



z/Architecture

解説書

SA88-8773-00
(英文原典 : SA22-7832-01)



z/Architecture

解説書

SA88-8773-00
(英文原典 : SA22-7832-01)

お願い:

本書および本書で紹介する製品をご使用になる前に、xviiページの『特記事項』に記載されている情報をお読みください。

ソフトコピーに関するご注意:

本書には、肩文字その他の多くの記号が含まれていますが、これらの記号はお使いの特定のハードウェアまたはソフトウェアで正しく表示されない場合もあります。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができます。

本書において、日本では発表されていないIBM製品（機械およびプログラム）、プログラミング、およびサービスについて言及または説明する場合があります。しかし、このことは、IBMがこのようなIBM製品、プログラミング、およびサービスを、必ずしも日本で発表する意図であることを示すものではありません。

本マニュアルに関するご意見やご感想は、次のURLからお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本IBM発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URLは、変更になる場合があります)

原 典： SA22-7832-01
z/Architecture
Principles of Operation
発 行： 日本アイ・ビー・エム株式会社
担 当： ナショナル・ランゲージ・サポート

第1刷 2004.3

©Copyright International Business Machines Corporation 1990-2001. All rights reserved.

©Copyright IBM Japan 2004

目次

特記事項	xvii	入出力装置と制御装置	2-7
商標	xvii	オペレーター機能	2-7
まえがき	xix	第3章 記憶機構	3-1
サイズと数値の表記法	xx	記憶機構のアドレッシング	3-2
バイト、文字、およびコード	xxi	情報のフォーマット	3-2
他の関連資料	xxi	規定境界	3-3
第 2 版での変更の要約 (英語版に適用)	xxi	アドレスのタイプとフォーマット	3-3
アドレスのタイプ		アドレスのタイプ	3-3
第1章 概説	1-1	絶対アドレス	3-3
z/Architecture の特長	1-1	実アドレス	3-4
64 ビット整数用の汎用レジスター	1-2	仮想アドレス	3-4
その他の新しい一般命令	1-2	1 次仮想アドレス	3-4
浮動小数点命令	1-4	2 次仮想アドレス	3-4
制御命令	1-4	AR 指定仮想アドレス	3-5
3 モード・アドレッシング	1-5	ホーム仮想アドレス	3-5
モーダル命令	1-5	論理アドレス	3-5
汎用レジスターのビット 0-31 への影響	1-5	命令アドレス	3-5
拡張変換機能 2	1-5	有効アドレス	3-5
入出力	1-6	アドレスのサイズと循環	3-5
アドレス循環		アドレス循環	3-6
ESA/390 のベース	1-7	記憶キー	3-8
ESA/370 および 370-XA のベース	1-13	保護	3-8
システム・プログラム	1-14	キー制御保護	3-8
互換性	1-14	記憶保護オーバーライド制御	3-9
z/Architecture システム間の互換性	1-14	取り出し保護オーバーライド制御	3-10
z/Architecture と ESA/390 の間の互換性	1-15	アクセス・リスト制御保護	3-10
制御プログラムの互換性	1-15	ページ保護	3-11
問題プログラム状態の互換性	1-15	低アドレス保護	3-11
可用性	1-15	保護時の抑止	3-12
第2章 編成	2-1	参照記録	3-13
主記憶機構	2-2	変更記録	3-13
拡張記憶機構	2-2	プレフィックシング	3-14
CPU	2-2	アドレス・スペース	3-15
PSW	2-3	異なるアドレス・スペースへの変更	3-16
汎用レジスター	2-3	アドレス・スペース番号	3-16
浮動小数点レジスター	2-3	ASN 変換	3-17
浮動小数点制御レジスター	2-4	ASN 変換制御	3-17
制御レジスター	2-4	制御レジスター 14	3-17
アクセス・レジスター	2-4	ASN 変換テーブル	3-18
暗号化機構	2-6	ASN 第 1 テーブル・エントリー	3-18
外部時刻参照	2-6	ASN 第 2 テーブル・エントリー	3-18
入出力	2-6	ASN 変換プロセス	3-20
チャンネル・サブシステム	2-6	ASN 第 1 テーブル索引	3-21
チャンネル・パス	2-6	ASN 第 2 テーブル索引	3-22

ASN 変換時の例外の認識	3-22	トレース	4-10
ASN 許可	3-22	制御レジスターの割り振り	4-13
ASN 許可制御	3-22	トレース・エントリー	4-13
制御レジスター 4	3-22	操作	4-23
ASN 第 2 テーブル・エントリー	3-22	プログラム・イベント記録	4-24
権限テーブル・エントリー	3-23	制御レジスター割り振りおよびアドレス・スペース制御エレメント	4-24
ASN 許可プロセス	3-23	操作	4-25
権限テーブルの索引	3-24	原因の識別	4-26
ASN 許可時の例外の認識	3-25	標識の優先順位	4-28
動的アドレス変換	3-25	記憶域の指示	4-29
変換制御	3-27	PER イベント	4-30
変換モード	3-27	ブランチ成功	4-30
制御レジスター 0	3-27	命令取り出し	4-31
制御レジスター 1	3-27	記憶域変更	4-31
制御レジスター 7	3-29	実アドレスを使用した格納	4-31
制御レジスター 13	3-29	他の割り込み条件と同時発生 of PER イベントの指示	4-32
変換テーブル	3-30	タイミング	4-33
領域テーブル・エントリー	3-30	TOD クロック	4-34
セグメント・テーブル・エントリー	3-31	フォーマット	4-34
ページ・テーブル・エントリー	3-32	状態	4-34
変換プロセス	3-33	クロック状態の変化	4-35
実スペース制御の検査	3-38	クロックの設定と検査	4-35
指定タイプ制御の検査	3-38	TOD プログラマブル・レジスター	4-36
アドレス・スペース制御エレメントにより指定されるテーブル内での検索	3-38	TOD クロックの同期化	4-38
領域テーブル・エントリーにより指定されるテーブル内での検索	3-39	クロック・コンパレーター	4-38
ページ・テーブルの検索	3-40	CPU タイマー	4-39
実アドレスの形成	3-41	外部開始機能	4-40
変換中の例外の認識	3-41	リセット	4-40
変換索引バッファ	3-41	CPU リセット	4-43
TLB の構造	3-42	初期 CPU リセット	4-44
TLB エントリーの形成	3-42	サブシステム・リセット	4-44
TLB エントリーの使用	3-43	クリア・リセット	4-45
変換テーブルの変更	3-44	パワーオン・リセット	4-45
アドレスの要約	3-46	初期プログラム・ローディング	4-46
変換されるアドレス	3-46	状況記憶	4-46
アドレスの取り扱い	3-47	マルチプロセッシング	4-47
割り当て済みの記憶位置	3-49	共用主記憶機構	4-47
第4章 制御	4-1	CPU アドレスの識別	4-47
停止、作動、ロード、およびチェック停止の各状態	4-1	CPU の信号発信と応答	4-48
停止状態	4-2	プロセッサ信号指令	4-48
作動状態	4-2	応答を決定する条件	4-51
ロード状態	4-2	指令コードの解釈を妨げる条件	4-51
チェック停止状態	4-2	状況ビット	4-52
プログラム状況ワード	4-3	第5章 プログラムの実行	5-1
プログラム状況ワードのフォーマット	4-5	命令	5-2
制御レジスター	4-7	オペランド	5-3

命令のフォーマット	5-3	リンケージ・テーブル・エントリー	5-28
レジスター・オペランド	5-6	エントリー・テーブル・エントリー	5-29
即値オペランド	5-6	PC 番号変換プロセス	5-31
記憶域オペランド	5-6	リンケージ・テーブル指定の取得	5-32
アドレスの生成	5-7	リンケージ・テーブル索引	5-32
3 モード・アドレッシング	5-7	エントリー・テーブル索引	5-32
命令アドレスの順次生成	5-7	PC 番号変換時の例外認識	5-33
オペランド・アドレスの生成	5-8	ホーム・アドレス・スペース	5-33
中間値の形成	5-8	アクセス・レジスターの概要	5-34
オペランド・アドレスの形成	5-8	要約	5-34
ブランチ・アドレスの生成	5-9	アクセス・レジスターの機能	5-34
中間値の形成	5-9	アクセス・レジスター指定のアドレス・ス	
ブランチ・アドレスの形成	5-9	ペース	5-34
命令の実行と順序付け	5-10	アクセス・レジスター命令	5-41
判断処理	5-10	アクセス・レジスターの変換	5-42
ループ制御	5-10	アクセス・レジスター変換制御	5-42
リンケージ・スタックなしのサブルーチン・リ		制御レジスター 2	5-42
ンケージ	5-10	制御レジスター 5	5-42
単純ブランチ命令	5-10	制御レジスター 8	5-43
その他のリンケージ命令	5-13	アクセス・レジスター	5-43
割り込み	5-19	アクセス・レジスター変換テーブル	5-44
命令終了のタイプ	5-19	ディスパッチ可能単位制御テーブルおよび	
完了	5-19	アクセス・リスト指定	5-44
抑止	5-20	アクセス・リスト・エントリー	5-45
無効化	5-20	ASN 第 2 テーブル・エントリー	5-46
中止	5-20	アクセス・レジスター変換プロセス	5-48
割り込み可能命令	5-20	アクセス・リスト・エントリー・トークン	
割り込み点	5-20	の選択	5-50
操作単位	5-20	1 次または 2 次アドレス・スペース制御エ	
割り込み可能命令の実行	5-20	レメントの取得	5-51
割り込み可能命令の代わりにの条件コード	5-21	ALET の第 1 バイトの検査	5-51
無効化と抑止に対する例外	5-22	有効アクセス・リスト指定の取得	5-51
DAT 関連アクセス例外の場合の記憶域の変		アクセス・リストの検索	5-51
更と復元	5-22	ASN 第 2 テーブル・エントリーの検索	5-52
DAT テーブル・エントリーの変更	5-23	アクセス・リスト・エントリーの使用許可	5-52
編集命令および変換命令の試行実行	5-23	アクセス・リスト制御保護の検査	5-52
許可メカニズム	5-23	2 次 ASN 第 2 テーブル・エントリーか	
モードの要件	5-24	らのアドレス・スペース制御エレメントの	
抽出権限制御	5-24	取得	5-52
PSW キー・マスク	5-24	アクセス・レジスター変換時の例外の認識	5-53
2 次スペース制御	5-24	ART 索引バッファ	5-53
サブシステム・リンケージ制御	5-25	ALB の構造	5-53
ASN 変換制御	5-25	ALB エントリーの形成	5-53
許可指標	5-25	ALB エントリーの使用	5-54
PC 番号の変換	5-28	ART テーブルの変更	5-54
PC 番号変換制御	5-28	サブスペース・グループ	5-55
制御レジスター 5	5-28	サブスペース・グループ・テーブル	5-55
PC 番号変換テーブル	5-28	サブスペース・グループ・ディスパッチ可	
		能単位制御テーブル	5-55

サブスペース・グループ ASN 第 2 テーブル・エントリー	5-57	記憶域オペランド格納参照	5-84
サブスペース置き換え操作	5-58	記憶域オペランド更新参照	5-85
リンケージ・スタックの概要	5-59	記憶域オペランドの一貫性	5-86
要約	5-59	単一アクセス参照	5-86
リンケージ・スタックの機能	5-60	複数アクセス参照	5-86
プログラム制御の転送	5-60	ブロック単位の参照	5-87
リンケージ・スタックを使用したブランチ	5-62	一貫性の指定	5-87
情報の追加と検索	5-62	オペランド・アクセス間の関係	5-88
許可のテスト	5-63	その他の記憶域参照	5-89
プログラム問題の分析	5-63	逐次化	5-89
リンケージ・スタック・エントリー・テーブル・エントリー	5-64	CPU の逐次化	5-89
リンケージ・スタックの操作	5-65	チャンネル・プログラムの逐次化	5-90
リンケージ・スタック操作制御	5-67	第6章 割り込み	6-1
制御レジスター 15	5-67	割り込みアクション	6-2
リンケージ・スタック	5-67	割り込みコード	6-5
エントリー記述子	5-67	使用可能と使用不可	6-6
ヘッダー・エントリー	5-68	浮動割り込み条件の取り扱い	6-6
トレーラー・エントリー	5-69	命令長コード	6-7
状態エントリー	5-69	ILC が 0 の場合	6-7
スタッキング・プロセス	5-72	命令取り出し時の例外の際の ILC	6-7
新規エントリー用のスペースの位置決め	5-72	PSW に関連した例外	6-9
新規エントリーの形成	5-73	早期例外認識	6-9
現エントリーの更新	5-74	遅延例外認識	6-9
制御レジスター 15 の更新	5-74	外部割り込み	6-10
スタッキング・プロセスでの例外の認識	5-75	クロック・コンパレーター	6-10
アンスタッキング・プロセス	5-75	CPU タイマー	6-11
現エントリーの位置決めとヘッダー・エントリーの処理	5-75	緊急信号	6-11
状態エントリーかどうかの検査	5-76	ETR	6-11
情報の復元	5-76	外部呼び出し	6-11
先行エントリーの更新	5-77	割り込みキー	6-12
制御レジスター 15 の更新	5-77	誤動作警報	6-12
アンスタッキング・プロセスでの例外の認識	5-77	サービス信号	6-12
記憶機構参照の順序	5-77	入出力割り込み	6-12
概念上の順序	5-78	マシン・チェック割り込み	6-13
命令実行における操作のオーバーラップ	5-78	プログラム割り込み	6-13
分割可能な命令実行	5-78	データ例外コード (DXC)	6-14
仮想記憶域参照のためのインターロック	5-79	データ例外の場合のプログラム割り込みの優先順位	6-14
命令間のインターロック	5-79	プログラム割り込みの条件	6-15
単一命令内のインターロック	5-80	アドレッシング例外	6-15
命令取り出し	5-81	AFX 変換例外	6-18
ART テーブルおよび DAT テーブルの取り出し	5-82	ALEN 変換例外	6-18
記憶キーのアクセス	5-83	ALE シーケンス例外	6-18
記憶域オペランドの参照	5-84	ALET 指定例外	6-18
記憶域オペランド取り出し参照	5-84	ASCE タイプ例外	6-18
		ASTE シーケンス例外	6-19
		ASTE 妥当性例外	6-19
		ASX 変換例外	6-20

暗号操作例外	6-20	第7章 一般命令	7-1
データ例外	6-20	データ・フォーマット	7-2
10 進除算例外	6-21	2 進整数の表現	7-3
10 進オーバーフロー例外	6-21	2 進数演算	7-3
実行例外	6-21	符号付き 2 進数演算	7-4
EX 変換例外	6-21	加算および減算	7-4
拡張権限例外	6-21	固定小数点オーバーフロー	7-4
固定小数点除算例外	6-22	符号なし 2 進数演算	7-4
固定小数点オーバーフロー例外	6-22	符号付き比較と論理比較	7-5
HFP 除算例外	6-22	命令	7-5
HFP 指数オーバーフロー例外	6-22	ADD	7-16
HFP 指数アンダーフロー例外	6-22	ADD HALFWORD	7-16
HFP 有効数字例外	6-23	ADD HALFWORD IMMEDIATE	7-16
HFP 平方根例外	6-23	ADD LOGICAL	7-17
LX 変換例外	6-23	ADD LOGICAL WITH CARRY	7-17
モニター・イベント	6-23	AND	7-18
オペランド例外	6-24	AND IMMEDIATE	7-19
操作例外	6-24	BRANCH AND LINK	7-20
ページ変換例外	6-25	BRANCH AND SAVE	7-20
PC 変換指定例外	6-25	BRANCH AND SAVE AND SET MODE	7-21
PER イベント	6-25	BRANCH AND SET MODE	7-22
1 次権限例外	6-26	BRANCH ON CONDITION	7-23
特権命令例外	6-26	BRANCH ON COUNT	7-24
記憶保護例外	6-27	BRANCH ON INDEX HIGH	7-25
領域第 1 変換例外	6-27	BRANCH ON INDEX LOW OR EQUAL	7-25
領域第 2 変換例外	6-28	BRANCH RELATIVE AND SAVE	7-26
領域第 3 変換例外	6-28	BRANCH RELATIVE AND SAVE LONG	7-26
2 次権限例外	6-29	BRANCH RELATIVE ON CONDITION	7-26
セグメント変換例外	6-29	BRANCH RELATIVE ON CONDITION	
スペース切り替えイベント	6-29	LONG	7-26
特殊操作例外	6-30	BRANCH RELATIVE ON COUNT	7-27
指定例外	6-31	BRANCH RELATIVE ON INDEX HIGH	7-28
スタック空例外	6-33	BRANCH RELATIVE ON INDEX LOW OR	
スタック一杯例外	6-33	EQUAL	7-28
スタック操作例外	6-33	CHECKSUM	7-29
スタック指定例外	6-33	COMPARE	7-33
スタック・タイプ例外	6-33	COMPARE AND FORM CODEWORD	7-33
トレース・テーブル例外	6-34	COMPARE AND SWAP	7-40
変換指定例外	6-34	COMPARE DOUBLE AND SWAP	7-40
プログラム割り込みの集合名	6-35	COMPARE HALFWORD	7-42
アクセス例外の認識	6-35	COMPARE HALFWORD IMMEDIATE	7-42
複数のプログラム割り込み条件	6-37	COMPARE LOGICAL	7-42
アクセス例外	6-40	COMPARE LOGICAL CHARACTERS	
ASN 変換例外	6-43	UNDER MASK	7-43
サブスペース置き換え例外	6-43	COMPARE LOGICAL LONG	7-44
トレース例外	6-44	COMPARE LOGICAL LONG EXTENDED	7-46
再始動割り込み	6-44	COMPARE LOGICAL LONG UNICODE	7-50
監視プログラム呼び出し割り込み	6-44	COMPARE LOGICAL STRING	7-53
割り込みの優先順位	6-44		

COMPARE UNTIL SUBSTRING EQUAL	7-54	MULTIPLY	7-105
COMPRESSION CALL	7-57	MULTIPLY HALFWORD	7-106
CONVERT TO BINARY	7-68	MULTIPLY HALFWORD IMMEDIATE	7-106
CONVERT TO DECIMAL	7-68	MULTIPLY LOGICAL	7-106
CONVERT UNICODE TO UTF-8	7-69	MULTIPLY SINGLE	7-107
CONVERT UTF-8 TO UNICODE	7-72	OR	7-108
COPY ACCESS	7-75	OR IMMEDIATE	7-109
DIVIDE	7-75	PACK	7-109
DIVIDE LOGICAL	7-75	PACK ASCII	7-110
DIVIDE SINGLE	7-76	PACK UNICODE	7-111
EXCLUSIVE OR	7-77	PERFORM LOCKED OPERATION	7-112
EXECUTE	7-78	ROTATE LEFT SINGLE LOGICAL	7-127
EXTRACT ACCESS	7-79	SEARCH STRING	7-128
EXTRACT PSW	7-79	SET ACCESS	7-129
INSERT CHARACTER	7-79	SET ADDRESSING MODE	7-129
INSERT CHARACTERS UNDER MASK	7-79	SET PROGRAM MASK	7-130
INSERT IMMEDIATE	7-80	SHIFT LEFT DOUBLE	7-130
INSERT PROGRAM MASK	7-81	SHIFT LEFT DOUBLE LOGICAL	7-131
LOAD	7-81	SHIFT LEFT SINGLE	7-132
LOAD ACCESS MULTIPLE	7-81	SHIFT LEFT SINGLE LOGICAL	7-132
LOAD ADDRESS	7-82	SHIFT RIGHT DOUBLE	7-133
LOAD ADDRESS EXTENDED	7-82	SHIFT RIGHT DOUBLE LOGICAL	7-133
LOAD ADDRESS RELATIVE LONG	7-83	SHIFT RIGHT SINGLE	7-134
LOAD AND TEST	7-83	SHIFT RIGHT SINGLE LOGICAL	7-135
LOAD COMPLEMENT	7-84	STORE	7-135
LOAD HALFWORD	7-84	STORE ACCESS MULTIPLE	7-135
LOAD HALFWORD IMMEDIATE	7-85	STORE CHARACTER	7-136
LOAD LOGICAL	7-85	STORE CHARACTERS UNDER MASK	7-136
LOAD LOGICAL CHARACTER	7-85	STORE CLOCK	7-137
LOAD LOGICAL HALFWORD	7-85	STORE CLOCK EXTENDED	7-137
LOAD LOGICAL IMMEDIATE	7-86	STORE HALFWORD	7-139
LOAD LOGICAL THIRTY ONE BITS	7-86	STORE MULTIPLE	7-139
LOAD MULTIPLE	7-86	STORE MULTIPLE HIGH	7-140
LOAD MULTIPLE DISJOINT	7-87	STORE PAIR TO QUADWORD	7-140
LOAD MULTIPLE HIGH	7-87	STORE REVERSED	7-140
LOAD NEGATIVE	7-88	SUBTRACT	7-141
LOAD PAIR FROM QUADWORD	7-88	SUBTRACT HALFWORD	7-142
LOAD POSITIVE	7-88	SUBTRACT LOGICAL	7-142
LOAD REVERSED	7-89	SUBTRACT LOGICAL WITH BORROW	7-143
MONITOR CALL	7-90	SUPERVISOR CALL	7-144
MOVE	7-91	TEST ADDRESSING MODE	7-144
MOVE INVERSE	7-91	TEST AND SET	7-144
MOVE LONG	7-92	TEST UNDER MASK (TEST UNDER MASK HIGH, TEST UNDER MASK LOW)	7-145
MOVE LONG EXTENDED	7-95	TRANSLATE	7-146
MOVE LONG UNICODE	7-99	TRANSLATE AND TEST	7-147
MOVE NUMERICS	7-102	TRANSLATE EXTENDED	7-148
MOVE STRING	7-103	TRANSLATE ONE TO ONE	7-150
MOVE WITH OFFSET	7-104		
MOVE ZONES	7-104		

TRANSLATE ONE TO TWO	7-150	すべての浮動小数点命令の要約	9-12
TRANSLATE TWO TO ONE	7-150		
TRANSLATE TWO TO TWO	7-150		
UNPACK	7-155		
UNPACK ASCII	7-155		
UNPACK UNICODE	7-156		
UPDATE TREE	7-157		
第8章 10 進数命令	8-1	第10章 制御命令	10-1
10 進数のフォーマット	8-1	BRANCH AND SET AUTHORITY	10-5
ゾーン・フォーマット	8-1	BRANCH AND STACK	10-9
パック・フォーマット	8-1	BRANCH IN SUBSPACE GROUP	10-12
10 進コード	8-2	COMPARE AND SWAP AND PURGE	10-17
10 進数操作	8-2	DIAGNOSE	10-18
10 進数演算命令	8-2	EXTRACT AND SET EXTENDED	
編集命令	8-3	AUTHORITY	10-19
10 進数命令の実行	8-3	EXTRACT PRIMARY ASN	10-19
10 進オペランドを対象とするその他の命令	8-3	EXTRACT SECONDARY ASN	10-20
10 進オペランド・データ例外	8-4	EXTRACT STACKED REGISTERS	10-20
命令	8-4	EXTRACT STACKED STATE	10-22
ADD DECIMAL	8-5	INSERT ADDRESS SPACE CONTROL	10-24
COMPARE DECIMAL	8-6	INSERT PSW KEY	10-25
DIVIDE DECIMAL	8-6	INSERT STORAGE KEY EXTENDED	10-25
EDIT	8-7	INSERT VIRTUAL STORAGE KEY	10-26
EDIT AND MARK	8-9	INVALIDATE PAGE TABLE ENTRY	10-27
MULTIPLY DECIMAL	8-11	LOAD ADDRESS SPACE PARAMETERS	10-28
SHIFT AND ROUND DECIMAL	8-11	LOAD CONTROL	10-37
SUBTRACT DECIMAL	8-12	LOAD PSW	10-37
TEST DECIMAL	8-12	LOAD PSW EXTENDED	10-38
ZERO AND ADD	8-13	LOAD REAL ADDRESS	10-39
		LOAD USING REAL ADDRESS	10-44
		MODIFY STACKED STATE	10-44
第9章 浮動小数点の概要とサポート命令	9-1	MOVE PAGE	10-46
レジスターと制御	9-2	MOVE TO PRIMARY	10-48
浮動小数点レジスター	9-2	MOVE TO SECONDARY	10-48
追加浮動小数点 (AFP) レジスター	9-2	MOVE WITH DESTINATION KEY	10-50
有効な浮動小数点レジスター指定	9-2	MOVE WITH KEY	10-50
浮動小数点制御 (FPC) レジスター	9-2	MOVE WITH SOURCE KEY	10-52
AFP レジスター制御ビット	9-2	PAGE IN	10-53
明示的な丸め方式	9-3	PAGE OUT	10-54
丸めアクションの要約	9-3	PROGRAM CALL	10-55
BFP と HFP の数表現の比較	9-4	PROGRAM RETURN	10-68
BFP と HFP の数の範囲	9-4	PROGRAM TRANSFER	10-72
BFP と HFP の間の等価の数表現	9-4	PURGE ALB	10-78
命令	9-6	PURGE TLB	10-78
CONVERT BFP TO HFP	9-8	RESET REFERENCE BIT EXTENDED	10-78
CONVERT HFP TO BFP	9-9	RESUME PROGRAM	10-79
LOAD	9-11	SET ADDRESS SPACE CONTROL FAST	10-81
LOAD ZERO	9-11	SET CLOCK	10-83
STORE	9-11	SET CLOCK COMPARATOR	10-84
		SET CLOCK PROGRAMMABLE FIELD	10-84
		SET CPU TIMER	10-84
		SET PREFIX	10-85
		SET PSW KEY FROM ADDRESS	10-85
		SET SECONDARY ASN	10-86

SET STORAGE KEY EXTENDED	10-89	システム損傷	11-15
SET SYSTEM MASK	10-89	命令処理損傷	11-15
SIGNAL PROCESSOR	10-89	システム・リカバリー	11-16
STORE CLOCK COMPARATOR	10-91	タイミング機構損傷	11-16
STORE CONTROL	10-91	外部損傷	11-16
STORE CPU ADDRESS	10-91	性能低下	11-17
STORE CPU ID	10-92	警告	11-17
STORE CPU TIMER	10-93	チャンネル報告保留	11-17
STORE FACILITY LIST	10-93	サービス・プロセッサ損傷	11-17
STORE PREFIX	10-93	チャンネル・サブシステム損傷	11-17
STORE REAL ADDRESS	10-94	サブクラス修飾子	11-17
STORE SYSTEM INFORMATION	10-95	バックアップ	11-18
STORE THEN AND SYSTEM MASK	10-105	遅延アクセス例外	11-18
STORE THEN OR SYSTEM MASK	10-105	補助報告	11-18
STORE USING REAL ADDRESS	10-105	同期マシン・チェック割り込み条件	11-18
TEST ACCESS	10-106	処理バックアップ	11-18
TEST BLOCK	10-108	処理損傷	11-19
TEST PROTECTION	10-110	記憶機構エラー	11-19
TRACE	10-113	未訂正の記憶機構エラー	11-19
TRAP	10-114	訂正済みの記憶機構エラー	11-19
		未訂正の記憶キー・エラー	11-19
		記憶機構性能低下	11-19
		間接記憶機構エラー	11-20
第11章 マシン・チェックの取り扱い	11-1	マシン・チェック割り込みコードの妥当性	11-20
マシン・チェックの検出	11-2	ット	11-20
マシン誤動作の訂正	11-2	PSW-MWP の妥当性	11-20
エラー検査/訂正	11-2	PSW のマスクとキーの妥当性	11-20
CPU 再試行	11-2	PSW プログラム・マスクと条件コードの妥当性	11-21
CPU 再試行の影響	11-3	PSW 命令アドレスの妥当性	11-21
チェックポイント同期化	11-3	故障記憶機構アドレスの妥当性	11-21
チェックポイント同期化の際のマシン・チェックの取り扱い	11-3	外部損傷コードの妥当性	11-21
チェックポイント同期化操作	11-3	浮動小数点レジスターの妥当性	11-21
チェックポイント同期化アクション	11-4	汎用レジスターの妥当性	11-21
チャンネル・サブシステム・リカバリー	11-4	制御レジスターの妥当性	11-21
ユニット削除	11-4	記憶機構の論理的妥当性	11-21
マシン・チェックの取り扱い	11-4	アクセス・レジスターの妥当性	11-21
有効化	11-5	TOD プログラマブル・レジスターの妥当性	11-21
記憶機構内の無効 CBC	11-6	浮動小数点レジスターの妥当性	11-21
プログラムによる記憶機構の有効化	11-6	CPU タイマーの妥当性	11-21
記憶キー内の無効 CBC	11-7	クロック・コンパレーターの妥当性	11-21
レジスター内の無効 CBC	11-9	マシン・チェック拡張割り込み情報	11-22
チェック停止状態	11-10	レジスター保管域	11-22
システム・チェック停止	11-11	外部損傷コード	11-22
マシン・チェック割り込み	11-11	故障記憶機構アドレス	11-22
緊急条件	11-11	マシン・チェック条件の取り扱い	11-23
抑制可能条件	11-11	浮動割り込み条件	11-23
割り込みアクション	11-12	浮動マシン・チェック割り込み条件	11-23
割り込み点	11-13		
マシン・チェック割り込みコード	11-14		
サブクラス	11-15		

浮動入出力割り込み	11-23	装置番号	13-5
マシン・チェック・マスキング	11-23	装置 ID	13-5
チャンネル報告保留サブクラス・マスク	11-24	入出力操作の実行	13-5
リカバリー・サブクラス・マスク	11-24	開始機能の開始	13-6
性能低下サブクラス・マスク	11-24	パス管理	13-6
外部損傷サブクラス・マスク	11-24	チャンネル・プログラムの実行	13-6
警告サブクラス・マスク	11-24	入出力操作の終結	13-7
マシン・チェック・ログアウト	11-24	入出力割り込み	13-8
マシン・チェックのマスキングの要約	11-24		
第12章 オペレーター機能	12-1	第14章 入出力命令	14-1
手動操作	12-1	入出力命令のフォーマット	14-1
基本オペレーター機能	12-1	入出力命令の実行	14-1
アドレス比較制御	12-1	逐次化	14-1
変更/表示制御	12-2	オペランドのアクセス	14-1
アーキテクチャー・モード標識ライト	12-2	条件コード	14-2
アーキテクチャー・モード選択制御	12-2	プログラム例外	14-2
チェック停止標識ライト	12-3	命令	14-2
IML 制御	12-3	CANCEL SUBCHANNEL	14-4
割り込みキー	12-3	CLEAR SUBCHANNEL	14-4
ロード標識ライト	12-3	HALT SUBCHANNEL	14-6
ロード消去キー	12-3	MODIFY SUBCHANNEL	14-7
ロード通常キー	12-3	RESET CHANNEL PATH	14-8
ロード装置アドレス制御	12-3	RESUME SUBCHANNEL	14-10
手動標識ライト	12-3	SET ADDRESS LIMIT	14-11
電源制御	12-4	SET CHANNEL MONITOR	14-12
速度制御	12-4	START SUBCHANNEL	14-14
再始動キー	12-4	STORE CHANNEL PATH STATUS	14-15
開始キー	12-4	STORE CHANNEL REPORT WORD	14-16
停止キー	12-4	STORE SUBCHANNEL	14-17
状況記憶キー	12-5	TEST PENDING INTERRUPTION	14-17
システム・リセット消去キー	12-5	TEST SUBCHANNEL	14-19
システム・リセット通常キー	12-5		
テスト標識ライト	12-5	第15章 基本入出力機能	15-1
TOD クロック制御	12-5	基本入出力機能の制御	15-1
待ち状態ライト	12-6	サブチャンネル情報ブロック	15-1
マルチプロセッシング構成	12-6	パス管理制御ワード	
		(Path-Management-Control Word)	15-2
		サブチャンネル状況ワード	
		(Subchannel-Status Word)	15-7
		モデル依存区域 (Model-Dependent Area)	15-7
		変更可能フィールドの要約	15-7
第13章 入出力の概要	13-1	チャンネル・パス従属関係	15-10
入出力 (I/O)	13-1	作業従属関係	15-10
チャンネル・サブシステム	13-1	アクティブ従属関係	15-11
サブチャンネル	13-2	専用従属関係	15-11
入出力装置の接続	13-2	チャンネル・パスの可用性	15-11
チャンネル・パス	13-2	制御装置タイプ	15-12
制御装置	13-3	消去機能	15-13
入出力装置	13-4	消去機能: パス管理	15-13
入出力アドレッシング	13-4		
チャンネル・パス ID	13-4		
サブチャンネル番号	13-5		

消去機能: サブチャンネル変更	15-13	初期状況割り込み制御 (I)	16-11
消去機能: 信号発信と完了	15-13	アドレス限界検査制御 (A)	16-11
停止機能	15-14	延期割り込み抑止 (U)	16-11
停止機能: パス管理	15-14	サブチャンネル制御フィールド	16-11
停止機能: 信号発信と完了	15-15	0 条件コード (Z)	16-11
開始機能と再開機能	15-17	拡張制御 (E)	16-11
開始機能と再開機能: パス管理	15-17	パス非稼働 (N)	16-11
入出力操作の実行	15-19	機能制御 (FC)	16-12
データのブロック化	15-20	活動制御 (AC)	16-13
操作要求ブロック	15-20	状況制御 (SC)	16-16
チャンネル・コマンド・ワード	15-25	CCW アドレス・フィールド	16-18
コマンド・コード	15-27	装置状況フィールド	16-22
記憶域の指定	15-27	サブチャンネル状況フィールド	16-23
チェーニング	15-29	プログラム制御割り込み	16-23
データ・チェーニング	15-31	誤長	16-23
コマンド・チェーニング	15-32	プログラム・チェック	16-23
スキップ	15-33	記憶保護チェック	16-26
プログラム制御割り込み	15-33	チャンネル・データ・チェック	16-26
CCW 間接データ・アドレッシング	15-34	チャンネル制御チェック	16-27
チャンネル・プログラム実行の延期	15-36	インターフェース制御チェック	16-27
コマンドとフラグ	15-38	チェーニング・チェック	16-28
チャンネル・プログラム内でのブランチ	15-38	カウント・フィールド	16-28
チャンネル内転送	15-39	拡張状況ワード	16-31
コマンド再試行	15-39	拡張状況フォーマット 0	16-31
開始前の入出力操作の終結	15-39	サブチャンネル・ログアウト	
開始時の入出力操作の終結	15-40	(Subchannel Logout)	16-31
入出力操作の即時終結	15-40	拡張報告ワード (Extended-Report Word)	16-35
データ転送時の入出力操作の終結	15-41	故障記憶機構アドレス	
チャンネル・パス・リセット機能	15-42	(Failing-Storage Address)	16-36
チャンネル・パス・リセット機能の信号発信	15-42	2 次 CCW アドレス	
チャンネル・パス・リセット機能完了信号の発信	15-43	(Secondary-CCW Address)	16-36
第16章 入出力割り込み	16-1	拡張状況フォーマット 1	16-37
割り込み条件	16-2	拡張状況フォーマット 2	16-37
中間割り込み条件	16-4	拡張状況フォーマット 3	16-38
1 次割り込み条件	16-4	拡張制御ワード	16-39
2 次割り込み条件	16-4	第17章 入出力サポート機能	17-1
警報割り込み条件	16-4	チャンネル・サブシステム・モニター	17-1
割り込みの優先順位	16-4	チャンネル・サブシステム・タイミング	17-2
割り込みアクション	16-5	チャンネル・サブシステム・タイマー	17-2
割り込み応答ブロック	16-6	測定ブロック更新	17-2
サブチャンネル状況ワード	16-6	測定ブロック	17-3
サブチャンネル・キー	16-7	測定ブロック起点	17-5
延期制御 (S)	16-8	測定ブロック・キー	17-5
拡張状況ワード・フォーマット (L)	16-8	測定ブロック指標	17-5
据え置き条件コード (CC)	16-8	測定ブロック更新モード	17-6
フォーマット (F)	16-10	測定ブロック更新可能	17-6
事前取り出し (P)	16-10	制御装置キューイング測定	17-6
		制御装置据え置き時間	17-6

装置アクティブ測定	17-6	SQUARE ROOT	18-18
時間間隔測定精度	17-7	SUBTRACT NORMALIZED	18-19
装置接続時間測定	17-7	SUBTRACT UNNORMALIZED	18-20
装置接続時間測定モード	17-7		
装置接続時間測定可能	17-8	第19章 2 進浮動小数点命令	19-1
信号およびリセット	17-8	2 進浮動小数点機能	19-1
信号	17-8	浮動小数点制御 (FPC) レジスター	19-2
停止信号	17-8	IEEE マスクおよびフラグ	19-3
消去信号	17-8	FPC DXC バイト	19-3
リセット信号	17-9	FPC レジスターに対する操作	19-4
リセット	17-9	BFP 演算	19-4
チャンネル・パス・リセット	17-9	BFP データ・フォーマット	19-4
入出力システム・リセット	17-10	BFP 短精度フォーマット	19-4
外部開始機能	17-13	BFP 長精度フォーマット	19-4
初期プログラム・ローディング	17-13	BFP 拡張精度フォーマット	19-4
入出力システムの再構成	17-16	バイアス指数	19-5
状況検査	17-16	仮数	19-5
アドレス限界検査	17-16	非ゼロ数の値	19-5
構成警報	17-17	BFP データのクラス	19-6
誤長表示抑止	17-17	ゼロ	19-6
並行センス	17-17	非正規化数	19-6
チャンネル・サブシステム・リカバリー	17-17	正規化数	19-6
チャンネル報告	17-18	無限大	19-7
チャンネル報告ワード	17-19	信号付き NaN とクァイエット NaN	19-7
チャンネル・サブシステム入出力優先順位決定機能	17-21	BFP フォーマット変換	19-7
チャンネル・サブシステム優先順位レベルの 数	17-21	BFP の丸め	19-8
		丸めモード	19-8
		正規化と非正規化	19-8
第18章 16 進浮動小数点命令	18-1	BFP の比較	19-9
HFP 演算	18-1	BFP 命令の条件コード	19-9
HFP 数の表現	18-1	剰余	19-9
正規化	18-3	IEEE 例外条件	19-11
HFP データのフォーマット	18-3	IEEE 無効操作	19-12
命令	18-4	IEEE ゼロ除算	19-12
ADD NORMALIZED	18-7	IEEE オーバーフロー	19-12
ADD UNNORMALIZED	18-8	IEEE アンダーフロー	19-13
COMPARE	18-9	IEEE 不正確	19-13
CONVERT FROM FIXED	18-10	結果の表	19-14
CONVERT TO FIXED	18-10	データ例外コード (DXC) とその略語	19-15
DIVIDE	18-11	命令	19-16
HALVE	18-12	ADD	19-19
LOAD AND TEST	18-12	COMPARE	19-24
LOAD COMPLEMENT	18-13	COMPARE AND SIGNAL	19-25
LOAD FP INTEGER	18-13	CONVERT FROM FIXED	19-27
LOAD LENGTHENED	18-14	CONVERT TO FIXED	19-28
LOAD NEGATIVE	18-15	DIVIDE	19-30
LOAD POSITIVE	18-15	DIVIDE TO INTEGER	19-32
LOAD ROUNDED	18-16	EXTRACT FPC	19-37
MULTIPLY	18-16	LOAD AND TEST	19-37

LOAD COMPLEMENT	19-37	COMPARE AND FORM CODEWORD	
LOAD FP INTEGER	19-38	(CFC)	A-14
LOAD FPC	19-39	COMPARE HALFWORD (CH)	A-14
LOAD LENGTHENED	19-40	COMPARE LOGICAL (CL, CLC, CLI,	
LOAD NEGATIVE	19-40	CLR)	A-15
LOAD POSITIVE	19-41	CLC の例	A-15
LOAD ROUNDED	19-41	CLI の例	A-15
MULTIPLY	19-42	CLR の例	A-15
MULTIPLY AND ADD	19-44	COMPARE LOGICAL CHARACTERS	
MULTIPLY AND SUBTRACT	19-44	UNDER MASK (CLM)	A-16
SET FPC	19-46	COMPARE LOGICAL LONG (CLCL)	A-16
SET ROUNDING MODE	19-47	COMPARE LOGICAL STRING (CLST)	A-18
SQUARE ROOT	19-47	CONVERT TO BINARY (CVB)	A-18
STORE FPC	19-47	CONVERT TO DECIMAL (CVD)	A-19
SUBTRACT	19-48	DIVIDE (D, DR)	A-19
TEST DATA CLASS	19-48	EXCLUSIVE OR (X, XC, XI, XR)	A-19
		XC の例	A-20
		XI の例	A-20
付録A. 数の表現と命令の使用例	A-1	EXECUTE (EX)	A-21
数の表現	A-2	INSERT CHARACTERS UNDER MASK	
2 進整数	A-2	(ICM)	A-22
符号付き 2 進整数	A-2	LOAD (L, LR)	A-22
符号なし 2 進整数	A-3	LOAD ADDRESS (LA)	A-23
10 進整数	A-4	LOAD HALFWORD (LH)	A-23
16 進浮動小数点数	A-5	MOVE (MVC, MVI)	A-23
変換の例	A-6	MVC の例	A-23
命令の使用例	A-6	MVI の例	A-24
マシン・フォーマット	A-6	MOVE INVERSE (MVCIN)	A-24
アセンブラー言語フォーマット	A-7	MOVE LONG (MVCL)	A-25
例で使用されているアドレッシング・モード	A-7	MOVE NUMERICS (MVN)	A-26
一般命令	A-7	MOVE STRING (MVST)	A-26
ADD HALFWORD (AH)	A-7	MOVE WITH OFFSET (MVO)	A-26
AND (N, NC, NI, NR)	A-7	MOVE ZONES (MVZ)	A-27
NI の例	A-8	MULTIPLY (M, MR)	A-27
リンケージ命令 (BAL, BALR, BAS,		MULTIPLY HALFWORD (MH)	A-28
BASR, BASSM, BSM)	A-8	OR (O, OC, OI, OR)	A-28
BALR および BASR のその他の例	A-9	OI の例	A-28
BRANCH AND STACK (BAKR)	A-10	PACK (PACK)	A-28
BAKR の例 1	A-10	SEARCH STRING (SRST)	A-29
BAKR の例 2	A-11	SRST の例 1	A-29
BAKR の例 3	A-11	SRST の例 2	A-29
BRANCH ON CONDITION (BC, BCR)	A-11	SHIFT LEFT DOUBLE (SLDA)	A-29
BRANCH ON COUNT (BCT, BCTR)	A-12	SHIFT LEFT SINGLE (SLA)	A-30
BRANCH ON INDEX HIGH (BXH)	A-12	STORE CHARACTERS UNDER MASK	
BXH の例 1	A-12	(STCM)	A-30
BXH の例 2	A-13	STORE MULTIPLE (STM)	A-31
BRANCH ON INDEX LOW OR EQUAL		TEST UNDER MASK (TM)	A-31
(BXLE)	A-13	TRANSLATE (TR)	A-32
BXLE の例 1	A-13	TRANSLATE AND TEST (TRT)	A-32
BXLE の例 2	A-14		

UNPACK (UNPK)	A-34	即値 OR を用いたプログラム障害の例	A-43
UPDATE TREE (UPT)	A-34	条件付きスワップ命令 (CS、CDS)	A-43
10 進数命令	A-34	単一ビットの設定	A-44
ADD DECIMAL (AP)	A-34	カウンターの更新	A-45
COMPARE DECIMAL (CP)	A-34	POST と WAIT のバイパス	A-45
DIVIDE DECIMAL (DP)	A-35	POST バイパス・ルーチン	A-45
EDIT (ED)	A-35	WAIT バイパス・ルーチン	A-46
EDIT AND MARK (EDMK)	A-36	ロック/アンロック	A-46
MULTIPLY DECIMAL (MP)	A-37	競合に対する LIFO キューイングを使用し	
SHIFT AND ROUND DECIMAL (SRP)	A-37	たロック/アンロック	A-46
10 進数の左シフト	A-37	競合に対する FIFO キューイングを使用し	
10 進数の右シフト	A-37	たロック/アンロック	A-47
10 進数の右シフトと丸め	A-38	フリー・プールの操作	A-49
10 の可変累乗による乗算	A-38	PERFORM LOCKED OPERATION (PLO)	A-50
ZERO AND ADD (ZAP)	A-38	ソート命令	A-51
16 進浮動小数点命令	A-39	ツリー・フォーマット	A-51
ADD NORMALIZED (AD、ADR、AE、		ソート命令の使用例	A-53
AER、AXR)	A-39		
ADD UNNORMALIZED (AU、AUR、		付録B. 命令一覧表	B-1
AW、AWR)	A-39		
COMPARE (CD、CDR、CE、CER)	A-40	付録C. 条件コードの設定	C-1
DIVIDE (DD、DDR、DE、DER)	A-40		
HALVE (HDR、HER)	A-41	付録D. 2 の累乗数の表	D-1
MULTIPLY (MD、MDR、MDE、MDER、			
MXD、MXDR、MXR)	A-41	付録E. 16 進数表	E-1
16 進浮動小数点数の変換	A-42		
固定小数点から 16 進浮動小数点へ	A-42	付録F. EBCDIC およびその他のコード	F-1
16 進浮動小数点から固定小数点へ	A-42		
マルチプログラミングとマルチプロセッシングの		索引	X-1
例	A-43		

特記事項

本書に記載の製品、プログラム、またはサービスが日本においては提供されていない場合があります。日本で利用可能な製品、プログラム、またはサービスについては、日本アイ・ピー・エムの営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の動作の評価と検証はお客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。使用許諾については、下記の宛先に書面にてご照会ください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

商標

以下は、IBM Corporation の商標です。

AIX/ESA
BookMaster
CICS
DB2
Enterprise Systems Architecture/370
Enterprise Systems Architecture/390
Enterprise Systems Connection Architecture
ESA/370
ESA/390
ESCON
FICON
IBM
IBMLink
MVS/ESA
OS/390
Processor Resource/Systems Manager
PR/SM
Sysplex Timer
System/370
VM/ESA
z/Architecture
z/OS

まえがき

本書には、参照を目的として、z/Architecture についての詳細な情報を収めてあります。

本書は、z/Architecture による定義に従って稼働するシステムのみにも適用されます。Enterprise Systems Architecture/390 (ESA/390) の定義に従って稼働するシステムの場合は、「IBM ESA/390 Principles of Operation」(SA22-7201) を参照してください。

本書では、それぞれの機能を使用するアセンブラー言語プログラムを作成するために必要と認められるレベルで、それぞれの機能について詳しく説明します。ただし、このようなプログラムを作成するときに従う必要がある表記法や規則については解説していないので、該当のアセンブラー言語関連資料を参照してください。

本書の内容は、主としてアセンブラー言語プログラマーのためのものですが、z/Architecture の機能の詳細に関心をお持ちの方なら、どなたにも役立つものです。

本書は参照用の資料として作成されているため、入門書または学習書としての利用には適していません。また、本書は、ユーザーがデータ処理システムの基本知識を持っていることを前提として書かれています。

本書で説明する各機能は、必ずしもすべてのモデルで利用可能なものとは限りません。さらに、定義の中には、ある程度の拡張性を持たせる構造になっているものがあり、したがって、現実にはどのモデルにもない能力について記述したり言及したりしている場合もあります。このような能力としては、例えば、サブシステム ID ワード内で 16 ビット・フィールドを使用して、サブチャンネル番号、CPU アドレスのサイズ、および主記憶機構を共用する CPU の数を識別する能力などがあります。この種の拡張性が可能であるとしても、それは IBM にこの種の能力を提供する意図があることを暗黙に示すものではありません。特定モデルにおける各種機能の特性と可用性については、そのモデルに関する機能特性の資料を参照してください。

本書は参照用として編成されているため、当然のことながら、各種の語句についての主要説明箇所より前に、該当の語句が現れることがあります。このような編成になっているので、途中で意味の分からない語句が出てきたときは、索引でその語句に関する主要説明が出ている場所を調べてください。

本書に示す情報は、19 の章と幾つかの付録に分かれています。

第1章、『概説』では、z/Architecture の主な機能を簡単に紹介します。

第2章、『編成』では、システム内での大きなグループ分類 (主記憶機構、拡張記憶機構、中央演算処理装置 (CPU)、外部時刻参照 (ETR)、および入出力) について説明し、これらのグループの構成と特性を簡単に示します。

第3章、『記憶機構』では、情報のフォーマット、記憶機構のアドレッシング、および記憶域保護のための各種機能について説明します。また、動的アドレス変換 (DAT) にも触れています。DAT を特殊なプログラミング・サポートと一緒に使用することで、仮想記憶域が使用できるようになります。

第4章、『制御』では、システム状況の切り替え、外部で開始される特殊操作、デバッグ、およびタイミングのための機能について説明します。特に、CPU 状態、制御モード、プログラム状況ワード (PSW)、制御レジスタ、トレース、プログラム・イベント記録、タイミング機能、リセット、保管状況、および初期プログラム・ローディングを取り扱います。

第5章、『プログラムの実行』では、プログラム実行の中で命令の役割について説明し、命令フォーマットについての詳細を示し、そして、プログラム状況ワード (PSW)、ブランチ、および割り込みの使用方法について簡単に説明します。この章には、ESA/370 で導入された拡張アドレス・スペース機能に関する詳しい説明を収めてあります。さらに、1 つの CPU でのプログラム実行を、他の CPU およびチャンネル・プログラムから見た場合の性質についても、詳しく説明します。

第6章、『割り込み』では、システム外部、システム内部、または CPU 自体の内部条件の結果として、CPU の状態が変化するメカニズムについて、詳しく説明します。割り込みは、6 つのクラスに分類されて説明されています。それは、マシン・チェック割り込み、プログラム割り込み、監視プログラム呼び出し割り込み、外部割り込み、入出力割り込み、および再始動割り込みです。

第7章、『一般命令』では、論理データおよび 2 進整数データのフォーマット、および、10 進命令と浮動小数点命令を除くすべての非特権命令について、詳しく説明します。

第8章、『10進数命令』では、10進データ・フォーマットと10進数命令について、詳しく説明します。

第9章、『浮動小数点の概要とサポート命令』には、浮動小数点操作の概要、16進浮動小数点操作と2進浮動小数点操作の両方に共通する命令、および、すべての浮動小数点命令の要約を収めてあります。

第10章、『制御命令』では、入出力命令を除くすべての準特権命令および特権命令について、詳しく説明します。

第11章、『マシン・チェックの取り扱い』では、マシンの誤動作を検出、修正、および報告するメカニズムについて説明します。

第12章、『オペレーター機能』では、システムの操作と制御のために使用できる基本的な手動の機能および制御について説明します。

本書の第13章から第17章までには、チャンネル・サブシステムが実行する機能、およびCPUとチャンネル・サブシステムの間で論理インターフェースが実行する機能についての詳細な定義を収めてあります。

第13章、『入出力の概要』では、チャンネル・サブシステムの基本的なコンポーネントと操作について、簡単に説明します。

第14章、『入出力命令』では、入出力命令について説明します。

第15章、『基本入出力機能』では、チャンネル・サブシステムが実行する基本入出力機能（入出力操作の開始、制御、終結など）について説明します。

第16章、『入出力割り込み』では、入出力割り込みおよび割り込み条件について説明します。

第17章、『入出力サポート機能』では、チャンネル・サブシステム使用状況モニター、リセット、初期プログラム・ローディング、再構成、およびチャンネル・サブシステム・リカバリーなどの機能について説明します。

第18章、『16進浮動小数点命令』では、16進浮動小数点（HFP）データ・フォーマットおよびHFP命令について、詳しく説明します。

第19章、『2進浮動小数点命令』では、2進浮動小数点（BFP）データ・フォーマットおよびBFP命令について、詳しく説明します。

『付録』には、以下の情報を収めてあります。

- 数の表現に関する情報
- 命令の使用例
- 何種類かの順序で並べた命令のリスト
- 条件コード設定の要約
- 2の累乗数の表
- 16進数を取り扱うときに役立つ表
- EBCDIC およびその他のコードの表

サイズと数値の表記法

本書では、英字の K、M、G、T、P、E は、それぞれ、乗数 2^{10} 、 2^{20} 、 2^{30} 、 2^{40} 、 2^{50} 、 2^{60} を表します。これらの文字は10進法の表記を流用したもので、それぞれ kilo (キロ) (10^3)、mega (メガ) (10^6)、giga (ギガ) (10^9)、tera (テラ) (10^{12})、peta (ペタ) (10^{15})、および exa (エクサ) (10^{18}) を表していますが、10進数の意味ではなく、それぞれ対応する10の累乗数に最も近い2の累乗数を表します。本書におけるそれぞれの文字の意味は以下のとおりです。

シンボル	値
K (キロ)	$1,024 = 2^{10}$
M (メガ)	$1,048,576 = 2^{20}$
G (ギガ)	$1,073,741,824 = 2^{30}$
T (テラ)	$1,099,511,627,776 = 2^{40}$
P (ペタ)	$1,125,899,906,842,624 = 2^{50}$
E (エクサ)	$1,152,921,504,606,846,976 = 2^{60}$

以下に示すのは、K、M、G、T、E の使用例です。

2,048 は 2K と表されます。

4,096 は 4K と表されます。

65,536 は 64K と表されます (65K ではありません)。

2^{24} は 16M と表されます。

2^{31} は 2G と表されます。

2^{42} は 4T と表されます。

2^{64} は 16E と表されます。

「千」および「百万」という語を使用しているときは、特に2の累乗数を表してはいないものとします。

本書に示されている数値は、特に2進数または16進数 (hex) と断っていない限り、すべて10進数であるものとします。

バイト、文字、およびコード

System/360 アーキテクチャーは、本来は Extended Binary-Coded-Decimal Interchange Code (EBCDIC) をサポートするために設計されたものですが、このアーキテクチャーの命令およびデータ・フォーマットのほとんどは、マシンが処理すべき外部コードから独立しています。ほとんどの命令では、各ビット・パターンが実際にどの文字を表しているかに関係なく、特定の 1 バイトを表す 256 通りのビット・パターンの組み合わせをすべて処理できます。ゾーン・フォーマットを使用する命令、および特定の外部コードに依存するきわめて少数の命令については、TRANSLATE 命令を使用して、1 つのコードから別のコードへとデータを変換することができます。したがって、z/Architecture に従って稼働するマシンは、EBCDIC、ASCII のほか、1 文字を 8 ビット以下で表すことのできる他のすべてのコードを処理できます。

本書では、特に断らない限り、特定の 1 バイトに与えられる値は、そのバイトのビットがバイナリー・コードを表すときに得られる値です。したがって、あるバイトにゼロが含まれると見なされる場合、その値は、EBCDIC 文字の「0」（つまり 16 進数 F0）ではなく、2 進数 00000000、つまり 16 進数 00 を意味します。

他の関連資料

並列入出力インターフェースについては、「*IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information*」(GA22-6974) で説明されています。

並列入出力チャンネル間アダプターについては、「*IBM Enterprise Systems Architecture/390 Channel-to-Channel Adapter for the System/360 and System/370 I/O Interface*」(SA22-7091) で説明されています。

本書において、FICON 入出力インターフェースと共に直列入出力インターフェースとして参照されているエンタープライズ・システム接続アーキテクチャー (Enterprise Systems Connection Architecture (ESCON)) 入出力インターフェースについては、「*IBM Enterprise Systems Architecture/390 ESCON I/O Interface*」(SA22-7202) で説明されています。

FICON 入出力インターフェースについては、ANSI 規格文書「*Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)*」で説明されています。

直列入出力インターフェース用のチャンネル間アダプターについては、「*IBM Enterprise Systems Architecture/390 ESCON Channel-to-Channel Adapter*」(SA22-7203) で説明されています。

z/Architecture に準拠するすべての入出力装置に共通のコマンド、状況、およびセンス・データについては、「*IBM Enterprise Systems Architecture/390 Common I/O-Device Commands and Self Description*」(SA22-7204) で説明されています。

圧縮機能については、「*IBM Enterprise Systems Architecture/390 Data Compression*」(SA22-7208) で説明されています。この資料には、z/Architecture 形式の COMPRESSION CALL 命令に関する説明が含まれています。

解釈実行機能については、「*IBM 370-XA Interpretive Execution*」(SA22-7095) で説明されています。

第 2 版での変更の要約 (英語版に適用)

本書の第 2 版 (現行版) では、旧版に対して、主として明確化と修正を目的とする変更が加えられています。主な変更は以下のとおりです。

- 第 1 章、『概説』
 - DIVIDE LOGICAL と MULTIPLY LOGICAL、TEST ADDRESSING MODE、SIGNAL PROCESSOR のアーキテクチャー設定指令、および STORE FACILITY LIST の要約が、追加または改良されました。
 - z/Architecture における入出力機能の拡張に併せて、要約が追加されました。
- 第 3 章、『記憶機構』
 - 絶対位置 0-23 の定義は、ESA/390 の初期プログラム・ロードのみに関連するものなので、削除されました。
 - STORE FACILITY LIST により保管される実記憶位置 200-203 の定義は、ビット 16 が拡張変換機能 2 を示すことを指示するように訂正されました。
 - PSW 内の未割り当てフィールドの説明は、ビット 4 が未割り当てで、ビット 31 が割り当て済みであることを示すように訂正されました。
 - M1 フィールドについて、RSL フォーマットおよび RIL フォーマットが追加されました。

- 第7章、『一般命令』
 - BRANCH AND SET MODE の定義が訂正されて、24 ビットまたは 31 ビット・アドレッシング・モードでは、R1 汎用レジスタのビット 63 は未変更のままとなり、ゼロには設定されないことが示されています。
 - PACK ASCII、PACK UNICODE、UNPACK ASCII、および UNPACK UNICODE の定義が明確化されました。
 - 以下の命令が、それぞれの記憶域オペランドに対する複数アクセス参照を行うことが明確化されました。
 - CHECKSUM
 - COMPARE AND FORM CODEWORD
 - CONVERT UNICODE TO UTF-8
 - CONVERT UTF-8 TO UNICODE
 - MOVE LONG、MOVE LONG EXTENDED、および MOVE LONG UNICODE 命令が、必ずしも、他の CPU で見られるように記憶域オペランドを左から右への順序では処理しないことが、明確化されています。他の CPU に見られるように左から右へ処理を進めるかどうか、および移動しているデータを後続の処理のためにキャッシュに入れるかどうかは、MOVE LONG および MOVE LONG EXTENDED の特殊埋め込み文字により指定されます。
- 第10章、『制御命令』では、以下の命令が、記憶域オペランドに対する複数アクセス参照を行うことが明確化されています。
 - LOAD ADDRESS SPACE PARAMETERS
 - RESUME PROGRAM
 - STORE SYSTEM INFORMATION

第13章から第17章まででは、以下に示す重要な変更のほかに、明確化のための多数の変更が加えられており、これらはすべて余白の縦線で示されています。

- 第13章、『入出力の概要』では、CCW 内の延期フラグに関する説明が明確化されて、関連 ORB の中の延期制御の値が 1 であり、このフラグが 1 とし

て指定されている場合に、フラグが有効であることが明記されています。

- 第14章、『入出力命令』では、指定したサブチャネルにおける 装置番号有効ビットがゼロのときの MODIFY SUBCHANNEL の結果が訂正されました。
- 第15章、『基本入出力機能』の変更は次のとおりです。
 - 関連 ORB 内の事前取り出し制御の値に関係なく、現 CCW および事前に取り出された CCW に関連したデータおよび IDAW の事前取り出しが無制限に行えることが明確化されました。
 - オペレーターのアクションが原因でチャネル・サブシステム入出力優先順位機能が稼働していない場合は、指定した制御装置優先順位番号は無視されます。
 - アドレス限界検査はデータ位置に適用されるもので、CCW または IDAW を含む記憶位置には適用されないことが、明確化されました。
- 第16章、『入出力割り込み』では、故障記憶機構アドレス・フィールドに格納されるアドレスの形式が、アドレッシング・モードではなく、フォーマット 2 IDAW 制御の観点から説明されています。
- 第17章、『入出力サポート機能』の変更は次のとおりです。
 - チャネル・サブシステム・モニター機能についての概要説明が明確化されました。
 - 測定ブロック更新機能による測定ブロックに対する参照は、単一アクセス参照であり、CPU からはワード単位の並行処理のように見えません。ブロック単位の並行処理には見えません。
 - チャネル・サブシステム入出力優先順位機能の説明が修正され、ファイバー・チャネル接続制御装置に関する接続優先順位の説明が組み込まれました。

上記の変更の影響により、示されている章以外の章も変更されていることがあります。本文または図表に対して技術的な変更がある個所では、余白に縦線を示してあります。

第1章 概説

z/Architecture の特長	1-1	ESA/390 のベース	1-7
64 ビット整数用の汎用レジスター	1-2	ESA/370 および 370-XA のベース	1-13
その他の新しい一般命令	1-2	システム・プログラム	1-14
浮動小数点命令	1-4	互換性	1-14
制御命令	1-4	z/Architecture システム間の互換性	1-14
3 モード・アドレッシング	1-5	z/Architecture と ESA/390 の間の互換性	1-15
モーダル命令	1-5	制御プログラムの互換性	1-15
汎用レジスターのビット 0-31 への影響	1-5	問題プログラム状態の互換性	1-15
拡張変換機能 2	1-5	可用性	1-15
入出力	1-6		

本書では、参照用としてお使いいただくことを目的として、z/Architecture について詳しく説明します。

システムのアーキテクチャーは、プログラマーから見たシステムの属性、つまりマシンの概念的な構造と機能上の動作を定義するものであって、特定のインプリメンテーションにおけるデータ・フローの編成、論理設計、物理設計、およびパフォーマンスとは別個のものです。幾つかの異なるマシンが、1つのアーキテクチャーに準拠していることもあります。異なるマシン・インプリメンテーションで同じプログラムのセットを実行したときに、1つのアーキテクチャーで定義されている一連の結果が生じる場合、これらのプログラムにとっては、それらのインプリメンテーション間に互換性があるものと見なすことができます。

z/Architecture の特長

z/Architecture は、システム/360 からシステム/370 へ、そしてシステム/370 拡張アーキテクチャー (370-XA)、Enterprise Systems Architecture/370 (ESA/370)、さらに Enterprise Systems Architecture/390 (ESA/390) へと発展してきた開発における、次の新たなステップです。z/Architecture には、非同期ページアウト、非同期データ・ムーバー、program-call-fast、およびベクトル機構を除き、ESA/390 のすべての機能および機構が組み込まれています。さらに、z/Architecture は、次のような重要な拡張機能も提供します。

- 64 ビットの汎用レジスターおよび制御レジスター
- ESA/390 の 24 ビットおよび 31 ビット・アドレッシング (これは z/Architecture にも受け継がれている)

ます)に加えて、64 ビット・アドレッシング・モード。

オペランド・アドレスと命令アドレスのどちらも、64 ビット・アドレスとすることができます。プログラム状況ワード (PSW) は、大きい命令アドレスを格納できるように、16 バイトに拡張されました。また、PSW には、64 ビット・アドレッシング・モードを指定するための、新たに割り当てられたビットも含まれています。

- 64 ビット仮想アドレスを変換するための、領域テーブルと呼ばれる動的アドレス変換 (DAT) テーブルの最大 3 つの追加レベル。

仮想アドレス・スペースは、ESA/390 の場合のようにセグメント・テーブル指定により指定することも、また領域テーブル指定により指定することもでき、どちらのタイプの指定もアドレス・スペース制御エレメント (ASCE) と呼ばれます。ASCE は実スペース指定とすることもでき、その場合は、仮想アドレスは、DAT テーブルを使用せずに単なる実アドレスとして扱われます。

- 新旧両方の大きい PSW およびレジスター保管域を格納できるようにするための 8K バイトのプレフィックス域。
- ESA/390 アーキテクチャー・モデルと z/Architecture アーキテクチャー・モデルの間で切り替えるための SIGNAL PROCESSOR 指令。

初期プログラム・ローディングでは、ESA/390 アーキテクチャー・モードに設定されます。その後で、新しい SIGNAL PROCESSOR 指令を使用して、z/Architecture モードに設定したり、z/Architecture から ESA/390 に戻ることができま

す。この指令を使用することにより、構成内のすべての CPU を常に同じアーキテクチャー・モードにしておくことができます。

- 多数の新しい命令が追加されており、その多くは 64 ビット 2 進整数を操作対象にします。

64 ビット 2 進整数を対象としない新規命令の中の幾つかは、ESA/390 にも追加されました。

z/Architecture には、前に挙げた 4 つの機能を除き、ESA/390 のすべての命令が含まれています。

z/Architecture の汎用レジスタおよび制御レジスタのビット位置の番号は、0 から 63 までです。ESA/390 において 32 ビット・レジスタのビット位置 0-31 を対象として働く ESA/390 命令は、z/Architecture では 64 ビット・レジスタのビット位置 32-63 を対象として働きます。

64 ビット整数用の汎用レジスタ

z/Architecture には、64 ビット 2 進整数を対象として ESA/390 の 32 ビット 2 進整数命令と同じ働きをする新しいアナログがあります。この種のアナログには次の 2 つのタイプがあります。

- 2 つの 64 ビット 2 進整数を使用して 1 つの 64 ビット 2 進整数を導き出すアナログ。例えば、ESA/390 の ADD 命令 (記憶域/レジスタ間操作の場合は A で、レジスタ間操作の場合は AR) に対応するアナログは、AG (記憶域からの 64 ビットを 64 ビット汎用レジスタの内容に加算)、および AGR (64 ビット汎用レジスタの内容を別の 64 ビット汎用レジスタの内容に加算) です。これらのアナログは、二モニクの中に「G」が含まれていることで区別できます。
- 1 つの 64 ビット 2 進整数と 1 つの 32 ビット 2 進整数を使用して 1 つの 64 ビット 2 進整数を導き出すアナログ。32 ビット整数は、符号付きの場合は符号ビットで拡張され、符号なしの場合は左側が 0 で拡張されます。例えば、ESA/390 の ADD 命令 (A または AR) に対応するアナログは、AGF (記憶域からの 32 ビットを 64 ビット汎用レジスタの内容に加算)、および AGFR (64 ビット汎用レジスタのビット位置 32-63 を別の 64 ビット汎用レジスタの内容に加算) です。これらのアナログは、二モニクの中に「GF」が含まれていることで区別できます。

その他の新しい一般命令

以下に示すのは新規追加または大幅に拡張されたその他の z/Architecture 一般命令です。

- ADD LOGICAL WITH CARRY および SUBTRACT LOGICAL WITH BORROW は、32 ビットまたは 64 ビットの符号なし 2 進整数を対象として操作を実行し、PSW の 2 ビット条件コードの左端ビットで表される繰り上がりまたは繰り下がり、計算を含めます。これにより、拡張精度整数 (64 ビットより長い整数) に対する操作のパフォーマンスが向上します。
- AND IMMEDIATE および OR IMMEDIATE は、2 バイトの即値オペランドを、64 ビットの汎用レジスタ内の 2 バイト境界上にある 2 バイトと結合します。
- BRANCH AND SAVE AND SET MODE および BRANCH AND SET MODE が拡張された結果、現行アドレッシング・モードが 64 ビット・モードである場合は R₁ 汎用レジスタのビット 63 が 1 に設定され、R₂ 汎用レジスタが 1 の場合は 64 ビット・アドレッシング・モードが設定されるようになりました。これにより、アドレッシング・モード (24 ビット、31 ビット、また 64 ビット・モード) の異なるプログラム間での「ポインター指示」によるリンケージが可能になります。
- 新しい BRANCH RELATIVE AND SAVE LONG 命令および BRANCH RELATIVE ON CONDITION LONG 命令は、ESA/390 の BRANCH RELATIVE AND SAVE 命令および BRANCH RELATIVE ON CONDITION 命令と同じ働きをしますが、32 ビット即値フィールドを使用する点が異なります。これにより、相対ブランチに使用できるターゲットの範囲が広がります。
- COMPARE AND FORM CODEWORD が拡張された結果、64 ビット・アドレッシング・モードでは、比較単位は 2 バイトでなく 6 バイトになり、結果のコードワードは 4 バイトでなく 8 バイトになりました。また、UPDATE TREE が拡張された結果、64 ビット・アドレッシング・モードでは、ノードは 8 バイトでなく 16 バイトになり、ノード内のコードワードは 4 バイトでなく 8 バイトになりました。これにより、長いキーを持つレコードをソートするときのパフォーマンスが向上します。
- DIVIDE LOGICAL および MULTIPLY LOGICAL は、符号なし 2 進整数に対する操作を

実行し、32 ビットまたは 64 ビットの商と剰余、または積を導き出します。

- **DIVIDE LOGICAL** は、64 ビットまたは 128 ビットの符号なし 2 進数を被除数とし、32 ビット (64 ビット被除数の場合) または 64 ビット (128 ビット被除数の場合) の符号なし 2 進数を除数として、それぞれ 32 ビットまたは 64 ビットの商と剰余を導き出します。**MULTIPLY LOGICAL** は、32 ビットまたは 64 ビットの符号なし 2 進数を被乗数および乗数として使用して、それぞれ 64 ビットまたは 128 ビットの積を導き出します。
- **DIVIDE SINGLE** は、64 ビットの被除数を 32 ビットまたは 64 ビットの除数で除算して、64 ビットの商と剰余を導き出します。**MULTIPLY SINGLE** が拡張された結果、64 ビットの被乗数に 32 ビットまたは 64 ビットの乗数を掛けて、64 ビットの積を導き出すことができるようになりました。
- **EXTRACT PSW** は、現 PSW 全体を抽出します。これにより、現在のマシンの状態、例えば、CPU が問題プログラム状態にあるのか監視プログラム状態にあるのかを判別することができます。
- **INSERT IMMEDIATE** は、64 ビット汎用レジスタ内の任意の 2 バイト境界に、2 バイトの即値オペランドを挿入します。**LOAD LOGICAL IMMEDIATE** は、これと同じことを行い、さらにレジスタの残り部分をクリアします。
- **LOAD ADDRESS RELATIVE LONG** は、符号付き 32 ビット即値フィールドを使用して、現行の (更新されていない) 命令アドレスを基準とする相対アドレスを作成します。
- **LOAD LOGICAL THIRTY ONE BITS** は、1 つの汎用レジスタまたは記憶域内の 1 ワードの右端の 31 ビットの左側に 33 個の 0 を付加したものを、汎用レジスタに入れます。
- **LOAD MULTIPLE DISJOINT** は、特定範囲内の各汎用レジスタの左端 32 ビットを記憶域内の 1 つの領域から取り出し、同じレジスタの右端 32 ビットを記憶域内の別の領域から取り出して、ロードします。これは、ロードされているレジスタの 1 つにより記憶域の 1 つがアドレッシングされているときに、**LOAD MULTIPLE HIGH** 命令および 32 ビットの **LOAD MULTIPLE** 命令の代わりに使用します。
- **LOAD MULTIPLE HIGH** および **STORE MULTIPLE HIGH** は、特定範囲内の各汎用レジスタの左端 32 ビットを、ロードまたは保管します。こ

れにより、**LOAD MULTIPLE** および **STORE MULTIPLE** を使用して右端 32 ビットをロードまたは保管する既存プログラムの機能を、拡張することができます。(64 ビット形式の **LOAD MULTIPLE** および **STORE MULTIPLE** も提供されています。)

- **LOAD PAIR FROM QUADWORD** および **STORE PAIR TO QUADWORD** は、64 ビット汎用レジスタの偶数/奇数ペアと、記憶域内のクワッドワード (quadword: 16 バイト境界上に位置合わせされた 16 バイト) との間で、操作を行います。これらの命令は、クワッドワードの整合性を確保します (マルチ CPU システムにおいてすべてのバイトが同時にロードまたは保管されるようになります)。(**COMPARE DOUBLE AND SWAP** でも、64 ビット形式の場合のみ、クワッドワードの整合性を提供します。)
- **LOAD REVERSED** および **STORE REVERSED** は、2 バイト、4 バイト、または 8 バイト単位を、左から右へのバイト順序を反転させて記憶域内にロードまたは保管します。**LOAD REVERSED** は、4 バイトまたは 8 バイト単位を 2 つの汎用レジスタ間で移動することもできます。これらの操作を使用して、「リトル・エンディアン」フォーマットと「ビッグ・エンディアン」フォーマットの間で変換を行うことができます。
- **PERFORM LOCKED OPERATION** は、複数セットの機能コードを使用できるように拡張されました。各セットは、それぞれ 6 種類の操作を提供します。追加されたセットの 1 つは、64 ビット汎用レジスタ内の 64 ビット・オペランドに対する操作を提供し、もう 1 つは、パラメーター・リスト内の 128 ビット・オペランドに対する操作を提供します。
- **ROTATE LEFT SINGLE LOGICAL** は、1 つの汎用レジスタから 32 ビットまたは 64 ビットを取り出し、循環 (右端ビットを左端ビットで置き換える) を行い、その結果を別の汎用レジスタに入れます (非破壊循環)。
- **SET ADDRESSING MODE** は、アドレッシング・モードを 24 ビット、31 ビット、および 64 ビットのいずれかに設定します。
- **SHIFT LEFT SINGLE**、**SHIFT LEFT SINGLE LOGICAL**、**SHIFT RIGHT SINGLE**、および **SHIFT RIGHT SINGLE LOGICAL** は、1 つの汎用レジスタからソース・オペランドを取り出して、結果のオペランドを別の汎用レジスタに入

るもの(被破壊シフト)ですが、それぞれに 64 ビット形式が加わりました。

- TEST ADDRESSING MODE は、現 PSW のビット 31 および 32 が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかを示す条件コードを設定します。
- TEST UNDER MASK HIGH および TEST UNDER MASK LOW (これらは ESA/390 の命令です)に、TEST UNDER MASK という代替名が与えられ、新たな 2 つの形式が追加されています。その結果、2 バイトの即値オペランドを使用して、64 ビット汎用レジスター内の任意の 2 バイト境界上にある 2 バイトのビットをテストできるようになりました。(ESA/390 の TEST UNDER MASK 命令は、1 バイトの即値オペランドを使用して記憶域内の 1 バイトをテストするものですが、これも引き続き使用できます。)

浮動小数点命令

z/Architecture の浮動小数点命令は、64 ビットの符号付き 2 進整数と 16 進または 2 進の浮動小数点データとの間で変換を行うための命令が追加されている以外は、ESA/390 と同じです。新たに追加された命令は、ニーモニックに「G」の文字が含まれています。

制御命令

以下の示すのは、z/Architecture で新規追加または拡張された制御命令の主なものです。

- EXTRACT AND SET EXTENDED AUTHORITY は、制御レジスター内の拡張許可指標を変更するための特権命令です。これにより、アクセス・リストを使用して、実スペース指定をさらに効率的に使用することができます。
- EXTRACT STACKED REGISTERS が拡張された結果、オプションとして、保管済みの 1 つ以上の汎用レジスターの内容のすべての 64 ビットを抽出できるようになりました。
- EXTRACT STACKED STATE が拡張された結果、オプションとして、保管済みの PSW の内容の全体 (64 ビット命令アドレスも含む) を抽出できるようになりました。
- LOAD CONTROL および STORE CONTROL が拡張されて、オプションとして 64 ビット制御レジスターを操作できるようになりました。

- LOAD PSW は、ESA/390 の場合と同様に 8 バイトの記憶域オペランドを使用し、このオペランドを 16 バイトの z/Architecture PSW に拡張します。
- LOAD PSW EXTENDED は、16 バイト PSW を直接ロードします。
- LOAD REAL ADDRESS は、ESA/390 形式で、アドレッシング・モードが 24 ビットまたは 31 ビットである場合は、変換が成功し、取得した実アドレスの値が 2G バイト未満であれば、ESA/390 の場合と同じ働きをします。LOAD REAL ADDRESS が、ESA/390 形式で 64 ビット・アドレッシング・モードである場合、または任意のアドレッシング・モードの拡張 z/Architecture 形式である場合は、64 ビットの実アドレスがロードされます。
- SIGNAL PROCESSER には新しい指令が加わっており、これを使用して、構成内のすべての CPU を、ESA/390 アーキテクチャー・モードから z/Architecture アーキテクチャー・モードに、またはその逆に切り替えることができます。(z/Architecture を使用してシステムを稼働させる場合でも、最初は ESA/390 モードで IPL を行う必要があります。)
- LOAD USING REAL ADDRESS および STORE USING REAL ADDRESS が拡張されて、オプションとして 64 ビット・オペランドを使用できるようになりました。
- STORE FACILITY LIST は、z/Architecture がインストールされているかどうか、そしてそれがアクティブになっているかどうかの指示を、実記憶位置 200 に格納する特権命令です。この命令は ESA/390 にも追加されており、ESA/390 に追加された新しい z/Architecture の命令が使用可能かどうかを示す指示も格納するようになりました。以前は、ほとんどのシステムでは実記憶位置 200 には全桁ゼロが含まれており、通常、STORE FACILITY LIST がインストールされているかどうかに関係なく、問題プログラム状態プログラムにより調べることができました。実記憶位置 200 に保管されている情報は、拡張変換機能 2 がインストールされているかどうかを示します。
- STORE REAL ADDRESS は、LOAD REAL ADDRESS に似ていますが、結果のアドレスをレジスターに入れる代わりに保管する点が異なります。

- TRACE が拡張されて、オプションとして 64 ビット汎用レジスタの内容を記録できるようになりました。

3 モード・アドレッシング

「3 モード・アドレッシング」とは、24 ビット、31 ビット、および 64 ビット・アドレッシング・モードを相互に切り替えることのできる能力を言います。切り替えには以下の方法を使用できます。

- 従来の BRANCH AND SAVE AND SET MODE 命令と BRANCH AND SET MODE 命令。どちらの命令も、R₂ 汎用レジスタのビット 63 が 1 である場合に、64 ビット・アドレッシング・モードを設定します。ビット 63 が 0 の場合は、これらの命令は、このレジスタのビット 32 が 0 であれば 24 ビット・アドレッシング・モードに、そしてビット 32 が 1 であれば 31 ビット・アドレッシング・モードに設定します。
- 新しい SET ADDRESSING MODE (SAM24、SAM31、および SAM64) 命令。この命令は、指定された命令コードに従って、24 ビット、31 ビット、または 64 ビット・アドレッシング・モードを設定します。

モーダル命令

3 モード・アドレッシングは、以下の例外を除き、一般命令において論理記憶域アドレスをどのように扱うかに対して影響を与えるだけです。

- BRANCH AND LINK、BRANCH AND SAVE、BRANCH AND SAVE AND SET MODE、BRANCH AND SET MODE、および BRANCH RELATIVE AND SAVE の各命令は、24 ビットまたは 31 ビット・アドレッシング・モードでは、ESA/390 の場合と同様に、汎用レジスタ R₁ のビット位置 32-39 に情報を入れ、64 ビット・アドレッシング・モードでは、これらのビットにアドレス・ビットを入れます。新しい BRANCH RELATIVE AND SAVE LONG 命令も同じことを行います。
- BRANCH AND SAVE AND SET MODE および BRANCH AND SET MODE 命令は、64 ビット・アドレッシング・モードでは、汎用レジスタ R₁ のビット位置 63 に 1 を入れます。24 ビットまたは 31 ビット・モードでは、BRANCH AND SAVE AND SET MODE はビット 63 を 0 に設定し、BRANCH AND SET MODE はこのビットを変更しません。

- 命令によっては、24 ビットまたは 31 ビット・アドレッシング・モードでは汎用レジスタのビット 0-31 を変更せず、64 ビット・アドレッシング・モードではこれらのビット内でアドレスまたは長さ情報を配置または更新するものもあります。この種の命令はモーダル命令とも呼ばれるもので、7-6 ページの「プログラミング上の注意」にリストされています。

汎用レジスタのビット 0-31 への影響

汎用レジスタのビット 0-31 を変更する命令には、2 つのタイプがあります。第 1 のタイプは、64 ビット・アドレッシング・モードで実行されているときのモーダル命令です (前のセクションを参照)。第 2 のタイプは、アドレッシング・モードに関係なく、64 ビットの結果オペランドを単一の汎用レジスタに入れ、128 ビットの結果オペランドを偶数/奇数ペアの汎用レジスタに入れる命令です。

第 2 タイプの命令のほとんどは、ニーモニックに「G」または「GF」の文字が含まれています。現行のアドレッシング・モードに関係なく汎用レジスタのビット 0-31 を変更する (または変更する可能性のある) その他の命令は、7-7 ページの「プログラミング上の注意」にリストされています。第 2 タイプのすべての命令を総称して、「G タイプ」命令とも言います。

64 ビット・アドレッシング・モードで実行されておらず、G タイプの命令も含まれていないプログラムは、どの汎用レジスタのビット 0-31 も変更することはできません。

拡張変換機能 2

拡張変換機能 2 は、z/Architecture をインプリメントするモデルで使用できます。この機能は、2 バイトデータ、ASCII データ、および 10 進データに対する操作を行います。2 バイト・データには、Unicode データを使用できます。Unicode データは、ユニコード万国共通文字規格 (Unicode Worldwide Character Standard) のバイナリー・コードを使用し、全世界のほとんどの言語の文字を使用可能にするデータです。この機能は以下の命令を提供します。

```
COMPARE LOGICAL LONG UNICODE
MOVE LONG UNICODE
PACK ASCII
PACK UNICODE
TEST DECIMAL
TRANSLATE ONE TO ONE
TRANSLATE ONE TO TWO
```

TRANSLATE TWO TO ONE
TRANSLATE TWO TO TWO
UNPACK ASCII
UNPACK UNICODE

ESA/390 にはすでに導入されていた拡張変換機能があり、現在ではそれが機能 1 と呼ばれるようになったため、拡張変換機能 2 は機能 2 と呼ばれています。z/Architecture には、標準機能として機能 1 が組み込まれています。機能 1 は以下の命令を提供します。

CONVERT UNICODE TO UTF-8
CONVERT UTF-8 TO UNICODE
TRANSLATE EXTENDED

マシンに機能 1 および機能 2 のいずれか一方または両方ともにインストールされていない場合は、OS/390 リリース 10 および z/OS が提供する MVS CSRUNIC マクロ命令によって、この両方の機能がシミュレートされます。

「OS/390 MVS Assembler Services Reference」(GC28-1910-10) に、CSRUNIC マクロ命令に関するプログラミング要件、レジスタ情報、構文、戻りコード、および例を収めてあります。

CSRUNIC を使用した場合は、本書に示されているプログラム例外によりプログラム割り込みが生じることはありません。CSRUNIC は、GC28-1910-10 に示されているように、戻りコードにより例外条件を示します。

入出力

z/Architecture をインストールするときに、追加の入出力関数および機能が提供されます。これらの機能は、ESA/390 および z/Architecture の両方のアーキテクチャー・モードで提供されるもので、以下のようなものがあります。

- 間接データ・アドレッシングが拡張されて、ダブルワード・フォーマット 2 IDAW が提供されるようになりました。この目的は、z/Architecture において 2G バイト絶対アドレス境界より上位のデータを対象として操作を実行できるようにすることです。従来の IDAW (31 ビット・アドレスを含むワード) は、現在はフォーマット 1 IDAW と呼ばれています。フォーマット 2 IDAW には 64 ビット・アドレスが含まれます。チャンネル・プログラムに関連した操作要求ブロック (operation-request block: ORB) の中に、そのプログラムがフォーマット 1 またはフォーマット 2 のどちらの IDAW を使用する

かを指定するビットがあります。さらに、チャンネル・プログラムのすべてのフォーマット 2 IDAW で 2K バイトまたは 4K バイトいずれかのデータ・ブロックを指定する能力も追加されており、これは ORB 内の別のビットにより決定されます。4K バイト・ブロックを使用すると、データ転送の効率が向上します。

- **FICON チャンネル機構** は、FICON 入出力インターフェースおよび FICON 変換入出力インターフェースのチャンネル・パスを接続する能力と、これらのチャンネル・パス・タイプをフル活用する能力を提供します。FICON チャンネル・パスを使用すると、ESCON チャンネル・パスに比べてデータ転送速度が高まり、また、複数のコマンドと関連データを「ストリーム化」して制御装置に渡すことでパフォーマンスがさらに向上するため、全体的なデータ・スループットが大幅に増加します。この機能は、次のような追加の制御メカニズムをサポートします。
 - ORB 内の変更制御ビットを使用することにより、プログラムは、チャンネル・プログラムを動的に変更するときの FICON チャンネルのパフォーマンスを最適化することができます。
 - ORB の同期制御ビットは、CCW の無制限の事前取り出しを行うとき、および書き込む必要があるデータが直前に読み取ったデータである場合に、直前の読み取りコマンドが完了するまで書き込みコマンドの実行を遅延させます。これにより、データ保全性と最大限のチャンネル・パス・パフォーマンスが確保されます。
 - ORB のストリーム・モード制御ビットを使用すると、プログラムは、コマンド・ストリーミングを回避する必要がある場合に、それを回避することができます。
 - 拡張状況ワードの「secondary-CCW-address (2 次 CCW アドレス)」フィールドは、制御装置に対するコマンド・ストリーミングを行っているときに、異常終了したチャンネル・プログラムをリカバリーするために役立ちます。このフィールドは、制御装置で失敗した CCW を識別します。
- **ORB 拡張機能** は、ORB のサイズを 3 ワードから 8 ワードに拡張します。これにより、チャンネル・サブシステム入出力優先順位機能でフィールドを使用できるようになります。
- **チャンネル・サブシステム入出力優先順位決定機能** を使用すると、プログラムは、保留中の入出力操作があるサブチャンネル間の優先順位関係を設定することができます。優先順位関係は、チャンネル・サブシステムが入出力操作をどの順序で開始するかを指定

します。さらに、ファイバー・チャンネル接続の制御装置の場合は、この機能を使用することにより、プログラムは、制御装置で保留状態にある入出力操作を実行する優先順位を指定することができます。

入出力に関する機能強化については、次のように、入出力関連の各章でその効果をさらに詳しく説明します。

- 第13章、『入出力の概要』では、FICON 入出力インターフェース、FICON 変換入出力インターフェース、およびフレーム多重方式を紹介します。
- 第14章、『入出力命令』
 - CANCEL SUBCHANNEL 命令について説明します。
 - TEST PENDING INTERRUPTION は、第 2 オペランド・アドレスが 0 の場合、3 ワードの入出力割り込みコードを実記憶位置 184-195 に保管します。新しい第 3 ワードには、入出力割り込みのソースをさらに限定的に識別する割り込み識別ワードが含まれます。
- 第15章、『基本入出力機能』
 - ORB は 8 ワードに拡張され、新たに、ストリーム・モード制御、変更制御、同期制御、フォーマット 2 IDAW 制御、2K-IDAW 制御、ORB 拡張制御、チャンネル・サブシステム優先順位、および制御装置優先順位が含まれるようになりました。
 - ダブルワード・フォーマット 2 IDAW と、フォーマット 2 IDAW によりオプションで指定可能な 4K バイト・データ・ブロックが、追加されました。
- 第16章、『入出力割り込み』
 - 拡張レポート・ワードに、2 次 CCW アドレス妥当性ビットおよび故障記憶機構アドレス・フォーマット・ビットが追加されました。
 - フォーマット 0 の拡張状況ワードに、2 ワードの故障記憶機構アドレスおよび 2 次 CCW アドレスが追加されました。
- 第17章、『入出力サポート機能』
 - 制御装置据え置き時間が追加されました。これは、測定ブロック内の装置接続時間および装置切断時間に影響を与えます。
 - 測定ブロック更新機能による測定ブロックに対する参照は、単一アクセス参照であり、CPU からはワード単位の並行処理として認識されません。

- 測定ブロックに、装置アクティブのみの時間が追加されました。
- チャンネル・サブシステム入出力優先順位決定機能機能が追加されました。これは、チャンネル・サブシステム優先順位および制御装置優先順位を提供します。

ESA/390 のベース

z/Architecture には、非同期ページアウト、非同期データ・ムーバー、program-call-fast、およびベクトル機構を除き、ESA/390 のすべての機能および機構が組み込まれています。このセクションでは、ESA/370 から ESA/390 への移行の段階で追加された、その他のほとんどの機能について、簡単に説明します。

ESA/390 については、「Enterprise Systems Architecture/390 Principles of Operation」(SA22-7201) で解説されています。

ここでは、ESA/390 で新たに導入された CPU 関連機能について概説します。ESA/390 は、1990 年 9 月に発表されました。その後追加された拡張機能については、該当の説明の後に、発表の日付を括弧に入れて示してあります。

以下の拡張機能に関する詳細説明は、SA22-7201 および本書に収めてあります。

- **アクセス・リスト制御保護 (Access-list-controlled protection)** を使用すると、特定のアドレス・スペースにアクセスするために使用するアクセス・リスト・エントリー内の 1 つのビットにより、そのスペースに対する格納タイプの記憶域参照を禁止することができます。したがって、それぞれ異なるアクセス・リストを持つ複数のユーザーに、同じアドレス・スペースへのそれぞれ異なる保管能力を持たせることができます。
- **プログラム・イベント記録機能 2 (PER 2)** は、最初の PER 機能 (現在は PER 1 と呼ばれています) に代わる機能です。(z/Architecture では、PER 1 と PER 2 のどちらの名前も使用されず、実際には単に「PER」と呼ばれています)。PER 2 には、ブランチ・ターゲットが指定の記憶域内にある場合のみブランチ成功イベントを発生させるためのオプションと、記憶域が指定のアドレス・スペース内にある場合のみ記憶域変更イベントを発生させるためのオプションがあります。これらのオプションを使用することにより、必要な PER イベントのみを発生させることができるので、パフォーマンスが向上します。

す。PER 2 では、汎用レジスター変更イベントのためのモニター機能が削除されています。

PER 2 には、PER イベントに関する追加情報を提供する拡張機能が組み込まれています。SA22-7201 の第 4 版より、これらの拡張機能についての詳細説明が加わっています。

- **並行センス** は、装置チェック条件が原因で割り込みが生じたときにセンス情報が提示されるようになります。これにより、センス情報を取得するためのみ入出力操作をする必要がなくなるため、パフォーマンスが向上します。
- **ブロードキャスト・ページ** は、動的アドレス変換およびアクセス・レジスター変換に関連したテーブルの更新、および複数の CPU 内の関連バッファのクリアを条件付きで行うための、COMPARE AND SWAP AND PURGE 命令を提供します。SA22-7201 の第 8 版より、この拡張機能についての詳細説明が加わっています。
- **記憶域保護オーバーライド** は、新しい形態のサブシステム記憶域保護機能です。これにより、エラーのある可能性があるアプリケーション・プログラムを含むアドレス・スペース内でサブシステムを実行したときの信頼性が向上します。制御レジスター・ビットにより記憶域保護オーバーライドがアクティブにされているときは、CPU が使用しているアクセス・キーに関係なく、記憶キー 9 の記憶位置で CPU が取り出しと格納することが許可されます。例えば、サブシステムはキー 8 の記憶域内にあって、PSW キー 8 を使用して実行されており、一方、アプリケーション・プログラムはキー 9 の記憶域内にあって、PSW キー 9 を使用して実行されている場合は、サブシステムからアプリケーション・プログラム領域へのアクセスは許可されますが、アプリケーション・プログラムからサブシステム領域へのアクセスは拒否されます。(1991 年 9 月)
- **ページ移動機能 2** は、ESA/370 で導入された MOVE PAGE 命令の機能を拡張するものです。その結果、指定したアクセス・キーをソース・オペランドと宛先オペランドのどちらにも使用できるようになり、宛先オペランドがすぐに参照される場合のパフォーマンスが向上し、また、特定オペランドが主記憶機構と拡張記憶機構の両方で無効であった場合にも、パフォーマンスが向上します。ESA/370 バージョンの MOVE PAGE は現在はページ移動機能 1 と呼ばれています(これについては第 7 章、『一般命令』で説明します)。また、ページ移動機能 2 の MOVE PAGE については、第 10 章、『制御命令』で説明します。MOVE PAGE の制御プログラム・サポートに関する手段についての詳細説明

は、一部省略されています。(1991 年 9 月)

(z/Architecture の MOVE PAGE 命令についての説明があるのは、本書の第 10 章のみです。MOVE PAGE では拡張記憶機構への(またはそこからの)データ移動はできなくなりました。MOVE PAGE に関しては、すべての詳細説明を収めてあります。)

- **平方根機能** は、SQUARE ROOT 命令と平方根例外から成っています。この命令は、浮動小数点オペランドの平方根を、長精度または短精度フォーマットで抽出します。この命令は、IBM 4341、4361、および 4381 プロセッサの一部のモデルで提供されているものと同じです。(1991 年 9 月)
- **入出力取り消し機能** は、プログラムが装置に通知せずに、指定のサブチャネルから保留状態の開始機能を撤回できるようにします。これは、ある種のエラー・リカバリー状況の場合に便利です。(1991 年 9 月)

入出力取り消し機能は、CANCEL SUB-CHANNEL 命令を提供します。この命令については、SA22-7201 の第 8 版より詳細説明が加わっています。

- **ストリング命令機能(または論理ストリング補助)** は、(1) 指定した終了バイトが見つかるまでバイトのストリングを移動する命令、(2) 不一致または指定した終了バイトが見つかるまで 2 つのストリングを論理的に比較する命令、および (3) 指定した長さのストリングの中で指定のバイトをサーチする命令を提供します。最初の 2 つの命令は、特に、通常は全桁ゼロの終了バイトでストリングが区切られる C プログラムの中で使用すると便利です。(1992 年 6 月)
- **保護時抑止機能** は、ページ保護が原因で記憶保護例外が生じたときに、命令の実行を終了するのではなく抑止し、また、保護ページのアドレスおよびアドレス・スペース ID が低位の記憶域に保管されるようにします。これは、AIX/ESA のコピー・オン・ライト機能を実行するとき便利です。コピー・オン・ライト機能では、AIX/ESA は、複数のアドレス・スペースの同一ページを、そのページ内で保管が試行されない限り実記憶域の単一のページ・フレームにマップし、特定のアドレス・スペース内で保管が試行されると、そのアドレス・スペース内で固有のページ・フレームを割り当てて、その新しいページ・フレームにページの内容をコピーします。(1993 年 2 月)
- **アドレス・スペース制御高速設定機能** は、SET ADDRESS SPACE CONTROL FAST (SACF) 命

令から成っています。この命令は、すべての SAC 機能が必要かどうかに応じて、前からあった SET ADDRESS SPACE CONTROL (SAC) 命令の代わりに使用できます。SACF は、SAC と異なり、逐次化およびチェックポイント同期機能を実行せず、事前に取り出された命令のコピーを破棄しません。SACF を使用すると、一部のモデルでのパフォーマンスが向上します。(1993 年 2 月)

- **サブスペース・グループ機能**には、**BRANCH IN SUBSPACE GROUP** 命令が含まれています。この命令を使用すると、サブスペース・グループと呼ばれるアドレス・スペースのグループの中で、1 つのアドレス・スペースから別のアドレス・スペースに制御を与えたり戻したりできます。これにより、**PROGRAM CALL** 命令と **PROGRAM RETURN** または **PROGRAM TRANSFER** 命令を使用した場合より、制御に引き渡しや戻しを実行するときのパフォーマンスが高まります。サブスペース・グループ内の一方のアドレス・スペースを基本スペースと呼び、グループ内のもう一方のアドレス・スペースをサブスペースと呼びます。各サブスペースには、基本スペース内の (サブスペース別に異なる) 記憶域のサブセットが 1 つずつ含まれています。基本スペースと各サブスペースには、それぞれサブシステム制御プログラム (CICS など) が 1 つと複数のアプリケーション・プログラムが含まれており、各サブスペースには、サブシステム制御プログラムの下で処理されている単一トランザクション用のデータが含まれています。個々のトランザクションごとに異なるサブスペースにデータを含めておくことで、1 つのトランザクションの処理により、誤って他のトランザクションのデータが損傷を受けるのを防ぐことができます。制御プログラムのデータは、記憶域保護オーバーライド機能によりトランザクション処理から保護することができます。(1994 年 4 月)
- **保護時抑止の仮想アドレス拡張**は、記憶保護例外が検出されたときに動的アドレス変換 (DAT) がオンになっている場合に、記憶保護域のアドレスが DAT で変換すべきアドレスの場合のみ、「保護時抑止」結果を指示してそのアドレスを格納します。格納するアドレスが実アドレスである場合は、「保護時抑止」結果は指示されません。この拡張機能により、格納されるアドレスが制御プログラムによって正しく変換されるので、例外が、キー制御による保護ではなくページ保護が原因で生じたものかどうかを判別できます。この拡張機能により、キー制御による保護を使用するオペレーティング・システム (例えば MVS/ESA) で、保護時抑制の機能をさらに効果的に使用できるようになります。(1994 年 9 月)
- **即時/相対命令機能**には、13の新しい命令が組み込まれています。これらの命令のほとんどは、符号付き 2 進算術演算または相対ブランチのどちらにも、ハーフワード即値を使用します。この機能により、汎用レジスタの必要度が減少し、特に、ブランチ・ターゲットをアドレッシングするために汎用レジスタを使用する必要がなくなります。したがって、多数のレジスタを必要とするプログラムの中で、汎用レジスタとアクセス・レジスタを効率的に割り振ることができます。(1996 年 9 月)
- **拡張比較/移動機能**は、新バージョンの **COMPARE LOGICAL LONG** および **MOVE LONG** 命令を提供します。これらの新バージョンでは、オペランド長指定のサイズが 24 ビットから 32 ビットに拡大されています。これは、16M バイトより大きいオブジェクトを、31 ビット・アドレッシングを使用して処理するときに便利です。また、これらの新バージョンでは、マルチプロセッシング・システムでのソフトウェア・ポーリングを可能にするために、処理が定期的に完了します。(1996 年 9 月)
- **チェックサム機能**は **CHECKSUM** 命令から成っています。この命令は、TCP/IP (transmission-control protocol/internet protocol) のパフォーマンスを高めることを目的として、16 ビットまたは 32 ビットのチェックサムを計算するために使用できます。(1996 年 9 月)
- **呼び出し先スペース識別機能**により、**PROGRAM CALL** 命令で作成されるリンケージ・スタック状態エントリ内で呼び出し先アドレス・スペースをさらに詳細に識別できるため、保守性が向上します。(1996 年 9 月)
- **ブランチおよび権限設定機能**は **BRANCH AND SET AUTHORITY** 命令から成ります。この命令を、**PROGRAM CALL**、**PROGRAM TRANSFER**、および **SET PSW KEY FROM ADDRESS** 命令の代わりに使用することにより、アドレス・スペース内でのリンケージのパフォーマンスを高めることができます。(1997 年 6 月)
- **ロック操作実行機能**は、非特権 **PERFORM LOCKED OPERATION** 命令から成ります。この命令により、複数の記憶域オペランドに対する並行インターロック更新参照が可能になります。この命令の機能コードには、6 つの操作のどれでも指定できます。6 つの操作とは、比較およびロード、比較およびスワップ、二重比較およびスワップ、比較およびスワップおよび格納、比較およびスワップおよび二重格納、比較およびスワップおよび三重格納です。機能コードは、さらに、ワード・オペランドま

たはダブルワード・オペランドを指定します。この命令を使用することにより、マルチプロセッシング・システムでプログラム式ロックを使用しないで済みます。(1997年6月)

- 4つの浮動小数点機能が新たに追加された結果、マシンの16進浮動小数点(HFP)機能が強化され、2進浮動小数点(BFP)機能が追加されました。これらの機能は以下のとおりです。
 - **基本浮動小数点拡張機能。**これにより12個の新しい浮動小数点レジスターが追加され、浮動小数点レジスターの合計数は16個になりました。この機能には、浮動小数点制御レジスターと、状況記憶操作またはマシン・チェック割り込みの際に新レジスターの内容を保管するための手段が含まれています。
 - **浮動小数点サポート(FPS)拡張機能。**これにより、8つの新しい命令が追加されました。そのうちの4つは、HFPフォーマットとBFPフォーマットの間でデータを変換するためのものです。
 - **16進浮動小数点(HFP)拡張機能。**これは、HFPフォーマットのデータを操作するための新しい26個の命令を提供します。これらの命令はすべて、BFP機能が提供する新しい命令に対応するもので、浮動小数点フォーマットと固定小数点フォーマットの間の変換のほか、拡張フォーマットに対する操作の強化セットが含まれています。
 - **2進浮動小数点(BFP)。**これは、短精度、長精度、および拡張精度の2進浮動小数点(BFP)データ・フォーマットを定義するもので、これらのフォーマットのデータを操作するための新しい87個の命令を提供します。BFP形式および操作は、浮動小数点数と10進数ストリングの間の変換を除き、IEEE標準(ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic (1985年8月12日))に準拠するために必要なすべてのものを提供します。浮動小数点数と10進数ストリングの間の変換は、ソフトウェアの中で対応する必要があります。(1998年5月)
- **プログラム再開機能**は、RESUME PROGRAM 命令から成っています。この命令は、指定した保管域から、現PSW内の命令アドレスおよび他の幾つかのフィールドのほか、アクセス・レジスターと汎用レジスターのペアの内容も復元します。RESUME PROGRAM は、復元する状態を含む保管域をアド

レッシングするために本来ならレジスターが必要な場合であっても、問題プログラム状態割り込み処理プログラムで割り込まれたプログラムの状態を復元し、そのプログラムに戻ることができるようにします。(1998年5月)

- **トラップ機能**は、TRAP 命令を提供します(2バイトのTRAP2命令と4バイトのTRAP4命令)。この命令では、アプリケーション・プログラム内の命令をオーバーレイして、処理中のデータ(例えば「2000年」問題を引き起こすおそれのある日付など)に対する修正操作を行うプログラムに制御を与えることができます。修正プログラムから戻るには、RESUME PROGRAM を使用できます。TRAP および RESUME PROGRAM を使用することにより、修正プログラムとの間で制御を受け渡しするために必要なプログラム割り込みを回避できるので、パフォーマンスが向上します。(1998年5月)
- **拡張TODクロック機能**では以下のように機能が強化されています。(1) TODクロックが64ビットから104ビットに拡張された結果、分解能が向上しました。(2) TODプログラマブル・レジスターが組み込まれました。このレジスターには、シスプレックス内のTODクロック値を提供する構成を識別するために使用できる、TODプログラマブル・フィールドが含まれています。(3) TODプログラマブル・レジスター内にTODプログラマブル・フィールドを設定するためのSET CLOCK PROGRAMMABLE FIELD 命令が組み込まれました。(4) STORE CLOCK EXTENDED 命令が組み込まれました。この命令は、長いTODクロック値およびTODプログラマブル・フィールドの両方を保管します。将来、TODクロックがさらに拡張されて、現在の2042年の制限を超える時刻値を含むようになったとき(TODクロックの現行ビット0からの繰り上がりが生じるとき)に、STORE CLOCK EXTENDED を使用できます。(1998年8月)
- **TODクロック制御オーバーライド機能**は、CPUのTODクロック制御を手動で行う必要をなくし、プログラム制御の下でTODクロックを設定できるようにするための制御レジスター・ビットを提供します。(1998年8月)
- **システム情報格納機能**は、STORE SYSTEM INFORMATION 特権命令を提供します。この命令は、仮想計算機、論理区画、または基本計算機のコンポーネントに関する情報を取得するために使用できます。(1999年1月)

- **拡張変換機能** (現在は拡張変換機能 1 と呼ばれています) には、**CONVERT UNICODE TO UTF-8**、**CONVERT UTF-8 TO UNICODE**、および **TRANSLATE EXTENDED** 命令が組み込まれており、いずれもパフォーマンスの向上をもたらします。これ以前には、**TRANSLATE AND TEST** 命令によりエスケープ文字を検出し、続いて **TRANSLATE** 命令によりそのエスケープ文字より前のバイトを変換するという方法がとられていましたが、その代わりに現在では **TRANSLATE EXTENDED** を使用することができます。(1999 年 4 月)

以下の拡張機能についての詳細説明は、他の資料に収められています。

- **エンタープライズ・システム接続アーキテクチャー (ESCON)** では、チャンネルと制御装置の間に光ファイバー通信リンクを使用する新しいタイプのチャンネルが導入されています。情報は、最大距離 60km の範囲内で、2 億ビット/秒の速度によりビット単位で直列転送されます。光ファイバー・テクノロジーとシリアル伝送により、ケーブルリングが単純化され、信頼性が向上します。「*IBM Enterprise Systems Architecture/390 ESCON I/O Interface*」(SA22-7202) を参照してください。
- **ESCON チャンネル間アダプター (ESCON CTCA)** は、並列入出力インターフェース・チャンネル・パス用に使用できるものと同じタイプの機能を、シリアル・チャンネル・パス用に提供します。「*IBM Enterprise Systems Architecture/390 ESCON Channel-to-Channel Adapter*」(SA22-7203) を参照してください。
- **入出力装置自己記述** では、装置が、装置自体およびその位置を入出力構成内で記述することができます。「*IBM Enterprise Systems Architecture/390 Common I/O-Device Commands and Self Description*」(SA22-7204) を参照してください。
- **圧縮機能** は、圧縮および拡張操作の前にプログラムで用意しておく静的 (非適応型) デイクショナリーを使用して、Ziv-Lempel タイプの圧縮と拡張を行います。デイクショナリーが静的であるため、圧縮機能は、長い順次データ・ストリーム (例えば、アーカイブ・データやネットワーク・データなど) だけでなく、ランダムにアクセスされる短いレコード (例えば、80 バイト) の場合にも、優れた圧縮能力を提供します。「*IBM Enterprise Systems Architecture/390 Data Compression*」(SA22-7208) を参照してください。(1993 年 2 月)
(z/Architecture の **COMPRESSION CALL** 命令に

ついては、本書で説明されています。ただし、概説情報、およびデイクショナリー形式に関する情報を提供しているのは、現時点では SA22-7208 のみです。)

ESA/390 のその他の拡張機能には以下のものがありますが、これらについては詳細説明は提供されていません。

- **内蔵の暗号化機構** は、データのプライバシーを保護するため、メッセージ認証および個人識別をサポートするため、および鍵管理を容易にするための幾つかの命令を提供します。この機構の高性能な暗号化能力は、金融トランザクション環境および大量暗号化環境用に設計されており、**Data Encryption Standard (DES)** に準拠しています。
 - 暗号化機構は仮想計算機環境で利用できるように拡張されており、その結果、VM/ESA の下で実行している MVS/ESA で使用できるようになりました。この VM/ESA は、他の VM/ESA の下または論理区画の中で実行しても構いません。(1991 年 9 月)
- **外部時刻参照機能** は、TOD クロックと外部時刻参照 (ETR) の同期を開始および維持するための手段を提供します。数マイクロ秒以内の同期誤差を達成でき、うるう秒の影響も計算に入れられます。この機能は、ETR 送信装置 (Sysplex Timer) (これは二重化が可能です)、複数の ETR 受信装置、および光ファイバー・ケーブルから成っています。ケーブルは、ETR 送信装置 (外部装置) を、構成の ETR 受信装置に接続するために使用されます。TOD クロックを ETR 送信装置が提供する値に設定するための CPU 命令が提供されています。
 - **ETR 伝搬遅延自動調整** 機能は、ETR から接続されているプロセッサに送られる時刻信号を調整して、プロセッサへのケーブル上で生じる伝搬遅延を補正することにより、長さの異なるケーブルを使用できるようにします。(1991 年 9 月)
 - **ETR 外部時刻ソース** 機能は、電話回線や無線を介してリモート・ロケーションから受信した時刻信号に ETR を同期させます。(1991 年 9 月)
- **拡張ソート** は、DB2 ソート機能のパフォーマンスを高める命令を提供します。
- 本書の第 4 版以降に記述されているもの以外の他の **PER 拡張機能** としては、PER 2 の能力が拡張され、解釈実行モードで使用できる追加の PER 機能が提供されるようになっています。

- **チャンネル・サブシステム呼び出し** は、入出力構成の管理のために使用できる各種の機能を提供します。この種の機能には、アクセス可能な構成エレメントから構成に関する情報を取得するものや、構成を動的に変更するものなどがあります。
- **操作拡張機能** により、以下のように他の幾つかの改善が加えられ、可用性と使いやすさが向上しています。
 - **自動再構成** により、LPAR 区画内のオペレーティング・システムは、即時に終了することを自身に対して宣言することができます。これは、通常、オペレーティング・システムの記憶域および CPU リソースを隣接区画が獲得して、失敗した他のシステムの作業負荷を動的に吸収できるようにするためです。その他の機能が、指定された関連区画を非活動化およびリセットします。
 - 新しい **記憶装置再構成** コマンドでは、サービス・プロセッサとの 1 回の通信により複数の記憶装置再構成要求を実行することができます。これにより、再構成に必要な時間が短縮されます。
 - **SCP 開始リセット** により、システム制御プログラム (SCP) は、何らかのチェック条件の後で割り込み禁止待ち状態に入る前に、入出力構成をリセットすることができます。
 - **コンソール統合** は、MVS が必要とするコンソールの数を 1 つ減らすことにより、構成要件を単純化します。
 - **プロセッサ可用性機能** では、いずれかの CPU にチェック停止を引き起こすようになりカバリー不能エラーが発生したときに、その CPU が自己の状態を保管し、構成内の他の CPU に警報を出すことができます。したがって、多くの場合、失敗した CPU で実行されていたプログラムの実行を、他の CPU が続行することができます。この機能は、ESA/390 モードと LPAR モードのどちらでも使用できます。(1991 年 4 月)
- **仮想計算機に関する拡張機能** により、以下のように、解釈実行機能に対する幾つかの改善が加えられています。
 - **VM データ・スペース機能** により、仮想計算機アプリケーションの中で ESA/390 アクセス・

レジスター・アーキテクチャーを従来より便利に利用できるようになります。この機能は、大量データをアドレッシングするため、およびデータを共用するための能力を向上させます。

VM/ESA が VM データ・スペース機能をどのように利用するかについては、「*VM/ESA CP Programming Services*」(SC24-5520) を参照してください。

- 新しい **記憶キー機能** は、従来使用されていた RCP 領域を不要にすることにより、パフォーマンスを向上させます。
 - 解釈済み **SIE** (領域再配置で使用可能) が改善されて、VM 自体がハイパフォーマンス・ゲストとして稼働しているときに、VM の下での優先ゲストを指定できるようになりました。
 - その他の改善事項としては、一部のゲスト状態情報を検索する特殊索引 (オプション) が加わったほか、インプリメンテーション選択の際の自由度が拡大されています。
 - **ESCON-multiple-image facility (EMIF)** では、複数の論理区画が ESCON チャンネル (および FICON チャンネル) を共用できるほか、オプションとして、それらの共用チャンネルに対して構成されている制御装置および関連の入出力装置も共用することができます。その結果、チャンネルの必要数が減少し、したがって、チャンネル使用状況と入出力接続が効率化されます。(1992 年 6 月)
 - **PR/SM LPAR** モードが拡張されて、単一イメージ構成では最大 10 個、そして物理分割構成では最大 20 個の論理区画が使用できるようになりました。これ以前は、それぞれ 7 個と 14 個に制限されていました。(1992 年 6 月)
- z/Architecture の登場に伴い、PR/SM LPAR モードでは 15 個の論理区画が使用できるようになり、物理分割はサポートされなくなりました。
- **結合機能** は、この機能を使用して接続されている MVS/ESA システム間にハイパフォーマンスなデータ共用を実現します。結合機能により、共用バッファ内のデータのキャッシング、共用リスト内での作業キューおよび状況情報の維持、および共用ロック制御によるデータのロックを目的として、記憶域を動的に分割することができます。MVS/ESA サービスは、結合機能の内容のアクセスおよび操作のための機能を提供します。(1994 年 4 月)

ESA/370 および 370-XA のベース

ESA/390 のベースには、ESA/370 のすべての機能が含まれています。このセクションでは、システム/370 と比較して 370-XA で追加された機能、および 370-XA と比較して ESA/370 で追加された機能のほとんどについての概要を、簡単に説明します。

370-XA で新規に追加された CPU 関連機能は、以下のとおりです。

- 2 モード・アドレッシングは、2 つの操作モードを提供します。それは、旧プログラムを実行するための 24 ビット・アドレッシング・モードと、31 ビット・アドレッシング・モードです。
- 31 ビット論理アドレッシングは、仮想アドレス・スペースを、24 ビット・モードでアドレッシング可能な 16M バイトから 2G バイト (2,147,483,648 バイト) に拡張します。
- 31 ビット実絶対アドレッシングは、最大 2G バイトの主記憶域のアドレス可能度を提供します。
- 370-XA の保護機能には、4K バイト・ブロックのみに対するキー制御の保護、ページ保護のほか、システム/370 と同じ 512 以下のアドレスに対する低アドレス保護が含まれています。取り出し保護オーバーライドは、記憶位置 0-2047 に対する取り出し保護を無効にします。
- トレース機能では、記憶域内の重要イベントを継続的に記録できます。これはシステムの問題を判別するときに役立ちます。
- COMPARE AND FORM CODEWORD 命令および UPDATE TREE 命令は、アプリケーションのソースを容易にします。
- 解釈実行機能を使用すると、幾つかのアーキテクチャーに従って稼働できる仮想計算機を作成できます。このような仮想計算機では、多くの仮想計算機機能がプログラムによりシミュレートされるのではなく直接解釈されるため、パフォーマンスが向上します。この機能については、「IBM 370-XA Interpretive Execution」(SA22-7095) で説明されています。
- サービス呼び出し論理プロセッサ (SCLP) 機能は、構成を記述および変更するために、制御プログラムとサービス・プロセッサの間で通信する手段を提供します。この機能については説明されていません。

370-XA チャンネル・サブシステムが採用された結果、370-XA と System/370 の間には以下のような入出力関連の相違があります。

- 入出力装置のパス独立アドレッシング。これは、どの CPU が入出力命令を実行しているか、または入出力装置がチャンネル・サブシステムにどのように接続しているかに関係なく、入出力操作を開始できるようにします。どの入出力割り込みも、その割り込みに対して使用可能にされているすべての CPU により実行することができます。
- パス管理。チャンネル・サブシステムは、これを使用して、どのパスが選択可能かを判別し、パスを選択し、そして、関連装置による入出力処理を開始しようとしたときに生じる使用中状態を管理することができます。
- 動的再接続。この機能を使用する入出力装置は、アクセス権を持っている任意の使用可能なチャンネル・パスに再接続して、コマンドのチェーンの実行を続けることができます。
- プログラマブル割り込みサブクラス。個々の入出力装置からの入出力割り込み要求を、8 つのマスク可能割り込みキューのうちの任意の 1 つに、プログラムにより割り当てることができます。
- チャンネル・プログラム内で 31 ビット・アドレスを直接使用するための、追加の CCW フォーマット。この新しい CCW フォーマットはフォーマット 1 と呼ばれるもので、システム/370 の CCW フォーマット (現在はフォーマット 0 と呼ばれています) に追加する形で提供されています。
- アドレス限界検査は、新しい記憶保護機能を提供するもので、指定した絶対アドレスより上または下の記憶位置へのデータ・アクセスを阻止します。
- モニター機能。プログラムがこの機能呼び出すと、チャンネル・サブシステムは、重要な入出力リソース使用率パラメータを測定し、累算します。
- 状況検査機能は、装置が示している装置状況ビットの不適切な組み合わせを報告します。
- 13 個の入出力命令のセットと、それに関連した制御ブロック。これは、チャンネル・サブシステムを制御するためのものです。

ESA/370 で新規に追加された機能には、以下のものがあります。

- 16 個のアクセス・レジスタ。プログラムは、これらのレジスタを使用して、そのプログラム自体が常駐しているアドレス・スペースも含めて、最大 16 個の 2G バイト・アドレス・スペース内の記憶

域オペランドに即時にアクセスできます。アクセス・レジスター・モードと呼ばれる動的アドレス変換モードでは、命令の B フィールド (命令によっては R フィールド) に、汎用レジスターとアクセス・レジスターの両方を指定し、保護テーブルの内容およびアクセス・レジスターの内容には、アクセスするオペランド・アドレス・スペースを指定します。許可メカニズムの制御下にあるプログラムは、アクセス・レジスターの内容を変更することにより、数百もの異なるオペランド・アドレス・スペースに高速でアクセスできます。

- リンケージ・スタックは、同一または異なるアドレス・スペース内のプログラム相互間で制御を受け渡すために、機能拡張されたメカニズムで使用されます。このメカニズムでは、以前からあった PROGRAM CALL 命令、拡張エントリー・テーブル・エントリー、および新しい PROGRAM RETURN 命令も使用します。このメカニズムは、呼び出しリンケージ時に、アクセス・レジスターや汎用レジスターの内容など各種の状況エレメントを保管し、その呼び出しリンケージ中に現在の状況を変更し、そして、戻りリンケージ時に、元の状況を復元します。リンケージ・スタックは、新しい BRANCH AND STACK 命令により実行されるブランチ・タイプのリンケージ時に、アクセス・レジスターおよび汎用レジスターの内容を保管するためにも使用できます。
- ホーム・スペース・モードと呼ばれる変換モードでは、制御プログラムは、ホーム・アドレス・スペースと呼ばれるアドレス・スペース内で効率的に制御を獲得することができます。このアドレス・スペースには、ディスパッチ可能単位 (タスクまたはプロセス) 用の基本制御ブロックが保持されていません。
- 準特権の MOVE WITH SOURCE KEY および MOVE WITH DESTINATION KEY を使用すると、異なる記憶キーを持つ記憶域相互間でデータを両方向に移動することができ、PSW キーを変更する必要はありません。
- 特権の LOAD USING REAL ADDRESS および STORE USING REAL ADDRESS 命令を使用すると、制御プログラムは、実記憶域内のデータに効率的にアクセスすることができます。
- 専用スペース機能を使用すると、アドレス・スペースに共通セグメントが含まれないようにして、低アドレス保護および取り出し保護オーバーライドがそのアドレス・スペースに適用されないようにすることができます。

- 非特権の MOVE PAGE 命令を使用すると、プログラムは、主記憶域と拡張記憶域の間でデータ・ページを移動することができます (ただし、ソース・ページと宛先ページが両方とも有効である場合に限られます)。MOVE PAGE の制御プログラム・サポート用の手段については、一部の詳細情報は提供されていません。ESA/370 バージョンの MOVE PAGE は、現在はページ移動機能 1 と呼ばれています。
- Processor Resource/Systems Manager (PR/SM) フィーチャーは、VM/XA の下での複数の優先ゲストに対するサポートを提供し、また、論理分割 (LPAR) モードを提供します。LPAR モードでは、複数の論理区画間でプロセッサ・リソースを柔軟に分割することができます。LPAR での PR/SM 使用の一部のアスペクトについては、「IBM ES/3090 Processor Complex Processor Resource/Systems Manager Planning Guide」(GA22-7123) に説明があります。
- COMPARE UNTIL SUBSTRING EQUAL 命令を使用すると、IMS ログ・データ・セットの圧縮のパフォーマンスが向上します。また、この命令は、他のプログラムの中でも便利に使用できます。

システム・プログラム

z/Architecture は、制御プログラムと共同して働くように設計されています。制御プログラムは、システム・リソースの使用を調整し、すべての入出力命令を実行し、例外条件を取り扱い、複数のプログラムのスケジューリングと実行を監視します。

互換性

z/Architecture システム間の互換性

z/Architecture で定義されているとおりに作動する複数のシステム相互間で、インプリメンテーションや物理的な能力が異なっていることはありますが、論理的には、これらのシステム間には下位互換性および上位互換性があります。互換性があれば、教育の手間が省け、システム・バックアップの可用性が高まり、システムの拡張も容易になります。特に、z/Architecture 用に書かれたプログラムは、下記の条件を満たしていれば、どのような z/Architecture インプリメンテーションにおいても同じ結果をもたらします。

1. 時間依存性がないこと。

2. 特定のシステム機能 (記憶容量、入出力装置、またはオプション機能など) が構成に含まれていないときに、その機能が存在することを前提とするプログラムではないこと。
3. 特定のシステム機能が構成に含まれているときに、その機能が存在しないことを前提とするプログラムではないこと。例えば、プログラムは、一部のモデルにはインストールされていない命令コードまたはコマンド・コードを使用した場合に発生する割り込みに依存するものであってはなりません。また、インストールされていない機能に関連したフィールドを使用する (またはそれに依存する) ものであってはなりません。例えば、他のモデルで固定ログアウト情報用に使用されている領域に、データを入れてはなりません。同様に、プログラムは、そのプログラムが使用するものとして明示的に使用可能にされていないマシン・フォーマット (制御レジスタ、命令フォーマットなど) の未割り当てフィールドを使用する (またはそれに依存する) ものであってはなりません。
4. 予測不能またはモデルに依存するものとして定義されている結果または機能、あるいは未定義として識別されている結果または機能に依存していないこと。これには、プログラムが装置番号および CPU アドレスの割り当てに依存してはならないという条件も含まれます。
5. 特定のモデル用の機能特性を記載した資料の中でこのアーキテクチャーから逸脱するものとして定義されている結果または機能に依存していないこと。
6. アーキテクチャーに対して加えられた変更のうち、互換性に影響を与えるものと認められる変更についての配慮がなされていること。

z/Architecture と ESA/390 の間の互換性

制御プログラムの互換性

ESA/390 用に使われた制御プログラムを、z/Architecture の定義に従って作動するシステムにそのまま移行することはできません。これは、汎用レジスタおよび制御レジスタのサイズ、PSW のサイズ、割り当てられる記憶位置、および動的アドレス変換が異なるためです。

問題プログラム状態の互換性

問題プログラム状態レベルでは、ESA/390 から z/Architecture に移行する場合の高度な互換性が確保されています。ユーザーのアプリケーションの大半は問題プログラム状態を想定して書かれているので、この問題プログラム状態の互換性は多くのインストール・システムで役立ちます。

ESA/390 用に使われた問題プログラム状態プログラムは、以下の条件を満たしていれば、z/Architecture でも稼働します。

1. 1-14ページの『z/Architecture システム間の互換性』に示されている制限事項に従っていること。
2. システムで使用できない制御プログラム機能に依存していないこと。

プログラミング上の注意: 本書では、各種の命令コード、命令内のビット位置、チャンネル・コマンド・ワード、レジスタ、テーブル・エントリ、および、記憶域内の低位 512 バイトおよびバイト 4096-8191 の中の固定位置に、それぞれ意味を割り当てています。特に断り書きがない限り、その他の命令コード、ビット位置、および低位記憶位置は、将来、新しい機能の追加や、アーキテクチャーのその他の拡張を行う際に割り当てることができるように、予約されています。

このような新規機能をインストールしたときに既存のプログラムが稼働できるようにするためには、プログラムは、現在の検索対象として定義されている値が無効になった結果として例外が指示されることを前提としてはなりません。検査されない未割り当て位置に値を入れる必要がある場合は、プログラムはそこに 0 を入れる必要があります。マシン提供のコードまたはフィールドの場合、プログラムでは、将来新しいコードおよびビットが割り当てられる可能性があるということを考慮に入れる必要があります。プログラムでは、未割り当ての低位記憶位置を情報保持のために使用してはなりません。これは、将来の割り当てによっては、マシンがこれらの記憶位置の内容を変更する可能性があるためです。

可用性

可用性とは、システムが個々のジョブを受け入れて正常に処理できる能力のことです。z/Architecture に従って稼働するシステムは、次の 2 つの能力により高度な可用性を達成します。(1) 広範囲にわたる多数のジョブを同時に処理できるため、システムは任意の特定ジョブに迅速にアクセスできます。(2) エラーの影響が限定され、エラーの原因が厳密に識別されるため、エラーの影響を受けるジョブの数が最小限に抑えられ、容易にエラーを訂正できます。

この能力は、以下のような設計上のアスペクトにより達成されています。

- プログラムの実行中に、命令およびデータの正確さが検査され、プログラム・エラーは装置エラーと区別して示されます。このような検査と報告は、障害の発生場所を突き止め、その影響を隔離するために役立ちます。
- 保護機能では、動的アドレス変換の機能、およびプログラムとデータをそれぞれ異なるアドレス・スペースに分離する機能を利用して、プログラムが無許可または誤った保管または取り出し操作を行ったことが原因で記憶域の内容が消滅したり誤用されたりする事態を防止します。これにより、ユーザーにとってのセキュリティが強化され、したがって、セキュリティ要件の異なるアプリケーションを他のアプリケーションと同時に処理することができます。
- 動的アドレス変換機能の働きにより、個々のアプリケーションを互いに他と分離させながら、同時に共通リソースを共用させることができます。また、この機能を利用して仮想計算機をインプリメントすることにより、新バージョンのオペレーティング・システムの設計とテストを行いながら、並行してアプリケーション・プログラムの処理を進めることもできます。さらに、この機能は、互換性のない複数のオペレーティング・システムの並行稼働も可能にします。
- アクセス・レジスターを使用することにより、プログラム、データ、および異なるデータ・コレクションを、異なるアドレス・スペースに配置することができます。このようにすれば、誤ったアドレスを使用した格納操作が原因で、誤った結果またはシステム全体にわたる障害が発生する可能性が減少します。
- マルチプロセッシングおよびチャンネル・サブシステムにより、記憶機構と処理能力の利用効率が高まり、CPU 間の直接通信が促進され、リソースの重

複が実現するため、マシン障害が発生してもシステムの継続稼働が可能になります。

- **MONITOR CALL**、プログラム・イベント記録、およびタイミング機能を使用すれば、プログラムのテストとデバッグを行う際に、手操作により介入する必要がなく、また並行処理されている他のプログラムへの影響もほとんどありません。
- ほとんどのモデルでは、主記憶域内でのエラー検出および訂正 (ECC)、CPU 再試行、およびコマンド再試行により、断続的な装置の誤動作を回避して、装置障害の発生回数を減らすことができます。
- 高度なマシン・チェック処理メカニズムにより、各モデル固有のものではない障害が分離されるので、未訂正のエラーの影響を受けるプログラムの数が減少します。さらに、このメカニズムは、モデルから独立してマシン状況情報を記録する機能も備えています。したがって、モデル間でのマシン・チェック処理の互換性が増大し、また、システム障害が発生したときに、異なるモデルにプログラムをロードして稼働させる能力も強化されます。
- 基本的なシステム動作に必要な手動制御の数は少数に抑えられており、オペレーターとシステムの間での対話の大部分は、入出力装置として働く 1 つの装置を介して行われるので、オペレーター・エラーの発生回数が減少します。
- **PR/SM** フィーチャーにより論理区画を使用可能にできるため、1 つ以上の区画で新規プログラミング・システムのテストを行いながら、他の区画では信頼性の高い実動処理を続行することができます。これは、特に非 VM 環境用の拡張機能です。
- **ESA/390** および **z/Architecture** の操作上の拡張機能およびチャンネル・サブシステム呼び出し機能により、システムに事故が起きたときにアプリケーション・プログラムの実行を続ける能力、および、操作を妨害せずに構成変更を行う能力がさらに増大します。

第2章 編成

主記憶機構	2-2	アクセス・レジスタ	2-4
拡張記憶機構	2-2	暗号化機構	2-6
CPU	2-2	外部時刻参照	2-6
PSW	2-3	入出力	2-6
汎用レジスタ	2-3	チャンネル・サブシステム	2-6
浮動小数点レジスタ	2-3	チャンネル・パス	2-6
浮動小数点制御レジスタ	2-4	入出力装置と制御装置	2-7
制御レジスタ	2-4	オペレーター機能	2-7

システムは、論理的には、主記憶機構、1つ以上の中央演算処理装置 (CPU)、オペレーター機能、チャンネル・サブシステム、および入出力装置から成っています。入出力装置は、制御装置を介してチャンネル・サブシステムに接続されます。チャンネル・サブシステムと制御装置の間の接続をチャンネル・パスと呼びます。

チャンネル・パスは、パラレル伝送プロトコルまたはシリアル伝送プロトコルのいずれかを使用するので、それに応じて、パラレル・チャンネル・パスまたはシリアル・チャンネル・パスと呼ばれます。シリアル・チャンネル・パスは、動的スイッチを介して制御装置に接続することができます。動的スイッチは、スイッチのポート間に異なる内部接続を提供します。

システムでは拡張記憶機構も使用でき、CPUには暗号装置を含めることができ、また、システムに外部時刻参照 (ETR) を接続することができます。

上記の機能の物理的な区別は、「モデル」と呼ばれるインプリメンテーションによって異なります。図2-1は、2つのCPUを備えたマルチプロセッシング・システムの論理構造を描いたものです。このシステムは、拡張記憶機構と暗号装置を備え、ETRに接続されています。

個々のプロセッサは、それぞれ、内部特性、インストールされている機能、チャンネル・サブシステムに接続できるサブチャンネル、チャンネル・パス、および制御装置の数、主記憶機構および拡張記憶機構のサイズ、および、オペレーター機能の表現が異なる場合があります。

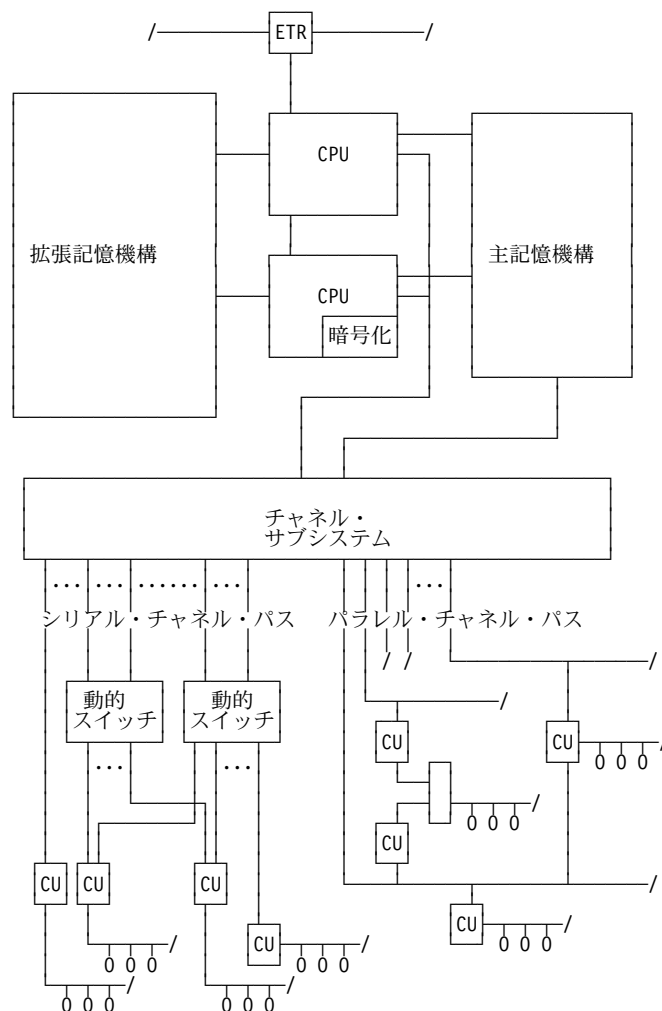


図 2-1. 2つのCPUを持つArchitectureシステムの論理構造

入出力装置を除外して見たシステムを、構成と呼びます。構成に含まれているものもそうでないものも含めたすべての物理装置の集合を、インストール・システムと呼びます。

モデルごとに異なる再構成制御機能があり、これによって、構成内の主記憶機構と拡張記憶機構の容量、および CPU とチャンネル・パスの数を変更することができます。場合によっては、再構成制御を使用して、1 つの構成を複数の構成に分割できることもあります。このようにして再構成された各構成は、それぞれ同じ構造を備えています。つまり、主記憶機構と拡張記憶機構、1 つ以上の CPU、そして、チャンネル・サブシステム内の 1 つ以上のサブチャンネルとチャンネル・パスから成っています。

各構成は互いに分離されており、1 つの構成内の主記憶機構および拡張記憶機構を、他の構成内の CPU およびチャンネル・サブシステムが直接アドレッシングすることはできません。ただし、共用入出力装置やチャンネルアダプターを媒介として、1 つの構成が他の構成と通信することはできます。任意の時点で 1 つのシステムの中で相互に接続されている記憶機構、CPU、サブチャンネル、およびチャンネル・パスは、構成に含まれているものと見なされます。個々の CPU、サブチャンネル、チャンネル・パス、主記憶位置、および拡張記憶位置は、時点では 1 つの構成のみに属することができます。

主記憶機構

直接アドレッシングが可能な主記憶機構は、CPU およびチャンネル・サブシステムによる高速データ処理の手段を提供します。データおよびプログラムを処理できるようにするには、その両方を入力装置から主記憶機構にロードしておく必要があります。システムで使用できる主記憶機構の容量はモデルによって異なり、また、モデルによっては、そのモデル固有の構成制御機能により構成内の容量が制御されることもあります。記憶域は、4K バイト・ブロックの倍数を単位として使用できます。どの時点でも、構成内のチャンネル・サブシステムおよびすべての CPU が、同時に同じ記憶域ブロックにアクセスし、同じ絶対アドレスを使用して主記憶機構位置の特定ブロックを参照することができます。

主記憶機構には高速アクセス・バッファ記憶域が含まれていることがあり、これはキャッシュとも呼ばれています。各 CPU にそれぞれ 1 つずつキャッシュを割り当てることができます。パフォーマンスの点を除けば、物理的な構造や異なるの記憶メディアの使用によって、プログラムが影響を受けることはありません。

拡張記憶機構

一部のモデルでは、拡張記憶機構を利用できます。拡張記憶機構が使用可能な場合、構成内のすべての CPU は、命令を使用して、4K バイト単位のデータ・ブロックを拡張記憶機構から主記憶機構へ、また主記憶機構から拡張記憶機構へ転送することができます。転送に使用する命令は PAGE IN 命令と PAGE OUT 命令で、これについては、第10章、『制御命令』で説明します。

拡張記憶機構の 4K バイト・ブロックは、拡張記憶機構ブロック番号と呼ばれる 32 ビットの符号なし 2 進整数を使用してアドレッシングされます。

CPU

中央演算処理装置 (CPU) は、システムの制御の中核です。CPU には、命令実行、割り込みアクション、タイミング機能、初期プログラム・ローディング、およびその他のマシン関連機能を順序付けして処理するための機能が含まれています。

CPU の物理的なインプリメンテーションはモデル間で異なる場合がありますが、論理的な機能は同じです。プログラムが互換性の規則に従っていれば、命令の実行結果はどのモデルでも同じです。

CPU は、命令を実行するときに、固定長の 2 進整数と浮動小数点数 (2 進数または 16 進数)、可変長の 10 進整数、および、固定長または可変長の論理情報を処理できます。処理は、並列の場合と直列の場合があります。処理エレメントの幅、シフト・パスの多重度、および、異なるタイプの演算を同時に実行できる程度は、CPU のモデルによってこととなりますが、論理的な結果は同じです。

CPU が実行する命令は 7 つのクラスに分けられます。それは、一般命令、10 進数命令、浮動小数点数サポート命令 (FPS)、2 進浮動小数点数命令 (BFP)、16 進浮動小数点数命令 (HFP)、制御命令、入出力命令です。一般命令は、2 進整数算術演算、論理演算、ブランチ演算、およびその他の非算術演算を行うために使用されます。10 進数命令は、10 進数フォーマットのデータを操作します。BFP および HFP は、それぞれ BFP フォーマットおよび HFP フォーマットのデータを操作します。FPS 命令は、フォーマットに関係なく浮動小数点数データを操作するほか、データを 1 つのフォーマットから別のフォーマットに変換する操作も行います。特権制御命令および入出力命令は、CPU が監視プログラム状態にあるときのみ実行できます。準特権制御命令は、該

当する許可メカニズムが許す範囲内で、問題プログラム状態で実行できます。

CPU は、プログラムでは使用できるが、主記憶機構内にはアドレッシング可能な表現を持たないレジスターを提供します。この種のレジスターには、現行プログラム状況ワード (PSW)、汎用レジスター、浮動小数点レジスター、制御レジスター、アクセス・レジスター、プレフィックス・レジスター、そして、クロック・コンパレーターおよび CPU タイマー用のレジスターがあります。インストール・システム内の各 CPU は、そのインストール・システム内のすべての CPU が共用する時刻 (TOD) クロックにアクセスできます。特定の操作の中でどのタイプのレジスターを使用するかは、命令で使用されている命令コードによって決まります。制御レジスター、アクセス・レジスター、汎用レジスター、および浮動小数点レジスターのフォーマットについては、2-5 ページの図2-2 を参照してください。

PSW

プログラム状況ワード (PSW) には、命令アドレスや条件コードのほか、命令の順序を制御したり CPU の状態を判別したりするために使用するその他の情報が含まれています。アクティブな PSW、つまり制御効力を持っている PSW を、「現 PSW」と呼びます。現 PSW は、現在実行されているプログラムを制御します。

CPU には割り込みの機能があり、例外条件や外部からの刺激が生じたときに、この機能を使用して迅速に他のプログラムに切り替えることができます。割り込みが発生すると、CPU は、その割り込みのクラス用として割り当てられている記憶位置 (「旧 PSW 位置」と呼ばれます) に現 PSW を入れます。そして、CPU は、第 2 の割り当て済み記憶位置から新しい PSW を取り出します。この新しい PSW によって、次に実行するプログラムが決まります。割り込みの処理が終了すると、割り込んだプログラムは旧 PSW を再ロードして、再びその PSW を現 PSW にすることができます。これにより、割り込まれたプログラムは実行を継続できるようになります。

割り込みには、外部割り込み、入出力割り込み、マシン・チェック割り込み、プログラム割り込み、再始動割り込み、監視プログラム呼び出し割り込みの 6 つのクラスがあります。各クラスについて、それぞれ専用の旧 PSW 位置と新 PSW 位置のペアが、実記憶域の中に永続的に割り当てられます。

汎用レジスター

命令では、16 個の汎用レジスターうちの 1 つ以上に含まれている情報を指定できます。汎用レジスターは、アドレス演算で基底アドレス・レジスターおよび指標レジスターとして、また、一般的な算術演算や論理演算でアキュムレーターとして使用できます。各レジスターは、それぞれ 64 個のビット位置を含むことができます。汎用レジスターは 0 から 15 までの番号で識別され、命令の中の 4 ビットの R フィールドに指定されます。命令によっては、複数 R フィールドを使用して複数の汎用レジスターをアドレッシングできることもあります。また、特定の汎用レジスターが、R フィールドにより明示指定されるのではなく暗黙に指定される命令もあります。

一部の演算では、2 つの隣接する汎用レジスターのビット 32-63 またはビット 0-63 を結合して、それぞれ 64 ビットまたは 128 ビット・フォーマットで使用される場合があります。この種の演算では、プログラムは偶数番号のレジスターを指定する必要があります。このレジスターには、左側 (高位) の 32 ビットまたは 64 ビットが含まれます。その次に大きい番号のレジスターには、右側 (低位) の 32 ビットまたは 64 ビットが含まれます。

16 個の汎用レジスターのうち 15 個は、一般的な算術演算や論理演算でアキュムレーターとして使用されるほかに、アドレス生成の際に基底アドレス・レジスターおよび指標レジスターとしても使用されます。その場合は、レジスターは、命令の中の 4 ビットの B フィールドまたは X フィールドに指定されます。B フィールドまたは X フィールドの値が 0 である場合は、基底または指標を適用しないことを意味します。したがって、汎用レジスター 0 を、基底アドレスまたは指標を含むものとして指定することはできません。

浮動小数点レジスター

浮動小数点数命令 (FPS、BFP、HFP) は、すべて同じ浮動小数点レジスターを使用します。CPU は 16 個の浮動小数点レジスターを備えています。浮動小数点レジスターは 0 から 15 までの番号で識別され、浮動小数点数命令の中の 4 ビットの R フィールドに指定されます。浮動小数点レジスターは、どれも長さは 64 ビットで、短精度 (32 ビット) または長精度 (64 ビット) のどちらの浮動小数点オペランドでも含むことができます。2-5 ページの図2-2 に示すように、浮動小数点レジスターをペアで使用して、拡張オペランド (128 ビット) を指定することもできます。8 つのペアは、それぞれ、番号の小さい方のレジスターの番号で参照されます。

浮動小数点制御レジスター

浮動小数点制御 (FPC) レジスターは、マスク・ビット、フラグ・ビット、データ例外コード、および丸めモード・ビットを含む 32 ビット・レジスターです。FPC レジスターについては、19-2ページの『浮動小数点制御 (FPC) レジスター』で説明します。

制御レジスター

CPU は 16 個の制御レジスターを備えており、それぞれ 64 個のビット位置があります。レジスター内のビット位置は、システムの特長機能 (例えばプログラム・イベント記録) に割り当てられており、特定の操作が可能であることを指定するため、またはその機能に必要な特殊情報を供給するために使用されます。

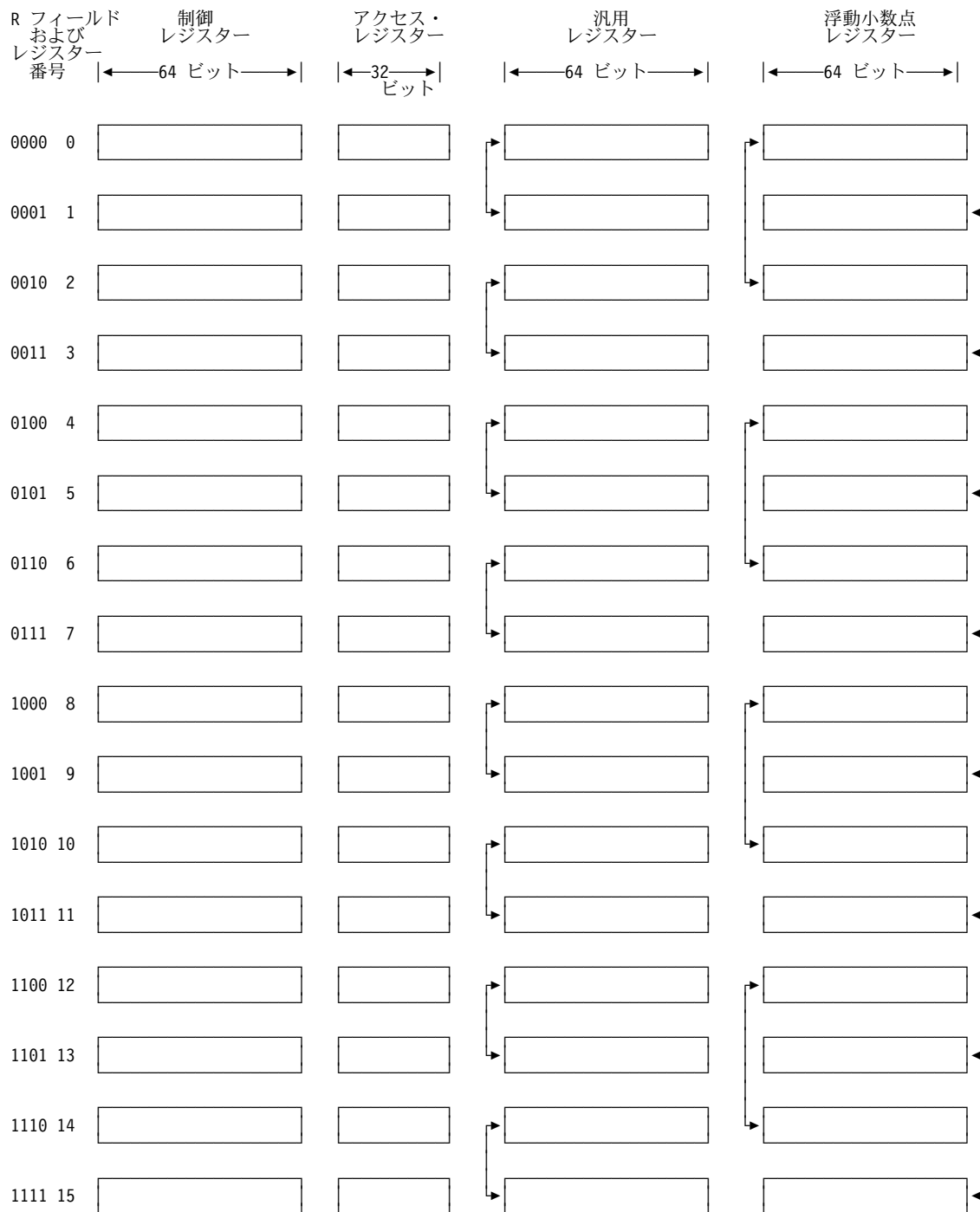
制御レジスターは 0 から 15 までの番号で識別され、LOAD CONTROL 命令および STORE CONTROL 命令の中の 4 ビットの R フィールドに指定されます。これらの命令では、それぞれ複数の制御レジスターをアドレッシングできます。

アクセス・レジスター

CPU は、0 から 15 までの番号が付いた 16 個のアクセス・レジスターを備えています。1 つのアクセス・レジスターは 32 個のビット位置から成り、アドレス・スペース制御エレメントの間接指定を含んでいます (間接指定についてはここでは詳しく説明しません)。アドレ

ス・スペース制御エレメントは、動的アドレス変換 (DAT) メカニズムが、対応するアドレス・スペースに対する参照を変換するために使用するパラメーターです。CPU が、アクセス・レジスター・モード (PSW 中のビットにより制御される) と呼ばれるモードになっているときは、命令の B フィールド (記憶域オペランド参照用の論理アドレスを指定するために使用される) が、アクセス・レジスターを指定します。そして、DAT は、そのアクセス・レジスターに指定されているアドレス・スペース制御エレメントを、現在行われている参照用で使用します。一部の命令では、B フィールドの代わりに R フィールドが使用されます。アクセス・レジスターの内容をロードおよび格納するための命令や、アクセス・レジスターの内容を別のアクセス・レジスターに移動するための命令が用意されています。

アクセス・レジスター 1-15 はどれも、現行命令スペース (1 次アドレス・スペース) も含めて任意のアドレス・スペースを指定できます。アクセス・レジスター 0 は、常に現行命令スペースを指定します。アクセス・レジスター 1-15 の 1 つを使用してアドレス・スペースが指定されている場合、CPU は、そのアクセス・レジスターの内容を変換することにより、どのアドレス・スペースが指定されているのかを判別します。アクセス・レジスター 0 を使用してアドレス・スペースが指定されている場合は、CPU は、そのアクセス・レジスターが現行命令スペースを指定しているものと見なし、アクセス・レジスターの実際の内容を調べることはしません。したがって、16 個のアクセス・レジスターは、任意の一時点において、現行命令スペースと、最大 15 個のその他のスペースを指定できることになります。



注: 矢印は、2つのレジスターを結合してダブルレジスター・ペアにできることを示しています。その場合は、番号の小さい方のレジスターをR フィールドに指定します。例えば、浮動小数点レジスター 13 と 15 のペアを指定するには、R フィールドに 2 進数 1101 を入力します。

図 2-2. 制御レジスター、アクセス・レジスター、汎用レジスター、および浮動小数点レジスター

暗号化機構

モデルによっては、CPU の拡張機構の 1 つとして統合暗号化機構が提供されていることがあります。CPU に備わっている暗号化機構は、CPU の一部として機能します。暗号化機構の利点については、1-11 ページに簡単な説明があります。ほかにはこの機構に関する説明はありません。

外部時刻参照

モデルによっては、外部時刻参照 (ETR) を構成に接続できる場合があります。ETR の利点については、1-11 ページに簡単な説明があります。ほかにはこの機構に関する説明はありません。

入出力

入出力 (I/O) 操作では、主記憶機構と入出力装置の間で情報の転送が行われます。入出力とそれぞれの制御装置はチャンネル・サブシステムに接続され、チャンネル・サブシステムがデータ転送を制御します。

チャンネル・サブシステム

チャンネル・サブシステムは、入出力装置と主記憶機構の間の情報の流れを制御します。これにより、CPU は入出力装置と直接通信する作業から解放され、入出力処理と並行してデータ処理を進めることができるようになります。チャンネル・サブシステムは、入出力装置の間の情報の流れを管理するために、1 つ以上のチャンネル・パスを通信リンクとして使用します。また、この入出力処理の一環として、チャンネル・サブシステムは、チャンネル・パスの可用性の検査、使用可能なチャンネル・パスの選択、および入出力装置との間の操作の実行開始など、パス管理機能を実行します。チャンネル・サブシステムにはサブチャンネルが含まれています。

チャンネル・サブシステムからアクセスできる各入出力装置に、それぞれ専用のサブチャンネルが 1 つずつ提供されます。サブチャンネルには、関連する入出力装置およびその装置とチャンネル・サブシステムの接続に関する情報を格納するための記憶域が備わっています。また、サブチャンネルは、入出力操作に関する情報、および関連入出力装置を必要とするその他の機能に関する情報を格納するための記憶域も備えています。サブチャンネルに含まれている情報には、チャンネル・サブシステムからも、また入出力命令を使用して CPU からアクセスできます。この情報は、関連入出力装置について CPU とチャンネル・サブシステムの間で通信を行うための手段として利用さ

れます。実際に提供されるサブチャンネルの数は、モデルと構成によって異なります。最大サブチャンネル数は、65,536 です。

チャンネル・パス

入出力装置は、制御装置を介し、チャンネル・パスを経由してチャンネル・サブシステムに接続されます。制御装置は複数のチャンネル・パスを介してチャンネル・サブシステムに接続でき、また、1 つの入出力装置を複数の制御装置に接続することもできます。モデルと構成によって異なりますが、1 つのチャンネル・サブシステムが、最大 8 つの異なるチャンネル・パスを使用して 1 つの入出力装置にアクセスできます。1 つのチャンネル・サブシステムが提供するチャンネル・パスの合計数は、モデルと構成によって異なります。最大チャンネル・パス数は、256 です。

チャンネル・パスには、次の 3 つのタイプのいずれかの通信リンクを使用できます。

- 並列入出力インターフェースと呼ばれる、システム/360 入出力インターフェースおよびシステム/370 入出力インターフェース。この場合のチャンネル・パスは、パラレル・チャンネル・パスと呼ばれます。
- 直列入出力インターフェースと呼ばれる、ESCON 入出力インターフェース。この場合のチャンネル・パスは、シリアル・チャンネル・パスと呼ばれます。
- 直列入出力インターフェースとも呼ばれる、FICON 入出力インターフェース。この場合のチャンネル・パスは、シリアル・チャンネル・パスと呼ばれます。

並列入出力インターフェースは、チャンネル・サブシステムと 1 つ以上の制御装置を結ぶ幾つかの電気信号回線から成っています。8 つの制御装置が、1 つの並列入出力インターフェースを共用することができます。1 つの並列入出力インターフェースで、最大 256 個の入出力装置をアドレッシングできます。並列入出力インターフェースについては、「*IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information*」(GA22-6974) で説明されています。

直列入出力インターフェースは、チャンネル・サブシステム、動的スイッチ、および制御装置のうちのいずれか 2 つの間を結ぶ 2 つの光ファイバー伝導体から成っていません。動的スイッチは、複数の直列入出力インターフェースを使用して、同一または複数の異なるチャンネル・サブシステムに、そして複数の制御装置に接続することができます。1 つのチャンネル・パスに接続できる制御装置の数は、チャンネル・サブシステムと動的スイッチの能力によって異なります。直列入出力インターフェースを使用

する 1 つの制御装置に接続できる入出力装置の数は、最大 256 です (制御装置によって異なります)。ESCON 入出力インターフェースについては、「*ESA/390 ESCON I/O Interface*」(SA22-7202) で説明されています。FICON 入出力インターフェースについては、ANSI 規格文書「*Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)*」で説明されています。

入出力装置と制御装置

入出力装置には、プリンター、磁気テープ装置、直接アクセス記憶装置、ディスプレイ、キーボード、通信コントローラー、テレプロセッシング装置、センサー・ベース装置などがあります。多くの入出力装置では、用紙や磁気テープなどの外部メディアが使用されます。また、ディスプレイや通信ネットワークなどのように、電気信号のみを扱う入出力装置もあります。どちらの場合も、入出力装置の動作は制御装置により制御されます。制御装

置は、関連の入出力装置を操作するために必要な論理機能とバッファリング機能を提供します。プログラミングの観点から見れば、制御装置のほとんどの機能は入出力装置の機能に併合されているように見えます。制御装置機能は、入出力装置または CPU に組み込むことも、また独立した制御装置を使用することもできます。

オペレーター機能

オペレーター機能は、オペレーターがマシンを制御するために必要な機能です。オペレーター機能に関連する装置の 1 つにオペレーター・コンソールがありますが、これはプログラムとの通信用の入出力装置としても使用できます。

オペレーター機能の主なものとしては、リセット、消去、初期プログラム・ローディング、開始、停止、変更、表示などがあります。

第3章 記憶機構

記憶機構のアドレッシング	3-2	ASN 第 2 テーブル索引	3-22
情報のフォーマット	3-2	ASN 変換時の例外の認識	3-22
規定境界	3-3	ASN 許可	3-22
アドレスのタイプとフォーマット	3-3	ASN 許可制御	3-22
アドレスのタイプ	3-3	制御レジスター 4	3-22
絶対アドレス	3-3	ASN 第 2 テーブル・エントリー	3-22
実アドレス	3-4	権限テーブル・エントリー	3-23
仮想アドレス	3-4	ASN 許可プロセス	3-23
1 次仮想アドレス	3-4	権限テーブルの索引	3-24
2 次仮想アドレス	3-4	ASN 許可時の例外の認識	3-25
AR 指定仮想アドレス	3-5	動的アドレス変換	3-25
ホーム仮想アドレス	3-5	変換制御	3-27
論理アドレス	3-5	変換モード	3-27
命令アドレス	3-5	制御レジスター 0	3-27
有効アドレス	3-5	制御レジスター 1	3-27
アドレスのサイズと循環	3-5	制御レジスター 7	3-29
アドレス循環	3-6	制御レジスター 13	3-29
記憶キー	3-8	変換テーブル	3-30
保護	3-8	領域テーブル・エントリー	3-30
キー制御保護	3-8	セグメント・テーブル・エントリー	3-31
記憶保護オーバーライド制御	3-9	ページ・テーブル・エントリー	3-32
取り出し保護オーバーライド制御	3-10	変換プロセス	3-33
アクセス・リスト制御保護	3-10	実スペース制御の検査	3-38
ページ保護	3-11	指定タイプ制御の検査	3-38
低アドレス保護	3-11	アドレス・スペース制御エレメントにより	
保護時の抑止	3-12	指定されるテーブル内での検索	3-38
参照記録	3-13	領域テーブル・エントリーにより指定され	
変更記録	3-13	るテーブル内での検索	3-39
プレフィックシング	3-14	ページ・テーブルの検索	3-40
アドレス・スペース	3-15	実アドレスの形成	3-41
異なるアドレス・スペースへの変更	3-16	変換中の例外の認識	3-41
アドレス・スペース番号	3-16	変換索引バッファ	3-41
ASN 変換	3-17	TLB の構造	3-42
ASN 変換制御	3-17	TLB エントリーの形成	3-42
制御レジスター 14	3-17	TLB エントリーの使用	3-43
ASN 変換テーブル	3-18	変換テーブルの変更	3-44
ASN 第 1 テーブル・エントリー	3-18	アドレスの要約	3-46
ASN 第 2 テーブル・エントリー	3-18	変換されるアドレス	3-46
ASN 変換プロセス	3-20	アドレスの取り扱い	3-47
ASN 第 1 テーブル索引	3-21	割り当て済みの記憶位置	3-49

この章では、主記憶機構内での情報の表現形式、アドレッシング、保護、および、参照と変更の記録について説明します。この章でアドレッシングについて説明する事

項には、アドレスのフォーマット、アドレス・スペースの概念、アドレスのタイプ、および、あるタイプのアドレスを別のタイプのアドレスに変換する方法などが含ま

れます。この章の終わりには、永続的に割り当てられている記憶位置のリストを示します。

主記憶機構は、直接アドレッシングできる高速アクセスのデータ記憶機構をシステムに提供します。データおよびプログラムを処理できるようにするには、その両方を入力装置から主記憶機構にロードしておくことが必要です。

主記憶機構は、キャッシュと呼ばれる、小さい高速アクセスのバッファ記憶機構を1つ以上備えていることがあります。各キャッシュは、通常、1つのCPUまたは1つの入出力プロセッサに関連付けられています。パフォーマンスの点を除けば、物理的な構造や異なるの記憶メディアの使用によって、プログラムが影響を受けることはありません。

CPUによるデータの取り出しと格納は、並行して行われているチャンネル・サブシステムの活動や、他のCPUによる同じ記憶位置への同時参照により、影響を受けることはありません。特定の主記憶位置に対する要求が同時に発生した場合は、通常、システムが決定する順序に従ってアクセスが許可されます。ある参照によりその記憶位置の内容が変更された場合、以後の記憶域取り出しでは、新しい内容が取得されます。

主記憶機構には、揮発性のものと不揮発性のものがあります。揮発性の場合、主記憶機構の内容は、電源が切れた後は保存されていません。不揮発性の場合、いったん電源をオフにし再度オンにしても、主記憶機構の内容は元のまま残されています。ただし、これは、電源を切るときに、すべてのCPUが停止状態にあり、主記憶機構に対する参照が何も行われていない場合に限られます。どちらのタイプの主記憶機構の場合も、記憶キーの内容は、電源を切った後もそのまま保存されるとは限りません。

注：本書の説明ではほとんどの場合は仮想記憶機構を対象としているので、「仮想記憶機構」という用語の代わりに、しばしば省略語の「記憶機構」が使用されています。また、「記憶機構」という用語は、意味が明らかな場合は、「主記憶機構」、「絶対記憶機構」、または「実記憶機構」の代わりに使用されることもあります。

「主記憶機構」および「絶対記憶機構」は、絶対アドレスによりアドレッシングできる記憶機構を表す用語として使用されます。これらの用語は、補助記憶機構に対立する概念として、例えば直接アクセス記憶装置(DASD)などが提供する高速アクセス記憶機構を表しています。「実記憶機構」は、プレフィックシングの効果を除けば「絶対記憶機構」と同義です。

記憶機構のアドレッシング

記憶機構は、水平の長いビットのストリングと考えることができます。ほとんどの操作では、記憶機構へのアクセスは左から右への順序で進められます。ビットのストリングは、8ビット単位で分割されます。この8ビットの単位はバイトと呼ばれ、すべての情報フォーマットを形成する基本単位です。

記憶機構内の個々のバイト位置は、負でない固有の整数で識別されます。この整数がバイト位置のアドレスで、簡単にバイト・アドレスと呼ばれます。隣接するバイト位置のアドレスは連続しており、左端の0から始めて、左から右への順に増加します。アドレスは符号なしの2進整数で、24ビット、31ビット、または64ビットです。アドレスについては、3-5ページの『アドレスのサイズと循環』で説明します。

情報のフォーマット

情報は、記憶機構とCPUまたはチャンネル・サブシステムとの間で、一度に1バイトずつ、または1グループ分のバイトを1単位として伝送されます。特に指定されない限り、記憶機構内のバイト・グループは、グループの左端のバイトによりアドレッシングされます。グループのバイト数は、実行する操作により暗黙に決定される場合と明示的に指定される場合があります。CPU操作に使用される場合、バイト・グループはフィールドと呼ばれます。

バイト・グループの中のビットは、左から右の順に番号付けされます。左側のビット群を「高位ビット」、右側のビット群を「低位ビット」と呼ぶことがあります。ただし、ビット番号は記憶域アドレスではありません。アドレッシングできるのはバイトだけです。記憶機構内の1つのバイトを形成する個々のビットを対象として操作を行うには、そのバイト全体にアクセスする必要があります。

1バイトの中のビットには、左から右の順に0から7までの番号が付きます。

1つのアドレスの中のビットには、24ビット・アドレスの場合は8-31または40-63の番号が付き、31ビット・アドレスの場合は1-31または33-63の番号が付き、64ビット・アドレスの場合は0-63の番号が付きません。複数バイトから成るその他の固定長フォーマットの中では、そのフォーマットを形成するビットには、0から始まる連続番号が付きます。

エラー検出のため、また一部のモデルではエラー訂正も行うために、個々のバイトまたはバイト・グループと一

緒に 1 つ以上の検査ビットが伝送されることがあります。この種の検査ビットはマシンにより自動的に生成されるもので、プログラムが直接制御することはできません。本書でデータ・フィールドやレジスターの長さについて述べる場合、関連の検査ビットは長さに含まれていません。記憶容量はすべてバイト数で表しています。

記憶域オペランド・フィールドの長さが命令コードで暗黙に決定される場合、そのフィールドは固定長であると見なされます。その長さは、1 バイト、2 バイト、4 バイト、または 8 バイトです。一部の命令では、これよりさらに長いフィールドが暗黙に指定されることもあります。

記憶域オペランド・フィールドの長さが暗黙指定されず、明示的に指定される場合は、そのフィールドは可変長と見なされます。可変長オペランドは、1 バイト刻みで長さが変化します。

情報が記憶機構に格納されるときは、指定したフィールドに含まれているバイト位置の内容のみが置き換えられます。これは、記憶機構への物理パスの幅が格納しようとしているフィールドより大きい場合でも同じです。

規定境界

情報単位には、記憶機構内の規定境界に合わせて配置しなければならないものがあります。情報単位が規定境界に合っていると見なされるのは、その情報単位の記憶域アドレスが、その単位の長さ (バイト数) の倍数である場合です。規定境界上にある 2 バイト、4 バイト、8 バイト、および 16 バイトのフィールドには、特殊名が与えられています。2 バイト境界上にある 2 個の連続したバイトのグループをハーフワードと呼びます。これは、命令を形成する基本単位です。4 バイト境界上にある 4 個の連続したバイトのグループを、ワードと呼びます。8 バイト境界上にある 8 個の連続したバイトのグループを、ダブルワードと呼びます。そして、16 バイト境界上にある 16 個の連続したバイトのグループを、クワッドワードと呼びます。(3-4 ページの図 3-1 を参照。)

記憶域アドレスが、ハーフワード、ワード、ダブルワード、およびクワッドワードを示している場合、そのアドレスの 2 進数表現では、それぞれ、右端の 1 個、2 個、3 個、または 4 個のビットが 0 になります。

命令は、2 バイト規定境界に配置する必要があります。また、CCW、IDAW、および、一部の命令の記憶域オペランドは、別の規定境界に配置する必要があります。

ほとんどの命令の記憶域オペランドについては、境界合わせは必要ありません。

プログラミング上の注意: フィールド長が 2 の累乗である場合の固定フィールド長操作では、記憶域オペランドがその長さの整数倍に相当するアドレスに置かれていないと、パフォーマンスが著しく低下することがあります。パフォーマンスを高めるには、使用頻度の高いオペランドは規定境界に合わせて置くようにしてください。

アドレスのタイプとフォーマット

アドレスのタイプ

主記憶機構のアドレッシングでは、3 つの基本タイプのアドレスが認識されます。それは、絶対アドレス、実アドレス、仮想アドレスです。これらのアドレス・タイプは、記憶機構アクセスの際にアドレスに適用される変換に基づいて区別されます。アドレス変換では、仮想アドレスが実アドレスに変換され、プレフィックシングでは、実アドレスが絶対アドレスに変換されます。この 3 つの基本アドレス・タイプに加えて、さらに幾つかのタイプが定義されていますが、これらは、命令と現行モードに応じて、3 つの基本タイプの 1 つとして扱われることもあり、まったく別のタイプとして扱われることもあります。

絶対アドレス

絶対アドレスは、主記憶機構内の特定位置に割り当てられているアドレスです。絶対アドレスは、まったく変換を行わずに、そのまま記憶機構のアクセスに使用されます。

構成内のチャンネル・サブシステムおよびすべての CPU は、共用主記憶域内の同じ位置を参照するときには、同じ絶対アドレスを使用します。使用可能な主記憶域には、通常、0 から始まる連続した絶対アドレスが割り当てられ、これらのアドレスは、常に、規定境界に位置合わせされた完全な 4K バイト・ブロック単位で割り当てられます。物理位置に割り当てられていないブロック内の絶対アドレスを使用しようとすると、例外が認識されます。一部のモデルでは、記憶機構の再構成制御機能があり、オペレーターはこれを使用して、絶対アドレスと物理位置の対応付けを変更できる場合があります。ただし、一時点で、1 つの物理位置が複数の絶対アドレスに関連付けられることはありません。

それぞれの絶対アドレスに従って順序付けされたバイト位置から成る記憶機構を、絶対記憶機構と言います。



図 3-1. 記憶域アドレスの規定境界

実アドレス

実アドレスは、実記憶機構内の位置を識別します。主記憶機構にアクセスするとき実アドレスが使用された場合は、そのアドレスは、プレフィックシングにより絶対アドレスに変換されます。

どの時点でも、構成内の個々の CPU ごとに、実アドレスから絶対アドレスへのマッピングが 1 つずつあります。CPU が実アドレスを使用して主記憶機構にアクセスしようとする、そのアドレスは、プレフィックシングによって絶対アドレスに変換されます。個々の場合の変換は、その CPU 用のプレフィックス・レジスター内の値により定義されます。

それぞれの実アドレスに従って順序付けされたバイト位置から成る記憶機構を、実記憶機構と言います。

仮想アドレス

仮想アドレスは、仮想記憶機構内の位置を識別します。仮想アドレスを使用して主記憶機構にアクセスしようとした場合は、そのアドレスは動的アドレス変換により実アドレスに変換され、さらに、プレフィックシングにより絶対アドレスに変換されます。

1 次仮想アドレス

1 次仮想アドレスは、1 次アドレス・スペース制御エレメントにより変換する必要がある仮想アドレスです。論理アドレスは、1 次スペース・モードにあるときは 1 次仮想アドレスとして扱われます。命令アドレスは、1 次スペース・モード、2 次スペース・モード、またはアクセス・レジスター・モードにあるときは、1 次仮想アドレスとして扱われます。MOVE TO PRIMARY の第 1 オペランドのアドレスと、MOVE TO SECONDARY の第 2 オペランドのアドレスは、常に 1 次仮想アドレスとして扱われます。

2 次仮想アドレス

2 次仮想アドレスは、2 次アドレス・スペース制御エレメントにより変換する必要がある仮想アドレスです。論理アドレスは、2 次スペース・モードにあるときは 2 次仮想アドレスとして扱われます。MOVE TO PRIMARY の第 2 オペランドのアドレスと、MOVE TO SECONDARY の第 1 オペランドのアドレスは、常に 2 次仮想アドレスとして扱われます。

AR 指定仮想アドレス

AR 指定仮想アドレスは、アクセス・レジスタ指定のアドレス・スペース制御エレメントにより変換する必要がある仮想アドレスです。論理アドレスは、アクセス・レジスタ・モードにあるときは AR 指定アドレスとして扱われます。

ホーム仮想アドレス

ホーム仮想アドレスは、ホーム・アドレス・スペース制御エレメントにより変換する必要がある仮想アドレスです。論理アドレスおよび命令アドレスは、ホーム・スペース・モードにあるときは、ホーム仮想アドレスとして扱われます。

論理アドレス

特に指定がない限り、ほとんどの命令の記憶域オペランド・アドレスは論理アドレスです。論理アドレスは、実モードでは実アドレスとして、1 次スペース・モードでは 1 次仮想アドレスとして、2 次スペース・モードでは 2 次仮想アドレスとして、アクセス・レジスタ・モードでは AR 指定仮想アドレスとして、そしてホーム・スペース・モードではホーム仮想アドレスとして扱われます。一部の命令では、命令に関連した記憶域オペランド・アドレスや記憶機構アクセスが、論理アドレスの規則に従っていない場合があります。このような場合には、命令定義の中に、アドレスのタイプについての定義が含まれています。

命令アドレス

記憶機構から命令を取り出すために使用するアドレスを、命令アドレスと言います。命令アドレスは、実モードでは実アドレスとして扱われ、1 次スペース・モード、2 次スペース・モード、またはアクセス・レジスタ・モードでは 1 次仮想アドレスとして扱われ、ホーム・スペース・モードではホーム仮想アドレスとして扱われます。現 PSW 内の命令アドレスと EXECUTE のターゲット・アドレスは、命令アドレスです。

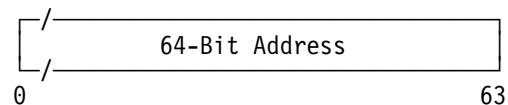
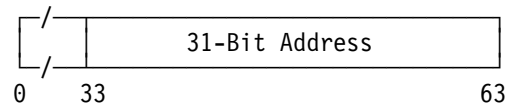
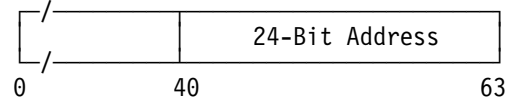
有効アドレス

状況によっては、「有効アドレス」という用語を使うと便利な場合があります。有効アドレスとは、動的アドレス変換またはプレフィックシングによる変換が行われる前に存在しているアドレスを指します。有効アドレスは、レジスタ内で直接指定することもでき、またアドレス計算の結果として生じることもあります。アドレス計算とは、基底と変位の加算、または基底、指標、および変位の加算のことです。

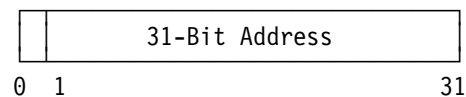
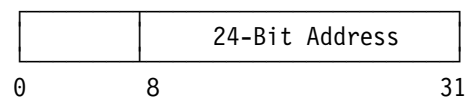
アドレスのサイズと循環

アドレス・サイズとは、1 つのアドレスを表すことのできる有効ビットの最大数です。アドレスのサイズには、24 ビット、31 ビット、64 ビットの 3 種類があります。24 ビット・アドレスでは最大 16,777,216 (16M) バイト、31 ビット・アドレスでは最大 2,147,483,648 (2G) バイト、そして 64 ビット・アドレスでは最大 18,446,744,073,709,551,616 (16E) バイトをアドレッシングできます。

現行のアドレッシング・モードの制御下でアドレス計算により生成される 24 ビット、31 ビット、または 64 ビットのアドレスのビットには、汎用レジスタ内の基底アドレスおよび指標ビットの番号付けに一致するように、それぞれ、40-63、33-63、および 0-63 の番号が付ききます。



31 ビット・アドレスのビットには、アドレッシング・モードに関係なく、1-31 の番号が付ききます。また、記憶域内の 4 バイト・フィールドに 24 ビットまたは 31 ビットのアドレスが含まれている場合は、ビットには、それぞれ、8-31 または 1-31 の番号が付ききます。



24 ビットまたは 31 ビットの仮想アドレスは、左側にそれぞれ 40 個または 33 個の 0 を付加して 64 ビットに拡張されてから、DAT プロセスにより変換されます。また、24 ビットまたは 31 ビットの実アドレスは、同じく 64 ビットに拡張されてから、プレフィックシングにより変換されます。24 ビットまたは 31 ビットの絶対アドレスは、主記憶機構にアクセスする前に、64 ビットに拡張されます。したがって、24 ビット・アドレスは、常に、64 ビット・アドレスによりアドレッシング可能な 16E バイトの記憶域の最初の 16M バイト・ブ

ロック内の位置を指定し、31 ビット・アドレスは、常に最初の 2G バイト・ブロック内の位置を指定します。

本書では、特に否定する記述がない限り、以下の定義が適用されます。すなわち、マシンが 24 ビットまたは 31 ビットのアドレスを生成してプログラムに渡すときには、そのアドレスは 32 ビット・フィールドに組み込まれ、フィールド内の左端のそれぞれ 8 ビットまたは 1 ビットが 0 に設定されて、使用可能にされます (記憶機構に格納されるか、汎用レジスターにロードされます)。アドレスが汎用レジスターにロードされる場合は、レジスターのビット 0-31 は変更されません。

有効アドレスのサイズは、PSW のビット 31 (拡張アドレッシング・モード・ビット) および 32 (基本アドレッシング・モード・ビット) により制御されます。ビット 31 および 32 が両方とも 0 である場合は、CPU は 24 ビット・アドレッシング・モードになっており、24 ビットオペランドおよび命令有効アドレスが指定されず。ビット 31 が 0 でビット 32 が 1 のときは、CPU は 31 ビット・アドレッシング・モードになっており、31 ビットオペランドおよび命令有効アドレスが指定されます。ビット 31 および 32 が両方とも 1 のときは、CPU は 64 ビット・アドレッシング・モードになっており、64 ビットオペランドおよび命令有効アドレスが指定されます (5-7 ページの『アドレスの生成』を参照)。

ASN 変換、ASN 許可、PC 番号変換、およびアクセス・レジスター変換プロセスで使用されるかまたは導き出される実アドレスまたは絶対アドレスのサイズは、現アドレッシング・モードに関係なく、常に 31 ビットです。同様に、DAT、スタッキング、アンスタッキング、およびトレース・プロセスで使用されるかまたは導き出される実アドレスまたは絶対アドレスのサイズは、常に 64 ビットです。

CCW に含まれるデータ・アドレスのサイズは、START SUBCHANNEL 命令で指定される操作要求ブロック (ORB) 内の CCW フォーマット制御ビットにより制御されます。24 ビット・アドレスおよび 31 ビット・アドレスを含む CCW は、それぞれフォーマット 0 およびフォーマット 1 の CCW と呼ばれます。フォーマット 0 およびフォーマット 1 の CCW については、第 15 章、『基本入出力機能』で説明します。同様に、IDAW に含まれるデータ・アドレスのサイズは、ORB 内の IDAW フォーマット制御ビットにより制御されず。31 ビットおよび 64 ビットのアドレスを含む IDAW は、それぞれフォーマット 1 およびフォーマット

2 の IDAW と呼ばれます。フォーマット 1 およびフォーマット 2 の IDAW については、第 15 章、『基本入出力機能』で説明します。

アドレス循環

CPU は、オペランドまたは命令のアドレスを形成するとき、または、該当するテーブル起点と指標からテーブル・エントリーのアドレスを作るときに、アドレス生成を行います。また、1 つのフィールドの連続したバイトにアクセスするためにアドレスを増分するときにも、CPU はアドレス生成を行います。同様に、チャンネル・サブシステムは、アドレスを増分する必要がある場合に、アドレス生成を行います。アドレスを増分するのは、(1) CCW を取り出すため、(2) IDAW を取り出すため、(3) データを転送するため、または、(4) 入出力測定ブロックのアドレスを計算するためです。

アドレスを生成するとき、アドレス・サイズとして許容される値 ($2^{24} - 1$ 、 $2^{31} - 1$ 、または $2^{64} - 1$) を超えるアドレスが得られた場合は、次の 2 つの処置のいずれかがとられます。

1. アドレスの高位ビット位置の繰り上がりが無視される。超過サイズのアドレスに対するこのような処理を、**循環**と呼びます。
2. 割り込み条件が認識される。

循環の効果として、アドレス・スペースが環状に見えるようになります。つまり、アドレス 0 が最大許容アドレスの次に続いているように見えます。DAT またはプレフィッキングによるアドレス変換が行われる場合は、アドレス計算と循環はその前に行われます。

CPU が生成するアドレスのうち仮想アドレスとなるものは、常に循環します。また、PROGRAM RETURN の結果、制御レジスター 15 の中のリンケージ・スタック・エントリー・アドレスが 0 より小さい値に減少したときも、循環が生じます。実アドレスまたは絶対アドレスによりアドレッシングされる CPU テーブル・エントリーの場合は、アドレスが循環するのか、それともアドレッシング例外が発生するのかは予測できません。

チャンネル・プログラムの実行において、生成アドレスがアドレス・サイズの値を超過した場合 (または、逆方向読み取りコマンドの場合はアドレスが 0 より小さい値になった場合) は、入出力プログラム・チェック条件が生じます。

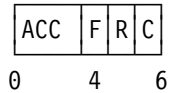
3-7 ページの図 3-2 は、各種のアドレス生成に対して適用されるしきい値、およびアドレスが許容値を超えた場合に行われる処置を示しています。

以下のものためのアドレス生成	アドレス・タイプ	アドレスが循環する場合の処置
EAM および BAM が 0 の場合の命令とオペランド	L,I,R,V	W24
EAM および BAM が 0 の場合の命令とオペランドの連続バイト	I,L,V ¹	W24
EAM が 0 で BAM が 1 の場合の命令とオペランド	L,I,R,V	W31
EAM が 0 で BAM が 1 の場合の命令とオペランドの連続バイト	I,L,V ¹	W31
EAM および BAM が 1 の場合の命令とオペランド	L,I,R,V	W64
EAM および BAM が 1 の場合の命令とオペランドの連続バイト	I,L,V ¹	W64
暗黙変換または LRA または STRAG に使用される場合の DAT テーブル・エントリー	A または R ²	X64
ASN 第 2 テーブル、権限テーブル (ASN 許可時)、リンケージ・テーブル、およびエントリー・テーブルのエントリー	R	X31
権限テーブル (アクセス・レジスタ変換時) およびアクセス・リストのエントリー	A または R ²	X31
リンケージ・スタック・エントリー	V	W64
入出力測定ブロック	A	P31
フォーマット 0 の CCW を持つチャンネル・プログラムの場合:		
連続する CCW	A	P24
連続する IDAW	A	P24
入出力データの連続するバイト (IDAW なし)	A	P24
入出力データの連続するバイト (フォーマット 1 の IDAW あり)	A	P31
入出力データの連続するバイト (フォーマット 2 の IDAW あり)	A	P24
フォーマット 1 の CCW を持つチャンネル・プログラムの場合:		
連続する CCW	A	P31
連続する IDAW	A	P31
入出力データの連続するバイト (IDAW なし)	A	P31
入出力データの連続するバイト (フォーマット 1 の IDAW あり)	A	P31
入出力データの連続するバイト (フォーマット 2 の IDAW あり)	A	P64
説明:		
¹ 実アドレスを使用してオペランドを指定する命令では、2 ²⁴ 、2 ³¹ 、または 2 ⁶⁴ の境界にまたがるオペランドを指定することはできないので、この場合は実アドレスは適用されません。 ² アドレスが絶対アドレスか実アドレスかは、予測不能です。 A 絶対アドレス。 BAM PSW に含まれる基本アドレッシング・モード・ビット。 EAM PSW に含まれる拡張アドレッシング・モード・ビット。 I 命令アドレス。 L 論理アドレス。 P24 アドレスが 2 ²⁴ - 1 を超過するかまたは 0 より小さくなると、入出力プログラム・チェック条件が生じます。 P31 アドレスが 2 ³¹ - 1 を超過するかまたは 0 より小さくなると、入出力プログラム・チェック条件が生じます。 P64 アドレスが 2 ⁶⁴ - 1 を超過するかまたは 0 より小さくなると、入出力プログラム・チェック条件が生じます。 R 実アドレス。 V 仮想アドレス。 W24 位置 2 ²⁴ - 1 を超えると位置 0 に循環します (またはその逆)。 W31 位置 2 ³¹ - 1 を超えると位置 0 に循環します (またはその逆)。 W64 位置 2 ⁶⁴ - 1 を超えると位置 0 に循環します (またはその逆)。 X31 アドレスが 2 ³¹ - 1 を超えた場合に、位置 2 ³¹ - 1 の後で位置 0 に循環するか、それともアドレッシング例外が生じるかは、予測不能です。 X64 アドレスが 2 ⁶⁴ - 1 を超えた場合に、位置 2 ⁶⁴ - 1 の後で位置 0 に循環するか、それともアドレッシング例外が生じるかは、予測不能です。		

図 3-2. アドレス循環

記憶キー

構成内の使用可能な個々の 4K バイト・ブロックに、記憶キーが 1 つずつ関連付けられます。記憶キーのフォーマットは以下のとおりです。



記憶キー内のビット位置は、以下のように割り振られません。

アクセス制御ビット (ACC): 参照がキー制御保護を受ける場合は、情報が格納される時、または取り出し保護が指定されている記憶位置から情報を取り出すときに、4 つのアクセス制御ビット (ビット 0-3) が 4 ビットのアクセス・キーと突き合わせられます。

取り出し保護ビット (F): 参照がキー制御保護を受ける場合は、取り出し保護ビット (ビット 4) によって、取り出しタイプの参照にキー制御保護を適用するかどうかが決まります。0 は、格納タイプの参照のみをモニターし、取り出しについては、アクセス・キーが何であってもすべて許可することを意味します。1 は、取り出しと格納の両方にキー制御保護を適用することを意味します。命令の取り出しとオペランドの取り出しとの区別はありません。

参照ビット (R): 参照ビット (ビット 5) は、通常、対応する記憶ブロック内のいずれかの記憶位置が、情報の格納または取り出しのために参照されるたびに、1 に設定されます。

変更ビット (C): 変更ビット (ビット 6) は、対応する記憶ブロック内のいずれかの位置に情報が格納されるたびに、1 に設定されます。

記憶キーは、アドレッシング可能な記憶機構の一部ではありません。記憶キー全体が、SET STORAGE KEY EXTENDED により設定され、INSERT STORAGE KEY EXTENDED により検査されます。さらに、RESET REFERENCE BIT EXTENDED 命令を使用すると、参照ビットと変更ビットを検査したり、参照ビットを 0 に設定したりできます。記憶キーのビット 0-4 は、INSERT VIRTUAL STORAGE KEY 命令により検査されます。TEST BLOCK 命令による使用可能度テストの実行中およびそれ以降の記憶キーの内容は、予測不能です。

保護

主記憶機構の内容が、エラーを含むプログラムや無許可のプログラムによって破壊されたり誤用されたりするのを防ぐために、4 つの保護機能が用意されています。それは、キー制御保護、アクセス・リスト制御保護、ページ保護、および低アドレス保護です。これらの保護機能は、それぞれ独立して適用されます。つまり、これらの保護機能のどれもアクセスを禁止していない場合のみ、主記憶機構へのアクセスが許されます。

キー制御保護は、不適正な格納に対する保護、および不適切な格納と取り出しの両方に対する保護は提供しますが、不適正な取り出しのみに対する保護は提供しません。

キー制御保護

記憶機構アクセスに対してキー制御保護が適用されているときは、格納が許可されるのは、記憶機構アクセス要求に関連したアクセス・キーと記憶キーが一致している場合だけです。取り出しが許可されるのは、この 2 つのキーが一致しているか、または記憶キーの取り出し保護ビットが 0 の場合です。

キーが一致したと見なされるのは、記憶キーの 4 つのアクセス制御ビットがアクセス・キーと等しいか、またはアクセス・キーが 0 の場合です。

3-9ページの図3-3 に、保護アクションを要約して示します。

記憶機構へのアクセスが CPU により開始され、キー制御保護が適用されているときは、PSW キーがアクセス・キーです。ただし、MOVE TO SECONDARY および MOVE WITH DESTINATION KEY の第 1 オペランド、および、MOVE TO PRIMARY、MOVE WITH KEY、および MOVE WITH SOURCE KEY の第 2 オペランドについては、アクセス・キーは汎用レジスターで指定されます。PSW キーは、現 PSW のビット位置 8-11 を占めます。

チャンネル・プログラムを実行することを目的として記憶機構にアクセスする場合は、そのチャンネル・プログラムに関連したサブチャンネル・キーがアクセス・キーです。チャンネル・プログラム用のサブチャンネル・キーは、操作要求ブロック (ORB) で指定されます。チャンネル・サブシステムをモニターすることを目的として測定ブロックにアクセスする場合は、測定ブロック・キーがアクセス・キーです。測定ブロック・キーは、SET CHANNEL MONITOR 命令で指定されます。

条件		記憶機構のアクセスが許可されるか？	
記憶キーの取り出し保護ビット	キーの関係	取り出し	格納
0	一致	可	可
0	不一致	可	不可
1	一致	可	可
1	不一致	不可	不可
説明			
一致	記憶キーの 4 つのアクセス制御ビットがアクセス・キーに等しいか、またはアクセス・キーが 0 です。		
可	アクセスが許可されます。		
不可	アクセスは許可されません。取り出しでは、該当情報はプログラムに対して使用可能にされず、格納では、該当の記憶位置の内容は変更されません。		

図 3-3. 保護アクションの要約

キー制御保護により CPU アクセスが禁止されている場合は、命令の実行は打ち切られ、記憶保護例外のためのプログラム割り込みが発生します。ただし、3-12ページの『保護時の抑止』で説明するように、操作単位または割り込み実行を抑止することができます。チャンネル・プログラム・アクセスが禁止されたときは、開始機能は終了し、関連の割り込み応答ブロック (IRB) に記憶保護チェック条件が示されます。測定ブロック・アクセスが禁止された場合は、入出力測定ブロック保護チェック条件が示されます。

キー制御保護により格納アクセスが禁止された場合は、記憶保護域の内容は変化しません。取り出しアクセスが禁止された場合は、保護されている情報は、レジスターにロードされることも、別の記憶位置に移されることも、入出力装置に渡されることもありません。命令取り出しが禁止された場合は、その命令は抑止され、何らかの命令長コードが示されます。

キー制御保護は、CPU が問題プログラム状態か監視プログラム状態かには無関係に働き、また、下記の場合を除き、実行されている CPU 命令またはチャンネル・コマンド・ワードのタイプにも依存しません。

特に別の指定がない限り、プログラムが明示的に指定し、CPU が情報の格納または取り出しのために使用する記憶位置へのアクセスは、すべてキー制御保護の対象となります。

記憶保護オーバーライド制御が 1 で、記憶キーの 4 つのアクセス制御ビットの値が 9 である場合は、キー制御保護は適用されません。取り出し保護オーバーライド制御が 1 の場合は、取り出しに対するキー制御保護は適用

される場合と適用されない場合があります。これは、有効アドレスと専用スペース制御によって決まります。

TEST BLOCK の第 2 オペランドへのアクセスは、キー制御保護の対象にはなりません。

チャンネル・サブシステムが入出力測定ブロックにアクセスするために行う記憶機構アクセス、あるいは、チャンネル・プログラムが CCW または IDAW を取り出すため、または CCW の実行時に指定されたデータ域にアクセスするために行う記憶機構アクセスは、すべてキー制御保護の対象となります。ただし、CCW、IDAW、または出力データが事前取り出しされている場合は、その CCW または IDAW が制御を受け取ることが確定的になるまで、または該当データが書き出されることが確定的になるまでは、保護チェックは示されません。

以下のような処理のために暗黙的に行われるアクセスには、キー制御保護は適用されません。

- 割り込み
- CPU ログアウト
- アクセス・レジスター変換、動的アドレス変換、PC 番号変換、ASN 変換、または ASN 許可のためのテーブル・エントリーの取り出し
- トレース
- 状況記憶機能
- TEST PENDING INTERRUPTION のオペランド・アドレスがゼロの場合の、実記憶位置 184-191 への格納
- 初期プログラム・ローディング

同様に、情報の変更や表示を目的としてオペレーター機能から開始されるアクセスに対しても、保護は適用されません。ただし、プログラムがその情報の記憶位置を明示的に指定した場合は、該当記憶位置は保護の対象となります。

記憶保護オーバーライド制御

制御レジスター 0 のビット 39 は、記憶保護オーバーライド制御です。このビットが 1 のときは、記憶保護オーバーライドはアクティブになっています。このビットが 0 のときは、記憶保護オーバーライドは非アクティブです。記憶保護オーバーライドがアクティブのときは、関連の記憶キーの値が 9 である記憶位置については、キー制御記憶保護は無視されます。記憶保護オーバーライドが非アクティブのときは、記憶キー値が 9 であっても、特別な処置は行われません。

記憶保護オーバーライドが適用されるのは、命令取り出しと、オペランド・アドレスが論理、仮想、または実アドレスである命令の取り出しアクセスと格納アクセスです。チャンネル・プログラムの実行またはチャンネル・サブ

システムのモニターを目的として行われるアクセスには、これは適用されません。

記憶保護オーバーライドは、キー制御保護の対象となっていないアクセスに対しては効力はありません。

プログラミング上の注意:

1. 記憶保護オーバーライドを使用することにより、信頼性のあるシステムで、エラーがある可能性のあるアプリケーション・プログラムを実行した場合の信頼性を高めることができます。ただし、そのアプリケーション・プログラムは、サブシステムがアクセスする記憶機構の一部分のみにアクセスしなければならないという条件があります。これを達成するための手法は、以下のとおりです。アプリケーション・プログラムがアクセスする記憶域には、記憶キー 9 を与えます。サブシステムのみがアクセスする記憶域には、0 以外の別の記憶キー (例えば 8) を与えます。アプリケーションは、PSW キー 9 を使用して実行されます。サブシステムは、PSW キー 8 (この例の場合) を使用して実行されます。したがって、サブシステムは、キー 8 および キー 9 の両方の記憶域にアクセスできますが、アプリケーション・プログラムがアクセスできるのはキー 9 の記憶域のみです。
2. 記憶保護オーバーライドは、CPU が行う記憶機構アクセスのほか、TEST PROTECTION の結果にも影響を与えます。ただし、問題プログラム状態の中で PSW キー・マスクをテストして、特定のキー値が使用できるかどうかを判別するための命令は、記憶保護オーバーライドがアクティブであるかどうかによって影響を受けることはありません。この種の命令の主なものとしては、MOVE WITH KEY と SET PSW KEY FROM ADDRESS があります。これらの命令が問題プログラム状態でアクセス・キー 9 を使用できるようにするには、PSW キー・マスクのビット 9 が 1 でなければなりません。

取り出し保護オーバーライド制御

制御レジスター 0 のビット 38 は、取り出し保護オーバーライド制御です。このビットが 1 のときは、有効アドレス 0-2047 の記憶位置に対する取り出し保護は無視されます。有効アドレスとは、動的アドレス変換またはプレフィックによる変換の前に存在しているアドレスです。ただし、有効アドレスが動的アドレス変換の対象となり、変換に使用されるアドレス・スペース制御エレメント内の専用スペース制御 (ビット 55) が 1 である場合は、取り出し保護は無視されません。

取り出し保護オーバーライドが適用されるのは、命令取り出しと、オペランド・アドレスが論理、仮想、または実アドレスである命令の取り出しアクセスです。チャンネル・プログラムの実行またはチャンネル・サブシステムのモニターを目的として行われる取り出しアクセスには、これは適用されません。このビットが 0 に設定されているときは、有効アドレス 0-2047 の記憶位置に対する取り出し保護が適用されるかどうかは、これらの記憶位置に関連した記憶キーの取り出し保護ビットの状態によって決まります。

取り出し保護オーバーライドは、キー制御保護の対象となっていないアクセスに対しては効力はありません。

プログラミング上の注意: 取り出し保護オーバーライド制御を使用すると、アドレス 0-2047 の記憶位置には取り出し保護は適用されず、アドレス 2048-4095 の記憶位置には取り出し保護が適用されます。

アクセス・リスト制御保護

アクセス・レジスター・モードでは、アクセス・リスト・エンタリーのビット 6 (取り出し専用ビット) によって、アクセス・リスト・エンタリーで指定されているアドレス・スペースに対して許されるオペランド参照のタイプが決まります。エンタリーが参照のアクセス・レジスター変換部分の中で使用されていて、ビット 6 が 0 であるときは、取り出しタイプと格納タイプの両方の参照が許可されます。ビット 6 が 1 のときは、取り出しタイプの参照のみが許可され、格納を行おうとすると記憶保護例外が発生し、命令の実行は抑止されます。

取り出し専用ビットは、ALB アクセス・リスト・エンタリーに含まれています。主記憶機構の中で、アクセス・リスト・エンタリー内の取り出し専用ビットに変更を加えても、記憶保護例外が認識されるかどうかについての効力は、必ずしもただちに発生するとは限りません。ただし、PURGE ALB、または ALB をページする COMPARE AND SWAP AND PURGE 命令の実行後は、このビット変更の効力はただちに発生します。

CPU がアクセス・レジスター・モードになっているときは、TEST PROTECTION の実行時にアクセス・リスト制御保護が適用されます。アクセス・リスト制御保護に違反すると、条件コード 1 が設定されます。ただし、条件コード 2 または 3 の条件が満たされたときは、これらの条件コードが設定されます。

プログラミング上の注意: アクセス・リスト制御保護に違反すると、常に抑止が発生します。その他のタイプの保護に対する違反は、終了をもたらすことがあります。

ページ保護

ページ保護機能は、個々のページ・テーブル・エントリーおよびセグメント・テーブル・エントリー内のページ保護ビットを使用して、仮想記憶域に対するアクセスを制御します。この機能は、不適正な格納に対する保護手段を提供します。

ページ・テーブル・エントリーのページ保護ビット (ビット 54) は、対応する 4K バイト・ページへの格納を許可するかどうかを制御します。このビットが 0 のときは、取り出しと格納の両方が許可され、このビットが 1 のときは取り出しのみが許可されます。保護されているページへの格納を行おうとした場合、そのページの内容は変更されず、操作単位または命令実行は抑止され、記憶保護のためのプログラム割り込みが生じます。

セグメント・テーブル・エントリーのページ保護ビット (ビット 54) は、セグメント・テーブル・エントリーで指定されたページ・テーブル内の各エントリーのページ保護ビット位置と OR 関係にあるものと見なされます。したがって、セグメント・テーブル・エントリーのページ保護ビットが 1 のときの効果は、指定されたページ・テーブル内の各エントリーのページ保護ビットが 1 の場合と同じです。

ページ保護は、仮想アドレスを使用するすべての格納タイプの参照に適用されます。

低アドレス保護

低アドレス保護機能は、割り込み処理中に、CPU が使用する主記憶機構の情報が破壊されるのを防ぎます。この保護を達成するために、命令が、0-511 および 4096-4607 の範囲内の有効アドレス (最初と 2 番目の 4K バイト有効アドレス・ブロックの最初の 512 バイト) を使用して格納するのを、禁止します。この範囲の基準は、動的アドレス変換またはプレフィックシングによるアドレス変換の前に適用されます。ただし、有効アドレスが動的変換の対象となり、その変換で使用されるアドレス・スペース制御エレメント内の専用スペース制御 (ビット 55) が 1 である場合は、この範囲基準は適用されず、したがって低アドレス保護は適用されません。他のタイプの例外が原因で、使用すべきアドレス・スペース制御エレメントが使用できない場合は、低アドレス保護は適用されません。

低アドレス保護は、低アドレス保護制御ビット、つまり制御レジスター 0 のビット 35 により制御されます。このビットが 0 のときは、低アドレス保護はオフであり、このビットが 1 のときは、低アドレス保護はオンになっています。

低アドレス保護によりアクセスが禁止された場合は、記憶保護域の内容は変更されず、命令の実行は終了し、記憶保護例外のためのプログラム割り込みが発生します。ただし、3-12 ページの『保護時の抑止』で説明するように、操作単位または割り込み実行を抑止することができます。

0-511 または 4096-4607 の範囲内の有効アドレスを使用して格納しようとするプログラムの試みは、すべて低アドレス保護の対象となります。低アドレス保護が適用されるのは、オペランド・アドレスが論理、仮想、または実アドレスである命令の格納アクセスです。また、低アドレス保護はトレース・テーブルにも適用されます。

低アドレス保護は、割り込み、CPU ログアウト、TEST PENDING INTERRUPTION による実記憶位置

184-191 への入出力割り込みコードの格納、および、初期プログラム・ローディングや状況記憶機能などのシーケンスのために、CPU またはチャンネル・サブシステムが行うアクセスには、適用されません。また、入出力データ転送中のデータの格納にも適用されません。ただし、プログラムがこれらの記憶位置を明示して格納を行う場合は、低アドレス保護の対象になります。

プログラミング上の注意:

1. 低アドレス保護とキー制御保護は、以下の例外を除き、同じ格納アクセスに対して適用されます。
 - a. チャンネル・サブシステムが行う格納には、キー制御保護は適用されますが、低アドレス保護は適用されません。
 - b. トレース、TEST BLOCK の第 2 オペランド、または、特にリンケージ・スタックを対象として作用する命令には、キー制御保護は適用されませんが、低アドレス保護は適用されません。
2. 取り出し保護オンライン場および低アドレス保護は、アドレス・スペース制御エラー内で専用スペース制御が 1 に設定されているアドレス・スペース内には適用されないため、そのアドレス・スペース内の記憶位置 0-2047 および 4096-4607 は、そのスペース内の他の記憶位置と同様に使用可能です。

保護時の抑止

一部の命令定義では、いずれかのタイプの保護が原因で記憶保護例外が認識された場合に、必ず操作を抑止することが指定されています。この指定がないときは、アクセス・リスト制御保護またはページ保護が原因で記憶保護例外が認識された場合は、命令の実行は常に抑止され、低アドレス保護またはキー制御保護が原因で記憶保護例外が認識された場合は、命令の実行は抑止されるかまたは終了します。

保護時抑止機能が使用されている場合、制御プログラムは、記憶保護例外を引き起こした仮想アドレスの変換に使用されているセグメント・テーブル・エントリおよびページ・テーブル・エントリを見つけて、その例外がページ保護によるものかどうかを判別します。¹ この機能は、Posix フォーク機能をインプリメントするために必要です (これについては『プログラミング上の注意』で説明します。) また、アドレスが仮想アドレスではない場合、または例外がアクセス・リスト制御保護によるものである場合は、この機能の働きにより、制御プログラムは、セグメント・テーブルおよびページ・テーブルのエントリを見つける作業を省略できます。

記憶保護例外を原因とするプログラム割り込みでは、実記憶位置 168-175 のビット位置 61 に、1 または 0 が格納されます。ビット位置 61 に 1 が格納されている場合は、以下のことを意味します。

- 例外が認識された操作単位または命令実行は、抑止されています。
- 動的アドレス変換 (DAT) がオンであった場合は (これはプログラムの旧 PSW 内の DAT モード・ビットに示されています)、例外を引き起こした有効アドレスは、DATにより変換されることになってきたアドレスです (有効アドレスとは、DAT またはプレフィックシングによる変換の前に存在しているアドレスです)。DAT がオンであっても、有効アドレスが実アドレスであるため DAT により変換すべきものでなかった場合は、ビット 61 は 0 に設定されます。DAT がオフであった場合は、ページ保護が原因で記憶保護例外が生じていることはありません。
- 実記憶位置 168-175 のビット位置 0-51 には、例外を引き起こした有効アドレスのビット 0-51 が含まれています。DAT がオンであった場合、つまり、有効アドレスが DAT により変更されることになっ

ていた場合は、実記憶位置 168-175 のビット位置 62 および 63 と、実記憶位置 160 には、ページ変換例外を原因とするプログラム割り込みで保管されるものと同じ情報が含まれています。この情報は、保護対象アドレスを含むアドレス・スペースを識別します。さらに、実記憶位置 168-175 のビット 60 は、記憶保護例外がアクセス・リスト制御保護により生じたものではない場合は 0 で、例外がアクセス・リスト制御保護により生じたものである場合は 1 です。ビット位置 60 が 1 である場合、それは、例外がページ保護を原因とするものではないことを示します。DAT がオフだった場合は、実記憶位置 168-175 のビット位置 60、62、および 63 の内容および実記憶位置 160 の内容は、予測不能です。実記憶位置 168-175 のビット位置 52-59 の内容は、常に予測不能です。

ビット 61 が 0 の場合は、操作が抑止または終了されたこと、および、実記憶位置 168-175 の残りの部分および実記憶位置 160 の内容は予測不能であることを示します。

記憶保護例外が、アクセス・リスト制御保護またはページ保護が原因で生じたものである場合は、ビット 61 は 1 に設定されます。また、低アドレス保護またはキー制御保護が原因で記憶保護例外が生じた場合も、ビット 61 が 1 に設定されることがあります。

記憶保護例外が、アクセス・リスト制御保護またはページ保護が原因で生じていると同時に、低アドレス保護またはキー制御保護も原因で生じている場合は、何が理由でその記憶保護例外が生じたかは予測不能であり、したがって、ビット 61 が 0 または 1 のどちらに設定されるかも予測不能です。

プログラミング上の注意:

1. 保護時抑止機能は、重複アドレス・スペースを作成する Posix フォーク機能を実行するときに便利です。フォークが生じると、制御プログラムは、複数の異なるアドレス・スペースの同じページを、実記憶機構の 1 つのページ・フレームにマップします (ただし、そのページへの格納が試行されていない場合に限り)。その後、特定のアドレス・スペースへの格納が試行されると、制御プログラムは、そのアドレス・スペース内の該当ページに固有のページ・フレームを割り当て、そのページの内容をその新しいページ・フレームにコピーします。この最

¹ 保護時抑止機能は、ESA/390 の保護時抑止機能を記憶装置としています。ページ保護用の抑止は、この機能の一部として新たに追加されたものです。

後のアクションは、しばしばコピー・オン・ライト機能と呼ばれます。制御プログラムは、ページへの格納の試行を検出するために、そのページ用のページ・テーブル・エントリ内のページ保護ビットを 1 に設定します。制御プログラムは、最初は、指定されたセグメント内のどの場所への格納の試行も検出できるように、セグメント・テーブル・エントリ内のページ保護ビットを 1 に設定します。

2. DAT がオンであるときに、実記憶位置 168-175 のビット 61 が 1 である場合は、記憶保護例外を引き起こしたアドレスが仮想アドレスであることを示します。この指示に基づいて、プログラム式のアクセス・レジスタ変換および動的アドレス変換の行うことにより、例外の原因が低アドレス保護またはキー制御保護ではなく、ページ保護にあるのかどうかを判別することができます。

3. 図3-4に、保護時抑止の結果を要約して示します。

LA または キー 制御 保護	DAT	ALC または ページ 保護	有効 アド レス	ビット 61	ビット 61 が 1 の場合	
					ビット 62、63、 および 記憶 位置 160	ビット 60
なし	オン	あり	Log.	1	P	
あり	オン	あり	Log.	U1	P	
あり	オフ	なし	Log. 実	U2	U	
あり	オフ	なし	Log. 実	U2	U P	
あり	オン	なし	Log. 実	OR	-	
説明： - 重要でないかまたは不適用。 0R 有効アドレスが実アドレスのため 0。 1A アクセス・リストで制御される保護が原因でビット 61 が 1 に設定されている場合は、1。その他の場合は 0。 ALC アクセス・リスト制御の対象。 LA 低アドレス。 Log. 論理 (Logical)。 P 予測可能。 U1 アクセス・リスト制御保護またはページ保護ではなく、低アドレス保護またはキー制御保護が認識される可能性があるため、予測不能。 U2 ビット 61 を 1 に設定する必要があるのは、アクセス・リスト制御保護またはページ保護の場合のみなので、予測不能。 U3 DAT がオフのため、予測不能。						

図 3-4. 保護時抑止の結果

参照記録

参照記録は、置換するページを選択するとき使用する情報を提供します。参照記録には、記憶キーの参照ビット (ビット 5) が使用されます。参照ビットは、DAT がオンかオフかに関係なく、対応する記憶ブロック内のいずれかの記憶位置が情報の取り出しまたは格納のために参照されるたびに、1 に設定されます。

参照記録は、常にアクティブになっており、CPU、オペレーター機能、またはチャンネル・サブシステムが行うアクセスも含めて、すべての記憶機構アクセスについて行われます。また、割り込みや入出力命令実行などの一環として暗黙的にマシンが行うアクセスについても、参照記録が行われます。

以下の命令のオペランド・アクセスについては、参照記録は行われません。これらの命令は、記憶位置にアクセスすることなく、記憶キーを直接参照するからです。

- INSERT STORAGE KEY EXTENDED
- RESET REFERENCE BIT EXTENDED (参照ビットは 0 に設定される)
- SET STORAGE KEY EXTENDED (参照ビットは指定値に設定される)

参照ビットから得られる記録は、おおむね正確です。参照ビットは、プログラムで指定も使用もされないデータまたは命令を取り出すことにより、1 に設定されることがあります。また、ある種の条件下では、参照が行われても参照ビットが 1 に設定されないこともあります。ある種の例外的な状況下では、明示的なプログラム・アクション以外の理由により、参照ビットが 0 に設定されることもあります。

変更記録

変更記録は、ページが主記憶機構内で置き換えられるときに、どのページを補助記憶機構に保管する必要があるかについての情報を提供します。変更記録には、記憶キーの変更ビット (ビット 6) が使用されます。

変更ビットは、対応する記憶ブロックの内容が格納アクセスにより変更されるたびに、1 に設定されます。記憶域の内容を変更しない格納アクセスの場合は、変更ビットは 1 に設定される場合とされない場合があります。

アクセスが禁止されているときに格納を行おうとした場合は、変更ビットは 1 に設定されません。特に、以下のような条件が存在します。

1. CPU による格納アクセスでは、そのアクセスについてアクセス例外が存在する場合、または、そのアクセスに関するアクセス例外より優先順位の高い例外が存在する場合は、その格納アクセスは禁止されます。
2. チャンネル・サブシステムによる格納アクセスの場合は、そのアクセスについてキー制御保護違反が存在する場合は、そのアクセスは禁止されます。

変更記録は、常にアクティブになっており、CPU、オペレーター機能、またはチャンネル・サブシステムが行うアクセスも含めて、すべての格納アクセスについて行われます。また、割り込みなどの一環としてマシンが行う暗黙的な参照についても、変更記録が行われます。

以下の命令のオペランドについては、変更記録は行われません。これらの命令は、記憶位置を変更せずに、記憶キーを直接変更するからです。

- RESET REFERENCE BIT EXTENDED
- SET STORAGE KEY EXTENDED (変更ビットは指定値に設定される)

0 から 1 に変更された変更ビットは、必ずしも CPU の再試行時に 0 に復元されるとは限りません (11-2 ページの『CPU 再試行』を参照)。ある種の異常な状況下での変更ビットの取り扱いについては、5-22 ページの『無効化と抑止に対する例外』を参照してください。

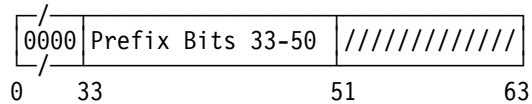
プレフィックシング

プレフィックシングを使用すると、0-8191 の範囲の実アドレスを、各 CPU ごとに絶対記憶機構内の異なるブロックに割り当てることができます。これにより、主記憶機構を共用する複数の CPU が、特に割り込み処理での干渉を最小限に抑えながら、同時に稼働できるようになります。

プレフィックシングを使用すると、0-8191 の範囲内の実アドレスが、該当 CPU 用のプレフィックス・レジスターのビット位置 0-50 の値で識別される 8K バイト絶対アドレスのブロック (プレフィックス域) に 1 対 1 で対応付けされ、プレフィックス・レジスター内のその値で識別される実アドレスのブロックが、絶対アドレス 0-8191 に 1 対 1 で対応付けされます。残りの実アドレスは、それぞれ対応する絶対アドレスと同じです。この変換により、各 CPU は、最初の 8K バイトと、他の CPU のプレフィックス・レジスターで指定されている記憶位置も含めて、主記憶域のすべてにアクセスできるようになります。

実アドレスと絶対アドレスの関係を図で表すと、3-15 ページの図3-5 のようになります。

プレフィックスは、プレフィックス・レジスターのビット位置 0-50 に含まれる 51 ビットの数量です。このレジスターのフォーマットは以下のとおりです。



レジスターのビット 0-32 は、常にすべて 0 です。レジスターのビット 33-50 は、特権命令 SET PREFIX により設定し、特権命令 STORE PREFIX により検査することができます。

SET PREFIX は、プレフィックス・レジスターのビット 33-50 を、記憶域内の 1 つのワードのビット位置 1-18 の値に設定し、そのワードのビット位置 0 および 19-31 の内容は無視します。STORE PREFIX は、プレフィックス・レジスターのビット位置 33-50 の値を、記憶域内の 1 つのワードのビット位置 1-18 に格納し、そのワードのビット位置 0 および 19-31 には 0 を格納します。

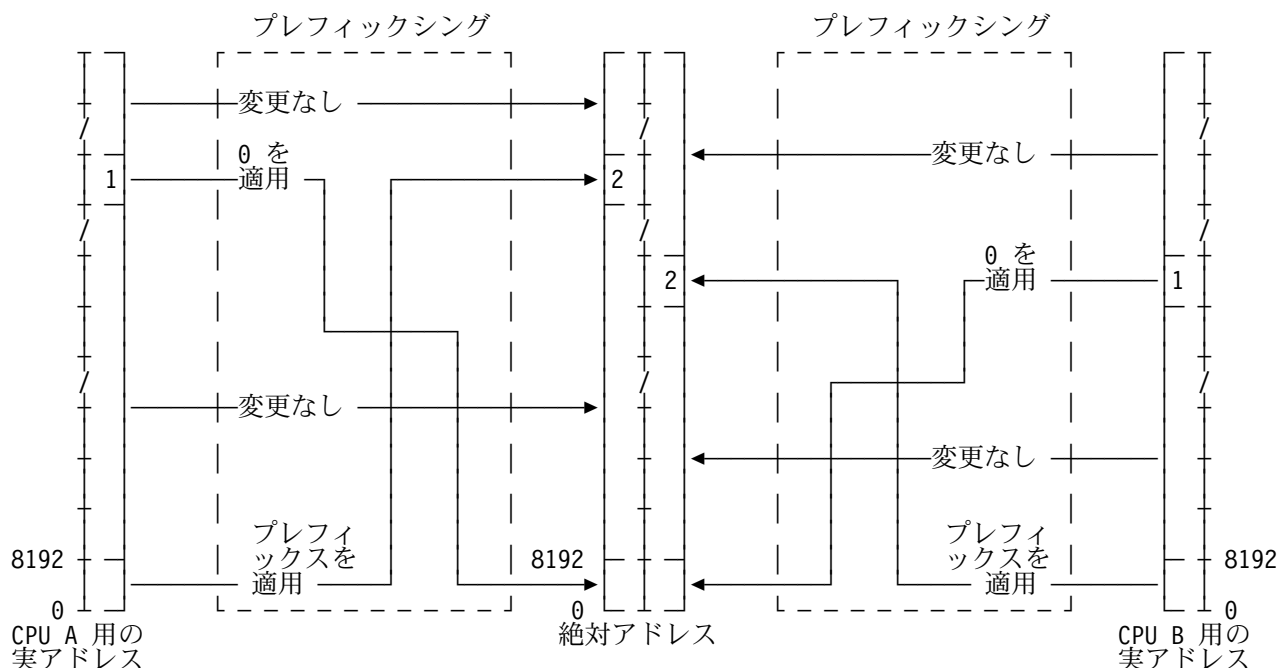
プレフィックス・レジスターの内容を変更した場合、その変更は次の順番に当たる命令が実行される時点で有効になります。

プレフィックシングを適用すると、実アドレスのビット 0-50 に応じて、以下のいずれかの規則を使用して、実アドレスが絶対アドレスに変換されます。

1. アドレスのビット 0-50 がすべて 0 である場合は、プレフィックスのビット 0-50 で置き換えられません。
2. アドレスのビット 0-50 がプレフィックスのビット 0-50 に等しい場合は、すべて 0 で置き換えられません。
3. アドレスのビット 0-50 がすべて 0 ではなく、プレフィックスのビット 0-50 に等しくもない場合は、変更されません。

プレフィックシングにより変換されるのは、記憶機構に対して提示されるアドレスのみです。アドレスのソースの内容は、変更しないまま残されます。

実アドレスと絶対アドレスの区別は、プレフィックス・レジスターの内容がすべて 0 の場合でも行われます。その場合、実アドレスとそれに対応する絶対アドレスは等しくなります。



(1) ビット 0-50 がこの CPU (A または B) 用のプレフィックスのビット 0-50 に等しい実アドレス。

(2) この CPU (A または B) 用の実記憶位置 0-8191 を含むブロックの絶対アドレス。

図 3-5. 実アドレスと絶対アドレスの関係

アドレス・スペース

アドレス・スペースは、連続した整数番号のシーケンス (仮想アドレス) と、各番号を記憶機構内の特定のバイト位置関連付けるための変換パラメーターから成っています。シーケンスは 0 から始まり、左から右へ進みます。

CPU が仮想アドレスを使用して主記憶機構にアクセスしようとする時、その仮想アドレスは、まず動的アドレス変換 (DAT) により実アドレスに変換され、次にプレフィックシングにより絶対アドレスに変換されます。

DAT は、5 レベルから 2 レベルのテーブル (領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、ページ・テーブル) を、変換パラメーターとして使用できます。特定のアドレス・スペース用の最高レベルのテーブルの指定 (起点と長さ) を、アドレス・スペース制御エレメントと呼びます。これは、DAT により使用できるように、制御レジスターに入っているか、アクセス・レジスターに指定されています。また、アドレス・スペースのアドレス・スペース制御エレメントは、実スペース指定のこともあります。実スペース指定は、DAT が、どのテーブルも使用せずに、仮想アドレスをそのまま実アドレスと見なすことを示します。

DAT は、そのときに応じて、異なる制御レジスターに入っているアドレス・スペース制御エレメント、またはアクセス・レジスターに指定されているアドレス・スペース制御エレメントを使用します。どれを選択するかは、現 PSW に指定されている変換モードによって決まります。使用可能な変換モードには、1 次スペース・モード、2 次スペース・モード、アクセス・レジスター・モード、ホーム・スペース・モードの 4 つがあります。変換モードに応じて、それぞれアドレッシング可能なアドレス・スペースが異なります。

CPU は、1 次スペース・モードまたは 2 次スペース・モードにあるときは、2 つのアドレス・スペース (1 次アドレス・スペースと 2 次アドレス・スペース) に属する仮想アドレスを変換できます。アクセス・レジスター・モードになっているときは、CPU は、最大 16 個のアドレス・スペース (1 次アドレス・スペースと最大 15 個の AR 指定アドレス・スペース) の仮想アドレスを変換できます。ホーム・スペース・モードになっている CPU は、ホーム・アドレス・スペースの仮想アドレスを変換できます。

1 次仮想アドレスから成っているアドレス・スペースは、1 次アドレス・スペースとして識別されます。1 次仮想アドレスは、1 次アドレス・スペース制御エレメント (ASCE) を使用して変換されます。同様に、2 次アド

レス・スペースは、2次 ASCE を使用して変換される 2 次仮想アドレスから成り、AR 指定アドレス・スペースは、AR 指定 ASCE を使用して変換される AR 指定アドレス・スペースから成り、そして、ホーム・アドレス・スペースは、ホーム ASCE を使用して変換されるホーム仮想アドレスから成っています。1 次 ASCE と 2 次 ASCE は、それぞれ制御レジスター 1 および 7 に入っています。AR 指定の ASCE は、制御レジスター 1 および 7 と、ASN 第 2 テーブル・エントリーと呼ばれるテーブル・エントリーに入っています。ホーム ASCE は、制御レジスター 13 に入っています。

異なるアドレス・スペースへの変更

プログラムで準特権命令 SET ADDRESS SPACE CONTROL または SET ADDRESS SPACE

CONTROL FAST を使用して、変換モードを 1 次スペース・モード、2 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードに変更することにより、さまざまな異なるアドレス・スペースをアドレッシング可能にすることができます。ただし、SET ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL FAST を使用してホーム・スペース・モードを設定できるのは、監視プログラム状態にあるときだけです。プログラムで他の準特権命令を使用して、制御レジスター 1 および 7 の中のアドレス・スペース制御エレメントを変更したり、非特権命令を使用してアクセス・レジスターの内容を変更したりすることによって、他のアドレス・スペースをアドレッシング可能にすることもできます。制御レジスター 13 の中のホーム・アドレス・スペース制御エレメントを変更できるのは、特権命令 LOAD CONTROL のみです。

アドレス・スペース番号

制御プログラムにより、アドレス・スペースにアドレス・スペース番号 (ASN) を割り当てることができます。ASN は、主記憶機構内の 2 レベルのテーブル構造の中に、アドレス・スペースに関する情報を含む ASN 第 2 テーブル・エントリーを指定します。ASN 第 2 テーブル・エントリーに有効のマークが付けられている場合は、そのエントリーには、アドレス・スペースを定義するアドレス・スペース制御エレメントが入っています。

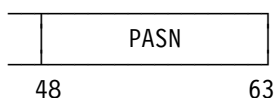
一定の状況下では、新規のアドレス・スペース制御エレメントを制御レジスター 1 または 7 に入れる準特権命令は、ASN 第 2 テーブル・エントリーからそのエレメントを取り出します。この種の命令の中には、ASN 変換メカニズムを使用するものが幾つかあります。ASN

変換命令は、与えられた ASN に基づいて、指定された ASN 第 2 テーブル・エントリーを見つけることができます。

ASN 16 ビットの符号なし 2 進数フォーマットであり、これにより 64K の固有の ASN が使用できるようになります。

1 次および 2 次アドレス・スペース用の ASN には、制御レジスター内のビット位置が割り当てられています。1 次アドレス・スペースの ASN (1 次 ASN) には、制御レジスター 4 のビット 48-63 が割り当てられ、2 次アドレス・スペースの ASN (2 次 ASN) には、制御レジスター 3 のビット 48-63 が割り当てられます。これらのレジスターのフォーマットは以下のとおりです。

制御レジスター 4



制御レジスター 3



1 次または 2 次アドレス・スペース・エレメントを該当の制御レジスターにロードする準特権命令は、対応する ASN もそれぞれ該当の制御レジスターにロードします。

ホーム・アドレス・スペースの ASN には、制御レジスター内のビット位置は割り当てられません。

0 または 1 の値を含むアクセス・レジスターは、それぞれ 1 次または 2 次アドレス・スペースを指定しており、そのアクセス・レジスターで指定されるアドレス・スペース制御エレメントは、それぞれ制御レジスター 1 または 7 に入っています。その他の値を含むアクセス・レジスターは、アクセス・リストと呼ばれるテーブル内のエントリーを指定します。指定されたアクセス・リスト・エントリーには、アクセス・レジスターで指定されたアドレス・スペース用の ASN 第 2 テーブル・エントリーの実アドレスが入っています。アクセス・レジスターで指定されるアドレス・スペース制御エレメントは、ASN 第 2 テーブル・エントリーに入っています。アクセス・レジスターの内容を変換して、DAT で使用するためのアドレス・スペース制御エレメントを入手する際には、ASN は使用されません。

注: アドレス・スペース内の仮想アドレスの順序に従って並べられたバイト位置から成る仮想記憶域を、一般に「記憶域」と呼びます。

プログラミング上の注意: ASN 第 2 テーブル・エントリーは、ASN ではなくアドレスを使用してアクセス・リスト・エントリーから突き止められるので、アクセス・リスト・エントリーが示す ASN 第 2 テーブル・エントリーは、「疑似」ASN 第 2 テーブル・エントリーとすることができます。疑似エントリーとは、ASN 変換プロセス指標付けできる 2 レベル構造の中に入らないエントリーです。固有の疑似 ASN 第 2 テーブル・エントリーの数は、固有の ASN の数より多くすることができます。この数は、ASN 第 2 テーブル・エントリー用として使用できる記憶容量によって制限されるだけです。したがって、ある意味では、可能なアドレス・スペースの数には制限はないと言えます。

ASN 変換

ASN 変換とは、ASN が示す ASN 第 2 テーブル・エントリーを見つけるために、16 ビットの ASN を変更するプロセスです。ASN 変換は、スペース切り替えを伴う PROGRAM TRANSFER (PT-ss) およびスペース切り替えを伴う SET SECONDARY ASN (SSAR-ss) の一環として行われるほか、LOAD ADDRESS SPACE PARAMETERS の一環として行われることもあります。PT-ss では、変換された ASN は制御レジスター 4 の中の 1 次 ASN と置き換わります。SSAR-ss では、変換された ASN は制御レジスター 3 の中の 2 次 ASN と置き換わります。この 2 つの変換プロセスは、それぞれ 1 次 ASN 変換および 2 次 ASN 変換と呼ばれるもので、LOAD ADDRESS SPACE PARAMETERS ではこの両方を行うことができます。ASN 変換プロセスは、1 次および 2 次のどちらの ASN 変換の場合も同じであり、プロセスの結果をどのように使用するかが異なるだけです。

ASN 変換は、PROGRAM RETURN の一環として行うこともできます。1 次 ASN 変換は、スペース切り替えを伴う PROGRAM RETURN (PR-ss) の一環として行われます。2 次 ASN 変換が行われるのは、PROGRAM RETURN (PR-ss、または現行の 1 次への PROGRAM RETURN) により復元された 2 次 ASN が、PROGRAM RETURN により復元された 1 次 ASN と等しくない場合です。

スペース切り替えを伴う PROGRAM CALL (PC-ss) は、1 次 ASN と、それに対応する ASN 第 2 テーブル

ル・エントリーをエントリー・テーブル・エントリーから入手することにより、1 次 ASN 変換と同様の処理を実行します。

ASN 変換プロセスは、ASN 第 1 テーブルおよび ASN 第 2 テーブルの 2 つのテーブルを使用します。これらのテーブルは、ASN 第 2 テーブル・エントリーを見つけるためと、ASN 許可を行うときに使用する、権限テーブルと呼ばれる第 3 のテーブルを見つけるために使用されます。

この変換のために、16 ビットの ASN は 2 つの部分から成っているものと見なされます。1 つは ASN 第 1 テーブル指標 (AFX) で、これは ASN の左側の 10 ビットです。もう 1 つは ASN 第 2 テーブル指標 (ASX) で、これは右側の 6 ビットです。ASN のフォーマットは以下のようになります。

ASN

	AFX		ASX	
0		10		15

AFX は、ASN 第 1 テーブルからエントリーを選択するために使用されます。ASN 第 1 テーブルの起点は、制御レジスター 14 の中の ASN 第 1 テーブル起点により指定されます。ASN 第 1 テーブル・エントリーには、ASN 第 2 テーブルの起点が含まれています。ASX は、ASN 第 2 テーブルからエントリーを選択するために使用されます。

1 次 ASN 変換の結果として、またスペース切り替えを伴う PROGRAM CALL の実行時に、検出された ASN 第 2 テーブル・エントリー (ASTE) が新しい 1 次 ASTE 起点 (PASTE0) として制御レジスター 5 に入れます。

ASN 変換制御

ASN 変換は、ASN 変換制御ビットと ASN 第 1 テーブル起点により制御されます。これらはどちらも制御レジスター 14 に入っています。

制御レジスター 14

	T		AFT0	
44				63

ASN 変換制御 (T): 制御レジスター 14 のビット 44 は、ASN 変換制御ビットです。このビットが持つメカニズムにより、制御プログラムは、特定プログラムの実行中に ASN 変換を行うことができるかどうか、および、スペース切り替えを伴う PROGRAM CALL の実行が許されるかどうかを示すことができます。以下の命令が実行可能なのは、ビット 44 が 1 になっているときです。

- LOAD ADDRESS SPACE PARAMETERS
- スペース切り替えを伴う PROGRAM CALL
- スペース切り替えを伴う PROGRAM RETURN、または復元された SASN が復元された PASN に等しくないときの PROGRAM RETURN
- スペース切り替えを伴う PROGRAM TRANSFER
- SET SECONDARY ASN

これらの命令について値が 1 でない場合は、特殊操作例外が認識されます。ASN 変換制御ビットは、問題プログラム状態と監視プログラム状態のどちらの場合も検査されます。

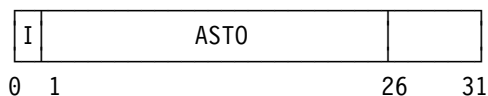
ASN 第 1 テーブル起点 (AFTO): 制御レジスター 14 のビット 45-63 と、その右側に付加された 12 個の 0 は、ASN 第 1 テーブルの始めを示す 31 ビットの実アドレスを形成します。

ASN 変換テーブル

ASN 変換プロセスは、ASN 第 1 テーブルおよび ASN 第 2 テーブルの 2 つのテーブルを使用した 2 レベルの索引から成ります。これらのテーブルは実記憶域に入っています。

ASN 第 1 テーブル・エン트리

ASN 第 1 テーブルのエントリーのフォーマットは以下のとおりです。



エン트리内では、次のようにフィールドが割り振られます。

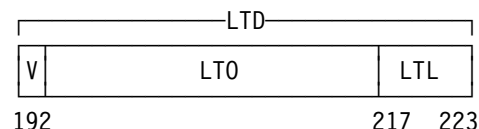
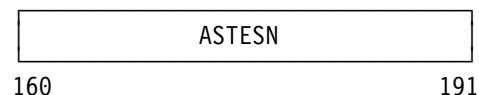
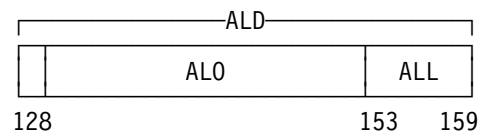
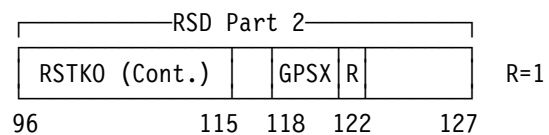
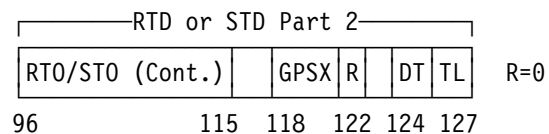
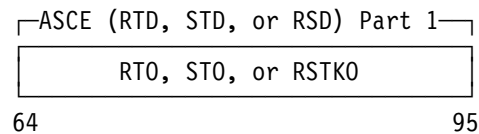
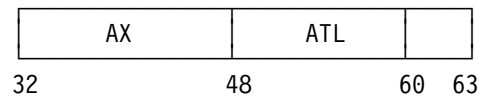
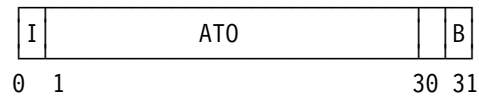
AFX 無効ビット (I): ビット 0 は、ASN 第 1 テーブル・エントリーに関連付けられている ASN 第 2 テーブルが使用可能かどうかを制御します。ビット 0 が 0 のときは、指定された ASN 第 2 テーブルを使用して

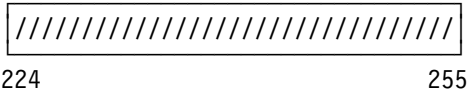
ASN 変換が進められます。このビットが 1 であれば、ASN 変換は続行できません。

ASN 第 2 テーブル起点 (ASTO): ビット 1-25 と、その右側に付加された 6 個の 0 は、ASN 第 2 テーブルの始めを示す 31 ビットの実アドレスを形成します。

ASN 第 2 テーブル・エン트리

ASN 第 2 テーブル・エントリーは長さが 64 バイトですが、現在使用されているのは最初の 32 バイトのみです。このエントリーのバイト 0-31 のフォーマットは以下のとおりです。





ASN 第 2 テーブル・エントリーのバイト 0-31 には、以下のようにフィールドが割り振られています。ここでは、ASN 変換または PROGRAM CALL (スペース切り替えを伴う) の中で、またはその結果として使用されるフィールドについてのみ、詳しく説明します。

ASX 無効ビット (I): ビット 0 は、ASN 第 2 テーブル・エントリーに関連付けられているアドレス・スペースが使用可能かどうかを制御します。ビット 0 が 0 のときは、ASN 変換が進められます。このビットが 1 であれば、ASN 変換は続行できません。

権限テーブル起点 (ATO): ビット 1-29 とその右側に付加された 2 個の 0 は、権限テーブルの始めを示す 31 ビットの実アドレスを形成します。

基本スペース・ビット (B): ビット 31 が 1 のときは、ASN 第 2 テーブル・エントリーに関連付けられているアドレス・スペースが、サブスペース・グループの基本スペースであることを示します。ビット 31 については、5-57ページの『サブスペース・グループ ASN 第 2 テーブル・エントリー』でさらに詳しく説明します。

許可指標 (AX): ビット 32-47 は、ASN 許可において、権限テーブル内で権限ビットを見つけるための指標として使用されます。この AX フィールドは、PROGRAM RETURN および PROGRAM TRANSFER による 1 次 ASN 変換、および場合によっては LOAD ADDRESS SPACE PARAMETERS による 1 次 ASN 変換の結果として使用されます。また、スペース切り替えを伴う PROGRAM CALL でも AX フィールドが使用されます。2 次 ASN の後は、AX フィールドは無視されます。

権限テーブルの長さ (ATL): ビット 48-59 は、権限テーブルの長さを 4 バイト単位で指定します。したがって、権限テーブルは可変長で、長さは 16 エントリーの倍数になります。権限テーブルの長さ (4 バイト単位) は、ATL の値に 1 を加えた値になります。ATL フィールドの内容は、特定の AX が示すエントリーが権限テーブル内に含まれているかどうかを確定するために使用されます。

アドレス・スペース制御エレメント(ASCE): ビット 64-127 は、8 バイトのアドレス・スペース制御エレメント (ASCE) で、これは領域テーブル指定 (RTD)、セグメント・テーブル指定 (STD)、または実スペース指定 (RSD) のいずれかです。(「領域テーブル指定」という用語は、領域第 1 テーブル指定、領域第 2 テーブル指定、または領域第 3 テーブル指定を意味します。)

ASCE フィールドは、ASN 変換の結果として、またはスペース切り替えを伴う PROGRAM CALL の中で、1 次 ASCE (PASCE) または 2 次 ASCE (SASCE) を置き換えるために使用されます。スペース切り替えを伴う PROGRAM CALL の場合は、ASCE フィールドは、PASCE (制御レジスター 1 のビット 0-63) と置き換わります。SET SECONDARY ASN の場合は、ASCE フィールドは、SASCE (制御レジスター 7 のビット 0-63) と置き換わります。LOAD ADDRESS SPACE PARAMETERS では、これらのアクションをそれぞれ独立して実行されることもあります。PROGRAM TRANSFER の場合は、ASCE フィールドは PASCE および SASCE の両方と置き換わります。PROGRAM RETURN の場合は、ASCE フィールドは、1 次 ASN 変換の結果として PASCE と置き換わり、2 次 ASN 変換の結果として SASCE と置き換わります。ASCE フィールド全体の内容は、妥当性検査を行わずに該当の制御レジスターに入れられます。

ASCE フィールドのビット 118 はサブスペース・グループ制御ビット (G) で、このビットが 1 のときは、ASCE が、基本スペースか、またはサブスペース・グループのサブスペースを指定することを意味します。このビットについては、5-57ページの『サブスペース・グループ ASN 第 2 テーブル・エントリー』でさらに詳しく説明します。

ASCE フィールドのビット 121 (X) は、スペース切り替えイベント制御ビットです。PROGRAM CALL、PROGRAM RETURN、および PROGRAM TRANSFER のスペース切り替え操作において、これらの命令の実行前または実行後に制御レジスター 1 の中でこのビットが 1 のときは、命令の実行が完了した後で、スペース切り替えイベントのためのプログラム割り込みが生じます。また、変換モードをホーム・スペース・モードへ (またはそのモードから) 変更する SET ADDRESS SPACE CONTROL、SET ADDRESS SPACE CONTROL FAST、または RESUME PROGRAM 命令において、制御レジスター 1 または制御レジスター 13 の中でこのビットが 1 である場合も、スペース切り替えイベント・プログラム割り込みが起こります。LOAD ADDRESS SPACE PARAMETERS

では、1 次 ASN 変換の際にこのビットが 1 である場合は、そのことが条件コードで示されます。

ASCE フィールドのビット 122 は実スペース制御ビット (R) で、このビットが 0 のときは、ASCE が領域テーブル指定またはセグメント・テーブル指定であることを示し、このビットが 1 のときは、ASCE が実スペース指定であることを示します。

ビット 122 が 0 のときは、指定タイプ制御ビット (DT) (ASCE フィールドのビット 124 および 125) は、ASCE の指定タイプを示します。ビット 124 および 125 の値は 2 進数 11、10、01、または 00 のいずれかで、これらは、それぞれ、領域第 1 テーブル指定、領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定を示します。

ASCE のその他のフィールド (RTO、STO、P、S、TL、および RSTKO) については、3-27ページの『制御レジスター 1』で説明します。

ASN 第 2 テーブル・エンタリーのリンケージ・テーブル指定 (LTD) フィールドについては、5-28ページの『PC 番号変換制御』で説明します。アドレス・リスト指定 (ALD) フィールドと ASTE シーケンス番号 (ASTESN) フィールドについては、5-46ページの『ASN 第 2 テーブル・エンタリー』で説明します。ASN 第 2 テーブル・エンタリーのビット 224-255 は、プログラミング用として使用できます。

プログラミング上の注意: バイト 0-31 の未使用フィールドおよびバイト 32-63 のすべてを含めて、ASN 第 2 テーブル・エンタリーの未使用フィールドは、すべて 0 に設定する必要があります。これらのフィールドは将来の拡張のための予約されており、これらのフィールドに非ゼロ値を入れるプログラムは、将来のマシンでは互換性を保って動作することができなくなるおそれがあります。

ASN 変換プロセス

このセクションでは、スペース切り替えを伴う PROGRAM RETURN、PROGRAM TRANSFER、および SET SECONDARY ASN の実行時、および、復元された 2 次 ASN が復元された 1 次 ASN に等しくない場合の PROGRAM RETURN の実行時に行われる ASN 変換プロセスについて説明します。LOAD ADDRESS SPACE PARAMETERS の場合の ASN 変換も同じですが、AFX 変換例外および ASX 変換例外は起きず、代わりに条件コードにより、その状態が示されるという点が異なります。ASN の変換は、ASN 第 1 テーブルおよび ASN 第 2 テーブルという 2 つのテーブルを使用して行われます。これらのテーブルは、どちらも主記憶機構内に常駐しています。

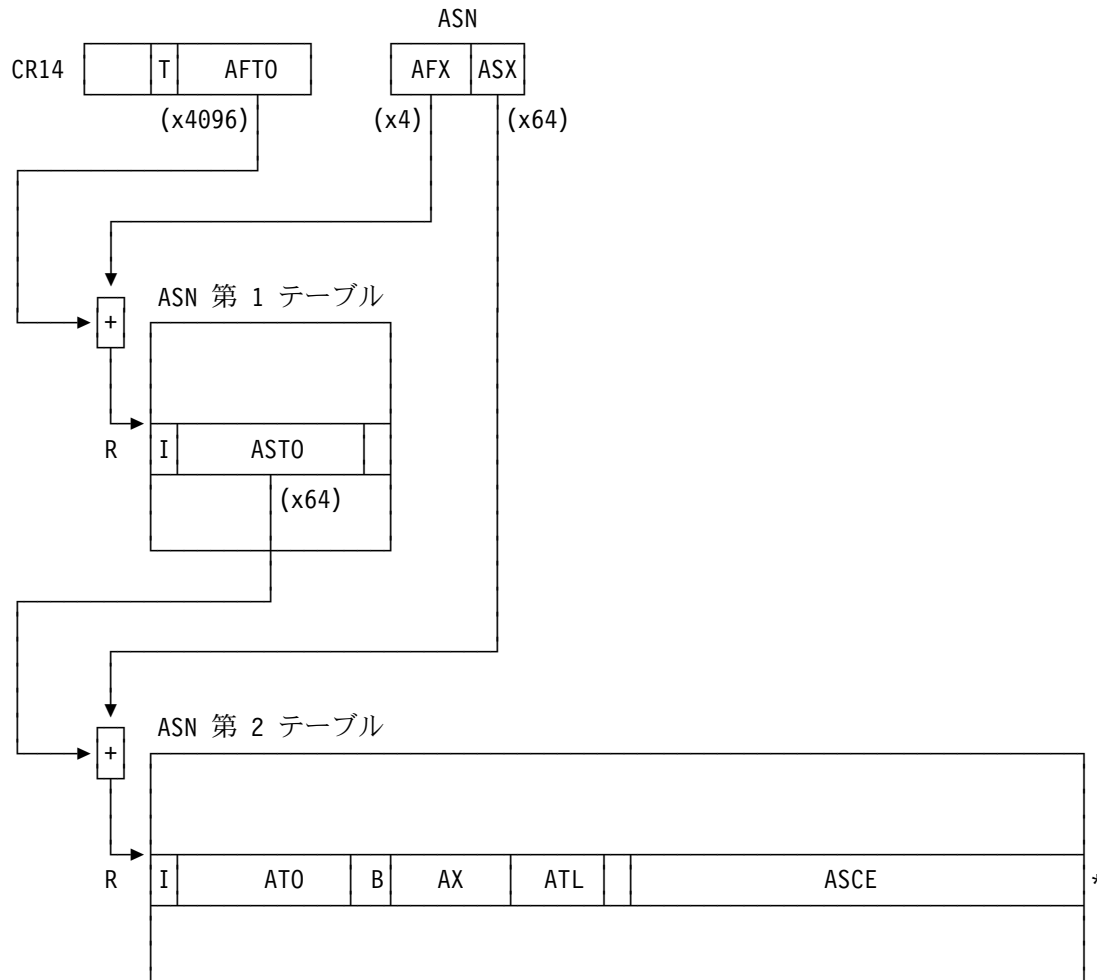
ASN 第 1 指標は、ASN 第 1 テーブルからエンタリーを選択するために使用されます。このエンタリーは、使用する ASN 第 2 テーブルを指示します。

ASN 第 2 指標は、ASN 第 2 テーブルからエンタリーを選択するために使用されます。

ASN 第 1 テーブル・エンタリーまたは ASN 第 2 テーブル・エンタリーのいずれかの I ビットが 1 であるときは、そのエンタリーは無効であり、ASN 変換プロセスは完了できません。この場合は、AFX 変換例外または ASX 変換例外が認識されます。

ASN 変換プロセスで、ASN 第 1 テーブルまたは ASN 第 2 テーブルからエンタリーを取り出すために主記憶機構にアクセスするときは、キー制御保護は適用されません。

3-21ページの図3-6 は、ASN 変換プロセスを示しています。



R: アドレスは実アドレスです。
 *: ASTE の最後の 48 バイトは示されていません。

図 3-6. ASN 変換

ASN 第 1 テーブル索引

ASN の AFX 部分と ASN 第 1 テーブル起点を使用し、ASN 第 1 テーブルからエントリーが選択されます。

ASN 第 1 テーブル・エントリーの 31 ビットの実アドレスは、制御レジスター 14 のビット位置 45-63 に含まれている AFT 起点の右側に 12 個の 0 を付加し、さらに、右側に 2 個の 0、そして左側に 19 個の 0 を付加した AFX 部分を加算したものです。この加算により、ビット位置 0 への繰り上がりが生じることはありません。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

他の CPU からは、ASN 第 1 テーブル・エントリーの 4 バイトすべてが同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。ASN 第 1 テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

4 バイトの AFT エントリーのビット 0 は、対応する AST が使用可能かどうかを指定します。このビットが 1 の場合は、AFX 変換例外が認識されます。AFT から取り出されたエントリーが、AST にアクセスするために使用されます。

ASN 第 2 テーブル索引

ASN の ASX 部分と、ASN 第 1 テーブル・エントリーに含まれている ASN 第 2 テーブル起点を使用して、ASN 第 2 テーブルからエントリーが選択されます。

ASN 第 2 テーブル・エントリーの 31 ビットの実アドレスは、ASN 第 1 テーブル・エントリーのビット 1-25 の右側に 6 個の 0 を付加し、さらに、右側に 6 個、そして左側に 19 個の 0 を付加した ASX を加算したものです。この加算によりビット位置 0 への繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{31} - 1$ から 0 へと循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

ASN 第 2 テーブル・エントリーの 64 バイトの取り出しは、他の CPU からはワード単位のように見え、左端のワードが最初に取り出されます。残りの 15 ワードが取り出される順序は、予測不能です。この取り出しアクセスは記憶保護の対象にはなりません。ASN 第 2 テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

ASN 第 2 テーブル・エントリーのビット 0 は、アドレス・スペースがアクセス可能かどうかを指定します。このビットが 1 の場合は、ASX 変換例外が認識されます。

ASN 変換時の例外の認識

ASN 変換プロセス中に検出される例外は、総称的に ASN 変換例外と呼ばれます。これらの例外のリストとそれぞれの優先順位については、第 6 章、『割り込み』を参照してください。

ASN 許可

ASN 許可は、現行の許可指標に関連付けられているプログラムが、特定のアドレス・スペースを設定することを許されているかどうかを確認するプロセスです。ASN 許可は、スペース切り替えを伴う PROGRAM TRANSFER 命令 (PT-ss) およびスペース切り替えを伴う SET SECONDARY ASN 命令 (SSAR-ss) の一環として行われるほか、LOAD ADDRESS SPACE PARAMETERS 命令の一環として行われることもあり

ます。これらの命令では、ASN 変換プロセスの後に ASN 許可が行われます。

PROGRAM RETURN 命令でも、復元された 2 次 ASN が復元された 1 次 ASN に等しくない場合に、ASN 許可が行われます。復元された 2 次 ASN の ASN 許可は、復元された 2 次 ASN の ASN 変換の後で行われます。

PT-ss の一部として行われる ASN 許可は、ASN が 1 次 ASN として設定できるかどうかを検査するためのものであり、1 次 ASN 許可と呼ばれます。LOAD ADDRESS SPACE PARAMETERS、PROGRAM RETURN、または SSAR-ss の一部として行われる ASN 許可は、ASN が 2 次 ASN として設定できるかどうかを検査するものであり、2 次 ASN 許可と呼ばれます。

ASN 許可は、実記憶域内の権限テーブルを使用して行われます。権限テーブルは、ASN 第 2 テーブル・エントリー内の権限テーブル起点フィールドと権限テーブル長さフィールドで指定されます。

ASN 許可制御

ASN 許可では、許可指標と共に、ASN 第 2 テーブル・エントリーの中の権限テーブル起点および権限テーブル長さが使用されます。

制御レジスター 4

PT-ss および SSAR-ss の場合は、制御レジスター 4 の現在の内容に許可指標が含まれています。LOAD ADDRESS SPACE PARAMETERS および PROGRAM RETURN の場合は、制御レジスター 4 の新しい内容となる値が使用されます。このレジスターのフォーマットは以下のとおりです。

AX	
32	48

許可指標 (AX): 制御レジスター 4 のビット 32-47 は、権限テーブル内の権限ビットを見つけるための指標として使用されます。

ASN 第 2 テーブル・エントリー

ASN 変換プロセスの一環として取り出される ASN 第 2 テーブル・エントリーには、権限テーブルを指定するために使用する情報が含まれています。ASN 第 2 テーブルのエントリーのフォーマットは以下のとおりです。

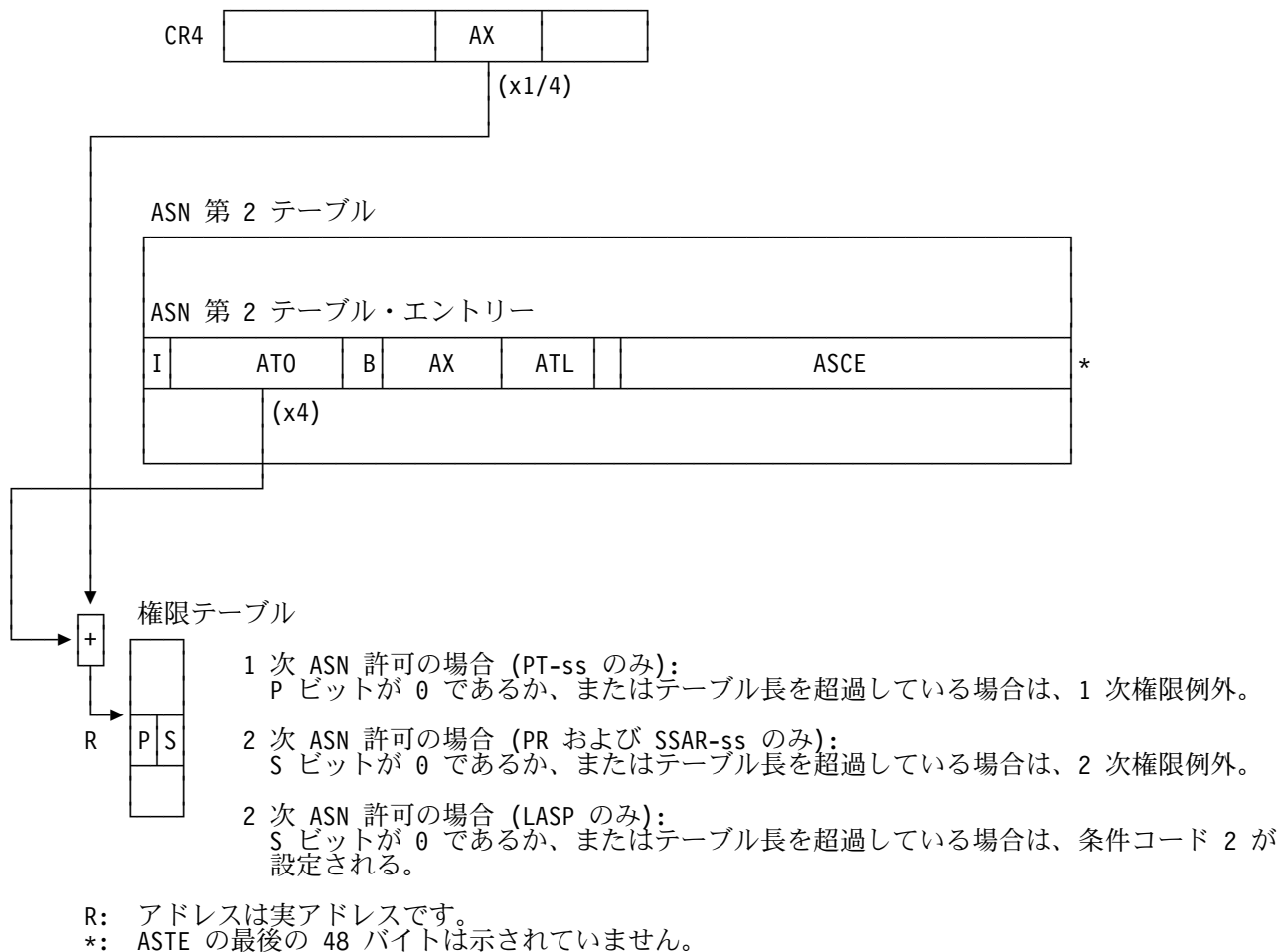


図 3-7. ASN 許可

権限テーブルの索引

許可指標と共に、ASN 第 2 テーブル・エントリーに含まれている権限テーブル起点を使用して、権限テーブルからエントリーが選択されます。

許可指標は、制御レジスター 4 のビット位置 32-47 に含まれています。

AST エントリーのビット位置 1-29 には、権限テーブルの 31 ビットの実アドレス (ATO) が含まれ、ビット位置 48-59 には、権限テーブルの長さ (ATL) が含まれています。

権限テーブル内の 1 つのバイトの 31 ビット実アドレスは、権限テーブルの起点の右側に 2 個の 0 を付加し、さらに許可指標の左端 14 ビットの左側に 17 個の 0 を付加したものを加算して、得られます。この加算によりビット位置 0 への繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{31} - 1$ から 0 へと循環します。現 PSW が、24 ビット、31 ビット、または 64 ビット

のいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

権限テーブル・エントリー索引プロセスの一環として、許可指標のビット 0-11 が権限テーブルの長さと比較されます。比較対象部分が権限テーブルの長さより大きいときは、PT-ss の場合は 1 次権限例外、そして SSAR-ss の場合は 2 次権限例外が認識されます。LOAD ADDRESS SPACE PARAMETERS の場合は、権限テーブルの長さを超過しているときは、条件コード 2 が設定されます。

権限テーブル内のバイトに対する取り出しアクセスは、保護の対象にはされません。バイトを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

このバイトには、それぞれ 2 ビットの権限テーブル・エントリーが 4 個含まれています。許可指標の左端 2 ビ

ット、つまり制御レジスター 4 のビット 46 および 47 を使用して、この 4 つのエントリーの 1 つが選択されます。次に、許可検査の対象が 1 次 ASN の場合は左ビット、2 次 ASN の場合は右ビットが検査されます。次の表は、PT-ss、SSAR-ss、PROGRAM RETURN、または LOAD ADDRESS SPACE PARAMETERS において、許可指標のビット 46 および 47 の値に応じてどのビットが選択されるかを示しています。

許可指標のビット 46 47		検査対象として権限テーブルのバイトから 選択されるビット	
		P ビット (PT-ss)	S ビット (SSAR-ss、 PR、または LASP)
0	0	0	1
0	1	2	3
1	0	4	5
1	1	6	7

選択されたビットが 1 であれば、ASN は許可され、AST エントリー内の該当フィールドが該当の制御レジスターにロードされます。選択されたビットが 0 であれば、ASN は許可されず、PT-ss の場合は 1 次権限例外が認識され、SSAR-ss または PROGRAM RETURN の場合は 2 次権限例外が認識されます。LOAD ADDRESS SPACE PARAMETERS では、ASN が許可されない場合は条件コード 2 が設定されます。

ASN 許可時の例外の認識

1 次または 2 次 ASN 許可プロセスで検出される各種例外と、それぞれの優先順位については、ASN 許可が行われる各命令の定義についての記述の中で説明します。

プログラミング上の注意: 1 次および 2 次権限例外が生じると、権限テーブルの動的変更ができるように、無効化が行われます。したがって、アドレス・スペースの作成時または「スワップイン」の時点ではまず権限テーブルの値をすべて 0 に設定しておいて、必要に応じて該当する権限ビットのみを 1 に設定することができます。

動的アドレス変換

動的アドレス変換 (DAT) を使用すると、任意の時点でプログラムの実行に割り込み、そのプログラムおよび関連データを直接アクセス記憶装置などの補助記憶装置に記録しておき、後で、そのプログラムとデータを主記憶機構内の別の場所に戻して、実行を再開することができます。

ます。主記憶機構と補助記憶機構間のプログラムとデータの転送は断片的に行われ、CPU がこれらの情報にアクセスして実行に使用することが必要になった時点で、主記憶機構に情報が戻されます。これらの操作は、プログラムとデータの変更または検査を必要とせずに実行でき、再配置されたプログラムについては明示的なプログラミング上の規則は何も適用されず、時間的遅延が生じる以外にはプログラムの実行が妨げられることもまったくありません。

動的アドレス変換機能がオペレーティング・システムによる適切なサポートを受けていれば、ユーザーは、システム構成の中で使用可能にされている主記憶機構より大きい記憶域を利用することができます。このような見かけの主記憶機構を仮想記憶機構と呼び、仮想記憶機構内の場所を指定するために使用されるアドレスを仮想アドレスと呼びます。1 人のユーザーに、構成内で使用可能な主記憶機構のサイズをはるかに上回る仮想記憶域を供給することが可能であり、通常この記憶域は補助記憶装置内に維持されます。仮想記憶機構は、ページと呼ばれるアドレスのブロックから成るものと見なされます。仮想記憶機構の中の最近参照されたページのみが、物理的な主記憶機構のブロックを占有するものとして割り当てられます。ユーザーが、主記憶機構内には存在しない仮想記憶機構ページを参照すると、そのページは主記憶機構に取り込まれて、必要度が低いと認められるページと置き換わります。このような記憶機構ページのスワップは、ユーザーに認識されない形でオペレーティング・システムが行うことができます。

1 つの仮想記憶機構に関連した仮想アドレスのシーケンスを、アドレス・スペースと呼びます。オペレーティング・システムによる適切なサポートがあれば、動的アドレス変換機能を使用して多数のアドレス・スペースを供給することができます。アドレス・スペースを使用することにより、ユーザー間の独立性の程度を高めることができます。このサポートにより、各ユーザーごとにそれぞれ別個のアドレス・スペースを用意して、完全な独立性を達成することができるほか、各アドレス・スペースの一部を 1 つの共通記憶域にマッピングすることにより、共用記憶域を供給することも可能です。さらに、準特権プログラムがこのようなアドレス・スペースの複数にアクセスできるようにするための命令も用意されています。動的アドレス変換では、制御レジスター内の変換パラメータを変更することなく、複数の異なるアドレス・スペースから仮想アドレスを変換できます。これらのアドレス・スペースは、1 次アドレス・スペース、2 次アドレス・スペース、および AR 指定アドレス・スペースと呼ばれます。特権プログラムでは、ホーム・アドレス・スペースへのアクセスを可能になります。

主記憶機構内のブロックを外部メディアからの新しい情報で置き換えるプロセスでは、どのブロックを置き換えるか、およびその置き換えられたブロックを補助記憶装

置に記録して保存する必要があるかどうかを、決定する必要があります。この決定プロセスを促進するために、記憶キーに参照ビットと変更ビットが関連付けられています。

動的アドレス変換は、CPU により生成される命令およびデータ・アドレスに対しては指定できますが、データのアドレッシングや、入出力操作での CCW および IDAW のアドレッシングには使用できません。仮想記憶環境での入出力操作を補助するためには、CCW 間接データ・アドレッシング機能が用意されています。

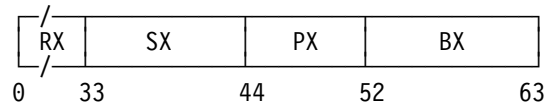
アドレス計算は、24 ビット、31 ビット、または 64 ビットのアドレッシング・モードで実行できます。アドレス計算が 24 ビットまたは 31 ビットのアドレッシング・モードで行われる場合は、それぞれ左側に 40 個または 33 個の 0 を付加して、64 ビット・アドレスが形成されます。したがって、計算結果の論理アドレスは常に 64 ビットの長さになります。動的アドレス変換により作られる実アドレス、およびその後でプレフィックシングにより作られる絶対アドレスは、常に 64 ビットの長さです。

動的アドレス変換は、記憶機構参照の際に、仮想アドレスをそれに対応する実アドレスに変換するプロセスです。仮想アドレスには、1 次仮想アドレス、2 次仮想アドレス、AR 指定仮想アドレス、ホーム仮想アドレスがあります。これらのアドレスは、それぞれ、1 次、2 次、AR 指定、またはホームの各アドレス・スペース制御エレメントを使用して変換されます。該当のアドレス・スペース制御エレメントが選択された後の変換プロセスは、4 つのタイプの仮想アドレスのどれについても同じです。アドレス・スペース制御エレメントとして使用されるのは、2G バイト・アドレス・スペースを指定するセグメント・テーブル指定か、4T バイト、8P バイト、または 16E バイトのスペースを指定する領域テーブル指定か、または 16E バイトのスペースを指定する実スペース指定です。(K、M、G、T、P、E の文字は、それぞれ、キロ (2^{10})、メガ (2^{20})、ギガ (2^{30})、テラ (2^{40})、ペタ (2^{50})、およびエクサ (2^{60}) を表します)。セグメント・テーブル指定または領域テーブル指定を使用した場合は、オペレーティング・システムが実記憶域または絶対記憶域の中に設定するテーブルを使用して、変換が行われます。実スペース指定の場合は、記憶域内のテーブルを使用せずに、仮想アドレスがそのまま実アドレスとして扱われます。

セグメント・テーブル指定または領域テーブル指定を使用した変換プロセスの対象となる情報単位には、認識済み領域、セグメント、およびページの 3 種類があります。領域とは、2G バイト境界で始まる 2G バイトの順

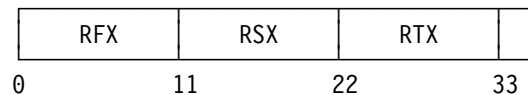
次仮想アドレスのブロックです。セグメントとは、1M バイト境界で始まる 1M バイトの順次仮想アドレスのブロックです。ページとは、4K バイト境界で始まる順次仮想アドレスのブロックです。

したがって、仮想アドレスは、4 つの基本フィールドに分けられます。ビット 0-32 は領域指標 (RX)、ビット 33-43 はセグメント指標 (SX)、ビット 44-51 はページ指標 (PX)、およびビット 52-63 はバイト指標 (BX) と呼ばれます。仮想アドレスのフォーマットは以下のとおりです。



仮想アドレス・スペースは、アドレス・スペース制御エレメントに応じて、1 個の領域から成る 2G バイトのスペースか、最大 8G 個の領域からなる 16E バイトのスペースになります。2G バイト・アドレス・スペースに適用される、仮想アドレスの RX 部分は、すべて 0 でなければなりません。さもないと例外が発生します。

仮想アドレスの RX 部分は、さらに 3 つのフィールドに分かれています。ビット 0-10 は領域第 1 指標 (RFX)、ビット 11-21 は領域第 2 指標 (RSX)、そしてビット 22-32 は領域第 3 指標 (RTX) と呼ばれます。仮想アドレスのビット 0-32 のフォーマットは以下のとおりです。



左端の有効部分が RTX である仮想アドレス (42 ビット・アドレス) では、4T バイト (2K 個の領域) をアドレッシングでき、左端の有効部分が RSX である仮想アドレス (53 ビット・アドレス) では、8P バイト (4M 個の領域) をアドレッシングでき、左端の有効部分が RFX である仮想アドレスでは、16E バイト (8G 個の領域) をアドレッシングできます。

RX が常に 0 である仮想アドレスは、セグメント・テーブルおよびページ・テーブルの 2 つの変換テーブルを使用して、実アドレスに変換できます。RX が非ゼロ値をとる可能性がある場合は、以下に述べるように、1 個から 3 個の追加の変換テーブルが必要です。RFX が非ゼロ値をとる可能性がある場合は、領域第 1 テーブル、領域第 2 テーブル、および領域第 3 テーブルが必要です。RFX は常に 0 で、RSX は 0 以外の可能性がある場合は、領域第 2 テーブルと領域第 3 テーブルが必要です。RFX と RSX は常に 0 で、RTX は 0 以外の可

能性がある場合は、領域第 3 テーブルが必要です。あるアドレス・スペース用のアドレス・スペース制御エレメントが、そのアドレス・スペースに対する参照を変換するために必要な最高レベルのテーブル (領域第 1 テーブルが最高で、セグメント・テーブルが最低) を示していない場合は、例外が認識されます。

領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブルを、単に領域テーブルと呼ぶこともあります。同様に、領域第 1 テーブル指定、領域第 2 テーブル指定、領域第 3 テーブル指定を、単に領域テーブル指定と呼ぶことがあります。

領域テーブル、セグメント・テーブル、ページ・テーブルは、実記憶機構の現在の割り当てを反映します。実記憶機構の割り当てはページを単位として行われ、ページの中には実記憶位置が連続して割り当てられます。ページは、一組の順次仮想アドレスに割り当てられる場合でも、実記憶機構の中で互いに隣接している必要はありません。

パフォーマンスを高めるために、変換は、通常、変換索引バッファ (TLB) と呼ばれる特殊バッファに入っているテーブル・コピーを使用して行われます。TLB には、実スペース指定で指定されている仮想/実記憶域間等価変換のためのエントリーも含まれていることがあります。

変換制御

アドレス変換は、PSW の 3 つのビットと、変換パラメーターと呼ばれる一組のビットにより制御されます。変換パラメーターは、制御レジスタ 0、1、7、および 13 の中にあります。また、変換テーブルにもその他の制御機能が含まれています。

そのほかに、第 5 章、『プログラムの実行』で説明する制御機能があります。これらの制御機能は、DAT で使用するアドレス・スペース・エレメントを入手するために、各アクセス・レジスタの内容を使用できるかどうかを判別します。

変換モード

PSW の中には、動的アドレス変換を制御するビットが 3 つあります。それは、DAT モード・ビットと呼ばれるビット 5 と、アドレス・スペース制御ビットと呼ばれるビット 16 および 17 です。DAT モード・ビットが 0 のときは DAT はオフであり、CPU は実モードになります。DAT モード・ビットが 1 のときは DAT はオンであり、CPU は、アドレス・スペース制御ビットが指

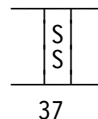
定する変換モードになります。00 は 1 次スペース・モード、01 はアクセス・レジスタ・モード、10 は 2 次スペース・モード、そして 11 はホーム・スペース・モードを示します。図 3-8 は、各種モードと、それぞれのモードでのアドレスの取り扱いを示しています。

PSW ビット			DAT	モード	アドレスの取り扱い	
5	16	17			命令 アドレス	論理 アドレス
0	0	0	オフ	実モード	実	実
0	0	1	オフ	実モード	実	実
0	1	0	オフ	実モード	実	実
0	1	1	オフ	実モード	実	実
1	0	0	オン	1 次スペース・ モード	1 次仮想	1 次仮想
1	0	1	オン	アクセス・ レジスタ・ モード	1 次仮想	AR 指定仮想
1	1	0	オン	2 次スペース・ モード	1 次仮想	2 次仮想
1	1	1	オン	ホーム・ スペース・ モード	ホーム仮想	ホーム仮想

図 3-8. 変換モード

制御レジスタ 0

制御レジスタ 0 には、動的アドレス変換を制御するためのビットが 1 つあります。このビットは以下のように割り当てられます。

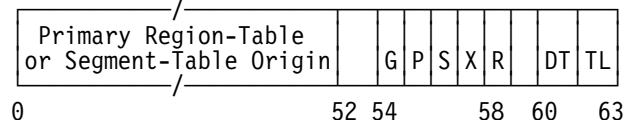


2 次スペース制御 (SS): 制御レジスタ 0 のビット 37 は、2 次スペース制御ビットです。このビットが 0 のときに、MOVE TO PRIMARY、MOVE TO SECONDARY、または SET ADDRESS SPACE CONTROL を実行しようとする、特殊操作例外が認識されます。このビットが 1 である場合は、CPU が 1 次スペース・モードになっているときに、2 次アドレス・スペース制御エレメントに指定されている領域テーブルまたはセグメント・テーブルが連結されていることを示します。

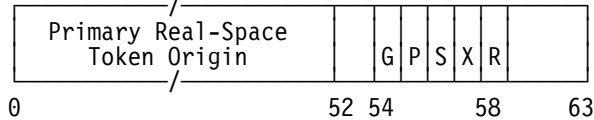
制御レジスタ 1

制御レジスタ 1 には、1 次アドレス・スペース制御エレメント (PASCE) が含まれています。このレジスタは、レジスタ内の実スペース制御ビット (R) に応じて、次の 2 つのフォーマットのいずれかになります。

1 次領域テーブルまたは
セグメント・テーブル指定 (R=0)



1 次実スペース指定 (R=1)



1 次アドレス・スペース制御エレメント内のフィールドは、以下のように割り振られます。

1 次領域テーブルまたは 2 次セグメント・テーブルの起点: 制御レジスター 1 の中の 1 次領域テーブル指定またはセグメント・テーブル指定のビット 0-51 と、その右側に付加された 12 個の 0 は、1 次領域テーブルまたはセグメント・テーブルの始めを示す 64 ビット・アドレスを形成します。これが実アドレスか絶対アドレスかは予測不能です。このテーブルは、1 次アドレス・スペース内の仮想アドレスを変換するために使用されるものなので、1 次領域テーブルまたはセグメント・テーブルと呼ばれます。

1 次サブスペース・グループ制御 (G): 制御レジスター 1 のビット 54 が 1 のときは、PASCE に指定されているアドレス・スペースが、基本スペースであるか、サブスペース・グループのサブスペースの 1 つであることを示します。ビット 54 が 0 のときは、アドレス・スペースはサブスペース・グループには属していません。

1 次専用スペース制御 (P): 制御レジスター 1 のビット 55 が 1 の場合は、次の事項が適用されます。(1) セグメント・テーブル・エントリーの変換索引バッファ (TLB) 表現内の共通セグメントの値が 1 であるときは、制御レジスター 1 の中のテーブルまたはトークンの起点と TLB エントリー内のテーブル起点が一致していても、そのエントリーおよびそのエントリーが指示する TLB ページ・テーブル・コピーは、1 次アドレス・スペースへの参照を変換するときに使用されません。(2) 低アドレス保護および取り出し保護オーバーライドは、1 次アドレス・スペースには適用されません。また、(3) 1 次アドレス・スペースへの参照が、記憶機構内のセグメント・テーブル・エントリーを使用して変換される場合に、そのエントリー内で共通セグメント・ビットが 1 のときは、変換指定例外が認識されます。上記の (2) は、制御レジスター 1 の内容が実スペース指定である場合も適用されます。

プログラミング上の注意: 上記の項目 (1) については、制御レジスター 1 の内容が実スペース指定であり、セグメント・テーブル・エントリーの TLB 表現内の共通セグメント・ビットの値が 1 であれば、その実スペース指定の中の専用スペース制御の値に関係なく、そのエ

ントリーおよびそれが指示する TLB ページ・テーブル・コピーは使用されません。

1 次記憶機構変更イベント制御 (S): 制御レジスター 9 の記憶機構変更スペース制御が 1 のときに、制御レジスター 1 のビット 56 が 1 の場合は、1 次アドレス・スペースに対して記憶機構変更イベントが生じることを指示します。ビット 56 が検査されるのは、PASCE を使用して、記憶機構オペランド格納参照のための動的アドレス変換を行うときです。記憶機構変更スペース制御が 0 のときは、ビット 56 は無視されます。

1 次スペース切り替えイベント制御 (X): 制御レジスター 1 のビット 57 が 1 の場合:

- スペース切り替え形式の PROGRAM CALL (PC-ss)、PROGRAM RETURN (PR-ss)、または PROGRAM TRANSFER (PT-ss) がの実行が完了すると、スペース切り替えイベント・プログラム割り込みが生じます。この割り込みが起こるのは、操作の前または後にビット 57 が 1 である場合です。
- 命令が取り出されるアドレス・スペースを、ホーム・アドレス・スペースへ、またはホーム・アドレス・スペースから変更する、RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令が完了すると、スペース切り替えイベント・プログラム割り込みが起こります。これは、操作の前後の両方ではなく、操作の前または後のいずれかにホーム・アドレスから命令が取り出されたときです。
- LOAD ADDRESS SPACE PARAMETERS により、条件コード 3 が設定されます。

1 次実スペース制御 (R): 制御レジスター 1 のビット 58 が 0 の場合は、レジスターには領域テーブル指定またはセグメント・テーブル指定が含まれています。ビット 58 が 1 の場合は、レジスターには実スペース指定が含まれています。ビット 58 が 1 のときは、セグメント・テーブル・エントリー内の共通セグメント・ビットの値が 1 であれば、制御レジスター 1 の中のトークン起点と TLB エントリー内のテーブル起点が一致していても、そのエントリーとそれが指示する TLB ページ・テーブル・コピーは、1 次アドレス・スペースに対する参照を変換するときに使用されません。

1 次指定タイプ制御 (DT): R が 0 の場合は、制御レジスター 1 の中では、次のように、ビット 60 および 61 によりテーブル指定のタイプが指定されます。

ビット 60

および 61 指定タイプ

11	領域第 1 テーブル
10	領域第 2 テーブル
01	領域第 3 テーブル
00	セグメント・テーブル

R が 0 の場合に、PASCE を使用して、左端の 1 ビットがアドレスのビット位置 0-10 のいずれかにある仮想アドレスを変換しようとするときは、ビット 60 および 61 は 2 進数 11 でなければなりません。同様に、左端の 1 ビットがアドレスのビット位置 11-21 のいずれかにあるときは、ビット 60 および 61 は 2 進数 11 または 10 でなければならず、左端の 1 ビットがアドレスのビット位置 22-32 のいずれかにあるときは、ビット 60 および 61 は 2 進数 11、10、または 01 でなければなりません。この条件が満たされていないと、ASCE タイプ例外が認識されます。

1 次領域テーブルまたはセグメント・テーブルの長さ

(TL): 制御レジスタ 1 中の 1 次領域テーブル指定またはセグメント・テーブル指定のビット 62 および 63 は、1 次領域テーブルまたはセグメント・テーブルの長さを 4096 バイト単位で指定します。したがって、領域テーブルまたはセグメント・テーブルの長さは可変であり、512 エントリーの倍数になります。1 次領域テーブルまたはセグメント・テーブルの長さ (4096 バイト単位) は、TL の値に 1 を加えた値になります。長さフィールドの内容は、テーブルを使用して変換される仮想アドレスの部分 ((RFX, RSX, RTX, または SX) が、そのテーブルに含まれるエントリーを指示しているかどうかを確認するために使用されます。

1 次実スペース・トークン起点: 制御レジスタ 1 中の 1 次実スペース指定のビット 0-51 と、その右に付加された 12 個の 0 は、64 ビット・アドレスを形成します。このアドレスを使用して、1 次アドレス・スペースに対する参照の仮想/実記憶域間等価変換を行うための TLB エントリーが形成され、使用されます。このアドレスは単なるトークンとして使用されるだけで、記憶域参照のためには使用されませんが、それでも、これは有効なアドレスでなければなりません。さもないと、制御レジスタ 1 の内容の使用時に誤った TLB エントリーが使用されるおそれがあります。

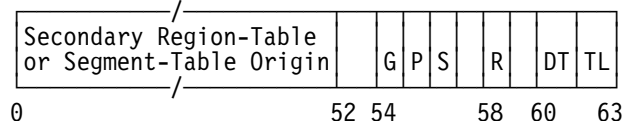
制御レジスタ 1 中には、割り当てられずに無視されるビットがいくつかあります。それは、このレジスタに領域テーブル指定またはセグメント・テーブル指定が含まれている場合のビット 52、53、および 59 と、レ

ジスタに実スペース指定が含まれている場合のビット 52、53、および 59-63 です。

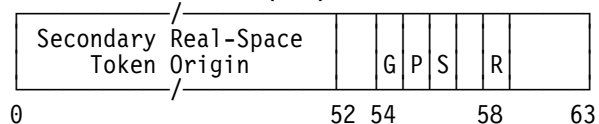
制御レジスタ 7

制御レジスタ 7 には、2 次アドレス・スペース制御エレメント (SASCE) が含まれています。このレジスタは、レジスタ内の実スペース制御ビット (R) に応じて、次の 2 つのフォーマットのいずれかになります。

2 次領域テーブルまたはセグメント・テーブル指定 (R=0)



2 次実スペース指定 (R=1)



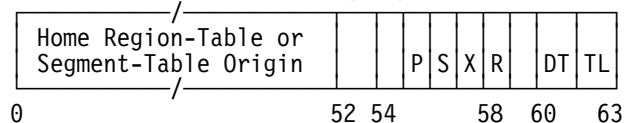
制御レジスタ 7 中の 2 次領域テーブル起点、2 次セグメント・テーブル起点、2 次サブスペース・グループ制御 (G)、2 次専用スペース制御 (P)、2 次記憶機構変更イベント制御 (S)、2 次実スペース制御 (R)、2 次指定タイプ制御 (DT)、2 次領域テーブルまたはセグメント・テーブルの長さ (TL)、および 2 次実スペース・トークン起点は、制御レジスタ 1 中の同じビット位置のフィールドと同じですが、制御レジスタ 7 は 2 次アドレス・スペースに適用されるという点が異なります。

制御レジスタ 7 中には、割り当てられずに無視されるビットがいくつかあります。それは、このレジスタに領域テーブル指定またはセグメント・テーブル指定が含まれている場合のビット 52、53、57、および 59 と、レジスタに実スペース指定が含まれている場合のビット 52、53、57、および 59-63 です。

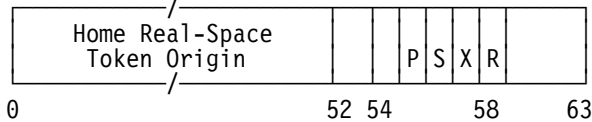
制御レジスタ 13

制御レジスタ 13 には、ホーム・アドレス・スペース制御エレメント (HASCE) が含まれています。このレジスタは、レジスタ内の実スペース制御ビット (R) に応じて、次の 2 つのフォーマットのいずれかになります。

ホーム領域テーブルまたはセグメント・テーブル指定 (R=0)



ホーム実スペース指定 (R=1)



ホーム・スペース切り替えイベント制御 (X): レジスタ 13 のビット 57 が 1 のときは、命令が取り出されるアドレス・スペースを、ホーム・アドレス・スペースへ、またはホーム・アドレス・スペースから変更する、RESUME PROGRAM, SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令が完了すると、スペース切り替えイベント・プログラム割り込みが起こります。これは、操作の前後の両方ではなく、操作の前または後のいずれかにホーム・アドレスから命令が取り出されたときです。

制御レジスタ 13 の中のホーム領域テーブル起点、ホーム・セグメント・テーブル起点、ホーム専用スペース制御 (P)、ホーム記憶機構変更イベント制御 (S)、ホーム実スペース制御 (R)、ホーム指定タイプ制御 (DT)、ホーム領域テーブルまたはセグメント・テーブルの長さ (TL)、およびホーム実スペース・トークン起点は、制御レジスタ 1 の中の同じビット位置にあるフィールドと同じに定義されていますが、制御レジスタ 13 はホーム・アドレス・スペースに適用されるという点が異なります。

制御レジスタ 13 には、割り当てられずに無視されるビットがいくつかあります。それは、このレジスタに領域テーブル指定またはセグメント・テーブル指定が含まれている場合のビット 52-54 および 59 と、レジスタに実スペース指定が含まれている場合のビット 52-54 および 59-63 です。

プログラミング上の注意:

1. 制御レジスタがロードされる時点では、動的アドレス変換に関する情報も含めて、レジスタにロードされる情報の妥当性は検査されません。この情報は、情報が使用される時点で検査され、そのときにプログラム例外があれば示されます。
2. 動的アドレス変換に関する情報は、DAT を伴う命令が実行される時、または、INVALIDATE PAGE TABLE ENTRY、LOAD REAL ADDRESS、または STORE REAL ADDRESS が実行される時に、使用されるものと見なされます。PSW で変換が指定されていても、命令の実行前に入出力割り込み、外部割り込み、再始動割り込み、またはマシン・チェック割り込みが生じたとき、あるいは PSW が待ち状態を指定している場合

は、この情報が使用されるものとは見なされません。

変換テーブル

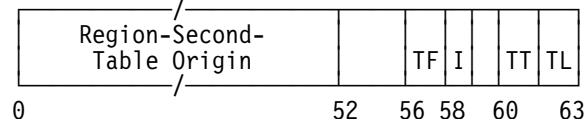
変換に使用されるアドレス・スペース制御エレメント (ASCE) が領域第 1 テーブル指定である場合、変換プロセスでは 5 つのテーブルを使用して 5 レベルの検索が行われます。5 つのテーブルとは、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、およびページ・テーブルです。これらのテーブルは、実記憶域または絶対記憶域の中に置かれています。ASCE が、領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定である場合は、指定のレベルより上位のレベルのテーブル内での検索は省略され、それら上位のテーブル自体も省略されます。

領域テーブル・エントリ

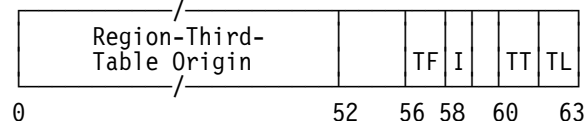
「領域テーブル・エントリ」という用語は、領域第 1 テーブル・エントリ、領域第 2 テーブル・エントリ、または領域第 3 テーブル・エントリを意味します。

領域第 1 テーブル、領域第 2 テーブル、および領域第 3 テーブルから取り出されたエントリのフォーマットは、以下のとおりです。エントリを含むテーブルのレベル (第 1、第 2、第 3) は、エントリ内のテーブル・タイプ (TT) ビットにより識別されます。

領域第 1 テーブル・エントリ (TT=11)



領域第 2 テーブル・エントリ (TT=10)



領域第 3 テーブル・エントリ (TT=01)



3 レベルの領域テーブル・エントリ内のフィールドは、次のように割り振られます。

領域第 2 テーブル起点、領域第 3 テーブル起点、およびセグメント・テーブル起点: 領域第 1 テーブル・エントリには、領域第 2 テーブル起点が含まれています。領域第 2 テーブル・エントリには、領域第 3 テ

ープル起点が含まれています。領域第 3 テーブル・エントリーには、セグメント・テーブル起点が含まれていません。以下の説明は、3 つの起点のそれぞれに適用されます。エントリーのビット 0-51 の右側に 12 個の 0 を付加したものが、次にレベルの低いテーブルの始めを示す 64 ビット・アドレスを形成します。これが実アドレスか絶対アドレスかは予測不能です。

領域第 2 テーブルのオフセット、領域第 3 テーブルのオフセット、およびセグメント・テーブルのオフセット (TF): 領域第 1 テーブル・エントリーには、領域第 2 テーブルのオフセットが含まれています。領域第 2 テーブル・エントリーには、領域第 3 テーブルのオフセットが含まれています。領域第 3 テーブル・エントリーには、セグメント・テーブルのオフセットが含まれていません。以下の説明は、3 つのオフセットのそれぞれに適用されます。このエントリーのビット 56 および 57 は、次に低いレベルのテーブルの始めの欠落している部分の長さを指定します。つまり、これらのビットは、次に低いレベルのテーブル内に実際に存在する最初のエントリーの位置を示します。これらのビットは、欠落部分の長さを 4096 バイト単位で指定するものであり、したがって、欠落部分は可変長で、長さは 512 エントリーの倍数になります。欠落部分の長さ (4096 バイト単位) は、TF の値と同じになります。オフセット・フィールドの内容は、長さフィールド (ビット 62 および 63) と一緒に使用されて、次に低いレベルのテーブルを使用して変換される仮想アドレスの部分 (RSX, RTX, または SX) が、テーブル内に実際に存在するエントリーを指示しているかどうかを確認されます。

領域無効ビット (I): 領域第 1 テーブル・エントリーまたは領域第 2 テーブル・エントリーのビット 58 は、そのエントリーに関連した領域のセットが使用可能かどうかを制御します。領域第 3 テーブル・エントリーのビット 58 は、そのエントリーに関連した単一の領域が使用可能かどうかを制御します。ビット 58 が 0 のときは、領域テーブル・エントリーを使用してアドレス変換が進められます。このビットが 1 のときは、エントリーは変換に使用できません。

テーブル・タイプ・ビット (TT): 領域第 1 テーブル・エントリー、領域第 2 テーブル・エントリー、および領域第 3 テーブル・エントリーのビット 60 および 61 は、以下のように、そのエントリーを含むテーブルのレベルを識別します。

ビット 60 および 61 領域テーブルのレベル

11	第 1
10	第 2
01	第 3

ビット 60 および 61 には、変換に使用しているアドレス・スペース制御要素であるテーブル指定のタイプ、およびこれまでに使用したテーブル・レベルの数を考慮に入れながら、正しいテーブル・レベルを指定する必要があります。指定に誤りがあると、変換指定例外が認識されます。

領域第 2 テーブルの長さ、領域第 3 テーブルの長さ、およびセグメント・テーブルの長さ (TL): 領域第 1 テーブル・エントリーには、領域第 2 テーブルの長さが含まれています。領域第 2 テーブル・エントリーには、領域第 3 テーブルの長さが含まれています。領域第 3 テーブル・エントリーには、セグメント・テーブルの長さが含まれています。以下の説明は、3 つの長さのそれぞれに適用されます。エントリーのビット 62 および 63 は、次に低いレベルのテーブルの長さを 4096 バイト単位で指定します。したがって、そのテーブルの長さは可変であり、512 エントリーの倍数になります。次に低いレベルのテーブルの長さ (4096 バイト単位) は、TL の値に 1 を加えた値になります。長さフィールドの内容は、オフセット・フィールド (ビット 56 および 57) と一緒に使用されて、次に低いレベルのテーブルを使用して変換される仮想アドレスの部分 (RSX, RTX, または SX) が、テーブル内に実際に存在しているエントリーを指示しているかどうかを確認されます。

セグメント・テーブル・エントリー

セグメント・テーブルから取り出されるエントリーのフォーマットは、以下のとおりです。

セグメント・テーブル・エントリー (TT=00)



セグメント・テーブル・エントリー内では、次のようにフィールドが割り振られます。

ページ・テーブル起点: ビット 0-52 の右側に 11 個の 0 を付加したものが、ページ・テーブルの始めを示す 64 ビット・アドレスを形成します。これが実アドレスか絶対アドレスかは予測不能です。

ページ保護ビット (P): ビット 54 は、このセグメント・テーブル・エントリーが指示するページ・テーブル内の各エントリーのページ保護ビットと OR 関係にあるものとして扱われます。したがって、このビットが 1 であれば、セグメント・テーブル・エントリーが指定するセグメント全体にページ保護が適用されます。

セグメント無効ビット (I): ビット 58 は、セグメント・テーブル・エントリーに関連したセグメントが使用可能かどうかを制御します。このビットが 0 であれば、セグメント・テーブル・エントリーを使用してアドレス変換が進められます。このビットが 1 のときは、セグメント・テーブル・エントリーは変換に使用できません。

共通セグメント・ビット (C): ビット 59 は、セグメント・テーブル・エントリーおよびそれが指示するページ・テーブルの変換索引バッファ (TLB) のコピーを使用するかどうかを制御します。0 は専用セグメントを示します。この場合、セグメント・テーブル・エントリーとそれが指示するページ・テーブルは、このセグメント・テーブル・エントリーが入っているセグメント・テーブルを指すセグメント・テーブル起点との関連付けの中でのみ、使用できます。1 は共通セグメントを示します。その場合は、セグメント・テーブル・エントリーおよびそれが指示するページ・テーブルは、異なるセグメント・テーブルが指定されていても、セグメント指標に対応するアドレスの変換に引き続き使用できます。ただし、変換に使用されるアドレス・スペース制御エレメント内で専用スペース制御 (ビット 55) が 1 である場合、またはそのアドレス・スペース制御エレメントが実スペースを指示している場合は、共通セグメント用のセグメント・テーブル・エントリーおよびページ・テーブルの TLB コピーを使用することはできません。使用しているアドレス・スペース制御エレメント内で専用スペース制御が 1 のときに、変換時に記憶機構からセグメント・テーブル・エントリーが取り出される場合は、共通セグメント・ビットは 0 でなければなりません。さもないと、変換指定例外が認識されます。

テーブル・タイプ・ビット (TT): セグメント・テーブル・エントリーのビット 60 および 61 は、そのエントリーが入っているテーブルのレベルを示す 2 進数 00 です。領域テーブル・エントリーまたはセグメント・テーブル・エントリーのビット 60 および 61 のすべての可能な値とそれぞれの意味は、以下のとおりです。

**ビット 60
および 61 テーブル・レベル**

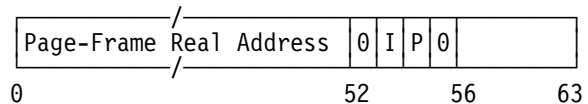
11	領域第 1
10	領域第 2
01	領域第 3
00	セグメント

ビット 60 および 61 には、変換に使用しているアドレス・スペース制御エレメントであるテーブル指定のタイプ、およびこれまでに使用したテーブル・レベルの数を考慮に入れながら、正しいテーブル・レベルを指定する必要があります。指定に誤りがあると、変換指定例外が認識されます。

セグメント・テーブル・エントリーのビット 53、55-57、62、および 63 は、将来の拡張に備えて予約されています。

ページ・テーブル・エントリー

ページ・テーブル・エントリーから取り出されるエントリーのフォーマットは、以下のとおりです。



ページ・テーブル・エントリー内のフィールドは、以下のように割り振られます。

ページ・フレーム実アドレス (PFRA): ビット 0-51 は、実記憶域アドレスの左端のビットを表します。これらのビットの右側に、仮想アドレスの 12 ビットのバイト指標フィールドを連結したものが、64 ビットの実アドレスとなります。

ページ無効ビット (I): ビット 53 は、ページ・テーブル・エントリーに関連したページが使用可能かどうかを制御します。このビットが 0 であれば、ページ・テーブル・エントリーを使用してアドレス変換が進められます。このビットが 1 のときは、ページ・テーブル・エントリーは変換に使用できません。

ページ保護ビット (P): ビット 54 は、ページに対する格納アクセスができるかどうかを制御します。この保護メカニズムは、キー制御保護メカニズムと低アドレス保護メカニズムに追加して使用する機能です。このビットには、取り出しアクセスには影響を与えません。このビットが 0 のときは、変換に使用されているセグメント・テーブル・エントリー内のページ保護ビット、およびその他の保護メカニズムによる制限の範囲内で、ページへの格納が許されます。このビットが 1 のときは、格納は許可されません。ページ保護ビットが 1 のときに格納を

行おうとした場合は、記憶保護例外が認識されます。ページに対してページ保護を適用するかどうかを決定する際に、セグメント・テーブル・エントリー内のページ保護ビットはビット 54 と OR 関係にあるものとして扱われます。

エントリーのビット位置 52 および 55 は 0 でなければなりません。さもないと、このエントリーを使用してアドレス変換を行う命令を実行するときに、変換指定例外が認識されます。ビット位置 56-63 には何も割り当てられず、無視されます。

変換プロセス

このセクションでは、仮想アドレスを使用して主記憶機構にアクセスする前に暗黙的に行われる変換プロセスについて説明します。明示的変換、つまり、LOAD REAL ADDRESS、STORE REAL ADDRESS、および TEST PROTECTION でのオペランド・アドレス変換のプロセスもこれと同じですが、LOAD REAL ADDRESS および TEST PROTECTION では、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、およびページ変換例外は認識されないという点が異なります。これらの条件は、代わりに条件コードで示されます。さらに、LOAD REAL ADDRESS および STORE REAL ADDRESS のオペランド・アドレスの変換では、CPU は実モードになることがあるという点も異なります。

仮想アドレスの変換は、PSW 中の DAT モード・ビットおよびアドレス・スペース制御ビットと、制御レジスター 1、7、および 13 中のアドレス・スペース制御エレメント (ASCE) により制御されるほか、アクセス・レジスターに指定されている内容によっても影響を受けます。変換に使用される ASCE が領域第 1 テーブル指定である場合は、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、およびページ・テーブルを使用して、変換が行われます。これらのテーブルは、すべて実記憶域または絶対記憶域に入っています。ASCE が、それより低レベルのタイプのテーブル指定 (領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定) である場合は、その指定されたレベルおよびそれ以下のテーブル・レベルのみを使用して変換が行われます。そして、指定レベルより高いテーブル・レベルを使用する必要がある仮想アドレス・ビット (値が 0 以外の場合) は、すべて 0 でなければなりません。さもないと、ASCE タイプ例外が認識されます。ASCE が実スペース指定である場合は、仮想アドレスは実アドレスと見なされ、実記憶域または絶対記憶域内のテーブル・エントリーは使用されません。

特定のアドレス変換に使用されるアドレス・スペース制御エレメント (ASCE) を、有効 ASCE と言います。したがって、1 次仮想アドレスを変換するときは、制御レジスター 1 の内容が有効 ASCE として使用されます。同様に、2 次仮想アドレスの場合は、制御レジスター 7 の内容が使用されます。AR 指定の仮想アドレスの場合は、アクセス・レジスターに指定されている ASCE が使用されます。ホーム仮想アドレスの場合は、制御レジスター 13 の内容が使用されます。

有効 ASCE 内の実スペース制御が 0 の場合は、その ASCE 中の指定タイプ制御により、ASCE のテーブル指定タイプが決まります。このテーブル指定は、領域第 1 テーブル指定、領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定です。仮想アドレスの対応する部分 (領域第 1 指標、領域第 2 指標、領域第 3 指標、またはセグメント指標) が、この指定中のテーブル長フィールドに照らして検査され、その指標をこの指定中の起点に加算することにより、指定されたテーブル内のエントリーが選択されます。指定中のテーブル長フィールドに従って計算した結果、選択されたエントリーがテーブルの外部に位置することになる場合、または、選択されたエントリーの中で I ビットが 1 である場合は、テーブル指定により指示されているテーブル・レベルに応じて、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外が認識されます。選択されたエントリー内のテーブル・タイプ・ビットが予期したテーブル・レベルを示していない場合は、変換指定例外が認識されます。

有効 ASCE を使用して選択されたテーブル・エントリーは、使用する次に低いレベルのテーブルを指示します。現在のテーブルが領域第 1 テーブル、領域第 2 テーブル、または領域第 3 テーブルである場合は、仮想アドレスの次の部分 (それぞれ、領域第 2 指標、領域第 3 指標、またはセグメント指標) が、現行テーブル・エントリー内のテーブル・オフセット・フィールドおよびテーブル長フィールドに照らして検査され、その指標をこのエントリー内の起点に加算することにより、次に低いレベルのテーブル内のエントリーが選択されます。現行テーブル・エントリー内のテーブル・オフセット・フィールドおよびテーブル長フィールドに従って計算した結果、次のテーブル内の選択されたエントリーがそのテーブルの外部に位置することになる場合、または選択されたエントリーの中で I ビットが 1 である場合は、次のテーブルのレベルに応じて、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外が認識されます。選択されたエントリー内のテーブル・タイプ・ビットが予期したテーブル・レベルを示していない場合は、変換指定例外が認識されます。

連続したテーブル・レベルを使用した仮想アドレスの部分の処理は、セグメント・テーブル・エントリーが選択されるまで続けられます。このエントリーは、使用するページ・テーブルを指示します。セグメント・テーブル・エントリーには、指定されたセグメント内のすべてのページに適用されるページ保護ビットが含まれています。

仮想アドレスのページ指標部分を、セグメント・テーブル・エントリー内のページ・テーブル起点に加算することにより、ページ・テーブル内のエントリーが選択されます。ページ・テーブル・エントリー内の I ビットが 1 の場合は、ページ変換例外が認識されます。ページ・テーブル・エントリーには、仮想アドレスの変換を表す実アドレスの左端ビットが含まれ、また、そのページ・テーブル・エントリーで指定されたページのみ適用されるページ保護ビットが含まれています。

仮想アドレスのバイト指標フィールドは、変更されずに、実アドレスの右端ビット位置として使用されます。

実記憶域または絶対記憶域内の変換テーブルを参照することによって遅延が生じないようにするために、一般に、テーブルから取り出された情報は変換索引バッファ (TLB) と呼ばれる特殊バッファにも置かれます。したがって、同じテーブル・エントリーを使用する後続の変換は、TLB に記録されている情報を使用して実行できます。TLB には、実スペース指定に関連した仮想/実記憶域間等価変換も記録されます。TLB の働きについては、3-41 ページの『変換索引バッファ』で説明します。

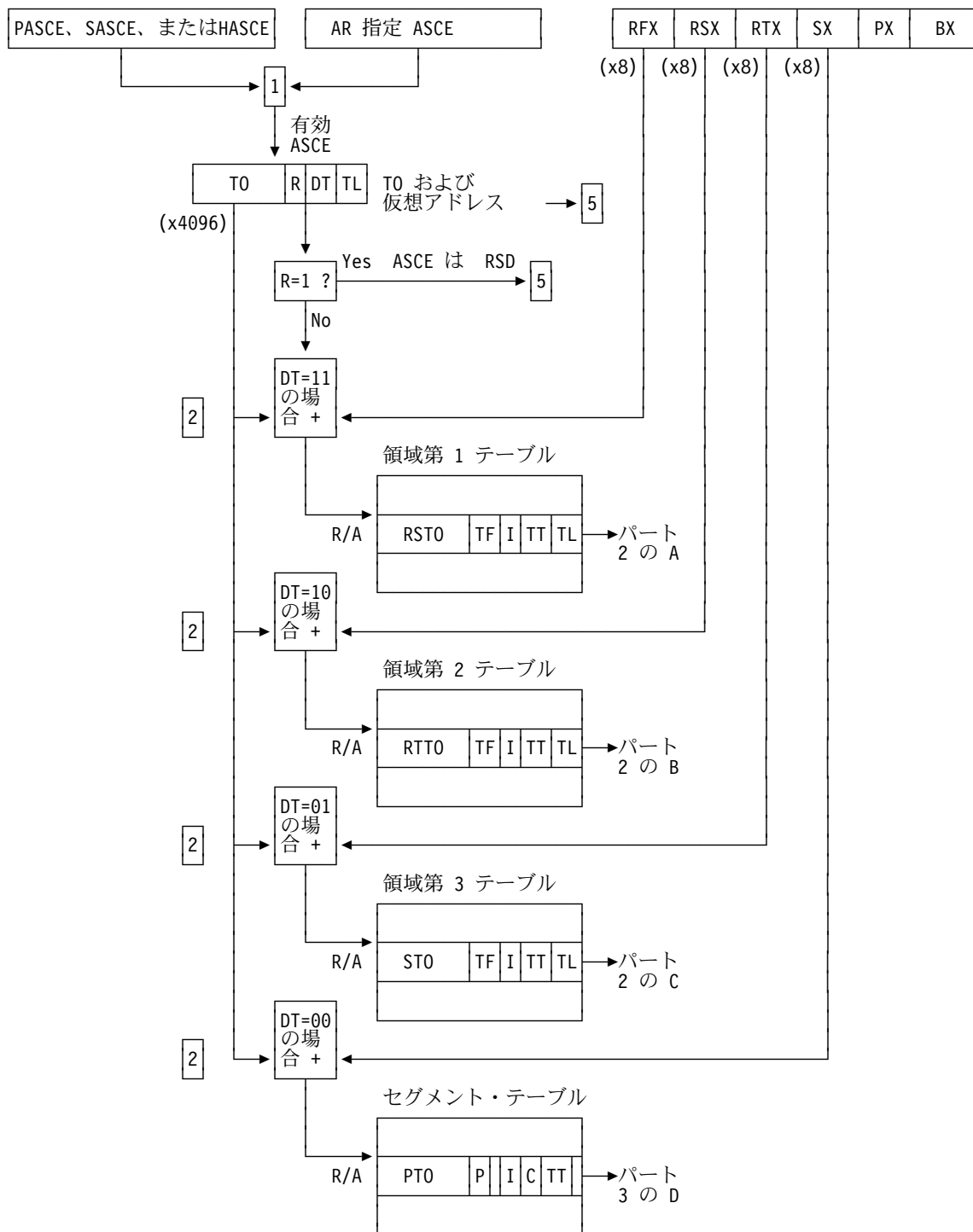
アドレス変換プロセスで、領域テーブル、セグメント・テーブル、またはページ・テーブルからエントリーを取り出すために実記憶域または絶対記憶域にアクセスするときは、キー制御保護は適用されません。

3-35 ページの図 3-9 は、TLB の効果も含めた変換プロセスを示しています。

制御レジスタ 1,
7, または 13

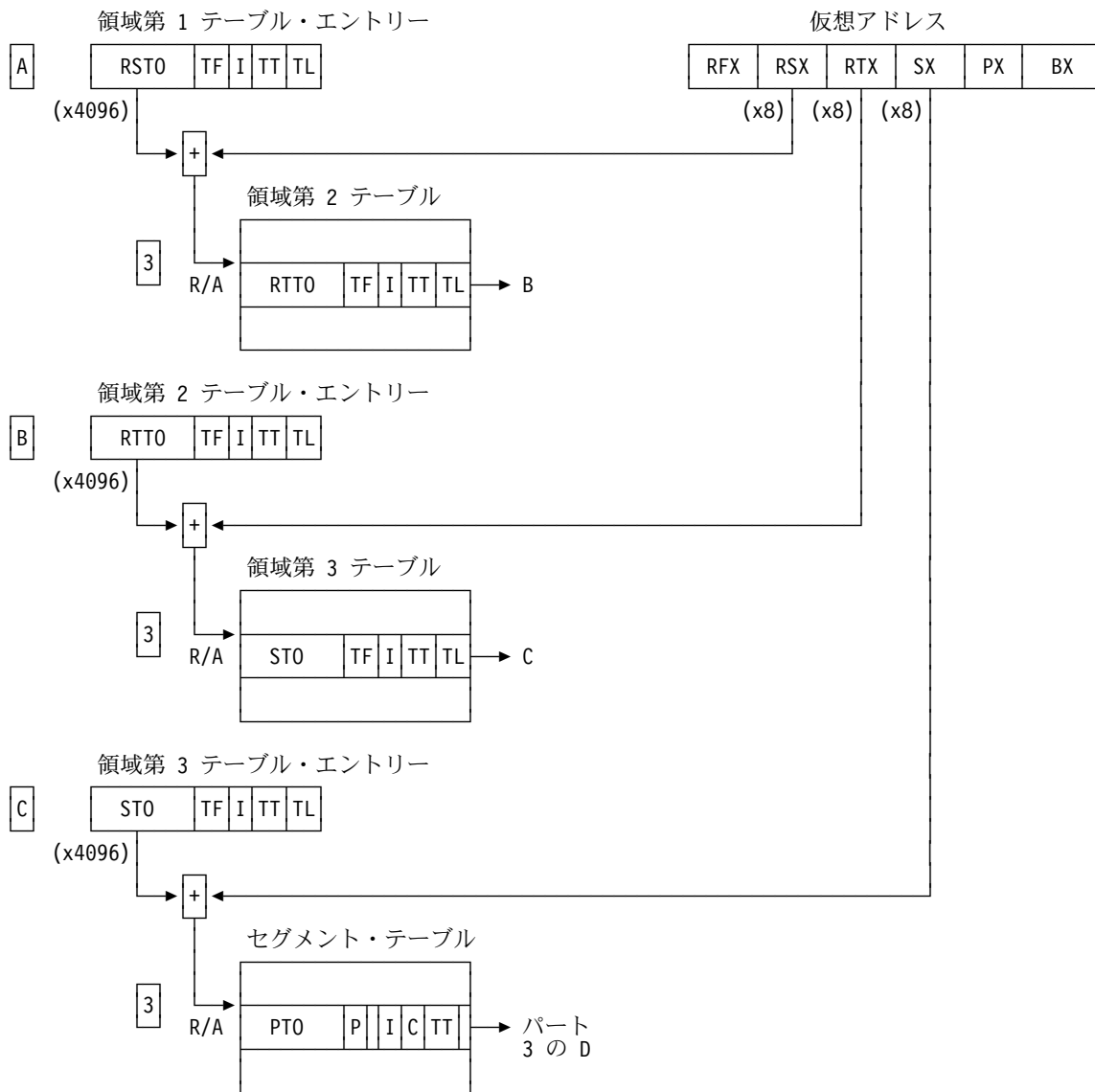
ASN 第 2 テーブル・エントリー

仮想アドレス



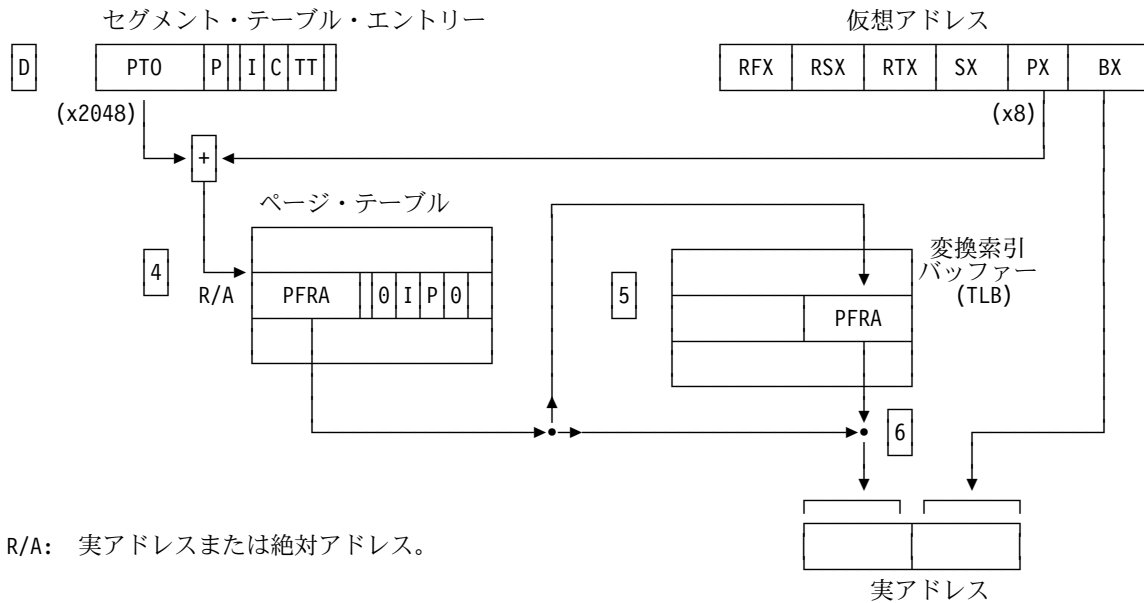
R/A: 実アドレスまたは絶対アドレス。

図 3-9 (1/3). 変換プロセス



R/A: 実アドレスまたは絶対アドレス。

図 3-9 (2/3). 変換プロセス



R/A: 実アドレスまたは絶対アドレス。

- 1 制御レジスタ 1 は、1 次仮想アドレスの変換のための 1 次アドレス・スペース制御エレメント (ASCE) を提供し、制御レジスタ 7 は、2 次仮想アドレスの変換のための 2 次 ASCE を提供し、制御レジスタ 13 は、ホーム仮想アドレスの変換のためのホーム ASCE を提供します。ASN 第 2 テーブル・エントリーは、AR 指定仮想アドレスの変換のための AR 指定 (アクセス・レジスタ指定) ASCE を提供します。
- 2 仮想アドレスの中の、DT により選択された指標の左側の部分は、0 でなければなりません。さもないと、ASCE タイプ例外が認識されます。指標のビット 0 および 1 は、ASCE の中の TL に等しいかまたはそれより小さくなくてはならず、選択されたテーブル・エントリー内の I は 0 でなければなりません。さもないと、DT により選択されたテーブル・レベルに応じて、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外が認識されます。選択されたテーブル・エントリー内の TT は、DT に等しくなくてはなりません。さもないと、変換指定例外が認識されます。
- 3 次の指標のビット 0 および 1 は、現行テーブル・エントリー内の TF に等しいかまたはそれより大きくなければならず、TL に等しいかまたはそれより小さくなくてはなりません。また、次に選択されるテーブル・エントリーの中の I は 0 でなければなりません。さもないと、次に選択されるエントリーのテーブル・レベルに応じて、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外が認識されます。次に選択されるエントリーの中の TT は、現行エントリーの中の TT から 1 を減算した値でなければなりません。さもないと、変換指定例外が認識されます。
- 4 ページ・テーブル・エントリーの中の I は 0 でなければなりません。さもないと、ページ変換例外が認識されます。ページ・テーブル・エントリーのビット 52 および 55 は 0 でなければなりません。さもないと、変換指定例外が認識されます。
- 5 TLB を検索するには、仮想アドレスの部分、および有効 ASCE の中のテーブル起点または実スペース・トークン起点などの情報が使用されます。
- 6 一致があれば、TLB からのページ・フレーム実アドレスを使用して実アドレスが形成されます。一致がなく、有効 ASCE がテーブル指定である場合は、実記憶域または絶対記憶域の中のテーブル・エントリーが取り出されます。そして、取り出されたエントリーを使用してアドレスが変換され、そのアドレスと検索情報を使用して、TLB 中のエントリーが形成されます。有効 ASCE が実スペース指定である場合は、仮想アドレスを等価の実アドレスに変換する TLB エントリーが形成されます。

図 3-9 (3/3). 変換プロセス

実スペース制御の検査

有効アドレス・スペース制御エレメント (ASCE) 内で、実スペース制御 (ビット 58) の値が 0 の場合は、ASCE は領域テーブル指定またはセグメント・テーブル指定です。実スペース制御が 1 の場合は、ASCE は実スペース指定です。

指定タイプ制御の検査

実スペース制御が 0 のときは、指定タイプ制御、つまり有効アドレス・スペース制御エレメント (ASCE) のビット 60 および 61 が、ASCE のテーブル指定タイプを示します。タイプによっては、変換される仮想アドレスの幾つかの左端ビットが 0 でなければなりません。さもないと、ASCE タイプ例外が認識されます。ビット 60 および 61 の可能なそれぞれの値について、0 でなければならぬテーブル指定タイプ・ビットおよび仮想アドレス・ビットは、以下のとおりです。

ビット 60 および	指定タイプ	0 でなければならぬ 仮想アドレス・ビット
11	領域第 1 テーブル	なし
10	領域第 2 テーブル	0-10
01	領域第 3 テーブル	0-21
00	セグメント・テーブル	0-32

アドレス・スペース制御エレメントにより指定されるテーブル内での検索

指定タイプ制御、つまり有効アドレス・スペース制御エレメント (ASCE) のビット 60 および 61 は、ASCE のテーブル指定タイプと、指定されたテーブルを使用して変換される仮想アドレスの部分を、次のように示します。

ビット 60 および	指定タイプ	テーブルにより 変換される 仮想アドレスの部分
11	領域第 1 テーブル	領域第 1 指標 (ビット 0-10)
10	領域第 2 テーブル	領域第 2 指標 (ビット 11-21)
01	領域第 3 テーブル	領域第 3 指標 (ビット 22-32)
00	セグメント・テーブル	セグメント指標 (ビット 33-43)

ビット 60 および 61 の値が 2 進数 11 であるときは、仮想アドレスの領域第 1 指標部分と、ASCE に含まれている領域第 1 テーブル起点を使用して、領域第 1 テーブルからエントリーが選択されます。

実記憶域または絶対記憶域内の領域第 1 テーブル・エントリーの 64 ビット・アドレスは、領域第 1 テーブル指定のビット 0-51 の右側に 12 個の 0 を付加し、さらに、右側に 3 個、左側に 50 個の 0 を付加した領域第 1 指標を加算したものです。この加算によりビット位置 0 の繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{64} - 1$ から 0 に循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、アドレスの 64 ビットがすべて使用されます。

領域第 1 テーブル検索プロセスの一環として、仮想アドレスのビット 0 および 1 (つまり領域第 1 指標のビット 0 および 1) を、テーブル長 (領域第 1 テーブル指定のビット 62 および 63) と比較することにより、アドレッシングされたエントリーが領域第 1 テーブル内にあるかどうかを確認されます。テーブル長フィールドの値が、仮想アドレスの対応するビット位置の値より小さい場合は、領域第 1 変換例外が認識されます。変換索引バッファ内に領域第 1 テーブル・エントリーと同等のものがあり、それを使用して変換が行われる場合は、テーブル長との比較は省略できます。

他の CPU からは、領域第 1 テーブル・エントリーの 8 バイトすべてが同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。領域第 1 テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作単位は抑止されます。

領域第 1 テーブルから取り出されたビット 58 は、対応する領域のセットが使用可能かどうかを示します。このビットが検査され、その値が 1 であれば、領域第 1 変換例外が認識されます。

領域第 1 テーブル・エントリー内のテーブル・タイプ・ビット (ビット 60 および 61) の値が、ASCE のビット 60 および 61 と同じでない場合は、変換指定例外が認識されます。

領域第 1 テーブル検索のプロセスで例外が認識されなかった場合は、領域第 1 テーブルから取り出されたエント

リーは、対応する領域第 2 テーブルの始めと、オフセットおよび長さを示しています。

ASCE のビット 60 および 61 の値が 2 進数 10 である場合は、仮想アドレスの領域第 2 指標部分と、ASCE に含まれている領域第 2 テーブル起点を使用して、領域第 2 テーブルからエンタリーが選択されます。仮想アドレスのビット 11 および 12 (つまり領域第 2 指標のビット 0 および 1) は、ASCE に含まれているテーブル長と比較されます。テーブル長フィールドの値が、仮想アドレスの対応するビット位置の値より小さい場合は、領域第 2 変換例外が認識されます。変換索引バッファー内に領域第 2 テーブル・エンタリーと同等のものがあ、それを使用して変換が行われる場合は、テーブル長との比較は省略できます。領域第 2 テーブル検索プロセスは、領域第 2 テーブル・エンタリー内のビット 58 が 1 の場合に領域第 2 変換例外が認識されるという点以外は、領域第 1 テーブル検索プロセスと同じです。例外が認識されなかった場合は、領域第 2 テーブルから取り出されたエンタリーは、対応する領域第 3 テーブルの始めと、オフセットおよび長さを示しています。

ASCE のビット 60 および 61 の値が 2 進数 01 である場合は、仮想アドレスの領域第 3 指標部分と、ASCE に含まれている領域第 3 テーブル起点を使用して、領域第 3 テーブルからエンタリーが選択されます。仮想アドレスのビット 22 および 23 (つまり領域第 3 指標のビット 0 および 1) は、ASCE に含まれているテーブル長と比較されます。テーブル長フィールドの値が、仮想アドレスの対応するビット位置の値より小さい場合は、領域第 3 変換例外が認識されます。変換索引バッファー内に領域第 3 テーブル・エンタリーと同等のものがあ、それを使用して変換が行われる場合は、テーブル長との比較は省略できます。領域第 3 テーブル検索プロセスは、領域第 3 テーブル・エンタリー内のビット 58 が 1 の場合に領域第 3 変換例外が認識されるという点以外は、領域第 3 テーブル・エンタリー内のテーブル・タイプ・ビットの検査も含めて、領域第 1 テーブル検索プロセスと同じです。例外が認識されなかった場合は、領域第 3 テーブルから取り出されたエンタリーは、対応するセグメント・テーブルの始めと、オフセットおよび長さを示しています。

ASCE のビット 60 および 61 の値が 2 進数 00 である場合は、仮想アドレスのセグメント指標部分と、ASCE に含まれているセグメント・テーブル起点を使用して、セグメント・テーブルからエンタリーが選択されます。仮想アドレスのビット 33 および 34 (つまりセグメント指標のビット 0 および 1) は、ASCE に含まれているテーブル長と比較されます。テーブル長フィールドの値が、仮想アドレスの対応するビット位置の値より小さい

場合は、セグメント変換例外が認識されます。変換索引バッファー内にセグメント・テーブル・エンタリーと同等のものがあ、それを使用して変換が行われる場合は、テーブル長との比較は省略できます。セグメント・テーブル・エンタリー内のビット 58 が 1 の場合は、セグメント変換例外が認識されます。また、(1) ASCE 中の専用スペース制御 (ビット 55) が 1 の場合、および、(2) セグメント・テーブルから取り出されたエンタリー内の共通セグメント・ビット (ビット 59) が 1 の場合は、変換指定例外が認識されます。その他の点では、セグメント・テーブル検索プロセスは、セグメント・テーブル・エンタリー内のテーブル・タイプ・ビットの検査も含めて、領域第 1 テーブル検索プロセスと同じです。例外が認識されなかった場合は、セグメント・テーブルから取り出されたエンタリーは、対応するページ・テーブルの始めを指し示します。

領域テーブル・エンタリーにより指定される テーブル内での検索

有効アドレス・スペース制御エレメント (ASCE) が領域テーブル指定である場合は、前のセクションの説明と同じ方法で領域テーブル・エンタリーが選択されます。そして、選択されたエンタリーと、仮想アドレスの次の指標部分を使用して、次に低いレベルのテーブル (別の領域テーブルまたはセグメント・テーブル) からエンタリーが選択されます。

ASCE を使用して選択されたテーブル・エンタリーが領域第 1 テーブル・エンタリーである場合は、仮想アドレスの領域第 2 指標部分と、その領域第 1 テーブル・エンタリーに含まれている領域第 2 テーブル起点を使用して、領域第 2 テーブルからエンタリーが選択されます。

実記憶域または絶対記憶域内の領域第 2 テーブル・エンタリーの 64 ビット・アドレスは、領域第 1 テーブル・エンタリーのビット 0-51 の右側に 12 個の 0 を付加し、さらに、右側に 3 個、左側に 50 個の 0 を付加した領域第 2 指標を加算したものです。この加算によりビット位置 0 の繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{64} - 1$ から 0 に循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、アドレスの 64 ビットがすべて使用されます。

領域第 2 テーブル検索プロセスの一環として、仮想アドレスのビット 11 および 12 (つまり領域第 2 指標のビット 0 および 1) を、テーブル・オフセット (領域第 1 テーブル・エンタリーのビット 56 および 57) と比較し、さらにテーブル長 (領域第 1 テーブル・エンタリーのビット 62 および 63) と比較することにより、アドレ

ッシングされたエントリーが領域第 2 テーブル内にあるかどうかを確認されます。テーブル・オフセット・フィールドの値が、仮想アドレスの対応するビット位置の値より大きい場合、または、テーブル長フィールドの値が、仮想アドレスの対応するビット位置の値より小さい場合は、領域第 2 変換例外が認識されます。

他の CPU からは、領域第 2 テーブル・エントリーの 8 バイトすべてが同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。領域第 2 テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作単位は抑止されます。

領域第 2 テーブルから取り出されたビット 58 は、対応する領域のセットが使用可能かどうかを示します。このビットが検査され、その値が 1 であれば、領域第 2 変換例外が認識されます。

領域第 2 テーブル・エントリー内のテーブル・タイプ・ビット 60 および 61 の値が、次に高いレベルのテーブル内の同じビットの値から 1 を引いた値でない場合は、変換指定例外が認識されます。

領域第 2 テーブル検索のプロセスで例外が認識されなかった場合は、領域第 2 テーブルから取り出されたエントリーは、対応する領域第 3 テーブルの始めと、オフセットおよび長さを示しています。

ASCE を使用して選択されたテーブル・エントリーが領域第 2 テーブル・エントリーである場合、または、領域第 1 テーブル・エントリーの内容を使用して領域第 2 テーブル・エントリーが選択された場合は、仮想アドレスの領域第 3 指標部分と、その領域第 2 テーブル・エントリーに含まれている領域第 3 テーブル起点を使用して、領域第 3 テーブルからエントリーが選択されます。仮想アドレスのビット 22 および 23 (つまり領域第 3 指標のビット 0 および 1) は、領域第 2 テーブル・エントリー内のテーブル・オフセットおよびテーブル長と比較されます。テーブル・オフセットがビット 22 および 23 より大きいか、テーブル長がビット 22 および 23 より小さいか、または領域第 3 テーブル・エントリー内のビット 58 が 1 である場合は、領域第 3 変換例外が認識されます。その他の点では、領域第 3 テーブル検索プロセスは、領域第 3 テーブル・エントリー内のテーブル・タイプ・ビットの検査も含めて、領域第 2 テーブル検索プロセスと同じです。例外が認識されなかった場合は、領域第 3 テーブルから取り出されたエントリーは、対応するセグメント・テーブルの始めと、オフセットおよび長さを示しています。

ASCE を使用して選択されたテーブル・エントリーが領域第 3 テーブル・エントリーである場合、または、領域第 2 テーブル・エントリーの内容を使用して領域第 3 テーブル・エントリーが選択された場合は、仮想アドレスのセグメント指標部分と、その領域第 3 テーブル・エントリーに含まれているセグメント・テーブル起点を使用して、セグメント・テーブルからエントリーが選択されます。仮想アドレスのビット 33 および 34 (つまりセグメント指標のビット 0 および 1) は、領域第 3 テーブル・エントリー内のテーブル・オフセットおよびテーブル長と比較されます。テーブル・オフセットがビット 33 および 34 より大きいか、テーブル長がビット 33 および 34 より小さいか、またはセグメント・テーブル・エントリー内のビット 58 が 1 である場合は、セグメント変換例外が認識されます。また、(1) ASCE の中の専用スペース制御 (ビット 55) が 1 の場合、および、(2) セグメント・テーブルから取り出されたエントリー内の共通セグメント・ビット (ビット 59) が 1 の場合は、変換指定例外が認識されます。その他の点では、セグメント・テーブル検索プロセスは、セグメント・テーブル・エントリー内のテーブル・タイプ・ビットの検査も含めて、領域第 2 テーブル検索プロセスと同じです。例外が認識されなかった場合は、セグメント・テーブルから取り出されたエントリーは、対応するページ・テーブルの始めを指し示します。

ページ・テーブルの検索

仮想アドレスのページ指標部分と、セグメント・テーブル・エントリーに含まれているページ・テーブル起点を使用して、ページ・テーブルからエントリーが選択されます。

実記憶域または絶対記憶域内のページ・テーブル・エントリーの 64 ビット・アドレスは、ページ・テーブル起点の右側に 11 個の 0 を付加し、さらに、右側に 3 個、左側に 53 個の 0 を付加したページ指標を加算したものです。ビット位置 0 の繰り上がりが生じることはありません。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、アドレスの 64 ビットがすべて使用されます。

他の CPU からは、ページ・テーブル・エントリーの 8 バイトすべてが同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。ページ・テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作単位は抑止されます。

ページ・テーブルから取り出されたエントリは、そのページが使用可能かどうかを示すものであり、ページ・フレーム実アドレスの左側のビット群が含まれています。ページ無効ビット (ビット 53) を検査することにより、対応するページが使用可能かどうかを確認されます。このビットが 1 の場合は、ページ変換例外が認識されます。ビット位置 52 または 55 に 1 が含まれている場合は、変換指定例外が認識されます。変換に使用するセグメント・テーブル・エントリ、またはページ・テーブル・エントリの中で、ページ保護ビット (ビット 54) が 1 であり、ここでやっている変換の目的となっている記憶機構参照が格納である場合は、記憶保護例外が認識されます。

実アドレスの形成

有効アドレス・スペース制御エレメント (ASCE) が、領域テーブル指定またはセグメント・テーブル指定であるときに、変換プロセスで何も例外が生じなかった場合は、ページ・テーブル・エントリからページ・フレーム実アドレスが得られます。ASCE が実スペース指定である場合は、仮想アドレスのビット 0-51 がページ・フレーム実アドレスとして使用されます。いずれの場合も、ページ・フレーム実アドレスと仮想アドレスのバイト指標部分が連結され、ページ・フレーム実アドレスが左側の部分となります。結果は、仮想アドレスに対応する実記憶域アドレスとなります。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、アドレスの 64 ビットがすべて使用されます。

変換中の例外の認識

無効なアドレスや無効なフォーマットがあると、変換プロセスの実行時に例外が認識されます。例外が認識されるのは、テーブル・エントリに含まれている情報が変換に使用されて、その情報が正しくないことが判明したときです。

DAT に関する情報は、DAT を伴う命令が実行される時、または、INVALIDATE PAGE TABLE ENTRY、LOAD REAL ADDRESS、または STORE REAL ADDRESS が実行される時に、使用されるものと見なされます。PSW で DAT が指定されていても、命令の実行前に入出力割り込み、外部割り込み、再始動割り込み、またはマシン・チェック割り込みが生じたとき、あるいは PSW が待ち状態を指定している場合は、この情報が使用されるものとは見なされません。仮想アドレスの変換時に使用中であると見なされるのは、そのアドレスを変更するために必要な情報のみです。特に、実際

に変換に使用されていないアドレス・スペース制御エレメントについては、そのアドレス・スペース制御エレメントの使用が原因で生じる可能性があるアドレッシング例外が認識されることはありません。

変換例外のリスト、各例外に対してとられる処置、および、複数の例外が該当する場合にそれらの例外が認識される優先順位は、6-35 ページの『アクセス例外の認識』に示されています。

変換索引バッファ

パフォーマンスを高めるために、通常、動的アドレス変換メカニズムがインプリメントされます。このメカニズムでは、領域テーブル、セグメント・テーブル、およびページ・テーブルに指定されている情報の一部が、変換索引バッファ (TLB) と呼ばれる特殊バッファに保持されます。CPU が実記憶域または絶対記憶域内の DAT テーブル・エントリを参照するのは、最初にそのエントリにアクセスするときだけです。この情報は TLB に入れておくことができるので、その後の変換は TLB 内の情報を使用して行うことができます。操作の整合性を維持するために、実スペース指定により指定された仮想/実記憶域間等価変換も、TLB 内の情報を使用して行うことができます。TLB の存在が変換プロセスに与える影響の範囲は、以下のとおりです。(1) 実記憶域または絶対記憶域内のテーブル・エントリの内容を変更しても、必ずしも変換に対して即時に効力が発生するとは限らない。(2) アドレス・スペース制御エレメント (ASCE) の中の領域第 1 テーブル起点、領域第 2 テーブル起点、領域第 3 テーブル起点、セグメント・テーブル起点、または実スペース・トークン起点のうち、2 つの起点のタイプが違っていても、同じ値の起点を含む ASCE を使用した形成された TLB エントリが選択されることがある。(3) 指定されたテーブル・エントリと等価の TLB が使用されていれば、アドレス・スペース制御エレメント内のテーブル長との比較は省略できる。複数 CPU 構成では、各 CPU について専用の TLB が 1 つずつあります。

TLB 内のエントリは、プログラムにより明示的にアドレッシングすることはできません。

TLB への情報の保存が許される条件が生じていても、必ず TLB に情報が保持されるとは限りません。さらに、TLB 内の情報は、消去が必須となる場合のほか、他の条件下でも消去されることがあります。

TLB の構造

ここでは、z/Architecture の定義に従って稼働するすべてのシステムのインプリメンテーションを対象として、TLB の論理構造について説明します。TLB エントリーには、TLB 領域/セグメント・テーブル結合エントリー、TLB ページ・テーブル・エントリー、および TLB 実スペース・エントリーの 3 種類があります。TLB 領域/セグメント・テーブル結合エントリーまたは TLB ページ・テーブル・エントリーは、実記憶域または絶対記憶域内のテーブル・エントリーから得られる情報と、その情報を記憶域から取り出すために使用する属性の両方を含むものと見なされます。TLB 実スペース・エントリーは、そのエントリーを形成するために使用するページ・フレーム実アドレス、実スペース・トークン起点、領域指標、セグメント指標、およびページ指標を含むものと見なされます。TLB 実スペース・エントリー内のトークン起点と、TLB 領域/セグメント・テーブル結合エントリー内のテーブル起点との区別はつきません。

注: 以下のセクションでは、どのような条件下で情報が TLB に入れられるか、どのような条件下で TLB からの情報がアドレス変換に使用されるか、そして、変換テーブルに対する変更が変換プロセスにどのような影響を与えるかについて説明します。

TLB エントリーの形成

実記憶域または絶対記憶域の中のテーブル・エントリーから、TLB 領域/セグメント・テーブル結合エントリーおよび TLB ページ・テーブル・エントリーがどのように形成されるか、および、プログラムで記憶域内のテーブル・エントリーの内容を操作したときにどのような影響が生じるかは、記憶域内のエントリーが特定の CPU に連結されているかどうか、およびそれらのエントリーが有効かどうかによって異なります。

テーブル・エントリーの連結された状態とは、そのエントリーが連結されている CPU が、暗黙的なアドレス変換のためにそのエントリーを使用できることを意味します。テーブル・エントリーは、同時に複数の CPU に連結することもできます。

テーブル・エントリーの有効な状態とは、そのテーブル・エントリーに関連した領域セット、領域、セグメント、またはページが使用可能であることを意味します。エントリー内の領域無効ビット、セグメント無効ビット、またはページ無効ビットが 0 であれば、そのエントリーは有効です。

TLB 領域/セグメント・テーブル結合エントリーを形成するために使用する領域テーブル・エントリー (存在する場合) およびセグメント・テーブル・エントリーを、**変換パス** と呼びます。変換パス内のすべてのエントリ

ーが連結されていて有効であり、変換に使用したときに変換指定例外を引き起こすおそれがないければ、その変換パスは、領域/セグメント・テーブル結合エントリーとして TLB に入れることができます。同様に、ページ・テーブル・エントリーが連結されていて有効であり、変換に使用したときに変換指定例外を引き起こすおそれがないければ、そのエントリーは TLB に入れることができます。

変換パス内の最高レベルのテーブル・エントリーが連結されるのは、そのエントリーが、連結するアドレス・スペース制御エレメント (ASCE) が指示しているテーブル内に存在する場合です。「テーブル内に存在する」とは、ASCE 内の起点フィールドおよび長さフィールドにより決定される状態を意味します。ASCE が連結 ASCE となるのは、以下の条件がすべて満たされているときです。

1. 現 PSW で DAT がオンに指定されている。
2. 現 PSW に、早期指定例外が認識される原因となるようなエラーが含まれていない。
3. ASCE が、下記の a、b、c、または d に示す要件を満たしている。
 - a. ASCE は制御レジスター 1 の中の 1 次 ASCE であり、CPU はホーム・スペース・モードではない。
 - b. ASCE は制御レジスター 7 の中の 2 次 ASCE であり、以下のいずれかの要件が満たされている。
 - CPU は、2 次スペース・モードまたはアクセス・レジスター・モードになっている。
 - CPU は 1 次スペース・モードであり、制御レジスター 0 の 2 次スペース制御 (ビット 37) が 1 である。
 - c. ASCE は、連結された有効な ASN 2 次テーブル・エントリー (ASTE) または使用可能な ALB ASTE の中にあり、CPU はアクセス・レジスター・モードになっている。ここで使用している用語の意味については、5-53 ページの『ART 索引バッファ』を参照してください。
 - d. ASCE は制御レジスター 13 の中でホーム ASCE であり、CPU は 2 次スペース・モードではない。

変換パス内のその他の各テーブル・エントリーが連結されるのは、次に高いレベルのエントリーが連結されて有

効な状態にあり、変換に使用した場合に変換指定例外を引き起こすおそれがなく、そして、そのエントリーが次に高いレベルのエントリーにより指示されているテーブル内に存在する場合があります。「テーブル内に存在する」とは、次に高いレベルのエントリー内の起点フィールド、オフセット・フィールド、および長さフィールドにより決定される状態を意味します。

ページ・テーブル・エントリーが連結されるのは、そのエントリーが、変換に使用した場合に変換指定例外を引き起こすおそれのない連結された有効なセグメント・テーブル・エントリーが指示しているページ・テーブルの中、または使用可能な TLB 領域/セグメント・テーブル結合エントリーが指示するページ・テーブルの中に存在する場合があります。使用可能な TLB 領域/セグメント・テーブル・エントリーについては、次のセクションで説明します。

領域テーブル・エントリーまたはセグメント・テーブル・エントリー内のテーブル・タイプ・ビット (ビット 60 および 61) と、変換プロセスで変換パスを使用するときにそのエントリーが出現するレベルとの間に整合性がない場合は、変換指定例外が生じます。セグメント・テーブル・エントリーを選択するために使用されるアドレス・スペース制御エレメントの中で専用スペース制御ビットが 1 で、そのエントリー内の共通セグメント・ビットが 1 である場合も、変換指定例外が生じます。また、ページ・テーブル・エントリー内のビット 52 または 55 が 1 である場合も、変換指定例外が生じます。

連結実スペース指定が存在するときは、いつでも TLB 実スペース・エントリーを形成できます。このエントリーは、実スペース指定の中の実スペース・トークン起点と、仮想アドレスのビット 0-51 のいずれかの値を使用して、形成されます。

TLB エントリーの使用

TLB エントリーの *使用可能な状態*とは、CPU が暗黙的なアドレス変換のためにその TLB エントリーを使用できることを意味します。使用可能な TLB エントリーは、次に低いレベルのテーブルがあればそれに連結し、暗黙アドレス変換の特定インスタンス用に使用できません。

TLB 領域/セグメント・テーブル結合エントリーが使用可能な状態にあると見なされるのは、以下の条件がすべて満たされているときです。

1. 現 PSW で DAT がオンに指定されている。
2. 現 PSW に、早期指定例外が認識される原因となるようなエラーが含まれていない。

3. TLB 領域/セグメント・テーブル結合エントリーが、以下のいずれか 1 つの要件を満たしている。

- a. TLB エントリー内の共通セグメント・ビットが 1 である。
- b. TLB エントリー内のテーブル起点 (TO) フィールドが、連結するアドレス・スペース制御エレメント内のテーブル起点フィールドまたはトークン起点フィールドに一致している。

TLB 領域/セグメント・テーブル結合エントリーを、暗黙的アドレス変換の特定インスタンス用に使用できるのは、その TLB エントリーが使用可能な状態にあり、かつ、その TLB エントリー内の共通セグメント・ビットが 1 であるか、あるいは、その TLB エントリー内のテーブル起点 (TO) フィールドが変換に使用されているアドレス・スペース制御エレメント内のテーブル起点フィールドまたはトークン起点フィールドに一致しており、かつ、その TLB エントリー内の領域指標フィールドおよびセグメント指標フィールドが、変換している仮想アドレスの対応するフィールドに一致している場合のみです。しかし、TLB 領域/セグメント・テーブル結合エントリー内の共通セグメント・ビットが 1 であり、かつ、変換に使用しているアドレス・スペース制御エレメント内の専用スペース制御ビットが 1 であるか、またはそのアドレス・スペース制御エレメントが実スペース指定である場合は、そのエントリーは使用されません。どちらの場合も、TLB エントリー内のテーブル起点フィールドと、アドレス・スペース制御エレメント内のテーブル起点フィールドまたはトークン起点フィールドが一致していても、そのエントリーは使用されません。

TLB ページ・テーブル・エントリーを暗黙的アドレス変換の特定インスタンス用に使用できるのは、そのエントリー内のページ・テーブル起点フィールドが、変換に使用しているセグメント・テーブル・エントリー内のページ・テーブル起点フィールドに一致しており、かつ、TLB ページ・テーブル・エントリー内のページ指標フィールドが、変換している仮想アドレスのページ指標に一致している場合のみです。

TLB 実スペース・エントリーを暗黙的アドレス変換に使用できるのは、その TLB エントリー内のトークン起点フィールドが、変換に使用しているアドレス・スペース制御エレメント内のテーブル起点フィールドまたはトークン起点フィールドに一致しており、かつ、その TLB エントリー内の領域指標フィールド、セグメント指標フィールド、およびページ指標フィールドが、変換している仮想アドレスの対応するフィールドに一致している場合のみです。

LOAD REAL ADDRESS のオペランド・アドレスは、DAT がオンでもオフでも、TLB の内容を使用して変換できますが、TLB エントリーが形成されるのは DAT がオンの場合のみです。

プログラミング上の注意:

1. テーブル・エントリーの内容は、そのテーブル・エントリーが連結されていて有効な状態にある場合のみ TLB にコピーできますが、そのコピーは、テーブル・エントリー自体が連結された状態または有効な状態でなくなった後も、TLB 内に残しておくことができます。
2. DAT がオフのときは、テーブル・エントリーは連結されていないため、内容を TLB にコピーすることはできません。特に、DAT がオフのときに LOAD REAL ADDRESS のオペランド・アドレスを変換した場合には、エントリーは TLB には入れられません。

逆に、DAT がオンのときは、現行の変換パラメーター、アドレス・スペース制御ビットの設定、およびアドレス・レジスターの内容に基づき、アドレス変換に使用できるすべての変換テーブル・エントリーから TLB に情報をコピーできます。TLB のロードは、エントリーが現在の命令の実行の一環として行われる変換に使用されるかどうかに関係なく行うことができ、また、このようなロードは、CPU が待ち状態になっているときにも発生します。
3. TLB の中には、1 つのテーブルの内容の複数のコピーが存在することがあります。例えば、インプリメンテーションによっては、有効なテーブル・エントリーの連結に使用される個々のアドレス・スペース制御エレメント内のテーブル起点ごとに、そのエントリーの内容のコピーが TLB に入れられることがあります。

変換テーブルの変更

連結されていて無効な状態にあるテーブル・エントリーが有効にされたときに、関連の仮想アドレスの変換に使用できるエントリーが TLB 内にない場合は、現在の操作単位の終わりまでにその変更の効力が発生します。同様に、有効ではあるが連結はされていないテーブル・エントリーが連結されたときに、関連の仮想アドレス用に使用できるエントリーが TLB 内にない場合も、現在の操作単位の終わりまでにその変更の効力が発生します。

連結された有効なテーブル・エントリーが変更された後、そのエントリーの代用としての適格性を備えたエントリーが TLB から消去される前に、変換のためにそのエントリーを必要とする仮想アドレスを使用して記憶域を参照しようとした場合は、以下に示す程度の範囲内で

予測不能の結果が生じます。つまり、新しい値の使用が開始されるのは、命令と命令の間のこともあり、また、変更の原因となった命令も含めていずれかの命令の実行中のこともあります。さらに、そのエントリーの代用となるエントリーが TLB から消去されるまでは、TLB には新旧両方の値が含まれていることがあり、特定のアクセスの際にどちらの値が選択されるかは予測不能です。

TLB 内に変換パスの新旧両方の値が存在する場合は、ページ・テーブル・エントリーがいずれか一方の値を使用して取り出され、もう一方の値に関連した TLB に入れられることがあります。パスの新しい値が例外の原因となるような値である場合は、その例外の結果割り込みが起きるときと起きないときがあります。割り込みが起きた場合は、その例外が通常抑止または無効化を生じさせるようなものであっても、命令の結果のフィールドが変更されることがあります。

TLB からのエントリーの消去は、以下の規則に従って行われます。

1. PURGE TLB または SET PREFIX を実行した場合、および CPU をリセットした場合は、TLB からすべてのエントリーが消去されます。
2. 構成内のいずれかに CPU で COMPARE AND SWAP AND PURGE が実行されると、その命令で使用されている汎用レジスター内のビット設定によっては、構成内のすべての TLB からすべてのエントリーが消去されることがあります。
3. 構成内のいずれかの CPU により INVALIDATE PAGE TABLE ENTRY が実行されると、構成内のすべての TLB から、選択されたエントリーが消去されます。
4. 上記に規則に従って必要とされた場合以外にも、一部またはすべての TLB エントリーが消去されることがあります。

プログラミング上の注意:

1. TLB 内のエントリーは、そのエントリーを形成する元になったテーブル・エントリーが非連結または無効状態になった後も、引き続き変換に使用されることがあります。TLB エントリーは、TLB から明示的に消去しない限り、必ず除去されるとは限りません。

連結された有効なエントリーに対する変更、またはテーブル・エントリーを連結し有効にするための変更があった場合、そのテーブル・エントリーの代用として使用できる TLB エントリーがなければ、その変更は、次の命令または次の命令より前の変換プロセスに反映されます。しかし、テーブル・エントリーを非連結または無効にするための変更は、その

テーブル・エントリーの代用としての適格性を備えたエントリーが TLB から消去されるまでは、変換プロセスに反映されない可能性があります。

- 命令実行の開始の一環として、オペランドのアクセス可能性を確認するための事前テストが行われたときに、動的アドレス変換に関連した例外が設定されることがあります。したがって、命令がオペランドを一度に 1 バイトずつ処理するものと見なされている場合に、命令が使用するテーブル・エントリーの妥当性検査を行い、そのテーブル・エントリーが有効であるように見えたとしても、実行開始の時点でそのテーブル・エントリーが無効であれば、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、またはページ変換例外が示されることがあります。

- 連結されているテーブル・エントリーに対して、I ビットを 0 に設定するかまたはページ・テーブル・エントリーの右端のビットを変更する以外の変更を行った場合に、そのエントリーの内容のすべてのコピーが TLB から消去される前に、そのエントリーが変換に使用されると、予測不能の結果が生じることがあります。つまり、新しい値の使用が開始されるのは、命令と命令の間のもあり、また、変更の原因となった命令も含めて、いずれかの命令の実行中のこともあります。例えば、MOVE (MVC) のような命令が連結されたテーブル・エントリーに変更を加え (そのエントリーを無効にする変更も含む)、続いてそのエントリーを変換に使用した場合は、事前に TLB からそのエントリーを消去することなく、変更されたエントリーが使用されるため、結果の値も例外認識の有無も予測不能になります。

連結されたテーブル・エントリーを操作することで、誤ったテーブル・エントリー値が TLB に記録されることがあります。例えば、変更が断片的に行われる場合は、連結された有効なエントリーを変更すると、部分的に更新されたエントリーが記録されることがあります。あるいは、変更のプロセスに中間値が導入された場合は、おそらく無効と認められるエントリーが一時的に有効とみなされ、TLB に記録されることがあります。このような中間値が導入される可能性があるのは、再試行された入出力操作により変更が行われた場合、または、1 つの命令の実行中に中間値が生じる場合です。

もう 1 つの例として、セグメント・テーブル・エントリーが別のページ・テーブルを指すように変更された後で、TLB を消去することなくそのエントリーが使用された場合は、新しいページ・テーブル・エントリーが取り出されて、古いページ・テーブル起点に関連付けられることがあります。この状況で、新しいページ・テーブル起点を指定した

INVALIDATE PAGE TABLE ENTRY を実行した場合に、新しいページ・テーブルから取り出されたページ・テーブル・エントリーが必ず消去されるとは限りません。

- ページ・テーブルの操作を容易にするために、INVALIDATE PAGE TABLE ENTRY 命令が用意されています。この命令は、ページ・テーブル・エントリーの I ビットを 1 に設定し、構成内のすべての TLB から、そのテーブル・エントリーに基づいて形成されたエントリーを消去します。

INVALIDATE PAGE TABLE ENTRY は、ページ・テーブル・エントリーの I ビットを 1 に設定して、構成内の各 CPU の TLB からそのエントリーの TLB コピーを消去したい場合に、便利です。INVALIDATE PAGE TABLE ENTRY を使用するときには、TLB 操作について以下の事項を考慮に入れる必要があります。(INVALIDATE PAGE TABLE ENTRY の項の『プログラミング上の注意』も参照してください。)

- ページ・テーブル・エントリーに対して、右端バイトを変更する以外の変更を加える場合は、その前に INVALIDATE PAGE TABLE ENTRY を実行してください。そうしないと、INVALIDATE PAGE TABLE ENTRY に選択的消去部分が指定されていても、そのエントリーの TLB コピーが消去されないことがあります。
- INVALIDATE PAGE TABLE ENTRY を使用して、1 つのページ・テーブル内のすべてのページ・テーブル・エントリーを無効にしても、そのページ・テーブルを指す領域/セグメント・テーブル結合エントリーがある場合に、必ずしも TLB からそのエントリーが消去されるとは限りません。領域/セグメント・テーブル結合エントリーの TLB を無効にして消去する必要がある場合は、以下の注 5 に示す規則に従う必要があります。
- 一時点で多数のページ・テーブル・エントリーを無効にするときは、各ページ・テーブル・エントリーごとに INVALIDATE PAGE TABLE ENTRY を発行するより、COMPARE AND SWAP AND PURGE (TLB をバージするもの) または PURGE TLB を使用し、下記の注 5 の規則に従う方が、オーバーヘッドが小さくてすむ可能性があります。

- テーブル・エントリーの操作は、以下の規則に従って行ってください。以下の規則に従っていれば、変換プロセスで常に実記憶域または絶対記憶域からの

テーブル・エントリーが使用されるものとして、変換が行われます。

- a. 有効なテーブル・エントリーがいずれかの CPU に連結されていて、その CPU による変換に使用される可能性がある間は、次の 3 つの場合を除き、そのエントリーを変更してはなりません。(1) INVALIDATE PAGE TABLE ENTRY を使用してそのエントリーを無効にする。(2) ページ・テーブル・エントリーのビット 56-63 を変更する。または、(3) TLB をページする COMPARE AND SWAP AND PURGE を使用して変更を加える。
- b. 連結されている有効なテーブル・エントリー、または連結されていないかまたは無効なテーブル・エントリーに対して、ページ・テーブル・エントリーのビット 56-63 を変更する以外の変換を加えた場合は、その変更の後で、そのエントリーから形成された TLB エントリーを持っている可能性がある各 CPU で暗黙的変換にそのエントリーが使用される前に、それらの CPU の TLB をページする必要があります。(INVALIDATE PAGE TABLE ENTRY 命令、または TLB をページする COMPARE AND SWAP AND PURGE 命令を使用して変更を行った場合は、別個にページを行う必要はないという点に注意してください)。テーブル・エントリーが連結されてかつ有効である場合は、変換のためにそのエントリーを必要とするプログラムが実行されていないことが分かっている場合でも、この規則が適用されます。
- c. 無効なテーブル・エントリーに変更を加えて、そのエントリー内に中間有効値が現れることができるようにした場合は、その変更の後で、そのエントリーが連結されている各 CPU による暗黙アドレス変換にそのエントリーが使用される前に、それらの CPU の TLB がページされるようにする必要があります。
- d. 特定のテーブルについて指定されているオフセットまたは長さを変更した場合は、その変更の後で、そのテーブル内に存在しなくなったテーブル・エントリーから形成された TLB エントリーを持つ可能性のある各 CPU による暗黙的変換にそのテーブルが使用される前に、それらの CPU の TLB がページされるようにする必要があります。

無効なページ・テーブル・エントリーが、中間有効値を導入することなく有効にされた場合、そのエントリーから形成された TLB エントリーを持っていない CPU では、TLB を消去する必要はありません。

同様に、無効な領域テーブル・エントリーまたはセグメント・テーブル・エントリーを、中間有効値を導入することなく有効にした場合は、その有効にされたエントリーから形成された TLB エントリーを持っておらず、かつその有効にされたエントリーを使用して連結されたページ・テーブル内のエントリーから形成された TLB エントリーを持っていない CPU では、TLB を消去する必要はありません。

一部のモデルでは、PURGE TLB、COMPARE AND SWAP AND PURGE、または SET PREFIX を実行すると、パフォーマンスが低下することがあります。したがって、これらの命令の使用は、上記の規則に従って最小限にとどめるようにしてください。

アドレスの要約

変換されるアドレス

プログラムにより明示的に指定され、記憶域を参照するために CPU により使用されるアドレスのほとんどは、命令アドレスまたは論理アドレスであり、したがって、DAT がオンのときは暗黙的変換の対象になります。同様に、割り込みの時点または命令実行の結果としてプログラムに提示される、対応するアドレスも、命令アドレスまたは論理アドレスです。LOAD REAL ADDRESS および STORE REAL ADDRESS のオペランド・アドレスは、PSW で DAT がオンに指定されているかオフに指定されているかに関係なく、明示的に変換されません。

命令の B フィールドおよび D フィールドに指定されている値から形成されているが、記憶域のアドレッシングには使用されない数値には、変換は適用されません。これには、LOAD ADDRESS、LOAD ADDRESS EXTENDED、MONITOR CALL、およびシフト命令のオペランド・アドレスなどが含まれます。また、これには、制御レジスター 10 および 11 の中の、PER の始めと終わりの位置を指示するアドレスも含まれます。

INSERT VIRTUAL STORAGE KEY および TEST PROTECTION は例外として、記憶キーを明示的に指示するアドレス (SET STORAGE KEY EXTENDED、INSERT STORAGE KEY EXTENDED、および RESET REFERENCE BIT EXTENDED のオペランド・アドレス) は、実アドレスです。同様に、割り込みなどのシーケンスのために CPU が暗黙的に使用するアドレスも、実アドレスです。

チャンネル・プログラムが、データを転送するため、および CCW または IDAW を参照するために使用するアドレスは、絶対アドレスです。

DIAGNOSE に関連した記憶域アドレスの取り扱い、モデルによって異なります。

アドレスの処理については、動的アドレス変換およびプレフィックシングも含めて、3-3ページの『アドレスのタイプ』で説明します。プレフィックシング (提供される場合) は、動的アドレス変換機能を使用してアドレス

が変換された後で適用されます。プレフィックシングの説明については、3-14ページの『プレフィックシング』を参照してください。

アドレスの取り扱い

図3-10 に、アドレスの取り扱いについての要約を示します。この図には、プログラムが会うすべてのアドレスをリストし、それぞれのアドレス・タイプを示しています。

仮想アドレス

- INSERT VIRTUAL STORAGE KEY の記憶域オペランドのアドレス。
- LOAD REAL ADDRESS および STORE REAL ADDRESS の中のオペランド・アドレス。
- MOVE TO PRIMARY および MOVE TO SECONDARY の記憶域オペランドのアドレス。
- ASCE タイプの例外、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、またはページ変換例外に対するプログラム割り込みの際に、実記憶位置 168 のダブルワードに記憶されるアドレス。
- 制御レジスター 15 の中のリンケージ・スタック・エントリー・アドレス。
- リンケージ・スタック・ヘッダー・エントリーの中のバックワード・スタック・エントリー・アドレス。
- リンケージ・スタック・トレーラー・エントリーの中のフォワード・セクション・ヘッダー・アドレス。
- ディスパッチ可能単位制御テーブルの中のトラップ制御ブロック・アドレス。
- トラップ制御ブロックの中のトラップ保管域アドレスおよびトラップ・プログラム・アドレス。

命令アドレス

- PSW 中の命令アドレス。
- ブランチ・アドレス。
- EXECUTE のターゲット。
- PER のためのプログラム割り込みの際に実記憶位置 152 のダブルワードに格納されるアドレス。
- BRANCH AND LINK、BRANCH AND SAVE、BRANCH AND SAVE AND SET MODE、BRANCH AND STACK、BRANCH IN SUBSPACE GROUP、BRANCH RELATIVE AND SAVE、BRANCH RELATIVE AND SAVE LONG、および PROGRAM CALL によって汎用レジスターに置かれるアドレス。
- BRANCH AND STACK によって汎用レジスター内で使用されるアドレス。
- 縮小権限状態で実行された BRANCH AND SET AUTHORITY によって汎用レジスター置かれるアドレス。

論理アドレス

- 命令の記憶域オペランドのアドレスで、他に指定されていないもの。
- EDIT AND MARK および TRANSLATE AND TEST によって汎用レジスター 1 に置かれるアドレス。
- MOVE LONG、MOVE LONG EXTENDED、COMPARE LOGICAL LONG、および COMPARE LOGICAL LONG EXTENDED によって汎用レジスター内で更新されるアドレス。
- CHECKSUM、COMPARE AND FORM CODEWORD、および UPDATE TREE によって汎用レジスター内で更新されるアドレス。
- 第 2 オペランド・アドレスが 0 以外である場合の TEST PENDING INTERRUPTION のアドレス。
- RESUME PROGRAM のパラメーター・リストのアドレス。

図 3-10 (1/2). アドレスの取り扱い

実アドレス

- INSERT STORAGE KEY EXTENDED、RESET REFERENCE BIT EXTENDED、および SET STORAGE KEY EXTENDED の記憶キーのアドレス。
- LOAD USING REAL ADDRESS、STORE USING REAL ADDRESS、および TEST BLOCK の記憶域オペランドのアドレス。
- LOAD REAL ADDRESS および STORE REAL ADDRESS によって生成される変換済みアドレス。
- ページ・テーブル・エントリー内のページ・フレーム実アドレス。
- 制御レジスター 12 の中のトレース・エントリー・アドレス。
- 制御レジスター 14 の中の ASN 第 1 テーブル起点。
- ASN 第 1 テーブル・エントリーの中の ASN 第 2 テーブル起点。
- アクセス・レジスター変換に使用される場合以外の、ASN 第 2 テーブル・エントリーの中の権限テーブル起点。
- 1 次 ASN 第 2 テーブル・エントリーの中のリンケージ・テーブルの起点。
- リンケージ・テーブル・エントリーの中のエントリー・テーブル起点。
- 制御レジスター 2 の中のディスパッチ可能単位制御テーブル起点。
- 制御レジスター 5 の中の 1 次 ASN 第 2 テーブル・エントリー起点。
- ディスパッチ可能単位制御テーブルの中の基本 ASN 第 2 テーブル・エントリー起点およびサブスペース ASN 第 2 テーブル・エントリー起点。
- エントリー・テーブル・エントリーおよびアクセス・リスト・エントリーの中の ASN 第 2 テーブル・エントリー・アドレス。

永続割り当て済み実アドレス

- 第 2 オペランド・アドレスが 0 のときに TEST PENDING INTERRUPTION が格納するダブルワードのアドレス。
- 割り込み時に使用される、PSW、割り込みコード、および関連情報のアドレス。
- マシン・チェック・ログアウトおよび保管域のために使用されるアドレス。
- STORE FACILITY LIST オペランドのアドレス。

実か絶対かを予測できないアドレス

- 制御レジスター 1、7、および 13 の中、アクセス・レジスター指定のアドレス・スペース制御エレメントの中、および、領域第 1 テーブル・エントリー、領域第 2 テーブル・エントリー、または領域第 3 テーブル・エントリーの中の、領域第 1 テーブル起点、領域第 2 テーブル起点、領域第 3 テーブル起点、またはセグメント・テーブル起点。
- セグメント・テーブル・エントリーの中、および INVALIDATE PAGE TABLE ENTRY の中のページ・テーブル起点。
- LOAD REAL ADDRESS が提供するセグメント・テーブル・エントリーまたはページ・テーブル・エントリー。
- アクセス・レジスター変換により使用される、ディスパッチ可能単位起点または 1 次スペース・アクセス・リスト起点および権限テーブル起点 (使用される ALE により指定された ASTE の中の)。

絶対アドレス

- プレフィックス値。
- ORB の中のチャンネル・プログラム・アドレス。
- CCW の中のデータ・アドレス。
- 間接データ・アドレッシングを指定する CCW 内の IDAW アドレス。
- チャンネル内の転送を指定する CCW 内の CCW アドレス。
- IDAW の中のデータ・アドレス。
- SET CHANNEL MONITOR で指定されている測定プログラム起点。
- SET ADDRESS LIMIT で指定されているアドレス限界。
- SIGNAL PROCESSOR 命令のアドレス指定状態記憶指令で使用されるアドレス。
- 実記憶位置 248 のダブルワードに格納される故障記憶機構アドレス。
- SCSW の中の CCW アドレス。

永続割り当て済み絶対アドレス

- 状況記憶機能のために使用されるアドレス。
- 初期プログラム・ローディング用に使用される PSW および最初の 2 つの CCW のアドレス。

記憶機構の参照には使用されないアドレス

- 制御レジスター 10 の中の PER 開始アドレス。
- 制御レジスター 11 の中の PER 終了アドレス。
- モニター・イベント用として実記憶位置 176 のダブルワードに格納されるアドレス。
- シフト命令や、記憶機構参照にアドレスを使用しないものとして指定されているその他の命令の中のアドレス。
- 実スペース指定の中の実スペース・トークン起点。

図 3-10 (2/2). アドレスの取り扱い

割り当て済みの記憶位置

3-55ページの図3-11に、記憶機構内の割り当て済み記憶位置のフォーマットと範囲を示しています。記憶位置は以下のように使用されます。

128-131 (実アドレス)

外部割り込みパラメーター：サービス信号または外部時刻参照 (ETR) を原因とする外部割り込みの際に、その割り込みに関連したパラメーターが記憶位置 128-131 に格納されます。

132-133 (実アドレス)

CPU アドレス：誤動作警報、緊急信号、または外部呼び出しを原因とする外部割り込みの際に、その割り込みの原因に関連した CPU アドレスが、記憶位置 132-133 に格納されます。その他のすべての外部割り込み条件の場合は、記憶位置 132-133 には 0 が格納されます。

134-135 (実アドレス)

外部割り込みコード：外部割り込みの際に、割り込みコードが記憶位置 134-135 に格納されます。

136-139 (実アドレス)

監視プログラム呼び出し識別：監視プログラム呼び出し割り込みの際に、記憶位置 137 のビット位置 5 および 6 に命令長コードが格納され、記憶位置 138-139 に割り込みコードが格納されます。記憶位置 136、および記憶位置 137 の残りのビット位置には、0 が格納されます。

140-143 (実アドレス)

プログラム割り込み識別：プログラム割り込みの際に、記憶位置 141 のビット位置 5 および 6 に命令長コードが格納され、記憶位置 142-143 に割り込みコードが格納されます。記憶位置 140、および記憶位置 141 の残りのビット位置には、0 が格納されます。

144-147 (実アドレス)

データ例外コード (DXC)：データ例外を原因とするプログラム割り込みの際に、記憶位置 147 にデータ例外コードが格納され、記憶位置 144-146 には 0 が格納されます。DXC に

ついては、6-14ページの『データ例外コード (DXC)』で説明します。

148-149 (実アドレス)

モニター・クラス番号：モニター・イベントを原因とするプログラム割り込みの際に、記憶位置 149 にモニター・クラス番号が格納され、記憶位置 148 には 0 が格納されます。

150-151 (実アドレス)

PER コード：PER イベントを原因とするプログラム割り込みの際に、記憶位置 150-151 のビット位置 0-2 および 4 に PER コードが格納され、さらに、4-26ページの『原因の識別』の説明に従ってその他の情報が格納されます。

152-159 (実アドレス)

PER アドレス：PER イベントを原因とするプログラム割り込みの際に、PER アドレスが記憶位置 152-159 に格納されます。

160 (実アドレス)

例外アクセス識別：ASCE タイプ例外、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、またはページ変換例外を原因とするプログラム割り込みの際に、その例外が適用されるアドレス・スペースを示す値が記憶位置 160 に格納されることがあります。CPU がアクセス・レジスター・モードにあり、アクセスが命令取り出し (EXECUTE 命令のターゲットの取り出しも含む) であった場合は、記憶位置 160 には 0 が格納されます。CPU がアクセス・レジスター・モードにあり、アクセスが、AR 指定のアドレス・スペース制御要素を使用する記憶域オペランド参照であった場合は、使用されているアクセス・レジスターの番号が記憶位置 160 のビット位置 4-7 に格納され、ビット位置 0-3 には 0 が格納されます。(上記 2 つの場合は、実記憶位置 168-175 のビット位置 62 および 63 に格納される値に関係なく、記憶位置 160 への格納が行われます)。CPU がアクセス・レジスター・モードにあり、アクセスがリンケージ・スタックに対する暗黙参照であった場合、または、CPU がアクセス・レジスター・モードではなかった場合は、記憶位置 160 の内容は予測不能です。

ALEN 変換例外、ALE シーケンス例外、ASTE 妥当性例外、ASTE シーケンス例外、または拡張権限例外を原因とするプログラム割り込みの際には、使用されたアクセス・レジスタの番号が記憶位置 160 のビット位置 4-7 に格納され、ビット位置 0-3 には 0 が格納されます。サブスペース置換操作で認識されたASTE 妥当性例外または ASTE シーケンス例外を原因とするプログラム割り込みの際には、記憶位置 160 にはすべて 0 が格納されます。

記憶保護例外を原因とするプログラム割り込みの際には、記憶位置 160 には、3-12ページの『保護時の抑止』に示すように情報が格納されます。

161 (実アドレス)

PER アクセス識別: PER 記憶域変更イベントを原因とするプログラム割り込みの際には、そのイベントが適用されるアドレス・スペースを示す値が、記憶位置 161 に格納されることがあります。このアクセスに AR 指定のアドレス・スペース制御エレメントが使用された場合は、使用されたアクセス・レジスタの番号が記憶位置 161 のビット位置 4-7 に格納され、ビット位置 0-3 には 0 が格納されます。(1) CPU がアクセス・レジスタ・モードであり、アクセスがリンケージ・スタックへの暗黙参照であった場合、または、(2) CPU がアクセス・レジスタ・モードではなかった場合は、記憶位置 161 の内容は予測不能です。

162 (実アドレス)

オペランド・アクセス識別: MOVE PAGE 命令で認識されたページ変換例外を原因とするプログラム割り込みの際には、この命令の R1 フィールドの内容が記憶位置 162 のビット位置 0-3 に格納され、R2 フィールドの内容がビット位置 4-7 に格納されます。MOVE PAGE 以外の命令の実行時にページ変換例外が認識された場合、または、ASCE タイプ例外、領域第 1 変換例外、領域第 2 変換例外、

領域第 3 変換例外、またはセグメント変換例外が認識された場合は、記憶位置 162 の内容は予測不能です。

163 (絶対アドレス)

状況記憶アーキテクチャー・モード識別: 状況記憶操作の実行時に、記憶位置 163 のビット位置 0-6 には 0 が格納され、ビット位置 7 には 1 が格納されます。

163 (実アドレス)

マシン・チェック・アーキテクチャー・モード識別: マシン・チェック割り込みの際に、記憶位置 163 のビット位置 0-6 には 0 が格納され、ビット位置 7 には 1 が格納されます。

168-175 (実アドレス)

変換例外識別: ASCE タイプ例外、領域第 1 変換、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、またはページ変換例外を原因とするプログラム割り込みの際には、その例外の原因となった仮想アドレスのビット 0-51 が、記憶位置 168-175 のビット位置 0-51 に格納されます。このアドレスは、変換例外アドレスと呼ばれることもあります。記憶位置 168-175 のビット 52-60 は予測不能です。例外が、MOVE PAGE の実行時に認識されたページ変換例外である場合は、記憶位置 168-175 のビット 61 は 1 に設定されます。例外が、MOVE PAGE 以外の命令の実行時に認識されたページ変換例外である場合は、ビット 61 は 0 に設定されません。例外が、ASCE タイプ例外、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外である場合は、記憶位置 168-175 のビット 61 は予測不能です。関連情報については、実記憶位置 162 の説明を参照してください。

記憶位置 168-175 のビット 62 および 63 は、以下のように、変換に使用されるアドレス・スペース制御エレメント (ASCE) を識別する値に設定されます。

ビット
62 63

意味

- 0 0 1次 ASCE が使用されました。
- 0 1 CPU はアクセス・レジスター・モードであり、アクセスは、命令取り出しか、または AR 指定の ASCE を使用する記憶域オペランド参照でした (つまり、アクセスはリンケージ・スタックに対する暗黙的参照ではありませんでした)。例外アクセス ID (実記憶位置 160) を調べることで、使用された ASCE を判別できます。ただし、1次、2次、またはホーム ASCE が使用された場合は、ビット 62 および 63 は、01 ではなく、それぞれ 00、10、または 11 に設定される場合があります。
- 1 0 2次 ASCE が使用されました。
- 1 1 ホーム ASCE が使用されました (リンケージ・スタックに対する暗黙参照の場合も含む)。

CPU がビット 62 および 63 を 01 に設定しないことがあるのは次の場合です。(1) アクセスが命令取り出しであることを認識した場合、(2) アクセス・リスト・エンタリー・トークン 00000000 または 00000001 (16 進数) が使用されたことを認識した場合、(3) または、アクセス・リスト・エンタリー・トークンが、アクセス・リスト・エンタリーを介して、1次 ASCE、2次 ASCE、またはホーム ASCE に等しい ASCE を含む ASN 第 2 テーブル・エンタリーを指定したことを認識した場合。

AFX 変換例外、ASX 変換例外、1次権限例外、または 2次権限例外を原因とするプログラム割り込みの際には、変換しようとしている ASN が記憶位置 174 および 175 に格納され、記憶位置 172 および 173 には 0 が格納され、記憶位置 168-171 の内容は変更されません。

スペース切り替えイベントを原因とするプログラム割り込みの際には、旧命令スペースの識別が記憶位置 174 および 175 に格納され、記憶位置 172 および 173 のビット位置 0 には旧命令スペースのスペース切り替え

イベント制御ビット、そしてビット位置 1-15 には 0 が入れられ、記憶位置 168-171 の内容は変更されません。格納される識別とビットは以下のとおりです。

- 操作の前に、CPU が 1次スペース・モード、2次スペース・モード、またはアクセス・レジスター・モードであった場合は、旧 PASN (操作の前の制御レジスター 4 のビット 48-63) が記憶位置 174 および 175 に格納され、旧 1次スペース切り替えイベント制御ビット (操作の前の制御レジスター 1 のビット 57) が、記憶位置 172 および 173 のビット位置 0 に入れられます。
- 操作の前に CPU がホーム・スペース・モードであった場合は、記憶位置 174 および 175 には 0 が格納され、記憶位置 172 および 173 のビット位置 0 には、ホーム・スペース切り替えイベント制御ビット (制御レジスター 13 のビット 57) が入れられます。

PROGRAM CALL により認識された LX 変換例外または EX 変換例外を原因とするプログラム割り込みの際には、記憶位置 172-175 のビット位置 12-31 に PC 番号が格納され、ビット位置 0-11 には 0 が格納され、記憶位置 168-171 の内容は変更されません。

記憶保護例外を原因とするプログラム割り込みの際には、記憶位置 168-175 には、3-12 ページの『保護時の抑止』に示すように情報が格納されます。

176-183 (実アドレス)

モニター・コード：モニター・イベントを原因とするプログラム割り込みの際に、モニター・コードが記憶位置 176-183 に格納されません。

184-187 (実アドレス)

サブシステム識別ワード：入出力割り込みの際に、サブシステム識別ワードが記憶位置 184-187 に格納されます。

188-191 (実アドレス)

入出力割り込みパラメーター：入出力割り込みの際に、関連のサブチャネルからの割り込みパラメーターが記憶位置 188-191 に格納されます。

- 192-195 (実アドレス)
 入出力割り込み識別ワード：入出力割り込みの際に、入出力割り込みのソースをさらに限定的に識別する入出力割り込み識別ワードが、記憶位置 192-195 に格納されます。
- 200-203 (実アドレス)
STFL 機能リスト：STORE FACILITY LIST 命令は、実記憶位置 200-203 に情報を格納します。ビット 0 が 1 の場合は、第7章、『一般命令』および第10章、『制御命令』に示されている命令の要約図の中で「N3」のマークが付けられている命令が、ESA/390 モードで使用できることを示します。ビット 1 が 1 の場合は、z/Architecture がインストールされていることを示します。ビット 2 が 1 の場合は、z/Architecture がアクティブであることを示します。ビット 16 が 1 の場合は、拡張変換機能 2 がインストールされていることを示します。ビット 3-15 および 17-31 には、0 が格納されます。
- 232-239 (実アドレス)
 マシン・チェック割り込みコード：マシン・チェック割り込みの際に、マシン・チェック割り込みコードが記憶位置 232-239 に格納されます。
- 244-247 (実アドレス)
 外部損傷コード：特定の外部損傷条件を原因とするマシン・チェック割り込みの際に、モデルに応じて外部損傷コードが記憶位置 244-247 に格納されることがあります。
- 248-255 (実アドレス)
 故障記憶機構アドレス：マシン・チェック割り込みの際に、64 ビットの故障記憶機構アドレスが記憶位置 248-255 に格納されることがあります。
- 288-303 (実アドレス)
 再始動時旧 PSW：再始動割り込みの際に、現 PSW が旧 PSW として記憶位置 288-303 に格納されます。
- 304-319 (実アドレス)
 外部旧 PSW：外部割り込みの際に、現 PSW が旧 PSW として記憶位置 304-319 に格納されます。
- 320-335 (実アドレス)
 監視プログラム呼び出し旧 PSW：監視プログラム呼び出し割り込みの際に、現 PSW が旧 PSW として記憶位置 320-335 に格納されます。
- 336-351 (実アドレス)
 プログラム旧 PSW：プログラム割り込みの際に、現 PSW が旧 PSW として記憶位置 336-351 に格納されます。
- 352-367 (実アドレス)
 マシン・チェック旧 PSW：マシン・チェック割り込みの際に、現 PSW が旧 PSW として記憶位置 352-367 に格納されます。
- 368-383 (実アドレス)
 入出力旧 PSW：入出力割り込みの際に、現 PSW が旧 PSW として記憶位置 368-383 に格納されます。
- 416-431 (実アドレス)
 再始動時新 PSW：再始動割り込みの際に、記憶位置 416-431 から新 PSW が取り出されません。
- 432-447 (実アドレス)
 外部新 PSW：外部割り込みの際に、記憶位置 432-447 から新 PSW が取り出されます。
- 448-463 (実アドレス)
 監視プログラム呼び出し新 PSW：監視プログラム呼び出し割り込みの際に、記憶位置 448-463 から新 PSW が取り出されます。
- 464-479 (実アドレス)
 プログラム新 PSW：プログラム割り込みの際に、記憶位置 464-479 から新 PSW が取り出されます。
- 480-495 (実アドレス)
 マシン・チェック新 PSW：マシン・チェック割り込みの際に、記憶位置 480-495 から新 PSW が取り出されます。
- 496-511 (実アドレス)
 入出力新 PSW：入出力割り込みの際に、記憶位置 496-511 から新 PSW が取り出されません。

4544-4607 (実アドレス)

プログラミング用に使用可能：記憶位置
4544-4607 は、プログラミング用に使用できません。

4608-4735 (絶対アドレス)

状況記憶浮動小数点レジスタ保管域：状況記憶操作の実行時に、浮動小数点レジスタの内容が記憶位置 4608-4735 に格納されません。

4608-4735 (実アドレス)

マシン・チェック浮動小数点レジスタ保管域：マシン・チェック割り込みの際に、浮動小数点レジスタの内容が記憶位置 4608-4735 に格納されます。

4736-4863 (絶対アドレス)

状況記憶汎用レジスタ保管域：状況記憶操作の実行時に、汎用レジスタの内容が記憶位置 4736-4863 に格納されます。

4736-4863 (実アドレス)

マシン・チェック汎用レジスタ保管域：マシン・チェック割り込みの際に、汎用レジスタの内容が記憶位置 4736-4863 に格納されます。

4864-4879 (絶対アドレス)

状況記憶 PSW 保管域：状況記憶操作の実行時に、現 PSW の内容が記憶位置 4864-4879 に格納されます。

4864-4879 (実アドレス)

固定ログアウト域：マシン・チェック割り込みの際に、モデルに応じて、ログアウト情報が記憶位置 4864-4879 に格納されます。

4888-4891 (絶対アドレス)

状況記憶プレフィックス保管域：状況記憶操作の実行時に、プレフィックス・レジスタの内容が記憶位置 4888-4891 に格納されません。

4892-4895 (絶対アドレス)

状況記憶浮動小数点制御レジスタ保管域：状況記憶操作の実行時に、浮動小数点制御レジスタの内容が 4892-4895 に格納されません。

4892-4895 (実アドレス)

マシン・チェック浮動小数点制御レジスタ保管域：マシン・チェック割り込みの際に、浮動小数点制御レジスタの内容が記憶位置 4892-4895 に格納されます。

4900-4903 (絶対アドレス)

状況記憶 TOD プログラマブル・レジスタ保管域：状況記憶操作の実行時に、TOD プログラマブル・レジスタの内容が記憶位置 4900-4903 に格納されます。

4900-4903 (実アドレス)

マシン・チェック TOD プログラマブル・レジスタ保管域：マシン・チェック割り込みの際に、TOD プログラマブル・レジスタが記憶位置 4900-4903 に格納されます。

4904-4911 (絶対アドレス)

状況記憶 CPU タイマー保管域：状況記憶操作の実行時に、CPU タイマーの内容が記憶位置 4904-4911 に格納されます。

4904-4911 (実アドレス)

マシン・チェック CPU タイマー保管域：マシン・チェック割り込みの際に、CPU タイマーの内容が記憶位置 4904-4911 に格納されません。

4913-4919 (絶対アドレス)

状況記憶クロック・コンパレータ保管域：状況記憶操作の実行時に、クロック・コンパレータのビット位置 0-55 の内容が記憶位置 4913-4919 に格納されます。この格納が行われるときは、記憶位置 4912 には 0 が格納されます。

4913-4919 (実アドレス)

マシン・チェック・クロック・コンパレータ保管域：マシン・チェック割り込みの際に、クロック・コンパレータのビット位置 0-55 の内容が記憶位置 4913-4919 に格納されます。この格納が行われるときは、記憶位置 4912 には 0 が格納されます。

4928-4991 (絶対アドレス)

状況記憶アクセス・レジスタ保管域：状況記憶操作の実行時に、アクセス・レジスタの内容が記憶位置 4928-4991 に格納されません。

4928-4991 (実アドレス)

マシン・チェック・アクセス・レジスター保管域: マシン・チェック割り込みの際に、アクセス・レジスターの内容が記憶位置 4928-4991 に格納されます。

4992-5119 (絶対アドレス)

状況記憶制御レジスター保管域: 状況記憶操作の実行時に、制御レジスターの内容が記憶位置 4992-5119 に格納されます。

4992-5119 (実アドレス)

マシン・チェック制御レジスター保管域: マシン・チェック割り込みの際に、制御レジスターの内容が記憶位置 4992-5119 に格納されます。

プログラミング上の注意:

1. CPU がアクセス・レジスター・モードにあるときは、複数のアドレス・スペース内のオペランドをアドレッシングする幾つかの命令 (MVCL など) は、1 つのアドレス・スペース内で記憶域変更 PER イベントを引き起こし、同時に別のアドレス・スペース内で領域変換例外、セグメント変換例外、またはページ変換例外を引き起こすことがあります。この場合、この 2 つの条件の原因となっているアクセス・レジスターはそれぞれ異なります。両方のアクセス・レジスターを識別するために、2 つのアクセス識別 (例外アクセス識別と PER アクセス識別) が提供されます。
2. 絶対記憶位置 163 の状況記憶アーキテクチャー・モード識別と、実記憶位置 163 のマシン・チェック・アーキテクチャー・モード識別は、CPU が z/Architecture アーキテクチャー・モードになっていることを示します。CPU に z/Architecture がインストールされていても、CPU が ESA/390 モードになっているときは、状況記憶操作およびマシン・チェック割り込み操作では、記憶位置 163 には 0 が格納されます。

16 進 数	10 進 数	フィールド
0	0	
4	4	
8	8	
C	12	
10	16	
14	20	
18	24	
1C	28	
20	32	
24	36	
28	40	
2C	44	
30	48	
34	52	
38	56	
3C	60	
40	64	
44	68	
48	72	
4C	76	
50	80	
54	84	
58	88	
5C	92	
60	96	
64	100	
68	104	
6C	108	
70	112	
74	116	
78	120	
7C	124	

図 3-11 (1/6). 割り当て済みの記憶位置

16 進 数	10 進 数	フィールド			
80	128	外部割り込みパラメーター			
84	132	CPU アドレス		外部割り込みコード	
88	136	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ILC	0	SVC 割り込みコード
8C	140	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ILC	0	プログラム割り込みコード
90	144	データ例外コード			
94	148	モニター・クラス番号	PER Cde	ATMID	AI
98	152	PER アドレス			
9C	156				
A0	160	実行アクセス ID	PER アクセス ID	命令 Id	SS/MC Ar-Md Id
A4	164				
A8	168	変換例外識別			
AC	172				
B0	176	モニター・コード			
B4	180				
B8	184	サブシステム識別ワード			
BC	188	入出力割り込みパラメーター			
C0	192	入出力割り込み識別ワード			
C4	196				
C8	200	STFL 機能リスト			
CC	204				
D0	208				
D4	212				
D8	216				
DC	220				
E0	224				
E4	228				
E8	232	マシン・チェック割り込みコード			
EC	236				
F0	240				
F4	244	外部損傷コード			
F8	248	故障記憶機構アドレス			
FC	252				

図 3-11 (2/6). 割り当て済みの記憶位置

16 進 数	10 進 数	フィールド
100	256	
104	260	
108	264	
10C	268	
110	272	
114	276	
118	280	
11C	284	
120	288	
124	292	
128	296	
12C	300	
130	304	
134	308	
138	312	
13C	316	
140	320	
144	324	
148	328	
14C	332	
150	336	
154	340	
158	344	
15C	348	
160	352	
164	356	
168	360	
16C	364	
170	368	
174	372	
178	376	
17C	380	

図 3-11 (3/6). 割り当て済みの記憶位置

16 進 数	10 進 数	フィールド
180	384	
184	388	
188	392	
18C	396	
190	400	
194	404	
198	408	
19C	412	
1A0	416	
1A4	420	
1A8	424	
1AC	428	
1B0	432	
1B4	436	
1B8	440	
1BC	444	
1C0	448	
1C4	452	
1C8	456	
1CC	460	
1D0	464	
1D4	468	
1D8	472	
1DC	476	
1E0	480	
1E4	484	
1E8	488	
1EC	492	
1F0	496	
1F4	500	
1F8	504	
1FC	508	

図 3-11 (4/6). 割り当て済みの記憶位置

16 進 数	10 進 数	フィールド
1000	4096	(448 バイト)
1004	4100	
1008	4104	
100C	4108	
1010	4112	
1014	4116	
/	/	
11A8	4520	
11AC	4524	
11B0	4528	
11B4	4532	
11B8	4536	
11BC	4540	
11C0	4544	
11C4	4548	(64 バイト)
/	/	
11F8	4600	
11FC	4604	

図 3-11 (5/6). 割り当て済みの記憶位置

16 進 数	10 進 数	フィールド
1200	4608	状況記憶浮動小数点レジスター保管域、またはマシン・チェック 浮動小数点レジスター保管域 (128 バイト)
1204	4612	
1278	4728	
127C	4732	
1280	4736	状況記憶汎用レジスター保管域、またはマシン・チェック 汎用レジスター保管域 (128 バイト)
1284	4740	
12F8	4856	
12FC	4860	
1300	4864	状況記憶 PSW 保管域、または固定ログアウト域
1304	4868	
1308	4872	
130C	4876	
1310	4880	
1314	4884	
1318	4888	状況記憶プレフィックス保管域
131C	4892	状況記憶浮動小数点制御レジスター保管域、またはマシン・チェック 浮動小数点制御レジスター保管域
1320	4896	
1324	4900	状況記憶 TOD プログラマブル・レジスター保管域、またはマシン・ チェック TOD プログラマブル・レジスター保管域
1328	4904	状況記憶 CPU タイマー保管域、またはマシン・チェック CPU タイマー保管域
132C	4908	
1330	4912	状況記憶クロック・コンパレーター・ビット 0-55 保管域、またはマシン・チェック・クロック・ コンパレーター・ビット 0-55 保管域
1334	4916	
1338	4920	
133C	4924	
1340	4928	状況記憶アクセス・レジスター保管域、またはマシン・チェック・ チェック・アクセス・レジスター保管域 (64 バイト)
137C	4988	
1380	4992	状況記憶制御レジスター保管域、またはマシン・チェック制御 レジスター保管域 (128 バイト)
1384	4996	
13F8	5112	
13FC	5116	

図 3-11 (6/6). 割り当て済みの記憶位置

第4章 制御

停止、作動、ロード、およびチェック停止の各状態	4-1	TOD クロック	4-34
停止状態	4-2	フォーマット	4-34
作動状態	4-2	状態	4-34
ロード状態	4-2	クロック状態の変化	4-35
チェック停止状態	4-2	クロックの設定と検査	4-35
プログラム状況ワード	4-3	TOD プログラマブル・レジスター	4-36
プログラム状況ワードのフォーマット	4-5	TOD クロックの同期化	4-38
制御レジスター	4-7	クロック・コンパレータ	4-38
トレース	4-10	CPU タイマー	4-39
制御レジスターの割り振り	4-13	外部開始機能	4-40
トレース・エンタリー	4-13	リセット	4-40
操作	4-23	CPU リセット	4-43
プログラム・イベント記録	4-24	初期 CPU リセット	4-44
制御レジスター割り振りおよびアドレス・スペース制御エレメント	4-24	サブシステム・リセット	4-44
操作	4-25	クリア・リセット	4-45
原因の識別	4-26	パワーオン・リセット	4-45
標識の優先順位	4-28	初期プログラム・ローディング	4-46
記憶域の指示	4-29	状況記憶	4-46
PER イベント	4-30	マルチプロセッシング	4-47
ブランチ成功	4-30	共用主記憶機構	4-47
命令取り出し	4-31	CPU アドレスの識別	4-47
記憶域変更	4-31	CPU の信号発信と応答	4-48
実アドレスを使用した格納	4-31	プロセッサ信号指令	4-48
他の割り込み条件と同時発生		応答を決定する条件	4-51
の PER イベント		指令コードの解釈を妨げる条件	4-51
の指示	4-32	状況ビット	4-52
タイミング	4-33		

この章では、1 つ以上の CPU の動作を制御、測定、および記録するための各種機能について、詳しく説明します。

停止、作動、ロード、およびチェック停止の各状態

停止、作動、ロード、およびチェック停止は CPU の 4 つの状態です。各状態は互いに排他的です。CPU が停止状態にある場合は、命令および割り込み (再始動割り込み以外) は実行されません。作動状態のときは、CPU は、プログラム状況ワード (PSW) および制御レジスターの制御に従い、オペレーター機能速度制御の設定で指定された方法により、命令を実行し、割り込みを取り入れます。CPU がロード状態になるのは、ESA/390 の初期プログラム・ローディング操作の間です。CPU チェ

ック停止状態になるのは、マシンの誤動作が生じたときだけです。

この 4 つの CPU 状態の間の切り替えは、オペレーター機能を使用されたとき、または CPU に対する特定の SIGNAL PROCESSOR 指令が受け入れられたときに行われます。これらの状態は、PSW 中のビットにより制御または識別されるものではありません。停止状態、ロード状態、およびチェック停止状態は、それぞれ、手動標識ライト、ロード標識ライト、およびチェック停止標識ライトでオペレーターに示されます。CPU が作動状態にあるときは、この 3 つの標識ライトは消えています。

CPU タイマーは、CPU が作動状態またはロード状態にあるときに更新されます。TOD クロックは、どの CPU の状態にも影響されません。

停止状態

CPU は、停止機能により作動状態から停止状態に変更されます。停止機能が実行されるのは以下の場合です。

- CPU が作動状態にあるときに、停止キーが活性化された。
- 作動状態にある CPU が、その CPU に対する SIGNAL PROCESSOR 命令で指定されている停止指令または停止/状況記憶指令を受け入れた。
- 速度制御が命令ステップ位置に設定されているときに、開始機能の実行により開始された 1 単位の操作の実行を CPU が終了した。

停止機能が実行されると、現行操作単位が終了した時点で、作動状態から停止状態への移行が生じます。PSW の待ち状態ビットが 1 のときは、CPU が使用可能にされている割り込みの中に保留中のものがなければ、この移行はただちに行われます。割り込み可能命令の場合は、1 つの操作単位の中で処理されるデータの量は、個々の命令によって異なるほか、モデルによっても異なる場合があります。

すべての保留可能割り込みは、停止機能によって停止状態に入る前の、CPU がまだ作動状態にある間に起こります。これらの割り込みの結果、停止状態に入る前に、旧 PSW が格納され、新 PSW が取り出されます。CPU が停止状態にあるときは、割り込み条件は保留のまま残されます。

以下の場合も、CPU は停止状態になります。

- CPU のリセットが完了したとき。ただし、この CPU の初期プログラム・ローディングの一環としてリセット操作が行われた場合は、CPU はロード状態にされ、必ずしも停止状態にはなりません。
- アドレス比較の結果が一致を示しており、一致した場合に停止することが指定されているとき。

リセットの実行については、4-40ページの『リセット』、またアドレス比較については、12-1ページの『アドレス比較制御』で説明します。

CPU が停止状態にあるときに、構成内の他の CPU で INVALIDATE PAGE TABLE ENTRY 命令が実行された場合は、その停止中の CPU が停止状態から脱する前に、TLB エントリーの消去が完了します。

作動状態

CPU が停止状態から作動状態に変わるのは、開始機能が使用されたとき、または再始動割り込み (第6章、『割り込み』を参照) が起きたときです。

開始機能が実行されるのは、CPU が停止状態にあるときに、(1) その CPU に関連した開始キーが活性化されるか、または、(2) その CPU に対する SIGNAL PROCESSOR 命令に指定されている開始指令を、CPU が受け入れた場合です。リセットにより停止状態になっていた場合は、開始機能の実行結果は予測不能です。

速度制御が処理位置に設定されているときに開始機能が実行された場合は、CPU は通常で動作を開始します。速度制御が命令ステップ位置に設定されており、待ち状態ビットが 0 のときは、1 つの命令 (割り込み可能命令の場合は 1 つの操作単位) が実行され、すべての保留可能割り込みは、CPU が停止状態に戻る前に発生します。速度制御が命令ステップ位置に設定されており、待ち状態ビットが 1 のときは、開始機能により命令は実行されませんが、すべての保留可能割り込みは、CPU が停止状態に戻る前に発生します。

ロード状態

CPU がロード状態に入るのは、ロード通常キーまたはロード消去キーが活性化されたときです。(4-46ページの『初期プログラム・ローディング』を参照。17-13ページの『初期プログラム・ローディング』も参照)。これにより、アーキテクチャー・モードが ESA/390 モードに設定されます。参考のために、ESA/390 のロード状態の説明に対する追加要素を下に挙げておきます。

初期プログラム・ローディング操作が正常に完了したときに、速度制御が処理位置に設定されていれば、CPU はロード状態から作動状態に変わります。速度制御が命令ステップ位置に設定されている場合は、CPU はロード状態から停止状態に変わります。

チェック停止状態

チェック停止状態は、ある種のマシン誤動作が起きたときに生じる CPU の状態ですが、これについては第11章、『マシン・チェックの取り扱い』で説明します。CPU リセットが行われると、CPU はチェック停止状態から解放されます。

プログラミング上の注意:

1. 実行時刻と実際の時刻との関係を除けば、CPU の停止によりプログラムの実行が影響を受けることはありません。
2. マシンの誤動作が原因で、CPU が命令の実行を終了できない場合は、停止機能は働かないので、代わりにリセット機能呼び出す必要があります。早期に認識される種類の PSW フォーマット・エラー、または CPU タイマーに基づく割り込みのような永続的割り込み条件が原因で、終了しない割り込みストリングが生じる場合も、同様の状況が発生します。
3. CPU が停止状態に入った後は、保留中の入出力操作が開始され、アクティブな入出力操作は、中断のままにするかまたは完了するまで続行することができます。CPU が停止状態にあるときは、入出力操作の中断または完了に伴う割り込み条件は保留のままになります。

プログラム状況ワード

CPU 内の現プログラム状況ワード (PSW) には、現在アクティブなプログラムの実行に必要な情報が含まれています。PSW は長さが 128 ビットで、命令アドレス、条件コード、およびその他の制御フィールドを含んでいます。一般に、PSW は、命令の順序を制御し、現在実行中のプログラムとの関連の中での CPU の状況を保持お

よび表示するために使用されます。その他の制御情報および状況情報は、制御レジスターと永続割り当て済み記憶位置に入っています。

CPU の状況は、新しい PSW または PSW の一部をロードすることにより変更できます。

CPU の割り込み時に、CPU の状況を保存するために現 PSW を保管し、新 PSW をロードすることにより、制御が切り替えられます。

LOAD PSW または LOAD PSW EXTENDED が実行されるか、初期プログラム・ローディングのシーケンスが正常に完了すると、新 PSW が導入されます。命令アドレスは、順次の命令実行により更新され、成功したブランチにより置き換えられます。ほかにも、PSW の各部分を操作するための命令が用意されています。4-4ページの図4-1 は、これらの命令の要約を示しています。

割り込みが完了するか、または PSW を変更する命令の実行が完了すると、新規または変更された PSW がアクティブになります (つまり、現 PSW に取り込まれている情報に基づいて CPU が制御されるようになります)。PSW を変更する命令に関連した PER のための割り込みは、操作の開始時に有効であった PER マスクの制御下で発生します。

PSW のビット 0-7 を、まとめてシステム・マスクと呼びます。

命令	システム・マスク (PSW ビット 0-7)		PSW キー (PSW ビット 8-11)		問題プログラム 状態 (PSW ビット 15)		アドレス・スペース 制御 (PSW ビット 16-17)		条件 コードおよびプロ グラム・マスク (PSW ビット 18-23)		基本 アドレッシング・ モード (PSW ビット 32)		拡張 アドレッシング・ モード (PSW ビット 31)	
	保管	設定	保管	設定	保管	設定	保管	設定	保管	設定	保管	設定	保管	設定
BRANCH AND LINK	-	-	-	-	-	-	-	-	24AM	-	31AM	-	-	-
BRANCH AND SAVE	-	-	-	-	-	-	-	-	-	-	BAM	-	-	-
BRANCH AND SAVE AND SET MODE	-	-	-	-	-	-	-	-	-	-	BAM	Yes ¹	Yes	Yes ¹
BRANCH AND SET AUTHORITY	-	-	Yes	Yes	Yes	Yes	-	-	-	-	BAM ²	BAM	-	-
BRANCH AND SET MODE	-	-	-	-	-	-	-	-	-	-	BAM ¹	Yes ¹	Yes ¹	Yes ¹
BRANCH AND STACK	Yes	-	Yes	-	Yes	-	Yes	-	Yes	-	BAM ³	-	Yes	-
BRANCH IN SUBSPACE GROUP	-	-	-	-	-	-	-	-	-	-	BAM ¹	BAM	-	-
BRANCH RELATIVE AND SAVE	-	-	-	-	-	-	-	-	-	-	BAM	-	-	-
BRANCH RELATIVE AND SAVE LONG	-	-	-	-	-	-	-	-	-	-	BAM	-	-	-
EXTRACT PSW	Yes	-	Yes	-	Yes	-	Yes	-	Yes	-	Yes	-	Yes	-
INSERT PROGRAM MASK	-	-	-	-	-	-	-	-	Yes	-	-	-	-	-
INSERT PSW KEY	-	-	Yes	-	-	-	-	-	-	-	-	-	-	-
INSERT ADDRESS SPACE CONTROL	-	-	-	-	-	-	Yes	-	-	-	-	-	-	-
Basic PROGRAM CALL	-	-	-	-	Yes	Yes	-	-	-	-	BAM	BAM	-	-
Stacking PROGRAM CALL	Yes	-	Yes	PKC	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes
PROGRAM RETURN	-	Yes ⁴	-	Yes	-	Yes	-	Yes	-	Yes	-	Yes	BAM	-
PROGRAM TRANSFER	-	-	-	-	-	Yes ⁵	-	-	-	-	-	BAM	-	-
RESUME PROGRAM	-	-	-	-	-	-	-	Yes	-	Yes	-	Yes	-	Yes
SET ADDRESS SPACE CONTROL	-	-	-	-	-	-	-	Yes	-	-	-	-	-	-
SET ADDRESSING MODE	-	-	-	-	-	-	-	-	-	-	-	Yes	-	Yes
SET PROGRAM MASK	-	-	-	-	-	-	-	-	-	Yes	-	-	-	-
SET PSW KEY FROM ADDRESS	-	-	-	Yes	-	-	-	-	-	-	-	-	-	-
SET SYSTEM MASK	-	Yes	-	-	-	-	-	-	-	-	-	-	-	-
STORE THEN AND SYSTEM MASK	Yes	AND	-	-	-	-	-	-	-	-	-	-	-	-
STORE THEN OR SYSTEM MASK	Yes	OR	-	-	-	-	-	-	-	-	-	-	-	-
TRAP	-	-	-	-	Yes	-	Yes	Yes	Yes	-	Yes	Yes	Yes	-
説明 - No. 1 命令内の関連する R フィールドが 0 でない場合のみ、アクションが実行されます。 2 縮小権限状態では、命令内の R ₁ フィールドが 0 でない場合のみ、アクションが実行されます。 3 64 ビット・アドレッシング・モードでは、命令内の R ₁ フィールドが 0 である場合も、アクションが実行されます。 4 PROGRAM RETURN は PER マスクを変更しません。 5 PROGRAM TRANSFER は、問題プログラム状態を 1 から 0 に変更しません。														

図 4-1 (1/2). PSW のフィールド操作

説明 (続き)	
BAM	基本アドレッシング・モード・ビットは、24 ビットまたは 31 ビット・アドレッシング・モードで保管または設定されます。
AND	命令内の即値フィールドと現行システム・マスクの論理積が、現行システム・マスクと置き換わります。
OR	命令内の即値フィールドと現行システム・マスクの論理和が、現行システム・マスクと置き換わります。
PKC	PSW キー制御ビット (エントリー・テーブル・エントリーのビット 131) が 0 のときは、PSW キーは変更されません。PSW キー制御ビットが 1 のときは、PSW キーは、エントリー・キー (エントリー・テーブル・エントリーのビット 136-139) を使用して設定されます。
24AM	条件コードとプログラム・マスクは、24 ビット・アドレッシング・モードで保管されます。
31AM	基本アドレッシング・モード・ビットは、31 ビット・アドレッシング・モードで保管されます。

図 4-1 (2/2). PSW のフィールド操作

プログラミング上の注意: 5-10ページの『リンケージ・スタックなしのサブルーチン・リンケージ』に、問題プログラム状態、アドレッシング・モード、および命令アドレスを保管または設定する操作の要約を示してあります。

プログラム状況ワードのフォーマット

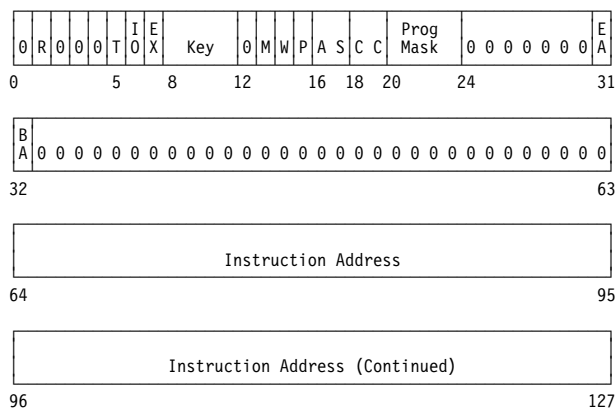


図 4-2. PSW のフォーマット

以下、PSW の各フィールドの働きについて簡単に説明します。(図4-2を参照。)

PER マスク (R): ビット 1 は、CPU がプログラム・イベント記録 (PER) に関連した割り込みを許可されるかどうかを制御します。このビットが 0 のときは、どの PER イベントも割り込みを生じさせることはできません。このビットが 1 のときは、制御レジスター 9 の PER イベント・マスク・ビットに従って、割り込みが許可されます。

DAT モード (T): ビット 5 は、記憶域のアクセスに使用される論理アドレスと命令アドレスの暗黙的な動的アドレス変換を行うかどうかを制御します。このビットが 0 のときは、DAT はオフであり、論理アドレスおよび命令アドレスは実アドレスとして扱われます。このビッ

トが 1 のときは、DAT はオンであり、動的アドレス変換メカニズムが呼び出されます。

入出力マスク (IO): ビット 6 は、CPU が入出力割り込みを許可されるかどうかを制御します。このビットが 0 のときは、入出力割り込みを起こすことはできません。このビットが 1 のときは、入出力割り込みが可能かどうかは、制御レジスター 6 の入出力割り込みサブクラス・マスク・ビットによって決まります。入出力割り込みサブクラス・マスク・ビットが 0 のときは、該当する入出力割り込みサブクラスの入出力割り込みを起こすことはできません。入出力割り込みサブクラス・マスクビットが 1 のときは、該当する入出力割り込みサブクラスの入出力割り込みを起こすことができます。

外部マスク (EX): ビット 7 は、外部クラスに含まれている条件によって、CPU が割り込みを許可されるかどうかを制御します。このビットが 0 のときは、外部割り込みを起こすことはできません。このビットが 1 のときは、外部割り込みが可能かどうかは、制御レジスター 0 の中の対応する外部サブクラス・マスク・ビットによって決まります。このサブクラス・マスク・ビットが 0 のときは、該当のサブクラスに関連した条件によって、割り込みを起こすことはできません。サブクラス・マスク・ビットが 1 のときは、該当のサブクラスの範囲内で割り込みを起こすことができます。

PSW キー: ビット 8-11 は、CPU が記憶域参照に使用するアクセス・キーです。参照がキー制御保護の対象となっている場合は、情報が格納される時、または、取り出し保護が指定されている記憶位置から情報が取り出される時に、PSW キーが記憶キーと突き合わせられます。ただし、MOVE TO PRIMARY、MOVE TO SECONDARY、MOVE WITH KEY、MOVE WITH SOURCE KEY、および MOVE WITH DESTINATION KEY のオペランドの 1 つについては、PSW キーの代わりに、オペランドとして指定されているアクセス・キーが使用されます。

マシン・チェック・マスク (M): ビット 13 は、CPU がマシン・チェック条件による割り込みを許可されるかどうかを制御します。このビットが 0 のときは、マシン・チェック割り込みを起こすことはできません。このビットが 1 のときは、システム損傷および命令処理損傷を原因とするマシン・チェック割り込みは許可されますが、その他のマシン・チェック・サブクラス条件を原因とする割り込みが許可されるかどうかは、制御レジスタ 14 のサブクラス・マスク・ビットによって決まります。

待ち状態 (W): ビット 14 が 1 のときは、CPU は待ち状態になっています。つまり、CPU による命令の処理は行われませんが、割り込みは生じることがあります。ビット 14 が 0 のときは、命令の取り出しと実行が通常どおりに行われます。このビットが 1 のときは、待ち状態ライトが点灯しています。

問題プログラム状態 (P): ビット 15 が 1 のときは、CPU は問題プログラム状態になっています。ビット 15 が 0 のときは、CPU は監視プログラム状態になっています。監視プログラム状態では、すべての命令が有効です。問題プログラム状態では、問題プログラムにとって意味のある情報を提供し、かつシステム保全性に影響を与えない命令のみが有効です。このような命令を非特権命令と呼びます。問題プログラム状態では有効にならない命令は、特権命令と呼ばれます。問題プログラム状態にある CPU が特権命令を実行しようとする時、特権命令例外が認識されます。もう 1 つの命令グループとして、準特権命令があります。これらの命令は、特定の権限テストの条件を満たした場合のみ、問題プログラム状態にある CPU により実行されます。さもないと、特権命令例外または特殊操作例外が認識されます。

アドレス・スペース制御 (AS): ビット 16 および 17 は、PSW ビット 5 との組み合わせで、変換モードを制御します。3-27ページの『変換モード』を参照してください。

条件コード (CC): ビット 18 および 19 は、2 ビットの条件コードです。条件コードは、一定の命令の実行結果に応じて、0、1、2、または 3 に設定されます。条件コードは、ほとんどの算術演算および論理演算、およびその他の一部の演算の結果として設定されます。

BRANCH ON CONDITION 命令では、条件コードの値を任意に選択して、ブランチの基準を指定できます。付録 C の表に、PSW の条件コードを設定するすべての命令について、設定できる条件コードの値を要約して示してあります。

プログラム・マスク: ビット 20-23 は、4 ビットのプログラム・マスクです。各ビットは、それぞれ次のようにプログラム例外に関連付けられています。

プログラム・マスク・ビット	プログラム例外
20	固定小数点オーバーフロー 10 進オーバーフロー HFP 指数アンダーフロー HFP 有効数字
21	
22	
23	

マスク・ビットが 1 のときは、例外が起きると割り込みが生じます。マスク・ビットが 0 のときは、割り込みは生じません。HFP 指数アンダーフロー・マスク・ビットまたは HFP 有効数字マスク・ビットは、対応する例外が起きたときに演算がどのように完了するかも決定します。

拡張アドレッシング・モード (EA): ビット 31 は、ビット 32 (基本アドレッシング・モード・ビット) との組み合わせにより、有効アドレスのサイズと有効アドレス生成を制御します。ビット 31 が 0 のときは、アドレッシング・モードはビット 32 により制御されます。ビット 31 および 32 が両方とも 1 のときは、64 ビット・アドレッシングが指定されます。

基本アドレッシング・モード (BA): ビット 31 および 32 は、有効アドレスのサイズと有効アドレス生成を制御します。ビット 31 と 32 が両方とも 0 のときは、24 ビット・アドレッシングが指定されます。ビット 31 が 0 で、ビット 32 が 1 のときは、31 ビット・アドレッシングが指定されます。ビット 31 および 32 が両方とも 1 のときは、64 ビット・アドレッシングが指定されます。ビット 31 が 1 でビット 32 が 0 の組み合わせは無効であり、指定例外が認識されます。PER アドレスのサイズ、または、DAT テーブル、ASN テーブル、デイスパッチ可能単位制御テーブル、リンケージ・テーブル、エントリー・テーブル、トレース・テーブル、アクセス・リスト、またはリンケージ・スタックにアクセスするために使用されるアドレスのサイズは、アドレッシング・モードでは制御されません。5-7ページの『アドレスの生成』および 3-5ページの『アドレスのサイズと循環』を参照してください。PSW のビット 31 および 32 によるアドレッシング・モードの制御を要約すると、以下のようになります。

PSW.31	PSW.32	アドレッシング・モード
0	0	24 ビット
0	1	31 ビット
1	1	64 ビット

命令アドレス: PSW のビット 64-127 は、命令アドレスです。このアドレスは、CPU が待ち状態 (PSW のビット 14 が 1) でないときに、次に実行すべき命令の左端のバイトの位置を示します。

ビット位置 0、2-4、24-30、および 33-63 は割り当てがなく、すべて 0 でなければなりません。これらのビット位置に 0 以外の値が入っていると、指定例外が認識されます。

PSW のビット 31 および 32 が 24 ビット・アドレッシング・モードを指定している場合は、命令アドレスの 64-103 はすべて 0 でなければならず、ビット 31 および 32 が 31 ビット・モードを指定している場合は、ビット 64-96 はすべて 0 でなければなりません。この条件が満たされていないと、指定例外が認識されます。また、ビット 31 が 1 でビット 32 が 0 の場合、またはビット位置 12 に 0 が含まれていない場合も、指定例外が認識されます。

LOAD PSW EXTENDED の第 2 オペランドは 16 バイトです。この命令は、このオペランドを現 PSW として検査することなく、未変更のままロードします。

LOAD PSW の第 2 オペランドは 8 バイトです。このオペランドは、ビット 31 (z/Architecture 拡張アドレッシング・モード・ビット) が 1 になることがある点以外は、ESA/390 PSW として扱われます。

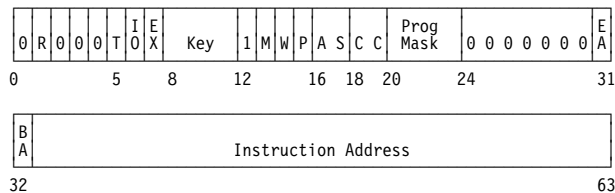


図 4-3. ESA/390 PSW のフォーマット (ただし、ビット 31 は EA と表記)

モデルにより、第 2 オペランドのビット 12 が 1 でないときに LOAD PSW が指定例外を認識する場合と、LOAD PSW の実行完了後に早期指定例外によりこのエラーが示される場合があります。LOAD PSW は、第 2 オペランドのビット 0-32 (ビット 12 の値を逆転したもの) およびビット 33-63 を、それぞれ現 PSW のビット 0-32 および 97-127 としてロードし、現 PSW のビット 33-96 を 0 に設定します。

制御レジスター

制御レジスターは、PSW の外部で制御情報を維持および操作するための手段です。64 ビットの制御レジスターが、16 個あります。

LOAD CONTROL (LCTLG) 命令では、この命令に指定されているレジスター内のすべての制御レジスター・ビット位置が、記憶域からロードされます。LOAD CONTROL (LCTL) 命令では、制御レジスターのビット位置 32-63 のみがロードされ、レジスターのビット 0-31 は未変更のまま残されます。BRANCH AND SET AUTHORITY、BRANCH IN SUBSPACE GROUP、LOAD ADDRESS SPACE PARAMETERS、SET SECONDARY ASN、BRANCH AND STACK、PROGRAM CALL、PROGRAM RETURN、および PROGRAM TRANSFER 命令は、情報をそれぞれ所定の制御レジスター・ビット位置に入れるための特殊な機能を備えています。

制御レジスターにロードされた情報は、そのロードの原因となった命令が完了した時点で、アクティブになります (つまりシステムの制御権を獲得します)。

レジスターに情報がロードされる時点では、使用不可または保護対象の記憶位置を指すアドレスの有無など、例外についての検査は行われません。情報の妥当性は、情報が実際に使用される時点で検査され、そのときに例外があれば示されます。

STORE CONTROL (STCTG) 命令では、この命令で指定されているレジスター内のすべての制御レジスター・ビット位置の内容が、記憶域に入れられます。STORE CONTROL (STCTL) 命令では、制御レジスターのビット位置 32-63 の内容が記憶域に入れられ、ビット 0-31 は無視されます。EXTRACT PRIMARY ASN、EXTRACT SECONDARY ASN、および PROGRAM CALL 命令は、所定の制御レジスター・ビット位置から情報を取得するための特殊な機能を備えています。

ここでは、制御レジスターの一般的な構造についてのみ説明します。個々の制御レジスター・ビット位置の定義については、それぞれに関連する機能に関する記述の中で説明されています。4-8 ページの図 4-4 は、割り当て済みの制御レジスター・ビット位置と、初期 CPU リセットの実行時点での各位置の初期値を示しています。図に示されていない制御レジスター・ビット位置は、すべて 0 に初期化されます。

プログラミング上の注意:

1. 個々の制御レジスター・ビット位置の詳しい定義は、索引の中の「制御レジスター割り当て」の項目を参照して見つけることができます。
2. これまで使用されていなかった制御レジスター・ビット位置を使用する新しい機能がインストールされ

た場合にも、既存のプログラムが正しく稼働できるようにするためには、各プログラムが未割り当ての位置に 0 をロードするようにしてください。

制御レジスター	ビット	フィールドの名前	関連の機能など	初期値
0	33	SSM 抑止制御	SET SYSTEM MASK	0
0	34	TOD クロック同期制御	TOD クロック	0
0	35	低アドレス保護制御	低アドレス保護	0
0	36	抽出権限制御	命令許可	0
0	37	2 次スペース制御	命令許可	0
0	38	取り出し保護オーバーライド制御	キー制御保護	0
0	39	記憶保護オーバーライド制御	キー制御保護	0
0	45	AFP レジスター制御	浮動小数点	0
0	48	誤動作警報サブクラス・マスク	外部割込み	0
0	49	緊急信号サブクラス・マスク	外部割込み	0
0	50	外部呼び出しサブクラス・マスク	外部割込み	0
0	52	クロック・コンパレーター・サブクラス・マスク	外部割込み	0
0	53	CPU タイマー・サブクラス・マスク	外部割込み	0
0	53	サービス信号サブクラス・マスク	外部割込み	0
0	54	未使用 ¹	外部割込み	1
0	57	割り込みキー・サブクラス・マスク	外部割込み	1
0	58	未使用 ¹	外部割込み	1
0	59	ETR サブクラス・マスク	外部割込み	0
0	61	暗号制御	暗号化	0
1	0-51	1 次領域テーブル起点 ²	動的アドレス変換	0
1	0-51	1 次セグメント・テーブル起点 ²	動的アドレス変換	0
1	0-51	1 次実スペース・トークン起点 ²	動的アドレス変換	0
1	54	1 次サブスペース・グループ制御	サブスペース・グループ	0
1	55	1 次専用スペース制御	動的アドレス変換	0
1	56	1 次記憶域変更イベント制御	プログラム・イベント記録	0
1	57	1 次スペース切り替えイベント制御	プログラム割り込み	0
1	58	1 次実スペース制御	動的アドレス変換	0
1	60-61	1 次指定タイプ制御 ³	動的アドレス変換	0
1	62-63	1 次テーブル長 ³	動的アドレス変換	0
2	33-57	ディスパッチ可能単位制御テーブル	アクセス・レジスター変換	0
3	32-47	PSW キー・マスク	命令許可	0
3	48-63	2 次 ASN	アドレス・スペース	0
4	32-47	許可指標	命令許可	0
4	48-63	1 次 ASN	アドレス・スペース	0

図 4-4 (1/3). 制御レジスターのフィールドの割り当て

制御レジスタ	ビット	フィールドの名前	関連の機能など	初期値
5	33-57	1 次 ASN 第 2 テーブル・エントリー 起点	アクセス・レジスタ変換	0 0
6	32-39	入出力割込みサブクラス・マスク	入出力割り込み	0
7	0-51	2 次セグメント・テーブル起点 ²	動的アドレス変換	0
7	0-51	2 次領域テーブル起点 ²	動的アドレス変換	0
7	0-51	2 次実スペース・トークン起点 ²	動的アドレス変換	0
7	54	2 次サブスペース・グループ制御	サブスペース・グループ	0
7	55	2 次専用スペース制御	動的アドレス変換	0
7	56	2 次記憶域変更イベント制御	プログラム・イベント記録	0
7	58	2 次実スペース制御	動的アドレス変換	0
7	60-61	2 次指定タイプ制御 ³	動的アドレス変換	0
7	62-63	2 次テーブル長 ³	動的アドレス変換	0
8	32-47	拡張許可索引	アクセス・レジスタ変換	0
8	48-63	モニター・マスク	MONITOR CALL	0
9	32	成功ブランチ・イベント・マスク	プログラム・イベント記録	0
9	33	命令取り出しイベント・マスク	プログラム・イベント記録	0
9	34	記憶域変更イベント・マスク	プログラム・イベント記録	0
9	36	実アドレスを使用した格納イベント・	プログラム・イベント記録	0
9	36	マスク		0
9	40	ブランチ・アドレス制御	プログラム・イベント記録	0
9	42	記憶域変更スペース制御	プログラム・イベント記録	0
10	0-63	PER 開始アドレス	プログラム・イベント記録	0
11	0-63	PER 終了アドレス	プログラム・イベント記録	0
12	0	ブランチ・トレース制御	トレース	0
12	1	モード・トレース制御	トレース	0
12	2-61	トレース・エントリー・アドレス	トレース	0
12	62	ASN トレース制御	トレース	0
12	63	明示トレース制御	トレース	0
13	0-51	ホーム・セグメント・テーブル起点 ²	動的アドレス変換	0
13	0-51	ホーム領域テーブル起点 ²	動的アドレス変換	0
13	0-51	ホーム実スペース・トークン起点 ²	動的アドレス変換	0
13	55	ホーム専用スペース制御	動的アドレス変換	0
13	56	ホーム記憶域変更イベント制御	プログラム・イベント記録	0
13	57	ホーム・スペース切り替えイベント 制御	プログラム割り込み	0
13	58	ホーム実スペース制御	動的アドレス変換	0
13	60-61	ホーム指定タイプ制御 ³	動的アドレス変換	0
13	62-63	ホーム・テーブル長 ³	動的アドレス変換	0

図 4-4 (2/3). 制御レジスタのフィールドの割り当て

制御レジスタ	ビット	フィールドの名前	関連の機能など	初期値
14	32	未使用 ¹		1
14	33	未使用 ¹		1
14	35	チャンネル・レポート保留サブクラス・マスク	入出力マシン・チェック取り扱い	0
14	36	リカバリー・サブクラス・マスク	マシン・チェック取り扱い	0
14	37	性能低下サブクラス・マスク	マシン・チェック取り扱い	0
14	38	外部損傷サブクラス・マスク	マシン・チェック取り扱い	1
14	39	警告サブクラス・マスク	マシン・チェック取り扱い	0
14	42	TOD クロック制御オーバーライド制御	TOD クロック	0
14	44	ASN 変換制御	命令許可	0
14	45-63	ASN 第 1 テーブル起点	ASN 変換	0
15	0-60	リンケージ・スタック・エントリー・アドレス	リンケージ・スタック操作	0

説明:

表にリストされていないフィールドは未割り当てです。リストされていない制御レジスタ・ビット位置の初期値は、すべて 0 です。

- このビットは使用されませんが、システム/370 の定義との整合性を維持するために 1 に初期化されます。
- 制御レジスタ内のアドレス・スペース制御エレメント (ASCE) は、レジスタ内のビット 58 (実スペース制御) およびレジスタのビット 60 および 61 (指定タイプ制御) に応じて、3 つのフォーマットのいずれかをとります。ビット 58 が 0 のときは、ビット 60 および 61 が 2 進数 11、10、または 01 であれば ASCE は領域テーブル指定であり、ビット 60 および 61 が 2 進数 00 であれば ASCE はセグメント・テーブル指定です。ビット 58 が 1 のときは、ASCE は実スペース指定です。ビット 0-51 は、ASCE が領域テーブル指定か、セグメント・テーブル指定か、または実スペース指定かに応じて、それぞれ、領域テーブル起点、セグメント・テーブル起点、または実スペース・トークン起点になります。
- ビット 60-63 が割り当てられるのは、制御レジスタ内の ASCE が領域テーブル指定またはセグメント・テーブル指定のときです。

図 4-4 (3/3). 制御レジスタのフィールドの割り当て

トレース

トレース機能は、重要イベントに関する継続的な記録を記憶機構内に維持します。これはシステムの問題を判別するときに役に立ちます。トレースは、ブランチ・トレース、ASN トレース、モード・トレース、および明示トレースという、個別に制御できる 4 つの機能からなっています。これらの機能は、トレース・テーブル内にエントリーを作成します。ブランチ・トレース、ASN トレース、およびモード・トレースは、総称的に暗黙トレースと呼ばれます。

ブランチ・トレースがオンのときは、ブランチを生じさせる特定のブランチ命令が実行されるたびに、トレース・テーブル内にブランチ・トレース・エントリーが作成されます。このトレース・エントリーにはブランチ・アドレスが入っています。また、トレース・エントリーは、そのブランチ・アドレスと、ブランチの後で有効になっているアドレッシング・モードについて、以下の事

項も示します。(1) CPU は 24 アドレッシング・モードになっている、(2) CPU は 31 ビット・アドレッシング・モードまたは 64 ビット・アドレッシング・モードであり、ブランチ・アドレスのビット 0-32 はすべて 0 である、または、(3) CPU は、64 ビット・アドレッシング・モードであり、ブランチ・アドレスのビット 0-32 は「すべて 0」ではありません。トレースされるブランチ命令には以下のものがあります。

- BRANCH AND LINK (BALR のみ): R₂ フィールドが 0 ではない場合。
- BRANCH AND SAVE (BASR のみ): R₂ フィールドが 0 ではない場合。
- BRANCH AND SAVE AND SET MODE: R₂ フィールドが 0 ではない場合。
- BRANCH AND SET AUTHORITY
- BRANCH AND STACK: R₂ フィールドが 0 ではない場合。

- BRANCH IN SUBSPACE GROUP
- RESUME PROGRAM
- TRAP

ただし、BRANCH IN SUBSPACE GROUP についてのブランチ・トレース・エントリーが作成されるのは、ASN トレースがオンになっていない場合のみです。

ブランチ・トレースとモード・トレースの両方がオンのときに、BRANCH AND SAVE AND SET MODE または RESUME PROGRAM が拡張アドレッシング・モード・ビット (PSW のビット 31) を変更した場合は、ブランチ・トレース・エントリーの代わりにモード切り替えブランチ・トレース・エントリーが作成されます。

ASN トレースがオンのときは、以下のいずれかの命令が実行されるたびに、命令と同じ名前のエントリーがトレース・テーブル内に作成されます。

- BRANCH IN SUBSPACE GROUP
- PROGRAM CALL
- PROGRAM RETURN
- PROGRAM TRANSFER
- SET SECONDARY ASN

ただし、PROGRAM RETURN についてのエントリーが作成されるのは、PROGRAM RETURN が PROGRAM CALL により形成されたリンケージ・スタック状態エントリーをアンスタックするときのみであり、PROGRAM RETURN が BRANCH AND STACK により形成されたエントリーをアンスタックするときには作成されません。

ASN トレースとモード・トレースが両方ともオンのときに、PROGRAM CALL が PSW のビット 31 を変更した場合は、まず PROGRAM CALL のトレース・エントリーが作成され、次にモード切り替えトレース・エントリーが作成されます。

モード・トレースは、基本アドレッシング・モード (24 ビットまたは 31 ビット) から拡張アドレッシング・モード (64 ビット) に、または拡張モードから基本モードへの切り替えを記録します。

モード・トレースがオンのときは、以下の命令のいずれかの実行により PSW のビット 31 が変更されるたびに、モード切り替えトレース・エントリーがトレース・テーブル内に作成されます。

- BRANCH AND SAVE AND SET MODE
- BRANCH AND SET MODE
- PROGRAM CALL
- PROGRAM RETURN
- RESUME PROGRAM
- SET ADDRESSING MODE

ただし、ASN トレースがオンのときに、PROGRAM RETURN が PROGRAM CALL により形成された状態エントリーをアンスタックする場合は、PROGRAM RETURN についてのモード切り替えトレース・エントリーは作成されません。代わりに PROGRAM RETURN トレース・エントリーが作成され、そのエントリーには PSW のビット 31 に関する情報が入れられます。

BRANCH AND SAVE AND SET MODE および RESUME PROGRAM については、ブランチ・トレースのみオンの場合はブランチ・トレース・エントリー、ブランチ・トレースとモード・トレースの両方がオンの場合はモード切り替えブランチ・トレース・エントリー、そして、モード・トレースのみがオンの場合はモード切り替えトレース・エントリーが作成されます。

4-12ページの図4-5は、暗黙トレースにより作成されるトレース・エントリーの要約を示しています。

明示トレースがオンのときは、TRACE (TRACE または TRACG) が実行されると、トレース・テーブル内にエントリーが作成されます。TRACE (TRACE) の場合のエントリーには、TOD クロックのビット 16-63、TRACE 命令の第 2 オペランド、および一連の汎用レジスターのビット 32-63 が含まれます。TRACE (TRACG) の場合のエントリーもほぼ同じですが、TOD クロックの 0-79、および一連の汎用レジスターのビット 0-63 が含まれる点が異なります。

命令	使用可能にされる暗黙トレース						
	ブランチ	ASN	モード	ブランチ および ASN	ブランチ および モード	ASN および モード	すべて
	トレース・エントリー・モード						
BAKR	B	-	-	B	B	-	B
BALR	B	-	-	B	B	-	B
BASR	B	-	-	B	B	-	B
BASSM	B	-	MS	B	B MSB	MS	B MSB
BSA	B	-	-	B	B	-	B
BSG	B	BSG	-	BSG	B	BSG	BSG
BSM	-	-	MS	-	MS	-	MS
PC	-	PC	MS	PC	MS	PC & MS	PC & MS
PR-b	-	-	MS	-	MS	MS	MS
PR-pc	-	PR	MS	PR	MS	PR	PR
PT	-	PT	-	PT	-	PT	PT
RP	B	-	MS	B	B MSB	MS	B MSB
SASN	-	SASN	-	SASN	-	SASN	SASN
SAM24/ 31/64	-	-	MS	-	MS	MS	MS
TRAP2/4	B	-	-	B	B	-	B

説明:

- なし。
- b PROGRAM RETURN がブランチ状態エントリーをアンスタックする場合。
- pc PROGRAM RETURN がプログラム呼び出し状態エントリーをアンスタックする場合。
- | または。
- & および。
- B ブランチ・トレース・エントリー。これが作成されるのは、ブランチが実行され、モード切り替えブランチ・トレース・エントリーが作成されない場合のみです。
- MS モード切り替えトレース・エントリー。これが作成されるのは、PSW のビット 31 が変更された場合のみです。
- MSB モード切り替えブランチ・トレース・エントリー。これが作成されるのは、PSW のビット 31 が変更された場合のみです (ビット 31 が変更されるのはブランチが実行された場合のみです)。

図 4-5. 暗黙トレースの要約

制御レジスタの割り振り

トレースを制御する情報は、以下のフォーマットで制御レジスタ 12 に含まれています。

B	M	Trace-Entry Address	A	E
0	1	2	62	63

ブランチ・トレース制御ビット (B): 制御レジスタ 12 のビット 0 は、ブランチ・トレースがオンかオフかを制御します。このビットが 0 のときは、ブランチ・トレースはオフであり、このビットが 1 のときは、ブランチ・トレースはオンです。

モード・トレース制御ビット (M): 制御レジスタ 12 のビット 1 は、モード・トレースがオンかオフかを制御します。このビットが 0 のときは、モード・トレースはオフであり、このビットが 1 のときは、モード・トレースはオンです。

トレース・エン트리・アドレス: 制御レジスタ 12 のビット 2-61 は、左側に 2 個、右側に 2 個の 0 ビットを付加されて、次に作成されるトレース・エントリーの実アドレスを形成します。

ASN トレース制御ビット (A): 制御レジスタ 12 のビット 62 は、ASN トレースがオンかオフかを制御します。このビットが 0 のときは、ASN トレースはオフであり、このビットが 1 のときは、ASN トレースはオンです。

明示トレース制御ビット (E): 制御レジスタ 12 のビット 63 は、明示トレースがオンかオフかを制御します。このビットが 0 の場合は、明示トレースはオフであり、したがって、TRACE 命令はノーオペレーションとして実行されます。このビットが 1 の場合は、TRACE 命令を実行すると、トレース・テーブル内にエントリー

が作成されます。ただし、TRACE 命令の第 2 オペランドのビット 0 が 1 のときは、エントリーは作成されません。

トレース・エントリー

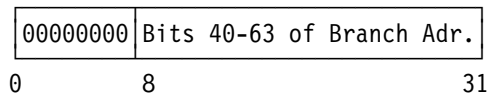
トレース・エントリーには 9 つのタイプがあり、ほとんどのタイプにはそれぞれ複数の詳細フォーマットがあります。各タイプおよびフォーマット数は以下のとおりです。

- ブランチ (3 フォーマット)
- BRANCH IN SUBSPACE GROUP (2 フォーマット)
- モード切り替え (3 フォーマット)
- モード切り替えブランチ (3 フォーマット)
- PROGRAM CALL (2 フォーマット)
- PROGRAM RETURN (9 フォーマット)
- PROGRAM TRANSFER (3 フォーマット)
- SET SECONDARY ASN (1 フォーマット)
- TRACE (2 フォーマット)

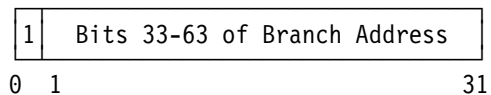
4-14ページの図4-6に、これらのエントリーを示してあります。図の中の各エントリーには「Fn」というラベルが付いていますが、これは、1つのトレース・エントリー・タイプの中の各フォーマットを参照するときに使用するフォーマット番号です。また、「ブランチ」は、ブランチ・トレース・エントリーを生じさせる命令のニーマニットの総称を表すもので、これは、BAKR、BALR、BASR、BASSM、BSA、またはBSGのいずれかを指します。

4-20ページの図4-7には、各トレース・エントリーを識別するビット・フィールドの値の昇順に従って、トレース・エントリーをリストしてあります。

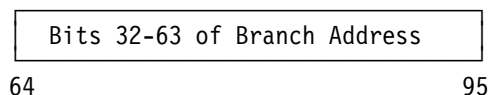
F1 ブランチ (結果のモードが 24 ビットである場合のブランチ、RP、または TRAP2/4)



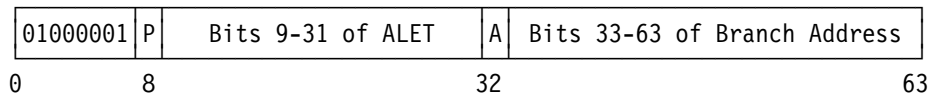
F2 ブランチ (結果のモードが 31 ビットの場合、または、結果の PSW ビット 31 が 1 で (注を参照)、ブランチ・アドレスのビット 0-32 がすべて 0 の場合の、ブランチ、RP、または TRAP2/4)



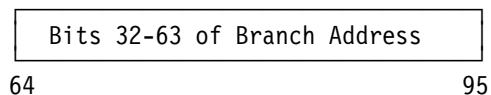
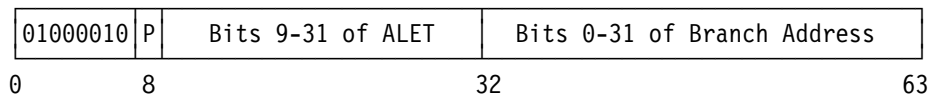
F3 ブランチ (結果の PSW ビット 31 が 1 で (注を参照)、ブランチ・アドレスのビット 0-32 がすべて 0 ではない場合の、ブランチ、RP、または TRAP2/4)



F1 BRANCH IN SUBSPACE GROUP (ASN トレースがオンで、24 ビットまたは 31 ビット・モードの場合)



F2 BRANCH IN SUBSPACE GROUP (ASN トレースがオンで、64 ビット・モードの場合)



F1 モード切り替え (結果の PSW ビット 31 が 1 (注を参照) の場合の、24 ビットまたは 31 ビット・モードからの BASSM、BSM、PC、PR、RP、または SAM64)

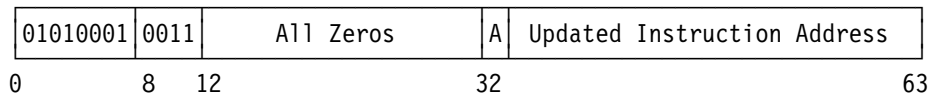
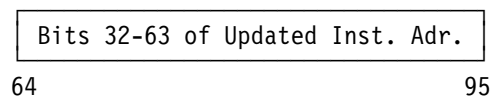


図 4-6 (1/7). トレース・エントリー

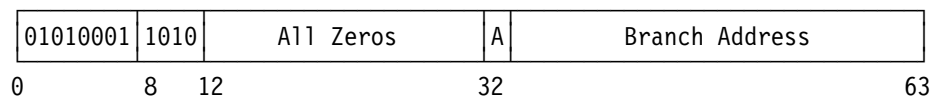
F2 モード切り替え (更新済み命令アドレスのビット 0-31 がすべて 0 の場合の、64 ビット・モードから 24 ビットまたは 31 ビット・モードへの、BASSM、BSM、PC、PR、RP、SAM24、または SAM31)



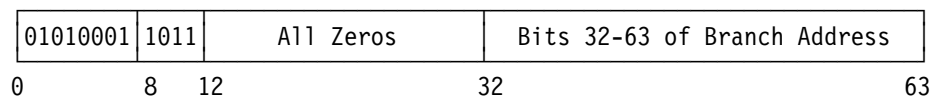
F3 モード切り替え (更新済み命令アドレスのビット 0-31 がすべて 0 ではない場合の、64 ビット・モードから 24 ビットまたは 31 ビット・モードへの、BASSM、BSM、PC、PR、RP、SAM24、または SAM31)



F1 モード切り替えブランチ (64 ビット・モードから 24 ビットまたは 31 ビット・モードへの、BASSM または RP)



F2 モード切り替えブランチ (結果の PSW ビット 31 が 1 で (注を参照)、ブランチ・アドレスのビット 0-31 がすべて 0 である場合の、24 ビットまたは 31 ビット・モードからの、BASSM または RP)



F3 モード切り替えブランチ (結果の PSW ビット 31 が 1 で (注を参照)、ブランチ・アドレスのビット 0-31 がすべて 0 ではない場合の、24 ビットまたは 31 ビット・モードからの、BASSM または RP)

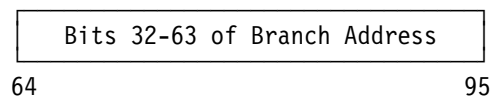
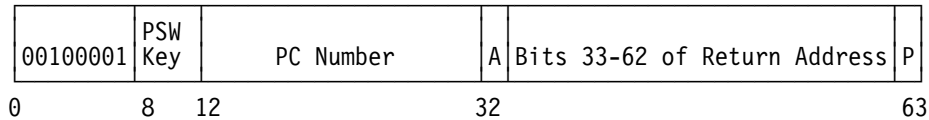
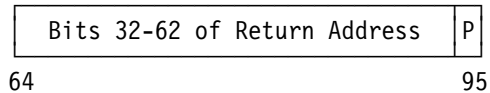
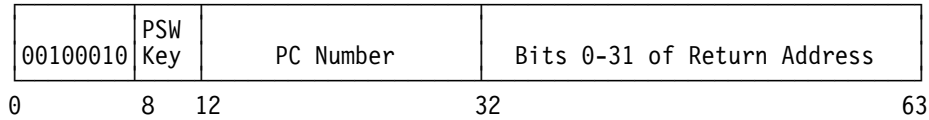


図 4-6 (2/7). トレース・エントリー

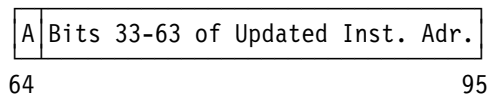
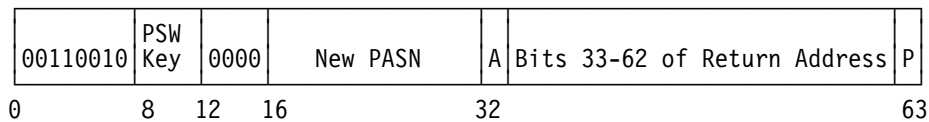
F1 PROGRAM CALL (結果のモードに関係なく、24 ビットまたは 31 ビット・モードのとき)



F2 PROGRAM CALL (結果のモードに関係なく、64 ビット・モードのとき)



F1 PROGRAM RETURN (結果のモードが 24 ビットまたは 31 ビットである場合の 24 ビットまたは 31 ビット・モード)



F2 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 で、結果のモードが 24 ビットまたは 31 ビットである場合の、64 ビット・モードのとき)

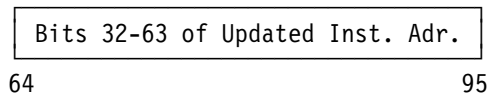
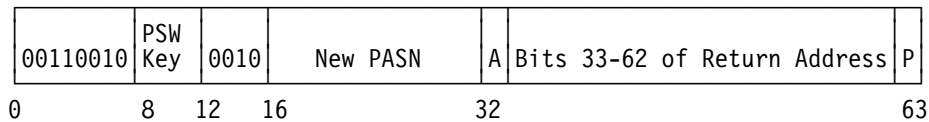
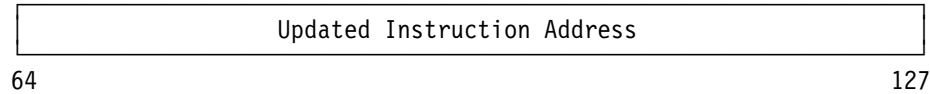
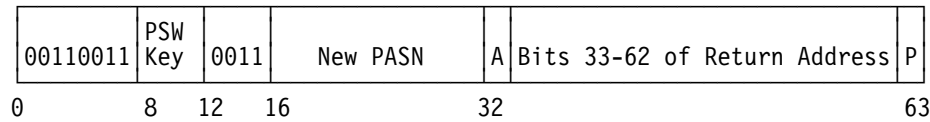
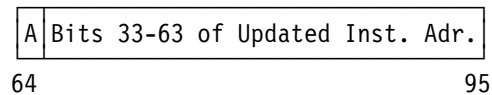
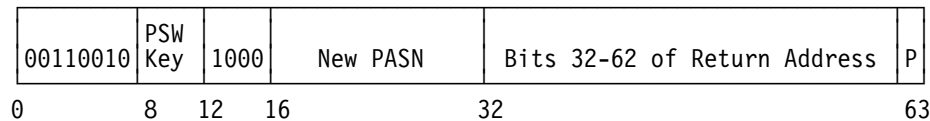


図 4-6 (3/7). トレース・エントリー

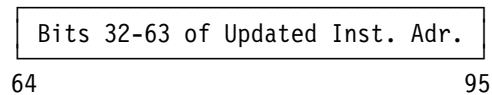
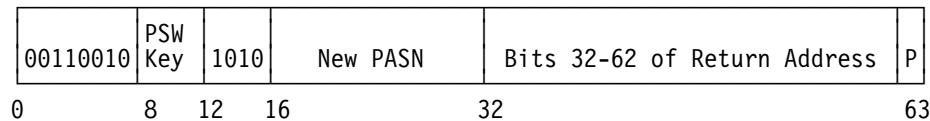
F3 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 ではなく、結果のモードが 24 ビットまたは 31 ビットである場合の、64 ビット・モードのとき)



F4 PROGRAM RETURN (結果の PSW ビット 31 が 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 である場合の、24 ビットまたは 31 ビット・モードのとき)



F5 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 で、結果の PSW ビット 31 が 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 である場合の、64 ビットモードのとき)



F6 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 ではなく、結果の PSW ビット 31 が 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 である場合の、64 ビット・モードのとき)

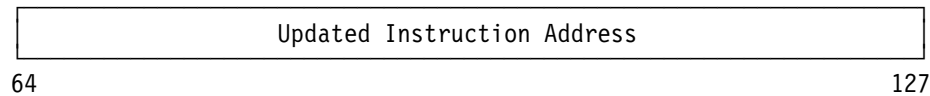
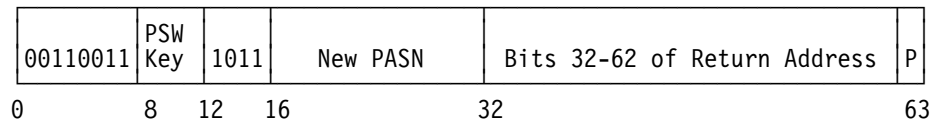
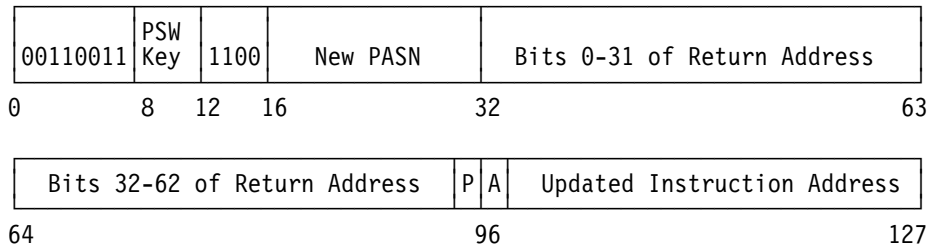
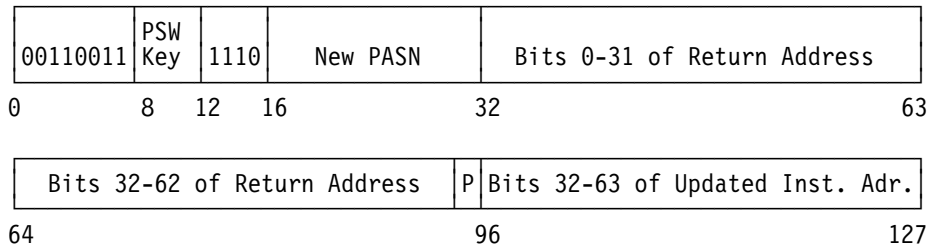


図 4-6 (4/7). トレース・エントリー

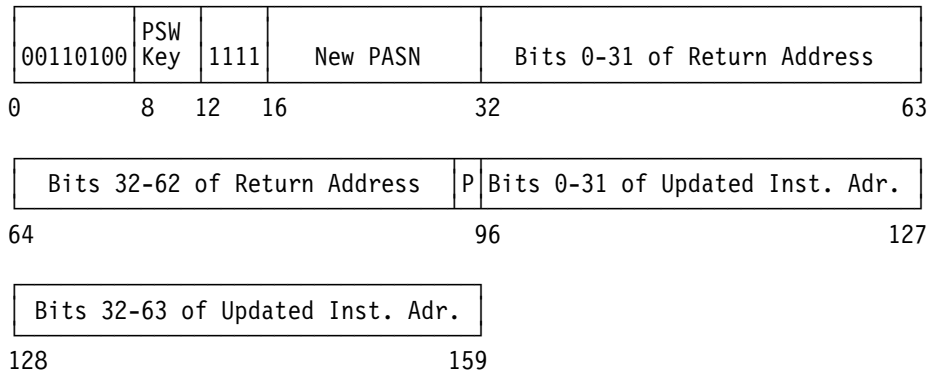
F7 PROGRAM RETURN (結果の PSW ビット 31 が 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 ではない場合の、24 ビットまたは 31 ビット・モードのとき)



F8 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 で、結果の PSW ビットが 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 ではない場合の、64 ビット・モードのとき)



F9 PROGRAM RETURN (更新済み命令アドレスのビット 0-31 がすべて 0 ではなく、結果の PSW ビット 31 が 1 で (注を参照)、戻りアドレスのビット 0-31 がすべて 0 ではない場合の、64 ビット・モードのとき)



F1 PROGRAM TRANSFER (24 ビットまたは 31 ビット・モードのとき)

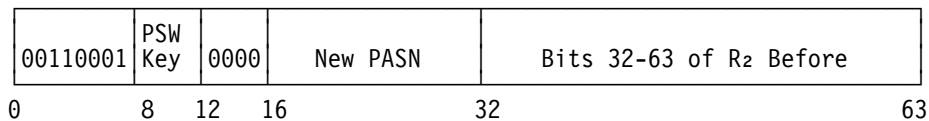
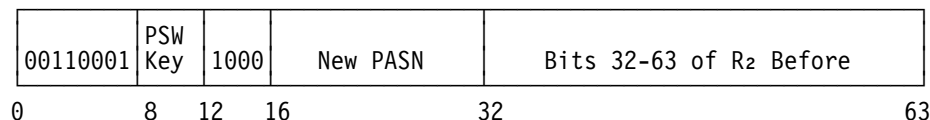
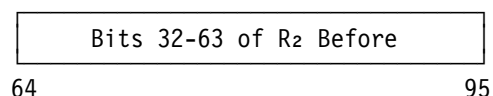
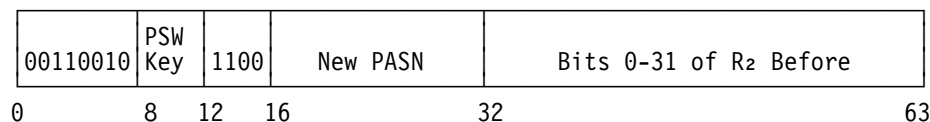


図 4-6 (5/7). トレース・エントリー

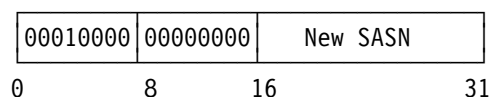
F2 PROGRAM TRANSFER (R₂ のビット 0-31 がすべて 0 の場合の、64 ビット・モードのとき)



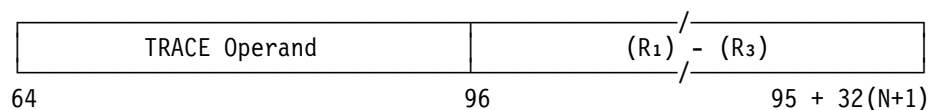
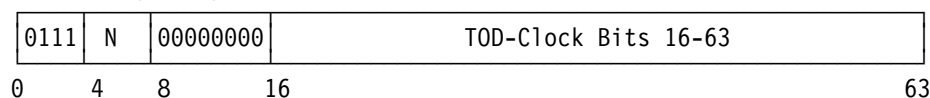
F3 PROGRAM TRANSFER (R₂ のビット 0-31 がすべて 0 ではない場合の、64 ビット・モードのとき)



F1 SET SECONDARY ASN



F1 TRACE (TRACE)



F2 TRACE (TRACG)

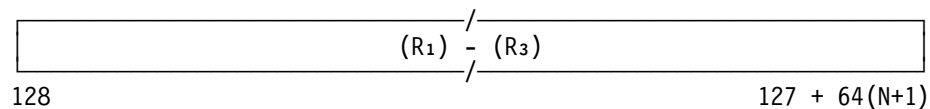
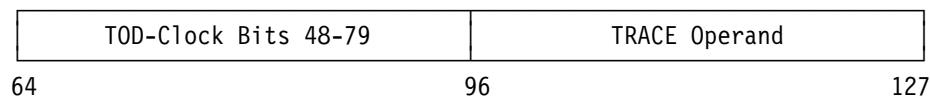
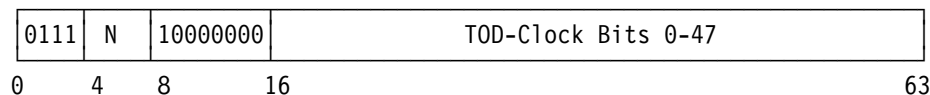


図 4-6 (6/7). トレース・エントリー

注: 結果の PSW ビット 32 が 0 の場合は、早期指定例外が認識されることになるので、「結果のモードが 64 ビットのとき」の代わりに「結果の PSW ビット 31 が 1 のとき」という表現が使用されています。PROGRAM RETURN は、PSW ビット 31 を 1 に、そしてビット 32 を 0 に設定できます。

図 4-6 (7/7). トレース・エントリー

トレース・エントリー・ビット			トレース・エントリー	
0-7	8-11	12-15	タイプ	フォーマット
00000000			ブランチ	1
00010000			SET SECONDARY ASN	1
00100001			PROGRAM CALL	1
00100010			PROGRAM CALL	2
00110001		0000	PROGRAM TRANSFER	1
00110001		1000	PROGRAM TRANSFER	2
00110010		0000	PROGRAM RETURN	1
00110010		0010	PROGRAM RETURN	2
00110010		1000	PROGRAM RETURN	4
00110010		1010	PROGRAM RETURN	5
00110010		1100	PROGRAM TRANSFER	3
00110011		0011	PROGRAM RETURN	3
00110011		1011	PROGRAM RETURN	6
00110011		1100	PROGRAM RETURN	7
00110011		1110	PROGRAM RETURN	8
00110100		1111	PROGRAM RETURN	9
01000001			BRANCH IN SUBSPACE GROUP	1
01000010			BRANCH IN SUBSPACE GROUP	2
01010001	0010		モード切り換え	2
01010001	0011		モード切り換え	1
01010001	1010		モード切り換えブランチ	1
01010001	1011		モード切り換えブランチ	2
01010010	0110		モード切り換え	3
01010010	1100		ブランチ	3
01010010	1111		モード切り換えブランチ	3
0111	0		TRACE	1
0111	1		TRACE	2
1			ブランチ	2

図 4-7. 識別ビットに従って配列したトレース・エントリー

トレース・エントリーの中のフィールドの定義は、以下のとおりです。フィールドは、4-14ページの図4-6の中での初出順に列挙してあります。

ブランチ・アドレス: ブランチ・アドレスは、ブランチが実行されたときに、次に実行する命令のアドレスです。ブランチの後で 24 ビット・アドレッシング・モードが有効になっているときに作成されるブランチ・トレース・エントリー (フォーマット 1 エントリー) では、ビット 8-31 にブランチ・アドレスのビット 40-63 が含まれます。ブランチの後で、31 ビット・アドレッシング・モードが有効になっているか、またはブランチの後で PSW ビット 31 が 1 であり、ブランチ・アドレスのビット 0-32 がすべて 0 である場合は、トレース・エントリー (フォーマット 2) のビット位置 1-31 に、ブランチ・アドレスのビット 33-63 が含まれます。ブランチの後で PSW のビット 31 が 1 であり、ブランチ・アドレスのビット 0-32 がすべて 0 ではない場合は、トレース・

エントリー (フォーマット 3) のビット位置 32-95 に、ブランチ・アドレスのビット 0-63 が含まれます。

24 ビットまたは 31 ビット・アドレッシング・モードでの実行時に作成される BRANCH IN SUBSPACE GROUP トレース・エントリー (フォーマット 1) では、ビット位置 33-63 にブランチ・アドレスのビット 33-63 が含まれます。また、64 ビット・アドレッシング・モードでは、トレース・エントリー (フォーマット 2) のビット位置 32-95 にブランチ・アドレスのビット 0-63 が含まれます。

64 ビット・アドレッシング・モードから 24 ビットまたは 31 ビット・アドレッシング・モードへの切り替えのときに作成されるモード切り替えブランチ・トレース・エントリー (フォーマット 1) では、ビット位置 33-63 にブランチ・アドレスのビット 33-63 が含まれます。また、PSW のビット 31 をオフからオンに切り替えるときには、ブランチ・アドレスのビット 0-31 がすべて 0

であれば、エンタリー (フォーマット 2) のビット位置 32-63 にブランチ・アドレスのビット 32-63 が含まれ、ブランチ・アドレスのビット 0-31 がすべて 0 ではない場合は、エンタリー (フォーマット 3) のビット 32-95 にブランチ・アドレスのビット 0-63 が含まれます。

ALET の 1 次リスト・ビット (P) およびビット

9-31: BRANCH IN SUBSPACE GROUP トレース・エンタリーのビット 8 には、この命令の R₂ フィールドに指定されているアクセス・レジスタ内のアクセス・リスト・エンタリー・トークン (ALET) のビット 7 が含まれます。トレース・エンタリーのビット 9-31 には、ALET のビット 9-31 が含まれます。

基本アドレッシング・モード・ビット (A): 24 ビットまたは 31 ビット・アドレッシング・モードでの実行時に作成される BRANCH IN SUBSPACE GROUP トレース・エンタリー (フォーマット 1 エンタリー) のビット位置 32 には、PSW のビット 32 と置き換わる基本アドレッシング・モード・ビットが含まれます。

PSW ビット 31 のオフからオンへの切り替えを示すモード切り替えトレース・エンタリー (フォーマット 1 エンタリー) のビット位置 32 には、モード切り替え操作の前に存在していた PSW ビット 32 の値が含まれます。

64 ビット・アドレッシング・モードから 24 ビットまたは 31 ビット・アドレッシング・モードへの切り替えを示すモード切り替えブランチ・トレース・エンタリー (フォーマット 1 エンタリー) のビット位置 32 には、PSW ビット 32 と置き換わる値が含まれます。

24 ビットまたは 31 ビット・アドレッシング・モード (結果のアドレッシング・モードには関係なく) での実行時に作成される PROGRAM CALL トレース・エンタリー (フォーマット 1 エンタリー) のビット位置 32 には、現 PSW から取られた基本アドレッシング・モード・ビット (ビット 32) が含まれます。

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードのときに作成される PROGRAM RETURN トレース・エンタリー (フォーマット 1、フォーマット 2、またはフォーマット 3 エンタリー) のビット 32 には、PSW のビット 32 と置き換わる基本アドレッシング・モード・ビットが含まれます。

24 ビットまたは 31 ビット・アドレッシング・モードで作成される PROGRAM RETURN トレース・エンタリーにおいて、戻りアドレスがエンタリー内の 1 ワードの

みを占めるときは (フォーマット 1 またはフォーマット 4 エンタリー)、そのエンタリーのビット 64 には、PROGRAM RETURN 命令の実行前に存在していた PSW ビット 32 の値が含まれます。戻りアドレスが 2 ワードを占めるときは (フォーマット 7 エンタリー)、ビット位置 96 には PSW ビット 32 の値が含まれます。

更新済み命令アドレス: PSW ビット 31 のオフからオンへの切り替えを示すモード切り替えトレース・エンタリー (フォーマット 1 エンタリー) のビット位置 33-63 には、PSW 内の更新済み命令アドレスのビット 33-63 (PSW のビット 97-127) が含まれます。これは、そのアドレスがモード切り替え命令により置き換えられる場合は、置き換えの前の値です。64 ビット・アドレッシング・モードから 24 ビットまたは 31 ビット・アドレッシング・モードへの切り替えを示すモード切り替えトレース・エンタリー (フォーマット 2) のビット位置 32-63 には、PSW 内の更新済み命令アドレスのビット 0-31 がすべて 0 である場合は、そのアドレスのビット 32-63 (PSW のビット 96-127) が含まれます。これは、そのアドレスがモード切り替え命令により置き換えられる場合は、置き換えの前の値です。また、その更新済み命令アドレスのビット 0-31 がすべて 0 ではない場合は、トレース・エンタリー (フォーマット 3) のビット位置 32-95 には、そのアドレスのビット 0-63 (PSW のビット 64-127) が含まれます。

PROGRAM RETURN トレース・エンタリーに関する以下の説明は、エンタリー内の戻りアドレスがエンタリー内の 1 ワードのみを占める場合に適用されます。24 ビットまたは 31 ビット・アドレッシング・モードでの実行時に作成されるトレース・エンタリー (フォーマット 1 またはフォーマット 4 エンタリー) のビット位置 65-95 には、PSW 内の更新済み命令アドレスがリンクページ・スタック状態エンタリーから置き換えられる前の、そのアドレスのビット 33-63 (PSW のビット 97-127) が含まれます。また、64 ビット・アドレッシング・モードでの実行の場合は、その更新済み命令アドレスのビット 0-31 がすべて 0 であるときは、トレース・エンタリー (フォーマット 2 または 5) のビット位置 64-95 に、その命令アドレスのビット 32-63 (PSW のビット 96-127) が含まれ、更新済み命令アドレスのビット 0-31 がすべて 0 ではないときは、トレース・エンタリー (フォーマット 3 または 6) のビット位置 64-127 に、その命令アドレスのビット 0-63 (PSW のビット 64-127) が含まれます。PROGRAM RETURN トレース・エンタリー内の戻りアドレスが 2 ワードを占めている場合は、エンタリー内の更新済み命令アドレスは、エ

ントリー内で 1 ワード右に移動します (フォーマット 7-9)。

PSW キー: PROGRAM CALL、PROGRAM TRANSFER、または PROGRAM RETURN トレース・エントリーのビット位置 8-11 には、現 PSW からの PSW キーが含まれます。

PC 番号: PROGRAM CALL トレース・エントリーのビット位置 12-31 には、第 2 オペランドのアドレスの右端 20 ビットの値が含まれます。

戻りアドレス: 24 ビットまたは 31 ビット・アドレッシング・モードでの実行時に作成される PROGRAM CALL トレース・エントリー (フォーマット 1 エントリー) のビット位置 33-62 には、PSW 内の更新済み命令アドレスがエントリー・テーブル・エントリーから置き換えられる前の、そのアドレスのビット 33-62 (PSW のビット 97-126) が含まれます。また、64 ビット・アドレッシング・モードでの実行の場合は、トレース・エントリー (フォーマット 2) のビット位置 32-94 に、その更新済み命令アドレスのビット 0-62 (PSW のビット 64-126) が含まれます。

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードであるときに作成される PROGRAM RETURN トレース・エントリー (フォーマット 1、フォーマット 2、またはフォーマット 3 エントリー) のビット位置 33-62 には、PSW のビット 64-127 と置き換わる命令アドレスのビット 33-62 が含まれます。また、結果の PSW ビット 31 が 1 のときは (この場合は、結果の PSW ビット 32 が 0 でなければアドレッシング・モードは 64 ビットになります)、命令アドレスのビット 0-31 がすべて 0 の場合は、トレース・エントリー (フォーマット 4-6) のビット位置 32-62 に、その命令アドレスのビット 0-62 が含まれ、命令アドレスのビット 0-31 がすべて 0 ではない場合は、トレース・エントリー (フォーマット 7-9) のビット位置 32-94 に、その命令アドレスのビット 0-62 が含まれます。

問題プログラム状態ビット (P): 24 ビットまたは 31 ビット・アドレッシング・モード (結果のモードに関係なく) での実行時に作成される PROGRAM CALL トレース・エントリー (フォーマット 1 エントリー) のビット位置 63、または、64 ビット・アドレッシング・モードでの実行時に作成されるエントリー (フォーマット 2) のビット 95 には、現 PSW からの問題プログラム状態ビットが含まれます。

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードであるときに作成される PROGRAM RETURN トレース・エントリー (フォーマット 1、フォーマット 2、またはフォーマット 3 エントリー)、または結果の PSW ビット 31 が 1 で戻りアドレスのビット 0-31 がすべて 0 であるときに作成される

PROGRAM RETURN トレース・エントリー (フォーマット 4-6) のビット位置 63 には、PSW のビット 15 と置き換わる問題プログラム状態ビットが含まれます。結果の PSW ビット 31 が 1 で、戻りアドレスのビット 0-31 がすべて 0 ではないときに作成される

PROGRAM RETURN トレース・エントリー (フォーマット 7-9) のビット位置 95 には、問題プログラム状態ビットが含まれます。

新規 PASN: PROGRAM TRANSFER トレース・エントリーのビット位置 16-31 には、汎用レジスタ R₁ のビット位置 48-63 に指定されている新規 PASN が含まれます (これは 0 のこともあります)。

PROGRAM RETURN トレース・エントリーのビット位置 16-31 には、リンケージ・スタック状態エントリーから復元された新規 PASN が含まれます。

実行前の R₂ のビット 32-63: 24 ビットまたは 31 ビット・アドレッシング・モードでの実行時に作成される PROGRAM TRANSFER トレース・エントリー (フォーマット 1 エントリー) のビット位置 32-63 には、この命令の R₂ フィールドに指定されている汎用レジスタのビット 32-63 が含まれます。(そのレジスタのビット 32 および 33-62 は、それぞれ、PSW のビット 32 および 97-126 と置き換わります。このレジスタのビット 63 は、PSW の問題プログラム状態ビットと置き換わります)。PROGRAM TRANSFER が 64 ビット・アドレッシング・モードで実行されているときは、R₂ 汎用レジスタのビット 0-31 がすべて 0 の場合は、トレース・エントリー (フォーマット 2) のビット位置 32-63 にレジスタのビット 32-63 が含まれ、レジスタのビット 0-31 がすべて 0 ではない場合は、トレース・エントリー (フォーマット 3) のビット位置 32-95 にレジスタのビット 0-63 が含まれます。

新規 SASN: SET SECONDARY ASN トレース・エントリーのビット位置 16-31 には、この命令が制御レジスタ 3 にロードした ASN 値が含まれます。

レジスタの数 (N): TRACE 用のトレース・エントリーのビット 4-7 には、これまででそのトレース・エントリーで提供された汎用レジスタの数から 1 を引いた

値が含まれます。N の値の範囲は、0 (トレース・エン
トリー内で 1 つの汎用レジスタの内容が提供されたこ
とを意味する) から、15 (16 個すべての汎用レジスタ
の内容が提供されたことを意味する) までです。

TOD クロック・ビット 16-63 または 0-79: TRACE
(TRACE) 用のトレース・エントリーのビット 16-63
は、TOD クロックのビット位置 16-63 から取られま
す。この値は、通常、TRACE 命令の実行時点で実行さ
れる STORE CLOCK 命令により提供されます。

TRACE (TRACG) 用のトレース・エントリーのビット
16-95 は、TOD クロックのビット位置 0-79 から取ら
れます。この値は、通常、TRACE 命令の実行時点で
実行される STORE CLOCK EXTENDED 命令により
提供されます。TOD クロックのビット位置 0 からの繰
り上がりについては、『プログラミング上の注意 2』を
参照してください。

TRACE オペランド: TRACE (TRACE) 用のトレ
ース・エントリーのビット位置 64-95 には、このエン
トリーの対象となった TRACE 命令の第 2 オペランドの 32
ビットのコピーが含まれます。TRACE (TRACG) 用の
トレース・エントリーのビット位置 96-127 には、これ
らのビットのコピーが含まれます。

(R₁)-(R₃): TRACE (TRACE) 用のトレース・エン
トリーのビット 96 から始まる一連の 4 バイトのフィー
ルドには、この TRACE 命令の R₁ および R₃ フィ
ールドに指定されている範囲内の汎用レジスタのビット
位置 32-63 の内容が含まれます。これらの汎用レジス
ターは、汎用レジスタ R₁ から始まり汎用レジスタ R₃
まで (このレジスタも含む)、レジスタ番号の昇順に
従って格納されます。汎用レジスタ 0 は、汎用レジス
ター 15 の後に続きます。同様に、TRACE (TRACG)
用のトレース・エントリーのビット 128 から始まる一
連の 8 バイト・フィールドには、これらのレジスタの
ビット位置 0-63 の内容が含まれます。

プログラミング上の注意:

- TRACE (TRACE) 用のトレース・エントリーのサイ
ズは、3 + (N + 1) ワードです。1 つのエン
トリーの最大サイズは 19 ワード、つまり 76 バイト
です。TRACE (TRACG) の場合は、サイズは
4 + 2(N + 1) ワードで、最大サイズは 36 ワ
ード、つまり 144 バイトです。
- 将来は、新しいモデルで TOD クロックの左端部分
が拡張され、クロックのビット位置 0 からこの拡張
部分への繰り上がりが可能になる予定です。『プロ
グラミング上の注意 13 (4-37ページ)』を参照し

てください。このようなモデルでは、拡張部分の右
端のビットが、TRACE (TRACG) トレース・エン
トリーのビット位置 15 に格納されます。TRACE
(TRACG) トレース・エントリーを処理するプログ
ラムでは、この将来の拡張を考慮に入れることをお
勧めします。

操作

トレースの対象となる命令が実行されたときに、対応す
るトレース機能がオンになっていれば、該当するタイプ
およびフォーマットのトレース・エントリーが作成され
ます。このトレース・エントリーの実アドレスは、制御
レジスタ 12 のビット位置 2-61 の値の左側に 2 個、
そして右側に 2 個の 0 ビットを付加したものです。そ
の結果、制御レジスタ 12 の中のアドレスは、作成さ
れたエントリーのサイズ分だけ増加します。

制御レジスタ 12 の中のアドレス増分が原因で、繰り
上がりがビット位置 51 まで及ぶ (つまり、トレース・
エントリーが次の 4K バイト・ブロックに入る) こと
になる場合は、トレース・エントリーは格納されませ
ん。作成するエントリーについてこの状況が生じたときは、
トレース・テーブル例外が認識されます。PROGRAM
CALL の結果、PROGRAM CALL トレース・エン
トリーとモード切り替えトレース・エントリーの両方が
作成されるときに、そのどちらかでビット位置 51 への繰
り上がりが生じる場合は、どちらのエントリーも格納さ
れません。TRACE 命令の場合は、トレース・テー
ブル例外が認識されるようにするために、実際の長さの代
わりに、最大長である 76 バイト (TRACE の場合) また
は 144 バイト (TRACG の場合) が使用されます。

トレース・エントリーの格納は、キー制御保護の対象に
はなりません (また、トレース・エントリー・アドレス
は実アドレスなので、ページ保護の対象にもなりません)
が、低アドレス保護の対象にはなりません。つまり、
作成する必要があるトレース・エントリーのアドレスが
0-511 または 4096-4607 の範囲内にあり、制御レジ
スター 0 のビット 35 が 1 である場合は、保護例外が認
識され、命令の実行は抑止されます。トレース・エン
トリーのアドレスが無効な場合は、アドレッシング例外が
認識され、命令の実行は抑止されます。

トレース・エントリーの格納に関連した 3 つの例外 (ア
ドレッシング例外、保護例外、およびトレース・テー
ブル例外) を、総称的にトレース例外と呼びます。

トレース例外条件以外の条件についてプログラム割り込
みが起こり、命令の実行が完了していない場合は、割り
込まれた命令について作成する必要があるトレース・エ

ントリーの一部または全部が、トレース・テーブルに格納されるかどうかは、予測不能です。したがって、一般に命令実行の無効化または抑止を引き起こすような条件が生じた場合は、制御レジスタ 12 に指定されている記憶位置から始まり、作成されるはずのエントリーの長さに達するまでの記憶位置が、変更されている可能性があります。

PROGRAM RETURN が、BRANCH AND STACK により形成されたリンケージ・スタック状態エントリーをアンスタックしたときに、ASN トレースがオンになっている場合は、トレース・エントリーが作成されず、トレース・エントリーがまったく格納されていないとしても、トレース例外が認識されることがあります。

トレース・エントリーに入れられる順序は、予測不能です。さらに、他の CPU およびチャネル・プログラムから見た場合、トレース・エントリーの対象となっている命令が完了するまでに、そのエントリーの 1 つのバイトの内容が複数回変更されているように見ることがあります。

制御レジスタ 12 の中のトレース・エントリー・アドレスは、トレース・エントリーが作成される命令の実行が完了したときのみ更新されます。

操作の開始前と完了後に、逐次化およびチェックポイント同期の機能が実行されます。

プログラム・イベント記録

PER の目的は、プログラムのデバッグを助けることにあります。これにより、以下のタイプのイベントについて、プログラムに警報を出すことができます。

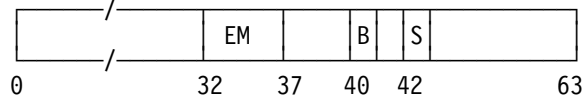
- ブランチ命令の実行の成功。ブランチ・ターゲット位置が指定した記憶域内にある場合のみ、イベントが生じるようにするためのオプションがあります。
- 指定した記憶域からの命令の取り出し。
- 指定した記憶域の内容の変更。該当の記憶域が指定したアドレス・スペース内にある場合のみ、イベントが生じるようにするためのオプションがあります。
- STORE USING REAL ADDRESS 命令の実行。

プログラムで、上記の 1 つまたは複数のタイプのイベントを選択的に指定することができます。ただし、STORE USING REAL ADDRESS に関するイベントは、必ず記憶域変更イベントと一緒に指定する必要があります。PER イベントに関する情報は、プログラム割り込みを使用してプログラムに渡されます。割り込みの原因は割り込みコードに示されます。

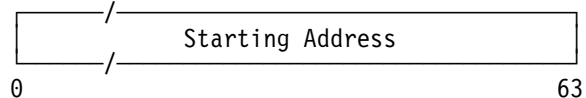
制御レジスタ割り振りおよびアドレス・スペース制御エレメント

PER を制御するための情報は、制御レジスタ 9、10、11、およびアドレス・スペース制御エレメントに入っています。制御レジスタ内の情報のフォーマットは、以下のとおりです。

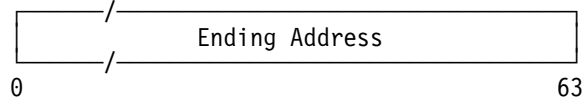
制御レジスタ 9



制御レジスタ 10



制御レジスタ 11



PER イベント・マスク (EM): ビット 32-34 および 36 は、どのタイプのイベントを認識するかを指定します。これらのビットは以下のように割り当てられます。

- ビット 32: ブランチ成功イベント
- ビット 33: 命令取り出しイベント
- ビット 34: 記憶域変更イベント
- ビット 36: 実アドレスを使用した格納イベント (ビット 34 も 1 でなければなりません)

32-34 および 36 が 1 の場合、対応するタイプのイベントを認識することを指定します。ただし、ビット 36 をこの目的に使用できるのは、ビット 34 も 1 である場合に限られます。ビット 34 が 1 のときは、記憶域変更イベントが認識されます。ビット 34 と 36 が 1 のときは、記憶域変更イベントと実アドレスを使用した格納イベントの両方が認識されます。ビットが 0 の場合は、それに対応するタイプのイベントは認識されません。ビット 34 が 0 のときは、記憶域変更イベントと実アドレスを使用した格納イベントのどちらも認識されません。

ブランチ・アドレス制御 (B): 制御レジスタ 9 のビット 40 が 1 の場合、指定した記憶域内の位置へのブランチについてのみ、ブランチ成功イベントが生じることを指定します。ビット 40 が 0 のときは、ブランチ・ターゲットのアドレスに関係なくブランチ成功イベントが生じます。

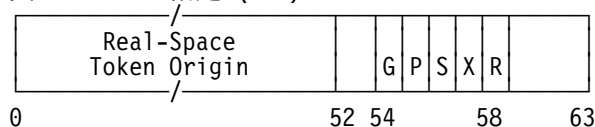
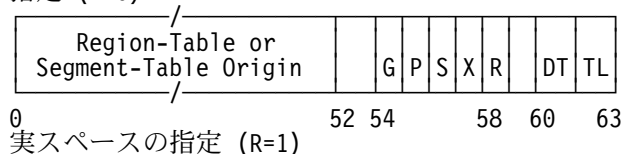
記憶域変更スペース制御 (S): 制御レジスタ 9 のビット 42 が 1 の場合、指定したアドレス・スペース内にある指定した記憶域を参照した結果として記憶域変換イベントが生じることを指定します。アドレス・スペースは、アドレス・スペースに対する参照を変換するために使用されるアドレス・スペース制御要素の記憶域変更イベント・ビットによって記憶域変更イベントが生じるスペースとして、指定されます。DAT がオフのときは、ビット 42 は無視されます。DAT がオフでビット 42 が 0 のときは、記憶域変更イベントの発生は特定のアドレス・スペースのみに限定されません。

PER 開始アドレス: 制御レジスタ 10 のビット 0-63 は、指定した記憶域の始めのアドレスです。

PER 終了アドレス: 制御レジスタ 11 のビット 0-63 は、指定した記憶域の終わりのアドレスです。

アドレス・スペース制御要素のフォーマットは、次のいずれかです。

領域テーブルまたはセグメント・テーブルの指定 (R=0)



記憶域変更イベント・ビット (S): 制御レジスタ 9 の記憶域変換スペース制御が 1 のときに、アドレス・スペース制御要素のビット 56 が 1 の場合、アドレス・スペース制御要素で定義されているアドレス・スペースが、記憶域変更イベントが生じるスペースであることを指定します。ビット 56 が検査されるのは、アドレス・スペース制御要素を使用して、記憶機構オペランド格納参照のための動的アドレス変換が行われるときです。アドレス・スペース制御要素は、制御レジスタ 1、7、または 13 中の PASCE、SASCE、または HASCE であることもあり、アクセス・レジスタ変換時に ASN 第 2 テーブル・エントリから取得されることもあります。ビット 56 は、主記憶機構内の ASN 第 2 テーブル・エントリからではなく、ART 索引バッファ (ALB) 内の ASN 第 2 テーブル・エントリから取得されることがあります。記憶域変更スペース制御が 0 のときは、ビット 56 は無視されます。

プログラミング上の注意:

1. CPU が PER イベントを使用できるように設定されているときは、モデルのパフォーマンスが低下することがあります。PER 機能の動作が原因で CPU のパフォーマンスが低下しないようにするには、この機能を使用しないプログラムでは PER イベントに対して CPU を使用不可にするようにしてください。そのためには、PSW 中の PER マスクを 0 に設定するか、制御レジスタ 9 中の PER イベント・マスクを 0 に設定するか、またはその両方を行います。これらのフィールドのいずれかが 0 のときは、PER が原因でパフォーマンスの低下が生じることはありません。
2. 一部のモデルでは、PER イベントに対して CPU を使用不可にしてあっても、制御レジスタ 9、10、および 11 がロードされるたびに、ややパフォーマンスが低下することがあります (4-29 ページの『記憶域の指示』の項の『プログラミング上の注意』を参照)。

操作

PER は、PSW のビット 1、つまり PER マスクにより制御されます。PER マスクと特定の PER イベント・マスク・ビットがすべて 1 のときは、CPU は対応するタイプのイベントに対して使用可能にされ、そうでないときは使用不可にされます。ただし、実アドレスを使用した格納イベントに対して CPU が使用可能にされるのは、記憶域変更マスク・ビットおよび実アドレスを使用した格納マスク・ビットが両方とも 1 のときだけです。

PER を原因とする割り込みが生じるのは、通常、そのイベントを発生させた命令の実行後です。イベントの発生により、命令の実行が影響を受けることはありません。つまり、命令は、完了、部分完了、中止、抑止、または無効化の状態を終了します。ただし、記憶域変更イベントが認識された場合は、そのイベントの原因となったバイト以降が格納されるのは最大 4K バイトまでであり、その結果、割り込み可能命令が部分完了の状態を終了することがあります。

ある PER イベントが生じたときに、PSW 内の PER マスクまたは制御レジスタ 9 中のマスクにより、その特定のイベントに対して CPU が使用不可にされている場合は、そのイベントは認識されません。

PSW 内の PER マスクまたは制御レジスタ 9、10、および 11 の PER 制御フィールドに変更を加えた場合、その次の命令の実行時から PER が変更されます。

制御レジスター 1、7、または 13 のアドレス・スペース制御エレメントの記憶域変更イベント・ビットに変更を加えた場合も、その次の命令の実行時から PER が変更されます。アクセス・レジスターの変換時に、主記憶機構または ALB 中の ASN 第 2 テーブル・エントリーから取得されるアドレス・スペース制御エレメントの記憶域変更イベント・ビットに変更を加えた場合は、必ずしも即時に PER が変更されるとは限りません。ただし、PURGE ALB の実行後は、即時に PER が変更されます。

ある PER イベント・タイプについて CPU を使用可能から使用不可に変更する命令を実行しているときに、そのタイプの PER イベントが起きた場合は、そのイベントは認識されます。

命令の試行実行の中で PER イベントが認識され、その後、実際の実行段階で、その命令、DAT テーブル・エントリー、およびオペランドが再取り出しされます。再取り出しされたフィールドが、試行実行から実際の実行までの間に別の CPU またはチャンネル・プログラムにより変更されていた場合は、示されている PER イベントが試行実行に関するものであるか、実際の実行に関するものであるかは、予測不能です。

本書には記載されていない特殊目的の命令については、PER の働きは、このセクションで述べた説明とはやや異なる場合があります。

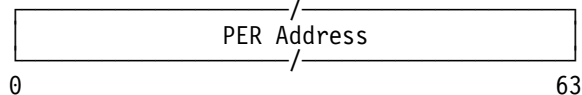
原因の識別

PER のためのプログラム割り込みが起きると、割り込みコードのビット 8 が 1 に設定され、実記憶位置 150-159 に識別情報が入れます。また、PER イベントが記憶域変更イベントである場合は、記憶位置 161 にも識別情報が入れます。さらに、プログラム旧 PSW および ILC 中の命令アドレスを使用して、追加の情報も提供されます。実記憶位置 150-159 および 161 に格納される情報のフォーマットは、以下のとおりです。

記憶位置 150-151:

PERC	000	ATMID	AI
0	5	8	13 15

記憶位置 152-159:



記憶位置 161:

0000	PAID
0	4 7

PER コード (PERC): PER イベントの発生は、ビット位置 0-2 および 4 中の 1 により示されます。PER コード内での特定タイプのイベントのビット位置は、制御レジスター 9 の PER イベント・マスク・フィールド内でのそのイベントのビット位置から 32 を引いた位置です。ただし、記憶位置 150 のビット位置 2 が 1 で、ビット位置 4 が 0 であれば、記憶域変更イベントを示します。また、ビット位置 2 および 4 がどちらも 1 のときは、実アドレスを使用した格納イベントを示します。プログラム割り込みが起きたときは、複数タイプの PER イベントが同時に示されることがあります。さらに、別のプログラム割り込み条件も存在している場合は、プログラム割り込みの割り込みコードには、PER イベントと同時にもう 1 つの条件も示されることがあります。

アドレッシングおよび変換モード識別 (ATMID): PER イベントが示されたときのプログラム割り込みの際に、そのイベントの原因となった命令の実行開始時点での PSW のビット 31、32、5、16、および 17 が、それぞれ、実記憶位置 150-151 のビット 8 および 10-13 に格納されることがあります。ビット 31、32、5、16、および 17 が格納された場合は、記憶位置 150-151 のビット位置 9 に 1 が格納されます。ビット 31、32、5、16、および 17 が格納されなかった場合は、記憶位置 150-151 のビット位置 8-13 に 0 が格納されます。

実記憶位置 150-151 のビット 8-13 を、アドレッシングおよび変換モード識別 (ATMID) と呼びます。ビット 9 は、ATMID 妥当性ビットと呼ばれます。ビット 9 が 0 のときは、無効な ATMID (すべて 0) が格納されたことを示します。

有効な ATMID のビットの意味は、以下のとおりです。

ビット	意味
8	PSW ビット 31
9	ATMID 妥当性ビット
10	PSW ビット 32
11	PSW ビット 5
12	PSW ビット 16
13	PSW ビット 17

必ず有効な AT MID が格納されるのは、PER イベントが、以下のいずれかの命令が原因で発生した場合のみです。

- BRANCH AND SAVE AND SET MODE (BASSM)
- BRANCH AND SET AUTHORITY (BSA)
- BRANCH AND SET MODE (BSM)
- BRANCH IN SUBSPACE GROUP (BSG)
- LOAD PSW (LPSW)
- LOAD PSW EXTENDED (LPSWE)
- PROGRAM CALL (PC)
- PROGRAM RETURN (PR)
- PROGRAM TRANSFER (PT)
- RESUME PROGRAM (RP)
- SET ADDRESS SPACE CONTROL (SAC)
- SET ADDRESS SPACE CONTROL FAST (SACF)
- SET ADDRESSING MODE (SAM24, SAM31, SAM64)
- SET SYSTEM MASK (SSM)
- STORE THEN AND SYSTEM MASK (STNSM)
- STORE THEN OR SYSTEM MASK (STOSM)
- SUPERVISOR CALL (SVC)
- TRAP (TRAP2, TRAP4)

上記以外の命令が原因で発生した PER イベントの場合は、有効な AT MID が格納されるかどうかは予測不能です。

PER ASCE 識別 (AI): PER コードに記憶域変更イベントが示されていて (ビット 2 が 1 でビット 4 が 0)、このイベントが起きたときに DAT がオンであった場合は、記憶位置 150-151 のビット 14 および 15 には、このイベントの原因となった参照を変換するために使用されたアドレス・スペース制御エレメント (ASCE) が、以下のように示されます。

ビット

14-15 意味

- | | |
|----|--|
| 00 | 1 次 ASCE が使用されました。 |
| 01 | AR 指定の ASCE が使用されました。PER アクセス ID (実記憶位置 161) を調べることでより、使用された ASCE を判別できます。ただし、1 次、2 次、またはホーム ASCE が使用された場合は、ビット 14 および 15 は、01 ではなく、それぞれ 00、10、または 11 に設定されることがあります。 |
| 10 | 2 次 ASCE が使用されました。 |
| 11 | ホーム ASCE が使用されました。 |

CPU は、アクセス・リスト・エントリー・トークン (ALET) 00000000 または 00000001 (16 進数) が使用されたことを認識するか、または、ALET が、アクセス・エントリー・リストを介して、1 次 ASCE、2 次 ASCE、またはホーム ASCE に等しい ASCE を含む ASN 第 2 テーブル・エントリーを指定したことを認識した場合、ビット 14 および 15 を 01 に設定しないことがあります。

PER コードに記憶域変更イベントが示されていないか (ビット 2 が 0、またはビット 4 が 1)、または DAT がオフの場合は、ビット位置 14 および 15 には 0 が格納されます。

記憶位置 150-151 のビット位置 3 および 5-7 には、0 が格納されます。

PER アドレス: 記憶位置 152-159 の PER アドレス・フィールドには、1 つ以上の PER イベントが認識されたときに実行されていた命令を取り出すために使用される命令アドレスが含まれます。その命令が EXECUTE のターゲットである場合は、EXECUTE 命令を取り出すために使用される命令アドレスが、PER アドレス・フィールドに入れられます。

PER アクセス識別 (PAID): PER コードに記憶域変更イベントが示されている場合は、そのイベントが適用されるアドレス・スペースを示す値が、記憶位置 161 に格納されることがあります。このアクセスに AR 指定のアドレス・スペース制御エレメントが使用された場合は、使用されたアクセス・レジスタの番号が記憶位置 161 のビット位置 4-7 に格納され、ビット位置 0-3 には 0 が格納されます。(1) CPU がアクセス・レジスタ・モードであり、アクセスがリンケージ・スタックへの暗黙参照であった場合、または、(2) CPU がアクセス・レジスタ・モードではなかった場合は、記憶位置 161 の内容は予測不能です。

命令アドレス: プログラム旧 PSW 内の命令アドレスは、別のプログラム状態が発生していない限り、次に実行されることになっていた命令のアドレスです。別のプログラム状態がある場合は、命令アドレスは、その条件が原因で終了しようとしている命令によって決まります。

ILC: ILC は、PER アドレスが示す命令の長さを示します。ただし、LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、または監視プログラム呼び出し割り込みにより取り込まれた PSW についての同時指定例外が原因で、ILC が 0 に設定されることがあります。

プログラミング上の注意:

1. PSW ビット 31 は拡張アドレッシング・モード・ビットで、PSW ビット 32 は基本アドレッシング・モード・ビットです。PSW ビット 31 および 32 が両方とも 1 である場合は、64 ビット・アドレッシング・モードであることを示します。PSW ビット 31 が 0 のときは、PSW ビット 32 が 0 であれば 24 ビット・アドレッシング・モードで、1 であれば 31 ビット・アドレッシング・モードです。PSW ビット 5 は DAT モード・ビットで、PSW ビット 16 および 17 はアドレス・スペース制御ビットです。異なる変換のモードでの命令アドレスおよび論理アドレスの取り扱い方法については、3-27 ページの『変換モード』を参照してください。
2. AT MID が有効な場合は、PER イベントを取り扱うプログラムは、そのイベントの原因となった命令がどのアドレス・スペースから取り出されたかを判別するほか、その命令の記憶域オペランド参照にどの変換モードが適用されたかを判別することができます。必ず有効な AT MID が格納される各命令では、PSW ビット 5、16、および 17 の 1 つまたは幾つかが変更されることがあります。その結果、PER イベントが原因でプログラム旧 PSW に格納されたこれらのビットは、必ずしも、イベントの原因となった命令の実行開始時に存在していた値と同じではなくなることがあります。PSW ビット 5、16、および 17 を変更できるのは、必ず有効な AT MID が格納される命令のみです。
3. 記憶域変更 PER イベントが示され、そのイベントの発生時に DAT がオンであった場合は、PER ASCE 識別 (実記憶位置 150-151 のビット 14 および 15) に、そのイベントの原因となった参照を変換するために使用されたアドレス・スペース制御エレメントが示されます。ビット 14 および 15 が、AR 指定のアドレス・スペース制御エレメントが使用されたことを示している場合は、実記憶位置 161 の PER アクセス識別を使用して、参照されたアドレス・スペースを判別できます。DAT がオンであったかどうかを判別するには、PER イベントを取り扱うプログラムは、まず AT MID 妥当性ビットを調べて、有効な AT MID が格納されたかどうかを判別し、有効な AT MID が格納されている場合は、その AT MID の中の DAT モード・ビットを調べる必要があります。有効な AT MID が格納されていない場合は、プログラムは、プログラム旧 PSW の中の DAT モード・ビットを調べる必要があります。

4. 有効な AT MID が格納されている場合は、PER イベントを取り扱うプログラムは、その PER イベントの原因となった命令について存在していたアドレッシング・モード (24 ビット、31 ビット、または 64 ビット) も判別できます。アドレッシング・モードが分かれば、プログラムは、命令のアドレスおよび命令の記憶域オペランド (存在する場合) のビット位置 0-39 に含まれている 1 の値の意味を確実に判別でき、したがって、命令およびオペランドの位置を正確に判別できます。命令のアドレスは、実記憶位置 152-159 の中の PER アドレスからは必ずしも正確には判別できないという点に注意してください。なぜなら、そのアドレスは EXECUTE 命令のアドレスである可能性があり、しかも、ターゲット命令のアドレスは、依然として、EXECUTE 命令の第 2 オペランド・アドレスを指定するフィールドから判別されるからです。もう 1 つの誤りの原因として、24 ビットまたは 31 ビット・アドレッシング・モードでは、それぞれ 16M バイトまたは 2G バイトの境界の直前で、命令またはオペランドが循環することがあるという点を忘れないでください。
5. アドレッシング・モード・ビットを変更できる命令の場合は、必ず有効な AT MID が格納されます。ただし、AT MID メカニズムには、PER イベントの原因となっている命令およびその命令のオペランドを、正確に見つけることができるという完全な保証はありません。これは、LOAD CONTROL および LOAD ADDRESS SPACE PARAMETERS が、その命令を取り出すために使用されたアドレス・スペース制御エレメントを変更する場合があるからです。

標識の優先順位

プログラム割り込みが発生したときに、複数の PER イベントが認識されていた場合は、認識されたすべての PER イベントが同時に PER コードに示されます。さらに、同時に別のプログラム割り込み条件も存在している場合は、プログラム割り込みの割り込みコードには、PER 条件と共にもう 1 つの条件も示されます。

SUPERVISOR CALL について命令取り出しイベントが生じた場合は、監視プログラム呼び出し割り込みの直後にプログラム割り込みが発生します。

命令の実行中に PER イベントが認識され、その命令によって持ち込まれた新 PSW に早期認識タイプの PSW フォーマット・エラーがある場合は (6-9 ページの『PSW に関連した例外』を参照)、プログラム割り込みの割り込みコードには、指定例外と PER の両方が同時に示されます。ただし、遅く認識されるタイプの PSW

フォーマット・エラーの場合は、割り込みコードには PER のみが示されます。どちらの場合も、無効な PSW はプログラム旧 PSW として格納されます。

PER が認識されても、通常は命令実行の終了が影響を受けることはありません。ただし、以下の場合、割り込み可能命令の実行は正常に完了しません。

1. 命令が、非同期的な条件 (入出力条件、外部条件、再始動条件、または抑制可能なマシン・チェック条件) のために割り込まれる場合は、最初に PER イベントに関するプログラム割り込みが発生し、その後で、その他の割り込みが、新規 PSW 内のマスク・ビットに従い、通常の優先順位で発生します。
2. 停止機能が実行されるときは、CPU が停止状態に入る前に、PER イベントを示すプログラム割り込みが起きます。
3. プログラム例外が認識されたときは、その命令実行で認識された PER イベントが同時に示されます。
4. モデルによっては、特定の状況下で PER イベントが認識された場合に、命令が早期に割り込まれることがあります。その結果、プログラム例外が同時に示されず、非同期的条件に関する割り込みは起こらず、また CPU が停止状態にならないことがあります。特に、記憶域変更イベントが認識された場合は、そのイベントの原因となったバイト以降が格納されるのは、最大 4K バイトまでです。

上記 1 および 2 において、認識された PER イベントが命令取り出しイベントのみであり、命令の他の操作単位がまだ実行されないままになっている場合は、そのイベントが破棄され、その結果プログラム割り込みは起こらないことがあります。イベントが破棄されるかどうかは予測不能です。

プログラミング上の注意:

1. 以下の場合、1 つの命令が、PER イベントを対象としたプログラム割り込みを引き起こすと同時に、PER イベントを対象とした割り込みを制御するフィールドの値を変更することがあります。この場合に、その PER イベントを対象とするプログラム割り込みが起きるかどうかは、元のフィールド値によって決まります。
 - a. LOAD PSW、LOAD PSW EXTENDED、SET SYSTEM MASK、STORE THEN AND SYSTEM MASK、および SUPERVISOR CALL 命令は、命令取り出しイベントを引き起こし、PER 割り込みに対して CPU を使用不可にすることがあります。さらに、STORE

THEN AND SYSTEM MASK の結果、記憶域変更イベントが示されることがあります。いずれの場合も、PER イベントを対象とするプログラム割り込みに関連したプログラム旧 PSW には、PER イベントに対して CPU が使用不可であることが示されることがあります。

- b. 制御レジスター 9 の中の PER イベント・マスクの値、または命令取り出しイベントの表示を制御する制御レジスター 10 および 11 の中のアドレスを変更する LOAD CONTROL 命令の実行時に、命令取り出しイベントが認識されることがあります。
 - c. アクセス・レジスター・モードでは、ASN 第 2 テーブル・エントリー (アクセス・リスト・エントリーに示されている) の中のアドレス・スペース制御エレメントの記憶域変更イベント・ビットの値が 1 のときに許される記憶域変更イベントが、そのビットの値を 1 から 0 に変更する任意の格納タイプの命令により引き起こされることがあります。
2. 割り込み可能命令の実行中に PER 割り込みが起きた場合は、ILC には、その命令または EXECUTE のいずれか該当する方の長さが示されます。LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、または SUPERVISOR CALL の結果として PER 割り込みが起きた場合は、これらの命令または EXECUTE のいずれか該当する方の長さが示されます。ただし、LOAD PSW、LOAD PSW EXTENDED、または PROGRAM RETURN で同時に指定例外が生じたときは、ILC は 0 になります。
 3. ブランチが原因で PER 割り込みが起きた場合は、PER アドレスはそのブランチ命令 (または EXECUTE) を指し示し、旧 PSW は次に実行すべき命令を指し示します。割り込み可能命令の実行中に割り込みが起きたときは、PER アドレスと、旧 PSW 内の命令アドレスは同じです。

記憶域の指示

2 つのタイプの PER イベント (命令取り出しイベントと記憶域変更イベント) では、常に記憶機構内の特定領域が指示されます。また、ブランチ成功イベントにも、この指示が含まれることがあります。この記憶域は、制御レジスター 10 の中の開始アドレスが示す記憶位置から始まり、制御レジスター 11 の中の終了アドレスが示す記憶位置で終わります (この記憶位置も含まれます)。この記憶域は、開始アドレスの右側に設けられます。

命令取り出しイベントは、命令の最初の 1 バイトまたは EXECUTE 命令ターゲットの最初の 1 バイトが指示された記憶域から取り出されるたびに発生します。記憶域変更イベントは、論理アドレスまたは仮想アドレスとして定義されたオペランド・アドレスを使用して、指定された記憶域への格納アクセスが行われたときに発生します。ただし、DAT がオンで、制御レジスター 9 の中の記憶域変更スペース制御が 1 のときは、記憶域変更イベントが起きるのは、対象の記憶域が、アドレス・スペース制御エレメント内の記憶域変更イベント・ビットが 1 になっているアドレス・スペースに含まれている場合のみです。実アドレスとして定義されているオペランド・アドレスを使用した格納アクセスの場合は、記憶域変更イベントは発生しません。制御レジスター 9 の中のブランチ・アドレス制御が 1 のときは、ブランチ・ターゲット命令の最初の 1 バイトが指示された記憶域内にあれば、ブランチ成功イベントが起きます。

ブランチ成功イベント、命令取り出しイベント、および記憶域変更イベント用として指定された一組のアドレスは、アドレス $2^{64} - 1$ の位置で循環します。つまり、アドレス 0 は、アドレス $2^{64} - 1$ の次に続くものと見なされます。開始アドレスが終了アドレスより小さいときは、その領域は連続していることとなります。開始アドレスが終了アドレスより大きいときは、開始アドレスからアドレス $2^{64} - 1$ までの領域と、アドレス 0 から終了アドレスまでの領域（終了アドレスを含む）を合わせた範囲が指定されたこととなります。開始アドレスと終了アドレスが同じである場合は、その 1 つの記憶位置が指定されます。

ブランチ成功イベント、命令取り出しイベント、および記憶域変更イベントのためのアドレス比較には、常に 64 ビット・アドレスが使用されます。そのために、24 ビットまたは 31 ビット・アドレッシング・モードでは、開始アドレスと終了アドレスを比較する前に、仮想、論理、または命令アドレスの左側に、それぞれ 39 個または 33 個の 0 が付加されます。

プログラミング上の注意: 一部のモデルでは、TLB 内の各ページ・テーブル・エントリーを拡張することにより、アドレス範囲検査のパフォーマンスが向上します。このようなインプリメンテーションでは、ブランチ成功、命令取り出し、または記憶域変更イベント・マスクが 1 のときに、制御レジスター 10 および 11 の内容を変更するか、または、これらの PER イベント・マスクを 1 に設定すると、TLB からエントリーが消去されることがあります。この性能低下は、CPU が PER イベントに対して使用不可にされている場合でも生じることがあります。したがって、可能な限り、プログラムで制御レジスター 9、10、または 11 をロードすることは避けてください。

PER イベント

ブランチ成功

制御レジスター 9 の中のブランチ・アドレス制御が 0 のときは、ブランチ・ターゲットのアドレスとは関係なくブランチ成功イベントが起きます。ブランチ・アドレス制御が 1 のときは、取り出されたブランチ・ターゲット命令の最初の 1 バイトが、制御レジスター 10 および 11 に指定されている記憶域に含まれている場合のみ、ブランチ成功イベントが起きます。

以下のいずれかの命令の結果ブランチが行われるたびに、ブランチ・アドレス制御の効力の範囲内で、ブランチ成功イベントが起きます。

- BRANCH AND LINK (BAL, BALR)
- BRANCH AND SAVE (BAS, BASR)
- BRANCH AND SAVE AND SET MODE (BASSM)
- BRANCH AND SET AUTHORITY (BSA)
- BRANCH AND SET MODE (BSM)
- BRANCH AND STACK (BAKR)
- BRANCH IN SUBSPACE GROUP (BSG)
- BRANCH ON CONDITION (BC, BCR)
- BRANCH ON COUNT (BCT, BCTR, BCTG, BCTGR)
- BRANCH ON INDEX HIGH (BXH, BXHG)
- BRANCH ON INDEX LOW OR EQUAL (BXLE, BXLEG)
- BRANCH RELATIVE AND SAVE (BRAS)
- BRANCH RELATIVE AND SAVE LONG (BRASL)
- BRANCH RELATIVE ON CONDITION (BRC)
- BRANCH RELATIVE ON CONDITION LONG (BRCL)
- BRANCH RELATIVE ON COUNT (BRCT, BRCTG)
- BRANCH RELATIVE ON INDEX HIGH (BRXH, BRXHG)
- BRANCH RELATIVE ON INDEX LOW OR EQUAL (BRXLE, BRXLG)
- RESUME PROGRAM (RP)
- TRAP (TRAP2, TRAP4)

また、以下の命令の結果ブランチが行われた場合も、ブランチ・アドレス制御の効力の範囲内で、ブランチ成功イベントが起きます。

- PROGRAM CALL (PC)
- PROGRAM RETURN (PR)
- PROGRAM TRANSFER (PT)

PROGRAM CALL、PROGRAM RETURN、および PROGRAM TRANSFER の場合は、ブランチ・ターゲット・アドレスが、命令が PSW に入れる新しい命令アドレスと見なされます。

ブランチ成功イベントが起きた場合、PER イベント・マスクのビット 32 が 1 で、PSW 中の PER マスクが 1 であれば、PER ブランチ成功イベントが認識されます。

PER ブランチ成功イベントは、PER コードのビット 0 を 1 に設定することにより示されます。

命令取り出し

命令取り出しイベントが起きるのは、命令の最初の 1 バイトが、制御レジスター 10 および 11 に指定されている記憶域の中にある場合です。また、EXECUTE のターゲットの最初のバイトが指定された記憶域内にある場合も、命令取り出しイベントが起きます。

命令取り出しイベントが起きた場合、PER イベント・マスクのビット 33 が 1 で、PSW 中の PER マスクが 1 であれば、PER 命令取り出しイベントが認識されます。

命令取り出しイベントは破棄されることがありますが、破棄されるかされないかは予測不能です。命令取り出しイベントが破棄されるのは、それが、非同期的な条件(入出力、外部、再始動、または抑制可能マシン・チェック条件)または停止機能のパフォーマンスが原因で割り込まれる割り込み可能命令について認識された唯一の PER イベントであり、かつ、その命令の操作単位の 1 つがまだ実行されないまま残っている場合です。

PER 命令取り出しイベントは、PER コードのビット 1 を 1 に設定することにより示されます。

記憶域変更

CPU が、論理アドレスまたは仮想アドレスを使用して、制御レジスター 10 および 11 に指定されている記憶域に対する格納アクセスを行い、アクセス例外が起きなければ、記憶域変更イベントが起きます。ただし、DAT がオンで、制御レジスター 9 中の記憶域変更スペース制御が 1 である場合は、記憶域変更イベントが起きるのは、DAT が記憶位置への参照を変換するために使用するアドレス・スペース制御エレメント中の記憶域変更イベント・ビットが 1 のときだけです。

記憶域の内容が変更されたと見なされるのは、CPU が、1 つのオペランドの全部または一部を指定の記憶域に格納させる命令を実行したときです。変更が生じると

見なされるのは、記憶保護例外を示す目的で格納が行われると見なされるときです。ただし、チャンネル・プログラムによるデータの格納については、認識はされません。(6-35ページの『アクセス例外の認識』を参照してください。) PER の目的から見た場合、格納される値が元の値と同じであっても、格納は記憶域変更と見なされません。

割り込みの実行過程で CPU が暗黙的に参照する記憶位置は、モニターされません。この種の記憶位置には、PSW および割り込みコードの記憶位置などがあります。ただし、命令によりこれらの位置に明示的に情報が格納されるときは、これらの位置もモニターされます。同様に、チャンネル・プログラムによるデータの格納の場合も、このモニターは行われません。リンケージ・スタックに作用する命令により格納される、リンケージ・スタック内の暗黙的な記憶位置は、モニターされます。

入出力命令は、格納が実際に行われる場合のみ、第 2 オペランド位置を変更するものと見なされます。

オペランドが実アドレスであるものとして指定されている命令には、記憶域変更は適用されません。したがって、記憶域変更は、INVALIDATE PAGE TABLE ENTRY、RESET REFERENCE BIT EXTENDED、SET STORAGE KEY EXTENDED、STORE USING REAL ADDRESS、TEST BLOCK、および TEST PENDING INTERRUPTION (有効アドレスが 0 の場合)には、適用されません。

記憶域変更イベントが起きた場合、PER イベント・マスクのビット 34 が 1 で、PSW 中の PER マスクが 1 であれば、PER 記憶域変更イベントが認識されます。PER 記憶域変更イベントを認識すべきかどうかを決定する際には、PER イベント・マスクのビット 36 は無視されます。

PER 記憶域変更イベントは、PER コードのビット 2 を 1 に設定し、PER コードのビット 4 を 0 に設定することにより、示されます。

実アドレスを使用した格納

実アドレスを使用した格納が起きるのは、STORE USING REAL ADDRESS 命令が実行されたときです。

実アドレスを使用した格納イベントと指定された記憶域の間には、関係はありません。

実アドレスを使用した格納イベントが起きた場合、PER イベント・マスクのビット 34 および 36 が 1 で、PSW

の中の PER マスクが 1 であれば、PER 実アドレスを使用した格納イベントが認識されます。

PER 実アドレスを使用した格納イベントは、PER コードのビット 2 および 4 を 1 に設定することにより示されます。

他の割り込み条件と同時発生時の PER イベントの指示

1 つの命令で、PER イベントと同時に、プログラム例外、モニター・イベント、スペース切り替えイベント、または監視プログラム呼び出し割り込みも発生した場合、PER イベントがどのように指示されるかについては、以下の規則が適用されます。

1. 命令取り出しイベントは、命令の実行が完了、中止、抑止、または無効化のいずれの状態になったかに関係なく示されます。ただし、命令の第 1、第 2、または第 3 のハーフワードにアクセス例外が適用された場合は、命令取り出しイベントが示されるかどうかは、予測不能です。同様に、アクセス例外が原因で、EXECUTE のターゲットの全部または一部へのアクセスが禁止された場合も、EXECUTE およびそのターゲットについて命令取り出しイベントが示されるかどうかは、予測不能です。
2. 操作の全部または一部が完了している場合は、プログラム例外、スペース切り替えイベント、またはモニター・イベントも同時に認識されているかどうかに関係なく、PER イベントが示されます。
3. 抑止または無効化された操作 (割り込み可能命令の場合は操作単位) については、ブランチ成功、記憶域変更、および実アドレスを使用した格納は示されません。
4. 命令の実行が中止された場合は、記憶域変更イベントが起きていれば、そのイベントが示されます。また、モデルによっては、命令が完了していればこのイベントが起きたであろうと想定される場合にも、このイベントを示すことがあります。この場合、結果フィールドの内容を変更するかどうかはオペランド値によって決まる場合でも、イベントが示されず。この定義の目的を達成するために、中止を許可する例外 (アドレッシング例外、記憶域保護例外、およびデータ例外) の発生は、結果の領域が変更されない場合でも、中止を引き起こすものと見なされます。
5. LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、RESUME PROGRAM、SET SYSTEM MASK、STORE THEN OR SYSTEM MASK、または SUPERVISOR CALL

の結果、何らかの PER 条件が発生すると同時に、PSW の活動化の直後に認識されるタイプの PSW フォーマット・エラーがある新 PSW が導入される場合は、割り込みコードは、その PER 条件と指定例外の両方を示します。LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、RESUME PROGRAM、または SUPERVISOR CALL の結果、後続の命令の実行の一部として認識されるタイプの PSW フォーマット・エラーが取り込まれる場合は、PSW は旧 PSW として格納され、指定例外は認識されません。

4-33 ページの図 4-8 は、PER イベントと他のプログラム割り込み条件が、どのようにして同時に表示されるかを、要約して示しています。

プログラミング上の注意:

1. 割り込み可能命令 MOVE LONG、TEST BLOCK、および COMPARE LOGICAL LONG の実行により、命令取り出しに関するイベントが起きることがあります。さらに、MOVE LONG は記憶域変更イベントを引き起こすことがあります。

このような命令の割り込みでは、同じ PER イベントが 2 回以上示されることがあります。したがって、プログラムにより PER データから余分なイベント指示を除去することが必要な場合があります。これらの命令の実行中に生じるイベントの指示については、以下の規則が適用されます。
 - a. 実行のために命令が取り出されると、それが初期実行か実行の再開に関係なく、命令取り出しイベントが示されます。ただし、示される PER イベントがこのイベントのみであり、割り込みが、非同期割り込み条件または停止機能のパフォーマンスが原因で生じたものであり、かつまだ実行されていない命令の操作単位が残っている場合は、このイベントは破棄される (つまり示されない) ことがあります。
 - b. 記憶域変更イベントが示されるのは、最後の実行開始から始まり、割り込みの前に転送された最後のバイトで終わる操作部分によって、指定された記憶域にデータが格納されたときだけです。完了前に割り込みが生じた場合、操作の再開後に再びこのイベントが起きるかどうかについての特別な指示は表示されません。指定記憶域が 1 バイトの記憶位置であるときは、MOVE LONG の実行時に 1 回だけ記憶域変更イベントが認識されます。
2. 割り込み可能命令の PER データから複数エントリを削除し、命令の個々の完全実行につきそれぞれ

1 つだけエントリーが取得されるようにするには、一般に、プログラムで以下のようなアクションを実行する必要があります。

a. PER アドレスが旧 PSW 中の命令アドレスと同じかどうか、および最後に実行された命令が割り込み可能かどうかを確認する。

b. 上記の両方の条件が満たされているときは、命令取り出しイベントを削除する。

c. 上記の両方の条件が満たされていて、イベントが記憶域変更イベントである場合は、残りの宛先オペランドの一部が指定記憶域内にあれば、そのイベントを削除する。

並行する条件	終了のタイプ	PER イベント			
		ブラ ンチ	命令取 り出し	記憶域 変更	実アドレスを 使用した格納
指定 PSW 内に奇数の命令アドレス 命令アクセス	S	No	No	No	No
	N または S	No	U	No	No
指定 EXECUTE ターゲット・アドレスが奇数 EXECUTE ターゲット・アクセス	S	No	U	No	-
	N または S	No	U	No	-
他の無効化	N	No	Yes	No ¹	-
他の抑止	S	No	Yes	No ¹	No
すべて中止	T	No	Yes	Yes ²	-
すべて完了	C	Yes	Yes	Yes	-

説明:

- 条件は適用されません。
- 1 割り込み可能命令の現行操作単位については、このタイプの PER イベントは示されませんが、割り込み可能命令の完了済み操作単位について認識されたこのタイプのイベントは示されます。
- 2 モデルによっては、実際にこのイベントが起きていなくても、実行が完了していればこのイベントが示されるものと想定される場合は、このイベントが示されることがあります。
- C 操作 (割り込み可能命令の場合は操作単位) は完了します。
- N 操作 (割り込み可能命令の場合は操作単位) は無効化されます。
- S 操作 (割り込み可能命令の場合は操作単位) は抑止されます。
- T 命令の実行は中止されます。
- Yes 該当の PER イベントが起きている場合、つまり、指定記憶域の内容が変更されたか、または、最初のバイトが指定記憶域内に含まれている命令が実行されようとした場合に、その PER イベントが他方のプログラム割り込み条件と共に示されます。
- No PER イベントは示されません。
- U PER イベントが示されるかどうかは、予測不能です。

図 4-8. 他の条件と同時発生の PER イベントの指示

タイミング

タイミング機構には、時間を測定するための 3 つの機構があります。それは、TOD タイマー、クロック・コンパレーター、および CPU タイマーです。TOD クロックには、TOD プログラマブル・レジスターが関連付けられます。

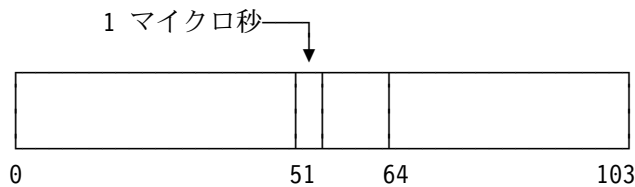
マルチプロセッシング構成では、すべての CPU が 1 つの TOD クロックを共用します。各 CPU は、それぞれ専用のクロック・コンパレーター、CPU タイマー、および TOD プログラマブル・レジスターを持っています。

TOD クロック

TOD (時刻) クロックは、日付と時刻を示すのに適した高分解能の実時間測定機構です。このクロックの周期は、約 143 年です。構成内のすべての CPU が、1 つの TOD クロックを共用します。

フォーマット

TOD クロックは 104 ビットのレジスターです。このレジスターは、次の図に示すフォーマットの 2 進カウンタです。



通常は、1 マイクロ秒ごとにビット位置 51 に 1 を加算することにより、TOD クロックの値が増分します。もっと分解能が高いか低いモデルでは、加算されるビット位置が異なりますが、加算の頻度は、クロックの進行速度が、1 マイクロ秒ごとにビット 51 に 1 を加算する場合と同じになるように設定されています。TOD クロックの分解能は、増分速度がモデルの命令実行速度に合うようになっています。

クロック値の増分により、ビット位置 0 からの繰り上がりが生じる場合は、繰り上がりは無視され、カウントは 0 から続けられます。これについてプログラムに警報が送られることはなく、オーバーフローの結果として何らかの割り込み条件が生成されることもありません。

クロックの動作は、システムの通常の活動やイベントの影響を受けることはありません。クロック値の増分は、PSW の待ち状態ビットが 1 かどうか、または、CPU が作動、ロード、停止、またはチェック停止のどの状態にあるかによって、変化することはありません。また、CPU リセット、初期 CPU リセット、クリア・リセット、または初期プログラム・ローディングが、クロックの動作に影響を与えることはありません。さらに、速度制御を設定すること、または初期マシン・ローディング操作によって、クロックの動作が影響を受けることもありません。モデルと構成に応じて、TOD クロックの電源は、CPU から独立して供給される場合とそうでない場合があります。

状態

TOD クロックの状態は、設定状態、非設定状態、停止状態、エラー状態、または非稼働状態として区別されません。状態によって、STORE CLOCK および STORE CLOCK EXTENDED の実行により設定される条件コードが決まります。設定状態または非設定状態にあるときは、クロックは増分し、この状態を走行中と呼びます。

非設定状態: クロックの電源がオンにされた時点では、クロックは 0 に設定され、非設定状態になります。非設定状態にあるときは、クロックは増分します。

クロックが非設定状態にあるときに、STORE CLOCK または STORE CLOCK EXTENDED を実行すると、条件コード 1 が設定され、走行中のクロックの現在値が格納されます。

停止状態: SET CLOCK が実行され、その結果としてクロックが設定されると、クロックは停止状態になります。この停止状態が生じるのは、SET CLOCK を実行したときに何の例外も発生せず、かつ、構成内で何らかの手动 TOD 制御が設定可能の位置に設定されているか、TOD クロック制御オーバーライド制御 (制御レジスター 14 のビット 42) が 1 である場合です。クロックは、設定状態、非設定状態、およびエラー状態から、停止状態にすることができます。停止状態にあるときは、クロックは増分しません。

クロックが停止状態にあるときに、STORE CLOCK または STORE CLOCK EXTENDED を実行すると、条件コード 3 が設定され、停止状態のクロックの値が格納されます。

設定状態: クロックは、停止状態からのみ設定状態に入ります。この状態変更は、クロックを最後に停止状態にした CPU の TOD クロック同期制御ビット (制御レジスター 0 のビット 34) により制御されます。このビットが 0 の場合は、クロックは、SET CLOCK の実行が完了した時点で停止状態になります。このビットが 1 の場合は、この CPU でこのビットが 0 に設定されるか、別の CPU がこのクロックを対象とした SET CLOCK 命令を実行するまで、クロックは停止状態のままになります。外部時刻参照 (ETR) がインストールされている場合は、ETR からの信号を使用して、停止状態から設定状態に設定することができます。

クロックの増分は、クロックが設定状態に入った後の最初のステップ・パルスから始まります。

クロックが設定状態にあるときに、STORE CLOCK または STORE CLOCK EXTENDED を実行すると、条件コード 0 が設定され、走行中のクロックの現在値が格納されます。

エラー状態: クロック値の妥当性に影響を与えたと思われる誤動作が検出されると、クロックはエラー状態に入ります。クロックがこの状態に入ることができるかどうかは、モデルによって異なります。クロックがエラー状態に入ると、構成内の各 CPU でそれぞれタイミング機構損傷マシン・チェック割り込み条件が生成されます。

クロックがエラー状態にあるときに、STORE CLOCK または STORE CLOCK EXTENDED を実行すると、条件コード 2 が設定され、0 の値が格納されます。

非稼働状態: クロックは、電源がオフになっているとき、または保守のために使用不可になっているときは、非稼働状態にあります。クロックがこの状態に入ることができるかどうかは、モデルによって異なります。クロックが非稼働状態に入ると、構成内の各 CPU でそれぞれタイミング機構損傷マシン・チェック割り込み条件が生成されます。

クロックが非稼働状態にあるときに、STORE CLOCK または STORE CLOCK EXTENDED を実行すると、条件コード 3 が設定され、0 が格納されます。

クロック状態の変化

SET CLOCK の実行により TOD クロックの値が変化した場合、またはクロックの状態が変化した場合、クロック・コンパレーターおよび CPU タイマーについて保留状態になっている割り込み条件が、その変化の後最大 1.048576 秒 (2²⁰ マイクロ秒) の間、認識されないことがあります。

TOD クロックに対する変更の結果、チャンネル・サブシステム・モニター機能操作の結果がどうなるかは、予測不能です。

クロックの設定と検査

構成内のいずれかの CPU の手動 TOD クロック制御が設定可能に位置するか、TOD クロック制御オーバーライド制御 (制御レジスター 14 のビット 42) が 1 であれば、SET CLOCK を実行することにより、クロックを特定の値に設定できます。SET CLOCK は、クロックのビットを、記憶域内のダブルワード・オペランドの対応するビット位置の内容に従って設定します。

クロックを設定すると、ビット位置 0 から、クロックの走行中に増分する右端の位置までのすべてのビット位置の値が置き換えられます。ただし、一部のモデルでは、指定された値のビット 52 またはそれより右側にある右端ビットは無視され、クロックの対応するビット位置には 0 が入れられます。また、クロックのビット位置 63 の右側の位置にも 0 が入れられます。

TOD クロックは、STORE CLOCK を実行することにより検査できます。これを実行すると、クロックのビット 0-63 が記憶域内の 8 バイト・オペランドに格納されます。また、STORE CLOCK EXTENDED を実行した場合は、クロックのビット 0-103 が記憶域内の 16 バイト・オペランドのバイト 1-13 に格納されます。STORE CLOCK EXTENDED は、記憶域オペランドの左端バイト (バイト 0) に 0 を格納し、TOD プログラムブル・レジスターのビット位置 16-31 から TOD プログラムブル・フィールドを取得して、それを記憶域オペランドのバイト位置 14 および 15 に格納します。STORE CLOCK EXTENDED が格納するオペランドのフォーマットは、以下のとおりです。

Zeros	TOD Clock	Programmable Field
0	8	112 127

将来の新しいモデルでは、STORE CLOCK EXTENDED は、TODクロックの左端拡張部分を記憶域オペランドのバイト位置 0 に格納するようになります。『プログラミング上の注意 13 (4-37ページ)』を参照してください。

同一構成内の異なる CPU で STORE CLOCK または STORE CLOCK EXTENDED を 2 回実行した場合は、クロックが走行中であれば、常に異なるクロック値が格納されます。クロックが停止している場合は、クロック値 (記憶域オペランドのビット 8-111) には、クロックの走行中に増分する右端のビット位置の右側の位置に 0 が格納されます。プログラムブル・フィールドは、クロックが停止していても引き続き格納されます。

STORE CLOCK または STORE CLOCK EXTENDED の実行順序をプログラムが確認できる場合は、走行中のクロックについて 1 つまたは複数の CPU でこれらの命令により格納される値は、常に、実行順序を正しく反映したものとなります。走行中のクロックの値が格納するときに、確実に固有の値が取得されるようにするために、増分する右端のビット位置の右側に 0 以外の値が格納されることがあります。STORE CLOCK EXTENDED が走行中のクロックの値を格納するときは、クロックのビット位置 64-103 (記憶域オペランドの

ビット位置 72-111) の値は、常に 0 以外の値になります。これで、STORE CLOCK EXTENDED により格納される値は、常に、STORE CLOCK が格納する値の右側に 0 を付加したものとは異なる値になります。

STORE CLOCK および STORE CLOCK EXTENDED の結果の固有性を確保し、発生順序を維持するために、STORE CLOCK が提供する 64 ビット値は右側に 40 個の 0 を付加することにより 104 ビットに拡張され、その結果、STORE CLOCK の値と STORE CLOCK EXTENDED のビット 8-111 は、どちらも 104 ビットの符号なし 2 進整数として扱われます。

複数の CPU がクロックにアクセスする構成では、SET CLOCK はインターロックされて、クロックの内容の全部が同時に更新されているように見えます。つまり、SET CLOCK が 2 つの CPU で同時に実行された場合、最終結果はそのいずれか一方の値になります。一方の CPU が SET CLOCK を実行し、もう一方が STORE CLOCK または STORE CLOCK EXTENDED を実行した場合は、STORE CLOCK または STORE CLOCK EXTENDED から得られる結果は、全部古い値のままになるか、全部新しい値になるかのいずれかです。一方の CPU が SET CLOCK を実行した場合、もう一方の CPU が実行した STORE CLOCK または STORE CLOCK EXTENDED 命令では、各 CPU の TOD クロック同期制御ビット (制御レジスター 0 のビット 34) が 0 であっても、クロックが停止状態にあるように見えることがあります。クロックは増分する前に設定状態に入るので、クロックが設定状態に入った後で最初に実行される STORE CLOCK または STORE CLOCK EXTENDED 命令では、依然として、SET CLOCK が導入した元の値が検出されることがあります。

TOD プログラマブル・レジスター

各 CPU は、TOD プログラマブル・レジスターを 1 つずつ備えています。このレジスターのビット 16-31 は、STORE CLOCK EXTENDED により TOD クロック値の右側に付加されるプログラマブル・フィールドです。このレジスターのフォーマットは以下のとおりです。

0000000000000000	Programmable Field
0	16 31

このレジスターは、SET CLOCK PROGRAMMABLE FIELD によりロードされます。レジスターの内容は、

初期 CPU リセットによりすべて 0 の値にリセットされます。

プログラミング上の注意:

1. クロックのビット 31 は、1.048576 秒ごとに増分します。したがって、アプリケーションによっては、クロックの左端 32 ビットを参照するだけで十分な精度が得られます。
2. クロック値に対応付けするカレンダー上の日付および時刻として標準の時刻起点を設定しておけば、システム間の通信が容易になります。この起点には、協定世界時 (Coordinated Universal Time: UTC) の 1900 年 1 月 1 日 午前 0 時を使用することをお勧めします。クロックの標準エポックはこの時刻から始まります。これは、TOD クロックを外部時刻参照 (ETR) と同期化するとき使用されるエポックでもあります。以前に使われていたグリニッジ標準時 (GMT) という用語は廃止され、現在はより厳密な UTC が使用されています。
3. クロック値を時刻および年月日の指示値として使用するプログラムは、そのプログラムの実行を制御するプログラミング・サポートとの間に整合性がなければなりません。プログラミング・サポートが標準エポックを使用している場合は、1972 年から 2041 年までの間は、クロックのビット 0 は 1 のままです。(ビット 0 は、1971 年 5 月 11 日の 11:56:53.685248 (UTC) にオンになりました。) たいていの場合は、ビット 0 が 1 かどうかを調べるだけで、クロック値が標準エポック内にあるかどうか分かります。
4. 現在の日付または時刻への変換、またはその逆の変換を行うときには、プログラミング・サポートは、時刻修正基準に従って「うるう秒」が挿入または削除されていることを考慮に入れる必要があります。TOD クロックが標準エポック内の特定時刻に正しく設定されているときは、UTC 時刻を算出するには、累積うるう秒の合計をクロック値から減算する必要があります。
5. クロック値は手動で設定するため精度に限界があるので、一般に、クロックの右端のビット位置 (1 秒未満の値を表す) は、正確な時刻を示すものとしては不適当です。ただし、経過時間の測定に使用する場合には、高精度の値が得られます。
6. 次の表は、TOD クロックの各ビット位置の値がどのような時間間隔で進められるかを示しています。この時間値は、該当ビットが 1 のときに表す加重時間値と考えることもできます。

TOD クロックの ビット	ステップの時間間隔			
	日	時間	分	秒
51				0.000 001
47				0.000 016
43				0.000 256
39				0.004 096
35				0.065 536
31				1.048 576
27				16.777 216
23			4	28.435 456
19		1	11	34.967 296
15		19	5	19.476 736
11		12 17	25	11.627 776
7		203 14	43	6.044 416
3		3257 19	29	36.710 656

7. 次の表は、幾つかの日付について、UTC 時刻 00:00:00 (0 am) の TOD クロック設定を示しています。設定されている時刻は、1900 年 1 月 1 日と 1972 年 1 月 1 日、そして、2000 年 11 月までに発生した 22 個のうるう秒が、それぞれ発生した直後の時刻です。これらのうるう秒は、それぞれ、リストされている日の前の日の 23:59:60 UTC から、リストされている日の 00:00:00 UTC までの間の UTC 時間枠の中に挿入されています。

年	月	日	うるう秒	クロック設定 (16 進数)
1900	1	1		0000 0000 0000 0000
1972	1	1		8126 D60E 4600 0000
1972	7	1	1	820B A981 1E24 0000
1973	1	1	2	82F3 00AE E248 0000
1974	1	1	3	84BD E971 146C 0000
1975	1	1	4	8688 D233 4690 0000
1976	1	1	5	8853 BAF5 78B4 0000
1977	1	1	6	8A1F E595 20D8 0000
1978	1	1	7	8BEA CE57 52FC 0000
1979	1	1	8	8DB5 B719 8520 0000
1980	1	1	9	8F80 9FDB B744 0000
1981	7	1	10	9230 5C0F CD68 0000
1982	7	1	11	93FB 44D1 FF8C 0000
1983	7	1	12	95C6 2D94 31B0 0000
1985	7	1	13	995D 40F5 17D4 0000
1988	1	1	14	9DDA 69A5 57F8 0000
1990	1	1	15	A171 7D06 3E1C 0000
1991	1	1	16	A33C 65C8 7040 0000
1992	7	1	17	A5EC 21FC 8664 0000
1993	7	1	18	A7B7 0ABE B888 0000
1994	7	1	19	A981 F380 EAAC 0000
1996	1	1	20	AC34 336F ECD0 0000
1997	7	1	21	AEE3 EFA4 02F4 0000
1999	1	1	22	B196 2F93 0518 0000

8. TOD クロックのビット位置 63 が使用されている場合、そのビットが進められる時間間隔は、 2^{-12} マ

イクロ秒、つまり約 244 ピコ秒です。この値をクロック単位と呼びます。

次の表は、各種の時間間隔を、16 進数表記のクロック単位数で示しています。

間隔	クロック単位 (16 進数)
1 マイクロ秒	1000
1 ミリ秒	3E 8000
1 秒	F424 0000
1 分	39 3870 0000
1 時間	D69 3A40 0000
1 日	1 41DD 7600 0000
365 日	1CA E8C1 3E00 0000
366 日	1CC 2A9E B400 0000
1,461 日*	72C E4E2 6E00 0000

* うるう年を含む 4 年間の日数。
1900 年はうるう年ではなかったことに注意してください。
したがって、1900 年から始まる 4 年間の日数は、1,460 日です。

9. 注 6 から 8 までの表は、STORE CLOCK により格納された値を調べるときに便利です。STORE CLOCK EXTENDED 命令により格納された値を調べるときに使用する表は、その命令に関する説明の終わりの『プログラミング上の注意』の中に収められています。
10. マルチプロセッシング構成では、CPU タイマー、クロック・コンパレータ、および TOD クロック同期チェックの割り込み条件が確実に CPU により認識されるようにするために、TOD クロックが設定されて走行を始めてから 2^{20} マイクロ秒間 (1.048576 秒間) は、プログラムの活動を遅らせるようにしてください。
11. STORE CLOCK および STORE CLOCK EXTENDED 結果の順序付けに関する規則があるため、TOD クロックのビット位置 63 より右側にあるビット位置が増分するモデルでは、STORE CLOCK の実行速度が STORE CLOCK EXTENDED よりかなり遅くなることがあります。
12. TOD プログラマブル・フィールドに構成識別を含めることにより、TOD クロック値の固有性を、異なる構成内にある複数のプロセッサにまで拡大して適用することができます。
13. 将来の新モデルでは、TOD クロックのビット位置 0 からの繰り上がりを使用して、追加の 8 ビット 2 進カウンターが増分するようになります。STORE CLOCK EXTENDED は、このカウンターを、記憶域オペランドのバイト位置 0 に格納します。SET CLOCK のバリエーションの 1 つとして、TOD ク

ロックに加えて、このカウンターを設定するようになります。また、SET CLOCK COMPARATOR および STORE CLOCK COMPARATOR のバリエーションは、クロック・コンパレーターの左側の比較可能バイトを操作するようになります。これらのアクションにより、ビット 0 からの繰り上がりが原因で生じる 143 年間という限界に達した後も、引き続き標準エポック内で TOD クロックによって時間を測定でき、クロック・コンパレーターを継続して使用できるようになります。16 バイトの STORE CLOCK EXTENDED オペランドを処理するプログラムでは、このような将来の発展を考慮に入れることをお勧めします。

TOD クロックの同期化

外部時刻参照 (ETR) がインストールされている場合は、以下の機能が提供されます。

- クロックが停止状態にあり、TOD クロック同期制御ビット (制御レジスター 0 のビット 34) が 1 に設定されている場合は、ETR 信号が発生すると、クロックは設定状態になり、増分が開始されます。
- TOD クロックと ETR のステップ速度が同期化されます。
- 設定状態にあるクロックのビット 32 から、増分する最右端のビットまでが、ETR の同じビットと比較されます。両者が一致していない場合は、外部損傷マシン・チェック割り込み条件により不一致条件が通知されます。このマシン・チェック割り込み条件は、不一致条件が生じた後、最大 1.048576 秒間 (2^{20} マイクロ秒間) 認識されないことがあります。

プログラミング上の注意:

1. TOD クロック同期化では、TOD クロックのビット 32 から、増分する最右端のビットまでが、同期化され、検査されるだけです。TOD クロックのビット 0-31 は、ETR の同じビットと違っていることがあります。
2. ETR がインストールされている場合は、SET CLOCK は、TOD クロックのビット 32 から増分する最右端のビットまでに、すべて 0 を入れる必要があります。

クロック・コンパレーター

クロック・コンパレーターは、TOD クロック値がプログラムにより指定された値を超えたときに、割り込みを発生させる手段を提供します。

複数の CPU を持つ構成では、各 CPU がそれぞれクロック・コンパレーターを 1 つずつ持っています。

クロック・コンパレーターのフォーマットは、TOD クロックのビット 0-63 と同じです。基本形式では、クロック・コンパレーターはビット 0-47 から成り、これらのビットが TOD クロックの対応するビットと比較されます。一部のモデルでは、48 個より多くのビットを使用して、もっと高い分解能を達成しています。クロック・コンパレーターが提供する位置のビットが、クロックの対応するビットと比較されます。クロックの分解能がクロック・コンパレーターの分解能より低い場合は、クロック・コンパレーターの内容は、STORE CLOCK の実行により格納されるクロック値と比較されます。

クロック・コンパレーターは、割り込みコード 1004 (16 進数) を伴う外部割り込みを引き起こします。次のいずれかの条件が生じると、クロック・コンパレーター割り込み要求が出ます。

1. TOD クロックが走行中で、クロック・コンパレーターの値がクロックの比較対象部分の値より小さく、両方の値が符号なしの 2 進整数と見なされる。この場合、比較は符号なしの 2 進数演算の規則に従って行われます。
2. TOD クロックが、エラー状態または非稼働状態にある。

クロック・コンパレーターの値が TOD クロックの値と等しいかそれより大きくなった場合、または、TOD クロックの値がクロック・コンパレーターの値より小さくなった場合は、クロック・コンパレーター割り込み要求は保留状態から解放されます。TOD クロックの値がクロック・コンパレーターの値より小さくなるのは、TOD クロックが設定されるか、または 0 に循環したときです。

クロック・コンパレーターは、STORE CLOCK COMPARATOR を実行して検査でき、また、SET CLOCK COMPARATOR 命令を実行して指定の値に設定できます。

クロック・コンパレーターの内容は、初期 CPU リセットにより 0 に初期化されます。

プログラミング上の注意:

1. クロック・コンパレーターの値が TOD クロックの値より小さいか、または TOD クロックがエラー状態または非稼働状態にある間は、クロック・コンパレーターの割り込み要求は持続します。したがって、クロック・コンパレーターの外部割り込みが発生した後、CPU が再び外部割り込みに対して使用可能にされる前に、次のいずれかのアクションを実行する必要があります。つまり、クロック・コンパレーターの値を置き換えるか、TOD クロックを設定するか、TOD クロックを 0 に循環させるか、または、クロック・コンパレーター・サブクラス・マスクを 0 に設定する必要があります。さもないと、外部割り込みのループが発生します。
2. STORE CLOCK または STORE CLOCK EXTENDED 命令は、クロック・コンパレーター割り込みに対して CPU が使用可能にされていても、クロック・コンパレーターの値より大きい値を格納することがあります。これは、割り込みの実行が開始されてからクロック値がアクセスされるまでの間に、TOD クロックが 1 回以上増分することがあるからです。このような場合は、STORE CLOCK または STORE CLOCK EXTENDED の実行が完了した時点で割り込みが発生します。

CPU タイマー

CPU タイマーは、経過した CPU 時間を測定し、指定された時間が経過したときに割り込みを発生させる手段を提供します。

複数の CPU を持つ構成では、各 CPU がそれぞれ CPU タイマーを 1 つずつ持っています。

CPU タイマーは、TOD クロックのビット 0-63 と同じフォーマットを持つ 2 進カウンターです。ただし、ビット 0 は符号と見なされます。基本的には、CPU タイマーは、1 マイクロ秒ごとにビット 51 から 1 を減算することにより、値が減少します。もっと分解能が高いか低いモデルでは、減少するビット位置が異なりますが、CPU タイマーの減少の速度は、1 マイクロ秒ごとにビット位置 51 から 1 を減算する場合と同じになるように設定されています。CPU タイマーの分解能は、ステップ速度がモデルの命令実行速度につり合うようになっています。

CPU タイマー値が負 (CPU タイマーのビット 0 が 1) のときは、CPU タイマーは、割り込みコード 1005 (16 進数) を伴う外部割り込みを要求します。CPU タイマー

値が負ではない値に変化すると、要求は保留状態から解放されます。

CPU タイマーと TOD クロックの両方とも走行中のときは、ステップ速度は、両方が同じ速度でステップするように同期化されます。通常は、CPU タイマーの減少は、並行して実行されている入出力活動の影響を受けません。しかし、一部のモデルでは、入出力活動が極度に活発化した場合や、その他類似の妨害状況が生じている場合は、CPU タイマーが停止することがあります。このような場合は、CPU タイマーが記録した時間の方が、CPU タイマーが止まらずにステップした場合に記録された場合の時間より、プログラムが使用した CPU 時間を正確に表しています。

CPU タイマーの値が減少するのは、CPU が作動状態またはロード状態にあるときです。手動速度制御が命令ステップに設定されているときは、CPU タイマーは、CPU が実際に操作単位を実行している間のみ減少します。ただし、モデルによっては、TOD クロックがエラー状態、停止状態、または非稼働状態にあるときは、CPU タイマーは減少する場合としない場合があります。

CPU がチェック停止状態にあるときは、CPU タイマーが減少するかどうかは、モデルによって異なります。

CPU タイマーは、STORE CPU TIMER 命令を実行することにより検査でき、SET CPU TIMER 命令を実行することにより指定の値に設定できます。

CPU タイマーは、処理 CPU リセットにより 0 に設定されます。

プログラミング上の注意:

1. CPU タイマーを特定のプログラムと関連付けて使用することにより、CPU 実行時間を測定すると共に、CPU での特定時間間隔の終わりを通知することができます。
2. 1 つの命令シーケンスの実行に要した時間の測定値は、入出力干渉、ページの可用性、および命令再試行などの影響によって異なります。したがって、同じインストール・システムで同じシーケンスを何度か測定した場合に、そのたびに測定値が異なることがあります。
3. CPU タイマーが正の値に設定されると、CPU タイマー割り込みは保留状態から解放されるので、不必要な割り込みが排除されます。このような割り込みが起きるのは、例えば、古い値が格納されてから新しい値が設定されるまでの間に、CPU タイマー割

り込みに対して CPU が使用不可にされ、CPU タイマー値割り込みが正から負に転じたときです。

4. CPU タイマーが負のときは (単に CPU タイマーが正から負に転じたときだけでなく)、常に CPU タイマー割り込み要求が出るので、CPU タイマーをある値に設定するときに、その値が正であることを事前に確認する必要はありません。

例えば、CPU タイマーにより時間が測定されているプログラムが、CPU タイマー以外の原因で割り込まれたときに、新しい PSW では外部割り込みが禁止されており、そして、STORE CPU TIMER により CPU タイマー値が保管されるとします。この場合に、割り込み以降に CPU タイマーが正から負に転じていれば、格納される値は負です。その後、計測対象のプログラムの実行が継続され、SET CPU TIMER により、CPU タイマーが保管されていた値に設定されたとします。この場合、保管されていた値が負であれば、再び外部割り込みが使用可能にされた直後に、CPU タイマー割り込みが発生します。

CPU タイマー割り込み要求がいつまでも消えないときは、CPU タイマーによる外部割り込みが起きた後で、再び外部割り込みに対して CPU を使用可能にする前に、CPU タイマーの値を正の値に循環させるか、CPU タイマー・サブクラス・マスクを 0 に設定する必要があります。さもないと、外部割り込みのループが発生します。

5. 割り込みに対して CPU が使用可能にされていても、STORE CPU TIMER 命令により負の値が格納されることがあります。これは、命令の実行が開始されてから CPU タイマーがアクセスされるまでの間に、CPU タイマーの値が 1 回以上減少することがあるからです。このような場合は、STORE CPU TIMER の実行が完了した時点で割り込みが発生します。

外部開始機能

リセット

以下の 5 つのリセット機能が提供されています。

- CPU リセット
- 初期 CPU リセット
- サブシステム・リセット
- クリア・リセット
- パワーオン・リセット

CPU リセットでは、装置チェック表示ライトを消去できるほか、CPU 状態に予測不能な結果が生じている場合に、情報の破壊を最小限に抑えてそのような結果を消去することができます。これは、特に、操作の分析や再開のために CPU 状態を保存しておく必要があるときに、チェック条件を消去するために使用されます。ロード通常キーの活動化が原因で CPU リセットが生じた場合は、アーキテクチャー・モードが ESA/390 モードに設定されます。

初期 CPU リセットでは、CPU リセットに加えて、現 PSW、CPU タイマー、クロック・コンバーター、プレフィックス、制御レジスター、浮動小数点制御レジスター、および TOD プログラマブル・レジスターの初期化もできます。ロード通常キーの活動化が原因で初期 CPU リセットが生じた場合は、アーキテクチャー・モードが ESA/390 モードに設定されます。

サブシステム・リセットでは、入出力システム・リセット機能呼び出せるほか、浮動割り込み条件を消去することができます。

クリア・リセットでは、CPU リセットとサブシステム・リセットができるほか、構成内のすべての CPU の中のすべての記憶位置およびレジスター (TOD クロックを除く) を消去または初期化できます。このような消去は、プログラムのデバッグやユーザーのプライバシーを確保したい場合に役立ちます。また、クリア・リセットでは、PERFORM LOCKED OPERATION 命令で使用されるすべてのロックが解除されます。さらに、クリア・リセットは、アーキテクチャー・モードを ESA/390 モードに設定します。この消去により、制御プログラムがアドレス不能ページの内容を保持するために使用している直接アクセス記憶装置など、外部記憶装置が影響を受けることはありません。

CPU パワーオン・リセットでは、初期 CPU リセットが実行されるほか、汎用レジスター、アクセス・レジスター、および浮動小数点レジスターの内容が、有効な検査ブロック・コードを伴って 0 にクリアされます。

PERFORM LOCKED OPERATION で使用されてこの CPU に関連付けられているロックは、すでに電源がオンになっている CPU により保持されている場合以外は、解除されます。TOD クロック、主記憶機構、およびチャンネル・サブシステムのパワーオン・リセット・シーケンスは、CPU パワーオン・シーケンスに含めることも、それぞれ単独で開始することもできます。CPU パワーオン・リセットにより構成が確立される場合は、アーキテクチャー・モードは ESA/390 モードに設定されます。それ以外の場合は、アーキテクチャー・モード

は、すでに構成に含まれている CPU のアーキテクチャ・モードに設定されます。

CPU リセット、初期 CPU リセット、サブシステム・リセット、およびクリア・リセットは、オペレーター機能を使用して手動で開始できます (第12章、『オペレーター機能』を参照)。初期 CPU リセットは、初期プログラム・ローディング機能の一部です。図4-9 は、この 4

通りのリセットを手動で開始する方法を要約したものです。パワーオン・リセットは、電源をオンにする動作の一環として行われます。リセット・アクションは、4-42 ページの図4-10 に要約して示してあります。SIGNAL PROCESSOR 命令でどのリセットを実行できるかについては、4-48 ページの『プロセッサ信号指令』を参照してください。

活動化するキー	実行された機能 ¹		
	キーが活動化された CPU	構成内の他の CPU	構成内のその他の部分
システム・リセット通常キー	CPU リセット	CPU リセット	サブシステム・リセット
システム・リセット消去キー	クリア・リセット ²	クリア・リセット ²	クリア・リセット ³
ロード通常キー	初期 CPU リセット、続いて IPL	CPU リセット	サブシステム・リセット
ロード消去キー	クリア・リセット ² 、続いて IPL	クリア・リセット ²	クリア・リセット ³
説明: ¹ システム・リセット・キーまたはロード・キーを活動化すると、入出力装置、記憶装置、および他の CPU との接続などの構成が変換することがあります。 ² このリセットの CPU 部分のみが適用されます。 ³ このリセットの非 CPU 部分のみが適用されます。			

図 4-9. 手動によるリセット開始

影響を受ける領域	リセット機能				
	サブシステム・リセット	CPUリセット	初期CPUリセット	クリア・リセット	パワー・オン・リセット
CPU	U	S	S ¹	S ¹	S
PSW	U	U/V#	C* ¹	C* ¹	C* ¹
プレフィックス	U	U/V	C	C	C
CPU タイマー	U	U/V	C	C	C
クロック・コンパレーター	U	U/V	C	C	C
TOD プログラマブル・レジスター	U	U/V	C	C	C
制御レジスター	U	U/V	I	I	I
浮動小数点制御レジスター	U	U/V	C	C	C
アクセス・レジスター	U	U/V	U/V	C	C
汎用レジスター	U	U/V	U/V	C	C
浮動小数点レジスター	U	U/V	U/V	C	C
記憶キー	U	U	U	C	C ²
揮発性主記憶機構	U	U	U	C	C ²
不揮発性主記憶機構	U	U	U	C	U
拡張記憶機構	U ³	U ³	U ³	U ³	C ²
TOD クロック	U ⁴	U ⁴	U ⁴	U ⁴	T ²
浮動割り込み条件	C	U	U	C	C ²
入出力システム	R	U	U	R	R ⁵
PERFORM LOCKED OPERATION ロック	U	U	U	RC	RP

説明:

アーキテクチャー・モードが z/Architecture から ESA/390 に変更されると、PSW は未変更のままになります。代わりに、PSW は 16 バイトから 8 バイトに変更され、この 8 バイト PSW のビットは以下のように設定されます。ビット 0-11 および 13-32 は 16 バイト PSW の同じビットと同じ値に設定され、ビット 12 は 1 に設定され、ビット 33-63 は 16 バイト PSW のビット 97-127 と同じ値に設定されます。PSW ビット 31 が 1 の場合は、PSW は ESA/390 モードでは無効です。

* アーキテクチャー・モードが ESA/390 である場合は、PSW の内容を 0 にクリアすると、PSW は無効になります。

1 この CPU でリセット機能の後に IPL シーケンスが続く場合は、この CPU は必ずしも停止状態にはならず、PSW は必ずしも 0 にリセットされません。

2 これらの装置がそれぞれ単独で電源を供給されている場合は、該当装置の電源がオンになったときのみ、このアクションが実行されます。

3 リセット機能の実行中に、拡張記憶機構を変更しようとしてアクセスすると、拡張記憶機構内の 4K バイト・ブロックの内容が予測不能になることがあります。拡張記憶機構を検査するためのアクセスは、拡張記憶機構の内容に影響を与えることはありません。

4 リセット機能の実行中に STORE CLOCK を使用して TOD クロックにアクセスした場合に、TOD クロックが影響を受けることはありません。

図 4-10 (1/2). リセット・アクションの要約

説明 (続き):

- 5 チャンネル・サブシステムが、単独で電源を供給されるか、またはそれぞれ単独で電源を供給される複数のエレメントから成っている場合は、リセット・アクションが適用されるのは、電源がオンになっているエレメントに関連したパス上にあるサブチャンネル、チャンネル・パス、および入出力制御装置と入出力装置のみです。
- C 条件または内容がクリアされます。影響を受ける領域がフィールドである場合は、内容は、有効な検査ブロック・コードを伴って、0 に設定されます。
- I 状態または内容が初期化されます。影響を受ける領域がフィールドである場合は、内容は、有効な検査ブロック・コードを伴って、初期値に設定されます。
- R チャンネル・サブシステムで、入出力システム・リセットが行われます。このリセットの一環として、チャンネル・サブシステムに接続しているすべての入出力制御装置および入出力装置に、システム・リセットが通知されます。
- RC 構成内のすべてのロックが解除されます。
- RP すでに電源がオンになっている CPU が保持しているロックを除き、構成内のすべてのロックが解除されます。
- S CPU はリセットされ、現在行われている操作があれば中止され、ALB および TLB からエンタリーが消去され、CPU 内の割り込み条件はクリアされ、CPU は停止状態になります。リセットの結果として停止状態になっているときは、開始機能の実行結果は予測不能です。システム・リセット消去キー、ロード通常キー、またはロード消去キーにより、または、構成を確立する CPU パワーオン・リセットにより開始されたリセットの場合は、アーキテクチャー・モードが ESA/390 モードに設定されます。さもないと、アーキテクチャー・モードは変更されませんが、パワーオン・リセットでは、すでに構成内にある CPU のモードに設定されます。
- T TOD クロックが 0 に初期化され、有効化されます。そして、TOD クロックは非設定状態になります。
- U 状態、条件、またはフィールド内容は変化しません。ただし、リセットの時点で、状態、条件、またはフィールド内容を変更する操作が進行中であった場合は、結果は予測不能です。
- U/V リセットの実行時点でフィールドが変更されなければ、内容は未変更のままです。ただし、一部のモデルでは、検査ブロック・コードが有効化されることがあります。リセットの時点で、フィールドの内容を変更する操作が進行中であった場合は、結果は予測不能です。

図 4-10 (2/2). リセット・アクションの要約

CPU リセット

CPU リセットでは以下のアクションが実行されます。

1. 現行の命令または他の処理シーケンス (割り込みなど) の実行は中止され、すべてのプログラム割り込み条件および監視プログラム呼び出し割り込み条件が消去されます。
2. CPU にとってローカルな保留中の外部割り込み条件が消去されます。浮動中の外部割り込み条件は、消去されません。
3. CPU にとってローカルな保留中のすべてのマシン・チェック割り込み条件およびエラー指示、およびすべてのチェック停止状態は、消去されます。浮動中のマシン・チェック割り込み条件は、消去されません。構成内のすべての CPU に報告されたマシン・チェック条件のうち、特定の CPU にとって保留状態にあるものを、その CPU にとってローカルなマシン・チェック条件と言います。
4. 事前取り出しされた命令またはオペランドのすべてのコピーが、消去されます。さらに、現行チェックポイント間隔内での命令実行により格納されることになっていた結果は、すべて消去されます。
5. ART 索引バッファおよび変換索引バッファのエンタリーが、消去されます。
6. 構成内のいずれかの CPU でロード通常キーを活性化したために生じたリセットの場合は、その CPU (および構成内の他のすべての CPU) のアーキテクチャー・モードが ESA/390 モードに設定されます。これにより、z/Architecture モードから

ESA/390 モードに変更された場合は、現 PSW は 16 バイトから 8 バイトに変更され、この 8 バイト PSW のビットは以下のように設定されます。ビット 0-11 および 13-32 は 16 バイト PSW の同じビットと同じ値に設定され、ビット 12 は 1 に設定され、ビット 33-63 は 16 バイト PSW のビット 97-127 に設定されます。

7. 上記のアクション 1-6 が完了すると、CPU は停止状態になります。その CPU でのリセット機能の後に IPL シーケンスが続く場合は、CPU は、リセット機能が完了した時点でロード状態に入り、必ずしもリセット操作の実行中に停止状態に入るとは限りません。

CPU にとって外部にあるレジスター、記憶機構内容、および各種条件の状態は、CPU リセットにより変更されません。ただし、リセットの時点で、レジスター、記憶位置、または状態の内容を変更する操作が進行中であつた場合は、リセット後の内容は予測不能です。

PERFORM LOCKED OPERATION の実行時に CPU が保持していたロックは、CPU リセットでは解除されません。

CPU が入出力命令または入出力割り込みを実行しているときに、CPU 内のリセット機能が開始された場合は、CPU とチャンネル・サブシステム間の現行操作は完了する場合も完了しない場合もあり、関連のチャンネル・サブシステム機能の結果の状態は予測不能になることがあります。

プログラミング上の注意:

1. CPU が停止状態にあるときは、状態、条件、またはフィールド内容を変更するようなほとんどの操作は、実行不能です。しかし、一部のプロセッサ信号命令や一部のオペレーター機能は、これらのフィールドを変更することがあります。CPU リセットが発行されたときにフィールドが失われる可能性をなくするためには、CPU を停止し、進行中のオペレーター機能がない状態にしておく必要があります。
2. アーキテクチャー・モードが ESA/390 モードに変更され、現 PSW のビット 31 が 1 のときは、その PSW は無効です。

初期 CPU リセット

初期 CPU リセットは、CPU リセットの機能に、以下の消去機能および初期化機能を加えたものです。

1. ロード通常キーの活動化が原因でリセットが生じた場合は、該当の CPU (および構成内の他のすべての CPU) のアーキテクチャー・モードが、ESA/390 モードに設定されます。
2. 現 PSW、プレフィックス、CPU タイマー、クロック・コンパレーター、および TOD プログラマブル・レジスターの内容が、0 に設定されます。CPU でのリセット機能の後に IPL シーケンスが続く場合は、必ずしも PSW の内容は 0 に設定されません。
3. 制御レジスターの内容が、それぞれの初期 z/Architecture 値に設定されます。CPU が ESA/390 モードか z/Architecture モードかに関係なく、制御レジスターの 64 ビットのすべてが設定されます。
4. 浮動小数点制御レジスターの内容が 0 に設定されません。

これらの消去機能および初期化機能には、妥当性検査が含まれます。

ESA/390 アーキテクチャー・モードでは、PSW ビット 12 は 1 でなければならないので、CPU がこのモードになっているときに現 PSW を 0 に設定すると、PSW は無効になります。したがって、この場合に、リセットの後で、最初に新規 PSW の導入を行わずに CPU を作動状態にすると、指定例外が認識されます。

サブシステム・リセット

サブシステム・リセットは、構成内の CPU 以外のエレメントに対してのみ働きます。このリセットでは以下のアクションが実行されます。

1. 入出力システム・リセットは、チャンネル・サブシステムにより行われます (17-10ページの『入出力システム・リセット』を参照)。
2. 構成内のすべての浮動割り込み条件が消去されます。

入出力システム・リセットの一環として、保留中の割り込み条件が消去され、チャンネル・サブシステムに接続しているすべての制御装置および入出力装置にシステム・リセットが通知されます (17-10ページの『入出力システム・リセット』を参照)。入出力制御装置および入出力装置に対するシステム・リセットの影響、およびその結果の制御装置および入出力装置の状態については、該当する制御装置または入出力装置に関するシステム・ライ

ブラリー資料を参照してください。一般に、システム・リセットでは、共用の制御装置または入出力装置内の機能のうち、リセット信号を出した特定のチャンネル・パスに関連した機能のみがリセットされます。

クリア・リセット

クリア・リセットは、初期 CPU リセット機能に、以下のアクションを起こす初期化機能を加えたものです。

1. 構成内のすべての CPU のアーキテクチャー・モードが、ESA/390 モードに設定されます。
2. 構成内のすべての CPU のアクセス・レジスター、汎用レジスター、および浮動小数点レジスターが、0 に設定されます。リセット操作の終わりに CPU が ESA/390 アーキテクチャー・モードになっても、汎用レジスターの 64 ビットのすべてが設定されます。
3. 構成内の主記憶機構の内容および関連の記憶キーが、有効な検査ブロック・コードを伴って、0 に設定されます。
4. **PERFORM LOCKED OPERATION** 命令を実行するときに構成内の CPU により使用されたロックが、解除されます。
5. サブシステム・リセットが行われます。

レジスターの設定および記憶機構と記憶キーの消去には、妥当性検査が含まれます。

プログラミング上の注意:

1. システム・リセット通常キーの活動化、**SIGNAL PROCESSOR CPU** リセットまたは初期 CPU リセット指令の実行では、アーキテクチャー・モードは変更されません。構成内のすべての CPU は、常に同じアーキテクチャー・モードになっています。
2. 変更してはならないフィールドの内容に CPU リセット操作が影響を与えないようにするには、リセットの時点で、CPU が何らかの命令を実行してはならず、また CPU はすべての割り込みに対して使用不可にされていなければなりません。CPU タイマーの動作およびマシン・チェック割り込み発生の可能性を除き、その他のすべての CPU 活動は、CPU を待ち状態にし、入出力割り込みおよび外部割り込みに対して CPU を使用不可にすることにより、停止できます。CPU タイマーが更新される時、またはマシン・チェック割り込みが生じるときに、リセットが行われる可能性を回避するには、CPU は停止状態になっていなければなりません。

3. CPU リセット、初期 CPU リセット、サブシステム・リセット、およびクリア・リセットは、TOD クロックの値と状態には影響を与えません。
4. CPU がどのような条件下でチェック停止状態に入るかは、モデルによって異なります。また、これらの条件には、現行操作の完了を妨げるような誤動作も含まれます。したがって、CPU がチェック停止状態にあるときに CPU リセットまたは初期 CPU リセットが実行された場合は、PSW、レジスター、および記憶位置の内容は、エラーの発生時にアクセスされていた記憶キーおよび記憶位置も含めて、予測不能の値をとることになります。また、場合によっては、リセットによりチェック停止状態がクリアされた後も、これらの内容がエラーのままになることもあります。このような場合は、エラーをクリアするためにクリア・リセットが必要になります。

パワーオン・リセット

マシンの特定コンポーネントに対するパワーオン・リセット機能は、そのコンポーネントのパワーオン・シーケンスの一環として実行されます。

TOD クロック、主記憶機構、拡張記憶機構、およびチャンネル・サブシステムのパワーオン・シーケンスは、CPU パワーオン・シーケンスに含めることも、それぞれ単独で開始することもできます。以下のセクションでは、CPU、TOD クロック、主記憶機構、拡張記憶機構、およびチャンネル・サブシステムのパワーオン・リセットについて説明します。チャンネル・サブシステム、制御装置、および入出力装置については、第17章、『入出力サポート機能』および該当のシステム・ライブラリー資料も参照してください。

CPU のパワーオン・リセット: パワーオン・リセットを行うと、初期 CPU リセットが行われます。チャンネル・サブシステムでは、入出力システム・リセットが行われる場合と行われない場合があります。汎用レジスター、アクセス・レジスター、および浮動小数点レジスターの内容が、有効な検査ブロック・コードを伴って、0 にクリアされます。**PERFORM LOCKED OPERATION** で使用されてこの CPU に関連付けられているロックは、すでに電源がオンになっている CPU により保持されている場合以外は、解除されます。リセットに伴って構成が確立される場合は、CPU は ESA/390 モードにされます。それ以外の場合は、CPU は、すでに構成内にある CPU アーキテクチャー・モードにされます。

TOD クロックのパワーオン・リセット: パワーオン・リセットにより、TOD クロックの値が有効な検査ブロック・コードを伴って 0 に設定され、クロックは非設定状態に入ります。

主記憶機構のパワーオン・リセット: 揮発性主記憶機構 (電源が切れたときに内容が保存されないもの) および記憶キーの場合は、パワーオン・リセットにより、これらのフィールドに、有効な検査ブロック・コードを伴って 0 が入れられます。不揮発性主記憶機構の場合は、検査ブロック・コードも含めて内容は変更されません。

拡張記憶機構のパワーオン・リセット: 拡張記憶機構の内容は、有効な検査ブロック・コードを伴って、0 にクリアされます。

チャンネル・サブシステムのパワーオン・リセット: チャンネル・サブシステムのパワーオン・リセットでは、チャンネル・サブシステム内で入出力システム・リセットが行われます。(17-10ページの『入出力システム・リセット』を参照してください。)

初期プログラム・ローディング

初期プログラム・ローディング (IPL) は、手動により指定の装置からプログラムを読み取り、そのプログラムの実行を開始するための手段を提供します。

モデルによっては、IPL に関連した追加の制御機能および指示機能を備えているものもあります。この追加情報については、該当モデル用のシステム・ライブラリー資料を参照してください。

手動で IPL を開始するには、ロード装置アドレス制御を、入力装置を指示する 4 桁番号に設定し、次に、特定 CPU についてロード消去キーまたはロード通常キーを活動化します。以下の説明では、「この CPU」という用語は、構成内の CPU のうち、ロード消去キーまたはロード通常キーが活動化された CPU を指します。

ロード消去キーを活動化すると、構成に対してクリア・リセットが行われます。

ロード通常キーを活動化すると、この CPU で初期 CPU リセットが行われ、構成内の他のすべての CPU に CPU リセットが伝搬され、構成の他の部分についてはサブシステム・リセットが行われます。

ロード消去キーまたはロード通常キーを活動化すると、アーキテクチャー・モードが ESA/390 に設定されま

す。参考のために、ESA/390 の初期プログラム・ローディングの説明に対する追加要素を下に挙げておきます。

操作のローディング部分の中で、リセットが行われた後は、この CPU はロード状態に入ります。この CPU は、リセット操作の実行中に必ず停止状態に入るとは限りません。CPU がロード状態にあるときは、ロード表示ライトが点灯しています。

続いて、ロード装置アドレス制御で指定されている入出力装置から、チャンネル・プログラム読み取り操作が開始されます。チャンネル・プログラムの実行結果は、絶対記憶位置 0 から始まるフォーマット 0 の CCW で、修飾ビットを 0、データ・アドレスを 0、バイト・カウントを 24、チェーン・コマンド・フラグおよび SLI フラグを 1、そしてその他のすべてのフラグを 0 に設定して、読み取りコマンドを指定した場合と同じになります。

IPL のチャンネル・サブシステム部分の詳細については、17-13ページの『初期プログラム・ローディング』で説明します。

IPL 入出力操作が正常に完了すると、IPL 装置のサブシステム識別ワードが絶対記憶位置 184-187 に格納され、絶対記憶位置 188-191 には 0 が格納され、絶対記憶位置 0-7 から新しい PSW がロードされます。PSW のロードが成功し、マシンの誤動作が検出されなければ、この CPU はロード状態から出て、ロード・ライトは消えます。速度制御が処理の位置に設定されている場合は、CPU は稼働状態になり、新しい PSW の制御下で CPU 操作が進行します。速度制御が命令ステップの位置に設定されている場合は、CPU は、新しい PSW がロードされた後で停止状態に入り、手動ライトが点灯します。

IPL 入出力操作または PSW ローディングが正常に完了しなかった場合は、CPU はロード状態のままになり、ロード・ライトは点灯したままになります。絶対記憶位置 0-7 の内容は予測不能です。

状況記憶

状況記憶操作では、アーキテクチャー・モード識別、および CPU レジスターの内容 (TOD クロックを除く) が、割り当て済みの記憶位置に格納されます。

4-47ページの図4-11 に、格納されるフィールド、それぞれの長さ、および主記憶機構内でのそれぞれの位置を示します。

フィールド	長さ (バイト)	絶対 アドレス
アーキテクチャー・ モード ID	1	163
浮動小数点レジスタ 0-15	128	4608
汎用レジスタ 0-15	128	4736
現 PSW	16	4864
プレフィックス	4	4888
浮動小数点レジスタ	4	4892
TOD プログラマブル・ レジスタ	4	4900
CPU タイマー	8	4904
0	1	4912
クロック・コンパレータ のビット 0-55	7	4913
アクセス・レジスタ 0-15	64	4928
制御レジスタ 0-15	128	4992

図 4-11. 状況記憶で使用する割り当て済み記憶位置

状況記憶操作が実行されると、絶対記憶位置 163 (状況記憶アーキテクチャー・モード識別) のビット位置 0-6 には 0 が記憶され、ビット位置 7 には 1 が記憶されます。

CPU が ESA/390 アーキテクチャー・モードになっているときは、状況記憶操作では、絶対記憶位置 163 にはすべて 0 が格納されます。

クロック・コンパレータのビット 0-55 が絶対記憶位置 4913 から格納される場合は、絶対記憶位置 4912 には 0 が格納されます。

レジスタの内容は変更されません。操作中にエラーが検出された場合は、CPU はチェック停止状態に入ります。

状況記憶操作は、状況記憶キーを使用して、手動で開始できます (第12章、『オペレーター機能』を参照)。また、状況記憶操作は、停止/状況記憶指令を指定した SIGNAL PROCESSOR を実行することにより、指定した CPU で開始することもできます。アドレス指定状況記憶指令を指定した SIGNAL PROCESSOR を実行すると、状況記憶アーキテクチャー・モード識別を除き、同じ状況情報を指定のアドレスに格納することができます (4-48ページの『プロセッサ信号指令』を参照)。

マルチプロセッシング

マルチプロセッシング機構は、システム可用性の向上およびデータとリソースの共用を目的として、共通の主記憶機構を介して複数の CPU を相互に接続する能力を提供します。マルチプロセッシング機構には以下の機構が含まれています。

- 共用主記憶機構
- CPU 間相互接続
- TOD クロック同期

これらの機構には、2つの外部割り込み条件 (TOD クロック同期チェックと誤動作警報) が関連しています。これについては、第6章、『割り込み』で説明します。ほかに、TOD クロック同期制御ビット用の制御レジスタ位置と、外部割り込み条件のマスク用の制御レジスタ位置があり、これらは、4-7ページの『制御レジスタ』にリストされています。

マルチプロセッシング構成の中のすべての CPU が、構成内のチャンネル・サブシステム (すべてのサブチャンネルを含む) にアクセスできます。入出力割り込み条件は浮動しており、構成内のどの CPU でもこれらの条件を受け入れることができます。

共用主記憶機構

共用主記憶機構を使用すると、複数の CPU が共通の主記憶機構の記憶位置にアクセスすることができます。共通主記憶機構の特定記憶位置にアクセスできるすべての CPU が、その記憶位置を含む 4K バイト・ブロック全体、および関連の記憶キーにアクセスできます。チャンネル・サブシステムおよび構成内のすべての CPU が、同じ絶対アドレスを使用して共用主記憶機構の記憶位置を参照します。

CPU アドレスの識別

各 CPU には、CPU アドレスと呼ばれる番号が割り当てられています。CPU アドレスは、構成内で個々の CPU を固有のものとして識別します。CPU を指示するには、SIGNAL PROCESSOR の CPU アドレス・フィールドにこのアドレスを指定します。また、ある CPU が、誤動作警報、緊急信号、または外部呼び出しを通知した場合は、割り込みにより CPU アドレス・フィールドにそのアドレスを格納することにより、該当の CPU が式ラベルされます。CPU アドレスはシステムのインストール時に割り当てられるもので、再構成変更により変更されることはありません。プログラムでは、STORE CPU ADDRESS を使用して、CPU のアドレスを判別することができます。

CPU の信号発信と応答

CPU 信号発信応答機能は、SIGNAL PROCESSOR と、幾つかの指令コードを解釈しそれに対処するメカニズムから成っています。この機能は、CPU 間の通信 (割り当てられた一組の指令コードの伝送、受信、デコードなど)、指定された操作の開始、および信号を発信した CPU への応答の機能を提供します。CPU は、SIGNAL PROCESSOR をその CPU 自体にあてて送ることもできます。SIGNAL PROCESSOR については、第10章、『制御命令』で説明します。

プロセッサ信号指令

プロセッサ信号指令は、SIGNAL PROCESSOR の第2オペランド・アドレスのビット位置 56-63 に指定されるもので、図4-12 に示すようにエンコードされます。

コード (16 進数)	指令
00	割り当てなし
01	センス
02	外部呼び出し
03	緊急信号
04	開始
05	停止
06	再始動
07	割り当てなし
08	割り当てなし
09	停止/状況記憶
0A	割り当てなし
0B	初期 CPU リセット
0C	CPU リセット
0D	プレフィックス設定
0E	アドレス指定状況記憶
0F-11	割り当てなし
12	アーキテクチャ設定
13-FF	割り当てなし

図 4-12. 指令のエラー・コード

各指令の定義は以下のとおりです。

センス: アドレッシングされた CPU は、指令を出した CPU に自身の状況を知らせます (ビットの定義については、4-52ページの『状況ビット』を参照)。アドレッシングされた CPU では、その他のアクションは何も生じません。ビットがすべて 0 ではない場合は、状況は SIGNAL PROCESSOR 命令の R1 フィールドに指定されている汎用レジスターに格納され、条件コード 1 が設定されます。状況ビットがすべて 0 である場合は、条件コード 0 が設定されます。

外部呼び出し: アドレッシングされた CPU では、外部呼び出し外部割り込み条件が生成されます。この割り込み条件は、SIGNAL PROCESSOR の実行中は保留にされます。関連の割り込みは、その条件に対して CPU が使用可能にされているときに起きるものであり、SIGNAL PROCESSOR の実行時に必ず起きるとは限りません。信号を送る CPU のアドレスは、割り込みが起きた時点で割り込みコードにより与えられます。1 つの CPU で保留にできる外部呼び出し条件は、一時点で 1 つだけです。この指令は、アドレッシングされた CPU が停止状態または稼働状態にある場合のみ有効です。

緊急信号: アドレッシングされた CPU では、緊急信号外部割り込み条件が生成されます。この割り込み条件は、SIGNAL PROCESSOR の実行中は保留にされません。関連の割り込みは、その条件に対して CPU が使用可能にされているときに起きるものであり、SIGNAL PROCESSOR の実行時に必ず起きるとは限りません。信号を送る CPU のアドレスは、割り込みが起きた時点で割り込みコードにより与えられます。受信側の CPU は、構成内の各 CPU (受信側 CPU 自身も含む) について、緊急信号条件を一時点で 1 つだけ保留のままにしておくことができます。この指令は、アドレッシングされた CPU が停止状態または稼働状態にある場合のみ有効です。

開始: アドレッシングされた CPU は、開始機能を実行します (4-1ページの『停止、作動、ロード、およびチェック停止の各状態』を参照)。この CPU は、SIGNAL PROCESSOR の実行中に必ず稼働状態になるとは限りません。この指令は、アドレッシングされた CPU が停止状態にある場合のみ有効です。リセットにより停止状態になっていた場合は、開始機能の実行結果は予測不能です。

停止: アドレッシングされた CPU は、停止機能を実行します (4-1ページの『停止、作動、ロード、およびチェック停止の各状態』を参照)。この CPU は、SIGNAL PROCESSOR の実行中に必ず停止状態になるとは限りません。この指令は、CPU が稼働状態にある場合のみ有効です。

再始動: アドレッシングされた CPU は、再始動機能を実行します (6-44ページの『再始動割り込み』を参照)。この CPU は、SIGNAL PROCESSOR の実行中に必ずこの操作を行うとは限りません。この指令は、アドレッシングされた CPU が停止状態または稼働状態にある場合のみ有効です。

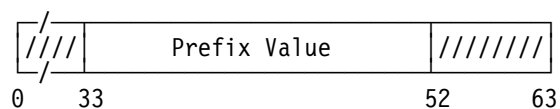
停止/状況記憶: アドレッシングされた CPU は、停止機能に続いて状況記憶操作を行います (4-46ページの『状況記憶』を参照)。この CPU は、SIGNAL PROCESSOR の実行中に必ずこの操作を完了するとは限らず、また停止状態にさえならないこともあります。この指令は、アドレッシングされた CPU が停止状態または稼働状態にある場合のみ有効です。

初期 CPU リセット: アドレッシングされた CPU は、初期 CPU リセットを行います (4-40ページの『リセット』を参照)。リセットの実行により、アーキテクチャー・モードまたは他の CPU が影響を受けることはなく、入出力がリセットされることもありません。リセット操作は、SIGNAL PROCESSOR の実行中に必ず完了するとは限りません。

CPU リセット: アドレッシングされた CPU は、CPU リセットを行います (4-40ページの『リセット』を参照)。リセットの実行により、アーキテクチャー・モードまたは他の CPU が影響を受けることはなく、入出力がリセットされることもありません。リセット操作は、SIGNAL PROCESSOR の実行中に必ず完了するとは限りません。

プレフィックス設定: SIGNAL PROCESSOR 命令のパラメーター・レジスターのビット位置 33-50 の内容は、プレフィックス値として扱われ、この値は、アドレッシングされた CPU のプレフィックス・レジスターのビット 33-50 と置き換わります。パラメーター・レジスターのビット 0-32 および 51-63 は無視されます。この指令が受け入れられるのは、アドレッシングされた CPU が停止状態にあり、プレフィックス・レジスターに入れる値が構成内で使用可能な 8K ブロックを指し示しており、かつ、この指令の受け入れを妨げるような他の条件が存在しない場合に限られます。SIGNAL PROCESSOR の実行時に、アドレッシングされた CPU が停止状態にあるかどうか、および指定された記憶域が使用可能かどうかの検査が行われます。この指令は、受け入れられたとしても、SIGNAL PROCESSOR の実行中に必ず完了するとは限りません。

パラメーター・レジスターのフォーマットは以下のとおりです。



プレフィックス設定指令は、以下のように実行されます。

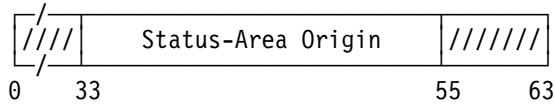
- アドレッシングされた CPU が停止状態になっていない場合は、この指令は受け入れられません。代わりに、SIGNAL PROCESSOR 命令の R₁ フィールドに指定されている汎用レジスターのビット 54 (誤り状態) が 1 に設定され、条件コード 1 が設定されます。
- アドレッシングされた CPU のプレフィックス・レジスターに入れる値が示している記憶域が、使用可能かどうかテストされます。パラメーター値のビット 33-50 の右側に 13 個、左側に 33 個の 0 を付加することにより、8K バイトの記憶域の絶対アドレスが形成されます。送信側および受信側の CPU が 24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードになっているかに関係なく、このアドレスは 64 ビットの絶対アドレスとして扱われます。新しいプレフィックス内にある 2 個の 4K バイト・ブロックから成る記憶域がアクセスされます。これらのブロックへのアクセスは保護の対象にはならず、関連の参照ビットは 1 に設定されることもされないこともあります。構成内でいずれかのブロックが使用可能でない場合は、アドレッシングされた CPU はこの指令を受け入れず、SIGNAL PROCESSOR 命令の R₁ フィールドに指定されている汎用レジスターのビット 55 (無効パラメーター) が 1 に設定され、条件コード 1 が設定されます。
- 上記の値が、アドレッシングされた CPU のプレフィックス・レジスターに入れられます。
- アドレッシングされた CPU の ALB および TLB の内容が消去されます。
- 新しいプレフィックス値が挿入された後で、アドレッシングされた CPU で逐次化およびチェックポイント同期機能が実行されます。

アドレス指定状況記憶: SIGNAL PROCESSOR のパラメーター・レジスターのビット位置 33-54 の内容は、アドレッシングされた CPU の状況が格納される、絶対記憶域内の 512 バイト境界上にある 512 バイト領域の起点として使用されます。パラメーター・レジスターのビット 0-32 および 55-63 は無視されます。

この指令が受け入れられるのは、アドレッシングされた CPU が停止状態にあり、状況域起点が構成内で使用可能な記憶位置を指し示しており、かつ、この指令の受け入れを妨げるような他の条件が存在しない場合に限られます。SIGNAL PROCESSOR の実行時に、アドレッシングされた CPU が停止状態にあるかどうか、および指定された記憶域が使用可能かどうかの検査が行われます。この指令は、受け入れられたとしても、SIGNAL

PROCESSOR の実行中に必ず完了するとは限りませ
ん。

パラメーター・レジスターのフォーマットは以下のとお
りです。



アドレス指定状況記憶指令は、以下のように実行されま
す。

- アドレッシングされた CPU が停止状態になってい
ない場合は、この指令は受け入れられません。代わ
りに、SIGNAL PROCESSOR 命令の R₁ フィールド
に指定されている汎用レジスターのビット 54 (誤
り状態) が 1 に設定され、条件コード 1 が設定され
ます。
- 状況を格納する領域が使用可能かどうかテストさ
れます。パラメーター値のビット 33-54 の右側に 9
個、左側に 33 個の 0 を付加することにより、512
バイトの記憶域の絶対アドレスが形成されます。送
信側および受信側の CPU が 24 ビット、31 ビッ
ト、または 64 ビットのいずれのアドレッシング・
モードになっているかに関係なく、このアドレスは
64 ビットの絶対アドレスとして扱われます。この
アドレスにある 512 バイトの記憶域ブロックがア
クセスされます。このアクセスは保護の対象にはな
らず、関連の参照ビットは 1 に設定されることもさ
れないこともあります。構成内でこのブロックが使
用可能でない場合は、アドレッシングされた CPU
はこの指令を受け入れず、SIGNAL PROCESSOR
命令の R₁ フィールドに指定されている汎用レジ
スターのビット 55 (無効パラメーター) が 1 に設定さ
れ、条件コード 1 が設定されます。
- アドレッシングされた CPU の状況が、指定の領域
に入れられます。格納される情報、およびその情報
を受け入れる領域のフォーマットは、停止/状況記憶
指令の場合と同じです。ただし、各フィールドは、
絶対記憶域の始めからの特定のオフセット位置に格
納されるのではなく、図4-13 に示すオフセット位
置にある指定領域に格納され、また、アーキテク
チャー・モード識別は格納されないという点が異なり
ます。指定の領域のバイト 288-291 および
312-319 は変更されません。
- 状況が格納された後で、アドレッシングされた
CPU で逐次化およびチェックポイント同期機能が
実行されます。

フィールド	長さ (バイト)	オフセット (バイト)
浮動小数点レジスター 0-15	128	0
汎用レジスター 0-15	128	128
現 PSW	16	256
プレフィックス	4	280
浮動小数点レジスター	4	284
TOD プログラマブル・ レジスター	4	292
CPU タイマー	8	296
0	1	304
クロック・コンパレーター のビット 0-55	7	305
アクセス・レジスター 0-15	64	320
制御レジスター 0-15	128	384

図 4-13. 指定した領域内での状況フィールドの位置

プログラミング上の注意: 指定領域に状況を格納した
CPU のアーキテクチャー・モードは、通常、その領域
内のオフセット 256 に格納される PSW のビット 12 に
示されます。この PSW は、ESA/390 モードと
z/Architecture モードのいずれの場合も、同じオフセッ
ト (256) に格納されます。ビット 12 は、ESA/390
PSW では 1 で、z/Architecture PSW では 0 です。状
況記憶操作および停止/状況記憶指令では、絶対記憶位置
163 にアーキテクチャー・モード識別が格納されま
すが、アドレス指定状況記憶指令ではこの識別情報は格納
されません。

アーキテクチャー設定: パラメーター・レジスターの
ビット位置 56-63 の内容は、構成内のすべての CPU が
設定されるアーキテクチャー・モードを指定するコード
として使用されます。コード 0 は ESA/390 モード、コ
ード 1 は z/Architecture モードを指定します。パラメ
ーター・レジスターのビット 0-55 は無視されます。
SIGNAL PROCESSOR 命令の CPU アドレス・レジ
スターの内容は、無視されます。構成内の他のすべての
CPU は、アドレッシングされるものと見なされます。

この指令が受け入れられるのは、上記のコードが 0 また
は 1 で、CPU がまだコードに指定されたモードになっ
ておらず、他のすべての CPU が停止状態またはチェッ
ク停止状態になっており、かつ、この指令の受け入れを
妨げるような他の条件が存在しない場合に限られます。
この指令は、受け入れられたとしても、SIGNAL
PROCESSOR の実行中にすべての CPU で完了する
とは限りません。CPU によってアーキテクチャー・モ
ードが異なることはありません。

アーキテクチャー設定指令は、以下のように実行されま
す。

- パラメーター・レジスター内のコードが 0 または 1 のどちらでもない場合、または CPU がすでにコードで指定されているアーキテクチャー・モードになっている場合は、この指令は受け入れられません。代わりに、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターのビット 55 (無効パラメーター) が 1 に設定され、条件コード 1 が設定されます。
- 構成内の他のすべての CPU が停止状態またはチェック停止状態になっていない場合は、この指令は受け入れられません。代わりに、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターのビット 54 (誤り状態) が 1 に設定され、条件コード 1 が設定されます。
- 構成内のすべての CPU のアーキテクチャー・モードが、コードで指定されているモードに設定されません。
- この指令により、アーキテクチャー・モードが ESA/390 モードから z/Architecture モードに変更された場合は、構成内のそれぞれの CPU について、8 バイトの現 PSW は 16 バイトに変更され、16 バイト PSW のビットは以下のように設定されます。ビット 0-11 および 13-32 は 8 バイト PSW の同じビットと同じ値に設定され、ビット 12 および ビット 33-96 は 0 に設定され、ビット 97-127 は 8 バイト PSW のビット 33-63 に設定されます。また、ESA/390 プレフィックスのビット 19 (これは z/Architecture プレフィックスのビット 51 になります) は、0 に設定されます。
- この指令により、アーキテクチャー・モードが z/Architecture から ESA/390 に変更された場合は、構成内の各 CPU について、16 バイトの現 PSW が 8 バイト PSW に変更され、8 バイト PSW のビットは以下のように設定されます。ビット 0-11 および 13-32 は 16 バイト PSW の同じビットと同じ値に設定され、ビット 12 は 1 に設定され、ビット 33-63 は 16 バイト PSW のビット 97-127 と同じ値に設定されます。
- 構成内のすべての CPU の ALB および TLB の内容が消去されます。
- 構成内のすべての CPU で、逐次化およびチェックポイント同期機能が実行されます。

この指令により、アーキテクチャー・モードが z/Architecture から ESA/390 に変更され、**SIGNAL PROCESSOR** 命令の結果命令取り出し PER イベントが起きた場合は、この命令のアドレスの右端 31 ビット

が ESA/390 PER アドレス・フィールドに格納されません。

プログラミング上の注意:

1. アーキテクチャー設定指令により、アーキテクチャー・モードが z/Architecture から ESA/390 に変更されたときに、PSW のビット 31 が 1 である場合は、その PSW は無効です。
2. **SIGNAL PROCESSOR** の指令の相対パフォーマンスに関する説明については、第10章、『制御命令』の中の **SIGNAL PROCESSOR** 命令に関する『プログラミング上の注意』を参照してください。

応答を決定する条件

指令コードの解釈を妨げる条件

以下のような状況が生じると、指令の開始が妨げられます。以下に挙げる状況は、複数の状況が同時に存在する場合に優先度が高くなる方から順に示してあります。

1. 同時に実行されている **SIGNAL PROCESSOR** が CPU 信号発信応答機能を使用しているため、アドレッシングされた CPU へのアクセス・パスが使用中の場合。この命令を同時に実行している CPU は、構成内のこの CPU 以外のいずれかの CPU であり、CPU アドレスは、この CPU のアドレスまたは無効なアドレスも含めた任意のアドレスです。この場合、指令は拒否されます。そして、条件コード 2 が設定されます。
2. アドレッシングされた CPU が非稼働状態にある場合。これは、その CPU がインストール・システムに含まれていないか、構成に含まれていないか、幾つかの技術員テスト・モードのいずれかになっているか、または電源がオフになっている場合です。この場合、指令は拒否されます。そして、条件コード 3 が設定されます。CPU が自身をアドレッシングしている場合は、**SIGNAL PROCESSOR** 命令の結果としてこの条件が生じることはありません。
3. アドレッシングされた CPU で以下のいずれかの条件が生じている場合。
 - a. アドレッシングされた CPU が、前に出された開始、停止、再始動、停止/状況記憶、プレフィックス設定、またはアドレス指定状況記憶指令を受け入れており、その指令で要求された機能の実行がまだ完了していない。
 - b. アドレッシングされた CPU で、手動による開始、停止、再始動、または状況記憶機能が開始されており、その機能がまだ完了していない。

CPU が自身をアドレッシングしている場合は、**SIGNAL PROCESSOR** 命令の結果としてこの条件が生じることはありません。

現在指定されている指令が、センス、外部呼び出し、緊急信号、開始、停止、再始動、停止/状況記憶、プレフィックス設定、アドレス指定状況記憶、またはアーキテクチャー設定である場合は、その指令は拒否され、条件コード 2 が設定されます。現在指定されている指令がリセット指令の 1 つであるか、あるいは、割り当てられていないかまたはインプリメントされていない指令である場合は、指令コードは、『状況ビット』の説明に従って解釈されません。

4. アドレッシングされた CPU で以下のいずれかの条件が生じている場合。

- a. アドレッシングされた CPU が、前に出された初期 CPU リセットまたは CPU リセット指令を受け入れており、その指令で要求された機能の実行がまだ完了していない。
- b. アドレッシングされた CPU で、手動リセット機能が開始されており、その機能がまだ完了していない。CPU が自身をアドレッシングしている場合は、**SIGNAL PROCESSOR** 命令の結果としてこの条件が生じることはありません。

現在指定されている指令が、センス、外部呼び出し、緊急信号、開始、停止、再始動、停止/状況記憶、プレフィックス設定、アドレス指定状況記憶、またはアーキテクチャー設定である場合は、その指令は拒否され、条件コード 2 が設定されます。現在指定されている指令がリセット指令の 1 つであるか、あるいは、割り当てられていないかまたはインプリメントされていない指令である場合は、その指令コードが拒否されて条件コード 2 が設定されるか、または、指令コードは『状況ビット』の説明に従って解釈されます。

上記の 3 および 4 で挙げた条件のいずれかが存在する場合は、アドレッシングされた CPU は「使用中」の状態にあると言われます。アドレッシングされた CPU がチェック停止状態にあるか、オペレーター介入条件が存在する場合は、使用中の標識は示されません。一般に、CPU 使用中条件が続くのは短い間です。ただし、上記の項目 3 で述べた条件は、一連の割り込みが原因で限りなく続くことがあります。しかし、この状況下では、どのリセット指令からも、CPU が使用中であるとは認識されません。

上記の項目 1 および 2 で述べた条件に該当せず、オペレーター介入状況条件や受信側チェック状況条件も存在しない場合は、アドレッシングされた CPU が前に受け

入れた指令が完了しているかどうかに関係なく、リセット指令が受け入れられることがあります。その結果、前の指令は、まだ一部しか完了していない状態で失われることがあります。失われた指令については、定義されている結果が得られるかどうかは予測不能です。

状況ビット

さまざまな状況条件が定義されており、指令の発行元の CPU およびアドレッシングされた CPU は、これらの条件に従って指令に対する応答を提示することができます。図4-14 は、各種の状況条件と、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスター内での各条件のビット位置を示しています。

ビット位置	状況条件
32	装置チェック
33-53	割り当てなし、0 が入る
54	誤り状態
55	無効パラメーター
56	外部呼び出し保留
57	停止
58	オペレーター介入
59	チェック停止
60	割り当てない、0 が入る
61	操作不能
62	無効指令
63	受信側チェック

図 4-14. 状況条件

ビット位置 32、およびビット位置 55 (指令がアーキテクチャー設定の場合) に割り当てられる状況条件は、**SIGNAL PROCESSOR** を実行する側の CPU で生成されます。その他の状況条件は、アドレッシングされる CPU で生成されます。

アーキテクチャー設定指令について無効パラメーター条件が存在する場合は、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターのビット 55 は 1 に設定され、ビット位置 32-63 の範囲内のその他のビットは 0 に設定され、ビット 0-31 は変更されず、そして条件コード 1 が設定されます。その他のアクションは何も行われません。

装置チェック条件が存在する場合は、アーキテクチャー設定指令について無効パラメーター条件が存在する場合は除き、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターのビット 32 は 1 に設定され、状況レジスターのビット位置 32-36 の見割り当てのビットは 0 に設定され、その他の状況ビットは予測不能であり、ビット 0-31 は変更されません。この場合

は、アドレッシングされた CPU へのアクセス・パスが使用中かどうか、および、アドレッシングされた CPU が、非稼働、使用中、またはゼロ状況提示の状態にあるかどうかに関係なく、条件コード 1 が設定されます。

アドレッシングされた CPU が使用中ではなく、かつ、アドレッシングされた CPU が稼働状態にあって、現在指定されている指令に対して使用中状態を示していない場合は、アドレッシングされた CPU は、自身の状況を発行元の CPU に提示します。これらの状況ビットには、次の 2 つのタイプがあります。

1. 状況ビット 54、55 (指令がアーキテクチャー設定ではない場合)、56-59、および 61 は、指令コードを受け取った時点で、アドレッシングされた CPU 内に対応する条件が存在することを示します。センス指令への応答の場合を除き、各条件が示されるのは、その条件が指定された指令の正常な実行を妨げる場合のみです。ただし、無効パラメーターは、他の妨害条件が存在するときは示されないこともあります。センス指令の場合は、存在するすべての状況条件が示されます。オペレーター介入条件は、インストールされているいずれかの指令の実行妨げる場合に、示されます。
2. 状況ビット 62 および 63 は、アドレッシングされた CPU が指令を受け取るときに、対応する条件が検出されたことを示します。

提示された状況がすべて 0 の場合は、アドレッシングされた CPU が指令を受け入れたことを意味し、発行元の CPU では条件コード 0 が設定されます。提示された状況がすべて 0 ではない場合は、指令は拒否されており、発行元の CPU では、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターに状況が格納され、そのレジスターのビット位置 32-63 の範囲内の未割り当ての位置には 0 が格納され、ビット 0-31 は変更されず、そして条件コード 1 が設定されます。

指令がアーキテクチャー設定である場合は、「アドレッシングされた CPU」は、構成内にあるその他の各 CPU を指します。これらの CPU については、条件コード 1、2、または 3 が設定される原因となるような条件が存在するかどうかテストされますが、どの順序でテストされるかは予測不能です。条件は、個々に CPU の中では、アドレッシングされた CPU が 1 つだけの場合と同様に優先順位付けされますが、CPU 間での優先順位の差はありません。条件が 1 つでも認識された場合は、それ以上 CPU のテストは行われず、その条件に対応する条件コードが設定され、**SIGNAL PROCESSOR** の実行は完了します。

状況条件は以下のように定義されています。

装置チェック: この条件が存在するのは、命令を実行している CPU が、この命令とそれに関連した指定の実行のみに影響を与える装置誤動作を検出したときです。指令コードは、伝送されている場合もされていない場合もあり、受け入れられている場合もない場合もあり、さらに、アドレッシングされた CPU が提供する状況ビットは誤っていることもあります。アーキテクチャー設定指令について無効パラメーター条件が検出された場合は、この条件は検出されません。

誤り状態: アドレッシングされた CPU が停止状態にならないため、プレフィックス設定指令またはアドレス指定状況記憶指令が拒否されたか、または、他のすべての CPU が停止またはチェック停止状態になっていないため、アーキテクチャー設定指令が拒否されました。該当する場合は、この状況は **SIGNAL PROCESSOR** の実行中に生成され、指令の実行を妨げる他の条件と一緒に示されます。ただし、アーキテクチャー設定指令について無効パラメーター条件が存在する場合は、この状況は生成されません。

無効パラメーター: この条件が存在するのは、次の 2 つの場合です。

1. プレフィックス設定指令またはアドレス指定状況記憶指令で与えられたパラメーター値が、構成内で使用できない記憶位置を示している。該当する場合は、この状況は **SIGNAL PROCESSOR** の実行中に生成されます。ただし、ほかにも指令の実行を妨げる条件が存在する場合は、この状況は生成されないこともあります。
2. アーキテクチャー設定指令で与えられたパラメーター値が、0 または 1 のいずれでもないか、または現アーキテクチャー・モードを示している。該当する場合は、この状況は **SIGNAL PROCESSOR** の実行中に生成され、他の状況は生成されません。

外部呼び出し保留: この条件が存在するのは、前に発行された **SIGNAL PROCESSOR** 指令があるために、アドレッシングされた CPU で外部呼び出し割り込み条件が保留になっているときです。この条件は、外部呼び出し指令が受け入れられてから、結果の外部割り込みが完了するか、CPU リセットが生じるまで、存続します。この条件の原因は、発行元の CPU にある場合も、相手の CPU にある場合もあります。この条件は、センスおよび外部呼び出しに対する応答の中でのみ示されます。

停止: この条件が存在するのは、アドレッシングされた CPU が停止状態にあるときです。この条件は、センスに対する応答の中でのみ示されます。CPU が自身をアドレッシングしている場合は、SIGNAL PROCESSOR 命令の結果としてこの条件が報告されることはありません。

オペレーター介入 (プロセッサ信号状況): この条件が存在するのは、アドレッシングされた CPU が、ローカルまたはリモートのオペレーター機能から開始された操作を実行しているときです。手動で開始されるような操作によりこの条件が引き起こされるかは、モデルと指定された指令によって異なります。アドレッシングされた CPU が、書替可能制御記憶機構を使用して指令を実行するときに、必要なライセンス内部コードが IML 機能によりロードされていない場合に、この条件が生じることがあります。オペレーター介入条件は、すべての指令に対する応答の中で示されます。オペレーター介入条件が存在し、それが原因で、インストールされているいずれかの指令の受け入れが妨げられている場合は、センスに対する応答の中でこの条件が示されます。また、未割り当てまたは未インストールの指令に対する応答の中でも、この条件が示されることがあります。CPU が自身をアドレッシングしている場合は、SIGNAL PROCESSOR 命令の結果としてこの条件が生じることがありません。

チェック停止: この条件が存在するのは、アドレッシングされた CPU がチェック停止状態にあるときです。この条件は、センス、外部呼び出し、緊急信号、開始、停止、再始動、プレフィックス設定、アドレス指定状況記憶、および停止/状況記憶に対する応答の中でのみ示されます。また、未割り当てまたは未インストールの指令に対する応答の中でも、この条件が示されることがあります。CPU が自身をアドレッシングしている場合は、SIGNAL PROCESSOR 命令の結果としてこの条件が報告されることはありません。

操作不能: この条件は、指令コードで指定されている操作を実行するために必要なサービス・プロセッサが操作不能であることを示しています。このサービス・プロセッサの障害については、すでにサービス・プロセッサ損傷マシン・チェック条件により報告されている場合もあります。センス、外部呼び出し、または緊急信号指令コードの場合は、操作不能条件が生じることがありません。

無効指令: この条件が存在するのは、SIGNAL PROCESSOR の実行に関連した通信において、未割り当てまたは未インストールの指令コードがデコードされたときです。

受信側チェック: この条件が存在するのは、SIGNAL PROCESSOR の実行に関連した通信において、アドレッシングされた CPU が装置の誤動作を検出したときです。この条件が示された場合は、指令はまだ開始されていません。また、残りの受信側状況ビットの生成がこの誤動作の影響を受けている可能性があるため、これらのビットは有効であるとは限りません。アドレッシング CPU では、マシン・チェック条件が生成されていることも、いないこともあります。

次の図は、各指令コードに対する応答の中で、発行元の CPU にどのような状況条件が提示されるかを示しています。

状況条件

63 受信側チェック ≠
 62 無効指令
 61 操作不能
 59 チェック停止
 58 オペレーター介入#
 57 停止
 56 外部呼び出し保留
 55 無効パラメーター
 54 誤り状態

指令

センス	0	0	X	X	X	X	0	0	X
外部呼び出し	0	0	X	0	X	X	0	0	X
緊急信号	0	0	0	0	X	X	0	0	X
開始	0	0	0	0	X	X	X	0	X
停止	0	0	0	0	X	X	X	0	X
再始動	0	0	0	0	X	X	X	0	X
停止/状況記憶	0	0	0	0	X	X	X	0	X
初期 CPU リセット	0	0	0	0	X	0	X	0	X
CPU リセット	0	0	0	0	X	0	X	0	X
プレフィックス設定	X	X	0	0	X	X	X	0	X
アドレス指定状況記憶	X	X	0	0	X	X	X	0	X
アーキテクチャー設定	X	X	0	0	X	0	X	0	X
未割り当ての指令	0	0	0	0	X	E	X	1	X

説明:

- # オペレーター介入条件の現在の状態は、解釈しようとしている指令コードによって異なることがあります。
- ≠ 受信側ビット位置に 1 が示されている場合は、他のビット位置に示されている値は有効であるとは限りません。
- 0 このビット位置には、この条件の現在の状態とは関係なく、0 が示されます。
- 1 このビット位置には、1 が示されます。
- X このビット位置には、対応する条件の現在の状態に応じて、0 または 1 が示されます。
- E 0 か、または対応する条件の現在の状態が示されます。

示されている状況ビットがすべて 0 である場合は、指令は受け入れられており、発行元の CPU では条件コード 0 が設定されます。1 つ以上の 1 が示されている場合は、指令は拒否されており、発行元の CPU は、**SIGNAL PROCESSOR** 命令の R1 フィールドに指定されている汎用レジスターに状況を格納し、条件コード 1 を設定します。

プログラミング上の注意:

1. アーキテクチャー設定指令 (これは事実上他のすべての CPU をアドレッシングし、すべての CPU に影響を与える) を除き、**SIGNAL PROCESSOR** のすべての指令は、それを発行する CPU 自身に対してアドレッシングすることができます。以下に、CPU が自身に対して **SIGNAL PROCESSOR** をアドレッシングすることにより得られる機能の例を示します。
 - a. **センス** では、外部呼び出し条件が保留になっているかどうかを示されます。
 - b. **外部呼び出し** および**緊急信号** では、対応する割り込み条件が生成されます。**外部呼び出し** は、前に生成された外部呼び出し条件が原因で拒否されることがあります。
 - c. **開始** では、条件コード 0 が設定されるだけで、その他の影響はありません。

- d. **停止** では、CPU は条件コード 0 を設定し、使用可能にされている保留中の割り込みを実行し、そして停止状態に入ります。
- e. **再始動** では、現 PSW を格納するための手段が提供されます。
- f. **停止/状況記憶** では、マシンは停止し、現在のすべての状況を格納します。

2. 2 つの CPU が、互いに他をアドレッシングして、同時に **SIGNAL PROCESSOR** を実行することができます。この場合、一方の CPU (両方ではなく) が、アクセス・パスが使用中であることを検出することがあります。これは、もう一方の CPU が実行している **SIGNAL PROCESSOR** に関連した指令コードまたは状況ビットが伝送されているためです。逆に、どちらの CPU もアクセス・パスを使用でき、互いに指令コードを伝送できることもあります。特に、2 つの CPU は、互いに同時に相手を停止、再始動、またはリセットすることができます。
3. アドレス指定状況記憶指令を使用して、チェック停止状態にある他の CPU から状況を取得するには、まず CPU リセット操作を使用して、その CPU を停止状態にする必要があります。このリセット指令は状況を変更することはなく、誤動作の性質によっては、アドレッシングされた CPU 内に、状況の入手を可能にするような条件を確立するための、最適の機会を提供します。

第5章 プログラムの実行

命令	5-2	PC 番号の変換	5-28
オペランド	5-3	PC 番号変換制御	5-28
命令のフォーマット	5-3	制御レジスター 5	5-28
レジスター・オペランド	5-6	PC 番号変換テーブル	5-28
即値オペランド	5-6	リンケージ・テーブル・エントリー	5-28
記憶域オペランド	5-6	エントリー・テーブル・エントリー	5-29
アドレスの生成	5-7	PC 番号変換プロセス	5-31
3 モード・アドレッシング	5-7	リンケージ・テーブル指定の取得	5-32
命令アドレスの順次生成	5-7	リンケージ・テーブル索引	5-32
オペランド・アドレスの生成	5-8	エントリー・テーブル索引	5-32
中間値の形成	5-8	PC 番号変換時の例外認識	5-33
オペランド・アドレスの形成	5-8	ホーム・アドレス・スペース	5-33
ブランチ・アドレスの生成	5-9	アクセス・レジスターの概要	5-34
中間値の形成	5-9	要約	5-34
ブランチ・アドレスの形成	5-9	アクセス・レジスターの機能	5-34
命令の実行と順序付け	5-10	アクセス・レジスター指定のアドレス・ス	
判断処理	5-10	ペース	5-34
ループ制御	5-10	アクセス・レジスター命令	5-41
リンケージ・スタックなしのサブルーチン・リ		アクセス・レジスターの変換	5-42
ンケージ	5-10	アクセス・レジスター変換制御	5-42
単純ブランチ命令	5-10	制御レジスター 2	5-42
その他のリンケージ命令	5-13	制御レジスター 5	5-42
割り込み	5-19	制御レジスター 8	5-43
命令終了のタイプ	5-19	アクセス・レジスター	5-43
完了	5-19	アクセス・レジスター変換テーブル	5-44
抑止	5-20	ディスパッチ可能単位制御テーブルおよび	
無効化	5-20	アクセス・リスト指定	5-44
中止	5-20	アクセス・リスト・エントリー	5-45
割り込み可能命令	5-20	ASN 第 2 テーブル・エントリー	5-46
割り込み点	5-20	アクセス・レジスター変換プロセス	5-48
操作単位	5-20	アクセス・リスト・エントリー・トークン	
割り込み可能命令の実行	5-20	の選択	5-50
割り込み可能命令の代わりの条件コード	5-21	1 次または 2 次アドレス・スペース制御エ	
無効化と抑止に対する例外	5-22	レメントの取得	5-51
DAT 関連アクセス例外の場合の記憶域の変		ALET の第 1 バイトの検査	5-51
更と復元	5-22	有効アクセス・リスト指定の取得	5-51
DAT テーブル・エントリーの変更	5-23	アクセス・リストの検索	5-51
編集命令および変換命令の試行実行	5-23	ASN 第 2 テーブル・エントリーの検索	5-52
許可メカニズム	5-23	アクセス・リスト・エントリーの使用許可	5-52
モードの要件	5-24	アクセス・リスト制御保護の検査	5-52
抽出権限制御	5-24	2 次 ASN 第 2 テーブル・エントリーか	
PSW キー・マスク	5-24	らのアドレス・スペース制御エレメントの	
2 次スペース制御	5-24	取得	5-52
サブシステム・リンケージ制御	5-25	アクセス・レジスター変換時の例外の認識	5-53
ASN 変換制御	5-25	ART 索引バッファ	5-53
許可指標	5-25	ALB の構造	5-53

ALB エントリーの形成	5-53	アンスタッキング・プロセス	5-75
ALB エントリーの使用	5-54	現エントリーの位置決めとヘッダー・エン	
ART テーブルの変更	5-54	トリーの処理	5-75
サブスペース・グループ	5-55	状態エントリーかどうかの検査	5-76
サブスペース・グループ・テーブル	5-55	情報の復元	5-76
サブスペース・グループ・ディスパッチ可		先行エントリーの更新	5-77
能単位制御テーブル	5-55	制御レジスター 15 の更新	5-77
サブスペース・グループ ASN 第 2 テー		アンスタッキング・プロセスでの例外の認	
ブル・エントリー	5-57	識	5-77
サブスペース置き換え操作	5-58	記憶機構参照の順序	5-77
リンケージ・スタックの概要	5-59	概念上の順序	5-78
要約	5-59	命令実行における操作のオーバーラップ	5-78
リンケージ・スタックの機能	5-60	分割可能な命令実行	5-78
プログラム制御の転送	5-60	仮想記憶域参照のためのインターロック	5-79
リンケージ・スタックを使用したブランチ	5-62	命令間のインターロック	5-79
情報の追加と検索	5-62	単一命令内のインターロック	5-80
許可のテスト	5-63	命令取り出し	5-81
プログラム問題の分析	5-63	ART テーブルおよび DAT テーブルの取り出	
リンケージ・スタック・エントリー・テーブル・		し	5-82
エントリー	5-64	記憶キーのアクセス	5-83
リンケージ・スタックの操作	5-65	記憶域オペランドの参照	5-84
リンケージ・スタック操作制御	5-67	記憶域オペランド取り出し参照	5-84
制御レジスター 15	5-67	記憶域オペランド格納参照	5-84
リンケージ・スタック	5-67	記憶域オペランド更新参照	5-85
エントリー記述子	5-67	記憶域オペランドの一貫性	5-86
ヘッダー・エントリー	5-68	単一アクセス参照	5-86
トレーラー・エントリー	5-69	複数アクセス参照	5-86
状態エントリー	5-69	ブロック単位の参照	5-87
スタッキング・プロセス	5-72	一貫性の指定	5-87
新規エントリー用のスペースの位置決め	5-72	オペランド・アクセス間の関係	5-88
新規エントリーの形成	5-73	その他の記憶域参照	5-89
現エントリーの更新	5-74	逐次化	5-89
制御レジスター 15 の更新	5-74	CPU の逐次化	5-89
スタッキング・プロセスでの例外の認識	5-75	チャンネル・プログラムの逐次化	5-90

通常、CPU の動作は記憶機構内の命令により制御されます。命令は、記憶域アドレスの昇順に従って、左から右へ一度に 1 つずつ順番に実行されます。この順次操作は、ブランチ、LOAD PSW、割り込み、SIGNAL PROCESSOR の指令、または手動の介入によって変更することができます。

命令

命令は、次の 2 つの主要部分から成っています。

- 命令コード (op コード)。これは、実行する操作を指定します。
- 関連のオペランドの指定。

オペランド

オペランドは3つのクラスに分類されます。それは、レジスター内に入れられるオペランド、即値オペランド、および記憶機構内のオペランドです。オペランドは、明示的に指定される場合と暗黙的に指定される場合があります。

レジスター・オペランドは、汎用レジスター、浮動小数点レジスター、アクセス・レジスター、または制御レジスターに含めることができ、レジスターのタイプは命令コードにより識別されます。オペランドを含むレジスターは、命令の中の R フィールドと呼ばれる4ビット・フィールド内のレジスターを識別することによって指定されます。命令によっては、オペランドは、暗黙的に指定されたレジスターに含まれていることがあります。このレジスターは、命令コードにより暗黙的に示されません。

即値オペランドは命令自体の中にそのまま含まれているものであり、即値オペランドを含む8ビット、16ビット、または32ビットのフィールドを、I フィールドと呼びます。

記憶機構内オペランドの長さは、暗黙に指定されるか、ビット・マスクにより指定されるか、命令内の L フィールドと呼ばれる4ビットまたは8ビットの長さ指定により指定されるか、または、汎用レジスターの内容により指定されます。記憶機構内オペランドのアドレスは、汎用レジスターの内容をアドレスの一部として使用するフォーマットにより指定されます。これにより、以下のことが可能になります。

1. 簡略表記法を使用して完全なアドレスを指定する。
2. 汎用レジスターをオペランドとして用いる命令を使用して、アドレス操作を行う。
3. 命令ストリームを変更せずに、プログラムによりアドレスを変更する。
4. 他のプログラムから受け取ったアドレスを直接使用して、データ域の位置に拘束されずに操作を行う。

記憶域を参照するために使用するアドレスは、命令の R フィールドが指すレジスターに含まれているか、または、命令の中で指定されている基底アドレス (B フィールド)、指標 (X フィールド)、および変位 (D フィールド) から計算されます。

CPU がアクセス・レジスター・モードにあるときは、B フィールドまたは R フィールドは、アドレスを指定するために使用されるほかに、アクセス・レジスターを指定することもできます。

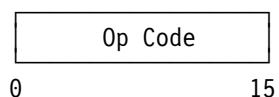
命令の実行を記述するために、オペランドは、第1オペランドと第2オペランド、そして場合によっては第3オペランドとして指定されます。

一般に、1つの命令の実行には2つのオペランドが関与し、実行結果は第1オペランドと置き換わります。ただし、CONVERT TO DECIMAL、TEST BLOCK、および、命令名に「STORE」が含まれている命令 (STORE THEN AND SYSTEM MASK および STORE THEN OR SYSTEM MASK を除く) は、格納先の場所を指定するために第2オペランドのアドレスを使用します。TEST AND SET、COMPARE AND SWAP、および COMPARE DOUBLE AND SWAP は、第2オペランドを更新することがあります。特に断り書きがない限り、操作のアドレッシングまたは実行部分に関するすべてのレジスターおよび記憶位置の内容は、変更されないうまみになります。

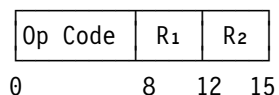
命令のフォーマット

命令の長さは1個、2個、または3個のハーフワードとし、記憶機構内でハーフワード境界に合わせて配置する必要があります。命令は、E、RR、RRE、RRF、RX、RXE、RXF、RS、RSE、RSL、RSI、RI、RIE、RIL、SI、S、SSE、SS の18種類の基本フォーマットのいずれかの形をとります。さらに、RRFには3つ、RS、RSE、およびRILには2つ、そしてSSには4つのバリエーションがあります。5-4ページの図5-1を参照してください。

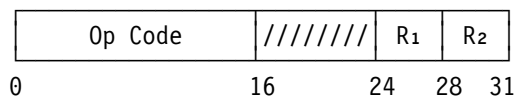
E フォーマット



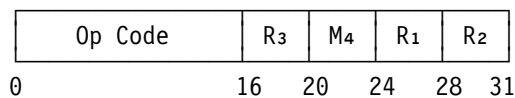
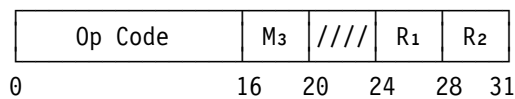
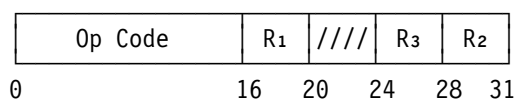
RR フォーマット



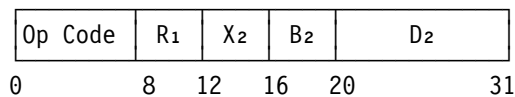
RRE フォーマット



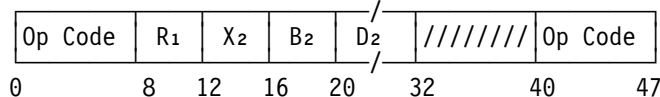
RRF フォーマット



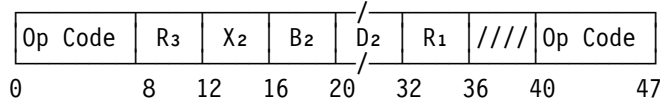
RX フォーマット



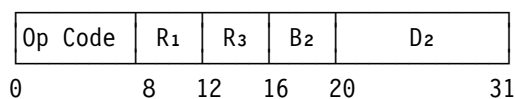
RXE フォーマット



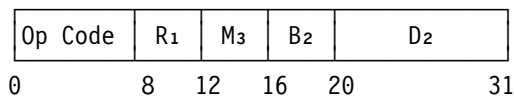
RXF フォーマット



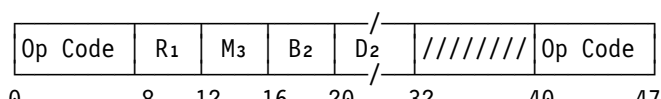
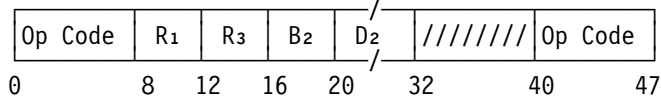
RS フォーマット



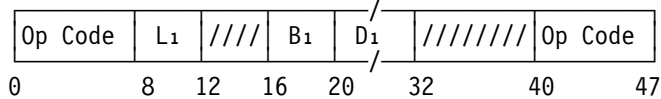
RS フォーマット (続き)



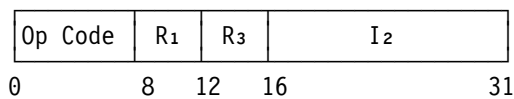
RSE フォーマット



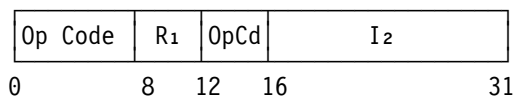
RSL フォーマット



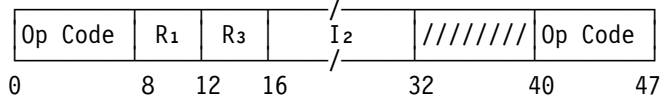
RSI フォーマット



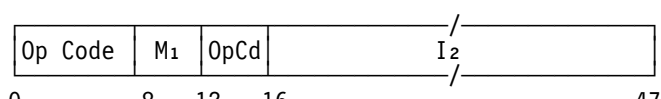
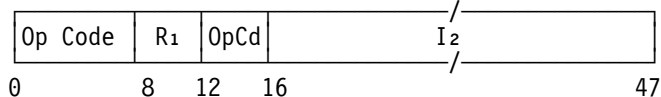
RI フォーマット



RIE フォーマット



RIL フォーマット



SI フォーマット

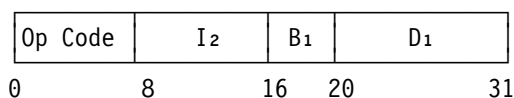
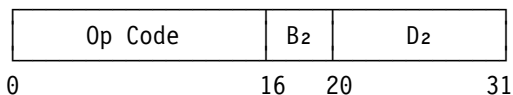


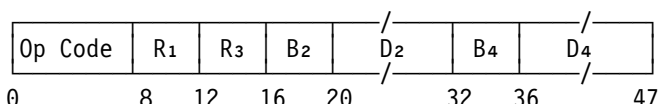
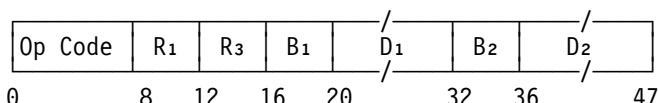
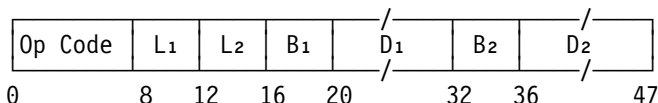
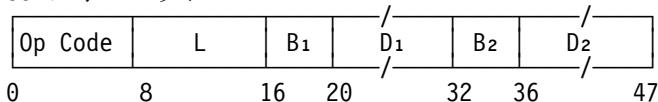
図 5-1 (1/3). 基本命令フォーマット

図 5-1 (2/3). 基本命令フォーマット

S フォーマット



SS フォーマット



SSE フォーマット

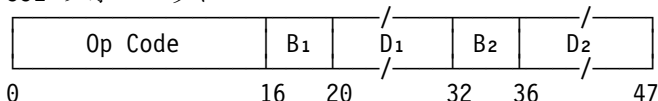


図 5-1 (3/3). 基本命令フォーマット

5-4ページの図5-1の中でスラッシュ (/) が示されているフィールドは、現在まだ割り当てられていないフィールドです。命令の中では、これらのフィールドには0が含まれている必要があります。さもないと、将来、プログラムの動作の互換性が失われることがあります。

命令によっては、基本フォーマットとはやや異なるフィールドを持つものもあり、また、このセクションで述べる一般規則に従わない動作をするものもあります。このような例外は、すべて、該当する個々の命令の説明の中で明確に示してあります。

フォーマットの名前は、該当の操作に関係するオペランドのクラスを示す一般概念を表しています。

- E は、暗黙オペランドを使用し、拡張命令コード・フィールドを持つ操作を表します。
- RR は、レジスター相互間の操作を表します。
- RRE は、拡張命令コード・フィールドを持つレジスター間操作を表します。
- RRF は、拡張命令コード・フィールドに加えて、R フィールド、M フィールド、またはその両方を持つレジスター間操作を表します。
- RX は、レジスター対指標付き記憶域操作を表します。
- RXE は、拡張命令コード・フィールドを持つレジスター対指標付き記憶域操作を表します。

- RXF は、拡張命令コード・フィールドに加えて R フィールドを持つ、レジスター対指標付き記憶域操作を表します。
- RS は、レジスター対記憶域操作を表します。
- RSE は、拡張命令コード・フィールドを持つレジスター対記憶域操作を表します。
- RSL は、記憶域操作を表します (RSE フォーマットから派生した命令フォーマットの1つ)。
- RSI は、レジスター対即値操作を表します。
- RI は、拡張命令コード・フィールドを持つレジスター対即値操作を表します。
- RIE は、長い拡張命令コード・フィールドを持つレジスター対即値操作を表します。
- RIL は、拡張命令コード・フィールドと長い即値フィールドを持つレジスター対即値操作を表します。
- SI は、記憶域対即値操作を表します。
- S 暗黙のオペランドと記憶域を使用した操作を表します。
- SS は、記憶域間操作を表します。
- SSE は、拡張命令コード・フィールドを持つ記憶域間操作を表します。

RR、RX、RS、RSI、SI、およびSSフォーマットでは、命令の最初の1バイトには命令コードが入ります。E、RRE、RRF、S、およびSSEフォーマットでは、命令の最初の2バイトに命令コードが入ります。ただし、Sフォーマットの一部の命令では、最初の1バイトのみに命令コードが入ります。RIおよびRILフォーマットでは、命令の最初の1バイトおよびビット位置12-15に命令コードが入ります。RXE、RXF、RSE、RSL、およびRIEフォーマットでは、命令コードは、命令の最初の1バイトおよびビット40-47に入ります。

命令コードの最初または唯一のバイトの最初の2ビットは、以下のように、命令の長さフォーマットを指定します。

ビット位置 0-1	命令の長さ (ハーフワード数)	命令のフォーマット
00	1	E/RR
01	2	RX
10	2	RRE/RRF/RX/RS/RSI/RI/SI/S
11	3	RXE/RXF/RSE/RSL/RIE/RIL/SS/SSE

個々の命令に関する説明の中のフォーマット図では、命令コード・フィールドに、単一引用符で囲んだ16進数で命令コードを示してあります。16進数表現では、0-9は2進数コード0000-1001を表し、A-Fは2進数コード1010-1111を表します。

各命令のフォーマット図の中のその他のフィールドは、コード名で示されています。コード名は、1 個の英字から成り、多くの場合英字の後に添え字番号が付いています。添え字番号は、そのフィールドが適用されるオペランドを表しています。

レジスター・オペランド

RR、RRE、RRF、RX、RXE、RXF、RS、RSE、RSI、RI、RIE、および RIL フォーマットでは、R₁ フィールドが指すレジスターの内容が、第 1 オペランドと呼ばれます。第 1 オペランドが入っているレジスターは、「第 1 オペランド位置」と呼ばれ、またときには「レジスター R₁」と呼ばれることもあります。RR、RRE、および RRF フォーマットでは、R₂ フィールドは第 2 オペランドが入っているレジスターを指します。R₂ フィールドは、R₁ と同じレジスターを指していることもあります。RRF、RXF、RS、RSE、RSI、および RIE フォーマットで、R₃ フィールドが使用されるかどうかは、命令によって異なります。RS および RSE フォーマットでは、R₃ フィールドは、代わりに、マスクを指定する M₃ フィールドの場合もあります。

R フィールドは、一般命令では汎用レジスターまたはアクセス・レジスターを指し、制御命令では汎用レジスターを指し、浮動小数点命令では浮動小数点レジスターを指します。ただし、EXTRACT STACKED REGISTERS および LOAD ADDRESS EXTENDED 命令では、R フィールドは汎用レジスターとアクセス・レジスターの両方を指し、LOAD CONTROL および STORE CONTROL 命令では、R フィールドは制御レジスターを指します。(この段落の内容はレジスター・オペランドのみに適用されるもので、記憶域オペランドをアドレッシングするときのアクセス・レジスターの使用には適用されません。)

アクセス・レジスターと浮動小数点レジスターの場合は、個々の命令の説明の中で特に断りがない限り、レジスター・オペランドの長さは、レジスター 1 個分(アクセス・レジスターの場合は 32 ビット、浮動小数点レジスターの場合は 64 ビット)で、第 2 オペランドの長さは第 1 オペランドと同じです。汎用レジスターと制御レジスターの場合は、レジスター・オペランドは、命令に応じて、64 ビット・レジスターのビット位置 32-63 を占める場合と、レジスター全体を占める場合があります。

即値オペランド

SI フォーマットでは、8 ビットの即値データ・フィールド(命令の I₂ フィールド)の内容が、そのまま第 2 オペランドとして使用されます。第 1 オペランドは B₁ フィールドと D₁ フィールドに指定され、長さは 1 バイトです。

ADD HALFWORD IMMEDIATE、COMPARE HALFWORD IMMEDIATE、LOAD HALFWORD IMMEDIATE、および MULTIPLY HALFWORD IMMEDIATE 命令の RI フォーマットでは、命令の 16 ビットの I₂ フィールドがそのまま符号付き 2 進整数として使用されます。また、R₁ フィールドは、第 1 オペランドを指定します。第 1 オペランドの長さは、命令に応じて、32 ビットの場合と 64 ビットの場合があります。TEST UNDER MASK (TMHH、TMHL、TMLH、TMLL) 命令の場合は、I₂ フィールドの内容はマスクとして使用され、R₁ フィールドは 64 ビットの長さの第 1 オペランドを指定します。INSERT IMMEDIATE、AND IMMEDIATE、OR IMMEDIATE、および LOAD LOGICAL IMMEDIATE 命令では、I₂ フィールドの内容は符号なし 2 進整数または論理値として使用され、R₁ フィールドは 64 ビットの長さの第 1 オペランドを指定します。

RI および RSI フォーマットの相対ブランチ命令の場合は、16 ビットの I₂ フィールドの内容が、ハーフワードの数を示す符号付き 2 進整数として使用されます。この数値をブランチ命令のアドレスに加えたものが、ブランチ・アドレスを示します。RIL フォーマットでは、I₂ フィールドは 32 ビットですが、用途は同じです。

記憶域オペランド

B フィールドおよび R フィールドを使用して、記憶域オペランドを指すアクセス・レジスターを指定する方法については、5-34ページの『アクセス・レジスター指定のアドレス・スペース』で説明します。

RSL、SI、SS、および SSE フォーマットでは、B₁ フィールドが指す汎用レジスターの内容が D₁ フィールドの内容に加えられ、その結果が第 1 オペランドのアドレスとなります。RS、RSE、S、SS、および SSE フォーマットでは、B₂ フィールドが指す汎用レジスターの内容が D₂ フィールドの内容に加えられ、その結果が第 2 オペランドのアドレスとなります。RX、RXE、および RXF フォーマットでは、X₂ フィールドと B₂ フィールドが指す汎用レジスターの内容が D₂ フィールドの内容に加えられ、その結果が第 2 オペランドのアドレスとなります。

汎用レジスターに、記憶域オペランドの 24 ビットまたは 32 ビットの長さが含まれている場合は、その長さは符号なし 2 進整数です。ただし、COMPARE UNTIL SUBSTRING EQUAL の場合は、この長さは符号付きの値であり、負の値は 0 と見なされます。同様に、命令の L、L₁、または L₂ フィールドの内容も、符号なし 2 進整数です。

8 ビットの長さフィールドを 1 つだけ持つ SS フォーマットでは、AND (NC)、EXCLUSIVE OR (XC)、MOVE (MVC)、MOVE NUMERICS、MOVE ZONES、および OR (OC) 命令の L フィールドは、第 1 オペランド・アドレスが指すバイトの右側に追加するオペランド・バイトの数を指定します。したがって、第 1 オペランドの長さは、L の長さコード 0-255 に対応して、1-256 バイトになります。格納結果は第 1 オペランドと置き換わり、アドレスと長さで指定されているフィールドの外部に格納されることはありません。このフォーマットでは、第 2 オペランドは第 1 オペランドと同じ長さになります。上記の定義には、EDIT、EDIT AND MARK、PACK ASCII、PACK UNICODE、TRANSLATE、TRANSLATE AND TEST、UNPACK ASCII、および UNPACK UNICODE に適用されるバリエーションがあります。

2 つの長さフィールドを持つ SS フォーマット、および RSL フォーマットでは、L₁ は、第 1 オペランド・アドレスが指すバイトの右側に追加するオペランド・バイトの数を指定します。したがって、第 1 オペランドの長さは、L₁ の中の長さコード 0-15 に対応して、1-16 バイトになります。同様に、L₂ は、第 2 オペランド・アドレスが指す記憶位置の右側に追加するオペランド・バイトの数を指定します。結果は第 1 オペランドと置き換わり、アドレスと長さで指定されているフィールドの外部に格納されることはありません。第 1 オペランドが第 2 オペランドより長い場合は、第 2 オペランドは第 1 オペランドの長さに等しくなるまで左方向に拡張され、その拡張部分にはすべて 0 が入ります。この拡張によって、記憶域内の第 2 オペランドが変更されることはありません。

MOVE TO PRIMARY、MOVE TO SECONDARY、および MOVE WITH KEY 命令に使用される、2 つの R フィールドを持つ SS フォーマットでは、R₁ フィールドが指す汎用レジスターの内容は 32 ビットの符号なしの値で、これは真の長さと呼ばれます。オペランドの長さは、どちらも、有効長と呼ばれる長さです。有効長は、真の長さとして 256 の、どちらか小さい方の長さです。命令では、条件コードを設定することによって、真の長さで指定された総バイト数を移動するためのループをプログラミングしやすくします。2 つの R フィール

ドを持つ SS フォーマットは、LOAD MULTIPLE DISJOINT 命令に使用する一連のレジスターおよび 2 つの記憶域オペランドを指定するため、また、PERFORM LOCKED OPERATION 命令に使用する 1 つまたは 2 つのレジスターと 1 つまたは 2 つの記憶域オペランドを指定するためにも、使用されます。

アドレスの生成

3 モード・アドレッシング

現 PSW のビット 31 と 32 は、アドレッシング・モード・ビットです。ビット 31 は拡張アドレッシング・モード・ビットで、ビット 32 は基本アドレッシング・モード・ビットです。これらのビットは、アドレス生成により作成される有効アドレスのサイズを制御します。現 PSW のビット 31 と 32 が両方とも 0 のときは、CPU は 24 ビット・アドレッシング・モードになっており、24 ビットの命令有効アドレスおよびオペランド有効アドレスが生成されます。現 PSW のビット 31 が 0 で、ビット 32 が 1 のときは、CPU は 31 ビット・アドレッシング・モードになっており、31 ビットの命令有効アドレスおよびオペランド有効アドレスが生成されます。現 PSW のビット 31 と 32 が両方とも 1 のときは、CPU は 64 ビット・アドレッシング・モードになっており、64 ビットの命令有効アドレスおよびオペランド有効アドレスが生成されます。

CPU が命令を実行するときに、命令およびオペランドのアドレスが生成されます。このセクションでは、ほとんどの命令で行われるアドレス生成について説明します。一部の命令では、この操作がこのセクションで述べる一般規則には従っていない場合もあります。このような例外は、すべて、該当する個々の命令の説明の中で明確に示してあります。

命令アドレスの順次生成

現 PSW が指す記憶位置から命令が取り出されると、命令アドレスがその命令に含まれるバイト数だけ増加してから、命令が実行されます。次の順番に当たっている命令を取り出すときは、新しい命令アドレスの値を使用して、同じステップが繰り返されます。

24 ビット・アドレッシング・モードでは、命令アドレスは循環し、命令アドレス $2^{24} - 2$ の位置のハーフワードの次には、命令アドレス 0 の位置のハーフワードが続きます。したがって、24 ビット・アドレッシング・モードでは、命令アドレスの更新の結果として PSW の

ビット位置 104 からの繰り上がりが生じる場合、その繰り上がりは失われます。31 ビットまたは 64 ビット・アドレッシング・モードの場合も、命令アドレスは同様に循環し、31 ビットの場合は命令アドレス $2^{31} - 2$ の位置、64 ビットの場合は命令アドレス $2^{64} - 2$ の位置にあるハーフワードの次には、命令アドレス 0 の位置にあるハーフワードが続きます。それぞれ PSW ビット位置 97 または 64 からの繰り上がりは失われます。

オペランド・アドレスの生成

中間値の形成

記憶域を参照するオペランド・アドレスは、中間値から導き出されます。中間値は、命令の R フィールドに指定されているレジスターに含まれているか、または、基底アドレス、指標、および変位の 3 つの 2 進数の和として求められます。

基底アドレス (B) は、命令内の B フィールドと呼ばれる 4 ビット・フィールドにプログラムで指定された汎用レジスターに含まれている、64 ビットの数値です。基底アドレスは、個々のプログラムおよびデータ域を個別にアドレッシングするための手段として使用できます。このアドレスは、配列型の計算では配列の位置を示し、レコード型の処理ではレコードを識別することができます。基底アドレスを利用して、記憶域全体をアドレッシングすることができます。また、基底アドレスは指標付けにも使用できます。

指標 (X) は、命令内の X フィールドと呼ばれる 4 ビット・フィールドにプログラムで指定された汎用レジスターに含まれている、64 ビットの数値です。指標は、RX、RXE、および RXF フォーマットの命令で指定されたアドレスの中のみ含まれています。RX、RXE、および RXF フォーマットの命令では、二重の指標付けが可能です。つまり、指標を使用して、配列の中の特定要素のアドレスを指定することができます。

変位 (D) は、命令内の D フィールドと呼ばれるフィールドに含まれている 12 ビットの数値です。変位を利用することにより、基底アドレスが指定する記憶位置を超えた最大 4,095 バイトまでの範囲内の相対アドレッシングが可能です。配列型の計算では、変位を使用して、1 つの要素に関連した多数の項目のうちの 1 つを指定することができます。レコードの処理では、変位を使用して、レコード内の項目を識別することができます。

中間合計値を形成する際には、基底アドレスと指標はどちらも 64 ビットの 2 進整数として扱われます。同様

に、変位は 12 ビットの符号なし 2 進整数として扱われ、左側に 52 個の 0 ビットが付加されます。この 3 つの値は 64 ビット 2 進数として加算され、オーバーフローは無視されます。合計値は常に 64 ビットの長さで、生成アドレスを形成するための中間値として使用されます。中間値のビットには、0-63 の番号が割り当てられます。

B₁、B₂、または X₂ フィールドのいずれかが 0 である場合は、それに対応するアドレス・コンポーネントが存在しないことを示します。存在しないコンポーネントについては、汎用レジスター 0 の内容に関係なく、中間合計値を形成する際に 0 が使用されます。変位が 0 であるということには、特別な意味はありません。

個々の命令の説明の中で、R フィールドが指している汎用レジスターの内容を使用して記憶域内のオペランドをアドレッシングすることが指定されている場合は、そのレジスターの内容が 64 ビットの中間値として使用されます。

1 つの命令で、アドレス計算とオペランド位置の両方に同じ汎用レジスターを指定できます。アドレス計算は、操作によりレジスターが変更される前に完了します。

個々の命令の定義の中で特に断り書きがない限り、生成されるオペランド・アドレスは、記憶域内でのオペランドの最左端のバイトを指します。

オペランド・アドレスの形成

生成されるオペランド・アドレスは常に 64 ビットの長さで、個々のビットには 0-63 の番号が割り当てられます。生成アドレスが中間値からどのように形成されるかは、現行アドレッシング・モードによって決まります。24 ビット・アドレッシング・モードでは、中間値のビット 0-39 は無視され、生成アドレスのビット 0-39 は 0 に強制的に設定され、中間値のビット 40-63 が、生成アドレスのビット 40-63 になります。31 ビットアドレッシング・モードでは中間値のビット 0-32 は無視され、生成アドレスのビット 0-32 は 0 に強制的に設定され、中間値のビット 33-63 が、生成アドレスのビット 33-63 になります。64 ビット・アドレッシング・モードでは、中間値のビット 0-63 が、生成アドレスのビット 0-63 になります。

プログラミング上の注意: 指標レジスターおよび基底レジスターの中では、負の値を使用できます。負の値の場合、31 ビット・アドレッシング・モードではビット 0-32 は無視され、24 ビット・アドレッシング・モードではビット 0-39 は無視されます。

ブランチ・アドレスの生成

中間値の形成

ブランチ命令において、ブランチがとられたときに次に実行すべき命令のアドレスを、ブランチ・アドレスと呼びます。使用できる命令フォーマットは、ブランチ命令に応じて、RR、RRE、RX、RXE、RS、RSE、RSI、RI、RIE、またはRILのいずれかです。

ブランチ・アドレスは、RS、RSE、RX、およびRXEフォーマットでは、基底アドレスと変位により指定され、RXおよびRXEフォーマットでは、基底アドレス、変位、および指標により指定されます。これらのフォーマットにおける中間値の生成は、オペランド・アドレスの中間値と同じ規則に従って行われます。

RRおよびRREフォーマットでは、R₂フィールドが指す汎用レジスタの内容が、ブランチ・アドレス形成の元となる中間値として使用されます。汎用レジスタ0を、ブランチ・アドレスを含むものとして指定することはできません。命令のR₂フィールドが0の場合は、その命令はブランチなしで実行されます。

相対ブランチ命令のフォーマットは、RSI、RI、RIE、およびRILです。RSI、RI、およびRIEフォーマットの相対ブランチ命令では、I₂フィールドの内容は、ハーフワードの数を示す16ビットの符号付き2進整数として扱われます。RILフォーマットでは、I₂フィールドの内容は、ハーフワードの数を示す32ビットの符号付き2進整数として扱われます。ブランチ・アドレスは、相対ブランチ命令のアドレスに、I₂フィールドに指定されているハーフワード数を加えたものです。

RSI、RI、RIE、またはRILフォーマットの相対ブランチ命令の場合の64ビットの中間値は、2つの加数の和であり、ビット位置0からのオーバーフローは無視されます。RSI、RI、またはRIEフォーマットでは、第1加数は、I₂フィールドの内容の右側に1個の0ビットを付加し、左側にその内容の符号ビットと同じビットを47個付加したものです。RILフォーマットでは、第1加数は、I₂フィールドの内容の右側に1個の0ビットを付加し、左側にその内容の符号ビットと同じビットを31個付加したものです。第2加数は、どのフォーマットの場合も、ブランチ命令の64ビット・アドレスです。ブランチ命令のアドレスは、次の順番に当たる命令をアドレッシングするために更新される前のPSW内の命令アドレスか、または、EXECUTE命令が使用され

ている場合はEXECUTEのターゲットのアドレスです。EXECUTEを24ビットまたは31ビット・アドレッシング・モードで使用している場合は、ブランチ命令のアドレスは、ターゲット・アドレスの左側にそれぞれ40個または33個の0を付加したものです。

ブランチ・アドレスの形成

ブランチ・アドレスは常に64ビットの長さで、個々のビットには0-63の番号が割り当てられます。ブランチ・アドレスは、現PSWのビット64-127と置き換わります。

ブランチ・アドレスが中間値からどのように形成されるかは、アドレッシング・モードによって決まります。アドレッシング・モードを変更するブランチ命令の場合は、変更後の新しいアドレッシング・モードが使用されます。24ビット・アドレッシング・モードでは、中間値のビット0-39は無視され、ブランチ・アドレスのビット0-39は0に強制的に設定され、中間値のビット40-63が、ブランチ・アドレスのビット40-63になります。31ビットアドレッシング・モードでは、中間値のビット0-32は無視され、ブランチ・アドレスのビット0-32は0に強制的に設定され、中間値のビット33-63が、ブランチ・アドレスのビット33-63になります。64ビット・アドレッシング・モードでは、中間値のビット0-63が、ブランチ・アドレスのビット0-63になります。

幾つかのブランチ命令では、指定された条件を満たしているかどうかによって、ブランチするかどうかが決まる場合があります。条件が満たされていない場合は、ブランチは行われず、命令の実行は通常の順序で続行され、ブランチ・アドレスは使用されません。ブランチが行われる場合は、ブランチ・アドレスのビット0-63が現PSWのビット64-127と置き換わります。ブランチ・アドレスは、ブランチ操作の一環として記憶域にアクセスするためには使用されません。

ブランチ・アドレスが奇数であることが原因で起きる指定例外、およびブランチ位置にある命令の取り出しが原因で起きるアクセス例外は、ブランチ操作の一環として認識されるのではなく、ブランチ位置にある命令の実行に関連した例外として認識されます。

1つのブランチ命令(例えば、BRANCH AND SAVE)で、ブランチ・アドレス計算とオペランド位置の両方に同じ汎用レジスタを指定できます。ブランチ・アドレス計算が完了してから、操作の残りの部分が実行されず。

命令の実行と順序付け

第4章、『制御』で説明したプログラム状況ワード (PSW) には、プログラムの適正な実行のために必要な情報が含まれています。PSW は、命令の順序を制御し、現在実行中のプログラムとの関連における CPU の状況を保持および提示するために、使用されます。アクティブな PSW、つまり制御効力を持っている PSW を、現行 PSW と呼びます。

ブランチ命令は、判断処理、ループ制御、およびサブルーチン・リンケージの機能を実行します。ブランチ命令は、現 PSW に新しい命令アドレスを導入することにより、命令の順序を変更します。16 ビットの I₂ フィールドを持つ相対ブランチ命令を使用すると、基底レジスターを使用せずに、ブランチ命令の位置を基準として、プラス 64K - 2 バイトまで、またはマイナス 64K バイトまでの範囲内のオフセット位置に、ブランチすることができます。32 ビットの I₂ フィールドを持つ相対ブランチ命令では、基底レジスターを使用せずに、ブランチ命令の位置を基準として、プラス 4G - 2 バイトまで、またはマイナス 4G バイトまでの範囲内のオフセット位置に、ブランチすることができます。

判断処理

判断処理の機能を持つ命令は、BRANCH ON CONDITION、BRANCH RELATIVE ON CONDITION、および BRANCH RELATIVE ON CONDITION LONG です。これらの命令は、ほとんどの算術操作、論理操作、および入出力操作の結果を反映する条件コードを検査します。条件コードは 2 ビットから成り、0、1、2、3 の 4 通りの条件コード設定が可能です。

各設定の意味は、条件コードを設定する操作によって異なります。例えば、条件コードは、0 である、0 以外、第 1 オペランドが高い、等しい、オーバーフロー、サブチャンネル使用中などの条件を反映します。いったん設定された条件コードは、別の条件コードを設定させる命令により変更されるまで、同じ設定のままです。条件コードを設定する命令の要約については、C-1ページの付録 C、『条件コードの設定』を参照してください。

ループ制御

BRANCH ON CONDITION、BRANCH RELATIVE ON CONDITION、または BRANCH RELATIVE ON CONDITION LONG 命令を使用して、アドレスの計算およびカウンタ操作の結果をテストすることにより、ループ制御を行うことができます。特に使用頻度の高い算術とテストの組み合わせに使用する命令として、BRANCH ON COUNT、BRANCH ON INDEX HIGH、および BRANCH ON INDEX LOW OR EQUAL が提供されており、また、これらの命令と同じ働きをする相対ブランチも提供されています。これらのブランチはこれらの作業専用に変換されているので、作業効率が向上します。

リンケージ・スタックなしのサブルーチン・リンケージ

このセクションでは、リンケージ・スタックを使用しないサブルーチン・リンケージの方法についてのみ説明します。リンケージ・スタックにより提供されるリンケージ拡張機能については、5-59ページの『リンケージ・スタックの概要』で説明します。(これらの拡張機能には、別の方法を用いた PROGRAM CALL 命令の操作のほか、BRANCH AND STACK および PROGRAM RETURN 命令が含まれています。)

単純ブランチ命令

アドレッシング・モードの変更が必要ない場合のサブルーチン・リンケージを行う命令には、BRANCH AND LINK および BRANCH AND SAVE があります。(BRANCH AND SAVE に関する以下の説明は、BRANCH RELATIVE AND SAVE および BRANCH RELATIVE AND SAVE LONG にも適用されます。)
この 2 つの命令は、どちらも、新しい命令アドレスを導入するだけでなく、戻りアドレスとそれに関連した情報を保存することもできます。戻りアドレスは、記憶域内のブランチ命令の後に続く命令のアドレスです。ただし、ブランチ命令をターゲットとする EXECUTE 命令の場合は、EXECUTE の後に続く命令のアドレスです。

BRANCH AND LINK と BRANCH AND SAVE のどちらにも、R₁ フィールドがあります。これらの命令は、それぞれの命令に応じたフィールドを使用して、ブランチ・アドレスを形成します。以下に、これらの命令の働きを要約して示します。

- 24 ビット・アドレッシング・モードでは、どちらの命令も、汎用レジスター R₁ のビット位置 40-63 に戻りアドレスを入れ、ビット 0-31 は変更しませ

ん。BRANCH AND LINK は、汎用レジスタ R₁ のビット位置 32-39 に、命令長コードと、PSW から取り出した条件コードおよびプログラム・マスクを入れます。BRANCH AND SAVE は、これらのビット位置に 0 を入れます。

- 31 ビット・アドレッシング・モードでは、どちらの命令も、汎用レジスタ R₁ のビット位置 33-63 に戻りアドレスを入れ、ビット位置 32 に 0 を入れ、ビット 0-31 は変更しません。
- 64 アドレッシング・モードでは、どちらの命令も、汎用レジスタ R₁ のビット位置 0-63 に戻りアドレスを入れます。
- どのアドレッシング・モードのときも、どちらの命令も現行アドレッシング・モードの制御下でブランチ・アドレスを生成します。これらの命令は、ブランチ・アドレスのビット 0-63 を PSW のビット位置 64-127 に入れます。RR フォーマットでは、R₂ フィールドが 0 のときは、どちらの命令もブランチを行いません。

24 ビットまたは 31 ビット・アドレッシング・モードでは、BRANCH AND SAVE は、基本アドレッシング・モード・ビット (PSW のビット 32) を、汎用レジスタ R₁ のビット位置 32 に入れることがあります。BRANCH AND LINK は、31 ビット・アドレッシング・モードの場合に同じことを行います。

BRANCH AND SAVE AND SET MODE および BRANCH AND SET MODE 命令は、リンケージ中にアドレッシング・モードの変更が必要な場合に使用するのためのものです。これらの命令には、R₁ フィールドと R₂ フィールドがあります。以下に、これらの命令の働きを要約して示します。

- BRANCH AND SAVE AND SET MODE は、汎用レジスタ R₁ の内容を BRANCH AND SAVE の場合と同じに設定します。さらに、この命令は、拡張アドレッシング・モード・ビット (PSW のビット 31) を、レジスタのビット位置 63 に入れます。
- BRANCH AND SET MODE は、R₁ が 0 以外のときは、以下のことを行います。24 ビットまたは 31 ビット・モードでは、この命令は、PSW のビット 32 を汎用レジスタ R₁ のビット 32 に入れ、レジスタのビット 0-31 および 33-63 は変更しません。このレジスタに命令アドレスが含まれている場合は、レジスタのビット 63 は 0 でなければならぬという点に注意してください。64 ビッ

ト・モードでは、この命令は、PSW のビット 31 (1 つ) を汎用レジスタ R₁ のビット位置 63 に入れ、ビット 0-62 は変更しません。

- R₂ が 0 以外のときは、どちらの命令も、以下のように入ります。アドレッシング・モードを設定し、ブランチを行います。汎用レジスタ R₂ のビット 63 が、PSW のビット位置 31 に入れられます。ビット 63 が 0 の場合は、レジスタのビット 32 が PSW のビット位置 32 に入れられます。ビット 63 が 1 の場合は、PSW のビット 32 は 1 に設定されます。その後は、新しいアドレッシング・モードの制御下で、レジスタの内容に基づいてブランチ・アドレスが生成されます。ただし、レジスタのビット 63 は 0 と見なされます。これらの命令は、ブランチ・アドレスのビット 0-63 を PSW のビット位置 64-127 に入れます。汎用レジスタ R₂ のビット 63 は変更されないまま残されているので、呼び出し先プログラムに入る時点で 1 になっていることがあります。R₂ が R₁ と同じである場合は、指定された汎用レジスタ内の結果は、R₁ レジスタについて指定されている結果と同じになります。

5-12ページの図5-2に、単純ブランチ命令の働きを要約して示します。この図には、対比のために、BRANCH AND COUNT 命令 (リンケージ用には使用されない)、および LOAD ADDRESS および LOAD ADDRESS EXTENDED 命令も示してあります。

プログラミング上の注意:

1. 64 ビット・アドレッシング・モードに入る呼び出し先プログラムは、エンタリー・ポイント・レジスタのビット 63 を使用して、呼び出しを行うために使用する命令、つまり、戻りリンケージを行うために使用する命令を判別することができます。ビット 63 が 0 の場合は、BRANCH AND SAVE (BAS または BASR) (または、場合によっては BAL、BALR、BRAS、または BRASL) が使用されていて、呼び出し側のアドレッシング・モードは戻りレジスタに示されているモードではなく、そして、アドレッシング・モードを変更せずに戻するために BRANCH ON CONDITION (BCR) を使用する必要があります。エンタリー・ポイント・レジスタのビット 63 が 1 の場合は、BASSM または BSM が使用されており、呼び出し側のアドレッシング・モードは戻りレジスタに示されているモードであり (または、BSM の場合は少なくともその可能性があり)、そして、戻ってから呼び出し側のアドレッシング・モードを復元するために BSM を使用する必要があります。

2. BSM が 24 ビットまたは 31 ビットのアドレッシング・モードで実行され、64 ビット・モードを設定するためのフォワード・リンケージの中で使用されており、命令の R₁ と R₂ が同じ値である場合は、呼び出し先プログラムに入るときに、指定された汎用レジスタのビット 63 は呼び出し側プログラムのモードを正しく示しません。(プログラムは R₂ レジスタのビット 63 を 1 に設定し、命令は、24 ビット・モードまたは 31 ビット・モードでは R₁ レジスタのビット 63 を変更しないので、

このビットは 0 でなく、1 になります。) BASSM は、常に、呼び出し側プログラムのアドレッシング・モードを正しく示します。

3. BAS か BASR または BASSM か BSM により、エンタリー・ポイントに 64 ビット・アドレッシング・モードでブランチできる場合に、BASSM または BSM の場合、アドレス可能性を提供する USING ステートメントでエンタリー・ポイント・レジスタを指定するときは、そのレジスタから 1 を減算する必要があります。

命令	フォーマット	モード	GR R ₁ に 入れられるアドレス				ブランチまたは 第2オペランド・ アドレス		R ₂ ビット 63	PSW ビット 31 の 設定	PSE ビット 32 の 設定
			ビット 0-31	ビット 32	ビット 33-62	ビット 63	ビット 0-32	ビット 33-63			
BALR*/BAL	RR/RX	24	U	***	***	IA	SIA	SIA	LSExc	U	U
		31	U	BAM	IA	IA	SIA	SIA	LSExc	U	U
		64	IA	IA	IA	IA	SIA	SIA	LSExc	U	U
BASR*/BAS/ BRAS/BRASL	RR/RX/ RI/RIL	24/31	U	BAM	IA	IA	SIA	SIA	LSExc	U	U
		64	IA	IA	IA	IA	SIA	SIA	LSExc	U	U
BASSM*	RR	24/31	U	BAM	IA	IA	SIA	SIA	0	0	R232
		24/31	U	BAM	IA	IA	SIA	SIA	1G0	1	1
		64	IA	IA	IA	1	SIA	SIA	0	0	R232
		64	IA	IA	IA	1	SIA	SIA	1G0	1	1
BSM**	RR	24/31	U	BAM	U	U	SIA	SIA	0	0	R232
		24/31	U	BAM	U	U	SIA	SIA	1G0	1	1
		64	U	U	U	1	SIA	SIA	0	0	R232
		64	U	U	U	1	SIA	SIA	1G0	1	1
BCTR*/BCT/ BCTGR*/BCTG	RR/RX/ RRE/RXE	24/31	NLA	NLA	NLA	NLA	SIA	SIA	LSExc	U	U
		64	NLA	NLA	NLA	NLA	SIA	SIA	LSExc	U	U
LA/LAE	RX/RX	24/31	U	0	Op2Ad	Op2Ad	FZ	SR1	0/1	U	U
		64	Op2Ad	Op2Ad	Op2Ad	Op2Ad	SR1	SR1	0/1	U	U

図 5-2 (1/2). 単純ブランチ・リンケージ命令と他の命令の要約

説明:

-	アドレスが存在しないか、ビットに特別の効力はありません。
*	R ₂ フィールドが 0 の場合は、このフィールドに関連したアクションは行われません。
**	R ₁ または R ₂ フィールドが 0 の場合は、そのフィールドに関連したアクションは行われません。
***	リンク・アドレスのビット位置 32-39 には命令長コード、条件コード、およびプログラム・マスクが保管され、ビット位置 40-63 には更新済みの命令アドレスのビット 40-63 が保管されます。
0/1	ビット 63 の値は 0 または 1 のいずれかです。
1G0	ビット 63 は 1 で、1 のまま残されますが、ブランチ・アドレスはこのビットが 0 であるものとして生成されます。
BAM	リンク・アドレスのビット 32 は、基本アドレッシング・モード・ビット (PSW のビット32) に設定されます。
FZ	24 ビットまたは 31 ビット・アドレッシング・モードでは、第 2 オペランド・アドレスのビット0-32 は 0 に設定されます。
IA	リンク・アドレスのビットは、更新済みの命令アドレスの対応するビットに設定されます。
LSExc	このビットが 1 の場合は、遅延指定例外が認識されます。
NLA	この命令はリンク・アドレスを生成しません。(この命令は、非リンケージ・ブランチ命令の例として示してあるだけです。)
Op2Ad	汎用レジスタ R ₁ が示すアドレスのビットは、第 2 オペランド・アドレスの対応するビットに設定されます。
R232	基本アドレッシング・モード・ビット (PSW のビット 32) は、汎用レジスタ R ₂ のビット 32 に設定されます。
SIA	ブランチ・アドレスのビット 0-63 は、PSW 内の命令アドレスを設定するために使用されます。24 ビット・アドレッシング・モードでは、ブランチ・アドレスのビット0-39 は 0 に設定されます。31 ビット・アドレッシング・モードでは、ビット 0-32 は 0 に設定されます。
SR1	第 2 オペランド・アドレスのビットは、R ₁ 汎用レジスタが指すアドレスの対応するビットを設定するために使用されます。24 ビット・アドレッシング・モードでは、第 2 オペランド・アドレスのビット 0-39 は 0 に設定されます。31 ビット・アドレッシング・モードでは、ビット 0-32 は 0 に設定されます。
U	変更されません。

図 5-2 (2/2). 単純ブランチ・リンケージ命令と他の命令の要約

その他のリンケージ命令

SUPERVISOR CALL 命令および **MONITOR CALL** 命令は、問題プログラム状態プログラムと、監視プログラムまたはモニター・プログラムとの間のリンケージを提供します。

PROGRAM CALL 命令および **PROGRAM**

TRANSFER 命令は、異なるアドレス・スペース内にある、権限の異なるプログラム間のリンケージ機能を提供します。**PROGRAM CALL** では、問題プログラム状態または監視プログラム状態にあり、かつ呼び出し側と同じアドレス・スペースまたは異なるアドレス・スペース内にある可能性がある、幾つかの事前割り当て済みプロ

グラムへのリンケージが可能です。この命令により、アドレッシング・モードを 24 ビットと 31 ビットの間で変更でき、また、PSW キー・マスク権限を増加することができます。この権限は、**SET PSW KEY FROM ADDRESS** 命令およびその他の機能を実行する権限です。一般に、**PROGRAM CALL** は、権限の大きいプログラムに制御を転送するために使用されます。

PROGRAM TRANSFER では、命令アドレスおよびアドレス・スペースを変更できるほか、アドレッシング・モードを 24 ビットと 31 ビットの間で変更できます。また、**PROGRAM TRANSFER** により、PSW キー・マスク権限を減少させること、および監視プログラム状態から問題プログラム状態に変更することもできます。

通常は、この命令は、1つのプログラムから、権限がそれと同じかもっと小さいプログラムに制御を転送するために使用されます。

呼び出しリンケージで権限を増加する必要がある場合は、PROGRAM CALL により呼び出しリンケージを行い、戻りリンケージを PROGRAM TRANSFER により行うことができます。逆に、呼び出しリンケージで権限を減少させる必要がある場合は、PROGRAM TRANSFER により呼び出しリンケージを行い、PROGRAM CALL により戻りリンケージを行うことができます。

PROGRAM CALL の働きは、エントリー・テーブル・エントリーを使用して制御されます。このエントリーは、命令の実行時にテーブル索引プロセスの一環として検索されます。エントリー・テーブル・エントリーは、基本（非スタッキング）操作またはスタッキング操作（5-59ページの『リンケージ・スタックの概要』を参照）のいずれかを指定します。この命令により1次アドレス・スペースが変更されるのは、エントリー・テーブル・エントリー内の ASN が 0 以外の場合のみです。1次アドレス・スペースが変更される場合、その操作は、スペース切り替えを伴う PROGRAM CALL (PC-ss) と呼ばれます。1次アドレス・スペースが変更されない場合は、その操作は、現1次に対する PROGRAM CALL (PC-cp) と呼ばれます。

PROGRAM TRANSFER は、新しい1次アドレス・スペースとなるアドレス・スペースを指定します。1次アドレス・スペースが変更される場合、その操作は、スペース切り替えを伴う PROGRAM TRANSFER (PT-ss) と呼ばれます。1次アドレス・スペースが変更されない場合は、その操作は現1次への PROGRAM TRANSFER (PT-cp) と呼ばれます。

基本 PROGRAM CALL および PROGRAM TRANSFER は、基本アドレッシング・モード（24ビットまたは31ビット）でも、拡張アドレッシング・モード（64ビット）でも、正常に実行できます。これらの命令では、基本アドレッシング・モードと拡張アドレッシング・モードの間の変更は行われません。

BRANCH AND SET AUTHORITY 命令は、PSW キー・マスク権限が減少する呼び出しリンケージを行うために使用される PT-cp 命令、および、PSW キー・マスク権限が復元される関連の戻りリンケージを行うために使用される PC-cp の代わりに使用でき、これによりパフォーマンスが向上します。BRANCH AND SET AUTHORITY は、監視プログラム状態と問題プログラム状態を切り替える機能も備えており、さらに、リンケ

ージ中に PSW キーを変更することによって、SET PSW KEY FROM ADDRESS に代わって使用することもできます。呼び出しリンケージ操作は、基本権限状態の BRANCH AND SET AUTHORITY (BSA-ba) と呼ばれ、戻りリンケージ操作は、縮小権限状態の BRANCH AND SET AUTHORITY (BSA-ra) と呼ばれます。

BRANCH IN SUBSPACE GROUP 命令を使用すると、サブスペース・グループと呼ばれるアドレス・スペースのグループ内でのリンケージが可能です。この場合、グループ内には基本スペースと呼ばれるアドレス・スペースが1つあり、その他のアドレス・スペースはサブスペースと呼ばれます。各サブスペースには、それぞれ基本スペース内の異なる記憶域サブセットが含まれており、基本スペースと各サブスペースにはそれぞれサブシステム制御プログラム (CICS など) とアプリケーション・プログラムが含まれており、各サブスペースには、サブシステム制御プログラムの下で処理されている単一トランザクション用のデータが含まれています。個々のトランザクションごとに異なるサブスペースにデータを含めておくことで、1つのトランザクションを処理するために実行されているプログラムが、誤って他のトランザクションのデータを損傷するのを防ぐことができます。1次アドレス・スペースは、制御プログラムが実行されているときは基本スペースであり、トランザクション処理のためにアプリケーション・プログラムが実行されているときは、そのトランザクション用のサブスペースです。BRANCH IN SUBSPACE GROUP は、PSW 中の命令アドレスだけでなく、制御レジスター 1 中の1次アドレス・スペース制御エレメントも変更します。BRANCH IN SUBSPACE GROUP は、制御レジスター 4 中の1次 ASN、または制御レジスター 5 中の1次 ASN 第2テーブル・エントリー起点は変更しません。したがって、基本スペースおよびサブスペースはすべて同じ ASN に関連付けられており、これらのアドレス・スペース内のプログラムは、すべて同じ権限を持っています。

上記で述べたように、サブスペースは基本スペースのサブセットにするのが本来の目的ですが、それが BRANCH IN SUBSPACE GROUP の必須条件というわけではありません。この命令は、上記以外にも、さまざまな方法で利用できます。

BRANCH IN SUBSPACE GROUP は、アクセス・レジスター内のアクセス・リスト・エントリー・トークン (ALET) を、制御を受け取るアドレス・スペースの ID として使用します。この命令は、戻りリンケージを可能にするために更新済みの命令アドレスを保管しますが、制御がどのアドレス・スペースから転送されたかを示す ID は保管しません。しかし、16進数 00000000 に等しい ALET (ALET 0 と呼ばれる) を使用して、サブスペ

ースから基本スペースに戻ることができ、また、16 進数 00000001 に等しい ALET (ALET 1 と呼ばれる) を使用して、基本スペースから最後に制御権を持っていたサブスペースに戻ることができます。

SET ADDRESSING MODE (SAM24、SAM31、SAM64) 命令を使用すれば、リンケージ操作の前または後に、24 ビット、31 ビット、または 64 ビット・アドレッシング・モードを設定することにより、リンケージを効率化することができます。

RESUME PROGRAM 命令は、問題プログラム状態割り込み割り込み処理プログラムが、割り込まれたプログラムに戻るために使用します。割り込み処理プログラムは、LOAD ACCESS MULTIPLE および LOAD MULTIPLE 命令を使用して、保管域から、割り込まれたプログラムのアクセス・レジスターおよび汎用レジスターの内容を復元します。ただし、1 個のアクセス/汎用

レジスター・ペアの内容は復元しません。次に、割り込み処理プログラムは、RESUME PROGRAM を使用して、所定の PSW フィールドの内容 (命令アドレスも含む) を復元し、さらに、残っているアクセス/汎用レジスター・ペアの内容を保管域から復元することができます。このペアは、まず、保管域をアドレッシングするための RESUME PROGRAM により使用されます。

TRAP 命令 (TRAP2、TRAP4) は、アプリケーション・プログラムが使用するデータの修正を行うために、アプリケーション・プログラム内の命令をオーバーレイし、トラップ・プログラムに制御を渡すことができます。トラップ・プログラムからアプリケーション・プログラムに制御を戻すには、RESUME PROGRAM 命令を使用できます。

5-16ページの図5-3 に、提供されているリンケージ命令と、それぞれが行う機能の要約を示します。

命令	フォーマット	命令アドレス PSW ビット 64-127		基本アドレッシング・ モード PSW ビット 32		拡張アドレッシング・ モード PSW ビット 31		問題 プログラム 状態 PSW ビット 15		PASN CR4 ビット 48-63		CR3 で 変更され る PSW キー・ マスク	トレー ス
		保管	設定	保管	設定	保管	設定	保管	設定	保管	設定		
BALR	RR	Yes*	R ₂ ¹	BAM31	-	-	-	-	-	-	-	-	R ₂ ¹
BAL	RX	Yes*	Yes	BAM31	-	-	-	-	-	-	-	-	-
BASR	RR	Yes	R ₂ ¹	BAM	-	-	-	-	-	-	-	-	R ₂ ¹
BAS	RX	Yes	Yes	BAM	-	-	-	-	-	-	-	-	-
BASSM	RR	Yes	R ₂ ¹	BAM	R ₂ ¹	Yes	R ₂ ¹	-	-	-	-	-	-
BRAS	RI	Yes	Yes	BAM	-	-	-	-	-	-	-	-	-
BRASL	RIL	Yes	Yes	BAM	-	-	-	-	-	-	-	-	-
BSA-ba	RRE	Yes	Yes	BAM	BAM	-	-	Yes	Yes ⁴	-	-	"AND" R ₁ ⁵	Yes
BSA-ra	RRE	R ₁ ¹	Yes	R ₁ ¹ BAM	BAM	-	-	-	Yes	-	-	Yes	Yes
BSG	RRE	Yes	Yes	R ₁ ¹ BAM	BAM	-	-	-	-	-	- ³	-	Yes
BSM	RR	-	R ₂ ¹	R ₁ ¹ BAM	R ₂ ¹	R ₁ ¹ EAM64	R ₂ ¹	-	-	-	-	-	-
MC# ²	SI	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-
PC-cp	S	Yes	Yes	BAM	BAM	-	-	Yes	Yes	-	-	"OR" EKM	Yes
PC-ss	S	Yes	Yes	BAM	BAM	-	-	Yes	Yes	Yes	Yes	"OR" EKM	Yes
PT-cp	RRE	-	R ₂	-	R ₂ BAM	-	-	-	R ₂ **	-	-	"AND" R ₁	Yes
PT-ss	RRE	-	R ₂	-	R ₂ BAM	-	-	-	R ₂ **	-	Yes	"AND" R ₁	Yes
RP	S	-	Yes	-	Yes	-	Yes	-	-	-	-	-	Yes
SAM24	E	-	-	-	Yes 0	-	Yes 0	-	-	-	-	-	Yes
SAM31	E	-	-	-	Yes 1	-	Yes 0	-	-	-	-	-	Yes
SAM64	E	-	-	-	Yes 1	-	Yes 1	-	-	-	-	-	Yes
SVC ²	RR	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-
TRAP2	E	Yes	Yes	Yes	Yes	Yes	-	Yes	-	-	-	-	Yes
TRAP4	S	Yes	Yes	Yes	Yes	Yes	-	Yes	-	-	-	-	Yes

説明:

- No

* 24 ビット・アドレッシング・モードでは、命令長コード、条件コード、およびプログラム・マスクは、R₁ 汎用レジスタのビット位置 32-39 に保管されます。

図 5-3 (1/2). リンケージ・スタックなしのリンケージ命令の要約

説明 (続き):

**	監視プログラム状態から問題プログラム状態への変更が許されます。問題プログラム状態から監視プログラム状態への変更が指定されているときは、特権命令例外が認識されます。
#	モニター・マスク・ビットは、選択したイベント・クラスについて、リンケージを禁止または許可するための手段として使用できます。
1	命令内の関連する R フィールドが 0 でない場合のみ、処置がとられます。
2	MC および SVC は、割り込みの一環として、現 PSW の全体を保管し、新しい PSW をロードします。
3	PASN が設定されていなくても、1 次アドレス・スペース制御エレメントは設定されます。
4	問題プログラム状態が設定されます。
5	PSW キーは汎用レジスター R ₁ から設定されます。
BAM	基本アドレッシング・モード・ビットは、24 ビットまたは 31 ビット・アドレッシング・モードの場合のみ保管または設定されます。
BAM31	基本アドレッシング・モード・ビットは、31 ビット・アドレッシング・モードでのみ保管されます。
EAM64	拡張アドレッシング・モード・ビットは、64 ビット・アドレッシング・モードでのみ保管されます。
R ₁	フィールドまたはビットは汎用レジスター R ₁ に保管されます。
R ₂	フィールドまたはビットは汎用レジスター R ₂ から設定されます。

図 5-3 (2/2). リンケージ・スタックなしのリンケージ命令の要約

プログラミング上の注意: この注で述べるのは、370-XA で導入され、ESA/370、ESA/390、および z/Architecture へと受け継がれた単純ブランチ・タイプのリンケージ命令です。以下の項では、これらのリンケージ命令の有用性と用途についての読者の理解を深めるために、各種のプログラム・リンケージと規則、およびそれぞれの状況下でのリンケージ命令の使用方法について説明します。

リンケージ命令の当初の目的は、システム/370 のプログラムをまったく変更せずに、またはごくわずかな変更を加えるだけで、370/XA システム (およびその後継システム) で実行できるようにすること、そして、370/XA の 31 ビット・アドレッシングの利点を活用するように設計されたプログラム用に追加機能を提供することにあります。リンケージ命令は、記憶機構内に新旧両方のプログラムが共存し、互いに通信することを可能にしました。そして現在は、リンケージ命令は、z/Architecture の 64 ビット・アドレッシングを使用できるように機能が拡張されています。

システム/370 のプログラムについては、修正されていない古いプログラムは、24 ビット・アドレッシング・モードで稼働し、同じく 24 ビット・アドレッシング・モードで稼働する他のプログラムのみを呼び出す (または直接通信する) ことが前提とされます。修正済みのプログラムは、通常は 24 ビット・アドレッシング・モードで稼働しますが、24 ビットと 31 ビットのいずれのア

ドレッシング・モードで稼働するプログラムでも呼び出すことができました。現在は、これらのプログラム、および同じく修正された 370-XA、ESA/370、および ESA/390 のプログラムは、24 ビット、31 ビット、または 64 ビットのどのアドレッシング・モードで稼働するプログラムでも呼び出すことができます。新規のプログラムは、どのアドレッシング・モードでも稼働するように作成でき、さらに、場合によっては、どのアドレッシング・モードの中でも呼び出すことが可能なプログラムも作成できます。

BRANCH AND SAVE AND SET MODE (BASSM)

は、アセンブラー/リンケージ・エディター制御セクション (CSECT) の外部にあるサブルーチンを呼び出すために、すべての新しいプログラム、特にリンケージ中にアドレッシング・モードを変更する必要があるプログラムが使用する、主要な呼び出し命令です。呼び出しシーケンスは、通常以下のとおりです。

```
L      15,ACON
BASSM 14,15

      EXTRN SUB
ACON  DC  A(X'80000000'+SUB)
```

ここで、ACON は A タイプ・アドレス定数で、X'80000000' は、31 ビット・アドレッシング・モードで制御を渡すときは指定する必要があり、24 ビット・アドレッシング・モードで制御を渡すときは省略する必要があります。

このようなルーチンからの戻りは、通常以下のとおりです。

BSM 0,14

A タイプ・アドレス定数は、拡張されて 64 ビットのエンタリー・ポイント・アドレスを含む 8 バイト・フィールドになります。このアドレスのビット 63 が 1 のときは、エンタリーが 64 ビット・アドレッシング・モードであることを示します。ここでは、この拡張定数は「ACONE」と示されています。呼び出しシーケンスは、通常以下のようになります。

```
LG      15,ACONE
BASSM  14,15
...
EXTRN  SUB
ACONE  DC  AD(X'1'+SUB)
```

このようなルーチンからの戻りは、通常以下のようになります。

BSM 0,14

アドレッシング・モードを変更する必要がない場合は、BASSM の代わりに BRANCH AND LINK または BRANCH AND SAVE を使用してください。

BRANCH AND LINK (BAL, BALR) 命令は、主としてシステム/370 との互換性を維持するために提供されています。この命令は、古い単純なプログラムが、最小限の変更で、またはまったく変更なしで、31 ビットおよび 64 ビットのアドレッシング・モードで稼働できる可能性を高めるために、31 ビット・モードと 64 ビット・モードの両方で働くように定義されています。ただし、24 ビット・アドレッシング・モードでは、BRANCH AND LINK は、汎用レジスタのビット位置 32-39 に 0 以外の情報を入れるため、何らかの問題が起きたりパフォーマンスが低下することがあります。したがって、この命令の代わりに、BRANCH AND SAVE (BAS および BASR) を使用することをお勧めします。BRANCH AND SAVE の代わりに、BRANCH RELATIVE AND SAVE および BRANCH RELATIVE AND SAVE LONG も使用できます。

24 ビットまたは 31 ビット・アドレッシング・モードで、BRANCH AND LINK (BAL または BALR) により呼び出されたサブルーチンからの通常の戻りは、以下のようにになります。

BCR 15,14

ただし、標準の「戻り命令」は以下のとおりです。

BSM 0,14

この命令は、24 ビット・アドレッシング・モードで実行されている呼び出し側の BAL の場合を除き、すべて

の場合に正しく働きます。24 ビット・アドレッシング・モードでは、BAL を実行すると、リンク・レジスタのビット位置 32 と 33 に ILC の値として 10 が入れられます。したがって、BSM では 31 ビット・アドレッシング・モードに戻ります。また、24 ビット・アドレッシング・モードで BALR を実行したときも、同じ ILC 効果が得られます。

BRANCH AND SAVE (BAS, BASR) 命令は、同じ CSECT 内にあるか、または同じアドレッシング・モードであることが分かっている、任意のプログラムへのサブルーチン・リンケージのために使用できます。また、R2 フィールドが 0 の BASR を使用すると、24 ビット・アドレッシング・モードでも左端のフィールドを整えて 31 ビット・アドレスが得られるので、命令ストリームへのアドレス可能性を高める効果が得られます。

24 ビットまたは 31 ビット・アドレッシング・モードで BRANCH AND SAVE (BAS または BASR) により呼び出されたルーチンからの戻りは、次のいずれかのようになります。

BCR 15,14

または:

BSM 0,14

64 ビット・アドレッシング・モードで BAS または BASR により呼び出されたルーチンから戻るための命令は、BCR でなければなりません。BSM では、リンク・レジスタのビット 63 (命令アドレスの最右端のビット) が 0 のため、リンク・レジスタのビット 32 (アドレス・ビット) に応じて、24 ビットまたは 31 ビットのアドレッシング・モードが設定されます。呼び出し命令として BASSM が使用されている場合は、常に BSM を戻り命令として使用できます。

場合によっては、まだ書き直されていない古いプログラムによって呼び出されるプログラムを、書き直すほうがよいことがあります。このような場合は、24 ビットまたは 31 ビット・アドレッシング・モードで稼働する古いプログラムに、中間プログラムのアドレスが割り当てられます。そして、その中間プログラムは、正しいエンタリー・モードと戻りモードをセットアップしてから、書き直されたプログラムを呼び出します。このような中間プログラムは、グルー・モジュールと呼ばれることがあります。R1 フィールドが 0 ではない BRANCH AND SET MODE (BSM) 命令は、この操作を効率的に行うために必要な機能を提供します。5-19 ページの図 5-4 は、24 ビット・モード・プログラムから 31 ビット・モード・プログラムへのリンケージの場合の、この操作を示しています。

古いプログラム	グルー・モジュール	新しいプログラム
L 15,OLDACON		
BALR 14,15		
•		
•		
•		
OLDACON DC EXTRN GLUE A(GLUE)	GLUE CSECT USING *,15 L 15,NEWACON BSM 14,15 EXTRN NEW NEWACON DC A(NEW)	
		NEW CSECT USING *,15 • • • BSM 0,14

図 5-4. 24 ビット・モードから 31 ビット・モードへのリンケージのためのグルー・モジュール

このグルー・モジュールの中の「BSM 14,15」が実行されると、64 ビット・アドレッシング・モードを示す値が汎用レジスター 14 のビット位置 63 に保管され、24 ビットまたは 31 ビットのいずれかのアドレッシング・モードを示す値がビット位置 32 に保管され、レジスターのその他のビットは変更されません。したがって、新しいプログラムの中で「BSM 0,14」が実行されると、制御権は、再びグルー・モジュールに渡されることなく、古いプログラムに直接渡されます。グルー・モジュールでは、4 バイトの A タイプ・アドレス定数の代わりに、ビット 63 を 1 に設定した 8 バイトの A タイプ・アドレス定数をロードすることによって、64 ビット・アドレッシング・モードの (おそらくは 2G バイトの境界より上位にある) プログラムに、制御を渡すこともできます。

割り込み

CPU は、割り込みを使用して、システムの外部、サブチャンネルまたは入出力 (I/O) 装置の中、他の CPU の中、またはこの CPU 自体の中に存在する条件に応じて、状態を変化させることができます。詳細については、第6章、『割り込み』を参照してください。

割り込みには、外部割り込み、入出力割り込み、マシン・チェック割り込み、プログラム割り込み、再始動割り込み、監視プログラム呼び出し割り込みの 6 つのクラスがあります。各クラスには、それぞれ新旧 2 つの

PSW が関連付けられており、いずれも永続的に割り当てられた実記憶位置に保管されています。どのクラスの割り込みの場合も、割り込みの原因を示す情報が格納され、現 PSW が旧 PSW の場所に格納され、そして、新 PSW の場所にある PSW が取り出されて、それが現 PSW になります。

旧 PSW には、割り込まれたプログラムを再開するために必要な CPU 状況情報が含まれています。割り込みにより呼び出されたプログラムの完了時に、LOAD PSW EXTENDED 命令を使用して、現 PSW を旧 PSW の値に戻すことができます。

命令終了のタイプ

命令実行の終了には、完了、無効化、抑止、中止、および部分完了の 5 つのタイプがあります。

命令実行の部分完了が生じるのは、割り込み可能命令の場合だけです。これについては、5-20ページの『割り込み可能命令』で説明します。

完了

命令の実行が完了した場合は、その命令の定義の中で要求されているとおりの結果が得られます。ある命令の実行の完了後に割り込みが起きたときは、旧 PSW の中の命令アドレスは、次の順番に当たる命令を指しています。

抑止

命令の実行が抑止されたときは、その命令は、「ノーオペレーション」を指定している場合と同じように実行されます。条件コードも含めて、結果フィールドはどれも変更されません。抑止の後で割り込みが起きた場合の旧 PSW 中の命令アドレスは、次の順番に当たる命令を指しています。

無効化

命令実行の無効化では、抑止の場合と同じ結果が生じます。ただし、ある命令の実行が無効化された後で割り込みが起きたときは、旧 PSW 中の命令アドレスは、次の順番に当たる命令ではなく、実行が無効化された命令(または、該当する場合は EXECUTE 命令)を指しています。

中止

命令実行が中止された場合は、その命令により変更されることになっていたフィールドの内容は、予測不能になります。操作の結果、指定された結果フィールドの内容の全部または一部が置き換えられることもあり、まったく置き換えられないこともあります。また、条件コードの変更を要求するような命令の場合は、条件コードが変更される可能性があります。マシン・チェック以外の原因による割り込みの場合は、PSW 内の命令アドレスの妥当性、割り込みコード、および ILC には影響はなく、また、マシンの状態または動作にも影響しません。中止の後で割り込みが起きた場合の旧 PSW 中の命令アドレスは、次の順番に当たる命令を指しています。

プログラミング上の注意: 抑止または無効化が起きた後の命令の実行はノーオペレーションとして扱われますが、一部の命令に関連付けられている暗黙的なトレース・アクションの結果、格納が行われることがあります。4-10ページの『トレース』を参照してください。

割り込み可能命令

割り込み点

ほとんどの命令では、1つの命令全体が1つの操作として実行されます。割り込みが許されるのは、操作と操作の間です。つまり、割り込みが発生する可能性があるのは、1つの操作が完了してから、その次の操作が開始されるまでの間です。

以下に示す命令は、命令の部分的な完了後に割り込みが許されるもので、これらを割り込み可能命令と呼びます。

- COMPARE AND FORM CODEWORD
- COMPARE LOGICAL LONG

- COMPARE UNTIL SUBSTRING EQUAL
- COMPRESSION CALL
- MOVE LONG
- MOVE LONG
- TEST BLOCK
- UPDATE TREE

操作単位

割り込み可能命令に実行中に生じる割り込みも含めて、本書で割り込み点について述べるときは、「操作単位」という用語を使用します。割り込み不能命令の場合は、命令の実行全体が、実際には1つの操作単位の実行ということになります。

割り込み可能命令の実行は、幾つかの操作単位の実行から成ると考えることができ、操作単位と操作単位の間では割り込みが許されます。1つの操作単位で処理されるデータの量は、命令によって異なるほか、モデルや、命令実行への割り込みの原因となった条件によっても異なります。

命令実行が幾つかの操作単位から成っているときに、幾つかの(すべてではなく)操作単位が完了した後で割り込みが起きた場合、その命令の状態を部分完了と呼びます。この場合は、操作単位には、いずれかの終了タイプ(完了、無効化、または抑止)が関連付けられています。中止の場合は、該当の操作単位だけではなく、命令全体が中止されます。

何らかの例外が原因で、割り込み可能命令の最初の操作単位が完了しないことがあります。その場合に、終了の形態が無効化または抑止であるときは、すべてのオペランド・パラメーターおよび結果の位置は変更されません。ただし、命令が条件コードを設定するように定義されている場合、条件コードは予測不能です。

記憶域変更 PER イベントが認識された場合は、割り込みによりそのイベントが示される前に、4K 未満の追加バイトが格納されます。

割り込み可能命令の実行

割り込み可能命令の実行は、その命令に関連するすべての操作単位が完了した時点で、完了します。ある操作単位の完了、無効化、または抑止の後で割り込みが起きたときには、その操作単位に先行する操作単位はすべて完了しており、それより後の操作単位および命令はまだ開始されていません。これらの終了タイプの間的主な違いは、現行操作単位がどのように取り扱われるか、および、旧 PSW に格納されている命令アドレスが、現行の命令を指すか次の順番に当たる命令を指すかにあります。

割り込みの時点では、割り込まれた命令より後の命令が記憶位置またはレジスタの内容に対して行うことになっている変更は、まだ行われていません。

完了: 割り込み可能命令の最後の操作単位が完了したときには、旧 PSW 中の命令アドレスは、次の順番に当たる命令を指しています。現操作単位の結果の位置は、更新されています。オペランド・パラメーターがどのように調整されるかは、命令によって異なります。最後の操作単位以外の操作単位が完了したときには、旧 PSW 中の命令アドレスは、割り込まれた命令または EXECUTE 命令のいずれか該当する方を指しています。現操作単位の結果の位置は、更新されています。オペランド・パラメーターは、割り込み次に格納された旧 PSW が現 PSW にされたときに、割り込まれた命令の実行が割り込み点から再開されるように、調整されます。

無効化: 操作単位が無効化されたときは、旧 PSW 中の命令アドレスは、割り込まれた命令または EXECUTE 命令のいずれか該当する方を指しています。現操作単位の結果の位置は、変更されていません。オペランド・パラメーターは、命令が再実行されるたびに、割り込まれた命令の実行が現操作単位から再開されるように、調整されます。

抑止: 操作単位が抑止されたときは、旧 PSW 中の命令アドレスは、次の順番に当たる命令を指しています。ただし、オペランド・パラメーターは、命令実行がどの程度まで完了しているかを示すように調整されません。抑止の原因となった条件が解消された後で命令が再実行される場合は、現操作単位から実行が再開されません。

中止: 中止の原因となった例外が割り込み可能命令の操作単位の中で起きた場合は、その操作全体が中止され、通常、その命令により変更されることになっていたフィールドの内容は、予測不能です。このような割り込みでは、旧 PSW 中の割り込みアドレスは、次の順番に当たる命令を指しています。

図5-5 は、4 つの終了タイプの間の違いを要約して示しています。

操作単位の状態	命令アドレス	オペランド・パラメーター	現在の結果の記憶位置
完了 最後の操作単位 その他の操作単位	次の命令	命令によって異なる	変更される
	現在の命令	次の操作単位	変更される
無効化	現在の命令	現在の操作単位	変更されない
抑止	次の命令	現在の操作単位	変更されない
中止	次の命令	予測不能	予測不能

図 5-5. 操作単位の終了タイプ

命令が条件コードを設定するものとして定義されている場合は、最後の操作単位が完了していない限り、命令実行の結果の条件コードは予測不能です。

割り込み可能命令の代わりにの条件コード

以下に示す命令は割り込み可能命令ではありませんが、代わりに、命令のパラメーターにより指定された処理の中の CPU が決定する部分が実行された後で、完了できる命令です。

- CHECKSUM
- COMPARE LOGICAL LONG EXTENDED
- COMPARE LOGICAL LONG UNICODE
- COMPARE LOGICAL STRING
- CONVERT UNICODE TO UTF-8
- CONVERT UTF-8 TO UNICODE
- MOVE LONG EXTENDED
- MOVE LONG UNICODE
- MOVE STRING
- SEARCH STRING
- TRANSLATE EXTENDED
- TRANSLATE ONE TO ONE
- TRANSLATE ONE TO TWO
- TRANSLATE TWO TO ONE
- TRANSLATE TWO TO TWO

指定されたすべての処理のうち CPU が決定した量の処理だけが行われた後で、上記の命令のいずれかが完了したときは、その命令で条件コード 3 が設定されます。この種の完了の場合は、PSW 中の命令アドレスは次の順番に当たる命令を指します。命令のオペランド・パラメーターが調整され、ブランチによりその命令に戻って再実行することによって命令が再開されます。命令が指定されたすべての処理を完了している場合は、3 以外の条件コードが設定されます。

上記のいずれかの命令で条件コード 3 が設定されるポイントは、割り込み可能命令における割り込み点に類似し

ており、また、隣接ポイント間で行われる処理の量は、割り込み可能命令の中の 1 つの操作単位に類似していません。ただし、命令は割り込み可能ではないので、1 回の実行が 1 操作単位の実行と見なされます。

条件コード 3 の設定を伴って完了した場合は、割り込みが可能になります。割り込みが必要ない状況では、モデルおよび命令によっては、条件コード 3 は設定される場合とされない場合があります。

記憶域変更 PER イベントが認識された場合は、割り込みによりそのイベントが示される前に、4K 未満の追加バイトが格納されます。

COMPARE UNTIL SUBSTRING EQUAL および COMPRESSION CALL 命令はどちらも割り込み可能命令であり、同時に、CPU が決定した量の処理を終えた後で条件コード 3 を設定する命令でもあります。

プログラミング上の注意:

1. 監視プログラム割り込みと一部のプログラム割り込みを除けば、どの割り込みでも、割り込み可能命令の一部が実行された後で起きる可能性があります。特に、外部割り込み、入出力割り込み、マシン・チェック割り込み、再始動割り込み、そしてアクセス例外および PER イベントに対するプログラム割り込みは、操作単位と操作単位の間で生じることがあります。
2. 割り込み可能命令の 1 操作単位内で処理されるデータの量は、モデルによって異なるほか、命令実行の割り込みまたは停止の原因となった条件のタイプによっても異なります。したがって、現操作単位の終了時点で割り込みが発生したときは、その操作単位の長さは割り込みのタイプによって異なる場合があります。さらに、割り込み可能命令の実行中に停止機能が要求された場合は、CPU は、現操作単位の実行が完了した時点で停止状態になります。また、命令ステップ・モードでは、操作単位が 1 つだけ実行されますが、その操作単位は、停止のさまざまなケースに応じて異なるものになることがあります。

無効化と抑止に対する例外

ある種の異常な状況下では、通常なら無効化または抑止を引き起こすような例外が発生しているにもかかわらず、格納型オペランドを持つ命令の結果フィールドが変更されることがあります。このような状況は、一般的な規則に反する例外的な状況です。通常は、無効化または抑止が必要な例外が認識されたときは、操作はノーオペレーションとして扱われます。このような状況が生じる

と、記憶域内の最終結果は無変更のままであるように見えても、格納型オペランドに関連した変更ビットがオンにされることがあります。また、個々の状況に応じて、さらに別の影響が見られることもあります。このような影響の範囲については、個々の状況ごとに説明します。

これらのどの状況の場合も、格納アクセスは発生せず、格納アクセスが禁止されているときは変更ビットも設定されません。CPU では、格納アクセスについてアクセス例外が存在する場合、または、そのアクセスに関するアクセス例外より優先順位の高い例外が存在する場合は、その格納アクセスは禁止されます。

これらの状況下で、抑止を必要とする例外のための割り込みが生じた場合は、旧 PSW の中の命令アドレスは次の順番に当たる命令を指しています。無効化を必要とする例外のための割り込みが起きたときは、部分的な結果が格納されていた場合でも、旧 PSW の中の割り込みアドレスはその例外の原因となった命令を指しています。

DAT 関連アクセス例外の場合の記憶域の変更と復元

このセクションでは、動的アドレス変換プロセスの中で生じる可能性がある例外を、「DAT 関連アクセス例外」と呼ぶことにします。この種の例外には、ASCE タイプ例外、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、ページ変換例外、変換指定例外、および、アドレッシング例外 (DAT テーブル・エントリーが構成内の使用可能でない位置に指定されているときに生じる) があります。上記の例外のうち最初 6 つは、通常は無効化を引き起こし、最後の 2 つは、通常は抑止を引き起こします。記憶保護例外 (ページ保護による例外も含む) は、DAT 関連アクセス例外とは見なされません。

一部のモデルでは、DAT 関連アクセス例外が起きたときに、チャンネル・プログラムが、以下に述べるような記憶機構への影響を認識することがあります。

格納型オペランドを持つ命令において、命令のいずれかのオペランドについて DAT 関連アクセス例外が認識されたときは、その格納型オペランドのうちの例外を引き起こさない部分が中間値に変更されることがあります。これは、後で元の値に復元されます。

DAT 関連アクセス例外が起きたときの記憶域変更および復元に関連したアクセスは、チャンネル・プログラムが認識するだけで、マルチプロセッシング構成内の他の CPU は認識できません。複数アクセス・オペランドを持つものとして定義されている命令を除き、中間値 (存在する場合) は、常に、DAT 関連アクセス例外が起きなければ得られたはずの最終値と同じになります。

プログラミング上の注意:

1. DAT 関連アクセス例外の場合の記憶域変更および復元が起きる主な状況には、以下の 2 つがあります。
 - a. ページ境界にまたがる格納型オペランドの部分について例外が認識され、その他の部分についてはアクセス例外がない場合。
 - b. 2 つの記憶域オペランドを持つ例外 (例えば、SS フォーマットの命令または MOVE LONG) の一方のオペランドについて例外が認識され、もう一方のオペランド (格納型オペランド) にはアクセス例外がない場合。
2. チャンネル・プログラムが、DAT 関連アクセス例外に伴う記憶域変更および復元 (特に CCW チェーンが変更されたとき) が原因で生じる中間オペランドを認識しないようにするには、CPU プログラムで次のいずれかを行う必要があります。
 - a. 一時点では 1 つの記憶域ページ上でのみ操作を行う。
 - b. 必要とするどのページにも例外が生じないことを確認するために、予備テストを行う。
 - c. DAT をオフにして操作を行う。

DAT テーブル・エントリーの変更

有効であり連結されている DAT テーブル・エントリーが、例外の原因となるような値に変更された後、そのエントリーと置き換わる適格性を備えたエントリーが TLB から消去される前に、変換にそのエントリーを必要とする仮想アドレスを使用して記憶域を参照しようとした場合は、命令により変更されることになっていたフィールドの内容は、すべて予測不能です。DAT テーブル・エントリーが変更された仮想アドレスに結果がある場合、その結果は、本来そのアドレスに関連付けられていた実記憶位置に入れられることがあります。さらに、最初は該当していなかったアクセス例外については、割り込みが起きるかどうかは予測不能です。一部のマシンでは、この状況が命令処理損傷マシン・チェックによって報告され、同時に遅延アクセス例外ビットも示されることがあります。

編集命令および変換命令の試行実行

EDIT、EDIT AND MARK、および TRANSLATE 命令の場合は、命令の実行が開始される前に、オペランドのアクセス可能性を調べる試行実行の中で、実際に操作に使用されるオペランドの部分が確定されることがあります。この試行実行では、結果が格納されない形で命令が実行されます。最初の試行実行の後、実行が完了する

までの間に、TRANSLATE の第 1 オペランド、または EDIT または EDIT AND MARK のいずれか一方のオペランドが、他の CPU またはチャンネル・プログラムにより変更された場合は、その命令により変更されることになっていたフィールドの内容は、いずれも予測不能です。さらに、最初は該当していなかったアクセス例外については、割り込みが起きるかどうかは予測不能です。

許可メカニズム

このセクションで説明する許可メカニズムにより、制御プログラムは、特定の準特権プログラムに提供する機能の程度を設定することができます。(許可メカニズムの概要は、5-26ページの図5-6に示してあります。) 許可メカニズムは、準特権を持つと見なされるプログラムが使用するためのものです。準特権プログラムは、問題プログラム状態で実行されますが、さらに多くの機能を使用する許可が与えられていることがあります。この許可制御を用いてプログラムの階層を確立することにより、高いレベルのプログラムに、低いレベルのプログラムより大きい特権または権限を持たせることができます。各レベルで使用できる機能の範囲、および低いレベルから高いレベルに制御を転送する能力は、制御プログラムが管理するテーブルの中に指定します。リンケージ・スタックを使用する場合は、非階層的な制御転送も指定できます。

準特権命令とは、問題プログラム状態で実行でき、このセクションで説明する 1 つ以上の許可メカニズムにより制御される命令です。準特権命令は 28 個あり、そのほかに、特権命令 LOAD ADDRESS SPACE PARAMETERS も許可メカニズムにより制御されます。これらすべての準特権命令と特権命令について、第10章、『制御命令』の中で説明されています。

許可メカニズムにより制御される命令は、5-26ページの図5-6にリストされています。この図には、特に準特権命令を制御するためのものではないその他の許可メカニズムも示されています。これらの許可メカニズムは、暗黙のアクセス・レジスター変換 (記憶域参照を行う命令の一環として行われるアクセス・レジスター変換) を制御するほか、LOAD REAL ADDRESS、STORE REAL ADDRESS、TEST ACCESS、および TEST PROTECTION 命令におけるアクセス・レジスター変換も制御します。これら追加のメカニズム (拡張許可指標、ALE シーケンス番号、および ASTE シーケンス番号) については、5-34ページの『アクセス・レジスター指定のアドレス・スペース』で説明します。

モードの要件

ほとんどの準特権命令は、DAT がオンになっていない限り実行できません。基本 PROGRAM CALL および PROGRAM TRANSFER は、1 次スペース・モードでのみ使用できます。(基本 PROGRAM CALL は、リンケージ・スタック使用しないときの PROGRAM CALL 操作です。リンケージ・スタックを使用しているときの PROGRAM CALL 操作は、スタッキング PROGRAM CALL と呼ばれます)。MOVE TO PRIMARY および MOVE TO SECONDARY は、1 次スペース・モードおよび 2 次スペース・モードでのみ使用できます。BRANCH AND STACK、スタッキング PROGRAM CALL、および PROGRAM RETURN は、1 次スペース・モードおよびアクセス・レジスター・モードでのみ使用できます。EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE は、1 次スペース・モード、アクセス・レジスター・モード、およびホーム・スペース・モードでのみ使用できます。無効な変換モードで準特権命令が実行されると、特殊操作例外が認識されます。

PROGRAM TRANSFER は、PSW 内の問題プログラム状態ビットの新しい値を指定します。問題プログラム状態にあるプログラムが PROGRAM TRANSFER を実行し、監視プログラム状態を設定しようとする、特権命令例外が認識されます。また、問題プログラム状態で、RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST を使用してホーム・スペース・モードを設定しようとした場合も、特権命令例外が認識されます。

抽出権限制御

抽出権限制御ビットは、制御レジスター 0 のビット位置 36 にあります。問題プログラム状態で、以下の命令が完了できるようにするには、ビット 36 は 1 でなければなりません。

- EXTRACT PRIMARY ASN
- EXTRACT SECONDARY ASN
- INSERT ADDRESS SPACE CONTROL
- INSERT PSW KEY
- INSERT VIRTUAL STORAGE KEY

ビット 36 が 1 でないときは、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御は検査されません。

PSW キー・マスク

PSW キー・マスクは、制御レジスター 3 のビット 32-47 から成っており、これらのビットはそれぞれ PSW キーの値 0-15 に対応しています。これらのビットは、問題プログラム状態で、どのキーとエントリー・ポイントがプログラムに許可されているかを制御するために使用されます。PSW キー・マスクは、PROGRAM TRANSFER により変更され、BRANCH AND SET AUTHORITY および PROGRAM CALL により変更またはロードされ、LOAD ADDRESS SPACE PARAMETERS および PROGRAM RETURN によりロードされます。PSW キー・マスクは、問題プログラム状態で以下のものを制御するために使用されます。

- SET PSW KEY FROM ADDRESS 命令を使用して設定できる PSW キーの値。
- 第 2 アクセス・キーを指定する 6 つの移動命令 (MOVE PAGE、MOVE TO PRIMARY、MOVE TO SECONDARY、MOVE WITH KEY、MOVE WITH SOURCE KEY、および MOVE WITH DESTINATION KEY) にとって有効な PSW キーの値。
- PROGRAM CALL を使用して呼び出すことができるエントリー・ポイント。この場合は、PSW キー・マスクとエントリー・テーブル・エントリー内の許可キー・マスクとの論理積が求められ、その結果が 0 の場合は、プログラムは許可されません。

問題プログラム状態にある命令が、PSW キー・マスクにより許可されていないキーを使用しようとする、特権命令例外が認識されます。また、問題プログラム状態にある命令が、PSW キー・マスクにより許可されていないエントリーを呼び出そうとした場合も、同じ例外が認識されます。監視プログラム状態では、PSW キー・マスクは検査されず、すべてのキーおよびエントリー・ポイントが有効です。

2 次スペース制御

制御レジスター 0 のビット 37 は、2 次スペース制御ビットです。このビットは、制御プログラムが、2 次の領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、またはセグメント・テーブルが設定されているかどうかを示すために使用するメカニズムです。SET ADDRESS SPACE CONTROL FAST を完了させるためには、ビット 37 は 1 でなければならないことがあります。また、以下の命令を完了させるには、ビット 37 は 1 でなければなりません。

- MOVE TO PRIMARY
- MOVE TO SECONDARY
- SET ADDRESS SPACE CONTROL

1 でない場合は、特殊操作例外が認識されます。2 次スペース制御は、問題プログラム状態と監視プログラム状態のどちらの場合も検査されます。

サブシステム・リンケージ制御

1 次 ASN 第 2 テーブル・エントリーのビット 192 は、サブシステム・リンケージ制御ビットです。以下の命令が完了できるようにするには、サブシステム・リンケージ制御は 1 でなければなりません。

- PROGRAM CALL
- PROGRAM TRANSFER

1 でない場合は、特殊操作例外が認識されます。サブシステム・リンケージ制御は、問題プログラム状態でも監視プログラム状態でも検査され、命令のスペース切り替えバージョンと現 1 次バージョンの両方を制御します。

ASN 変換制御

制御レジスター 14 のビット 44 は、ASN 変換制御ビットです。このビットのメカニズムを利用して、制御プログラムは、特定のプログラムの実行中に ASN 変換を実行できるかどうかを指示することができます。以下の命令を完了するには、ビット 44 は 1 でなければなりません。

- LOAD ADDRESS SPACE PARAMETERS
- SET SECONDARY ASN
- スペース切り替えを伴う PROGRAM CALL
- スペース切り替えを伴う場合、および復元された 2 次 ASN が復元された 1 次 ASN に等しくない場合の PROGRAM RETURN
- スペース切り替えを伴う PROGRAM TRANSFER

1 でない場合は、特殊操作例外が認識されます。ASN 変換制御は、問題プログラム状態と監視プログラム状態のどちらの場合も検査されます。PROGRAM CALL では、ASN 変換を行わずに、エントリー・テーブル・エントリーから ASN 第 2 テーブル・エントリーのアドレスを直接入手する場合でも、ASN 変換制御が検査されます。

許可指標

許可指標は、制御レジスター 4 のビット位置 32-47 に含まれています。許可指標は 1 次アドレス・スペースに関連付けられるものであり、スペース切り替えを伴う PROGRAM CALL、スペース切り替えを伴う PROGRAM RETURN、スペース切り替えを伴う PROGRAM TRANSFER、または LOAD ADDRESS SPACE PARAMETERS が実行されるときに、PASN と一緒にロードされます。許可指標は、プログラムが特定のアドレス・スペースを設定する権限を持っているか

どうかを判別するために、使用されます。プログラムは、アドレス・スペースを 2 次アドレス・スペースとして設定することを許可されている場合と、1 次アドレス・スペースとして設定することを許可されている場合と、その両方があります。許可指標は、問題プログラム状態と監視プログラム状態のどちらの場合も検査されます。

各アドレス・スペースには、権限テーブルが 1 つずつ関連付けられています。この権限テーブルからエントリーを選択するために、許可指標が使用されます。各エントリーに含まれている 2 つのビットは、その許可指標を持つプログラムが、アドレス・スペースを 1 次アドレス・スペースとして設定できるか、2 次アドレス・スペースとして設定できるか、それともその両方ができるのかを示します。

スペース切り替えを伴う SET SECONDARY ASN 命令、および復元された 2 次 ASN が復元された 1 次 ASN に等しくない場合の PROGRAM RETURN 命令は、許可指標を使用して権限テーブル・エントリーの中の 2 次権限ビットをテストし、アドレス・スペースを 2 次アドレス・スペースとして設定できるかどうかを判定します。このテスト対象ビットは 1 でなければならず、そうでない場合は 2 次権限例外が認識されます。

スペース切り替えを伴う PROGRAM TRANSFER 命令は、許可指標を使用して権限テーブル・エントリー内の 1 次権限ビットをテストし、アドレス・スペースを 1 次アドレス・スペースとして設定できるかどうかを判定します。このテスト対象ビットは 1 でなければならず、そうでない場合は 1 次権限例外が認識されます。

スペース切り替えを伴う PROGRAM CALL 命令は、ASN 第 2 テーブル・エントリーから新しい許可指標をロードさせます。これにより、呼び出し先プログラムに、呼び出し側プログラムより多くのアドレス・スペースまたは異なるアドレス・スペースにアクセスすることを許可する許可指標を与えることができます。スペース切り替えを伴う PROGRAM RETURN 命令およびスペース切り替えを伴う PROGRAM TRANSFER 命令は、戻り先アドレス・スペースに関連した許可指標を復元します。

権限テーブル・エントリー内の 2 次権限ビットを拡張許可指標と一緒に使用することにより、プログラムが、アクセス・レジスター変換の際にアクセス・リスト・エントリーの使用を許可されているかどうか、判別されることもあります。これについては、5-34 ページの『アクセス・レジスター指定のアドレス・スペース』で説明します。

機能または命令	モードの要件		許可メカニズム									スペースリクエスト制御 (1.57, 13.57)
	特権命令	変換モード	サブシステムリンク制御 ⁶	2次スペース制御 (0.37)	ASN変換制御 (14.44)	抽出権限制御 (0.36)	PSWキー・マスク (3.32-3.47)	許可指標 (4.32-4.47)	拡張許可指標 (8.32-8.47)	ALEシーケンス番号 ⁷	ASTEシーケンス番号 ⁸	
暗黙の AR 変換 BAKR BSA-ba BSA-ra BSG EPAR		A SO-PA SO-PSAH SO-PSAH					Q		EA	ALQ	ASQ ASQ	
EREG EREGG ESAR ESTA IAC IPK		SO-PAH SO-PAH SO-PSAH SO-PAH SO-PSAH				Q Q Q						
IVSK LASP LRA LRAG MSTA MVCDK	P P P	SO-PSAH SO-PAH			SO	Q		CC	CCA CCA	CCA CCA	CCA CCA	CC
MVCP MVCS MVCSK bPC-cp sPC-cp bPC-ss		SO-PS SO-PS SO-P SO-PA SO-P		SO SO			Q Q Q Q ¹ Q ¹ Q ¹					X1
sPC-ss PR-cp PR-ss PT-cp PT-ss RP	Q ² Q ²	SO-PA SO-PA SO-PA SO-P SO-P	SO SO SO		SO SO ⁴ SO SO		Q ¹	SA ⁵ PASA ⁵ PA				X1 X1 X1 X2
SAC SACF SPKA SSAR-cp SSAR-ss STRAG	Q ³ Q ³ P	SO-PSAH SO-PSAH SO-PSAH SO-PSAH		SO SO ⁹			Q	SA	EA	ALQ	ASQ	X2 X2
TAR TPROT	P							CC CC	CC CC	CC CC		

図 5-6. 許可メカニズムの要約

許可メカニズムの要約の説明:

1	PSW キー・マスクと、エントリー・テーブル・エントリー内の許可キー・マスクとの論理積が求められます。	CRx.y	制御レジスター x のビット位置 y。
2	問題プログラム状態のときに監視プログラム状態を設定しようとする、例外が認識されます。	EA	拡張権限例外。
3	問題プログラム状態のときにホーム・スペース状態を設定しようとする、例外が認識されます。	P	特権命令に関する特権命令例外。
4	新しい SASN について ASN 変換が行われます。例外が認識されることがあるのは、新しい SASN が新しい PASN に等しくない場合のみです。	PA	1 次権限例外。
5	新しい SASN について 2 次権限が検査されます。例外が認識されることがあるのは、新しい SASN が新しい PASN に等しくない場合のみです。	PASA	1 次権限例外または 2 次権限例外。
6	サブシステム・リンケージ制御は、1 次 ASN 第 2 テーブル・エントリーのビット 192 です。	Q	準特権命令に関する特権命令例外。権限は、問題プログラム状態の場合のみ検査されます。
7	ALE シーケンス番号は、アクセス・リスト・エントリー・トークンのビット 8-15、およびアクセス・リスト・エントリーのビット 8-15 です。	SA	2 次権限例外。
8	ASTE シーケンス番号は、アクセス・リスト・エントリーのビット 96-127、および ASN 第 2 テーブル・エントリーの 160-191 です。	SO	特殊操作例外。
9	例外が認識されるかどうかは予測不能です。	SO-P	CPU は 1 次スペース・モードでなければなりません。CPU が、2 次スペース・モード、アクセス・レジスター・モード、ホーム・スペース・モード、または実モードのときは、特殊操作例外が起きます。
A	アクセス・レジスター変換が行われるのは、アクセス・レジスター・モードの場合のみです。	SO-PA	CPU は 1 次スペース・モードまたはアクセス・レジスター・モードでなければなりません。CPU が、2 次スペース・モード、ホーム・スペース・モード、または実モードのときは、特殊操作例外が起きます。
ALQ	ALE シーケンス例外。	SO-PAH	CPU は 1 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードでなければなりません。CPU が、2 次スペース・モードまたは実モードのときは、特殊操作例外が起きます。
ASQ	ASTE シーケンス例外。	SO-PS	CPU は 1 次スペース・モードまたは 2 次スペース・モードでなければなりません。CPU が、ホーム・スペース・モード、アクセス・レジスター・モード、または実モードのときは、特殊操作例外が起きます。
bPC	基本 (非スタッキング) PROGRAM CALL。	SO-PSAH	CPU は 1 次スペース・モード、2 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードでなければなりません。CPU が実モードのときは、特殊操作例外が起きます。
CC	テストの結果、条件コードが設定されません。	sPC	スタッキング PROGRAM CALL。
CCA	テストの結果、条件コードが設定されません。テストが行われるのは、アクセス・レジスター・モードの場合のみです。	X1	制御レジスター 1 のビット 57 が 1 のときは、スペース切り替えイベントが認識されます。操作は完了します。
		X2	制御レジスター 1 または 13 のビット 57 が 1 で、命令スペースがホーム・アドレス・スペースへ、またはホーム・アドレス・スペースから変更されたときは、スペ

ース切り替えイベントが認識されます。操作は完了します。

PC 番号の変換

PC 番号の変換は、PROGRAM CALL 命令の実行の一環としてエントリー・テーブル・エントリーを見つけるために、20 ビットの PC 番号を変換するプロセスです。この変換を行うために、20 ビットの PC 番号は 2 つのフィールドに分割されます。左端 12 ビットはリンクエッジ指標 (LX) で、右端 8 ビットはエントリー指標 (EX) です。PC 番号の元になる有効アドレスのフォーマットは、以下のとおりです。



変換は、リンクエッジ・テーブルおよびエントリー・テーブルの 2 つのテーブルを使用して行われます。これらのテーブルは両方とも実記憶域に入っています。リンクエッジ・テーブル指定は、1 次 ASN 第 2 テーブル・エントリー (1 次 ASTE) と呼ばれる、記憶域内の 3 番目の区域に入っており、ASTE の起点は、制御レジスター 5 の中にあります。エントリー・テーブルは、リンクエッジ・テーブル・エントリーにより指定されます。

PC 番号変換制御

PC 番号変換は、制御レジスター 5 の内容が指し示す 1 次 ASN 第 2 テーブル・エントリー内のリンクエッジ・テーブル指定により制御されます。

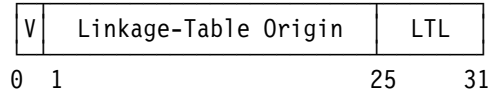
制御レジスター 5

制御レジスター 5 は、1 次 ASN 第 2 テーブル・エントリーの位置を指定します。このレジスターのフォーマットは以下のとおりです。



1 次 ASTE 起点 (PASTE0): 制御レジスター 5 のビット 33-57 の右側に 6 個の 0 を付加したものが、1 次 ASTE の始めを示す 31 ビットの実アドレスを形成します。1 次 ASTE のバイト 24-27 は、リンクエッジ・テーブル指定です。

リンクエッジ・テーブル指定のフォーマットは、以下のとおりです。



サブシステム・リンクエッジ制御 (V): リンケージ・テーブル指定のビット 0 は、サブシステム・リンクエッジ制御ビットです。以下の命令が完了できるようにするには、ビット 0 は 1 でなければなりません。

- PROGRAM CALL
- PROGRAM TRANSFER

1 でない場合は、特殊操作例外が認識されます。サブシステム・リンクエッジ制御は、問題プログラム状態でも監視プログラム状態でも検査され、命令のスペース切り替えバージョンと現 1 次バージョンの両方を制御します。

リンクエッジ・テーブル起点: リンケージ・テーブル指定のビット 1-24 の右側に 7 個の 0 を付加したものが、リンクエッジ・テーブルの始めを示す 31 ビットの実アドレスを形成します。

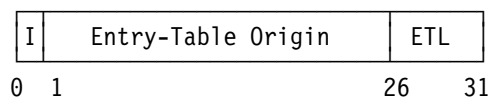
リンクエッジ・テーブル長 (LTL): リンケージ・テーブル指定の 25-31 は、リンクエッジ・テーブル長を 128 バイト単位で指定します。したがって、リンクエッジ・テーブル変数の長さは、32 個の 4 バイト・エントリーの整数倍になります。128 バイト単位の場合のリンクエッジ・テーブル長は、ビット位置 25-31 の値に 1 を加えた値です。リンクエッジ・テーブル長を、PC 番号のリンクエッジ指標部分の左端 7 ビットと比較することにより、リンクエッジ指標がリンクエッジ・テーブル内のエントリーの 1 つを指しているかどうかを判別されます。

PC 番号変換テーブル

PC 番号変換プロセスは、リンクエッジ・テーブルおよびエントリー・テーブルの 2 つのテーブルを使用した 2 段階の索引から成ります。これらのテーブルは実記憶域に入っています。

リンクエッジ・テーブル・エントリー

リンクエッジ・テーブルから取り出されるエントリーのフォーマットは、以下のとおりです。



リンクエッジ・テーブル・エントリー内のフィールドは、以下のように割り振られています。

LX 無効ビット (I): ビット 0 は、リンケージ・テーブル・エントリーに関連したテーブル・エントリーが使用可能かどうかを制御します。

このビットが 0 のときは、リンケージ・テーブル・エントリーを使用して PC 番号変換が進められます。このビットが 1 のときは、LX 変換例外が認識されます。

エントリー・テーブル起点: ビット 1-25 の右側に 6 個の 0 を付加したものが、エントリー・テーブルの始めを示す 31 ビットの実アドレスを形成します。

エントリー・テーブル長 (ETL): ビット 26-31 は、エントリー・テーブルの長さを 128 バイト単位で指定します。したがって、テーブルは可変長で、長さは 4 個の 32 バイト・エントリーの整数倍になります。128 バイト単位でのエントリー・テーブルの長さは、ビット位置 26-31 の値に 1 を加えた値です。エントリー・テーブル長をエントリー指標の左端 6 ビットと比較することにより、エントリー指標がエントリー・テーブル内のエントリーのどれかを指しているかどうかを判別されます。

エントリー・テーブル・エントリー

エントリー・テーブル・エントリーのビット 0-63 のフォーマットは、PROGRAM CALL 操作の後で有効になっているアドレッシング・モードが、拡張アドレッシング・モード (64 ビット) か基本アドレッシング・モード (24 ビットまたは 31 ビット) かにによって異なります。そして、どちらのアドレッシング・モードかは、エントリー・テーブル・エントリーのビット 128 および 129 によって決まります。

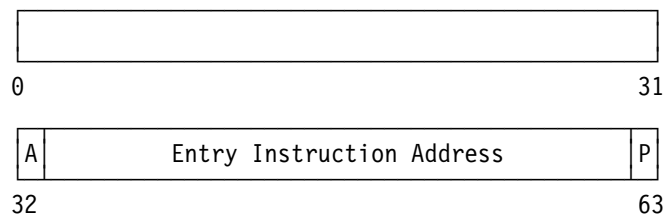
エントリー・テーブル・エントリーのビット 128 (T) は、PC タイプ・ビットです。ビット 128 が 0 のときは、PROGRAM CALL は基本 (非スタッキング) 操作を行います。ビット 128 が 1 のときは、PROGRAM CALL はスタッキング操作を行います。

エントリー・テーブル・エントリーのビット 129 (G) は、エントリー拡張アドレッシング・モード・ビットです。PROGRAM CALL の基本操作では、現 PSW のビット 31 (拡張アドレッシング・モード・ビット) は、ビット 129 に等しくなければなりません。等しくないとき、特殊操作例外が認識されます。ビット 129 が 0 のときのスタッキング・オペレーションでは、現 PSW のビット 31 は 0 に設定され、PSW のビット 32 (基本アドレッシング・モード・ビット) は、エントリー・テーブル・エントリーのビット 32 (A) (エントリー基本アドレッシング・モード・ビット) の値に設定されます。ビ

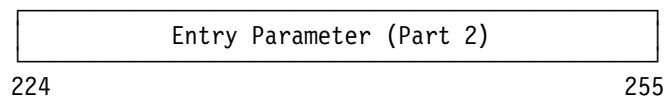
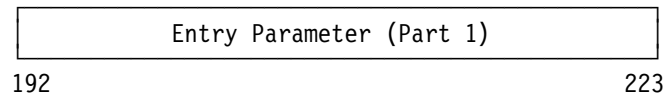
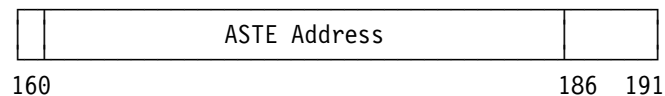
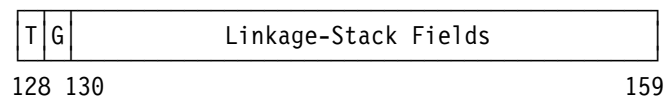
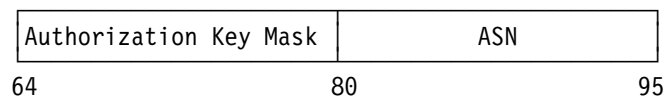
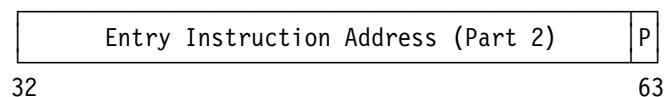
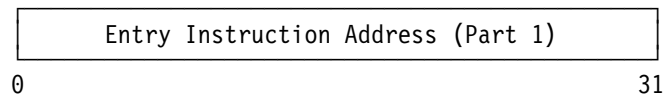
ット 129 が 1 のときのスタッキング操作では、現 PSW のビット 31 および 32 はどちらも 1 に設定されます。したがって、PROGRAM CALL の基本操作では、拡張アドレッシング・モードと基本アドレッシング・モードの間の切り替えは行われませんが、24 ビット・モードと 31 ビット・モードの間の切り替えは可能です。スタッキング操作では、どのアドレッシング・モードでも設定できます。

32 バイトのエントリー・テーブル・エントリーのフォーマットは、以下のとおりです。

ビット 129 が 0 の場合



ビット 129 が 1 の場合



エントリー・テーブル・エントリー内のフィールドは、以下のように割り振られます。

エンタリー基本アドレッシング・モード (A): ビット 129 が 0 のときは、PROGRAM CALL 操作の一環として、ビット 32 は、基本アドレッシング・モード・ビット (現 PSW のビット 32) と置き換わります。この場合、ビット 32 が 0 のときは、ビット 33-39 も 0 でなければなりません。そうでない場合は、PC 変換指定例外が認識されます。ビット 129 が 1 のときは、ビット 32 はエンタリー命令アドレスのビットの 1 つで、PSW のビット 32 は 1 のままか、または 1 に設定されます。

エンタリー命令アドレス: ビット 129 が 0 のときは、ビット 33-62 の左側に 33 個の 0 を付加し、右側に 1 個の 0 を付加したものが、命令アドレスを形成します。このアドレスは、PROGRAM CALL 命令の一環として、PSW 内の命令アドレスと置き換えられます。ビット 129 が 1 のときは、ビット 0-62 の右側に 1 個の 0 を付加したものが、命令アドレスを形成します。

エンタリー問題プログラム状態 (P): ビット 63 は、PROGRAM CALL 操作の一環として、問題プログラム状態ビット (現 PSW のビット 15) と置き換えられます。

許可キー・マスク: ビット 64-79 は、問題プログラム状態のときに、PROGRAM CALL を発行するプログラムが、このエンタリー・ポイントを呼び出すことを許可されるかどうかを検査するために使用されます。許可キー・マスクと制御レジスター 3 の中の現 PSW キー・マスクの論理積が求められ、その結果がすべて 0 かどうかを検査されます。結果がすべて 0 の場合は、特権命令例外が認識されます。監視プログラム状態のときは、この検査は行われません。

ASN: ビット 80-95 は、スペース切り替え操作 (PC-ss) または現 1 次操作 (PC-cp) のどちらが起きるかを示します。ビット 80-95 が 0 のときは、PC-cp が指定されます。ビット 80-95 がすべて 0 ではない場合は、PC-ss が指定され、これらのビットは 1 次 ASN と置き換わる ASN として扱われます。

エンタリー・キー・マスク: ビット 96-111 は、PROGRAM CALL 操作の一環として、論理和演算されて制御レジスター 3 の中の PSW キー・マスクに組み込まれるか、または、その PSW キー・マスクと置き換え

られます。どちらになるかは、ビット位置 130-159 の中のビットの 1 つにより決定されます。

PC タイプ・ビット (T): ビット 128 は、0 であれば基本 PROGRAM CALL 操作を指定し、1 であればスタッキング PROGRAM CALL 操作を指定します。

エンタリー拡張アドレッシング・モード (G): 基本 PROGRAM CALL 操作では、ビット 129 は拡張アドレッシング・モード・ビット (現 PSW のビット 31) に一致していなければなりません。一致していないと、特殊操作例外が認識されます。スタッキング操作では、ビット 129 は PSW のビット 31 と置き換わります。

ASTE アドレス: ビット 80-95 がすべて 0 ではない場合は、ビット 161-185 の右側に 6 個の 0 を付加したものが、31 ビットの ASN 第 2 テーブル・エンタリー・アドレスを形成します。このアドレスは、ビット 80-95 に ASN 変換プロセスを適用した結果として得られるアドレスです。

エンタリー・パラメーター: ビット 129 が 0 のときは、PC 操作の一環として、ビット 224-255 が汎用レジスター 4 のビット位置 32-63 に入れられ、レジスターのビット 0-31 は変更されません。ビット 129 が 1 のときは、PC 操作の一環として、ビット 192-255 が汎用レジスター 4 に入れられます。

130-159 はリンケージ・スタックと関連して使用されます。これについては、5-64 ページの『リンケージ・スタック・エンタリー・テーブル・エンタリー』で説明します。

ビット 112-127、160、および 186-191 は、将来の拡張に備えて予約されており、いずれも 0 でなければなりません。

プログラミング上の注意: エンタリー・パラメーターは、環境によって異なる必要情報を見つける場合にアドレッシング可能性の基盤として使用できるアドレスを、呼び出し先プログラムに与えるためのものです。このパラメーターは、異なるエンタリー・テーブルをセットアップすることにより、個々の環境に合わせて変更することができます。これに代わる方法として、呼び出し側プログラムからこの情報を入手することもできますが、その場合は、広範囲にわたる妥当性検査が必要になったり、保全性の問題が起きることがあります。

PC 番号変換プロセス

PC 番号の変換には、リンケージ・テーブルとエントリー・テーブルが使用されます。これらのテーブルはどちらも実記憶域の中にあります。この変換には、1 次 ASN 第 2 テーブル・エントリーも必要です。このエントリーも実記憶域内にあります。

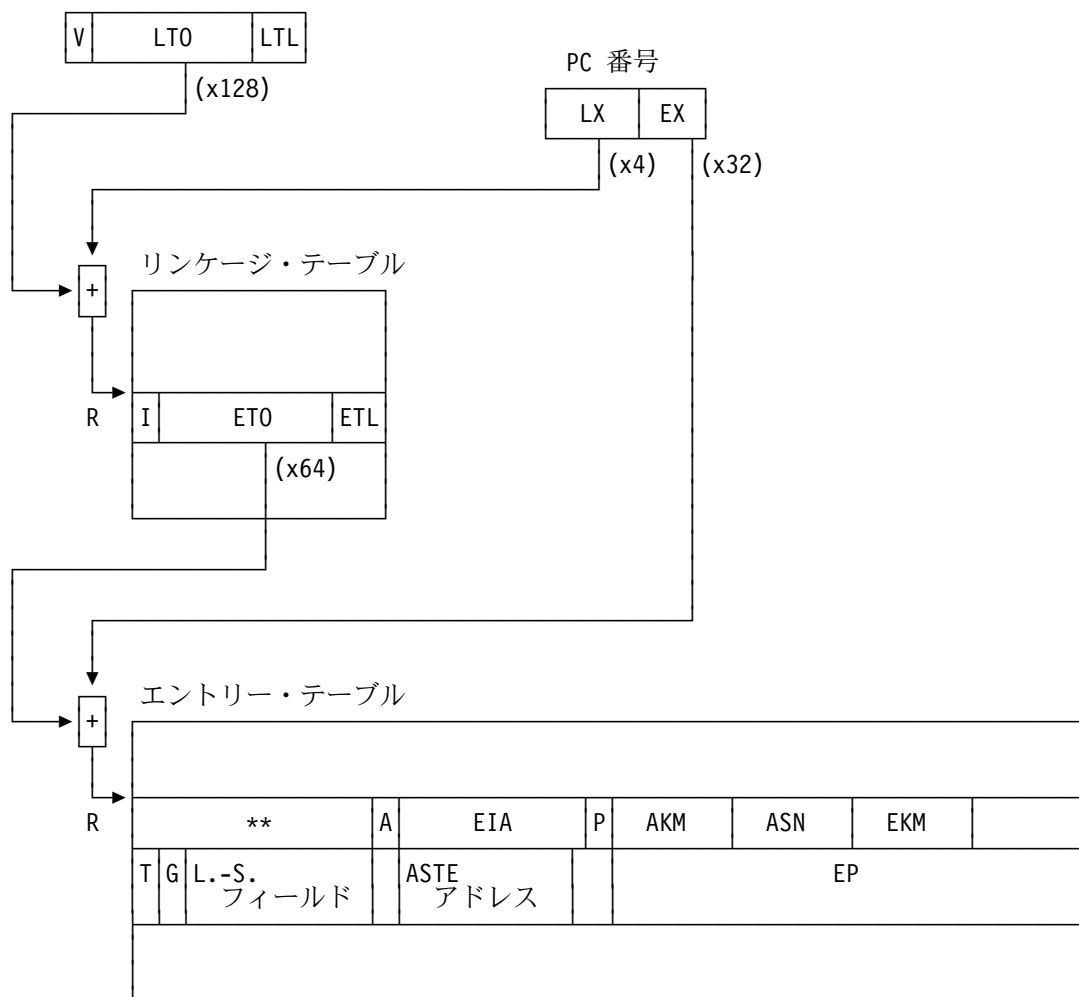
PC 番号変換のために、20 ビットの PC 番号が 2 つの部分に分割されます。左端 12 ビットは、リンケージ指標 (LX) と呼ばれ、右端 8 ビットは、エントリー指標 (EX) と呼ばれます。LX は、リンケージ・テーブルからエントリーを選択するために使用されます。このテ

ブルの開始アドレスと長さは、1 次 ASTE 中のリンケージ・テーブル指定により指定されます。このエントリーは、使用するエントリー・テーブルを指定します。そして、このエントリー・テーブルからエントリーを選択するために、PC 番号の EX フィールドが使用されます。

PC 番号変換の目的で、1 次 ASTE、リンケージ・テーブル、およびエントリー・テーブルからエントリーを取り出すために主記憶機構へのアクセスが行われるときには、キー制御保護は適用されません。

PC 番号変換プロセスを、図5-7 に示します。

1 次 ASTE 中のリンケージ・テーブル指定



R: アドレスは実アドレスです。
 **: G が 1 の場合は、ETE の最初のワードおよび A は EIA のビット 0-32 です。

図 5-7. PC 番号変換

リンケージ・テーブル指定の取得

リンケージ・テーブル指定は、1 次 ASN 第 2 テーブル・エントリーのバイト 24-27 から取得されます。このエントリーの開始アドレスは、制御レジスター 5 の内容により指定されます。

リンケージ・テーブル指定の 31 ビット実アドレスは、1 次 ASTE 起点 (制御レジスター 5 のビット 33-57) の右側に 6 個の 0 を付加し、それに 24 を加算したものです。この加算により、ビット位置 0 への繰り上がりが生じることはありません。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

他の CPU からは、リンケージ・テーブル指定の 4 バイトすべてが、1 次 ASTE から同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。リンケージ・テーブル指定を取り出すために生成される記憶アドレスが、構成内で使用可能でない位置を指定している場合は、アドレッシング例外が認識され、操作は抑止されます。リンケージ・テーブル指定以外には、1 次 ASTE のフィールドは検査されません。

リンケージ・テーブル索引

PC 番号のリンケージ指標部分 (LX) がリンケージ・テーブル起点と一緒に使用されて、リンケージ・テーブルからエントリーが選択されます。

リンケージ・テーブル・エントリーの 31 ビット実アドレスは、リンケージ・テーブル指定のビット 1-24 の内容の右側に 7 個の 0 を付加し、さらに、右側に 2 個、左側に 17 個の 0 を付加したリンケージ指標を加算したものです。この加算によりビット位置 0 への繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{31} - 1$ から 0 へと循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

リンケージ・テーブル索引プロセスの一環として、リンケージ指標の左端 7 ビットをリンケージ・テーブル長 (リンケージ・テーブル指定のビット 25-31) と比較することにより、アドレッシングされたエントリーがリンケージ・テーブル内にあるかどうか判定されます。リンケージ・テーブル長フィールドの値が、リンケージ指標の左端 7 ビットの値より小さい場合は、LX 変換例外が認識されます。

他の CPU からは、リンケージ・テーブル・エントリーの 4 バイトすべてが同時に取り出されているように見えます。この取り出しアクセスは記憶保護の対象にはなりません。リンケージ・テーブル・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

リンケージ・テーブル・エントリーのビット 0 は、リンケージ指標に対応するエントリー・テーブルが使用可能かどうかを指定します。このビットが検査され、その値が 1 であれば、LX 変換例外が認識されます。

リンケージ・テーブル索引のプロセスで例外が認識されなければ、リンケージ・テーブルから取り出されたエントリーは、対応するエントリー・テーブルの起点と長さを指しています。

エントリー・テーブル索引

PC 番号のエントリー指標部分 (EX) と、リンケージ・テーブル・エントリーに含まれているエントリー・テーブル起点を使用して、エントリー・テーブルからエントリーが選択されます。

エントリー・テーブル・エントリーの 31 ビット・アドレスは、エントリー・テーブル起点の右側に 6 個の 0 を付加し、それに、右側に 5 個、左側に 18 個の 0 を付加したエントリー指標を加算したものです。この加算によりビット位置 0 への繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、テーブルは $2^{31} - 1$ から 0 へと循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

エントリー・テーブル索引プロセスの一環として、エントリー指標の左端 6 ビットをエントリー・テーブル長 (リンケージ・テーブル・エントリーのビット 26-31) と比較することにより、アドレッシングされたエントリーがテーブル内にあるかどうか判定されます。エントリー・テーブル長フィールドの値が、エントリー指標の左端 6 ビットの値より小さい場合は、EX 変換例外が認識されます。

上記の実アドレスを使用して、32 バイトのエントリー・テーブル・エントリーが取り出されます。他の CPU からは、このエントリーの取り出しはワード単位のように見え、最左端のワードが最初に取り出されません。残りの 7 ワードが取り出される順序は、予測不能です。この取り出しアクセスは記憶保護の対象にはなりません。エントリー・テーブル・エントリーを取り出すた

めに生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

エントリー・テーブル・エントリーから取り出された情報の利用については、PROGRAM CALL 命令の定義の中で説明されています。

PC 番号変換時の例外認識

PC 番号変換プロセスで検出される例外と、それぞれの優先順位については、PROGRAM CALL の定義の中で説明されています。

プログラミング上の注意: 1 次 ASTE のビット 0 (ASX 無効ビット) の値に関係なく、リンケージ・テーブル指定は 1 次 ASN 第 2 テーブル・エントリーから正常に取り出されます。他の環境では、このビットの値が 1 のときは例外が認識されることがあります。

ホーム・アドレス・スペース

制御プログラムなどのような特権プログラムが、ディスパッチ可能単位 (例えばタスク) のホーム・アドレス・スペースの制御権とアクセス権を獲得するために使用できる機能が用意されています。

通常、個々のディスパッチ可能単位にそれぞれのアドレス・スペースが 1 つずつ関連付けられており、制御プログラムは、そのディスパッチ可能単位を表す基本制御ブロックをそのアドレス・スペース内に保持しています。このアドレス・スペースを、ディスパッチ可能単位のホーム・アドレス・スペースと呼びます。異なるディスパッチ可能単位が、同じホーム・アドレス・スペースを持つことも、それぞれ異なるホーム・アドレス・スペースを持つこともできます。制御プログラムは、ディスパッチ可能単位を開始するときに、1 次アドレス・スペースと 2 次アドレス・スペースを、そのディスパッチ可能単位のホーム・アドレス・スペースと同じに設定することができます。しかし、その後、ディスパッチ可能単位での PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、または SET SECONDARY ASN 命令の使用の結果、一般に、例えば制御プログラムが割り込みを処理するためなどにホーム・アドレス・スペースにアクセスする必要があるときに、1 次アドレス・スペースまたは 2 次アドレス・スペースのいずれかがホーム・アドレス・スペースであるという保証はなくなります。したがって、通常、制御プログラムは、ホーム・アドレス・スペースにアクセスする必要があるとき

に、確実にホーム・アドレス・スペースをアドレッシングするための特殊なアクションをとる必要があります。ホーム・アドレス・スペース機能は、このアクションをとるための効率的な手段を提供します。

ホーム・アドレス・スペース機能には以下のものがあります。

- 制御レジスター 13 の中のホーム・アドレス・スペース制御エレメント (HASCE)。HASCE は、制御レジスター 1 の中の 1 次アドレス・スペース制御エレメント (PASCE)、および制御レジスター 7 の中の 2 次アドレス・スペース制御エレメント (SASCE) と同様の方法で、DAT により使用されません。
- ホーム・スペース・モード。このモードになるのは、DAT がオンで、アドレス・スペース制御 (PSW ビット 16 および 17) の値が 2 進数 11 のときです。CPU がホーム・スペース・モードになっているときは、命令アドレスと論理アドレスはホーム仮想アドレスであり、DAT は HASCE を使用してこれらのアドレスを変換します。
- RESUME PROGRAM、SET ADDRESS SPACE CONTROL、および SET ADDRESS SPACE CONTROL FAST 命令によって監視プログラム状態でホーム・スペース・モードを設定する能力、および、INSERT ADDRESS SPACE CONTROL 命令によってホーム・スペース・モードの指示を戻す能力。
- ホーム・スペース切り替えイベント制御 (制御レジスター 13 のビット 57)。
- RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令が完了したときの、スペース切り替えイベントの認識。これが認識されるのは、CPU が、操作の前または後のいずれかに (両方ではなく) ホーム・スペース・モードになっており、(1) 1 次スペース切り替えイベント制御 (制御レジスター 1 のビット 57) が 1 であるか、(2) ホーム・スペース切り替えイベント制御が 1 であるか、または、(3) PER イベントが示される場合です。

スペース切り替えイベントは、特定のアドレス・スペース内で命令の取り出しが開始または終了するときに、PER またはトレースを使用可能または使用不可にするために使用できます。

アクセス・レジスタの概要

アクセス・レジスタに関連したほとんどの機能は、このセクションと、5-10ページの『リンケージ・スタックなしのサブルーチン・リンケージ』、5-42ページの『アクセス・レジスタの変換』、および5-77ページの

『記憶機構参照の順序』で説明しています。さらに、変換モードおよびアクセス・リスト制御保護については、第3章、『記憶機構』で説明します。PER を使用して記憶域変更イベントを指定のアドレス・スペースのみに限定する方法と、リセット時および状況記憶操作時のアクセス・レジスタの取り扱い方については、第4章、『制御』で説明します。割り込みについては、第6章、『割り込み』で説明します。命令については、第7章、『一般命令』および第10章、『制御命令』で説明します。マシン・チェック割り込み時のアクセス・レジスタの取り扱いと、プログラムによるアクセス・レジスタの妥当性検査については、第11章、『マシン・チェックの取り扱い』で説明します。アクセス・レジスタの変更/表示制御については、第12章、『オペレーター機能』で説明します。

要約

提供されている主な機能には、以下のものがあります。

- 準特権プログラムが即時および同時に使用できる最大 16 個のアドレス・スペース (命令スペースを含む)。アドレス・スペースは、アクセス・レジスタと呼ばれる 16 個のレジスタにより指定されます。
- アクセス・レジスタの内容の検査および変更を行う命令。

さらに、上記の機能を制御するための制御メカニズムと権限メカニズムが組み込まれています。

アクセス・レジスタを使用すると、一連の命令、または MOVE (MVC) や MOVE LONG (MVCL) などの単一の命令を、複数のアドレス・スペース内の記憶域オペランドを対象として実行することができ、しかも、個々のアドレス・スペースごとに変換モードやその他の制御情報を変更する必要はありません。したがって、1 つのアドレス・スペース内に常駐するプログラムが、完全な命令セットを使用して、そのアドレス・スペースとその他の最大 15 個のアドレス・スペース内のデータを処理し、これらすべてのアドレス・スペース間で任意にデータを移動することができます。また、プログラムは、アクセス・レジスタの内容を変更することにより、さらに別のアドレス・スペースにもアクセスできます。

アクセス・レジスタの内容を検査および変更する命令は非特権命令であり、第7章、『一般命令』で説明されています。この種の命令には以下のものがあります。

- COPY ACCESS
- EXTRACT ACCESS
- LOAD ACCESS MULTIPLE
- LOAD ADDRESS EXTENDED
- SET ACCESS
- STORE ACCESS MULTIPLE

アクセス・レジスタに関連して使用される PURGE ALB 特権命令および COMPARE AND SWAP AND PURGE 命令については、第10章、『制御命令』で説明されています。

アクセス・レジスタは、CPU がアクセス・レジスタ・モードにあるときのアドレス・スペースを指定します。アクセス・レジスタ・モードを設定するには、SET ADDRESS SPACE CONTROL 命令および SET ADDRESS SPACE CONTROL FAST 命令を使用し、アクセス・レジスタ・モードを指示するには、INSERT ADDRESS SPACE CONTROL 命令を使用します。また、スタッキング命令 PROGRAM CALL、PROGRAM RETURN、および RESUME PROGRAM も、アクセス・レジスタ・モードを設定するために使用できます。これらの命令は、すべて、第10章、『制御命令』で説明されています。

BRANCH IN SUBSPACE GROUP 命令では、アクセス・レジスタは特殊な使い方がされます。この命令でのアクセス・レジスタの使用法について詳しく説明してあるのは、第10章、『制御命令』の中のこの命令に関する定義の中だけです。ただし、BRANCH IN SUBSPACE GROUP によるディスパッチ可能単位制御テーブルおよび拡張 ASN 第 2 テーブルの使用については、5-55ページの『サブスペース・グループ・テーブル』で説明します。

アクセス・レジスタの機能

アクセス・レジスタ指定のアドレス・スペース

CPU には、0-15 の番号が付いた 16 個の 32 ビット・アクセス・レジスタが含まれています。DAT がオンで、PSW のビット 16 および 17 が 2 進数 01 であれば、アクセス・レジスタ・モードになります。このモードでは、記憶域オペランドの論理アドレスを指定するために使用する命令の B または R フィールドは、汎用レジスタだけでなくアクセス・レジスタも指定します。指定された汎用レジスタは、通常どおりに、記憶

域オペランドの論理アドレスを形成するために使用されます。指定されたアクセス・レジスタは、その論理アドレスに関連するアドレス・スペースを指定するために使用されます。アクセス・レジスタは、アドレス・スペースのアドレス・スペース制御エレメントを使用してアドレス・スペースを指定し、DAT はそれを論理アドレスに変換します。アクセス・レジスタは、アドレス・スペース制御エレメントを直接含めるのではなく、間接的な方法でアドレス・スペース制御エレメントを指定します。

アクセス・レジスタには、制御レジスタ 1 または 7 の中の 1 次または 2 次アドレス・スペース制御エレメントを指定することも、ASN 第 2 テーブル・エントリに含まれているアドレス・スペース制御エレメントを指定することもできます。後者の場合は、アクセス・レジスタは、テーブル内のアクセス・リストと呼ばれるエントリを指示し、そのアクセス・リスト・エントリが ASN 第 2 テーブル・エントリを指示します。

アクセス・レジスタの内容を使用して、DAT が使用するアドレス・スペース制御エレメントを取得するプロセスを、アクセス・レジスタ変換 (ART) と呼びます。図 5-8 はこのプロセスを示しています。

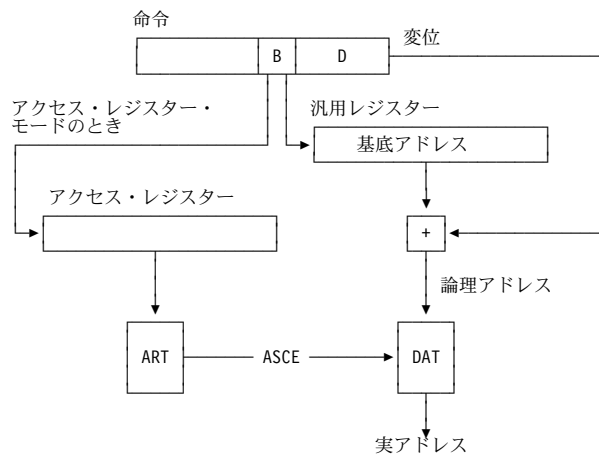


図 5-8. アクセス・レジスタの使用

アクセス・レジスタは、AR 指定アドレス・スペース制御エレメントを使用して AR 指定アドレス・スペースを指定するものです。AR 指定アドレス・スペース内の仮想アドレスを、AR 指定仮想アドレスと呼びます。

アクセス・レジスタ・モードでは、記憶域オペランド・アドレスはすべて AR 指定仮想アドレスですが、命令アドレスは 1 次仮想アドレスです。

アクセス・レジスタの指示: アクセス・レジスタ・モードでは、命令の B または R フィールドは、以下の条件下でアクセス・レジスタ変換に使用するアクセス・レジスタを指示します。

- フィールドが、基底アドレスを含む汎用レジスタを指示する B フィールドである。基底アドレスと一緒に、変位 (D) のほか場合によっては指標 (X) が使用されて、記憶域オペランドの論理アドレスを形成します。
- フィールドが、記憶域オペランドの論理アドレスを含む汎用レジスタを指示する R フィールドである。

例えば、次のような命令があるとします。

MVC 0(L,1),0(2)

この場合、長さ L の第 2 オペランドが、第 1 オペランド位置に移動されます。第 2 オペランドの論理アドレスは汎用レジスタ 2 の中にあり、第 1 オペランド位置の論理アドレスは汎用レジスタ 1 の中にあります。第 2 オペランドを含むアドレス・スペースはアクセス・レジスタ 2 で指定され、第 1 オペランド位置を含むアドレス・スペースはアクセス・レジスタ 1 で指定されます。この 2 つのアドレス・スペースは異なるアドレス・スペースでもよく、また、それぞれ現命令スペース (1 次アドレス・スペース) と違っていても構いません。

PSW ビット 16 および 17 が 01 のときは、LOAD REAL ADDRESS および STORE REAL ADDRESS 命令の B₂ フィールドは、DAT がオンかオフに関係なく、アクセス・レジスタ変更に使用するアクセス・レジスタを指示します。

COMPARE AND FORM CODEWORD および UPDATE TREE 命令は、暗黙指定された汎用レジスタおよびアクセス・レジスタにより、記憶域オペランドを指定します。

MOVE TO PRIMARY および MOVE TO SECONDARY 命令は、1 次仮想アドレスおよび 2 次仮想アドレスを使用して記憶域オペランドを指定します。これらの命令にはアクセス・レジスタは適用されません。これらの命令をアクセス・レジスタ・モードで実行すると、例外が認識されます。アクセス・レジスタ・モードでは、MOVE TO PRIMARY および MOVE TO SECONDARY の代わりに MOVE WITH KEY 命令を使用できます。また、MOVE WITH SOURCE KEY および MOVE WITH DESTINATION KEY 命令も使用できます。

命令の R フィールドには、アクセス・レジスタ変換以外の目的に使用するアクセス・レジスタも指定できます。

各命令の項の冒頭にある要約図には、アクセス・レジスタ変換を目的とするものかどうかに関係なく、アクセス・レジスタを指定できるフィールドがすべて示されています。

アドレス・スペース制御エレメントの取得: このセクション以降の幾つかのセクションで、アクセス・レジスタ変換プロセスの概要を説明し、アクセス・リストに関連した概念を紹介します。

アクセス・レジスタに指定されるアドレス・スペース制御エレメントは、以下のようにして、アクセス・レジスタ変換により取得されます。

- アクセス・レジスタに 16 進数 00000000 が含まれている場合は、指定されるアドレス・スペース制御エレメントは 1 次アドレス・スペース制御エレメント (PASCE) で、これは制御レジスタ 1 から取得されます。
- アクセス・レジスタに 16 進数 00000001 が含まれている場合は、指定されるアドレス・スペース制御エレメントは 2 次アドレス・スペース制御エレメント (SASCE) で、これは制御レジスタ 7 から取得されます。
- アクセス・レジスタにその他の値が含まれている場合は、指定されるアドレス・スペース制御エレメントは ASN 第 2 テーブル・エントリから取得されます。アクセス・レジスタの内容は、ASN 第 2 テーブル・エントリの実起点を含むアクセス・リスト・エントリを指示します。

アクセス・レジスタ変換では、アクセス・レジスタ 0 は特殊な取り扱い方がされます。つまり、このレジスタは 16 進数 00000000 を含むものとして扱われ、実際の内容は検査されません。したがって、アクセス・レジスタ・モードで、B または R フィールドが 0 の場合に指定される論理アドレスは、アクセス・レジスタ 0 の内容に関係なく、常に 1 次アドレス・スペースに関連したアドレスです。ただし、アクセス・レジスタ 0 の取り扱いには例外が 1 つあります。つまり、TEST ACCESS 命令では、アクセス・レジスタ 0 は 16 進数 00000000 を含むものとは見なされず、レジスタの実際の内容が使用されます。

アクセス・レジスタ・モードにおいて、16 進数 00000000 を値として含むアクセス・レジスタが現 1 次アドレス・スペースを指示するものと見なす処理によ

り、アクセス・リスト・エントリを使用せずに、そのアドレス・スペースをアドレッシングすることができません。これは、スペース切り替えを伴う PROGRAM CALL (PC-ss)、PROGRAM RETURN (PR-ss)、または PROGRAM TRANSFER (PT-ss) 命令により 1 次アドレス・スペースが変更される場合に、便利です。同様に、16 進数値 00000001 を含むアクセス・レジスタが 2 次アドレス・スペースを指示するものと見なす処理により、スペース切り替え操作の後で、アクセス・リスト・エントリを使用せずにそのスペースをアドレッシングすることができます。

アクセス・レジスタの内容は、PROGRAM CALL および PROGRAM TRANSFER 命令によって変更されることはありません。したがって、16 進数 00000000 または 00000001 を含むアクセス・レジスタは、PROGRAM CALL または PROGRAM TRANSFER の実行後には、実行前とは異なるアドレス・スペースを指定していることがあります。例えば、スペース切り替えを伴う PROGRAM CALL 命令が実行された場合、16 進数 00000000 を含むアクセス・レジスタは、実行前には古い 1 次アドレス・スペースを示し、実行後には新しい 1 次アドレス・スペースを示しています。

アクセス・レジスタ変換により、ASN 第 2 テーブル・エントリからアドレス・スペース制御エレメントが取得される場合は、そのエントリのビット 0 (ASX 無効ビット) は 0 でなければなりません。さもないと例外が認識されます。

アクセス・リスト: アクセス・レジスタの内容が指示するアクセス・リスト・エントリは、ディスパッチ可能単位アクセス・リストおよび 1 次スペース・アクセス・リストの 2 つのアクセス・リストのいずれかに入れていることができます。2 つのアクセス・リストのどちらに指定のエントリが含まれているかは、アクセス・レジスタ内のビットの 1 に示されます。これらのアクセス・リストは、どちらも実記憶域または絶対記憶域の中にあります。アクセス・リストの位置は、制御レジスタ 2 および 5 に指定されます。

制御レジスタ 2 には、ディスパッチ可能単位制御テーブルと呼ばれる実記憶域の起点が含まれています。ディスパッチ可能単位制御テーブルには、ディスパッチ可能単位アクセス・リストの指定 (実起点または絶対起点および長さ) が含まれています。

制御レジスタ 5 には、1 次 ASN 第 2 テーブル・エントリと呼ばれる実記憶域の起点が含まれています。1 次 ASN 第 2 テーブル・エントリには、1 次スペース・アクセス・リストの指定が含まれています。

アクセス・リスト (ディスパッチ可能アクセス・リストまたは 1 次スペース・アクセス・リスト) には、最大 1,024 エントリーの範囲内で、8 個の 16 バイト・エントリーの整数倍に相当するエントリーが含まれていません。

プログラムとディスパッチ可能単位: アクセス・リストについて考えるときは、「プログラム」と「ディスパッチ可能単位」という用語を区別することが必要です。プログラムは命令のシーケンスであり、プログラム・モジュールと呼ばれることもあります。プログラムは、呼び出し側プログラムと呼び出し先プログラムから成るシーケンスの場合もあります。ディスパッチ可能単位 (プロセスまたはタスクとも呼ばれます) は、1 つの CPU によるプログラム実行で一度に行われる作業の単位です。

ディスパッチ可能単位アクセス・リストは、特定のディスパッチ可能単位に関連付けるように意図されています。つまり、ディスパッチ可能単位を処理するために現在どのプログラムが実行されているかに関係なく、同じディスパッチ可能単位は常に同じディスパッチ可能単位アクセス・リストを持つように意図されています。

LOAD CONTROL 命令以外には、制御レジスター 2 中のディスパッチ可能制御テーブル起点を変更する手段はありません。

1 次スペース・アクセス・リストは、制御レジスター 4 中の 1 次 ASN および制御レジスター 1 中の 1 次アドレス・スペース制御エレメントにより指定される 1 次アドレス・スペースと関連付けられます。あるディスパッチ可能単位の 1 次アドレス・スペースが変化すると、つまり、そのディスパッチ可能単位を処理するために別の 1 次アドレス・スペース内にあるプログラムの実行が開始されると、そのディスパッチ可能単位で使用できる 1 次スペース・アクセス・リストが変わります。

LOAD ADDRESS SPACE PARAMETERS、PROGRAM CALL、PROGRAM RETURN、または PROGRAM TRANSFER 命令は、制御レジスター 4 中の 1 次 ASN および制御レジスター 1 中の 1 次アドレス・スペース制御エレメントを置き換えるときに、制御レジスター 5 中の 1 次 ASN 第 2 テーブル・エントリー起点も置き換えます。

したがって、1 つのディスパッチ可能単位については、ディスパッチ可能単位アクセス・リストは常に一定であるように意図されています (後で述べるようにリスト・エントリーが変化することはあります)。また、1 次スペース・アクセス・リストは、どのプログラムが実行され

ているかによって、そしてそのプログラムの 1 次アドレス・スペースによって決まります。さらに、同じ 1 次アドレス・スペース内のすべてのディスパッチ可能単位およびプログラムが、同じ 1 次スペース・アクセス・リストを持ちます。

アクセス・リスト・エントリー・トークン: アクセス・レジスターの内容は、一般にアクセス・リスト内のエントリーを指定するものなので、アクセス・リスト・エントリー・トークン (ALET) と呼ばれます。ALET のフォーマットは以下のとおりです。

0000000	P	ALESN	ALEN
0	7 8	16	31

ALET の 1 次リスト・ビット (P) は、指定されたアクセス・リスト・エントリーがどちらのアクセス・リストに含まれているかを指定します。このビットが 0 ならディスパッチ可能単位アクセス・リストで、1 なら 1 次スペース・アクセス・リストです。指定されたアクセス・リストを、有効アクセス・リストと呼びます。

ALET には、アクセス・リスト・エントリー番号 (ALEN) も含まれています。この番号に 16 を掛けた値が、有効アクセス・リストの始めから、指定されたアクセス・リスト・エントリーまでのバイト数になります。アクセス・レジスター変換の際に、ALEN が有効アクセス・リスト内に存在しないエントリーを指している場合、または ALET の左端 7 ビットがすべて 0 ではない場合は、例外が認識されます。

ALET 中のアクセス・リスト・エントリー・シーケンス番号 (ALESN) については、この次のセクションで説明します。

ALET が 16 進数 00000000 または 00000001 であるときは、上記の ALET のフォーマットは適用されません。

ALET は、アクセス・レジスター、汎用レジスター、または記憶域内に置くことができ、問題プログラムによる操作からの特別な保護はありません。どのプログラムも、アクセス・レジスター、汎用レジスター、および記憶域の間で任意に ALET を転送することができます。呼び出し先プログラムは、そのプログラムが使用できる任意の記憶域にアクセス・レジスターの内容を保管し、そのプログラム自身の目的のためにアクセス・レジスターをロードして使用し、そして、呼び出し側に戻るまえにアクセス・レジスターの元の内容を復元することができます。

アクセス・リスト・エントリーの割り振りと無効化:

アクセス・リストは、制御プログラムから提供され、どの問題プログラムによる直接的な操作からも保護されるものとして規定されています。この保護を達成するには、キー制御保護を利用する方法と、DAT を使用して、どの問題プログラムからもアクセスできない実記憶域にアクセス・リストを入れる方法があります。

アクセス・リスト・エントリーは有効なときと無効なときがあり、これはエントリー内のビットの 1 つによって決まります。有効なアクセス・リスト・エントリーは特定のアドレス・スペースを指定するものであり、適切な許可を持つプログラムがそのスペースにアクセスするために使用できます。無効なアクセス・リスト・エントリーは、有効なエントリーとして割り振るために使用できます。有効なアクセス・リスト・エントリーを割り振り、割り振り済みのエントリーを無効化するサービスは、制御プログラムが提供する必要があります。

アクセス・リスト・エントリーの割り振りは、以下のようなステップから成っています。まず、問題プログラムが、アドレス・スペースを表す何らかの識別を制御プログラムに渡し、また、ディスパッチ可能単位アクセス・リストかそれとも 1 次スペース・アクセス・リストかの指定を渡します。制御プログラムは、何らかの手段で、問題プログラムがそのアドレス・スペースにアクセスする権限を持っているかどうかを検査します。問題プログラムがその権限を持っている場合は、制御プログラムは、指定されたアクセス・リスト内の無効なエントリーを選択し、それを対象アドレス・スペースを指定する有効なエントリーに変更し、そして、割り振られたエントリーを示すアクセス・リスト・エントリー・トークン (ALET) を問題プログラムに戻します。これで、問題プログラムは、アクセス・レジスターにその ALET を入れることにより、このアドレス・スペースにアクセスできるようになります。後で、制御プログラムの無効化サービスを使用して、割り振られたアクセス・リスト・エントリーを無効化することができます。無効なアクセス・リスト・エントリーを指定する ALET が使用されると、アクセス・レジスター変換の時点で例外が認識されます。

特定のアクセス・リスト・エントリーが割り振られた後、無効化され、さらに、最初とは異なるアドレス・スペースを指定して再び割り振られることがあります。概念的に正しくないアドレス・スペースを指定する ALET が誤って使用されるのを防止するために、ALET とアクセス・リスト・エントリーの両方にアクセス・リスト・エントリー・シーケンス番号 (ALESN) が指定されます。制御プログラムは、アクセス・リスト・エントリーを割り振るときに、そのエントリーと、そのエントリーを指すものとして問題プログラムに戻す ALET に、同じ ALESN を入れる必要があります。そして、アクセ

ス・リスト・エントリーを再割り振りするときに、制御プログラムは、ALESN の値を変更する必要があります。使用された ALET 中の ALESN が、指定されたアクセス・リスト・エントリー内の ALESN と同じでない場合は、例外が認識されます。

ALET は問題プログラムから保護されるものではなく、問題プログラムは ALET 中の ALESN を任意の値に変更できるので、ALESN 検査は権限メカニズムではなく、信頼性確保のためのメカニズムと考えることができます。さらに、ALESN は 1 バイトであり、各割り振りのたびに値が 1 ずつ増加するものとすれば、256 回の再割り振りの後で値は循環するので、ALESN 検査はフェイルプールの信頼性メカニズムでもありません。

アクセス・リスト・エントリーの使用許可: アクセス・リストは、特定のディスパッチ可能単位または 1 次アドレス・スペースと関連付けられるものですが、このリスト内の有効なエントリーは、それぞれ、ディスパッチ可能単位の作業を行うために一定の順序で実行される、異なるプログラムと関連付けられるようになっています。各プログラムは、それぞれのプログラムに関連付けられたアクセス・リスト・エントリーのみを使用できる、特定の権限を持つことができます。ここで言う権限は、制御レジスター 8 の中の 16 ビット拡張許可指標 (EAX) が表しているものです。

関連の許可メカニズムで使用されるその他のエレメントには、(1) アクセス・リスト・エントリー内の専用ビット、(2) アクセス・リスト・エントリー中のアクセス・リスト・エントリー許可指標 (ALEAX)、および、(3) 権限テーブルがあります。

プログラムが、アクセス・レジスター変換の際にアクセス・リスト・エントリーの使用を許可されるのは、以下の条件のいずれかが満たされている場合です。

1. アクセス・リスト・エントリー内の専用ビットが 0 である。この条件は、特定のディスパッチ可能単位を処理するために実行されるすべてのプログラムに許可を与えるための高性能な手段です。
2. アクセス・リスト・エントリー内の ALEAX が、制御レジスター 8 の中の EAX に等しい。この条件は、特定のプログラムのみ許可を与えるための高性能な手段です。
3. EAX が、アクセス・リスト・エントリーにより指定されているアドレス・スペースに関連した権限テーブル内で、1 である 2 次ビットを選択している。権限テーブルの位置は、アクセス・リスト・エントリーにアドレス・スペースの ASN 第 2 テーブル・エントリー (ASTE) の実起点が含まれており、

ASTE に権限テーブルの実起点が含まれていることから、突き止められます。この条件は、特定プログラムのみを許可するための別の手段ですが、上記の条件 2 より性能は劣ります。ただし、条件 2 の代わりに条件 3 が満たされるようにした場合は、それぞれ異なる EAX を使用して実行される複数のプログラムが、同じアクセス・リスト・エントリーを使用して特定のアドレス・スペースにアクセスできるという利点があります。

アクセス・レジスタ変換では、上記の 3 つの条件が満たされているかどうか、上記のバージョン順にテストされます。小さい番号の条件がどれか満たされていれば、それより大きい番号の条件が満たされているかどうかのテストは行われません。条件が 1 つも満たされていない場合は、例外が認識されます。

図5-9 は、許可メカニズムをどのように使用できるかの例を示しています。図の中の「PBZ」は専用ビットが 0 であることを意味し、「PBO」は専用ビットが 1 であることを意味します。

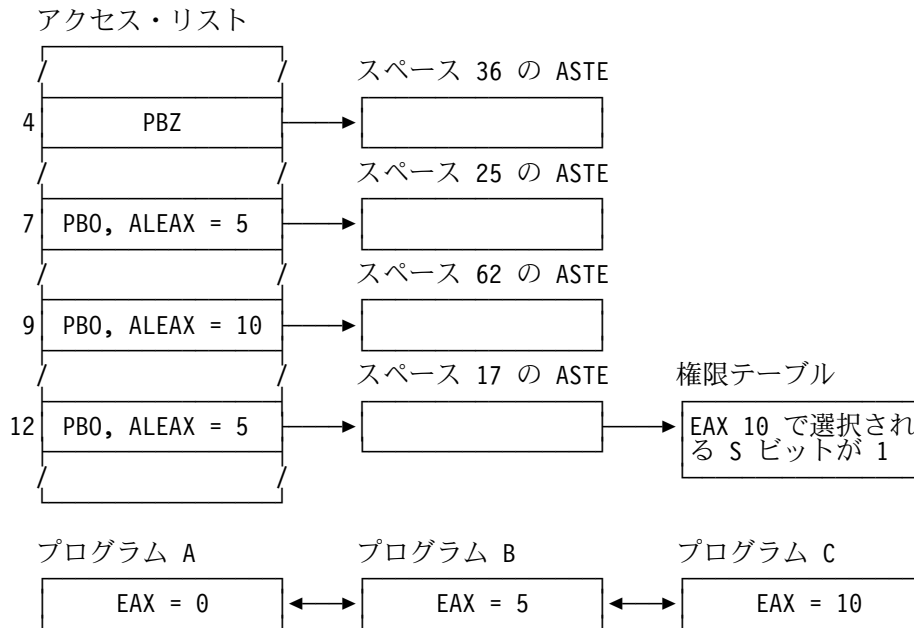


図 5-9. アクセス・リスト・エントリーの使用許可の例

この図に示されているアクセス・リスト (ディスパッチ可能単位アクセス・リスト) では、注目すべきエントリーは、4、7、9、および 12 です。各アクセス・リスト・エントリーには、専用ビット、ALEAX、および、アドレス・スペースの ASTE の実起点が含まれています。エントリー 4 の専用ビットは 0 であり、したがってエントリー 4 の ALEAX の値は重要ではないので、示されていません。エントリー 7、9、および 12 はの専用ビットは 1 なので、これらのエントリーの ALEAX は示されています。アドレス・スペースを識別するために使用されている番号 (36、25、62、17) は、任意の数値です。これらの番号はアドレス・スペースの ASN とすることもできますが、アクセス・レジスタ変換では ASN は使用されません。権限テーブルは、アドレス・スペース 17 についてのみ示されています。このテーブルでは、EAX 10 により 2 次ビットとして 1 が選択さ

れています。その他のスペースの権限テーブルでは、2 次ビットはどれも 1 ではないものと想定しています。

この図は、A、B、および C という 3 つのプログラムのシーケンスも示しています。このシーケンスは、アクセス・リストに関連したディスパッチ可能単位の作業を行うために実行されます。これらのプログラムは、同じアドレス・スペース内にあっても、異なるアドレス・スペース内にあっても構いません。この 3 つのプログラムが実行される時の制御レジスタ 8 の EAX は、それぞれ、0、5、および 10 です。

アクセス・リスト・エントリー (ALE) 4 の専用ビットが 0 なので、プログラム A、B、および C は、それぞれ ALE 4 を使用してアドレス・スペース 36 にアクセスできます。ALE 7 の ALEAX はプログラム B の EAX に等しいので、プログラム B は ALE 7 を使用してスペース 25 にアクセスできますが、他のプログラム

はこの ALE を使用することはできません。同様に、ALE 9 を使用できるのは、プログラム C のみです。ALE 12 の ALEAX とプログラム B の EAX が等しいので、プログラム B は ALE 12 を使用できます。また、プログラム C の EAX は、スペース 17 の権限テーブル内で 1 である 2 次ビットを選択しているため、プログラム C も ALE 12 を使用できます。

この例で、プログラム A、B、および C がすべて同じアドレス・スペース内にあり、アクセス・リストがそのスペースの 1 次スペース・アクセス・リストである場合でも、結果は同じです。

専用ビットが 0 である ALE は、現 EAX の値に関係なくどのプログラムでも使用できるものなので、共用 ALE と呼ばれます。専用ビットが 1 である ALE は、プログラムがその ALE を使用できるかどうかは現 EAX によって決まるので、専用 ALE と呼ばれます。

許可メカニズムについての注意: コンピュータ科学の分野で使用されている「可能性」という言葉の意味に従えば、アクセス・リストは一種の可能性リストと考えることができます。つまり、アクセス・リスト内のエントリを割り振るために使用するポリシーを確立するのは制御プログラムの責任であり、割り振り時に必要とされるプログラミングされた許可検査は、非常に複雑で長いものになることがあります。アクセス・リスト内に有効なエントリが作成されると、上記の許可メカニズムを使用してアクセス・レジスタ変換プロセスが実行され、効率的な方法で制御プログラム・ポリシーが適用されます。

ASN と権限テーブルのみを使用するよりも、アクセス・リストを使用する方が効率的です。例えば、あるアクセス・レジスタに ASN が含まれており、アクセス・レジスタ変換では、その ASN の ASN 変換が行われた後で、EAX を使用して権限テーブルがテストされるものとします。この場合、EAX は既存のすべてのアドレス・スペースに関連付けられることになるため、EAX の管理およびプログラムへの EAX の割り当ては複雑になります。ALET およびアクセス・レジスタ変換の実際の定義を使用すれば、EAX は、現行のディスパッチ可能単位および 1 次スペース・アクセス・リストで表されているアドレス・スペースのみに関連したものになります。また、ASN 変換はアクセス・レジスタ変換の一部としては行われないので、同時に存在するアドレス・スペースの数 (ASN 第 2 テーブル・エントリにより表される) が、使用可能な ASN の数 (64K) より大きくなる場合があります。

EAX をプログラムに割り当て、プログラム・リンケージの際に制御レジスタ 8 内で EAX を変更するには、エントリ・テーブル・エントリとリンケージ・スタックを使用することができます。これらのコンポーネントについては、5-59 ページの『リンケージ・スタックの概要』で説明します。制御レジスタ 8 の中の EAX の保管と変更には、特権命令 EXTRACT AND SET EXTENDED AUTHORITY も使用できます。

アクセス・レジスタの使用に当たっては、SET SECONDARY ASN 命令および許可指標 (AX)(制御レジスタ 4 のビット 32-47) が一定の役割を果たすことができます。スペース切り替え形式の SET SECONDARY ASN (SSAR-ss) は、新しい 2 次スペースに関連した権限テーブル内で AX により選択された 2 次ビットが 1 である場合に、新しい 2 次アドレス・スペースを確立します。2 次スペースは、16 進数 00000001 の値を持つ ALET を使用してアドレッシングできます。

アクセス能力の取り消し: アクセス・レジスタ変換の一部となっているもう 1 つのメカニズムとして、権限メカニズムと保全性メカニズムを組み合わせたものがあります。このセクションではこのメカニズムについて説明します。

アクセス・リスト・エントリ (ALE) には ASN 第 2 テーブル・エントリ・シーケンス番号 (ASTESN) が含まれており、ALE が指示する ASTE にもこの番号が含まれています。アクセス・レジスタ変換の際には、ALE 内の ASTESN は指示されている ASTE 内の ASTESN に等しくなければなりません。さもないと、例外が認識されます。

制御プログラムは、ALE を割り振るときに、指示されている ASTE から ALE に ASTESN をコピーする必要があります。その上で、制御プログラムは、ASTE 内の ASTESN を変更することにより、ALE が表しているアドレッシング能力を事実上取り消すことができます。ASTE 内の ASTESN を変更すると、その ASTE を指定している前に使用可能だったすべての ALE が、使用不能になります。

このように ALE を使用不能にする必要があるのは、次の 2 つのいずれかの場合です。

1. 制御プログラムのポリシーの中で、プログラムが ASTE により指定されているアドレス・スペースへのアクセス権を持つかどうかを決定する何らかの要素が、変更された場合。これは、そのアドレス・スペースに対するアクセスを持ち、ALE を割り振られている一部またはすべてのプログラムが、

もはやアクセス権を許可されなくなることを意味することがあります。

ASTE 内の ASTESN を変更すると、その ASTE を指定しているすべての ALE が、使用不能になります。この能力取り消しを選択的に行う必要がある場合は、ASTESN が等しくないために例外が認識されたときは、制御プログラムは、権限を決定するためにプログラムされているプロシージャを再適用することができ、使用可能のままにしておく必要がある ALE に新しい ASTESN をコピーすることにより、その ALE を再び使用可能にすることができます。ALE が再び使用可能になったら、通常、例外が検出された命令を制御プログラムで再実行する必要があります。

2. ASTE が再割り当てされた結果、概念的に異なるアドレス・スペースが指定され、旧アドレス・スペースを指定していた ALE ではその新しいアドレス・スペースを指定してはならない場合。(ASTE のビット 0、つまり ASX 無効ビットを 1 に設定すれば、特定のアドレス・スペースへの ASTE の割り当てを削除でき、これによりアクセス・レジスタ変換でその ASTE が使用されないようにすることができます。ただし、再割り当ての後には、通常、ビット 0 は 0 に戻されます。)

ASTESN メカニズムは、上記の最初の場合については権限メカニズムと考えることができ、2 番目の場合については保水性メカニズムと考えることができます。

ASTESN メカニズムは、制御プログラムが、各 ASTE を指示する ALE を含むアクセス・リストを追跡して記録する必要がないという点で、特に便利です。さらに、このメカニズムを使用すれば、アクセス・レジスタ変換で ALE が使用されないようにするために、アクセス・リストから ALE を検索して無効化する必要もありません。後者の操作は、ときには大量の時間を消費することがあるほか、きわめて困難な管理上の問題を引き起こすこともあります。これは、ALE を無効化することが必要になったときに、アクセス・リストが直接アクセス記憶装置のような補助記憶機構に収められていることがあるからです。

ASTESN は 4 バイトのフィールドです。権限ポリシーの変更またはアドレス・スペースの再割り当てが妥当な頻度で行われていれば、約 40 億個の ASTESN 値により、システムの全存続期間にわたってフェイルプールの権限メカニズムと保水性メカニズムを達成できます。

格納参照の防止: アクセス・リスト・エントリーには取り出し専用ビットが含まれており、このビットが 1 のときは、そのアクセス・リスト・エントリーは記憶域オペランド格納参照を行うためには使用できないことを示します。取り出し専用ビットの効果については、3-10 ページの『アクセス・リスト制御保護』で説明されています。

変換のパフォーマンスの向上: アクセス・レジスタ変換 (ART) は、概念的には、アクセス・レジスタ・モードで論理アドレスを使用して記憶域オペランドが参照されるたびに起こります。パフォーマンスを高めるために、ART では、通常、ART テーブル (アクセス・リスト指定ソース、アクセス・リスト、ASN 第 2 テーブル、および権限テーブル) に含まれる一部またはすべての情報を、ART 索引バッファ (ALB) と呼ばれる特殊なバッファ内に維持するという手法が用いられます。これにより、CPU が実記憶域内の ART テーブル・エントリーを参照する必要があるのは、そのエントリーへの最初のアクセスのときだけですみます。このエントリー内の情報が ALB の中に保持されていれば、以後の変換は ALB を使用して行うことができます。

実記憶域内で ART テーブル・エントリーのどれかに変更が加えられた場合は、PURGE ALB 命令および COMPARE AND SWAP AND PURGE 命令を使用して、ALB からすべての情報を消去することができます。

アクセス・レジスタ命令

アクセス・レジスタの内容を検査および変更するための命令には、以下のものがあります。

- COPY ACCESS
- EXTRACT ACCESS
- LOAD ACCESS MULTIPLE
- LOAD ADDRESS EXTENDED
- SET ACCESS
- STORE ACCESS MULTIPLE

SET ACCESS 命令、指定されたアクセス・レジスタの内容を、指定された汎用レジスタの内容で置き換えます。EXTRACT ACCESS 命令は、逆に、アクセス・レジスタの内容を汎用レジスタに移します。COPY ACCESS 命令は、アクセス・レジスタの内容を別のアクセス・レジスタに移します。

LOAD ACCESS MULTIPLE 命令は、指定された連続番号のアクセス・レジスタのセットを、そのアクセス・レジスタ数に等しい長さ (ワード数) の指定記憶位置からロードします。STORE ACCESS MULTIPLE

命令は、逆に、1 セットのアクセス・レジスタの内容を特定の記憶位置に格納します。

LOAD ADDRESS EXTENDED 命令は、命令の B、X、および D フィールドに指定されている有効アドレスを持つ指定の汎用レジスタをロードするという点で、LOAD ADDRESS 命令に似ています。さらに、LOAD ADDRESS EXTENDED は、ロードされる汎用レジスタと同じ番号のアクセス・レジスタも操作します。アドレス・スペース制御 (PSW ビット 16 および 17) が 2 進数 00、10、または 11 のときは、LOAD ADDRESS EXTENDED は、それぞれ 16 進数 00000000、00000001、または 00000002 をアクセス・レジスタにロードします。アドレス・スペース制御が 2 進数 01 のときは、LOAD ADDRESS EXTENDED は、命令の B フィールドによって決定される値をターゲット・アクセス・レジスタにロードします。B フィールドが 0 であれば、LOAD ADDRESS EXTENDED は、16 進数 00000000 をターゲット・アクセス・レジスタにロードします。B フィールドが 0 でない場合は、LOAD ADDRESS EXTENDED は、B フィールドに指定されているアクセス・レジスタの内容を、ターゲット・アクセス・レジスタにロードします。ただし、最後の場合、B に指定されているアクセス・レジスタのビット 0-6 がすべて 0 でないときは、ターゲットの汎用レジスタおよびアクセス・レジスタに生じる結果は、予測不能です。

DAT がオンの場合に、アドレス・スペース制御値が 2 進数 00、01、10、または 11 のときは、それぞれ、1 次スペース・モード、2 次スペース・モード、またはホーム・スペース・モードを示します。LOAD ADDRESS EXTENDED は、DAT がオンかオフかに関係なく同じ働きをします。

アクセス・レジスタ変換に使用される場合、アクセス・レジスタの 16 進数値 00000000 および 00000001 は、それぞれ 1 次アドレス・スペースおよび 2 次アドレス・スペースを指定し、16 進数値 00000002 は、ディスパッチ可能単位アクセス・リスト内のエントリー 2 を指定します。アドレス・スペース制御が 2 進数 11 のときに、ターゲット・アクセス・レジスタに 16 進数 00000002 をロードするのは、制御プログラムでディスパッチ可能単位アクセス・リスト内のエントリー 2 を割り当てることで、ホーム・アドレス・スペースを指定できるようにするためです。

アクセス・レジスタの変換

アクセス・レジスタ変換の概要については、5-34ページの『アクセス・レジスタ指定のアドレス・スペース』で説明されています。

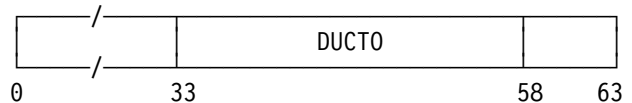
アクセス・レジスタ変換制御

アクセス・レジスタ変換は、アドレス・スペース制御、および制御レジスタ 2、5、および 8 に指定されている制御機能により、制御されます。アドレス・スペース制御 (PSW ビット 16 および 17) については、3-27ページの『変換モード』で説明されています。以下の項ではその他の制御機能について説明します。

その他の制御は、アクセス・レジスタ変換テーブルの中にあります。

制御レジスタ 2

制御レジスタ 2 には、ディスパッチ可能単位制御テーブルの位置が示されています。このレジスタのフォーマットは以下のとおりです。



ディスパッチ可能単位制御テーブル起点 (DUCTO):

制御レジスタ 2 のビット 33-57 の右側に 6 個の 0 を付加したものが、ディスパッチ可能単位制御テーブルの始めを示す 31 ビットの実アドレスを形成します。アクセス・レジスタ変換では、このディスパッチ可能単位制御テーブルから、ディスパッチ可能単位アクセス・リスト指定を取得することができます。

制御レジスタ 5

制御レジスタ 5 には、1 次 ASN 第 2 テーブル・エントリーの位置が指定されます。このレジスタのフォーマットは以下のとおりです。

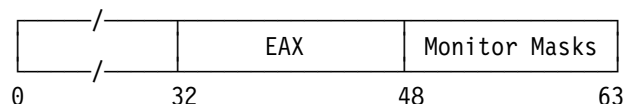


1 次 ASTE 起点 (PASTE0): 制御レジスタ 5 のビット 33-57 の右側に 6 個の 0 を付加したものが、1 次 ASN 第 2 テーブル・エントリーの始めを示す 31 ビットの実アドレスを形成します。アクセス・レジスタ変換では、1 次 ASTE から 1 次スペース・アクセス・リスト指定を取得することができます。1 次 ASTE 起点は、LOAD ADDRESS SPACE PARAMETERS が PASN 変換を行うときに設定されるほか、スペース切り

替えを伴う PROGRAM CALL、PROGRAM RETURN、および PROGRAM TRANSFER によっても設定されます。これらの命令はどれも、制御レジスター 5 に 1 次 ASTE 起点を入れると同時に、このレジスターのビット位置 32 および 58-63 に 0 を入れ、ビット 0-31 は変更しません。制御レジスター 5 のビット 0-32 および 58-63 は将来の割り当てに備えて確保されているものであり、これらのビットは 0 であってもなくても結果は変わりません。

制御レジスター 8

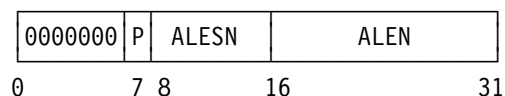
制御レジスター 8 には、拡張許可指標が入っています。このレジスターのフォーマットは以下のとおりです。



拡張許可指標 (EAX): 制御レジスター 8 のビット 32-47 は、拡張許可指標です。アクセス・レジスター変換では、EAX をアクセス・リスト・エン트리内のアクセス・リスト・エン트리許可指標 (ALEAX) と比較し、また、権限テーブル内の 2 次ビットを見つけるための指標として EAX を使用することができます。EAX はスタッキング PROGRAM CALL 操作により設定され、PROGRAM RETURN より格納されます。また、EAX は、特権命令 EXTRACT AND SET EXTENDED AUTHORITY により保管および設定することもできます。

アクセス・レジスター

32 ビットのアクセス・レジスターが 16 個あり、0-15 の番号が付けられています。アクセス・レジスターの内容を、アクセス・リスト・エン트리・トークン (ALET) と呼びます。ALET のフォーマットは以下のとおりです。



ALET 内のフィールドは以下のようにフィールドが割り当てられます。

1 次リスト・ビット (P): ALET 16 進数 00000000 または 00000001 のいずれでもないときは、ビット 7 は、アクセス・レジスター変換に使用するアクセス・リストを指定します。ビット 7 が 0 のときは、ディスパッチ可能単位アクセス・リストが使用されます。このリストは、制御レジスター 2 の内容が指し示すディスパッチ可能単位制御テーブル内の、ディスパッチ可能単位ア

クセス・リスト指定により指定されます。ビット 7 が 1 のときは、1 次スペース・アクセス・リストが使用されます。このリストは、制御レジスター 5 の内容が指し示す 1 次 ASTE の中の、1 次スペース・アクセス・リスト指定により指定されます。

アクセス・リスト・エン트리・シーケンス番号

(ALESN): ビット 8-15 は、ALET が取得された後で、ALET が指し示すアクセス・リスト・エントリーが無効化され、再割り振りされているかどうかを検査するために使用できます。アクセス・レジスター変換の際に、ALET が 16 進数 00000000 または 00000001 のいずれでもないときは、ALET のビット 8-15 が、指定されているアクセス・リスト・エントリーの中の、アクセス・リスト・エン트리・シーケンス番号 (ALESN) と比較されます。

アクセス・リスト・エン트리番号 (ALEN): ALET が 16 進数 00000000 または 00000001 のいずれでもないときは、ALET のビット 16-31 は、ビット 7 の値に応じて、ディスパッチ可能単位アクセス・リストまたは 1 次スペース・アクセス・リストの中のエントリーを示します。使用されるアクセス・リスト指定を、有効アクセス・リスト指定と呼びます。これは、有効アクセス・リスト起点と有効アクセス・リスト長から成っています。

アクセス・レジスター変換の際には、ALEN の右側に 4 個の 0 を付加したものが、有効アクセス・リスト起点に指定されている 31 ビットの実アドレスまたは絶対アドレスに加算され、その結果が、指定されたアクセス・リスト・エントリーの実アドレスまたは絶対アドレスとなります。ALEN と有効アクセス・リスト長を比較することにより、指定されたアクセス・リスト・エントリーがリスト内にあるかどうかを判別され、そのエントリーがリストの外部にある場合は、ALEN 変換例外が認識されます。ALEN の可能な最大値は 65,535 ですが、1 つのアクセス・リストに含めることができるエントリーの最大数は 1,024 エントリーです。

アクセス・レジスター変換時には、ビット 0-6 は 0 でなければならず、そうでない場合は ALET 指定例外が認識されます。

ALET が 16 進数 00000000 または 00000001 のときは、それぞれ 1 次または 2 次アドレス・スペースを示し、上に示したフォーマットは適用されません。

アクセス・レジスター変換では、通常、アクセス・レジスター 0 は 16 進数 00000000 を含むものと見なされ、実際の内容は検査されません。唯一の例外として、

TEST ACCESS の一環として行われるアクセス・レジスタ変換の場合は、この検査が行われます。また、アクセス・レジスタ 0 は、PSW ビット 16 および 17 が 2 進数 01 のときに、LOAD ADDRESS EXTENDED の B フィールドにより指定されている場合も、16 進数 00000000 を含むものと見なされます。アクセス・レジスタ 0 が、TEST ACCESS に指定されている場合、または COPY ACCESS、EXTRACT ACCESS、または STORE ACCESS MULTIPLE のソースとして指定されている場合は、このアクセス・レジスタの実際の内容が使用されます。アクセス・レジスタ 0 は、他のアクセス・レジスタと同様に、COPY ACCESS、LOAD ACCESS MULTIPLE、LOAD ADDRESS EXTENDED、および SET ACCESS によりロードできます。

ALET の 16 進数が 00000000 または 00000001 である場合のその他の定義については、10-12ページの『BRANCH IN SUBSPACE GROUP』で説明されています。

アクセス・レジスタ変換テーブル

変換される ALET が 16 進数 00000000 または 00000001 のどちらでもないときは、アクセス・レジスタ変換では、2 段階の検索により、まず有効アクセス・リスト指定、次に有効アクセス・リスト内のエントリが見つけ出されます。有効アクセス・リスト指定は、実記憶域の中にあります。有効アクセス・リストは、実記憶域または絶対記憶域の中にあります。

アクセス・レジスタ変換では、アクセス・リスト・エントリ内の起点を使用して、ASN 第 2 テーブル・エントリが見つけ出されます。また、1 段階の検索により、権限テーブル内のエントリが見つけ出されることもあります。ASN 第 2 テーブル・エントリは、実記憶域の中にあります。権限テーブルは、実記憶域または絶対記憶域の中にあります。

権限テーブルのエントリについては、3-23ページの『権限テーブル・エントリ』で説明されています。以下のセクションでは、アクセス・リスト指定、アクセス・リスト・エントリ、および ASN 第 2 テーブル・エントリについて説明します。

ディスパッチ可能単位制御テーブルおよびアクセス・リスト指定

変換される ALET が 16 進数 00000000 または 00000001 のどちらでもない場合、アクセス・レジスタ変換では、ALET のビット 7 が 0 であればディスパッチ可能単位アクセス・リスト指定が取得され、ビット 7 が 1 であれば 1 次スペース・アクセス・リスト指定が取得されます。ここで取得されたアクセス・リスト指定を、有効アクセス・リスト指定と呼びます。

ディスパッチ可能単位アクセス・リスト指定 (DUALD) は、ディスパッチ可能単位制御テーブル (DUCT) と呼ばれる 64 バイト区域のバイト 16-19 にあります。DUCT は実記憶域の中にあり、その位置は制御レジスタ 2 の中の DUCT 起点により指定されます。

ディスパッチ可能単位制御テーブルのフォーマットは、以下のとおりです。

16 進数 10 進数

0	0	BASTE0			
4	4	S	SSASTE0		
8	8				
C	12	SSASTESN			
10	16	DUALD			
14	20	PSW-Key Mask	PSW Key	R A	P
18	24				
1C	28	////////////////////			

24 ビットまたは 31 ビット・アドレッシング・モードのとき

20	32				
24	36	B	Bits 33-63 of Return Address		
		A			

64 ビット・アドレッシング・モードのとき

20	32	Bits 0-31 of Return Address			
24	36	Bits 33-63 of Return Address			

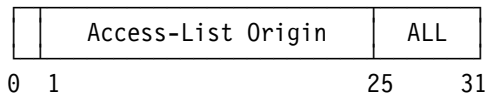
28	40			
2C	44	Trap-Control-Block Address		E
30	48			
3C	60			

DUCT のバイト 0-7 (BASTE0、SA、および SSASTE0) および 12-15 (SSASTESN) については、5-55ページの『サブスペース・グループ・ディスパッチ可能単位制御テーブル』で説明します。バイト 20-23 (PSW キー・マスク、PSW キー、RA、および P) および 32-39 (BA および戻りアドレス) については、10-5ページの『BRANCH AND SET AUTHORITY』で説明します。バイト 44-47 (トラップ制御ブロック・アドレスおよび E) については、10-114ページの『TRAP』で説明します。バイト 8-11、24-27、40-43、および 48-63 は、将来に拡張に備えて予約されているものであり、これらのバイトはすべて 0 でなければなりません。バイト 28-31 はプログラミング用に使用できます。

1 次スペース・アクセス・リスト指定 (PSALD) は、1 次 ASN 第 2 テーブル・エン트리と呼ばれる 64 バイト区域のバイト 16-19 にあります。1 次 ASTE は実記憶域内にあり、その位置は制御レジスタ 5 の中の 1 次 ASTE 起点により指定されます。1 次 ASTE のフォーマットについては、5-46ページの『ASN 第 2 テーブル・エン트리』で説明します。

ディスパッチ可能単位アクセス・リスト指定と 1 次スペース・アクセス・リスト指定は、どちらも同じフォーマットです。

アクセス・リスト指定



アクセス・リスト指定のフィールドは、以下のように割り振られています。

アクセス・リスト起点: アクセス・リスト指定のビット 1-24 の右側に 7 個の 0 を付加したものが、アクセス・リストの始めを示す 31 ビットのアドレスを形成します。このアドレスは、実アドレスまたは絶対アドレスとして扱われますが、どちらになるかは予測不能です。

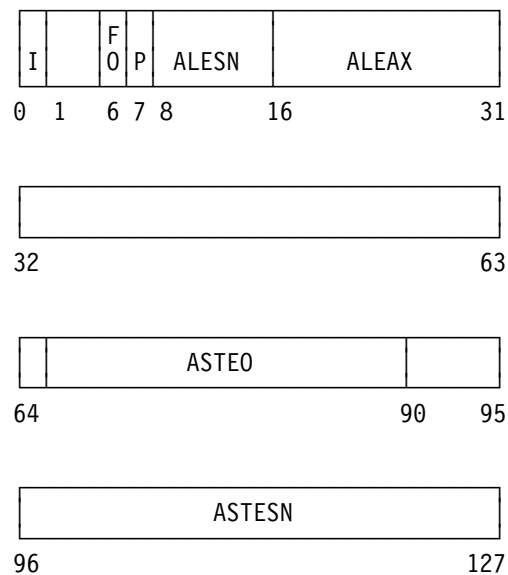
アクセス・リスト長 (ALL): アクセス・リスト指定のビット 25-31 は、アクセス・リストの長さを 128 バイト単位で指定します。したがって、アクセス・リストの長さは可変であり、8 個の 16 バイト・エントリーの倍数になります。アクセス・リストの長さ (128 バイト単位) は、ビット位置 25-31 の値に 1 を加えた値です。アクセス・リスト長の左側に 6 個の 0 を付加したものが、アクセス・リスト・エン트리番号のビット 0-12 (アクセス・リスト・エン트리・トークンのビット 16-28) と比較されて、そのアクセス・リスト・エン트리番号がアクセス・リスト内にあるエントリーを示しているかどうか判別されます。

ビット 0 は将来の拡張に備えて予約されており、値は 0 でなければなりません。

プログラミング上の注意: 1 つのアクセス・リスト指定で許されるアクセス・リスト・エントリーの最大数は、1,024 です。どの時点でも、2 つのアクセス・リストを使用できます。したがって、制御プログラムの介入を必要とせずに、最大 2,048 個の 16E バイト・アドレス・スペース (合計 2^{75} バイト) をアドレッシングすることができます。

アクセス・リスト・エン트리

有効アクセス・リストは、変換される ALET のビット 7 が 0 ならディスパッチ可能単位アクセス・リストであり、ビット 7 が 1 なら 1 次スペース・アクセス・リストです。有効アクセス・リストから取り出されるエントリーの長さは 16 バイトで、フォーマットは以下のとおりです。



アクセス・リスト・エントリーのフィールドは、次のように割り振られています。

ALEN 無効ビット (I): ビット 0 が 0 のときは、アクセス・リスト・エントリーがアドレス・スペースを指定していることを示します。アドレス・レジスター変換時にビット 0 が 1 であるときは、ALEN 変換例外が認識されます。

取り出し専用ビット (FO): ビット 6 は、アクセス・リスト・エントリーで指定されているアドレス・スペースに対してどのようなタイプのオペランド参照が許されるかを制御します。ビット 6 が 0 のときは、取り出しタイプおよび格納タイプの両方の参照が許されます。ビット 6 が 1 のときは、取り出しタイプの参照のみが許され、格納を行おうとすると、アクセス・リスト制御保護に対する保護例外が認識され、操作は抑止されます。

専用ビット (P): ビット 7 が 0 のときは、どのプログラムでも、アクセス・レジスター変換時にこのアクセス・リスト・エントリーの使用を許可されることを示します。ビット 7 が 1 のときは、許可はビット 16-31 の内容に従って決定されます。

アクセス・リスト・エントリー・シーケンス番号

(ALESN): ビット 8-15 は、アクセス・レジスター変換時に、ALET 中の ALESN と比較されます。両者が等しくないと、ALE シーケンス例外が認識されます。制御プログラムは、アクセス・リスト・エントリーを再割り振りするたびに、ビット 8-15 を変更するように定義されています。

アクセス・リスト・エントリー許可指標 (ALEAX):

ビット 16-31 は、アクセス・レジスター変換が行われているプログラムが、このアクセス・リスト・エントリーの使用を許可されているかどうかを判別するために使用できます。以下のいずれかの条件が満たされていれば、プログラムは許可されます。

1. ビット 7 が 0 である。
2. ビット 16-31 が、制御レジスター 8 中の拡張許可指標 (EAX) に等しい。
3. EAX が、指定されたアドレス・スペース用の権限テーブル内で、1 である 2 次ビットを選択している。

上記の条件が 1 つも満たされていない場合は、例外が認識されます。

ASN 第 2 テーブル・エントリー起点 (ASTE0): ビット 65-89 の右側に 6 個の 0 を付加したものが、指定されたアドレス・スペースの ASTE の 31 ビットの実

アドレスを形成します。アクセス・レジスター変換では、ASTE からそのアドレス・スペース用のアドレス・スペース制御エレメントが取得されます。

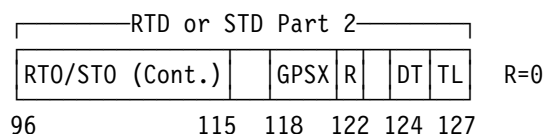
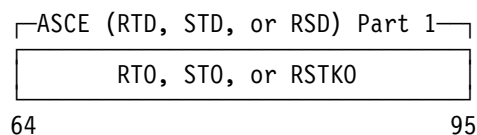
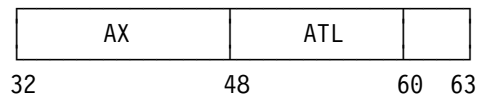
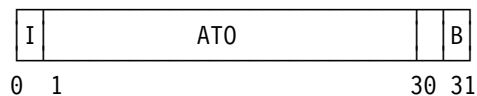
ASTE シーケンス番号 (ASTESN): ビット 96-127 は、アクセス・リスト・エントリーが表しているアドレス・レジスタリング能力を取り消すために使用できます。ビット 96-127 は、アクセス・レジスター変換の際に、指定された ASTE 中の ASTE シーケンス番号 (ASTESN) と比較されます。

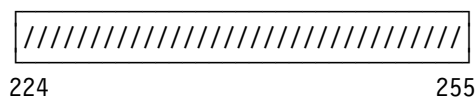
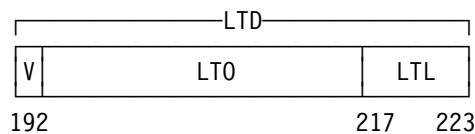
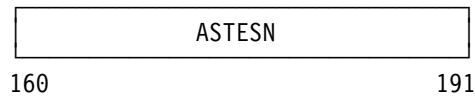
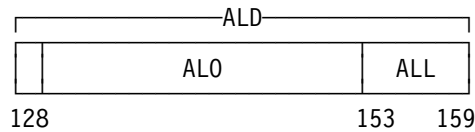
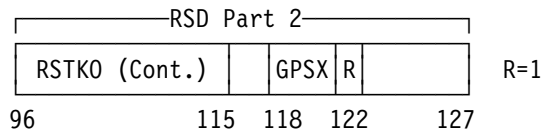
ビット 1-5、32-64、および 90-95 は将来の拡張に備えて予約されており、いずれも 0 でなければなりません。

ディスパッチ可能単位アクセス・リストおよび 1 次スペース・アクセス・リストのどちらにおいても、アクセス・リスト・エントリー 0 および 1 はアクセス・レジスター変換には使用しないものとされています。アクセス・リスト・エントリー 0 のビット 1-127 およびアクセス・リスト・エントリー 1 のビット 1-63 は、将来の拡張に備えて予約されており、いずれも 0 でなければなりません。アクセス・リスト・エントリー 0 のビット 0、およびアクセス・リスト・エントリー 1 のビット 64-127 は、プログラミング用に使用できます。制御プログラムは、ALEN が 0 または 1 である ALET によりアクセス・リスト・エントリー 0 および 1 が使用されないようにするために、これらのエントリーのビット 0 を 1 に設定する必要があります。

ASN 第 2 テーブル・エントリー

64 バイトの ASN 第 2 テーブル・エントリーの最初の 32 バイトのフォーマットは、以下のとおりです。





ASN 第 2 テーブル・エンタリー (ASTE) のバイト 0-31 にある各フィールドは、3-18ページの『ASN 第 2 テーブル・エンタリー』の中で、特定のメカニズムおよび命令との関連で定義されています。ASTE の中の各フィールドは、5-57ページの『サブスペース・グループ ASN 第 2 テーブル・エンタリー』の中で、BRANCH IN SUBSPACE GROUP 命令との関連で定義されています。アクセス・レジスタ変換のみについて、および BRANCH IN SUBSPACE GROUP 以外の命令については、ASTE のフィールドは以下のように割り振られています。

ASX 無効ビット (I): ビット 0 は、ASTE に関連したアドレス・スペースが使用可能かどうかを制御します。ビット 0 が 0 のときは、アクセス・レジスタ変換が進められます。このビットが 1 のときは、ASTE 妥当性例外が認識されます。

権限テーブル起点 (ATO): ビット 1-29 の右側に 2 個の 0 を付加したものが、権限テーブルの始めを示す 31 ビット・アドレスを形成します。このアドレスは、実アドレスまたは絶対アドレスとして扱われますが、どちらになるかは予測不能です。ただし、ASN 許可用としては、このアドレスは実アドレスとして扱われます。アクセス・レジスタ変換の際に権限テーブルがアクセスされるのは、アクセス・リスト・エンタリーの専用ビット

が 1 で、アクセス・リスト・エンタリー内のアクセス・リスト・エンタリー許可指標 (ALEAX) が、制御レジスタ 8 の中の拡張許可指標 (EAX) に等しくない場合のみです。

基本スペース・ビット (B): アクセス・レジスタ変換では、ビット 31 は無視されます。ビット 31 については、5-57ページの『サブスペース・グループ ASN 第 2 テーブル・エンタリー』でさらに詳しく説明します。

許可指標 (AX): ビット 32-47 は、アクセス・レジスタ変換では使用されません。

権限テーブル長 (ATL): ビット 48-59 は、権限テーブルの長さを 4 バイト単位で指定します。したがって、権限テーブルは可変長で、長さは 16 エンタリーの倍数になります。権限テーブルの長さ (4 バイト単位) は、ATL の値に 1 を加えた値になります。ATL フィールドの内容は、特定の EAX が示すエンタリーが権限テーブル内に含まれているかどうかを確定するために使用されます。このエンタリーがテーブル内にはない場合は、拡張権限例外が認識されます。

アドレス・スペース制御エレメント (ASCE): ビット 64-127 は、8 バイトのアドレス・スペース制御エレメント (ASCE) で、これはセグメント・テーブル指定 (STD)、領域テーブル指定 (RTD)、または実スペース指定 (RSD) のいずれかです。(「領域テーブル指定」という用語は、領域第 1 テーブル指定、領域第 2 テーブル指定、または領域第 3 テーブル指定を意味します。) ASCE フィールドは、アクセス・レジスタ変換の結果として取得され、現在行われている記憶域オペランド参照用の論理アドレスを変換するために、DAT で使用されます。ビット 121 (スペース切り替えイベント制御) は、アクセス・レジスタ変換では (またはその変換の結果としては) 使用されません。ASCE のその他のフィールド (RTO、STO、RSTKO、G、P、S、R、DT、および TL) については、3-27ページの『制御レジスタ 1』で説明します。

アクセス・リスト指定 (ALD): この ASTE が、制御レジスタ 5 の中の 1 次 ASTE 起点により指定されている場合は、ビット 128-159 は 1 次スペース・アクセス・リスト指定 (PSALD) です。5-44ページの『ディスパッチ可能単位制御テーブルおよびアクセス・リスト指定』の中のアクセス・リスト指定に関する説明を参照してください。アクセス・レジスタ変換時に、ALET の中の 1 次リスト・ビット (ビット 7) が 1 であれば、PSALD は有効アクセス・リスト指定です。

ASN 第 2 テーブル・エンタリー・シーケンス番号

(ASTESN): ビット 160-191 は、この ASTE を指定しているアクセス・リスト・エンタリーが表すアクセス能力の取り消しを制御するために使用されます。アクセス・レジスター変換の際に、ビット 160-191 はアクセス・リスト・エンタリー内の ASTESN と比較され、両者が等しくないときは ASTE シーケンス例外が認識されます。制御プログラムは、ASTE で指定されているアドレス・スペース用の許可ポリシーが変化したとき、または ASTE が再割り当てされて別のアドレス・スペースを指定するようになったときに、ビット 160-191 を変更するものとされています。

リンケージ・テーブル指定 (LTD): ビット 192-223 は、アクセス・レジスター変換では使用されません。

ASTE のビット 224-255 は、プログラミング用に使用できます。

プログラミング上の注意: バイト 0-31 の未使用フィールドおよびバイト 32-63 のすべてを含めて、ASTE の未使用フィールドは、すべて 0 に設定する必要があります。これらのフィールドは将来の拡張のための予約されており、これらのフィールドに非ゼロ値を入れるプログラムは、将来のマシンでは互換性を保って動作することができなくなるおそれがあります。

アクセス・レジスター変換プロセス

このセクションでは、アクセス・レジスター・モードでの記憶域オプション参照時に行われるアクセス・レジスター変換プロセスについて説明します。PSW ビット 16 および 17 が 2 進数 01 であるときの LOAD REAL ADDRESS と STORE REAL ADDRESS、任意の変換モードでの TEST ACCESS、およびアクセス・レジスター・モードでの TEST PROTECTION は、以下に説明するおりのアクセス・レジスター変換を行います。ただし、LOAD REAL ADDRESS、TEST ACCESS、および TEST PROTECTION では、以下の例外が起きた場合に、プログラム割り込み条件として扱われる代わりに、条件コードが設定されます。

- ALET 指定例外
- ALEN 変換例外
- ALE シーケンス例外
- ASTE 妥当性例外
- ASTE シーケンス例外
- 拡張権限例外

BRANCH IN SUBSPACE GROUP は、10-12ページの『BRANCH IN SUBSPACE GROUP』で説明されているとおりにアクセス・レジスター変換を行います。

アクセス・レジスター変換は、DAT が使用するアドレス・スペース制御エレメントを取得するために、記憶域オペランド参照で指定されているアクセス・レジスターを対象として行われます。アクセス・レジスター 1-15 の1つが指定されている場合、そのアクセス・レジスター内にあるアクセス・リスト・エンタリー・トークン (ALET) を使用して、アドレス・スペース制御エレメントが取得されます。アクセス・レジスター 0 が指定されているときは、16 進数値 00000000 を持つ ALET が使用されます。ただし、TEST ACCESS では、アクセス・レジスター 0 の実際の内容が使用されます。

ALET が 16 進数 00000000 または 00000001 のときは、それぞれ、1 次または 2 次アドレス・スペース制御エレメントが取得されます。

ALET が 16 進数 00000000 または 00000001 のいずれでもないときは、ALET の左端 7 ビットが 0 かどうかを検査され、ALET の 1 次リスト・ビットおよび制御レジスター 2 または 5 の内容を使用して有効アクセス・リスト指定が取得され、ALET の中のアクセス・リスト・エンタリー番号 (ALEN) を使用して有効アクセス・リスト内のエンタリーが選択されます。

選択されたアクセス・リスト・エンタリーが妥当であるかどうか、および、正しいアクセス・リスト・エンタリー・シーケンス番号 (ALESN) を含んでいるかどうかを検査されます。

このアクセス・リスト・エンタリーでアドレッシングされている ASN 第 2 テーブル・エンタリー (ASTE) が妥当であるかどうか、および正しい ASN 第 2 テーブル・エンタリー・シーケンス番号 (ASTESN) を含んでいるかどうかを検査されます。

プログラムがこのアクセス・リスト・エンタリーの使用を許可されるかどうかは、次の 1 つまたは複数の要素に基づいて決定されます。(1) アクセス・リスト・エンタリーの中の専用ビットおよびアクセス・リスト・エンタリー許可指標 (ALEAX)、(2) 制御レジスター 8 の中の拡張許可指標 (EAX)、および、(3) ASN 第 2 テーブル・エンタリーによりアドレッシングされる権限テーブル内のエンタリー。

格納タイプの参照を行う必要がある場合は、アクセス・リスト・エンタリー内の取り出し専用ビットが 0 かどうかを検査されます。

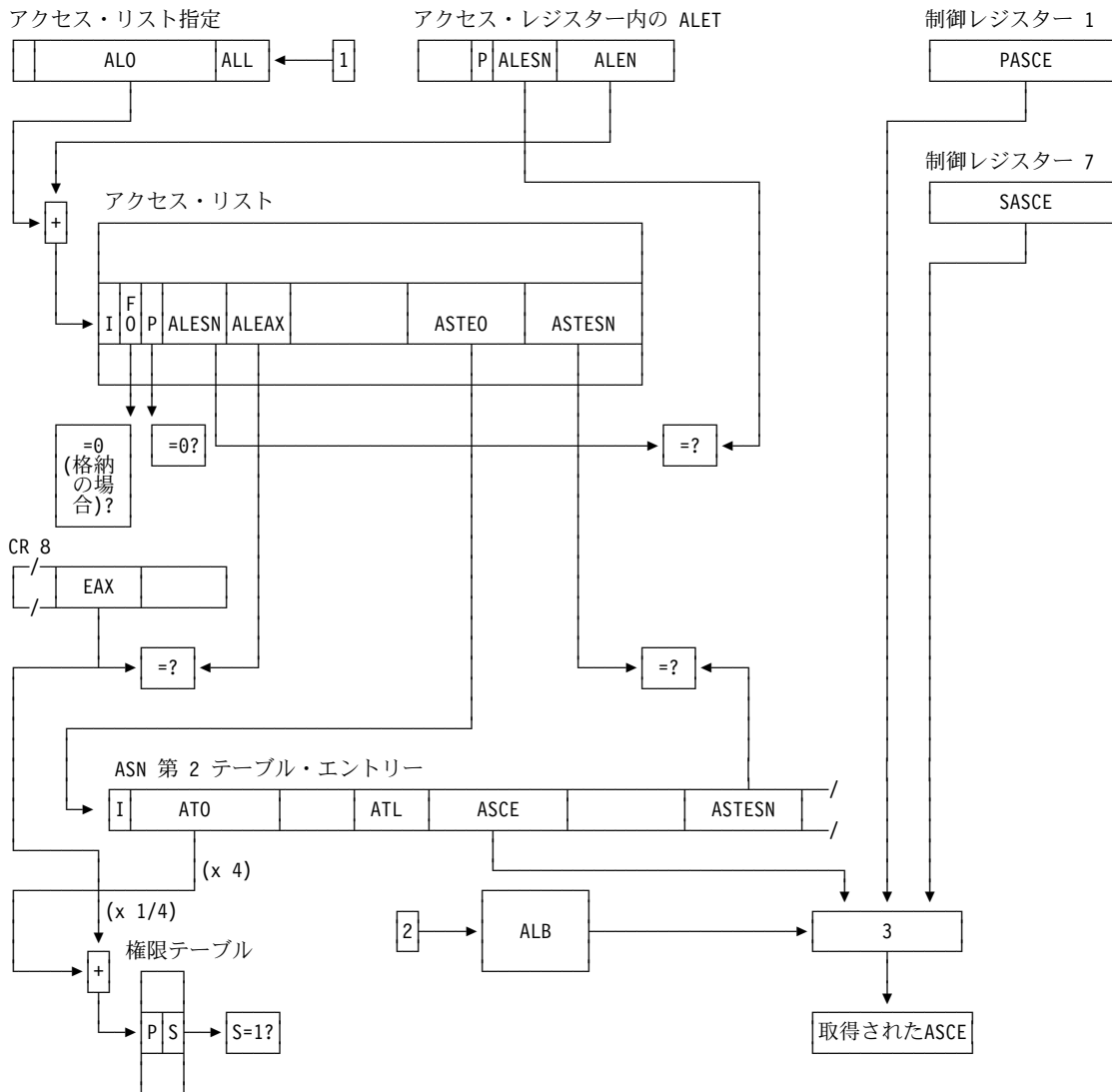
例外が 1 つも認識されなければ、ASN 第 2 テーブル・エントリー内のアドレス・スペース制御エレメントが取得されます。

実記憶域または絶対記憶域への参照に伴う遅延を避けるために、実記憶域または絶対記憶域から取り出された情報は、通常、ART 索引バッファ (ALB) と呼ばれる特殊バッファにも入れられます。したがって、同じ情報に関連する後続の変換は、ALB の内容を使用して実行

できます。ALB の働きについては、5-53ページの『ART 索引バッファ』で説明します。

アクセス・レジスタ変換の際に、アクセス・リスト指定ソース、アクセス・リスト、ASN 第 2 テーブル、または権限テーブルからエントリーを取り出すために、実記憶域または絶対記憶域にアクセスするときには、キー制御保護は適用されません。

5-50ページの図5-10 は、ALB の効果も含めて、アクセス・レジスタ変換の主な機能を示しています。



説明:

- 1 該当する ALD が次のように取得されます。
 ALET 中の P が 0 であれば (そして ALET が 0 または 1 でなければ)、DUCT 中の DUALD が取得されます。ALET 中の P が 1 であれば、1 次 ASTE 中の PSALD が取得されます。
- 2 ALD ソース起点、ALET、ALO、および EAX を含む情報を使用して、ALB がサーチされます。この情報は、ALE、ASTE、および ATE からの情報と一緒に ALB に入られます。
- 3 該当する ASCE が次のように取得されます。
 ALET が 0 のときは、CR 1 中の PASCE が取得されます。
 ALET が 1 のときは、CR 7 中の SASCE が取得されます。
 ALET が 1 より大きいときは、次のようになります。
 一致するものがあれば、ALB からの ASCE が使用されます。
 一致するものがない場合は、実記憶域または絶対記憶域からテーブルが取り出されます。
 ASTE から結果の ASCE が取得され、ALB 内にエントリーが形成されます。

図 5-10. アクセス・レジスタ変換

アクセス・リスト・エントリー・トークンの選択

アクセス・レジスタ 1-15 のどれか 1 つが指定されている場合、または、TEST ACCESS の R1 フィールドで指定されたアクセス・レジスタの場合は、アクセス・レジスタ変換では、そのアクセス・レジスタ内にあるアクセス・リスト・エントリー・トークン

(ALET) が使用されます。アクセス・レジスタ 0 が指定されているときは、TEST ACCESS の場合を除き、16 進数値 00000000 を持つ ALET が使用され、アクセス・レジスタ 0 の内容は検査されません。

1 次または 2 次アドレス・スペース制御エレメントの取得

変換されている ALET が 16 進数 00000000 であるときは、制御レジスタ 1 中の 1 次アドレス・スペース制御エレメントが取得されます。ALET が 16 進数 00000001 であるときは、制御レジスタ 7 中の 2 次アドレス・スペース制御エレメントが取得されます。上記 2 つのいずれの場合も、アクセス・レジスタ変換は完了します。

ALET の第 1 バイトの検査

変換されている ALET が 16 進数 00000000 または 00000001 のいずれでもない場合は、ALET のビット 0-6 がすべて 0 かどうかを検査されます。ビット 0-6 がすべて 0 ではない場合は、ALET 指定例外が認識され、操作は抑止されます。

有効アクセス・リスト指定の取得

ALET の 1 次リスト・ビット (ビット 7) は、有効アクセス・リスト指定を入手するための検索に使用されます。ビット 7 が 0 のときは、有効 ALD は、ディスパッチ可能単位制御テーブル (DUCT) のバイト 16-19 に示されているディスパッチ可能単位 ALD です。ビット 7 が 1 のときは、有効 ALD は、1 次 ASN 第 2 テーブル・エン트리 (1 次 ASTE) のバイト 16-19 に示されている 1 次スペース ALD です。

ビット 7 が 0 のときは、DUCT 起点 (制御レジスタ 2 のビット 33-57) の右側に 6 個の 0 を付加し、それに 16 を加算することにより、ディスパッチ可能 ALD の 31 ビット実アドレスが求められます。この加算により、ビット位置 0 への繰り上がりが生じることはありません。

ビット 7 が 1 のときは、1 次 ASTE 起点 (制御レジスタ 5 のビット 33-57) の右側に 6 個の 0 を付加し、それに 16 を加算することにより、1 次スペース ALD の 31 ビット実アドレスが求められます。この加算により、ビット位置 0 への繰り上がりが生じることはありません。

取得された 31 ビット実アドレスを使用して、有効 ALD が取り出されます。これは、ALET のビット 7 に応じて、ディスパッチ可能単位 ALD または 1 次スペース ALD のいずれかになります。他の CPU からは、有効 ALD の取り出しはワード単位のように見え、またこの操作は保護の対象にはなりません。有効 ALD を取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

シング例外が認識され、操作は抑止されます。1 次スペース ALD が取り出されるときは、1 次 ASTE のビット 0 (ASX 無効ビット) は無視されます。

アクセス・リストの検索

有効アクセス・リスト内で検索が行われます。有効アクセス・リストは、ALET のビット 7 が 0 ならディスパッチ可能単位アクセス・リストであり、ビット 7 が 1 なら 1 次スペース・アクセス・リストです。有効アクセス・リストは、実記憶域または絶対記憶域の中にあるものと見なされますが、どちらになるかは予測不能です。

ALET のアクセス・リスト・エン트리番号 (ALEN) 部分を使用して、有効アクセス・リスト内のエントリーが選択されます。有効 ALD のビット 1-24 の右側に 7 個の 0 を付加し、それに、右側に 4 個、左側に 11 個の 0 を付加した ALEN を加算することにより、アクセス・リスト・エントリーの 31 ビットの実アドレスまたは絶対アドレスが求められます。この加算によりビット位置 0 への繰り上がりが生じる場合は、アドレッシング例外が認識されるか、または、繰り上がりが無視されて、アクセス・リストは $2^{31} - 1$ から 0 へと循環します。現 PSW が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードを指定しているかに関係なく、この 31 ビット・アドレスが形成され、使用されます。

アクセス・リスト検索プロセスの一環として、ALEN の左端 13 ビットが有効アクセス・リスト長 (有効 ALD のビット 25-31) と比較することにより、アドレッシングされたエントリーがアクセス・リスト内にあるかどうか判定されます。この比較では、アクセス・リスト長は、左側に 6 個の 0 を付けて拡張されます。アクセス・リスト長から形成された値が ALEN の左端 13 ビットの値より小さい場合は、ALEN 変換例外が認識され、操作は無効化されます。

上記の実アドレスまたは絶対アドレスを使用して、16 バイトのアクセス・リスト・エントリーが取り出されます。他の CPU からは、このエントリーの取り出しはワード単位のように見え、最左端のワードが最初に取り出されます。残りの 3 ワードが取り出される順序は、予測不能です。この取り出しアクセスは記憶保護の対象にはなりません。アクセス・リスト・エントリーを取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

アクセス・リスト・エントリーのビット 0 は、アクセス・リスト・エントリーが、ASN 第 2 テーブル・エントリーを指示することによりアドレス・スペースを指定

しているかどうかを示します。このビットが検査され、その値が 1 であれば、ALEN 変換例外が認識され、操作は無効化されます。

ビット 0 が 0 のときは、アクセス・リスト・エントリーのビット位置 8-15 にあるアクセス・リスト・エントリー・シーケンス番号 (ALESN) を ALET 内の ALESN と比較することにより、ALET が概念的に正しいアクセス・リスト・エントリーを指示しているかどうかを判別されます。両者が等しくないときは、ALE シーケンス例外が認識され、操作は無効化されます。

ASN 第 2 テーブル・エントリーの検索

ASN 第 2 テーブル・エントリー (ASTE) を検索するには、アクセス・リスト・エントリー内の ASTE 起点が使用されます。アクセス・リスト・エントリーのビット 65-89 の右側に 6 個の 0 を付加したものが、ASTE の 31 ビット実アドレスを形成します。

その実アドレスを使用して、64 バイトの ASTE が取り出されます。他の CPU からは、このエントリーの取り出しはワード単位のように見え、最左端のワードが最初に取り出されます。残りのワードが取り出される順序は、予測不能です。この取り出しアクセスは記憶保護の対象にはなりません。ASTE を取り出すために生成された記憶域アドレスが、構成内で使用可能になっていない位置を示している場合は、アドレッシング例外が認識され、操作は抑止されます。

ASTE のビット 0 は、ASTE がアドレス・スペースを指定しているかどうかを示します。このビットが検査され、その値が 1 であれば、ASTE 妥当性例外が認識され、操作は無効化されます。

ビット 0 が 0 のときは、ASTE のビット位置 160-191 にある ASTE シーケンス番号 (ASTESN) を、アクセス・リスト・エントリーのビット位置 96-127 にある ASTESN と比較することにより、このアクセス・リスト・エントリーが表しているアドレッシング能力が取り消されているかどうかを判別されます。両者が等しくないときは、ASTE シーケンス例外が認識され、操作は無効化されます。

アクセス・リスト・エントリーの使用許可

アクセス・リスト・エントリーの専用ビット (ビット 7) は、プログラムがこのアクセス・リスト・エントリーの使用を許可されているかどうかを判別するために使用されます。ほかに、アクセス・リスト・エントリーのビット位置 16-31 にあるアクセス・リスト・エントリー許可指標 (ALEAX)、制御レジスター 8 のビット位置 32-47

にある拡張許可指標 (EAX)、および ASTE に示されている権限テーブルが使用されることもあります。

専用ビットが 0 のときは、プログラムは許可されており、アクセス・レジスター変換の許可ステップは完了します。

専用ビットが 1 であっても、ALEAX が EAX に等しければ、プログラムは許可されており、アクセス・レジスター変換の許可ステップは完了します。

専用ビットが 1 で、かつ ALEAX が EAX に等しくない場合は、拡張許可プロセスと呼ばれるプロセスが実行されます。拡張許可では、EAX を使用して、ASTE が示している権限テーブル内のエントリーが選択され、その選択されたエントリー内の 2 次権限ビットが 1 であるかどうかをテストされます。そして、そのテスト対象ビットが 1 であれば、プログラムは許可されます。

拡張許可は、以下の点を除いて、3-22ページの『ASN 許可』で説明した 2 次 ASN 許可プロセスと同じです。

- 権限テーブル起点は、実アドレスとしてではなく、実アドレスまたは絶対アドレスとして扱われます。
- 制御レジスター 4 の許可指標 (AX) の代わりに、制御レジスター 8 の EAX が使用されます。
- EAX のビット位置 0-11 の値が ASTE 内の権限テーブル長 (ATL) より大きい場合、2 次権限例外ではなく拡張権限例外が認識されます。拡張権限例外が認識された場合は、操作は無効化されます。

専用ビットが 1 であり、ALEAX が EAX に等しくなく、しかも、EAX により選択された権限テーブル・エントリー内の 2 次ビットが 1 ではないときは、拡張権限例外が認識され、操作は無効化されます。

アクセス・リスト制御保護の検査

格納タイプの参照を行うことになっているときに、アクセス・リスト・エントリー内の取り出し専用ビット (ビット 6) が 1 である場合は、記憶保護例外が認識され、操作は抑止されます。

2 次 ASN 第 2 テーブル・エントリーからのアドレス・スペース制御エレメントの取得

変換されている ALET が 16 進数 00000000 または 00000001 のどちらでもなく、上記の一連のステップで何も例外が認識されなかった場合は、アクセス・レジスター変換では、ASTE のビット位置 64-127 からアドレス・スペース制御エレメントが取得されます。

アクセス・レジスタ変換時の例外の認識

アクセス・レジスタ変換プロセス中に検出される可能性がある例外と、それぞれの優先順位は、第6章、『割り込み』の中の『アクセス例外』のセクションに示してあります。

プログラミング上の注意: アクセス・リスト・エントリまたは ASN 第 2 テーブル・エントリを更新する際には、プログラムは、更新の最後のステップとして、そのエントリを無効から有効に変更する (エントリのビット 0 を 0 に設定する) 必要があります。左端のワードが最初に取り出されるので、これにより、部分的に更新されたエントリのワードは取り出されないようになります。

ART 索引バッファ

パフォーマンスを高めるために、通常、アクセス・レジスタ変換 (ART) メカニズムでは、アクセス・リスト、ASN 第 2 テーブル、および権限テーブル内に指定されているアクセス・リスト指定および情報を、ART 索引バッファ (ALB) と呼ばれる特殊バッファ内に保持する方法を利用しています。アクセス・リスト指定、アクセス・リスト・エントリ、ASN 第 2 テーブル・エントリ、および権限テーブル・エントリを、総称的に ART テーブル・エントリと呼びます。これにより、CPU が実記憶域または絶対記憶域内の ART テーブル・エントリを参照する必要があるのは、そのエントリへの最初のアクセスのときだけですみます。このエントリ内の情報が ALB の中に保持されていれば、以後の ART 操作は ALB を使用して行うことができます。ALB の存在が ART プロセスに与える影響としては、(1) 実記憶域または絶対記憶域内で ART テーブルが変更されたときに、変換へのその影響は必ずしも即時には現れない、(2) ALB アクセス・リスト・エントリを使用している場合は、記憶域内にあって変換に使用されるアクセス・リスト指定内のアクセス・リスト長との比較は省略できる、および、(3) ALB 権限テーブル・エントリを使用している場合は、記憶域内にあって変換に使用される ASN 第 2 テーブル・エントリ内の権限テーブル長との比較は省略できる、という点が挙げられます。複数 CPU 構成では、各 CPU について専用の ALB が 1 つずつあります。

ALB 内のエントリは、プログラムにより明示的にアドレスリングすることはできません。

ALB への情報の保持が可能なすべての条件が満たされていても、必ず ALB に情報が保持されるとは限りませ

ん。さらに、ALB 内の情報は、消去が必須となる条件に加えて、他の条件下でも消去されることがあります。

ALB の構造

ここでは、z/Architecture の定義に従って稼働するすべてのシステムのインプリメンテーションを対象として、ALB の論理構造について説明します。ALB エントリには 4 つのタイプがあると考えられます。それは、ALB アクセス・リスト指定 (ALB ALD)、ALB アクセス・リスト・エントリ (ALB ALE)、ALB ASN 第 2 テーブル・エントリ (ALB ASTE)、および ALB 権限テーブル・エントリ (ALB ATE) です。1 つの ALB エントリは、実記憶域または絶対記憶域内の ART テーブル・エントリから得られる情報と、ART テーブル・エントリを実記憶域または絶対記憶域から取り出すために使用する属性の両方を含むものと見なされます。ALB ALD には、実記憶域内の ALD を選択するために使用された ALD ソース起点が、ディスパッチ可能単位制御テーブル起点か 1 次 ASTE 起点かは、示されません。

注: 以下のセクションでは、どのような条件下で情報が ALB に入れられるか、どのような条件下で ALB からの情報がアクセス・レジスタ変換に使用されるか、そして、テーブルに対する変更が ART プロセスにどのような影響を与えるかについて説明します。

ALB エントリの形成

ALB エントリが形成されるかどうか、および、プログラムが実記憶域または絶対記憶域内の ART テーブル・エントリの内容を操作したときにどのような影響が生じるかは、そのエントリが特定の CPU に連結されているかどうか、およびそのエントリが有効かどうかによって異なります。

ART テーブル・エントリの連結された状態とは、そのエントリが連結されている CPU が、アクセス・レジスタ変換のためにそのエントリを使用できることを意味します。ART テーブル・エントリは、同時に複数の CPU に連結することもできます。

アクセス・リスト・エントリまたは ASN 第 2 テーブル・エントリが有効であるのは、そのエントリに関連した無効ビットが 0 のときです。アクセス・リスト指定および権限テーブル・エントリには無効ビットはなく、常に有効です。1 次スペース・アクセス・リスト指定は、1 次 ASTE の無効ビットの値に関係なく有効です。

ART テーブル・エントリは、連結されていて、かつ有効であれば、ALB に入れられる可能性があります。

アクセス・リスト指定が CPU に連結されるのは、その指定が、制御レジスター 2 中のディスパッチ可能単位制御テーブル起点が示すディスパッチ可能単位制御テーブル内にあるか、または、制御レジスター 5 中の 1 次 ASTE 起点が示す 1 次 ASTE 内にあるときです。

アクセス・リスト・エントリーが CPU に連結されるのは、そのエントリーが、連結されたアクセス・リスト指定 (ALD) または使用可能な ALB ALD の中にあるときです。使用可能な ALB ALD については、次のセクションで説明します。

ASN 第 2 テーブル・エントリーが CPU に連結されるのは、そのエントリーが、連結されている有効なアクセス・リスト・エントリー (ALE) または使用可能な ALB ALE 中の ASTE 起点により指定されているときです。使用可能な ALB ALE については、次のセクションで説明します。

権限テーブル・エントリーが CPU に連結されるのは、そのエントリーが、連結されている有効な ASN 第 2 テーブル・エントリー (ASTE) または使用可能な ALB ASTE が示している権限テーブル内にあるときです。使用可能な ALB ASTE については、次のセクションで説明します。

ALB エントリーの使用

ALB エントリーの使用可能な状態とは、CPU がアクセス・レジスター変換のためにその ALB エントリーを使用できることを意味します。使用可能な ALB エントリーは、次に低いレベルのテーブルがあればそれに連結し、アクセス・レジスター変換の特定インスタンス用に使用できます。

ALB ALD が使用可能な状態にあるのは、ALB ALD 中の ALDSO フィールドが、現行のディスパッチ可能単位制御テーブル起点または現行の 1 次 ASTE 起点に一致しているときです。

ALB ALD がアクセス・レジスター変換の特定インスタンスに使用されることがあるのは、以下のいずれかの条件が満たされているときです。

1. 変換する ALET 中の 1 次リスト・ビットが 0 で、ALB ALD 中の ALDSO フィールドが現行のディスパッチ可能単位制御テーブル起点に一致している。
2. 変換する ALET 中の 1 次リスト・ビットが 1 で、ALB ALD 中の ALDSO フィールドが現行の 1 次 ASTE 起点に一致している。

ALB ALE の ALO フィールドが、連結されている ALD または使用可能な ALB ALD の ALO フィールドに一致しているときに、ALB ALE が使用可能状態にある。

ALB ALE がアクセス・レジスター変換の特定インスタンスに使用されることがあるのは、以下のすべての条件が満たされているときです。

1. 変換する ALET の値が 1 より大きい。(ALET が 0 または 1 の場合は、CR 1 または CR 7 の内容が使用されます。)
2. ALB ALE の ALO フィールドが、変換に使用している ALD または ALB ALD の ALO フィールドに一致している。
3. ALB ALE の ALEN フィールドが、変換している ALET の ALEN フィールドに一致している。

ALB ASTE が使用可能な状態にあるのは、ALB ASTE の ASTEO フィールドが、連結されている有効な ALE または使用可能な ALB ALE の ASTEO フィールドに一致しているときです。

ALB ASTE がアクセス・レジスター変換の特定インスタンスに使用されることがあるのは、ALB ASTE の ASTEO フィールドが、変換に使用している ALE または ALB ALE の ASTEO フィールドに一致しているときです。

ALB ATE がアクセス・レジスター変換の特定インスタンスに使用されることがあるのは、以下の条件が両方とも満たされているときです。

1. ALB ATE の ATO フィールドが、変換に使用している ASTE または ALB ASTE の ATO フィールドに一致している。
2. ALB ATE の EAX フィールドが現 EAX に一致している。

ART テーブルの変更

連結されてはいるが無効な ART テーブル・エントリーが有効にされたとき、または連結されてはいるが無効な ART テーブル・エントリーが連結されたときに、その ART エントリーから形成されたエントリーがまだ ALB 内にはない場合は、その変更の効力は現命令の終わりまでの間に発生します。

連結された有効な ART テーブル・エントリーが変更された場合に、ALB からそのエントリーのコピーが消去される前に、そのエントリーを必要とする ART を実行しようとした場合は、以下に示す程度の範囲内で予測不

能の結果が生じます。つまり、新しい値の使用が開始されるのは、命令と命令の間のこともあり、また、変更の原因となった命令も含めて、いずれかの命令の実行中のこともあります。さらに、ALB からそのエントリーのコピーが消去されるまでは、ALB には新旧両方の値が含まれていることがあり、特定の ART 操作でどちらの値が選択されるかは予測不能です。新旧両方の値が有効スペース指定を表すものとして使用されている場合は、有効スペース指定が同じであることが認識されず、その結果、オペランドのオーバーラップが認識されないことがあります。有効スペース指定およびオペランド・オーバーラップについては、5-80ページの『単一命令内のインターロック』で説明します。

LOAD ACCESS MULTIPLE または LOAD

CONTROL が ART に関連したパラメーターを変更した場合は、この命令の操作が終了するまでは、操作の開始時点でのパラメーターの値が効力を維持します。

PURGE ALB、ALB をパージする COMPARE AND SWAP AND PURGE、または SET PREFIX 命令が実行されるか、または CPU リセットが行われると、ALB のすべてのエントリーが消去されます。

サブスペース・グループ

サブスペース・グループ機能には、BRANCH IN SUBSPACE GROUP 命令、および、アドレス・スペース制御エレメント、ディスパッチ可能単位制御テーブル、そして ASN 第 2 テーブル・エントリーの中のフィールドの割り振り、および、PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、LOAD ADDRESS SPACE PARAMETERS 命令のサブスペース置き換え操作があります。BRANCH IN SUBSPACE GROUP については、5-10ページの『リンケージ・スタックなしのサブルーチン・リンケージ』で概要を紹介し、10-12ページの『BRANCH IN SUBSPACE GROUP』で詳しく説明します。

サブスペース・グループ・テーブル

このセクションでは、サブスペース・グループ機能が、ディスパッチ可能単位制御テーブルおよび ASN 第 2 テーブル・エントリーをどのように使用するかについて説明します。

サブスペース・グループ・ディスパッチ可能単位制御テーブル

ディスパッチ可能単位制御テーブルのフォーマットは、以下のとおりです。

16 進数 10 進数

0	0	BASTE0			
4	4	S	SSASTE0		
8	8				
C	12	SSASTESN			
10	16	DUALD			
14	20	PSW-Key Mask	PSW Key	R A	P
18	24				
1C	28	////////////////////			

24 ビットまたは 31 ビット・アドレッシング・モードのとき

20	32				
24	36	B	Bits 33-63 of Return Address		
		A			

64 ビット・アドレッシング・モードのとき

20	32	Bits 0-31 of Return Address			
24	36	Bits 33-63 of Return Address			

28	40				
2C	44	Trap-Control- Block Address			E
30	48				
3C	60	/ /			

サブスペース・グループ機能が使用するディスパッチ可能単位制御テーブルのフィールドは、以下のように割り振られます。

基本 ASTE 起点 (BASTE0): バイト 0-3 のビット 1-25 の右側に 6 個の 0 を付加したものが、ディスパッチ可能単位に関連したサブスペース・グループの基本スペースを指定する ASN 第 2 テーブル・エントリーの始めを示す 31 ビット実アドレスを形成します。

BRANCH IN SUBSPACE GROUP は、バイト 0-3 のビット 1-25 を、制御レジスター 5 のビット位置 33-57

にある 1 次 ASTE 起点 (PASTE0) と比較して、現行の 1 次アドレス・スペースが現行のディスクパッチ可能単位用のサブスペース・グループの中にあるかどうかを判別します。この比較を行うために、ビット 1-25 が PASTE0 と比較されることも、バイト 0-3 の内容全体が制御レジスター 5 のビット位置 33-63 の内容と比較されることもあります。BRANCH IN SUBSPACE GROUP は、バイト 0-3 のビット 1-25 を、ALET 0 および 1 以外の ALET のアクセス・レジスター変換によりアクセス・リスト・エントリから得られた宛先 ASTE 起点 (DASTE0) と比較して、宛先 ASTE が基本スペース ASTE かどうかを判別します。この比較では、ビット 1-25 が DASTE0 に比較される場合と、バイト 0-3 の内容全体が、左側に 1 個、右側に 6 個の 0 を付加した DASTE0 と比較される場合があります。PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、および LOAD ADDRESS SPACE PARAMETERS は、バイト 0-3 のビット 1-25 を、ASN 変換により得られた ASTE 起点 (ASTE0) と比較します。この比較では、ビット 1-25 が ASTE0 と比較される場合と、バイト 0-3 の内容全体が、左側に 1 個、右側に 6 個の 0 を付加した ASTE0 と比較される場合があります。BRANCH IN SUBSPACE GROUP が ALET 0 を使用するときは、バイト 0-3 のビット 1-25 の右側に 6 個の 0 を付加したものが、宛先 ASTE を示します。

サブスペース・アクティブビット (SA): バイト 4-7 のビット 0 が 1 のときは、ディスクパッチ可能単位について実行された最後の BRANCH IN SUBSPACE GROUP 命令が、そのディスクパッチ可能単位に関連したサブスペース・グループのサブスペースの 1 つに制御を転送したことを示します。ビット 0 が 0 のときは、そのディスクパッチ可能単位について実行された最後の BRANCH IN SUBSPACE GROUP 命令が、サブスペース・グループの基本スペースに制御を転送したか、そのディスクパッチ可能単位についてまだ BRANCH IN SUBSPACE GROUP が実行されていないか、または、そのディスクパッチ可能単位がどのサブスペース・グループにも関連付けられていないことを示します。BRANCH IN SUBSPACE GROUP は、ディスクパッチ可能単位に関連したサブスペース・グループのサブスペースのどれかに制御を転送するときは、バイト 4-7 のビット 0 を 1 に設定し、サブスペース・グループの基本スペースに制御を転送するときは、ビット 0 を 0 に設定します。

サブスペース ASTE 起点 (SSASTE0): バイト 4-7 のビット 1-25 の右側に 6 個の 0 を付加したものが、ディスクパッチ可能単位に対して実行された BRANCH

IN SUBSPACE GROUP 命令により最後に制御が渡されたサブスペースを指定する ASN 第 2 テーブル・エントリの始めを示す 31 ビット実アドレスを形成します。BRANCH IN SUBSPACE GROUP は、ALET 1 以外の ALET を使用してサブスペースに制御を転送するときは、サブスペースの ASTE0 (宛先 ASTE0) をバイト 4-7 のビット位置 1-25 に入れ、ビット位置 26-31 に 0 を入れ、サブスペース・アクティブ・ビット (バイト 4-7 のビット 0) を 1 に設定します。BRANCH IN SUBSPACE GROUP が ALET 1 を使用してサブスペースに制御を転送するときは、バイト 4-7 のビット 1-25 の右側に 6 個の 0 を付加したものが宛先 ASTE を示し、BRANCH IN SUBSPACE GROUP は、サブスペース・アクティブ・ビットを 1 に設定し、バイト 4-7 のビット 26-31 を 0 に設定するか、またはこれらのビットを無変更のままにします。ただし、ビット 1-25 がすべて 0 の場合は、特殊操作例外が認識されません。BRANCH IN SUBSPACE GROUP は、サブスペース・グループの基本スペースに制御を転送するときは、サブスペース・アクティブ・ビットを 0 に設定し、バイト 4-7 のビット 1-31 は変更しません。PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、および LOAD ADDRESS SPACE PARAMETERS は、バイト 4-7 のビット 1-25 を使用して、サブスペース ASTE の中の ASCE のビット 0-55 および 57-63 から、制御レジスター 1 の 1 次 ASCE または制御レジスター 7 の 2 次 ASCE の同じビット設定します。

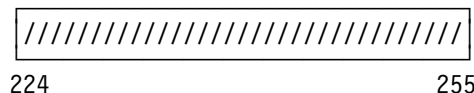
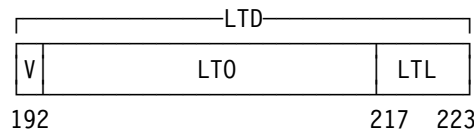
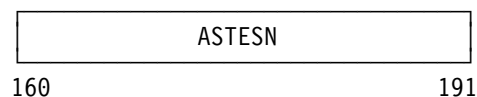
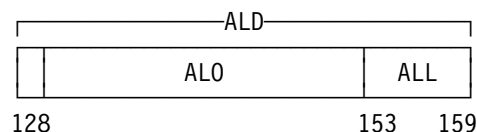
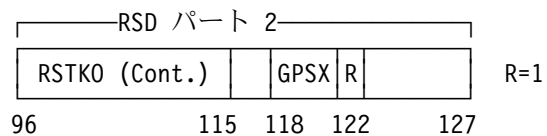
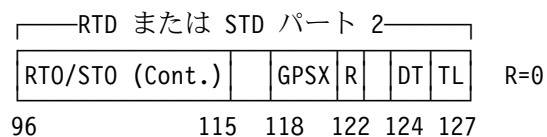
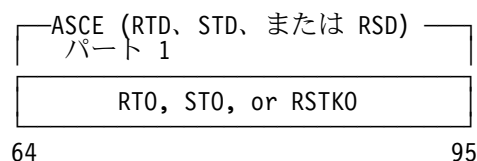
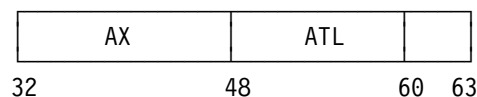
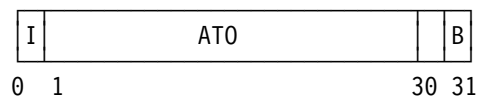
サブスペース ASTE シーケンス番号 (SSASTESN): バイト 12-15 は、DUCT の中の SSASTE0 (バイト 4-7 のビット 1-25) が表しているリンケージ能力を取り消すために使用できます。BRANCH IN SUBSPACE GROUP は、ALET 1 以外の ALET 1 を使用してサブスペースに制御を転送するときは、サブスペース ASTE の中の ASTESN を取得し、それをバイト 12-15 に入れます。ALET 1 を使用してサブスペースに制御を転送するときは、BRANCH IN SUBSPACE GROUP は、バイト 12-15 をサブスペース ASTE の中の ASTESN と比較し、両者が等しくない場合は、ASTE シーケンス例外を認識します。PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、および LOAD ADDRESS SPACE PARAMETERS 命令は、SSASTE0 を使用して、制御レジスター 1 の中の 1 次 ASCE または制御レジスター 7 の中の 2 次 ASCE のビット 0-55 および 57-63 を設定するときは、まず、バイト 12-15 をサブスペース ASTE の中の ASTESN と比較し、この 2 つのフィー

ルドが等しくない場合は、ASTE シーケンス例外を認識します。

バイト 16-19 については、5-44ページの『ディスパッチ可能単位制御テーブルおよびアクセス・リスト指定』で説明されています。バイト 20-23 については、10-5ページの『BRANCH AND SET AUTHORITY』で説明します。バイト 32-39 および 44-47 については、10-114ページの『TRAP』で説明します。バイト 24-27、40-43、および 48-63 は将来の拡張に備えて予約されているものであり、すべて 0 でなければなりません。バイト 28-31 はプログラミング用に使用できます。

サブスペース・グループ ASN 第 2 テーブル・エントリー

64 バイトの ASN 第 2 テーブル・エントリーのフォーマットは、以下のとおりです。



BRANCH IN SUBSPACE GROUP の場合、ASTE のバイト 0-31 のフィールドは以下のように割り振られます。

ASX 無効ビット (I): ビット 0 は、ASTE に関連したアドレス・スペースが使用可能かどうかを制御します。ALET 1 のアクセス・レジスター変換時、または、BRANCH IN SUBSPACE GROUP の場合は ALET 0 および 1 以外の ALET のアクセス・レジスター変換時に、ビット 0 が 0 であるときは、変換が進められます。このビットが 1 のときは、ASTE 妥当性例外が認識されます。ALET 0 のアクセス・レジスター変換では、このビットは無視されます。ASTE が、ディスパッチ可能単位制御テーブル内のサブスペース ASTE 起点 (SSASTE0) で指定されている場合は、ビット 0 は、ビット 160-191 (ASTESN) の定義で説明する用途にも使用されます。

権限テーブル起点 (ATO): ビット 1-29 は、BRANCH IN SUBSPACE GROUP では使用されません。

基本スペース・ビット (B): ビット 31 が 1 のときは、ASTE に関連付けられているアドレス・スペースが、サブスペース・グループの基本スペースであることを示します。BRANCH IN SUBSPACE GROUP が、ALET 0 および 1 以外の ALET を使用して宛先 ASTE を検索する場合に、宛先 ASTE 起点がディスパッチ可能単位制御テーブル内の基本 ASTE 起点に等しくなく、その宛先 ASTE のビット 31 が 1 であるときは、特殊操作例外が認識されます。

許可指標 (AX): ビット 32-47 は、BRANCH IN SUBSPACE GROUP では使用されません。

権限テーブル長 (ATL): ビット 48-59 は、BRANCH IN SUBSPACE GROUP では使用されません。

アドレス・スペース制御エレメント (ASCE): ビット 64-127 は、8 バイトのアドレス・スペース制御エレメント (ASCE) で、これはセグメント・テーブル指定

(STD)、領域テーブル指定 (RTD)、または実スペース指定 (RSD) のいずれかです。(「領域テーブル指定」という用語は、領域第 1 テーブル指定、領域第 2 テーブル指定、または領域第 3 テーブル指定を意味します。) ASCE フィールドは、BRANCH IN SUBSPACE GROUP で行われるアクセス・レジスタ変換の結果として取得されます。BRANCH IN SUBSPACE GROUP が、ALET 0 および 1 以外の ALET を使用して宛先 ASTE を検索する場合に、宛先 ASTE 起点がディスパッチ可能単位制御テーブル内の基本 ASTE 起点に等しくなく、その宛先 ASTE のサブスペース・グループ制御ビット (ビット 118) が 0 であるときは、特殊操作例外が認識されます。BRANCH IN SUBSPACE GROUP は、現行のディスパッチ可能単位に関連したサブスペース・グループの基本スペースに制御を転送するときは、ビット 64-127 を制御レジスタ 1 に入れます。また、サブスペース・グループのサブスペースの 1 つに制御を転送するときは、BRANCH IN SUBSPACE GROUP は、ビット 65-119 および 121-127 を、それぞれ、制御レジスタ 1 のビット位置 0-55 および 57-63 に入れます。PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、および LOAD ADDRESS SPACE PARAMETERS による ASN 変換の後には、ビット 64-127 は、3-18 ページの『ASN 第 2 テーブル・エントリ』の説明のとおり使用されます。

リンケージ・テーブル指定 (LTD): ビット 192-223 は、BRANCH IN SUBSPACE GROUP では使用されません。

アクセス・リスト指定 (ALD): この ASTE が、制御レジスタ 5 の中の 1 次 ASTE 起点により指定されている場合は、ビット 128-159 は 1 次スペース・アクセス・リスト指定 (PSALD) です。アクセス・レジスタ変換時に、ALET の中の 1 次リスト・ビット (ビット 7) が 1 であれば、PSALD は有効アクセス・リスト指定です。

ASN 第 2 テーブル・エントリ・シーケンス番号 (ASTESN): ビット 160-191 は、この ASTE を指定しているアクセス・リスト・エントリが表すアクセス能力の取り消しを制御するために使用されます。アクセス・レジスタ変換の際に、ビット 160-191 はアクセス・リスト・エントリ内の ASTESN と比較され、両者が等しくないときは ASTE シーケンス例外が認識されます。

ASTE のビット 224-255 は、プログラミング用に使用できます。ASTE がディスパッチ可能単位制御テーブル

内のサブスペース ASTE 起点 (SSASTE0) により指定されている場合は、ビット 160-191 は、その SSASTE0 が表しているリンケージ能力の取り消しを制御するためにも使用されます。BRANCH IN SUBSPACE GROUP が、ALET 1 を使用して、SSASTE0 で指定されているサブスペースに制御を転送するとき、または PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、SET SECONDARY ASN、または LOAD ADDRESS SPACE PARAMETERS が、SSASTE0 を使用して、制御レジスタ 1 の 1 次 ASCE または制御レジスタ 7 の 2 次 ASCE のビット 0-55 および 57-63 を、サブスペース ASTE 内の ASCE の同じビットから設定するときは、これらの命令は、まずサブスペース ASTE のビット 0 が 0 かどうかをテストし、0 でなければ ASTE 妥当性例外を認識します。次に、ビット 160-191 をディスパッチ可能単位制御テーブル内のサブスペース ASTE シーケンス番号 (SSASTESN) と比較して、両者が等しくない場合は ASTE シーケンス例外を認識します。ただし、LOAD ADDRESS SPACE PARAMETERS において上記の 2 つの例外が存在する場合は、この命令は、例外を認識する代わりに条件コード 1 または 2 を設定します。

プログラミング上の注意: バイト 0-31 の未使用フィールドおよびバイト 32-63 のすべてを含めて、ASTE の未使用フィールドは、すべて 0 に設定する必要があります。これらのフィールドは将来の拡張のための予約されており、これらのフィールドに非ゼロ値を入れるプログラムは、将来のマシンでは互換性を保って動作することができなくなるおそれがあります。

サブスペース置き換え操作

サブスペース・グループ機能には、PROGRAM CALL、PROGRAM TRANSFER、PROGRAM RETURN、SET SECONDARY ASN、および LOAD ADDRESS SPACE PARAMETERS のサブスペース置き換え操作が含まれています。これらの操作が適用されるのは、上記 5 つの命令のいずれかが実行されるディスパッチ可能単位が、サブスペース・アクティブと呼ばれる状態にあるときです。ディスパッチ可能単位がサブスペース・アクティブの状態にあるのは、そのディスパッチ可能単位が、BRANCH IN SUBSPACE GROUP を使用してサブスペース・グループのサブスペースの 1 つに制御を転送した後で、まだ、BRANCH IN SUBSPACE GROUP を使用してグループの基本スペースに制御を戻していない場合です。

サブスペース置き換え操作の定義は、第10章、『制御命令』の中の上記の各命令に関する定義に収めてありま

す。ここでは、これらの操作の概要を説明しておきます。ASN 変換の結果、アドレス・スペースが 1 次または 2 次 アドレス・スペースとして設定されたときに、そのアドレス・スペースがいずれかのサブスペース・グループ内にある、つまり、そのアドレス・スペースのアドレス・スペース制御エレメント (ASCE)(制御レジスター 1 の中の新しい PASCE または制御レジスター 7 の中の SASCE) のサブスペース・グループ制御ビット (ビット 54 (G)) が 1 である場合、およびそのディスパッチ可能単位がサブスペース・アクティブの状態にある、つまり、ディスパッチ可能単位制御テーブル (DUCT) 内のサブスペース・アクティブ・ビット (ワード 1 のビット 0 (SA) が 1 である場合は、ASN 変換の結果として得られたそのアドレス・スペースの ASN 第 2 テーブル・エントリ (ASTE) 起点 (ASTE0) が、DUCT の基本 ASTE 起点 (BASTE0)(ワード 0 のビット 1-25) と比較されます。その ASTE0 と BASTE0 が等しければ、以下の操作が行われます。このディスパッチ可能単位が入った最後のサブスペースの ASTE、つまり DUCT の中のサブスペース ASTE 起点 (SSASTE0) が示す ASTE の中で、ビット 0 が 1 である場合は、ASTE 妥当性例外が認識されます。サブスペース ASTE のワード 5 の中の ASTE シーケンス番号 (ASTESN) が、DUCT のワード 3 の中のサブスペース ASTESN (SSASTESN) に等しくない場合は、ASTE シーケンス例外が認識されます。ただし、LOAD ADDRESS SPACE PARAMETERS は、ASTE 妥当性例外または ASTE シーケンス例外を認識する代わりに、0 以外の条件コードを設定します。例外が存在しない場合は、アドレス・スペースの ASCE のビット 0-55 および 57-63 (制御レジスター 1 の中の PASCE または制御レジスター 7 の中の SASCE) が、サブスペース ASTE のワード 2 の中の ASCE の同じビットで置き換えられます。

DUCT またはサブスペース ASTE にアクセスしようとしたときにアドレッシング例外が認識された場合は、命令の実行は抑止されます。ASTE 妥当性例外または ASTE シーケンス例外が認識された場合は、命令の実行は無効化されます。このように無効化または抑止が生じた場合は、レジスターの内容は変更されず、すべて命令実行の開始時点の内容のまま残されます。

DUCT またはサブスペース ASTE へのアクセスには、キー制御保護は適用されません。

ASN 変換の結果得られた ASTE0 を BASTE0 と比較する際には、ASTE0 が BASTE0 と比較される場合と、ASTE0 の左側に 1 個、右側に 6 個の 0 を付加したものが、DUCT のワード 0 の内容全体と比較される場合があります。

DUCT の中の SSASTE0 を使用してサブスペース ASTE にアクセスする際に、SSASTE0 がすべて 0 であるかどうかの検査は行われません。

DUCT およびサブスペース ASTE に対する参照は単一アクセス参照であり、他の CPU からはワード単位のアクセスのように見えます。DUCT のワードがアクセスされる順序は一定していません。サブスペース ASTE のワードのアクセス順序は、ワード 0 が最初にアクセスされるという点以外は不定です。

サブスペース置き換え操作中に認識される例外を、総称的にサブスペース置き換え例外と呼びます。これらの例外は、6-43ページの『サブスペース置き換え例外』に優先度の順にリストしてあります。

リンケージ・スタックの概要

リンケージ・スタックに関連したほとんどの機能は、このセクションと、5-65ページの『リンケージ・スタックの操作』で説明されています。さらに、スタッキング PROGRAM CALL 命令および PROGRAM RETURN 命令のトレースについては、第5章、『プログラムの実行』、割り込みについては、第6章、『割り込み』、そして命令については、第10章、『制御命令』で説明されています。

要約

提供されている主な機能には、以下のものがあります。

1. テーブル・ベースのサブルーチン・リンケージ・メカニズム。これは、PSW および制御レジスターの状況変化を制御し、リンケージ・スタック内のエントリを使用して、汎用レジスターおよびアクセス・レジスターのこの状況と内容を保管および復元します。
2. リンケージ・スタックを使用するブランチ・タイプのリンケージ・メカニズム。
3. 追加の 2 ワードを現行のリンケージ・スタック・エントリに入れ、そのエントリ内にある汎用レジスターおよびアクセス・レジスターの内容のすべての状況を検索するための命令。
4. プログラムが特定のアクセス・リスト・エントリ・トークンの使用を許可されているかどうかを判別する命令。
5. プログラムの問題分析のための援助機能。

さらに、上記の機能を制御するための制御メカニズムと権限メカニズムが組み込まれています。

個々のディスパッチ可能単位ごとに、それぞれ専用のリンケージ・スタックを関連付けて使用できるようになっています。特定のディスパッチ可能単位用のリンケージ・スタックは、そのディスパッチ可能単位のホーム・アドレス・スペースに入っています。

ディスパッチ可能単位のリンケージ・スタックは、キー制御保護によりディスパッチ可能単位から保護されるようになっています。リンケージ・スタックに情報を入れたりそこから情報を検索したりするリンケージ・スタック命令には、キー制御保護は適用されません。

リンケージ・スタック機能は、準特権を持つものと見なされるプログラムが使用するためのものです。準特権プログラムは、問題プログラム状態で実行されるものですが、さらに多くの機能を使用する許可が与えられています。このような許可制御を利用して、プログラムの非階層編成を設定できます。この編成では、呼び出し側プログラムと呼び出し先プログラムのシーケンス内の各プログラムが、そのシーケンス内でそれぞれの前後にあるプログラムとは異なる任意のレベルの権限を持つことができます。各プログラムが使用できる機能の範囲と、プログラムから別のプログラムに制御を転送する能力は、制御プログラムが管理するテーブルの中で規定されています。

リンケージ・スタック命令 (準特権命令) については、第10章、『制御命令』で説明します。この種の命令には以下のものがあります。

- BRANCH AND STACK
- EXTRACT STACKED REGISTERS
- EXTRACT STACKED STATE
- MODIFY STACKED STATE
- PROGRAM RETURN
- TEST ACCESS

さらに、オプションとして、PROGRAM CALL 命令はリンケージ・スタック内のエントリーを形成します。リンケージ・スタックに対して働く PROGRAM CALL 命令を、スタッキング PROGRAM CALL と呼びます。PROGRAM CALL がスタッキング PROGRAM CALL として認識されるかどうかは、エントリー・テーブル・エントリー内のビットの 1 つにより制御されます。

リンケージ・スタックの機能

プログラム制御の転送

リンケージ・スタックを使用することにより、任意の異なる権限レベルで作動するプログラムが、制御プログラムの介入を必要とせずに直接リンクすることができます。呼び出し側プログラムと呼び出し先プログラムのシーケンスの中での各プログラムの権限はそれぞれ異なってもよく、したがって、プログラムの非階層編成を設定できます。主として、各プログラム・モジュールにそれぞれ拡張許可指標を関連付けることにより、モジュラー許可制御を達成できます。これにより、異なる権限を持つプログラム・モジュールが、同じアドレス・スペース内に共存することができます。一方、呼び出し側プログラム・モジュールに関連する拡張許可指標が呼び出し先プログラム・モジュールの実行中に効力を持つ拡張許可指標となるので、異なるディスパッチ可能単位には異なる権限を使用して、呼び出し先プログラム・モジュールを実行することができます。別のプログラムまたは同じ呼び出し先プログラムに異なる権限を関連付けるためのその他の手段として、PSW キー・マスクおよび 2 次 ASN に関するオプションがあります。各プログラムの権限は、制御プログラムが管理するテーブル内で規定されます。異なる PC 番号を使用して同じプログラムを呼び出せるようにテーブルをセットアップすることにより、そのプログラムを呼び出すために使用される PC 番号に応じて、そのプログラムに異なる権限を割り当てることができます。また、プログラムが他のプログラムを呼び出すためにどの PC 番号を使用するかも、テーブルにより制御することができます。

スタッキング PROGRAM CALL および PROGRAM RETURN のリンケージ操作により、異なるアドレス・スペース内にあり権限レベルの異なるプログラムをリンクすることができます。実行状態と、汎用レジスターおよびアクセス・レジスターの内容は、スタッキング PROGRAM CALL の実行時に保管され、PROGRAM RETURN の実行時に部分的に復元されます。リンケージ・スタックは、リンケージ操作時に、実行状態とレジスターの内容の両方を保管および復元するための、効率的な手段を提供します。

PROGRAM CALL 命令の実行時には、PC 番号変換プロセスにより 32 バイトのエントリー・テーブル・エントリーが突き止められます。エントリー・テーブル・エントリー内の PC タイプ・ビットが 1 であれば、スタッキング PROGRAM CALL 操作が指定され、そうでなければ基本 PROGRAM CALL 操作が指定されます。

エントリー・テーブル・エントリーには、基本 PROGRAM CALL およびスタッキング PROGRAM

CALL の両方に適用される情報 (これは 5-28ページの『PC 番号の変換』で説明されている情報で、これには、許可キー・マスクと、新規 ASN、アドレッシング・モード、命令アドレス、問題プログラム状態、PSW キー・マスク、1 次 ASTE アドレス、およびエントリー・パラメーターの指定が含まれます) に加えて、PSW 内のアドレス・スペース制御と PSW キーに関するオプション、および、制御レジスター内の PSW キー・マスク、拡張許可指標、および 2 次 ASN が含まれています。

スタッキング PROGRAM CALL 操作時に、エントリー・テーブル・エントリー内の追加情報を使用して、PSW 内のアドレス・スペース制御を、1 次スペース・モードまたはアクセス・レジスター・モードを指定するように設定できます。PSW キーは、未変更のままにしておくことも、エントリー・テーブル・エントリーからの情報で置き換えることもできます。制御レジスター 3 の中の PSW キー・マスクは、エントリー・テーブル・エントリーからの情報と論理和演算するか、またはその情報で置き換えることができます。制御レジスター 8 の中の拡張許可指標は、未変更のままにしておくことも、エントリー・テーブル・エントリーからの情報で置き換えることもできます。制御レジスター 3 の中の 2 次 ASN は、呼び出し側プログラムまたは呼び出し先プログラムの 1 次 ASN と同じに設定することができます。これにより、呼び出し先プログラムが呼び出し側プログラムの 1 次アドレス・スペースにアクセスする能力を、制御することができます。

スタッキング PROGRAM CALL 操作では、常に、リンケージ・スタック内に状態エントリーと呼ばれるエントリーが作成されます。このエントリーは、実行状態と、汎用レジスター 0-15 および アクセス・レジスター 0-15 の内容を保管するためのものです。保管される実行状態に含まれる情報は、使用された PC 番号、呼び出されたスペースの ID、エントリー・テーブル・エントリーによる変更が行われる前の更新済み PSW、操作の前の拡張許可指標、PSW キー・マスク、1 次 ASN、および 2 次 ASN、そして、操作の後の拡張アドレッシング・モード・ビットです。ただし、保管された更新済み PSW の中の PER マスクの値は、予測不能です。リンケージ・スタック状態エントリーには、そのエントリーが PROGRAM CALL により作成されたものであることを示すエントリー・タイプ・コードも含まれます。

エントリー・テーブル・エントリーに指定されているアドレス・スペース番号 (ASN) が 0 以外の値のときは、スペース切り替え操作が行われます。スペース切り替えが行われる場合の操作を、スペース切り替えを伴う PROGRAM CALL (PC-ss) と呼びます。スペース切り

替えが行われない操作は、現 1 次への PROGRAM CALL (PC-cp) と呼ばれます。

スペース切り替えを伴う PROGRAM CALL は、エントリー・テーブル・エントリーから新しい 1 次 ASTE 起点を取得し、その新しい 1 次 ASTE から新しい 1 次アドレス・スペース制御エレメントを取得し、これらの情報をそれぞれ制御レジスター 5 および 1 に入れます。さらに、2 次 ASN を旧 1 次 ASN または新 1 次 ASN のどちらかに設定したかに応じて、制御レジスター 7 の中の 2 次アドレス・スペース制御エレメントを、それぞれ旧 1 次アドレス・スペース制御エレメントまたは新 1 次アドレス・スペース制御エレメントと同じに設定します。現 1 次への PROGRAM CALL は、2 次 ASN を 1 次 ASN と同じに設定し、2 次アドレス・スペース制御エレメントを 1 次アドレス・スペース制御エレメントと同じに設定します。

PROGRAM RETURN 命令は、スタッキング PROGRAM CALL 操作によりリンケージ・スタックに保管されている情報のほとんどを復元します。復元される情報は、PSW、拡張許可指標、PSW キー・マスク、1 次 ASN、2 次 ASN、および、汎用レジスター 2-14 とアクセス・レジスター 2-14 の内容です。ただし、現 PSW の中の PER マスクは変更されません。PROGRAM RETURN の操作は、PROGRAM RETURN が状態エントリーをアンスタックする操作と考えることができます。

PROGRAM RETURN でスペース切り替え操作が行われるのは、復元された 1 次 ASN が、操作の前に存在していた 1 次 ASN に等しくない場合です。スペース切り替えが行われる場合の操作を、スペース切り替えを伴う PROGRAM RETURN (PR-ss) と呼びます。スペース切り替えが行われない操作は、現 1 次への PROGRAM RETURN (PR-cp) と呼ばれます。

スペース切り替えを伴う PROGRAM RETURN は、復元された 1 次 ASN の ASN 変換を実行して、新しい 1 次 ASTE 起点および新しい 1 次アドレス・スペース制御エレメントを取得し、それぞれ制御レジスター 5 および 1 に入れます。スペース切り替えを伴う PROGRAM RETURN または現 1 次への PROGRAM RETURN では、(1) 復元された 2 次 ASN が復元された 1 次 ASN と同じである場合は、制御レジスター 7 の中の 2 次アドレス・スペース制御エレメントは、制御レジスター 1 の中の 1 次アドレス・スペース制御エレメントと同じに設定され、(2) 復元された 2 次 ASN が復元された 1 次 ASN と同じではない場合は、復元された ASN の ASN 変換と ASN 許可を行うことにより、新しい 2 次

アドレス・スペース制御エレメントが取得されて、制御レジスター 7 に入れられます。

スタッキング PROGRAM CALL 操作および

PROGRAM RETURN 操作が正常に実行されるのは、1 次スペース・モードまたはアクセス・レジスター・モードの場合のみです。CPU が、実モード、2 次スペース・モード、またはホーム・スペース・モードのときは、例外が認識されます。

リンケージ・スタック状態エントリー内でアンスタック抑止ビットと呼ばれるビットを 1 に設定することにより、PROGRAM RETURN がこのエントリーをアンスタックしようとしたときに例外が起きるようにすることができます。このビットが 1 であっても、このエントリーに情報を追加したり、このエントリーから情報を検索したりする命令は、実行できます。アンスタック抑止ビットは、このビットの値が 1 になっている状態エントリーがアンスタックされようとしたときに、制御プログラムが制御権を獲得できるようにするためのものです。

リンケージ・スタックを使用したブランチ

BRANCH AND STACK 命令を使用することにより、リンケージ・スタックに、実行状態と、汎用レジスターおよびアクセス・レジスターの内容も保管することができます。BRANCH AND STACK は、PC 番号を使用する代わりに、他のブランチ命令の場合と同様にブランチ・アドレスを使用します。BRANCH AND STACK を PROGRAM RETURN と一緒に使用することにより、同一アドレス・スペース内にあり同じレベルの権限を持つ複数のプログラムを、互いにリンクさせることができます。つまり、BRANCH AND STACK は、命令アドレス以外の実行状態を変更することはできません。

BRANCH AND STACK は、PROGRAM CALL が作成するものと同じリンケージ・スタック状態エントリーを作成します。この 2 つのタイプの状態エントリーを区別する必要があるときは、PROGRAM CALL が作成するエントリーをプログラム呼び出し状態エントリーと呼び、BRANCH AND STACK が作成するエントリーをブランチ状態エントリーと呼びます。ブランチ状態エントリーがプログラム呼び出し状態エントリーと異なる点は、(1) そのエントリーがブランチ状態エントリーであることを示す異なるエントリー・タイプ・コードを含んでいること、および、(2) PC 番号および呼び出し先スペース ID の代わりに、操作の後で存在する基本アドレッシング・モード・ビットおよび命令アドレスを含んでいることです。PSW ビット 32 および 64-127 のこれらの新しい値は、完全な PSW に追加して状態エントリーに保管されます。

BRANCH AND STACK では、状態エントリーに保管されている完全な PSW の一部となっている基本および拡張アドレッシング・モード・ビットと命令アドレスは、現行の (操作の開始時点の) アドレッシング・モード・ビットおよび更新済み命令アドレス (次の順番に当たる命令のアドレス) でもよく、レジスターの中で指定することもできます。このレジスターは、BRANCH AND LINK (BALR のみ)、BRANCH AND SAVE、BRANCH AND SAVE AND SET MODE、または BRANCH AND SET MODE 命令によりリンク情報が入れられたレジスターとすることができます。

BRANCH AND STACK は、呼び出し側プログラムの中、または呼び出し先プログラムのエントリー・ポイント (またはその近く) で使用でき、どちらの場合も、呼び出し先プログラムの終わりにある PROGRAM RETURN 命令は、正しく呼び出し側プログラムに戻ります。エントリー・ポイントで BRANCH AND STACK を使用できるため、旧呼び出し側プログラムを変更せずにリンケージ・スタックを使用できます。

BRANCH AND STACK の R₂ フィールドが 0 のときは、この命令はブランチを起こさずに実行されます。

PROGRAM RETURN は、ブランチ状態エントリーをアンスタックするときに、エントリー内の拡張許可指標、PSW キー・マスク、1 次 ASN、および 2 次 ASN を無視します。PROGRAM RETURN 命令は、エントリーに保管されている PSW および汎用レジスター 2-14 およびアクセス・レジスター 2-14 の内容を復元します。ただし、現 PSW 内の PER マスクは変更されません。

BRANCH AND STACK が正常に実行されるのは、1 次スペース・モードまたはアクセス・レジスター・モードの場合のみです。CPU が、実モード、2 次スペース・モード、またはホーム・スペース・モードのときは、例外が認識されます。

ブランチ状態エントリーの場合も、アンスタック抑止ビットの効果はプログラム呼び出し状態エントリーの場合と同じです。

情報の追加と検索

プログラムで MODIFY STACKED STATE を使用することにより、指定された汎用レジスターのペアに含まれている 2 ワードの情報を、現行のリンケージ・スタック状態エントリー (ブランチ状態エントリーまたはプログラム呼び出し状態エントリー) の変更可能域と呼ばれる区域に入れることができます。これは、呼び出し先プログラムが、必要に応じて、制御プログラムから制御権

を渡されるリカバリー・ルーチンを設定できるようにするためです。

プログラムで EXTRACT STACKED REGISTERS 命令および EXTRACT STACKED STATE 命令を使用することにより、BRANCH AND STACK または PROGRAM CALL が現行の状態エントリーに保管した情報、または MODIFY STACKED STATE がそのエントリーに入れた情報を、取り出すことができます。

EXTRACT STACKED REGISTERS (EREGG) は、指定された範囲の汎用レジスタおよびアクセス・レジスタの内容を、その内容が保管されたときの元のレジスタに戻します。EXTRACT STACKED REGISTERS (EREG) も同じことを行いますが、汎用レジスタのビット 32-63 のみを復元し、ビット 0-31 は変更しないという点が異なります。EXTRACT STACKED STATE は、状態エントリーに保管されている (またはそこに入れられている) 非レジスタ情報のワードのペアを取得し、指定された汎用レジスタのペアのビット位置 32-63 に入れます。また、EXTRACT STACKED STATE は、状態エントリーに保管されている PSW を含む 2 個のダブルワードを取得し、指定された汎用レジスタのペアのビット位置 0-63 に入れます。

EXTRACT STACKED STATE は、現行の状態エントリーがブランチ状態エントリーかプログラム呼び出し状態エントリーかを示す条件コードを設定します。

許可のテスト

TEST ACCESS 命令は、指定されたアクセス・レジスタ内のアクセス・リスト・エントリー・トークン (ALET)、および指定された汎用レジスタ内の拡張許可指標 (EAX) を、オペランドとして使用します。

TEST ACCESS は、制御レジスタ 8 の中の現 EAX の代わりに指定された EAX を使用してアクセス・レジスタ変換プロセスを ALET に適用し、その結果を示す条件コードを設定します。この条件コードは以下のいずれかを示します。(1) ALET は 16 進数 00000000 である。(2) ALET はディスパッチ可能単位アクセス・リスト内のエントリーを示しており、アクセス・レジスタ変換において例外を起こさずに変換できる。

(3) ALET は 1 次スペース・アクセス・リスト内のエントリーを示しており、アクセス・レジスタ変換において例外を起こさずに変換できる。(4) ALET は 16 進数 00000001 であるか、アクセス・レジスタ変換において例外を引き起こす。

TEST ACCESS の主な目的は、呼び出し先プログラムが、呼び出し側プログラムの EAX を使用して、呼び出し側プログラムから渡された ALET が呼び出し側プログラムから使用を許可されたものであるかどうかを判別

できるようにすることです。これは、呼び出し先プログラムが自身の EAX を使用して AR 指定アドレス・スペースを操作するためには、呼び出し側プログラムが、呼び出し側プログラムの EAX を使用してそのスペースを操作する許可を持っていないなければならないというプログラミング規則をサポートするものです。呼び出し先プログラムは、EXTRACT STACKED STATE 命令を使用して、TEST ACCESS で使用する呼び出し側プログラムの EAX を現行のリンケージ・スタック状態エントリーから取得することができます。

TEST ACCESS のもう 1 つの目的は、ALET が 1 次アドレス・スペースを指定する 16 進数 00000000 か、または 2 次アドレス・スペースを指定する 16 進数 00000001 であるという、特殊なケースを示すことにあります。PROGRAM CALL は、1 次および 2 次アドレス・スペースを変更することがあるので、ALET の 16 進数 00000000 および 16 進数 00000001 という値は、呼び出し先プログラムが使用するときには、呼び出し側プログラムが使用していたときとは異なるアドレス・スペースを指していることがあります。

さらに、TEST ACCESS には、ALET が 1 次スペース・アクセス・リスト内のエントリーを指定しているかどうかを示すという目的もあります。これは、スペース切り替えを伴うプログラム・リンケージ操作により 1 次アドレス・スペースが変更された後は、この指定が誤っていることがあるためです。

プログラム問題の分析

プログラム問題の分析を効率化するために、リンケージ・スタックが使用されているときに、3 つの追加のリンケージ操作に関するトレース・エントリーを暗黙的に作成するオプションが用意されています。ブランチ・トレースがオンのときは、BRANCH AND STACK 命令が実行されてブランチが起きるたびに、このトレース・エントリーが作成されます。ASN トレースがオンのときは、スタッキング PROGRAM CALL 操作が行われるたびに、そして PROGRAM RETURN が PROGRAM CALL により作成されたリンケージ・スタック状態エントリーをアンスタックするたびに、トレース・エントリーが作成されます。モード・トレースがオンのときは、スタッキング PROGRAM CALL 操作または PROGRAM RETURN 操作が行われ、PSW ビット 31 が変更されるたびに、トレース・エントリーが作成されます。ただし、PROGRAM RETURN の場合は、ASN トレースの結果としてトレース・エントリーが作成される場合は、モード・トレースに関するトレース・エントリーは作成されません。トレースの詳細な定義は、4-10ページの『トレース』に収めてあります。

さらに分析を効率化するための手段として、ブランチを起こす BRANCH AND STACK、スタッキング PROGRAM CALL、および PROGRAM RETURN は、PER ブランチ成功イベントとしても認識されます。PROGRAM RETURN では、BRANCH AND STACK または PROGRAM CALL によりアンスタック状態エントリーが作成されていることがあります。

スペース切り替えを伴うスタッキング PROGRAM CALL 命令または PROGRAM RETURN 命令を実行したときに、操作の前または後で 1 次スペース切り替えイベント制御が 1 であるか、PER イベントが示される場合は、スペース切り替えイベントが発生します。

リンケージ・スタック・エントリー・テーブル・エントリー

エントリー・テーブル・エントリー内の、ビット 130-159 を除くすべてのフィールドについては、5-29 ページの『エントリー・テーブル・エントリー』で説明されています。このセクションでは、ビット 130-159 についてのみ説明します。

エントリー・テーブル・エントリーのフォーマットは、以下のとおりです。

ビット 129 が 0 の場合

0			31
A	Entry Instruction Address		P
32			63

ビット 129 が 1 の場合

		Entry Instruction Address (Part 1)	
0			31
		Entry Instruction Address (Part 2)	
32			63

Authorization Key Mask		ASN	
64	80	95	
Entry Key Mask			
96	112	127	

T	G	K	M	E	C	S	EK		Entry Ext. Auth. Index
128	131						136	140	144
									159

ASTE Address									
								186	191

Entry Parameter (Part 1)									
								192	123

Entry Parameter (Part 2)									
								224	255

ビット位置 130-159 の中のフィールドは、以下のように割り振られます。

PSW キー制御 (K): ビットが 131 のときは、スタッキング PROGRAM CALL 操作の一環として、ビット 136-139 が PSW 中の PSW キーと置き換わることを示します。このビットが 0 のときは、PSW キーは変更されません。基本 PROGRAM CALL 操作では、ビット 131 は無視されます。

PSW キー・マスク制御 (M): ビット 132 が 1 のときは、スタッキング PROGRAM CALL 操作の一環として、ビット 96-111 が制御レジスター 3 中の PSW キー・マスクと置き換わることを示します。このビットが 0 のときは、スタッキング PROGRAM CALL 操作の一環として、ビット 96-111 は制御レジスター 3 中の PSW キー・マスクと論理和演算されます。基本 PROGRAM CALL 操作では、ビット 132 は無視されます。

拡張許可指標制御 (E): ビット 133 が 1 のときは、スタッキング PROGRAM CALL 操作の一環として、ビット 144-159 が制御レジスター 8 中の現行の拡張許可指標と置き換わることを示します。このビットが 0 のときは、現行の拡張許可指標は変更されません。基本 PROGRAM CALL 操作では、ビット 133 は無視されます。

アドレス・スペース制御の制御 (C): ビット 134 が 1 のときは、スタッキング PROGRAM CALL 操作の一環として、現 PSW のビット 17 を 1 に設定することを示します。このビットが 0 のときは、ビット 17 は 0 に設定されます。スタッキング PROGRAM CALL 命令の発行時には CPU は 1 次スペース・モードまたはアクセス・レジスター・モードになっていなければならないので、結果の CPU のモードは、ビット 134 が 1 ならアクセス・レジスター・モードになり、ビット 134

が 0 なら 1 次スペース・モードになります。基本 PROGRAM CALL 操作では、ビット 134 は無視されます。

2 次 ASN 制御 (S): ビット 135 が 1 のときは、スペース切り替えを伴うスタッキング PROGRAM CALL 操作 (PC-ss) の一環として、ビット 80-95 が新しい 2 次 ASN になり、新しい SASCE を新しい PASCE と同じに設定することを示します。このビットが 0 のときは、新しい SASN および SASCE は、それぞれ、呼び出し側プログラムの PASN および PASCE と同じに設定されます。基本 PROGRAM CALL 操作、および現 1 次へのスタッキング PROGRAM CALL 操作 (PC-cp) では、ビット 135 は無視されます。

エンタリー・キー (EK): PSW キー制御ビット (ビット 131) が 1 であれば、スタッキング PROGRAM CALL 操作の一環として、ビット 136-139 は PSW の中の PSW キーと置き換わります。ビット 131 が 0 のときは、ビット 136-139 は無視され、現 PSW キーは変更されません。基本 PROGRAM CALL 操作では、ビット 136-139 は無視されます。

エンタリー拡張許可指標: 拡張許可指標制御 (ビット 133) が 1 であれば、スタッキング PROGRAM CALL 操作の一環として、ビット 144-159 が制御レジスター 8 のビット 32-47 と置き換わります。ビット 133 が 0 のときは、ビット 144-159 は無視され、現行の拡張許可指標は変更されません。基本 PROGRAM CALL 操作では、ビット 144-159 は無視されます。

ビット 130 および 140-143 は、将来の拡張に備えて予約されており、いずれも 0 でなければなりません。

リンケージ・スタックの操作

リンケージ・スタックは、各ディスパッチ可能単位ごとに制御プログラムにより形成することができます。リンケージ・スタックは、BRANCH AND STACK 操作およびスタッキング PROGRAM CALL 操作中に、実行状態と、汎用レジスターおよびアクセス・レジスターの内容を保管するために使用されます。また、PROGRAM RETURN 操作で、実行状態と汎用レジスターおよびアクセス・レジスターの内容の一部を復元するためにも、リンケージ・スタックが使用されます。

リンケージ・スタックは、仮想記憶域内に常駐します。特定のディスパッチ可能単位用のリンケージ・スタックは、そのディスパッチ可能単位用のホーム・アドレス・

スペースの中にあります。ホーム・アドレス・スペースは、制御レジスター 13 の中のホーム・アドレス・スペース制御エレメントで指定されます。

リンケージ・スタックは、問題プログラム状態のプログラムから保護されるようになっています。したがって、問題プログラム状態のプログラムは、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE 命令以外の方法で、リンケージ・スタックに保管されている情報を検査または変更することはできません。この保護は、キー制御保護を利用して達成できます。

リンケージ・スタックは、互いにチェーニングされた複数のリンケージ・スタック・セクションにより形成することができます。リンケージ・スタック・セクションは可変長です。各リンケージ・スタック・セクションの最大長は、65,560 バイトです。

リンケージ・スタック内には、ヘッダー・エンタリー、トレーラー・エンタリー、および状態エンタリーの 3 種類のエンタリーがあります。ヘッダー・エンタリーとトレーラー・エンタリーは、それぞれリンケージ・スタック・セクションの始めと終わりにあり、リンケージ・スタック・セクションを相互にチェーニングするために使用されます。ヘッダー・エンタリーおよびトレーラー・エンタリーは、制御プログラムにより形成されます。状態エンタリーは、BRANCH AND STACK またはスタッキング PROGRAM CALL 操作で保管された実行状態およびレジスターの内容を収容するためのもので、その操作中に作成されます。状態エンタリーはさらに細分されて、BRANCH AND STACK により形成された場合はブランチ状態エンタリー、そして PROGRAM CALL により形成された場合はプログラム呼び出し状態エンタリーとして区別されます。

BRANCH AND STACK およびスタッキング

PROGRAM CALL 操作で、状態エンタリーを作成し、そこに情報を保管するアクションを、スタッキング・プロセスと呼びます。PROGRAM RETURN 操作で、状態エンタリーから情報を復元し、論理的にそのエンタリーを削除するアクションは、アンスタッキング・プロセスと呼ばれます。アンスタッキング・プロセスの一環として状態エンタリーを見つける操作は、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE 操作の中でも行われます。

どのタイプのリンケージ・スタック・エンタリーも、長さは 8 バイトの倍数です。ヘッダー・エンタリーとトレーラー・エンタリーの長さは、それぞれ 16 バイトです。状態エンタリーの長さは 296 バイトです。

ヘッダー・エンタリー、トレーラー・エンタリー、および状態エンタリーのどれにも、末尾に共通の 8 バイト域があり、これをエンタリー記述子と呼びます。制御レジスター 15 の中のリンケージ・スタック・エンタリー・アドレスは、リンケージ・スタック・セクション内のトレーラー・エンタリーを除く最後のリンケージ・スタック・エンタリーのエンタリー記述子の最左端のバイトを指定します。このエンタリーを現リンケージ・スタック・エンタリーと呼び、このセクションを現リンケージ・スタック・セクションと呼びます。

リンケージ・スタック・セクション内の各エンタリー記述子(そのセクションのトレーラー・エンタリー内の記述子を除く)には、そのエンタリー記述子の終わりからトレーラー・エンタリーの始めまでの間のスペースの量を示すフィールドがあります。このフィールドを残余フリー・スペース・フィールドと呼びます。トレーラー・エンタリー内の残余フリー・スペース・フィールドは、使用されません。

スタック・プロセスにおいて、リンケージ・スタック内に新しい状態エンタリーを作成する必要がある場合は、現リンケージ・スタック・セクション内に新規エンタリーを収容できる十分な残余フリー・スペースがあれば、現リンケージ・スタック・エンタリーのエンタリー記述子の直後に、新規エンタリーが置かれます。現セクション内に十分な残余フリー・スペースがなくても、現セクション内のトレーラー・エンタリーが現セクションの後に別のセクションが続いていることを示している場合は、その後続セクション内に十分な残余フリー・スペースがあれば、後続セクションのヘッダー・エンタリーのエンタリー記述子の直後に、新規エンタリーが置かれます。トレーラー・エンタリーが後続セクションがないことを示している場合は、例外が認識され、プログラム割り込みが発生します。その場合に、新たなセクションを割り振り、それを現セクションにチェーニングし、**BRANCH AND STACK** または **スタッキング PROGRAM CALL** 命令を再実行させるのは、制御プログラムの責任です。後続セクションがあっても、十分な残余フリー・スペースがないときは、例外が認識されます。

トレーラー・エンタリーの位置を指定するために使用される残余フリー・スペース値が 8 の倍数でない場合は、例外が認識されます。リンケージ・スタック・セクションのトレーラー・エンタリーの位置を指定するために使用される残余フリー・スペース値が 8 の倍数になるようにするためには、そのセクション内のヘッダー・エンタリーの中で、残余フリー・スペース値を 8 の倍数に設定することが必要です。

スタッキング・プロセスで新しい状態エンタリーが正常に形成されると、制御レジスター 15 の中のリンケ

ージ・スタック・エンタリー・アドレスが、新規エンタリーのエンタリー記述子の最左端のバイトを指すように更新され、その新規エンタリーが新しい現リンケージ・スタック・エンタリーになります。

PROGRAM RETURN によるアンスタッキング・プロセスでは、現リンケージ・スタック・エンタリーが状態エンタリーである場合は、プロセスはそのエンタリーに対して行われ、リンケージ・スタック・エンタリー・アドレスは、同じリンケージ・スタック・セクション内の前のエンタリーのエンタリー記述子を指すように更新されます。そして、その前エンタリーが現エンタリーになります。新しい現エンタリーは、別の状態エンタリーのこともあり、ヘッダー・エンタリーのこともあります。

リンケージ・スタック・セクションのヘッダー・エンタリーは、先行するセクションがあるかどうかを示します。先行セクションがある場合は、ヘッダー・エンタリーには、その先行セクション内のトレーラー・エンタリーを除く最後のリンケージ・スタック・エンタリーのアドレスが含まれています。その最後のエンタリーは、リンケージ・スタック内にエラーがない限り、他のヘッダー・エンタリーではなく、状態エンタリーです。

アンスタッキング・プロセスが実行されるときに、現リンケージ・スタック・エンタリーがヘッダー・エンタリーであり、そのヘッダー・エンタリーが先行するリンケージ・スタック・セクションが存在することを示している場合は、先行セクションの中で指定されているエンタリー状態エンタリーであれば、そのエンタリーが現エンタリーであるものとしてアンスタッキング・プロセスが進められます。ヘッダー・エンタリーが先行セクションがあることを示していないか、先行セクション内で指定されているエンタリーが状態エンタリーでない場合は、例外が認識されます。

EXTRACT STACKED REGISTERS, EXTRACT STACKED STATE、または **MODIFY STACKED STATE** でアンスタッキング・プロセスが実行されるときは、状態エンタリーを見つける操作は行われますが、制御レジスター 15 の中のリンケージ・スタック・エンタリー・アドレスは変更されません。

リンケージ・スタック・セクション内の各エンタリー記述子には、そのセクション内の次のリンケージ・スタック・エンタリー(トレーラー・エンタリーを除く)の長さを指定するフィールドが含まれています。スタッキング・プロセスで状態エンタリーが作成されるときには、そのエンタリー内のこのフィールドには 0 が入れられ、そのエンタリーの長さは前のエンタリーのこのフィールドに入れられます。**PROGRAM RETURN** によるアンスタッキング・プロセスで状態エンタリーが論理的に削除されたときは、先行するエンタリーのこのフィールド

に 0 が入れられます。このフィールドは、次エントリー・サイズ・フィールドと呼ばれます。

リンケージ・スタックに対するスタッキングまたはアンスタッキング・プロセスには、キー制御保護は適用されませんが、低アドレス保護とページ保護は適用されません。

リンケージ・スタック操作制御

リンケージ・スタックの使用は、制御レジスター 13 中のホーム・アドレス・スペース制御エレメントと、制御レジスター 15 中のリンケージ・スタック・エントリー・アドレスにより制御されます。ホーム・アドレス・スペース制御エレメントについては、3-25ページの『動的アドレス変換』で説明されています。ここでは、リンケージ・スタック・エントリー・アドレスについて説明します。

制御レジスター 15

制御レジスター 15 は、現リンケージ・スタック・エントリーのエントリー記述子の位置を指定します。このレジスターのフォーマットは以下のとおりです。

Linkage-Stack-Entry Address	
0	61 63

リンケージ・スタック・エントリー・アドレス: 制御レジスター 15 のビット 0-60 の右側に 3 個の 0 を付加したものが、現リンケージ・スタック・セクション内の現リンケージ・スタック・エントリーのエントリー記述子の 64 ビットのホーム仮想アドレスを形成します。ビット 0-60 は、BRANCH AND STACK およびスタッキング PROGRAM CALL によるスタッキング・プロセス、および PROGRAM RETURN によるアンスタッキング・プロセスで、変更されます。制御レジスター 15 のビット 0-60 が変更された場合は、ビット 61-63 は 0 に設定されます。

リンケージ・スタック

リンケージ・スタックは、リンケージ・スタック・エントリーを含む 1 つ以上のリンケージ・スタック・セクションから成っています。リンケージ・スタック・エントリーの基本タイプには、ヘッダー・エントリー、トレーラー・エントリー、状態エントリーの 3 つがあります。状態エントリーは、さらに、ブランチ状態エントリーとプログラム呼び出し状態エントリーに細分されます。

どのタイプのリンケージ・スタック・エントリーにも、末尾にエントリー記述子があります。制御レジスター

15 中のリンケージ・スタック・エントリー・アドレスは、現リンケージ・スタック・セクション内の現リンケージ・スタック・エントリーのエントリー記述子の最左端バイトを指定します。

リンケージ・スタックは、制御レジスター 13 中のホーム・アドレス・スペース制御エレメントが指しているホーム・アドレス・スペース内に常駐します。

エントリー記述子

各リンケージ・スタック・エントリーの終わりには、エントリー記述子があります。エントリー記述子は長さが 8 バイトで、フォーマットは以下のとおりです。

U	ET	SI	RFS	NES		
0	1	8	16	32	48	63

エントリー記述子のフィールドは以下のように割り振られています。

アンスタック抑止ビット (U): PROGRAM RETURN によるアンスタッキング・プロセスで検出されたヘッダー・エントリーまたは状態エントリーのエントリー記述子のビット 0 が 1 であるときは、スタック操作例外が認識されます。トレーラー・エントリーの場合、および EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE によるアンスタッキング・プロセスの場合は、ビット 0 は無視されます。制御プログラムでは、現リンケージ・スタック・エントリー (ヘッダー・エントリーまたは状態エントリー) 中のビット 0 を、一時的に 1 に設定することができます。これにより、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE の実行は成功するが、PROGRAM RETURN の実行は成功しないようにすることができます。スタッキング・プロセスで状態エントリーが形成される時点では、そのエントリーのエントリー記述子のビット 0 は 0 に設定されます。

エントリー・タイプ (ET): ビット 1-7 は、エントリー記述子を含むリンケージ・スタック・エントリーのタイプを示すコードです。割り当てられているコードには以下のものがあります。

コード	エントリー・タイプ
(2 進数)	
0001001	ヘッダー・エントリー
0001010	トレーラー・エントリー
0001100	ブランチ状態エントリー
0001101	プログラム呼び出し状態エントリー

2 進数コード 0000000-0001000、0001011、および 0001110-0111111 は、将来の割り当てのために予約されています。2 進数コード 1000000-1111111 は、プログラミング用に使用できます。

スタッキング・プロセスで状態エントリーが形成される時点では、そのエントリーのエントリー記述子のビット 1-7 は、0001100 または 0001101 に設定されます。

EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN によるアンスタッキング・プロセスにおいて、現リンケージ・スタック・エントリーのビット 1-7 が、そのエントリーが状態エントリーまたはヘッダー・エントリーであることを示していない場合、または、現エントリーがヘッダー・エントリーであるときに、そのヘッダー・エントリー内の逆方向スタック・エントリー・アドレスで指定されているエントリーのビット 1-7 が、その指定されたエントリーが状態エントリーであることを示していない場合は、スタック・タイプ例外が認識されます。ただし、現エントリーと指定されたエントリーが両方ともヘッダー・エントリーである場合は、スタック・タイプ例外ではなくスタック指定例外が認識されます。

セクション ID (SI): ビット 8-15 は、エントリー記述子を含むリンケージ・スタック・セクションの ID で、これは制御プログラムから提供されます。スタッキング・プロセスにより形成される状態エントリーでは、ビット 8-15 は、先行するリンケージ・スタック・エントリー内の制御 ID フィールドの内容と同じに設定されます。

残余フリー・スペース (RFS): ビット 16-31 は、このエントリー記述子の終わりから、同じリンケージ・スタック・セクション内のトレーラー・エントリーの始めまでの間にあるバイトの数を指定します (ただし、トレーラー・エントリーでは、このフィールドは意味を持ちません)。したがって、セクション内の最後の状態エントリー、または状態エントリーがない場合のヘッダー・エントリーでは、ビット 16-31 は、そのセクション内でスタッキング・プロセスの実行のために使用できるバイト数を示します。スタッキング・プロセスにより形成される状態エントリーでは、ビット 16-31 は、先行するリンケージ・スタック・エントリー内の残余フリー・スペース・フィールドの内容から、新規エントリーのサイズ (バイト数) を減じた値に設定されます。リンケージ・スタック・セクション内のヘッダー・エントリーのエントリー記述子では、ビット 16-31 は 8 の倍数でなければなりません (ビット 29-31 は 0 でなければなりません)。さもないと、8 の倍数でない値がそのセクション内

の各状態エントリーのエントリー記述子のビット 16-31 に転搬されます。そして、スタッキング・プロセスが、次のセクションに進むためにそのセクション内のトレーラー・エントリーを見つけようとしたときに、スタック指定例外が認識されます。

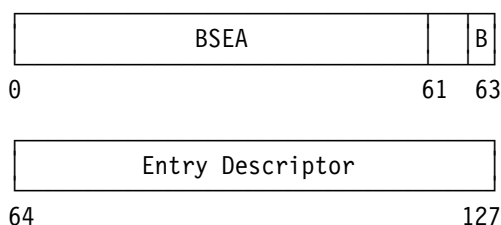
次エントリー・サイズ (NES): 32-47 は、同じリンケージ・スタック・セクション内の、トレーラー・エントリーを除く次のリンケージ・スタック・エントリーのサイズを、バイト数で指定します。現リンケージ・スタック・エントリーでは、このフィールドの内容はすべて 0 です。トレーラー・エントリーでは、このフィールドには意味はありません。スタッキング・プロセスで状態エントリーが形成されるときに、その新規エントリーの次エントリー・サイズ・フィールドには 0 が入れられ、先行するエントリーの次エントリー・サイズ・フィールドに、新規エントリーのサイズが入れます。アンスタッキング・プロセスは、状態エントリーを論理的に削除するときは、先行するエントリーの次エントリー・サイズ・フィールドに 0 を入れ、そのエントリーが現エントリーになります。

スタッキング・プロセスで状態エントリーが形成される時点では、そのエントリーのビット 48-63 は 0 に設定されます。ヘッダー・エントリー、トレーラー・エントリー、または状態エントリーでは、ビット 48-63 は将来の拡張に備えて予約されており、常に 0 でなければなりません。

プログラミング上の注意: 最左端のビットが 1 であるエントリー・タイプ・コードは、割り当てられません。制御プログラムは、現リンケージ・スタック・エントリー (ヘッダー・エントリーまたは状態エントリー) のエントリー・タイプ・コードの最左端のビットを一時的に 1 に設定して、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN の実行が成功しないようにすることができます。

ヘッダー・エントリー

各リンケージ・スタック・セクションの始めには、ヘッダー・エントリーがあります。ヘッダー・エントリーは長さが 16 バイトで、フォーマットは以下のとおりです。



ヘッダー・エントリーの最初の 8 バイトに含まれるフィールドは、以下のように割り振られています。

逆方向スタック・エントリー妥当性ビット (B): ビット 63 が 1 のときは、先行するリンケージ・スタック・セクションが使用可能であり、逆方向スタック・エントリー・アドレス (ビット 0-60) が有効であることを示します。スタッキング・プロセスにおいて、先行セクション内にプロセスを実行するための十分な使用可能スペースがないために、プロセスが先行セクションからこのセクションに移って進められるときは、ビット 63 は 1 に設定されます。アンスタッキング・プロセスでは、このヘッダー・エントリーが現リンケージ・スタック・エントリーであるときに、ビット 63 が 0 である場合は、スタック空例外が認識されます。

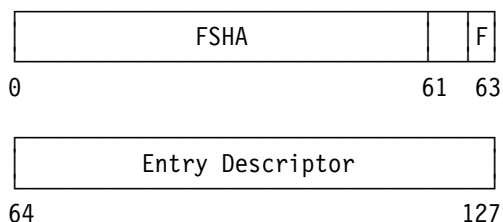
逆方向スタック・エントリー・アドレス (BSEA): ビット 63 が 1 のときは、ビット 0-60 の右側に 3 個の 0 を付加したものが、先行するリンケージ・スタック・セクション内の、トレーラー・エントリーを除く最後のリンケージ・スタック・エントリーのエントリー記述子の 64 ビットのホーム仮想アドレスを形成します。ただし、現リンケージ・スタック・エントリーが、先行するリンケージ・スタック・セクションまたはそれより前のセクションの中にある場合は、ビット 0-60 に指定されているエントリーおよびそれより前のエントリーが論理的に削除されてしまっている可能性があるため、ビット 0-60 は意味を持たないことがあります。ビット 0-60 が設定されるのは、スタッキング・プロセスにおいて、先行セクション内にプロセスを実行するための十分な使用可能スペースがないために、先行セクションからこのセクションに移ってプロセスが進められる場合です。アンスタッキング・プロセスでは、このヘッダー・エントリーが現リンケージ・スタック・エントリーであり、かつ、ビット 63 が 1 であれば、このエントリーは現エントリーとして扱われます。0-60 は、現エントリーとして扱われます。

スタッキング・プロセスでビット 0-60 が設定されるとき、ビット 61 および 62 は 0 に設定されます。ビット 61 および 62 は、将来の拡張に備えて予約されています。

トレーラー・エントリー

各リンケージ・スタック・セクションの終わりには、トレーラー・エントリーがあります。トレーラー・エントリーは、同じリンケージ・スタック・セクション内のヘッダー・エントリーおよび各状態エントリーのエントリー記述子の残余フリー・スペース・フィールドに指定されている区域の直後から始まります。トレーラー・エン

トリーは長さが 16 バイトで、フォーマットは以下のとおりです。



トレーラー・エントリーの最初の 8 バイトに含まれるフィールドは、以下のように割り振られています。

正方向セクション妥当性ビット (F): ビット 63 が 1 のときは、次のリンケージ・スタック・セクションが使用可能であり、正方向セクション・ヘッダー・アドレス (ビット 0-60) が有効であることを示します。スタッキング・プロセスにおいて、現リンケージ・スタック・セクション内に、プロセスを実行するための十分な使用可能スペースがなく、かつ現セクションのトレーラー・エントリー内のビット 63 が 0 である場合は、スタック一杯例外が認識されます。

正方向セクション・ヘッダー・アドレス (FSHA): ビット 63 が 1 のときは、ビット 0-60 の右側に 3 個の 0 を付加したものが、次のリンケージ・スタック・セクション内のヘッダー・エントリーのエントリー記述子の 64 ビットのホーム仮想アドレスを形成します。スタッキング・プロセスにおいて、現セクション内にプロセスを実行するための十分な使用可能スペースがなく、かつビット 63 が 1 である場合は、ビット 0-60 に指定されているヘッダー・エントリーが現リンケージ・スタック・エントリーになります。

ビット 61 および 62 は、将来の拡張に備えて予約されています。

プログラミング上の注意: トレーラー・エントリー内のフィールドは、すべて制御プログラムのみによって設定されます。

状態エントリー

各リンケージ・スタック・セクション内のヘッダー・エントリーの後には、0 個または 1 個以上の状態エントリーが続いています。状態エントリーには、BRANCH AND STACK 命令により形成されるブランチ状態エントリーと、スタッキング PROGRAM CALL 命令により形成されるプログラム呼び出し状態エントリーがあります。状態エントリーは長さが 296 バイトで、フォーマットは以下のとおりです。

16 進数	10 進数		
0 8	0 8	Contents of General Registers 0-15	128 Bytes
70 78	112 120		
80 88	128 136	Other Status Information	96 Bytes
D0 D8	208 216		
E0 E8	224 232	Contents of Access Registers 0-15	64 Bytes
110 118	272 280		
120	288	Entry Descriptor	8 Bytes

状態エントリーのバイト 0-127 には、汎用レジスター 0-15 の内容が、レジスター番号の昇順に従って含まれています。バイト 224-287 には、アクセス・レジスター 0-15 の内容が、レジスター番号の昇順に従って含まれています。これらのフィールドの内容は、BRANCH AND STACK およびスタッキング PROGRAM CALL 操作時に、レジスターから状態エントリーに移されます。PROGRAM RETURN 操作では、汎用レジスター 2-14 およびアクセス・レジスター 2-14 の内容が、状態エントリーからレジスターに復元されます。EXTRACT STACKED REGISTERS では、指定した範囲の汎用レジスターおよびアクセス・レジスターの内容を、状態エントリーからレジスターに復元することができます。

状態エントリーのバイト 128-223 には、BRANCH AND STACK、スタッキング PROGRAM CALL、および MODIFY STACKED STATE によりその他の状況情報が入られます。この状況情報の一部は、PROGRAM RETURN により PSW および制御レジスターに復元され、またすべての情報を EXTRACT STACKED STATE により検査できます。バイト 288-295 には、エントリー記述子が入っています。EXTRACT STACKED STATE は、エントリー記述子内のエントリー・タイプ・コードが、ブランチ状態エントリーまたはプログラム呼び出し状態エントリーのどちらを指定しているかを示す条件コードを設定します。

状態エントリーのバイト 128-223 の詳細なフォーマットは、以下のとおりです。

PKM	SASN	EAX	PASN
128	130	132	134 135

PSW Bits 0-63

136 143
24 ビットまたは 31 ビット・モードで作成されるブランチ状態エントリーの場合

A	Bits 33-63 of Branch Address
---	---------------------------------

144 148 151
64 ビット・モードで作成されるブランチ状態エントリーの場合

Bits 0-62 of Branch Address	1
-----------------------------	---

144 151
結果のモードが 24 ビットまたは 31 ビットになるプログラム呼び出し状態エントリーの場合

Called-Space Id.	0	PC Number
------------------	---	-----------

144 148 151
結果のモードが 64 ビットになるプログラム呼び出し状態エントリーの場合

Called-Space Id.	1	PC Number
------------------	---	-----------

144 148 151

Modifiable Area

152 159

All Zeros

160 167

PSW Bits 64-127

168 175

Unpredictable

176 223

バイト 128-175 に含まれるフィールドは、以下のように割り振られています。以下の説明の中で、「呼び出し側プログラムの」という表現は、状態エントリーを作成した BRANCH AND STACK またはスタッキング PROGRAM CALL 命令の実行開始の時点で存在していた値を意味します。

PSW キー・マスク (PKM): バイト 128-129 には、呼び出し側プログラムの PSW キー・マスク (制御レジスター 3 のビット 32-47) が入っています。PSW キー・マスクは、BRANCH AND STACK またはスタッキング PROGRAM CALL により状態エントリーに保

管され、スタッキング PROGRAM CALL により作成されたエントリーをアンスタックする PROGRAM RETURN 命令により、制御レジスターに復元されます。

2次 ASN (SASN): バイト 130-131 には、呼び出し側プログラムの 2 次 ASN (制御レジスター 3 のビット 48-63) が入っています。SASN は、BRANCH AND STACK またはスタッキング PROGRAM CALL により状態エントリーに保管され、スタッキング PROGRAM CALL により作成されたエントリーをアンスタックする PROGRAM RETURN 命令により、制御レジスターに復元されます。

拡張許可指標 (EAX): バイト 132-133 呼び出し側プログラムの拡張許可指標 (制御レジスター 8 のビット 32-47) が入っています。EAX は、BRANCH AND STACK またはスタッキング PROGRAM CALL により状態エントリーに保管され、スタッキング PROGRAM CALL により作成されたエントリーをアンスタックする PROGRAM RETURN 命令により、制御レジスターに復元されます。

1次 ASN (PASN): バイト 134-135 には、呼び出し側プログラムの 1 次 ASN (制御レジスター 4 のビット 48-63) が入っています。PASN は、BRANCH AND STACK またはスタッキング PROGRAM CALL により状態エントリーに保管され、スタッキング PROGRAM CALL により作成されたエントリーをアンスタックする PROGRAM RETURN 命令により、制御レジスターに復元されます。

プログラム状況ワード (PSW): R₁ フィールドが 0 の BRANCH AND STACK 命令により作成されるブランチ状態エントリー、およびプログラム呼び出し状態エントリーでは、バイト 136-143 および 168-175 には呼び出し側プログラムの更新済み PSW が入っています。バイト 136-143 には PSW のビット 0-63、そしてバイト 168-175 には PSW のビット 64-127 が入れられます。したがって、この PSW 中の基本アドレッシング・モード・ビットおよび拡張アドレッシング・モード・ビットは、呼び出し側プログラムのアドレッシング・モードを指定しており、命令アドレスは、この状態エントリーを作成した BRANCH AND STACK またはスタッキング PROGRAM CALL 命令の次の順番に当たっている命令か、または、BRANCH AND STACK またはスタッキング PROGRAM CALL 命令をターゲット命令とする EXECUTE 命令の次の順番に当たっている命令を示します。R₁ フィールドが 0 ではない BRANCH AND STACK 命令により作成されるブランチ状態エン

トリーでは、バイト 136-143 および 168-175 には、呼び出し側プログラムの PSW が入っています。ただし、バイト 136-139 のビット位置 31 の拡張アドレッシング・モード・ビット、バイト 140 のビット位置 0 の基本アドレッシング・モード・ビット、およびバイト 168-175 の命令アドレスは、R₁ フィールドが示している汎用レジスターの内容に指定されている値になります。基本アドレッシング・モード・ビット、拡張アドレッシング・モード・ビット、および命令アドレスがどのように指定されるかについては、第10章、『制御命令』の中の BRANCH AND STACK の定義を参照してください。バイト 136-143 の PER マスクの値は、常に予測不能です。PSW は、BRANCH AND STACK またはスタッキング PROGRAM CALL により状態エントリーに保管され、PROGRAM RETURN により現 PSW として復元されます。ただし、PER マスクは復元されません。PROGRAM RETURN は、現 PSW 中の PER マスクは変更しません。

基本アドレッシング・モード (A): 24 ビットまたは 31 ビット・アドレッシング・モードで作成されたブランチ状態エントリーでは、バイト 148-151 のビット位置 0 には、その状態エントリーを作成した BRANCH AND STACK 命令の実行終了時点での、基本アドレッシング・モード・ビット (PSW のビット 32) が入っています。基本アドレッシング・モード・ビットは、BRANCH AND STACK によりバイト 148-151 のビット位置 0 に保管されます。BRANCH AND STACK は、PSW 内の基本アドレッシング・モード・ビットは変更しません。

ブランチ・アドレス: 24 ビットまたは 31 ビット・アドレッシング・モードで作成されたブランチ状態エントリーでは、バイト 148-151 のビット 1-31 には、その状態エントリーを作成した BRANCH AND STACK の実行終了時点での、PSW 内の命令アドレスのビット 33-63 が入っており、バイト 144-147 の内容は予測不能です。64 ビット・アドレッシング・モードで作成されたブランチ状態エントリーでは、バイト 144-151 には、命令アドレスのビット 0-62 の右側に 1 個の 1 を付加したものが入っています。命令アドレスは、BRANCH AND STACK により、バイト 148-151 または 144-151 に保管されます (どちらになるかはアドレッシング・モードによって決まります)。R₂ フィールドが 0 でない BRANCH AND STACK 命令では、ブランチが生じるので、バイト 148-151 または 144-151 の命令アドレスはブランチ・アドレスになります。R₂ フィールドが 0 である BRANCH AND STACK 命令は、ブランチなしで実行されるので、バイト 148-151 または 144-151 中の命令アドレスは、BRANCH AND

STACK 命令の次の順番に当たっている命令か、
BRANCH AND STACK 命令をターゲット命令として
実行された EXECUTE 命令の次の順番に当たっている
命令です。

呼び出し先スペース ID: プログラム呼び出し状態エン
トリーでは、バイト 144-147 には呼び出し先スペース
ID (CSI) が入っています。CSI は、スタッキング
PROGRAM CALL により状態エントリーに保管されま
す。スペース切り替えを伴う PROGRAM CALL 操作
の場合は、CSI のバイト 0 および 1 (状態エントリーの
バイト 144 および 145) には、PROGRAM CALL 命令
が制御レジスター 4 に入れた新しい 1 次 ASN が入っ
ており、CSI のバイト 2 および 3 (状態エントリーのバ
イト 146 および 147) には、PROGRAM CALL 命令が
制御レジスター 5 に入れたアドレスにある新しい 1 次
ASTE 中の ASTE シーケンス番号 (ASTESN) の右
端 2 バイトが入っています。現 1 次への PROGRAM
CALL 操作の場合は、CSI はすべて 0 です。

PC 番号: プログラム呼び出し状態エントリーでは、
バイト 148-151 のビット位置 12-31 には、そのエン
トリーを作成したスタッキング PROGRAM CALL 命令
で使用された PC 番号が入っています。スタッキング
PROGRAM CALL は、バイト 148-151 のビット位置
12-31 には PC 番号を入れ、ビット位置 1-11 には 0 を
入れ、そして、ビット位置 0 には、結果のアドレッシ
ング・モードが 24 ビットまたは 31 ビット・モードなら
0、結果のアドレッシング・モードが 64 ビット・モー
ドなら 1 を入れます。

変更可能域: バイト 152-159 は、MODIFY
STACKED STATE により設定されるフィールドです。
BRANCH AND STACK およびスタッキング
PROGRAM CALL は、バイト 152-159 にはすべて 0
を入れます。

バイト 160-167 には、BRANCH AND STACK および
スタッキング PROGRAM CALL によりすべて 0 が入
られます。

バイト 176-223 の内容は予測不能です。

スタッキング・プロセス

スタッキング・プロセスは、BRANCH AND STACK
またはスタッキング PROGRAM CALL 操作の一環と
して実行されます。このプロセスでは、新しいリンケー
ジ・スタック状態エントリー用のスペースの位置が突き
止められ、そのエントリーが作成され、先行エントリー
の次エントリー・サイズ・フィールドが更新され、制御
レジスター 15 中のリンケージ・スタック・エントリー
・アドレスが更新されて、新規エントリーが現リンケ
ージ・スタック・エントリーになります。

スタッキング・プロセスを正常に実行するには、DAT
がオンで、かつ CPU が 1 次スペース・モードまたはア
クセス・レジスター・モードになっていなければなりま
せん。さもないと、特殊操作例外が認識され、操作は抑
止されます。

上記で述べた条件を除いて、スタッキング・プロセス
は、現 PSW のビット 31、32、16、および 17 に指定
されている現行アドレッシング・モードおよび変換モー
ドとは関係なく、実行されます。スタッキング・プロセ
スで使用されるすべてのアドレスは、常に 64 ビットの
ホーム仮想アドレスです。

スタッキング・プロセスにおいて、別のアドレスとの間
で一定の値を加算または減算することによりアドレスが
形成される場合は、そのアドレスのビット位置 0 からの
繰り上がり、またはビット位置 0 への繰り下がりは無視
されます。

スタッキング・プロセスにおいて、変換後に、構成内で
使用可能になっていない位置を示すアドレスを使用して
取り出しましたは格納が行われると、アドレッシング例外
が認識され、操作は抑止されます。

スタッキング・プロセスで行われるアクセスには、キー
制御保護は適用されませんが、ページ保護および低アド
レス保護は適用されます。記憶保護例外が起きた場合
は、操作は抑止されます。

新規エントリー用のスペースの位置決め

現リンケージ・スタック・エントリーを突き止めるに
は、制御レジスター 15 中のリンケージ・スタック・
エントリー・アドレスが使用されます。制御レジスター
15 のビット 0-60 の右側に 3 個の 0 を付加したもの
が、現リンケージ・スタック・エントリーのエントリー
記述子の最左端のバイトの 64 ビット・ホーム仮想アド
レスを形成します。

現リンケージ・スタック・エントリーのエントリー記述
子の第 1 ワードが、64 ビット・ホーム仮想アドレスを

使用して取り出されます。この取り出しの目的は、このワードの中のセクション ID フィールドと残余フリー・スペース・フィールドを入手することにあります。このワード内のアンスタック抑止ビットとエントリー・タイプ・フィールドは検査されません。

残余フリー・スペース・フィールド内の 16 ビットの符号なし 2 進数値 (エントリー記述子のビット 16-31) が、作成するリンケージ・スタック・エントリーのサイズ (バイト数) と比較されます。状態エントリーのサイズは、296 バイトです。このフィールドの値が、作成するエントリーのサイズに等しいかそれより大きい場合は、『新規エントリーの形成』の説明に従って処理が続行されます。そうでない場合は、以下に述べるように処理が進められます。

現リンケージ・スタック・エントリー内の残余フリー・スペース・フィールドが、現リンケージ・スタック・セクション内に新規エントリーを作成するための十分な使用可能スペースがないことを示している場合は、現セクションのトレーラー・エントリーの最初のダブルワードが取り出されます。このダブルワードを取り出すためのアドレスは、次のように決定されます。つまり、制御レジスター 15 の内容から形成されたアドレスに 8 を加算して、現エントリーのエントリー記述子の後の最初のバイトのアドレスが決定され、さらに、そのアドレスに現エントリーの残余フリー・スペース・フィールドの内容を加算して、トレーラー・エントリーの最初のバイトが決定されます。この加算に使用する残余フリー・スペース値は、8 の倍数でなければなりません。そうでないときは、スタック指定例外が認識され、操作は無効化されます。

トレーラー・エントリーの正方向セクション妥当性ビット (ビット 63) が 0 である場合は、スタック一杯例外が認識され、操作は無効化されます。そうでない場合は、トレーラー・エントリー内の正方向セクション・ヘッダー・アドレスを使用して、次のリンケージ・スタック・セクション内のヘッダー・エントリーが突き止められます。トレーラー・エントリーのビット 0-60 の右側に 3 個の 0 を付加したものが、次のセクション内のヘッダー・エントリーのエントリー記述子の左端バイトの 64 ビット・ホーム仮想アドレスを形成します。

次のリンケージ・スタック・セクション内のヘッダー・エントリーのエントリー記述子の第 1 ワードが、取り出されます。この取り出しの目的は、このワードの中のセクション ID フィールドと残余フリー・スペース・フィールドを入手することにあります。このワード内のアンスタック抑止ビットとエントリー・タイプ・フィールドは検査されません。

次のリンケージ・スタック・セクション内のヘッダー・エントリーの残余フリー・スペース・フィールドの値が、作成するエントリーのサイズ (バイト数) と比較されます。このフィールドの値が、作成するエントリーのサイズに等しいかそれより大きい場合は、以下のことが行われます。

- リンケージ・スタック・エントリー・アドレス (制御レジスター 15 のビット 0-60) が、次のリンケージ・スタック・セクション内のヘッダー・エントリーのビット位置 0-60 に逆方向スタック・エントリー・アドレスとして入れられ、ビット位置 61 および 62 には 0 が入れられます。
- 次のセクション内のヘッダー・エントリーの逆方向スタック・エントリー妥当性ビット (ビット 63) が、1 に設定されます。
- 次のセクション内のヘッダー・エントリーのエントリー記述子の 64 ビット・ホーム仮想アドレスのビット 0-60 が、制御レジスター 15 のビット位置 0-60 に入れられ、制御レジスター 15 のビット位置 61-63 には 0 が入れられます。したがって、次のセクション内のヘッダー・エントリーが現リンケージ・スタック・エントリーになり、次のセクションが現リンケージ・スタック・セクションになります。
- 処理は、『新規エントリーの形成』の説明に従って続行されます。

次のセクション内のヘッダー・エントリーの残余フリー・スペース・フィールドの値 (次のセクションが現セクションになる前の) が、形成されるリンケージ・スタック・エントリーのサイズより小さい場合は、スタック指定例外が認識され、操作は無効化されます。

新規エントリーの形成

現リンケージ・スタック・エントリー内の残余フリー・スペース・フィールドが、現リンケージ・スタック・セクション内に新規エントリーを形成するための十分な使用可能スペースがあることを示している場合は、現エントリーのエントリー記述子の直後の位置を始点として新規エントリーが形成されます。

新規エントリーは状態エントリーです。新規エントリーのバイト 0-127 には、汎用レジスター 0-15 の内容が、レジスター番号の昇順に従って保管されます。新規エントリーのバイト 224-287 には、アクセス・レジスター 0-15 の内容が、レジスター番号の昇順に従って保管されます。新規エントリーのバイト 128-129、130-131、132-133、および 134-135 には、それぞれ、新規エントリーの PSW キー・マスク (制御レジスター 3 のビット 32-47)、2 次 ASN (制御レジスター 3 のビット

48-63)、拡張許可指標 (制御レジスター 8 のビット 32-47)、および 1 次 ASN (制御レジスター 4 のビット 48-63) が格納されます。新規エントリーのバイト 136-143 および 168-175 には、命令アドレスが更新された現 PSW が格納されます。バイト 136-143 には PSW のビット 0-63、そしてバイト 168-175 には PSW のビット 64-127 が入れられます。ただし、PER マスクの値 (格納された PSW のビット 1) は、予測不能です。さらに、実行されている命令が、R₁ フィールドが 0 ではない BRANCH AND STACK 命令である場合は、新規エントリーのバイト 136 と 140 にそれぞれ格納される拡張アドレッシング・モード・ビットと基本アドレッシング・モード・ビット、および新規エントリーのバイト 168-175 に格納される命令アドレスは、R₁ フィールドが示している汎用レジスターの内容により指定されているものです。

命令が PROGRAM CALL である場合は、新規エントリーのバイト 144-147 には、呼び出し先スペース ID が格納されます。スペース切り替えを伴う PROGRAM CALL 命令の場合は、呼び出し先スペース ID は、この命令で使用されているエントリー・テーブル・エントリー内の 2 バイト ASN (バイト 10 および 11) の後に、ASN で指定されている ASN 第 2 テーブル・エントリー内の ASTE シーケンス番号のバイト 2 および 3 を付加したもので、バイト 2 および 3 はビット 176-191 を占めます。現 1 次への PROGRAM CALL 操作を行う命令の場合は、呼び出し先スペース ID はすべて 0 です。

命令が、24 ビットまたは 31 ビット・アドレッシング・モードの BRANCH AND STACK である場合は、現 PSW の基本アドレッシング・モード・ビットが、状態エントリーのバイト 148 のビット位置 0 に格納され、ブランチ・アドレス (ブランチを伴わない命令の場合は更新済み命令アドレス) のビット 33-63 が、バイト 148-151 のビット位置 1-31 に格納され、バイト 144-147 の内容は予測不能です。64 ビット・アドレッシング・モードでは、ブランチ・アドレスまたは更新済み命令アドレスのビット 0-62 の右側に 1 個の 1 を付加したものが、状態エントリーのバイト 144-151 に格納されます。

命令が PROGRAM CALL である場合は、使用された 20 ビットの PC 番号が、バイト 148-151 のビット位置 12-31 に格納されます。PROGRAM CALL の実行後の結果のアドレッシング・モードが、24 ビットまたは 31 ビット・アドレッシング・モードである場合は、バイト

148 のビット位置 0 に 0 が格納されます。結果のアドレッシング・モードが 64 ビット・アドレッシング・モードである場合は、バイト 148 のビット位置 0 には、0 ではなく 1 が格納されます。バイト 148-151 のビット位置 1-11 には、結果のアドレッシング・モードに関係なく 0 が格納されます。

新規エントリーのバイト 152-167 には、0 が格納されます。バイト 176-223 の内容は予測不能です。

新規エントリーのバイト 288-295 は、エントリー記述子です。このエントリー記述子のアンスタック抑止ビット (ビット 0) は、0 に設定されます。このエントリー記述子のエントリー・タイプ・フィールド (ビット 1-7) には、実行されている命令が BRANCH AND STACK であれば、2 進数コード 0001100 が格納されます。命令が PROGRAM CALL である場合は、2 進数コード 0001101 が格納されます。このエントリー記述子のセクション ID フィールド (ビット 8-15) には、現リンケージ・スタック・エントリーのセクション ID フィールドの値が格納されます。このエントリー記述子の残余フリー・スペース・フィールドには、現エントリーの残余フリー・スペース・フィールドの値から新規エントリーのサイズ (バイト数) を引いた値が、格納されます。このエントリー記述子の次エントリー・サイズ・フィールド (ビット 32-47) およびビット位置 48-63 には、0 が格納されます。

他の CPU からは、新規エントリーへの格納はワード単位で行われているように見えます。格納が行われる順序は予測不能です。

現エントリーの更新

現エントリーの次エントリー・サイズ・フィールドには、新しいリンケージ・スタック・エントリーのサイズ (バイト数) が格納されます。現エントリーのその他の部分は変更されません。

現エントリーおよび新規エントリーへの格納の順序は、予測不能です。

制御レジスター 15 の更新

制御レジスター 15 のビット位置 0-60 (リンケージ・スタック・エントリー・アドレス) には、新しいリンケージ・スタック・エントリーのエントリー記述子の 64 ビット・ホーム仮想アドレスのビット 0-60 が入れられます。制御レジスター 15 のビット位置 61-63 には、0 が入れられます。したがって、新規エントリーは現リンケージ・スタック・エントリーになります。

スタッキング・プロセスでの例外の認識

スタッキング・プロセス中に検出される可能性がある例外と、それぞれの優先順位については、BRANCH AND STACK および PROGRAM CALL の定義の中で説明されています。

プログラミング上の注意: BRANCH AND STACK および PROGRAM CALL の実行中に例外が認識されたときは、無効化または抑止が起こります。したがって、例外が認識された場合は、スタッキング・プロセスは、リンケージ・スタック・エントリーへの格納も、制御レジスター 15 の内容の変更も行いません。

アンスタッキング・プロセス

アンスタッキング・プロセスは、PROGRAM RETURN 操作の一環として実行されます。このプロセスでは、リンケージ・スタック内の最後の状態エントリーの位置が突き止められ、そのエントリー内の情報の一部が CPU レジスターに復元され、先行エントリーの次エントリー・サイズ・フィールドが更新され、制御レジスター 15 中のリンケージ・スタック・エントリー・アドレスが更新されて、先行エントリーが現リンケージ・スタック・エントリーになります。アンスタッキング・プロセスの一環として最後の状態エントリーを見つける操作は、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE 操作の一部としても行われます。

アンスタッキング・プロセスを正常に実行するには、DAT がオンで、かつ CPU が 1 次スペース・モードまたはアクセス・レジスター・モードになっていなければなりません。さもないと、特殊操作例外が認識され、操作は抑止されます。ただし、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、または MODIFY STACKED STATE の一環として行われるアンスタッキング・プロセスの場合は、CPU は、1 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードのどれであっても構いません。

上記で述べた条件を除いて、アンスタッキング・プロセスは、現 PSW のビット 31、32、16、および 17 に指定されている現行アドレッシング・モードおよび変換モードとは関係なく、実行されます。アンスタッキング・プロセスで使用されるすべてのアドレスは、常に 64 ビットのホーム仮想アドレスです。

アンスタッキング・プロセスにおいて、別のアドレスとの間で一定の値を加算または減算することによりアドレスが形成される場合は、そのアドレスのビット位置 0 か

らの繰り上がり、またはビット位置 0 への繰り下がりは無視されます。

アンスタッキング・プロセスにおいて、変換後に、構成内で使用可能になっていない位置を示すアドレスを使用して取り出したり格納が行われると、アドレッシング例外が認識され、操作は抑止されます。

アンスタッキング・プロセスで行われるアクセスには、キー制御保護は適用されませんが、ページ保護および低アドレス保護は適用されます。記憶保護例外が起きた場合は、操作は抑止されます。

現エントリーの位置決めとヘッダー・エントリーの処理

現リンケージ・スタック・エントリーを突き止めるには、制御レジスター 15 中のリンケージ・スタック・エントリー・アドレスが使用されます。制御レジスター 15 のビット 0-60 の右側に 3 個の 0 を付加したものが、現リンケージ・スタック・エントリーのエントリー記述子の最左端のバイトの 64 ビット・ホーム仮想アドレスを形成します。

現リンケージ・スタック・エントリーのエントリー記述子の第 1 ワードが、64 ビット・ホーム仮想アドレスを使用して取り出されます。エントリー記述子のビット 1-7 にあるエントリー・タイプ・コードが 2 進数 0001001 ではない場合、つまりこのエントリーがヘッダー・エントリーではないことを示している場合は、5-76 ページの『状態エントリーかどうかの検査』の説明に従って処理が続行されます。そうでない場合は、処理は以下に示すように続行されます。

現リンケージ・スタック・エントリー内のエントリー・タイプ・コードが、ヘッダー・エントリーを示す 2 進数 0001001 であるときは、次の処理はどの命令が実行されているかによって異なります。PROGRAM RETURN 操作の一環としてアンスタッキング・プロセスが実行されており、現エントリーのエントリー記述子内のアンスタック抑止ビット (ビット 0) が 1 のときは、スタック操作例外が認識され、操作は無効化されます。

EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、または MODIFY STACKED STATE の一環としてアンスタッキング・プロセスが実行されているときは、アンスタック抑止ビットは無視されます。

アンスタック抑止ビットを原因とする例外が起きていなければ、現リンケージ・スタック・エントリー (ヘッダー・エントリー) の最初のダブルワードが取り出されず。このダブルワードのアドレスは、現エントリーのエ

ントリー記述子のアドレスから 8 を引くことにより求められます。

現エントリーの逆方向スタック・エントリー妥当性ビット (ビット 63) が 0 の場合は、スタック空例外が認識され、操作は無効化されます。そうでない場合は、現エントリー内の逆方向スタック・エントリー・アドレスを使用して、ここで指定エントリーとして参照されているリンケージ・スタック・エントリーが突き止められます。現エントリーのビット 0-60 の右側に 3 個の 0 を付加したものが、指定されたエントリーのエントリー記述子の最左端バイトの 64 ビット・ホーム仮想アドレスを形成します。

このアンスタッキング・プロセスの定義では、指定されたリンケージ・スタック・エントリーは、先行するリンケージ・スタック・セクション内の、トレーラー・エントリーを除く最後のエントリーであることが前提になります。この前提は、明示的に記述されていない処理を暗黙に含むものではありません。

指定されたエントリーのエントリー記述子の第 1 ワードが取り出されます。このエントリー記述子内のエントリー・タイプ・コードが 2 進数 0001001 ではない場合、つまりこのエントリーがヘッダー・エントリーではないことを示している場合は、以下のことが行われます。

- PROGRAM RETURN 操作の一環としてアンスタッキング・プロセスが実行された場合は、指定されたエントリーのエントリー記述子の 64 ビット・ホーム仮想アドレスのビット 0-60 が、制御レジスター 15 のビット位置 0-60 に入れられ、制御レジスター 15 のビット位置 61-63 には 0 が入れられます。したがって、指定されたエントリーは現リンケージ・スタック・エントリーになり、先行セクション (上記の前提に基づく) は現リンケージ・スタック・セクションになります。EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、または MODIFY STACKED STATE の一環としてアンスタッキング・プロセスが実行される場合は、制御レジスター 15 の内容は変更されませんが、命令の定義のその他の部分の実行中に、指定されたエントリーは一時的に現リンケージ・スタック・エントリーと呼ばれます。
- 処理は、『状態エントリーかどうかの検査』の説明に従って続行されます。

指定されたエントリー内のエントリー・タイプ・コードが、ヘッダー・エントリーを示す 2 進数 0001001 である場合は、スタック指定例外が認識され、操作は無効化されます。

状態エントリーかどうかの検査

現リンケージ・スタック・エントリー内のエントリー・タイプ・コードが、このエントリーがヘッダー・エントリーではないことを示している場合は、そのコードが、状態エントリーに割り当てられているコードである 2 進数 0001100 または 0001101 であるかどうかを検査されます。

現リンケージ・スタック・エントリーが状態エントリーである場合は、次の処理はどの命令が実行されているかによって異なります。PROGRAM RETURN 操作の一環としてアンスタッキング・プロセスが実行されているときは、『情報の復元』の説明に従って処理が続行されます。EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、または MODIFY STACKED STATE の一環としてプロセスが実行されているときは、プロセスは完了します。つまり、アンスタッキング・プロセスの一環としてこれ以上処理が行われることはありません。

現リンケージ・スタック・エントリーが状態エントリーではない (したがって必然的にヘッダー・エントリーでもない) 場合は、スタック・タイプ例外が認識され、操作は無効化されます。

情報の復元

アンスタッキング・プロセスの残りの部分は、PROGRAM RETURN 操作でのみ実行されます。

現リンケージ・スタック・エントリーは、状態エントリーです。このエントリー内のアンスタック抑止ビットが 1 の場合は、スタック操作例外が認識され、操作は無効化されます。

アンスタック抑止ビットを原因とする例外が起きていなければ、現リンケージ・スタック・エントリーの内容の一部が、CPU レジスターに復元されます。制御レジスター 2-14 およびアクセス・レジスター 2-14 の内容は、それぞれ、スタッキング・プロセスにより現エントリーに保管されたときの取り出し元のレジスターに復元されます。現エントリー内のエントリー・タイプ・コードが、プログラム呼び出し状態エントリーを示す 2 進数 0001101 であるときは、制御レジスター 3 の中の PSW キー・マスクと 2 次 ASN、制御レジスター 8 の中の拡張許可指標、および制御レジスター 4 の中の 1 次 ASN も、同様に復元されます。この復元では、制御レジスター 4 の中の許可指標および制御レジスター 8 の中のモニター・マスクは変更されません。(許可指標は、アンスタッキング・プロセスの後で実行される PROGRAM RETURN の一環として変更されることがあります。) エントリー・タイプ・コードが、ブランチ状態エントリ

ーを示す 2 進数 0001100 であるときは、現エントリー内の PSW キー・マスク、2 次 ASN、拡張許可指標、および 1 次 ASN は無視され、制御レジスターのすべての内容が無変更のまま残されます。現エントリーがブランチ状態エントリーまたはプログラム呼び出し状態エントリーであるときは、現エントリーのバイト 136-143 および 168-175 から、それぞれ、現 PSW のビット 0-63 および 64-127 が復元されます。ただし、PER マスクは復元されません。現 PSW の中の PER マスクは、無変更のまま残されます。現エントリーのバイト 144-159 および 160-167 は無視されます。

他の CPU からは、現エントリーからの取り出しはワード単位で行われているように見えます。取り出しが行われる順序は予測不能です。

先行エントリーの更新

先行するリンケージ・スタック・エントリーのエントリー記述子の次エントリー・サイズ・フィールド (ビット 32-47) には、0 が格納されます。先行エントリーのその他の部分は変更されません。先行エントリーのエントリー記述子のアドレスは、現エントリーのエントリー記述子のアドレスから、現エントリーのサイズ (バイト数) を引くことにより求められます。

先行エントリーへの格納および現エントリーからの取り出しの順序は、予測不能です。

制御レジスター 15 の更新

制御レジスター 15 のビット位置 0-60 (リンケージ・スタック・エントリー・アドレス) には、先行リンケージ・スタック・エントリーのエントリー記述子の 64 ビット・ホーム仮想アドレスのビット 0-60 が入れられます。制御レジスター 15 のビット位置 61-63 には、0 が入れられます。したがって、先行エントリーは現リンケージ・スタック・エントリーになります。

アンスタッキング・プロセスでの例外の認識

アンスタッキング・プロセス中に検出される可能性がある例外と、それぞれの優先順位については、PROGRAM RETURN 命令の定義の中で説明されています。EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、および MODIFY STACKED STATE に適用される例外については、それぞれの命令の定義の中で説明されています。

プログラミング上の注意:

1. EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN の実行中に何らかの例外が認識された場合は、無効化または抑止のいずれかが発生します。したがって、例外が認識された場合は、アンスタッキング・プロセスにより、どの CPU レジスターの内容も変更されることはなく (ただし抑止の場合は PSW の中の命令アドレスが更新されます)、また、どのリンケージ・スタック・エントリーへの格納も行われません。
2. PROGRAM RETURN によるアンスタッキング・プロセスでは、PSW の中の PER マスクは復元されません。これは、関連の BRANCH AND STACK または PROGRAM CALL 命令の実行後、PROGRAM RETURN 命令の実行前に、PER がオンまたはオフに切り替えられた場合に、その位置が打ち消されないようにするためです。スペース切り替えを伴う PROGRAM CALL または PROGRAM RETURN の場合は、必要があれば、スペース切り替えイベントを、PER をオンまたはオフに切り替えるための仕事として使用できます。

記憶機構参照の順序

以下のセクションでは、CPU プログラムの中で操作がオーバーラップしたり実行が断片的になった結果として、記憶機構内にどのような影響が現れるかについて説明します。以下に述べる影響のほとんどは、複数の CPU またはチャネル・プログラムが同時に実行され、共通の記憶位置にアクセスしている場合のみ見られるものです。したがって、これらの影響のほとんどは、プログラムが他の CPU またはチャネル・プログラムと相互に作用する場合以外は、考慮に入れる必要はありません。

以下のセクションで説明する影響のいくつかは、他の CPU またはチャネル・プログラムとの相互関係とは無関係に発生します。したがって、この種の影響は容易に認識することができます。また、これらの影響は、事前取り出しされる命令と、単一の命令のオーバーラップするオペランドに関連しています。これらの影響については、『概念上の順序』および 5-79 ページの『仮想記憶域参照のためのインターロック』で説明されています。

概念上の順序

実モード、1 次スペース・モード、または 2 次スペース・モードでは、CPU は、概念的には一度に 1 つずつ命令を実行します。つまり、1 つの命令の実行が終わってから、次の命令が実行されます。ブランチ成功の結果指定された命令は、そのブランチの実行が終わった後で実行されます。同様に、割り込みは、命令と命令の間、または、割り込み可能命令の中では操作単位と操作単位の間で発生します。

上記の処理により暗黙に発生するイベントの順序は、概念上の順序と呼ばれることがあります。

プログラムとしては、命令実行の各操作は順次に行われ、前の操作が終わった後で、現行の命令が取り出された上で、現行操作が始まるように見えます。記憶機構のインプリメンテーション特性や、命令実行と記憶機構アクセスのオーバーラップにより、実際の処理順序が違っていても、この外見上の順序は変わりません。最終的な結果は、概念上の順序で操作が行われた場合と同じになります。したがって、ある命令が記憶域内の後続の命令を変更することも可能です。

アクセス・レジスター・モードまたはホーム・スペース・モードでの操作も、他の変換モードの場合と同じですが、唯一の例外として、前の命令の格納タイプ・オペランドである命令は、格納が行われる前に取り出されるように見えることがあります。したがって、ある命令が後続の命令を変更できるという保証はありません。この例外が生じるのは、格納を行う命令と格納される命令のどちらかが、アクセス・レジスター・モードまたはホーム・スペース・モードで実行される場合です。

さらに、変換モードに関係なく、事前取り出しされた命令のコピーが必ずしも破棄されないケースが 1 つあります。それは、取り出しと格納が、同じ実アドレスにマップされる異なる有効アドレスを使用して行われる場合です。このケースについては、5-79 ページの『仮想記憶域参照のためのインターロック』で詳しく説明します。

命令実行における操作のオーバーラップ

複数の操作がオーバーラップすることのない単純なモデルでは、概念上の順序と実際の順序は事実上同じです。しかし、もっと複雑なマシンでは、操作のオーバーラップ、オペランドと結果のバッファリング、および装置間の伝搬遅延に相当する実行時間が原因で、実際の順序が概念上の順序とは大幅に異なることがあります。このようなマシンでは、操作間の依存関係を検知する特殊回路が使用されており、この回路により、操作を生成する CPU から見た最終的な結果が、それらの操作を概念所の順序に従って実行した場合に得られる結果と同じなる

ことが保証されます。しかし、特に制約がない限り、他の CPU およびチャンネル・プログラムから、実際の順序が概念上の順序とは違って見えることがあります。

分割可能な命令実行

通常は、各命令の実行は 1 つの分割不可能なイベントとして発生するものと見なすことができます。しかし、実際の操作では、命令の実行は一連の独立したステップから成っています。命令によっては、オペランドが断片的に取り出されたり格納されたりするため、オペランドの取り出しと結果の格納の間にある程度の遅延が生じることがあります。その結果、他の CPU およびチャンネル・プログラムには、中間結果や部分的にしか完了していない結果が見えることがあります。

プログラムと他の CPU 上の操作、またはプログラムとチャンネル・プログラムが相互に作用し合うときは、そのプログラムでは、1 つの操作が一連の記憶域参照から成っていることがあること、そして個々の記憶域参照がさらに一連のアクセスから成っていることがあること、そしてこれらのアクセスの概念上の順序と外見上の順序が異なる場合があることを、考慮に入れる必要があります。

命令実行に関連した記憶域参照のタイプには、命令の取り出し、ART テーブルと DAT テーブルの取り出し、および記憶域オペランド参照があります。記憶域参照の順序に関する説明では、ASN 変換、PC 番号変換、トレース、およびリンケージ・スタックのスタッキング・プロセスおよびアンスタッキング・プロセスを実行するための記憶域へのアクセスを、記憶域オペランド参照と見なすことにします。

プログラミング上の注意: CPU の実行順序は、以下のような点で単純な概念上の定義とは異なることがあります。

- CPU 自体から見た場合、概念上の格納の時点または事前取り出しされた命令の実行時点で該当のモードが存在するかどうかに関係なく、命令はアクセス・レジスター・モードまたはホーム・スペース・モードで事前取り出しされるように見えることがあります。また、異なる有効アドレスが使用されているときにも、命令が事前取り出しされるように見えることがあります。(5-79 ページの『仮想記憶域参照のためのインターロック』を参照してください。)
- チャンネル・プログラムおよび他の CPU から見た場合、1 つの命令の実行が一連の断片的なステップとして行われるように見えることがあります。これについては、以下のセクションで個々の記憶域参照のタイプごとに説明します。

- チャンネル・プログラムおよび他の CPU から見た場合、1 つの命令に関連した記憶域オペランド・アクセスが、必ずしも概念上の順序に従って行われていないように見えることがあります。(5-88ページの『オペランド・アクセス間の関係』を参照してください。)
- チャンネル・プログラムから見た場合、ある種の異常な状況において、記憶域の内容が、いったん変更された後で再び元の値に戻るように見えることがあります。(5-22ページの『DAT 関連アクセス例外の場合の記憶域の変更と復元』を参照してください。)

仮想記憶域参照のためのインターロック

上記の幾つかのセクションで説明したように、幾つかの例外を除いて、CPU の操作は、その CPU 自体にとっては順次に行われるように見えます。つまり、一般に、1 つの命令による格納が完了してから、次の命令とそのオペランドが取り出されているように見えます。互いにオーバーラップするマシンでは、共通の記憶位置へのアクセスを検出するインターロック回路を使用して、この外見上の順序が維持されます。

記憶域の内容を変更し、複数のオペランドを持つ命令の場合、通常、命令定義は、記憶域内でオペランドがオーバーラップするときに得られる結果を記述しており、したがって、この定義は格納と取り出しの順序に基づいているものと考えられます。オペランドのオーバーラップが存在するかどうかは、インターロック回路を使用して判別されます。

このセクションの目的は、どのような場合に、マシンが順次に作動しているように見える必要があるのか、そして、どのような場合に、単一命令のオペランドがオーバーラップしているもの見なされなければならないか、またはそのように見なされなければならないかを、定義することにあります。

有効アドレスを比較することにより、ある程度は適正な操作が達成されます。この定義では、「有効アドレス」という用語は、仮想アドレスか、実アドレスか、または絶対アドレスかに関係なく、変換の前のアドレスを意味するものとします(変換が行われる場合)。2 つの有効アドレスが同じ値を持っている場合、一方が実アドレスであっても、また異なるアドレス・スペース内にあるものであっても、それらの有効アドレスは同じであると見なされます。

2 つの仮想有効アドレスの値は、必ずしも、それらのアドレスが同じ記憶位置を指しているかどうかを示すとは限りません。アドレス変換テーブルは、異なる有効アドレスが同じ実アドレスにマップするようにセットアップされることも、異なるアドレス・スペース内にある同じ有効アドレスがそれぞれ異なる実アドレスにマップするようにセットアップされることもあります。

仮想記憶域参照用のインターロックが考慮される状況としては、1 つの命令の記憶域参照が他の命令に影響を与える場合と、単一命令の複数記憶域参照の場合の、2 つがあります。

命令間のインターロック

CPU 自体から見た場合、各命令用のオペランドを見つけるための記憶域アクセスは、使用される有効アドレスには関係なく、概念上の順序で生じるように見えます。つまり、1 つの命令のオペランド格納が完了してから、次の命令のオペランド取り出しが始まるように見えます。命令取り出しの場合に、1 つの命令についてのオペランド格納が、必ず次の命令の取り出しの前に完了しているように見えるのは、そのオペランド格納と命令取り出しに同じ有効アドレスが使用されているときのみであり、したがって、実モード、1 次スペース・モード、または 2 次スペース・モードのときのみです。

ある命令が、直接的にまたは EXECUTE を使用して、概念上その次の命令が実行されることになっている主記憶位置の内容を変更し、かつ、結果を格納するときと命令を取り出すときにそれぞれ異なる有効アドレスを使用してその記憶位置が指定された場合は、格納より前に命令の取り出しが行われるように見えることがあります。格納を行う命令またはそれに続く命令が、アクセス・レジスター・モードまたはホーム・スペース・モードで実行されるときは、値を格納するために使用された有効アドレスと、命令を取り出すために使用された有効アドレスが同じであっても、記憶域の内容に対する変更は必ず認識されるとは限りません。途中で介入する操作が原因で事前取り出しされた命令が破棄された場合は、更新後の値が認識されます。どのようなときに事前取り出しされた命令を破棄する必要があるかの定義は、5-81ページの『命令取り出し』に示されています。

記憶キーに対する変更は、関連の記憶域ブロックに対する概念上次の参照が、仮想アドレス、実アドレス、または絶対アドレスのどれを使用して行われたかに関係なく、その参照より前に完了しているように見えます。同様に、記憶域ブロックに対する概念上前の順番に当たっている参照は、そのブロック用のキーが変更または検査される時点では、すでに完了しているように見えます。

単一命令内のインターロック

記憶域の内容を変更し、複数のオペランドを持つ命令の場合、通常、命令定義は、記憶域内でオペランドがオーバーラップするときに得られる結果を記述します。この結果は、通常、記憶域アクセスの順序に基づいて定義されます。つまり、格納タイプのオペランドの結果は、他のオペランドの一部が取り出される前に、記憶域に入れられるように見えることが必要です。この定義が適用されるのは、格納アクセスと取り出しアクセスが、同じ有効アドレスおよび同じ有効スペース指定により指定される場合です。

アクセス・レジスター・モードで複数のアドレス・スペースが関与する場合に、2つのスペースが同じであるかどうかを判別するためにマシンが使用する値を、「有効スペース指定」と呼びます。アクセス・レジスター・モードでは、各記憶域オペランド・アドレスに関連した32ビットのアクセス・リスト・エントリー・トークン(ALET)の値が、有効スペース指定と呼ばれています。Bフィールドに0が指定されているときは、有効スペース指定にはすべて0の値が使用されます。有効スペース指定が異なる場合は、両方のALETが同じアドレス・スペース制御エレメント値にマップされる場合でも、2つのスペースは異なるものと見なされます。

格納アクセスと取り出しアクセスが、それぞれ異なる有効スペース指定を使用して指定されるか、または異なる有効アドレスを使用して指定された場合は、オペランド取り出しがオペランド格納の前に行われるように見えることがあります。

5-81ページの図5-11は、オーバーラップのさまざまなケースと、それぞれの結果を要約して示しています。また、各ケースについて、MOVE LONG (MVCL) が条件コード3を設定するかどうかも示されています。

有効スペース指定はALBエントリーにより表すことができるので、ALBエントリーを比較することによって、2つの有効スペース指定が同じかどうかをテストできます。プログラムが、連結されている有効なARTテーブル・エントリーを変更し、その後で、PURGE ALB命令またはALBをページするCOMPARE AND SWAP AND PURGEを実行しなかった場合は、同一である2つの有効スペース指定がALB内に異なる表現を持つことになり、オペランドのオーバーラップが認識できないことがあります。有効スペース指定が異なるときは、ALBの使用によりオーバーラップが認識されることはありません。

プログラミング上の注意: 複数のアドレスを使用して1つの主記憶位置にアクセスするには、幾つかの方法があります。

- 1つのアドレス・スペース内の複数のアドレス、または複数のアドレス・スペース内の複数のアドレス(実スペース指定により指定されている実アドレスも含む)が、単一の実アドレスにマップされるように、DATテーブルをセットアップできます。
- 論理アドレス、命令アドレス、および仮想アドレスの変換を変更できます。そのためには、制御レジスターにDATパラメーターをロードするか、PSW内のアドレス・スペース制御ビットを変更するか、あるいは、論理アドレスおよび命令アドレスの場合にDATをオンまたはオフに切り替えます。
- アクセス・レジスターモードでは、各アクセス・レジスターを使用して異なるアドレス・スペースを選択できます。さらに、命令の取り出しとEXECUTEのターゲットには1次アドレス・スペースが選択されます。
- STORE USING REAL ADDRESSは、実アドレスを使用して格納を行います。
- 他の幾つかの命令も、実アドレスを使用します(上記のケース1の場合のように論理アドレスが実スペース指定により変換されていなくても構いません)。また、MOVE TO PRIMARY および MOVE TO SECONDARY 命令は、2つのアドレス・スペースにアクセスします。
- 割り込みのための情報の格納および取り出しを目的とする記憶域アクセスは、実アドレスを使用して行われ、状況記憶機能のための記憶域アクセスは絶対アドレスを使用して行われ、さらに、プログラムによるアクセスは仮想アドレスを使用して行われます。
- 実アドレスから絶対アドレスへのマッピングは、SET PREFIX 命令またはリセットにより変更できます。
- チャンネル・プログラムは絶対アドレスを使用し、CPUは実アドレスまたは仮想アドレスを使用して、特定の主記憶位置にアクセスできます。
- 他のCPUは1つのタイプのアドレスを使用し、このCPUは別のタイプのアドレスを使用して、特定の主記憶位置にアクセスできます。

インターロックに関するこのセクションの主な目的は、上記のケース1、3、および4がもたらす影響について説明することにあります。

ケースの2では、変換パラメーターが変更されると事前取り出しされた命令は破棄され、CPUによる格納の遅延はそのCPU自体には見えないため、影響は現れません。

有効 スペース 指定が 等しいか?	有効 アドレスの 破壊的 オーバーラップ があるか?	実記憶域内に オペランドの 破壊的 オーバーラップ があるか?	オーバーラップが 認識されるか?	
			MVCL が CC3 を 設定	オペランド 結果
Yes	No	No	No	No
Yes	No	Yes	No	Unp.
Yes	Yes	No	*	*
Yes	Yes	Yes	Yes	Yes
No	No	No	No	No
No	No	Yes	No	Unp.
No	Yes	No	No	No
No	Yes	Yes	No	Unp.

説明:

* このケースは発生しません。
Unp. オーバーラップが認識されるかどうかは予測不能です。

図 5-11. 単一命令内の仮想記憶域インターロック

ケース 5 では、実アドレスを使用して取り出しを行う命令 (例えば、LOAD REAL ADDRESS は、セグメント・テーブル・エントリーとページ・テーブル・エントリーのほか、領域テーブル・エントリーも取り出すことがある) については、命令間でのオペランド・アクセスしか関係しないので、影響は何も現れません。実アドレスを使用して格納を行う命令 (STORE USING REAL ADDRESS を除く)、または複数のアドレス・スペースにまたがって格納を行う命令 (アクセス・レジスター・モードにあるものを除く) は、すべて事前取り出しされた命令を破棄させるので、影響は現れません。

ケース 6 および 7 は逐次化を引き起こすように定義されている状況であり、結果として事前取り出しされた命令は破棄されます。これらのケースでは、影響は何も現れません。

ケース 8 および 9 の取り扱いには、チャンネル・プログラムおよび他の CPU から見たアクセスが関係します。これについては、以下のセクションで説明します。

命令取り出し

命令取り出しでは、現 PSW 内の命令アドレスが指し示している 1 個、2 個、または 3 個のハーフワードが取り出されます。命令取り出しの一環として、命令の即値フィールドがアクセスされます。ただし、ある命令が、その命令自体が占めている記憶位置にある記憶域オペランドを指定している場合は、その記憶位置は、命令および記憶域オペランドの両方としてアクセスされます。EXECUTE のターゲット命令の取り出しも、命令取り出しの一種と見なされます。

命令のバイトは断片的に取り出されることがあり、必ずしも左から右への方向にアクセスされるとは限りません。このような命令は、1 回の実行中に複数回取り出されることがあります。例えば、オペランドのアドレス可能性をテストするため、または PER イベントを検査するために命令が取り出され、実際の実行時に再度取り出されることがあります。

命令は、必ずしも概念上の実行順序のとおりに取り出されるわけではなく、また、必ずしも実行のたびに取り出されるとも限りません。特に、ある命令の取り出しが、概念上それより前にある命令のための記憶域オペランド参照より前に行われることがあります。命令取り出しは、概念上それより後にあるすべての命令のすべての記憶域オペランド参照より前に発生します。

仮想アドレスを使用して命令が事前取り出しされることがあるのは、関連する DAT テーブル・エントリーが連結されていて有効である場合、または、それらのテーブル・エントリーと置き換えるための適格性を備えたエントリーが TLB 内に存在する場合に限られます。事前取り出しされた命令が実行のために解釈されるのは、その命令の対象が、事前取り出しされたときに目的となっていた仮想アドレスと同一である場合のみです。

事前取り出しが可能な命令の数には制限はなく、また、1 つの記憶位置の内容の複数のコピーを取り出すこともできます。したがって、実行される命令は、必ずしも最後に取り出されたコピーであるとは限りません。チャンネル・プログラムおよび他の CPU により格納が行われたときに、事前取り出しされた命令は、必ずしもそれを反映して変更されるとは限りません。ただし、概念上前にある格納に使用される CPU と有効アドレスが、後で命令の取り出しのときに使用されるものと同じであり、か

つ、CPU が、格納命令が実行される時には実モード、1 次スペース・モード、または 2 次スペース・モードのいずれかであり、後続の命令が実行されるときにもこれらのモードのいずれかである場合は、更新済みの情報が取得されます。有効アドレスが異なる場合、あるいは、格納の実行時または格納の目的となっている命令の実行時のいずれかにおいて、CPU がアクセス・レジスター・モードまたはホーム・スペース・モードになっている場合は、必ずしも更新済みの情報が取得されるとは限りません。ただし、DAT がオンまたはオフに切り替えられた結果として事前取り出しされた命令が破棄されたために、いずれかの実行が実モードになっている場合は、更新済みの情報が取得されます。

事前取り出しされた命令のコピーは、以下の各時点ですべて破棄されます。

- 逐次化機能が実行される時。
- CPU が作動状態に入るとき。
- DAT がオンまたはオフに切り替えられたとき。
- 1 次スペース・モード、2 次スペース・モード、またはアクセス・レジスター・モードで、制御レジスター 1 の中の変換パラメーターが変更されたとき、または、ホーム・スペース・モードで制御レジスター 13 の中の変換パラメーターが変更されたとき。

SET ADDRESS SPACE CONTROL 命令は、1 次スペース・モード、2 次スペース・モード、アクセス・レジスター・モード、およびホーム・スペース・モードの間で切り替えを行うことができ、さらに逐次化を行います。SET ADDRESS SPACE CONTROL FAST 命令も同じモード変更を行うことができますが、逐次化は行いません。

プログラミング上の注意:

1. CPU から見ると、その CPU 自身の命令取り出しが明白であるように見えるのは、異なる有効アドレスが単一の実アドレスにマップしている場合、または CPU がアクセス・レジスター・モードまたはホーム・スペース・モードで実行されている場合です。これについては、5-78ページの『概念上の順序』および 5-79ページの『仮想記憶域参照のためのインターロック』で説明されています。
2. SET ADDRESS SPACE CONTROL FAST 命令以外の手段で PSW のビット 16 および 17 を変更した場合は、逐次化が行われ、事前取り出しされた命令は破棄されます。また、DAT をオンまたはオフに切り替えると、事前取り出しされた命令は破棄されます。したがって、SET ADDRESS SPACE CONTROL FAST 以外の方法で変換モードが変更された場合は、必ず事前取り出しされた命令が破棄されます。

3. 1 つの CPU で行われた命令の事前取り出しは、チャンネル・プログラムまたは他の CPU に以下のような影響を与えます。

1 つの CPU が得意の記憶位置の内容を事前取り出した後で、チャンネル・プログラムまたは別の CPU がその記憶位置の内容を変更し、そして、変更が行われたことを示すフラグを設定することができます。その後、最初の CPU はテストによりその設定されたフラグを検出し、変更された位置にブランチし、そして事前取り出しされた元の内容を実行することができます。

CPU は、チャンネル・プログラムまたは他の CPU が同時に命令を変更している場合に、その命令の一部のビット位置 (すべてではなく) に対する変更を認識することができます。

1 つの CPU が命令を事前取り出した後で、その命令が実行される前に、別の CPU が記憶キーを変更することができます。その結果、最初の CPU は、保護された記憶位置から命令を実行しているように見えます。しかし、実行される命令のコピーは、その記憶位置が保護される前に事前取り出しされているコピーです。

ART テーブルおよび DAT テーブルの取り出し

アクセス・レジスター変換 (ART) テーブル・エントリーには、アクセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、および権限テーブル・エントリーがあります。動的アドレス変換 (DAT) テーブル・エントリーには、領域テーブル・エントリー、セグメント・テーブル・エントリー、およびページ・テーブル・エントリーがあります。これらのエントリーの取り出しは、以下のように行われます。

1. ART テーブル・エントリーは、事前取り出しして ART 索引バッファ (ALB) に入れておくことができます。ALB 内のエントリーは、COMPARE AND SWAP AND PURGE、PURGE ALB、または SET PREFIX 命令、CPU リセット、または SIGNAL PROCESSOR 命令のアーキテクチャー設定指令により消去されるまでは、記憶域から再取り出しすることなく使用できます。DAT テーブル・エントリーは、事前取り出しして変換索引バッファ (TLB) に入れておくことができます。TLB 内のエントリーは、COMPARE AND SWAP AND PURGE、INVALIDATE PAGE TABLE ENTRY、PURGE TLB、または SET PREFIX 命令、CPU リセット、または SIGNAL

PROCESSOR 命令のアーキテクチャ設定指令により消去されるまでは、記憶域から再取り出しすることなく使用できます。ART テーブル・エントリーおよび DAT テーブル・エントリーは、必ずしも、概念上必要とされる順序で取り出されるとは限りません。エントリーが連結されて有効になっているときは、概念上前にある命令の実行中も含めて、いつでも取り出される可能性があります。

2. 他の CPU から見れば、アクセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、および DAT テーブル・エントリーの取り出しは、ワード単位で行われているように見えます。ただし、チャンネル・プログラムから見た場合、エントリーに対する参照は一度に 1 バイトずつアクセスしているように見ることがあります。
3. アクセス・リスト・エントリーまたは ASN 第 2 テーブル・エントリーの各ワードが取り出される順序は、エントリーの最左端のワードが最初に取り出されること以外は、予測不能です。ただし、**BRANCH IN SUBSPACE GROUP** でアクセス・リスト・エントリー・トークンの 16 進数 00000000 が変換されるときは、ASN 第 2 テーブル・エントリーの左端ワードは取り出されません。
4. ART テーブル・エントリーまたは DAT テーブル・エントリーは、命令の一部のオペランド参照が発生した後でも、取り出されることがあります。この取り出しは、該当の ART テーブル・エントリーまたは DAT テーブル・エントリーを必要とする実際のバイト・アクセスの直前に、初めて行われることもあります。
5. ART テーブル・エントリーまたは DAT テーブル・エントリーは、試行実行も含めてアドレス使用のたび、および各オペランドの各バイトに対する各参照のたびに、取り出されることがあります。
6. DAT ページ・テーブル・エントリーの取り出しは、そのページに対する参照の前に行われます。TLB 内に該当のページ・テーブル・エントリーのコピーがないときは、そのページ・テーブル・エントリーの取り出しの前に、関連のセグメント・テーブル・エントリーが取り出されます。TLB 内にそのセグメント・テーブル・エントリーのコピーがないときは、そのセグメント・テーブル・エントリーの取り出しの前に、領域第 3 テーブル・エントリーが取り出されます (そのエントリーが必要な場合)。同様に、領域第 2 テーブル・エントリーが必要な場合は、そのエントリーが領域第 3 テーブル・エントリーの取り出しの前に取り出され、さらに、領域第 1 テーブル・エントリーが必要な場合は、そのエン

トリーが領域第 2 テーブル・エントリーの取り出しの前に取り出されます。

7. ART により取得されたアドレス・スペース制御エレメントで指定されている領域テーブル・エントリーまたはセグメント・テーブル・エントリーのコピーが TLB 内に存在しないときは、DAT 領域テーブル・エントリーまたはセグメント・テーブル・エントリーの取り出しに先だって、ASN 第 2 テーブル・エントリーの ART 取り出しが行われます。必要な権限テーブル・エントリーのコピーが ALB 内に存在しない場合は、その権限テーブル・エントリーの取り出しに先立って、関連の ASN 第 2 テーブル・エントリーの ART 取り出しが行われます。必要な ASN 第 2 テーブル・エントリーのコピーが ALB 内に存在しない場合は、その ASN 第 2 テーブル・エントリーの取り出しに先立って、関連のアクセス・リスト・エントリーが取り出されます。必要なアクセス・リスト・エントリーのコピーが ALB 内に存在しないときは、そのアクセス・リスト・エントリーの取り出しに先立って、関連のアクセス・リスト指定が取り出されます。

記憶キーのアクセス

記憶キーの参照は以下のように扱われます。

1. 記憶域に対する参照が行われ、その参照にキー制御保護が適用される場合は、参照対象の記憶位置に関連した 4 個のアクセス制御ビットと取り出し保護ビットの検査が、その記憶位置に対する参照と並行して行われます。
2. 格納が行われるときは、その格納操作と並行して、関連の記憶キー内で変更ビットが設定されます。
3. **SET STORAGE KEY EXTENDED** 命令を実行すると、それと並行して、記憶キー内の 7 個のビットがすべて設定されます。**SET STORAGE KEY EXTENDED** では、記憶キーへのアクセスは記憶域オペランド格納参照の場合と同じ順序規則に従って行われます。また、これは単一アクセス参照です。
4. **INSERT STORAGE KEY EXTENDED** 命令は、記憶キーのビット 0-6 の一貫性のあるイメージを提供します。同様に、**INSERT VIRTUAL STORAGE KEY** および **TEST PROTECTION** 命令は、記憶キーのビット 0-4 の一貫性のあるイメージを提供します。これらの命令では、記憶キーへのアクセスは記憶域オペランド取り出し参照の場合と同じ順序規則に従って行われます。また、このアクセスは単一アクセス参照です。

5. RESET REFERENCE BIT EXTENDED 命令は、参照ビットのみを変更します。記憶キーのその他のビットはどれも変更されません。参照ビットと変更ビットが同時に調べられて、条件コードが設定されます。RESET REFERENCE BIT EXTENDED では、記憶キーへのアクセスは、記憶域オペランド更新参照の場合の順序規則に従って行われます。更新されるビットは参照ビットのみです。

参照ビットが示す参照記録は、必ずしも正確とは限りません。また、参照ビットの取り扱いは並行性の規則には従いません。ただし、ほとんどの場合は、参照記録は記憶域参照とほぼ一致しています。

変更ビットは、格納が行われていない場合でも設定されることがあります。5-22ページの『無効化と抑止に対する例外』を参照してください。

記憶域オペランドの参照

記憶域オプション参照とは、命令で指定されている記憶位置内にある 1 つまたは複数の明示オペランドを取り出すかまたは格納することです。

命令の実行過程では、その命令用の一部またはすべてのオペランドが取り出されたり、後で変更できるように中間結果が保管されたり、最終結果が記憶域に移される前に一時的に保持されたりします。チャンネル・プログラムまたは他の CPU による格納は、必ずしもこれらの中間結果に影響を与えるとは限りません。

記憶域オペランド参照には、取り出し、格納、更新の 3 種類があります。

記憶域オペランド取り出し参照

記憶域オペランドのバイトが、ソースとしてのみ命令の実行に関与する場合、そのオペランドを取り出しタイプ・オペランドと呼び、そのオペランドの記憶位置に対する参照を記憶域オペランド取り出し参照と呼びます。個々の命令のオペランド定義の中で、アクセス例外が取り出しに関するものであることが示されていれば、それは取り出しタイプ・オペランドです。

取り出しタイプ・オペランドでは、1 つのバイトのすべてのビットが同時にアクセスされます。1 つのオペランドが複数のバイトから成っているときは、それらのバイトは、記憶域から一度に 1 バイトずつ個別に取り出されることがあります。特に明記されていない限り、バイトは必ずしも一定の順序で取り出されるとは限りません。

チャンネル・プログラムおよび他の CPU から見れば、1 つの命令の記憶域オペランド取り出し参照は、先行するすべての命令の記憶域オペランド取り出し参照が終わった後で、後続の命令の記憶域オペランド取り出し参照が始まる前に、行われるように見えます。1 つの命令のオペランドは、その命令について指定されている順序で取り出されます。CPU は、命令が実行される前に命令のオペランドを取り出すことがあります。オペランドが取り出されてから、そのオペランドが使用されるまでの時間の長さについては、特定の制限はありません。しかし、CPU 自身から見れば、記憶域オプション参照は概念上の順序に従って行われます。

記憶域オペランド格納参照

記憶域オペランドのバイトが、宛先としてのみ命令の実行に関与し、結果により置き換えられるだけの場合は、そのオペランドを格納タイプ・オペランドと呼び、そのオペランドの記憶位置に対する参照を記憶域オペランド格納参照と呼びます。個々の命令のオペランド定義の中で、アクセス例外が格納に関するものであることが示されていれば、それは格納タイプ・オペランドです。

格納タイプ・オペランドでは、1 つのバイトのすべてのビットが同時にアクセスされます。1 つのオペランドが複数のバイトから成っているときは、それらのバイトは、一度に 1 バイトずつばらばらに記憶域に格納されることがあります。特に明記されていない限り、バイトは必ずしも一定の順序で格納されるとは限りません。

CPU は、記憶域への結果の格納を遅らせることがあります。結果が、格納されないまま保留状態になっている時間の長さについては、特に決まった制限はありません。この遅延によって、結果が記憶域に格納される順序が変わることはありません。

チャンネル・プログラムまたは他の CPU から見れば、1 つの命令の結果は、先行するすべての命令の結果が記憶域に格納された後で、かつ後続の命令の結果が格納される前に、格納されるように見えます。1 つの命令の結果は、その命令について指定されている順序で格納されます。

CPU は、ある記憶位置を宛先とするすべての情報を格納してしまうまでは、その記憶位置から、オペランド、ART テーブル・エントリー、または DAT テーブル・エントリーを取り出しません。事前取り出しされた命令は、情報が記憶域内に現れる前に更新されているように見ることがあります。

格納は、必ず逐次化操作の結果としてのみ実行され、CPU が停止状態になる前に完了します。

記憶域オペランド更新参照

命令によっては、記憶域オペランド位置が、ソースと宛先の両方の役割を兼ねることがあります。その場合は、その記憶位置への参照は、最初は取り出しであり、次は格納になります。このようなオペランドを更新タイプ・オペランドと呼び、2種類のアクセスの組み合わせを更新参照と呼びます。第1オペランド位置に対する更新を行う命令には、MOVE ZONES、TRANSLATE、OR (OC、OI)、および ADD DECIMAL などがあります。個々の命令のオペランド定義の中で、アクセス例外が取り出しと格納の両方に関するものであることが示されていれば、それは格納タイプ・オペランドです。

更新タイプ・オペランドを持つほとんどの命令では、1つの更新参照に関連した取り出しアクセスと格納アクセスは必ずしも連続して発生するとは限らず、その間に、チャンネル・プログラムまたは他のCPUが、同じ記憶位置に対する取り出しアクセスおよび格納アクセスを行うこともあります。このような更新参照は、非インターロック更新記憶域参照と呼ばれることがあります。

一部の特殊な命令では、他のCPUによる特定のアクセスに対して、更新参照がインターロックされます。このような更新参照を、インターロック更新参照と呼びます。1つのインターロック更新参照に関連した取り出しアクセスと格納アクセスは、必ずしも連続して発生するとは限りませんが、1つのインターロック更新参照について取り出しアクセスが行われてから格納アクセスが行われるまでの間は、他のCPUは、同じ記憶位置を対象として、そのインターロック更新参照に関連した格納アクセスおよび取り出し/格納アクセスを行うことはできません。チャンネル・プログラムは、インターロック期間中でも、該当の記憶位置にアクセスできます。

以下の命令に関する記憶域オペランド更新参照は、他のCPUからは、インターロック更新参照のように見えます。TEST AND SET、COMPARE AND SWAP、および COMPARE DOUBLE AND SWAP は、インターロック更新参照を実行します。0のマスクを持つ STORE CHARACTERS UNDER MASK 命令が、第2オペランド・アドレスが指すバイトの取り出しと格納を行うようなモデルでは、取り出しアクセスおよび格納アクセスはインターロック更新参照です。

更新参照に関連した取り出しアクセスと格納アクセスは、上記の制約の範囲内で、前のセクションで述べた取り出しおよび格納の場合と同じ規則に従って行われます。

プログラミング上の注意:

- 2つのCPUが、非インターロック更新参照を使用して同じ主記憶位置を更新しようとした場合は、両方のCPUが、情報を取り出した後で格納アクセスを行う可能性があります。その場合は、結果を格納するために最初のCPUが行った変更は失われます。同様に、一方のCPUが、非インターロック更新参照を使用してあるフィールドの内容を更新しているときに、取り出しから格納までの間に、もう一方のCPUがそのフィールドに対する格納アクセスを行った場合は、最初の格納の結果は失われます。また、あるCPUにより非インターロック更新参照の取り出し部分と格納部分の間で、別のCPUが同じ記憶域フィールドに対して、格納アクセスでなくインターロック更新参照を行った場合は、そのインターロック更新参照により得られた内容は失われます。
- TEST AND SET、COMPARE AND SWAP、および COMPARE DOUBLE AND SWAP 命令は、複数のCPUによる共通記憶域フィールドの更新を容易にします。どのCPUによる変更も失われないようにするには、すべてのCPUが、インターロック更新参照を行う命令を使用する必要があります。さらに、チャンネル・プログラムは、インターロック更新参照の取り出し部分と格納部分の間で格納を行うことがあるので、同じ記憶位置に対してこのような格納を行うことがないようにプログラミングする必要があります。
- 両方の操作の結果フィールドに含まれているバイトのみが、共通の主記憶位置の一部と見なされます。ただし、共通バイトの中のすべてのビットは、2つの操作により変更されるビットが同じでなくても、共通であると見なされます。例えば、(1) 一方のCPUが、第2オペランド位置の長さを1、値を16進数80としてOR(OC)命令を実行し、(2) もう一方のCPUが、第2オペランド位置の長さを1、値を16進数FEとしてAND(NC)を実行し、(3) 両方の命令の第1オペランドのタイプが同じである場合は、いずれかの更新の結果は失われます。
- 格納アクセスがCPUによる更新参照の一部であるときには、必ずしも、格納される情報が該当記憶位置の元の内容と違っているかどうかによって、格納の実行が左右されるわけではありません。具体的には、指定されたすべてのバイト位置が置き換えられ、フィールド内の各バイトについて、バイトの内容全体が置き換えられます。

モデルによって異なりますが、情報を格納するためのアクセスは、例えば以下のような場合に行われます。

- a. 第 2 オペランドがすべて 0 の OR 命令 (OI または OC) の実行。
- b. 第 1 オペランド・フィールドと第 2 オペランド・フィールドが一致している OR (OC) の実行。
- c. TRANSLATE の第 1 オペランド位置で、引き数値と関数値が同じである場合。

- RESUME PROGRAM
- STORE REVERSED
- STORE SYSTEM INFORMATION
- TEST BLOCK
- TRANSLATE
- TRANSLATE EXTENDED
- TRANSLATE ONE TO ONE
- TRANSLATE ONE TO TWO
- TRANSLATE TWO TO ONE
- TRANSLATE TWO TO TWO
- UNPACK
- UNPACK ASCII
- UNPACK UNICODE
- UPDATE TREE

記憶域オペランドの一貫性

単一アクセス参照

取り出し参照は、データ・フィールドの各バイトへの 1 回のアクセスにより値が取り出される場合に、単一アクセス参照と呼ばれます。オーバーラップ・オペランドの場合、記憶位置は各オペランドごとに 1 回ずつアクセスされます。格納タイプ参照は、データ・フィールド内の各バイト位置に対して 1 つの格納アクセスがある場合に、単一アクセス参照と呼ばれます。更新参照は、取り出しアクセスと格納アクセスが両方とも単一アクセスである場合に、単一アクセスと呼ばれます。

複数アクセス参照に関連したアクセス、および DAT 関連のアクセス例外の場合の記憶域の変更と復元に関連した格納を除き、記憶域オペランド参照はすべて単一アクセス参照です。

複数アクセス参照

場合によっては、記憶域オペランドのすべてまたは一部のバイトに対して複数のアクセスが行われることがあります。複数アクセス参照が行われるのは、以下のような場合です。

1. 以下の命令の記憶域オペランド。

- CHECKSUM
- COMPARE AND FORM CODEWORD
- COMPARE UNTIL SUBSTRING EQUAL
- CONVERT TO BINARY
- CONVERT TO DECIMAL
- CONVERT UNICODE TO UTF-8
- CONVERT UTF-8 TO UNICODE
- LOAD ADDRESS SPACE PARAMETERS
- LOAD REVERSED
- MOVE INVERSE
- MOVE PAGE
- MOVE WITH OFFSET
- PACK
- PACK ASCII
- PACK UNICODE

2. MOVE LONG、MOVE LONG EXTENDED、または MOVE LONG UNICODE の第 1 オペランドの埋め込みバイトで満たされる部分への格納。
3. 10 進数命令の記憶域オペランド。
4. PAGE IN および PAGE OUT の主記憶域オペランド。
5. 入出力命令の記憶域オペランド。
6. トレース・エントリへの格納。
7. 停止/状況記憶およびアドレス指定状況記憶 SIGNAL PROCESSOR 指令に関連した格納。
8. TRAP により使用されるトラップ制御ブロックおよびトラップ保管域。
9. COMPRESSION CALL のオペランド、辞書、およびシンボル変換テーブル。

ある記憶位置への記憶域オペランド格納参照が単一アクセス参照でないときは、同じバイト位置に格納される値は、それぞれの格納アクセスのたびに必ずしも同じになるとは限りません。したがって、単一バイト位置に中間結果が入り、チャンネル・プログラムおよび他の CPU にはその中間結果が提示されてしまうことがあります。

プログラミング上の注意:

1. チャンネル・プログラムまたは他の CPU が変更している単一バイトに対して複数の取り出しアクセスまたは格納アクセスが行われた場合の結果は、必ずしも、各ビットを個別に取り出したまたは格納した場合に得られる結果と同じになるとは限りません。例えば、MULTIPLY DECIMAL の実行では、加算と減算が反復的に行われ、そのたびに記憶域から第 2 オペランドが取り出され、第 1 オペランドが記憶域内で更新されることがあります。

2. ある記憶位置をチャンネル・プログラムまたは他の CPU がアクセスしているときに、同時に、複数アクセス参照を行う CPU 命令を使用してその記憶位置を変更した場合、CPU による単一バイトへの複数の格納アクセスの結果、チャンネル・プログラムまたは CPU からは中間結果が観察されることがあります。例えば CCW チェーンを変更するときなどにこのような中間値が生じるのを避けるためには、単一アクセス参照を行う命令のみを使用するようにしてください。
3. 命令取り出し (EXECUTE のターゲットの取り出しも含む) は、複数アクセス参照です。

ブロック単位の参照

一部の参照では、1 つのハーフワード、ワード、ダブルワード、またはクワッドワード内のすべてのバイトに対するアクセスが、他の CPU からはブロック単位に見えるように指定されます。チャンネル・プログラムからは、これらのアクセスは一度に複数のバイトを含むようには見えないことがあります。このセクションでは、ハーフワード、ワード、ダブルワード、またはクワッドワードをブロックと呼びます。取り出しタイプ参照がブロック単位に見えるように指定されている場合、そのブロックに含まれるバイトが取り出されている間は、他の CPU によるそのブロックへの格納アクセスは許可されません。取り出しと取り出しの間に、チャンネル・プログラムがブロック内のバイトにアクセスすることはあります。格納タイプ参照がブロック単位に見えるように指定されている場合、そのブロック内のバイトが格納されている間は、他の CPU によるそのブロックへのアクセス (取り出しまたは格納) は許可されません。格納と格納の間に、チャンネル・プログラムがブロック内のバイトにアクセスすることはあります。

一貫性の指定

CONVERT TO DECIMAL、CONVERT TO BINARY、LOAD PSW EXTENDED、LOAD REVERSED、RESUME PROGRAM、STORE CLOCK EXTENDED、STORE REVERSED、STORE SYSTEM INFORMATION、TRAP、および入出力命令を除き、S、RX、または RXE フォーマットのすべての命令では、オペランドのサイズに不可欠な境界上でオペランドがアドレッシングされているときは、記憶域オペランド参照は、他の CPU からはブロック単位のように見えます。LOAD PSW EXTENDED の場合は、記憶域オペランドの 2 つのダブルワードのそれぞれに対するアクセスは、他の CPU からはダブルワード単位のように見えます。

COMPARE AND SWAP、COMPARE AND SWAP AND PURGE、および COMPARE DOUBLE AND SWAP 命令の場合は、他の CPU からは、記憶域オペランドに対するすべての参照がブロック単位のように見えます。

PERFORM LOCKED OPERATION 命令では、他の CPU から見た場合、偶数番号の記憶域オペランドへのアクセスは、機能コードが 4 の倍数であればワード単位のように見え、機能コードが 4 の倍数に 1、2、または 3 を加えた値であればダブルワード単位のように見えます。パラメーター・リスト内のダブルワードへのアクセスは、他の CPU から見れば、機能コードに関係なくダブルワード単位のように見えます。

オペランド (1 つまたは複数) がワード境界から始まっている LOAD MULTIPLE、LOAD MULTIPLE DISJOINT、LOAD MULTIPLE HIGH、STORE MULTIPLE、および STORE MULTIPLE HIGH 命令、および、COMPARE LOGICAL (CLC)、COMPARE LOGICAL CHARACTERS UNDER MASK、INSERT CHARACTERS UNDER MASK、LOAD CONTROL (LCTLG)、STORE CHARACTERS UNDER MASK、および STORE CONTROL (STCTG) 命令は、左から右への方向で記憶域オペランドにアクセスし、他の CPU からは、1 つのダブルワード内のバイトはすべて同時にアクセスされているように見えます。

LOAD ACCESS MULTIPLE、LOAD CONTROL (LCTL)、STORE ACCESS MULTIPLE、および STORE CONTROL (STCTL) 命令は、左から右への方向で記憶域オペランドにアクセスし、他の CPU からは、1 つのワード内のすべてのバイトが同時にアクセスされているように見えます。

破壊的オーバーラップが存在しない場合は、MOVE (MVC)、MOVE WITH KEY、MOVE TO PRIMARY、および MOVE TO SECONDARY のオペランドは以下のようにアクセスされます。

1. 第 1 オペランドは左から右への方向にアクセスされ、他の CPU からは、1 つのダブルワード内でアクセスされるすべてのバイトが同時にアクセスされているように見えます。
2. 第 2 オペランドは左から右への方向にアクセスされ、他の CPU からは、第 2 オペランドのダブルワード内のバイトのうち、第 1 オペランド内の 1 つのダブルワードに移されるすべてのバイトが、同時に取り出されているように見えます。したがって、第 1 オペランドと第 2 オペランドがダブルワード内の同じバイト・オフセットから始まっている場合

は、他の CPU からは、第 2 オペランドの取り出しはダブルワード単位のように見えます。ダブルワード内での両者のオフセットが 4 バイト違っている場合は、他の CPU からは、第 2 オペランドの取り出しはワード単位のように見えます。

破壊的オーバーラップが存在すると見なされるのは、処理が一度に 1 バイトずつ行われるときに、結果の格納後に、結果の記憶位置がソースとして使用される場合です。

MOVE WITH SOURCE KEY、MOVE WITH DESTINATION KEY、および MOVE STRING のオペランドへのアクセスは、破壊的オーバーラップが存在しないものと見なされる点以外は、MOVE (MVC) の場合と同じです。

チャンネル・プログラムまたは他の CPU から見た場合、MOVE LONG、MOVE LONG EXTENDED、および MOVE LONG UNICODE のオペランドへのアクセスは、各命令についての説明の中で特に明記されていない限り、必ずしも左から右への方向で行われるとは限りません。他の CPU から見たときに、これらの命令のオペランドがダブルワード単位でアクセスされるように見えるのは、以下のすべての条件が満たされている場合です。

- 両方のオペランドがダブルワード境界から始まり、長さがダブルワードの整数倍である。
- オペランドがオーバーラップしていない。
- 操作の非理め込み部分が実行されている。

COMPARE LOGICAL LONG、COMPARE LOGICAL LONG EXTENDED、および COMPARE LOGICAL LONG UNICODE のオペランドが両方ともダブルワード境界から始まっていて、長さがダブルワードの整数倍である場合は、他の CPU からは、これらのオペランドがダブルワード単位でアクセスされているように見えます。

COMPARE LOGICAL STRING のオペランドが両方ともダブルワード境界から始まっている場合は、他の CPU からは、オペランドがダブルワード単位でアクセスされているように見えます。SEARCH STRING のオペランドがダブルワード境界から始まっている場合は、他の CPU からは、そのオペランドはダブルワード単位でアクセスされているように見えます。

EXCLUSIVE OR (XC) では、オペランドは左から右への方向に処理され、第 1 オペランドと第 2 オペランドが一致しているときは、他の CPU からは、1 つのダブ

ルワード内のすべてのバイトが同時にアクセスされているように見えます。

プログラミング上の注意: 2 つのオペランドが正確に一致している EXCLUSIVE OR (XC) の場合は、オペランド・フィールド内のバイトは、3 回 (取り出しが 2 回と格納が 1 回) アクセスされるように見えることがあります。1 回目は第 1 オペランド更新の取り出し部分、2 回目は第 2 オペランドの取り出し、そして 3 回目は第 1 オペランド更新の格納部分です。他の CPU からは、この 3 回のアクセスはどれもダブルワード単位で行われているように見えますが、必ずしも 3 回のアクセスが連続して発生しているように見えるとは限りません。オペランドを取り出さずに命令を完了できる場合は、一方または両方の取り出しアクセスが省略されることがあります。

オペランド・アクセス間の関係

チャンネル・プログラムおよび他の CPU から見た場合、1 つの命令の実行に関連した記憶域オペランド取り出しは、概念上それより後にある命令のすべての記憶域オペランド参照参照より前に行われるように見えます。1 つの命令で指定されている記憶域オペランド格納は、概念上それより後にある命令で指定されているすべての記憶域オペランド格納より前に行われるように見えますが、必ずしも、概念上後にある命令で指定されている記憶域オペランド取り出しより前になるとは限りません。ただし、格納と取り出しが同じ主記憶位置を対象としている場合は、記憶域オペランド格納は、概念上それより後にある記憶域オペランド取り出しより前に、行われるように見えます。

1 つの命令に 2 つの記憶域オペランドがあり、両方とも取り出し参照を起こさせるものである場合は、どちらのオペランドが最初に取り出されるか、または、一方のオペランドが取り出される前にもう一方のオペランドがどれだけ取り出されるのかは、予測不能です。2 つのオペランドがオーバーラップしている場合は、両者に共通する記憶位置が各オペランドごとに独立して取り出されることがあります。

1 つの命令に 2 つのオペランドがあり、第 1 オペランドが格納を起こさせ、第 2 オペランドが取り出し参照を起こさせる場合は、結果が格納される前に第 2 オペランドがどれだけ取り出されるかは、予測不能です。破壊的オーバーラップ・オペランドの場合は、第 2 オペランドの第 1 オペランドと共通する部分は、記憶域から必ず取り出されるとは限りません。

1 つの命令に 2 つの記憶域オペランドがあり、第 1 オペランドが更新参照を起こさせ、第 2 オペランドが取り

出し参照を起こさせる場合は、どちらのオペランドが最初に取り出されるか、または、一方のオペランドが取り出される前にもう一方のオペランドがどれだけ取り出されるかは、予測不能です。同様に、結果が記憶域に戻される前にどれだけ処理されるかも、予測不能です。破壊的オーバーラップ・オペランドの場合は、第 2 オペランドの第 1 オペランドと共通する部分は、記憶域から必ず取り出されるとは限りません。

次に示す状況下では、2 つのオペランドのそれぞれについて、同じ記憶位置を別々に取り出すことにより、プログラムの実行が影響を受けることがあります。1 つの命令の 2 つのオペランド・アドレスが同じ記憶位置を指しているときに、その命令の実行中に、チャンネル・プログラムまたは他の CPU によりその記憶位置の内容が変更された場合は、その記憶位置の新旧両方の値が同時に使用されることがあります。例えば、あるフィールドをそれ自体と比較した結果が等値にならなかったり、あるフィールドとそれ自体との排他論理和演算の結果が 0 にならなかったりすることがあります。

その他の記憶域参照

再始動、プログラム、監視プログラム呼び出し、外部、入出力、およびマシン・チェックの各 PSW は、他の CPU からは、ダブルワード単位でアクセスされるように見えます。これらの参照は、概念上前の操作単位の後で、かつ概念上後の操作単位の前に発生するように見えます。新 PSW の取り出し、旧 PSW の格納、および割り込みコードの格納の間の関係は、予測不能です。

割り込みコードの格納アクセスは、必ずしも単一アクセス格納とは限りません。外部割り込みおよび監視プログラム割り込みのコードの格納アクセスは、概念上前の操作と概念上後の操作の間に起きるように見えます。プログラム割り込みコードの格納アクセスは、プログラム割り込みの原因となる命令に関連した記憶域オペランド参照より前に行われることがあります。

逐次化

通常は、1 つの CPU が機能を実行する順序は、チャンネル・プログラムおよび他の CPU が実行する機能とは無関係です。同様に、1 つのチャンネル・プログラムが機能を実行する順序は、通常、CPU および他のチャンネル・プログラムが実行する機能とは無関係です。しかし、実行中の一定の時点で、CPU の逐次化が起こります。また、チャンネル・プログラムも、一定の時点で逐次化されます。

CPU の逐次化

すべての割り込み、および一部の命令の実行は、CPU 操作の逐次化を引き起こします。チャンネル・プログラムおよび他の CPU から見た場合、逐次化操作では、この CPU において概念上前にあるすべての記憶域アクセスが、概念上後にある記憶域アクセスが起きる前に完了されます。逐次化は、ART テーブル・エントリーおよび DAT テーブル・エントリーの取り出しに関連したアクセスを除き、記憶域および記憶キーに対するすべての CPU アクセスの順序に影響を与えます。

逐次化を起こさせる要因は、CPU リセット、すべての割り込み、および以下の命令の実行です。

- 一般命令 BRANCH ON CONDITION (BCR) (M₁ フィールドがすべて 1 で、R₂ フィールドがすべて 0 の場合)、および、COMPARE AND SWAP、COMPARE DOUBLE AND SWAP、STORE CLOCK、SUPERVISOR CALL、および TEST AND SET。
- LOAD PSW、LOAD PSW EXTENDED、および SET STORAGE KEY EXTENDED。
- すべての入出力命令。
- PURGE ALB、PURGE TLB、および SET PREFIX。PURGE ALB および SET PREFIX では、ALB 内のすべてのエントリーの消去も行われます。また、PURGE TLB および SET PREFIX では、TLB 内のすべてのエントリーの消去も行われます。
- SIGNAL PROCESSOR。アーキテクチャー設定 SIGNAL PROCESSOR 指令は、構成内のすべての CPU の逐次化を引き起こします。
- INVALIDATE PAGE TABLE ENTRY。
- TEST BLOCK。
- MOVE TO PRIMARY、MOVE TO SECONDARY、PROGRAM CALL、PROGRAM CALL FAST、PROGRAM TRANSFER、SET ADDRESS SPACE CONTROL、および SET SECONDARY ASN。
- アンスタックすべき状態エントリーがプログラム呼び出し状態エントリーである場合の PROGRAM RETURN。
- PERFORM LOCKED OPERATION。逐次化は、ロックが獲得された直後に行われるほか、ロックが解除される直前に再度行われます。ただし、ロック

が獲得される前にパラメーター・リストから取り出された値が、必ず再取り出しされるとは限りません。

- 4つのトレース機能 (ブランチ・トレース、ASN トレース、モード・トレース、および明示トレース) では、トレース・アクションの開始前と完了後に、逐次化が行われます。
- PAGE IN および PAGE OUT。
- COMPARE AND SWAP AND PURGE。これは、すべての CPU において、ALB および TLB 内のすべてのエントリーの消去も行います。

逐次化操作に伴うイベントの順序は、以下のとおりです。

1. チャンネル・プログラムおよび他の CPU から見た場合、この CPU が行う記憶域アクセスのうち、概念上前にあるすべてのアクセスが完了します。これには、概念上前にある、記憶キーのすべての格納と変更が含まれます。
2. 逐次化操作に関連した通常の機能が実行されます。命令実行の場合は、オペランドが取り出され、結果の格納が完了します。例外として、LOAD PSW、LOAD PSW EXTENDED、および SET PREFIX では、オペランドは前の格納が完了する前に取り出されることがあり、また、割り込みでは、割り込みコードおよび関連のフィールドが逐次化の前に格納されることがあります。逐次化を行う命令の取り出しは、その命令の実行の前に行われ、また、場合によってはそれより前の命令の実行に先だって行われることもあります。割り込みの場合は、旧 PSW、割り込みコード、およびその他の情報 (存在する場合) が格納され、新 PSW が取り出されますが、必ずしもこの順序で行われるとは限りません。
3. 最後に、概念上後に続く操作のための命令取り出しおよびオペランド・アクセスが開始されます。

逐次化機能は、逐次化が行われている CPU の制御下にある記憶域アクセスの順序に影響を与えます。チャンネル・プログラムおよび他の CPU の制御下にある記憶域アクセスの順序には、影響はありません。

プログラミング上の注意:

1. 逐次化操作がもたらす影響には、以下のようなものがあります。
 - a. ある命令の実行により、後続の命令のソースとして使用される記憶位置の内容が変更され、かつ、結果を格納するときと命令を取り出すときに、同じ絶対位置を指定するために異なるアド

レスが使用されている場合に、その変更の後の逐次化により、確実に変更後の命令が実行されるようになります。

- b. 逐次化が行われると、チャンネル・プログラムおよび他の CPU からは、命令とオペランドの取り出しおよび結果の格納が、逐次化操作により確立された順序で行われているように見えます。
2. 逐次化命令が取り出される記憶位置への格納が行われた場合、その格納の後で、かつ逐次化命令の実行の前に逐次化機能が実行される場合を除けば、その格納が必ず逐次化の実行に影響を与えないとは限りません。
 3. 以下の例は逐次化の効果を示しています。記憶位置 A には、最初は X'FF' が含まれています。

CPU 1		CPU 2	
MVI	A,X'00'	G CLI	A,X'00'
BCR	15,0	BNE	G

CPU 1 が実行する BCR 15,0 命令は逐次化命令で、これにより、記憶位置 A での CPU 1 による格納が確実に完了します。しかし、CPU 2 は、無限にループするか、または CPU で次に入出力割り込みまたは外部割り込みが発生するまでループする可能性があります。なぜなら、CPU 2 は、CLI 命令の実行のために、すでに記憶位置 A からの取り出しを行っていることがあるからです。CPU 2 が必ず再度記憶位置 A から取り出しを行うようにするためには、CPU-2 のループの中で逐次化命令が実行されることが必要です。

チャンネル・プログラムの逐次化

チャンネル・プログラムの逐次化は、以下のように行われます。

1. チャンネル・プログラムによるすべての記憶域アクセスおよび記憶キー・アクセスは、CPU および他のチャンネル・プログラムから見た場合、START SUBCHANNEL の実行開始の後、あるいは中断状態のときは RESUME SUBCHANNEL の実行開始の後で行われます。これには、CCW、IDAW、およびデータのすべてのアクセスが含まれます。
2. チャンネル・プログラムによるすべての記憶域アクセスおよび記憶キー・アクセスは、CPU および他のチャンネル・プログラムから見た場合、1 次状況での状況保留を示すサブチャンネル状況がいずれかの CPU にとって使用可能になる前に、完了します。

3. CCW に、PCI フラグまたは値が 1 の中断フラグが含まれている場合は、CCW チェーン内の先行する CCW が原因で生じるすべての記憶域アクセスおよび記憶キー・アクセスは、CPU および他のチャンネル・プログラムから見た場合、中間状況 (PCI または中断) での状況保留を示しているサブチャンネル状

況がいずれかの CPU にとって使用可能になる前に、完了します。

チャンネル・プログラムの逐次化は、他のチャンネル・プログラムまたは他の CPU プログラムによる記憶域アクセスまたは記憶キー・アクセスの順序には影響を与えません。

第6章 割り込み

割り込みアクション	6-2	固定小数点除算例外	6-22
割り込みコード	6-5	固定小数点オーバーフロー例外	6-22
使用可能と使用不可	6-6	HFP 除算例外	6-22
浮動割り込み条件の取り扱い	6-6	HFP 指数オーバーフロー例外	6-22
命令長コード	6-7	HFP 指数アンダーフロー例外	6-22
ILC が 0 の場合	6-7	HFP 有効数字例外	6-23
命令取り出し時の例外の際の ILC	6-7	HFP 平方根例外	6-23
PSW に関連した例外	6-9	LX 変換例外	6-23
早期例外認識	6-9	モニター・イベント	6-23
遅延例外認識	6-9	オペランド例外	6-24
外部割り込み	6-10	操作例外	6-24
クロック・コンパレーター	6-10	ページ変換例外	6-25
CPU タイマー	6-11	PC 変換指定例外	6-25
緊急信号	6-11	PER イベント	6-25
ETR	6-11	1 次権限例外	6-26
外部呼び出し	6-11	特権命令例外	6-26
割り込みキー	6-12	記憶保護例外	6-27
誤動作警報	6-12	領域第 1 変換例外	6-27
サービス信号	6-12	領域第 2 変換例外	6-28
入出力割り込み	6-12	領域第 3 変換例外	6-28
マシン・チェック割り込み	6-13	2 次権限例外	6-29
プログラム割り込み	6-13	セグメント変換例外	6-29
データ例外コード (DXC)	6-14	スペース切り替えイベント	6-29
データ例外の場合のプログラム割り込みの		特殊操作例外	6-30
優先順位	6-14	指定例外	6-31
プログラム割り込みの条件	6-15	スタック空例外	6-33
アドレッシング例外	6-15	スタック一杯例外	6-33
AFX 変換例外	6-18	スタック操作例外	6-33
ALEN 変換例外	6-18	スタック指定例外	6-33
ALE シーケンス例外	6-18	スタック・タイプ例外	6-33
ALET 指定例外	6-18	トレース・テーブル例外	6-34
ASCE タイプ例外	6-18	変換指定例外	6-34
ASTE シーケンス例外	6-19	プログラム割り込みの集合名	6-35
ASTE 妥当性例外	6-19	アクセス例外の認識	6-35
ASX 変換例外	6-20	複数のプログラム割り込み条件	6-37
暗号操作例外	6-20	アクセス例外	6-40
データ例外	6-20	ASN 変換例外	6-43
10 進除算例外	6-21	サブスペース置き換え例外	6-43
10 進オーバーフロー例外	6-21	トレース例外	6-44
実行例外	6-21	再始動割り込み	6-44
EX 変換例外	6-21	監視プログラム呼び出し割り込み	6-44
拡張権限例外	6-21	割り込みの優先順位	6-44

割り込みメカニズムを利用することにより、CPU は、構成の外部、構成の内部、または CPU 自体の中の条件に基づいて、自身の状況を変化させることができます。優先順位の高い条件にすばやく対応し、条件のタイプを即時に認識できるようにするために、割り込み条件は 6 つのクラスに分類されています。それは、外部、入出力、マシン・チェック、プログラム、再始動、および監視プログラム呼び出しです。

割り込みアクション

割り込みでは、現 PSW が旧 PSW として格納され、割り込みの原因を示す情報が格納され、そして新 PSW が取り出されます。処理は、新 PSW の指定内容に従って再開されます。

通常、割り込みが生じたときに格納される旧 PSW には、割り込まれたプログラムを再開できるようにするた

めに、割り込みが起きなかったとすれば次に実行されるはずだった命令のアドレスが含まれています。プログラム割り込みおよび監視プログラム呼び出し割り込みの場合は、プログラムが割り込みの原因に対処できるように、最後に実行された命令の長さを示すコードも格納されます。あるプログラム条件に対する通常の応答が、割り込みの原因となった命令の再実行である場合は、命令アドレスは、最後に実行された命令を直接識別していません。

再始動割り込みを除き、割り込みが起きるのは CPU が作動状態にあるときだけです。再始動割り込みが起きることがあるのは、CPU が停止状態または作動状態にあるときです。

原因の判別、位置の判別、および命令の実行については、6-3ページの図6-1に要約を示し、その後の各セクションで詳しく説明します。

原因の識別	割り込みコード	PSW-マスク・ビット	制御レジスタ内のマスク・ビット	ILC設定	旧 PSW で指定されている命令の実行
MACHINE CHECK (旧 PSW 352、 新 PSW 480)	位置 232-239 ¹				
緊急条件		13		u	中止または無効化 ²
抑制可能条件		13	14, 35-39	u	影響なし ²
SUPERVISOR CALL (旧 PSW 320、 新 PSW 448)	位置 138-139				
命令ビット	00000000 ssssssss			1,2	完了
PROGRAM (旧 PSW 336、 新 PSW 464)	位置 142-143				
	2 進数	16 進数 ³			
操作	00000000 p0000001	0001		1,2,3	抑止
特権操作	00000000 p0000010	0002		2,3	抑止
実行	00000000 p0000011	0003		2	抑止
保護	00000000 p0000100	0004		1,2,3	抑止または中止
アドレッシング	00000000 p0000101	0005		1,2,3	抑止または中止
指定	00000000 p0000110	0006		0,1,2,3	抑止または完了
データ	00000000 p0000111	0007		1,2,3	抑止、中止、または完了
固定小数点オーバーフロー	00000000 p0001000	0008	20	1,2,3	完了
固定小数点除算	00000000 p0001001	0009		1,2,3	抑止または完了
10 進オーバーフロー	00000000 p0001010	000A	21	2,3	完了
10 進除算	00000000 p0001011	000B		2,3	抑止
HFP 指数オーバーフロー	00000000 p0001100	000C		1,2,3	完了
HFP 指数アンダーフロー	00000000 p0001101	000D	22	1,2,3	完了
HFP 有効数字	00000000 p0001110	000E	23	1,2	完了
HFP 除算	00000000 p0001111	000F		1,2	抑止
セグメント変換	00000000 p0010000	0010		1,2,3	無効化
ページ変換	00000000 p0010001	0011		1,2,3	無効化
変換指定	00000000 p0010010	0012		1,2,3	抑止
特殊操作	00000000 p0010011	0013	0, 33	1,2,3	抑止
オペランド	00000000 p0010101	0015		2	抑止
トレース・テーブル	00000000 p0010110	0016		1,2	無効化
スペース切り替え	00000000 p0011100	001C	1, 57	0,1,2	完了
HFP 平方根	00000000 p0011101	001D		2	抑止
PC 変換指定	00000000 p0011111	001F		2	抑止

図 6-1 (1/3). 割り込みアクション

原因の識別	割り込みコード		PSW-マスク・ビット	制御レジスター内のマスク・ビット	レスジスター・ビット	ILC設定	旧 PSW で指定されている命令の実行
AFX 変換	00000000	p0100000	0020			1,2	無効化
ASX 変換	00000000	p0100001	0021			1,2	無効化
LX 変換	00000000	p0100010	0022			2	無効化
EX 変換	00000000	p0100011	0023			2	無効化
1 次権限	00000000	p0100100	0024			2	無効化
2 次権限	00000000	p0100101	0025			1,2	無効化
ALET 指定	00000000	p0101000	0028			1,2,3	抑止
ALEN 変換	00000000	p0101001	0029			1,2,3	無効化
ALE シーケンス	00000000	p0101010	002A			1,2,3	無効化
ASTE 妥当性	00000000	p0101011	002B			1,2,3	無効化
ASTE シーケンス	00000000	p0101100	002C			1,2,3	無効化
拡張権限	00000000	p0101101	002D			1,2,3	無効化
スタック一杯	00000000	p0110000	0030			2	無効化
スタック空	00000000	p0110001	0031			1,2	無効化
スタック指定	00000000	p0110010	0032			1,2	無効化
スタック・タイプ	00000000	p0110011	0033			1,2	無効化
スタック操作	00000000	p0110100	0034			1,2	無効化
ASCE タイプ	00000000	p0111000	0038			1,2,3	無効化
領域第 1 変換	00000000	p0111001	0039			1,2,3	無効化
領域第 2 変換	00000000	p0111010	003A			1,2,3	無効化
領域第 3 変換	00000000	p0111011	003B			1,2,3	無効化
モニター・イベント	00000000	p1000000	0040	8, 32-47		2	完了 ⁵
PER イベント	00000000	lnnnnnn ⁵	0080	9, 32-36		0,1,2,3	完了 ⁶
暗号操作	00000001	p0011001	0119			2	無効化
EXTERNAL (旧 PSW 304, 新 PSW 432)	位置 134-135						
	2 進数	16 進数 ³					
割り込みキー	00000000	01000000	0040	7	0, 57	u	影響なし
クロック・コンパレーター	00010000	00000100	1004	7	0, 52	u	影響なし
CPU タイマー	00010000	00000101	1005	7	0, 53	u	影響なし
誤動作警報	00010010	00000000	1200	7	0, 48	u	影響なし
緊急信号	00010010	00000001	1201	7	0, 49	u	影響なし
外部呼び出し	00010010	00000010	1202	7	0, 50	u	影響なし
ETR	00010100	00000110	1406	7	0, 59	u	影響なし
サービス信号	00100100	00000001	2401	7	0, 54	u	影響なし

図 6-1 (2/3). 割り込みアクション

原因の識別	割り込みコード	PSW-マスク・ビット	制御レジスタ内のマスク・ビット レジスタ・ビット	ILC 設定	旧 PSW で指定されている命令の実行
INPUT/OUTPUT (旧 PSW 368、 新 PSW 496) 入出力割り込みサブクラス	位置 184-191	6	6, 32-39 ₄	u	影響なし
RESTART (旧 PSW 288、 新 PSW 416) 再始動キー	なし			u	影響なし
説明: 旧 PSW、新 PSW、および割り込みコードの位置は、実記憶位置です。 1 モデルによって異なる 64 ビットのマシン・チェック割り込みコードは、実記憶位置 232-239 に格納されます。 2 マシン・チェック条件の影響は、マシン・チェック割り込みコード内のビットにより示されます。これらのビットの設定は、損傷の程度と、操作単位が無効化されたか、中止されたか、影響なしを示します。 3 「16 進数」欄に示されている割り込みコードは、基本割り込みの 16 進数コードです。このコードは、「2 進数」欄の n または p で表されている並行割り込み条件の影響を示すものではありません。 4 制御レジスタ 6 のビット 32-39 は、それぞれ、入出力割り込みサブクラス 0-7 の詳細なマスクングを提供します。 5 割り込みコードが PER イベントを示しているときに、0 の ILC が格納されることがあるのは、割り込みのビット 8-15 が 1000110 (PER、指定) である場合のみです。 6 並行して示されているプログラム例外が原因で操作単位が無効化、抑止、または中止された場合を除き、操作単位は完了します。 n 別のプログラム割り込み条件を示す可能な 0 以外のコード。 p このビットが 1 のときは、並行 PER イベント割り込み条件を示します。 s SUPERVISOR CALL の I フィールドのビット。 u 格納されません。					

図 6-1 (3/3). 割り込みアクション

割り込みコード

割り込みの 6 つのクラス (外部、入出力、マシン・チェック、プログラム、再始動、監視プログラム呼び出し) は、旧 PSW がどの記憶位置に格納されているか、および新 PSW がどの記憶位置から取り出されるかによって区別されます。ほとんどのクラスの割り込みについては、割り込みコードにより、そして一部のクラスについては、割り込み時に永続割り当て済みの実記憶位置に格納された追加情報により、さらに詳しく原因を究明することができます。(3-49ページの『割り当て済みの記憶位置』も参照してください。) 外部割り込み、プログラム割り込み、および監視プログラム割り込みの場合は、割り込みコードは 16 ビットから成っています。

外部割り込みの場合は、割り込みコードは実記憶位置 134-135 に格納されます。実記憶位置 128-131 にはパ

ラメーター、実記憶位置 132-133 には CPU アドレスが格納されることがあります。

入出力割り込みの場合は、割り込みコードは実記憶位置 184-191 に格納されます。入出力割り込みコードは、32 ビットのサブシステム ID ワードと 32 ビットの割り込みパラメーターから成っています。

マシン・チェック割り込みの場合は、割り込みコードは 64 ビットから成り、実記憶位置 232-239 に格納されます。マシン・チェック故障記憶機構アドレスの内容および固定ログアウト域とマシン・チェック保管域の内容から、割り込みの原因を判別するため、およびマシンの状態をリカバリーするための追加情報が得られることがあります。(第11章、『マシン・チェックの取り扱い』を参照してください。)

プログラム割り込みの場合は、割り込みコードは実記憶位置 142-143 に格納され、実記憶位置 141 のビット位置 5 および 6 には命令長コードが格納されます。さらに、実記憶位置 144-162 および 168-183 に格納される、データ例外コード (DXC)、モニター・クラス番号、PER コード、アドレッシングおよび変換モード ID、PER アドレス、例外アクセス ID、PER アクセス ID、オペランド・アクセス ID、変換例外 ID、およびモニター・コードからも、情報が得られることがあります。

使用可能と使用不可

現 PSW、浮動小数点制御 (FPC) レジスター、および制御レジスター内のマスク・ビットを使用することにより、すべての外部割り込み、入出力割り込み、およびマシン・チェック割り込み、そして一部のプログラム割り込みについて、CPU 使用可能または使用不可にすることができます。あるマスク・ビットが 1 であれば、CPU はそれに対応するクラスの割り込みに対して CPU が使用可能にされ、その割り込みが発生可能になります。

マスク・ビットが 0 のときは、CPU はそれに対応する割り込みに対して使用不可にされます。入出力割り込みを起こす条件は、保留のままになります。外部割り込み条件は、保留状態のままになるか、または原因が取り除かれるまで持続します。マシン・チェック割り込み条件は、タイプに応じて、無視されるか、保留状態のままになるか、CPU をチェック停止状態にします。禁止されたプログラム割り込み条件は無視されます。ただし、原因の一部は条件コードの設定によって示されることもあり、また IEEE 例外では FPC レジスター内にフラグが設定されます。HFP 有効数字および HFP 指数アンダーフローのプログラム・マスク・ビットの設定は、対応する条件が発生したときに HFP 演算がどのように完了するかに影響を与えます。同様に、FPC レジスター内の IEEE マスク・ビットの設定は、対応する条件が発生したときに BFP 演算がどのように完了するかに影響を与えます。

プログラミング上の注意:

1. PSW のマスク・ビットを使用することにより、マスク可能なほとんどの割り込みを禁止できます。したがって、1 つの割り込みにより取り込まれた新 PSW により後続の割り込みを禁止することができます。さらに、マスク・ビットを使用して、割り込み優先順位の階層を確立し、あるクラスの条件が発生したときに別のクラスの条件を処理しているプログラムに割り込むことはできるが、その逆はできな

いように設定することができます。必要なハウスキーピング・ステップが終わらないうちに割り込み処理ルーチンが割り込まれるのを防ぐためには、新 PSW では、現在のクラスまたはそれより優先順位が低いクラスの割り込みが生じないように、CPU を使用不可にする必要があります。

2. 制御レジスター内のマスク・ビットは割り込みプロシージャの一環としては変更されないため、これらのマスクを使用して、前の割り込みの直後に同じクラスの割り込みが生じるのを防ぐことはできません。制御レジスター内のマスク・ビットは、同一クラス内の一部のソースについてのみ選択的に CPU を使用可能にし、他のソースについては使用不可にするための手段を提供します。
3. 幾つかのプログラム割り込みには、制御ビットのみがあり、PSW 内に対応するマスク・ビットがないものがあります。この種のビットには、FPC レジスター内の IEEE マスク・ビット、制御レジスター 8 のビット位置 48-63 のモニター・マスク、および制御レジスター 1 の 1 次スペース切り替えイベント制御ビット (ビット 57) があります。この性質を持つビットが「マスク」ビットと見なされるかどうかは幾分恣意的なもので、ビットが 1 のときに割り込みが使用可能にされるという極性を持つ場合のみ、マスク・ビットと見なされます。したがって、例えば、SSM 抑止制御ビット (制御レジスター 0 のビット 33) はマスク・ビットと見なされますが、AFP レジスター制御ビット (制御レジスター 0 のビット 45) はマスク・ビットとは見なされません。このような制御ビットの極性とは関係なく、他のプログラム割り込みを回避するには、割り込み処理ルーチンは、これらのビットが適正に設定されるまでは、これらのビットにより左右されるような命令を発行しないようにする必要があります。

浮動割り込み条件の取り扱い

構成内のいずれかの CPU に提示できる割り込み条件を、浮動割り込み条件と呼びます。この条件は、構成内の、対応する割り込みに対して使用可能にされていて、かつその割り込みを実行できる最初の CPU に提示されます。この時点でその条件は消去され、構成内の他の CPU には提示されません。CPU が割り込みを実行できないのは、チェック停止状態にあるとき、無効なプレフィックスを持っているとき、早期に認識されるタイプの指定例外を原因とするプログラム割り込みのストリングの中にあるとき、または、停止状態にあるときです。ただし、速度制御が命令ステップに設定されている CPU は、開始キーが活動化されたときに割り込みを実行できます。

サービス信号条件、入出力条件、およびある種のマシン・チェック条件は、浮動割り込み条件です。

命令長コード

命令長コード (ILC) は 2 個のビット位置を占め、最後に実行された命令の長さを示します。旧 PSW 内の命令アドレスが次の順番に当たる命令を指しているときは、この命令長コードから、割り込みの原因となった命令を突き止めることができます。24 ビット・アドレッシング・モードでは、BRANCH AND LINK 命令によっても ILC が提供されます。

プログラム割り込みおよび監視プログラム割り込みの場合の ILC は、それぞれ、実記憶位置 141 および 137 のバイトのビット位置 5 および 6 に格納されます。外部割り込み、入出力割り込み、マシン・チェック割り込み、および再始動割り込みの場合は、ILC は、最後に実行された命令の長さとは関係がないので、保管されません。

監視プログラム呼び出し割り込みおよびプログラム割り込みの場合は、0 以外の ILC は、最後に実行された命令の長さをハーフワード数で示します。その命令は、命令アドレスが奇数であるために指定例外が認識された命令か、または、命令の取り出し時に、アクセス例外 (アドレッシング例外、ASCE タイプ例外、ページ変換例外、保護例外、領域変換例外、セグメント変換例外、または変換指定例外) が認識された命令です。EXECUTE により命令が実行される時は、命令長コード 2 は、ターゲット命令ではなく EXECUTE の長さを示すように設定されます。

0 以外の命令長コードの値は、命令の左端 2 ビットに関連付けられています。この値は、命令コードが割り当てられているかどうか、または命令がインストールされているかどうかには無関係です。次の表は、命令長コードの意味を要約したものです。

ILC		命令の ビット 0-1	命令長
10 進数	2 進数		
0	00		使用不可能
1	01	00	1 ハーフワード
2	10	01	2 ハーフワード
2	10	10	2 ハーフワード
3	11	11	3 ハーフワード

ILC が 0 の場合

プログラム割り込みの後で命令長コードが 0 のときは、旧 PSW に格納されている命令アドレスが、割り込みの原因となった命令を指してはいないことを示します。

ILC が 0 になるのは、早期の例外認識の一環として、PSW フォーマット・エラーを原因とする指定例外が認識され、しかも、LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、または割り込みにより PSW が導入された場合です。(6-9 ページの『PSW に関連した例外』を参照してください。)

LOAD PSW、LOAD PSW EXTENDED、または PROGRAM RETURN の場合は、その命令または EXECUTE の命令アドレスは、新 PSW の中の命令アドレスで置き換えられています。何らかの割り込みが原因で無効 PSW が導入された場合は、PSW フォーマット・エラーを命令に起因するものと考えられません。

LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、および監視プログラム呼び出し割り込みの場合は、ILC が 0 である指定例外と同時に、PER イベントが示されることがあります。

スペース切り替えイベントと PSW フォーマット・エラーの両方を引き起こす PROGRAM RETURN 命令の場合は、スペース切り替えイベントが認識されますが、ILC が 0 または 1 (EXECUTE の場合は 0 または 2) のどちらになるかは予測不能です。

命令取り出し時の例外の際の ILC

命令へのアクセスを禁止する例外が原因でプログラム割り込みが生じたときは、その命令は実行されたものと見なされますが、命令の最初の 2 ビットに基づいて命令長コードを設定することはできません。この場合の ILC の意味について考えるときは、次の 2 つの状況が区別されます。

1. 奇数の命令アドレスが原因で指定例外が認識された場合、または、命令を取り出すときに、アドレッシング例外、記憶保護例外、または変換指定例外が検出された場合は、ILC は 1、2、または 3 に設定されます。これは、命令アドレスが、2 にこの値を掛けた値だけ増加したことを意味します。命令アドレスの値の増加が、2、4、または 6 のいずれになるかは、予測不能です。旧 PSW 内の命令アドレスを、ILC が示しているハーフワード位置の数だけ減らすことにより、PSW 内の最初に入っていた命令アドレスを求めることができます。
2. 命令を取り出しているときに、ASCE タイプ例外、領域変換例外、セグメント変換例外、またはページ

変換例外が認識された場合は、ILC は、恣意的に 1、2、または 3 のいずれかに設定されます。この場合は、操作は無効化され、命令アドレスは増加しません。

命令の最初のハーフワードを取り出すことができても、第 2 または第 3 のハーフワードを取り出すときにアクセス例外が認識された場合は、ILC は必ずしも命令の最初の 2 ビットに関連付けられているとは限りません。これらの場合、ILC は恣意的に 1、2、または 3 に設定されます。上記の状況 1 および 2 で述べたように、命令アドレスは、更新される場合とされない場合があります。

EXECUTE のターゲット命令を取り出しているときに何らかの例外が検出されたときは、ILC は 2 です。

プログラミング上の注意:

1. プログラム割り込みの時点で命令長が 0 以外であるときは、その数に相当するハーフワード位置をプログラムの旧 PSW 内の命令アドレスから引くことにより、最後に実行された命令のアドレスが得られることを示します。ただし、次のいずれかの状況が存在する場合は例外です。
 - a. 無効化をもたらす例外が原因で割り込みが起きた場合。
 - b. 割り込み可能命令の実行前に PER に関連した割り込みが生じ、同時に他のプログラム割り込み条件が示されていない場合。
 - c. LOAD PSW、LOAD PSW EXTENDED、または、ブランチ命令またはリンケージ命令 (SUPERVISOR CALL は含むが MONITOR CALL は含まない) に起因する PER イベントまたはスペース切り替えイベントが原因で、割り込みが起きた場合。
 - d. 命令が取り出されるアドレス・スペースを指定しているアドレス・スペース制御エレメントを含む制御レジスタ (1 または 13) をロードする LOAD CONTROL 命令の記憶域オペランドに関するアドレッシング例外または記憶保護例外が原因で、割り込みが起きた場合。

上記の a および b の場合は、PSW 内の命令アドレスは増加せず、命令アドレスが指す命令は最後に実行された命令と同じものです。旧 PSW 内の命令アドレスが例外の原因となった命令を指しているのは、これらの場合のみです。状況 b は、PSW 内の命令アドレスが実記憶位置 152 にあるダブルワード内の PER アドレスと同じであるという点で、割

り込み可能命令または割り込み不能命令の完了後に示される PER イベントと区別されます。

状況 c の場合は、命令アドレスは操作の一環として置き換えられており、最後に実行された命令のアドレスは、プログラムの旧 PSW に入っているアドレスを使用して計算することはできません。

状況 d の場合は、最後に実行された命令の有効アドレスは計算できますが、命令アドレス・スペースのアドレス・スペース制御エレメントが予測不能なので、対応する実アドレスは不明です。

2. PER イベントが示されている場合は、記憶位置 152 にあるダブルワード内の PER アドレスが、割り込みの原因となった命令 (または該当する場合は EXECUTE 命令) を指しているため、命令長コード (ILC) は余分です。同様に、操作が無効化された場合も、PSW 内の命令アドレスは増加しないため ILC は余分です。この場合に ILC 値が必要であれば、旧 PSW が示している命令の命令コードからその値を求めることができます。
3. 以下のいずれかの状況が存在する場合は、プログラム割り込みの前に最後に実行されたアドレスだけでは、プログラムの問題を突き止めるには不十分です。
 - a. 命令の取り出し中に検出されたアクセス例外が原因で割り込みが発生し、その命令アドレスが、順次操作以外の手段で (つまり、ブランチ命令、リンケージ命令、LOAD PSW、LOAD PSW EXTENDED、割り込み、または、IPL シーケンスの終結により) PSW に導入された場合。
 - b. 命令アドレスが奇数であるため発生した指定例外が原因で割り込みが生じた場合。奇数の命令アドレスは、命令アドレスが PSW に導入された結果としても必然的に発生します。
 - c. 実モードとの間の切り替えを行い、PSW のビット位置 0-7 に無効な値を取り込む STORE THEN OR SYSTEM MASK または SET SYSTEM MASK 命令により早期指定例外が生じ、その結果として割り込みが発生した場合。

上記の状況 a および b の場合は、命令アドレスは最後の命令実行の前の操作で置き換えられ、その先行する操作に関連したプログラム位置のアドレスは使用不能です。

状況 c の場合は、最後に実行された命令のアドレスは使用可能ですが、それに対応する実アドレスは不明です。

4. LOAD PSW、LOAD PSW EXTENDED、または PROGRAM RETURN 命令、あるいは割り込みに起因する早期指定例外が原因で割り込みが生じた場合は、最後に実行された命令は使用不能です。

PSW に関連した例外

現 PSW 内の誤った情報に関連した例外は、その情報が PSW に取り込まれるときに認識されることも、次の命令の実行の一環として認識されることもあります。指定例外条件に該当する PSW 内の誤りを、PSW フォーマット・エラーと呼びます。

早期例外認識

以下のエラー条件の場合は、指定例外に関するプログラム割り込みは、PSW がアクティブになった直後に起きます。

- 未割り当てのビット (0、2-4、24-30、または 33-63) のいずれかが 1 である。
- ビット 12 が 1 である。
- ビット 31 が 0、ビット 32 が 1 であり、ビット 64-96 がすべて 0 ではない。
- ビット 31 および 32 が両方とも 0 であり、ビット 64-103 がすべて 0 ではない。
- ビット 31 が 1 で、ビット 32 が 0 である。

割り込みは、待ち状態が指定されているかどうかに関係なく起きます。PSW が無効であるために、CPU が、保留中の入出力割り込み、外部割り込み、またはマシン・チェック割り込みに対して使用可能にされた場合は、代わりにプログラム割り込みが起き、保留中の割り込みは、プログラム割り込みにより導入される新 PSW のマスク・ビットに基づいて制御されます。

割り込みか、あるいは LOAD PSW、LOAD PSW EXTENDED、または PROGRAM RETURN の実行により、上記のいずれかのエラー条件を持つ PSW が導入された場合は、命令長コードは 0 に設定され、新たに導入された PSW は、変更されずに旧 PSW として格納されます。SET SYSTEM MASK または STORE THEN OR SYSTEM MASK の実行により上記のエラー条件の 1 つが導入された場合は、命令長コードは 2 に設定され、命令アドレスは 4 だけ増加します。システム・マスク・フィールドに導入された無効な値を含む PSW は、旧 PSW として保管されます。

遅延例外認識

以下の条件の場合は、例外は次の命令の実行の一環として認識されます。

- PSW に奇数の命令アドレスが含まれている (PSW のビット 127 が 1 である) ために指定例外が認識される場合。
- 命令アドレスが指している位置、または指定された命令アドレスで始まる命令の第 2 または第 3 ハーフワードの位置に、アクセス例外 (アドレッシング、ASCE タイプ、ページ変換、記憶保護、領域変換、セグメント変換、または変換指定) が関連している場合。

これらの条件の下でプログラムの旧 PSW に格納される命令長コードおよび命令アドレスについては、6-7ページの『命令取り出し時の例外の際の ILC』で説明されています。

入出力割り込み、外部割り込み、またはマシン・チェック割り込み条件が保留状態にあるときに、PSW により CPU がその条件に対して使用可能にされた場合は、該当の割り込みが発生し、PSW は遅延認識される例外の有無について検査されません。同様に、待ち状態を指定している PSW についても、遅延認識される例外の有無は検査されません。

プログラミング上の注意:

1. アドレッシング例外または記憶保護例外が起きたときは、LOAD ADDRESS SPACE PARAMETERS、LOAD PSW、LOAD PSW EXTENDED、PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、RESUME PROGRAM、SET PREFIX、SET SECONDARY ASN、SET SYSTEM MASK、STORE THEN AND SYSTEM MASK、および STORE THEN OR SYSTEM MASK の実行は抑止され、したがって、例外を引き起こしたプログラムに関する情報は、プログラムの旧 PSW から得られます。
2. 命令の第 1 ハーフワードは取り出せたが、第 2 または第 3 のハーフワードを取り出すときにアクセス例外が認識されたという場合は、ILC は必ずしもその命令コードに関連したものであるとは限りません。
3. 割り込みにより導入された新 PSW に PSW フォーマット・エラーがある場合は、一連の割り込みが起きることがあります。(6-44ページの『割り込みの優先順位』を参照してください。)

外部割り込み

外部割り込みは、CPU が構成の内外部で発信される各種の信号に応答するための手段を提供します。

外部割り込みが起きると、旧 PSW が実記憶位置 304-319 に格納され、実記憶位置 432-447 から新 PSW が取り出されます。

割り込みのソースは、実記憶位置 134-135 に格納されている割り込みコードが示しています。命令長コードは格納されません。

さらに、誤動作警報、緊急信号、および外部呼び出し条件については、16 ビットの CPU アドレスが割り込みのソースに関連付けられ、実記憶位置 132-133 に格納されます。CPU アドレスが格納されると、割り込みコードのビット 6 が 1 に設定されます。その他のすべての条件については、CPU アドレスは格納されず、割り込みコードのビット 6 は 0 に設定され、実記憶位置 132-133 には 0 が格納されます。

ETR 割り込みおよびサービス信号割り込みの場合は、その割り込みに 32 ビットのパラメーターが関連付けられ、実記憶位置 128-131 に格納されます。外部割り込みコードのビット 5 は、パラメーターが格納されたことを示しています。ビット 5 が 0 であれば、実記憶位置 128-131 内容は変更されていません。

外部割り込み条件には、2 つのタイプがあります。それは、割り込み要求条件が保留にされるものと、割り込みが直接要求されるものです。外部割り込みを直接要求する条件としては、クロック・コンパレーターおよび CPU タイマーがあります。外部割り込みを直接要求する条件が、その要求が受理される前に除去された場合は、要求は保留状態にはされず、割り込みは起きません。逆に言えば、その要求は割り込みによって消去されないため、その条件が持続する場合は、1 回の条件発生の結果、複数の割り込みが起きることがあります。

1 つのソースについて、実際に割り込みが起きる前に幾つかの割り込み要求が生成され、割り込み条件が保留されるタイプのものである場合は、そのソースについては、1 つの要求のみが保存され、保留状態のまま残されます。

特定のソースについて外部割り込みが起きるのは、CPU がそのソースによる割り込みに対して使用可能にされている場合のみです。外部割り込みは、操作単位が完了した時点で起きます。CPU が特定のソースに対して使用

可能にされるかどうかを制御するのは、外部マスク、PSW のビット 7、および、制御レジスター 0 の外部サブクラス・マスク・ビットです。外部割り込みの個々のソースについて、サブクラス・マスク・ビットが 1 つずつ割り当てられています。外部マスク・ビットが 1 で、対応するサブクラス・マスク・ビットが 1 である場合に限り、該当のソースが割り込みを起こすことができます。

保留中の外部割り込み条件に対して CPU が使用可能にされると、その使用可能化の原因となった命令実行または割り込みが完了した時点で、割り込みが起きます。

複数のソースが、同時に外部割り込みに対する要求を出すことがあります。CPU が、同時に保留されている複数の要求に対して使用可能にされると、最も優先順位の高い条件について割り込みが起きます。

外部割り込み要求の優先順位は、高い方から並べると以下のようになります。

- 割り込みキー
- 誤動作警報
- 緊急信号
- 外部呼び出し
- クロック・コンパレーター
- CPU タイマー
- ETR
- サービス信号

すべての要求は、一度に 1 つずつ受理されます。同時に複数の緊急信号が存在する場合、または同時に複数の誤動作警報が存在する場合は、最も小さい CPU アドレスに関連した要求が最初に受理されます。

クロック・コンパレーター

クロック・コンパレーターに関する割り込み要求が出されるのは、次のいずれかの条件が満たされているときです。

1. TOD クロックが設定または非設定状態にあり、クロック・コンパレーターの値が TOD クロックの比較対象部分の値より小さい (比較値は両方とも符号なしの 2 進整数と見なされます)。
2. TOD クロックが、エラー状態または非稼働状態にある。

要求が受理される前に、その要求の原因となった条件が取り除かれた場合は、要求は保留状態にはされず、割り込みは起きません。逆に言えば、その要求は割り込みによって消去されないため、その条件が持続する場合は、

1 回の条件発生の結果、複数の割り込みが起きることがあります。

TOD クロックが設定されるかまたはその状態が変更された場合、以後最大 1.048576 秒間は、クロック・コンパレーターを原因とする割り込み条件は、認識されることも認識されないこともあります。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 52 にあります。このビットは 0 に初期化されます。

クロック・コンパレーター条件は、外部割り込みコード 1004 (16 進数) により示されます。

CPU タイマー

CPU タイマーの値が負 (CPU タイマーのビット 0 が 1) になると、CPU タイマーに関する割り込み要求が発生します。要求が受理される前にこの値が正に変わった場合は、要求は保留状態にはされず、割り込みは起きません。逆に言えば、その要求は割り込みによって消去されないため、その条件が持続する場合は、1 回の条件発生の結果、複数の割り込みが起きることがあります。

TOD クロックが設定されるかまたはその状態が変更された場合、以後最大 1.048576 秒間は、CPU タイマーを原因とする割り込み条件は、認識されることも認識されないこともあります。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 53 にあります。このビットは 0 に初期化されます。

CPU タイマー条件は、外部割り込みコード 1005 (16 進数) により示されます。

緊急信号

CPU が、この CPU をアドレッシングしている SIGNAL PROCESSOR 命令で指定された緊急信号指令を受け入れると、緊急信号に関する割り込み要求が生成されます。この命令は、この CPU で実行されていることも、構成内の他の CPU で実行されていることもあります。この要求は、消去されるまでは、それを受け入れた CPU の中で保存され、保留されます。保留中の要求が消去されるのは、その要求が割り込みを生じさせるか、または CPU がリセットされたときです。

構成内の各 CPU (受け入れ CPU も含む) について別々の緊急信号要求を、受け入れ CPU 内に保留状態で保持するための機能があります。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 49 にあります。このビットは 0 に初期化されます。

緊急信号条件は、外部割り込みコード 1201 (16 進数) により示されます。SIGNAL PROCESSOR 命令を実行した CPU のアドレスは、実記憶位置 132-133 に格納されます。

ETR

現 CPC ポート・グループ内のいずれかのポートでポート可用性が変更されるか、または ETR 警報が発生すると、ETR に関する割り込み要求が生成されます。本書では、ETR に固有の用語については定義されていません。

割り込みが起きる前に、同じ ETR 条件が複数回発生しても、要求が生成されるのは 1 回だけです。この要求は、構成内のすべての CPU に対して生成されます。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 59 にあります。このビットは 0 に初期化されます。

ETR 条件は、外部割り込みコード 1406 (16 進数) により示されます。

外部呼び出し

CPU が、この CPU をアドレッシングしている SIGNAL PROCESSOR 命令で指定された外部呼び出し指令を受け入れると、外部呼び出しに関する割り込み要求が生成されます。この命令は、この CPU で実行されていることも、構成内の他の CPU で実行されていることもあります。この要求は、消去されるまでは、それを受け入れた CPU の中で保存され、保留されます。保留中の要求が消去されるのは、その要求が割り込みを生じさせるか、または CPU がリセットされたときです。

一時点で 1 つの CPU において保留状態にできるのは、1 つの外部要求とプロセッサ・アドレスのみです。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 50 にあります。このビットは 0 に初期化されます。

外部呼び出し条件は、外部割り込みコード 1202 (16 進数) により示されます。SIGNAL PROCESSOR 命令を実行した CPU のアドレスは、実記憶位置 132-133 に格納されます。

割り込みキー

オペレーターが割り込みキーを活動化すると、割り込みキーに関する割り込み要求が生成されます。この要求は、消去されるまでは、CPU の中で保存され保留されます。保留中の要求が消去されるのは、その要求が割り込みを生じさせるか、または CPU がリセットされたときです。

CPU がロード状態にあるときに割り込みキーが活動化された場合は、割り込み要求が生成されるか、それとも該当の条件が失われるかは、モデルによって異なります。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 57 にあります。このビットは 1 に初期化されます。

割り込みキー条件は、外部割り込みコード 0040 (16 進数) により示されます。

誤動作警報

構成内の他の CPU がチェック停止状態になるか、またはその CPU の電源が切れると、誤動作警報に関する割り込み要求が生成されます。この要求は、消去されるまでは、それを受け入れた CPU の中で保存され、保留されます。保留中の要求が消去されるのは、その要求が割り込みを生じさせるか、または CPU がリセットされたときです。

構成内の他の各 CPU について別々の誤動作警報要求を受け入れ CPU 内に保留状態で保持するための機能があります。構成からいずれかの CPU を取り外しても、誤動作警報条件は発生しません。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 48 にあります。このビットは 0 に初期化されます。

誤動作警報条件は、外部割り込みコード 1200 (16 進数) により示されます。この条件を生成した CPU のアドレスは、実記憶位置 132-133 に格納されます。

サービス信号

ある種の構成制御/保守機能 (例えば、モデルによって異なる DIAGNOSE 命令により開始された機能) が完了すると、サービス信号に関する割り込み要求が生成されません。プログラムは、この割り込みに伴って提供される 32 ビットのパラメーターを使用して、どの操作についてこの割り込みが報告されたのかを判別することができます。

サービス信号は浮動割り込み条件の 1 つであり、構成内のこの割り込みを実行できる最初の CPU に提示されません。この割り込み条件が消去されるのは、この条件によりいずれかの CPU で割り込みが起きるか、またはサブシステム・リセットが行われたときです。

サブクラス・マスク・ビットは、制御レジスター 0 のビット位置 54 にあります。このビットは 0 に初期化されます。

サービス信号条件は、外部割り込みコード 2401 (16 進数) により示されます。32 ビットのパラメーターが、実記憶位置 128-131 に格納されます。

入出力割り込み

入出力 (I/O) 割り込みは、CPU が、入出力装置およびチャンネル・サブシステムで発生した条件に応答するための手段を提供します。

入出力割り込みに関する要求は、いつでも発生する可能性があり、また同時に複数の要求が発生することもあります。この要求は、CPU により受け入れられるか、または他の方法 (サブシステム・リセットなど) により消去されるまでは、保存され、保留状態のままになっています。

入出力割り込みは、操作単位が完了した時点で起きます。要求間の優先順位が設定されるので、各 CPU ではそれぞれ一度に 1 つの割り込みだけが処理されます。入出力割り込みサブクラスが異なる割り込みに関する要求間の優先順位は、制御レジスター 6 の中の入出力割り込みサブクラス・マスクの設定と、入出力割り込みサブクラスの数値 (0 が最高優先順位) に従って決定されません。詳しくは、第16章、『入出力割り込み』を参照してください。

CPU が入出力割り込みに対して使用可能にされ、チャンネル・サブシステムが保留中の入出力割り込み条件の優先順位を設定してあれば、その使用可能化の原因となっ

た命令実行または割り込みが完了した時点で、割り込みが起きます。

入出力割り込みが起きると、旧 PSW が実記憶位置 368-383 に格納され、新 PSW が実記憶位置 496-511 から取り出されます。また、実記憶位置 184-191 には、8 バイトの入出力割り込みコードの形で追加情報が格納されます。入出力割り込みコードは、32 ビットのサブシステム ID ワードと 32 ビットの割り込みパラメーターから成っています。

入出力割り込みが起きる可能性があるのは、CPU が、要求を出している割り込みサブクラスに対して使用可能にされているときだけです。CPU が特定の入出力割り込みに対して使用可能にされているかどうかは、入出力マスク・ビット (PSW のビット 6) と、制御レジスター 6 の中の入出力割り込みサブクラス・マスクによって決まります。

入出力割り込みは、0-7 の番号が付いた 8 つの入出力割り込みサブクラスに分類されます。各入出力割り込みサブクラスには、制御レジスター 6 の中のビット位置 32-39 の入出力割り込みサブクラス・マスク・ビットが、1 つずつ関連付けられています。また、各サブチャネルには、それぞれ入出力割り込みサブクラス値が関連付けられています。CPU が特定の入出力割り込みサブクラスの入出力割り込みに対して使用可能にされるのは、PSW のビット 6 が 1 で、かつ制御レジスター 6 の中の関連の入出力割り込みサブクラス・マスク・ビットも 1 である場合のみです。対応する入出力割り込みサブクラス・マスク・ビットが 0 の場合は、CPU は、そのサブクラス値を持つ入出力割り込みに対しては使用不可にされます。PSW のビット 6 が 0 のときは、すべてのサブクラスの入出力割り込みが禁止されます。

マシン・チェック割り込み

マシン・チェック割り込みは、装置の誤動作が起きたことをプログラムに通知するための手段です。プログラムは、渡された情報に基づいて、障害の原因と損傷の程度を判別することができます。

マシン・チェック割り込みが起きると、旧 PSW が実記憶位置 352-367 に格納され、新 PSW が実記憶位置 480-495 から取り出されます。

誤動作の原因と重大度は、実記憶位置 232-239 に格納されている 64 ビットのマシン・チェック割り込みコードに示されます。また、実記憶位置 244-255 および 4608-5119 には、割り込みの原因と障害の場所をさらに詳しく示す情報が格納されていることがあります。

割り込みアクションと、関連情報の格納は、PSW のビット 13 および制御レジスター 14 のビットにより制御されます。詳しくは、第11章、『マシン・チェックの取り扱い』を参照してください。

プログラム割り込み

プログラム割り込みは、プログラムの実行中に起きた例外およびイベントを知らせるために使用されます。

プログラム割り込みが起きると、旧 PSW が実記憶位置 336-351 に格納され、新 PSW が実記憶位置 464-479 から取り出されます。

割り込み原因は割り込みコードに示されます。割り込みコードは実記憶位置 142-143 に格納されます。命令長コードは実記憶位置 141 のバイトのビット位置 5 および 6 に格納され、そのバイトのその他のビットは 0 に設定されます。実記憶位置 140 にはすべて 0 が格納されます。原因によっては、割り込みの理由を示す追加情報が実記憶位置 144-183 に格納されていることがあります。

PER イベントおよび暗号操作例外の場合を除き、割り込みコードの右端 7 個のビット位置に格納されたコード値が、割り込みの原因となった条件を示しています。ここに提示されるのは、一度に 1 つの条件だけです。割り込みコードのビット 0-7 は、0 に設定されます。

PER イベントの場合は、割り込みコードのビット 8 が 1 に設定されます。これが唯一の条件である場合は、ビット 0-7 および 9-15 は 0 に設定されます。PER イベントと同時に他のプログラム割り込み条件も存在する場合は、ビット 8 が 1 に設定されるほか、その条件に応じてビット 0-7 および 9-15 が設定されます。

暗号操作例外は、割り込みコード 0119 (16 進数) により示されます。ただし、PER イベントも生じている場合は、0199 (16 進数) で示されます。

暗号操作例外は、割り込みコード 0119 (16 進数) により示されます。ただし、PER イベントも生じている場合は、0199 (16 進数) で示されます。

あるプログラム割り込みに対応するマスク・ビットがある場合、そのプログラム割り込みが起きる可能性があるのは、そのマスク・ビットが 1 のときだけです。PSW 内のプログラム・マスクは 4 つの例外を制御し、FPC レジスター内の IEEE マスクは IEEE 例外を制御しま

す。制御レジスター 0 のビット 33 は SET SYSTEM MASK が特殊操作例外を起こすかどうかを制御し、制御レジスター 8 のビット 48-63 はモニター・イベントが引き起こす割り込みを制御します。さらに、PER イベントが引き起こす割り込みを制御する階層形式のマスクがあります。制御マスク・ビットが 0 であれば、それに該当する条件は無視されます。つまり、その条件は保留のまま残されることはありません。

プログラミング上の注意:

1. プログラム割り込み用の新 PSW に PSW フォーマット・エラーがあるか、または、命令取り出し中にその PSW が原因で例外が認識された場合は、一連のプログラム割り込みが起きることがあります。このような一連のプログラム割り込みを打ち切る方法については、6-44 ページの『割り込みの優先順位』を参照してください。
2. プログラム例外として示される条件の幾つかは、チャンネル・サブシステムにより認識されることもあり、その場合は、例外はサブチャンネル状況ワードまたは拡張状況ワードに示されます。

データ例外コード (DXC)

データ例外が原因でプログラム割り込みが起きたときは、データ例外コード (DXC) が記憶位置 147 に格納され、記憶位置 144-146 には 0 が格納されます。DXC は、データ例外条件のタイプを識別します。AFP レジスター (追加浮動小数点レジスター) 制御ビット (制御レジスター 0 のビット 45) が 1 のときは、浮動小数点制御 (FPC) レジスターの DXC フィールドにも DXC が入れられます。その他のプログラム例外が報告されても、FPC レジスター内の DXC フィールドは変更されません。DXC は、データ例外の原因を示す 8 ビットのコードです。図 6-2 および 6-15 ページの図 6-3 は、データ例外とデータ例外コードを示しています。

データ例外の場合のプログラム割り込みの優先順位

複数のデータ例外が該当し、かつどれも使用可能にされている場合は、最も小さい DXC 値を持つ例外が報告されます。したがって、例えば、DXC 2 (BFP 命令) はどの IEEE 例外条件よりも優先されます。

指定例外と AFP レジスター・データ例外の両方が該当する場合は、どちらが報告されるかは予測不能です。

DXC (16 進数)	データ例外
00	10 進オペランド
01	AFP レジスター
02	BFP 命令
08	IEEE 不正確および切り捨て
0C	IEEE 不正確および増加
10	IEEE アンダーフロー、正確
18	IEEE アンダーフロー、不正確および切り捨て
1C	IEEE アンダーフロー、不正確および増加
20	IEEE オーバーフロー、正確
28	IEEE オーバーフロー、不正確および切り捨て
2C	IEEE オーバーフロー、不正確および増加
40	IEEE ゼロ除算
80	IEEE 無効操作

図 6-2. データ例外コード (DXC)

例外	該当する 命令タイプ	CR0.45 の影響	FPC マスク	FPC フラグ	DXC (2進数)	割り込み アクション	実記憶位置 147 への DXC の 格納	FPC バイト 2 への DXC の 格納
10 進オペランド	10 進数 ¹	0	なし	なし	0000 0000	抑止または中 止	あり	なし
		1					あり	あり
AFP レジスタ	FPS & HFP	0*	なし	なし	0000 0001	抑止	あり	なし
BFP 命令	BFP	0*	なし	なし	0000 0010	抑止	あり	なし
IEEE 無効操作	BFP	1*	0.0	1.0	1000 0000	抑止	あり	あり
IEEE ゼロ除算	BFP	1*	0.1	1.1	0100 0000	抑止	あり	あり
IEEE オーバーフロー	BFP	1*	0.2	1.2	0010 xy00	完了	あり	あり
IEEE アンダーフロー	BFP	1*	0.3	1.3	0001 xy00	完了	あり	あり
IEEE 不正確	BFP	1*	0.4	1.4	0000 1y00	完了	あり	あり

説明:

¹ 10 進オペランド・データ例外は、10 進命令 (第 8 章) と、一般命令 COMPRESSION CALL および CONVERT TO BINARY (第 7 章) に適用されます。

0* この例外が認識されるのは、CR0.45 が 0 の場合のみです。

1* この例外が認識されるのは、CR0.45 が 1 の場合のみです。

xy IEEE オーバーフローおよび IEEE アンダーフローの場合は、DXC のビット 4 および 5 は、2 進数 00、10、または 11 に設定されます。これらの値は、それぞれ、結果が「正確である」、「不正確で切り捨てられた」、または「不正確で増加された」ことを示します。

y 「IEEE 不正確」の場合は、DXC のビット 5 は 0 または 1 に設定されます。これらの値は、それぞれ、結果が「不正確で切り捨てられた」または「不正確で増加された」ことを示します。

BFP 2 進浮動小数点命令 (第 19 章)。

FPS 浮動小数点サポート命令 (第 9 章)。

HFP 1 進浮動小数点命令 (第 18 章)。

図 6-3. データ例外

プログラム割り込みの条件

以下、個々のプログラム割り込み条件について詳しく説明します。

アドレッシング例外

アドレッシング例外が認識されるのは、CPU が、構成内で使用可能になっていない主記憶位置を参照しようとした場合です。ある主記憶位置が構成内で使用できないのは、その位置がインストールされていないか、該当の記憶装置が構成に含まれていないか、またはその記憶装置の電源がオフになっているときです。構成内で使用可能でない記憶位置を指定しているアドレスを、無効アドレスと言います。

命令のアドレスが無効な場合は、操作は抑止されます。同様に、EXECUTE のターゲット命令のアドレスが無

効な場合も、操作は抑止されます。また、テーブルまたはテーブル・エンタリーにアクセスしようとしたときにアドレッシング例外が検出された場合は、操作単位が抑止されます。この規則が適用されるテーブルおよびテーブル・エンタリーは、ディスパッチ可能単位制御テーブル、1 次 ASN 第 2 テーブル・エンタリーのほか、アクセス・リスト、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、ページ・テーブル、リンケージ・テーブル、エンタリー・テーブル、ASN 第 1 テーブル、ASN 第 2 テーブル、権限テーブル、リンケージ・スタック、およびトレース・テーブル内のエンタリーです。領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、およびページ・テーブルに対する参照についてアドレッシング例外が検出された場合は、動的アドレス変換のときの暗黙参照の場合も、また LOAD REAL ADDRESS、STORE REAL ADDRESS、および TEST

PROTECTION の実行に関連した参照の場合も、操作は抑止されます。同様に、ディスパッチ可能単位制御テーブル、1次 ASN 第 2 テーブル・エントリ、アクセス・リスト、ASN 第 2 テーブル、または権限テーブルへのアクセスについてアドレッシング例外が検出された場合は、暗黙的なアクセス・レジスタ変換の場合も、また LOAD REAL ADDRESS、STORE REAL ADDRESS、TEST ACCESS、または TEST PROTECTION の一環としてのアクセス・レジスタ変換の場合も、操作は抑止されます。実行が抑止される一部の命令を除き、オペランド・アドレスの変換はできても、そのアドレスが使用不能の位置を示している場合は、操作は中止されます。6-17ページの図6-4を参照してください。

中止の場合は、変更されるのは結果フィールドだけです。ここで言う「結果フィールド」には、条件コード、レジスタ、および、用意されている記憶位置のうち命令により変更されるものとして指定されている位置が含まれます。したがって、ある命令が、記憶域内の特定フ

ィールドの内容のみを変更することになっているときに、そのフィールドのすべてのバイトが、構成内の使用可能でない位置にある場合は、操作は抑止されます。構成内で、オペランド位置の一部は使用でき、一部は使用できないという場合は、構成内の使用可能に部分には格納を行うことができます。

命令を取り出すとき、または命令取り出しに関連した DAT テーブルを取り出すときにアドレッシング例外が起きたときは、ILC が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲットの取り出しに関連して例外が起きたときは、ILC は 2 です。

命令取り出しに関連していないすべてのアドレッシング例外では、ILC は 1、2、3 のいずれかで、これは参照の原因となった命令の長さを示します。

アドレッシング例外は、16 進数 0005 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0085)。

例外	アクションの対象			
	テーブル・エントリー取り出し ¹	テーブル・エントリー格納 ²	命令取り出し	オペランド参照
アドレッシング例外	抑止	抑止	抑止	IPTE、LASP、LPSW、LPSWE、MSCH、PLO ⁶ 、RP、SCKC、SPT、SPX、SSCH、SSM、STCRW、STNSM、STOSM、TPI、および TPROT の場合は抑止。その他の場合はすべて中止。 ⁴
キー制御保護の場合の保護例外	--	--	抑止	IPTE、LASP、LPSW、LPSWE、MSCH、PLO ⁶ 、RP、SCKC、SPT、SPX、SSCH、SSM、STCRW、STNSM、STOSM、および TPI ⁵ の場合は抑止。その他の場合はすべて中止。 ⁴
アクセス・リスト制御保護の場合の保護例外	--	--	--	抑止
ページ保護の場合の保護例外	--	抑止 ³	--	抑止 ⁵
低アドレス保護の場合の保護例外	--	抑止	--	PTE、STCRW、STNSM、STOSM、および TPI ⁵ の場合は抑止。 その他の場合はすべて中止。 ⁴

説明:

-- 不適用。

1 テーブル・エントリーには、領域テーブル、セグメント・テーブル、ページ・テーブル、リンケージ・テーブル、エントリー・テーブル、ASN 第 1 テーブル、ASN 第 2 テーブル、権限テーブル、ディスパッチ可能単位制御テーブル、1 次 ASN テーブル・エントリー、アクセス・リスト、およびリンケージ・スタックが含まれます。

2 テーブル・エントリーには、リンケージ・スタックとトレース・テーブルが含まれます。

3 ページ保護は、リンケージ・スタックには適用されますが、トレース・テーブルには適用されません。

4 中止の場合は、変更されるのは結果フィールドだけです。ここで言う「結果」フィールドには、条件コードとレジスタのほか、命令により変更されることになっている記憶位置が含まれます。ただし、参照がアクセス例外を引き起こす場合は、記憶位置または記憶キーへの変更は行われません。したがって、命令が主記憶機構内のフィールドの内容だけを変更することになっており、そのフィールドの各バイトがアクセス例外を引き起こすことになる場合は、結果は操作が抑止された場合と同じになります。キー制御保護および低アドレス保護の場合、アクションは中止ではなく抑止です。第 3 章、『記憶機構』の『保護時の抑止』を参照してください。

5 TPI の有効アドレスが 0 のときは、格納アクセスは暗黙的に実記憶位置 184-191 に対して行われ、キー制御保護、ページ保護、および低アドレス保護は適用されません。

6 抑止が起きるのは、比較およびロード操作と比較およびスワップ操作の場合のみです。

図 6-4. アドレッシング例外および記憶保護例外の場合のアクションの要約

AFX 変換例外

AFX 変換例外が認識されるのは、スペース切り替え形式の PROGRAM RETURN、PROGRAM TRANSFER、または SET SECONDARY ASN での ASN 変換、または復元後の SASN が復元後の PASN に等しくないときの PROGRAM RETURN での ASN 変換において、使用された ASN 第 1 テーブル・エントリーのビット 0 が 0 ではない場合です。

変換されている ASN は実記憶位置 174-175 に格納され、実記憶位置 172-173 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1 または 2 です。

AFX 変換例外は、16 進数 0020 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A0)。

ALEN 変換例外

ALEN 変換例外が認識されるのは、アクセス・レジスタ変換時に、次のいずれかの条件が生じている場合です。

1. 使用されたアクセス・レジスタに含まれているアクセス・リスト・エントリー番号が、有効アクセス・リスト指定が指しているアクセス・リストの終わりを超えるアクセス・リスト・エントリーを示している。
2. アクセス・リスト・エントリーのビット 0 が 0 でない。

アクセス・レジスタの番号が実記憶位置 160 のビット位置 4-7 に格納され、ビット 0-3 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1、2、または 3 です。

ALEN 変換例外は、16 進数 0029 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A9)。

ALE シーケンス例外

ALE シーケンス例外が認識されるのは、アクセス・レジスタ変換で使用されるアクセス・レジスタに含まれている ALESN が、そのアクセス・レジスタが指すアクセス・リスト・エントリー内のアクセス・リスト・エントリー・シーケンス番号 (ALESN) に等しくないときです。

アクセス・レジスタの番号が実記憶位置 160 のビット位置 4-7 に格納され、ビット 0-3 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1、2、または 3 です。

ALE 変換例外は、16 進数 002A のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00AA)。

ALET 指定例外

ALET 指定例外が認識されるのは、アクセス・レジスタ変換に使用されるアクセス・レジスタ内のアクセス・リスト・エントリーのビット位置 0-6 が、すべて 0 ではないときです。ただし、アクセス・レジスタ 0 の場合は、TEST ACCESS で使用される場合を除き、すべて 0 を含むものとして扱われるので、この例外は認識されません。TEST ACCESS は、アクセス・レジスタ 0 の実際の内容を使用します。

操作は抑止されます。

命令長コードは 1、2、または 3 です。

ALET 指定例外は、16 進数 0028 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A8)。

ASCE タイプ例外

ASCE タイプ例外が認識されるのは、動的アドレス変換時に以下のいずれかが真である場合です。

1. 使用するアドレス・スペース制御エレメントが領域第 2 テーブル指定であり、変換する仮想アドレスのビット 0-10 がすべて 0 ではない。
2. 使用するアドレス・スペース制御エレメントが領域第 3 テーブル指定であり、変換する仮想アドレスのビット 0-21 がすべて 0 ではない。

3. 使用するアドレス・スペース制御エレメントがセグメント・テーブル指定であり、変換する仮想アドレスのビット 0-32 がすべて 0 ではない。

この例外は、命令アドレスまたはオペランド・アドレスの変換を必要とする命令の実行中に認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49 ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令長コードは 1、2、または 3 です。

ASCE タイプ例外は、16 進数 0038 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B8)。

ASTE シーケンス例外

ASTE シーケンス例外が認識されるのは、以下のいずれかに該当する場合です。

1. 下記 2 以外の場合に、アクセス・レジスター変換で使用されるアクセス・リスト・エントリーに、そのアクセス・リスト・エントリーが指している ASN 第 2 テーブル・エントリー内の ASN 第 2 テーブル・エントリー・シーケンス番号 (ASTESN) に等しくない ASTESN が含まれている。このアクセス・リスト・エントリーは、使用されたアクセス・レジスターが指しているものです。
2. BRANCH IN SUBSPACE GROUP による ALET 1 のアクセス・レジスター変換時に、ディスパッチ可能単位制御テーブル (DUCT) 内のサブスペース ASTESN (SSASTESN) が、DUCT 内のサブスペース ASTE 起点 (SSASTE0) が指しているサブスペース ASTE の中の ASTESN に等しくない。
3. サブスペース置き換え操作において、ディスパッチ可能単位制御テーブル (DUCT) の中のサブスペース ASTESN (SSASTESN) が、DUCT 内のサブスペース ASTE 起点 (SSASTE0) が指しているサブスペース ASTE の中の ASTESN に等しくない。

上記 1 および 2 の場合は、アクセス・レジスターの番号が実記憶位置 160 のビット位置 4-7 に格納され、ビ

ット 0-3 は 0 に設定されます。上記 3 の場合は、実記憶位置 160 にはすべて 0 が保管されます。

操作は無効化されます。

命令長コードは 1、2、または 3 です。

ASTE シーケンス例外は、16 進数 002C のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00AC)。

プログラミング上の注意: アクセス・レジスター変換にアクセス・レジスター 0 を使用した結果としてこの例外が起きることはあり得ないので、サブスペース置き換え操作で ASTE シーケンス例外が認識された場合に実記憶位置 160 に 0 を格納するのは、固有の指示です。

ASTE 妥当性例外

ASTE 妥当性例外が認識されるのは、以下のいずれかに該当する場合です。

1. 下記 2 以外の場合に、アクセス・レジスター変換に使用されるアクセス・リスト・エントリーが、ビット 0 が 0 ではない ASN 第 2 テーブル・エントリーを指している。このアクセス・リスト・エントリーは、使用されたアクセス・レジスターが指しているものです。
2. BRANCH IN SUBSPACE GROUP による ALET 1 のアクセス・レジスター変換において、ディスパッチ可能単位制御テーブル内のサブスペース ASTE 起点 (SSASTE0) が、ビット 0 が 0 ではない ASN 第 2 テーブル・エントリーを指している。
3. サブスペース置き換え操作で、ディスパッチ可能単位制御テーブル内のサブスペース ASTE 起点 (SSASTE0) が、ビット 0 が 0 ではない ASN 第 2 テーブル・エントリーを指している。

上記 1 および 2 の場合は、アクセス・レジスターの番号が実記憶位置 160 のビット位置 4-7 に格納され、ビット 0-3 は 0 に設定されます。上記 3 の場合は、実記憶位置 160 にはすべて 0 が保管されます。

操作は無効化されます。

命令長コードは 1、2、または 3 です。

ASTE 妥当性例外は、16 進数 002B のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00AB)。

プログラミング上の注意: アクセス・レジスタ変換にアクセス・レジスタ 0 を使用した結果としてこの例外が起きることはあり得ないので、サブスペース置き換え操作で ASTE 妥当性例外が認識された場合に実記憶位置 160 に 0 を格納するのは、固有の指示です。

ASX 変換例外

ASX 変換例外が認識されるのは、スペース切り替え形式の PROGRAM CALL の実行時、スペース切り替え形式の PROGRAM RETURN、PROGRAM TRANSFER、または SET SECONDARY ASN での ASN 変換時、または、復元後の SASN が復元後の PASN に等しくないときの PROGRAM RETURN での ASN 変換時に、使用された ASN 第 1 テーブル・エントリーのビット 0 が 0 ではない場合です。

変換されている ASN は実記憶位置 174-175 に格納され、実記憶位置 172-173 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1 または 2 です。

ASX 変換例外は、16 進数 0021 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A1)。

暗号操作例外

暗号操作例外が認識されるのは、暗号化機能がインストールされて使用可能になっている CPU で、制御レジスタ 0 のビット 61 が 0 のときに、暗号化機能命令が実行された場合です。また、暗号化機能命令が実行されるときに、この CPU には暗号化機能はインストールされていない (または使用可能な状態にない) が、この CPU または構成内の他の CPU で、プログラムに対して暗号化機能を使用可能にできるという場合も、暗号操作例外が認識されます。

暗号化機能命令が実行されるときに、構成内に現在存在するかまたは構成に組み込み可能ななどの CPU にも暗号化機能がインストールされていない場合は、暗号操作例外または操作例外のどちらが認識されるかは、モデルによって異なります。

暗号操作例外が認識された場合は、操作は無効化されます。

命令長コードは 2 です。

暗号操作例外は、16 進数 0119 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0199)。

データ例外

データ例外条件は、6-15ページの図6-3 に示されています。各条件について記載されているように、割り込みが起きるかどうか、マスク・ビットにより制御される場合とされない場合があります。

マスク不能のデータ例外条件が認識されたときは、必ずデータ例外に関するプログラム割り込みが起きます。

IEEE 例外条件は、それぞれ、浮動小数点制御 (FPC) レジスタ内の 1 つのマスク・ビットにより制御されます。これらの条件の取り扱いについては、19-11ページの『IEEE 例外条件』で説明します。

データ例外が認識されるのは以下の場合です。

- **10 進オペランド**・データ例外が認識されるのは、10 進オペランドを操作する命令で、無効な 10 進数字または符号コードが見つかったか、オペランドの指定に誤りがある場合です。操作は抑止されますが、EDIT および EDIT AND MARK の場合は操作は中止されます。詳しくは、8-4ページの『10 進オペランド・データ例外』を参照してください。また、COMPRESSION CALL で辞書にエラーが見つかった場合も 10 進オペランド・データ例外が認識されます。その場合は操作は中止されます。10 進オペランド・データ例外は、DXC 0 により通知されます。
- **AFP レジスタ**・データ例外が認識されるのは、制御レジスタ 0 のビット 45 が 0 であり、かつ、浮動小数点サポート (FPS) 命令または 16 進浮動小数点 (HFP) 命令が、0、2、4、または 6 以外の浮動小数点レジスタを指定しているときです。操作は抑止され、DXC 1 によりそれが通知されます。
- **BFP 命令**データ例外が認識されるのは、制御レジスタのビット 45 が 0 のときに、BFP 命令が実行された場合です。操作は抑止され、DXC 2 によりそれが通知されます。
- **IEEE 例外条件**データ例外が認識されるのは、BFP 命令で例外条件が検出された場合です。操作は、条件のタイプに応じて、抑止されるかまたは完了します。詳しくは、19-11ページの『IEEE 例外条件』を参照してください。

命令長コードは 1、2、または 3 です。

データ例外は、16 進数 0007 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0087)。

10 進除算例外

10 進除算例外が認識されるのは、10 進除算において、除数が 0 のとき、または商が指定されたデータ・フィールドのサイズを超えるときです。

10 進除算例外が示されるのは、除数と被除数の両方の符号コードが有効であり、例外が起きたときに使用されていた数字が有効である場合に限られます。

操作は抑止されます。

命令長コードは 2 または 3 です。

10 進除算例外は、16 進数 000B のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008B)。

10 進オーバーフロー例外

10 進オーバーフロー例外が認識されるのは、10 進数演算の宛先フィールドが短すぎて結果を収容できないために、0 以外の数字が 1 つ以上失われる場合です。

10 進オーバーフロー・マスク (PSW ビット 21) により、割り込みを禁止できます。

操作は完了します。オーバーフロー数字を無視することにより演算結果が求められ、条件コード 3 が設定されます。

命令長コードは 2 または 3 です。

10 進オーバーフロー例外は、16 進数 000A のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008A)。

実行例外

実行例外が認識されるのは、EXECUTE のターゲット命令が別の EXECUTE である場合です。

操作は抑止されます。

命令長コードは 2 です。

実行例外は、16 進数 0003 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0083)。

EX 変換例外

EX 変換例外が認識されるのは、PROGRAM CALL による PC 番号変換において、PC 番号のエントリー指標部分に示されているエントリー・テーブル・エントリーが、リンケージ・テーブル・エントリーが指しているエントリー・テーブルの終わりを越える場合です。

実記憶位置 172 にあるワードのビット位置 12-31 に PC 番号が格納され、そのワードの左端 12 ビットは 0 に設定されます。

操作は無効化されます。

命令長コードは 2 です。

EX 変換例外は、16 進数 0023 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A3)。

拡張権限例外

拡張権限例外が認識されるのは、アクセス・レジスター変換において、以下のすべてが真である場合です。

1. 使用されたアクセス・リスト・エントリー内の専用ビットが 1 である。
2. アクセス・リスト・エントリー内のアクセス・リスト・エントリー権限指標 (ALEAX) が、制御レジスター 8 の中の拡張権限指標 (EAX) に等しくない。
3. 次のいずれかが真である。
 - a. EAX が指している権限テーブル・エントリーが、使用された権限テーブルの長さを超えている。この権限テーブルは、使用されたアクセス・リスト・エントリーに示されている ASN 第 2 テーブル・エントリーが指しているものです。
 - b. EAX が示している 2 次権限ビットが 0 である。

このアクセス・リスト・エントリーは、使用されたアクセス・レジスターが指しているものです。

アクセス・レジスターの番号が実記憶位置 160 のビット位置 4-7 に格納され、ビット 0-3 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1、2、または 3 です。

拡張権限例外は、16 進数 002D のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00AD)。

固定小数点除算例外

固定小数点例外が認識されるのは、以下のいずれかに該当する場合です。

1. 符号付きまたは符号なしの 2 進数除算において、結果が 32 ビットとして定義されているか、除数が 0 であるか、または、商が 32 ビットの符号付きまたは符号なしの 2 進整数で表せない。
2. 符号付きまたは符号なしの 2 進数除算において、結果が 64 ビットとして定義されているか、除数が 0 であるか、または、商が 64 ビットの符号付きまたは符号なしの 2 進整数で表せない。
3. CONVERT TO BINARY の結果が、32 ビットの符号付き 2 進整数 (結果が 32 ビットとして定義されている場合)、または 64 ビットの符号付き 2 進整数 (結果が 64 ビットとして定義されている場合) で表せない。

除算の場合は、操作は抑止されます。CONVERT TO BINARY (CVB) の場合は、レジスターに収まらない左端ビットを無視して、実行が完了します。CONVERT TO BINARY (CVBG) の実行は抑止されます。

命令長コードは 1、2、または 3 です。

固定小数点除算例外は、16 進数 0009 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0089)。

固定小数点オーバーフロー例外

固定小数点オーバーフロー例外が認識されるのは、符号付き 2 進数算術演算または符号付き左シフト操作でオーバーフローが起きた場合です。

固定小数点オーバーフロー・マスク (PSW ビット 20) により、割り込みを禁止できます。

操作は完了します。オーバーフロー情報を無視することにより演算結果が求められ、条件コード 3 が設定されません。

命令長コードは 1、2、または 3 です。

固定小数点オーバーフロー例外は、16 進数 0008 のプログラム割り込みコードにより示されます (ただし、同

時に PER イベントも示されている場合は 16 進数 0088)。

HFP 除算例外

HFP 除算例外が認識されるのは、HFP 除算において除数の小数部が 0 の場合です。

操作は抑止されます。

命令長コードは 1 または 2 です。

HFP 除算例外は、16 進数 000F のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008F)。

HFP 指数オーバーフロー例外

HFP 指数オーバーフロー例外が認識されるのは、HFP 演算の結果の指数部が 127 を超え、結果の小数部が 0 でない場合です。

操作は完了します。小数部は正規化され、結果の符号と小数部は正しい値を維持します。結果の指数部は、正しい指数部より 128 小さく設定されます。

命令長コードは 1、2、または 3 です。

HFP 指数例外は、16 進数 000C のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008C)。

HFP 指数アンダーフロー例外

HFP 指数アンダーフロー例外が認識されるのは、HFP 演算の結果の指数部が 0 より小さく、結果の小数部が 0 でない場合です。拡張フォーマットの HFP 結果の場合は、HFP 指数アンダーフローが示されるのは、指数部の高位でアンダーフローが起きた場合のみです。

HFP 指数アンダーフロー・マスク (PSW ビット 22) により、割り込みを禁止できます。

操作は完了します。HFP 指数アンダーフロー・マスクは、演算の結果にも影響を与えます。このマスク・ビットが 0 のときは、符号、指数部、および小数部は 0 に設定され、結果は真の 0 になります。このマスク・ビットが 1 のときは、小数部は正規化され、指数部は正しい指数部より 128 大きくされ、符号と小数部は正しい値を維持します。

命令長コードは 1、2、または 3 です。

HFP 指数アンダーフロー例外は、16 進数 000D のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008D)。

HFP 有効数字例外

HFP 有効数字例外が認識されるのは、HFP 加算または減算の結果の小数部が 0 のときです。

HFP 有効数字マスク (PSW ビット 23) により、割り込みを禁止できます。

操作は完了します。HFP 有効数字マスクは、演算の結果にも影響を与えます。このマスク・ビットが 0 のときは、結果を真の 0 で置き換えて演算が完了します。このマスク・ビットが 1 のときは、結果の指数部に何も変更を加えずに演算が完了します。

命令長コードは 1 または 2 です。

HFP 有効数字例外は、16 進数 000E のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 008E)。

HFP 平方根例外

HFP 平方根例外が認識されるのは、HFP SQUARE ROOT 命令の第 2 オペランドが 0 より小さい場合です。

操作は抑止されます。

命令長コードは 2 または 3 です。

HFP 平方根例外は、16 進数 001D のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 009D)。

LX 変換例外

LX 変換例外が認識されるのは、PROGRAM CALL による PC 番号変換時に、次のいずれかの条件が生じている場合です。

1. PC 番号のリンケージ指標部分が指すリンケージ・テーブル・エントリーが、使用されているリンケージ・テーブル指定が指すリンケージ・テーブルの終わりを超えている。
2. リンケージ・テーブル・エントリーのビット 0 が 0 でない。

実記憶位置 172 にあるワードのビット位置 12-31 に PC 番号が格納され、そのワードの左端 12 ビットは 0 に設定されます。

操作は無効化されます。

命令長コードは 2 です。

LX 変換例外は、16 進数 0022 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A2)。

モニター・イベント

モニター・イベントが認識されるのは、MONITOR CALL が実行されたときに、命令のビット 12-15 に指定されたクラスに対応する制御レジスター 8 のモニター・マスク・ビットが、1 の場合です。制御レジスター 8 の情報のフォーマットは、以下のとおりです。

制御レジスター 8

Monitor Masks	
48	63

モニター・マスク・ビット (制御レジスター 8 のビット 48-63) は、それぞれモニター・クラス 0-15 に対応しています。モニター・マスク・ビットは、同時に幾つでもオンにすることができます。現時点でオンになっているすべてのモニター・イベント・クラスが、モニターされます。これらのマスク・ビットは 0 に初期化されます。

MONITOR CALL が実行されるときに、対応するモニター・マスク・ビットが 1 の場合は、モニター・イベントに関するプログラム割り込みが起きます。

実記憶位置 148-149 および 176-183 に、追加情報が格納されます。これらの位置に格納される情報のフォーマットは、以下のとおりです。

実記憶位置 148-149

00000000	Monitor Class No.
0	8 15

実記憶位置 176-183

Monitor Code
0

MONITOR CALL 命令のビット位置 8-15 の内容が実記憶位置 149 に格納され、モニター・クラス番号を構成します。実記憶位置 148 には 0 が格納されます。命令の B1 フィールドと D1 フィールドに指定されている有効アドレスはモニター・コードを形成し、実記憶位置 176 にあるダブルワードに格納されます。このアドレスの値は、アドレッシング・モード (現 PSW のビット 31 および 32) によって異なります。24 ビット・アドレッシング・モードでは、アドレスのビット 0-39 は 0 であり、31 ビット・アドレッシング・モードでは、ビット 0-32 は 0 です。

操作は完了します。

命令長コードは 2 です。

モニター・イベントは、16 進数 0040 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00C0)。

オペランド例外

オペランド例外が認識されるのは、以下のいずれかが真である場合です。

1. CLEAR SUBCHANNEL、HALT SUBCHANNEL、MODIFY SUBCHANNEL、RESUME SUBCHANNEL、START SUBCHANNEL、STORE SUBCHANNEL、または TEST SUBCHANNEL を実行しようとしたときに、汎用レジスター 1 のビット位置 32-47 に 16 進数 0001 が含まれていない。ただし、レジスターのビット位置 32-39 に 1 が含まれていることが原因で生じる例外は、認識されることも認識されないこともあります。
2. MODIFY SUBCHANNEL を実行しようとしたときに、SCHIB オペランドのワード 1 のビット 1 および 6 が 0 でないか、または SCHIB オペランドのワード 6 のビット 9-10 および 25-30 がすべて 0 ではない。
3. MODIFY SUBCHANNEL を実行しようとしたときに、SCHIB オペランドのビット 9 および 10 が両方とも 1 である。
4. RESET CHANNEL PATH を実行しようとしたときに、汎用レジスター 1 のビット 40-55 がすべて 0 ではない。

5. SET ADDRESS LIMIT を実行しようとしたときに、汎用レジスター 1 のビット 32 および 48-63 がすべて 0 ではない。
6. SET CHANNEL MONITOR を実行しようとしたときに、汎用レジスター 1 のビット 62 は 1 で、汎用レジスター 2 のビット 59-63 はすべて 0 ではない。
7. SET CHANNEL MONITOR を実行しようとしたときに、汎用レジスター 1 のビット位置 40-47 の値が無効である。
8. SET CHANNEL MONITOR を実行しようとしたときに、汎用レジスター 1 のビット 36-39 および 48-60 がすべて 0 ではない。
9. SET CHANNEL MONITOR を実行しようとしたときに、汎用レジスター 1 のビット 39 は 1 で、汎用レジスター 1 のビット 40-47 はすべて 0 ではない。
10. START SUBCHANNEL を実行しようとしたときに、ORB オペランドのワード 1 のビット 5、13、および 25-28 がすべて 0 ではない。
11. START SUBCHANNEL を実行しようとしたときに、ORB オペランドのワード 1 のビット 11 が 0 でない。この例外は、認識される場合とされない場合があります。

操作は抑止されます。

命令長コードは 2 です。

オペランド例外は、16 進数 0015 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0095)。

操作例外

操作例外が認識されるのは、CPU が、無効な命令コードを持つ命令を実行しようとした場合です。その命令コードは割り当てられていないものであるか、またはその命令コードを持つ命令が CPU にインストールされていない可能性があります。

操作は抑止されます。

命令長コードは 1、2、または 3 です。

操作例外は、16 進数 0001 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0081)。

プログラミング上の注意:

1. 一部のモデルでは、例えば特殊機構や特注機構の補助のためまたはそれらの一部として提供されている命令などのように、本書では説明していない命令が用意されています。したがって、本書で説明していない命令コードを使用しても、必ず操作例外が認識されるとは限りません。さらに、この種の命令では、特定の操作モードがセットアップされたり、後続の命令の実行に影響するような変更がマシンに加えられたりすることもあります。このような操作が生じるのを回避するために、本書で説明されていない命令コードを持つ命令を実行するのは、その命令コードに関連した特定の機能が必要な場合に限るようにしてください。
2. 2 バイト命令フォーマットでは、現在、命令コード 00 は割り当てられていません。将来この命令が割り当てられる可能性はほとんどありません。

ページ変換例外

ページ変換例外が認識されるのは、ページ無効ビットが 1 のときです。

この例外は、命令アドレスまたはオペランド・アドレスの変換にページ・テーブル・エントリーを必要とする命令の実行の一部として認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。また、MOVE PAGE のオペランド・アドレスの場合も、条件コード・オプション・ビット (汎用レジスタ 0 のビット 55) が 1 であれば、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の参照時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

ページ変換例外は、16 進数 0011 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0091)。

PC 変換指定例外

PC 変換指定例外が認識されるのは、PROGRAM CALL による PC 番号変換に使用されるエントリー・テーブル・エントリー (ETE) について、次のいずれかが真である場合です。

1. PROGRAM CALL 操作が、24 ビットまたは 31 ビット・アドレッシング・モード (PSW のビット 31 が 0) での基本操作 (ETE のビット 128 が 0) であり、ETE のビット 32 が 0 であり (24 ビット・モードを示す)、ETE のビット 33-39 がすべて 0 ではない。
2. PROGRAM CALL 操作がスタッキング操作であり (ETE のビット 128 が 1)、ETE のビット 32 および 129 が 0 であり (24 ビット・モードを示す)、ETE のビット 33-39 がすべて 0 ではない。

操作は抑止されます。

命令長コードは 2 です。

PC 変換指定例外は、16 進数 001F のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 009F)。

PER イベント

PER イベントが認識されるのは、CPU が PER に対して使用可能にされているときに、1 つ以上の PER イベントが発生した場合です。

CPU が PER に対して使用可能にされるかどうかは、PER マスク (PSW のビット 1) により制御されます。PER マスクが 0 のときは、PER イベントは認識されません。このビットが 1 のときは、制御レジスタ 9 の PER イベント・マスク・ビットに従って PER イベントが認識されます。

他の条件が原因で操作単位が禁止、無効化、抑止、または中止されない限り、操作単位は完了します。

このイベントを識別する情報が、実記憶位置 150-159 に格納されるほか、場合によっては実記憶位置 161 にも格納されます。

命令長コードは 0、1、2、または 3 です。コード 0 が設定されるのは、同時に指定例外も示されている場合のみです。

PER イベントは、プログラム割り込みコードのビット 8 を 1 に設定することにより示されます。

PER イベントとそれに関連した割り込み情報についての詳細は、4-24ページの『プログラム・イベント記録』を参照してください。

1 次権限例外

1 次権限例外が認識されるのは、スペース切り替えを伴う PROGRAM TRANSFER (PT-ss) による ASN 許可の際に、次のいずれかである場合です。

1. 制御レジスター 4 の中の許可指標が指している権限テーブル・エントリーが、使用された権限テーブルの終わりを越えている。この権限テーブルは、使用された ASN 用の ASN 第 2 テーブル・エントリーで指定されているものです。
2. 許可指標が示している 1 次権限ビットが 0 である。

使用された ASN は実記憶位置 174-175 に格納され、実記憶位置 172-173 は 0 に設定されます。

操作は無効化されます。

命令長コードは 2 です。

1 次権限例外は、16 進数 0024 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A4)。

特権命令例外

特権命令例外が認識されるのは、以下のいずれかに該当する場合です。

1. 問題プログラム状態にあるときに、特権命令を実行しようとした。
2. PROGRAM TRANSFER 命令の R₂ フィールドに指定されている汎用レジスターの最右端ビットの値が 0 であるため、PSW の問題プログラム状態ビットが、問題プログラム状態 (1) から監視プログラム状態 (0) に変更されることになる。
3. 問題プログラム状態において、SET PSW KEY FROM ADDRESS 命令の第 2 オペランドに指定されているキー値について、制御レジスター 3 の中の PSW キー・マスク・ビットが 0 である。

4. 問題プログラム状態において、MOVE WITH KEY 命令の R₃ フィールドに示されているレジスターの最右端バイトが指しているキー値について、制御レジスター 3 の中の PSW キー・マスク・ビットが 0 である。

5. 問題プログラム状態において、MOVE TO PRIMARY、MOVE TO SECONDARY、または MOVE WITH KEY 命令の R₃ フィールドに示されているレジスターの最右端バイトが指しているキー値について、制御レジスター 3 の中の PSW キー・マスク・ビットが 0 である。

6. 問題プログラム状態において、

- EXTRACT PRIMARY ASN
- EXTRACT SECONDARY ASN
- INSERT ADDRESS SPACE CONTROL
- INSERT PSW KEY
- INSERT VIRTUAL STORAGE KEY

上記のいずれかの命令が検出され、抽出権限制御 (制御レジスター 0 のビット 4) が 0 である。

7. 問題プログラム状態において、PROGRAM CALL の実行時に、許可キー・マスク (AKM) と制御レジスター 3 の中の PSW キー・マスクとの論理積演算の結果が 0 になる。

8. 問題プログラム状態において、SET ADDRESS SPACE CONTROL または SET ADDRESS SPACE CONTROL FAST 命令の第 2 オペランド・アドレスのビット 20-23 の値が、2 進数 0011 である。

9. 問題プログラム状態において、MOVE WITH SOURCE KEY または MOVE WITH DESTINATION KEY 命令の汎用レジスター 1 の最右端バイトが指しているキー値について、制御レジスター 3 の中の PSW キー・マスク・ビットが 0 である。

10. 問題プログラム状態において、BRANCH AND SET AUTHORITY 命令の R₁ フィールドに示されているレジスターの最右端バイトが指しているキー値について、制御レジスター 3 の中の PSW キー・マスク・ビットが 0 である。

11. 問題プログラム状態において、RESUME PROGRAM の第 2 オペランド内で、PSW フィールドのビット 16 および 17 の値が 2 進数 11 である。

操作は抑止されます。

命令長コードは 2 または 3 です。

特権命令例外は、16進数 0002 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16進数 0082)。

記憶保護例外

記憶保護例外が認識されるのは、以下のいずれかに該当する場合です。

1. **キー制御保護:** CPU がある記憶位置にアクセスしようとしたが、その記憶位置は行おうとした参照のタイプに対して保護されており、アクセス・キーが記憶キーと一致しない。
2. **アクセス・リスト制御保護:** アクセス・レジスター・モードで、CPU が、取り出し専用ビットが 1 に設定されているアクセス・リスト・エントリーを使用して格納を行おうとした。
3. **低アドレス保護:** 有効アドレスが 0-511 または 4096-4607 の範囲内にあり、低アドレス保護制御 (制御レジスター 0 のビット 35) が 1 であるときに、CPU が低アドレス保護の対象となる格納を行おうとした。
4. **ページ保護:** DAT がオンのときに、CPU が、変換に使用されるページ・テーブル・エントリーまたはセグメント・テーブル・エントリー内でページ保護ビットが 1 に設定されているページへの格納を行おうとした。

命令の位置が取り出しに対して保護されている場合は、操作は抑止されます。同様に、EXECUTE のターゲット命令が取り出しに対して保護されている場合も、操作は抑止されます。

アクセス・リスト制御保護およびページ保護の場合は、操作は抑止されます。その他の 2 つのタイプの保護の場合は、実行が抑止される幾つかの特殊な命令を除き、オペランド位置の参照時に保護例外が検出されると、操作は中止されます。6-17ページの図6-4を参照してください。ただし、3-12ページの『保護時の抑止』で説明されているように、操作は抑止されることもあります。

中止の場合は、変更されるのは結果フィールドだけです。ここで言う「結果フィールド」には、条件コードおよびレジスターのほか、該当の命令により変更されることになっている記憶位置があれば、それも含まれます。ただし、ある記憶位置の参照が記憶保護例外を引き起こす場合は、その記憶位置に対する変更は行われません。したがって、ある命令が記憶域内の特定フィールドの内容のみを変更することになっているときに、そのフィールドのどのバイトも記憶保護例外を引き起こすことになる場合は、操作は抑止されます。取り出しの際に中止が

起きたときは、保護されている情報は、アドレス可能レジスターにロードされることも、他の記憶位置に移されることもありません。

例外の原因となっているアドレスに関する情報が、実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 にも格納されます。3-12ページの『保護時の抑止』を参照してください。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置が保護されている場合は、ILC は 1、2、または 3 のいずれかで、これは参照の原因となった命令の長さを示します。

記憶保護例外は、16進数 0004 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16進数 0084)。

領域第 1 変換例外

領域第 1 変換例外が認識されるのは、仮想アドレスの変換に使用する変換パス内に領域第 1 テーブルがあり、かつ次のいずれかである場合です。

1. 仮想アドレスの領域第 1 指標部分に示されている領域第 1 テーブル・エントリーが、領域第 1 テーブルの外部にある。
2. 領域無効ビットが 1 である。

この例外は、単に領域変換例外と呼ばれることもあります。この用語は、領域第 2 変換例外および領域第 3 変換例外も意味します。

この例外は、命令アドレスまたはオペランド・アドレスの変換に領域第 1 テーブル・エントリーを必要とする命令の実行中に認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

領域第 1 変換例外は、16 進数 0039 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B9)。

領域第 2 変換例外

領域第 2 変換例外が認識されるのは、仮想アドレスの変換に使用する変換パス内に領域第 2 テーブルがあり、かつ次のいずれかである場合です。

1. 仮想アドレスの領域第 2 指標部分に示されている領域第 2 テーブル・エントリーが、領域第 2 テーブルの外部にある。
2. 領域無効ビットが 1 である。

この例外は、単に領域変換例外と呼ばれることもあります。この用語は、領域第 1 変換例外および領域第 3 変換例外も意味します。

この例外は、命令アドレスまたはオペランド・アドレスの変換に領域第 2 テーブル・エントリーを必要とする命令の実行中に認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

領域第 2 変換例外は、16 進数 003A のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00BA)。

領域第 3 変換例外

領域第 3 変換例外が認識されるのは、仮想アドレスの変換に使用する変換パス内に領域第 3 テーブルがあり、かつ次のいずれかである場合です。

1. 仮想アドレスの領域第 3 指標部分に示されている領域第 3 テーブル・エントリーが、領域第 3 テーブルの外部にある。
2. 領域無効ビットが 1 である。

この例外は、単に領域変換例外と呼ばれることもあります。この用語は、領域第 1 変換例外および領域第 2 変換例外も意味します。

この例外は、命令アドレスまたはオペランド・アドレスの変換に領域第 3 テーブル・エントリーを必要とする命令の実行中に認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

領域第 3 変換例外は、16 進数 003B のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00BB)。

2 次権限例外

2 次権限例外が認識されるのは、スペース切り替えを伴う SET SECONDARY ASN で ASN 許可が行われるとき、または、PROGRAM RETURN による許可の際に、復元された SASN が復元された PASN と等しくないときに、次のいずれかである場合です。

1. 制御レジスター 4 の中の許可指標が指している権限テーブル・エントリーが、使用された権限テーブルの終わりを超えている。この権限テーブルは、使用された ASN 用の ASN 第 2 テーブル・エントリーで指定されているものです。PROGRAM RETURN の場合は、この ASN は、使用されたリンケージ・スタック状態エントリーから復元される SASN です。
2. 許可指標が示している 2 次権限ビットが 0 である。

使用された ASN は実記憶位置 174-175 に格納され、実記憶位置 172-173 は 0 に設定されます。

操作は無効化されます。

命令長コードは 1 または 2 です。

2 次権限例外は、16 進数 0025 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00A5)。

セグメント変換例外

セグメント変換例外が認識されるのは、次のいずれかの場合です。

1. 仮想アドレスのセグメント指標部分に示されているセグメント・テーブル・エントリーが、セグメント・テーブルの外部にある。
2. セグメント無効ビットが 1 である。

この例外は、命令アドレスまたはオペランド・アドレスの変換にセグメント・テーブル・エントリーを必要とする命令の実行中に認識されます。ただし、LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの場合は、条件コードの設定によりこの条件が示されます。

割り込みが起きたときは、例外の原因となった仮想アドレスに関する情報が実記憶位置 168-175 に格納されるほか、場合によっては実記憶位置 160 および 162 にも格納されます。この情報の詳細については、3-49 ページの『割り当て済みの記憶位置』を参照してください。

操作単位は無効化されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

セグメント変換例外は、16 進数 0010 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0090)。

スペース切り替えイベント

スペース切り替えイベントが認識されるのは、以下の場合に操作が完了したときです。

1. スペース切り替え形式の PROGRAM CALL、PROGRAM RETURN、または PROGRAM TRANSFER が実行されたときに、次のいずれかが真である場合。
 - a. 操作の前に、1 次スペース切り替えイベント制御ビット (制御レジスター 1 のビット 57) が 1 である。
 - b. 操作の後で、1 次スペース切り替えイベント制御ビットが 1 である。
 - c. PER イベントが示されている。
2. RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST が実行され、操作前または操作後のいずれかに (両方ではなく) CPU がホーム・スペース・モードになっており、かつ次のいずれかが真である場合。
 - a. 1 次スペース切り替えイベント制御ビット (制御レジスター 1 のビット 57) が 1 である。
 - b. ホーム・スペース切り替えイベント制御ビット (制御レジスター 13 のビット 57) が 1 である。
 - c. PER イベントが示されている。

PROGRAM CALL、PROGRAM RETURN、および PROGRAM TRANSFER 命令、そして、変換モードをホーム・スペース・モードに変更する RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令の場合は、操作前の制御レジスター 4 のビット位置 48-63 にあった旧 PASN が、実記憶位置 174-175 に格

納され、実記憶位置 172-173 のビット位置 0 には旧 1 次スペース切り替えイベント制御ビットが入れられ、ビット位置 1-15 には 0 が入れられます。

変換モードをホーム・スペース・モードから別のモードに変更する RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令の場合は、実記憶位置 174-175 には 0 が入れられ、実記憶位置 172-173 のビット位置 0 にはホーム・スペース切り替えイベント制御ビット、そしてビット位置 1-15 には 0 が入れられます。

PSW フォーマット・エラーを引き起こす PROGRAM RETURN 命令の場合は、命令長コードが、0 または 1 (EXECUTE が実行された場合は 0 または 2) のどちらになるかは予測不能です。

操作は完了します。

命令長コードは 0、1、または 2 です。

スペース切り替えイベントは、16 進数 001C のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 009C)。

プログラミング上の注意:

1. 制御プログラムは、スペース切り替えイベントを使用して、プログラムが特定のアドレス・スペースに入るか、またはそこから出るたびに、制御を受け取ることができます。制御レジスタ 1 がロードされるたびに、1 次スペース切り替えイベント制御ビットが、1 次アドレス・スペース制御エレメントの残りのビットと共に、制御レジスタ 1 にロードされます。
2. スペース切り替えイベントは、プログラミングされた許可検査を行うとき、追加のトレース情報が記録されるようにしたいとき、または、PER またはトレースに対して CPU を使用可能または使用不可にしたいときに便利です。
3. PC-ss、PR-ss、および PT-ss では、操作の一環として、ASN 第 2 テーブル・エントリ (ASTE) のビット 121 が、レジスタ 1 のビット位置 57 にロードされます。特定のアドレス・スペースについて ASTE のビット 121 が 1 に設定されている場合は、プログラムが、PC-ss、PR-ss、または PT-ss のいずれかを使用してそのアドレス・スペースに入るかまたはそこから出ると、スペース切り替えイベントが認識されます。

4. PC-ss、PR-ss、または PT-ss 操作が完了し何らかの PER イベントが示されるとき、または、ホーム・スペース・モードとの間の変更を行う RESUME PROGRAM、SET ADDRESS SPACE CONTROL、または SET ADDRESS SPACE CONTROL FAST 命令の実行が完了し何らかの PER イベントが示されるときに、スペース切り替えイベントが起きることにより、制御プログラムは、その PER イベントの原因となった命令がどのアドレス・スペースから取り出されたかを判別することができます。

特殊操作例外

特殊操作例外が起きるのは、以下のいずれかに該当する場合です。

1. 監視プログラム状態で SET SYSTEM MASK を実行しようとしたときに、SSM 抑止制御ビット (制御レジスタ 0 のビット 33) が 1 である。
2. DAT がオフのときに、以下の命令のいずれかを実行しようとした。
 - EXTRACT PRIMARY ASN
 - EXTRACT SECONDARY ASN
 - INSERT ADDRESS SPACE CONTROL
 - INSERT VIRTUAL STORAGE KEY
 - SET ADDRESS SPACE CONTROL
 - SET SECONDARY ASN
3. MOVE TO PRIMARY または MOVE TO SECONDARY を実行しようとしたときに、CPU が 1 次スペース・モードまたは 2 次スペース・モードになっていない。
4. 基本 PROGRAM CALL または PROGRAM TRANSFER を実行しようとしたときに、CPU が 1 次スペース・モードになっていない。
5. BRANCH AND STACK、スタッキング PROGRAM CALL、PROGRAM RETURN、または TRAP を実行しようとしたときに、CPU が 1 次スペース・モードまたはアクセス・レジスタ・モードになっていない。
6. EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、または MODIFY STACKED STATE を実行しようとしたときに、CPU が 1 次スペース・モード、アクセス・レジスタ・モード、またはホーム・スペース・モードになっていない。
7. LOAD ADDRESS SPACE PARAMETERS、スペース切り替えを伴う PROGRAM CALL (PC-ss)、

スペース切り替えを伴う PROGRAM TRANSFER (PT-ss)、または SET SECONDARY ASN (SSAR-cp または SSAR-ss) を実行しようとしたとき、または、PASN 変換または SASN 変換を必要とする PROGRAM RETURN 命令を実行しようとしたときに、ASN 変換制御 (制御レジスター 14 のビット 34) が 0 である。

8. PROGRAM CALL または PROGRAM TRANSFER を実行しようとしたときに、サブシステム・リンケージ制御 (1 次 ASN 第 2 テーブル・エントリーのビット 192) が 0 である。
9. SET ADDRESS SPACE CONTROL、MOVE TO PRIMARY、または MOVE TO SECONDARY を実行しようとしたときに、2 次スペース制御 (制御レジスター 0 のビット 37) が 0 である。これが原因で、SET ADDRESS SPACE CONTROL FAST を実行しようとしたときに例外が認識されることがあります。
10. BRANCH IN SUBSPACE GROUP を実行しようとしたときに、次のいずれかが真である。
 - a. 現 1 次アドレス・スペースが、現ディスパッチ可能単位に関連したサブスペース・グループに含まれていない。つまり、制御レジスター 5 の中の 1 次 ASTE 起点 (PASTE0) が、ディスパッチ可能単位制御テーブル (DUCT) の中の基本 ASTE 起点 (BASTE0) に等しくない。
 - b. アクセス・レジスター R₂ の中のアクセス・リスト・エントリー・トークン (ALET) が ALET 1 であるのに対し、ディスパッチ可能単位は、まだ、BRANCH IN SUBSPACE GROUP を使用してどのサブスペースにも入っていない。つまり、DUCT 内のサブスペース ASTE 起点 (SSASTE0) がすべて 0 である。
 - c. 使用されている ALET が ALET 0 でも ALET 1 でもなく、宛先 ASTE (DASTE) がサブスペース・グループの基本スペースまたはサブスペースを指定していない。つまり、アクセス・リスト・エントリーから取得した DASTE 起点 (DASTE0) が DUCT 内の BASTE0 と等しくなく、かつ、DASTE 内のアドレス・スペース制御エレメントの中のサブスペース・グループ・ビット (G) が 0 であるか、DASTE 内の基本スペース・ビット (B) が 1 である。
11. BRANCH AND SET AUTHORITY を実行しようとしたときに、R₂ フィールドが、基本権限状態の

場合に 0 であるか、縮小権限状態の場合に 0 以外である。

12. TRAP を実行しようとしたときに、TRAP 使用可能ビット (ディスパッチ可能単位制御テーブルのバイト 44-47 のビット 31) が 0 である。
13. 基本 PROGRAM CALL を実行しようとしたときに、拡張アドレッシング・モード・ビット (現 PSW のビット 31) が、エントリー拡張アドレッシング・モード・ビット (エントリー・テーブル・エントリーのビット 129) に等しくない。
14. 24 ビットまたは 31 ビット・アドレッシング・モードで LOAD REAL ADDRESS (LRA) を実行しようとしたときに、結果の実アドレスまたは絶対アドレスのビット 0-32 がすべて 0 ではない。

操作は抑止されます。

命令長コードは 1、2、または 3 です。

特殊操作例外は、16 進数 0013 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0093)。

指定例外

指定例外が認識されるのは、以下のいずれかが真である場合です。

1. PSW の未割り当てのビット位置 (つまりビット位置 0、2-4、24-30、または 33-63 のいずれか) に 1 が入れられている。これは、早期 PSW 指定例外として扱われます。
2. PSW のビット位置 12 に 1 が入れられている。これは、早期 PSW 指定例外として扱われます。
3. 次のいずれかの理由により PSW が無効である。
 - a. PSW のビット 31 が 1 で、ビット 32 が 0 である。
 - b. PSW のビット 31 および 32 が 0 (24 ビット・アドレッシング・モードを示す) で、PSW のビット 64-103 がすべて 0 ではない。
 - c. PSW のビット 31 が 0 でビット 32 が 1 (31 ビット・アドレッシング・モードを示す) であり、PSW のビット 64-96 がすべて 0 ではない。

これは、早期 PSW 指定例外として扱われます。

4. PSW に奇数の命令アドレスが含まれている。

5. 規定境界指定を必要とする命令で、オペランド・アドレスが規定境界を指していない。
 6. 偶数番号のレジスタ指定を必要とする命令で、R フィールドに奇数番号の汎用レジスタが指定されている。
 7. 拡張オペランドに、0、1、4、5、8、9、12、または 13 以外の浮動小数点レジスタが指定されている。
 8. 10 進数演算の乗数または除数が、15 桁の数字と符号を合わせた長さを超えている。
 9. 10 進数の乗算または除算において、第 1 オペランド・フィールドの長さが、第 2 オペランド・フィールドの長さと同じかまたはそれより小さい。
 10. MONITOR CALL のビット位置 8-11 が 0 でない。
 11. SET ADDRESS SPACE CONTROL または SET ADDRESS SPACE CONTROL FAST の第 2 オペランド・アドレスのビット 52 および 53 のいずれかまたは両方とも 0 でない。
 12. PSW の拡張アドレッシング・モード・ビットが 0 のときに、PROGRAM TRANSFER の R₂ フィールドに指定されている汎用レジスタの基本アドレッシング・モード・ビット (ビット 32) が 0 であるのに対し、同じレジスタ内の命令アドレスのビット 33-39 がすべて 0 ではない。
 13. COMPARE AND FORM CODEWORD を実行しようとしたときに、汎用レジスタ 1、2、および 3 の初期値が偶数値でない。
 14. UPDATE TREE を実行しようとしたときに、汎用レジスタ 4 および 5 の初期値が、24 ビットまたは 31 ビット・アドレッシング・モードの場合に 8 の倍数になっていないか、64 ビット・アドレッシング・モードの場合に 16 の倍数になっていない。
 15. MOVE PAGE を実行しようとしたときに、汎用レジスタ 0 のビット位置 48-51 に 0 が含まれていないか、ビット 52 および 53 が両方とも 1 である。
 16. COMPARE LOGICAL STRING、MOVE STRING、または SEARCH STRING を実行しようとしたときに、汎用レジスタ 0 のビット 32-55 がすべて 0 ではない。
 17. EXECUTE を実行しようとしたときに、ターゲット・アドレスが奇数である。
 18. RESUME PROGRAM を実行しようとしたときに、第 2 オペランド内の PSW フィールドのビット 31、32、および 64-127 が、現 PSW に入れる値として有効でない。この例外が認識されるのは、以下のいずれかの場合です。
 - ビット 31 および 32 が両方とも 0 で、ビット 64-103 がすべて 0 ではない。
 - ビット 31 が 0、ビット 32 が 1 で、ビット 64-96 がすべて 0 ではない。
 - ビット 31 が 1 で、ビット 32 が 0 である。
 - ビット 127 が 1 である。
 19. オプションとして LOAD PSW を実行しようとしたときに、第 2 オペランド・アドレスの位置にあるダブルワードのビット 12 が 0 である。
 20. SET ADDRESSING MODE (SAM24) を実行しようとしたときに、PSW 内の更新されない命令アドレスのビット 0-39 (PSW のビット 64-103) がすべて 0 ではない。
 21. SET ADDRESSING MODE (SAM31) を実行しようとしたときに、PSW 内の更新されない命令アドレスのビット 0-32 (PSW のビット 64-96) がすべて 0 ではない。
- 旧 PSW が示している命令の実行は抑止されます。ただし、早期 PSW 指定例外 (上記の原因 1 から 3 まで) の場合は、新 PSW を導入する操作は完了しますが、その後ただちに割り込みが発生します。
- 以下に示す場合以外は、命令長コード (ILC) は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。
- 命令アドレスが奇数 (原因 4 (6-31 ページ)) の場合は、ILC が 1、2、3 のいずれになるかは予測不能です。
- 早期 PSW 指定例外 (上記の原因 1 から 3 まで) のために例外が認識され、LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、または割り込みによりその例外が導入された場合は、ILC は 0 です。例外が SET ADDRESSING MODE (SAM24、SAM31) により導入された場合は、ILC は 1 で、ADDRESSING MODE が EXECUTE のターゲットだった場合は、ILC は 2 です。例外が、SET SYSTEM MASK または STORE THEN OR SYSTEM MASK により導入された場合は、ILC は 2 です。

指定例外は、16 進数 0006 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0086)。

プログラミング上の注意: PSW に関連した例外がどのようなときに認識されるかの定義については、6-9ページの『PSW に関連した例外』を参照してください。

スタック空例外

スタック空例外が認識されるのは、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN でのアンスタッキング・プロセスにおいて、現リンケージ・スタック・エントリーがヘッダー・エントリーで、そのヘッダー・エントリー内の逆方向スタック・エントリー妥当性ビットが 0 である場合です。

操作は無効化されます。

命令長コードは 1 または 2 です。

スタック空例外は、16 進数 0031 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B1)。

スタック一杯例外

スタック一杯例外が認識されるのは、BRANCH AND STACK またはスタッキング PROGRAM CALL でのスタッキング・プロセスにおいて、現リンケージ・スタック・セクション内の十分な残余フリー・スペースがなく、トレーラー・エントリー内の正方向セクション妥当性ビットが 0 である場合です。

操作は無効化されます。

命令長コードは 2 です。

スタック一杯例外は、16 進数 0030 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B0)。

スタック操作例外

スタック操作例外が認識されるのは、PROGRAM RETURN でのアンスタッキング・プロセスにおいて、検出されたいずれかのリンケージ・スタック状態エントリーまたはヘッダー・エントリーのアンスタック抑止ビットが 1 である場合です。

操作は無効化されます。

命令長コードは 1 または 2 です。

スタック操作例外は、16 進数 0034 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B4)。

スタック指定例外

スタック指定例外が認識されるのは、以下の場合です。

1. BRANCH AND STACK またはスタッキング PROGRAM CALL でのスタッキング・プロセスにおいて、現リンケージ・スタック・セクション内に十分な残余フリー・スペースがなく、かつ次のいずれかが真である場合。
 - a. 現セクションのトレーラー・エントリーを見つげるために使用されている残余フリー・スペース値が、8 の倍数でない。
 - b. 次のセクション内に十分な残余フリー・スペースがない。

2. EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN でのアンスタッキング・プロセスにおいて、現リンケージ・スタック・エントリーがヘッダー・エントリーであり、そのヘッダー・エントリー内の逆方向スタック・エントリー・アドレスが別のヘッダー・エントリーを指し示している。

操作は無効化されます。

命令長コードは 1 または 2 です。

スタック指定例外は、16 進数 0032 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B2)。

スタック・タイプ例外

スタック・タイプ例外が認識されるのは、EXTRACT STACKED REGISTERS、EXTRACT STACKED STATE、MODIFY STACKED STATE、または PROGRAM RETURN でのアンスタッキング・プロセスにおいて、以下のいずれかである場合です。

1. 現リンケージ・スタック・エントリーが、ヘッダー・エントリーまたは状態エントリーでない。
2. 現リンケージ・スタック・エントリーがヘッダー・エントリーである場合に、そのヘッダー・エントリー内の逆方向スタック・エントリー・アドレスが指し示している先行エントリーが、ヘッダー・エントリーまたは状態エントリーでない。(先行エントリ

ーがヘッダー・エンタリーである場合は、スタック指定例外が認識されます。)

操作は無効化されます。

命令長コードは 1 または 2 です。

スタック・タイプ例外は、16 進数 0033 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 00B3)。

トレース・テーブル例外

トレース・テーブル例外が認識されるのは、CPU が、次の 4K バイト・ブロック境界に達するかまたはそれを超えるトレース・テーブル・エンタリーを格納しようとしたときです。TRACE 命令でこの例外を認識できるようにするために、明示トレース・エンタリーの長さは、TRACE (TRACE) の場合は 76 バイト、TRACE (TRACG) の場合は 144 バイトとして扱われます。PROGRAM CALL トレース・エンタリーとモード切り替えトレース・エンタリーの両方を格納する PROGRAM CALL 命令の場合は、第 1 または第 2 のエンタリーのいずれかが境界に達するかまたはそれを超えることになるときは、第 1 エンタリーについて例外が認識されます。

操作は無効化されます。

命令長コードは 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

トレース・テーブル例外は、16 進数 0016 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0096)。

変換指定例外

変換指定例外が認識されるのは、仮想アドレスを変換する際に、以下のいずれかが真である場合です。

1. 変換に使用するアドレス・スペース制御エレメントが示しているテーブル内での検索において、選択したテーブル・エンタリー内のテーブル・タイプ・ビットが、アドレス・スペース制御エレメント内の指定タイプ・ビットに等しくない。
2. 領域第 1 テーブル、領域第 2 テーブル、または領域第 3 テーブル内のエンタリーが示すテーブルの中での検索において、選択したテーブル・エンタリー内のテーブル・タイプ・ビットの値が、それを指定

しているテーブル・エンタリー内の同じビットの値から 1 を引いた値に等しくない。

3. 専用スペース制御 (変換に使用されるアドレス・スペース制御エレメントのビット 55) が 1 で、変換に使用されるセグメント・テーブル・エンタリーが有効で、セグメント・テーブル・エンタリー内の共通セグメント・ビット (ビット 59) が 1 である。
4. 変換に使用されるページ・テーブル・エンタリーが有効で、そのエンタリーのビット位置 52 および 55 に 0 が含まれていない。

上記の理由は、どれも、DAT テーブル・エンタリーにフォーマット・エラーがある場合と考えることができません。

この例外が認識されるのは、アドレス変換を使用する命令の実行中のみです。つまり、DAT がオンで、論理アドレス、命令アドレス、または仮想アドレスを変換する必要があるとき、あるいは、LOAD REAL ADDRESS または STORE REAL ADDRESS が実行されるときです。

操作単位は抑止されます。

命令の取り出し時にこの例外が起きたときは、命令長コード (ILC) が 1、2、3 のいずれになるかは予測不能です。EXECUTE のターゲット命令の取り出し時にこの例外が起きたときは、ILC は 2 です。

オペランド位置の参照時にこの例外が起きたときは、ILC は 1、2、または 3 のいずれかで、例外を引き起こした命令の長さを示します。

変換指定例外は、16 進数 0012 のプログラム割り込みコードにより示されます (ただし、同時に PER イベントも示されている場合は 16 進数 0092)。

プログラミング上の注意: 命令アドレスの変換プロセスで変換指定例外が認識されたときは、操作は抑止されます。この場合、旧 PSW 内の命令アドレスは、命令長コード (ILC) に示されている値の分だけ増加しているもので、命令のアドレスを導き出すには ILC が必要です。領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、セグメント変換例外、およびページ変換例外の場合は、操作は無効化され、旧 PSW 内の命令アドレスに該当の命令が示され、ILC は恣意的に 1、2、3 のいずれかに設定されます。

プログラム割り込みの集合名

便宜上、ある種のプログラム割り込みはグループにまとめられ、集合名で呼ばれています。これらの集合名は、割り込みの定義の中などで、例外のセット全体を参照する必要がありときに使用されます。集合名には以下の4つがあります。

- アクセス例外
- ASN 変換例外

- サブスペース置き換え例外
- トレース例外

個々の例外とそれぞれの優先順位は、6-37ページの『複数のプログラム割り込み条件』に示されています。

アクセス例外の認識

図6-5 は、アクセス例外を引き起こす可能性のある条件と、それぞれが検出されたときに実行されるアクションを要約したものです。

条件 ²	LRA または LRA の仮想アドレスの変換		TAR および TPROT の場合の変換と、TPROT の論理アドレスの場合のアクセス ¹		その他のアドレスの場合の変換とアクセス	
	表示	アクション	表示	アクション	表示	アクション
アクセス・レジスター ³ ビット 0-6 がすべて 0 ではない	cc3	完了	cc3	完了	AS	抑止
有効アクセス・リスト指定 ³ 指定のアドレスが無効	A	抑止	A	抑止	A	抑止
アクセス・リスト・エンタリー ³ アクセス・リスト長違反 エントリーのアドレスが無効	cc3 A	完了 抑止	cc3 A	完了 抑止	AT A	無効化 抑止
I ビットがオン	cc3	完了	cc3	完了	AT	無効化
アクセス・レジスター内のシーケンス番号がエンタリー内のシーケンス番号に等しくない	cc3	完了	cc3	完了	ALQ	無効化
ASN 第 2 テーブル・エンタリー ³ エンタリーのアドレスが無効	A	抑止	A	抑止	A	抑止
I ビットがオン	cc3	完了	cc3	完了	AV	無効化
アクセス・リスト・エンタリー内のシーケンス番号がエンタリー内のシーケンス番号に等しくない	cc3	完了	cc3	完了	ASQ	無効化
権限テーブル・エンタリー ³ 権限テーブル長違反	cc3	完了	cc3	完了	EA	無効化
エンタリーのアドレスが無効	A	抑止	A	抑止	A	抑止
2 次権限ビットが 1 でない	cc3	完了	cc3	完了	EA	無効化
アドレス・スペース制御エレメントが、領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定であるときに、それぞれ、命令アドレスまたはオペランド・アドレスのビット 0-10、0-21、または 0-32 がすべて 0 ではない	cc3	完了	cc3	完了	ATY	無効化
アドレス・スペース制御エレメントまたは領域テーブル・エンタリーで指定されている領域テーブル・エンタリー	cc3	完了	cc3	完了	RT	無効化
テーブルの外部のエンタリー エンタリーのアドレスが無効	A	抑止	A	抑止	A	抑止
I ビットがオン	cc3	完了	cc3	完了	RT	無効化
エンタリー内の TT が、アドレス・スペース制御エレメント内の DT に等しくないか、または次に高いレベルのエンタリー内の TT から 1 を引いた値に等しくない。	TS	抑止	TS	抑止	TS	抑止

図 6-5 (1/3). アクセス例外の取り扱い

条件 ²	LRA または LRAg の仮想アドレスの変換		TAR および TPROT の場合の変換と、TPROT の場合の論理アドレスのアクセス ¹		その他のアドレスの場合の変換とアクセス	
	表示	アクション	表示	アクション	表示	アクション
アドレス・スペース制御エレメントまたは領域テーブル・エントリーで指定されているセグメント・テーブル・エントリー	cc3	完了	cc3	完了	ST	無効化
テーブルの外部のエントリー	A	抑止	A	抑止	A	抑止
エントリーのアドレスが無効	cc1	完了	cc3	完了	ST	無効化
I ビットがオン (以下の場合を除く)	cc3	完了	-	-	-	-
I ビットがオン (エントリー・アドレスの 0-32 がすべて 0 ではないときは 24 ビットまたは 31 ビット・モードでの LRA)						
0 かどうか検査されるビット位置が 1 ⁵	TS	抑止	TS	抑止	TS	抑止
エントリー内の TT が、アドレス・スペース制御エレメント内の DT に等しくないか、次に高いレベルのエントリー内の TT から 1 を引いた値に等しくない (TT は 0 でない)	TS	抑止	TS	抑止	TS	抑止
ページ・テーブル・エントリー						
エントリーのアドレスが無効	A	抑止	A	抑止	A	抑止
I ビットがオン (以下の場合を除く)	cc2	完了	cc3	完了	PT	無効化
I ビットがオン (エントリー・アドレスの 0-32 がすべて 0 ではないときは 24 ビットまたは 31 ビット・モードでの LRA)	cc3	完了	cc3	完了	PT	無効化
0 かどうか検査されるビット位置の値が 1 ⁵	TS	抑止	TS	抑止	TS	抑止
命令取り出しのためのアクセス						
記憶位置が保護されている (キー制御保護)	-	-	-	-	P	抑止
無効なアドレス	-	-	-	-	A	抑止
オペランド取得のためのアクセス						
記憶位置が保護されている (低アドレス保護、ページ保護、またはキー制御保護)	-	-	cc set ⁶	完了	P	中止*
無効なアドレス	-	-	A	抑止	A	中止*

説明:

- 条件は適用されません。
- * 本書で他の指定がない限り、アクションは中止です。
- 1 TAR は論理アドレスを持っていません。「アドレス・スペース制御エレメント」から「オペランド取得のためのアクセス」までの行は、TPROT のみに適用されるもので、TAR には適用されません。
- 2 保護は、命令取り出しおよびオペランド取得のためのアクセスのみに適用されます。有効アクセス・リスト指定またはリストされたエントリーの取り出しには、適用されません。
- 3 アクセス・レジスター、有効アクセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、または権限テーブル・エントリーに関連した例外は、アクセス・レジスター・モードのときのみ認識されます。ただし、LOAD REAL ADDRESS および STORE REAL ADDRESS において、PSW ビット 16 および 17 が 2 進数 01 のときは、これらの例外が認識されます。また、TEST ACCESS では、変換モードに関係なくこれらの例外が認識されます。
- 4 アクセス・リスト・エントリー内の専用ビットが 0 の場合、または、アクセス・リスト・エントリー内のアクセス・リスト・エントリー権限指標が、制御レジスター 8 の中の拡張権限指標に等しい場合は、権限テーブルはアクセスされず、2 次権限ビットは検査されません。
- 5 テーブル・エントリー内にフォーマット・エラーがある場合の変換指定例外が認識されるのは、命令の実行のためにアドレスを変換するときにそのエントリーが必要とされる場合のみです。

図 6-5 (2/3). アクセス例外の取り扱い

説明 (続き):

- 6 条件コードは以下のように設定されます。
 - 0 オペランドの記憶位置は保護されていない。
 - 1 取り出しは許可されるが、格納は許可されない。
 - 2 取り出しも格納も許可されない。
- A アドレッシング例外。
- ALQ ALE シーケンス例外。
- AS ALET 指定例外。
- ASQ ASTE シーケンス例外。
- AT ALEN 変換例外。
- ATY ASCE タイプ例外。
- AV ASTE 妥当性例外。
- cc1 条件コード 1 が設定される。
- cc2 条件コード 2 が設定される。
- cc3 条件コード 3 が設定される。
- EA 拡張権限例外。
- P 記憶保護例外。
- PT ページ変換例外。
- RT テーブルのレベルに応じて、領域第 1 変換例外、領域第 2 変換例外、または領域第 3 変換例外。
- ST セグメント変換例外。
- TS 変換指定例外。

図 6-5 (3/3). アクセス例外の取り扱い

アクセス例外は、すべて、その例外が関連している命令の実行の一部として認識されます。CPU が、使用可能でない記憶位置から事前取り出しをしようとした場合、または、他の何らかのアクセス例外条件を検出したが、ブランチ命令または割り込みにより命令順序が変更された結果、該当の命令が実行されない場合は、アクセス例外は認識されません。

どの命令でも、命令取り出しが原因でアクセス例外が認識されることがあります。さらに、記憶機構内のオペランドへのアクセスが原因で、命令実行に関連したアクセス例外が起きることがあります。

命令の取り出しを原因とするアクセス例外が示されるのは、例外を起こさずに命令の第 1 ハーフワードを取り出すことができないときです。命令の第 1 ハーフワードについてアクセス例外がないときは、命令の最初の 2 ビットに指定されている命令長に従って、残りのハーフワードについてアクセス例外が示されることがあります。ただし、命令の第 2 または第 3 のハーフワードにアクセスせずに操作を実行できる場合は、その未使用部分についてアクセス例外が示されるかどうかは、予測不能です。命令取り出しに関するアクセス例外の指示はすべての命令について同じなので、個々の命令の定義の中では説明していません。

個々の命令の説明の中で特に明記されていない限り、オペランド位置へのアクセスに関連した例外には、以下の規則が適用されます。取り出しタイプ・オペランドの場合は、アクセス例外が必ず示されるのは、操作を完了するために必要なオペランド部分についてのみです。取り出しタイプ・オペランドの中の、操作の完了に必要な部分については、アクセス例外が示されるかどうかは予測不能です。格納タイプ・オペランドの場合は、オペランドのアクセス不能部分を使用せずに操作を完了できる場合であっても、オペランド全体についてアクセス例外

が認識されます。格納タイプ・オペランドの値が予測不能として定義されるような状況では、アクセス例外が認識されるかどうかは予測不能です。

オペランド位置へのアクセスが原因でアクセス例外が認識される可能性がある場合は、該当の命令の説明の中のプログラム例外のリストに、必ず「アクセス」という語が含まれています。また、このエントリーには、どのオペランドが例外認識の原因となる可能性があるか、および、そのオペランド位置への取り出しアクセスまたは格納アクセスのどちらで例外が認識されるかも示されています。アクセス例外は、オペランドの中の、個々の命令について定義されている部分についてのみ認識されません。

複数のプログラム割り込み条件

PER イベントを除き、1 つのプログラム割り込みと共に示されるプログラム割り込み条件は 1 つだけです。しかし、1 つの条件が存在するという事は、他の条件が存在しないことを意味しているわけではありません。複数の割り込み条件が存在しているときは、割り込みコードには最も優先順位の高い条件のみが示されます。

同じ優先順位の条件が 2 つある場合、そのどちらが示されるかは予測不能です。特に、ページまたは保護の境界にまたがっているオペランドの 2 つの部分に関連したアクセス例外の優先順位は予測不能であり、必ずしも、オペランド内のバイトのアクセスについて指定されている順序に対応しているとは限りません。

終了のタイプ (無効化、抑止、または中止) は、割り込みコードに示されている例外のタイプについて定義されている終了タイプになります。ただし、中止を許す条件が示されているときに、無効化または抑止を引き起こす他

の条件も同時に存在する場合は、操作単位は抑止されま
す。

6-39ページの図6-6 は、PER イベントと一部の複雑な制
御命令に関連した割り込みを除き、すべてのプログラム
割り込み条件の優先順位を示しています。命令の特定の
ハーフワードまたは特定のオペランド・バイトの記憶域
に対する参照に関連したすべての例外は、「アクセス」
という項にまとめてあります。6-41ページの図6-7 は、
1 つのアクセスにおけるアクセス例外の優先順位を示し
ています。したがって、2 番目の図は、命令の特定部分
へのアクセスまたはオペランドに関連した特定アクセス
の中で検出される複数の例外のうち、どれが最も優先順
位が高いかを示しています。また、最初の図は、その条
件の優先順位と、操作中に検出される他の条件との関係
を示しています。同様に、6-43ページの図6-8 および
6-44ページの図6-10 には、それぞれ、ASN 変換および
トレースの一環として発生する例外の優先順位を示して
あります。

一部の命令については、個々の命令に関する説明の中に
優先順位が示されています。

リストされている任意の 2 つの条件の間の相対的な優先
順位は、図の中の優先順位番号を、左から右へ、一致し
なくなるところまで比較すれば分かります。最初の不一
致箇所が数字の部分にある場合は、2 つの条件は互いに
排他的な関係にあるか、または、両方の条件が起り得
る場合は番号の小さい方の条件が示されます。最初の不
一致が英字の部分にある場合は、2 つの条件は排他的で

はなく、両方が起きたときにどちらが示されるかは予測
不能です。

表の使い方を分かりやすくするために、6 バイト命令で
ある ADD DECIMAL 命令を例として考えてみましょう。
この命令の最初の 4 バイトはアクセスできますが、
この命令が記憶域境界にまたがっているため、最後の 2
バイトについてアドレッシング例外が存在するものとし
ます。さらに、この命令によりアドレッシングされる最
初のオペランドに無効な 10 進数字が含まれており、ま
た、そのオペランドは、キー制御保護の対象となってい
るため取り出しはできるが格納はできない記憶位置に
あるものとします。この状況で ADD DECIMAL を実行
しようとする、結果として次の 3 つの例外が起きる可
能性があります。

優先順位 番号	例外
7.B	命令の第 3 ハーフワードに関する アクセス例外。 アクセス例外 (オペランド 1)。 データ例外。
8.B	
8.D	

最初の不一致 (7≠8) は数字部分にあるので、アドレッシ
ング例外が示されます。しかし、ADD DECIMAL 命令
全体を取り出すことができ、上記の 2 番目以降の 2 つ
の例外のみが存在する場合は、不一致 (B≠D) は英字部
分にあるので、記憶保護例外とデータ例外のどちらが示
されるかは予測不能です。

1. 即時割り込みを起こすタイプの PSW エラーに起因する指定例外。¹
2. PSW 内に奇数の命令アドレスがあることに起因する指定例外。
3. EXECUTE の第 1 ハーフワードに関するアクセス例外。²
4. EXECUTE の第 2 ハーフワードに関するアクセス例外。²
5. EXECUTE のターゲット命令がハーフワード境界上に指定されていないことに起因する指定例外。²
6. 命令の第 1 ハーフワードに関するアクセス例外。
- 7.A 命令の第 2 ハーフワードに関するアクセス例外。³
- 7.B 命令の第 3 ハーフワードに関するアクセス例外。³
- 7.C.1 操作例外。
- 7.C.2 特権命令に関する特権命令例外。
- 7.C.3 実行例外。
- 7.C.4 特殊操作例外。⁴
- 8.A 上記の 1、2、および 5 に含まれるもの以外の条件に起因する指定例外。
- 8.B⁵ 記憶機構内のオペランドへのアクセスに関するアクセス例外。⁶
- 8.C⁵ 記憶域内のオペランドに対するその他のアクセスに関するアクセス例外。⁶
- 8.D データ例外。⁷
- 8.E 10 進除算例外。⁸
- 8.F トレース例外。
9. PER イベント以外のイベント、結果が完了となる例外、固定小数点除算例外、浮動小数点除算例外、オペランド例外、平方根例外、および非正規化オペランド例外。これらの例外およびイベントは相互に排他的であるか、または、それぞれの優先順位は該当の定義の中で指定されています。

説明:

数字は優先順位を示しており、「1」が最高優先順位です。英字は優先順位を示すものではありません。

- 1 即時割り込みを起こす PSW エラーは、割り込みの結果としてロードされる新しい PSW により導入されることも、PROGRAM RETURN、SET SYSTEM MASK および STORE THEN OR SYSTEM MASK により導入されることもあります。表に示されている優先順位は、割り込みにより導入された PSW エラーに関する優先順位ですが、これは、前の命令により導入された PSW エラーに関する優先順位と考えることもできます。このエラーが導入されるのは、命令で他の例外が検出されなかった場合のみです。結果として生じる割り込みの優先順位は、次に実行されることになっていた命令が起こす割り込みより高くなります。ただし、エラーのある PSW を導入した命令が起こす割り込みの優先順位よりは低くなります。
- 2 優先順位 3、4、および 5 は EXECUTE 命令に関するものであり、6 以降の優先順位はターゲット命令に関するものです。EXECUTE がいない場合は、優先順位 3、4、および 5 は適用されません。
- 3 命令の各ハーフワードごとに、別々にアクセスが行われることがあります。命令の第 2 ハーフワードは、命令のビット 0-1 が両方とも 0 ではない場合のみ、アクセスされます。命令の第 3 ハーフワードは、命令のビット 0-1 が両方とも 1 である場合のみ、アクセスされます。これらのハーフワードのいずれかを使用せずに命令を完了できる場合、またはそのハーフワードを使用せずに優先順位の低い例外を判別できる場合は、そのハーフワードについては、必ずアクセス例外が認識されるとは限りません。

図 6-6 (1/2). プログラム割り込み条件の優先順位

説明 (続き):

- 4 LOAD REAL ADDRESS により認識される特殊操作例外の優先順位は 9 です。
- 5 命令取り出しの場合と同様に、オペランドの各部分について別々にアクセスが行われることがあります。これらのアクセスも、また異なるオペランドに対するアクセスも、優先順位は同じであり、8.B と 8.C の 2 つの項目をリストしてあるのは、これらのアクセスの任意の 2 つに関する例外の相対優先順位を表すためです。8.B には、INSERT STORAGE KEY EXTENDED、INSERT VIRTUAL STORAGE KEY、INVALIDATE PAGE TABLE ENTRY、LOAD REAL ADDRESS、STORE REAL ADDRESS、RESET REFERENCE BIT EXTENDED、SET STORAGE KEY EXTENDED、および TEST PROTECTION の場合のアクセス例外も含まれます。
- 6 MOVE LONG、MOVE LONG EXTENDED、COMPARE LOGICAL LONG、および COMPARE LOGICAL LONG EXTENDED の場合は、特定オペランドについてアクセス例外が示されるのは、そのオペランドの R フィールドに偶数番号のレジスターが示されている場合のみです。
- 7 この例外が示されるのは、この例外の原因となる符号または数字がアクセス例外なしに取り出された場合のみです。
- 8 この例外が示されるのは、例外の設定に使用された数字、および符号が、アクセス例外なしで取り出され、かつ符号が有効であり、かつ例外の設定に使用された数字が有効である場合のみです。

図 6-6 (2/2). プログラム割り込み条件の優先順位

アクセス例外

アクセス例外は、絶対アドレス、命令アドレス、論理アドレス、実アドレス、または仮想アドレスを使用して記憶機構にアクセスしているときに検出される可能性のある例外です。したがって、アクセス・レジスター・モードでの例外には、以下のものがあります。

1. ALET 指定例外
2. ALEN 変換例外
3. ALE シーケンス例外
4. ASTE 妥当性例外
5. ASTE シーケンス例外
6. 拡張権限例外
7. アドレッシング例外 (ART テーブル)
8. ASCE タイプ例外
9. 領域第 1 変換例外
10. 領域第 2 変換例外
11. 領域第 3 変換例外
12. セグメント変換例外
13. ページ変換例外
14. 変換指定例外
15. アドレッシング例外 (DAT テーブル)
16. アドレッシング例外 (オペランドまたは命令)
17. 記憶保護例外 (キー制御保護、アクセス・リスト制御保護、ページ保護、および低アドレス保護)

DAT がオンであっても、アクセス・レジスター・モード以外のモードでは、上記の例外 8-17 (アクセス・リスト制御保護の場合を除く) が検出されることがあります。

DAT がオフの場合の例外は以下のとおりです。

1. アドレッシング例外 (オペランドまたは命令)
2. 記憶保護例外 (キー制御および低アドレス)

さらに、DAT がオフであっても、STORE REAL ADDRESS 命令では例外 1-17 が検出されることがあり、LOAD REAL ADDRESS 命令では例外 7、14、および 15 が検出されることがあり、INVALIDATE PAGE TABLE ENTRY 命令では例外 15 が検出されることがあります。

6-41ページの図6-7には、アクセス例外についてもっと詳しい情報を示してあります。

プログラミング上の注意: 6-41ページの図6-7の中の優先順位番号は変更することもできますが、ESA/390での対応する優先順位との比較を容易にするために、ESA/390の優先順位に合わせてあります。特に、B.1.A.1 - B.1.A.9はB.1 - B.9に変更することもできますが、ESA/390には、「B.1.B 制御レジスター 0のビット 8-12の正しくないエンコードによる変換指定例外」が含まれています。

A.	有効アドレスが 0-511 または 4096-4607 の範囲内にある格納タイプ・オペランド参照に起因する記憶保護例外（低アドレス保護）。DAT がオンであり、かつ、他の例外が存在するために変換に使用するアドレス・スペース制御エレメントを取得できない場合は、この例外は認識されません。
B.1.A.1	アクセス・レジスタのビット 0-6 がすべて 0 ではないことに起因する ALET 指定例外。 ¹
B.1.A.2	有効アクセス・リスト指定へのアクセスに関するアドレッシング例外。 ²
B.1.A.3	アクセス・リスト・エンタリーがリストの外部にあることに起因する ALEN 変換例外。 ¹
B.1.A.4	アクセス・リスト・エンタリーへのアクセスに関するアドレッシング例外。 ²
B.1.A.5	アクセス・リスト・エンタリーの I ビットの値が 1 であることに起因する ALEN 変換例外。 ¹
B.1.A.6	アクセス・レジスタ内のアクセス・リスト・エンタリー・シーケンス番号 (ALESN) が、アクセス・リスト・エンタリー内の ALESN に等しくないことに起因する ALE シーケンス例外。 ¹
B.1.A.7	ASN 第 2 テーブル・エンタリーへのアクセスに関するアドレッシング例外。 ²
B.1.A.8	ASN 第 2 テーブル・エンタリーの I ビットの値が 1 であることに起因する ASTE 妥当性例外。 ¹
B.1.A.9	アクセス・リスト・エンタリー内の ASN 第 2 テーブル・エンタリー・シーケンス番号 (ASTESN) が、ASN 第 2 テーブル・エンタリー内の ASTESN に等しくないことに起因する ASTE シーケンス例外。 ¹
	注： 例外 B.1.A.10 から B.1.A.12 までが認識されるのは、アクセス・リスト・エンタリーの専用ビットが 1 で、そのエンタリー内の ALEAX が制御レジスタ 8 の中の EAX に等しくない場合のみです。
B.1.A.10	権限テーブル・エンタリーがテーブルの外部にあることに起因する拡張権限例外。 ¹
B.1.A.11	権限テーブル・エンタリーへのアクセスに関するアドレッシング例外。 ²
B.1.A.12	(1) アクセス・リスト・エンタリー内の専用ビットが 0 ではなく、(2) アクセス・リスト・エンタリー内のアクセス・リスト・エンタリー許可指標が、制御レジスタ 8 の中の拡張許可指標に等しくなく、かつ (3) 拡張許可指標により選択された 2 次権限ビットが 1 ではないために起きる拡張権限例外。 ¹

図 6-7 (1/3). アクセス例外の優先順位

B.2.A	格納に対して保護されている仮想アドレスに対する格納タイプ・オペランド参照による記憶保護例外 (アクセス・リスト制御保護)。 ¹
B.2.B.1	アドレス・スペース制御エレメントが、領域第 2 テーブル指定、領域第 3 テーブル指定、またはセグメント・テーブル指定であるときに、命令アドレスまたはオペランド・アドレスのビット 0-10、0-21、または 0-32 が 0 でないことに起因する ASCE タイプの例外。 ³
B.2.B.2	アドレス・スペース制御エレメントで指定されているテーブル内の必須エントリーがテーブルの外部にあることに起因する、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外。 ³ 注: 例外 B.2.B.3 から B.2.B.6 までは、領域第 1 テーブル・エントリー、領域第 2 テーブル・エントリー、領域第 3 テーブル・エントリー、およびセグメント・テーブル・エントリーについて、それらのエントリーが使用される順序で認識されます。
B.2.B.3	テーブル・エントリーへのアクセスに関するアドレッシング例外。 ⁴
B.2.B.4	テーブル・エントリー内の I ビットの値が 1 であることに起因する、領域第 1 変換例外、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外。 ³
B.2.B.5	(1) テーブル・エントリー内の TT が、それを指定しているアドレス・スペース制御エレメント内の DT に等しくないか、指定している次に高いレベルのテーブル・エントリー内の TT より 1 だけ小さくないこと、または、(2) このエントリーがセグメント・テーブル・エントリーである場合はセグメント・テーブル・エントリー (アドレス・スペース・制御エレメント内の専用スペース・ビットが 1 の場合は共通セグメント・ビット) の中に無効な 1 があることに起因する変換指定例外。 ⁴
B.2.B.6	次に低いレベルのテーブル・エントリー (存在する場合) の必須エントリーがテーブルの外部にあることに起因する、領域第 2 変換例外、領域第 3 変換例外、またはセグメント変換例外。 ³
B.2.B.7	ページ・テーブル・エントリーへのアクセスに関するアドレッシング例外。 ⁵
B.2.B.8	ページ・テーブル・エントリー内の I ビットの値が 1 であることに起因するページ変換例外。 ^{3 7}
B.2.B.9	I ビットが 0 のページ・テーブル・エントリー (ビット 52 および 55) の中に無効な 1 があることに起因する変換指定例外。 ⁴ 注: 例外 B.3.A、B.3.B、および B.4 が認識されるのは、DAT がオフで、ページ・テーブル・エントリー内の I ビットが 0 の場合のみです。
B.3.A	格納に対して保護されている仮想アドレスに対する格納タイプ・オペランド参照に起因する記憶保護例外 (ページ保護)。 ⁶
B.3.B	命令またはオペランドへのアクセスに関するアドレッシング例外。
B.4.	保護されている命令位置またはオペランド位置にアクセスしようとしたことに起因する記憶保護例外 (キー制御保護)。

図 6-7 (2/3). アクセス例外の優先順位

説明:

- 1 アクセス・レジスター・モードでないときは、適用されません。TEST ACCESS の実行、および LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスの変換の場合は、適用されません。
- 2 アクセス・レジスター・モードでないときは適用されません。ただし、TEST ACCESS の実行の場合は適用されます。また、PSW ビット 16 が 2 進数 17 および 01 であるときは、LOAD REAL ADDRESS のオペランド・アドレスおよび STORE REAL ADDRESS の第 2 オペランド・アドレスの変換にも適用されます。
- 3 STORE REAL ADDRESS の第 2 オペランド・アドレスの変換の場合を除き、DAT がオフのときは適用されません。LOAD REAL ADDRESS および TEST PROTECTION のオペランド・アドレスには、適用されません。
- 4 LOAD REAL ADDRESS のオペランド・アドレスおよび STORE REAL ADDRESS の第 2 オペランド・アドレスの変換の場合を除き、DAT がオフのときは適用されません。
- 5 INVALIDATE PAGE TABLE ENTRY の実行の場合、および、LOAD REAL ADDRESS のオペランド・アドレスおよび STORE REAL ADDRESS の第 2 オペランド・アドレスの変換の場合を除き、DAT がオフのときは適用されません。
- 6 DAT がオフのときは適用されません。
- 7 MOVE PAGE で、両方のオペランドについて条件が真の場合は、第 2 オペランドについて例外が認識されます。また、条件コード・オプション・ビットが 1 の場合は、例外は認識されません。代わりに、第 1 オペランドについてのみ条件が真の場合は条件コード 1 が設定され、第 2 オペランドまたは両方のオペランドについて条件が真の場合は条件コード 2 が設定されます。

図 6-7 (3/3). アクセス例外の優先順位

ASN 変換例外

ASN 変換例外は、PROGRAM RETURN、PROGRAM TRANSFER、および SET SECONDARY ASN 命令で ASN を変換するプロセスに共通する例外です。図6-8 は、これらの例外と、それぞれが検出される優先順位を示しています。

1. ASN 第 1 テーブル・エントリへのアクセスに関するアドレッシング例外。
2. ASN 第 1 テーブル・エントリ内の I ビット (ビット 0) が 1 であることに起因する AFX 変換例外。
3. ASN 第 2 テーブル・エントリへのアクセスに関するアドレッシング例外。
4. ASN 第 2 テーブル・エントリ内の I ビット (ビット 0) が 1 であることに起因する ASX 変換例外。

図 6-8. ASN 変換例外の優先順位

サブスペース置き換え例外

サブスペース置き換え例外は、PROGRAM CALL、PROGRAM RETURN、PROGRAM TRANSFER、または SET SECONDARY ASN によるサブスペース置き換え操作中に認識される可能性がある例外です。図 6-9 は、これらの例外とそれぞれの優先順位を示しています。

1. ディスパッチ可能単位制御テーブルへのアクセスに関するアドレッシング例外。
2. サブスペース ASN 第 2 テーブル・エントリへのアクセスに関するアドレッシング例外。
3. サブスペース ASN 第 2 テーブル・エントリ内のビット 0 が 1 であることに起因する ASTE 妥当性例外。
4. ディスパッチ可能単位制御テーブル内の ASN 第 2 テーブル・エントリ・シーケンス番号が、サブスペース ASN 第 2 テーブル・エントリ内の ASN 第 2 テーブル・エントリ・シーケンス番号に等しくないことに起因する ASTE シーケンス例外。

図 6-9. サブスペース置き換え例外の優先順位

トレース例外

トレース例外は、トレース・テーブル・エントリーを形成しているときに発生する可能性がある例外です。6-44 ページの図6-10 は、これらの例外とそれぞれの優先順位を示しています。

- | | |
|-----|---|
| A. | エントリー・アドレスが 0-511 または 4096-4607 の範囲内にあることに起因する記憶保護例外 (低アドレス保護)。 |
| B.1 | 新規エントリーが次の 4K バイト境界に達するかまたはそれを越えることに起因するトレース・テーブル例外。 |
| B.2 | トレース・テーブル・エントリーへのアクセスに関するアドレッシング例外。 |

図 6-10. トレース例外の優先順位

再始動割り込み

再始動割り込みは、オペレーターまたは他の CPU が、指定したプログラムの実行を開始するための手段を提供します。この割り込みに対して CPU を使用不可にすることはできません。

再始動割り込みが起きると、旧 PSW が実記憶位置 288-303 に格納され、実行するプログラムの始めを示す新 PSW が実記憶位置 416-431 から取り出されます。命令長コードおよび割り込みコードは格納されません。

CPU が作動状態にある場合は、PSW の交換が行われるのは、現操作単位が完了し、この CPU が使用可能にされている他のすべての保留中の割り込み条件が処理された後です。CPU が停止状態にある場合は、CPU は作動状態になり、他の保留中の割り込みを処理する前に、まず PSW を交換します。

再始動割り込みは、再始動キーを活動化したときに開始されます。また、再始動指令を指定する SIGNAL PROCESSOR 命令を実行することにより、指定した CPU でこの操作を開始することもできます。

速度制御が命令ステップ位置に設定されているときは、再始動により PSW が交換された後で、操作単位または追加の割り込みのどちらが処理されるかは、予測不能です。

プログラミング上の注意: CPU がチェック停止状態にあるときに再始動を行うには、CPU をリセットする必要があります。情報の損失を最小限に抑えてリセットするには、システム・リセット通常キーを使用することができます。これを使用すると、制御レジスターも含めて

プログラムでアドレッシングできるレジスターの内容は消去されず、チャンネル・サブシステムがリセットされません。CPU リセット SIGNAL PROCESSOR 指令を使用すれば、チャンネル・サブシステムに影響を与えずに CPU をクリアすることができます。

監視プログラム呼び出し割り込み

監視プログラム呼び出し割り込みは、SUPERVISOR CALL 命令が実行されたときに発生します。この割り込みに対して CPU を使用不可にすることはできません。また、この命令が実行されると、即時に割り込みが起きます。

監視プログラム呼び出し割り込みが起きると、旧 PSW が実記憶位置 320-335 に格納され、新 PSW が実記憶位置 448-463 から取り出されます。

SUPERVISOR CALL 命令のビット位置 8-15 が、割り込みコードの右端の 1 バイトに入れられます。割り込みコードの左端の 1 バイトは 0 に設定されます。命令長コードは 1 ですが、この命令が EXECUTE を介して実行された場合は、命令長コードは 2 になります。

割り込みコードは実記憶位置 138-139 に格納され、命令長コードは実記憶位置 137 にあるバイトのビット位置 5 および 6 に格納され、実記憶位置 136 には 0 が入れられます。

割り込みの優先順位

1 つの命令の実行中に、割り込みの原因となるイベントが同時に幾つか起きることがあります。例えば、その命令がプログラム割り込みを引き起こすこともあり、外部割り込み要求を受け取るとこともあり、装置の誤動作が検出されることもあり、入出力割り込み要求が出されることあり、また、再始動キーが活動化されることもあります。プログラム割り込みの代わりに、監視プログラム呼び出し割り込みが起きることがあります。また、PER がアクティブのときは、その両方が起きることがあります。同時に出された複数の割り込み要求は、あらかじめ定められた順序で処理されます。

最高優先順位を持つのは緊急マシン・チェック条件です。この条件が起きたときは、現在の操作は中止または無効化されます。現在の操作の結果として発生するプログラム割り込みおよび監視プログラム呼び出し割り込みは、排除される可能性があります。保留中の抑制可能マシン・チェック条件がある場合は、緊急マシン・チェック割り込みと共にその条件も示されることがあります。緊急マシン・チェック条件の副次作用を制限するために

妥当と認められるすべての試みが行われるので、通常は、外部割り込み、入出力割り込み、および再始動割り込みに対する要求に影響が出ることはありません。

緊急マシン・チェック条件が存在しないときは、操作単位の終了時に同時に存在する複数の割り込み要求は、以下に示す優先順位の降順に従って受け入れられます。

- 監視プログラム呼び出し割り込み
- プログラム割り込み
- 抑制可能マシン・チェック割り込み
- 外部割り込み
- 入出力割り込み
- 再始動割り込み

同時に発生した複数の割り込み要求が処理されるときは、旧 PSW が格納され、最初に受け入れる命令に属する新 PSW が取り出されます。この新 PSW は、どの命令も実行されないうちに格納され、次の割り込みに関連した新 PSW が取り出されます。このようにして、処理すべき割り込みがなくなるまで、PSW の格納と取り出しが繰り返されます。新しい PSW がロードされるたびに、優先順位が再評価されます。再評価の際には、そのときまでに新たに保留状態になった追加の割り込みも評価の対象にされます。また、外部割り込みと入出力割り込み、および、抑制可能条件に起因するマシン・チェック割り込みが起きるのは、再評価の時点での現 PSW に、CPU がその原因による割り込みが可能な状態にあることが示されている場合のみです。

命令の実行は、最後に取り出された PSW を使用して再開されます。したがって、割り込みサブルーチンの実行順序は、PSW が取り出された順序の逆になります。

あるプログラム割り込み用の新 PSW で待ち状態が指定されておらず、命令アドレスが奇数である場合や、その PSW が原因でアクセス例外が認識される場合は、別のプログラム割り込みが起きます。この第 2 の割り込みは第 1 の割り込みと同じ受け入れ不能の PSW を導入するので、割り込みのストリングが設定されることとなります。この種のプログラム例外は後続の命令の実行の一部として認識されるものであり、割り込みのストリングは、外部割り込み、入出力割り込み、マシン・チェック割り込み、再始動割り込み、または停止機能により断ち切ることができます。

プログラム割り込みの新 PSW のビット位置 12 または未割り当てのビット位置に 1 が含まれている場合、または、ビット 31 および 32 が 24 ビット・アドレッシングを示しているときに、割り込みアドレスの左端 40 ビットが 0 でないか、ビット 31 および 32 が 31 ビット・アドレッシングを示しているときに、左端 33 ビットが 0 でない場合、または、ビット 31 が 1 のときにビット 32 が 0 である場合は、別のプログラム割り込みが起きます。この条件は、再始動条件、入出力条件、外部条件、抑制可能マシン・チェック条件、または停止機能より優先順位が高いため、割り込みのストリングを断ち切るには CPU リセットを使用する必要があります。

新 PSW が、直前に発生した割り込みを許可するものである場合は、他の割り込みクラスの割り込みのストリングが存在していることがあります。これには、マシン・チェック割り込み、外部割り込みのほか、ループを形成する CCW が原因で生成される PCI 条件による入出力割り込みが含まれます。さらに、複数の割り込みクラスを含む割り込みストリングが存在する可能性もあります。例えば、CPU タイマーが負で、CPU タイマー・サブクラス・マスクが 1 であるとします。ここで、外部の新 PSW の未割り当てビットの 1 つが 1 であり、プログラムの新 PSW が外部割り込みに対して使用可能にされているとすれば、外部割り込みとプログラム割り込みが交互に起きる割り込みのストリングが発生します。これよりさらに複雑な割り込みのストリングが生じる可能性もあります。処理を要する割り込みが残っている限り、停止機能を使用して割り込みのストリングを断ち切ることはできないので、CPU リセットが必要です。

同様に、CPU が使用可能することのできない記憶位置を示すプレフィックス値を使用して割り込みが試みられたときに存在する条件を打ち切るためにも、CPU リセットを呼び出す必要があります。

CPU が使用可能にされているすべての要求についての割り込みは、CPU が停止状態になる前に起きます。CPU が停止状態にあるときは、最高優先順位を持つのは再始動です。

プログラミング上の注意: 同時に発生する複数の割り込み要求が受け入れられる順序は、マスキングによってある程度変更できます。

第7章 一般命令

データ・フォーマット	7-2	COMPARE LOGICAL LONG UNICODE	7-50
2 進整数の表現	7-3	COMPARE LOGICAL STRING	7-53
2 進数演算	7-3	COMPARE UNTIL SUBSTRING EQUAL	7-54
符号付き 2 進数演算	7-4	COMPRESSION CALL	7-57
加算および減算	7-4	CONVERT TO BINARY	7-68
固定小数点オーバーフロー	7-4	CONVERT TO DECIMAL	7-68
符号なし 2 進数演算	7-4	CONVERT UNICODE TO UTF-8	7-69
符号付き比較と論理比較	7-5	CONVERT UTF-8 TO UNICODE	7-72
命令	7-5	COPY ACCESS	7-75
ADD	7-16	DIVIDE	7-75
ADD HALFWORD	7-16	DIVIDE LOGICAL	7-75
ADD HALFWORD IMMEDIATE	7-16	DIVIDE SINGLE	7-76
ADD LOGICAL	7-17	EXCLUSIVE OR	7-77
ADD LOGICAL WITH CARRY	7-17	EXECUTE	7-78
AND	7-18	EXTRACT ACCESS	7-79
AND IMMEDIATE	7-19	EXTRACT PSW	7-79
BRANCH AND LINK	7-20	INSERT CHARACTER	7-79
BRANCH AND SAVE	7-20	INSERT CHARACTERS UNDER MASK	7-79
BRANCH AND SAVE AND SET MODE	7-21	INSERT IMMEDIATE	7-80
BRANCH AND SET MODE	7-22	INSERT PROGRAM MASK	7-81
BRANCH ON CONDITION	7-23	LOAD	7-81
BRANCH ON COUNT	7-24	LOAD ACCESS MULTIPLE	7-81
BRANCH ON INDEX HIGH	7-25	LOAD ADDRESS	7-82
BRANCH ON INDEX LOW OR EQUAL	7-25	LOAD ADDRESS EXTENDED	7-82
BRANCH RELATIVE AND SAVE	7-26	LOAD ADDRESS RELATIVE LONG	7-83
BRANCH RELATIVE AND SAVE LONG	7-26	LOAD AND TEST	7-83
BRANCH RELATIVE ON CONDITION	7-26	LOAD COMPLEMENT	7-84
BRANCH RELATIVE ON CONDITION		LOAD HALFWORD	7-84
LONG	7-26	LOAD HALFWORD IMMEDIATE	7-85
BRANCH RELATIVE ON COUNT	7-27	LOAD LOGICAL	7-85
BRANCH RELATIVE ON INDEX HIGH	7-28	LOAD LOGICAL CHARACTER	7-85
BRANCH RELATIVE ON INDEX LOW OR		LOAD LOGICAL HALFWORD	7-85
EQUAL	7-28	LOAD LOGICAL IMMEDIATE	7-86
CHECKSUM	7-29	LOAD LOGICAL THIRTY ONE BITS	7-86
COMPARE	7-33	LOAD MULTIPLE	7-86
COMPARE AND FORM CODEWORD	7-33	LOAD MULTIPLE DISJOINT	7-87
COMPARE AND SWAP	7-40	LOAD MULTIPLE HIGH	7-87
COMPARE DOUBLE AND SWAP	7-40	LOAD NEGATIVE	7-88
COMPARE HALFWORD	7-42	LOAD PAIR FROM QUADWORD	7-88
COMPARE HALFWORD IMMEDIATE	7-42	LOAD POSITIVE	7-88
COMPARE LOGICAL	7-42	LOAD REVERSED	7-89
COMPARE LOGICAL CHARACTERS		MONITOR CALL	7-90
UNDER MASK	7-43	MOVE	7-91
COMPARE LOGICAL LONG	7-44	MOVE INVERSE	7-91
COMPARE LOGICAL LONG EXTENDED	7-46	MOVE LONG	7-92

MOVE LONG EXTENDED	7-95	STORE ACCESS MULTIPLE	7-135
MOVE LONG UNICODE	7-99	STORE CHARACTER	7-136
MOVE NUMERICS	7-102	STORE CHARACTERS UNDER MASK	7-136
MOVE STRING	7-103	STORE CLOCK	7-137
MOVE WITH OFFSET	7-104	STORE CLOCK EXTENDED	7-137
MOVE ZONES	7-104	STORE HALFWORD	7-139
MULTIPLY	7-105	STORE MULTIPLE	7-139
MULTIPLY HALFWORD	7-106	STORE MULTIPLE HIGH	7-140
MULTIPLY HALFWORD IMMEDIATE	7-106	STORE PAIR TO QUADWORD	7-140
MULTIPLY LOGICAL	7-106	STORE REVERSED	7-140
MULTIPLY SINGLE	7-107	SUBTRACT	7-141
OR	7-108	SUBTRACT HALFWORD	7-142
OR IMMEDIATE	7-109	SUBTRACT LOGICAL	7-142
PACK	7-109	SUBTRACT LOGICAL WITH BORROW	7-143
PACK ASCII	7-110	SUPERVISOR CALL	7-144
PACK UNICODE	7-111	TEST ADDRESSING MODE	7-144
PERFORM LOCKED OPERATION	7-112	TEST AND SET	7-144
ROTATE LEFT SINGLE LOGICAL	7-127	TEST UNDER MASK (TEST UNDER MASK HIGH, TEST UNDER MASK LOW)	7-145
SEARCH STRING	7-128	TRANSLATE	7-146
SET ACCESS	7-129	TRANSLATE AND TEST	7-147
SET ADDRESSING MODE	7-129	TRANSLATE EXTENDED	7-148
SET PROGRAM MASK	7-130	TRANSLATE ONE TO ONE	7-150
SHIFT LEFT DOUBLE	7-130	TRANSLATE ONE TO TWO	7-150
SHIFT LEFT DOUBLE LOGICAL	7-131	TRANSLATE TWO TO ONE	7-150
SHIFT LEFT SINGLE	7-132	TRANSLATE TWO TO TWO	7-150
SHIFT LEFT SINGLE LOGICAL	7-132	UNPACK	7-155
SHIFT RIGHT DOUBLE	7-133	UNPACK ASCII	7-155
SHIFT RIGHT DOUBLE LOGICAL	7-133	UNPACK UNICODE	7-156
SHIFT RIGHT SINGLE	7-134	UPDATE TREE	7-157
SHIFT RIGHT SINGLE LOGICAL	7-135		
STORE	7-135		

この章では、10 進数命令と浮動小数点数命令以外の、本書に現れるすべての非特権命令について説明します。

データ・フォーマット

一般命令は、データを、符号付き 2 進整数、符号なし 2 進整数、非構造化論理データ、および 10 進数データの 4 つのタイプのいずれかとして取り扱います。変換命令、パック命令、およびアンパック命令では、データを 10 進数として扱います。10 進数データについては、第 8 章、『10 進数命令』を参照してください。

一般命令が取り扱うデータは、汎用レジスターまたは記憶域の中にあるか、あるいは命令ストリームから取り出されます。一般命令の中には、PSW または TOD クロック内のデータを操作するものもあります。

記憶域と記憶域との間の操作では、オペランド・フィールドの定義の結果として両方のオペランド・フィールドがオーバーラップする場合があります。このオーバーラップの影響は操作に応じて異なります。COMPARE または TRANSLATE AND TEST のように、オペランドが変更されない場合には、オーバーラップがあっても操作の実行に影響はありません。MOVE や TRANSLATE のような命令の場合には、一方のオペランドが新規データで置き換えられるので、オーバーラップの程度とデータの取り出しまたは格納の仕方によっては操作の実行に影響が出ることがあります。オーバーラップしたオペランドの影響を評価する場合は、データは一度に 8 ビットのバイト単位で処理されるものと見なされます。MOVE LONG および MOVE INVERSE のオペランドには、特殊な規則が適用されます。アクセス・レジスタ

ー・モードでのオーバーラップの検出方法については、5-80ページの『単一命令内のインターロック』を参照してください。

2 進整数の表現

2 進整数は、符号付きまたは符号なしのデータとして扱われます。

符号なし 2 進整数では、すべてのビットは、その数の絶対値を表すために使用されます。長さの異なる 2 つの符号なし 2 進整数を加算する場合、短い方の数の左側には 0 が埋め込まれて長さがそろえられます。

操作によっては、結果を求めるために、該当の数の 1 の補数が使用されることがあります。ある数の 1 の補数とは、その数の個々のビットを、符号を含めて逆転させたものです。

符号付き 2 進整数では、左端のビットが符号を表し、その後に数値フィールドが続きます。正の数値は、符号ビットを 0 に設定した真の 2 進表記法で表されます。値が 0 の場合は、符号ビットも含めてすべてのビットが 0 に設定されます。負の数値は、符号ビットの位置に 1 を持った 2 の補数の 2 進表記法で表されます。

つまり、負の数値は、それと同じ絶対値を持つ正数の 2 の補数で表されます。ある数の 2 の補数は、その数の 1 の補数を求め、その右端のビット位置に値 1 を加算し、符号位置への繰り上がりを許し、符号位置からの繰り上がりは無視するという方法で行われます。

この数表現では、右端の部分はその数の無限に長い表記とみなすことができます。正数の場合は、その数の最上位ビットの左側にあるビットはすべて 0 です。負数の場合は、これらのビットはすべて 1 です。したがって、符号付きオペランドの左側をビットで拡張する必要がある場合は、これらのビットをオペランドの符号ビットと等価に設定することによって、この拡張が行われます。

符号付き 2 進整数の表記には負の 0 は含まれていません。一定の長さをとった場合の数の範囲は、0 でない負数の集合が、0 でない正数の集合より 1 つだけ大きくなります。最大正数は、符号ビットが 0 で、それに続くすべてのビットが 1 である構成です。一方、最大負数（絶対値が最大の負数）は、符号ビットが 1 で、それに続くすべてのビットが 0 である構成です。

符号付き 2 進整数は、どちらの符号の場合でも、0 および最大負数を除き、2 の補数を作ることによって同じ大きさの反対符号の数に変換できます。ある数の 2 の補数

を作ることは、その数を 0 から引くことと同じです。0 の 2 の補数は 0 です。

最大負数の 2 の補数を、その数と同じビット数で表すことはできません。LOAD COMPLEMENT などの操作で最大負数の 2 の補数を生成しようとする、結果は最大負数となり、固定小数点オーバーフロー例外が認識されます。ただし、最大負数の補数が中間結果として計算されるだけで、最終結果が表現可能な範囲内に収まる場合には、オーバーフローは発生しません。これに該当する例が、-1 から最大負数を引くという減算です。一定の長さの 2 つの最大負数の積は、その 2 倍の長さの正数として表現できます。

本書で符号付き 2 進整数という場合、それには符号ビットも含まれているものとします。したがって、「32 ビットの符号付き 2 進整数」とは 31 個の数字ビットと 1 個の符号ビットから成る整数であり、「64 ビットの符号付き 2 進整数」とは 63 個の数字ビットと 1 個の符号ビットから成る整数のことです。

算術演算では、符号付き 2 進整数の数値フィールドの繰り上がりは符号ビットに繰り越されます。ただし、代数的な左シフトでは、有効数字ビットが次の桁にシフトされても、符号ビットは変化しません。

プログラミング上の注意:

1. 符号付き 2 進整数の 2 の補数を作るために、右端の 1 ビットの左側にあるすべてのビットを逆転させ、右端の 1 ビットとその右側にあるすべての 0 のビットをそのまま残すという代替方法もあります。
2. 符号付き 2 進整数では、数字ビットが正の値を表し、符号が値 0、または最大負数の値を表すと考えることもできます。

2 進数演算

レジスターと記憶域との間、またはレジスターとレジスターとの間の 2 進数演算を実行する命令の多くは、3 種類のオペランド長の組み合わせに対応する 3 つの命令のセットとして提供されています。これらの 3 つの命令は、名前は同じですが、命令コードとニーモニックは異なります。例えば、ADD (A) は 32 ビットのオペランドに対して実行されて 32 ビットの結果を生成し、ADD (AG) は 64 ビットのオペランドに対して実行されて 64 ビットの結果を生成し、ADD (AGF) は 64 ビットのオペランドおよび 32 ビットのオペランドに対して実行されて 64 ビットの結果を生成します。ニーモニックの中

に文字「G」のみがあれば、64 ビット専用の操作を表し、文字「GF」があれば、32 ビットから 64 ビットへの操作を表します。

32 から 64 ビットへの操作では、中間結果は 64 ビットです。LOAD COMPLEMENT (LCGFR) は、オーバーフローを起こさずに、32 ビットの最大負数の 2 の補数を生成します。

汎用レジスター内の 32 ビット・オペランドは、そのレジスターのビット位置 32-63 に入っています。オペランドに対する操作 (ADD (A) など) では、レジスターのビット 0-31 は使用されず、したがって変更もされません。汎用レジスター内の 64 ビット・オペランドは、そのレジスターのビット位置 0-63 に入り、すべてのビットがオペランドに対する操作 (ADD (AG) など) に使用されます。しかし、ニーモニックに「G」が含まれていない一部の命令では、64 ビット・オペランドが使用され、このオペランドの左端の 32 ビットは奇数/偶数の汎用レジスター・ペアの偶数レジスターのビット位置 32-63 に、右端の 32 ビットはこのペアの奇数レジスターのビット位置 32-63 に入っている値です。

記憶域内での 32 ビット・オペランドのビットの番号は 0-31 です。このオペランドが汎用レジスターのビット位置 32-63 に入ると、ビット番号は 32-63 になります。

符号付き 2 進数演算

加算および減算

符号付き 2 進整数の加算では、各オペランドのすべてのビット (符号ビットも含む) が足し合わされます。オペランドの片方が短い場合は、短い方のオペランドの左側が符号ビットの値で拡張され、長さがそろえられます。

汎用レジスター内の 32 ビット符号付き 2 進整数の場合、符号ビットはそのレジスターのビット 32 です。汎用レジスター内の 64 ビット符号付き 2 進整数の場合、符号ビットはそのレジスターのビット 0 です。

減算は、第 2 オペランドの 1 の補数と値 1 とを、第 1 オペランドに加算することによって行われます。

固定小数点オーバーフロー

符号付き 2 進数の加算または減算で固定小数点オーバーフロー条件が発生するのは、符号ビット位置からの繰り上がりと左端の数字ビット位置からの繰り上がり不一致の場合です。オーバーフローが検出されても、加算の結果には影響しません。数学的に言えば、符号付きの加算および減算では、結果が符号付き 2 進整数の表現可

能な範囲を超えると、固定小数点オーバーフローが発生します。具体的に言えば、32 ビットの符号付き 2 進整数を取り扱う ADD (A) および SUBTRACT (S) の場合、求める真の結果が $+2^{31}$ 以上または -2^{31} 未満になるとオーバーフローが発生します。オーバーフロー後に汎用レジスターに入れられる実際の結果は、真の結果とは 2^{32} だけ違ったものになります。固定小数点オーバーフローは、プログラム・マスクで許されていれば、プログラム割り込みの原因になります。同様に、64 ビットの符号付き 2 進整数を取り扱う ADD (AG) および SUBTRACT (SG) の場合、求める真の結果が $+2^{63}$ 以上または -2^{63} 未満になるとオーバーフローが発生します。オーバーフロー後に実際に汎用レジスターにロードされる値は、真の結果とは 2^{64} だけ違ったものになります。ADD (AGF) と SUBTRACT (SGF) では、64 ビットの結果もオーバーフローに関する規則も ADD (AG) と SUBTRACT (SG) の場合と同じです。

SHIFT LEFT SINGLE 命令と SHIFT LEFT DOUBLE 命令では、結果が符号付き 2 進整数の範囲を超えると、オーバーフローが発生します。実際の結果は、元の符号が結果の符号として残るという点で、加減算の場合とは異なります。

符号なし 2 進数演算

符号なし 2 進整数の加算では、各オペランドのすべてのビット (符号ビットも含む) が足し合わされます。減算は、第 2 オペランド (減数) の 1 の補数と値 1 とを、第 1 オペランド (被減数) に加算することによって行われます。どちらの場合も、オペランドの片方が短い場合は、短い方のオペランドの左側が 0 で拡張され、長さがそろえられます。減算では、この拡張はオペランドの補数を取る前に適用され、その結果が値 1 に適用されません。

符号なし 2 進数演算は、フィールド X、B、および D を加算するときにアドレス演算で使用されます。(5-7 ページの『アドレスの生成』を参照してください。) また、TRANSLATE および TRANSLATE AND TEST で関数バイトのアドレスを得るときにも、符号なし 2 進数演算が使用されます。さらに、32 ビットおよび 64 ビットの符号なし 2 進整数の場合にも、ADD LOGICAL、ADD LOGICAL WITH CARRY、DIVIDE LOGICAL、MULTIPLY LOGICAL、SUBTRACT LOGICAL、および SUBTRACT LOGICAL WITH BORROW の各命令で符号なし 2 進数演算が使用されます。

オペランドの長さが同じなら、ADD (A, AG, AGF) と ADD LOGICAL (AL, ALG, ALGF) では、同じ 32 ビットまたは 64 ビットの結果が生成されます。ただ、この 2 種類の命令では結果の解釈が異なります。ADD では、結果は符号付き 2 進整数であると解釈され、符号、大きさ、およびオーバーフローについての検査の後で、その検査結果に基づいて条件コードが設定されます。ADD LOGICAL では、結果は符号なし 2 進整数であると解釈され、結果が 0 であるかどうか、ビット位置 32 (32 ビット整数の場合) またはビット位置 0 (64 ビット整数の場合) から繰り上がりがあったかどうかに基づいて、条件コードが設定されます。このような繰り上がりはオーバーフローとは見なされず、ADD LOGICAL では、オーバーフローによるプログラム割り込みは発生しません。

SUBTRACT LOGICAL が ADD LOGICAL と異なるのは、第 2 オペランドの 1 の補数と値 1 とが、第 1 オペランドに加算されるという点です。

ADD LOGICAL WITH CARRY の場合は、直前の操作からの繰り上がりがあれば、現 PSW のビット 18 に値 1 が入ります。ビット 18 は、PSW の 2 ビット条件コードの左端のビットです。SUBTRACT LOGICAL WITH BORROW の場合は、直前の操作からの繰り下がりがあれば、ビット 18 に値 0 が入ります。繰り下がりがあるということは、繰り上がりがないということです。

プログラミング上の注意:

1. 論理加算と論理減算は、複数精度 2 進整数オペランドの算術演算を実行するために使用できます。したがって、複数精度の加算の場合、まず ADD LOGICAL を使用して、両オペランドの対応する下位部分を加算します。次に ADD LOGICAL WITH CARRY を使用して、両オペランドのその他の対応部分を、右から左へと移動しながら加算します。複数精度オペランドが符号付きの場合は、最上位部分に対して ADD を使用する必要があります。その結果、条件コードにオーバーフロー、または正しい符号と結果全体の大きさが示されます。オーバーフローは、プログラム・マスクにより許されていれば、固定小数点オーバーフローのプログラム割り込みによっても示されます。ADD を使用するとき、最上位部分の和に示されている条件コードがそのすぐ下位の部分の加算からの繰り上がりである場合は、最上位部分の和に値 1 を追加する必要があります。
2. ADD LOGICAL のもう 1 つの使い方としては、オーバーフローを示さずにすべて 1 からすべて 0 に

循環することが許されている、2 進カウンターの値を増加させるために使用できます。

符号付き比較と論理比較

比較演算では、2 つのオペランドが等しいかどうかを判別し、ほとんどの比較演算では、等しくない 2 つのオペランドのどちらが大きい (高い) かも判別します。符号付き 2 進比較演算では、オペランドは符号付き 2 進整数として扱われ、論理比較演算では、オペランドは符号なし 2 進整数または非構造化データとして扱われます。

COMPARE (C, CG, CGF) および COMPARE HALFWORD は符号付き 2 進比較演算です。これらの命令は、SUBTRACT (S, SG, SGF) および SUBTRACT HALFWORD を実行し、どちらのオペランドも置き換えず、結果として得られる差を条件コードの設定のみに使用する場合と同等です。この操作では、符号が反対で差が 2^{63} 以上あるような数字の比較ができません。したがって、SUBTRACT とは異なり、COMPARE ではオーバーフローは起きません。

論理比較では、2 つのオペランドが左から右へと 1 バイトずつ比較されます。すべての対応バイトが等しければ、2 つのオペランドが等しいことになります。オペランドが等しくない場合は、最初の等しくない一対のバイトの中の対応ビット位置を左から右へと比較することによって、比較結果が決定されます。最初の等しくない一対のビットに、0 の値のビットがある方が小さいオペランドで、1 の値のビットがある方が大きいオペランドです。残りのビットやバイトの位置で比較結果が変わることはないので、最初の等しくないビットのペアを検出したら、そこから先のオペランド比較を続行する必要はありません。

命令

一般命令およびその二モニク、フォーマット、および命令コードを、7-8 ページの図 7-1 に示します。この図には、ESA/390 に比べて z/Architecture で新しく追加された命令、条件コードが設定される場合、アクセス・レジスターを指定する命令フィールド、および、オペランド指定、データ、または結果でプログラム割り込みを起こすような例外条件も示してあります。

図 7-1 では、z/Architecture で新規に加わった命令は「N」で示されています。z/Architecture の新規命令の中には、ESA/390 にも追加されているものが幾つかあり、それらの命令は「N3」で示されています。

命令のオペランドが両方とも 32 ビット・オペランドの場合は、その命令のニーモニックにはオペランドの長さを示す文字は含まれていません。名前が同じで 64 ビット・オペランドの命令の場合は、ニーモニックに「G」の文字が含まれています。名前は同じだが第 1 オペランドが 64 ビットで第 2 オペランドが 32 ビットの場合は、ニーモニックに「GF」の文字が含まれています。図7-1 では、32 ビット・オペランドのみの命令と、それと同名でニーモニックに「G」または「GF」が追加されているその他の命令がある場合は、最初の命令には名前の後に「(32)」が付き、その他の命令には (G か GF かに応じて) それぞれ「(64)」または「(64<32)」が付いています。32 ビット・オペランド長の命令の中には、対応する 64 ビット・オペランド長の命令がないものもあります。このような命令の名前の後には「(32)」は付いていません。ただし、乗算または除算の命令にはすべて、オペランド長が示されています (概算の場合もあります)。

命令フォーマット、オペランドの指定と長さ、およびアドレス生成に関する詳しい定義については、5-2 ページの『命令』を参照してください。そこで述べた一般規則から外れる例外があれば、個々の命令の説明の中で明示してあります。

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、32 ビット・オペランドの LOAD AND TEST では、LTR がニーモニックで、R₁、R₂ がオペランド指定です。

プログラミング上の注意:

1. 3 モード・アドレッシングは、以下の例外を除き、一般命令において論理記憶域アドレスをどのように扱うかに対して影響を与えるだけです。

BRANCH AND LINK (BAL、BALR)、
BRANCH AND SAVE (BAS、BASR)、
BRANCH AND SAVE AND SET MODE、
BRANCH AND SET MODE、BRANCH RELATIVE AND SAVE、および BRANCH RELATIVE AND SAVE LONG の各命令は、24 ビットまたは 31 ビット・アドレッシング・モードでは、ESA/390 の場合と同様に、汎用レジスター R₁ のビット位置 32-39 に情報を入れ、64 ビット・アドレッシング・モードでは、これらのビットにアドレス・ビットを入れます。

BRANCH AND SAVE AND SET MODE 命令は、24 ビットまたは 31 ビット・アドレッシング・モードでは汎用レジスター R₁ のビット位置

63 に 0 を入れ、64 ビット・アドレッシング・モードではそのビット位置に 1 を入れます。

BRANCH AND SET MODE 命令は、24 ビットまたは 31 ビット・アドレッシング・モードでは汎用レジスター R₁ のビット位置 63 の内容を無変更のままにし、64 ビット・アドレッシング・モードではそのビット位置に 1 を入れます。

次に示す命令は、24 ビットまたは 31 ビット・アドレッシング・モードでは汎用レジスターのビット 0-31 を変更せず、64 ビット・アドレッシング・モードではこれらのビット内でアドレスまたは長さ情報を配置または更新するものもあります。また、レジスター内の結果の左端バイトの取り扱いは、アドレッシング・モードが 24 ビット・モードか 31 ビット・モードかに応じて異なる場合もあります。

- BRANCH AND LINK (BAL、BALR)
- BRANCH AND SAVE (BAS、BASR)
- BRANCH AND SAVE AND SET MODE
- BRANCH RELATIVE AND SAVE
- BRANCH RELATIVE AND SAVE LONG
- CHECKSUM
- COMPARE AND FORM CODEWORD
- COMPARE LOGICAL LONG
- COMPARE LOGICAL LONG EXTENDED
- COMPARE LOGICAL LONG UNICODE
- COMPARE LOGICAL STRING
- COMPARE UNTIL SUBSTRING EQUAL
- COMPRESSION CALL
- CONVERT UNICODE TO UTF-8
- CONVERT UTF-8 TO UNICODE
- LOAD ADDRESS
- LOAD ADDRESS EXTENDED
- LOAD ADDRESS RELATIVE LONG
- MOVE LONG
- MOVE LONG EXTENDED
- MOVE LONG UNICODE
- MOVE STRING
- SEARCH STRING
- TRANSLATE EXTENDED
- TRANSLATE AND TEST
- TRANSLATE ONE TO ONE
- TRANSLATE ONE TO TWO
- TRANSLATE TWO TO ONE
- TRANSLATE TWO TO TWO
- UPDATE TREE

上のリストに示した命令は、モーダル命令と呼ばれることもあります。

2. 汎用レジスタのビット 0-31 を変更する命令には、2 つのタイプがあります。第 1 のタイプは、64 ビット・アドレッシング・モードで実行されているときのモーダル命令です (上のリストを参照)。第 2 のタイプは、アドレッシング・モードに関係なく、64 ビットの結果オペランドを単一の汎用レジスタに入れ、128 ビットの結果オペランドを偶数/奇数ペアの汎用レジスタに入れる命令です。

第 2 タイプの命令のほとんどは、ニーモニックに「G」または「GF」の文字が含まれています。現行のアドレッシング・モードに関係なく汎用レジスタのビット 0-31 を変更する (または変更する可能性のある) その他の命令は、次のとおりです。

- AND IMMEDIATE (NIHH, NIHL のみ)
- INSERT CHARACTERS UNDER MASK (ICMH のみ)
- INSERT IMMEDIATE (IIHH, IIHL のみ)
- LOAD LOGICAL IMMEDIATE
- LOAD MULTIPLE DISJOINT
- LOAD MULTIPLE HIGH
- LOAD PAIR FROM QUADWORD
- OR IMMEDIATE (OIHH, OIHL のみ)

第 2 タイプのすべての命令を総称して、「G タイプ」命令とも言います。

64 ビット・アドレッシング・モードで実行されておらず、G タイプの命令も含まれていないプログラムは、どの汎用レジスタのビット 0-31 も変更することはできません。

3. G タイプの命令を含んでいない古いプログラムを 64 ビット・アドレッシング・モードで正常に実行できる状況は、開発意図に含まれておらず、予期されてもいません。しかし、場合によっては、特に (プログラミング規則により) 古いプログラムに制御権が渡ったときに汎用レジスタのビット 0-31 が常にすべて 0 になるように設定されている場合は、正常実行が可能なこともあります。

4. 拡張変換機能 2 がインストールされている場合は、次に示す追加の一般命令を使用できます。

- COMPARE LOGICAL LONG UNICODE
- MOVE LONG UNICODE
- PACK ASCII
- PACK UNICODE
- TRANSLATE ONE TO ONE
- TRANSLATE ONE TO TWO
- TRANSLATE TWO TO ONE
- TRANSLATE TWO TO TWO
- UNPACK ASCII
- UNPACK UNICODE

命令	ニーモニック	特性					命令コード
ADD (32) ADD (64) ADD (64<32) ADD (32) ADD (64)	AR AGR AGFR A AG	RR C RRE C N RRE C N RX C RXE C N			IF IF IF IF IF	B ₂ B ₂	1A B908 B918 5A E308
ADD (64<32) ADD HALFWORD ADD HALFWORD IMMEDIATE (32) ADD HALFWORD IMMEDIATE (64) ADD LOGICAL (32)	AGF AH AHI AGHI ALR	RXE C N RX C RI C RI C N RR C	A A		IF IF IF IF	B ₂ B ₂	E318 4A A7A A7B 1E
ADD LOGICAL (64) ADD LOGICAL (64<32) ADD LOGICAL (32) ADD LOGICAL (64) ADD LOGICAL (64<32)	ALGR ALGFR AL ALG ALGF	RRE C N RRE C N RX C RXE C N RXE C N	A A A			B ₂ B ₂ B ₂	B90A B91A 5E E30A E31A
ADD LOGICAL WITH CARRY (32) ADD LOGICAL WITH CARRY (64) ADD LOGICAL WITH CARRY (32) ADD LOGICAL WITH CARRY (64) AND (32)	ALCR ALCGR ALC ALCG NR	RRE C N3 RRE C N RXE C N3 RXE C N RR C	A A			B ₂ B ₂	B998 B988 E398 E388 14
AND (64) AND (32) AND (64) AND (文字) AND (即値)	NGR N NG NC NI	RRE C N RX C RXE C N SS C SI C	A A A A			ST ST B ₁ B ₁	B980 54 E380 D4 94
AND IMMEDIATE (高 高) AND IMMEDIATE (高 低) AND IMMEDIATE (低 高) AND IMMEDIATE (低 低) BRANCH AND LINK	NIHH NIHL NILH NILL BALR	RI C N RI C N RI C N RI C N RR			T	B	A54 A55 A56 A57 05
BRANCH AND LINK BRANCH AND SAVE BRANCH AND SAVE BRANCH AND SAVE AND SET MODE BRANCH AND SET MODE	BAL BASR BAS BASSM BSM	RX RR RX RR RR			T T T	B B B B B	45 0D 4D 0C 0B
BRANCH ON CONDITION BRANCH ON CONDITION BRANCH ON COUNT (32) BRANCH ON COUNT (64) BRANCH ON COUNT (32)	BCR BC BCTR BCTGR BCT	RR RX RR RRE N RX			φ ¹	B B B B B	07 47 06 B946 46
BRANCH ON COUNT (64) BRANCH ON INDEX HIGH (32) BRANCH ON INDEX HIGH (64) BRANCH ON INDEX LOW OR EQUAL (32) BRANCH ON INDEX LOW OR EQUAL (64)	BCTG BXH BXHG BXLE BXLEG	RXE N RS RSE N RS RSE N				B B B B B	E346 86 EB44 87 EB45

図 7-1 (1/8). 一般命令の要約

命令	二 モ ニ ック	特性						命 令 コ ー ド
BRANCH RELATIVE AND SAVE BRANCH RELATIVE AND SAVE LONG BRANCH RELATIVE ON CONDITION BRANCH RELATIVE ON CONDITION LONG BRANCH RELATIVE ON COUNT (32)	BRAS BRASL BRC BRCL BRCT	RI RIL N3 RI RIL N3 RI					B B B B B	A75 C05 A74 C04 A76
BRANCH RELATIVE ON COUNT (64) BRANCH RELATIVE ON INDEX HIGH (32) BRANCH RELATIVE ON INDEX HIGH (64) BRANCH RELATIVE ON INDEX L OR E (32) BRANCH RELATIVE ON INDEX L OR E (64)	BRCTG BRXH BRXHG BRXLE BRXLG	RI N RSI RIE N RSI RIE N					B B B B B	A77 84 EC44 85 EC45
CHECKSUM COMPARE (32) COMPARE (64) COMPARE (64<32) COMPARE (32)	CKSM CR CGR CGFR C	RRE C RR C RRE C N RRE C N RX C	A SP A				R ₂ B ₂	B241 19 B920 B930 59
COMPARE (64) COMPARE (64<32) COMPARE AND FORM CODEWORD COMPARE AND SWAP (32) COMPARE AND SWAP (64)	CG CGF CFC CS CSG	RXE C N RXE C N S C RS C RSE C N	A A A SP A SP A SP	II	GM \$ \$		I1 ST ST B ₂ B ₂	E320 E330 B21A BA EB30
COMPARE DOUBLE AND SWAP (32) COMPARE DOUBLE AND SWAP (64) COMPARE HALFWORD COMPARE HALFWORD IMMEDIATE (32) COMPARE HALFWORD IMMEDIATE (64)	CDS CDSG CH CHI CGHI	RS C RSE C N RX C RI C RI C N	A SP A SP A		\$ \$	ST ST	B ₂ B ₂ B ₂	BB EB3E 49 A7E A7F
COMPARE LOGICAL (32) COMPARE LOGICAL (64) COMPARE LOGICAL (64<32) COMPARE LOGICAL (32) COMPARE LOGICAL (64)	CLR CLGR CLGFR CL CLG	RR C RRE C N RRE C N RX C RXE C N	A A A				B ₂ B ₂	15 B921 B931 55 E321
COMPARE LOGICAL (64<32) COMPARE LOGICAL (文字) COMPARE LOGICAL (即値) COMPARE LOGICAL C. UNDER MASK (高) COMPARE LOGICAL C. UNDER MASK (低)	CLGF CLC CLI CLMH CLM	RXE C N SS C SI C RSE C N RS C	A A A A A				B ₁ B ₂ B ₁ B ₂ B ₂	E331 D5 95 EB20 BD
COMPARE LOGICAL LONG COMPARE LOGICAL LONG EXTENDED COMPARE LOGICAL LONG UNICODE COMPARE LOGICAL STRING COMPARE UNTIL SUBSTRING EQUAL	CLCL CLCLE CLCLU CLST CUSE	RR C RS C RSE C E2 RRE C RRE C	A SP A SP A SP A SP A SP	II II	GO GM		R ₁ R ₂ R ₁ R ₃ R ₁ R ₂ R ₁ R ₂ R ₁ R ₂	0F A9 EB8F B25D B257

図 7-1 (2/8). 一般命令の要約

命令	二 モ ニ ック	特性						命 令 コ ー ド
COMPRESSION CALL CONVERT TO BINARY (32) CONVERT TO BINARY (64) CONVERT TO DECIMAL (32) CONVERT TO DECIMAL (64)	CMPSC CVB CVBG CVD CVDG	RRE C RX RXE N RX RXE N	A A A A A	SP SP SP SP SP	II Dd Dd Dd	GM IK IK	ST R ₁ R ₂ B ₂ ST ST B ₂	B263 4F E30E 4E E32E
CONVERT UNICODE TO UTF-8 CONVERT UTF-8 TO UNICODE COPY ACCESS DIVIDE (32<64) DIVIDE (32<64)	CUUTF CUTFU CPYA DR D	RRE C RRE C RRE RR RX	A A A A A	SP SP SP SP SP		IK IK	ST ST R ₁ R ₂ R ₁ R ₂ U ₁ U ₂ B ₂	B2A6 B2A7 B24D 1D 5D
DIVIDE LOGICAL (32<64) DIVIDE LOGICAL (64<128) DIVIDE LOGICAL (32<64) DIVIDE LOGICAL (64<128) DIVIDE SINGLE (64)	DLR DLGR DL DLG DSGR	RRE N3 RRE N RXE N3 RXE N RRE N	A A A A A	SP SP SP SP SP		IK IK IK IK IK	B ₂ B ₂	B997 B987 E397 E387 B90D
DIVIDE SINGLE (64<32) DIVIDE SINGLE (64) DIVIDE SINGLE (64<32) EXCLUSIVE OR (32) EXCLUSIVE OR (64)	DSGFR DSG DSGF XR XGR	RRE N RXE N RXE N RR C RRE C N	A A A A A	SP SP SP C C N		IK IK IK	B ₂ B ₂	B91D E30D E31D 17 B982
EXCLUSIVE OR (32) EXCLUSIVE OR (64) EXCLUSIVE OR (文字) EXCLUSIVE OR (即値) EXECUTE	X XG XC XI EX	RX C RXE C N SS C SI C RX	A A A A A	SP SP SP C SP		EX	ST ST B ₁ B ₂ B ₁	57 E382 D7 97 44
EXTRACT ACCESS EXTRACT PSW INSERT CHARACTER INSERT CHARACTERS UNDER MASK (高) INSERT CHARACTERS UNDER MASK (低)	EAR EPSW IC ICMH ICM	RRE RRE N3 RX RSE C N RS C	A A A A A				U ₂ B ₂ B ₂ B ₂	B24F B98D 43 EB80 BF
INSERT IMMEDIATE (高 個) INSERT IMMEDIATE (高 低) INSERT IMMEDIATE (低 高) INSERT IMMEDIATE (低 低) INSERT PROGRAM MASK	IIHH IIHL IILH IILL IPM	RI N RI N RI N RI N RRE						A50 A51 A52 A53 B222
LOAD (32) LOAD (64) LOAD (64<32) LOAD (32) LOAD (64)	LR LGR LGFR L LG	RR RRE N RRE N RX RXE N	A A A A A				B ₂ B ₂	18 B904 B914 58 E304
LOAD (64<32) LOAD ACCESS MULTIPLE LOAD ADDRESS LOAD ADDRESS EXTENDED LOAD ADDRESS RELATIVE LONG	LGF LAM LA LAE LARL	RXE N RS RX RX RIL N3	A A A A A	SP SP SP SP SP			B ₂ UB U ₁ BP	E314 9A 41 51 C00

図 7-1 (3/8). 一般命令の要約

命令	ニーモニック	特性						命令コード
LOAD AND TEST (32) LOAD AND TEST (64) LOAD AND TEST (64<32) LOAD COMPLEMENT (32) LOAD COMPLEMENT (64)	LTR LTGR LTGFR LCR LCGR	RR C RRE C N RRE C N RR C RRE C N						12 B902 B912 13 B903
LOAD COMPLEMENT (64<32) LOAD HALFWORD (32) LOAD HALFWORD (64) LOAD HALFWORD IMMEDIATE (32) LOAD HALFWORD IMMEDIATE (64)	LCGFR LH LGH LHI LGHI	RRE C N RX RXE N RI RI N		A A			B ₂ B ₂	B913 48 E315 A78 A79
LOAD LOGICAL (64<32) LOAD LOGICAL (64<32) LOAD LOGICAL CHARACTER LOAD LOGICAL HALFWORD LOAD LOGICAL IMMEDIATE (高高)	LLGFR LLGF LLGC LLGH LLIHH	RRE N RXE N RXE N RXE N RI N		A A A			B ₂ B ₂ B ₂	B916 E316 E390 E391 A5C
LOAD LOGICAL IMMEDIATE (高低) LOAD LOGICAL IMMEDIATE (低高) LOAD LOGICAL IMMEDIATE (低低) LOAD LOGICAL THIRTY ONE BITS LOAD LOGICAL THIRTY ONE BITS	LLIHL LLILH LLILL LLGTR LLGT	RI N RI N RI N RRE N RXE N						A5D A5E A5F B917 E317
LOAD MULTIPLE (32) LOAD MULTIPLE (64) LOAD MULTIPLE DISJOINT LOAD MULTIPLE HIGH LOAD NEGATIVE (32)	LM LMG LMD LMH LNR	RS RSE N SS N RSE N RR C		A A A A			B ₂ B ₂ B ₂ B ₄ B ₂	98 EB04 EF EB96 11
LOAD NEGATIVE (64) LOAD NEGATIVE (64<32) LOAD PAIR FROM QUADWORD LOAD POSITIVE (32) LOAD POSITIVE (64)	LNGR LNGFR LPQ LPR LPGR	RRE C N RRE C N RXE N RR C RRE C N			A SP		B ₂	B901 B911 E38F 10 B900
LOAD POSITIVE (64<32) LOAD REVERSED (32) LOAD REVERSED (64) LOAD REVERSED (16) LOAD REVERSED (32)	LPGFR LRVR LRVGR LRVH LRV	RRE C N RRE N3 RRE N RXE N3 RXE N3					B ₂ B ₂	B910 B91F B90F E31F E31E
LOAD REVERSED (64) MONITOR CALL MOVE (文字) MOVE (即値) MOVE INVERSE	LRVG MC MVC MVI MVCIN	RXE N SI SS SI SS		A A A A	SP	MO	B ₂ ST B ₁ B ₂ ST B ₁ B ₂ ST B ₁ B ₂	E30F AF D2 92 E8
MOVE LONG MOVE LONG EXTENDED MOVE LONG UNICODE MOVE NUMERICS MOVE STRING	MVCL MVCLE MVCLU MVN MVST	RR C RS C RSE C E2 SS RRE C		A SP A SP A SP A A SP	II		ST R ₁ R ₂ ST R ₁ R ₃ ST R ₁ R ₂ ST B ₁ B ₂ ST R ₁ R ₂	0E A8 EB8E D1 B255

図 7-1 (4/8). 一般命令の要約

命令	ニーモニック	特性				命令コード		
MOVE WITH OFFSET MOVE ZONES MULTIPLY (64<32) MULTIPLY (64<32) MULTIPLY HALFWORD (32)	MVO MVZ MR M MH	SS SS RR RX RX		A A SP SP A		ST ST	B ₁ B ₂ B ₁ B ₂ B ₂ B ₂	F1 D3 1C 5C 4C
MULTIPLY HALFWORD IMMEDIATE (32) MULTIPLY HALFWORD IMMEDIATE (64) MULTIPLY LOGICAL (64<32) MULTIPLY LOGICAL (128<64) MULTIPLY LOGICAL (64<32)	MHI MGHI MLR MLGR ML	RI RI N RRE N3 RRE N RXE N3					B ₂	A7C A7D B996 B986 E396
MULTIPLY LOGICAL (128<64) MULTIPLY SINGLE (32) MULTIPLY SINGLE (64) MULTIPLY SINGLE (64<32) MULTIPLY SINGLE (32)	MLG MSR MSGR MSGFR MS	RXE N RRE RRE N RRE N RX		A SP A A A			B ₂ B ₂	E386 B252 B90C B91C 71
MULTIPLY SINGLE (64) MULTIPLY SINGLE (64<32) OR (32) OR (64) OR (32)	MSG MSGF OR OGR O	RXE N RXE N RR C RRE C N RX C		A A A A			B ₂ B ₂ B ₂	E30C E31C 16 B981 56
OR (64) OR (文字) OR (即値) OR IMMEDIATE (高 高) OR IMMEDIATE (高 低)	OG OC OI OIHH OIHL	RXE C N SS C SI C RI C N RI C N		A A A		ST ST	B ₁ B ₂ B ₁	E381 D6 96 A58 A59
OR IMMEDIATE (低 高) OR IMMEDIATE (低 低) PACK PACK ASCII PACK UNICODE	OILH OILL PACK PKA PKU	RI C N RI C N SS SS E2 SS E2		A A SP A SP		ST ST ST	B ₁ B ₂ B ₁ B ₂ B ₁ B ₂	A5A A5B F2 E9 E1
PERFORM LOCKED OPERATION ROTATE LEFT SINGLE LOGICAL (32) ROTATE LEFT SINGLE LOGICAL (64) SEARCH STRING SET ACCESS	PLO RLL RLLG SRST SAR	SS C RSE N3 RSE N RRE C RRE		A SP A SP A SP	\$ GM G0	ST	FC R ₂ U ₁	EE EB1D EB1C B25E B24E
SET ADDRESSING MODE (24) SET ADDRESSING MODE (31) SET ADDRESSING MODE (64) SET PROGRAM MASK SHIFT LEFT DOUBLE	SAM24 SAM31 SAM64 SPM SLDA	E N3 E N3 E N RR L RS C		SP SP SP	T T T IF			010C 010D 010E 04 8F
SHIFT LEFT DOUBLE LOGICAL SHIFT LEFT SINGLE (32) SHIFT LEFT SINGLE (64) SHIFT LEFT SINGLE LOGICAL (32) SHIFT LEFT SINGLE LOGICAL (64)	SLDL SLA SLAG SLL SLLG	RS RS C RSE C N RS RSE N		SP	IF IF			8D 8B EB0B 89 EB0D

図 7-1 (5/8). 一般命令の要約

命令	ニーモニック	特性						命令コード		
SHIFT RIGHT DOUBLE SHIFT RIGHT DOUBLE LOGICAL SHIFT RIGHT SINGLE (32) SHIFT RIGHT SINGLE (64) SHIFT RIGHT SINGLE LOGICAL (32)	SRDA SRDL SRA SRAG SRL	RS RS RS RSE RS	C C C N		SP SP			8E 8C 8A EB0A 88		
SHIFT RIGHT SINGLE LOGICAL (64) STORE (32) STORE (64) STORE ACCESS MULTIPLE STORE CHARACTER	SRLG ST STG STAM STC	RSE RX RXE RS RX	N N 		A A A A A		ST ST ST ST ST	B ₂ B ₂ UB B ₂	EB0C 50 E324 9B 42	
STORE CHARACTERS UNDER MASK (高) STORE CHARACTERS UNDER MASK (低) STORE CLOCK STORE CLOCK EXTENDED STORE HALFWORD	STCMH STCM STCK STCKE STH	RSE RS S S RX	N C C 		A A A A A		\$ \$	ST ST ST ST ST	B ₂ B ₂ B ₂ B ₂ B ₂	EB2C BE B205 B278 40
STORE MULTIPLE (32) STORE MULTIPLE (64) STORE MULTIPLE HIGH STORE PAIR TO QUADWORD STORE REVERSED (16)	STM STMG STMH STPQ STRVH	RS RSE RSE RXE RXE	 N N N N3		A A A A A			ST ST ST ST ST	B ₂ B ₂ B ₂ B ₂ B ₂	90 EB24 EB26 E38E E33F
STORE REVERSED (32) STORE REVERSED (64) SUBTRACT (32) SUBTRACT (64) SUBTRACT (64<32)	STRV STRVG SR SGR SGFR	RXE RXE RR RRE RRE	N3 N C C C N		A A		IF IF IF	ST ST	B ₂ B ₂	E33E E32F 1B B909 B919
SUBTRACT (32) SUBTRACT (64) SUBTRACT (64<32) SUBTRACT HALFWORD SUBTRACT LOGICAL (32)	S SG SGF SH SLR	RX RXE RXE RX RR	C C C C C		A A A A		IF IF IF IF		B ₂ B ₂ B ₂ B ₂	5B E309 E319 4B 1F
SUBTRACT LOGICAL (64) SUBTRACT LOGICAL (64<32) SUBTRACT LOGICAL (32) SUBTRACT LOGICAL (64) SUBTRACT LOGICAL (64<32)	SLGR SLGFR SL SLG SLGF	RRE RRE RX RXE RXE	C C C C C N N N N		A A A A				B ₂ B ₂ B ₂	B90B B91B 5F E30B E31B
SUBTRACT LOGICAL WITH BORROW (32) SUBTRACT LOGICAL WITH BORROW (64) SUBTRACT LOGICAL WITH BORROW (32) SUBTRACT LOGICAL WITH BORROW (64) SUPERVISOR CALL	SLBR SLBGR SLB SLBG SVC	RRE RRE RXE RXE RR	C C C C N3 N3 N N		A A		¢		B ₂ B ₂	B999 B989 E399 E389 0A
TEST ADDRESSING MODE TEST AND SET TEST UNDER MASK TEST UNDER MASK (高 高) TEST UNDER MASK (高 低)	TAM TS TM TMHH TMHL	E S SI RI RI	C C C C C N3 N N N N		A A		\$	ST	B ₁ B ₂	010B 93 91 A72 A73

図 7-1 (6/8). 一般命令の要約

命令	二 モ ニ ツ ク	特性						命 令 コ ド	
TEST UNDER MASK (低 高) TEST UNDER MASK (低 低) TEST UNDER MASK HIGH TEST UNDER MASK LOW TRANSLATE	TMLH TMLL TMH TML TR	RI RI RI RI SS	C C C C S	N N C C S	A		ST B ₁ B ₂	A70 A71 A70 A71 DC	
TRANSLATE AND TEST TRANSLATE EXTENDED TRANSLATE ONE TO ONE TRANSLATE ONE TO TWO TRANSLATE TWO TO ONE	TRT TRE TROO TROT TRTO	SS RRE RRE RRE RRE	C C C C C	E2 C E2 E2 E2	A A A A A	SP SP SP SP SP	GM ST GM GM GM	B ₁ B ₂ R ₁ R ₂ RM R ₂ RM R ₂ RM R ₂	DD B2A5 B993 B992 B991
TRANSLATE TWO TO TWO UNPACK UNPACK ASCII UNPACK UNICODE UPDATE TREE	TRTT UNPK UNPKA UNPKU UPT	RRE SS SS SS E	C C C C C	E2 C E2 E2 C	A A A A A	SP C SP SP SP	GM ST ST ST GM	ST B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ I4	B990 F3 EA E2 0102

説明:

- ¢ 逐次化とチェックポイント同期化を起こします。
- ¢¹ M₁ フィールドがすべて 1 で R₂ フィールドがすべて 0 のとき、逐次化とチェックポイント同期化を起こします。
- \$ 逐次化を起こします。
- A 論理アドレスについてのアクセス例外。
- A¹ アクセス例外。すべてのアクセス例外が起きるわけではありません。詳細については、命令の説明を参照してください。
- AI 命令アドレスについてのアクセス例外。
- B PER ブランチ・イベント。
- B₁ アクセス・レジスター・モードでは、B₁ フィールドはアクセス・レジスターを指定します。
- B₂ アクセス・レジスター・モードでは、B₂ フィールドはアクセス・レジスターを指定します。
- BP PSW ビット 16 および 17 の値が 2 進数 01 のとき、B₂ フィールドはアクセス・レジスターを指定します。
- C 条件コードが設定されます。
- Dd 10 進オペランドのデータ例外。
- E 命令フォーマット。
- E2 拡張変換機能 2。
- EX 実行例外。
- FC アクセス・レジスターの指定は、命令の機能コードに応じて決まります。
- GO 命令の実行時に、汎用レジスター 0 が暗黙的に使用されます。
- GM 命令の実行時に、複数の汎用レジスターが暗黙的に使用されます。
COMPARE AND FORM CODEWORD では、汎用レジスター 1、2、および 3。
COMPARE UNTIL SUBSTRING EQUAL および PERFORM LOCKED OPERATION では、汎用レジスター 0 および 1。
COMPRESSION CALL、TRANSLATE ONE TO ONE、TRANSLATE ONE TO TWO、TRANSLATE TWO TO ONE、および TRANSLATE TWO TO TWO では、汎用レジスター 0 および 1。
TRANSLATE AND TEST では、汎用レジスター 1 および 2。
UPDATE TREE では、汎用レジスター 0-5。
- I1 アクセス・レジスター・モードでは、アクセス・レジスター 1 が暗黙的に指定されます。
- I2 アクセス・レジスター・モードでは、アクセス・レジスター 2 が暗黙的に指定されます。
- I4 アクセス・レジスター・モードでは、アクセス・レジスター 4 が暗黙的に指定されます。
- IF 固定小数点オーバーフロー例外。
- II 割り込み可能命令。
- IK 固定小数点除算例外。
- L 新しい条件コードがロードされます。

図 7-1 (7/8). 一般命令の要約

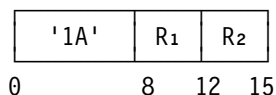
説明 (続き):

MO	モニター・イベント。
N	z/Architecture の新規の命令で、ESA/390 にはないものです。
N3	z/Architecture の新規の命令で、ESA/390 にも追加されたものです。
R1	アクセス・レジスター・モードでは、R1 フィールドはアクセス・レジスターを指定します。
R2	アクセス・レジスター・モードでは、R2 フィールドはアクセス・レジスターを指定します。
R3	アクセス・レジスター・モードでは、R3 フィールドはアクセス・レジスターを指定します。
RI	RI 命令フォーマット。
RIE	RIE 命令フォーマット。
RIL	RIL 命令フォーマット。
RR	RR 命令フォーマット。
RRE	RRE 命令フォーマット。
RS	RS 命令フォーマット。
RSE	RSE 命令フォーマット。
RSI	RSI 命令フォーマット。
RX	RX 命令フォーマット。
RXE	RXE 命令フォーマット。
S	S 命令フォーマット。
SI	SI 命令フォーマット。
SP	指定例外。
SS	SS 命令フォーマット。
ST	PER 記憶域変更イベント。
T	トレース例外 (トレース・テーブル、アドレッシング、および低アドレス保護を含む)。
U1	R1 フィールドは、無条件にアクセス・レジスターを指定します。
U2	R2 フィールドは、無条件にアクセス・レジスターを指定します。
UB	R1 および R3 フィールドは無条件にアクセス・レジスターを指定し、B2 フィールドはアクセス・レジスター・モードの場合にアクセス・レジスターを指定します。

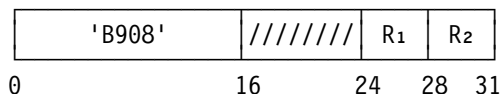
図 7-1 (8/8). 一般命令の要約

ADD

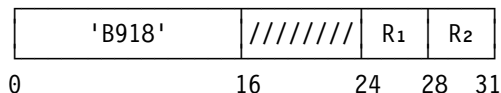
AR R₁,R₂ [RR]



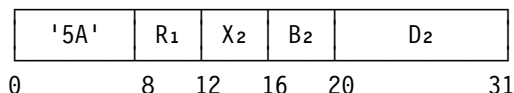
AGR R₁,R₂ [RRE]



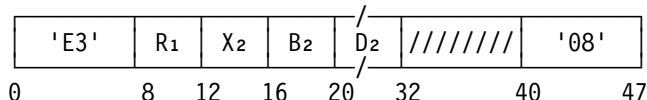
AGFR R₁,R₂ [RRE]



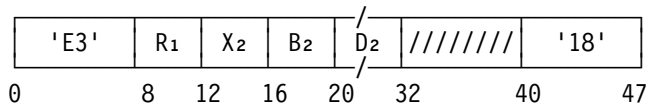
A R₁,D₂(X₂,B₂) [RX]



AG R₁,D₂(X₂,B₂) [RXE]



AGF R₁,D₂(X₂,B₂) [RXE]



第 2 オペランドが第 1 オペランドに加算され、和が第 1 オペランド位置に入れられます。ADD (AR、A) では、オペランドと和は 32 ビットの符号付き 2 進整数として扱われます。ADD (AGR、AG) では、オペランドと和は 64 ビットの符号付き 2 進整数として扱われます。ADD (AGFR、AGF) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、第 1 オペランドと和は 64 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がりを許し、符号ビット位置からの繰り上がりは無視する方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定

小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

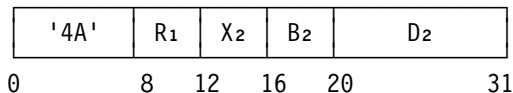
- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

- アクセス例外 (取り出し: A、AG、および AGF のオペランド 2 のみ)。
- 固定小数点オーバーフロー例外

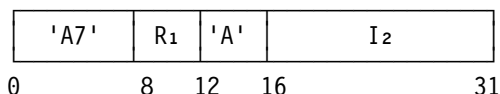
ADD HALFWORD

AH R₁,D₂(X₂,B₂) [RX]

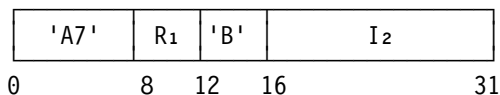


ADD HALFWORD IMMEDIATE

AHI R₁,I₂ [RI]



AGHI R₁,I₂ [RI]



第 2 オペランドが第 1 オペランドに加算され、和が第 1 オペランド位置に入れられます。第 2 オペランドは、長さが 2 バイトで、16 ビットの符号付き 2 進整数として扱われます。ADD HALFWORD および ADD HALFWORD IMMEDIATE (AHI) では、第 1 オペランドと和は 32 ビットの符号付き 2 進整数として扱われます。ADD HALFWORD IMMEDIATE (AGHI) では、第 1 オペランドと和は 64 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がりを許し、符号ビット位置からの繰り上がりは無視す

る方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

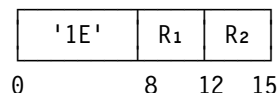
プログラム例外:

- アクセス例外 (取り出し: AH のオペランド 2 のみ)
- 固定小数点オーバーフロー例外

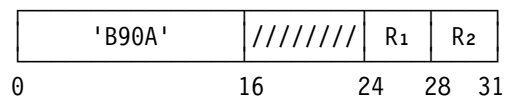
プログラミング上の注意: ADD HALFWORD 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。

ADD LOGICAL

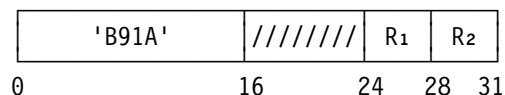
ALR R₁,R₂ [RR]



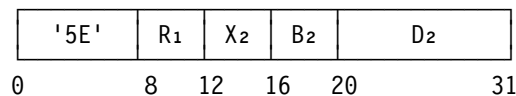
ALGR R₁,R₂ [RRE]



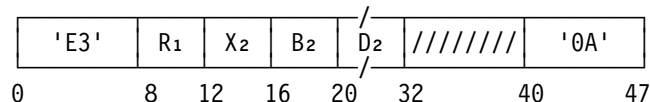
ALGFR R₁,R₂ [RRE]



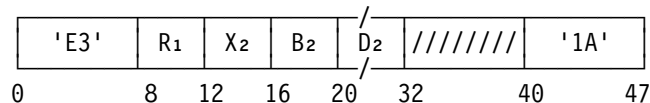
AL R₁,D₂(X₂,B₂) [RX]



ALG R₁,D₂(X₂,B₂) [RXE]



ALGF R₁,D₂(X₂,B₂) [RXE]



第 2 オペランドが第 1 オペランドに加算され、和が第 1 オペランド位置に入れられます。ADD LOGICAL (ALR, AL) では、オペランドと和は 32 ビットの符号なし 2 進整数として扱われます。ADD LOGICAL (ALGR, ALG) では、オペランドと和は 64 ビットの符号なし 2 進整数として扱われます。ADD LOGICAL (ALGFR, ALGF) では、第 2 オペランドは 32 ビットの符号なし 2 進整数として扱われ、第 1 オペランドと和は 64 ビットの符号なし 2 進整数として扱われます。

結果の条件コード:

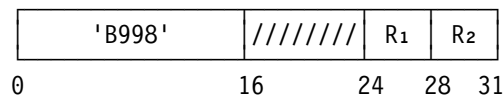
- 0 結果は 0 で、繰り上がりはありません。
- 1 結果は 0 でなく、繰り上がりはありません。
- 2 結果は 0 で、繰り上がりがあります。
- 3 結果は 0 でなく、繰り上がりがあります。

プログラム例外:

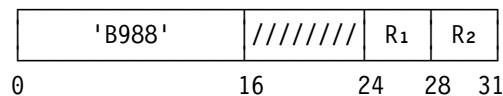
- アクセス例外 (取り出し: AL, ALG, および ALGF のオペランド 2 のみ)

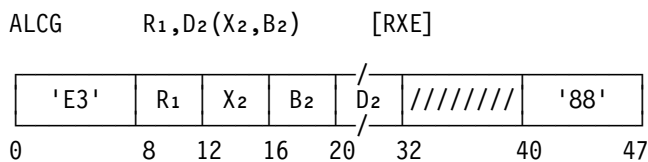
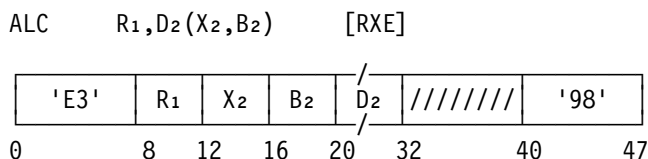
ADD LOGICAL WITH CARRY

ALCR R₁,R₂ [RRE]



ALCGR R₁,R₂ [RRE]





第 2 オペランドと繰り上がりが第 1 オペランドに加算され、その和が第 1 オペランド位置に入れます。ADD LOGICAL WITH CARRY (ALCR、ALC) では、オペランド、繰り上がり、および和は、32 ビットの符号なし 2 進整数として扱われます。ADD LOGICAL WITH CARRY (ALCGR、ALCG) では、これらは 64 ビットの符号なし 2 進整数として扱われます。

結果の条件コード:

- 0 結果は 0 で、繰り上がりはありません。
- 1 結果は 0 でなく、繰り上がりはありません。
- 2 結果は 0 で、繰り上がりがあります。
- 3 結果は 0 でなく、繰り上がりがあります。

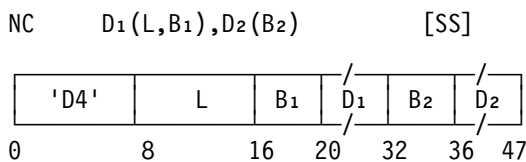
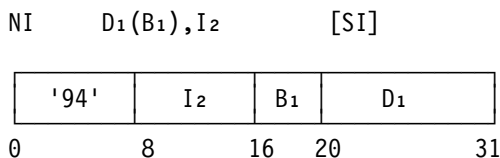
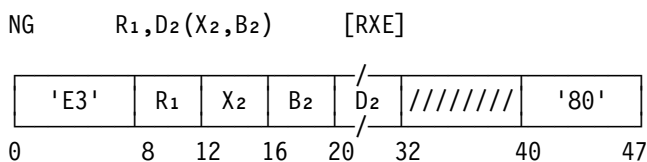
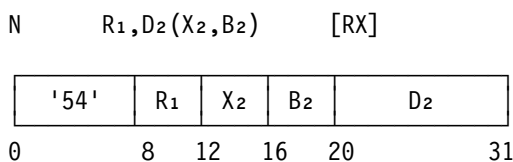
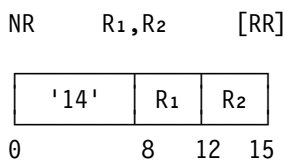
プログラム例外:

- アクセス例外 (取り出し: ALC および ALCG のオペランド 2 のみ)

プログラミング上の注意:

1. 繰り上がりがあれば、現 PSW のビット 18 に値 1 が入ります。ビット 18 は、PSW の 2 ビット条件コードの左端のビットです。ADD LOGICAL 命令または ADD LOGICAL WITH CARRY 命令の実行の結果、結果のビット位置 0 からの繰り上がりが発生されると、ビット 18 が 1 に設定されます。
2. モデルによっては、ADD および ADD LOGICALの方が ADD LOGICAL WITH CARRY よりパフォーマンスがよくなる場合があります。

AND



第 1 オペランドと第 2 オペランドの論理積 (AND) が、第 1 オペランド位置に入れます。

連結子 AND は、両方のオペランドにビット単位で適用されます。両方のオペランドの対応するビット位置に 1 が含まれている場合は結果のビット位置の内容が 1 に設定され、それ以外の場合は結果ビットが 0 に設定されます。

AND (NC) では、各オペランドが左から右へと処理されます。オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理され、必要なオペランド・バイトを取り出した直後に各結果バイトが格納されるときと同様の方法で、結果が取得されます。

AND (NI) では、第 1 オペランドの長さは 1 バイトであり、1 バイトのみが格納されます。

オペランドは、AND (NR、N) では 32 ビット、AND (NGR、NG) では 64 ビットです。

結果の条件コード:

- 0 結果は 0 です。
- 1 結果は 0 ではありません。
- 2 --
- 3 --

プログラム例外:

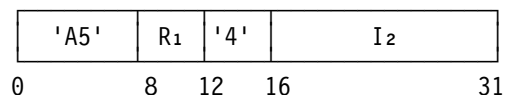
- アクセス例外 (取り出し: オペランド 2、N、NG、および NC。取り出しおよび格納: オペランド 1、NI および NC)

プログラミング上の注意:

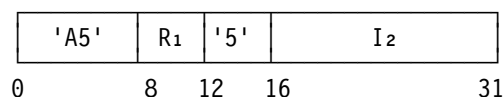
1. AND 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. AND 命令は、ビットを 0 に設定するために使用できます。
3. AND (NI) および AND (NC) の第 1 オペランドへのアクセスは、第 1 オペランドの 1 バイトを記憶域から取り出す操作と、更新済みの値を格納する後続操作から成っています。しかし、ある 1 つのバイトについて見ると、必ずしも取り出しアクセスの直後に格納アクセスが行われるとは限りません。このため、他の CPU またはチャネル・プログラムが更新する可能性のある記憶位置を AND 命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、A-43ページの『マルチプログラミングとマルチプロセッシングの例』で OR (OI) について示してあります。

AND IMMEDIATE

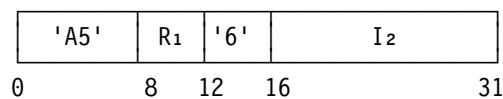
NIHH R₁,I₂ [RI]



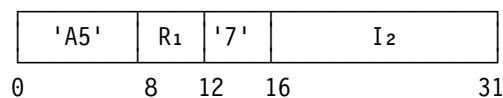
NIHL R₁,I₂ [RI]



NILH R₁,I₂ [RI]



NILL R₁,I₂ [RI]



第 2 オペランドと第 1 オペランドのビットとの論理積 (AND) が求められ、その結果が第 1 オペランドの該当ビットと置き換わります。第 1 オペランドのその他のビットは変更されません。

命令別に見ると、第 2 オペランドとの論理積演算後に置換される第 1 オペランドのビットは次のとおりです。

命令	論理積演算後に置換されるビット
NIHH	0-15
NIHL	16-31
NILH	32-47
NILL	48-63

連結子 AND は、両方のオペランドにビット単位で適用されます。両方のオペランドの対応するビットが 1 であれば、結果のビット位置の内容は 1 に設定され、それ以外の場合は、結果ビットは 0 に設定されます。

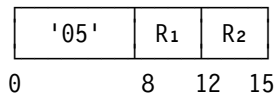
結果の条件コード:

- 0 16 ビットの結果で、0 です。
- 1 16 ビットの結果で、0 ではありません。
- 2 --
- 3 --

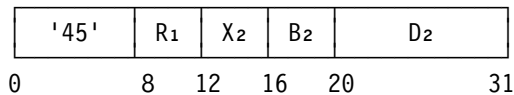
プログラム例外: なし。

BRANCH AND LINK

BALR R₁,R₂ [RR]

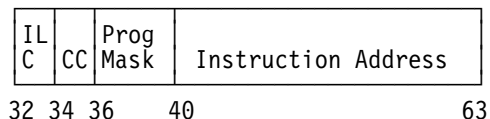


BAL R₁,D₂(X₂,B₂) [RX]



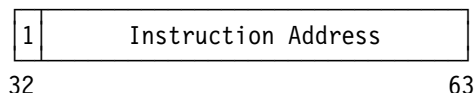
現 PSW からの情報 (更新済み命令アドレスを含む) が、第 1 オペランド位置にリンク情報として保管されます。その後、PSW の中の命令アドレスがブランチ・アドレスで置き換えられます。

24 ビット・アドレッシング・モードでは、リンク情報は、命令長コード (ILC)、条件コード (CC)、プログラム・マスク・ビット、および更新済み命令アドレスの右端の 24 ビットで構成され、次に示すフォーマットで、第 1 オペランド位置のビット位置 32-63 に入れられます。



命令長コードは 1 または 2 です。

31 ビット・アドレッシング・モードでは、リンク情報は、PSW のビット 32、基本アドレッシング・モード・ビット (常に 1)、および、更新済み命令アドレスの右端の 31 ビットで構成され、次のフォーマットで、第 1 オペランド位置のビット位置 32-63 に入れられます。



24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置のビット 0-31 は変更されません。

64 ビット・アドレッシング・モードでは、リンク情報は、更新済み命令アドレスから成り、第 1 オペランド位置のビット位置 0-63 に入れられます。

RX フォーマットでは、第 2 オペランドのアドレスがブランチ・アドレスとして使用されます。RR フォーマットでは、汎用レジスター R₂ の内容を使用してブランチ・アドレスが生成されます。ただし、R₂ フィールドが 0 の場合は、操作はブランチなしで実行されます。ブランチ・アドレスは、汎用レジスター R₁ の変更前に計算されます。

条件コード: コードは変更されません。

プログラム例外:

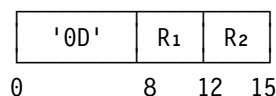
- トレース例外 (R₂ が 0 でないとき。BALR のみ)

プログラミング上の注意:

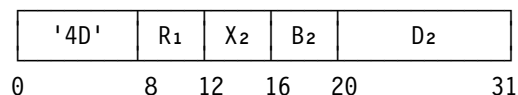
1. BRANCH AND LINK 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. RR フォーマットの R₂ フィールドが 0 でない場合は、リンク情報はブランチなしでロードされます。
3. BRANCH AND LINK 命令 (BAL および BALR) は、互換性を確保するために提供されているものです。可能な限り BRANCH AND SAVE 命令 (BAS および BASR)、BRANCH RELATIVE AND SAVE、または BRANCH RELATIVE AND SAVE LONG を使用し、BRANCH AND LINK は避けてください。BRANCH AND LINK は、24 ビット・アドレッシング・モードのときにリンク・レジスターのビット位置 32-39 に 0 以外の情報を入れるので、それが原因で問題が起きることがあります。さらに、24 ビット・アドレッシング・モードでは BRANCH AND LINK は他の命令より速度が遅くなる場合があります。なぜなら、BRANCH AND LINK では、ILC、条件コード、およびプログラム・マスクを作成してリンク・レジスターのビット位置 32-39 に入れる必要があるからです。
4. 条件コードおよびプログラム・マスク情報は、INSERT PROGRAM MASK 命令を使用すれば、どのアドレッシング・モードでも取得できます。ただし、24 ビット・アドレッシング・モードの場合のみはリンク情報の左端バイトに示されます。

BRANCH AND SAVE

BASR R₁,R₂ [RR]



BAS R₁, D₂ (X₂, B₂) [RX]



現 PSW からの情報 (更新済み命令アドレスを含む) が、第 1 オペランド位置にリンク情報として保管されま
す。その後、PSW 中の命令アドレスがブランチ・アド
レスで置き換えられます。

24 ビットまたは 31 ビットのアドレッシング・モードで
は、リンク情報は PSW のビット 32 および 97-127
で、基本アドレッシング・モード・ビットと、更新済み
命令アドレスの右端 31 ビットから成っています。この
リンク情報は、第 1 オペランド位置のビット位置 32 お
よび 33-63 にそれぞれ入れられ、第 1 オペランド位置
のビット 0-31 は変更されません。

64 ビット・アドレッシング・モードでは、リンク情報
は更新済み命令アドレスから成り、第 1 オペランド位置
のビット位置 0-63 に入れられます。

RX フォーマットでは、第 2 オペランドのアドレスがブ
ランチ・アドレスとして使用されます。RR フォーマ
ットでは、汎用レジスター R₂ の内容を使用してブランチ
・アドレスが生成されます。ただし、R₂ フィールド
が 0 の場合は、操作はブランチなしで実行されます。ブ
ランチ・アドレスは、汎用レジスター R₁ の変更前に計
算されます。

条件コード: コードは変更されません。

プログラム例外:

- トレース例外 (R₂ フィールドが 0 でないとき。
BASR のみ)

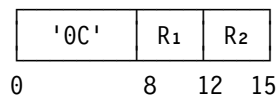
プログラミング上の注意:

1. BRANCH AND SAVE 命令の使用例は、付録A、
『数の表現と命令の使用例』に示されています。
2. BRANCH AND SAVE 命令 (BAS および BASR)
は、呼び出し側と同じアドレッシング・モードであ
ることが判明しているプログラムへのリンケージに
使用するためのものです。BRANCH AND LINK
命令 (BAL および BALR) の代わりに、この命令
を使用してください。この命令と、その他のリンケ
ージ命令の詳しい説明については、『リンケージ・
スタックなしのサブルーチン・リンケージ』という
セクションの 5-11 ページおよび 5-17 ページの『

プログラミング上の注意』を参照してください。
BRANCH AND SAVE 命令の利点の説明につい
て、『BRANCH AND LINK』の項の『プログラ
ミング上の注意』も参照してください。

BRANCH AND SAVE AND SET MODE

BASSM R₁, R₂ [RR]



現 PSW からの情報 (更新済み命令アドレスを含む)
が、第 1 オペランド位置にリンク情報として保管されま
す。その後、R₂ フィールドが 0 でない場合は、PSW
内のアドレッシング・モード・ビットと命令アドレス
が、第 2 オペランドの指定内容で置き換えられます。

24 ビットまたは 31 ビットのアドレッシング・モードで
は、リンク情報は PSW のビット 32 および 97-127
で、基本アドレッシング・モード・ビットと、更新済み
命令アドレスの右端 31 ビットから成っています。この
リンク情報は、第 1 オペランド位置のビット位置 32 お
よび 33-63 にそれぞれ入れられ、第 1 オペランド位置
のビット 0-31 は変更されません。64 ビット・アドレ
ッシング・モードでは、リンク情報は PSW のビット
64-126 の右端に 1 が付加されたもので、第 1 オペラ
ンド位置のビット位置 0-63 に入れられます。

汎用レジスター R₂ の内容は、新規アドレッシング・モ
ードおよびブランチ・アドレスを指定します。ただし、
R₂ フィールドが 0 の場合は、操作はブランチなしで実
行され、アドレッシング・モード・ビットも設定されま
せん。

汎用レジスター R₂ の内容が使用され、このレジスタ
ーのビット 63 が 0 である場合は、現 PSW のビット 31
(拡張アドレッシング・モード・ビット) が 0 に設定さ
れ、レジスターのビット 32 が新しい基本アドレッシ
ング・モードを表して PSW のビット 32 と置き換わり
ます。そして、新しいアドレッシング・モードの制御下
で、レジスターの内容からブランチ・アドレスが生成さ
れます。このブランチ・アドレスが、PSW 内の命令ア
ドレスと置き換わります。

汎用レジスター R₂ の内容が使用され、このレジスタ
ーのビット 63 が 1 の場合は、次のことが起きます。現
PSW のビット 31 および 32 が 1 に設定されます。新

しい拡張アドレッシング・モードの制御下で、レジスタの内容からブランチ・アドレスが生成されますが、ただし、レジスタのビット 63 は 0 として扱われます。このブランチ・アドレスが、PSW 内の命令アドレスと置き換わります。レジスタの 63 は 1 のままです。ただし、R₂ が R₁ と同じである場合は、指定された汎用レジスタ内の結果は、R₁ レジスタについて指定されているものと同じになります。

PSW の新しい値は、汎用レジスタ R₁ の変更前に計算されます。

条件コード: コードは変更されません。

プログラム例外:

- トレース例外 (R₂ フィールドが 0 でないとき)

プログラミング上の注意:

1. BRANCH AND SAVE AND SET MODE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. BRANCH AND SAVE AND SET MODE は、呼び出し側のモードとは異なるアドレッシング・モードで動作する可能性のあるサブルーチンに対する基本呼び出し命令として使用します。この命令と、その他のリンケージ命令の詳しい説明については、『リンケージ・スタックなしのサブルーチン・リンケージ』というセクションの 5-11 ページおよび 5-17 ページの『プログラミング上の注意』を参照してください。
3. 24 ビットまたは 31 ビットの旧プログラムでも、汎用レジスタ R₂ のビット 0-31 がすべて 0 であれば、まったく変更を加えずに、BRANCH AND SAVE AND SET MODE を使用して新しい 64 ビット・プログラムを呼び出すことができます。旧プログラムは、汎用レジスタ R₂ のビット位置 32-63 に 4 バイトのアドレス定数 (プログラムの外部から供給) をロードできます。この場合、レジスタのビット 63 (記憶域内の定数のビット 31) は 1 の場合も 1 でない場合もあります。BRANCH AND SAVE AND SET MODE 命令の実行によってアドレッシング・モードが 64 ビットに変更されない場合は (あるいは変更される場合であっても)、呼び出し先プログラムから SET ADDRESSING MODE (SAM64) 命令を発行することによって、64 ビット・モードを設定することができます。
4. 5-11 ページ (『単純ブランチ命令』の項) のプログラミング上の注意を参照してください。

BRANCH AND SET MODE

BSM R₁,R₂ [RR]

'0B'	R ₁	R ₂
0	8	12 15

24 ビットまたは 31 ビットのアドレッシング・モードでは、現 PSW のビット 32 (基本アドレッシング・モード・ビット) が、第 1 オペランドのビット位置 32 に挿入され、オペランドのビット 0-31 および 33-63 は変更されません。64 ビット・アドレッシング・モードでは、第 1 オペランドのビット位置 63 に 1 が挿入され、オペランドのビット 0-62 は変更されません。その後、PSW 内のアドレッシング・モード・ビットおよび命令アドレスが第 2 オペランドの指定内容で置き換えられます。関連した R フィールドが 0 の場合は、オペランドに関連したアクションは実行されません。

汎用レジスタ R₂ の内容は、新規アドレッシング・モードおよびブランチ・アドレスを指定します。ただし、R₂ フィールドが 0 の場合は、操作はブランチなしで実行され、アドレッシング・モード・ビットも設定されません。

汎用レジスタ R₂ の内容が使用され、このレジスタのビット 63 が 0 である場合は、現 PSW のビット 31 (拡張アドレッシング・モード・ビット) が 0 に設定され、レジスタのビット 32 が新しい基本アドレッシング・モードを表して PSW のビット 32 と置き換わります。そして、新しいアドレッシング・モードの制御下で、レジスタの内容からブランチ・アドレスが生成されます。このブランチ・アドレスで、PSW 内の命令アドレスが置き換えられます。

汎用レジスタ R₂ の内容が使用され、このレジスタのビット 63 が 1 の場合は、次のことが起きます。現 PSW のビット 31 および 32 が 1 に設定されます。新しい拡張アドレッシング・モードの制御下で、レジスタの内容からブランチ・アドレスが生成されますが、ただし、レジスタのビット 63 は 0 として扱われます。このブランチ・アドレスが、PSW 内の命令アドレスと置き換わります。レジスタの 63 は 1 のままです。ただし、R₂ が R₁ と同じである場合は、指定された汎用レジスタ内の結果は、R₁ レジスタについて指定されているものと同じになります。

PSW の新しい値は、汎用レジスタ R₁ の変更前に計算されます。

条件コード: コードは変更されません。

プログラム例外:

- トレース例外

プログラミング上の注意:

1. BRANCH AND SET MODE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されていません。
2. BRANCH AND SET MODE の R₁ フィールドが 0 の場合、これは BRANCH AND SAVE AND SET MODE を用いて開始されるプログラムでの、標準の戻り命令となります。また、BRANCH AND SAVE、BRANCH RELATIVE AND SAVE、または BRANCH RELATIVE AND SAVE LONG を使用して 24 ビットまたは 31 ビットのアドレッシング・モードで開始されるプログラムでの戻り命令としても使用できます。
BRANCH AND SET MODE の R₁ フィールドが 0 でない場合、これは「グルー・モジュール」の中で使用され、24 ビットの旧プログラムと 31 ビット・アドレッシング・モードで実行されるそれより新しいプログラム、あるいは、24 ビットまたは 31 ビットの旧プログラムと 64 ビット・アドレッシング・モードで実行される新しいプログラムを接続する役割を果たします。この命令と、その他のリンケージ命令の詳しい説明については、『リンケージ・スタックなしのサブルーチン・リンケージ』というセクションの 5-11 ページおよび 5-17 ページの『プログラミング上の注意』を参照してください。

BRANCH ON CONDITION

BCR M₁,R₂ [RR]

'07'	M ₁	R ₂
0	8	12 15

BC M₁,D₂(X₂,B₂) [RX]

'47'	M ₁	X ₂	B ₂	D ₂
0	8	12	16 20	31

条件コードが M₁ で指定された値の 1 つになると、現 PSW 内の命令アドレスがブランチ・アドレスで置き換えられます。それ以外の場合は、通常の命令順序に従って、更新済み命令アドレスを用いて処理が行われます。

RX フォーマットでは、第 2 オペランドのアドレスがブランチ・アドレスとして使用されます。RR フォーマットでは、汎用レジスター R₂ の内容を使用してブランチ・アドレスが生成されます。ただし、R₂ フィールドが 0 の場合は、操作はブランチなしで実行されます。

M₁ フィールドは、4 ビットのマスクとして使用されます。4 つの条件コード (0、1、2、および 3) は、次のように左から右への順で、マスクの 4 つのビットに対応します。

条件コード	命令内のマスクのビット番号	マスク位置の値
0	8	8
1	9	4
2	10	2
3	11	1

現行の条件コードを使用して、対応するマスク・ビットが選択されます。条件コードに基づいて選択されたマスク・ビットが 1 であれば、ブランチは成功です。選択されたマスク・ビットが 0 であれば、次の順番に当たる命令を使用して、通常の命令順序に従って処理が進められます。

BRANCH ON CONDITION (BCR) の M₁ フィールドがすべて 1 で、R₂ フィールドがすべて 0 の場合は、逐次化とチェックポイント同期化機能が実行されます。

条件コード: コードは変更されません。

プログラム例外: なし。

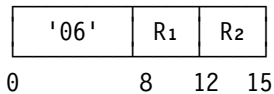
プログラミング上の注意:

1. BRANCH ON CONDITION 命令の使用例は、付録A、『数の表現と命令の使用例』に示されていません。
2. 複数の条件に基づいてブランチを発生させる場合は、関連の条件コードをそのマスク位置の値の和の形でマスクに指定します。例えば、マスク 12 は、条件コードが 0 または 1 のときにブランチを実行することを示します。
3. 4 個のマスク・ビットがすべて 0 の場合、または RR フォーマットの R₂ フィールドに 0 が含まれている場合は、ブランチ命令はノーオペレーションと同じことです。4 個のマスク・ビットがすべて 1 の場合、つまりマスク値が 15 の場合は、RR フォーマットの R₂ フィールドが 0 である場合を除き、無条件にブランチが実行されます。

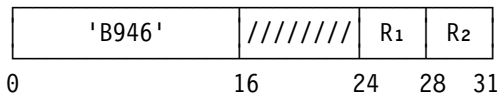
4. BCR 15,0 (つまり、16 進数 07F0 の値を持つ命令) を実行すると、著しい性能低下を招くことがあります。最適なパフォーマンスを確保するためには、逐次化またはチェックポイント同期化機能が本当に必要な場合以外は、プログラムで BCR 15,0 を使用することは避けてください。
5. ブランチ・アドレスの指定における RR フォーマットと RX フォーマットの関係は、オペランド・アドレスの指定の場合とは異なるので、注意してください。RX フォーマットのブランチ命令では、ブランチ・アドレスは、X₂、B₂、および D₂ で指定されるアドレスです。RR フォーマットでは、ブランチ・アドレスは R₂ で指定されるレジスタに含まれています。オペランドの場合は、X₂、B₂、および D₂ で指定されるアドレスはオペランド・アドレスですが、R₂ で指定されるレジスタに含まれているのはオペランドであって、オペランド・アドレスではありません。

BRANCH ON COUNT

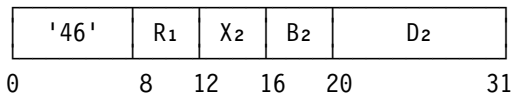
BCTR R₁,R₂ [RR]



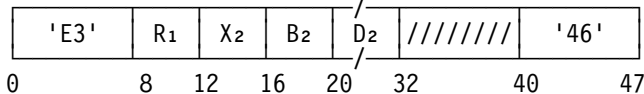
BCTGR R₁,R₂ [RRE]



BCT R₁,D₂(X₂,B₂) [RX]



BCTG R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドから 1 が減算され、結果が第 1 オペランド位置に入れられます。BRANCH AND COUNT

(BCT、BCTR) では、第 1 オペランドおよび結果は、32 ビットの 2 進整数として扱われ、オーバーフローは無視されます。BRANCH AND COUNT (BCTG、BCTGR) では、第 1 オペランドと結果は、64 ビットの 2 進整数として扱われ、オーバーフローは無視されません。結果が 0 であれば、更新済み命令アドレスを使用して通常の命令順序に従って処理が進められます。結果が 0 でない場合は、現 PSW 中の命令アドレスがブランチ・アドレスで置き換えられます。

RX または RXE フォーマットでは、第 2 オペランドのアドレスがブランチ・アドレスとして使用されます。

RR または RRE フォーマットでは、汎用レジスタ R₂ の内容を使用してブランチ・アドレスが生成されません。ただし、R₂ フィールドが 0 の場合は、操作はブランチなしで実行されます。ブランチ・アドレスは、汎用レジスタ R₁ の変更前に生成されます。

条件コード: コードは変更されません。

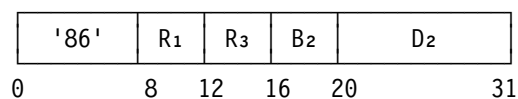
プログラム例外: なし。

プログラミング上の注意:

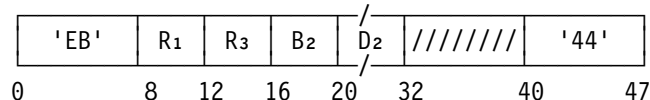
- BRANCH ON COUNT 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
- 2 進数減算の結果は符号付きの場合でも符号なしの場合でも同じになるため、第 1 オペランドおよび結果は、符号付き 2 進整数または符号なし 2 進整数のどちらと見なすこともできます。
- 初期カウントが 1 の場合、結果は 0 で、ブランチは実行されません。初期カウントが 0 の場合、結果は -1 で、ブランチが実行されます。初期カウントが -1 の場合、結果は -2 で、ブランチが実行されません。以後、同様の処理が続きます。ループ内では、結果が再び 0 に戻るまでは、命令が実行されるたびにブランチも実行されます。数値範囲上の理由で次のようになる点に注意してください。BCT または BCTR の場合は、初期カウントが -2^{31} なら、結果は正の値 $2^{31} - 1$ になります。BCTG または BCTGR の場合は、初期カウントが -2^{63} なら、結果は正の値 $2^{63} - 1$ になります。
- RR または RRE フォーマットの R₂ フィールドに 0 が含まれている場合は、カウントはブランチなしで行われます。

BRANCH ON INDEX HIGH

BXH $R_1, R_3, D_2(B_2)$ [RS]

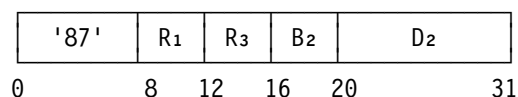


BXHG $R_1, R_3, D_2(B_2)$ [RSE]

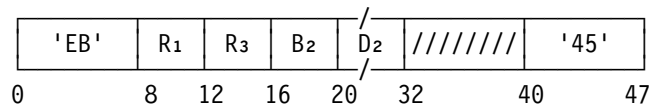


BRANCH ON INDEX LOW OR EQUAL

BXLE $R_1, R_3, D_2(B_2)$ [RS]



BXLEG $R_1, R_3, D_2(B_2)$ [RSE]



第 1 オペランドに増分が加算され、その和が比較値と比較されます。比較の結果に基づいて、ブランチを実行するかどうかが決まります。その後、和が第 1 オペランド位置に入れられます。第 2 オペランドのアドレスは、ブランチ・アドレスとして使用されます。R₃ フィールドは、増分および比較値が入っているレジスターを指示します。

BRANCH ON INDEX HIGH では、和の方が大きい場合は、現 PSW 中の命令アドレスがブランチ・アドレスで置き換えられます。和が比較値より小さいかまたは等しい場合は、更新済み命令アドレスを使用して通常の命令順序に従って処理が進められます。

BRANCH ON INDEX LOW OR EQUAL では、和が比較値より小さいかまたは等しい場合は、現 PSW 中の命令アドレスがブランチ・アドレスで置き換えられます。和の方が大きい場合は、更新済み命令アドレスを使用して、通常の命令順序に従って処理が進められます。

R₃ フィールドが偶数なら、一对のレジスターを示します。このペアの偶数レジスターの内容が増分として使用

され、奇数レジスターの内容が比較値として使用されます。R₃ フィールドが奇数なら、単一レジスターを示し、このレジスターの内容が増分と比較値の両方として使用されます。

加算および比較が目的の場合、すべてのオペランドと結果は、BXH および BXLE の場合は 32 ビットの符号付き 2 進整数として扱われ、BXHG および BXLEG の場合は 64 ビットの符号付き 2 進整数として扱われます。加算により生じたオーバーフローは無視されます。

比較値レジスターが第 1 オペランド位置にも入ることが指定されている場合でも、比較値として使用されるのは、比較値レジスターの元の内容です。ブランチ・アドレスは、汎用レジスター R₁ の変更前に生成されます。

ブランチが実行されるかどうかに関係なく、和は第 1 オペランド位置に入れられます。

条件コード: コードは変更されません。

プログラム例外: なし。

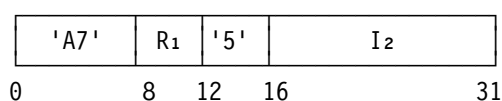
プログラミング上の注意:

- BRANCH ON INDEX HIGH 命令および BRANCH ON INDEX LOW OR EQUAL 命令の使用例が幾つか、付録A、『数の表現と命令の使用例』に示されています。
- これらの命令の名前に含まれている「INDEX」(指標)という語は、命令の主目的の 1 つが指標値に増分を加えてテストすることであることを示しています。増分は、符号付き 2 進整数であり、整数サイズの制限に従った任意の指定量で、汎用レジスター R₁ 内の値を増加または減少させるために使用できます。
- 31 ビット・アドレッシング・モードでは、記憶機構内のデータ域が 31 ビット・アドレス・スペースの右端にあり、BRANCH ON INDEX HIGH (BXH) または BRANCH ON INDEX LOW OR EQUAL (BXLE) 命令を使用してデータを上方にステップ処理で進める場合には、注意が必要です。これらの命令の実行時に行われる加算および比較演算では、オペランドは 32 ビットの符号付き 2 進整数として扱われるので、 $2^{31} - 1$ の後に続く値は、 2^{31} ではなく (このフォーマットでは表現できません)、 -2^{31} になります。命令には、このようなオーバーフローを示す機能はありません。したがって、データ域がアドレス $2^{31} - 1$ で終わっていると、これらの命令の使用に基づいて実行されるある種の

共通ループ技法は機能しません。付録A、『数の表現と命令の使用例』の BRANCH ON INDEX LOW OR EQUAL の例に、この問題が示されています。64 ビット・アドレッシング・モードで、データが 64 ビット・アドレス・スペースの終わりにあり、BRANCH ON INDEX HIGH (BXHG) または BRANCH ON INDEX LOW OR EQUAL (BXLEG) が使用されている場合にも、同じ注意が適用されます。

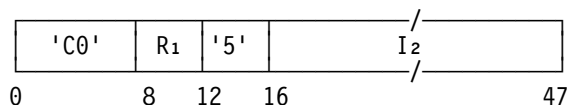
BRANCH RELATIVE AND SAVE

BRAS R₁, I₂ [R1]



BRANCH RELATIVE AND SAVE LONG

BRASL R₁, I₂ [R1L]



現 PSW からの情報 (更新済み命令アドレスを含む) が、第 1 オペランド位置にリンク情報として保管されません。その後、PSW の中の命令アドレスがブランチ・アドレスで置き換えられます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、リンク情報は PSW のビット 32 および 97-127 で、基本アドレッシング・モード・ビットと、更新済み命令アドレスの右端 31 ビットから成っています。このリンク情報は、第 1 オペランド位置のビット位置 32 および 33-63 にそれぞれ入れられ、第 1 オペランド位置のビット 0-31 は変更されません。

64 ビット・アドレッシング・モードでは、リンク情報は更新済み命令アドレスから成り、第 1 オペランド位置のビット位置 0-63 に入れられます。

I₂ フィールドの内容は符号付き 2 進整数で、命令のアドレスに追加してブランチ・アドレスを生成するためのハーフワードの個数を指定します。

条件コード: コードは変更されません。

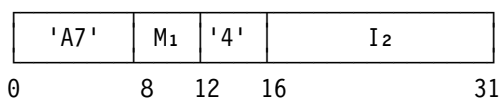
プログラム例外: なし。

プログラミング上の注意:

1. ブランチ・アドレスを指定する方法以外は、操作は BRANCH AND SAVE (BAS) 命令の場合と同じです。BRANCH AND SAVE の使用例は『付録A』に示されています。
2. BRANCH RELATIVE AND SAVE 命令および BRANCH RELATIVE AND SAVE LONG 命令は、BRANCH AND SAVE 命令と同様に、呼び出し側と同じアドレッシング・モードであることが判明しているプログラムへのリンケージに使用するためのものです。BRANCH AND LINK 命令 (BAL および BALR) の代わりに、この命令を使用してください。この命令と、その他のリンケージ命令の詳しい説明については、『リンケージ・スタックなしのサブルーチン・リンケージ』というセクションの 5-11 ページおよび 5-17 ページの『プログラミング上の注意』を参照してください。BRANCH RELATIVE AND SAVE、BRANCH RELATIVE AND SAVE LONG、および BRANCH AND SAVE 命令の利点の説明について、『BRANCH AND LINK』の項の『プログラミング上の注意』も参照してください。
3. 命令が EXECUTE のターゲットである場合は、ブランチはターゲット・アドレスを基準とする相対的なものになります。5-9 ページの『ブランチ・アドレスの生成』を参照してください。

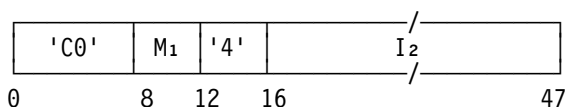
BRANCH RELATIVE ON CONDITION

BRC M₁, I₂ [R1]



BRANCH RELATIVE ON CONDITION LONG

BRCL M₁, I₂ [R1L]



条件コードが M1 で指定された値の 1 つになると、現 PSW 内の命令アドレスがブランチ・アドレスで置き換えられます。それ以外の場合は、通常の命令順序に従って、更新済み命令アドレスを用いて処理が行われま

I2 フィールドの内容は符号付き 2 進整数で、命令のアドレスに追加してブランチ・アドレスを生成するためのハーフワードの個数を指定します。

M1 フィールドは、4 ビットのマスクとして使用されます。4 つの条件コード (0、1、2、および 3) は、次のように左から右への順で、マスクの 4 つのビットに対応します。

条件コード	命令内のマスクのビット番号	マスク位置の値
0	8	8
1	9	4
2	10	2
3	11	1

現行の条件コードを使用して、対応するマスク・ビットが選択されます。条件コードに基づいて選択されたマスク・ビットが 1 であれば、ブランチは成功です。選択されたマスク・ビットが 0 であれば、次の順番に当たる命令を使用して、通常の命令順序に従って処理が進められます。

条件コード: コードは変更されません。

プログラム例外: なし。

プログラミング上の注意:

1. ブランチ・アドレスを指定する方法以外は、操作は BRANCH ON CONDITION 命令の場合と同じです。BRANCH ON CONDITION の使用例は『付録A』に示されています。
2. 複数の条件に基づいてブランチを発生させる場合は、関連の条件コードをそのマスク位置の値の和の形でマスクに指定します。例えば、マスク 12 は、条件コードが 0 または 1 のときにブランチを実行することを示します。
3. 4 個のマスク・ビットがすべて 0 の場合は、ブランチ命令はノーオペレーションと同じことです。4 個のマスク・ビットがすべて 1 の場合、つまりマスク値が 15 の場合は、無条件にブランチが実行されま

4. 命令が EXECUTE のターゲットである場合は、ブランチはターゲット・アドレスを基準とする相対的なものになります。5-9 ページの『ブランチ・アドレスの生成』を参照してください。

BRANCH RELATIVE ON COUNT

BRCT R1, I2 [RI]

'A7'	R1	'6'	I2	
0	8	12	16	31

BRCTG R1, I2 [RI]

'A7'	R1	'7'	I2	
0	8	12	16	31

第 1 オペランドから 1 が減算され、結果が第 1 オペランド位置に入れられます。BRANCH RELATIVE ON COUNT (BRCT) では、第 1 オペランドおよび結果は、32 ビットの 2 進整数として扱われ、オーバーフローは無視されます。BRANCH RELATIVE ON COUNT (BRCTG) では、第 1 オペランドおよび結果は、64 ビットの 2 進整数として扱われ、オーバーフローは無視されます。結果が 0 であれば、更新済み命令アドレスを使用して通常の命令順序に従って処理が進められます。結果が 0 でない場合は、現 PSW の中の命令アドレスがブランチ・アドレスで置き換えられます。

I2 フィールドの内容は符号付き 2 進整数で、命令のアドレスに追加してブランチ・アドレスを生成するためのハーフワードの個数を指定します。

条件コード: コードは変更されません。

プログラム例外: なし。

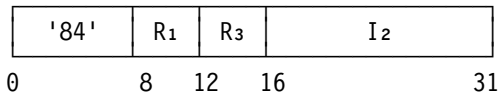
プログラミング上の注意:

1. ブランチ・アドレスを指定する方法以外は、操作は BRANCH ON COUNT 命令の場合と同じです。BRANCH ON COUNT の使用例は『付録A』に示されています。
2. 2 進数減算の結果は符号付きの場合でも符号なしの場合でも同じになるため、第 1 オペランドおよび結果は、符号付き 2 進整数または符号なし 2 進整数のどちらと見なすこともできます。

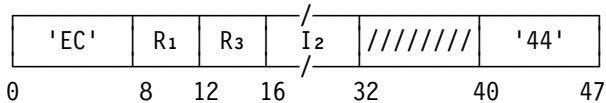
- 初期カウントが 1 の場合、結果は 0 で、ブランチは実行されません。初期カウントが 0 の場合、結果は -1 で、ブランチが実行されます。初期カウントが -1 の場合、結果は -2 で、ブランチが実行されます。以後、同様の処理が続きます。ループ内では、結果が再び 0 に戻るまでは、命令が実行されるたびにブランチも実行されます。 数値範囲上の理由で次のようになる点に注意してください。BRCT の場合は、初期カウントが -2^{31} なら、結果は正の値 $2^{31} - 1$ になります。BRCTG の場合は、初期カウントが -2^{63} なら、結果は 正の値 $2^{63} - 1$ になります。
- 命令が EXECUTE のターゲットである場合は、ブランチはターゲット・アドレスを基準とする相対的なものになります。5-9ページの『ブランチ・アドレスの生成』を参照してください。

BRANCH RELATIVE ON INDEX HIGH

BRXH R_1, R_3, I_2 [RSI]

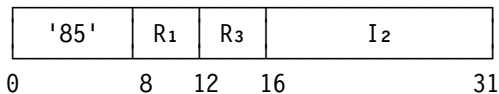


BRXHG R_1, R_3, I_2 [RIE]

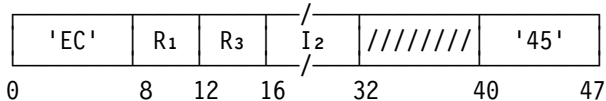


BRANCH RELATIVE ON INDEX LOW OR EQUAL

BRXLE R_1, R_3, I_2 [RSI]



BRXLG R_1, R_3, I_2 [RIE]



第 1 オペランドに増分が加算され、その和が比較値と比較されます。比較の結果に基づいて、ブランチを実行するかどうか決定されます。その後、和が第 1 オペランド位置に入られます。 R_3 フィールドは、増分および比較値が入っているレジスターを指示します。

I_2 フィールドの内容は符号付き 2 進整数で、命令のアドレスに追加してブランチ・アドレスを生成するためのハーフワードの個数を指定します。

BRANCH RELATIVE ON INDEX HIGH では、和の方が大きい場合は、現 PSW 中の命令アドレスがブランチ・アドレスで置き換えられます。和が比較値より小さいかまたは等しい場合は、更新済み命令アドレスを使用して通常の命令順序に従って処理が進められます。

BRANCH RELATIVE ON INDEX LOW OR EQUAL では、和が比較値より小さいかまたは等しい場合は、現 PSW 中の命令アドレスがブランチ・アドレスで置き換えられます。和の方が大きい場合は、更新済み命令アドレスを使用して、通常の命令順序に従って処理が進められます。

R_3 フィールドが偶数なら、一对のレジスターを示します。このペアの偶数レジスターの内容が増分として使用され、奇数レジスターの内容が比較値として使用されます。 R_3 フィールドが奇数なら、単一レジスターを示し、このレジスターの内容が増分と比較値の両方として使用されます。

加算および比較が目的の場合、すべてのオペランドと結果は、BRXH および BRXLE の場合は 32 ビットの符号付き 2 進整数として扱われ、BRXHG および BRXLG の場合は 64 ビットの符号付き 2 進整数として扱われます。加算により生じたオーバーフローは無視されます。

比較値レジスターが第 1 オペランド位置にも入ることが指定されている場合でも、比較値として使用されるのは、比較値レジスターの元の内容です。

ブランチが実行されるかどうかに関係なく、和は第 1 オペランド位置に入られます。

条件コード: コードは変更されません。

プログラム例外: なし。

プログラミング上の注意:

1. ブランチ・アドレスを指定する方法以外は、操作は BRANCH ON INDEX HIGH 命令および BRANCH ON INDEX LOW OR EQUAL 命令の場合と同じです。BRANCH ON INDEX HIGH および BRANCH ON INDEX LOW OR EQUAL の使用例が幾つか、『付録A』に示されています。
2. これらの命令の名前に含まれている「INDEX」(指標) という語は、命令の主目的の 1 つが指標値を増分を加えてテストすることであることを示しています。増分は、符号付き 2 進整数であり、任意の指定量で汎用レジスタ R₁ 内の値を増加または減少させるために使用できます。
3. 31 ビット・アドレッシング・モードでは、記憶機構内のデータ域が 31 ビット・アドレス・スペースの右端にあり、BRANCH RELATIVE ON INDEX HIGH (BRXH) または BRANCH RELATIVE ON INDEX LOW OR EQUAL (BRXLE) 命令を使用してデータを上方にステップ処理で進める場合には、注意が必要です。これらの命令の実行時に行われる加算および比較演算では、オペランドは 32 ビットの符号付き 2 進整数として扱われるので、 $2^{31} - 1$ の後に続く値は、 2^{31} ではなく (このフォーマットでは表現できません)、 -2^{31} になります。命令には、このようなオーバーフローを示す機能はありません。したがって、データ域がアドレス $2^{31} - 1$ で終わっていると、これらの命令の使用に基づいて実行されるある種の共通ループ技法は機能しません。『付録』の BRANCH ON INDEX LOW OR EQUAL の例に、この問題が示されています。64 ビット・アドレッシング・モードで、データが 64 ビット・アドレス・スペースの終わりにあり、BRANCH RELATIVE ON INDEX HIGH (BRXHG) または BRANCH RELATIVE ON INDEX LOW OR EQUAL (BRXLG) が使用されている場合にも、同じ注意が適用されます。
4. 命令が EXECUTE のターゲットである場合は、ブランチはターゲット・アドレスを基準とする相対的なものになります。5-9 ページの『ブランチ・アドレスの生成』を参照してください。

CHECKSUM

CKSM R₁, R₂ [RRE]

	'B241'	////////	R ₁	R ₂
0		16	24	28 31

第 2 オペランドの連続する 4 バイトのエレメントが、汎用レジスタ R₁ のビット位置 32-63 にある第 1 オペランドに加算され、これらのビット位置に 32 ビットのチェックサムを形成します。第 1 オペランドおよび 4 バイト・エレメントは、32 ビットの符号なし 2 進整数として扱われます。エレメントを 1 つ加算するたびに、第 1 オペランドのビット位置 32 からの繰り上がりが第 1 オペランドのビット位置 63 に加算されます。汎用レジスタ R₁ のビット 0-31 は、常に無変更のままです。第 2 オペランドが 4 バイトの倍数になっていない場合は、最後の 1 バイト、2 バイト、または 3 バイトは、4 バイト・エレメントを形成するために必要な個数のすべて 0 のバイトを右側に付加して処理されます。第 1 オペランドへの 4 バイト・エレメントの追加は、第 2 オペランド全体、または CPU で決定される量の第 2 オペランドの処理が終わるまで行われます。結果は条件コードで示されます。

R₂ フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 2 オペランドの左端バイトの位置は、R₂ 汎用レジスタの内容によって指定されます。第 2 オペランド位置のバイトの数は、R₂ + 1 汎用レジスタ内の 32 ビットまたは 64 ビットの符号なし 2 進整数によって指定されます。

汎用レジスタ R₂ 内のアドレス、および汎用レジスタ R₂ + 1 内の長さの取り扱いは、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、レジスタのビット位置 0-63 の内容がアドレスを形成します。24 ビットまたは 31 ビットのアドレッシング・モードでは、長さを示すのは、汎用レジスタ R₂ + 1 のビット位置 32-63 にある 32 ビットの符号なし 2 進整数で、ビット位置 0-31 の内容は無視されます。64 ビット

ト・アドレッシング・モードでは、長さは、レジスターの 64 ビットの符号なし 2 進整数で示されます。

第 2 オペランドの 4 バイト・エレメントを第 1 オペランドに加算する操作は左から右へと 4 バイト・エレメント単位で進められ、次のいずれかの条件が発生すると同時に終了します。(1) 第 2 オペランド全体の処理が終わる。または、(2) 全体量より少ないが、CPU で決定された量の第 2 オペランドの処理が終わる。どちらの場合も、汎用レジスター R₁ のビット位置 32-63 に示される結果が、処理された第 2 オペランド部分に関する 32 ビットのチェックサムです。第 2 オペランドが 4 バイトの倍数になっていない場合は、第 2 オペランドの最後の、4の倍数を超える残りのバイトは、適切な数のすべて 0 のバイトが右側に概念的に付加され、最後の 4 バイト・エレメントを形成するものと見なされます。

第 2 オペランド全体が処理されたために操作が終了した場合は、条件コードが 0 に設定されます。全体より少ない CPU で決定される量の第 2 オペランド部分が処理されたために操作が終了した場合は、条件コードが 3 に設定されます。条件コードが 3 に設定されて操作が終了する場合は、第 1 オペランドのビット位置 32 からの繰り上がりは、操作の終了前に、第 1 オペランドのビット位置 63 に加算されます。

操作の完了時に、R₂ + 1 レジスターの 32 ビットまたは 64 ビットのオペランド長フィールドが、第 1 オペランドに実際に追加された第 2 オペランドのバイト数 (概念的に追加されたすべて 0 のバイトを含まない数) だけ減少し、R₂ レジスター内のアドレスが同じ数だけ増加します。したがって、条件コードが 0 に設定された場合は、32 ビットまたは 64 ビットのオペランド長フィールドには 0 が含まれ、条件コードが 3 に設定された場合は 0 でない値が含まれます。24 ビットまたは 31 ビットのアドレッシング・モードでは、R₂ + 1 レジスターのビット 0-31 は、常に無変更のままです。

条件コード 3 が設定された場合、命令が使用している汎用レジスターは、ブランチで戻って同じ命令を再実行しさえすれば第 2 オペランドの残り部分を処理できるように設定されています。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。最小量は、4 バイトか、または R₂ + 1 汎用レジスターに指定されたバイト数の、どちらか小さい方です。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスター R₂ のビット位置 32-63 内の、アドレスの一部となっていない左端ビットは、レジスター R₂ + 1 の長さの初期値が 0 の場合でも、0 に設定されることもあり、変更されないこともあります。汎用レジスター R₂ のビット 0-31 は変更されません。

R₁ レジスターが R₂ または R₂ + 1 レジスターと同一の場合は、結果は予測不能です。

処理された最後のバイトの右側にある第 2 オペランド部分については、アクセス例外が認識される場合とされない場合があります。第 2 オペランドが 4K バイトより長い場合は、処理された最後のバイトを 4K バイトを超える位置より後にある記憶位置については、アクセス例外は認識されません。

R₂ フィールドが奇数の場合は、アクセス例外は認識されません。第 2 オペランドの長さが 0 の場合も、アクセス例外は認識されません。

結果の条件コード:

- 0 第 2 オペランド全体が処理されました。
- 1 --
- 2 --
- 3 CPU で決定された量の第 2 オペランド部分が処理されました。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 指定例外

プログラミング上の注意:

1. R₁ 汎用レジスターのビット位置 32-63 の初期内容は、32 ビットのチェックサムを提供する元になります。プログラムでは、通常、CHECKSUM 命令を発行する前に、これらの内容をすべて 0 に設定する必要があります。
2. 16 ビットのチェックサムは、例えば、TCP/IP アプリケーションで使用されます。次のプログラムを CHECKSUM 命令の後で実行することで、汎用レジスター R₂ のビット位置 32-63 に、汎用レジスター R₁ のビット位置 32-63 の 32 ビットのチェックサムから、16 ビットのチェックサムを生成することができます。プログラムには、各命令を実行した後の R₂ および R₂ + 1 レジスターのビット位置 32-63 の内容を示す注釈を付けてあります。R₁ レジスターのビット位置 32-63 の内容は、A,B で表

されています。これは、ビット位置 32-47 の値 A、およびビット位置 48-63 の値 B を指しています。値 C は、A + B からの繰り上がりです。CHECKSUM で条件コードが 0 に設定されたときは、レジスタ R2 + 1 のビット位置 32-63 にはすべて 0 が含まれることが既知の事実であることに注意してください。

プログラム		R2 の ビット 32-63	R2+1 の ビット 32-63
LR	R2, R1	A, B	0, 0
SRDL	R2, 16	0, A	B, 0
ALR	R2, R2+1	B, A	B, 0
ALR	R2, R1	A+B+C, A+B	B, 0
SRL	R2, 16	0, A+B+C	B, 0

3. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。
4. CHECKSUM の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)
5. 7-32ページの図7-2 に、この操作の要約を示します。

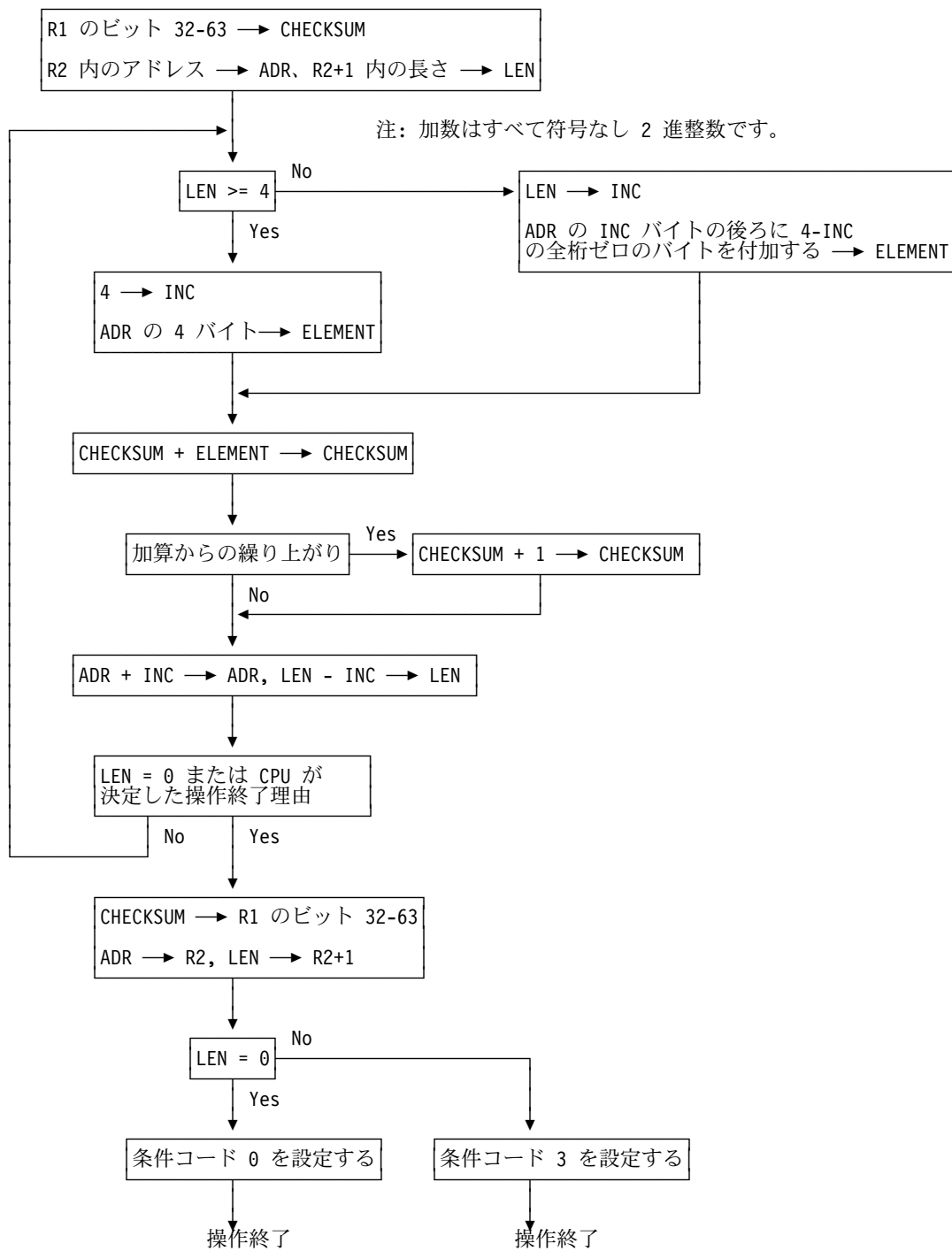
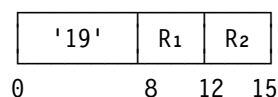


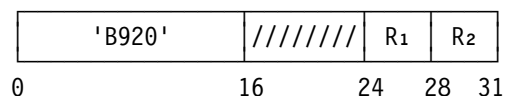
図 7-2. CHECKSUM の実行

COMPARE

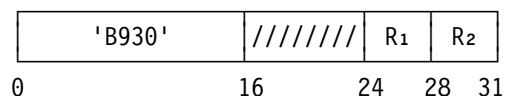
CR R₁,R₂ [RR]



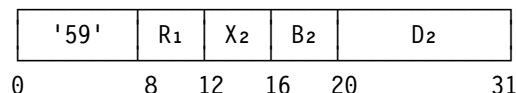
CGR R₁,R₂ [RRE]



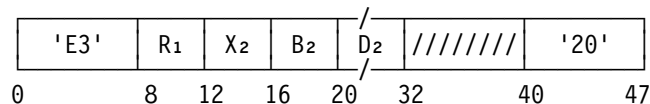
CGFR R₁,R₂ [RRE]



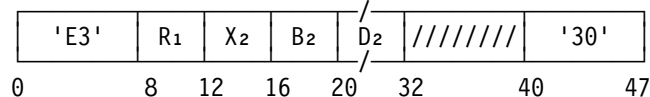
C R₁,D₂(X₂,B₂) [RX]



CG R₁,D₂(X₂,B₂) [RXE]



CGF R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。COMPARE (CR、C) では、オペランドは 32 ビットの符号付き 2 進整数として扱われます。COMPARE (CGR、CG) では、オペランドは 64 ビットの符号付き 2 進整数として扱われます。COMPARE (CGFR、CGF) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、第 1 オペランドは 64 ビットの符号付き 2 進整数として扱われます。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。

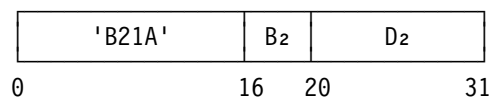
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: C、CG、および CGF のオペランド 2 のみ)

COMPARE AND FORM CODEWORD

CFC D₂(B₂) [S]



汎用レジスタ 2 には指標が含まれています。この指標は、汎用レジスタ 1 および 3 の内容と共に使用されて、第 1 オペランドおよび第 3 オペランドと呼ばれる、記憶域内の 2 つのフィールドの開始アドレスが指定されます。第 1 オペランドと第 3 オペランドが論理比較され、ソート・マージ・アルゴリズムで使用するためのコードワードが形成されます。

第 2 オペランドのアドレスは、データのアドレッシングには使用されません。第 2 オペランド・アドレスのビット 49-62 は、右端に 1 個、左端に 1 個の 0 が付加されて、16 ビットの指標限界として使用されます。第 2 オペランド・アドレスのビット 63 はオペランド制御ビットです。ビット 63 が 0 であれば、コードワードは高位オペランドから形成されます。ビット 63 が 1 であれば、コードワードは低位オペランドから形成されます。第 2 オペランド・アドレスのその他の部分は無視されません。

汎用レジスタ 1 および 3 には、第 1 オペランドおよび第 3 オペランドの基底アドレスが含まれています。汎用レジスタ 2 のビット 48-63 は、第 1 オペランドおよび第 3 オペランドをアドレッシングするための指標として使用されます。汎用レジスタ 1、2、および 3 には、最初はすべて偶数の値が含まれていなければなりません。さもないと、指定例外が認識されます。

アクセス・レジスタ・モードでは、アクセス・レジスタ 1 は、第 1 オペランドおよび第 3 オペランドが含まれているアドレス・スペースを指定します。

第 1 オペランドと第 3 オペランドが比較されるときは、単位のサイズ、結果のコードワードのサイズ、および汎用レジスタ 1、2、および 3 のビット 0-31 が操作に

どう関与するかは、アドレッシング・モードによって異なります。24 ビットまたは 31 ビットのアドレッシング・モードでは、比較単位は 2 バイト、コードワードは 4 バイトで、ビット 0-31 は無視されて無変更のままになります。64 ビット・アドレッシング・モードでは、比較単位は 6 バイト、コードワードは 8 バイトで、ビット 0-31 は操作で使用され、場合によっては変更されます。

24 ビットまたは 31 ビットのアドレッシング・モードの操作

この操作では、第 1 オペランドと第 3 オペランドがハーフワード単位で比較され、等しくないハーフワードのペアが見つかるか、または指標が指標限界を超えるまで、指標が増加されます。この操作は操作単位ごとのまとまりで進められ、操作単位間で割り込みが生じることもあります。

1 つの操作単位の開始時に、指標 (汎用レジスター 2 のビット 48-63) が指標限界と論理比較されます。指標が指標限界より大きい場合は、汎用レジスター 3 のビット 32-63 が、ビット 32 を 1 に設定して汎用レジスター 2 のビット位置 32-63 に入れられ、条件コード 0 が設定されて、命令が完了します。

指標が指標限界より小さいかそれに等しい場合は、比較の対象になるハーフワードの現行ペアを見つけるために、その指標が第 1 オペランドおよび第 3 オペランドの基底アドレスに適用されます。指標の左側に 48 個の 0 を付加したものと、汎用レジスター 1 のビット 32-63 の左側に 32 個の 0 を付加したものが加算されて、64 ビットの間値が形成されます。ビット位置 32 からの繰り上がりが生じても、無視されます。現行の第 1 オペランド・ハーフワードのアドレスは、オペランド・アドレスの生成に関する次の通常規則に基づき、中間値から生成されます。現行の第 3 オペランド・ハーフワードのアドレスは、汎用レジスター 3 のビット 32-63 と指標を加算することによって、同様の方法で形成されます。

現行の第 1 オペランド・ハーフワードと第 3 オペランド・ハーフワードが論理比較されます。等しい場合は、汎用レジスター 2 の内容が 2 だけ増加され、操作単位が終了します。

比較値が等しくない場合は、汎用レジスター 2 の内容が 2 だけ増加され、次に、16 個分のビット位置だけ左へ論理的にシフトされます。シフトは、ビット位置 32-63 の範囲内でのみ行われます。オペランド制御ビットが 0 の場合は、(1) 高位ハーフワードの 1 の補数が汎用レジスター 2 のビット位置 48-63 に入れられ、さらに (2)

オペランド 1 の方が大きかった場合は、汎用レジスター 1 と 3 のビット 32-63 が交換されます。オペランド制御ビットが 1 の場合は、(1) 低位ハーフワードが汎用レジスター 2 のビット位置 48-63 に入れられ、さらに (2) オペランド 1 の方が小さかった場合は、汎用レジスター 1 と 3 のビット 32-63 が交換されます。

アクセス例外を認識するために、オペランド 1 およびオペランド 3 は両方とも、指標限界から指標を引いた値より 2 大きい値の長さを持つものと見なされます。

64 ビット・アドレッシング・モードの操作

この操作では、第 1 オペランドと第 3 オペランドが一度に 6 バイト単位で比較され、等しくない 6 バイト単位のペアが見つかるか、または指標が指標限界を超えるまで、指標が増加されます。この操作は操作単位ごとのまとまりで進められ、操作単位間で割り込みが生じることもあります。

1 つの操作単位の開始時に、指標 (汎用レジスター 2 のビット 48-63) が指標限界と論理比較されます。指標が指標限界より大きい場合は、汎用レジスター 3 のビット 0-63 が、ビット 0 を 1 に設定して汎用レジスター 2 のビット位置 0-63 に入れられ、条件コード 0 が設定されて、命令が完了します。

指標が指標限界より小さいかそれに等しい場合は、比較の対象になる 6 バイト単位の現行ペアを見つけるために、その指標が第 1 オペランドおよび第 3 オペランドの基底アドレスに適用されます。指標の左側に 48 個の 0 を付加したものと、汎用レジスター 1 のビット 0-63 が加算されて、現行の第 1 オペランドの 6 バイト単位の 64 ビット・アドレスを形成します。ビット位置 0 からの繰り上がりが生じても、無視されます。現行の第 3 オペランドの 6 バイト単位のアドレスは、汎用レジスター 3 のビット 0-63 と指標を加算することによって、同様の方法で形成されます。

現行の第 1 オペランドと第 3 オペランドの 6 バイト単位が論理比較されます。等しい場合は、汎用レジスター 2 の内容が 6 だけ増加され、操作単位が終了します。

比較値が等しくない場合は、汎用レジスター 2 の内容が 6 だけ増加され、次に 48 個分のビット位置だけ左へ論理的にシフトされます。オペランド制御ビットが 0 の場合は、(1) 高位の 6 バイト単位の 1 の補数が汎用レジスター 2 のビット位置 16-63 に入れられ、さらに (2) オペランド 1 の方が大きかった場合は、汎用レジスター 1 と 3 のビット 0-63 が交換されます。オペランド制御ビ

ットが 1 の場合は、(1) 低位の 6 バイト単位が汎用レジスタ 2 のビット位置 16-63 に入れられ、さらに (2) オペランド 1 の方が小さかった場合は、汎用レジスタ 1 と 3 のビット 0-63 が交換されます。

アクセス例外を認識するために、オペランド 1 およびオペランド 3 は両方とも、指標限界から指標を引いた値より 6 大きい長さを持つものと見なされます。

アドレッシング・モードからは独立している指定

命令が割り込まれたときの条件コードは予測不能です。

初期時点で指標が指標限界より大きい場合は、記憶域オペランドに関するアクセス例外は認識されません。4K バイトより長いオペランドの場合は、処理対象バイトより後の 4K バイトを超える部分については、アクセス例外は認識されません。指定例外条件が存在する場合は、アクセス例外は認識されません。

B₂ フィールドが汎用レジスタ 2 を指定している場合は、指標限界が再計算されるかどうかは予測不能です。したがって、この場合、オペランド長は予測不能です。ただし、どのような場合でもオペランドの長さが 2¹⁵ バイトを超えることはできません。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 オペランド制御ビットが 0 でオペランド 1 が小さい、またはオペランド制御ビットが 1 でオペランド 3 が小さい。
- 2 オペランド制御ビットが 0 でオペランド 1 が大きい、またはオペランド制御ビットが 1 でオペランド 3 が大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 3)
- 指定例外

プログラミング上の注意:

1. COMPARE AND FORM CODEWORD の使用例は、付録 A、『数の表現と命令の使用例』の『ソート命令』に示されています。
2. 第 1 オペランドおよび第 3 オペランドの、比較の開始点となるハーフワードまたは 6 バイト単位のオフセット (アドレッシング・モードによって決まる) は、COMPARE AND FORM CODEWORD を実

行する前に、汎用レジスタ 2 のビット位置 48-63 に入っている必要があります。第 2 オペランドのアドレスから派生する指標限界は、比較の対象となる第 1 オペランドおよび第 3 オペランドの最後のハーフワードまたは 6 バイト単位のオフセットでなければなりません。両方のオペランドの比較結果が等しくなかった場合は、COMPARE AND FORM CODEWORD の実行により、汎用レジスタ 2 に形成されるコードワードの左端の 16 ビット (24 ビットまたは 31 ビットのアドレッシング・モードではレジスタのビット 32-47 で、64 ビット・アドレッシング・モードではビット 0-15) に、最初のハーフワードまたは 6 バイト単位のオフセットが示されます。UPDATE TREE 操作でのコードワードの比較結果が等しかった場合は、汎用レジスタ 2 のビット位置 32-47 (24 ビットまたは 31 ビットのアドレッシング・モードの場合) またはビット位置 0-15 (64 ビット・アドレッシング・モードの場合) には、コードワードの結合を崩すために別の COMPARE AND FORM CODEWORD で比較を再開するために使用する、オフセットが含まれています。オペランド制御ビット値 0 または 1 は、それぞれ、オペランドを昇順または降順でソートするために使用されます。ソートにおけるコードワードの使用方法については、A-51 ページの『ソート命令』を参照してください。

3. 条件コードは、オペランドの比較結果を最長 32,768 バイトまで示します。オペランドが等しかった場合は、汎用レジスタ 2 のビット位置 32-63 (24 ビットまたは 31 ビット・アドレッシング・モードの場合) またはビット位置 0-63 (64 ビット・アドレッシング・モードの場合) に、負のコードワードが含まれています。ただし、24 ビットまたは 31 ビット・モードで、指標限界が 32,766 であって比較対象のオペランドが最後の 2 バイトのみで異なっている場合、または 64 ビット・モードで、指標限界が 32,762 であってオペランドが最後の 6 バイトのみで異なっている場合にも、負のコードワードが含まれる結果になります。この後者の場合のコードワードが UPDATE TREE で使用されると、汎用レジスタ 0 および 1 には誤った結果が示されることがあります。したがって、結果のコードワードを UPDATE TREE で使用する場合は、指標限界が 32,764 (24 ビットまたは 31 ビット・モードの場合) または 32,760 (64 ビット・モードの場合) を超えないようにする必要があります。
4. COMPARE AND FORM CODEWORD を EXECUTE のターゲットにする場合は、特別な注意が必要です。EXECUTE の項の『プログラミング

グ上の注意』の、割り込み可能命令についての説明を参照してください。

5. 割り込み可能命令については、さらに詳しいプログラミング上の注意が第5章、『プログラムの実行』の『割り込み可能命令』に示されています。
6. COMPARE AND FORM CODEWORD の記憶域オペランド参照は、複数アクセス参照となることも

あります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

7. 図7-3 および 7-37ページの図7-4 に、24 ビットまたは 31 ビットのアドレッシング・モードの場合の操作の要約を示し、7-38ページの図7-5 および 7-39ページの図7-6 に、64 ビット・アドレッシング・モードの場合の操作の要約を示します。

オペランド制御ビット	関係	結果の条件コード	GR2 内の結果 (ビット 32-63)	GR1 内の結果 (ビット 32-63)	GR3 内の結果 (ビット 32-63)
0	op1 = op3	0	OGR3b1	-	-
0	op1 < op3	1	X, nop3	-	-
0	op1 > op3	2	X, nop1	OGR3	OGR1
1	op1 = op3	0	OGR3b1	-	-
1	op1 < op3	2	X, top1	OGR3	OGR1
1	op1 > op3	1	X, top3	-	-

説明:

- ビットは変更されません。
- OGR1 GR1 のビット 32-63 の元の値。
- OGR3 GR3 のビット 32-63 の元の値。
- OGR3b1 ビット 32 が 1 に設定されている GR3 のビット 32-63 の元の値。
- X GR2 の 32-47 には、最初の不一致のハーフワードの指標より 2 だけ大きい値が入ります。
- nop1 GR2 のビット 48-63 には、オペランド 1 内の最初の不一致のハーフワードの 1 の補数が入ります。
- nop3 GR2 のビット 48-63 には、オペランド 3 内の最初の不一致のハーフワードの 1 の補数が入ります。
- top1 GR2 のビット 48-63 には、オペランド 1 内の最初の不一致のハーフワードが入ります。
- top3 GR2 のビット 48-63 には、オペランド 3 内の最初の不一致のハーフワードが入ります。

図 7-3. COMPARE AND FORM CODEWORD の操作 (24 ビットおよび 31 ビット・アドレッシング・モードの場合)

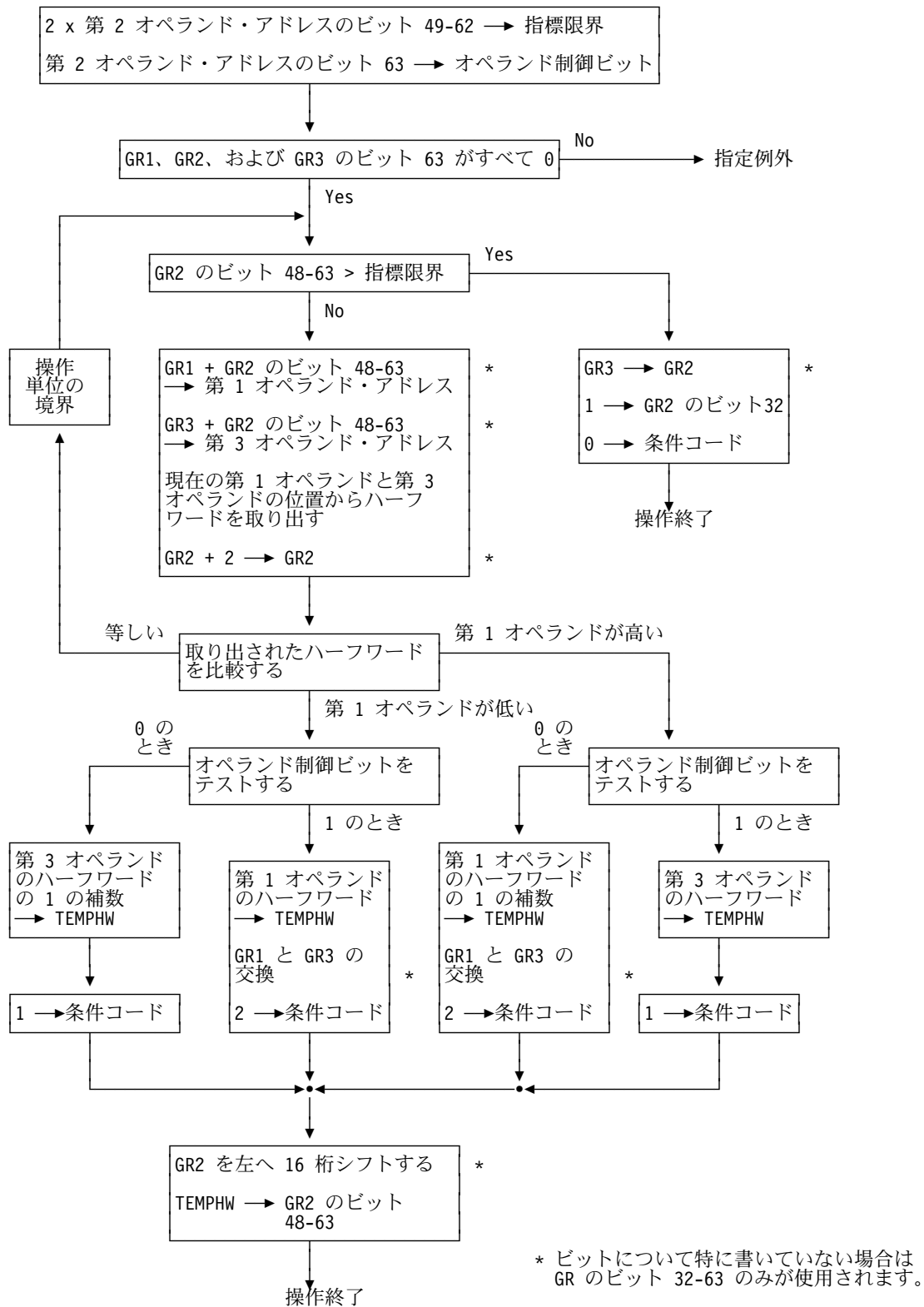


図 7-4. COMPARE AND FORM CODEWORD の実行 (24 ビットまたは 31 ビットのアドレッシング・モードの場合)

オペランド制御ビット	関係	結果の条件コード	GR2 内の結果 (ビット 0-63)	GR1 内の結果 (ビット 0-63)	GR3 内の結果 (ビット 0-63)
0	op1 = op3	0	OGR3b1	-	-
0	op1 < op3	1	X, nop3	-	-
0	op1 > op3	2	X, nop1	OGR3	OGR1
1	op1 = op3	0	OGR3b1	-	-
1	op1 < op3	2	X, top1	OGR3	OGR1
1	op1 > op3	1	X, top3	-	-

説明:

- ビットは変更されません。
- OGR1 GR1 のビット 0-63 の元の値。
- OGR3 GR3 のビット 0-63 の元の値。
- OGR3b1 ビット 0 が 1 に設定されている GR3 のビット 0-63 の元の値
- X GR2 のビット 0-15 には、最初の不一致の 6 バイト単位の指標より 6 だけ大きい値が入ります。
- nop1 GR2 のビット 16-63 には、オペランド 1 内の最初の不一致の 6 バイト単位の 1 の補数が入ります。
- nop3 GR2 のビット 16-63 には、オペランド 3 内の最初の不一致の 6 バイト単位の 1 の補数が入ります。
- top1 GR2 のビット 16-63 には、オペランド 1 内の最初の不一致の 6 バイト単位が入ります。
- top3 GR2 のビット 16-63 には、オペランド 3 内の最初の不一致の 6 バイト単位が入ります。

図 7-5. COMPARE AND FORM CODEWORD の操作 (64 ビット・アドレッシング・モードの場合)

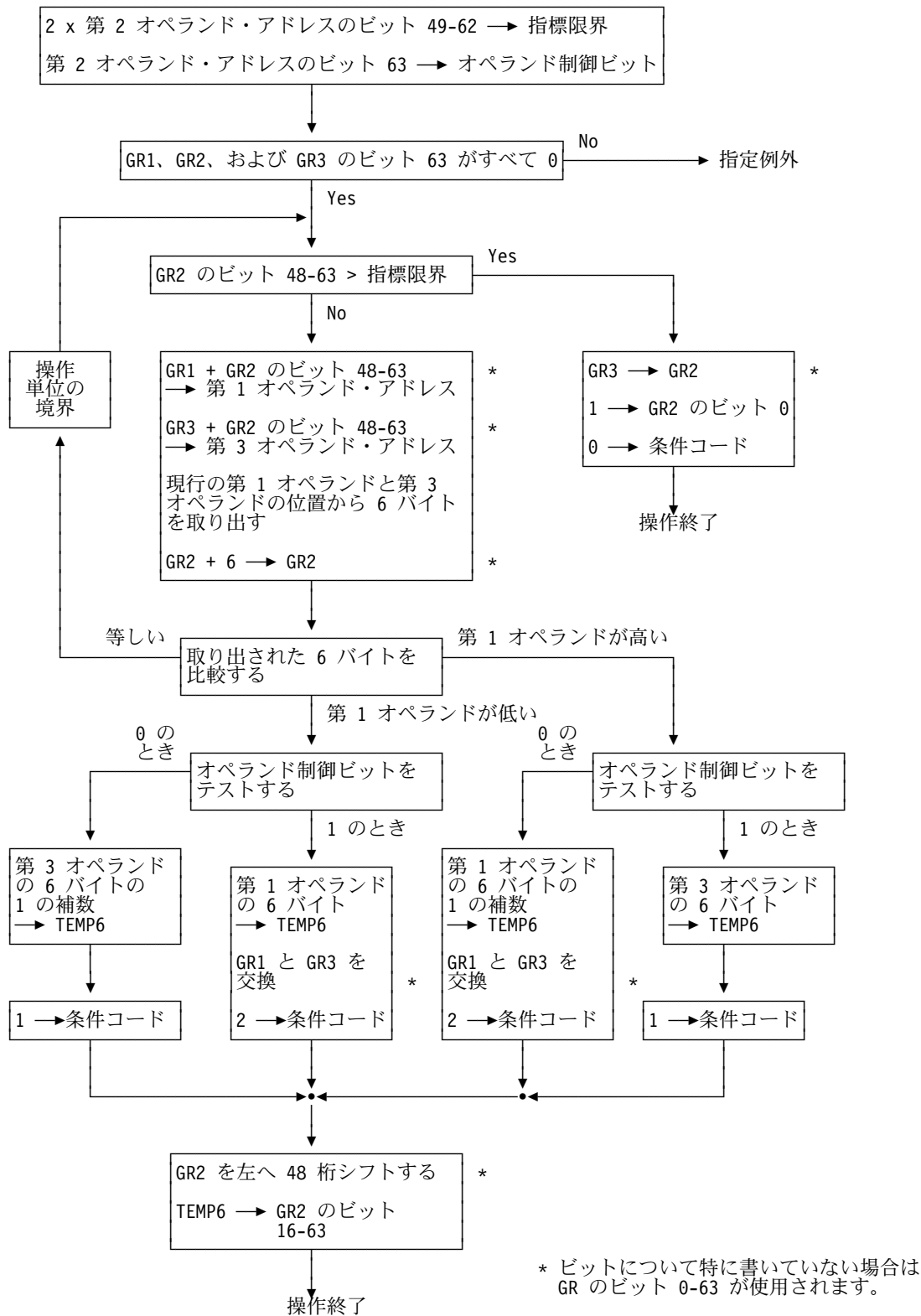


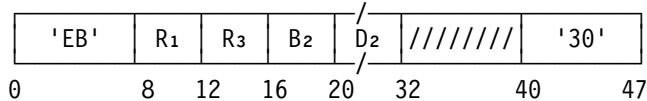
図 7-6. COMPARE AND FORM CODEWORD の実行 (64 ビット・アドレッシング・モードの場合)

COMPARE AND SWAP

CS R₁,R₃,D₂(B₂) [RS]

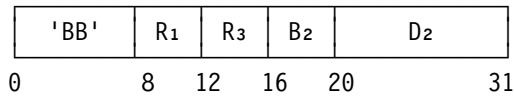


CSG R₁,R₃,D₂(B₂) [RSE]

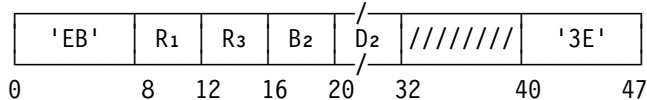


COMPARE DOUBLE AND SWAP

CDS R₁,R₃,D₂(B₂) [RS]



CDSG R₁,R₃,D₂(B₂) [RSE]



第 1 オペランドと第 2 オペランドが比較されます。両方が等しければ、第 3 オペランドが第 2 オペランド位置に格納されます。等しくない場合は、第 2 オペランドが第 1 オペランド位置にロードされます。比較の結果は条件コードで示されます。

COMPARE AND SWAP (CS) では、第 1 オペランドおよび第 3 オペランドはどちらも長さが 32 ビットで、それぞれのオペランドが汎用レジスタのビット位置 32-63 を占めます。第 2 オペランドは記憶域内の 1 ワードです。

COMPARE AND SWAP (CSG) では、第 1 オペランドおよび第 3 オペランドはどちらも長さが 64 ビットで、それぞれのオペランドが汎用レジスタのビット位置 0-63 を占めます。第 2 オペランドは記憶域内のダブルワードです。

COMPARE DOUBLE AND SWAP (CDS) では、第 1 オペランドおよび第 3 オペランドは、どちらも長さが 64 ビットです。オペランドの最初の 32 ビットは、汎用レジスタの奇偶ペアの偶数番号のレジスタのビット

位置 32-63 を占め、2 番目の 32 ビットは、同じペアの奇数番号のレジスタのビット位置 32-63 を占めます。第 2 オペランドは記憶域内のダブルワードです。

COMPARE DOUBLE AND SWAP (CDSG) では、第 1 オペランドおよび第 3 オペランドは、どちらも長さが 128 ビットです。オペランドの最初の 64 ビットは、汎用レジスタの奇偶ペアの偶数番号のレジスタのビット位置 0-63 を占め、2 番目の 64 ビットは、同じペアの奇数番号のレジスタのビット位置 0-63 を占めます。第 2 オペランドは記憶域内のクワッドワードです。

比較結果が等しければ、第 3 オペランドが第 2 オペランド位置に格納されます。比較のための第 2 オペランドの取り出しと第 2 オペランド位置への格納は、他の CPU からは、ブロック単位のインターロック更新参照のように見えます。

比較の結果が等しくなければ、第 2 オペランド位置は無変更のままになります。ただし、一部のモデルには、値が取り出され、その後で第 2 オペランド位置に変更なしで戻されるものもあります。この更新は、他の CPU からはブロック単位のインターロック更新参照のように見えます。

オペランドが取り出される前、および操作が完了した後で、逐次化機能が実行されます。

COMPARE AND SWAP (CS) の第 2 オペランドは、ワード境界に指定されていなければなりません。

COMPARE AND SWAP (CSG) および COMPARE DOUBLE AND SWAP (CDS) の第 2 オペランドは、ダブルワード境界に指定されていなければなりません。

COMPARE DOUBLE AND SWAP (CDSG) の第 2 オペランドは、クワッドワード境界に指定されていなければなりません。COMPARE DOUBLE AND SWAP の R₁ フィールドおよび R₃ フィールドは、それぞれ偶数番号のレジスタを指定していなければなりません。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 第 1 オペランドと第 2 オペランドが等しく、第 2 オペランドは第 3 オペランドで置き換えられました。
- 1 第 1 オペランドと第 2 オペランドが等しく、第 1 オペランドは第 2 オペランドで置き換えられました。
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (取り出しおよび格納: オペランド 2)
- 指定例外

プログラミング上の注意:

1. COMPARE AND SWAP および COMPARE DOUBLE AND SWAP 命令の使用例が幾つか、付録A、『数の表現と命令の使用例』に示されています。
2. 以下の注のうち、オペランド・サイズに関する説明の幾つかは、CS および CDS についてのものです。同様の注で、もっと大きいオペランド・サイズに関する説明は、CSG および CDSG に適用されるものです。
3. COMPARE AND SWAP は、マルチプログラミング環境またはマルチプロセッシング環境で共通記憶域を共用する複数の CPU プログラムで使用できます。次に 2 つの例を挙げます。
 - a. CPU プログラムの実行中に別の CPU プログラムが割り込んで同じ記憶位置を更新する可能性がある場合、または別の CPU プログラムがその記憶位置を同時に更新する可能性がある場合でも、実行中の CPU プログラムは、次のプロシージャーを実行することによって、その記憶位置の内容を変更することができます。まず、更新対象の 1 つ以上のバイトを含んでいるワード全体を汎用レジスターにロードします。次に、更新後の値を計算し、それを別の汎用レジスターに置きます。そして、R₁ フィールドで元の値が入っているレジスターを指定し、R₃ フィールドで更新後の値が入っているレジスターを指定して、COMPARE AND SWAP を実行します。更新が成功であれば、条件コード 0 が設定されます。記憶位置に元の値がもう含まれていなければ、更新は失敗です。

COMPARE AND SWAP 命令の R₁ フィールドで指定された汎用レジスターには、記憶位置の新しい現行値が含まれていて、条件コード 1 が設定されます。条件コード 1 が設定されると、CPU プログラムはその新しい現行値を使用して、同じプロシージャーを繰り返すことができます。
 - b. COMPARE AND SWAP は、共通記憶域の共用の制御に使用することができ、共通記憶域が使用中のときにはメッセージを (メッセージのチェーン・リストの中に) 残す機能も備えています。この機能を達成するには、記憶域内の 1 ワードを制御ワードとして使用します。このワ

ードの値が 0 であれば、共通域は使用中でなく、メッセージは存在しないことを示します。負の値であれば、共通域は使用中であって、メッセージは存在しないことを示します。0 以外の正の値であれば、共通域は使用中であって、その値がリストに追加された最新メッセージのアドレスであることを示します。この方法により、その共通域の使用権を得ようとする CPU プログラムの数が幾つであっても、COMPARE AND SWAP を使用して、制御ワードを更新し、共通域が使用中であることを示したり、メッセージをリストに追加することができます。共通域の使用権を獲得した単一の CPU プログラムは、COMPARE AND SWAP を使用して、リストからメッセージを削除することもできます。

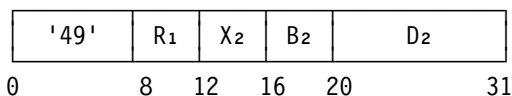
4. COMPARE DOUBLE AND SWAP は、COMPARE AND SWAP と同様の方法で使用できます。そのほかに、もう 1 つ別の使用方法があります。上記の『プログラミング上の注意の 3b』で述べたような、リスト内の最初のメッセージをアドレッシングするために使用する制御ワードが設定されている、チェーン・リストを考えてみましょう。複数の CPU プログラム (共通域の使用権を獲得した単一の CPU プログラムのみでなく) が COMPARE AND SWAP を使用してメッセージを削除することが許される場合は、リストが誤って更新される可能性があります。この状況が起きるのは、例えば、1 つの CPU プログラムが最新メッセージを削除するためにそのメッセージのアドレスを取り出した後で、別の CPU プログラムが最初の 2 つのメッセージを削除し、最初のメッセージをチェーンに戻した場合です。処理を継続中の最初の CPU は、リストが変更されたことを簡単には検出できません。制御ワードのサイズをダブルワードに拡大し、最初のメッセージ・アドレスと、リストが変更されるたびに増加する変更番号が含まれている 1 ワードとをそのダブルワードに含め、COMPARE DOUBLE AND SWAP を使用して両方のフィールドを同時に更新することで、リストが誤った状態に更新される可能性は、無視できる程度にまで減少します。つまり、誤った更新が行われるのは、最初の CPU プログラムの処理が遅れている間に正確に 2³² の倍数に等しい数の変更が行われ、しかも最後の変更で元のメッセージ・アドレスが制御ワードに戻された場合のみになるからです。
5. COMPARE AND SWAP および COMPARE DOUBLE AND SWAP には、チャンネル・プログラムからの記憶域アクセスに対するインターロック機能はありません。したがって、チャンネル・プログラムが格納先にする可能性のある位置を更新するとき

に、これらの命令を使用してはなりません。チャンネル・プログラムのデータが失われる場合があります。

- 2 つ以上の CPU による同一の共通域フィールドの更新が常に正常に実行できるようにするには、すべての更新をインターロック更新参照を使用して行う必要があります。インターロック更新参照を実行する命令は、COMPARE AND SWAP、COMPARE AND SWAP AND PURGE、COMPARE DOUBLE AND SWAP、および TEST AND SET のみです。例えば、ある CPU が OR IMMEDIATE を実行し、別の CPU が COMPARE AND SWAP を実行して同じバイトを更新しようとした場合、OR IMMEDIATE による取り出しは、COMPARE AND SWAP による取り出しの前に実行されることもあり、COMPARE AND SWAP による取り出しと格納の間に実行されることもあります。さらに、OR IMMEDIATE による格納が、COMPARE AND SWAP による格納の後で実行されることもあり、この場合は COMPARE AND SWAP による変更は失われることとなります。
- 条件コードが 1 に設定される場合、モデルによっては、COMPARE AND SWAP および COMPARE DOUBLE AND SWAP が原因で、第 2 オペランド位置に関して、PER 記憶域変更イベントが認識される、格納に関する記憶保護例外が認識される、変更ビットが 1 に設定される (アクセス例外が存在しないとき)、のいずれかの状態が生じることがあります。記憶域の内容は無変更のままになるので、PER 記憶域変更イベントが認識された場合、変更ビットは 1 になる場合もあり、ならない場合もあります。

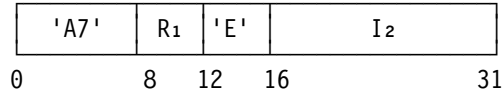
COMPARE HALFWORD

CH $R_1, D_2 (X_2, B_2)$ [RX]

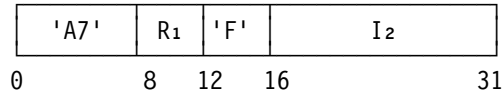


COMPARE HALFWORD IMMEDIATE

CHI R_1, I_2 [RI]



CGHI R_1, I_2 [RI]



第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。第 2 オペランドは、長さが 2 バイトで、16 ビットの符号付き 2 進整数として扱われます。COMPARE HALFWORD および COMPARE HALFWORD IMMEDIATE (CHI) では、第 1 オペランドは 32 ビットの符号付き 2 進整数として扱われます。COMPARE HALFWORD IMMEDIATE (CGHI) では、第 1 オペランドは 64 ビットの符号付き 2 進整数として扱われます。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

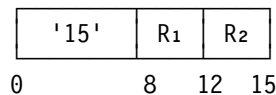
プログラム例外:

- アクセス例外 (取り出し: CH のオペランド 2 のみ)

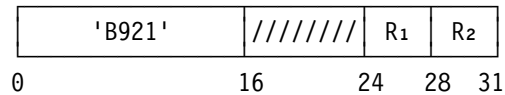
プログラミング上の注意: COMPARE HALFWORD 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。

COMPARE LOGICAL

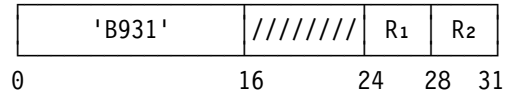
CLR R_1, R_2 [RR]



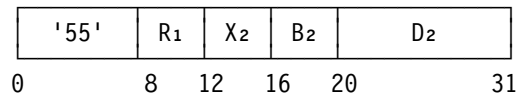
CLGR R₁,R₂ [RRE]



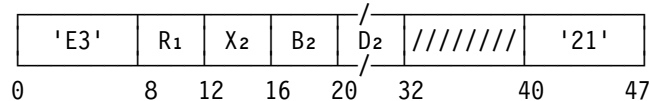
CLGFR R₁,R₂ [RRE]



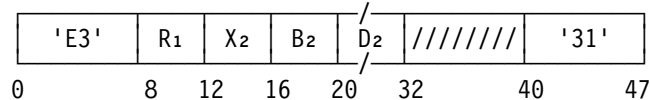
CL R₁,D₂(X₂,B₂) [RX]



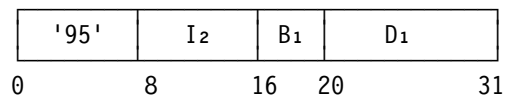
CLG R₁,D₂(X₂,B₂) [RXE]



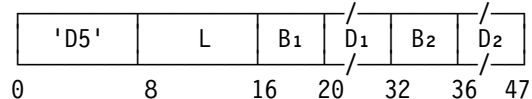
CLGF R₁,D₂(X₂,B₂) [RXE]



CLI D₁(B₁),I₂ [SI]



CLC D₁(L,B₁),D₂(B₂) [SS]



第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。

COMPARE LOGICAL (CLR、CL) では、両オペランドは 32 ビットとして扱われます。COMPARE LOGICAL (CLGR、CLG) では、両オペランドは 64 ビットとして扱われます。COMPARE LOGICAL (CLGFR、CLGF) では、第 1 オペランドは 64 ビットとして扱われ、第 2 オペランドは左に 0 が付加された 32 ビットとして扱われます。

比較は左から右へとバイト単位で行われ、不一致が見つかったとき、またはフィールドの終わりに達したときに終了します。COMPARE LOGICAL (CL、CLG、CLGF、CLC) では、最初の不一致バイトの右の記憶域オペランド部分について、アクセス例外が認識される場合とされない場合があります。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

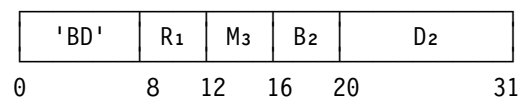
- アクセス例外 (取り出し: オペランド 2、CL、CLG、CLGF、および CLC。取り出し: オペランド 1、CLI および CLC)

プログラミング上の注意:

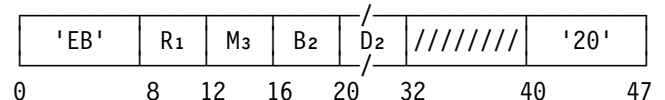
1. COMPARE LOGICAL 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. COMPARE LOGICAL は、各オペランドのすべてのビットを、非構造化論理データのフィールドの一部として、同様に扱います。COMPARE LOGICAL (CLC) では、フィールド長 256 バイトまで、比較を拡張できます。

COMPARE LOGICAL CHARACTERS UNDER MASK

CLM R₁,M₃,D₂(B₂) [RS]



CLMH R₁,M₃,D₂(B₂) [RSE]



マスクの制御下で第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。

M₃ フィールドの内容がマスクとして使用されます。この 4 個のビットは、汎用レジスタ R₁ の中の 4 個の

バイトと、左から右へ 1 対 1 の関係で対応しています。COMPARE LOGICAL CHARACTERS UNDER MASK (CLM) では、マスク・ビットが対応する 4 個のバイトは、汎用レジスタ R₁ のビット位置 32-63 に入っています。COMPARE LOGICAL CHARACTERS UNDER MASK (CLMH) では、この 4 個のバイトはレジスタの高位の半分で、ビット位置 0-31 を占めています。マスク内の 1 に対応するバイト位置の集合が 1 つの連続したフィールドと見なされ、第 2 オペランドと比較されます。第 2 オペランドは、記憶域内の連続フィールドで、第 2 オペランド・アドレスから始まり、マスクの中にある 1 の個数と同じ長さです。マスク内の 0 に対応する汎用レジスタ内のバイトは、操作には関係しません。

比較は左から右へとバイト単位で行われ、不一致が見つかったとき、またはフィールドの終わりに達したときに終了します。

マスクが 0 でない場合、記憶域オペランド・アクセスに関連した例外が認識されるのは、マスクで指定されたバイト数の範囲内です。最初の不一致バイトの右側にある記憶域オペランド部分については、アクセス例外が認識される場合とされない場合があります。マスクが 0 の場合は、第 2 オペランド・アドレスにある 1 バイトについて、アクセス例外が認識されます。

結果の条件コード:

- 0 両オペランドが等しいか、またはマスク・ビットがすべて 0。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)

プログラミング上の注意: COMPARE LOGICAL CHARACTERS UNDER MASK 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。

COMPARE LOGICAL LONG

CLCL R₁,R₂ [RR]

'0F'	R ₁	R ₂
0	8	12 15

第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。短い方のオペランドは、右側が埋め込みバイトで拡張されているものと見なされます。

R₁ フィールドおよび R₂ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 のビット位置 40-63 にある符号なし 2 進整数によって指定されます。汎用レジスタ R₂ + 1 のビット位置 32-39 には、埋め込みバイトが含まれています。汎用レジスタ R₁ + 1 のビット位置 0-39、および汎用レジスタ R₂ + 1 のビット位置 0-31 の内容は、無視されます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱い方は、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

7-45 ページの図 7-7 は、上記で説明したレジスタの内容を示しています。

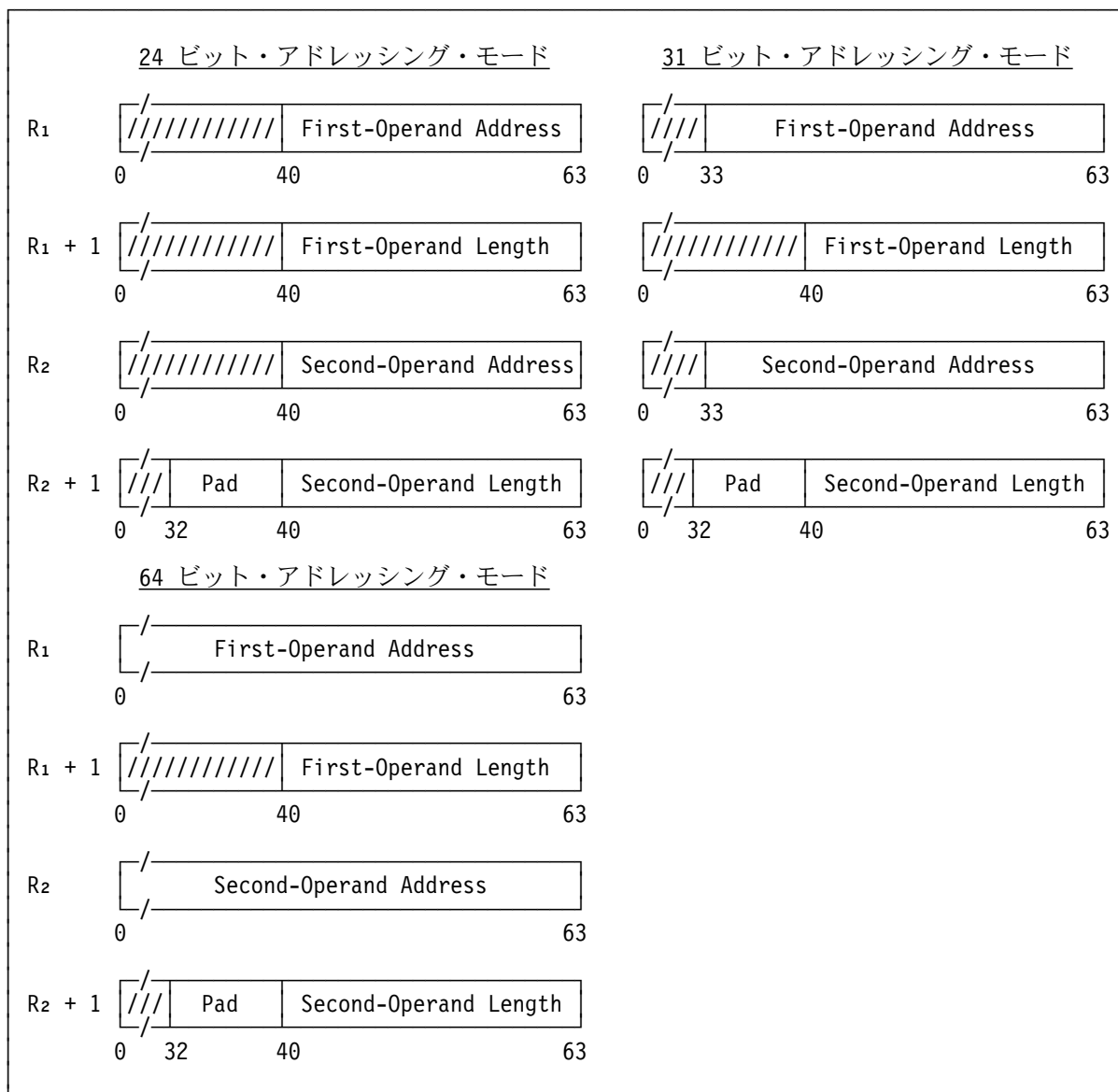


図 7-7. COMPARE LOGICAL LONG で使用するレジスタの内容

比較は左から右へとバイト単位で行われ、不一致が見つかったとき、または長い方のオペランドの終わりに達したときに終了します。2つのオペランドの長さが異なる場合、短い方のオペランドは、右側が適切な数の埋め込みバイトで拡張されているものと見なされます。

両オペランドの長さがどちらも0であるときは、2つのオペランドは等しいと見なされます。

この命令の実行は割り込み可能です。割り込みが起きると、それが操作を中止させる割り込みでない限り、汎用レジスタ R1 + 1 および R2 + 1 内の長さが比較済みのバイト数だけ減少し、汎用レジスタ R1 および R2 内のアドレスが同じバイト数だけ増加しています。これにより、命令の実行が再開されたときは、割り込み

点から処理が再開されます。24ビットまたは31ビットのアドレッシング・モードでは、汎用レジスタ R1 および R2 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは0に設定され、ビット位置 0-31 の内容は変更されません。どのアドレッシング・モードでも、汎用レジスタ R1 + 1 および R2 + 1 のビット位置 0-39 の内容は変更されず、条件コードは予測不能になります。短い方のオペランドが使い尽くされた後で操作に割り込みが起きた場合は、短い方のオペランドに関する長さフィールドは0となり、それに応じてアドレスも更新されます。

不一致が見つかったために操作が終了した場合には、終了時に、汎用レジスタ R1 および R2 のアドレス・フィールドに、各オペランドの中の最初の不一致バイトが

示されています。汎用レジスター R₁ + 1 および R₂ + 1 のビット位置 40-63 の長さは、一致したバイトの数だけ減少しています。ただし、埋め込みバイトにより不一致が起きた場合は、短い方のオペランドの長さフィールドが 0 に設定されます。汎用レジスター R₁ および R₂ の中のアドレスは、それに対応する長さフィールドが減少した量と同じだけ、増加します。

埋め込みバイトがある場合はそれも含めて、2 つのオペランドが一致した場合は、操作の完了時に両方の長さフィールドが 0 に設定され、両方のアドレスが、対応するオペランド長の値と同じだけ増加します。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスター R₁ および R₂ のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、どちらかまたは両方の長さの初期値が 0 の場合でも、0 に設定されます。どのアドレッシング・モードでも、汎用レジスター R₁ + 1 および R₂ + 1 のビット位置 0-39 は変更されません。

最初の不一致バイトの右側にある記憶域オペランド部分については、アクセス例外が認識される場合とされない場合があります。2K バイトより長いオペランドの場合は、処理されるバイトより後にあり、2K バイトを超える部分については、アクセス例外は認識されません。最初の不一致バイトより後の 2K バイトを超える位置については、アクセス例外は示されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。

結果の条件コード:

- 0 両オペランドが等しいか、または両方の長さが 0。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 2)
- 指定例外

プログラミング上の注意:

1. COMPARE LOGICAL LONG 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. R₁ フィールドと R₂ フィールドが同じである場合は、同じ内容を持つ 2 つの別個のレジスター・ペアが指定された場合と同様の方法で操作が進みます。ただし、指定されているレジスターの内容が増加または減少する量は、比較されたバイト数の 2 倍でなく、比較されたバイト数と同数になります。他の CPU またはチャネル・プログラムがオペランド域を動的に変更していなかった場合は、条件コード 0 が設定されます。しかし、記憶域にアクセスせずに操作が完了する場合もあるので、このオペランドについてアクセス例外が認識されるかどうかは予測不能です。
3. COMPARE LOGICAL LONG を EXECUTE のターゲットにする場合は、特別な注意が必要です。EXECUTE の項の『プログラミング上の注意』の、割り込み可能命令についての説明を参照してください。
4. 割り込み可能命令に関するその他のプログラミング上の注意は、第5章、『プログラムの実行』の『割り込み可能命令』に示されています。
5. アクセス・レジスター・モードでは、アクセス・レジスター 0 は、アクセス・レジスター 0 の内容に関係なく、1 次アドレス・スペースを示します。

COMPARE LOGICAL LONG EXTENDED

CLCLE R₁,R₃,D₂(B₂) [RS]

'A9'	R ₁	R ₃	B ₂	D ₂	
0	8	12	16	20	31

第 1 オペランドと第 3 オペランドの比較は、不一致バイトが見つかるか、長い方のオペランドの終わりに達するか、または CPU で決定されるバイト数分の比較が終わるか、いずれかの条件に最初に到達するまで続けられます。短い方のオペランドは、右側が埋め込みバイトで拡張されているものと見なされます。結果は条件コードで示されます。

R₁ フィールドおよび R₃ フィールドは、それぞれ汎用レジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 3 オペランドの左端バイトの位置は、それぞれ、汎用レジスター **R1** および **R3** の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスター **R1 + 1** および **R3 + 1** のビット位置 32-63 の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスター **R1 + 1** および **R3 + 1** 全体の内容で指定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスター **R1** および **R3** のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスター **R1** および **R3** のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスターのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

第 2 オペランド・アドレスはデータのアドレッシングには使用されません。第 2 オペランド・アドレスの右端 8 ビット (ビット 56-63) は、埋め込みバイトです。第 2 オペランド・アドレスのビット 0-55 は無視されます。

7-48 ページの図 7-8 は、上記で説明したレジスターの内容およびアドレスを示しています。

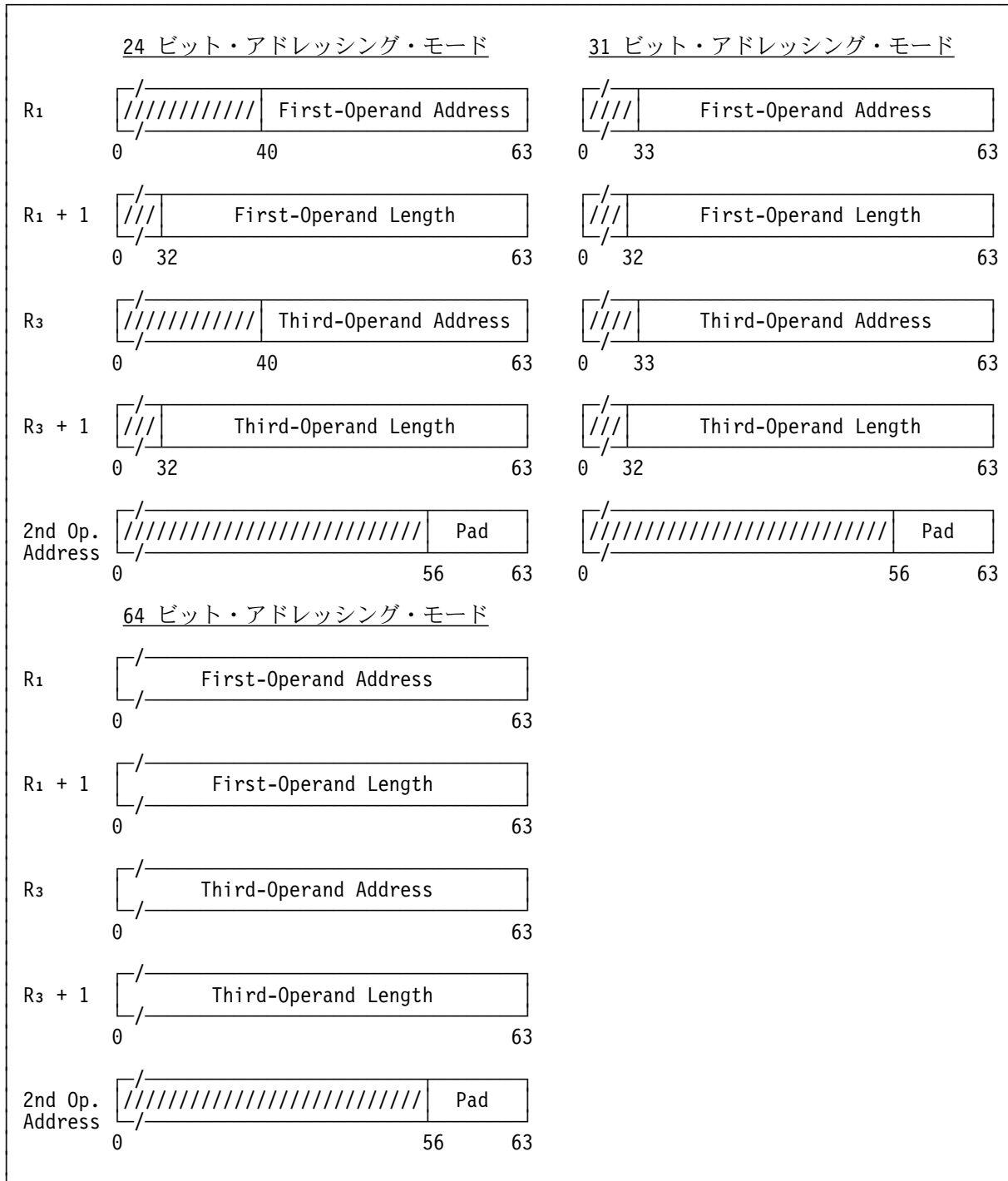


図 7-8. COMPARE LOGICAL LONG EXTENDED で使用するレジスタの内容および第2 オペランド・アドレス

比較は左から右へとバイト単位で行われ、不一致が見つかるか、長い方のオペランドの終わりに達するか、または CPU で決定されるバイト数分の比較が終わるか、いずれか最初に発生した条件によって終了します。2 つのオペランドの長さが異なる場合、短い方のオペランドは、右側が適切な数の埋め込みバイトで拡張されているものと見なされます。

両オペランドの長さがどちらも 0 であるときは、2 つのオペランドは等しいと見なされます。

不一致が見つかって操作が終了した場合には、終了時の汎用レジスタ R1 および R3 のアドレス・フィールドに、各オペランドの中の最初の不一致バイトが示されています。汎用レジスタ R1 + 1 および R3 + 1 のビ

ット位置 32-63 (24 ビットまたは 31 ビットのアドレッシング・モードの場合) またはビット位置 0-63 (64 ビット・アドレッシング・モードの場合) は、一致したバイトの数だけ減少しています。ただし、埋め込みバイトにより不一致が起きた場合は、短い方のオペランドの長さフィールドが 0 に設定されます。汎用レジスタ R₁ および R₃ 内のアドレスは、それに対応する長さフィールドが減少した量と同じだけ、増加します。第 1 オペランドが小さければ条件コード 1 が設定され、第 1 オペランドが大きければ条件コード 2 が設定されます。

埋め込みバイトがある場合はそれも含めて、2 つのオペランドが一致した場合は、操作の完了時に両方の長さフィールドが 0 に設定され、両方のアドレスが、対応するオペランド長の値と同じだけ増加します。そして、条件コード 0 が設定されます。

不一致が見つかることも長い方のオペランドの終わりに達することもなく、CPU で決定された数のバイトの比較が完了した場合は、汎用レジスタ R₁ + 1 および R₃ + 1 内の長さが、比較済みのバイト数だけ減少し、汎用レジスタ R₁ および R₃ 内のアドレスが同じバイト数だけ増加しています。したがって、この命令の再実行時には、次に比較するバイトから処理が再開されます。短い方のオペランドが使い尽くされた後で操作が完了した場合は、短い方のオペランドに関する長さフィールドは 0 となり、それに応じてオペランド・アドレスも更新されます。そして、条件コード 3 が設定されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R₁、R₁ + 1、R₃、および R₃ + 1 のビット位置 0-31 の内容は、常に無変更のままになります。

埋め込みバイトは、命令の実行中に D₂(B₂) から複数回形成される可能性があり、R₁ および R₃ により指定されるレジスタは複数回更新される可能性があります。したがって、B₂ が R₁、R₁ + 1、R₃、または R₃ + 1 に等しく、命令の実行中に変化する可能性がある場合は、結果は予測不能です。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。最大量は、どちらのオペランドでも約 4K バイトです。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスタ R₁ および R₃ のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、片方または両方の長さの初期値が 0 の場合でも、

0 に設定されるか、または元の値のまま変更されません。

最初の不一致バイトの右側にある記憶域オペランド部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、処理対象バイトより後の 4K バイトを超える部分については、アクセス例外は認識されません。最初の不一致バイトより後の 4K バイトを超える位置については、アクセス例外は示されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。

結果の条件コード:

- 0 全バイトを比較。両オペランドが等しいか、または両方の長さが 0。
- 1 全バイトを比較。第 1 オペランドが小さい。
- 2 全バイトを比較。第 1 オペランドが大きい。
- 3 CPU で決定されたバイト数を比較。不一致は見つかりません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 3)
- 指定例外

プログラミング上の注意:

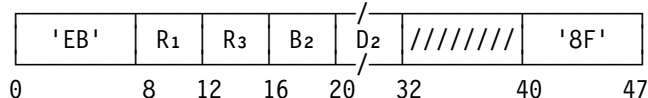
1. COMPARE LOGICAL LONG EXTENDED は、オペランド長が 32 ビット 2 進整数として指定されている場合に、COMPARE LOGICAL LONG の代わりに使用するためのものです。COMPARE LOGICAL LONG なら割り込みが発生するケースで、COMPARE LOGICAL LONG EXTENDED は条件コード 3 を設定します。
2. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、比較処理を継続できます。比較済みのバイト数をプログラムで判別する必要はありません。
3. どちらのオペランドも約 4K バイト以上は処理しないという機能は、長時間の操作の実行中に別の CPU に関してプログラムが設定した可能性のあるフラグを、ソフトウェアでポーリングできるようにするためです。
4. R₁ フィールドと R₃ フィールドが同じ場合は、同じ内容を持つ 2 つの別個のレジスタ・ペアが指定された場合と同様の方法で操作が進みます。ただ

し、指定されているレジスタの内容が増加または減少する量は、比較されたバイト数の 2 倍でなく、比較されたバイト数と同数になります。他の CPU またはチャンネル・プログラムがオペランド域を動的に変更していなかった場合は、条件コードは、多くの場合は 3 に設定された後で、最終的に 0 に設定されます。しかし、記憶域にアクセスせずに操作が完了する場合もあるので、このオペランドについてアクセス例外が認識されるかどうかは予測不能です。記憶域にアクセスしなかった場合、オペランド長に関係なく、条件コード 3 が設定されることもあり、されないこともあります。

5. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。

COMPARE LOGICAL LONG UNICODE

CLCLU R₁,R₃,D₂(B₂) [RSE]



第 1 オペランドと第 3 オペランドの比較は、一致しない 2 バイトの Unicode 文字が見つかるか、長い方のオペランドの終わりに達するか、または CPU で決定される文字数分の比較が終わるか、いずれかの条件に最初に到達するまで続けられます。短い方のオペランドは、右側が 2 バイト埋め込み文字で拡張されているものと見なされます。結果は条件コードで示されます。

R₁ フィールドおよび R₃ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 3 オペランドの左端文字の位置は、それぞれ、汎用レジスタ R₁ および R₃ の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₃ + 1 のビット位置 32-63 の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₃ + 1 のビット位置 0-63 の内容で指

定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ R₁ + 1 および R₃ + 1 の内容は、偶数個のバイトを示している必要があります。さもないと、指定例外が認識されます。

汎用レジスタ R₁ および R₃ のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₃ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、レジスタのビット位置 0-63 の内容がアドレスを形成します。

第 2 オペランド・アドレスはデータのアドレッシングには使用されません。第 2 オペランド・アドレスの右端 16 ビット (ビット 48-63) は、2 バイト埋め込み文字です。第 2 オペランド・アドレスのビット 0-47 は無視されます。

7-51 ページの図 7-9 は、上記で説明したレジスタの内容およびアドレスを示しています。

比較は左から右へと文字単位で行われ、不一致が見つかるか、長い方のオペランドの終わりに達するか、または CPU で決定される文字数分の比較が終わるか、いずれか最初に発生した条件によって終了します。2 つのオペランドの長さが異なる場合、短い方のオペランドは、右側が適切な数の 2 バイト埋め込み文字で拡張されているものと見なされます。

両オペランドの長さがどちらも 0 であるときは、2 つのオペランドは等しいと見なされます。

不一致が見つかったために操作が終了した場合には、終了時に、汎用レジスタ R₁ および R₃ のアドレス・フィールドに、各オペランドの中にある最初の不一致の 2 バイト文字が示されています。汎用レジスタ R₁ + 1 および R₃ + 1 のビット位置 32-63 (24 ビットまたは 31 ビットのアドレッシング・モードの場合) またはビット位置 0-63 (64 ビット・アドレッシング・モードの場合) は、一致した文字数の 2 倍だけ減少しています。ただし、2 バイト埋め込み文字により不一致が起きた場合は、短い方のオペランドの長さフィールドが 0 に設定されます。汎用レジスタ R₁ および R₃ 内のアドレスは、それに対応する長さフィールドが減少した量と同じ

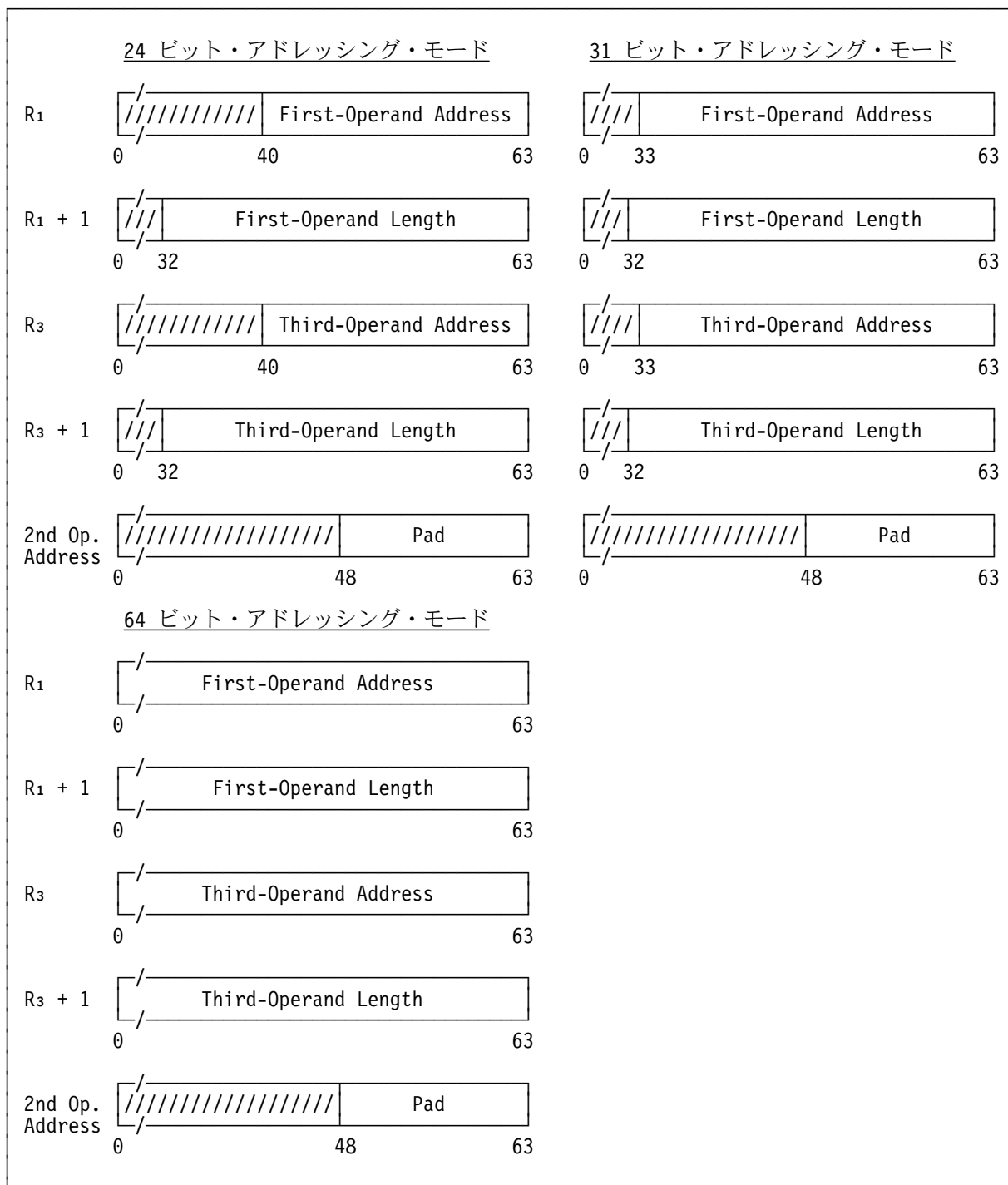


図 7-9. COMPARE LOGICAL LONG UNICODE で使用するレジスタの内容および第 2 オペランド・アドレス

だけ、増加します。第 1 オペランドが小さければ条件コード 1 が設定され、第 1 オペランドが大きければ条件コード 2 が設定されます。

2 バイト埋め込み文字がある場合はそれも含めて、2 つのオペランドが一致した場合は、操作の完了時に両方の長さフィールドが 0 に設定され、両方のアドレスが、対

応するオペランド長の値と同じだけ増加します。そして、条件コード 0 が設定されます。

不一致が見つかることも長い方のオペランドの終わりに達することなく、CPU で決定された数の文字の比較が完了した場合は、汎用レジスタ R1 + 1 および R3 + 1 内の長さが、比較済みの文字数の 2 倍だけ減少

し、汎用レジスター **R1** および **R3** 内のアドレスが同じ文字数だけ増加します。したがって、この命令の再実行時には、次に比較する文字から処理が再開されます。短い方のオペランドが使い尽くされた後で操作が完了した場合は、短い方のオペランドに関する長さフィールドは **0** となり、それに応じてオペランド・アドレスも更新されます。そして、条件コード **3** が設定されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスター **R1**、**R1 + 1**、**R2**、および **R2 + 1** のビット位置 0-31 の内容は、常に無変更のままになります。

2 バイト埋め込み文字は、命令の実行中に **D2(B2)** から複数回形成される可能性があり、**R1** および **R3** により指定されるレジスターは複数回更新される可能性があります。したがって、**B2** が **R1**、**R1 + 1**、**R3**、または **R3 + 1** に等しく、命令の実行中に変化する可能性がある場合は、結果は予測不能です。

条件コード **3** が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスター **R1** および **R3** のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、片方または両方の長さの初期値が **0** の場合でも、**0** に設定されるか、または元の値のままに変更されません。

最初の不一致文字の右側にある記憶域オペランド部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、処理対象文字より後の 4K バイトを超える部分については、アクセス例外は認識されません。最初の不一致文字より後の 4K バイトを超える位置については、アクセス例外は示されません。

オペランドの長さが **0** の場合、そのオペランドについてはアクセス例外は認識されません。オペランドに関連した **R** フィールドまたは長さが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。

結果の条件コード:

- 0 全文字を比較。両オペランドが等しいか、または両方の長さが **0**。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。

- 3 CPU で決定された文字数を比較。不一致は見つからない。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 3)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. COMPARE LOGICAL LONG UNICODE は、2 バイト文字を比較する場合に、COMPARE LOGICAL LONG または COMPARE LOGICAL LONG EXTENDED の代わりに使用するためのものです。比較する文字は、Unicode 文字の場合も、他の 2 バイト文字の場合もあります。COMPARE LOGICAL LONG なら割り込みが発生するケースで、COMPARE LOGICAL LONG UNICODE は条件コード **3** を設定します。
2. 条件コード **3** が設定されると、プログラムはブランチに従って命令に戻り、比較を続けることができます。比較済みの文字数をプログラムで判別する必要はありません。
3. **R1** フィールドと **R3** フィールドが同じ場合は、同じ内容を持つ 2 つの別個のレジスター・ペアが指定された場合と同様の方法で操作が進みます。ただし、指定されているレジスターの内容が増加または減少する量は、比較された文字数の 2 倍であって、比較されたバイト数の 4 倍ではありません。他の CPU またはチャネル・プログラムがオペランド域を動的に変更していなかった場合は、条件コードは、多くの場合は **3** に設定された後で、最終的に **0** に設定されます。しかし、記憶域にアクセスせずに操作が完了する場合もあるので、このオペランドについてアクセス例外が認識されるかどうかは予測不能です。記憶域にアクセスしなかった場合、オペランド長に関係なく、条件コード **3** が設定されることもあり、されないこともあります。
4. アクセス・レジスター・モードでは、アクセス・レジスター **0** は、アクセス・レジスター **0** の内容に関係なく、1 次アドレス・スペースを示します。
5. Unicode のスペース文字 (または表記が 16 進数 FFF 以下の任意の文字) による埋め込みが必要な場合は、命令の変位フィールドにこの文字を次の例のように表記することができます。

```
CLCLU 6,8,X'020'
```

COMPARE LOGICAL STRING

CLST R₁,R₂ [RRE]

'B25D'	////////	R ₁	R ₂
0	16	24	28 31

第 1 オペランドと第 2 オペランドの比較は、不一致バイトが見つかるか、どちらかのオペランドの終わりに達するか、または CPU で決定されるバイト数分の比較が終わるか、いずれかの条件に最初に到達するまで続けられます。CPU で決定される数の最小値は 256 です。結果は条件コードで示されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

第 1 オペランドおよび第 2 オペランドは、長さが同じであっても異なっていても構いません。オペランドの終わりは、そのオペランドの最後のバイト位置にある終了文字で指示します。オペランドの終わりを判別するために使用する終了文字は、汎用レジスタ 0 のビット位置 56-63 で指定します。汎用レジスタ 0 のビット位置 32-55 は、将来の拡張に備えて予約されており、すべて 0 が含まれていなければなりません。さもないと、指定例外が認識されます。

操作は左から右へとバイト単位で行われ、どちらか一方、または両方のオペランドの中で終了文字が検出されるか、終了文字以外の不一致バイトが見つかるか、または CPU で決定されるバイト数分の比較が終わるか、いずれか最初に発生した条件によって終了します。CPU で決定される数の最小値は 256 です。両方のオペランドで同時に終了文字が検出された場合は、それがオペランドの先頭バイト位置であった場合も含めて、両方のオペランドは同じ長さであり、等しいと見なされ、条件コード 0 が設定されます。片方のオペランドの中でのみ終

了文字が検出された場合は、そのオペランド (短い方のオペランド) が小さいと見なされ、条件コード 1 または 2 が設定されます。第 1 オペランドが小さければ条件コード 1 が設定され、第 2 オペランドが小さければ条件コード 2 が設定されます。同様に、比較により終了文字以外の不一致バイトが見つかったとき、低位の方のバイトが第 1 オペランドの中にあれば条件コード 1 が設定され、低位の方のバイトが第 2 オペランドの中にあれば条件コード 2 が設定されます。CPU で決定されたバイト数の比較が終わった場合には、条件コード 3 が設定されます。

条件コード 1 または 2 が設定された場合、第 1 オペランドおよび第 2 オペランド内で処理された最後のバイトのアドレスが、それぞれ汎用レジスタ R₁ および R₂ に入れます。つまり、条件コード 1 が設定された場合は、第 1 オペランド内の終了文字または最初の不一致バイト (どちらか検出された方) のアドレスが汎用レジスタ R₁ に入れられ、その第 1 オペランド・バイトに対応する位置にある第 2 オペランド・バイトのアドレスが、汎用レジスタ R₂ に入れられます。条件コード 2 が設定された場合は、第 2 オペランド内の終了文字または最初の不一致バイト (どちらか検出された方) のアドレスが汎用レジスタ R₂ に入れられ、その第 2 オペランド・バイトに対応する位置にある第 1 オペランド・バイトのアドレスが、汎用レジスタ R₁ に入れられます。条件コード 3 が設定された場合は、第 1 オペランドおよび第 2 オペランド内で次に処理されるバイトのアドレスが、それぞれ汎用レジスタ R₁ および R₂ に入れられます。汎用レジスタにアドレスが入れるたびに、そのレジスタのビット 32-39 (24 ビット・アドレッシング・モードの場合) またはビット 32 (31 ビット・アドレッシング・モードの場合) が 0 に設定されます。R₁ および R₂ レジスタのビット 0-31 は、24 ビット・モードの場合も 31 ビット・モードの場合も変更されません。

条件コード 0 が設定された場合、汎用レジスタ R₁ および R₂ の内容は変更されません。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

第 1 オペランドおよび第 2 オペランドに関するアクセス例外が認識されるのは、実行中の操作で必ず検査されるオペランド部分についてのみです。

結果の条件コード:

- 0 両オペランド全体が等しい。汎用レジスタ R₁ および R₂ は変更されません。
- 1 第 1 オペランドが小さい。汎用レジスタ R₁ および R₂ は、処理された最後のバイトのアドレスで更新されています。
- 2 第 1 オペランドが大きい。汎用レジスタ R₁ および R₂ は、処理された最後のバイトのアドレスで更新されています。
- 3 CPU で決定された数のバイトが等しい。汎用レジスタ R₁ および R₂ は、次のバイトのアドレスで更新されています。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 2)
- 指定例外

プログラミング上の注意:

1. COMPARE LOGICAL STRING 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 条件コード 0 が設定された場合、どちらの終了文字についても、その位置は示されません。
3. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、比較処理を継続できます。比較済みのバイト数をプログラムで判別する必要はありません。
4. R₁ または R₂ が 0 の場合もあります。この場合、汎用レジスタ 0 は、アドレスのほかに終了文字も含んでいるものとして扱われます。
5. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。

COMPARE UNTIL SUBSTRING EQUAL

CUSE R₁,R₂ [RRE]

'B257'	////////	R ₁	R ₂
0	16	24	28 31

第 1 オペランドと第 2 オペランドの比較は、指定の長さを持つ等しいサブstring (一連のバイト) が見つかるか、長い方のオペランドの終わりに達するか、または CPU で決定される不一致バイト数の比較が終わるか、いずれかの条件に最初に到達するまで続けられます。短い方のオペランドは、右側が埋め込みバイトで拡張されているものと見なされます。CPU で決定される数の最小値は 256 です。結果は条件コードで示されません。

R₁ フィールドおよび R₂ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 のビット位置 32-63 の 32 ビットの符号なし 2 進整数によって指定されます。64 ビット・アドレッシング・モードでは、バイト数は、これらのレジスタのビット位置 0-63 の、64 ビットの 2 進整数によって指定されます。オペランド長が負の値の場合は、0 として扱われ、命令の完了時には変更されていません。

汎用レジスタ 0 のビット 56-63 は、符号なしのサブstringの長さをバイト数で指定します (値は 0-255)。汎用レジスタ 1 のビット 56-63 は埋め込みバイトです。汎用レジスタ 0 および 1 のビット 0-55 は無視されます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いには、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

7-55ページの図7-10は、上記で説明したレジスタの内容を示しています。

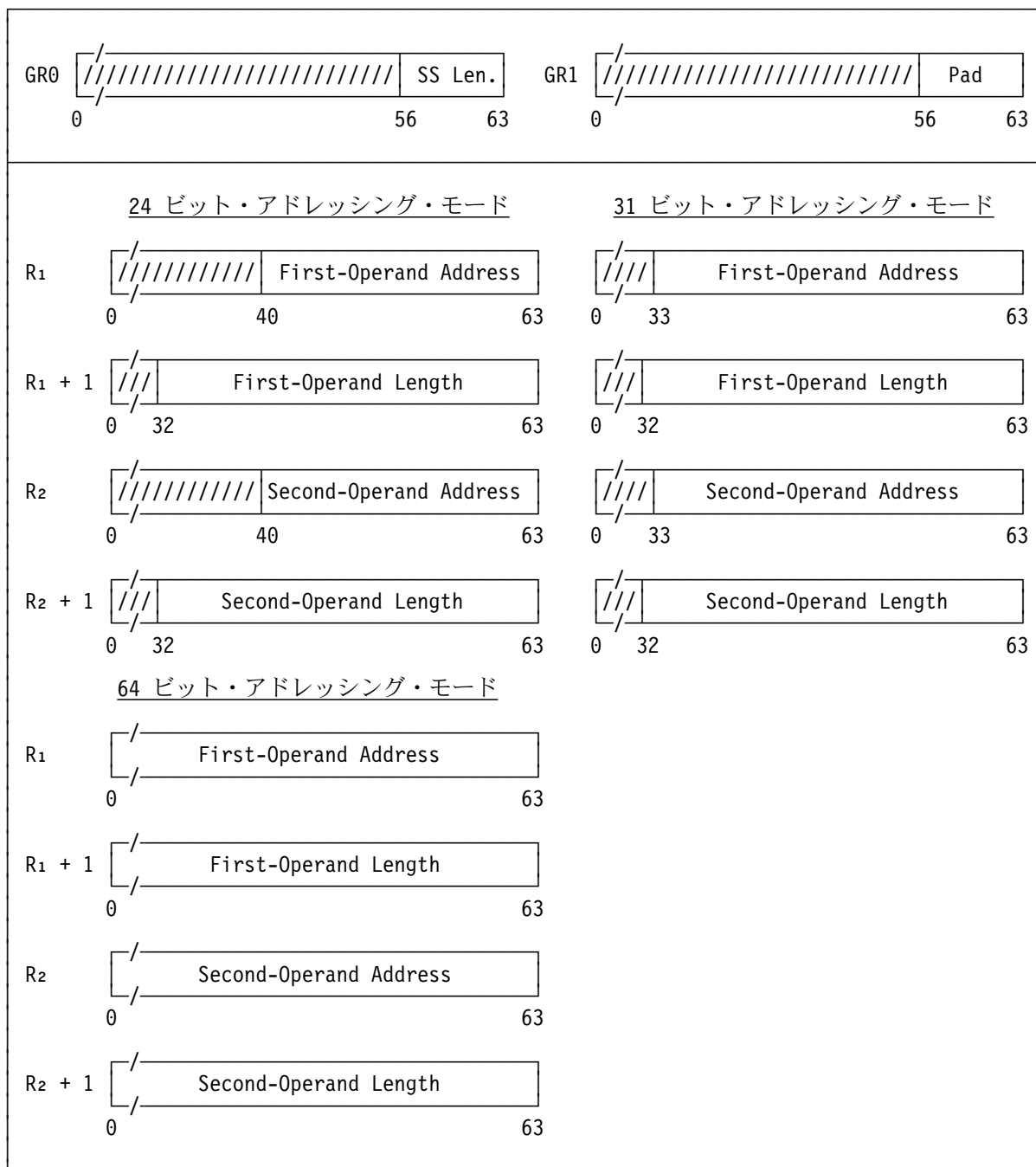


図 7-10. `COMPARE UNTIL SUBSTRING EQUAL` で使用するレジスターの内容

結果は、両方のオペランドを左から右へ処理したものと
して求められます。ただし、各オペランドのすべてまたは
一部のバイトに対して、複数のアクセスが行われること
があります。

比較は左から右へとバイト単位で行われ、以下のいずれ
かの条件が成立したときに終了します。(1) 指定された
長さの、一致サブストリングが見つかった。(2) 長い方
のオペランドの終わりに達したが、指定された長さの一
致サブストリングが見つからなかった。または、

(3) CPU で決定されるバイト数の比較が完了したが、
比較された最後のバイトが一致しなかった。CPU で決
定される数の最小値は 256 です。2 つのオペランドの長
さが異なる場合、短い方のオペランドは、右側が適切
な数の埋め込みバイトで拡張されているものと見なされ
ます。

指定の長さの一致サブストリングが見つかったために操
作が終了した場合は、条件コードが 0 に設定されます。
長い方のオペランドの終わりに達しても指定の長さの一

致サブストリングが見つからなかったために操作が終了した場合は、最後に比較されたバイトが一致バイトなら条件コードが 1 に設定され、最後に比較されたバイトが不一致バイトなら条件コードが 2 に設定されます。

CPU で決定されるバイト数の比較が完了し、不一致バイトの比較で操作が終了した場合は、条件コードが 3 に設定されます。

指定されたサブストリング長が 0 の場合は、指定長の一致サブストリングが見つかったものと見なされ、条件コード 0 が設定されます。

両オペランドの長さが 0 で、指定されたサブストリング長が 0 ではない場合は、比較された最後のバイトが不一致バイトだったときに長い方のオペランドの終わりに達したと見なされ、条件コード 2 が設定されます。

すでに比較されたものがすべて一致バイトだったとしても、その後の比較で不一致バイトが見つかった場合は、その時点までに比較されたすべてのバイトが不一致と見なされます。

操作の完了時に、R1 + 1 および R2 + 1 レジスタのオペランド長フィールドは、比較された不一致バイトの数 (比較された不一致バイトの前にある一致バイトも含む) だけ減少し、R1 および R2 レジスタ内のアドレスが同じ数だけ増加します。ただし、長い方のオペランドが埋め込みバイトと比較された場合は、短い方のオペランドの長さフィールドが 0 以下に減少することはない、対応するアドレスも、短い方のオペランドの後の最初のバイトのアドレスより大きい値に増加することはありません。24 ビットまたは 31 ビットのアドレッシング・モードでは、レジスタ R1 および R2 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、サブストリング長が 0 の場合、または両方のオペランド長の初期値が 0 の場合でも、0 に設定されます。

したがって、条件コード 0 または 1 が設定されたとき、R1 および R2 レジスタに入っている結果のアドレスが 2 つのオペランド内の一致サブストリングの最初のバイトを示し、R1 + 1 および R2 + 1 レジスタに入っている長さが、その一致サブストリングの前にあるバイト数だけ減少しています。ただし、短い方のオペランド内の一致サブストリングが埋め込みバイトで始まっている場合は、短い方のオペランドの長さフィールドは 0 になり、対応するアドレス・フィールドは短い方のオペランドの長さ分だけ増加しています。条件コード 2 が設定されたときは、各アドレス・フィールドは対応するオペランドの後の最初のバイトを示し、両方の長さフィールドは 0 になります。条件コード 3 が設定されたときは、各アドレス・フィールドは対応するオペランドの最後に比較されたバイトの後の最初のバイトを示し、両

方の長さフィールドは、比較されたバイト数だけ減少します。ただし、長さフィールドが 0 より小さくなることはありません。

R1 および R2 フィールドの内容が同じであるときは、第 1 オペランドと第 2 オペランドが比較される場合と、オペランドの比較を行わずに条件コードが 0 または 1 に設定される場合があります。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R1、R1 + 1、R2、および R2 + 1 のビット位置 0-31 の内容は、常に無変更のままになります。

サブストリング長および埋め込みバイトは、命令の実行中に汎用レジスタ 0 および 1 から複数回取り出される可能性があり、R1 および R2 により指定されるレジスタは複数回更新される可能性があります。したがって、R1 または R2 が 0 のときは、結果は予測不能です。

条件コード 3 が設定された場合、命令が使用している汎用レジスタは、ブランチで戻って同じ命令を再実行しさえすればオペランドの残り部分を処理できるように設定されています。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

命令の実行は、比較された最後のバイトが不一致のときに割り込み可能になります。最後のバイトが一致した場合は、割り込み可能にはなりません。割り込みが起きると、それが操作を中止させる割り込みでない限り、R1 フィールドおよび R2 フィールドによって指定されるレジスタの内容は、通常の命令完了時と同様に更新されます。その結果、この命令の実行が再開されたときは、割り込み点から処理が再開されます。条件コードは予測不能です。

処理された最後のバイトの右側にある記憶域オペランド部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、最後の処理バイトより後の 4K バイトを超える記憶位置については、アクセス例外は認識されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。サブストリング長として 0 が指定されている場合はオペランドのアドレ

ス・フィールドおよび長さフィールドは変更されませんが、アクセス例外の認識は必ずしも防止できません。

結果の条件コード:

- 0 指定の長さの一致サブstringが見つかりました。
- 1 長い方のオペランドの終わりに達し、比較された最後のバイトが一致しています。
- 2 長い方のオペランドの終わりに達し、比較された最後のバイトが一致していません。
- 3 CPU で決定されたバイト数の比較が完了し、比較された最後のバイトが一致していません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1 および 2)
- 指定例外

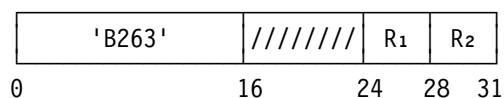
プログラミング上の注意:

1. R1 フィールドと R2 フィールドが同じ場合は、同じ内容を持つ 2 つの別個のレジスター・ペアが指定された場合と同様の方法で操作が進みます。そして、別の CPU またはチャンネル・プログラムによるオペランド域の動的変更がない場合には、条件コード 0、1、または 2 が設定されます (次の注 2 を参照)。しかし、記憶域にアクセスせずに操作が完了する場合もあるので、このオペランドについてアクセス例外が認識されるかどうかは予測不能です。
2. R1 フィールドと R2 フィールドの内容が同じで、オペランド長が 0 でなく、別の CPU またはチャンネル・プログラムがオペランドを変更中でない場合は、次のようになります。オペランド長が指定されたサブstring長に等しいかそれより長いときは、条件コード 0 が設定されます。オペランド長が指定されたサブstring長より短いときは、条件コード 1 が設定されます。両方のオペランド長が 0 の場合は、R1 が R2 に等しいかどうかに関係なく次のようになります。指定されたサブstring長が 0 のときは、条件コード 0 が設定され、指定されたサブstring長が 0 以外のときは、条件コード 2 が設定されます。どの場合も、R1 および R2 レジスター内のアドレスと、R1 + 1 および R2 + 1 レジスター内の長さは、変更されません。
3. COMPARE UNTIL SUBSTRING EQUAL を EXECUTE のターゲットにする場合は、特別な注意が必要です。EXECUTE の項の『プログラミング上の注意』の、割り込み可能命令についての説明を参照してください。

4. 割り込み可能命令に関するその他の詳しいプログラミング上の注意は、5-20ページの『割り込み可能命令』に示されています。
5. アクセス・レジスター・モードでは、アクセス・レジスター 0 は、アクセス・レジスター 0 の内容に関係なく、1 次アドレス・スペースを示します。
6. COMPARE UNTIL SUBSTRING EQUAL の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

COMPRESSION CALL

CMPSC R1,R2 [RRE]



この説明では、「Enterprise Systems Architecture/390 Data Compression」(SA22-7208-01) に示されている概説および辞書フォーマットの情報についての知識があることが前提になっています。

第 2 オペランドは、汎用レジスター 0 の指定に応じて圧縮または展開され、結果が第 1 オペランド位置に入れます。圧縮データ・オペランドは、通常、汎用レジスター 1 内のアドレスによって指定される辞書の中のエントリーに対応する指標記号からなっています。この辞書は、圧縮操作中は圧縮辞書であり、展開操作中は展開辞書です。汎用レジスター 0 にフォーマット 1 の兄弟記述子が指定されている圧縮時には、圧縮辞書の直後に展開辞書が続いています。汎用レジスター 0 に記号変換オプションが指定されているときの圧縮時には、圧縮の結果得られる指標記号は、汎用レジスター 1 内のアドレスおよびオフセットによって指定される記号変換テーブルを使用して交換記号に変換されます。この交換記号が、第 1 オペランド位置に入れます。圧縮データ・オペランド内の記号のビット数は、汎用レジスター 0 で指定されます。操作は、どちらかのオペランドの終わりに達するか、CPU で決定されるデータ量の処理が済むか、どちらかの条件が先に発生するまで続けられます。結果は条件コードで示されます。

R1 フィールドおよび R2 フィールドは、それぞれ汎用レジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドのビットのどれかを含む左端バイトの位置は、それぞれ、汎用レジスター R₁ および R₂ 内のアドレスによって指定されます。第 1 オペランドとおよび第 2 オペランドのビットを含むバイトの数は、それぞれ、汎用レジスター R₁ + 1 および R₂ + 1 のビット 32-63 (24 ビットまたは 31 ビットのアドレッシング・モードの場合)、またはこれらのレジスターのビット 0-63 (64 ビット・アドレッシング・モードの場合) によって指定されます。汎用レジスター R₁ + 1 および R₂ + 1 の内容は、24 ビットまたは 31 ビット・アドレッシング・モードでは 32 ビットの符号なし 2 進整数として、64 ビット・アドレッシング・モードでは 64 ビットの符号なし 2 進整数として扱われます。

圧縮時の圧縮辞書または展開時の展開辞書の左端バイトの位置は、汎用レジスター 1 内のアドレスによって 4K バイト境界上に指定されます。

汎用レジスター R₁、R₂、および 1 内のアドレスの取り扱い、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、レジスター R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスター R₁ および R₂ のビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、レジスター R₁ および R₂ のビット位置 0-63 の内容がアドレスを形成します。24 ビット・アドレッシング・モードでは、レジスター 1 のビット位置 40-51 の内容の右側に 12 個の 0 を付加したものがアドレスを形成し、ビット位置 0-39 および 52-63 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスター 1 のビット位置 33-51 の内容の右側に 12 個の 0 を付加したものがアドレスを形成し、ビット位置 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、レジスター 1 のビット位置 0-51 の内容の右側に 12 個の 0 を付加したものがアドレスを形成し、ビット 52-63 は無視されます。

汎用レジスター 1 のビット位置 52-63 の内容は、今説明したように無視されますが、この内容は次のように使用されます。このレジスターのビット位置 61-63 は、圧縮データ・ビット番号 (CBN) です。操作の開始時には、CBN は圧縮データ・オペランドの左端バイト内の左端ビットを指しています。圧縮データ・オペランドは、圧縮時は第 1 オペランドで、展開時は第 2 オペランドです。圧縮時に記号変換オプションが指定されてい

るときは、レジスターのビット位置 52-60 の内容の右側に 7 個の 0 が付加されたものが、圧縮辞書の先頭から記号変換テーブルの左端バイトまでのバイト・オフセットになります。展開時には記号変換は指定できません。展開時には、ビット位置 52-60 の内容は無視されます。

7-59ページの図7-11 は、上記で説明したレジスターの内容、および汎用レジスター 0 の内容を示しています。

汎用レジスター 0 のビット 55 (E) が 0 なら圧縮操作を示し、1 なら展開操作を示します。

汎用レジスター 0 のビット 47 (ST) は、記号変換オプション・ビットです。ビット 47 が 0 のときの圧縮操作では、文字ストリングを表す圧縮辞書エントリーに対する指標記号と呼ばれる指標が生成され、それらの指標記号が圧縮データ・オペランドに入れられます。ビット 47 が 1 のときの圧縮操作では、指標記号は生成されませんが、その後で、その指標記号が交換記号と呼ばれる別の記号に変換され、交換記号が圧縮データ・オペランドに入れられます。この記号変換は、汎用レジスター 1 中のアドレスおよびオフセットで指定された記号変換テーブルを使用して行われます。展開時には、ビット 47 と、汎用レジスター 1 内のオフセットは無視されます。展開時には、圧縮データ・オペランドには常に、展開辞書のエントリーを指す指標記号が含まれています。

汎用レジスター 0 のビット 48-51 (CDSS) は、図に示すように、圧縮データ・オペランド内の指標記号または交換記号のビット数を示します。ビット 48-51 の値は、2 進数 0000 または 0110-1111 であってはなりません。これらのいずれかの値のときは、指定例外が認識されます。記号変換が指定されていないときは、ビット 48-51 は、図に示すように、圧縮辞書および展開辞書のそれぞれの中の 8 バイトのエントリーの数も指定し、したがって、それぞれの辞書のサイズをバイト数で指定することになります。記号変換が指定されているときは、圧縮辞書は記号変換テーブルの先頭まで拡張されるものと見なされます。つまり、圧縮辞書のサイズ (バイト数) は、汎用レジスター 1 のビット位置 52-60 内のオフセットの右側に 7 個の 0 を付加したものになります。記号変換テーブルのサイズ (バイト数) は、圧縮辞書のサイズの 1/4 と見なされます。ただし、汎用レジスター 1 内のオフセットは、少なくとも、記号変換が指定されおらず CDSS が実際のサイズより 1 小さかった場合の圧縮辞書のサイズと同じでなければならず、CDSS が 2 進数 0001 の場合は、オフセットは最小でも 2K バイトでなければなりません。それ以外の場合は、結果は予測不能です。例えば、CDSS が 0101 であれば、オフセットは少なくとも 32K バイトでなければなりません。

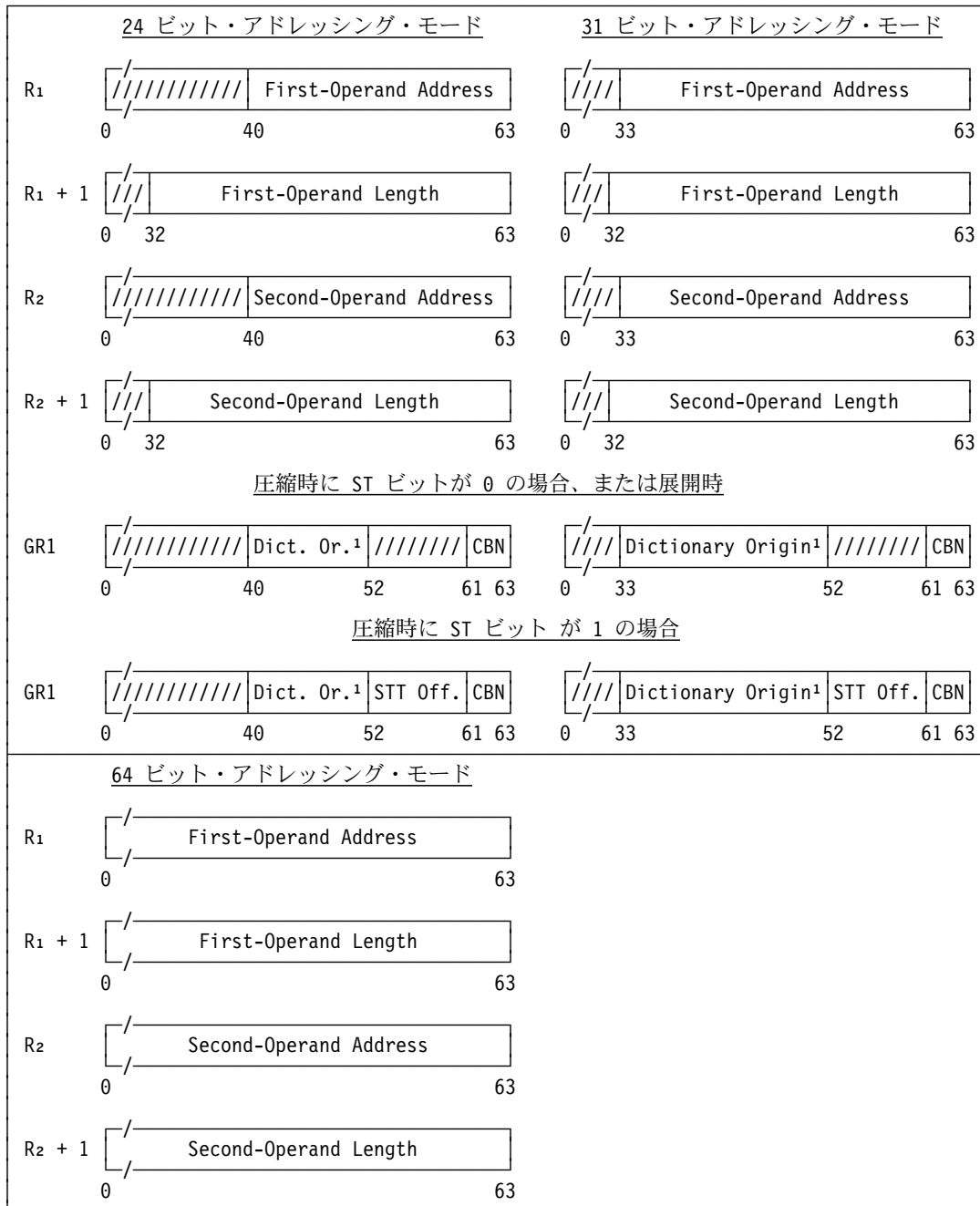


図 7-11 (1/2). COMPRESSION CALL で使用するレジスタの内容

汎用レジスタ 0 のビット 54 (F1) は、0 のときは圧縮辞書にフォーマット 0 の兄弟記述子が含まれることを指定し、1 のときはフォーマット 1 の兄弟記述子が含まれることを指定します。兄弟記述子は、圧縮操作時に使用されます。フォーマット 0 兄弟記述子は 8 バイトで、圧縮辞書の指標位置に入っています。フォーマット 1 兄弟記述子は 16 バイトで、最初の 8 バイトは圧縮辞書の指標位置に、2 番目の 8 バイトは展開辞書の同じ指標位置に入っています。ビット 54 が 1 のときの圧縮時には、展開辞書は、汎用レジスタ 1 内のアドレスで指

定されている圧縮辞書の直後に続くものと見なされません。ビット 54 は、展開時には無視されます。

汎用レジスタ 0 のビット 47 および 54 は両方共 1 であってはなりません。1 の場合、結果は予測不能です。

汎用レジスタ 0 の未使用のビット位置は、将来の拡張に備えて予約されているもので、ここには 0 が含まれていなければなりません。さもないと、将来プログラムの動作の互換性が失われることがあります。

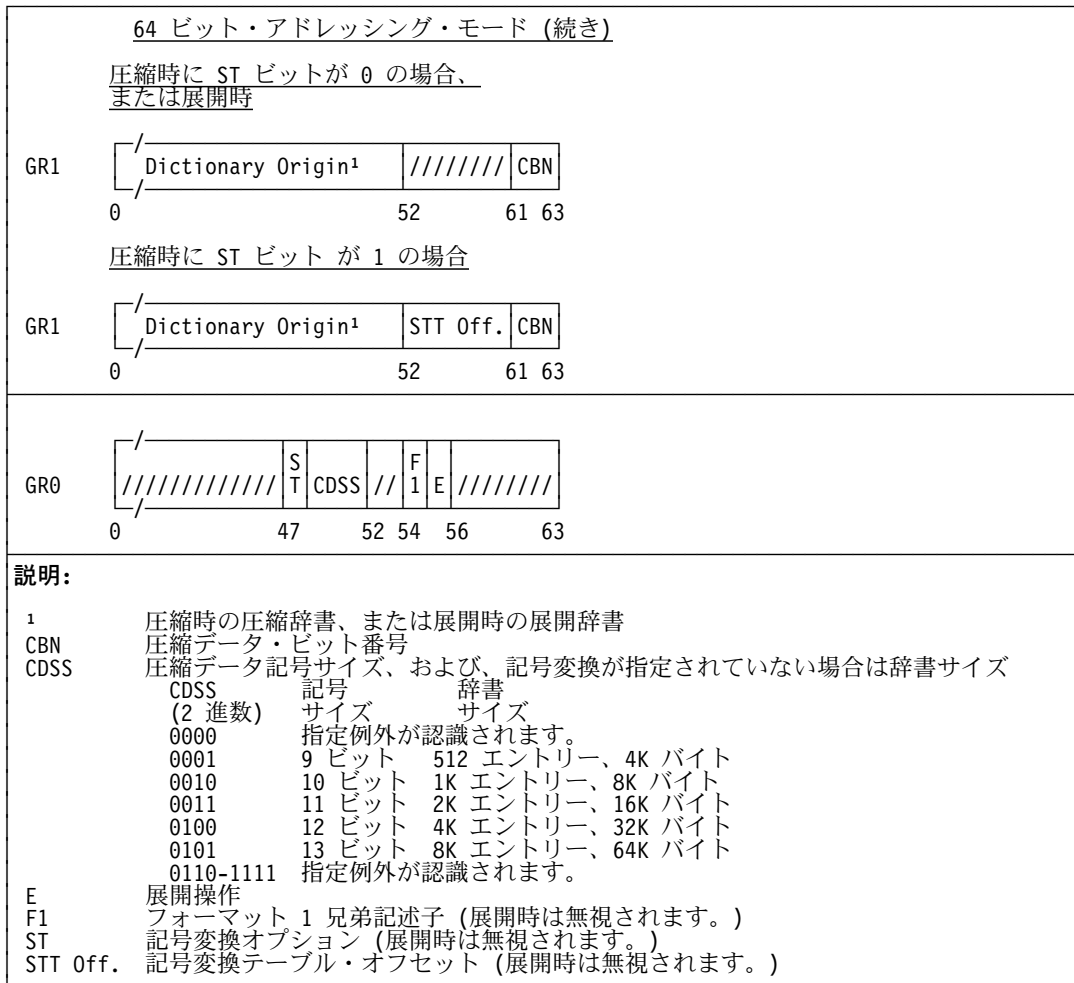


図 7-11 (2/2). COMPRESSION CALL で使用するレジスターの内容

アクセス・レジスター・モードでは、アクセス・レジスター R₁ の内容は第 1 オペランドにアクセスするために使用され、アクセス・レジスター R₂ の内容は、第 2 オペランド、辞書、および記号変換テーブルにアクセスするために使用されます。

操作は両オペランドの左端から開始され、右へと進みません。操作は、どちらかのオペランドの終わりに達するか、または CPU で決定されるデータ量の処理が完了するか、どちらかの条件が先に発生したときに終了します。

圧縮操作時には、第 1 オペランド位置に残っている未使用ビット位置の数が、追加の圧縮データを収容するには不十分になった時点で、第 1 オペランドの終わりに達したと見なされます。

展開操作時には、次の 2 つの条件のどちらかが起きたときに、第 1 オペランドの終わりに達したと見なされます。

1. 第 1 オペランド位置に残っている未使用バイト位置の数が、次の指標記号を展開した結果生じるすべての文字数を収容するには不十分になったとき。
2. 未使用バイト位置の数が 0 になった直後。つまり、1 つの指標記号を展開した結果、第 1 オペランド位置が完全に埋まった直後。

展開操作時には、次の順番の指標記号が第 2 オペランド位置の中に完全に入らない場合には、第 2 オペランド位置の終わりに達したものと見なされます。汎用レジスター 1 のビット位置 61-63 にある圧縮データ・ビット番号に関係なく、汎用レジスター R₂ 内のアドレスと汎用レジスター R₂ + 1 内の長さの合計によって指定されるバイトの先頭で、第 2 オペランド位置は終了します。

第 2 オペランドの終わりに達したために操作が終了したときは、条件コード 0 が設定されます。第 1 オペランドの終わりに達したために操作が終了したときは、条件コード 1 が設定されます。ただし、第 2 オペランドの

終わりにも同時に達した場合は、条件コード 0 が設定されます。CPU で決定されるデータ量の処理が完了したために操作が終了したときは、条件コード 3 が設定されます。

操作の完了時には、汎用レジスター $R_1 + 1$ 内の長さは、第 1 オペランド位置に格納されている完全バイトの数だけ減少し、汎用レジスター R_1 内のアドレスは同じバイト数だけ増加しています。圧縮時には、あるバイトのすべてのビット位置に圧縮データのビットが含まれている場合のみ、その完全バイトが格納されたものと見なされます。圧縮時に、格納される圧縮データの先頭ビットがバイトのビット位置 0 にない場合は、そのバイトの中で圧縮データの先頭ビットより左にあるビットは、無変更のままになります。圧縮時に、格納される最後のバイトが圧縮データを完全に含んでいない場合は、そのバイト内の圧縮データの右端ビットの右にあるビットは、無変更のままになるか、または 0 に設定されるか、どちらかです。

汎用レジスター $R_2 + 1$ 内の長さは、第 2 オペランド位置で処理された完全バイトの数だけ減少し、汎用レジスター R_2 内のアドレスは同じバイト数だけ増加しています。展開時には、あるバイトのすべてのビットが展開データの生成に使用された場合にのみ、その完全バイトが処理されたとみなされます。

汎用レジスター R_1 および R_2 内の、アドレスの一部ではない左端ビットは、0 に設定されるかまたは無変更のままになります。ただし、24 ビットまたは 31 ビットのアドレッシング・モードでは、これらのレジスターのビット 0-31、および汎用レジスター $R_1 + 1$ および $R_2 + 1$ のビット 0-31 は、常に無変更のままになります。

処理済みの圧縮データの最後のビットの直後のビットのビット番号は、汎用レジスター 1 のビット位置 61-63 に入れられ、このレジスターのビット 52-60 と、レジスター内のアドレスの一部ではない左端は、0 に設定されるかまたは無変更のままになります。ただし、元の長さ値の片方または両方が小さすぎて処理された圧縮データがなかった場合は、レジスター内のすべてのビットが無変更のままになることがあります。汎用レジスター 0 のビット 47 が 1 のときは、汎用レジスター 1 のビット 52-60 は常に無変更のままになります。また、24 ビットまたは 31 ビットのアドレッシング・モードでは、このレジスターのビット 0-31 は、常に無変更のままです。

両オペランドが互いにオーバーラップしているか、または第 1 オペランドが記憶域内で辞書または記号変換テー

ブルに何らかの形でオーバーラップしている場合、結果は予測不能です。

記号変換が指定された場合、記号変換テーブルは 2 バイト単位のエンタリーで構成され、各エンタリーの右端ビット位置には交換記号が含まれています。交換記号の長さは、汎用レジスター 0 のビット 48-51 により指定されます。記号変換テーブル・エンタリー内の交換記号の一部ではない左側のビットは、0 でなければなりません。それ以外の場合、結果は予測不能です。

指標記号を交換記号に変換するためには、指標記号に 2 を掛けた結果が記号変換テーブルの先頭のアドレスに追加され、テーブル内の特定エンタリーの位置が求められ、そのエンタリーから交換記号が取得されます。

この命令の実行は割り込み可能です。割り込みが起きると、それが操作を中止させる割り込みでない限り、 R_1 フィールドおよび R_2 フィールドが指すレジスターの内容、および汎用レジスター 1 の内容は、通常の命令完了時と同様に更新されます。これにより、この命令の実行が再開されたときは、割り込み点から処理が再開されます。条件コードは予測不能です。

4K バイトより長いオペランドの場合は、処理中の現在の記憶位置より後の 4K バイトを超える記憶位置については、アクセス例外は認識されません。辞書および記号変換テーブルの領域を使用することが指定されている場合は、操作中には記憶位置が使用されない場合であっても、辞書および記号変換テーブル内のすべての記憶位置について、アクセス例外が認識されることがあります。オペランドに関連した R フィールドが奇数の場合は、そのオペランド、辞書、または記号変換テーブルについてはアクセス例外は認識されません。また、 R_1 フィールドが奇数の場合は、 PER 記憶域変更イベントは認識されず、変更ビットも設定されません。

どちらかのオペランドか、または辞書か記号変換テーブルについて、アクセス例外が認識されるはずの状況では、結果は、この例外が認識されるか、条件コード 3 が設定されるか、どちらかです。条件コード 3 が設定された場合は、命令が実行されて同じオペランドの処理を続けようとしたときに例外条件がまだ残っていれば、アクセス例外が認識されます。

圧縮時には、例外が認識されるか条件コード 3 が設定されるかに関係なく、アクセス例外条件の無効化またはページ保護例外条件の抑止は、アクセス例外条件が存在しなかった場合に生成されるはずの指標記号のみが生成される結果を生む形で処理されます。

圧縮時または展開時には、例外が認識されるか条件コード 3 が設定されるかに関係なく、アクセス例外条件の無効化または抑止の結果として、汎用レジスター R1 内の最終アドレスにより指定される記憶位置またはその右にある第 1 オペランド位置に、データが格納されることがあります。この結果は、真の無効化または抑止ではありません。格納されるデータの量は、アクセス例外条件の理由に応じて異なります。辞書または記号変換テーブルへの参照が原因の例外条件の場合は、最終アドレスが指す記憶位置またはその右側に、最大 4K バイトのデータが格納されている可能性があります。第 1 オペランドまたは第 2 オペランドへの参照が原因の例外条件の場合は、圧縮時なら 1 個の指標または交換記号の一部、展開時なら 1 個の文字記号の一部が、最終アドレスにより指定された記憶位置またはその右側に格納されている可能性があります。どの場合も、命令が再実行されて同じオペランドの処理が続行されると、格納が繰り返されます。

第 1 オペランドの終わりに達し、第 2 オペランドについてアクセス例外が認識される状況が発生した場合は、条件コード 1 が設定されるかアクセス例外が認識されるかは予測不能です。

展開時に、展開辞書が論理的に正しくない場合は、「Enterprise Systems Architecture/390 Data Compression」(SA22-7208-01) の第 1 章『Expansion Process』で説明されているように、誤った格納が行われることがあります。この場合のアクセス例外の結果は、真の無効化または抑止ではない場合があります。

特別な条件

各文字記号の圧縮時には、記号の中の文字数、または記号の文字数を表す辞書の文字エンタリー数 (兄弟記述子ではない) がカウントされ、このカウント値が大きくなりすぎると、データ例外が認識されます。カウントが少なくとも 260 に達するまでは、例外の認識は起きません。

圧縮時には、各親エンタリーの処理中に処理された子文字または兄弟文字の数がカウントされ、このカウント値が大きくなりすぎると、データ例外が認識されます。カウントが少なくとも 260 に達するまでは、例外の認識は起きません。つまり、親が持つ子は 260 個以下でなければなりません。さもないと、データ例外が認識されます。

各文字記号の展開時には、記号の中の文字数、または記号の文字数を表す辞書エンタリー数がカウントされ、このカウント値が大きくなりすぎると、データ例外が認識されます。記号内の文字数がカウントされる場合、カウ

ントが少なくとも 260 に達するまでは、例外の認識は起きません。記号内の文字数を表す辞書エンタリーがカウントされる場合、カウントが少なくとも 127 に達するまでは、例外の認識は起きません。

辞書内で起きるある種のエラー条件は、データ例外が認識されて操作が中止される原因になります。このようなエラー条件の一部は、資料 SA22-7208-01 の第 1 章の『Expansion Process』および『Results of Dictionary Errors』に説明されています。残りは SA22-7208-01 の第 2 章に説明されています。

結果の条件コード:

- 0 第 2 オペランドの終わりに達しました。
- 1 第 1 オペランドの終わりに達し、第 2 オペランドの終わりに達していません。
- 2 --
- 3 CPU で決定された量のデータが処理されました。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2、辞書、および記号変換テーブル。格納: オペランド 1)
- データ例外
- 指定例外

プログラミング上の注意:

1. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、操作を継続できます。処理済みのデータの量をプログラムで判別する必要はありません。
2. 圧縮時には、アクセス例外の無効化が認識されることになる場合、指標記号が生成されるのは、それが、アクセス例外条件が存在しなかった場合に生成されるはずの指標記号である場合のみです。この結果、この命令を 1 回以上実行し、同じ辞書を使用して同じ展開データを圧縮すると、常に同じ圧縮データが生成されることになります。つまり、(1) 辞書内での最良一致が常に第 2 オペランド内の文字について検出されるか、そうでなければ、実行が終了して CC3 が設定されるかまたは例外が認識されません。そして、(2) 圧縮の結果は (多くの場合は命令の実行数を変えることで) 反復可能で、予測可能です。

例えば、圧縮対象のストリング内の次の文字が ABC で、辞書エンタリー A に子 B があり、B にさらに子 C があつた場合、通常の操作では、ABC を単一の指標記号として圧縮します。しかし、このストリングの C が無効ページ (ページ変換例外が認

識されるページ)の先頭バイト内にあった場合は、指標記号は生成されません。もっと具体的に言えば、文字記号 AB に対応する指標記号は生成されません。なぜなら、これはアクセス例外条件が存在しなかった場合に生成されることになる指標記号ではないからです。

上記の「最良一致」とは、第 2 オペランド内の文字についてのみ言っています。上の例では、第 2 オペランド(ストリング)の次の 2 文字が AB で、これらが第 2 オペランドの最後の 2 文字の場合、最良一致は AB になります。第 2 オペランドにもう 1 バイトあり、そこに C が含まれていたとすれば、一致は ABC になります。

展開は、通常では常時反復可能です。中止の原因となる例外が認識された場合以外は、指標記号は常に、それが表す文字記号そのものに展開されます。

3. 展開時には、第 1 オペランド位置に少なくとも 1 個の未使用のバイト位置が残っていれば、COMPRESSION CALL は、第 1 オペランド位置に次の指標記号の展開結果のデータを収容できる未使用バイト位置が十分にないと判別する前に、第 2 オペランド内の次の指標記号の処理を完了してしまうことがあります。この、次の指標記号が原因で不良辞書エントリが検出された場合は、結果は、データ例外が起きるか、または条件コード 1 が設定されることとなります。

指標記号の処理の結果として第 1 オペランド位置が正確に埋まると、COMPRESSION CALL は即時に条件コード 1 を設定します。ただし、同時に第 2 オペランド位置の終わりに達した場合は、条件コード 0 を設定します。条件コード 1 を即時に設定することには、ある辞書を使用してデータを圧縮し、すぐにその直後(おそらくはビット境界で)別のタイプのデータを別の辞書を使用して圧縮できるといった利点があります。最初の辞書を使用して圧縮されたデータの展開時に、第 1 オペランド位置の長さが生成される展開データの長さになるように正確に指定されていれば、この圧縮データは正常に展開することができます。第 1 オペランド位置がいっぱいになれば、条件コード 1 が設定され、その時点で辞書の指定が変更されて、圧縮データの残り部分は 2 番目の辞書を使用して展開できるようになります。定義により条件コード 1 が設定されないようになっている場合は、2 番目の辞書の使用結果として、次の指標記号を最初の辞書により展開しようとする試みが行われ、その結果データ例外が認識されることがあります。例えば、次の指標記号が 2 番目の辞書

の文字エントリを正しく指定していても、その指標記号が最初の辞書内のフォーマット 1 兄弟記述子の後半部分を指定していて、その後半部分が例えば 0 (16 進数 F0) などの文字で始まっている場合、これは文字エントリ内では無効な部分記号長と見なされることとなります。

4. 辞書または記号変換テーブルへの参照が原因のアクセス例外条件無効化の場合は、汎用レジスター R₁の最終アドレスが指す記憶位置またはその右側に、データが格納されている可能性があります。この格納および処理は、格納されたデータを生成するために必要であったもので、COMPRESSION CALL が再実行されて同じオペランドの処理が続行されると、また繰り返されます。この反復処理は命令実行のパフォーマンスの低下を招くので、避ける必要があります。そのためには、プログラムが 1 回だけ実行され、辞書および記号変換テーブルに関するページ変換例外条件がめったに起きない環境を確保することが重要です。
5. 圧縮データ・ビット番号(CBN)の使用法と設定方法の例を次に示します。この例では、次のことが前提になっています。

- 操作は展開操作です。
- 汎用レジスター 0 の CDSS は 2 進数 0010 です。したがって、展開辞書内には 1K エントリがあり、指標記号の長さは 10 ビットです。
- 第 2 オペランド(圧縮データ・オペランド)は、16 進数 6000 の記憶位置から始まり、長さは 5 バイトです。初期 CBN は 7 です。したがって、展開する指標記号は 3 個で、最終 CBN は 5 になります。
- 16 進数 6000 の記憶位置から始まる圧縮データは、16 進数 0081FF9FF8 です。したがって、3 個の指標記号は 16 進数 103、3FC、および 3FF です。
- 第 1 オペランド(拡張データ・オペランド)は、16 進数 5000 の記憶位置から始まり、長さは 64 バイトです。3 個の指標記号は、合計 14 バイトの展開データに展開されます。

次の図は、汎用レジスター R₁、R₁ + 1、R₂、および R₂ + 1 の初期内容および最終内容、16 進数 6000-6004 の記憶位置の内容の 2 進数表現、および CBN に対応するカーソルが展開操作中に進む様子を示しています。

レジスター	初期 内容 (16 進数)	最終 内容 (16 進数)
R ₁	5000	500E
R ₁₊₁	40	32
R ₂	6000	6004
R ₂₊₁	5	1

16 進数 6000-6004 の記憶位置の内容の 2 進数表現

00000000 10000001 11111111 10011111 11111000

初期 CBN (7) ↑ ↑ ↑ ↑ ↑ 最終 CBN (5)

6. 親が 260 個を超える子を持つことを許可しない理由は、次のとおりです。親は、5 つまで、同一の子文字を含むことができます。さらに、255 個の異なる兄弟文字が可能です。これらの兄弟文字はすべて相互に異ならなければならない、子文字とも異なるものでなければなりません。さもないと、インプリメンテーションによっては、むだになる(突き合わせに使用されない)ことがあります。この範囲内で、個々の可能な子はすべて許可されます。

7. 記号変換は VTAM で使用するためのものです。VTAM は、ソフトウェアおよび適応辞書を使用して圧縮を行うことにより、処理を開始します。適応辞書が成熟し、圧縮度が十分なレベルに達する(なんらかのしきい値を超える)と、VTAM は、その辞書を「凍結」(適応を停止)し、セッションの相手方にも凍結するよう通知します。そして、適応辞書を COMPRESSION CALL で使用される形式に変換し、その後、COMPRESSION CALL を使用して圧縮を続行します。セッションの相手方は、凍結された適応辞書の使用を続けることができます。

次に、STT オフセットについて説明します。VTAM が 4K エントリーの適応辞書 (VTAM が使用する最大サイズ) を使用しているとします。この VTAM 辞書には兄弟記述子は存在していないので、この辞書内のエントリーはすべて文字記号に対

応しています。COMPRESSION CALL 辞書では幾つかのエントリーは兄弟記述子であることが必要なため、VTAM 辞書は 1 対 1 で COMPRESSION CALL 辞書にマップすることはできません。したがって、VTAM には、基本圧縮操作で使用するための 8K エントリー辞書が必要です。使用する必要があるのは 8K の後半 4K の中の最初の 100 個前後のエントリーのみで、これらのエントリーが、兄弟記述子である前半の 4K 内のエントリーの埋め合わせをする(とって代わる)働きをします。STT は、後半 4K のこれらの 100 個前後のエントリーの直後から開始でき、また、スペース節約のためにはそうする必要があります。この例では、指標記号は 13 ビットですが、12 ビットの交換記号に変換されます。

8. プログラムは、先頭でフィールドをチェーニングすることにより管理されるページに、辞書を入れることがあります。この場合、圧縮後に連続する位置に入れるために 1 つの辞書の一部を移動する必要がありますか、または辞書内にホール生じさせる必要があります。COMPRESSION CALL の定義には、明示的にホールをサポートするものは何も含まれていません。しかし、展開データ内に絶対に現れない文字が少なくとも 1 文字あると仮定すると、その文字を親エントリー内の子文字として、または兄弟記述子内の兄弟文字として使用して、絶対に参照されない 1 つまたは複数の子を指定することができます。この方法でホールを作成することができます。
9. COMPRESSION CALL のオペランド、辞書、および記号変換テーブルに対する参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)
10. 7-65ページの図7-12 および 7-67ページの図7-13 は、圧縮プロセスおよび展開プロセスの可能な形式を示しています(可能な形式がこれだけという意味ではありません)。この図には、辞書エラーを検出するためのテストや、辞書エラーの結果は示されていません。

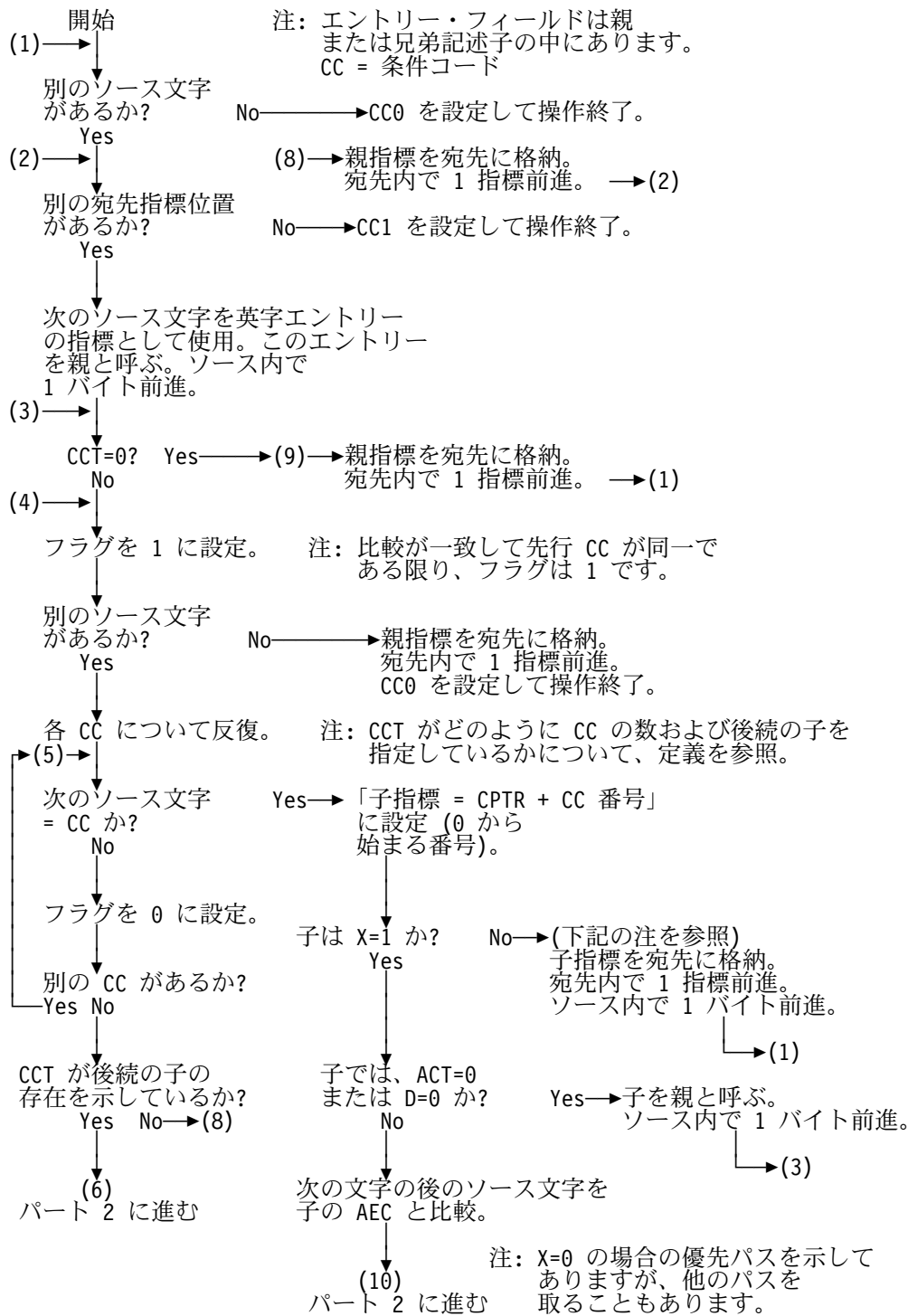


図 7-12 (1/2). 圧縮プロセス

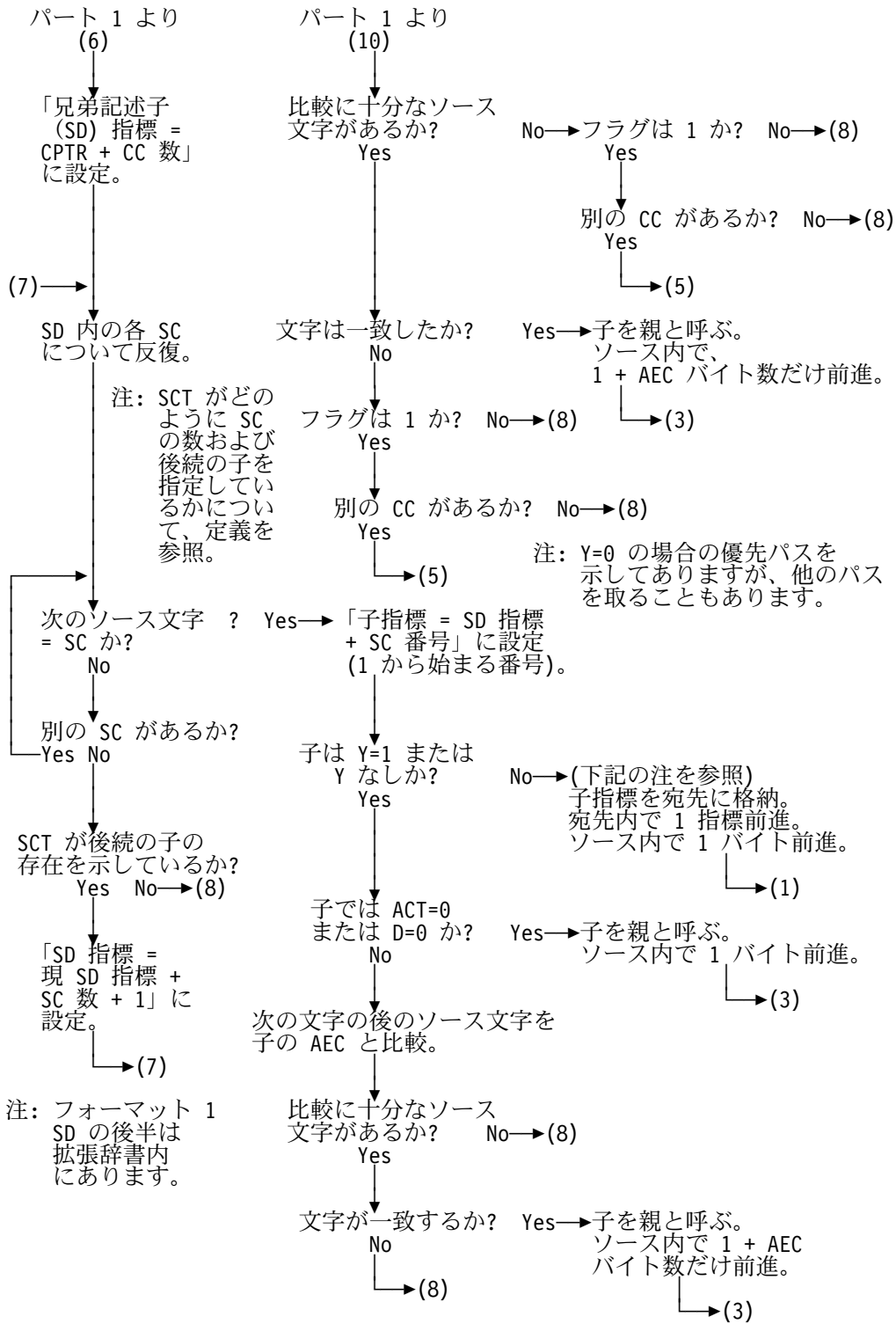


図 7-12 (2/2). 圧縮プロセス

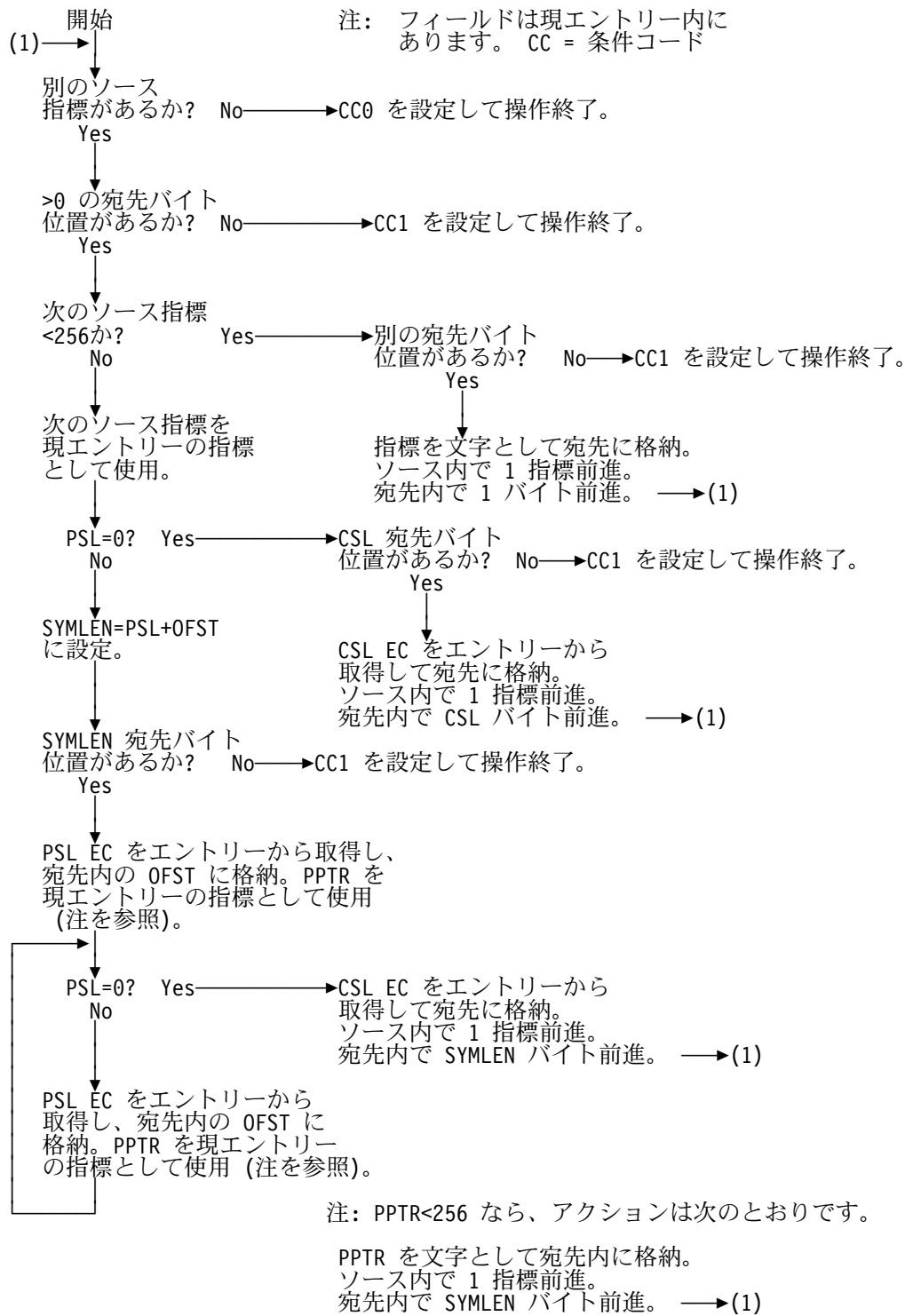
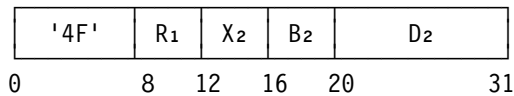


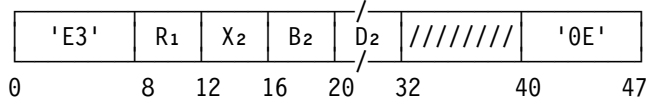
図 7-13. 展開プロセス

CONVERT TO BINARY

CVB R₁,D₂(X₂,B₂) [RX]



CVBG R₁,D₂(X₂,B₂) [RXE]



第 2 オペランドが 10 進数から 2 進数に変換され、結果が第 1 オペランド位置に入られます。

CONVERT TO BINARY (CVB) では第 2 オペランドは記憶域内で 8 バイトを占め、CONVERT TO BINARY (CVBG) では第 2 オペランドは記憶域内で 16 バイトを占めます。第 2 オペランドは、第 8 章、『10 進数命令』で説明するように、パック 10 進数データのフォーマットを備えています。第 2 オペランドは、符号コードおよび数字コードが有効かどうか検査され、無効なコードが検出されると、10 進オペランド・データ例外が認識されます。

CONVERT TO BINARY (CVB) では、変換の結果は 32 ビットの符号付き 2 進整数で、汎用レジスタ R₁ のビット位置 32-63 に入られます。このレジスタのビット 0-31 は変更されません。変換可能でしかも 32 ビット位置に収容可能な最大正数は、2,147,483,647 です。変換可能な最大負数 (絶対値が最大の負数) は、-2,147,483,648 です。この範囲の外部にある 10 進数については、結果の 2 進数の右端 32 ビットをこのレジスタに入れて操作が完了し、固定小数点除算例外が認識されます。

CONVERT TO BINARY (CVBG) では、変換の結果は 64 ビットの符号付き 2 進整数で、汎用レジスタ R₁ のビット位置 0-63 に入られます。変換可能でしかも 64 ビット・レジスタに収容可能な最大正数は 9,223,372,036,854,775,807 です。変換可能な最大負数 (絶対値が最大の負数) は、-9,223,372,036,854,775,808 です。この範囲の外部にある 10 進数については、固定小数点除算例外が認識され、操作は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

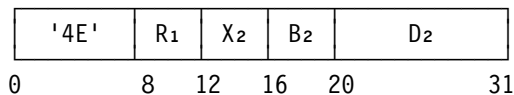
- アクセス例外 (取り出し: オペランド 2)
- データ例外
- 固定小数点除算例外

プログラミング上の注意:

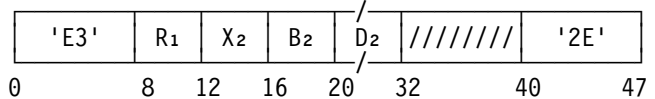
1. CONVERT TO BINARY 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. 第 2 オペランドが負の値なら、結果は 2 の補数表記になります。
3. CONVERT TO BINARY の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

CONVERT TO DECIMAL

CVD R₁,D₂(X₂,B₂) [RX]



CVDG R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドが 2 進数から 10 進数に変換され、結果が第 2 オペランド位置に格納されます。

CONVERT TO DECIMAL (CVD) では、第 1 オペランドは 32 ビットの符号付き 2 進整数として扱われ、結果は記憶域内で 8 バイトを占めます。CONVERT TO DECIMAL (CVDG) では、第 1 オペランドは 64 ビットの符号付き 2 進整数として扱われ、結果は記憶域内で 16 バイトを占めます。

結果は、第 8 章、『10 進数命令』で説明するように、パック 10 進数データのフォーマットで表されます。結果の右端 4 ビットは符号を表します。正符号は 1100 としてエンコードされ、負符号は 1101 としてエンコードされます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意:

1. CONVERT TO DECIMAL 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. CVD では、変換元の数値は、汎用レジスタから取得される 32 ビットの符号付き 2 進整数です。結果には 15 桁の 10 進数を使用でき、31 ビットの等価の 10 進数に必要な最大桁は 10 桁なので、オーバーフローは発生しません。同様に、CVDG では、31 桁の 10 進数を使用でき、63 ビットの等価の 10 進数に必要な最大桁は 19 桁なので、オーバーフローは発生しません。
3. CONVERT TO DECIMAL の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

CONVERT UNICODE TO UTF-8

CUUTF R₁,R₂ [RRE]

'B2A6'	////////	R ₁	R ₂
0	16	24	28 31

第 2 オペランドの 2 バイト Unicode 文字は、UTF-8 文字に変換されて第 1 オペランド位置に入れられます。UTF-8 文字は、変換元の Unicode 文字に応じて、1 バイト、2 バイト、3 バイト、または 4 バイトです。操作は、第 1 オペランドまたは第 2 オペランドの終わりに達するか、CPU で決定される文字数の変換が完了するか、どちらかの条件が先に発生するまで続けられます。結果は条件コードで示されます。

R₁ フィールドおよび R₂ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 のビット位置 32-63

の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 全体の内容で指定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

7-70ページの図7-14 は、上記で説明したレジスタの内容を示しています。

第 2 オペランドの文字は、左から右への順に 1 つずつ選択されて変換されます。変換の結果として生じたバイトは、左から右への順に第 1 オペランド位置に入れられます。操作は、第 1 オペランド位置または第 2 オペランド位置が使い尽くされるか、または CPU で決定された第 2 オペランドの文字数の変換が完了するまで続けられます。

Unicode 文字から UTF-8 文字への変換方式を示すために、ここでは Unicode 文字を次のように英字で表すことにします。

Unicode 文字 ビット番号	111111 01234567 89012345
識別ビット文字	abcdefghijklmnop

Unicode サロゲート・ペア (高サロゲートと呼ばれる 1 文字と、それに続く低サロゲートと呼ばれる 1 文字で構成される文字ペア) の場合、ビットは、英字で表すと次のように識別されます。

Unicode 高サロゲート ビット番号	111111 01234567 89012345
識別ビット文字	110110ab cdefghij

Unicode 低サロゲート ビット番号	11112222 22222233 67890123 45678901
識別ビット文字	110111kl mnopqrst

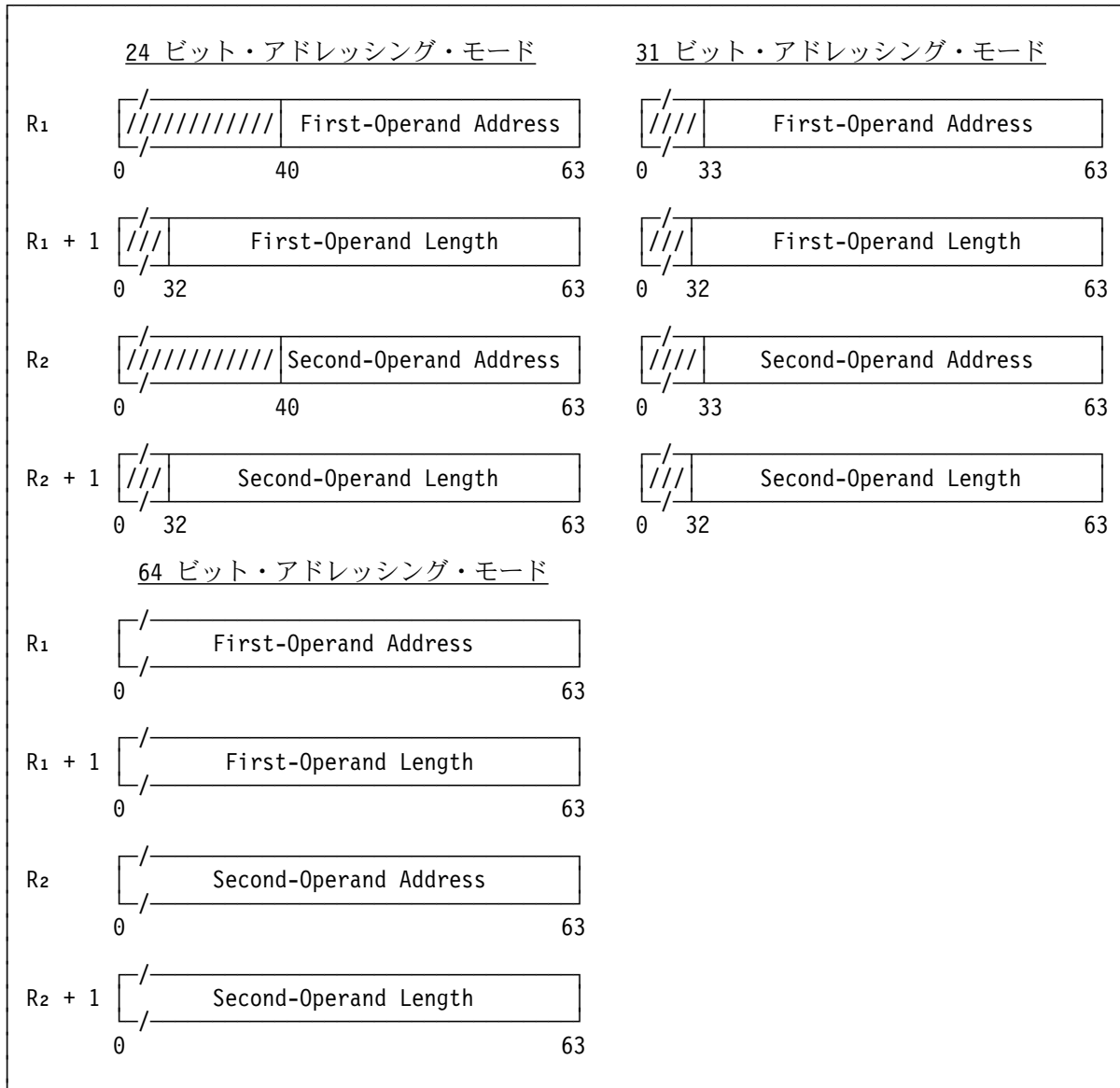


図 7-14. `CONVERT UNICOD TO UTF-8` で使用するレジスターの内容

16 進数 0000 から 007F までの範囲内にある Unicode 文字は、次のように、1 バイトの UTF-8 文字に変換されます。

Unicode 文字	00000000 0jklmnop
UTF-8 文字	0jklmnop

16 進数 0080 から 07FF までの範囲内にある Unicode 文字は、次のように、2 バイトの UTF-8 文字に変換されます。

Unicode 文字	00000fgh iklmnop
UTF-8 文字	110fghij 10klmnop

16 進数 0800 から D7FF まで、および DC00 から FFFF までの範囲内にある Unicode 文字は、次のように、3 バイトの UTF-8 文字に変換されます。

Unicode 文字	abcdefgh iklmnop
UTF-8 文字	1110abcd 10efghij 10klmnop

16 進数 D800 から DBFF までの範囲内の高サロゲートで始まる Unicode サロゲート・ペアは、次のように、4 バイトの UTF-8 文字に変換されます。

Unicode 文字	110110ab cdefghij 110111kl mnopqrst
UTF-8 文字	11110uvw 10xyefgh 10ijklmn 10opqrst

ここで、 $uvwxyz = abcd + 1$

2 番目の Unicode 文字の最初の 6 ビットは無視されま
す。

第 2 オペランド位置が使い尽くされたと見なされるの
は、少なくとも 2 バイトの残りバイトがもう含まれてい
ないとき、または、先頭 2 バイトが Unicode 高サロゲ
ートの場合には、少なくとも 4 バイトの残りバイトがも
う含まれていないときです。第 1 オペランド位置が使い
尽くされたと見なされるのは、次の第 2 オペランド文字
またはサロゲート・ペアの変換結果として生じる
UTF-8 文字を入れるために少なくとも必要な、1 バイ
ト、2 バイト、3 バイト、または 4 バイトがもう残って
いないときです。

第 2 オペランド位置が使い尽くされたときは、条件コー
ド 0 が設定されます。第 1 オペランド位置が使い尽く
されたときは、条件コード 1 が設定されます。ただし、
第 2 オペランド位置も同時に使い尽くされた場合は、条
件コード 0 が設定されます。CPU で決定される文字数
の変換が完了した場合は、条件コード 3 が設定されま
す。

操作の完了時には、汎用レジスター R2 + 1 の内容は、
変換されたバイトの数だけ減少し、汎用レジスター R2
の内容は同じバイト数だけ増加しています。また、汎用
レジスター R1 + 1 の内容は、第 1 オペランド位置に
入れられたバイトの数だけ減少し、汎用レジスター R1
の内容は、同じバイト数だけ増加しています。24 ビッ
トまたは 31 ビットのアドレッシング・モードで汎用レ
ジスター R1 および R2 が更新されたとき、それらのレ
ジスターのビット 32-39 (24 ビット・モードの場合) ま
たはビット 32 (31 ビット・モードの場合) が、0 に設
定されるかまたは無変更のままになることがあります。

24 ビットまたは 31 ビットのアドレッシング・モードで
は、汎用レジスター R1、R1 + 1、R3、および
R3 + 1 のビット位置 0-31 の内容は、常に無変更のま
まになります。

条件コード 3 が設定された場合は、レジスターは更新さ
れているので、命令の再実行時には、次に処理する順番

に当たるバイト位置から処理を再開することができま
す。

条件コード 3 が設定される結果になる処理量は、システ
ム・パフォーマンスの向上という観点から CPU が決定
するもので、この量は命令を実行するたびに異なる場合
があります。

R1 レジスターが R2 レジスターと同一の場合は、結果
は予測不能です。

第 2 オペランドが第 1 オペランドとオーバーラップし
ている場合も、結果は予測不能です。

処理された最後のバイトの右側にあるオペランド部分に
ついては、アクセス例外が認識される場合とされない場
合があります。4K バイトより長いオペランドの場合
は、処理された最後のバイトを 4K バイトを超える位置よ
り後にある記憶位置については、アクセス例外は認識さ
れません。

オペランドの長さが 0 の場合、そのオペランドについて
はアクセス例外は認識されません。オペランドに関連し
た R フィールドが奇数の場合は、そのオペランドにつ
いてはアクセス例外は認識されません。

結果の条件コード:

- 0 第 2 オペランド全体が処理されました。
- 1 第 1 オペランドの終わりに達しました。
- 2 --
- 3 CPU で決定された文字数が変換されました。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペ
ランド 1)
- 指定例外

プログラミング上の注意:

1. 条件コード 3 が設定されると、プログラムはブラン
チに従って命令に戻り、変換処理を継続できます。
処理済みの第 1 オペランドまたは第 2 オペランド
のバイト数をプログラムで判別する必要はありませ
ん。
2. CONVERT UNICODE TO UTF-8 の記憶域オペラ
ンド参照は、複数アクセス参照となることもありま
す。(5-86ページの『記憶域オペランドの一貫性』
を参照してください。)

CONVERT UTF-8 TO UNICODE

CUTFU R₁,R₂ [RRE]

'B2A7'		////////	R ₁	R ₂
0	16	24	28	31

第 2 オペランドの 1 バイト、2 バイト、3 バイト、または 4 バイトの UTF-8 文字が、2 バイト Unicode 文字に変換され、第 1 オペランド位置に入れられます。操作は、第 1 オペランドまたは第 2 オペランドの終わりに達するか、CPU で決定された文字数の変換が完了するか、または無効な UTF-8 文字が検出されるか、いずれかの条件が最初に発生するまで続けられます。結果は条件コードで示されます。

R₁ フィールドおよび R₂ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 のビット位置 32-63 の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 全体の内容で指定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

7-73 ページの図 7-15 は、上記で説明したレジスタの内容を示しています。

第 2 オペランドの文字は、左から右への順に 1 つずつ選択されて変換されます。変換の結果として生じたバイトは、左から右への順に第 1 オペランド位置に入れられます。操作は、第 1 オペランド位置または第 2 オペランド位置が使い尽くされるか、CPU で決定された数の第 2 オペランド文字の変換が完了するか、または無効な UTF-8 文字が第 2 オペランドで検出されるまで続けられます。

UTF-8 文字から Unicode 文字への変換方式を示すために、ここでは Unicode 文字を次のように英字で表すことにします。

Unicode 文字	111111
ビット番号	01234567 89012345
識別ビット文字	abcdefghijklmnop

Unicode サロゲート・ペア (高サロゲートと呼ばれる 1 文字と、それに続く低サロゲートと呼ばれる 1 文字で構成される文字ペア) の場合、ビットは、英字で表すと次のように識別されます。

Unicode 高サロゲート	111111
ビット番号	01234567 89012345
識別ビット文字	110110ab cdefghij

Unicode 低サロゲート	11112222 22222233
ビット番号	67890123 45678901
識別ビット文字	110111kl mnopqrst

UTF-8 文字の先頭バイトの内容が 16 進数 00 から 7F までの範囲内にある場合は、文字は 1 バイト文字であり、次のように 2 バイトの Unicode 文字に変換されます。

UTF-8 文字	0jklmnop
Unicode 文字	00000000 0jklmnop

UTF-8 文字の先頭バイトの内容が 16 進数 C0 から DF までの範囲内にある場合は、文字は 2 バイト文字であり、次のように 2 バイトの Unicode 文字に変換されます。

UTF-8 文字	110fghij 10klmnop
Unicode 文字	00000fgh ijklmnop

UTF-8 文字の 2 番目のバイトの最初の 2 ビットは無視されます。

UTF-8 文字の先頭バイトの内容が 16 進数 E0 から EF までの範囲内にある場合は、文字は 3 バイト文字であ

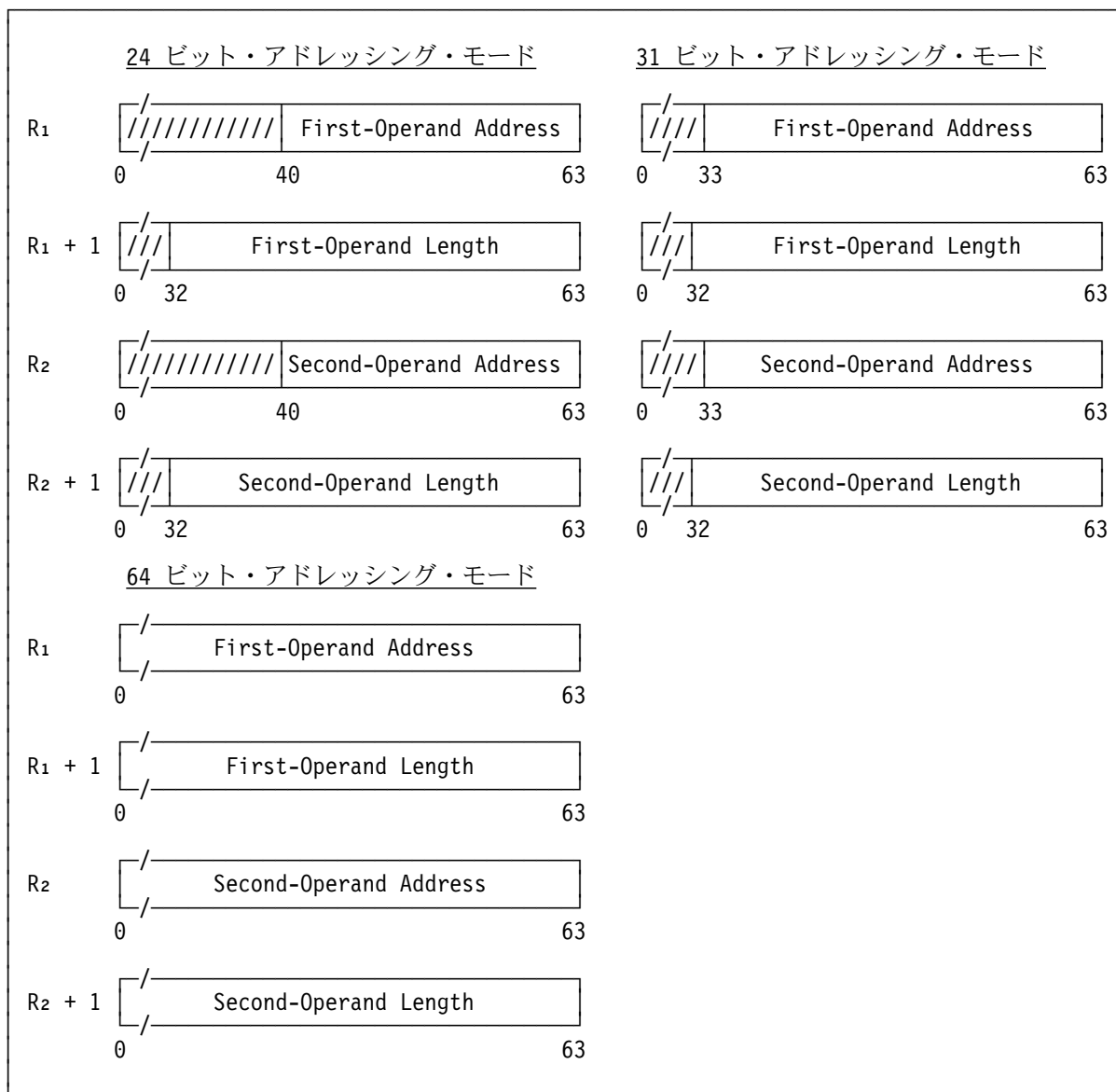


図 7-15. CONVERT UTF-8 TO UNICODE で使用するレジスタの内容

り、次のように 2 バイトの Unicode 文字に変換されます。

UTF-8 文字	1110abcd 10efghij 10klmnop
Unicode 文字	abcdefghijklmnop

UTF-8 文字の 2 番目および 3 番目のバイトの最初の 2 ビットは無視されます。

UTF-8 文字の先頭バイトの内容が 16 進数 F0 から F7 までの範囲内にある場合は、文字は 4 バイト文字であり、次のように、2 個の 2 バイト Unicode 文字 (サロゲート・ペア) に変換されます。

UTF-8 文字	11110uvw 10xyefgh 10ijklmn 10opqrst
----------	-------------------------------------

Unicode 文字	110110ab cdefghij 110111kl mnopqrst
------------	-------------------------------------

ここで、zabcd = uvwxy -1

UTF-8 文字の 2 番目、3 番目、および 4 番目のバイトの最初の 2 ビットは無視されます。減算操作によって生成される高位ビット (z) は 0 になるはずですが、これは無視されます。

第 2 オペランド位置が使い尽くされたと見なされるのは、少なくとも 1 個の残りバイトが含まれていないとき、または、先頭の残りバイトの内容が示す 2 バイト、3 バイト、または 4 バイトの UTF-8 文字を入れるため

に少なくとも必要な 2 バイト、3 バイト、または 4 バイトがもう残っていないときです。第 1 オペランド位置が使い尽くされたと見なされるのは、少なくとも 2 バイトの残りバイトがもう含まれていないとき、または、4 バイトの UTF-8 文字を変換する場合は、少なくとも 4 バイトの残りバイトがもう含まれていないときです。

第 2 オペランド位置が使い尽くされたときは、条件コード 0 が設定されます。第 1 オペランド位置が使い尽くされたときは、条件コード 1 が設定されます。ただし、第 2 オペランド位置も同時に使い尽くされた場合は、条件コード 0 が設定されます。CPU で決定される文字数の処理が完了した場合は、条件コード 3 が設定されません。

次の UTF-8 文字の先頭バイトの内容が 16 進数 80 から BF、または 16 進数 F8 から FF までの範囲にある場合は、その文字は無効であり、条件コード 2 が設定されます。

条件コード 1 と 2 を設定する条件が同時に発生した場合は、条件コード 2 が設定されます。

操作の完了時には、汎用レジスター $R_2 + 1$ の内容は、変換されたバイトの数だけ減少し、汎用レジスター R_2 の内容は同じバイト数だけ増加しています。また、汎用レジスター $R_1 + 1$ の内容は、第 1 オペランド位置に入れられたバイトの数だけ減少し、汎用レジスター R_1 の内容は、同じバイト数だけ増加しています。24 ビットまたは 31 ビットのアドレッシング・モードで汎用レジスター R_1 および R_2 が更新されたとき、それらのレジスターのビット 32-39 (24 ビット・モードの場合) またはビット 32 (31 ビット・モードの場合) が、0 に設定されるかまたは無変更のままになることがあります。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスター R_1 、 $R_1 + 1$ 、 R_2 、および $R_2 + 1$ のビット位置 0-31 の内容は、常に無変更のままになります。

条件コード 3 が設定された場合は、レジスターは更新されているので、命令の再実行時には、次に処理する順番に当たるバイト位置から処理を再開することができます。

条件コード 2 が設定された場合、汎用レジスター R_2 には無効な UTF-8 文字が含まれています。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

R_1 レジスターが R_2 レジスターと同一の場合は、結果は予測不能です。

第 2 オペランドが第 1 オペランドとオーバーラップしている場合も、結果は予測不能です。

処理された最後のバイトの右側にあるオペランド部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、処理された最後のバイトを 4K バイトを超える位置より後にある記憶位置については、アクセス例外は認識されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。

結果の条件コード:

- 0 第 2 オペランド全体が処理されました。
- 1 第 1 オペランドの終わりに達しました。
- 2 無効な UTF-8 文字です。
- 3 CPU で決定された文字数が処理されました。

プログラム例外:

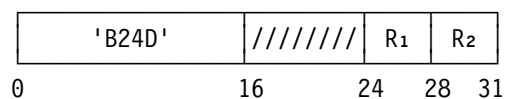
- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 指定例外

プログラミング上の注意:

1. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、変換処理を継続できます。処理済みの第 1 オペランドまたは第 2 オペランドのバイト数をプログラムで判別する必要はありません。
2. 複数バイトの UTF-8 文字の継続バイトのビット 0 および 1 については、変換のパフォーマンス向上のための検査は行われません。したがって、無効な継続バイトは検出されません。
3. CONVERT UTF-8 TO UNICODE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

COPY ACCESS

CPYA R₁,R₂ [RRE]



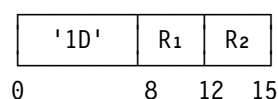
アクセス・レジスタ R₂ の内容が、アクセス・レジスタ R₁ に入れます。

条件コード: このコードは変更されません。

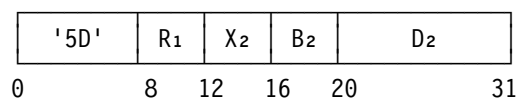
プログラム例外: なし。

DIVIDE

DR R₁,R₂ [RR]



D R₁,D₂(X₂,B₂) [RX]



64 ビットの第 1 オペランド (被除数) が 32 ビットの第 2 オペランド (除数) で割られ、32 ビットの剰余と商が第 1 オペランド位置に入れます。

R₁ フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定する必要があります。さもないと、指定例外が認識されます。

被除数は、64 ビットの符号付き 2 進整数として扱われます。被除数の左端 32 ビットは、汎用レジスタ R₁ のビット位置 32-63 にあり、右端 32 ビットは汎用レジスタ R₁ + 1 のビット位置 32-63 にあります。

除数、剰余、および商は、32 ビットの符号付き 2 進整数として扱われます。DIVIDE (DR) では、除数は汎用レジスタ R₂ のビット位置 32-63 にあります。剰余は汎用レジスタ R₁ のビット位置 32-63 に入れられ、商は汎用レジスタ R₁ + 1 のビット位置 32-63 に入れられます。両方のレジスタのビット 0-31 は変更されません。

商の符号は代数規則に基づいて決定され、剰余の符号は被除数と同じになります。ただし、商が 0 の場合、および剰余が 0 の場合は、符号は常に正です。

除数が 0 の場合、または被除数および除数の絶対値の商が 32 ビット符号付き 2 進整数で表せない場合は、固定小数点除算例外が認識されます。これには、0 を 0 で割る場合も含まれます。

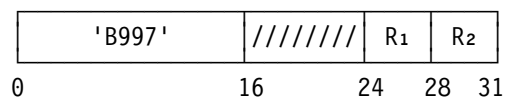
条件コード: このコードは変更されません。

プログラム例外:

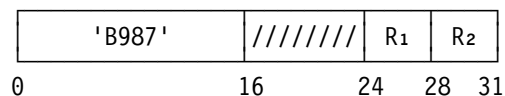
- アクセス例外 (取り出し: D のオペランド 2 のみ)
- 固定小数点除算例外
- 指定例外

DIVIDE LOGICAL

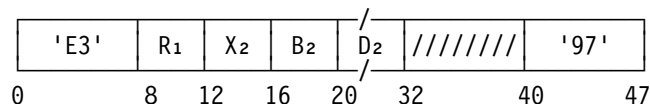
DLR R₁,R₂ [RRE]



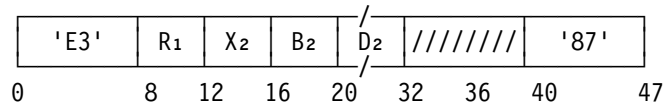
DLGR R₁,R₂ [RRE]



DL R₁,D₂(X₂,B₂) [RXE]



DLG R₁,D₂(X₂,B₂) [RXE]



64 ビットまたは 128 ビットの第 1 オペランド (被除数) が 32 ビットまたは 64 ビットの第 2 オペランド (除数) で割られ、32 ビットまたは 64 ビットの剰余と商が第 1 オペランド位置に入れます。

R₁ フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定する必要があります。さもないと、指定例外が認識されます。

DIVIDE LOGICAL (DLR, DL) では、被除数は 64 ビットの符号なし 2 進整数として扱われます。被除数の左端 32 ビットは、汎用レジスタ R_1 のビット位置 32-63 にあり、右端 32 ビットは汎用レジスタ $R_1 + 1$ のビット位置 32-63 にあります。

除数、剰余、および商は、32 ビットの符号なし 2 進整数として扱われます。DIVIDE LOGICAL (DLR) では、除数は汎用レジスタ R_2 のビット位置 32-63 にあります。剰余は汎用レジスタ R_1 のビット位置 32-63 に入れられ、商は汎用レジスタ $R_1 + 1$ のビット位置 32-63 に入れられます。両方のレジスタのビット 0-31 は変更されません。

DIVIDE LOGICAL (DLGR, DLG) では、被除数は 128 ビットの符号なし 2 進整数として扱われます。被除数の左端 64 ビットは汎用レジスタ R_1 にあり、右端 64 ビットは汎用レジスタ $R_1 + 1$ にあります。除数、剰余、および商は、64 ビットの符号なし 2 進整数として扱われます。剰余は汎用レジスタ R_1 に入れられ、商は汎用レジスタ $R_1 + 1$ に入れられます。

除数が 0 の場合、または被除数および除数の絶対値の商が 32 ビット符号なし 2 進整数 (DIVIDE LOGICAL (DLR, DL) の場合) または 64 ビット符号なし 2 進整数 (DIVIDE LOGICAL (DLGR, DLG) の場合) で表せない場合は、固定小数点除算例外が認識されます。これには、0 を 0 で割る場合も含まれます。

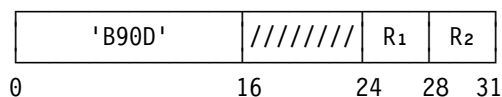
条件コード: コードは変更されません。

プログラム例外:

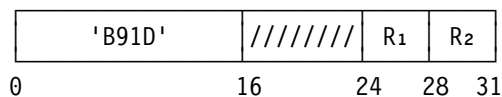
- アクセス例外 (取り出し: DL または DLG のオペランド 2 のみ)
- 固定小数点除算例外
- 指定例外

DIVIDE SINGLE

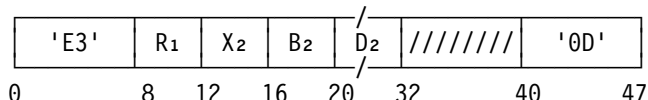
DSGR R_1, R_2 [RRE]



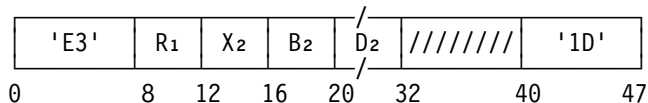
DSGFR R_1, R_2 [RRE]



DSG $R_1, D_2 (X_2, B_2)$ [RXE]



DSGF $R_1, D_2 (X_2, B_2)$ [RXE]



汎用レジスタ $R_1 + 1$ の 64 ビットの内容 (被除数) が 64 ビットまたは 32 ビットの第 2 オペランド (除数) で割られ、64 ビットの剰余が汎用レジスタ R_1 に入れられ、64 ビットの商が汎用レジスタ $R_1 + 1$ に入れられます。

R_1 フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

被除数、商、および剰余は、64 ビットの符号付き 2 進整数として扱われます。DIVIDE SINGLE (DSGR, DSG) では、除数は 64 ビットの符号付き 2 進整数として扱われます。DIVIDE SINGLE (DSGFR, DSGF) では、除数は 32 ビットの符号付き 2 進整数として扱われます。DSGFR では、除数は汎用レジスタ R_2 のビット位置 32-63 にあります。

商の符号は代数規則に基づいて決定され、剰余の符号は被除数と同じになります。ただし、商が 0 の場合、および剰余が 0 の場合は、符号は常に正です。

除数が 0 の場合、または被除数および除数の絶対値の商が 64 ビット符号付き 2 進整数で表せない場合は、固定小数点除算例外が認識されます。これには、0 を 0 で割る場合も含まれます。

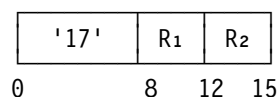
条件コード: コードは変更されません。

プログラム例外:

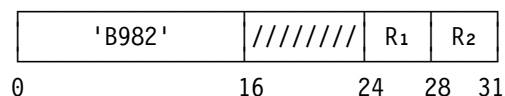
- アクセス例外 (取り出し: DSG および DSGF のオペランド 2 のみ)
- 固定小数点除算例外
- 指定例外

EXCLUSIVE OR

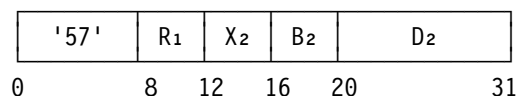
XR R₁,R₂ [RR]



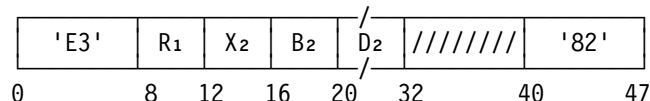
XGR R₁,R₂ [RRE]



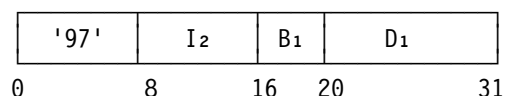
X R₁,D₂(X₂,B₂) [RX]



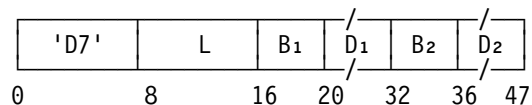
XG R₁,D₂(X₂,B₂) [RXE]



XI D₁(B₁),I₂ [SI]



XC D₁(L,B₁),D₂(B₂) [SS]



第 1 オペランドと第 2 オペランドの排他論理和 (EXCLUSIVE OR) が第 1 オペランド位置に入れます。

連結子 EXCLUSIVE OR は、両方のオペランドにビット単位で適用されます。2 つのオペランドの対応するビット位置内のビットが異なる場合は結果のビット位置の内容が 1 に設定され、それ以外の場合は結果ビットが 0 に設定されます。

EXCLUSIVE OR (XC) では、各オペランドが左から右へと処理されます。オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理さ

れ、必要なオペランド・バイトを取り出した直後に各結果バイトが格納される時と同様の方法で、結果が取得されます。

EXCLUSIVE OR (XI) では、第 1 オペランドの長さは 1 バイトであり、1 バイトのみが格納されます。

オペランドは、EXCLUSIVE OR (XR、X) では 32 ビットで、EXCLUSIVE OR (XGR、XG) では 64 ビットです。

結果の条件コード:

- 0 結果は 0 です。
- 1 結果は 0 ではありません。
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 2、X、XG、および XC。取り出しおよび格納: オペランド 1、XI および XC)

プログラミング上の注意:

1. EXCLUSIVE OR 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. EXCLUSIVE OR は、ビットを反転するために使用できます。この操作は、プログラム式 2 進スイッチのテストおよび設定をするときに特に便利です。
3. あるフィールドとそのフィールド自体との EXCLUSIVE OR 演算の結果は、すべて 0 になります。
4. EXCLUSIVE OR (XR または XGR) では、A EXCLUSIVE-OR B、B EXCLUSIVE-OR A、A EXCLUSIVE-OR B というシーケンスを実行すると、追加の汎用レジスターを使用せずに A と B の内容が交換されます。
5. EXCLUSIVE OR (XI) および EXCLUSIVE OR (XC) の第 1 オペランドへのアクセスは、第 1 オペランドの 1 バイトを記憶域から取り出す操作と、更新済みの値を格納する後続操作から成っています。しかし、ある 1 つのバイトについて見ると、必ずしも取り出しアクセスの直後に格納アクセスが行われるとは限りません。このため、他の CPU またはチャンネル・プログラムが更新する可能性がある記憶位置を EXCLUSIVE OR を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』

の『マルチプログラミングとマルチプロセッシングの例』で OR (OI) について示してあります。

EXECUTE

EX R₁,D₂(X₂,B₂) [RX]

'44'	R ₁	X ₂	B ₂	D ₂	
0	8	12	16	20	31

第 2 オペランド・アドレスにある単一命令が汎用レジスタ R₁ の内容により変更され、結果の命令 (ターゲット命令といいます) が実行されます。

R₁ フィールドが 0 でない場合は、第 2 オペランド・アドレスが示す命令のビット 8-15 と汎用レジスタ R₁ のビット 56-63 との論理和が求められます。この論理和演算は、汎用レジスタ R₁ の内容も記憶域内の命令も変更することではなく、実行する命令を解釈する働きのみをします。R₁ フィールドが 0 の場合は、論理和演算は実行されません。

ターゲット命令の長さは、2 バイト、4 バイト、または 6 バイトです。ターゲット命令の実行および例外処理は、命令アドレスおよび命令長コードを除き、そのターゲット命令が通常の順次操作で取得された場合とまったく同じです。

現 PSW 内の命令アドレスは、EXECUTE の長さの分だけ増加します。この更新済みアドレスおよび EXECUTE の命令長コードは、例えば、ターゲット命令が BRANCH AND LINK の場合にはリンク情報として使用されます。ターゲット命令が正常なブランチ命令の場合は、現 PSW 内の命令アドレスが、ターゲット命令で指定されるブランチ・アドレスで置き換えられません。

ターゲット命令が逆に EXECUTE を実行した場合は、実行例外が認識されます。

EXECUTE の有効アドレスは偶数でなければなりません。さもないと、指定例外が認識されます。ターゲット命令の長さが 2 または 3 ハーフワードで、2 番目または 3 番目のハーフワードを取り出さなくても実行できる命令の場合は、その使用されないハーフワードについて

アクセス例外が認識されるかどうかは予測不能です。第 2 オペランド・アドレスが奇数の場合は、そのアドレスについてはアクセス例外は認識されません。

EXECUTE の第 2 オペランド・アドレスは、論理アドレスでなく命令アドレスです。したがって、1 次スペース・モード、2 次スペース・モード、またはアクセス・レジスタ・モードでは、ターゲット命令は 1 次アドレス・スペースから取り出されます。

条件コード: コードはターゲット命令により設定されることがあります。

プログラム例外:

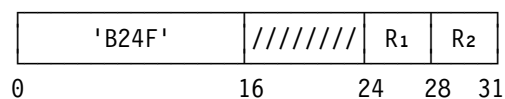
- アクセス例外 (取り出し: ターゲット命令)
- 実行例外
- 指定例外

プログラミング上の注意:

1. EXECUTE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 汎用レジスタからの 8 ビットと指定の命令との論理和演算によって、長さ、指標、マスク、中間データ、レジスタ、または拡張命令コード・フィールドを間接的に指定することができます。
3. ターゲット命令の取り出しは、プログラム・イベントの記録、およびアクセス例外の報告を目的とする命令取り出しと考えることができます。
4. EXECUTE またはターゲット命令が原因で、アクセス例外または指定例外が起きることがあります。
5. 割り込み可能命令を EXECUTE のターゲットにする場合、プログラムでは、通常、その割り込み可能命令により更新されるレジスタを EXECUTE の R₁、X₂、または B₂ レジスタとして指定してはなりません。さもないと、命令の後の実行再開時、または割り込みなしで同じ命令が再取り出しされたときに、これらのレジスタの更新済みの値が EXECUTE の実行で使用されてしまいます。同様に、通常は、プログラムで、割り込み可能命令の記憶域内の宛先フィールドに、EXECUTE の記憶位置が含まれないようにする必要があります。なぜなら、実行の再開時にその記憶位置の新しい内容が解釈される可能性があるからです。

EXTRACT ACCESS

EAR R₁,R₂ [RRE]



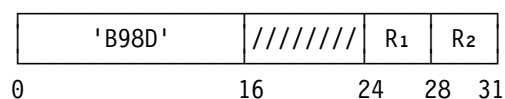
アクセス・レジスタ R₂ の内容が、汎用レジスタ R₁ のビット位置 32-63 に入れます。汎用レジスタ R₁ のビット 0-31 は変更されません。

条件コード: コードは変更されません。

プログラム例外: なし。

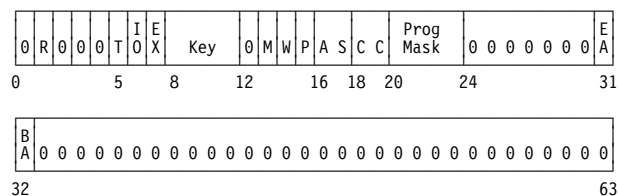
EXTRACT PSW

EPSW R₁,R₂ [RRE]



現 PSW のビット 0-31 が第 1 オペランドのビット位置 32-63 に入れられ、このオペランドのビット 0-31 は無変更のままになります。その後、現 PSW のビット 32-63 が第 2 オペランドのビット位置 32-63 に入れられ、このオペランドのビット 0-31 は無変更のままになります。R₂ が 0 の場合は、第 2 オペランドに関連したアクションは実行されません。

PSW のビット 0-63 のフォーマットは以下のとおりです。

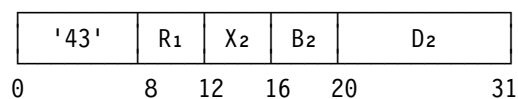


条件コード: コードは変更されません。

プログラム例外: なし。

INSERT CHARACTER

IC R₁,D₂(X₂,B₂) [RX]



第 2 オペランド位置にあるバイトが、汎用レジスタ R₁ のビット位置 56-63 に挿入されます。レジスタ内の残りのビットは変更されません。

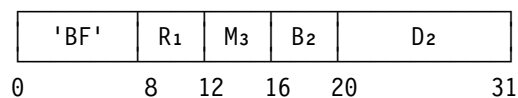
条件コード: コードは変更されません。

プログラム例外:

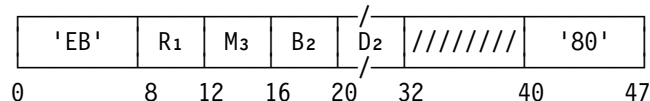
- アクセス例外 (取り出し: オペランド 2)

INSERT CHARACTERS UNDER MASK

ICM R₁,M₃,D₂(B₂) [RS]



ICMH R₁,M₃,D₂(B₂) [RSE]



第 2 オペランド・アドレスから始まる連続した記憶位置から取り出されたバイトが、マスクの制御下で汎用レジスタ R₁ に挿入されます。

M₃ フィールドの内容がマスクとして使用されます。この 4 個のビットは、左から右への順で、汎用レジスタ R₁ の左から右への 4 個のバイトと、1 対 1 で対応しています。INSERT CHARACTERS UNDER MASK (ICM) では、マスク・ビットが対応する 4 個のバイトは、汎用レジスタ R₁ のビット位置 32-63 に入っています。INSERT CHARACTERS UNDER MASK (ICMH) では、この 4 個のバイトはレジスタの高位の半分で、ビット位置 0-31 を占めています。マスク内の 1 に対応するバイト位置が、左から右へと、第 2 オペランド・アドレスから始まる連続記憶位置のバイトで埋められます。マスクが 0 でない場合は、第 2 オペランドの長さはマスク内の 1 の個数と同じになります。マスク内の 0 に対応する汎用レジスタ内のバイトは、変更さ

れません。ICM ではレジスタのビット 0-31 は変更されず、ICMH ではビット 32-63 は変更されません。

結果の条件コードは、マスク、および挿入されるビットの値に基づいて決まります。マスクが 0 であるか、またはすべての挿入ビットが 0 の場合は、条件コードは 0 に設定されます。挿入ビットのすべてが 0 ではない場合は、コードは記憶域オペランドの左端ビットに従って設定されます。つまり、このビットが 1 ならコードは 1 に設定され、このビットが 0 ならコードは 2 に設定されます。

マスクが 0 でない場合は、マスクによって指定された数のバイトについてのみ、記憶域オペランド・アクセスに関連した例外が認識されます。マスクが 0 の場合は、第 2 オペランド・アドレスにある 1 バイトについて、アクセス例外が認識されます。

結果の条件コード:

- 0 すべての挿入ビットが 0 か、またはマスク・ビットのすべてが 0 です。
- 1 左端の挿入ビットが 1 です。
- 2 左端の挿入ビットが 0 で、挿入ビットがすべて 0 ではありません。
- 3 --

プログラム例外:

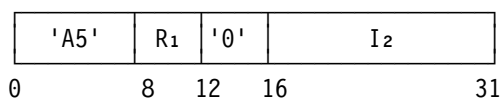
- アクセス例外 (取り出し: オペランド 2)

プログラミング上の注意:

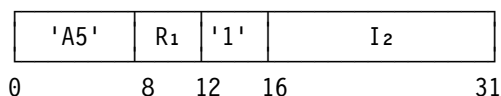
1. INSERT CHARACTERS UNDER MASK 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. INSERT CHARACTERS UNDER MASK (ICM のみ) の条件コードの定義により、マスクが 1111 のときは、この命令では LOAD AND TEST (LTR のみ) の場合に設定されるコードと同じ条件コードが設定されます。したがって、この命令は、記憶域からレジスタへのロードおよびテストの操作に使用できます。
3. INSERT CHARACTERS UNDER MASK (ICM) は、マスク 1111 または 0001 を指定されている場合、条件コードの設定以外は、それぞれ LOAD (L) 命令または INSERT CHARACTER (IC) 命令と同じ機能を実行します。ただし、パフォーマンスの点では INSERT CHARACTERS UNDER MASK の方が遅くなることがあります。

INSERT IMMEDIATE

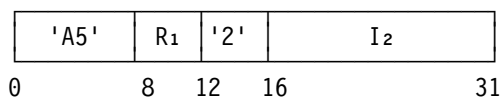
IIHH R₁, I₂ [RI]



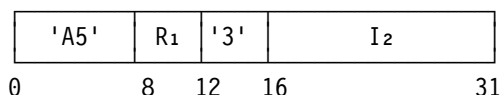
IIHL R₁, I₂ [RI]



II LH R₁, I₂ [RI]



II LL R₁, I₂ [RI]



第 2 オペランドが第 1 オペランドのビット位置に入れます。第 1 オペランドのその他のビットは変更されません。

命令別に見ると、第 2 オペランドがロードされる第 1 オペランドのビット位置は次のとおりです。

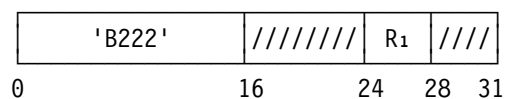
命令	ロードされる ビット位置
IIHH	0-15
IIHL	16-31
II LH	32-47
II LL	48-63

条件コード: コードは変更されません。

プログラム例外: なし。

INSERT PROGRAM MASK

IPM R₁ [RRE]



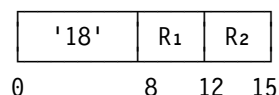
現 PSW から取り出された条件コードおよびプログラム・マスクが、それぞれ、汎用レジスター R₁ のビット位置 34 および 35、および 36-39 に挿入されます。このレジスターのビット 32 および 33 は 0 に設定され、ビット 0-31 および 40-63 は変更されません。

条件コード: コードは変更されません。

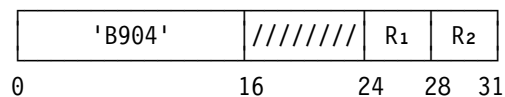
プログラム例外: なし。

LOAD

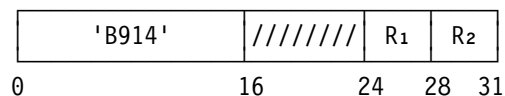
LR R₁,R₂ [RR]



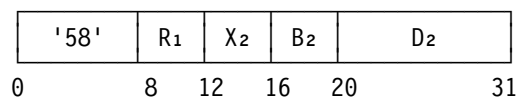
LGR R₁,R₂ [RRE]



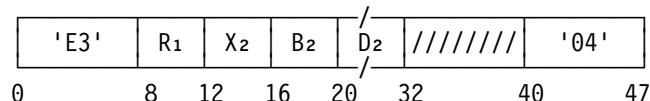
LGFR R₁,R₂ [RRE]



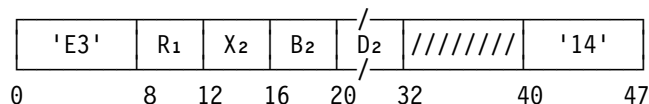
L R₁,D₂(X₂,B₂) [RX]



LG R₁,D₂(X₂,B₂) [RXE]



LGF R₁,D₂(X₂,B₂) [RXE]



第 2 オペランドが、変更なしで第 1 オペランド位置に入られます。ただし、LOAD (LGFR, LGF) の場合は、第 2 オペランドは符号拡張されます。

オペランドは、LOAD (LR, L) では 32 ビットで、LOAD (LGR, LG) では 64 ビットです。LOAD (LGFR, LGF) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、第 1 オペランドは 64 ビットの符号付き 2 進整数として扱われます。

条件コード: コードは変更されません。

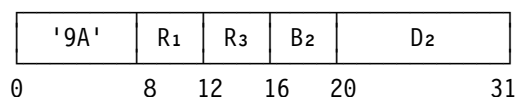
プログラム例外:

- アクセス例外 (取り出し: L, LG、および LGF のオペランド 2 のみ)

プログラミング上の注意: LOAD 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。

LOAD ACCESS MULTIPLE

LAM R₁,R₃,D₂(B₂) [RS]



アクセス・レジスター R₁ から始まりアクセス・レジスター R₃ で終わるアクセス・レジスターのセットが、第 2 オペランド・アドレスが指す記憶位置からロードされます。

アクセス・レジスターの内容が取り出される記憶域は、第 2 オペランド・アドレスが指す位置から始まり、指定されたアクセス・レジスターの数と同数の記憶域ワードに達するまで続きます。これらのアクセス・レジスターは、アクセス・レジスター R₁ から始まりアクセス・レジスター R₃ まで (このレジスターも含む)、レジスター番号の昇順に従ってロードされます。アクセス・レジスター 0 はアクセス・レジスター 15 の後に続きます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 指定例外

LOAD ADDRESS

LA R₁,D₂(X₂,B₂) [RX]

'41'	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20
				31

X₂、B₂、および D₂ フィールドにより指定されるアドレスが、汎用レジスター R₁ に入れます。アドレスの計算は、アドレス演算の規則に従って行われます。

24 ビット・アドレッシング・モードでは、アドレスはビット位置 40-63 に入れられ、ビット 32-39 は 0 に設定され、ビット 0-31 は変更されません。31 ビット・アドレッシング・モードでは、アドレスはビット位置 33-63 に入れられ、ビット 32 は 0 に設定され、ビット 0-31 は変更されません。64 ビット・アドレッシング・モードでは、アドレスはビット位置 0-63 に入れられます。

オペランドに関する記憶域参照は行われず、アドレスについてアクセス例外が検査されることもありません。

条件コード: コードは変更されません。

プログラム例外: なし。

プログラミング上の注意:

1. LOAD ADDRESS 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. LOAD ADDRESS は、命令の D₂ フィールドの内容を増分として、レジスター 0 以外の汎用レジスターの右端ビットを増加させるために使用できます。増加させるレジスターを R₁、および X₂ (B₂ を 0 に設定) または B₂ (X₂ を 0 に設定) で指定します。この命令は、24 ビット・アドレッシング・モードでは 24 ビットを更新し、31 ビット・アドレッシング・モードでは 31 ビットを更新し、64 ビット・アドレッシング・モードでは 64 ビットを更新します。

LOAD ADDRESS EXTENDED

LAE R₁,D₂(X₂,B₂) [RX]

'51'	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20
				31

X₂、B₂、および D₂ フィールドにより指定されるアドレスが、汎用レジスター R₁ に入れます。アクセス・レジスター R₁ にロードされる値は、アドレス・スペース制御ビット (PSW のビット 16 および 17) の現在の値によって決まります。アドレス・スペース制御ビットが 2 進数 01 のときは、このアクセス・レジスターに入る値は、B₂ フィールドが 0 か 0 でないかによって異なります。

アドレスの計算は、アドレス演算の規則に従って行われます。24 ビット・アドレッシング・モードでは、アドレスは汎用レジスター R₁ のビット位置 40-63 に入れられ、ビット 32-39 は 0 に設定され、ビット 0-31 は変更されません。31 ビット・アドレッシング・モードでは、アドレスはビット位置 33-63 に入れられ、ビット 32 は 0 に設定され、ビット 0-31 は変更されません。64 ビット・アドレッシング・モードでは、アドレスはビット位置 0-63 に入れられます。

アクセス・レジスター R₁ に入られる値は、次の表のとおりです。

PSW の
ビット
16 およ
び 17

アクセス・レジスター R₁ に入れられる値

00	16 進数 00000000 (ビット位置 0-31 が 0)
10	16 進数 00000001 (ビット位置 0-30 が 0 で、ビット位置 31 が 1)
01	B ₂ フィールドが 0 の場合: 16 進数 00000000 (ビット位置 0-31 が 0) B ₂ フィールドが 0 でない場合: アクセス・レジスター B ₂ の内容
11	16 進数 00000002 (ビット位置 0-29 および 31 が 0、ビット位置 30 が 1)

ただし、PSW のビット 16 および 17 が 2 進数 01 で、B₂ フィールドが 0 でない場合は、アクセス・レジスター B₂ のビット位置 0-6 にはすべて 0 が含まれて

いる必要があります。さもないと、汎用レジスター R₁ およびアクセス・レジスター R₁ 中の結果は予測不能になります。

オペランドに関する記憶域参照は行われず、アドレスについてアクセス例外が検査されることもありません。

条件コード: コードは変更されません。

プログラム例外: なし。

プログラミング上の注意:

1. DAT がオンのときは、次のように、アドレス・スペース制御ビットのそれぞれの値が変換モードに対応します。

PSW のビット

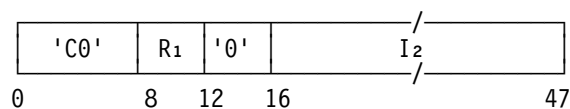
16 および 17 変換モード

00	1 次スペース・モード
10	2 次スペース・モード
01	アクセス・レジスター・モード
11	ホーム・スペース・モード

2. アクセス・レジスター・モードでは、アクセス・レジスター内の 16 進数値 00000000 は 1 次アドレス・スペースを指定し、16 進数値 00000001 は 2 次アドレス・スペースを指定します。制御プログラムがディスパッチ可能単位アクセス・リストのエントリー 2 をホーム・アドレス・スペースを指定するものとして割り当てており、そのエントリーにアクセス・リスト・エントリー・シーケンス番号 (ALESN) 0 を入れていれば、16 進数値 00000002 は、ホーム・アドレス・スペースを指定します。

LOAD ADDRESS RELATIVE LONG

LARL R₁, I₂ [RIL]



I₂ フィールドにより指定されるアドレスが、汎用レジスター R₁ に入れられます。アドレスの計算は、BRANCH RELATIVE ON CONDITION LONG およ

び BRANCH RELATIVE AND SAVE LONG のブランチ・アドレスの規則に従って行われます。

24 ビット・アドレッシング・モードでは、アドレスはビット位置 40-63 に入れられ、ビット 32-39 は 0 に設定され、ビット 0-31 は変更されません。31 ビット・アドレッシング・モードでは、アドレスはビット位置 33-63 に入れられ、ビット 32 は 0 に設定され、ビット 0-31 は変更されません。64 ビット・アドレッシング・モードでは、アドレスはビット位置 0-63 に入れられます。

I₂ フィールドの内容は符号付き 2 進整数で、計算済みアドレスを生成するために命令のアドレスに加算するハーフワードの個数を指定します。

オペランドに関する記憶域参照は行われず、アドレスについてアクセス例外が検査されることもありません。

条件コード: コードは変更されません。

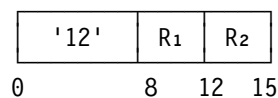
プログラム例外: なし。

プログラミング上の注意:

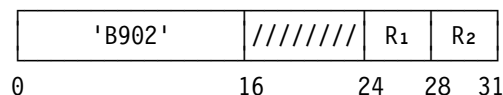
1. 偶数アドレス (ハーフワード・アドレス) のみが生成可能です。奇数アドレスが必要な場合は、LOAD ADDRESS RELATIVE LONG を使用して、LOAD ADDRESS RELATIVE LONG で形成したアドレスに 1 を加算することができます。
2. LOAD ADDRESS RELATIVE LONG が EXECUTE のターゲットのときは、生成されるアドレスは EXECUTE 命令でなく、LOAD ADDRESS RELATIVE LONG 命令の位置を基準とする相対アドレスです。これは、相対ブランチ命令の動作とも一貫しています。

LOAD AND TEST

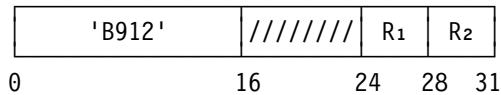
LTR R₁, R₂ [RR]



LTGR R₁, R₂ [RRE]



LTGFR R₁,R₂ [RRE]



第 2 オペランドが、変更なしで第 1 オペランド位置に入れます。ただし、LOAD AND TEST (LTGFR) の場合は、第 2 オペランドは符号拡張されます。第 2 オペランドの符号と絶対値は、符号付き 2 進整数として扱われ、条件コードに示されます。

オペランドは、LOAD AND TEST (LTR) では 32 ビットで、LOAD AND TEST (LTGR) では 64 ビットです。LOAD AND TEST (LTGFR) では、第 2 オペランドは 32 ビットで、第 1 オペランドは 64 ビットの符号付き 2 進整数として扱われます。

結果の条件コード:

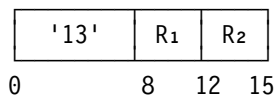
- 0 結果は 0 です。
- 1 結果が 0 より小さい。
- 2 結果が 0 より大きい。
- 3 --

プログラム例外: なし。

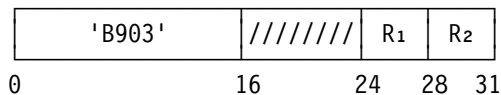
プログラミング上の注意: LOAD AND TEST (LTR および LTGR) で、R₁ フィールドと R₂ フィールドが同じレジスターを指す場合、この操作はデータ移動を伴わないテストと同等です。

LOAD COMPLEMENT

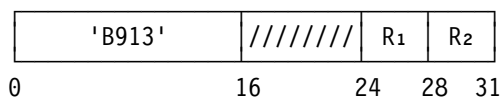
LCR R₁,R₂ [RR]



LCGR R₁,R₂ [RRE]



LCGFR R₁,R₂ [RRE]



第 2 オペランドの 2 の補数が、第 1 オペランド位置に入れます。LOAD COMPLEMENT (LCR) では、第 2 オペランドおよび結果は、32 ビットの符号付き 2 進整数として扱われます。LOAD COMPLEMENT (LCGR) では、これらは 64 ビットの符号付き 2 進整数として扱われます。LOAD COMPLEMENT (LCGFR) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、結果は 64 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がりを許し、符号ビット位置からの繰り上がりは無視する方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

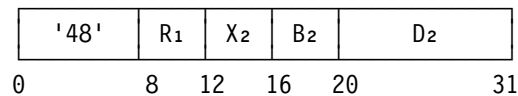
プログラム例外:

- 固定小数点オーバーフロー例外

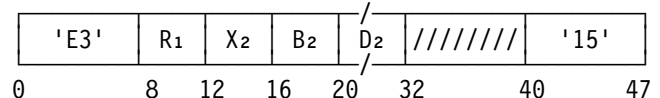
プログラミング上の注意: この操作は、すべての数の補数を生成します。0 は変更されません。LCR または LCGR では、それぞれ、32 ビットまたは 64 ビットの最大負数自体は変更されません。最大負数の補数が生成されるとオーバーフロー条件が起きます。LCGFR は、オーバーフローを起こさずに、32 ビットの最大負数の補数を生成します。

LOAD HALFWORD

LH R₁,D₂(X₂,B₂) [RX]

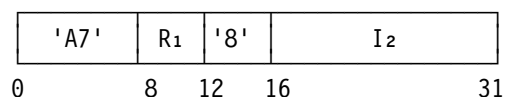


LGH R₁,D₂(X₂,B₂) [RXE]

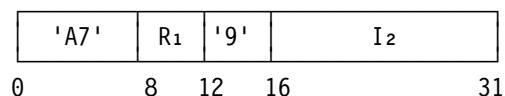


LOAD HALFWORD IMMEDIATE

LHI R₁, I₂ [RI]



LGHI R₁, I₂ [RI]



第2オペランドが符号拡張され、第1オペランド位置に入れられます。第2オペランドは、長さが2バイトで、16ビットの符号付き2進整数として扱われます。LOAD HALFWORD (LH) および LOAD HALFWORD IMMEDIATE (LHI) では、第1オペランドは32ビットの符号付き2進整数として扱われます。LOAD HALFWORD (LGH) および LOAD HALFWORD IMMEDIATE (LGHI) では、第1オペランドは64ビットの符号付き2進整数として扱われます。

条件コード: コードは変更されません。

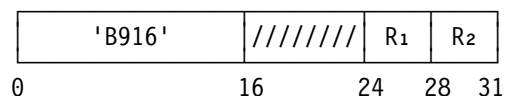
プログラム例外:

- アクセス例外 (取り出し: LH および LGH のオペランド 2)

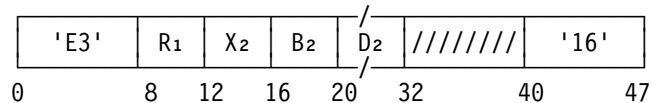
プログラミング上の注意: LOAD HALFWORD 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。

LOAD LOGICAL

LLGFR R₁, R₂ [RRE]



LLGF R₁, D₂ (X₂, B₂) [RXE]



4バイトの第2オペランドが、汎用レジスタ R₁ のビット位置 32-63 に入れられ、0 が汎用レジスタ R₁ のビット位置 0-31 に入れられます。

LOAD LOGICAL (LLGFR) では、第2オペランドは汎用レジスタ R₂ のビット位置 32-63 に入ります。

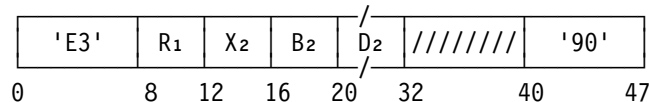
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: LLGF のオペランド 2 のみ)

LOAD LOGICAL CHARACTER

LLGC R₁, D₂ (X₂, B₂) [RXE]



1バイトの第2オペランドが、汎用レジスタ R₁ のビット位置 56-63 に入れられ、0 が汎用レジスタ R₁ の 0-55 に入れられます。

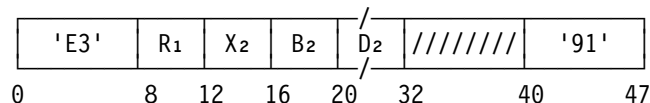
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)

LOAD LOGICAL HALFWORD

LLGH R₁, D₂ (X₂, B₂) [RXE]



2バイトの第2オペランドが、汎用レジスタ R₁ のビット位置 48-63 に入れられ、0 が汎用レジスタ R₁ のビット位置 0-47 に入れられます。

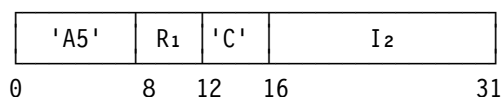
条件コード: コードは変更されません。

プログラム例外:

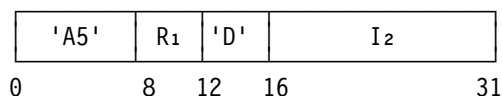
- アクセス例外 (取り出し: オペランド 2)

LOAD LOGICAL IMMEDIATE

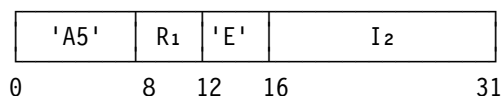
LLIHH R₁, I₂ [RI]



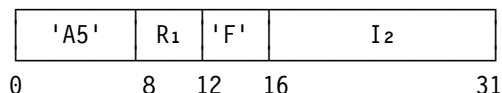
LLIHL R₁, I₂ [RI]



LLILH R₁, I₂ [RI]



LLILL R₁, I₂ [RI]



第 2 オペランドが第 1 オペランドのビット位置に入れます。第 1 オペランドの残りの部分は 0 に設定されます。

命令別に見ると、第 2 オペランドがロードされる第 1 オペランドのビット位置は次のとおりです。

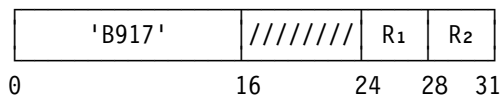
命令	ロードされるビット位置
LLIHH	0-15
LLIHL	16-31
LLILH	32-47
LLILL	48-63

条件コード: コードは変更されません。

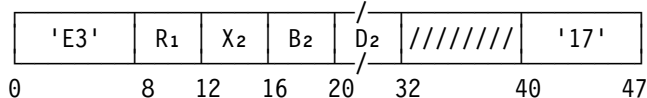
プログラム例外: なし。

LOAD LOGICAL THIRTY ONE BITS

LLGTR R₁, R₂ [RRE]



LLGT R₁, D₂ (X₂, B₂) [RXE]



LLGTR では、汎用レジスタ R₂ のビット 33-63 が、その左側に 33 個の 0 が付加されて、汎用レジスタ R₁ に入れます。LLGT では、第 2 オペランド位置にある 4 バイトのビット 0-31 が、その左側に 33 個の 0 を付加されて、汎用レジスタ R₁ に入れます。

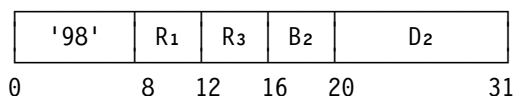
条件コード: コードは変更されません。

プログラム例外:

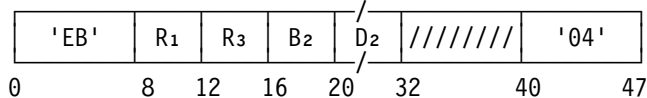
- アクセス例外 (取り出し: LLGT のオペランド 2 のみ)

LOAD MULTIPLE

LM R₁, R₃, D₂ (B₂) [RS]



LMG R₁, R₃, D₂ (B₂) [RSE]



汎用レジスタ R₁ から始まり、汎用レジスタ R₃ で終わる一連のレジスタのビット位置に、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置の内容がロードされます。

LOAD MULTIPLE (LM) では、これらの汎用レジスタのビット位置 32-63 に、第 2 オペランド・アドレスから始まる 4 バイトの連続フィールドの内容がロードされ、レジスタのビット 0-31 は変更されません。

LOAD MULTIPLE (LMG) では、これらの汎用レジスタのビット位置 0-63 に、第 2 オペランド・アドレスから始まる 8 バイトの連続フィールドの内容がロードされます。

これらの汎用レジスタは、汎用レジスタ R₁ から始まり汎用レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従ってロードされます。汎用レジスタ 0 は、汎用レジスタ 15 の後に続きます。

条件コード: コードは変更されません。

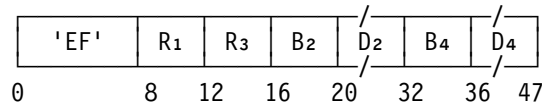
プログラム例外:

- アクセス例外 (取り出し: オペランド 2)

プログラミング上の注意: R₁ および R₃ により指定されるレジスタ番号のすべての組み合わせが有効です。レジスタ番号が等しい場合は、LM の場合は 4 バイトのみ、LMG の場合は 8 バイトのみが伝送されます。R₃ で指定された番号が R₁ で指定された番号より小さい場合は、レジスタ番号は 15 から 0 に循環します。

LOAD MULTIPLE DISJOINT

LMD R₁,R₃,D₂(B₂),D₄(B₄) [SS]



汎用レジスタ R₁ から始まり、汎用レジスタ R₃ で終わる一連のレジスタのビット位置 0-31 に、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置の内容がロードされます。同様に、同じ一連のレジスタのビット位置 32-63 に、第 4 オペランド・アドレスが指す記憶位置から始まる記憶域の内容がロードされます。

これらの汎用レジスタは、汎用レジスタ R₁ から始まり汎用レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従ってロードされます。汎用レジスタ 0 は、汎用レジスタ 15 の後に続きます。

条件コード: コードは変更されません。

プログラム例外:

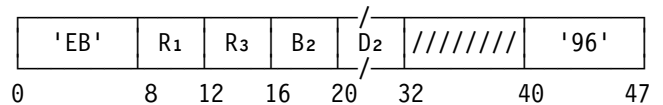
- アクセス (取り出し: オペランド 2 および 4)

プログラミング上の注意:

1. R₁ および R₃ により指定されるレジスタ番号のすべての組み合わせが有効です。レジスタ番号が等しい場合は、8 バイトのみが伝送されます。R₃ で指定された番号が R₁ で指定された番号より小さい場合は、レジスタ番号は 15 から 0 に循環します。
2. 第 2 オペランドおよび第 4 オペランドのアドレスは、レジスタの内容が変更される前に計算されます。
3. LOAD MULTIPLE 命令と LOAD MULTIPLE HIGH 命令を組み合わせると、同じレジスタ範囲に対する LOAD MULTIPLE DISJOINT 命令と同じかそれ以上のパフォーマンスが得られます。LOAD MULTIPLE DISJOINT は、ロードされるレジスタの 1 つを使用して第 2 オペランドまたは第 4 オペランドをアドレッシングする必要がある場合に使用するためのものです。

LOAD MULTIPLE HIGH

LMH R₁,R₃,D₂(B₂) [RSE]



汎用レジスタ R₁ から始まり、汎用レジスタ R₃ で終わる一連のレジスタの高位半分、つまりビット位置 0-31 に、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置の内容がロードされます。つまり、ビット位置 0-31 に、第 2 オペランド・アドレスから始まる連続する 4 バイト・フィールドの内容がロードされます。これらのレジスタのビット 32-63 は変更されません。

これらの汎用レジスタは、汎用レジスタ R₁ から始まり汎用レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従ってロードされます。汎用レジスタ 0 は、汎用レジスタ 15 の後に続きます。

条件コード: コードは変更されません。

プログラム例外:

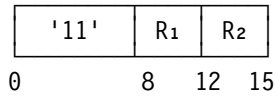
- アクセス例外 (取り出し: オペランド 2)

プログラミング上の注意: R₁ および R₃ により指定されるレジスタ番号のすべての組み合わせが有効です。レジスタ番号が等しい場合は、4 バイトのみが伝送さ

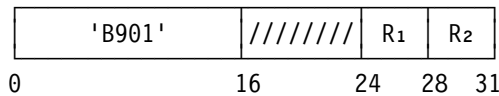
れます。R₃ で指定された番号が R₁ で指定された番号より小さい場合は、レジスター番号は 15 から 0 に循環します。

LOAD NEGATIVE

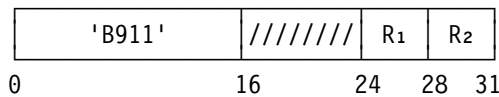
LNR R₁,R₂ [RRE]



LNGR R₁,R₂ [RRE]



LNGFR R₁,R₂ [RRE]



第 2 オペランドの絶対値の 2 の補数が、第 1 オペランド位置に入れられます。LOAD NEGATIVE (LNR) では、第 2 オペランドおよび結果は 32 ビットの符号付き 2 進整数として扱われ、LOAD NEGATIVE (LNGR) では、第 2 オペランドおよび結果は 64 ビットの符号付き 2 進整数として扱われます。LOAD NEGATIVE (LNGFR) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、結果は 64 ビットの符号付き 2 進整数として扱われます。

結果の条件コード:

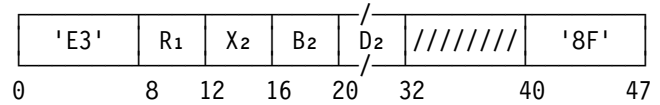
- 0 結果は 0。
- 1 結果が 0 より小さい。
- 2 --
- 3 --

プログラム例外: なし。

プログラミング上の注意: この操作は正数の補数を生成します。負数は変更されません。数値 0 は変更されません。

LOAD PAIR FROM QUADWORD

LPQ R₁,D₂(X₂,B₂) [RXE]



クワッドワードの第 2 オペランドが第 1 オペランド位置にロードされます。他の CPU からは、第 2 オペランドは、クワッドワード単位で取り出されているように見えます。クワッドワードの左寄りのダブルワードは汎用レジスター R₁ にロードされ、右寄りのダブルワードは汎用レジスター R₁ + 1 にロードされます。

R₁ フィールドは、汎用レジスターの奇偶ペアを指示するもので、偶数番号のレジスターを指定している必要があります。第 2 オペランドは、クワッドワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

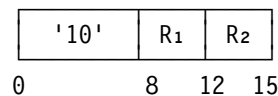
プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 指定例外

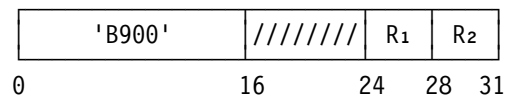
プログラミング上の注意: LOAD MULTIPLE (LM または LMG) 命令は、必ずしもクワッドワード単位のアクセスを提供するわけではありません。

LOAD POSITIVE

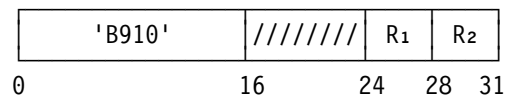
LPR R₁,R₂ [RR]



LPGR R₁,R₂ [RRE]



LPGFR R₁,R₂ [RRE]



第 2 オペランドの絶対値が、第 1 オペランド位置に入れます。LOAD POSITIVE (LPR) では、第 2 オペランドおよび結果は 32 ビットの符号付き 2 進整数として扱われ、LOAD POSITIVE (LPGR) では、これらは 64 ビットの符号付き 2 進整数として扱われます。

LOAD POSITIVE (LPGFR) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、結果は 64 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がり許し、符号ビット位置からの繰り上がりは無視する方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 --
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

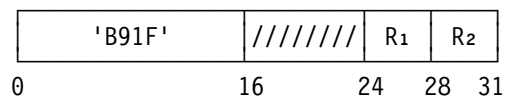
プログラム例外:

- 固定小数点オーバーフロー例外

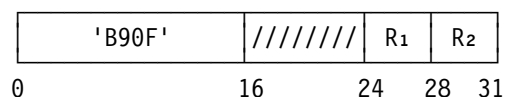
プログラミング上の注意: この操作は、負数の補数を生成します。正数および 0 は変更されません。LPR または LPGR では、それぞれ、32 ビットまたは 64 ビットの最大負数の補数が生成されるとオーバーフロー条件が起きます。最大負数自体は変更されません。LPGFR は、オーバーフローを起こさずに 32 ビットの最大負数の補数を生成します。

LOAD REVERSED

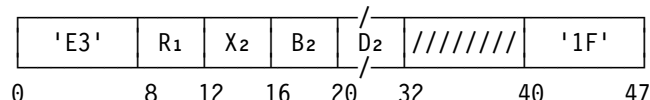
LRVR R₁,R₂ [RRE]



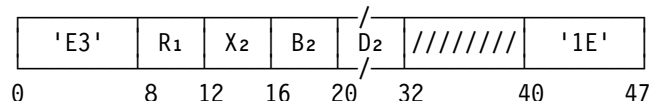
LRVGR R₁,R₂ [RRE]



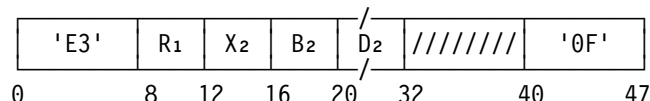
LRVH R₁,D₂(X₂,B₂) [RXE]



LRV R₁,D₂(X₂,B₂) [RXE]



LRVG R₁,D₂(X₂,B₂) [RXE]



第 2 オペランドが、左から右へのバイト順序を反転させて、第 1 オペランド位置に入れます。

LOAD REVERSED (LRVH) では、第 2 オペランドは 2 バイトで、結果は汎用レジスタ R₁ のビット位置 48-63 に入れられ、このレジスタのビット 0-47 は変更されません。

LOAD REVERSED (LRVR, LRV) では、第 2 オペランドは 4 バイトで、結果は汎用レジスタ R₁ のビット位置 32-63 に入れられ、このレジスタのビット 0-31 は変更されません。LOAD REVERSED (LRVR) では、第 2 オペランドは汎用レジスタ R₂ のビット位置 32-63 に入ります。

LOAD REVERSED (LRVGR, LRVG) では、第 2 オペランドは 8 バイトです。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: LRVH, LRV, LRVG のオペランド 2 のみ)

プログラミング上の注意:

1. この命令は、2 バイト、4 バイト、または 8 バイトを「リトル・エンディアン」フォーマットから「ビッグ・エンディアン」フォーマットへ変換するため、または逆の変換をするために使用します。ビッグ・エンディアン・フォーマットでは、左から右へのバイト順序は、最上位から最下位への順になります。リトル・エンディアン・フォーマットでは、バイトは最下位から最上位への順に配置されます。例

例えば、ビッグ・エンディアン・フォーマットのバイト ABCD は、リトル・エンディアン・フォーマットでは DCBA になります。

2. **LOAD REVERSED (LRVR)** は、次の例に示すように、レジスター内にすでに入っている 2 バイト値に対して使用できます。この例では、対象の 2 バイトは R1 レジスターのビット位置 48-63 に入っています。

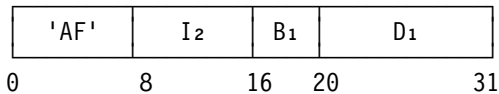
```
LRVR   R1,R1
SRA    R1,16
```

LOAD REVERSED 命令は、対象の 2 バイトを、バイトの順序を反転してレジスターのビット位置 32-47 に入れます。**SHIFT RIGHT SINGLE (SRA)** 命令は、対象の 2 バイトを、レジスターのビット位置 48-63 にシフトして、左側 (つまりビット位置 32-47) を符号ビットで拡張します。対象の 2 バイトが符号なしの値の場合は、この命令でなく、**SHIFT RIGHT SINGLE LOGICAL (SRL)** 命令を使用する必要があります。

3. **LOAD REVERSED** の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86 ページの『記憶域オペランドの一貫性』を参照してください。)

MONITOR CALL

```
MC      D1(B1),I2      [SI]
```



制御レジスター 8 の中の該当のモニター・マスク・ビットが 1 であれば、プログラム割り込みが起きます。

モニター・マスク・ビットは、制御レジスター 8 のビット位置 48-63 に入っており、それぞれモニター・クラス 0-15 に対応しています。

I₂ フィールドのビット位置 12-15 には、16 個のモニター・クラスの 1 つを指定する 2 進数が入っています。I₂ フィールドにより指定されたクラスに対応するモニター・マスク・ビットが 1 であれば、モニター・イベント・プログラム割り込みが起きます。I₂ フィールドの内容は記憶位置 149 に格納され、記憶位置 148 には 0 が格納されます。プログラム割り込みコードのビット 9 は 1 に設定されます。

第 1 オペランド・アドレスはデータのアドレッシングには使用されません。代わりに、B₁ フィールドおよび D₁ フィールドに指定されているアドレスがモニター・コードを形成し、記憶位置 176 にあるダブルワードに格納されます。アドレス計算はアドレスの演算規則に従って行われます。24 ビット・アドレッシング・モードでは、ビット 0-39 は 0 に設定されます。31 ビット・アドレッシング・モードでは、ビット 0-32 は 0 に設定されます。

命令のビット 12-15 に指定されたクラスに対応するモニター・マスク・ビットが 0 のときは、割り込みは起きず、命令はノーオペレーションとして実行されます。

命令のビット位置 8-11 にはすべて 0 が含まれていなければなりません。さもないと、指定例外が認識されません。

条件コード: コードは変更されません。

プログラム例外:

- モニター・イベント例外
- 指定例外

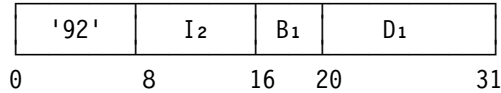
プログラミング上の注意:

1. **MONITOR CALL** は、モニター対象プログラムの選択された地点に到達したときに、モニター・プログラムに制御権を渡すための機能を提供します。これを実現するには、モニター対象プログラム内の望みの場所に **MONITOR CALL** 命令を挿入します。この機能は、さまざまな計測機能を実行するとき役に立ちます。特に、どのプログラムが実行されたかを示すトレース情報を生成したり、特定のプログラムの使用頻度を示すカウント情報を生成したり、特定のプログラムの実行に要する時間を示すタイミング情報を生成する場合に利用できます。
2. モニター・マスクを使用すると、すべてのモニター・イベント・プログラム割り込みを禁止することも、すべてのクラスまたは選択したクラスについてモニター・イベント・プログラム割り込みを許可することもできます。
3. モニター・コードを使用すると、クラス番号に加えて、記述情報を個々の **MONITOR CALL** に関連付けることができます。基底レジスターを使用せずに、最大 4,096 通りの異なるモニター・コードをモニター割り込みに関連付けることができます。基底レジスターを使用する場合 (B₁ フィールドに 0 以外の値が指定されている場合) は、アドレッシング・モードに応じて、個々のモニター割り込みを

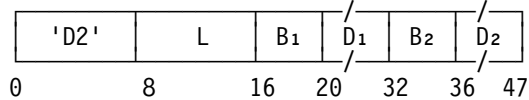
24 ビット、31 ビット、または 64 ビットのコードにより識別することができます。

MOVE

MVI $D_1(B_1), I_2$ [SI]



MVC $D_1(L, B_1), D_2(B_2)$ [SS]



第 2 オペランドが第 1 オペランドの位置に入れられます。

MOVE (MVC) では、各オペランドは左から右へと処理されます。オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理されて必要なオペランド・バイトを取り出した直後に各結果バイトが格納されると同様の方法で結果が取得されます。

MOVE (MVI) では、第 1 オペランドの長さは 1 バイトであり、1 バイトのみが格納されます。

条件コード: コードは変更されません。

プログラム例外:

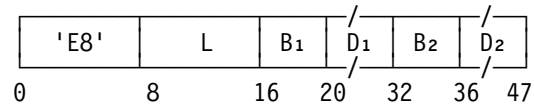
- アクセス例外 (取り出し: MVC のオペランド 2。格納: MVI および MVC のオペランド 1)

プログラミング上の注意:

1. MOVE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 第 2 オペランドの 1 バイト右から第 1 オペランドを開始すると、その 1 バイトをフィールド全体に波及させることができます。

MOVE INVERSE

MVCIN $D_1(L, B_1), D_2(B_2)$ [SS]



第 2 オペランドが、左から右へのバイト順序を逆にし、第 1 オペランド位置に入れられます。

第 1 オペランド・アドレスは、第 1 オペランドの左端のバイトを指定します。第 2 オペランド・アドレスは、第 2 オペランドの右端のバイトを指定します。両方のオペランドの長さは同じです。

結果は、第 2 オペランドを右から左へ、第 1 オペランドを左から右へ処理したものとして求められます。第 2 オペランドは、24 ビット・アドレッシング・モードでは記憶位置 0 から記憶位置 $2^{24} - 1$ へ、31 ビット・アドレッシング・モードでは記憶位置 0 から記憶位置 $2^{31} - 1$ へ、64 ビット・アドレッシング・モードでは記憶位置 0 から記憶位置 $2^{64} - 1$ へと循環可能です。第 1 オペランドは、24 ビット・アドレッシング・モードでは記憶位置 $2^{24} - 1$ から記憶位置 0 へ、31 ビット・アドレッシング・モードでは記憶位置 $2^{31} - 1$ から記憶位置 0 へ、64 ビット・アドレッシング・モードでは記憶位置 $2^{64} - 1$ から記憶位置 0 へと循環可能です。

オペランドが 2 バイト以上オーバーラップする場合は、結果フィールドのオーバーラップ部分の内容は予測不能です。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)

プログラミング上の注意:

1. MOVE INVERSE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 移動される個々のバイトの内容は変更されません。
3. MOVE INVERSE は、第 2 オペランド・アドレスが第 2 オペランドの左端ではなく右端のバイトを指定する唯一の SS フォーマット命令です。

4. MOVE INVERSE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

MOVE LONG

MVCL R₁,R₂ [RR]

'0E'	R ₁	R ₂
0	8	12 15

両方のオペランド位置のオーバーラップが、第 1 オペランド位置の最終内容に影響を及ぼさない場合は、第 2 オペランドが第 1 オペランド位置に入れられます。第 1 オペランド位置の右端に幾つかのバイト位置が残る場合は、その位置は埋め込みバイトで満たされます。

R₁ フィールドおよび R₂ フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。第 1 オペランド位置および第 2 オペランド位置のバイト数は、それぞれ、汎用レジスタ R₁ + 1 および R₂ + 1 のビット位置 40-63 にある符号なし 2 進整数によって指定されます。汎用レジスタ R₂ + 1 のビット位置 32-39 には、埋め込みバイトが含まれています。汎用レジスタ R₁ + 1 のビット位置 0-39、および汎用レジスタ R₂ + 1 のビット位置 0-31 の内容は、無視されます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

7-93ページの図7-16 は、上記で説明したレジスタの内容を示しています。

移動が両方のフィールドの左端から開始され、1 バイトずつ処理されて右へ進むように見える方法で、結果が求められます。汎用レジスタ R₁ + 1 のビット 40-63 で指定された数のバイトが第 1 オペランド位置に移動されると、操作は終了します。第 2 オペランドが第 1 オペランドより短い場合は、第 1 オペランド位置の残りの右端バイトは、埋め込みバイトで満たされます。

この命令の実行の一部として、2 つの長さフィールドの値が比較され、その結果に基づいて条件コードが設定されます。また、2 つのオペランドの間に破壊的オーバーラップがないかどうかも検査されます。論理オペランド・アドレスを使用してオーバーラップの検査を行うことが前提になる場合、第 1 オペランド位置にデータが移動された後で、その位置がソースとして使用されるときは、2 つのオペランドが破壊的にオーバーラップしていると言います。オペランドが破壊的にオーバーラップしているときは、移動は行われず、条件コード 3 が設定されます。

第 1 オペランドの左端バイトが、操作に関与する第 2 オペランドの各バイトのうち、左端の 1 バイトを除くどのバイトとも一致していないときは、その 2 つのオペランドは破壊的オーバーラップをしておらず、したがって移動が実行されます。オペランドが記憶位置 2²⁴ - 1 (または 2³¹ - 1 または 2⁶⁴ - 1) から記憶位置 0 へ循環する場合は、記憶位置 2²⁴ - 1 (または 2³¹ - 1 または 2⁶⁴ - 1) まで (この記憶位置を含む) の記憶位置のオペランド・バイトは、記憶位置 0 以降のバイトの左にあるものと見なされます。

24 ビット・アドレッシング・モードでは記憶位置 2²⁴ - 1 から記憶位置 0 へと循環し、31 ビット・アドレッシング・モードでは記憶位置 2³¹ - 1 から記憶位置 0 へと循環し、64 ビット・アドレッシング・モードでは記憶位置 2⁶⁴ - 1 から記憶位置 0 へと循環します。

アクセス・レジスタ・モードでは、アクセス・レジスタ R₁ およびアクセス・レジスタ R₂ の内容が比較されます。R₁ フィールドまたは R₂ フィールドが 0 なら、アクセス・レジスタ 0 の内容でなく、32 個の 0 が使用されます。比較された値の 32 ビットがすべて等しい場合は、破壊的オーバーラップかどうかのテストが行われます。比較された値の 32 ビットすべてが等しいわけではない場合は、破壊的オーバーラップは存在しないと宣言されます。この場合、オペランドが実際には実記憶域内でオーバーラップしていても、オーバーラップ条件が結果に反映されるかどうかは予測不能です。

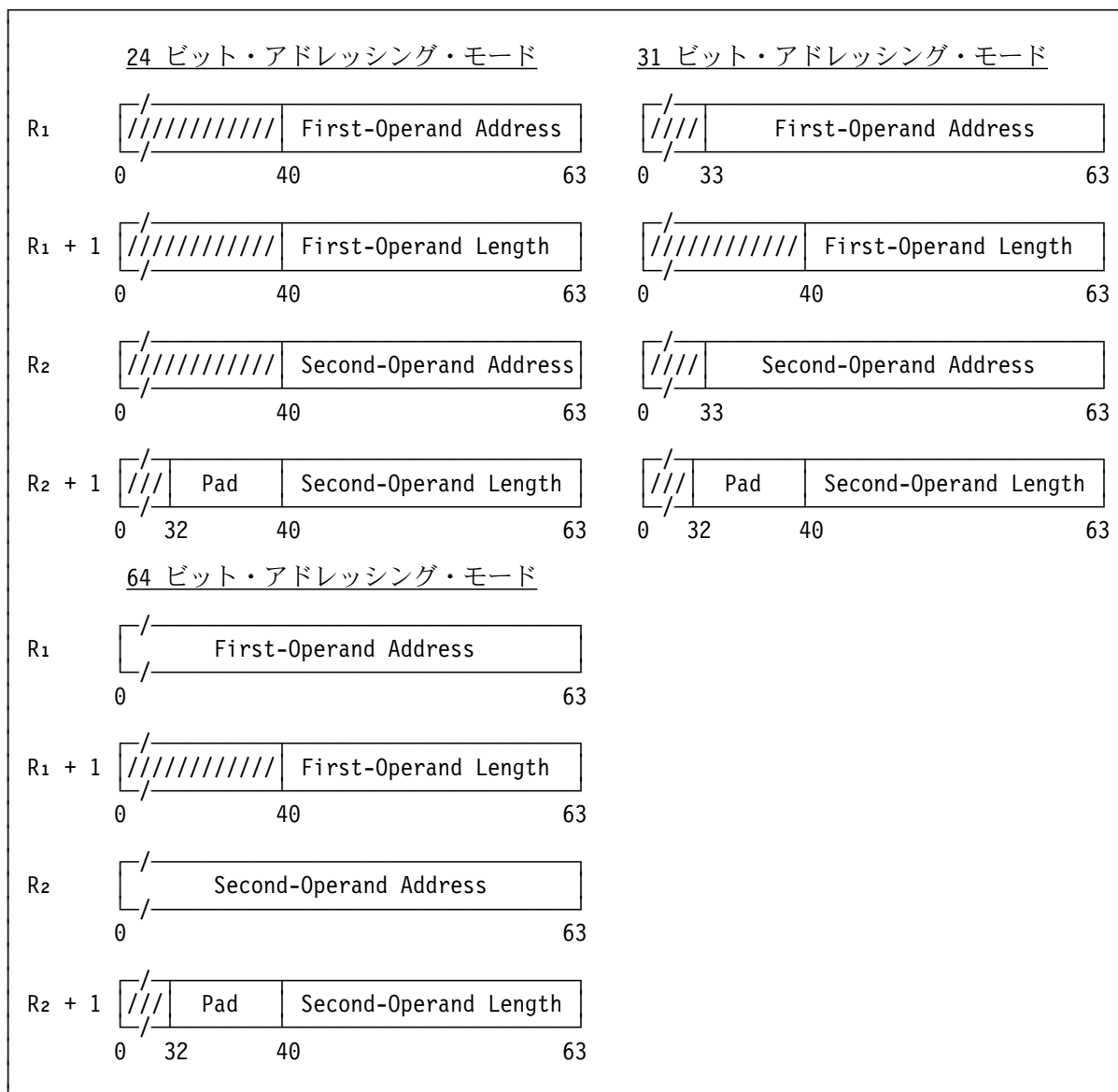


図 7-16. MOVE LONG で使用するレジスタの内容

汎用レジスタ $R_1 + 1$ のビット 40-63 で指定された長さが 0 であれば、移動は行われず、長さの相対値を示す条件コード 0 または 1 が設定されます。

この命令の実行は割り込み可能です。割り込みが起きると、それが操作を中止させる割り込みでない限り、汎用レジスタ $R_1 + 1$ および $R_2 + 1$ 内の長さが移動済みのバイト数だけ減少し、汎用レジスタ R_1 および R_2 内のアドレスが同じバイト数だけ増加しています。これにより、命令の実行が再開されたときは、割り込み点から処理が再開されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R_1 および R_2 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは 0 に設定され、ビット位置 0-31 の

内容は変更されません。どのアドレッシング・モードでも、汎用レジスタ $R_1 + 1$ および $R_2 + 1$ のビット位置 0-39 の内容は変更されず、条件コードは予測不能になります。埋め込み操作の途中でこの操作への割り込みが起きた場合は、汎用レジスタ $R_2 + 1$ の長さフィールドは 0 になり、汎用レジスタ R_2 のアドレスは、汎用レジスタ $R_2 + 1$ の元の長さ分だけ増加します。汎用レジスタ R_1 および $R_1 + 1$ には、埋め込み操作がどこまで進んだかが反映されています。

第 1 オペランド位置にその命令の記憶位置または EXECUTE の記憶位置が含まれている場合は、実行中に割り込みが起きなくても、命令が記憶域から再取り出しされ、再解釈されることがあります。このような再取り出しが実行中のどこで起きるかは予測不能です。

モデルにより場合によっては、操作の非埋め込み部分の実行中に、16 進数 B0 および 16 進数 B8 の 2 つの埋め込みバイト値を使用して、それぞれ、キャッシュをバイパスして移動を行うかキャッシュを使用して移動を行うかを指示する場合があります。この場合、埋め込みバイトの 16 進数 B0 は移動後に宛先領域を参照しないことを指示し、埋め込みバイトの 16 進数 B8 は、宛先領域を参照することを指示します。

操作の非埋め込み部分では、MOVE LONG のオペランドへのアクセスは単一アクセス参照になります。埋め込みバイトが 16 進数 B1 の場合を除き、このようなアクセスは、チャンネル・プログラムおよび他の CPU から見て必ずしも左から右への方向で行われるようには見えません。両オペランドがダブルワード境界から始まっていて、長さがダブルワードの整数倍で、オーバーラップしていない場合は、操作の非埋め込み部分の実行中は、他の CPU からはこれらのオペランドがダブルワード単位でアクセスされているように見えます。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランドの中の埋め込みバイトで満たされる部分は、格納が左から右への方向に行われるとは限りません。また、2 回以上格納されるように見えることもあります。

操作の完了時には、汎用レジスタ R1 + 1 内の長さは、第 1 オペランド位置に格納されているバイトの数だけ減少し、汎用レジスタ R1 内のアドレスは、同じバイト数だけ増加しています。汎用レジスタ R2 + 1 内の長さは、第 2 オペランド位置から取り出して移動されたバイトの数だけ減少し、汎用レジスタ R2 内のアドレスは同じバイト数だけ増加しています。24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R1 および R2 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、どちらかまたは両方の長さの初期値が 0 の場合または条件コード 3 が設定された場合でも、0 に設定されます。両方のレジスタのビット位置 0-31 は変更されません。どのアドレッシング・モードでも、汎用レジスタ R1 + 1 および R2 + 1 のビット位置 0-39 は変更されません。

条件コード 3 が設定された場合、オペランド・アクセスに関連した例外は認識されません。オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。同様に、第 2 オペランドが第 1 オペランドより長い場合も、第 1 オペランド・フィールドより長い第 2 オペランド・フィールド部分については、アクセス例外は認識されません。2K バイトより長いオペランドの場合は、処理中の現在の記憶位置より後の 2K バ

イトを超える記憶位置については、アクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。また、R1 フィールドが奇数の場合は、PER 記憶域変更イベントは認識されず、変更ビットも設定されません。

結果の条件コード:

- 0 オペランド長が等しく、破壊的オーバーラップはありません。
- 1 第 1 オペランドの長さが小さく、破壊的オーバーラップはありません。
- 2 第 1 オペランドの長さが大きく、破壊的オーバーラップはありません。
- 3 破壊的オーバーラップがあるために、移動は行われませんでした。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 指定例外

プログラミング上の注意:

1. MOVE LONG 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. 埋め込みバイトを 0 に設定し、第 2 オペランドの長さを 0 に設定することによって、MOVE LONG を使用して記憶域を消去することができます。ほとんどのモデルでは、256 バイトを超える記憶域を消去するには、これが最も高速の命令です。しかし、この消去に関連した格納は複数アクセス格納になることがあります。したがって、チャンネル・プログラムまたは他の CPU が記憶域を 0 と見なすとすぐにアクセスして使用する可能性がある場合には、そのような記憶域をこの方法で消去してはなりません。詳しくは、5-86 ページの『記憶域オペランドの一貫性』を参照してください。
3. どちらのオペランドについても、アドレッシング例外を起こすような長さをプログラムで指定することは避ける必要があります。アドレッシング (および保護) 例外の結果、現在の操作単位のみでなく、操作全体が中止されることもあります。この中止の結果、すべての結果フィールドの内容が予測不能になることがあります。MOVE LONG の場合、この内容には、条件コードと 2 つの奇偶汎用レジスタ・ペア、および主記憶域の第 1 オペランド位置も含まれます。次に示すのは、幾つかのモデルで実際に生じた状況の例です。

- a. 長さが数ブロックの第 1 オペランドの 4K バイト・ブロックの 1 つで保護例外が起きると、保護対象ブロックへの格納は抑止されます。しかし、第 1 オペランドの保護されていない後続ブロックへの移動は続行されます。同様に、あるブロックでアドレッシング例外が起きても、使用可能な後続ブロックの処理が抑止されるとは限りません。
 - b. 一部のモデルでは、外部割り込み、入出力割り込み、抑制可能なマシン・チェック割り込み、または再始動割り込みが起きたとき、あるいは、操作単位の無効化または抑止を必要とするようなプログラム割り込みが起きたときのみ、汎用レジスターが更新される場合があります。したがって、第 1 オペランドの数ブロックへの移動が行われた後でアドレッシング例外または記憶保護例外が起きた場合、汎用レジスターが変更されないことがあります。
4. 第 1 オペランドの長さが 0 の場合は、条件コードが設定され、24 ビットまたは 31 ビットのアドレッシング・モードでは汎用レジスター R1 および R2 のビット位置 32-63 の左端ビットが 0 に設定されます。
 5. R1 フィールドと R2 フィールドの内容が同じ場合は、指定されているレジスターの内容が、移動されたバイト数の 2 倍でなく、移動されたバイト数と同数だけ増加または減少します。そして、条件コード 0 が設定されます。
 6. 移動が行われる場合、つまり破壊的オーバーラップが存在しない場合について、以下で詳しく説明します。

アクセス・レジスター・モードでは、使用されるアクセス・レジスターの内容は、有効スペース指定と呼ばれます。有効スペース指定が等しくない場合は、破壊的オーバーラップは存在しないと宣言され、移動が行われます。有効スペース指定が同じである場合、またはアクセス・レジスター・モードではない場合は、次のケースが適用されます。

第 2 オペランドが記憶位置 $2^{24} - 1$ (あるいは、アドレッシング・モードに応じて、 $2^{31} - 1$ または $2^{64} - 1$) から記憶位置 0 に循環するかどうかによって、次のケースに示す移動が行われます。

- a. 第 2 オペランドが循環しない場合は、第 1 オペランドの左端バイトが第 2 オペランドの左端バイトと一致しているかまたはそれより左にあるとき、または第 1 オペランドの左端バイトが操作に関与する右端の第 2 オペランド・バイトより右にあるときに、移動が行われます。

- b. 第 2 オペランドが循環する場合は、第 1 オペランドの左端バイトが第 2 オペランドの左端バイトと一致しているかまたはそれより左にあるときで、かつ第 1 オペランドの左端バイトが操作に関与する右端の第 2 オペランド・バイトより右にあるときに、移動が行われます。

右端の第 2 オペランド・バイトは、第 1 オペランド長と第 2 オペランド長のうち、短い方の長さを使用して決定されます。

第 2 オペランド長が 1 または 0 の場合は、破壊的オーバーラップは存在しません。

7. MOVE LONG を EXECUTE のターゲットにする場合は、特別な注意が必要です。EXECUTE の項の『プログラミング上の注意』の、割り込み可能命令についての説明を参照してください。
8. MOVE LONG の実行は割り込み可能なので、実行中に割り込みが発生しないということを前提としたプログラムでは、この命令は使用できません。同様に、通常は、プログラムで、MOVE LONG の第 1 オペランドにこの命令の記憶位置または EXECUTE の記憶位置が含まれないようにする必要があります。なぜなら、割り込み後の処理の再開時にその記憶位置の新しい内容が解釈されたり、割り込みなしで命令が再取り出しされたりする可能性があるからです。
9. 割り込み可能命令については、さらに詳しいプログラミング上の注意が第 5 章、『プログラムの実行』の 5-20 ページの『割り込み可能命令』に示されています。
10. アクセス・レジスター・モードでは、アクセス・レジスター 0 は、アクセス・レジスター 0 の内容に関係なく、1 次アドレス・スペースを示します。

MOVE LONG EXTENDED

MVCLC R1,R3,D2(B2) [RS]

'A8'	R1	R3	B2	D2	
0	8	12	16	20	31

第 3 オペランド全体またはその一部が、第 1 オペランド位置に入れられます。第 1 オペランド位置の右端に幾つかのバイト位置が残る場合は、その位置は埋め込みバイトで満たされます。操作は、第 1 オペランド位置の終わりに達するか、CPU で決定される文字数が第 1 オペランド位置に移動されてしまうか、どちらかの条件が先

に発生するまで続けられます。結果は条件コードで示されます。

R1 フィールドおよび **R3** フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 3 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ **R1** および **R3** の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ **R1 + 1** および **R3 + 1** のビット位置 32-63 の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ **R1 + 1** および **R3 + 1** 全体の内容で指定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ **R1** および **R3** のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ **R1** および **R3** のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

第 2 オペランド・アドレスはデータをアドレッシングするためには使用されません。第 2 オペランド・アドレスの右端 8 ビット (ビット 56-63) は、埋め込みバイトです。第 2 オペランド・アドレスのビット 0-55 は無視されます。

7-97ページの図7-17 は、上記で説明したレジスタの内容およびアドレスを示しています。

移動が両方のフィールドの左端から開始され、1 バイトずつ処理されて右へ進むように見える方法で、結果が求められます。操作は、汎用レジスタ **R1 + 1** で指定された数のバイトが第 1 オペランド位置に入れられるか、または CPU で決定された数のバイトが第 1 オペランド位置に入れられるか、どちらかの条件が先に発生した時点で終了します。第 3 オペランドが第 1 オペランドよ

り短い場合は、第 1 オペランド位置の残りの右端バイトは、埋め込みバイトで満たされます。

第 1 オペランドの終わりに達したために操作が完了した場合、2 つのオペランドの長さが等しければ条件コードは 0 に設定され、第 1 オペランド長が第 3 オペランド長より小さければ条件コードは 1 に設定され、第 1 オペランド長が第 3 オペランド長より大きければ条件コードは 2 に設定されます。第 1 オペランドの終わりに達しないうちに CPU で決定されたバイト数が移動され、そのために操作が完了した場合には、条件コード 3 が設定されます。

破壊的オーバーラップに関してはテストは実行されず、したがって、破壊的オーバーラップが存在している場合には、第 1 オペランド位置に入る結果は予測不能です。第 1 オペランド位置にデータが移動された後で、その位置がソースとして使用されるときは、2 つのオペランドが破壊的にオーバーラップしていると言います。

第 1 オペランドの左端バイトが、操作に関与する第 3 オペランドの各バイトのうち、左端の 1 バイトを除くどのバイトとも一致していないときは、その 2 つのオペランドは破壊的オーバーラップをしていません。オペランドが記憶位置 $2^{24} - 1$ (または $2^{31} - 1$ または $2^{64} - 1$) から記憶位置 0 へ循環する場合は、記憶位置 $2^{24} - 1$ (または $2^{31} - 1$ または $2^{64} - 1$) まで (この記憶位置を含む) の記憶位置のオペランド・バイトは、記憶位置 0 以降のバイトの左にあるものと見なされません。

24 ビット・アドレッシング・モードでは、記憶位置 $2^{24} - 1$ から記憶位置 0 へと循環し、31 ビット・アドレッシング・モードでは、記憶位置 $2^{31} - 1$ から記憶位置 0 へと循環し、64 ビット・アドレッシング・モードでは、記憶位置 $2^{64} - 1$ から記憶位置 0 へと循環します。

汎用レジスタ **R1 + 1** で指定された長さが 0 であれば、移動は行われず、長さの相対値を示す条件コード 0 または 1 が設定されます。

モデルにより場合によっては、操作の非埋め込み部分の実行中に、16 進数 B0 および 16 進数 B8 の 2 つの埋め込みバイト値を使用して、それぞれ、キャッシュをバイパスして移動を行うかキャッシュを使用して移動を行うかを指示する場合があります。この場合、埋め込みバイトの 16 進数 B0 は移動後に宛先領域を参照しないことを指示し、埋め込みバイトの 16 進数 B8 は、宛先領域を参照することを指示します。

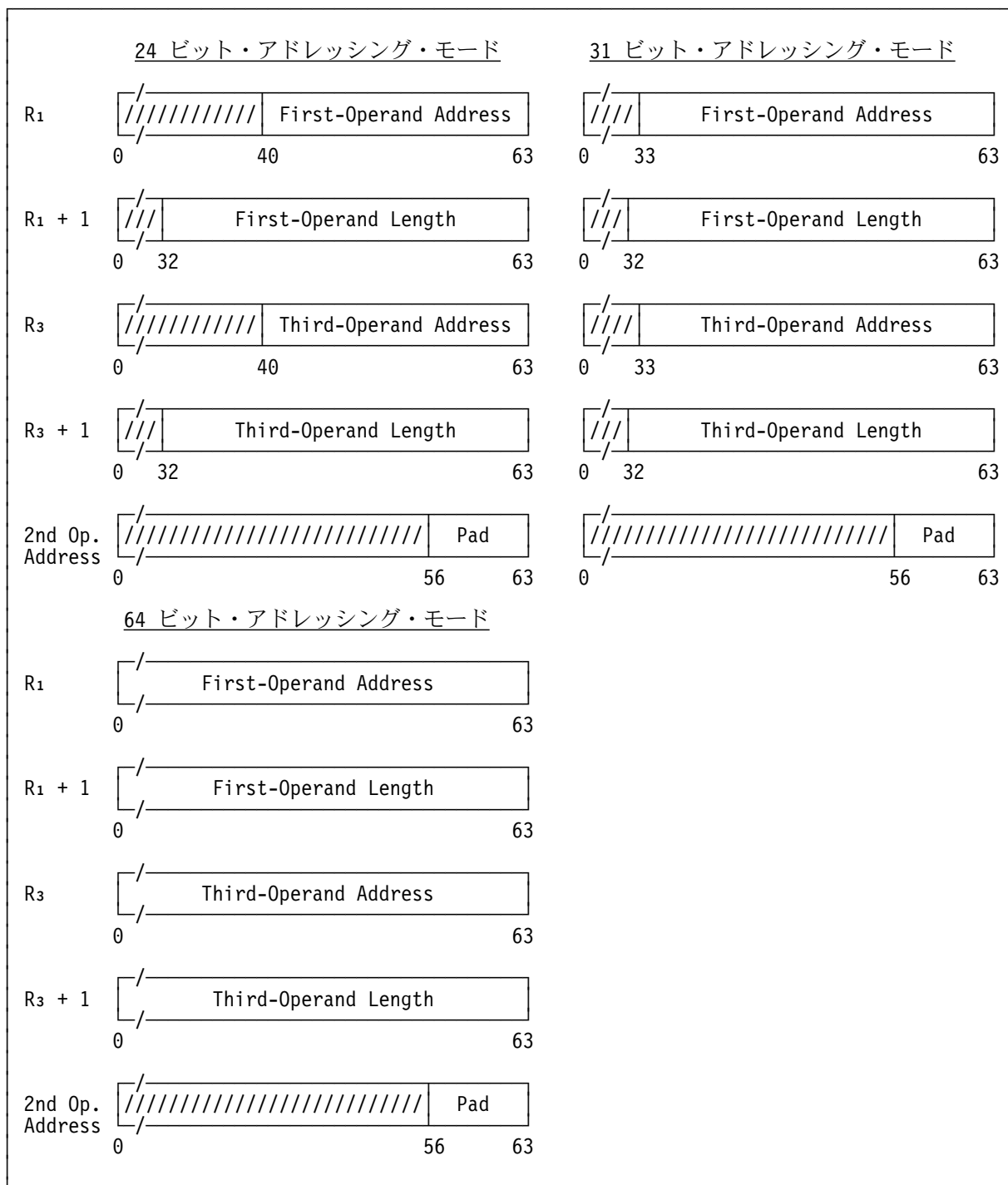


図 7-17. MOVE LONG EXTENDED で使用するレジスタの内容および第 2 オペランド・アドレス

操作の非埋め込み部分では、MOVE LONG のオペランドへのアクセスは単一アクセス参照になります。埋め込みバイトが 16 進数 B1 の場合を除き、このようなアクセスは、チャンネル・プログラムおよび他の CPU から見て必ずしも左から右への方向で行われるようには見えません。両オペランドがダブルワード境界から始まっていて、長さがダブルワードの整数倍で、オーバーラップし

ていない場合は、操作の非埋め込み部分の実行中は、他の CPU からこれらのオペランドがダブルワード単位でアクセスされているように見えます。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランドの中の埋め込みバイトで満たされる部分は、格納が左から右への方向に行われるとは限りませ

ん。また、2 回以上格納されるように見えることもあります。

操作の完了時には、汎用レジスター $R_1 + 1$ 内の長さは、第 1 オペランド位置に格納されているバイトの数だけ減少し、汎用レジスター R_1 内のアドレスは、同じバイト数だけ増加しています。汎用レジスター $R_3 + 1$ 内の長さは、第 3 オペランド位置から取り出して移動されたバイトの数だけ減少し、汎用レジスター R_3 内のアドレスは同じバイト数だけ増加しています。

第 1 オペランドの終わりに達しないうちに CPU で決定されたバイト数の移動が完了したために操作が完了した場合は、汎用レジスター $R_1 + 1$ および $R_3 + 1$ 内の長さが、移動済みのバイト数だけ減少し、汎用レジスター R_1 および R_3 内のアドレスが同じバイト数だけ増加しています。したがって、この命令の再実行時には、次に移動するバイトから処理が再開されます。埋め込みの実行中に操作が完了した場合は、汎用レジスター $R_3 + 1$ の長さフィールドは 0 になり、汎用レジスター R_3 のアドレスは、汎用レジスター $R_3 + 1$ の元の長さ分だけ増加します。汎用レジスター R_1 および $R_1 + 1$ には、埋め込み操作がどこまで進んだかが反映されています。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスター R_1 、 $R_1 + 1$ 、 R_3 、および $R_3 + 1$ のビット位置 0-31 の内容は、常に無変更のままになります。

埋め込みバイトは、命令の実行中に $D_2(B_2)$ から複数回形成される可能性があり、 R_1 および R_3 により指定されるレジスターは複数回更新される可能性があります。したがって、 B_2 が R_1 、 $R_1 + 1$ 、 R_3 、または $R_3 + 1$ に等しく、命令の実行中に変化する可能性がある場合は、結果は予測不能です。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。最大量は、どちらのオペランドでも約 4K バイトです。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスター R_1 および R_3 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、片方または両方の元の長さ値が 0 の場合でも、0 に設定されるか、または元の値のまま変更されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。同様に、第 3 オペランドが第 1 オペランドより長い場合も、第 1 オペランド・フィールドより長い第 3 オペランド・フィールド部分については、アクセス例外は認識されません。4K バイトより長いオペランドの場合は、処理中の現在の記憶位置より後の 4K バイトを超える記憶位置については、アクセス例外は認識されません。オペランドに関連した R フィールドが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。また、 R_1 フィールドが奇数の場合は、PER 記憶域変更イベントは認識されず、変更ビットも設定されません。

結果の条件コード:

- 0 全バイトを移動。両方のオペランドの長さが等しい。
- 1 全バイトを移動。第 1 オペランド長が小さい。
- 2 全バイトを移動。第 1 オペランド長が大きい。
- 3 CPU で決定されたバイト数を移動。第 1 オペランドの終わりに達していません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 3。格納: オペランド 1)
- 指定例外

プログラミング上の注意:

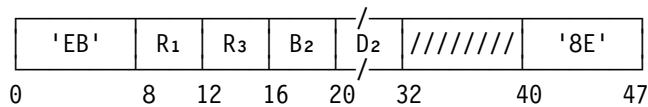
1. MOVE LONG EXTENDED は、オペランド長が 32 ビットまたは 64 ビットの 2 進整数として指定され、破壊的オーバーラップのテストが必要ない場合に、MOVE LONG の代わりに使用するためのものです。MOVE LONG なら割り込みが発生するケースで、MOVE LONG EXTENDED は条件コード 3 を設定します。
2. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、移動処理を継続できます。移動済みのバイト数をプログラムで判別する必要はありません。
3. どちらのオペランドも約 4K バイト以上は処理しないという機能は、長時間の操作の実行中に別の CPU に関してプログラムが設定した可能性のあるフラグを、ソフトウェアでポーリングできるようにするためです。
4. 埋め込みバイトを 0 に設定し、第 3 オペランドの長さを 0 に設定することによって、MOVE LONG EXTENDED を使用して記憶域を消去することができます。しかし、この消去に関連した格納は複数アクセス格納になることがあります。したがって、チ

チャンネル・プログラムまたは他の CPU が記憶域を 0 と見なすすぐにアクセスして使用する可能性がある場合には、そのような記憶域をこの方法で消去してはなりません。詳しくは、5-86ページの『記憶域オペランドの一貫性』を参照してください。

5. R1 フィールドと R3 フィールドの内容が同じ場合は、指定されているレジスタの内容が、移動されたバイト数の 2 倍でなく、移動されたバイト数と同数だけ増加または減少します。条件コードは、多くの場合は 3 に設定された後で、最終的に 0 に設定されます。
6. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。

MOVE LONG UNICODE

MVCLU R1,R3,D2(B2) [RSE]



第 3 オペランド全体またはその一部が、第 1 オペランド位置に入れられます。第 1 オペランド位置の右端に 2 バイト文字の文字位置が幾つか残る場合は、その位置は 2 バイトの埋め込み文字で満たされます。操作は、第 1 オペランド位置の終わりに達するか、CPU で決定される文字数が第 1 オペランド位置に入れられるか、どちらかの条件が先に発生するまで続けられます。結果は条件コードで示されます。

R1 フィールドおよび R3 フィールドは、それぞれ汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 3 オペランドの左端文字の位置は、それぞれ、汎用レジスタ R1 および R3 の内容によって指定されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ R1 + 1 および R3 + 1 のビット位置 32-63 の内容で指定され、これらの内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置および第 3 オペランド位置のバイト数は、それぞれ、汎用レジスタ R1 + 1 および R3 + 1 のビット位置 0-63 の内容で指定され、これらの内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ R1 + 1 および R3 + 1 の内容は、偶数個のバイトを示している必要があります。さもないと、指定例外が認識されます。

汎用レジスタ R1 および R3 のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R1 および R3 のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、レジスタのビット位置 0-63 の内容がアドレスを形成します。

第 2 オペランド・アドレスはデータのアドレッシングには使用されません。第 2 オペランド・アドレスの右端 16 ビット (ビット 48-63) は、2 バイト埋め込み文字です。第 2 オペランド・アドレスのビット 0-47 は無視されます。

7-100ページの図7-18 は、上記で説明したレジスタの内容およびアドレスを示しています。

移動が両方のフィールドの左端から開始され、1 文字ずつ処理されて右へ進むように見える方法で、結果が求められます。操作は、汎用レジスタ R1 + 1 の内容で指定された数の文字が第 1 オペランド位置に入れられるか、または CPU で決定された数の文字が入れられるか、どちらかの条件が先に発生したときに終了します。第 3 オペランドが第 1 オペランドより短い場合は、第 1 オペランド位置の残りの右端の文字位置は、2 バイトの埋め込み文字で満たされます。

第 1 オペランドの終わりに達したために操作が完了した場合、2 つのオペランドの長さが等しければ条件コードは 0 に設定され、第 1 オペランド長が第 3 オペランド長より小さければ条件コードは 1 に設定され、第 1 オペランド長が第 3 オペランド長より大きければ条件コードは 2 に設定されます。第 1 オペランドの終わりに達しないうちに CPU で決定された文字数が移動され、そのために操作が完了した場合には、条件コード 3 が設定されます。

破壊的オーバーラップに関してはテストは実行されず、したがって、破壊的オーバーラップが存在している場合には、第 1 オペランド位置に入る結果は予測不能です。第 1 オペランド位置にデータが移動された後で、その位

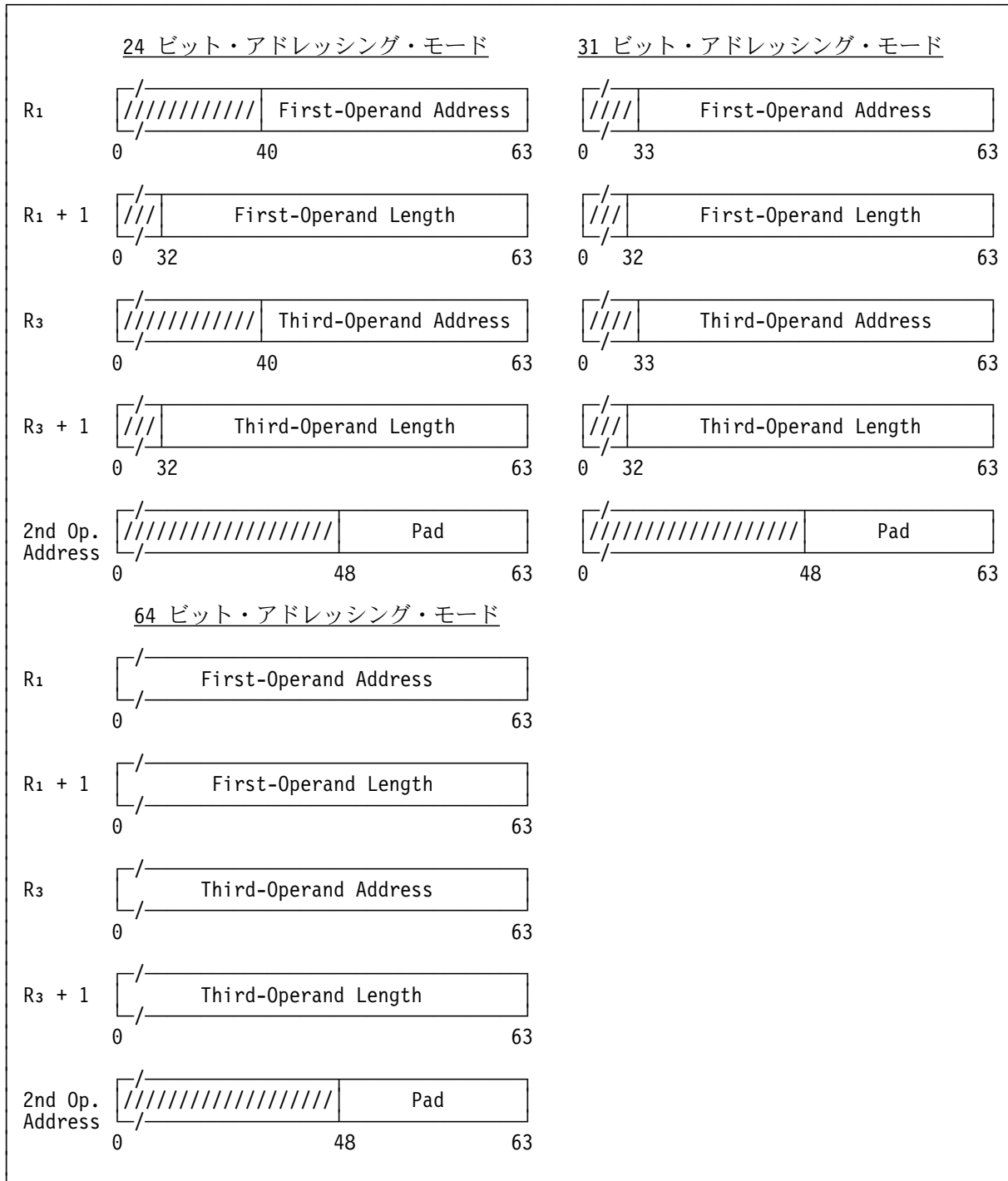


図 7-18. MOVE LONG UNICODE で使用するレジスターの内容および第 2 オペランド・アドレス

置がソースとして使用されるときは、2 つのオペランドが破壊的にオーバーラップしていると言います。

第 1 オペランドの左端文字が、操作に関与する第 3 オペランドの各文字のうち、左端の 1 文字を除くどの文字とも一致していないときは、その 2 つのオペランドは破

壊的オーバーラップをしていません。オペランドが記憶位置 $2^{24} - 1$ (または $2^{31} - 1$ または $2^{64} - 1$) から記憶位置 0 へ循環する場合は、記憶位置 $2^{24} - 1$ (または $2^{31} - 1$ または $2^{64} - 1$) まで (この位置を含む) の記憶位置のオペランド文字は、記憶位置 0 以降の文字の左にあるものと見なされます。

24 ビット・アドレッシング・モードでは、記憶位置 $2^{24} - 1$ から記憶位置 0 へと循環し、31 ビット・アドレッシング・モードでは、記憶位置 $2^{31} - 1$ から記憶位置 0 へと循環し、64 ビット・アドレッシング・モードでは、記憶位置 $2^{64} - 1$ から記憶位置 0 へと循環します。

汎用レジスタ $R_1 + 1$ で指定された長さが 0 であれば、移動は行われず、長さの相対値を示す条件コード 0 または 1 が設定されます。

操作の非埋め込み部分では、MOVE LONG UNICODE のオペランドへのアクセスは単一アクセス参照になります。チャンネル・プログラムまたは他の CPU から見た場合、このようなアクセスは必ずしも左から右への方向で行われるようには見えません。両オペランドがダブルワード境界から始まっていて、長さがダブルワードの整数倍で、オーバーラップしていない場合は、操作の非埋め込み部分の実行中は、他の CPU からはこれらのオペランドがダブルワード単位でアクセスされているように見えます。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランドの中の 2 バイトの埋め込み文字で満たされる部分は、格納が左から右への方向に行われるとは限りません。また、二度以上格納されるように見えることもあります。

操作の完了時には、汎用レジスタ $R_1 + 1$ 内の長さは、第 1 オペランド位置に格納されている文字数の 2 倍だけ減少し、汎用レジスタ R_1 内のアドレスは、同じ文字数だけ増加しています。汎用レジスタ $R_3 + 1$ 内の長さは、第 3 オペランド位置から取り出して移動された文字数の 2 倍だけ減少し、汎用レジスタ R_3 内のアドレスは、同じく移動された文字数の 2 倍だけ増加しています。

第 1 オペランドの終わりに達しないうちに CPU で決定されたバイト数の移動が完了したために操作が完了した場合は、汎用レジスタ $R_1 + 1$ および $R_3 + 1$ 内の長さが、移動済みの文字数の 2 倍だけ減少し、汎用レジスタ R_1 および R_3 内のアドレスが同じ文字数だけ増加しています。したがって、この命令の再実行時には、次に移動する文字から処理が再開されます。埋め込みの実行中に操作が完了した場合は、汎用レジスタ $R_3 + 1$ の長さフィールドは 0 になり、汎用レジスタ R_3 内のアドレスは、オペランド 3 から移動された文字数の 2 倍だけ増加しています。汎用レジスタ R_1 および $R_1 + 1$ には、埋め込み操作がどこまで進んだかが反映されています。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R_1 、 $R_1 + 1$ 、 R_2 、および $R_2 + 1$ のビット位置 0-31 の内容は、常に無変更のままになります。

2 バイト埋め込み文字は、命令の実行中に $D_2(B_2)$ から複数回形成される可能性があり、 R_1 および R_3 により指定されるレジスタは複数回更新される可能性があります。したがって、 B_2 が R_1 、 $R_1 + 1$ 、 R_3 、または $R_3 + 1$ に等しく、命令の実行中に変化する可能性がある場合は、結果は予測不能です。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

24 ビットまたは 31 ビットのアドレッシング・モードの操作の完了時には、汎用レジスタ R_1 および R_3 のビット位置 32-63 内の、アドレスの一部ではない左端ビットは、片方または両方の長さの元の値が 0 の場合でも、0 に設定されるか、または元の値のまま変更されません。

オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。同様に、第 3 オペランドが第 1 オペランドより長い場合も、第 1 オペランド・フィールドより長い第 3 オペランド・フィールド部分については、アクセス例外は認識されません。4K バイトより長いオペランドの場合は、処理中の現在の記憶位置より後の 4K バイトを超える記憶位置については、アクセス例外は認識されません。オペランドに関連した R フィールドまたは長さが奇数の場合は、そのオペランドについてはアクセス例外は認識されません。また、 R_1 フィールドまたは長さが奇数の場合は、PER 記憶域変更イベントは認識されず、変更ビットも設定されません。

結果の条件コード:

- 0 全文字を移動。両方のオペランドの長さが等しい。
- 1 全文字を移動。第 1 オペランド長が小さい。
- 2 全文字を移動。第 1 オペランド長が大きい。
- 3 CPU で決定された文字数を移動。第 1 オペランドの終わりには達していません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 3。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)

- 指定例外

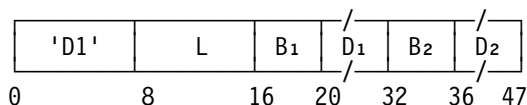
プログラミング上の注意:

1. MOVE LONG UNICODE は、埋め込み文字が 2 バイト文字の場合に、MOVE LONG または MOVE LONG EXTENDED の代わりに使用するためのものです。埋め込み文字は、Unicode 文字でも、その他の任意の 2 バイト文字でも構いません。MOVE LONG なら割り込みが発生するケースで、MOVE LONG UNICODE は条件コード 3 を設定します。
2. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、移動処理を継続できます。移動済みの文字数をプログラムで判別する必要はありません。
3. MOVE LONG UNICODE を使用して、第 2 オペランド・アドレスに埋め込み文字を入れ、第 3 オペランド長を 0 に設定することによって、記憶域を埋め込み文字で満たすことができます。しかし、この消去に関連した格納は複数アクセス格納になることがあります。したがって、記憶域が 0 と見なされると即時にチャネル・プログラムまたは他の CPU がアクセスして使用する可能性がある場合、そのような記憶域をこの方法で消去してはなりません。詳しくは、5-86ページの『記憶域オペランドの一貫性』を参照してください。
4. R₁ フィールドと R₃ フィールドの内容が同じ場合は、指定されているレジスタの内容が、移動された文字数の 2 倍だけ増加または減少しています (移動された文字数の 4 倍ではありません)。条件コードは、多くの場合は 3 に設定された後で、最終的に 0 に設定されます。
5. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。
6. Unicode のスペース文字 (または表記が 16 進数 FFF 以下の任意の文字) による埋め込みが必要な場合は、命令の変位フィールドにこの文字を次の例のように表記することができます。

```
MVCLU 6,8,X'020'
```

MOVE NUMERICS

MVN D₁(L,B₁),D₂(B₂) [SS]



第 2 オペランド内の各バイトの右端 4 ビットが、第 1 オペランド内の対応するバイトの右端ビット位置に入れます。第 1 オペランド内の各バイトの左端 4 ビットは変更されません。

各オペランドは左から右へ処理されます。オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理され、必要なオペランド・バイトを取り出した直後に各結果バイトが格納されるときと同様の方法で、結果が取得されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)

プログラミング上の注意:

1. MOVE NUMERICS 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. MOVE NUMERICS は、ゾーン・フォーマットの 10 進数データ・フィールドの数値部分を移動します。ゾーン 10 進数 フォーマットについては、第 8 章、『10 進数命令』の説明を参照してください。どちらのオペランドについても、符号コードおよび数字コードの妥当性検査は行われません。
3. MOVE NUMERICS の第 1 オペランドへのアクセスは、第 1 オペランドの各バイトの右端 4 ビットを取り出す操作と、そのバイトの更新済みの値を格納する後続操作から成っています。しかし、ある 1 つのバイトについて見ると、必ずしも取り出しアクセスの直後に格納アクセスが行われるとは限りません。このため、他の CPU またはチャネル・プログラムが更新する可能性のある記憶位置をこの命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』の『マルチプログラミングとマルチプロセッシングの例』で OR (OI) について示してあります。

MOVE STRING

MVST R1,R2 [RRE]

'B255'	////////	R1	R2
0	16	24	28 31

第 2 オペランドの全体またはその一部が、第 1 オペランド位置に入れられます。操作は、第 2 オペランドの終わりに達するか、CPU で決定されるバイト数の移動が完了するか、どちらかの条件が先に発生するまで続けます。CPU で決定される数の最小値は 1 です。結果は条件コードで示されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R1 および R2 の内容によって指定されます。

汎用レジスタ R1 および R2 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R1 および R2 のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

第 2 オペランドの終わりは、そのオペランドの最後のバイト位置にある終了文字で指示します。第 2 オペランドの終わりを判別するために使用される終了文字は汎用レジスタ 0 の 56-63 で指定します。汎用レジスタ 0 のビット位置 32-55 は、将来の拡張に備えて予約されており、すべて 0 が含まれていなければなりません。さもないと、指定例外が認識されます。

操作は左から右へと進み、第 2 オペランドの終了文字が移動されるか、CPU で決定される数の第 2 オペランド・バイトの移動が完了するか、どちらかの条件が先に発生した時点で終了します。CPU で決定される数の最小値は 1 です。終了文字が第 2 オペランドの先頭バイト位置にある場合は、終了文字のみが移動されます。終了文字が移動された場合は、条件コード 1 が設定されず。CPU で決定される数の第 2 オペランド・バイト (終了文字はこの数に含まれません) が移動された場合

は、条件コード 3 が設定されます。破壊的オーバーラップは認識されません。第 2 オペランドが宛先として使用された後でソースとして使用された場合、第 2 オペランド内の終了文字が認識されない場合があるという点で、結果は予測不能です。

条件コード 1 が設定された場合は、第 1 オペランド内の終了文字が汎用レジスタ R1 に入れられ、汎用レジスタ R2 の内容は変更されません。条件コード 3 が設定された場合は、第 1 オペランドおよび第 2 オペランド内で次に処理されるバイトのアドレスが、それぞれ汎用レジスタ R1 および R2 に入れられます。汎用レジスタにアドレスが入られるたびに、そのレジスタのビット 32-39 (24 ビット・アドレッシング・モードの場合) またはビット 32 (31 ビット・アドレッシング・モードの場合) が 0 に設定されます。R1 および R2 レジスタのビット 0-31 は、24 ビット・モードの場合も 31 ビット・モードの場合も変更されません。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

第 1 オペランドおよび第 2 オペランドに関するアクセス例外が認識されるのは、実行中の操作で必ず使用されるオペランド部分についてのみです。

破壊的オーバーラップが認識されないという点を除けば、記憶域オペランド一貫性の規則は、MOVE (MVC) 命令の場合と同じです。

結果の条件コード:

- 0 --
- 1 第 2 オペランド全体を移動。汎用レジスタ R1 が更新され、終了文字のアドレスが第 1 オペランドに入っています。汎用レジスタ R2 は変更されていません。
- 2 --
- 3 CPU で決定されたバイト数を移動。汎用レジスタ R1 および R2 が更新され、次のバイトのアドレスが入っています。

プログラム例外:

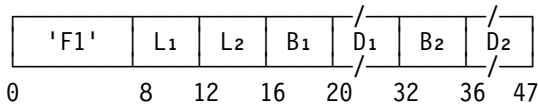
- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 指定例外

プログラミング上の注意:

1. MOVE STRING 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、データ移動を継続できません。移動済みのバイト数をプログラムで判別する必要はありません。
3. R₁ または R₂ が 0 であっても構いません。その場合、汎用レジスター 0 は、アドレスと共に終了文字も含んでいるものとして扱われます。
4. アクセス・レジスター・モードでは、アクセス・レジスター 0 は、アクセス・レジスター 0 の内容に関係なく、1 次アドレス・スペースを示します。

MOVE WITH OFFSET

MVO D₁(L₁,B₁),D₂(L₂,B₂) [SS]



第 2 オペランドが、第 1 オペランドの右端 4 ビットの左側に隣接する位置に入られます。

第 1 オペランドの右端 4 ビットは、右端ビットとして第 2 オペランドに連結され、第 2 オペランドのビットは 4 ビット位置だけオフセットされ、結果が第 1 オペランド位置に入られます。

結果は、両方のオペランドを右から左へ処理したものとして求められます。必要があれば、第 2 オペランドは左側が 0 で拡張されたものとして扱われます。第 1 オペランドが短すぎて第 2 オペランド全体を収容できない場合は、第 2 オペランドの左端の残り部分は無視されます。第 2 オペランドの使用されなかった部分については、アクセス例外が示されることもあり、示されないこともあります。

オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理され、必要なオペランド・バイトを取り出した直後に各結果バイトが格納され、個々の第 2 オペランド・バイトの左の数字が、次の結果バイトの処理に使用可能なままの状態になっていて再取り出しが必要ないものとして、結果が取得されません。

条件コード: コードは変更されません。

プログラム例外:

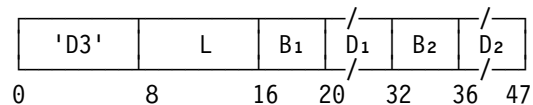
- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)

プログラミング上の注意:

1. MOVE WITH OFFSET 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. MOVE WITH OFFSET を使用して、パック 10 進データを奇数個の桁位置だけシフトすることができます。パック 10 進数フォーマットについては、第 8 章、『10 進数命令』の説明を参照してください。どちらのオペランドについても、符号コードおよび数字コードの妥当性検査は行われません。しかし、多くの場合、SHIFT AND ROUND DECIMAL の方が使用には便利です。
3. MOVE WITH OFFSET の第 1 オペランドの右端バイトへのアクセスは、第 1 オペランドの各バイトの右端 4 ビットを取り出す操作と、このバイトの更新済みの値を格納する操作から成っています。第 1 オペランドの右端バイトへの取り出しアクセスおよび格納アクセスは、必ずしも取り出しアクセスの直後に格納アクセスが行われるわけではありません。このため、他の CPU またはチャンネル・プログラムが更新する可能性のある記憶位置をこの命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』の『マルチプログラミングとマルチプロセッシングの例』で OR (OI) について示してあります。
4. MOVE WITH OFFSET の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

MOVE ZONES

MVZ D₁(L,B₁),D₂(B₂) [SS]



第 2 オペランド内の各バイトの左端 4 ビットが、第 1 オペランド内の対応するバイトの左端 4 ビット位置に入られます。第 1 オペランドの各バイトの右端 4 ビットは変更されません。

各オペランドは左から右へ処理されます。オペランドがオーバーラップしているときには、オペランドが一度に1バイトずつ処理されて、必要なオペランド・バイトを取り出した直後に各結果バイトが格納される場合と同様の方法で、結果が取得されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)

プログラミング上の注意:

1. MOVE ZONES 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. MOVE ZONES は、ゾーン・フォーマットの10進数フィールドのゾーン部分を移動します。ゾーン・フォーマットについては、第8章、『10進数命令』の説明を参照してください。どちらのオペランドについても、符号コードおよび数字コードの妥当性検査は行われません。
3. MOVE ZONES の第1オペランドへのアクセスは、第1オペランドの各バイトの左端4ビットを取り出す操作と、そのバイトの更新済みの値を格納する後続操作から成っています。しかし、ある1つのバイトについて見ると、必ずしも取り出しアクセスの直後に格納アクセスが行われるとは限りません。このため、他のCPUまたはチャンネル・プログラムが更新する可能性のある記憶位置をこの命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』の『マルチプログラミングとマルチプロセッシングの例』でOR(OI)命令について示してあります。

MULTIPLY

MR R₁,R₂ [RR]

'1C'	R ₁	R ₂
0	8	15

M R₁,D₂(X₂,B₂) [RX]

'5C'	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20
				31

32ビットの第1オペランド(被乗数)に32ビットの第2オペランド(乗数)が掛けられ、64ビットの積が第1オペランド位置に入れます。

R₁ フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定する必要があります。さもないと、指定例外が認識されます。

被乗数および乗数は、どちらも32ビットの符号付き2進整数として扱われます。被乗数は、汎用レジスタ R₁ + 1 のビット位置 32-63 にあります。MULTIPLY (MR) では、乗数は汎用レジスタ R₂ のビット位置 32-63 にあります。汎用レジスタ R₁ の内容、および汎用レジスタ R₁ + 1 (および MR では汎用レジスタ R₂) のビット位置 0-31 の内容は無視されます。

積は、64ビットの符号付き2進整数です。積のビット 0-31 は、汎用レジスタ R₁ のビット 32-63 と置き換わります。積のビット 32-63 は、汎用レジスタ R₁ + 1 のビット 32-63 と置き換わります。汎用レジスタ R₁ および R₁ + 1 のビット 0-31 は変更されません。オーバーフローは起きません。

積の符号は、乗数および被乗数の符号に基づき、代数規則に従って決定されます。ただし、結果が0の場合は、符号は常に正です。

条件コード: コードは変更されません。

プログラム例外:

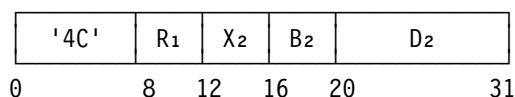
- アクセス例外 (取り出し: M のオペランド 2 のみ)
- 指定例外

プログラミング上の注意:

1. MULTIPLY 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 積の有効数字部分は、常に62個以下のビット位置を占めます。32ビットの最大負数を2個掛け合わせた場合のみ、63個の有効数字ビットの積が形成されます。

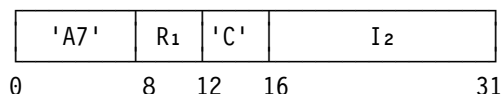
MULTIPLY HALFWORD

MH R₁,D₂(X₂,B₂) [RX]

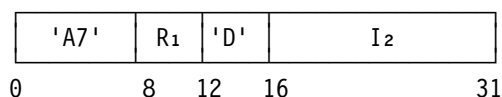


MULTIPLY HALFWORD IMMEDIATE

MHI R₁,I₂ [RI]



MGHI R₁,I₂ [RI]



32 ビットまたは 64 ビットの第 1 オペランド (被乗数) に 16 ビットの第 2 オペランド (乗数) が掛けられ、積の右端 32 ビットまたは 64 ビットが第 1 オペランド位置に入れます。第 2 オペランドは、長さが 2 バイトで、16 ビットの符号付き 2 進整数として扱われます。

MULTIPLY HALFWORD および MULTIPLY HALFWORD IMMEDIATE (MHI) では、被乗数は、汎用レジスタ R₁ のビット位置 32-63 に入っている 32 ビットの符号付き 2 進整数として扱われ、符号付き 2 進整数である積の右端 32 ビットで置き換えられます。積の 32 個の右端ビットの左側にあるビットが有効数字であるかどうかはテストされず、オーバーフロー指示も提供されません。汎用レジスタ R₁ のビット 0-31 は、無視され、無変更のままになります。

MULTIPLY HALFWORD IMMEDIATE (MGHI) では、被乗数は、汎用レジスタ R₁ のビット位置 0-63 に入っている 64 ビットの符号付き 2 進整数として扱われ、符号付き 2 進整数である積の右端 64 ビットで置き換えられます。積の 64 個の右端ビットの左側にあるビットが有効数字であるかどうかはテストされず、オーバーフロー指示も提供されません。

積の符号は、乗数および被乗数の符号に基づき、代数規則に従って決定されます。ただし、結果が 0 の場合は、符号は常に正です。

条件コード: コードは変更されません。

プログラム例外:

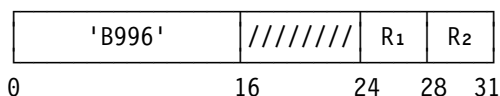
- アクセス例外 (取り出し: MH のオペランド 2 のみ)

プログラミング上の注意:

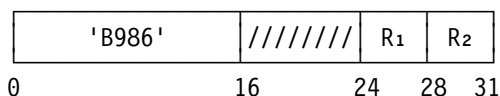
1. MULTIPLY HALFWORD 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. MH および MHI では、積の有効数字部分は常に 46 個以下のビット位置を占めます。2 個の最大負数を掛け合わせた場合のみ、47 個の有効数字ビットの積が形成されます。積の右端 32 ビットは無変更のまま第 1 オペランド位置に入れられ、その左にあるビットはすべて無視されるので、オーバーフローが生じた場合は、結果の符号ビットは積の真の符号とは異なる場合があります。負数の積の場合は、レジスタ R₁ に入れられる 32 ビットは、2 の補数で表された積の右端部分です。MGHI では、積の有効数字部分は常に 78 個以下のビット位置を占めます。

MULTIPLY LOGICAL

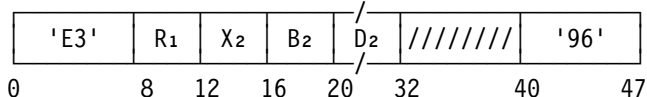
MLR R₁,R₂ [RRE]



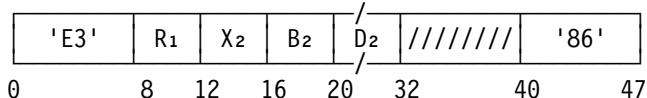
MLGR R₁,R₂ [RRE]



ML R₁,D₂(X₂,B₂) [RXE]



MLG R₁,D₂(X₂,B₂) [RXE]



32 ビットまたは 64 ビットの第 1 オペランド (被乗数) に 32 ビットまたは 64 ビットの第 2 オペランド (乗数) が掛けられ、64 ビットまたは 128 ビットの積が第 1 オペランド位置に入れられます。

R1 フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

MULTIPLY LOGICAL (MLR, ML) では、被乗数と乗数はどちらも 32 ビットの符号なし 2 進整数として扱われます。被乗数は、汎用レジスタ R1 + 1 のビット位置 32-63 に入れられます。MULTIPLY LOGICAL (MLR) では、乗数は汎用レジスタ R2 のビット位置 32-63 に入れられます。汎用レジスタ R1 の内容、および汎用レジスタ R1 + 1 (および MLR では汎用レジスタ R2) のビット位置 0-31 の内容は無視されます。積は 64 ビットの符号なし 2 進整数です。積のビット 0-31 は、汎用レジスタ R1 のビット 32-63 と置き換わり、積のビット 32-63 は、汎用レジスタ R1 + 1 のビット 32-63 と置き換わります。汎用レジスタ R1 および R1 + 1 のビット 0-31 は変更されません。オーバーフローは起きません。

MULTIPLY LOGICAL (MLGR, MLG) では、被乗数と乗数はどちらも 64 ビットの符号なし 2 進整数として扱われます。被乗数は汎用レジスタ R1 + 1 に入れられます。汎用レジスタ R1 の内容は無視されます。積は 128 ビットの符号なし 2 進整数です。積のビット 0-63 は、汎用レジスタ R1 の内容と置き換わり、積のビット 64-127 は汎用レジスタ R1 + 1 の内容と置き換わります。オーバーフローは起きません。

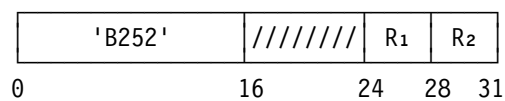
条件コード: コードは変更されません。

プログラム例外:

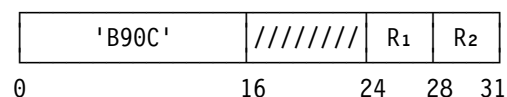
- アクセス例外 (取り出し: ML および MLG のオペランド 2 のみ)
- 指定例外

MULTIPLY SINGLE

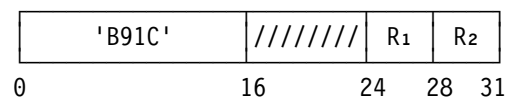
MSR R1,R2 [RRE]



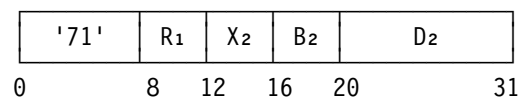
MSGR R1,R2 [RRE]



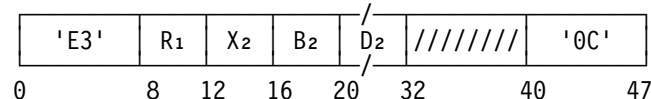
MSGFR R1,R2 [RRE]



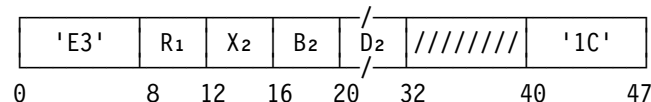
MS R1,D2(X2,B2) [RX]



MSG R1,D2(X2,B2) [RXE]



MSGF R1,D2(X2,B2) [RXE]



第 1 オペランド (被乗数) に第 2 オペランド (乗数) が掛けられ、積の右端 32 ビットまたは 64 ビットが第 1 オペランド位置に入れられます。

MULTIPLY SINGLE (MSR, MS) では、被乗数、乗数、および積は、32 ビットの符号付き 2 進整数として扱われます。被乗数は、汎用レジスタ R1 のビット位置 32-63 からとられ、符号付き 2 進整数である積の右端 32 ビットで置き換えられます。汎用レジスタ R1 のビット 0-31 は変更されません。MSR では、乗数は汎用レジスタ R2 のビット位置 32-63 にあります。積の 32 個の右端ビットの左側にあるビットが有効数字であるかどうかはテストされず、オーバーフロー指示も提供されません。

MULTIPLY SINGLE (MSGR, MSGFR, MSG, MSGF) では、被乗数、乗数、および積は、64 ビットの符号付き 2 進整数として扱われます。ただし、MSGFR および MSGF の場合のみ、乗数は 32 ビットの符号付き 2 進整数として扱われます。被乗数は、汎用レジスタ R1 からとられ、符号付き 2 進整数である積の右端 64 ビットで置き換えられます。MSGFR では、乗数は

汎用レジスタ R₂ のビット位置 32-63 にあります。積の 64 個の右端ビットの左側にあるビットが有効数字であるかどうかはテストされず、オーバーフロー指示も提供されません。

積の符号は、乗数および被乗数の符号に基づき、代数規則に従って決定されます。ただし、結果が 0 の場合は、符号は常に正です。

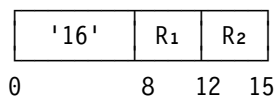
条件コード: コードは変更されません。

プログラム例外:

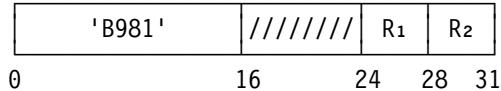
- アクセス例外 (取り出し: MS、MSG、MSGF のオペランド 2 のみ)

OR

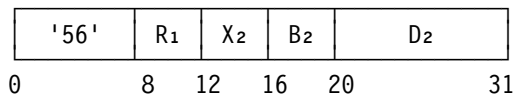
OR R₁,R₂ [RR]



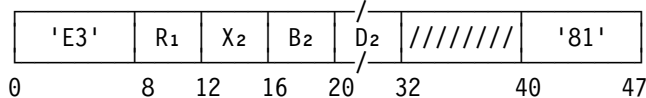
OGR R₁,R₂ [RRE]



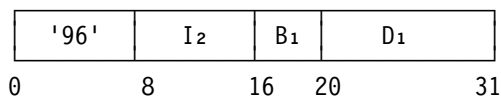
O R₁,D₂(X₂,B₂) [RX]



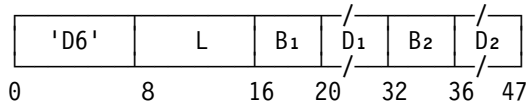
OG R₁,D₂(X₂,B₂) [RXE]



OI D₁(B₁),I₂ [SI]



OC D₁(L,B₁),D₂(B₂) [SS]



第 1 オペランドと第 2 オペランドの論理和 (OR) が、第 1 オペランド位置に入れられます。

連結子 OR は、両方のオペランドにビット単位で適用されます。片方または両方のオペランドの対応するビット位置に 1 が含まれている場合は結果のビット位置の内容が 1 に設定され、それ以外の場合は結果ビットが 0 に設定されます。

OR (OC) では、各オペランドが左から右へと処理されます。オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理され、必要なオペランド・バイトを取り出した直後に各結果バイトが格納される時と同様の方法で、結果が取得されます。

OR (OI) では、第 1 オペランドの長さは 1 バイトであり、1 バイトのみが格納されます。

OR (OR、O) ではオペランドは 32 ビットで、OR (OGR、OG) ではオペランドは 64 ビットです。

結果の条件コード:

- 0 結果は 0 です。
- 1 結果は 0 ではありません。
- 2 --
- 3 --

プログラム例外:

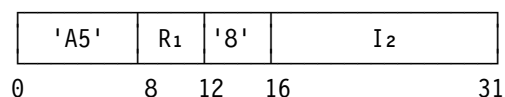
- アクセス例外 (取り出し: オペランド 2、O、OG、および OC。取り出しおよび格納: オペランド 1、OI および OC)

プログラミング上の注意:

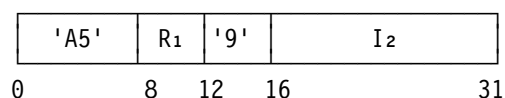
1. OR 命令の使用例は、付録A、『数の表現と命令の使用例』に示されます。
2. OR は、ビットを 1 に設定するために使用できません。
3. OR (OI) および OR (OC) の第 1 オペランドへのアクセスは、第 1 オペランドの 1 バイトを記憶域から取り出す操作と、更新済みの値を格納する後続操作から成っています。しかし、ある 1 つのバイトについて見ると、必ずしも取り出しアクセスの直後に格納アクセスが行われるとは限りません。このため、他の CPU またはチャネル・プログラムが更新する可能性のある記憶位置を OR 命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』の『マルチプログラミングとマルチプロセスングの例』に示してあります。

OR IMMEDIATE

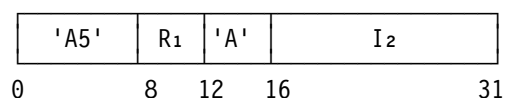
OIHH R₁, I₂ [RI]



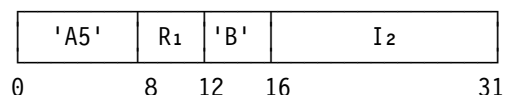
OIHL R₁, I₂ [RI]



OILH R₁, I₂ [RI]



OILL R₁, I₂ [RI]



第 2 オペランドと第 1 オペランドのビットとの論理和 (OR) が求められ、その結果が第 1 オペランドの該当ビットと置き換わります。第 1 オペランドのその他のビットは変更されません。

命令別に見ると、第 2 オペランドとの論理和演算後に置換される第 1 オペランドのビットは次のとおりです。

命令	論理和演算後に置換されるビット
OIHH	0-15
OIHL	16-31
OILH	32-47
OILL	48-63

連結子 OR は、両方のオペランドにビット単位で適用されます。片方または両方のオペランドの対応するビット位置に 1 が含まれている場合は結果のビット位置の内容が 1 に設定され、それ以外の場合は結果ビットが 0 に設定されます。

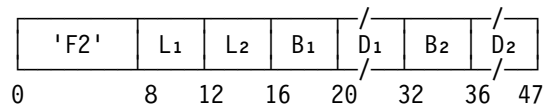
結果の条件コード:

- 0 16 ビットの結果で、0 です。
- 1 16 ビットの結果で、0 ではありません。
- 2 --
- 3 --

プログラム例外: なし。

PACK

PACK D₁(L₁, B₁), D₂(L₂, B₂) [SS]



第 2 オペランドのフォーマットがゾーンからパックに変換され、結果が第 1 オペランド位置に入れられます。ゾーン・フォーマットおよびパック・フォーマットについては、第 8 章、『10 進数命令』で説明されています。

第 2 オペランドは、ゾーン・フォーマットであるものとして扱われます。各バイトの数字ビットは 10 進数として扱われます。右端バイト内のゾーン・ビットは符号として扱われますが、それ以外のゾーン・ビットは無視されます。

符号および数字は無変更のまま第 1 オペランドに移動され、有効なコードかどうかの検査は行われません。符号は、結果フィールドの右端バイトの右端 4 ビット位置に入れられ、数字は、符号に隣接し、数字相互も隣接する形で、結果フィールドの残りの部分に入れられます。

結果は、両方のオペランドを右から左へ処理したのとして求められます。必要があれば、第 2 オペランドは左側が 0 で拡張されたものとして扱われます。第 1 オペランドが短すぎて第 2 オペランドのすべての数字を収容できない場合は、第 2 オペランドの左端の残り部分は無視されます。第 2 オペランドの使用されなかった部分については、アクセス例外が示されることもあり、示されないこともあります。

オペランドがオーバーラップしているときには、必要なオペランド・バイトを取り出した直後に各結果バイトが格納されるときと同様の方法で、結果が取得されます。個々の結果バイトごとに、2 個の第 2 オペランド・バイトが必要です。ただし、結果フィールドの右端バイトのみは別で、ここには右端の第 2 オペランド・バイトしか必要ではありません。

条件コード: コードは変更されません。

プログラム例外:

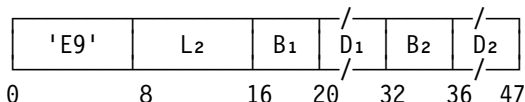
- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)

プログラミング上の注意:

1. PACK 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. PACKは、L₁ フィールドおよび L₂ フィールドに 0 を指定し、両方のオペランドに同じアドレスを指定することによって、1 バイト内の 2 つの 16 進数字を交換するために使用できます。
3. 1 つのフィールドのすべてのバイト (右端バイトも含む) のゾーン・ビットを除去するには、両方のオペランドの右側をダミー・バイトで拡張し、その後の結果フィールドではこのダミー・バイトが無視されるようにします。
4. PACK の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

PACK ASCII

PKA D₁(B₁),D₂(L₂,B₂) [SS]



第 2 オペランドのフォーマットが ASCII からパックに変換され、結果が第 1 オペランド位置に入れます。パック・フォーマットについては、第8章、『10 進数命令』の説明を参照してください。

第 2 オペランドのバイトは、0-9 に対応する 2 進エンコード 0000-1001 で表された 10 進数字が、右端 4 ビット位置に含まれているものとして扱われます。バイトの左端 4 ビット位置は無視されます。第 2 オペランドは正の値と見なされます。

暗黙の正符号 (2 進数 1100) およびソース数字は、第 1 オペランド位置に入れます。ソース数字は、無変更のままで移動され、有効なコードかどうかの検査はされません。符号は、結果フィールドの右端バイトの右端 4 ビットに入れられ、数字は、符号に隣接し、数字相互も隣接する形で、結果フィールドの残りの部分に入れられます。

結果は、両方のオペランドを右から左へ処理したものと求められます。必要があれば、第 2 オペランドは左側が 0 で拡張されたものとして扱われます。

第 1 オペランドの長さは 16 バイトです。

第 2 オペランドの長さは、L₂ フィールドの内容によって指定されます。第 2 オペランドの長さは 32 バイトを超えてはなりません (L₂ が 31 以下でなければなりません)。さもないと、指定例外が認識されます。

第 2 オペランドの長さが 32 バイトの場合は、左端バイトは無視されます。

第 1 オペランドと第 2 オペランドがどのような形であれオーバーラップしている場合、結果は予測不能です。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランド位置は、必ずしも特定の順序で格納されてはいないように見えます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. PACK ASCII の主目的は ASCII 10 進数字のフォーマットを変更することですが、各バイトの左端 4 ビットは無視されるので、使用が ASCII に限定されているわけではありません。
2. 次の例は、ASCII 数字をパックする場合のこの命令の使用法を示しています。

```
ASDIGITS DS      0CL31
          DC      X'3132333435'
          DC      X'3637383930'
          DC      X'3132333435'
          DC      X'3637383930'
          DC      X'3132333435'
          DC      X'3637383930'
          DC      X'31'
PKDIGITS DS      PL16
          ...
          PKA     PKDIGITS,ASDIGITS(31)
```

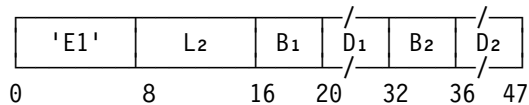
3. この命令を使用して、EBCDIC 数字をパックすることもできます。この方法は、第 2 オペランドの長さが 16 バイト (つまり PACK の第 2 オペランドの限度) より大きい場合に、特に便利です。

```
EBDIGITS DS    0CL31
          DC    X'F1F2F3F4F5'
          DC    X'F6F7F8F9F0'
          DC    X'F1F2F3F4F5'
          DC    X'F6F7F8F9F0'
          DC    X'F1F2F3F4F5'
          DC    X'F6F7F8F9F0'
          DC    X'F1'
PKDIGITS DS    PL16
          ...
          PKA   PKDIGITS,EBDIGITS(31)
```

4. PACK ASCII の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

PACK UNICODE

```
PKU      D1(B1),D2(L2,B2)  [SS]
```



第 2 オペランドのフォーマットが Unicode からパックに変換され、結果が第 1 オペランド位置に入れられます。パック・フォーマットについては、第 8 章、『10 進数命令』の説明を参照してください。

第 2 オペランドの 2 バイト文字は、右端 4 ビット位置に 0-9 に対応する 2 進エンコード 0000-1001 で表された 10 進数字が含まれている、Unicode Basic Latin 文字として扱われます。文字の左端 12 ビット位置は無視されます。第 2 オペランドは正の値と見なされます。

暗黙の正符号 (2 進数 1100) およびソース数字は、第 1 オペランド位置に入れられます。ソース数字は、無変更のまま移動され、有効なコードかどうかの検査はされません。符号は、結果フィールドの右端バイトの右端 4 ビットに入れられ、数字は、符号に隣接し、数字相互も隣接する形で、結果フィールドの残りの部分に入れられます。

結果は、両方のオペランドを右から左へ処理したものとして求められます。必要があれば、第 2 オペランドは左側が 0 で拡張されたものとして扱われます。

第 1 オペランドの長さは 16 バイトです。

第 2 オペランドのバイト長は L2 フィールドの内容によって指定されます。第 2 オペランドの長さは 32 文字 (つまり 64 バイト) を超えてはならず、バイト長は偶数でなければなりません (L2 は 63 以下で、かつ奇数でなければなりません)。さもないと、指定例外が認識されます。

第 2 オペランドの長さが 32 文字 (64 バイト) の場合は、左端バイトは無視されます。

第 1 オペランドと第 2 オペランドがどのような形であれオーバーラップしている場合、結果は予測不能です。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランド位置は、必ずしも特定の順序で格納されてはいないように見えます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. 次の例は、ヨーロッパ表記の数をパックする場合の PACK UNICODE の使用法を示しています。

```
UNDIGITS DS    0CL62
          DC    X'00310032003300340035'
          DC    X'00360037003800390030'
          DC    X'00310032003300340035'
          DC    X'00360037003800390030'
          DC    X'00310032003300340035'
          DC    X'00360037003800390030'
          DC    X'0031'
```

```
PKDIGITS DS    PL16
          ...
          PKU   PKDIGITS,UNDIGITS(62)
```

2. 各文字の左端 12 ビットは無視されるので、数字 0 の右端 4 ビットが 0 であるような Unicode 10 進数も、この命令を使用してパックできます。例えば、タイ語の数字の場合は次のようになります。

```

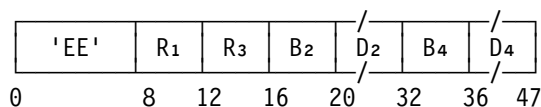
UNDIGITS DS    0CL62
          DC    X'0E510E520E530E540E55'
          DC    X'0E560E570E580E590E50'
          DC    X'0E510E520E530E540E55'
          DC    X'0E560E570E580E590E50'
          DC    X'0E510E520E530E540E55'
          DC    X'0E560E570E580E590E50'
          DC    X'0E51'
PKDIGITS DS    PL16
          ...
          PKU   PKDIGITS,UNDIGITS(62)

```

3. PACK UNICODE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

PERFORM LOCKED OPERATION

PLO R₁,D₂(B₂),R₃,D₄(B₄) [SS]



汎用レジスター 1 に指定されているロックが取得された後で、汎用レジスター 0 内の機能コードにより指定された操作が実行され、その後、ロックが解除されます。ただし、他の CPU から見ると、(1) ロックが取得される前に、記憶域オペランド (使用される可能性のあるパラメーター・リスト内のフィールドを含む) が取り出され、テスト対象位置への格納が可能な場合は格納タイプのアクセス例外についてのテストが行われ、(2) ロックが解除された後で、オペランドがパラメーター・リストに格納されることがあります。パラメーター・リストに含まれていないオペランドがロック取得前に取り出された場合、そのオペランドはロックの取得後に再び取り出されます。

機能コードには、6 つの操作のどれでも指定できます。6 つの操作とは、比較およびロード、比較およびスワップ、二重比較およびスワップ、比較およびスワップおよび格納、比較およびスワップおよび二重格納、比較およびスワップおよび三重格納です。

汎用レジスター 0 のテスト・ビットが 1 のときは、ロックの取得も 6 つの操作の実行もまったく行わず、代わりに機能コードの妥当性をテストすることを指定します。これは、追加操作のための追加機能コードを将来割り当てる場合に便利です。この定義は、他に説明がない限り、テスト・ビットが 0 であるものとして書かれています。

比較およびロードを指定した場合は、第 1 オペランド比較値と第 2 オペランドとが比較されます。両者が等しければ、第 4 オペランドが第 3 オペランド位置に入れます。比較の結果が等しくない場合は、第 2 オペランドが、新しい第 1 オペランド比較値として、第 1 オペランド比較値が入っていた位置に入れます。

比較およびスワップを指定した場合は、第 1 オペランド比較値と第 2 オペランドとが比較されます。両者が等しければ、第 1 オペランド置換値が第 2 オペランド位置に格納されます。比較の結果が等しくない場合は、第 2 オペランドが、新しい第 1 オペランド比較値として、第 1 オペランド比較値が入っていた位置に入れます。

二重比較およびスワップを指定した場合は、第 1 オペランド比較値と第 2 オペランドとが比較されます。両者が等しければ、第 3 オペランド比較値と第 4 オペランドとが比較されます。2 回の比較の結果が両方とも等しければ、第 1 オペランドと第 3 オペランド置換値が、それぞれ第 2 オペランド位置および第 4 オペランド位置に格納されます。最初の比較の結果が等しくない場合は、第 2 オペランドが、新しい第 1 オペランド比較値として、第 1 オペランド比較値が入っていた位置に入れます。最初の結果は等しいが 2 番目の結果が等しくないという場合は、第 4 オペランドが、新しい第 3 オペランド比較値として、第 3 オペランド比較値が入っていた位置に入れます。

比較およびスワップおよび格納、二重格納、または三重格納を指定した場合は、第 1 オペランド比較値と第 2 オペランドとが比較されます。両者が等しければ、第 1 オペランド置換値が第 2 オペランド位置に格納され、第 3 オペランドが第 4 オペランド位置に格納されます。そして、二重格納または三重格納操作の場合は第 5 オペランドが第 6 オペランド位置に格納され、三重格納操作の場合は第 7 オペランドが第 8 オペランド位置に格納されます。第 1 オペランド比較の結果で等しくないことが示された場合は、第 2 オペランドが、新しい第 1 オペランド比較値として、第 1 オペランド比較値が入っていた位置に入れます。

この 6 つの操作のいずれの場合も、操作の後で、比較の結果が条件コードで示されます。

機能コード (FC) は、汎用レジスター 0 のビット位置 56-63 に入っています。機能コードは、実行する操作のみでなく、オペランドの長さ、および、第 1 オペランドの比較値および置換値と、第 3 オペランドまたは第 3 オペランドの比較値および置換値 (これらの値をまとめ

て、集合的に単に第 1 オペランド、第 3 オペランドと
いいます) が汎用レジスターに入っているのかパラメー
ター・リストに入っているのかも指定します。機能コー
ドのパターンは次のとおりです。

- 4 の倍数 (0 を含む) の機能コードは、32 ビット長
で、第 1 オペランドおよび第 3 オペランドは汎用
レジスターのビット位置 32-63 に入っていることを
指定します。
- 4 の倍数より 1 大きい機能コードは、64 ビット長
で、第 1 オペランドおよび第 3 オペランドはパラ
メーター・リストに入っていることを指定します。
- 4 の倍数より 2 大きい機能コードは、64 ビット長
で、第 1 オペランドおよび第 3 オペランドが汎用
レジスターのビット位置 0-63 に入っていることを
指定します。
- 4 の倍数より 3 大きい機能コードは、128 ビット長
で、第 1 オペランドおよび第 3 オペランドはパラ
メーター・リストに入っていることを指定します。

図7-19 は、機能コード、操作名、オペランド長、およ
び説明の中でその操作を示すために使用される記号を示
しています。例えば、機能コード 8 を伴う **PERFORM
LOCKED OPERATION** を表すには、**PLO.DCS** を使用
できます。記号の中で、文字「G」は 64 ビットのオ
ペランド長を示し、文字「R」は 64 ビットのオペラン
ドの一部またはすべてが汎用レジスターに入っているこ
とを示し、文字「X」は 128 ビットのオペランド長を示
します。

CPU は、図7-19 にリストした機能コードで指定される
操作をすべて実行することができます。CPU が実行可
能な操作を指定している機能コードは、有効と表現され
ます。操作に割り当てられていない機能コード、または
操作がインプリメント (インストール) されていないた
めに CPU で実行不可能な操作を指定している機能コー
ドは、無効と呼ばれます。

汎用レジスター 0 のビット 55 はテスト・ビット (T) で
す。ビット 55 が 0 のときは、汎用レジスター 0 内の
機能コードは有効でなければなりません。さもないと、
指定例外が認識されます。ビット 55 が 1 のときは、機
能コードが有効であれば条件コードが 0 に設定され、機
能コードが無効であれば条件コードが 3 に設定されま
す。それ以外の操作は行われません。

汎用レジスター 0 のビット 32-54 は、すべて 0 でなけ
ればなりません。さもないと、指定例外が認識されま
す。このレジスターのビット 55 が 1 のときは、これは

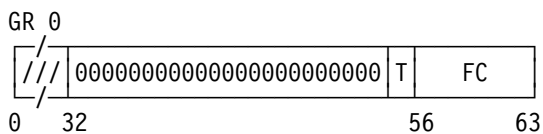
機能 コード	操作	オペランド 長 (ビット)	機能記号
0	比較およびロード	32	CL
1	0 と同じ	64	CLG
2	0 と同じ	64	CLGR
3	0 と同じ	128	CLX
4	比較およびスワップ	32	CS
5	4 と同じ	64	CSG
6	4 と同じ	64	CSGR
7	4 と同じ	128	CSX
8	二重比較およびスワ ップ	32	DCS
9	8 と同じ	64	DCSG
10	8 と同じ	64	DCSGR
11	8 と同じ	128	DCSX
12	比較およびスワップ および格納	32	CSST
13	12 と同じ	64	CSSTG
14	12 と同じ	64	CSSTGR
15	12 と同じ	128	CSSTX
16	比較およびスワップ および二重格納	32	CSDST
17	16 と同じ	64	CSDSTG
18	16 と同じ	64	CSDSTGR
19	16 と同じ	128	CSDSTX
20	比較およびスワップ および三重格納	32	CSTST
21	20 と同じ	64	CSTSTG
22	20 と同じ	64	CSTSTGR
23	20 と同じ	128	CSTSTX

図 7-19. **PERFORM LOCKED OPERATION** の機能コード
と操作

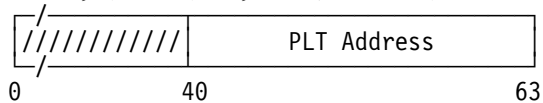
認識される唯一の例外となります。汎用レジスター 0 の
ビット 0-31 は無視されます。

使用されるロックは、プログラム・ロック・トークン (PLT: program lock token) で表されます。このトークンの論理アドレスは、汎用レジスター 1 に指定されます。24 ビット・アドレッシング・モードでは、PLT アドレスは汎用レジスター 1 のビット 40-63 で、このレジスターのビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、PLT アドレスはこのレジスターのビット 33-63 で、ビット 0-32 は無視されます。64 ビット・アドレッシング・モードでは、PLT アドレスはこのレジスターのビット 0-63 です。

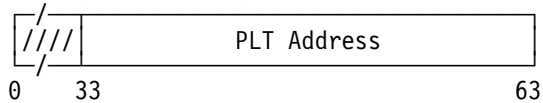
上記で説明した汎用レジスター 0 および 1 の内容は、以下のとおりです。



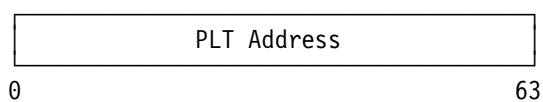
24 ビット・アドレッシング・モードのときの GR 1



31 ビット・アドレッシング・モードのときの GR 1



64 ビット・アドレッシング・モードのときの GR 1



偶数番号の機能コード (0 も含む) では、第 1 オペランド比較値は汎用レジスター R_1 にあります。4 で始まる偶数番号の機能コードでは、第 1 オペランド置換値が汎用レジスター $R_1 + 1$ にあります。 R_1 はレジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されま

す。機能コード 0 および 2 では、 R_1 は偶数でも奇数でも構いません。

機能コード 0、2、12、および 14 では、第 3 オペランドが汎用レジスター R_3 にあります。 R_3 は偶数でも奇数でも構いません。

機能コード 8 および 10 では、第 3 オペランド比較値が汎用レジスター R_3 にあり、第 3 オペランド置換値が汎用レジスター $R_3 + 1$ にあります。 R_3 はレジスターの奇偶ペアを差し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

すべての機能コードで、この命令の B_2 および D_2 フィールドは、第 2 オペランド・アドレスを指定します。

機能コード 0、2、8、10、12、および 14 では、この命令の B_4 および D_4 フィールドは、第 4 オペランド・アドレスを指定します。

機能コード 1、3、5、7、9、11、13、15、および 16-23 では、この命令の B_4 および D_4 フィールドは、命令で使用されるパラメーター・リストのアドレスを指定します。このアドレスは第 4 オペランド・アドレスとは呼ばれません。パラメーター・リストには、比較値および置換値が入っている奇数番号のオペランド、および第 2 オペランド以外の偶数番号のオペランドのアドレスが含まれています。アクセス・レジスター・モードの場合は、このパラメーター・リストに、偶数番号オペランドのアドレスに関連したアクセス・リスト・エンター・トークン (ALET) も含まれています。

アクセス・レジスター・モードでは、ALET が含まれたパラメーター・リストを使用することになる機能コードの場合、 R_3 は 0 であってはなりません。0 の場合は、指定例外が認識されます。

R_1 および R_3 に関する規則と、 B_4 および D_4 の使用法の要約が、7-115 ページの図 7-20 に示されています。

機能コード	操作	R ₁	R ₃	D ₄ (B ₄)
0 および 2	比較およびロード	E0	E0	Op4a
1 および 3	比較およびロード	-	NZ	PLa
4 および 6	比較およびスワップ	E	-	-
5 および 7	比較およびスワップ	-	-	PLa
8 および 10	二重比較およびスワップ	E	E	Op4a
9 および 11	二重比較およびスワップ	-	NZ	PLa
12 および 14	比較およびスワップ および格納	E	E0	Op4a
13 および 15	比較およびスワップ および格納	-	NZ	PLa
16 および 18	比較およびスワップ 二重格納	E	NZ	PLa
17 および 19	比較およびスワップ および二重格納	-	NZ	PLa
20 および 22	比較およびスワップ および三重格納	E	NZ	PLa
21 および 23	比較およびスワップ および三重格納	-	NZ	PLa

説明:

- 無視されます。
- E 偶数でなければなりません。
- E0 偶数でも奇数でも構いません。
- NZ アクセス・レジスター・モードの場合は 0 以外の値でなければなりません。その他の場合は無視されます。
- Op4a D₄(B₄) はオペランド 4 のアドレスです。
- PLa D₄(B₄) はパラメーター・リストのアドレスです。

図 7-20. PERFORM LOCKED OPERATION のレジスター規則と D₄(B₄) の使用方法

7-116ページの図7-21 は、オペランド (オペランドの比較値および置換値も含む)、オペランド・アドレス、および命令で使用されるパラメーター・リストの、それぞれの記憶位置を示しています。

パラメーター・リスト内のオペランド・アドレスは、使用される場合、リスト内でダブルワードを占めます。24 ビット・アドレッシング・モードでは、オペランド・アドレスはダブルワードのビット 40-63 で、このダブルワードのビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、オペランド・アドレスはダブルワードのビット 33-63 で、このダブルワードのビット 0-32 は無視されます。64 ビット・アドレッシング・モードでは、オペランド・アドレスはダブルワードのビット 0-63 です。

アクセス・レジスター・モードでは、アクセス・レジスター 1 はプログラム・ロック・トークン (PLT) が入っているアドレス・スペースを指定し、アクセス・レジスター B₂ は第 2 オペランドが入っているアドレス・スペースを指定し、アクセス・レジスター B₄ は第 4 オペランドまたは 7-116ページの図7-21 に示すパラメーター・リストが入っているアドレス・スペースを指定し

ます。また、パラメーター・リストにアドレスが含まれているオペランドの場合は、アクセス・リスト・エンターリー・トークン (ALET) がアドレスと共にそのリストに含まれていて、オペランドが入っているアドレス・スペースを指定するためにアクセス・レジスター・モードで使用されます。

アクセス・レジスター・モードでは、アドレスがパラメーター・リストに含まれているオペランドに関してアクセス例外または PER 記憶域変更イベントが認識されると、パラメーター・リスト内のそれに関連した ALET が、例外またはイベントが認識されたときにアクセス・レジスター R₃ にロードされます。そして、結果として発生するプログラム割り込みのときに、値が例外アクセス ID として実記憶位置 160 に、または PER アクセス ID として実記憶位置 161 に格納される場合には、R₃ が格納されます。例外もイベントも認識されずに命令の実行が完了した場合は、アクセス・レジスター R₃ の内容は予測不能です。アクセス・レジスター・モードでない場合、または ALET が含まれたパラメーター・リストが使用されない場合は、アクセス・レジスター R₃ の内容は変更されません。

偶数番号 (2、4、6、および 8) の記憶域オペランドは、規定境界上に指定する必要があります。規定境界とは、4 の倍数の機能コードではワード境界であり、4 の倍数より 1 または 2 大きい機能コードではダブルワード境界であり、4 の倍数より 3 大きい機能コードではクワッドワード境界です。パラメーター・リストは、使用する場合はダブルワード境界上に指定する必要があります。さもないと、指定例外が認識されます。汎用レジスター 1 内のプログラム・ロック・トークン (PLT) アドレスは、境界合わせをする必要はありません。

パラメーター・リスト内のすべての未使用フィールドには、すべての桁に 0 が含まれていなければなりません。さもないと、将来、プログラムの動作の互換性が失われることがあります。

逐次化操作は、ロックが獲得された直後に行われるほか、ロックが解除される直前に再度行われます。ただし、ロックが獲得される前にパラメーター・リストから取り出された値が、必ず再取り出しされるとは限りません。テスト・ビット (汎用レジスター 0 のビット 55) が 1 の場合は、逐次化操作は実行されません。

以下の図は、さまざまな機能コードのパラメーター・リストを示しています。左側に示されているオフセットはバイト値です。

機能 コード ¹	操作	Op1c	Op1r	Op2a	Op3 または		Op4a	Op5	Op7	
					Op3c	Op3r		Op6a	Op8a	
0 および 2	比較およびロード	R ₁	-	D ₂ (B ₂)	R ₃		D ₄ (B ₄)	-	-	-
1 および 3	比較およびロード	PL	-	D ₂ (B ₂)	PL		PL	-	-	D ₄ (B ₄)
4 および 6	比較およびスワップ	R ₁	R ₁₊₁	D ₂ (B ₂)	-		-	-	-	-
5 および 7	比較およびスワップ	PL	PL	D ₂ (B ₂)	-		-	-	-	D ₄ (B ₄)
8 および 10	二重比較および スワップ	R ₁	R ₁₊₁	D ₂ (B ₂)	R ₃	R ₃₊₁	D ₄ (B ₄)	-	-	-
9 および 11	二重比較および スワップ	PL	PL	D ₂ (B ₂)	PL	PL	PL	-	-	D ₄ (B ₄)
12 および 14	比較およびスワップ および格納	R ₁	R ₁₊₁	D ₂ (B ₂)	R ₃		D ₄ (B ₄)	-	-	-
13 および 15	比較およびスワップ および格納	PL	PL	D ₂ (B ₂)	PL		PL	-	-	D ₄ (B ₄)
16 および 18	比較およびスワップ および二重格納	R ₁	R ₁₊₁	D ₂ (B ₂)	PL		PL	PL	-	D ₄ (B ₄)
17 および 19	比較およびスワップ および二重格納	PL	PL	D ₂ (B ₂)	PL		PL	PL	-	D ₄ (B ₄)
20 および 22	比較およびスワップ および三重格納	R ₁	R ₁₊₁	D ₂ (B ₂)	PL		PL	PL	PL	D ₄ (B ₄)
21 および 23	比較およびスワップ および三重比較	PL	PL	D ₂ (B ₂)	PL		PL	PL	PL	D ₄ (B ₄)

説明:

- ¹ 4 の倍数の機能コード (0 を含む)、または 4 の倍数より 1 大きい機能コードでは、汎用レジスタ内のオペランドはレジスタのビット位置 32-63 に入り、レジスタのビット 0-31 は無視されて無変更のままになります。4 の倍数より 2 大きい機能コードでは、汎用レジスタ内のオペランドはレジスタのビット位置 0-63 に入ります。
- オペランド、値、またはアドレスは、操作には使用されません。
- OpNc オペランド N の比較値。
- OpNr オペランド N の置換値。
- OpNa オペランド N のアドレス。
- PL オペランド、値、またはアドレスはパラメーター・リストに入っています。
- PLa パラメーター・リストのアドレス。

図 7-21. PERFORM LOCKED OPERATION のオペランドおよびアドレスの位置

機能コード 0-3 (比較およびロード)

命令で使用されるオペランドおよびアドレスの記憶位置は、7-116ページの図7-21 に示されています。

機能コード 1 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 1 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	
32	
40	Operand 3
48	
56	
64	Operand-4 ALET
72	Operand-4 Address

機能コード 3 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 3 のパラメーター・リスト

0	Operand-1 Comparison Value
8	Operand-1 Comp. Value (continued)
16	
24	
32	Operand 3
40	Operand 3 (continued)
48	
56	
64	Operand-4 ALET
72	Operand-4 Address

第 1 オペランド比較値が第 2 オペランドと比較されず。第 1 オペランド比較値が第 2 オペランドに等しい場合は、第 3 オペランドが第 4 オペランドで置き換えられ、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない場合は、第 1 オペランド比較値が第 2 オペランドで置き換えられ、条件コード 1 が設定されます。

機能コード 4-7 (比較およびスワップ)

命令で使用されるオペランドおよびアドレスの記憶位置は、7-116ページの図7-21 に示されています。

機能コード 5 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 5 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	Operand-1 Replacement Value

機能コード 7 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 7 のパラメーター・リスト

0	Operand-1 Comparison Value
8	Operand-1 Comp. Value (continued)
16	Operand-1 Replacement Value
24	Operand-1 Repl. Value (continued)

第 1 オペランド比較値が第 2 オペランドと比較されず。第 1 オペランド比較値が第 2 オペランドに等しい場合は、第 1 オペランド置換値が第 2 オペランド位置に格納され、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない場合は、第 1 オペランド比較値が第 2 オペランドで置き換えられ、条件コード 1 が設定されます。

機能コード 8-11 (二重比較およびスワップ)

命令で使用されるオペランドおよびアドレスの記憶位置は、7-116ページの図7-21 に示されています。

機能コード 9 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 9 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	Operand-1 Replacement Value
32	
40	Operand-3 Comparison Value
48	
56	Operand-3 Replacement Value
64	Operand-4 ALET
72	Operand-4 Address

機能コード 11 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 11 のパラメーター・リスト

0	Operand-1 Comparison Value
8	Operand-1 Comp. Value (continued)
16	Operand-1 Replacement Value
24	Operand-1 Repl. Value (continued)
32	Operand-3 Comparison Value
40	Operand-3 Comp. Value (continued)
48	Operand-3 Replacement Value
56	Operand-3 Repl. Value (continued)
64	Operand-4 ALET
72	Operand-4 Address

第 1 オペランド比較値が第 2 オペランドと比較されず。第 1 オペランド比較値が第 2 オペランドに等しい場合は、第 3 オペランド比較値が第 4 オペランドと比較されます。第 3 オペランド比較値が第 4 オペランドに等しい場合 (第 1 オペランド比較値が第 2 オペランドに等しいことが判明した後) は、第 1 オペランド置換値が第 2 オペランド位置に格納され、第 3 オペランド置換値が第 4 オペランド位置に格納されて、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない場合は、第 1 オペランド比較値が第 2 オペランドで置き換えられ、条件コード 1 が設定されます。

第 3 オペランド比較値が第 4 オペランドに等しくない場合 (第 1 オペランド比較値が第 2 オペランドに等しいことが判明した後) は、第 3 オペランド比較値が第 4 オペランドで置き換えられ、条件コード 2 が設定されます。

機能コード 12-15 (比較およびスワップおよび格納)

命令で使用されるオペランドおよびアドレスの記憶位置は、7-116ページの図7-21 に示されています。

機能コード 13 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 13 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	Operand-1 Replacement Value
32	
40	
48	
56	Operand 3
64	Operand-4 ALET
72	Operand-4 Address

機能コード 15 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 15 のパラメーター・リスト

0	Operand-1 Comparison Value	
8	Operand-1 Comp. Value (continued)	
16	Operand-1 Replacement Value	
24	Operand-1 Repl. Value (continued)	
32		
40		
48	Operand 3	
56	Operand 3 (continued)	
64		Operand-4 ALET
72	Operand-4 Address	

第 1 オペランド比較値が第 2 オペランドと比較されません。第 1 オペランド比較値が第 2 オペランドに等しい場合は、第 1 オペランド置換値が第 2 オペランド位置に格納され、第 3 オペランドが第 4 オペランド位置に格納されて、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない場合は、第 1 オペランド比較値が第 2 オペランドで置き換えられ、条件コード 1 が設定されます。

機能コード 16-19 (比較およびスワップおよび二重格納)

命令で使用されるオペランドおよびアドレスの記憶位置は、7-116ページの図7-21 に示されています。

機能コード 16 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 16 のパラメーター・リスト

0		
8		
16		
24		
32		
40		
48		
56		Operand 3
64		Operand-4 ALET
72	Operand-4 Address	
80		
88		Operand 5
96		Operand-6 ALET
104	Operand-6 Address	

機能コード 17 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 17 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	Operand-1 Replacement Value
32	
40	
48	
56	Operand 3
64	Operand-4 ALET
72	Operand-4 Address
80	
88	Operand 5
96	Operand-6 ALET
104	Operand-6 Address

機能コード 18 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 18 のパラメーター・リスト

0	
8	
16	
24	
32	
40	
48	
56	Operand 3
64	Operand-4 ALET
72	Operand-4 Address
80	
88	Operand 5
96	Operand-6 ALET
104	Operand-6 Address

機能コード 19 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 19 のパラメーター・リスト

0	Operand-1 Comparison Value	
8	Operand-1 Comp. Value (continued)	
16	Operand-1 Replacement Value	
24	Operand-1 Repl. Value (continued)	
32		
40		
48	Operand 3	
56	Operand 3 (continued)	
64		Operand-4 ALET
72	Operand-4 Address	
80	Operand 5	
88	Operand 5 (continued)	
96		Operand-6 ALET
104	Operand-6 Address	

第 1 オペランド比較値が第 2 オペランドと比較されま
す。第 1 オペランド比較値が第 2 オペランドに等しい
場合は、第 1 オペランド置換値が第 2 オペランド位置
に格納され、第 3 オペランドが第 4 オペランド位置に
格納されて、第 5 オペランドが第 6 オペランド位置に
格納され、そして、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない
場合は、第 1 オペランド比較値が第 2 オペランドで置
き換えられ、条件コード 1 が設定されます。

機能コード 20-23 (比較およびスワップおよび三重格納)

命令で使用されるオペランドおよびアドレスの記憶位置
は、7-116ページの図7-21 に示されています。

機能コード 20 の場合に使用されるパラメーター・リス
トのフォーマットは、以下のとおりです。

機能コード 20 のパラメーター・リスト

0		
8		
16		
24		
32		
40		
48		
56		Operand 3
64		Operand-4 ALET
72	Operand-4 Address	
80		
88		Operand 5
96		Operand-6 ALET
104	Operand-6 Address	
112		
120		Operand 7
128		Operand-8 ALET
136	Operand-8 Address	

機能コード 21 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 21 のパラメーター・リスト

0	
8	Operand-1 Comparison Value
16	
24	Operand-1 Replacement Value
32	
40	
48	
56	Operand 3
64	Operand-4 ALET
72	Operand-4 Address
80	
88	Operand 5
96	Operand-6 ALET
104	Operand-6 Address
112	
120	Operand 7
128	Operand-8 ALET
136	Operand-8 Address

機能コード 22 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 22 のパラメーター・リスト

0	
8	
16	
24	
32	
40	
48	
56	Operand 3
64	Operand-4 ALET
72	Operand-4 Address
80	
88	Operand 5
96	Operand-6 ALET
104	Operand-6 Address
112	
120	Operand 7
128	Operand-8 ALET
136	Operand-8 Address

機能コード 23 の場合に使用されるパラメーター・リストのフォーマットは、以下のとおりです。

機能コード 23 のパラメーター・リスト

0	Operand-1 Comparison Value	
8	Operand-1 Comp. Value (continued)	
16	Operand-1 Replacement Value	
24	Operand-1 Repl. Value (continued)	
32		
40		
48	Operand 3	
56	Operand 3 (continued)	
64		Operand-4 ALET
72	Operand-4 Address	
80	Operand 5	
88	Operand 5 (continued)	
96		Operand-6 ALET
104	Operand-6 Address	
112	Operand 7	
120	Operand 7 (continued)	
128		Operand-8 ALET
136	Operand-8 Address	

第 1 オペランド比較値が第 2 オペランドと比較されず。第 1 オペランド比較値が第 2 オペランドに等しい場合は、第 1 オペランド置換値が第 2 オペランド位置に格納され、第 3 オペランドが第 4 オペランド位置に格納され、第 5 オペランドが第 6 オペランド位置に格納され、第 7 オペランドが第 8 オペランド位置に格納されます。そして、条件コード 0 が設定されます。

第 1 オペランド比較値が第 2 オペランドに等しくない場合は、第 1 オペランド比較値が第 2 オペランドで置き換えられ、条件コード 1 が設定されます。

ロック

ロックは、操作の開始時に取得され、操作の終了時に解放されます。取得されるロックは、プログラム・ロック・トークン (PLT) で表されます。このトークンの論理アドレスは、前に述べたように、汎用レジスター 1 に指定されます。

PLT は、PLT 論理アドレスのモデル依存変換によって生成される値です。モデルに応じて、PLT は PLT 論理アドレスから直接導き出される場合もあり、DAT がオンの場合には PLT 論理アドレスの変換の結果である実アドレスから DAT によって導き出される場合もあります。DAT を使用する場合は、アクセス・レジスター・モードでは、DAT の前にアクセス・レジスター変換 (ART) が行われます。

PLT は、構成内のロックの 1 つを選択します。ロックの数はモデルによって異なります。異なる CPU で実行中の複数のプログラムが同一ロックを指定できるのは、それらの CPU がすべて同一の PLT 論理アドレスを指定し、しかもそれらのアドレスを同じ実アドレスに変換できる場合に限られます。

同じモデルでも、PLT を形成するときに ART および DAT を使用することもしないこともあるので、ART および DAT の実行中に検出可能なアクセス例外条件は、例外として認識されることもあり、認識されないこともあります。PLT により指定された記憶位置へのアクセスは行われませんが、その記憶位置についてのアドレッシング例外が認識されることはあります。どのような理由でも、PLT 論理アドレスの処理中に記憶保護例外が認識されることはありません。

1 つの CPU は、一時点で 1 つのロックしか保持できません。

CPU が PERFORM LOCKED OPERATION が実行して、他の CPU が PERFORM LOCKED OPERATION 命令の実行によってすでに保持しているロックを使用しようとする、後からそのロックを使用しようとした CPU の実行は、ロックが解放されるまで延期されず。過度の遅延が起きるのはマシン誤動作の場合のみで、これはマシン・チェック条件に該当します。

同一ロックに対する複数の要求が満たされる順序は決まっています。

ロックを保持している間に CPU に回復不能障害が起きた場合は、マシン・チェックが起きる、チェック停止状態に入る、またはシステム・チェック停止に陥るなどの結果になることがあります。マシン・チェックは、すべ

てのオペランドにまったく損傷がない場合は処理バックアップであり、レジスター・オペランドが損傷している場合は処理損傷です。結果がマシン・チェックまたはチェック停止状態のときは、どの記憶域オペランドも変更されていないか、または変更される予定のすべての記憶域オペランドが変更されているか、どちらかであって、どちらの場合もロックは解放されています。記憶域オペランドが、その正常な元の状態でも正常な最終状態でもない場合は、結果はシステム・チェック停止になります。

記憶域オペランドの参照

偶数番号の記憶域オペランドへのアクセスは、他の CPU から見た場合、4 の倍数の機能コードの場合はワード単位のように見え、4 の倍数より 1、2、または 3 大きい機能コードの場合はダブルワード単位のように見えます。パラメーター・リスト内のダブルワードへのアクセスは、他の CPU から見れば、機能コードに関係なくダブルワード単位のように見えます。

他の CPU から見れば、すべての記憶域オペランドは、ロックの取得前にアクセス例外の有無がテストされるように見えます。(チャンネル・プログラムはロックを監視できません。)

他の CPU から見れば、比較およびスワップ操作 (これには第 4 オペランドがありません) 以外のすべての操作で、ロックが保持されている間に第 4 オペランドへのアクセスが行われるのは、ロックの保持中に第 1 オペランド比較値と第 2 オペランドとの比較により等しいという結果が示された場合のみです。これらの操作において、ロックが保持される前に第 4 オペランドにアクセスされるのは、第 1 オペランド比較値と第 2 オペランドとの比較結果が等しい場合のみです。また、DAT がオンのときは、第 2 オペランドの取り出し後に他の CPU により実行された INVALIDATE PAGE TABLE ENTRY 命令が原因で、第 4 オペランドについてのページ変換例外が認識される結果にならなかった場合に限りです。この例外が認識されるのは、この CPU が該当のページ・テーブル・エントリーに対応する TLB エントリーを持っていないために、この命令が第 4 オペランド用のページ・テーブル・エントリー内のページ無効ビットを 1 に設定した場合です。比較およびスワップおよび二重格納操作、および比較およびスワップおよび三重格納操作では、第 6 オペランドは (三重格納操作では第 8 オペランドも) 上で説明した第 4 オペランドと同様に扱われません。INVALIDATE PAGE TABLE ENTRY に関するこの指定の理由は、『プログラミング上の注意 6 (7-125 ページ)』で説明されています。

オペランドへのアクセスが上記で述べたように禁止されていない場合は、比較の結果が理由となって格納が行われなかった場合にも、オペランドについての格納タイプのアクセス例外が認識されることがあります。記憶域変更 PER イベントが認識され、変更ビットが設定されるのは、格納が行われた場合のみです。

ロックの取得前にオペランド比較値とオペランドとの間で比較が行われ、等しいという結果が示された場合でも、ロックは取得されます。ロックが保持されている間に行われた比較の結果としてのみ、条件コードが設定されます。条件コード 1 または 2 が設定された場合、第 1 オペランド比較値または第 3 オペランド比較値は、他の CPU から見ると、ロックが保持されている間にそれぞれ第 2 オペランドまたは第 4 オペランドを取り出すことによってのみ置き換えられます。

格納が第 2 オペランド位置と、第 4、第 6、および第 8 オペランド位置のうちの 1 つ以上に対して行われる場合は、他の CPU およびチャンネル・プログラムから見ると、第 2 オペランド位置への格納が常に最後に行われます。

パラメーター・リストへの格納は、ロックが保持されている間に行われることも、ロックが解放されてから行われることもあります。

逐次化操作は、ロックが獲得された直後に行われるほか、ロックが解除される直前に再度行われます。ただし、ロックが獲得される前にパラメーター・リストから取り出された値が、必ず再取り出しされるとは限りません。テスト・ビット (汎用レジスター 0 のビット 55) が 1 の場合は、逐次化操作は実行されません。

パラメーター・リスト位置についてのアクセス例外は、その位置が操作に必要な場合でも認識されることがあります。これに該当するのは、オフセット 0 で始まり、使用されている機能コード用に定義された最後の記憶位置までです。

比較およびロード操作、および比較およびスワップ操作では、何らかのアドレッシング例外および保護例外が起きたときは、操作が抑止されます。

ロックを保持している間に CPU に回復不能障害が起きた結果、マシン・チェックが起きるかまたはチェック停止状態に入った場合は、どの記憶域オペランドもまったく変更されていないか、または変更される予定のすべての記憶域オペランドが正常に変更されているか、どちらかです。正常な変更は、障害の発生前に正常に実行された格納を繰り返すことによって達成されます。したがって、格納タイプのオペランドに対して、最初の格納値と 2 番目の格納値が等しい、2 つの単一アクセス格納参照

(多くの場合は更新参照の格納部分と、それに続く格納参照) がある場合があります。

結果の条件コード:

テスト・ビットが 0 のときは、次のようになります。

- 0 すべての比較結果が等しい。1 つ以上の置換値が格納またはロードされました。
- 1 第 1 オペランドの比較が等しくない。第 1 オペランド比較値が置き換えられました。
- 2 -- (二重比較およびスワップ以外のすべての操作)
- 2 第 1 オペランドの比較は等しいが、第 3 オペランドの比較が等しくない。第 3 オペランド比較値は置き換えられました (二重比較およびスワップ)。
- 3 --

テスト・ビットが 1 のときは、次のようになります。

- 0 機能コードは有効です。
- 1 --
- 2 --
- 3 機能コードは無効です。

プログラム例外:

- アクセス例外 (すべての機能コードのときの取り出し: ただし、PLT 位置のアドレッシング例外および保護例外、プログラム・ロック・トークン例外、モデル依存例外を除く。すべての機能コードのときの取り出しおよび格納: オペランド 2。偶数番号の機能コードのときの取り出しおよび格納: パラメーター・リスト。機能コード 16、18、20、および 22 のときの取り出し: パラメーター・リスト。機能コード 0-3 のときの取り出し: オペランド 4。機能コード 8-11 のときの取り出しおよび格納: オペランド 4。機能コード 12-23 のときの格納: オペランド 4。機能コード 16-23 のときの格納: オペランド 6。機能コード 20-23 のときの格納: オペランド 8)
- 指定例外

プログラミング上の注意:

1. **PERFORM LOCKED OPERATION** 命令の使用例は、『付録A、『数の表現と命令の使用例』』に示されています。
2. 記憶位置の内容を変更する **PERFORM LOCKED OPERATION** 命令が同一ロックを使用して複数の CPU によって並行して実行される場合、パラメーター・リストに含まれていないオペランドに対する変更は、CPU のうちのどれか 1 つが変更を完了してから、他の CPU による変更が開始されます。変

更の開始順序は、どの CPU が最初にロックを取得するかによって決まります。

3. **PERFORM LOCKED OPERATION** の比較およびスワップ機能の実行には、インターロック更新参照は使用されません。他の CPU による記憶域オペランドへの並行格納参照は、インターロック更新参照であった場合でも、**PERFORM LOCKED OPERATION** の本来意図された操作に予測不能な干渉を生じ、結果のレジスターおよび記憶域の内容に影響が出ることがあります。予測可能な記憶域結果を取得する必要がある場合は、記憶位置の内容に対する変更はすべて、同一ロックを使用する **PERFORM LOCKED OPERATION** 命令によって行う必要があります。
4. **PERFORM LOCKED OPERATION** の実行中に CPU に回復不能な障害が発生すると、格納タイプ オペランドに同じ値が 2 つ格納されることがあるので、別の CPU が **PERFORM LOCKED OPERATION** を使用せずに同じオペランドに対して同時に格納操作をした場合、その値は失われることがあります。
5. **DAT** がオンのときに異なるアドレス・スペース内のプログラムが同一ロックを使用する場合は、それらのプログラムは、変換が行われるアドレス・スペースに関係なく、同一であってかつ同一の実アドレスに変換される **PLT** 論理アドレスを使用する状態を確保する必要があります。この状態を確保できないと、プログラム間で実際には異なるロックを使用する可能性があります。
6. 7-124 ページの『記憶域オペランドの参照』に、**INVALIDATE PAGE TABLE ENTRY (IPTE)** 命令の指定に関する説明があります。次の例は、この指定がなくても成り立つものですが、この指定の必要性を示しています。
 - a. CPU 1 は、機能コード 8 を伴う **PERFORM LOCKED OPERATION** 命令の実行を開始します (この命令を **PLO.DCS** と呼ぶことにします)。オペランド 2 は記憶位置 **Qtail** で、キューの最後のエレメント (エレメント **X**) のアドレス (第 1 オペランド比較値) が入っています。オペランド 4 はそのエレメント内の記憶位置で、キュー内の次の (この場合は存在していない) エレメントのアドレス (0、第 3 オペランド比較値) が入っています。**PLO** 命令の目的は、エレメントのアドレス (第 1 オペランドおよび第 3 オペランド置換値) をオペランド 2 および オペランド 4 の両方に入れることによって、エレメントをエンキューすることです。ロ

ックが保持されていない場合は、PLO 命令はオペランド 2 を取り出し、それを第 1 オペランド比較値と比較して、等しいという結果を得ます。

- b. CPU 2 は、PLO.DCS 命令を完全に実行して、キューの唯一の元素である元素 X を、このキューから除去 (デキュー) します。PLO 命令は、Qtail および Qhead に 0 を格納します。Qhead は、キューの最初の元素のアドレスが入っている記憶位置です。CPU 2 のプログラムは、デキューされた元素を処理し、次に制御プログラムの主記憶域解放 (freemain) サービスを呼び出して、この元素が含まれていた記憶域の割り振りを解除します。主記憶域解放サービスは、IPTE を使用して、元素 X を含むページのページ・テーブル・エントリ内のページ無効ビットを 1 に設定します。IPTE 命令は、即時にページ無効ビットを 1 に設定し、他のすべての CPU がそのページに対応する各自のエントリの TLB をクリアしたことを示す信号を待ちます。
- c. CPU 1 は、オペランド 4 を取り出そうと試みます。CPU 1 は、オペランド 4 ページ用の TLB エントリを持っていません。CPU 1 は、CPU 2 の IPTE 命令が先に進んでもよいことを示す信号を CPU 2 に送ります。
- d. CPU 2 は、その IPTE 命令を完了します。CPU 2 のプログラムは、ページ・テーブル・エントリのソフトウェア・ビットを 1 に設定します。これによって、該当ページが主記憶域

から解放されたこと、したがってそのページへの参照を行うと、参照を行ったプログラムに対して制御プログラムがアドレッシング例外を提示する結果になることが示されます。

- e. CPU 1 は、オペランド 4 についての DAT を実行しようとして、ページ無効ビットが 1 であることを検出します。CPU 1 は、ページ変換例外を示すプログラム割り込みを実行します。例外ハンドラーは、主記憶域解放を示すソフトウェア・ビットが 1 であることを検出し、CPU 1 プログラムにアドレッシング例外を提示し、それによってこのプログラムは異常終了します。

CPU 1 が該当ページの TLB エントリを持っている場合は、PLO 命令に対する割り込みは発生せず、ロックが保持されている間の第 1 オペランド比較値と第 2 オペランドとの比較の結果、CPU 2 が第 2 オペランドを変更したことが示されます。この場合、PLO 命令は条件コード 1 を設定します。

CPU 1 が TLB エントリを持っておらず、CPU 1 が命令を実行している間に IPTE がページ無効ビットを 1 に設定できなかった場合でも、CPU 1 は正常にオペランド 4 アドレスを変換でき、ロックが保持されている間にオペランド 2 が変更済みであることを検出します。オペランド 2 が元素 X を指しているが、元素 X ページの主記憶域解放ビットが 1 であるという場合は、プログラミング・エラーです。

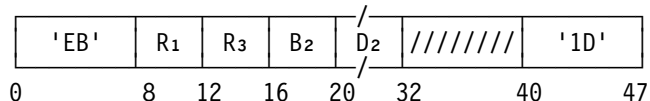
- 7. 7-127ページの図7-22 に、この操作の結果の要約を示します。

Op1c=Op2	Op3c=Op4	条件 コード	アクション	
機能コード 0-3 (比較およびロード)				
No	-	1		Op2 → Op1c
Yes	-	0	Op4 → Op3	
機能コード 4-7 (比較およびスワップ)				
No	-	1		Op2 → Op1c
Yes	-	0	Op1r → Op2	
機能コード 8-11 (二重比較およびスワップ)				
No	-	1		Op2 → Op1c
Yes	No	2		Op4 → Op3c
Yes	Yes	0	Op1r → Op2 Op3r → Op4	
機能コード 12-15 (比較およびスワップおよび格納)				
No	-	1		Op2 → Op1c
Yes	-	0	Op1r → Op2 Op3 → Op4	
機能コード 16-19 (比較およびスワップおよび二重格納)				
No	-	1		Op2 → Op1c
Yes	-	0	Op1r → Op2 Op3 → Op4 Op5 → Op6	
機能コード 20-23 (比較およびスワップおよび三重格納)				
No	-	1		Op2 → Op1c
Yes	-	0	Op1r → Op2 Op3 → Op4 Op5 → Op6 Op7 → Op8	
説明:				
- 適用されません。				
OpNc オペランド N の比較値。				
OpNr オペランド N の置換値。				

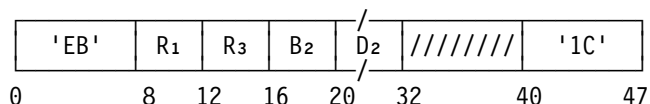
図 7-22. PERFORM LOCKED OPERATION の結果の要約

ROTATE LEFT SINGLE LOGICAL

RLL R₁, R₃, D₂ (B₂) [RSE]



RLLG R₁, R₃, D₂ (B₂) [RSE]



32 ビットまたは 64 ビットの第 3 オペランドが、第 2 オペランド・アドレスにより指定されるビット数だけ左に回転され、結果が第 1 オペランド位置に入れられます。第 3 オペランドは、汎用レジスタ R₃ 内で無変更のままになります。ROTATE LEFT SINGLE LOGICAL (RLL) では、汎用レジスタ R₁ および R₃ のビット 0-31 は変更されません。

第 2 オペランド・アドレスはデータのアドレッシングには使用されません。その右端 6 ビットは、回転するビット位置の数を示します。アドレスの残りの部分は無視されます。

RLL では、第 1 オペランドおよび第 3 オペランドは、それぞれ、汎用レジスタ R₁ および R₃ のビット位置 32-63 に入っています。RLLG では、両オペランドは、これらのレジスタのビット位置 0-63 に入っています。

第 3 オペランドの 32 ビットまたは 64 ビットのすべてが左シフトの対象となります。シフトによってオペランドの左端ビット位置から外に出た各ビットは、同じオペランドの右端ビット位置に再び入ります。

条件コード: コードは変更されません。

プログラム例外: なし。

SEARCH STRING

SRST R₁,R₂ [RRE]

0	16	24	28	31
'B25E'	////////	R ₁	R ₂	

第 2 オペランド内での検索は、指定された文字が見つかるか、第 2 オペランドの終わりに達するか、または CPU で決定されるバイト数の検索が完了するか、いずれかの条件が最初に発生するまで続けられます。CPU で決定される数の最小値は 256 です。結果は条件コードで示されます。

第 2 オペランドの左端バイトの位置は、汎用レジスタ R₂ の内容によって指定されます。第 2 オペランドの後に続く最初のバイトの位置は、汎用レジスタ R₁ の内容によって指定されます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いには、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

アクセス・レジスタ・モードでは、第 2 オペランドが含まれるアドレス・スペースは、アクセス・レジスタ R₂ のみを使用して指定されます。アクセス・レジスタ R₁ の内容は無視されます。

検索対象の文字は、汎用レジスタ 0 のビット位置 56-63 に指定されます。汎用レジスタ 0 のビット位置 32-55 は、将来の拡張に備えて予約されており、すべて 0 が含まれていなければなりません。さもないと、指定例外が認識されます。

操作は左から右へと進み、指定された文字が第 2 オペランド内で見つかるか、次に検査する第 2 オペランド・バイトのアドレスが汎用レジスタ R₁ 内のアドレスに一致するか、または CPU で決定される数の第 2 オペランド・バイトの検査が完了するか、いずれかの条件が最初に発生した時点で終了します。CPU で決定される数の最小値は 256 です。指定された文字が見つかった場合は、条件コード 1 が設定されます。次に検査する第 2 オペランド・バイトのアドレスが汎用レジスタ R₁ 内のアドレスに一致する場合は、条件コード 2 が設定されます。CPU で決定される数の第 2 オペランド・バイトの検査が完了した場合は、条件コード 3 が設定されます。CPU で決定される数の第 2 オペランド・バイトの検査が完了し、次の第 2 オペランド・バイトのアドレスが汎用レジスタ R₁ に入っている場合、条件コード 2 と 3 のどちらが設定されるかは予測不能です。

条件コード 1 が設定された場合は、第 2 オペランド内で見つかった指定の文字が汎用レジスタ R₁ に入れられ、汎用レジスタ R₂ の内容は変更されません。条件コード 3 が設定された場合は、第 2 オペランド内の次に処理するバイトのアドレスが汎用レジスタ R₂ に入れられ、汎用レジスタ R₁ の内容は変更されません。条件コード 2 が設定された場合、汎用レジスタ R₁ および R₂ の内容は変更されません。汎用レジスタにアドレスが入れられるたびに、そのレジスタのビット 32-39 (24 ビット・アドレッシング・モードの場合) またはビット 32 (31 ビット・アドレッシング・モードの場合) が 0 に設定されます。R₁ および R₂ レジスタのビット 0-31 は、24 ビット・モードの場合も 31 ビット・モードの場合も変更されません。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

第 2 オペランドに関するアクセス例外が認識されるのは、必ず検査されるオペランド部分についてのみです。

記憶域オペランド一貫性の規則は、COMPARE LOGICAL LONG 命令の場合と同じです。

結果の条件コード:

- 0 --
- 1 指定された文字が見つかりました。汎用レジスター R1 はその文字のアドレスで更新され、汎用レジスター R2 は変更されていません。
- 2 第 2 オペランド全体を検索しても、指定された文字は見つかりませんでした。汎用レジスター R1 および R2 は変更されていません。
- 3 CPU で決定されたバイト数の検索が完了しました。汎用レジスター R1 は変更されていません。汎用レジスター R2 は、次のバイトのアドレスで更新されています。

プログラム例外:

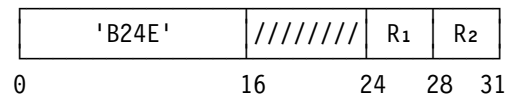
- アクセス例外 (取り出し: オペランド 2)
- 指定例外

プログラミング上の注意:

1. SEARCH STRING 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 条件コード 3 が設定されると、プログラムはブランチに従って命令に戻り、検索を継続できます。検索済みのバイト数をプログラムで判別する必要はありません。
3. 汎用レジスター R1 内のアドレスが汎用レジスター R2 内のアドレスに等しい場合は、即時に条件コード 2 が設定され、アクセス例外は認識されません。汎用レジスター R1 内のアドレスが汎用レジスター R2 内のアドレスより小さい場合は、オペランドが記憶域の先頭から記憶位置 0 に循環する場合のみ、条件コード 2 が設定されます。
4. R1 または R2 が 0 の場合もあります。この場合、汎用レジスター 0 は、アドレスのほかに指定された文字も含んでいるものとして扱われます。
5. 長さが不明なstring内で終了文字を検索する必要がありますが、次の 3 つの条件が成り立つとします。
 - (1) stringが、記憶位置 256 (終了文字が 16 進数 00 の場合は、記憶位置 1) より低位の位置から始まっていない、(2)stringが記憶位置 0 に循環しない、そして (3) 汎用レジスター 0 内の指定された文字を保存する必要がない。この場合は、SEARCH STRING が使用する汎用レジスターを 3 つでなく 2 つだけにするために、R1 を 0 にすることができます。

SET ACCESS

SAR R1,R2 [RRE]



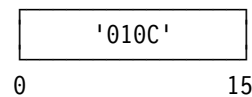
汎用レジスター R2 のビット位置 32-63 の内容が、アクセス・レジスター R1 に入られます。

条件コード: コードは変更されません。

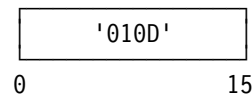
プログラム例外: なし。

SET ADDRESSING MODE

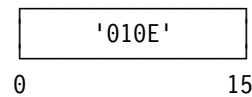
SAM24 [E]



SAM31 [E]



SAM64 [E]



拡張アドレッシング・モード・ビット (現 PSW のビット 31) および基本アドレッシング・モード・ビット (現 PSW のビット 32) を以下のように設定することによって、アドレッシング・モードを設定できます。

命令	PSW ビット 31	PSW ビット 32	結果のアドレッシング・モード
SAM24	0	0	24 ビット
SAM31	0	1	31 ビット
SAM64	1	1	64 ビット

PSW 内の命令アドレスは、以下のように、新しいアドレッシング・モードの設定に従って更新されます。PSW のビット位置 64-127 の内容に、値 2 (命令長) が追加されるか、または、命令が EXECUTE のターゲットの場合は値 4 が追加されます。どちらの場合も、ビット位置

0 からの繰り上がりは無視されます。次に、新規アドレッシング・モードが 24 ビット・モードのときは PSW のビット 64-103 が 0 に設定され、新規アドレッシング・モードが 31 ビット・モードのときはビット 64-96 が 0 に設定されます。

命令が完了するのは、新規アドレッシング・モードと PSW 内の未更新の命令アドレスの組み合わせが有効な場合のみです。新規アドレッシング・モードが 24 ビット・モードのときは、未更新 PSW のビット 64-103 はすべて 0 でなければなりません。または、新規アドレッシング・モードが 31 ビット・モードのときは、未更新 PSW のビット 64-96 はすべて 0 でなければなりません。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- 指定例外 (SAM24 および SAM31 のみ)
- トレース例外

プログラミング上の注意: 未更新の命令アドレスを検査することで、命令の完了が妨げられる主なケースが 2 つあります。それは、命令がアドレス 2^{24} 以上の位置にあり、新規アドレッシング・モードが 24 ビットである場合、または、命令がアドレス 2^{31} 以上の位置にあり、新規アドレッシング・モードが 24 ビット・モードまたは 31 ビット・モードの場合です。これらのケースでは、命令が完了したとしても、新規アドレッシング・モードの設定に基づいて命令アドレスが更新された結果、アドレスの 1 つ以上の左端ビットが 0 に設定されてしまい、その結果として、次の順番に当たる記憶位置以外の位置から次の命令が取り出されることがあります。このアクションは、「誤った分岐」と呼ばれることがあります。命令の記憶位置が $2^{24} - 2$ または $2^{31} - 2$ の場合、あるいは、EXECUTE が使用されている場合には $2^{24} - 4$ または $2^{31} - 4$ の場合にも、誤った分岐が起きる場合があります。

SET PROGRAM MASK

SPM R₁ [RR]

'04'	R ₁	////
0	8	12 15

第 1 オペランドを使用して、条件コード、および現 PSW のプログラム・マスクが設定されます。

汎用レジスター R₁ のビット 34 および 35 が条件コードと置き換わり、ビット 36-39 がプログラム・マスクと置き換わります。汎用レジスター R₁ のビット 0-33 および 40-63 は無視されます。

条件コード: コードは、汎用レジスター R₁ のビット 34 および 35 で指定されているとおりに設定されます。

プログラム例外: なし。

プログラミング上の注意:

1. 汎用レジスターのビット 34-39 は、24 ビット・アドレッシング・モードで BRANCH AND LINK を実行することによって、あるいは、24 ビットまたは 31 ビットのどちらかのアドレッシング・モードで INSERT PROGRAM MASK を実行することによって、PSW からロードされています。
2. SET PROGRAM MASK では、問題プログラム状態または監視プログラム状態のどちらでも、条件コードおよびマスク・ビットを設定することができます。
3. プログラムでは、プログラム・マスクを設定すると後続のプログラム実行に重大な影響が出る可能性があることを考慮に入れておく必要があります。4 個のマスク・ビットがそれに対応する割り込みの発生を制御するだけでなく、指数アンダーフローおよび有効数字マスクも、取得される結果を決定する要因になります。

SHIFT LEFT DOUBLE

SLDA R₁,D₂(B₂) [RS]

'8F'	R ₁	////	B ₂	D ₂
0	8	12	16	20 31

符号付き第 1 オペランドの 63 ビットの数値部分が、第 2 オペランド・アドレスにより指定されるビット数だけ左にシフトされ、結果が第 1 オペランド位置に入られます。第 1 オペランドは、汎用レジスター R₁ のビット 32-63 と、その右側に続く汎用レジスター R₁ + 1 のビット 32-63 から成っています。

R₁ フィールドは、汎用レジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

第 1 オペランドは、64 ビットの符号付き 2 進整数として扱われます。第 1 オペランドの符号ビット (偶数番号のレジスターのビット 32) は、変更されません。奇数番号のレジスターのビット位置 32 には、数字ビットが含まれています。この数字ビットは、他の数字ビットと同じ方法でシフトに関与します。右側の空いたビット位置には 0 が入れられます。汎用レジスター R₁ および R₁ + 1 のビット 0-31 は変更されません。

符号ビットとは異なる 1 つ以上のビットが、シフトによって偶数番号レジスターのビット位置 32 から外に出た場合は、オーバーフローが生じて、条件コード 3 が設定されます。固定小数点オーバーフロー・マスク・ビットが 1 であれば、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

- 固定小数点オーバーフロー例外
- 指定例外

プログラミング上の注意:

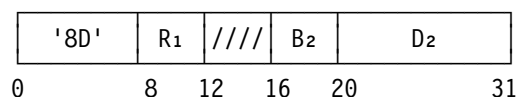
1. SHIFT LEFT DOUBLE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. ESA/390 と z/Architecture の両方に存在する 8 個のシフト命令では、1 個の汎用レジスター内の 32 ビットに関して、あるいは倍長の場合は 2 個の汎用レジスターのそれぞれの 32 ビットに関して、左または右、単長または倍長、符号付きまたは論理の、3 組のシフトが選択できます。z/Architecture で追加されている 4 個のシフト命令では、1 個の汎用レジスター内に 64 ビットに関して、左または右、符号付きまたは論理シフトが選択可能です。符号付きシフトと論理シフトの相違点は、符号付きシフトで

は、オーバーフローが認識され、条件コードが設定され、左端ビットが符号として扱われることです。

3. 2 つの符号付き倍長シフト操作で、シフト量として 0 を指定すると、倍長の符号および絶対値のテストを行うことができます。
4. 第 2 オペランド・アドレスの生成には基底レジスターが関与するので、基底レジスターにシフト量を入れることによって、シフト量の間接指定ができます。B₂ フィールドに 0 を入れると、間接シフト指定はないことを意味します。

SHIFT LEFT DOUBLE LOGICAL

SLDL R₁,D₂(B₂) [RS]



64 ビットの第 1 オペランドが、第 2 オペランド・アドレスにより指定されるビット数だけ左にシフトされ、結果が第 1 オペランド位置に入れられます。第 1 オペランドは、汎用レジスター R₁ のビット 32-63 と、その右側に続く汎用レジスター R₁ + 1 のビット 32-63 から成っています。

R₁ フィールドは、汎用レジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

第 1 オペランドの 64 ビットすべてがシフトの対象になります。シフトによって偶数番号レジスターのビット位置 32 から外に出たビットは、検査されず、失われます。右側の空いたビット位置には 0 が入れられます。汎用レジスター R₁ および R₁ + 1 のビット 0-31 は変更されません。

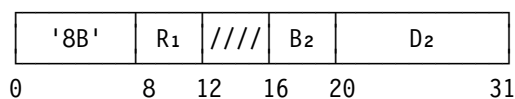
条件コード: コードは変更されません。

プログラム例外:

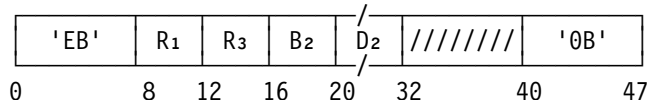
- 指定例外

SHIFT LEFT SINGLE

SLA $R_1, D_2(B_2)$ [RS]



SLAG $R_1, R_3, D_2(B_2)$ [RSE]



SHIFT LEFT SINGLE (SLA) の場合は、符号付き第 1 オペランドの 31 ビットの数値部分が、第 2 オペランド・アドレスで指定されるビット数だけ左にシフトされ、結果が第 1 オペランド位置に入れられます。汎用レジスタ R_1 のビット 0-31 は変更されません。

SHIFT LEFT SINGLE (SLAG) の場合は、符号付き第 3 オペランドの 63 ビットの数値部分が、第 2 オペランド・アドレスで指定されるビット数だけ左にシフトされ、その結果の左側に第 3 オペランドの符号ビットを付加したものが、第 1 オペランド位置に入れられます。第 3 オペランドは、汎用レジスタ R_3 内で無変更のままになります。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

SLA の場合は、第 1 オペランドは、汎用レジスタ R_1 のビット位置 32-63 にある 32 ビット符号付き 2 進整数として扱われます。第 1 オペランドの符号は変更されません。このオペランドの 31 個の数字ビットのすべてが、左シフトの対象となります。

SLAG の場合は、第 1 オペランドおよび第 3 オペランドは、それぞれ、汎用レジスタ R_1 および R_3 のビット位置 0-63 にある 64 ビット符号付き 2 進整数として扱われます。第 1 オペランドの符号は、第 3 オペランドの符号と同じに設定されます。第 3 オペランドの 63 個の数字ビットのすべてが、左シフトの対象となります。

SLA および SLAG のどちらの場合も、右側の空いたビット位置には 0 が入れられます。

シフトにより、符号ビットとは異なる 1 つ以上のビットが、SLA の場合はビット位置 33 から、また SLAG の場合はビット位置 1 から外へ出る場合は、オーバーフローが起き、条件コード 3 が設定されます。固定小数点オーバーフロー・マスク・ビットが 1 であれば、固定小数点オーバーフローに関するプログラム割り込みが起きません。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

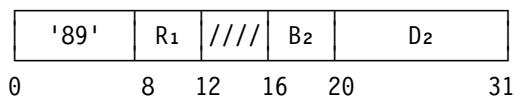
- 固定小数点オーバーフロー例外

プログラミング上の注意:

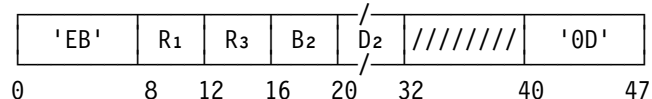
- SHIFT LEFT SINGLE 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
- SHIFT LEFT SINGLE (SLA) では、 -2^{30} 以上で 2^{30} 未満である数値を左に 1 ビット位置だけシフトすることは、その数値に 2 を掛けるのと同じことです。SHIFT LEFT SINGLE (SLAG) では、これに相当する値は -2^{62} および 2^{62} です。
- SHIFT LEFT SINGLE (SLA) では、シフト桁数が 31 から 63 までの範囲内である場合は、シフトにより数字部分の全体がレジスタから送り出され、結果として負の最大数かまたは 0 が残ります。どちらになるかは、初期内容が負であったかどうかによって決まります。SHIFT LEFT SINGLE (SLAG) では、シフト桁数が 63 の場合に同じ結果が生じます。

SHIFT LEFT SINGLE LOGICAL

SLL $R_1, D_2(B_2)$ [RS]



SLLG R₁,R₃,D₂(B₂) [RSE]



SHIFT LEFT SINGLE LOGICAL (SLL) の場合は、32 ビットの第 1 オペランドが、第 2 オペランド・アドレスに指定されているビット数だけ左にシフトされ、その結果が第 1 オペランド位置に入れられます。汎用レジスター R₁ のビット 0-31 は変更されません。

SHIFT LEFT SINGLE LOGICAL (SLLG) の場合は、64 ビットの第 3 オペランドが、第 2 オペランド・アドレスに指定されているビット数だけ左にシフトされ、その結果が第 1 オペランド位置に入れられます。第 3 オペランドは、汎用レジスター R₃ 内で無変更のままになります。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

SLL では、第 1 オペランドは、汎用レジスター R₁ のビット位置 32-63 です。このオペランドの 32 ビットすべてが左シフトの対象となります。

SLLG では、第 1 オペランドおよび第 3 オペランドは、それぞれ、汎用レジスター R₁ および R₃ のビット位置 0-63 です。第 3 オペランドの 64 ビットすべてが、左シフトの対象となります。

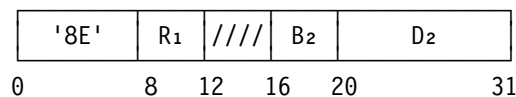
SLL および SLLG のどちらの場合も、右側の空いたビット位置には 0 が入れられます。

条件コード: コードは変更されません。

プログラム例外: なし。

SHIFT RIGHT DOUBLE

SRDA R₁,D₂(B₂) [RS]



符号付き第 1 オペランドの 63 ビットの数値部分が、第 2 オペランド・アドレスにより指定されるビット数だけ

右にシフトされ、結果が第 1 オペランド位置に入れられます。第 1 オペランドは、汎用レジスター R₁ のビット 32-63 と、その右側に続く汎用レジスター R₁ + 1 のビット 32-63 から成っています。

R₁ フィールドは、汎用レジスターの奇偶ペアを指し、偶数番号のレジスターを指定している必要があります。さもないと、指定例外が認識されます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

第 1 オペランドは、64 ビットの符号付き 2 進整数として扱われます。第 1 オペランドの符号ビット (偶数番号のレジスターのビット 32) は、変更されません。奇数番号のレジスターのビット位置 32 には、数字ビットが含まれています。この数字ビットは、他の数字ビットと同じ方法でシフトに関与します。シフトによって奇数番号レジスターのビット位置 63 から外に出たビットは、検査されず、失われます。左側の空いたビット位置には、符号と同じビットが入れられます。汎用レジスター R₁ および R₁ + 1 のビット 0-31 は変更されません。

結果の条件コード:

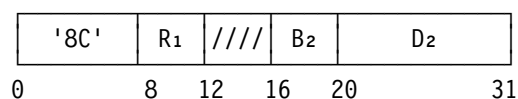
- 0 結果は 0 です。
- 1 結果は 0 より小さくなります。
- 2 結果は 0 より大きくなります。
- 3 --

プログラム例外:

- 指定例外

SHIFT RIGHT DOUBLE LOGICAL

SRDL R₁,D₂(B₂) [RS]



64 ビットの第 1 オペランドが、第 2 オペランド・アドレスにより指定されるビット数だけ右にシフトされ、結果が第 1 オペランド位置に入れられます。第 1 オペランドは、汎用レジスター R₁ のビット 32-63 と、その右側に続く汎用レジスター R₁ + 1 のビット 32-63 から成っています。

R1 フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定する必要があります。さもないと、指定例外が認識されます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

第 1 オペランドの 64 ビットすべてがシフトに関与します。シフトによって奇数番号レジスタのビット位置 63 から外に出たビットは、検査されず、失われます。左側の空いたビット位置には 0 が入れられます。汎用レジスタ **R1** および **R1 + 1** のビット 0-31 は変更されません。

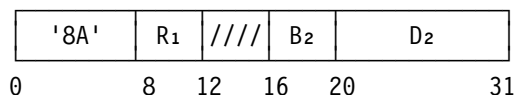
条件コード: コードは変更されません。

プログラム例外:

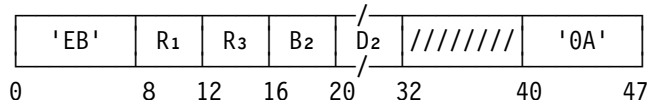
- 指定例外

SHIFT RIGHT SINGLE

SRA **R1, D2(B2)** [RS]



SRAG **R1, R3, D2(B2)** [RSE]



SHIFT RIGHT SINGLE (SRA) の場合は、符号付き第 1 オペランドの 31 ビットの数値部分が、第 2 オペランド・アドレスで指定されるビット数だけ右にシフトされ、結果が第 1 オペランド位置に入れられます。汎用レジスタ **R1** のビット 0-32 は変更されません。

SHIFT RIGHT SINGLE (SRAG) の場合は、符号付き第 3 オペランドの 63 ビットの数値部分が、第 2 オペランド・アドレスで指定されるビット数だけ右にシフトされ、その結果の左側に第 3 オペランドの符号ビットを付加したものが、第 1 オペランド位置に入れられます。第 3 オペランドは、汎用レジスタ **R3** 内で無変更のままになります。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトする

ビット位置の数を示します。アドレスの残りの部分は無視されます。

SRA の場合は、第 1 オペランドは、汎用レジスタ **R1** のビット位置 32-63 にある 32 ビット符号付き 2 進整数として扱われます。第 1 オペランドの符号は変更されません。このオペランドの 31 個の数字ビットのすべてが、右シフトの対象となります。

SRAG の場合は、第 1 オペランドおよび第 3 オペランドは、それぞれ、汎用レジスタ **R1** および **R3** のビット位置 0-63 にある 64 ビット符号付き 2 進整数として扱われます。第 1 オペランドの符号は、第 3 オペランドの符号と同じに設定されます。第 3 オペランドの 63 個の数字ビットのすべてが、右シフトの対象となります。

SRA および SRAG のどちらの場合も、シフトによりビット位置 63 から外に出るビットは、検査されずに失われます。左側の空いたビット位置には、符号と同じビットが入れられます。

結果の条件コード:

- 0 結果は 0 です。
- 1 結果は 0 より小さくなります。
- 2 結果は 0 より大きくなります。
- 3 --

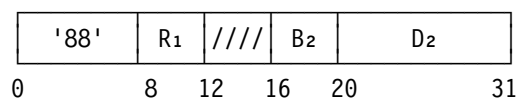
プログラム例外: なし。

プログラミング上の注意:

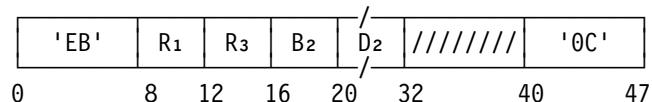
1. ビット位置を 1 つ分右にシフトすることは、2 で割って結果の端数を切り捨てることと同じです。偶数の数値を右に 1 桁シフトした場合は、結果はその数値を 2 で割るのと同じことです。奇数の数値を右に 1 桁シフトした場合は、その数の次に小さい数値を 2 で割るのと同じことです。例えば、+5 を右に 1 ビット位置だけシフトすると +2 になりますが、-5 の場合は -3 になります。
2. SHIFT RIGHT SINGLE (SRA) では、シフト桁数が 31 から 63 までの範囲内である場合は、シフトにより数字部分の全体がレジスタから送り出され、結果として -1 または 0 が残ります。どちらになるかは、初期内容が負であったかどうかによって決まります。SHIFT RIGHT SINGLE (SRAG) では、シフト桁数が 63 の場合と同じ結果が生じます。

SHIFT RIGHT SINGLE LOGICAL

SRL $R_1, D_2(B_2)$ [RS]



SRLG $R_1, R_3, D_2(B_2)$ [RSE]



SHIFT RIGHT SINGLE LOGICAL (SRL) の場合は、32 ビットの第 1 オペランドが、第 2 オペランド・アドレスに指定されているビット数だけ右にシフトされ、その結果が第 1 オペランド位置に入られます。汎用レジスタ R_1 のビット 0-31 は変更されません。

SHIFT RIGHT SINGLE LOGICAL (SRLG) の場合は、64 ビットの第 3 オペランドが、第 2 オペランド・アドレスに指定されているビット数だけ右にシフトされ、その結果が第 1 オペランド位置に入られます。第 3 オペランドは、汎用レジスタ R_3 内で無変更のままになります。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。その右端 6 ビットは、シフトするビット位置の数を示します。アドレスの残りの部分は無視されます。

SRL では、第 1 オペランドは、汎用レジスタ R_1 のビット位置 32-63 です。このオペランドの 32 ビットすべてが右シフトの対象となります。

SRLG では、第 1 オペランドおよび第 3 オペランドは、それぞれ、汎用レジスタ R_1 および R_3 のビット位置 0-63 です。第 3 オペランドの 64 ビットすべてが、右シフトの対象となります。

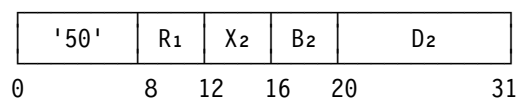
SRL および SRLG のどちらの場合も、シフトによりビット位置 63 から外に出るビットは、検査されずに失われます。左側の空いたビット位置には 0 が入れられます。

条件コード: コードは変更されません。

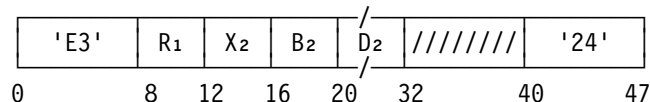
プログラム例外: なし。

STORE

ST $R_1, D_2(X_2, B_2)$ [RX]



STG $R_1, D_2(X_2, B_2)$ [RXE]



第 1 オペランドが、無変更のままで第 2 オペランド位置に入られます。

STORE (ST) ではオペランドは 32 ビットで、STORE (STG) ではオペランドは 64 ビットです。

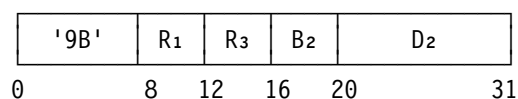
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)

STORE ACCESS MULTIPLE

STAM $R_1, R_3, D_2(B_2)$ [RS]



アクセス・レジスタ R_1 からアクセス・レジスタ R_3 までの一連のアクセス・レジスタの内容が、第 2 オペランド・アドレスが指す記憶位置に格納されます。

アクセス・レジスタの内容が入られる記憶域は、第 2 オペランド・アドレスが指す位置から始まり、指定されたアクセス・レジスタの数と同数の記憶域ワードに達するまで続きます。アクセス・レジスタの内容は、アクセス・レジスタ R_1 から始まりアクセス・レジスタ R_3 まで (このレジスタも含む)、レジスタ番号の昇順に従って格納されます。アクセス・レジスタ 0 はアクセス・レジスタ 15 の後に続きます。アクセス・レジスタの内容は変更されません。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

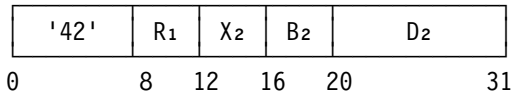
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 指定例外

STORE CHARACTER

STC R₁,D₂(X₂,B₂) [RX]



汎用レジスタ R₁ のビット 56-63 が、無変更のまま第 2 オペランド位置に入れられます。第 2 オペランドの長さは 1 バイトです。

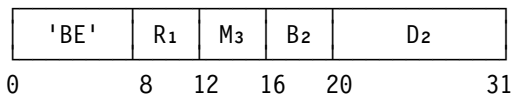
条件コード: コードは変更されません。

プログラム例外:

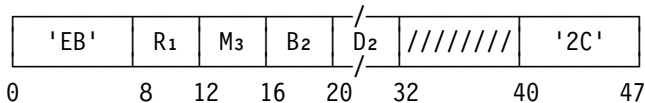
- アクセス例外 (格納: オペランド 2)

STORE CHARACTERS UNDER MASK

STCM R₁,M₃,D₂(B₂) [RS]



STCMH R₁,M₃,D₂(B₂) [RSE]



マスクに従って汎用レジスタ R₁ から選択されたバイトが、第 2 オペランド・アドレスから始まる連続したバイト位置に入れられます。

M₃ フィールドの内容がマスクとして使用されます。この 4 個のビットは、汎用レジスタ R₁ の中の 4 個のバイトと、左から右へ 1 対 1 の関係で対応しています。STORE CHARACTERS UNDER MASK (STCM) では、マスク・ビットに対応する 4 個のバイトは、汎用レジスタ R₁ のビット位置 32-63 に入っています。STORE CHARACTERS UNDER MASK (STCMH) で

は、この 4 個のバイトはレジスタの高位の半分で、ビット位置 0-31 を占めています。マスク内の 1 に対応する各バイトが、第 2 オペランド・アドレスから始まる連続した記憶位置に、同じ順序で入れられます。マスクが 0 でない場合は、第 2 オペランドの長さはマスク内の 1 の個数と同じになります。汎用レジスタの内容は変更されません。

マスクが 0 でない場合は、マスクによって指定された数のバイトについてのみ、記憶域オペランド・アクセスに関連した例外が認識されます。

マスクが 0 の場合は、第 2 オペランドに指定されている 1 バイトは変更されません。ただし、モデルによっては、その値がいったん取り出されて、再び同じ記憶位置に無変更のまま格納されることがあります。この更新は、他の CPU からはインターロック更新参照のように見えます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意:

1. STORE CHARACTERS UNDER MASK 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 0111 のマスクを指定して STORE CHARACTERS UNDER MASK を使用することにより、例えば CCW 内のアドレスを変更するためなどに、3 バイトのアドレスを格納することができます。
3. 1111、0011、または 0001 のマスクを指定した STORE CHARACTERS UNDER MASK (STCM) は、それぞれ、STORE (ST)、STORE HALFWORD、または STORE CHARACTER と同じ働きをします。ただし、ほとんどのモデルでは、STORE CHARACTERS UNDER MASK の方が効率の点で劣ります。
4. 0 のマスクを指定して STORE CHARACTERS UNDER MASK を使用することは、避けてください。このような使い方をすると、モデルによっては、第 2 オペランド・アドレスに指定されている 1 バイトの取り出しと格納が行われることがあります。この参照は、チャンネル・プログラムによるアクセスに対してインターロックされません。さらに、第 2 オペランド・アドレスに指定されているバイトについて、PER 記憶域変更イベントが認識される、アクセス例外が認識される、または、アクセス

例外が存在しない場合に変更ビットが 1 に設定される、などの現象が生じるおそれがあります。記憶域の内容は無変更のままになるので、PER 記憶域変更イベントが認識された場合、変更ビットは 1 になる場合もあり、ならない場合もあります。

STORE CLOCK

STCK D₂(B₂) [S]

	'B205'	B ₂	D ₂	
0		16	20	31

TOD クロックが、設定状態、停止状態、または非設定状態にある場合に、クロックのビット 0-63 の現在の値が、第 2 オペランド・アドレスに指定されている 8 バイト・フィールドに格納されます。

クロックが停止している場合は、クロックの走行中に増分する右端のビット位置の右側の位置に 0 が格納されます。走行中のクロックの値が格納される時は、増分される右端のビットの右側の位置に 0 以外の値が格納されることがあります。これは、必ず固有の値が格納されるようにするためです。

クロックから値が供給されない右端のビット位置には、すべて 0 が格納されます。

クロックがエラー状態または非稼働状態にあるときは、オペランド位置に 0 が格納されます。

この命令により格納されるクロック値の性質は、結果の条件コードの設定により示されます。

クロックの値が取り出される前と、値が記憶域に入れられた後で、逐次化機能が実行されます。

結果の条件コード:

- 0 クロックは設定状態にあります。
- 1 クロックは非設定状態にあります。
- 2 クロックはエラー状態にあります。
- 3 クロックは停止状態または非稼働状態にあります。

プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意:

1. クロックのビット 31 は、1.048576 秒ごとに増分します。したがって、人間の応答が関係するタイミング・アプリケーションでは、左端のクロック・ワードを使用するだけで十分な分解能が得られる場合があります。
2. 条件コード 0 は、通常、クロックが制御プログラムにより設定されていることを意味します。したがって、結果の値は、経過時間測定に使用することも、有効な時刻と日付の表示として使用することもできます。条件コード 1 は、クロック値が、クロックの電源がオンになってから経過した時間であることを示します。この場合は、値は経過時間測定には使用できませんが、有効な時刻表示ではありません。条件コード 2 および 3 は、STORE CLOCK から提供された値が、時間の測定や表示には使用できないことを意味します。
3. 条件コード 3 は、クロックが停止状態または非稼働状態にあることを示します。通常の場合、クロックが非稼働状態のときはすべて 0 の値が格納されるので、この 2 つの状態を区別することができます。
4. z/Architecture 用に書かれた問題プログラムをシステム/370 モードのシステムでも実行する必要がある場合は、システム/370 モードでは、条件コードが 2 のときに格納されている値は必ずしも 0 ではないということを考慮に入れてください。

STORE CLOCK EXTENDED

STCKE D₂(B₂) [S]

	'B278'	B ₂	D ₂	
0		16	20	31

TOD クロックが、設定状態、停止状態、または非設定状態にある場合に、クロックのビット 0-103 の現在の値が、第 2 オペランド・アドレスに指定されている 16 バイト・フィールドのバイト位置 1-13 に格納されます。バイト位置 0 には 0 が格納されます。バイト位置 14 および 15 には、TOD プログラマブル・フィールド (TOD プログラマブル・レジスターのビット 16-31) が格納されます。

上記で述べたオペランドのフォーマットは、以下のとおりです。

Zeros	TOD Clock	Programmable Field
0	8	112 127

クロックが停止している場合は、クロック値の中の、クロックの走行中に増分する右端のビット位置の右側の位置に 0 が格納されます。この場合も、プログラマブル・フィールドは格納されます。

走行中のクロックの値が格納される時は、クロックのビット位置 64-103 (記憶域オペランドのビット位置 72-111) の値は、常に 0 以外の値になります。これで、STORE CLOCK EXTENDED により格納される値は、必ず、STORE CLOCK が格納する値の右側に 0 を付加したものと異なる値になります。

クロックがエラー状態または非稼働状態にあるときは、オペランド位置に 0 が格納されます。

この命令により格納されるクロック値の性質は、結果の条件コードの設定により示されます。

クロックの値が取り出される前と、値が記憶域に入れられた後で、逐次化機能が実行されます。

結果の条件コード:

- 0 クロックは設定状態にあります。
- 1 クロックは非設定状態にあります。
- 2 クロックはエラー状態にあります。
- 3 クロックは停止状態または非稼働状態にあります。

プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意:

1. 条件コード 0 は、通常、クロックが制御プログラムにより設定されていることを意味します。したがって、結果の値は、経過時間測定に使用することも、有効な時刻と日付の表示として使用することもできます。条件コード 1 は、クロック値が、クロックの電源がオンになってから経過した時間であることを示します。この場合は、値は経過時間測定には使用できませんが、有効な時刻表示ではありません。条件コード 2 および 3 は、STORE CLOCK

EXTENDED から提供された値が、時間の測定や表示には使用できないことを意味します。

2. 4-36から始まる『プログラミング上の注意』には、STORE CLOCK 命令により格納される TOD クロックの値に関連した 16 進数値が示されています。以下の注意事項 3 から 5 までは、上記と同じ注意事項の繰り返しですが、STORE CLOCK EXTENDED が格納する値のビット 0-71 に合わせて、説明文と 16 進数値を変えてあります。
3. 次の表は、STORE CLOCK EXTENDED が格納する TOD クロック値の各ビットが、どのような時間間隔で進められるかを示しています。この時間値は、該当ビットが 1 のときに表す加重時間値と考えることもできます。ビット番号は、STORE CLOCK EXTENDED のオペランドのビットを指しています。

STCKE ビット	時間間隔			
	日	時間	分	秒
59				0.000 001
55				0.000 016
51				0.000 256
47				0.004 096
43				0.065 536
39				1.048 576
35				16.777 216
31			4	28.435 456
27		1	11	34.967 296
23		19	5	19.476 736
19	12	17	25	11.627 776
15	203	14	43	6.044 416
11	3257	19	29	36.710 656

4. 次の表は、幾つかの日付について、UTC 時刻 00:00:00 (0 am) での STORE CLOCK EXTENDED のオペランドのビット 0-63 の設定を示しています。設定されている時刻は、1900 年 1 月 1 日と 1900 年 1 月 1 日、1972 年 1 月 1 日、そして、1999 年 1 月までに発生した 22 個のうるう秒が、それぞれ発生した直後の時刻です。これらのうるう秒は、それぞれ、リストされている日の前の日の 23:59:60 UTC から、リストされている日の 00:00:00 UTC までの間の UTC 時間枠の中に挿入されています。

年	月	日	うるう秒	STCKE 値 (16 進数) ビット 0-63
1900	1	1		0000 0000 0000 0000
1972	1	1		0081 26D6 0E46 0000
1972	7	1	1	0082 0BA9 811E 2400
1973	1	1	2	0082 F300 AEE2 4800
1974	1	1	3	0084 BDE9 7114 6C00
1975	1	1	4	0086 88D2 3346 9000
1976	1	1	5	0088 53BA F578 B400
1977	1	1	6	008A 1FE5 9520 D800
1978	1	1	7	008B EACE 5752 FC00
1979	1	1	8	008D B5B7 1985 2000
1980	1	1	9	008F 809F DBB7 4400
1981	7	1	10	0092 305C 0FCD 6800
1982	7	1	11	0093 FB44 D1FF 8C00
1983	7	1	12	0095 C62D 9431 B000
1985	7	1	13	0099 5D40 F517 D400
1988	1	1	14	009D DA69 A557 F800
1990	1	1	15	00A1 717D 063E 1C00
1991	1	1	16	00A3 3C65 C870 4000
1992	7	1	17	00A5 EC21 FC86 6400
1993	7	1	18	00A7 B70A BEB8 8800
1994	7	1	19	00A9 81F3 80EA AC00
1996	1	1	20	00AC 3433 6FEC D000
1997	7	1	21	00AE E3EF A402 F400
1999	1	1	22	00B1 962F 9305 1800

5. TOD クロックのビット位置 63 が使用されている場合、そのビットが進められる時間間隔は、 2^{-12} マイクロ秒、つまり約 244 ピコ秒です。この値をクロック単位と呼びます。

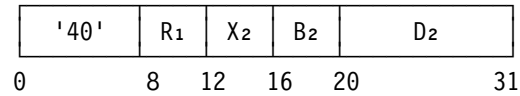
次の表は、各種の時間間隔を、16 進数表記のクロック単位数で示しています。この表が示す値は、STORE CLOCK EXTENDED のオペランドのビット位置 0-71 に格納される値です。オペランドのビット 71 が 1 クロック単位を表しています。

時間間隔	クロック単位 (16 進数) ビット 0-71
1 マイクロ秒	0010 00
1 ミリ秒	3E80 00
1 秒	00F4 2400 00
1 分	3938 7000 00
1 時間	000D 693A 4000 00
1 日	0141 DD76 0000 00
365 日	0001 CAE8 C13E 0000 00
366 日	0001 CC2A 9EB4 0000 00
1,461 日 *	0007 2CE4 E26E 0000 00

* うるう年を含む 4 年間の日数。
1900 年はうるう年ではないという点に注意してください。したがって、1900 年に始まる 4 年間の日数は 1,460 日です。

STORE HALFWORD

STH $R_1, D_2(X_2, B_2)$ [RX]



汎用レジスター R₁ のビット 48-63 が、無変更のまま第 2 オペランド位置に入れます。第 2 オペランドの長さは 2 バイトです。

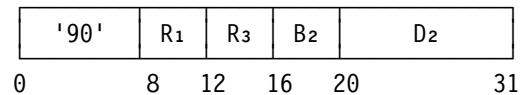
条件コード: コードは変更されません。

プログラム例外:

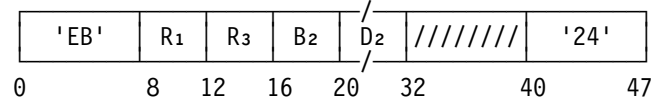
- アクセス例外 (格納: オペランド 2)

STORE MULTIPLE

STM $R_1, R_3, D_2(B_2)$ [RS]



STMG $R_1, R_3, D_2(B_2)$ [RSE]



汎用レジスター R₁ から始まり、汎用レジスター R₃ で終わる一連の汎用レジスターの内容が、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置に入れます。

STORE MULTIPLE (STM) の場合は、該当の汎用レジスターのビット位置 32-63 の内容が、第 2 オペランド・アドレスから始まる連続した 4 バイトのフィールドに格納されます。STORE MULTIPLE (STMG) の場合は、該当の該当の汎用レジスターのビット位置 0-63 の内容が、第 2 オペランド・アドレスから始まる連続した 8 バイトのフィールドに格納されます。

これらの汎用レジスターは、汎用レジスター R₁ から始まり汎用レジスター R₃ まで (このレジスターも含む)、レジスター番号の昇順に従って格納されます。汎用レジスター 0 は、汎用レジスター 15 の後に続きます。

条件コード: コードは変更されません。

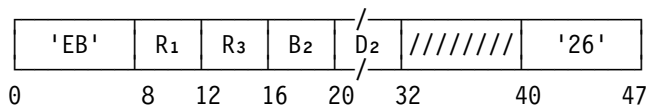
プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意: STORE MULTIPLE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。

STORE MULTIPLE HIGH

STMH R₁,R₃,D₂(B₂) [RSE]



汎用レジスタ R₁ から始まり、汎用レジスタ R₃ で終わる一連の汎用レジスタの高位半分、つまりビット位置 0-31 の内容が、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置に格納されます。つまり、ビット位置 0-31 の内容が、第 2 オペランド・アドレスから始まる連続する 4 バイトのフィールドに格納されます。これらのレジスタのビット 32-63 は無視されます。

これらの汎用レジスタは、汎用レジスタ R₁ から始まり汎用レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従って格納されます。汎用レジスタ 0 は、汎用レジスタ 15 の後に続きます。

条件コード: コードは変更されません。

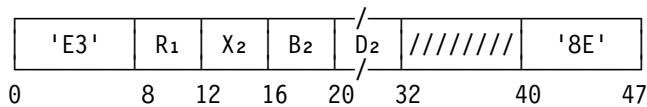
プログラム例外:

- アクセス例外 (格納: オペランド 2)

プログラミング上の注意: R₁ および R₃ により指定されるレジスタ番号のすべての組み合わせが有効です。レジスタ番号が等しい場合は、4 バイトのみが伝送されます。R₃ で指定された番号が R₁ で指定された番号より小さい場合は、レジスタ番号は 15 から 0 に循環します。

STORE PAIR TO QUADWORD

STPQ R₁,D₂(X₂,B₂) [RXE]



クワッドワードの第 1 オペランドが、第 2 オペランド位置に格納されます。他の CPU からは、第 2 オペラ

ド位置への格納はクワッドワード単位で行われているように見えます。第 1 オペランドの左側のダブルワードは汎用レジスタ R₁ の中にあり、右側のダブルワードは汎用レジスタ R₁ + 1 の中にあります。

R₁ フィールドは、汎用レジスタの奇偶ペアを指定し、偶数番号のレジスタを指定している必要があります。第 2 オペランドは、クワッドワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

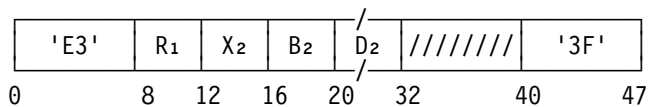
プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 指定例外

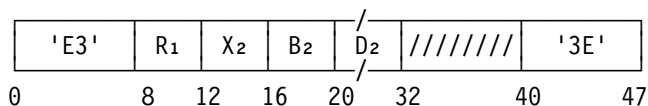
プログラミング上の注意: STORE MULTIPLE (STM または STMG) 命令では、必ずしもクワッドワード単位のアクセスが行われるわけではありません。

STORE REVERSED

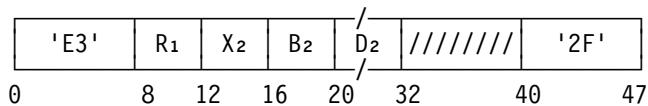
STRVH R₁,D₂(X₂,B₂) [RXE]



STRV R₁,D₂(X₂,B₂) [RXE]



STRVG R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドが、左から右へのバイト順序を反転させて、第 2 オペランド位置に入れられます。

STORE REVERSED (STRVH) の場合は、第 1 オペランドは、汎用レジスタ R₁ のビット位置 48-63 を占める 2 バイトです。STORE REVERSED (STRV) の場合は、第 1 オペランドは、汎用レジスタ R₁ のビット位置 32-63 を占める 4 バイトです。STORE REVERSED (STRVG) の場合は、第 1 オペランドは、

汎用レジスター R₁ のビット位置 0-63 を占める 8 バイトです。

条件コード: コードは変更されません。

プログラム例外:

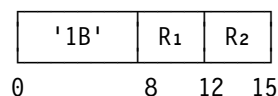
- アクセス例外 (格納: オペランド 2)

プログラミング上の注意:

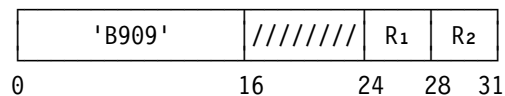
1. この命令は、2 バイト、4 バイト、または 8 バイトを「リトル・エンディアン」フォーマットから「ビッグ・エンディアン」フォーマットへ変換するため、または逆の変換をするために使用できます。ビッグ・エンディアン・フォーマットでは、左から右へのバイト順序は、最上位から最下位への順になります。リトル・エンディアン・フォーマットでは、バイトは最下位から最上位への順に配置されます。例えば、ビッグ・エンディアン・フォーマットのバイト ABCD は、リトル・エンディアン・フォーマットでは DCBA になります。
2. STORE REVERSED の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

SUBTRACT

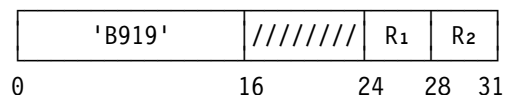
SR R₁,R₂ [RR]



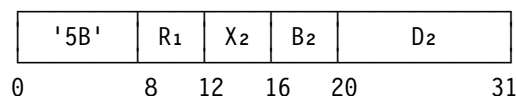
SGR R₁,R₂ [RRE]



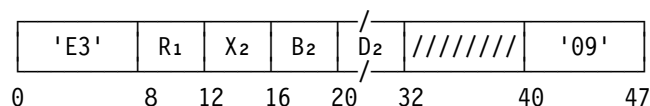
SGFR R₁,R₂ [RRE]



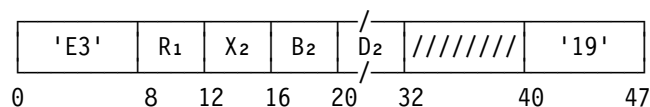
S R₁,D₂(X₂,B₂) [RX]



SG R₁,D₂(X₂,B₂) [RXE]



SGF R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドから第 2 オペランドが減算され、差が第 1 オペランド位置に入れられます。SUBTRACT (SR, S) では、オペランドおよび差は 32 ビットの符号付き 2 進整数として扱われます。SUBTRACT (SGR, SG) では、オペランドおよび差は 64 ビットの符号付き 2 進整数として扱われます。SUBTRACT (SGFR, SGF) では、第 2 オペランドは 32 ビットの符号付き 2 進整数として扱われ、第 1 オペランドおよび差は 64 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がり許し、符号ビット位置からの繰り上がりは無視する方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

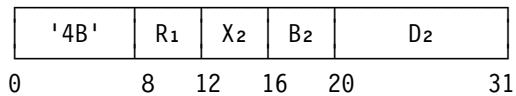
- アクセス例外 (取り出し: S, SG, および SGF のオペランド 2 のみ)
- 固定小数点オーバーフロー例外

プログラミング上の注意:

1. SR および SGR では、R₁ と R₂ が同じレジスターを指定している場合は、この減算はそのレジスターの内容を消去することと同じです。
2. 最大の負の数値から同じ数値を減算すると、結果は 0 になり、オーバーフローは起きません。

SUBTRACT HALFWORD

SH R₁,D₂(X₂,B₂) [RX]



第 1 オペランドから第 2 オペランドが減算され、差が第 1 オペランド位置に入られます。第 2 オペランドは、長さが 2 バイトで、16 ビットの符号付き 2 進整数として扱われます。第 1 オペランドおよび差は、32 ビットの符号付き 2 進整数として扱われます。

オーバーフローが起きると、符号ビット位置への繰り上がりを許し、符号ビット位置からの繰り上がりは無視する方法で結果が得られ、条件コード 3 が設定されます。固定小数点オーバーフロー・マスクが 1 のときは、固定小数点オーバーフローに関するプログラム割り込みが起きます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

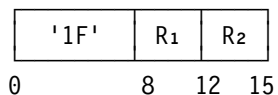
プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 固定小数点オーバーフロー例外

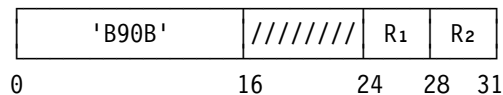
プログラミング上の注意: SUBTRACT HALFWORD IMMEDIATE 命令は提供されていませんが、I₂ フィールドに負の値を指定した ADD HALFWORD IMMEDIATE 命令を使用すれば、それと同じ機能が得られます。

SUBTRACT LOGICAL

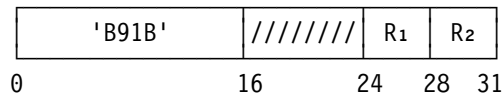
SLR R₁,R₂ [RR]



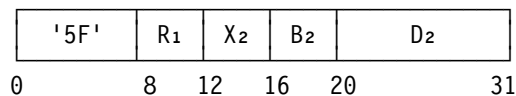
SLGR R₁,R₂ [RRE]



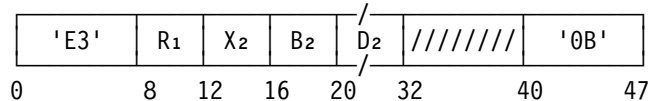
SLGFR R₁,R₂ [RRE]



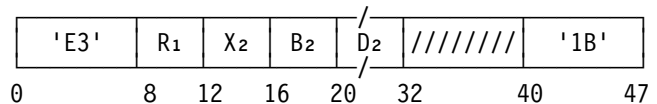
SL R₁,D₂(X₂,B₂) [RX]



SLG R₁,D₂(X₂,B₂) [RXE]



SLGF R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドから第 2 オペランドが減算され、差が第 1 オペランド位置に入られます。SUBTRACT LOGICAL (SLR, SL) では、オペランドおよび差は 32 ビットの符号なし 2 進整数として扱われます。SUBTRACT LOGICAL (SLGR, SLG) では、オペランドおよび差は 64 ビットの符号なし 2 進整数として扱われます。SUBTRACT LOGICAL (SLGFR, SLGF) では、第 2 オペランドは 32 ビットの符号なし 2 進整数として扱われ、第 1 オペランドおよび差は 64 ビットの符号なし 2 進整数として扱われます。

結果の条件コード:

- 0 --
- 1 結果は 0 でなく、繰り下がりがありません。
- 2 結果は 0 で、繰り下がりはありません。
- 3 結果は 0 でなく、繰り下がりはありません。

プログラム例外:

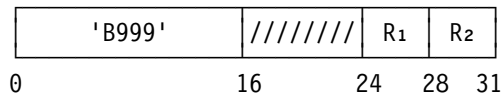
- アクセス例外 (取り出し: SL, SLG、および SLGF のオペランド 2 のみ)

プログラミング上の注意:

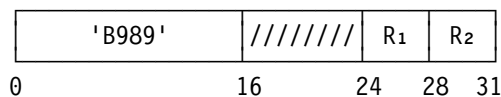
1. 論理減算は、第 2 オペランドの 1 の補数と値 1 とを、第 1 オペランドに加算することによって行われます。第 2 オペランドの 2 の補数の代わりに、1 の補数と値 1 を使用するため、第 2 オペランドが 0 のときは繰り上がりが生じます。
2. SUBTRACT LOGICAL と SUBTRACT の間には、条件コードの意味の違いと、オーバーフローについての割り込みが生じないという違いがあるだけです。
3. 差が 0 のときは、常に、SLGR、SLGFR、SLG、および SLGF ではビット位置 0、そして SLR および SL ではビット位置 32 からの繰り上がりが生じ、したがって繰り下がりはありません。
4. SUBTRACT LOGICAL についての条件コード設定は、以下のように、繰り上がりの有無を示すものと解釈することもできます。
 - 1 結果は 0 でなく、繰り上がりはありません。
 - 2 結果は 0 で、繰り上がりがあります。
 - 3 結果は 0 でなく、繰り上がりがあります。

SUBTRACT LOGICAL WITH BORROW

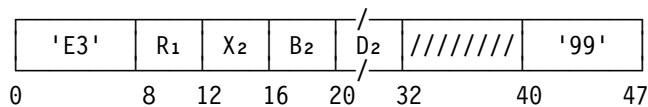
SLBR R₁,R₂ [RRE]



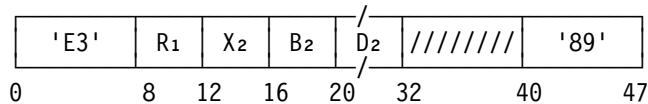
SLBGR R₁,R₂ [RRE]



SLB R₁,D₂(X₂,B₂) [RXE]



SLBG R₁,D₂(X₂,B₂) [RXE]



第 1 オペランドから第 2 オペランドと繰り下がりが減算され、差が第 1 オペランド位置に入られます。

SUBTRACT LOGICAL WITH BORROW (SLBR、SLB) では、オペランド、繰り下がり、および差は 32 ビットの符号なし 2 進整数として扱われます。SUBTRACT LOGICAL WITH BORROW (SLBGR、SLBG) では、これらは 64 ビットの符号なし 2 進整数として扱われます。

結果の条件コード:

- 0 結果は 0 で、繰り下がりがありません。
- 1 結果は 0 でなく、繰り下がりがありません。
- 2 結果は 0 で、繰り下がりはありません。
- 3 結果は 0 でなく、繰り下がりはありません。

プログラム例外:

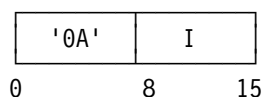
- アクセス例外 (取り出し: SLB および SLBG のオペランド 2 のみ)

プログラミング上の注意:

1. 繰り下がりがあれば、現 PSW のビット 18 に値 0 が入ります。ビット 18 は、PSW の 2 ビット条件コードの左端のビットです。ビット 18 が 0 に設定されるのは、SUBTRACT LOGICAL または SUBTRACT LOGICAL WITH BORROW 命令の実行によって、32 ビットまたは 64 ビットの結果の左端ビット位置への繰り下がりが生じる場合です。
2. 繰り下がりを伴う論理減算は、第 2 オペランドの 1 の補数と現 PSW のビット 18 を、第 1 オペランドに加算することによって行われます。したがって、ビット 18 が 1 (繰り下がりが無いことを示す) であれば、この加算は SUBTRACT LOGICAL の場合と同じです。
3. SUBTRACT LOGICAL WITH BORROW (SLBR、SLB) では、条件コード 0 が設定されるのは、PSW のビット 18 が繰り下がり無しを示しているときに、0 から、最大の 32 ビット符号なし 2 進整数 (2³²-1) が減算された場合です。SUBTRACT LOGICAL WITH BORROW (SLBGR、SLBG) で条件コード 0 が設定されるのは、PSW のビット 18 が繰り下がり無しを示しているときに、0 から最大の 64 ビット符号なし 2 進整数 (2⁶⁴-1) が減算された場合です。
4. モデルによっては、SUBTRACT および SUBTRACT LOGICAL WITH BORROW よりパフォーマンスがよくなる場合があります。

SUPERVISOR CALL

SVC I [RR]



この命令は、監視プログラム呼び出し割り込みを引き起こします。この命令の I フィールドは、割り込みコードの右端のバイトを示します。

この命令のビット 8-15 の左側に 8 個の 0 を付加したものが、割り込みの過程で格納される監視プログラム割り込みコードの中に入れられます。6-44ページの『監視プログラム呼び出し割り込み』を参照してください。

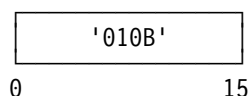
逐次化およびチェックポイント同期化機能が実行されず。

条件コード: コードは変更されず、旧 PSW の一部として保管されます。監視プログラム呼び出し割り込みの一部として、新しい条件コードがロードされます。

プログラム例外: なし。

TEST ADDRESSING MODE

TAM [E]



拡張アドレッシング・モード・ビットおよび基本アドレッシング・モード・ビット (それぞれ現 PSW のビット 31 および 32) がテストされ、結果が条件コードで示されます。

結果の条件コード:

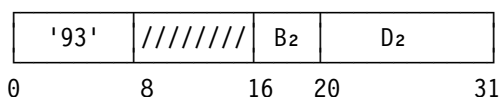
- 0 PSW のビット 31 および 32 が 0 (24 ビット・アドレッシング・モードを示す)。
- 1 PSW のビット 31 が 0 でビット 32 が 1 (31 ビット・アドレッシング・モードを示す)。
- 2 --
- 3 PSW ビット 31 および 32 が 1 (64 ビット・アドレッシング・モードを示す)。

プログラム例外: なし。

プログラミング上の注意: PSW のビット 31 が 1 でビット 32 が 0 のときは、早期の PSW 指定例外が認識されます。

TEST AND SET

TS D₂(B₂) [S]



第 2 オペランド・アドレスにあるバイトの左端ビット (ビット位置 0) に基づいて条件コードが設定され、そのバイトがすべて 0 に設定されます。

この命令のビット 8-15 は無視されます。

記憶域内の該当のバイトは、ビット 0 のテストのために取り出されるときに、すべて 1 に設定されます。この更新は、他の CPU からはインターロック更新参照のように見えます。

バイトが取り出される前と、すべて 1 が格納された後で、逐次化機能が実行されます。

結果の条件コード:

- 0 左端のビットが 0。
- 1 左端のビットが 1。
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (取り出しおよび格納: オペランド 2)

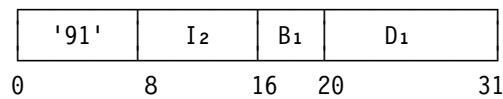
プログラミング上の注意:

1. TEST AND SET は、異なる CPU で実行されている複数のプログラムによる共通記憶域の共用を制御するために使用できます。この命令は、主として、システム/360 用にかかれたプログラムとの互換性を確保するために用意されたものです。同じ CPU 上で実行されるプログラム間での共用、および割り込みが可能なプログラムのためには、COMPARE AND SWAP および COMPARE DOUBLE AND SWAP 命令が提供する機能の方が適しています。詳細については、これらの命令の説明と、関連の『プログラミング上の注意』を参照してください。

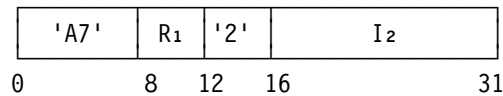
2. TEST AND SET は、チャンネル・プログラムによる記憶域アクセスに対するインターロックは行いません。したがって、チャンネル・プログラムが格納先として使用する可能性のある記憶位置を更新する目的には、この命令は使用しないでください。これを行うと、そのチャンネル・プログラムのデータが失われることがあります。

TEST UNDER MASK (TEST UNDER MASK HIGH, TEST UNDER MASK LOW)

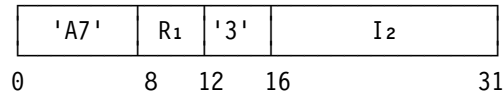
TM $D_1(B_1), I_2$ [SI]



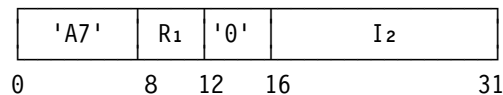
TMHH R_1, I_2 [RI]



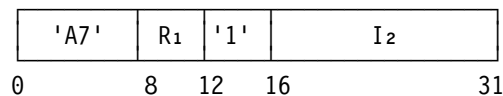
TMHL R_1, I_2 [RI]



TMLH または TMH R_1, I_2 [RI]



TMLL または TML R_1, I_2 [RI]



マスクに基づいて第 1 オペランドからビットが選択され、その結果が条件コードで示されます。

TEST UNDER MASK は、ESA/390 の TEST UNDER MASK HIGH (TMH) および TEST UNDER MASK LOW (TML) の新しい名前であり、それぞれの新しいニックネームは、TMLH および TMLL です。

TEST UNDER MASK (TM) では、即値データのバイト I_2 が 8 ビットのマスクとして使用されます。マスクのビットは、第 1 オペランド・アドレスで指定されている記憶域バイトのビットと 1 対 1 で対応しています。

あるマスク・ビットの値が 1 であれば、それに対応する記憶域ビットがテストされます。マスク・ビットが 0 であれば、対応する記憶域ビットは無視されます。このようにして選択されたすべての記憶域ビットが 0 である場合は、条件コード 0 が設定されます。マスク・ビットがすべて 0 のときも、条件コード 0 が設定されます。選択されたビットがすべて 1 のときは、条件コード 3 が設定されます。その他の場合は、条件コード 1 が設定されます。

マスクがすべて 0 であっても、記憶域オペランドに関するアクセス例外が 1 バイトについて認識されます。

TEST UNDER MASK (TMHH, TMHL, TMLH, TMLL) では、 I_2 フィールドの内容は 16 ビットのマスクとして使用されます。各命令について、マスクのビットは、以下のように、第 1 オペランドの 16 ビットと 1 対 1 で対応しています。

命令	テストされるビット
TMHH	0-15
TMHL	16-31
TMLH (または TMH)	32-47
TMLL (または TML)	48-63

あるマスク・ビットの値が 1 であれば、それに対応する第 1 オペランド・ビットがテストされます。マスク・ビットが 0 のときは、対応する第 1 オペランド・ビットは無視されます。このようにして選択されたすべての第 1 オペランド・ビットが 0 である場合は、条件コード 0 が設定されます。マスク・ビットがすべて 0 のときも、条件コード 0 が設定されます。選択されたビットが 0 と 1 の混合である場合は、選択された左端のビットが 0 であれば条件コード 1 が設定され、選択された左端のビットが 1 であれば条件コード 2 が設定されます。選択されたビットがすべて 1 のときは、条件コード 3 が設定されます。

結果の条件コード:

- 0 選択されたビットがすべて 0 か、マスク・ビットがすべて 0 です。
- 1 選択されたビットは 0 と 1 の混合です (TM のみ)。
- 1 選択されたビットは 0 と 1 の混合で、左端のビットは 0 です (TMHH, TMHL, TMLH, TMLL)。
- 2 -- (TM のみ)
- 2 選択されたビットは 0 と 1 の混合で、左端のビットは 1 です (TMHH, TMHL, TMLH, TMLL)。
- 3 選択されたビットはすべて 1 です。

プログラム例外:

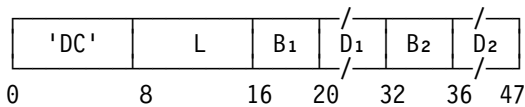
- アクセス例外 (取り出し: TM のオペランド 1)

プログラミング上の注意:

1. TEST UNDER MASK (TM) 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. TMHH, TMHL, TMLH, または TMLL のマスクにより 2 ビットが選択された場合は、その選択された 2 ビットが実際に条件コードにロードされます。

TRANSLATE

TR $D_1(L, B_1), D_2(B_2)$ [SS]



第 1 オペランドのバイトが、第 2 オペランド・アドレスが指すリストを参照するための 8 ビットの引き数として使用されます。リストから選択された各関数バイトが、第 1 オペランド内の対応する引き数と置き換わります。

L フィールドは、第 1 オペランドの長さのみを指定します。

第 1 オペランドのバイトは、変換のために、左から右への順に 1 つずつ選択されます。各引き数バイトが、第 2 オペランドの初期アドレスに加算されます。この加算は、アドレス演算の規則に従って行われ、引き数バイトは 8 ビットの符号なし 2 進整数として扱われ、左側が 0 で拡張されます。加算の和が関数バイトのアドレスとして使用され、元の引き数バイトと置き換わります。

操作は、第 1 オペランド・フィールドの終わりに達するまで続けられます。オーバーラップが生じない限り、リストは変更されません。

両オペランドがオーバーラップしているときは、各結果バイトが、対応する関数バイトの取り出しの直後に格納されたものとして、結果が求められます。

アクセス例外は、第 2 オペランド内の実際に必要なバイトについてのみ認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)

プログラミング上の注意:

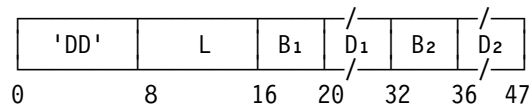
1. TRANSLATE 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. TRANSLATE は、データを、1 つのコードから別のコードに変換するために使用できます。
3. この命令はデータを再配置するためにも使用できます。そのためには、宛先領域に特定のパターンを入れ、TRANSLATE の第 1 オペランドにそのパターンを指定し、再配置するデータを第 2 オペランドに指定します。パターンの各バイトには、この位置を宛先とするバイトを指定する 8 ビットの数値が含まれています。命令が実行されると、パターンに従って、第 2 オペランドのバイトが所要の順序で選択されます。
4. 個々の 8 ビットの引き数バイトが第 2 オペランドの初期アドレスに加算されて、それぞれ 1 つの関数バイトのアドレスが求められるので、リストの長さは最大 256 バイトになります。8 ビットの引き数値がすべて発生するのではないことが分かっている場合は、リストのサイズを削減することができます。
5. DAT が オンのときに、TRANSLATE の第 2 オペランド・アドレスが 4K バイト境界の左側 256 バイトの位置より小さい位置を指定している場合は、パフォーマンスが著しく低下することがあります。これは、マシンが、第 2 オペランドが実際にその境界を超えることになるかどうかを判断するために、命令の試行実行を行うことがあるためです。
6. 第 1 オペランド・フィールド内の特定の 1 バイトへの取り出しアクセスとそれに続く格納アクセスは、必ずしも連続して発生するわけではありません。このため、他の CPU またはチャンネル・プログ

ラムが更新する可能性のある記憶位置をこの命令を使用して更新するのは、安全と言えない場合があります。このような場合の影響の例を、付録A、『数の表現と命令の使用例』の『マルチプログラミングとマルチプロセッシングの例』でOR (OI) について示してあります。

7. TRANSLATE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

TRANSLATE AND TEST

TRT $D_1(L, B_1), D_2(B_2)$ [SS]



第1オペランドのバイトが、第2オペランド・アドレスが指すリストから関数バイトを参照するための8バイトの引き数として使用されます。0でない最初の関数バイトが汎用レジスタ2に挿入され、それに対応する引き数のアドレスが汎用レジスタ1に挿入されます。

L フィールドは、第1オペランドの長さのみを指定します。

第1オペランドのバイトは、変換のために、左から右への順に1つずつ選択されます。第1オペランドは、無変更のまま記憶域内に残されます。関数バイトのアドレスの計算は、TRANSLATE 命令の場合と同じ方法で行われます。リストから検索された関数バイトについて、値が0かどうかが検査されます。

関数バイトが0のときは、第1オペランド内の次のバイトに移って操作が進められます。第1オペランド・フィールドの終わりに達しても、0でない関数バイトが見つからないときは、条件コードを0に設定して操作が完了します。汎用レジスタ1および2の内容は変更されません。

0でない関数バイトが見つかったら、汎用レジスタ2にその関数バイトが挿入され、対応する引き数のアドレスが汎用レジスタ1に挿入されて、操作は完了します。このアドレスは、最後に変換された引き数バイトを指しています。この関数バイトは汎用レジスタ2のビット56-63と置き換わり、ビット0-55は変更されません。

24ビット・アドレッシング・モードでは、汎用レジスタ1のビット40-63がアドレスで置き換えられ、ビット0-39は変更されません。31ビット・アドレッシング・モードでは、汎用レジスタ1のビット33-63がアドレスで置き換えられ、ビット32は0に設定され、ビット0-31は変更されません。64アドレッシング・モードでは、汎用レジスタ1のビット0-63がアドレスで置き換えられます。

関数バイトが0以外であるときは、該当の引き数バイトが第1オペランドの右端バイトかどうかに応じて、条件コード1または2が設定されます。変換すべき引き数バイトが1つ以上残っている場合は、条件コード1が設定されます。残っている引き数バイトがない場合は、条件コード2が設定されます。

アクセス・レジスタ1の内容は、常に無変更のままです。

アクセス例外は、第2オペランド内の実際に必要なバイトについてのみ認識されます。第1オペランド内のバイトのうち、0でない関数バイトが得られた最初のバイトより右にあるバイトについては、アクセス例外は認識されません。

結果の条件コード:

- 0 すべての関数バイトが0です。
- 1 0でない関数バイトがあり、まだ第1オペランド・フィールドの終わりに達していません。
- 2 0でない関数バイトがあり、まだ第1オペランド・フィールドの終わりに達しています。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド1および2)

プログラミング上の注意:

1. TRANSLATE AND TEST 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. TRANSLATE AND TEST は、第1オペランドをスキャンして、特殊な意味を持つ文字を見つけるために使用できます。第2オペランド(つまりリスト)では、スキップすべき文字については全桁0の関数バイトが設定され、検出すべき文字については0以外の関数バイトが設定されます。

TRANSLATE EXTENDED

TRE R₁,R₂ [RRE]

'B2A5'		////////	R ₁	R ₂
0	16	24	28	31

第 1 オペランドのバイトが、汎用レジスタ 0 中のテスト・バイトと比較されます。比較結果が一致しなかった場合は、第 1 オペランド内のバイトは、第 2 オペランド・アドレスで指定される 256 バイトの変換テーブルを参照するための 8 ビットの引き数として使用されます。第 2 オペランドから選択された各関数バイトは、第 1 オペランド内の対応する引き数と置き換わります。この操作は、テスト・バイトに等しい第 1 オペランド・バイトが検出されるか、第 1 オペランドの終わりに達するか、または、CPU で決定されたバイト数の処理が完了するか、そのいずれかの条件が最初に発生するまで続けられます。結果は条件コードで示されます。

R₁ フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定している必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ R₁ および R₂ の内容によって指定されます。24 ビットまたは 31 ビット・アドレッシング・モードでは、第 1 オペランド位置のバイト数は、汎用レジスタ R₁ + 1 のビット位置 32-63 の内容により指定され、この内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 1 オペランド位置のバイト数は、汎用レジスタ R₁ + 1 のすべての内容により指定され、この内容は 64 ビットの符号なし 2 進整数として扱われます。

汎用レジスタ R₁ および R₂ 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタのビット位置 33-63 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、ビット位置 0-63 の内容がアドレスを形成します。

テスト・バイトは、汎用レジスタ 0 のビット位置 56-63 で、このレジスタのビット位置 0-55 の内容は無視されます。

7-149ページの図7-23 は、上記で挙げたレジスタの内容を示しています。

第 1 オペランドのバイトは、変換のために、左から右への順に 1 つずつ選択されます。各引き数バイトは、まず汎用レジスタ 0 中のテスト・バイトと比較されます。比較結果が等しい場合は、操作は完了します。引き数バイトがテスト・バイトに等しくない場合は、その引き数バイトは第 2 オペランドの初期アドレスに加算されます。この加算は、アドレス演算の規則に従って行われ、引き数バイトは 8 ビットの符号なし 2 進整数として扱われ、左側が 0 で拡張されます。加算の和が関数バイトのアドレスとして使用され、元の引き数バイトと置き換わります。オーバーラップが生じない限り、第 2 オペランドは変更されません。

この操作は、テスト・バイトに等しい第 1 オペランド・バイトが検出されるか、第 1 オペランド位置の終わりに達するか、または、CPU で決定された数の第 1 オペランド・バイトの処理が完了するまで続けられます。

第 1 オペランド位置の終わりに達しても、テスト・バイトに等しいバイトが見つからないときは、条件コード 0 が設定されます。テスト・バイトに等しい第 1 オペランド・バイトが見つかったときは、条件コード 1 が設定されます。CPU で決定されたバイト数の処理が完了したときは、条件コード 3 が設定されます。第 1 オペランド位置の終わりに達したとき、または次に処理すべきバイトがテスト・バイトに等しいときも、条件コード 3 が設定されることがあります。これらの場合は、命令が再度実行されるときに、条件コード 0 または 1 が設定されます。

条件コード 0 で操作が完了した場合は、汎用レジスタ R₁ の内容に汎用レジスタ R₁ + 1 の内容が加算され、汎用レジスタ R₁ + 1 の内容は 0 に設定されます。条件コード 1 で操作が完了した場合は、汎用レジスタ R₁ + 1 の内容から、テスト・バイトに等しい第 1 オペランド・バイトが見つかる前に処理されたバイト数が減算され、汎用レジスタ R₁ にそれと同じ数が加算され、その結果、汎用レジスタ R₁ には等しいバイトのアドレスが入ることになります。条件コード 3 で操作が完了した場合は、汎用レジスタ R₁ + 1 の内容から処理されたバイト数が減算され、汎用レジスタ R₁ にそれと同じ数が加算されるので、命令の再実行は、次に処理すべきバイトから再開されます。24 ビットまたは 31 ビットのアドレッシング・モードで汎用レジスタ

R1 が更新されたとき、そのレジスターのビット 32-39 (24 ビット・モードの場合) またはビット 32 (31 ビット・モードの場合) が、0 に設定されるかまたは無変更のままになることがあります。

24 ビットまたは 31 ビット・アドレッシング・モードでは、汎用レジスター R1 および R1 + 1 のビット位置 0-31 の内容は、常に無変更のままになります。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

R2 レジスターが R1 または R1 + 1 と同じレジスターである場合は、結果は予測不能です。

R1 または R2 が 0 のときは、結果は予測不能です。

第 2 オペランドが第 1 オペランドとオーバーラップしている場合も、結果は予測不能です。

処理された最後のバイトの右側にある第 1 オペランド部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、処理された最後のバイトを 4K バイトを超える位置より後にある記憶位置については、アクセス例外は認識されません。

すべてのバイトが使用されていない場合であっても、第 2 オペランドの 256 バイトすべてについてアクセス例外が認識されることがあります。

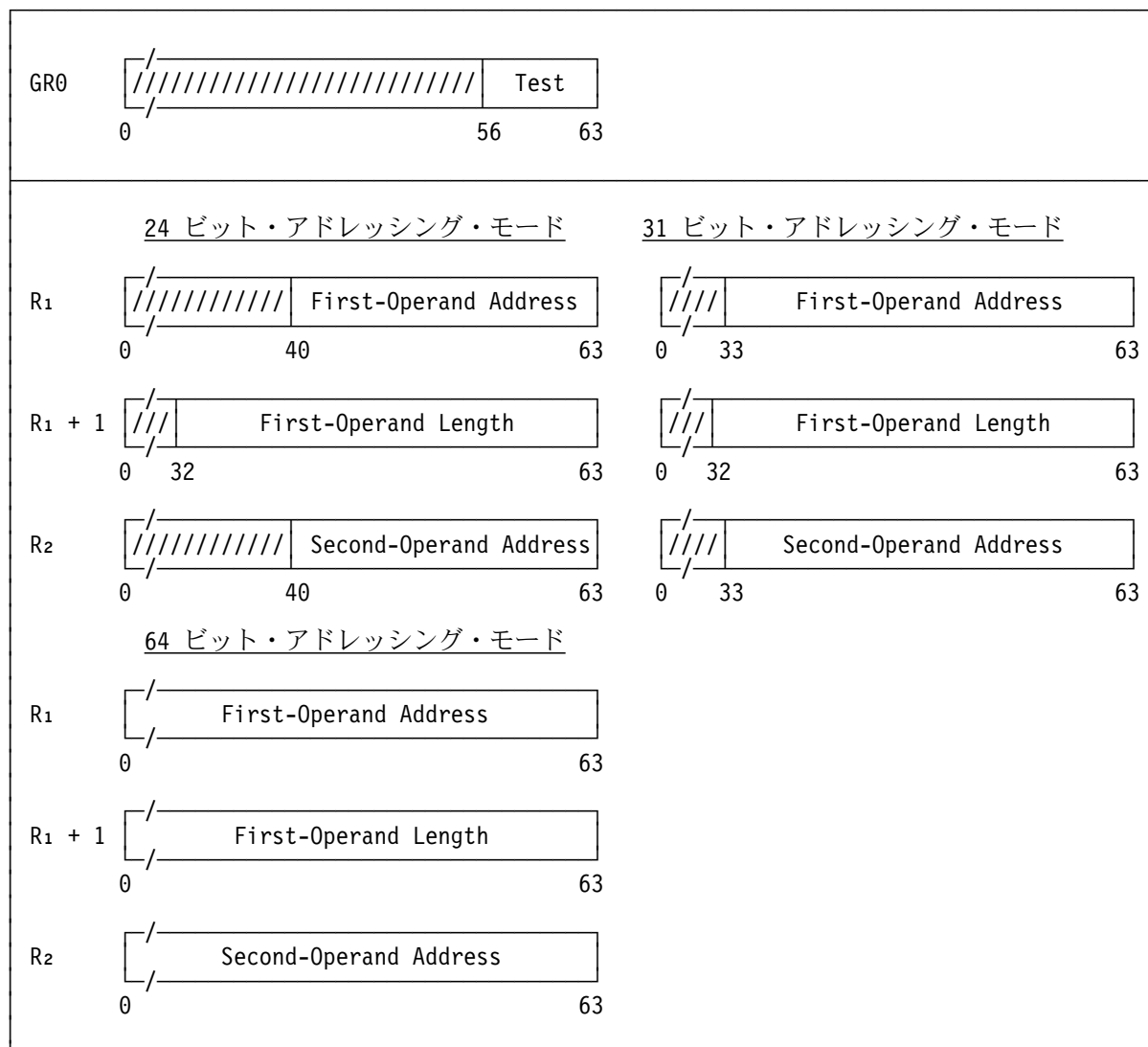


図 7-23. TRANSLATE EXTENDED で使用するレジスターの内容

R1 フィールドが奇数の場合は、アクセス例外は認識されません。第 1 オペランドの長さが 0 の場合、そのオペランドについてはアクセス例外は認識されません。

結果の条件コード:

- 0 第 1 オペランド全体が処理され、テスト・バイトに等しいバイトが見つかりました。
- 1 第 1 オペランドの最初のバイトがテスト・バイトに一致しました。
- 2 --
- 3 CPU で決定されたバイト数が処理されました。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 指定例外

プログラミング上の注意:

1. 条件コード 3 が設定されたときは、プログラムはブランチに従ってこの命令に戻り、変換処理を継続できます。変換済みのバイト数をプログラムで判別する必要はありません。
2. 以前には、TRANSLATE AND TEST 命令によりエスケープ文字を検出し、続いて TRANSLATE 命令によりそのエスケープ文字より前のバイトを変換するという方法がとられていましたが、その代わりにこの命令を使用することにより、パフォーマンスを高めることができます。
3. TRANSLATE EXTENDED の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

TRANSLATE ONE TO ONE

TR00 R1,R2 [RRE]

'B993'	////////	R1	R2
0	16	24	28 31

TRANSLATE ONE TO TWO

TROT R1,R2 [RRE]

'B992'	////////	R1	R2
0	16	24	28 31

TRANSLATE TWO TO ONE

TRTO R1,R2 [RRE]

'B991'	////////	R1	R2
0	16	24	28 31

TRANSLATE TWO TO TWO

TRTT R1,R2 [RRE]

'B990'	////////	R1	R2
0	16	24	28 31

第 2 オペランドの文字を引き数として使用して、汎用レジスター 1 の中のアドレスが指す変換テーブルから関数文字が選択されます。変換テーブルから選択された各関数文字は、汎用レジスター 0 の中のテスト文字と比較され、一致していなければ、その文字が第 1 オペランド位置に入れます。この操作は、テスト文字に等しい関数文字が選択されるか、第 2 オペランドの終わりに達するか、または CPU で決定された文字数の処理が完了するか、そのいずれかの条件が最初に発生するまで続けます。結果は条件コードで示されます。

オペランドおよびテスト文字の長さは、以下のとおりです。

- TRANSLATE ONE TO ONE の場合は、第 2 オペランド、第 1 オペランド、およびテスト文字は、いずれも 1 バイトです。
- TRANSLATE ONE TO TWO の場合は、第 2 オペランド文字は 1 バイトで、第 1 オペランド文字とテスト文字は 2 バイトです。
- TRANSLATE TWO TO ONE の場合は、第 2 オペランド文字は 2 バイトで、第 1 オペランド文字とテスト文字は 1 バイトです。

- TRANSLATE TWO TO TWO の場合は、第 2 オペランド、第 1 オペランド、およびテスト文字は、いずれも 2 バイトです。

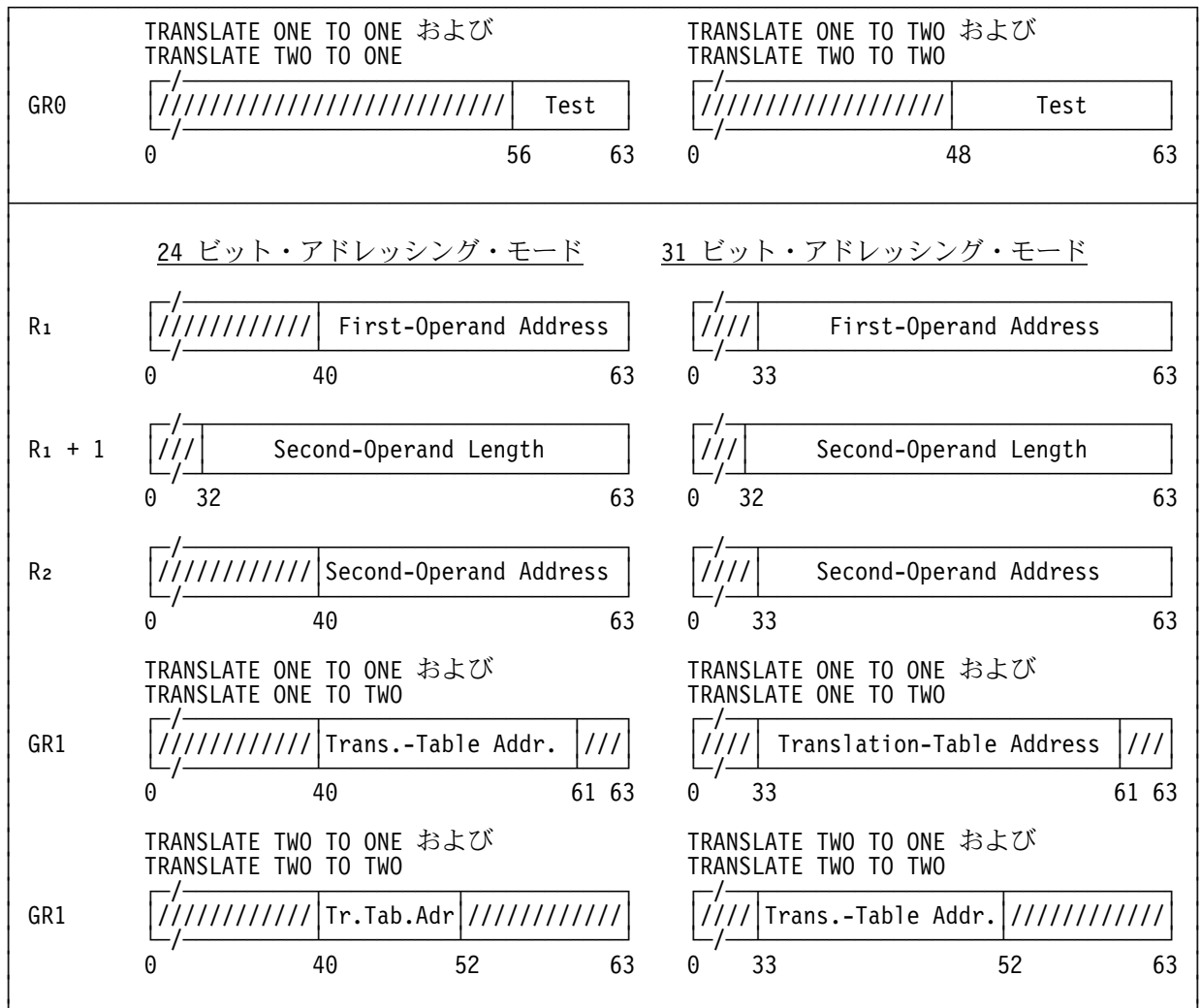


図 7-24 (1/2). TRANSLATE ONE TO ONE、TRANSLATE ONE TO TWO、TRANSLATE TWO TO ONE、および TRANSLATE TWO TO TWO で使用するレジスタの内容

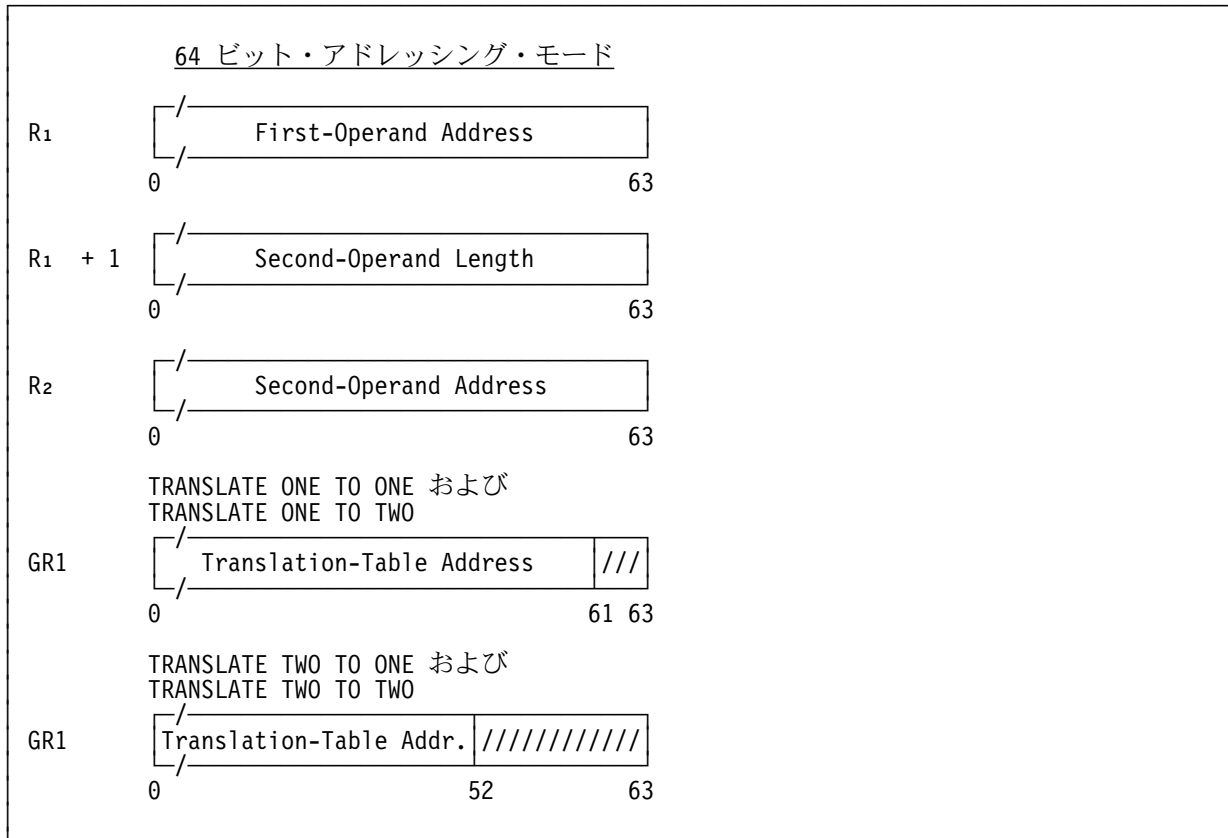


図 7-24 (2/2). TRANSLATE ONE TO ONE、TRANSLATE ONE TO TWO、TRANSLATE TWO TO ONE、およびTRANSLATE TWO TO TWO で使用するレジスタの内容

TRANSLATE ONE TO ONE および TRANSLATE TWO TO ONE では、テスト文字は、汎用レジスタ 0 のビット位置 56-63 に入っています。TRANSLATE ONE TO TWO および TRANSLATE TWO TO TWO では、テスト文字は、汎用レジスタ 0 のビット位置 48-63 に入っています。

R1 フィールドは、汎用レジスタの奇偶ペアを指し、偶数番号のレジスタを指定する必要があります。さもないと、指定例外が認識されます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスタ **R1** および **R2** の内容によって指定されます。24 ビットまたは 31 ビット・アドレッシング・モードでは、第 2 オペランド位置のバイト数は、汎用レジスタ **R1 + 1** のビット位置 32-63 の内容により指定され、この内容は 32 ビットの符号なし 2 進整数として扱われます。64 ビット・アドレッシング・モードでは、第 2 オペランド位置のバイト数は、汎用レジスタ **R1 + 1** のビット位置 0-63 の内容により指定され、この内容は 64 ビットの符号なし 2 進整数

として扱われます。第 1 オペランド位置の長さは、TRANSLATE ONE TO ONE および TRANSLATE TWO TO TWO の場合は第 2 オペランドの長さと同じ、TRANSLATE ONE TO TWO の場合は第 2 オペランドの長さの 2 倍、そして TRANSLATE TWO TO ONE の場合は第 2 オペランドの長さの 2 分の 1 と見なされます。

TRANSLATE TWO TO ONE および TRANSLATE TWO TO TWO では、汎用レジスタ **R1 + 1** の中の長さは、偶数のバイト数でなければなりません。さもないと、指定例外が認識されます。

変換テーブルは、TRANSLATE ONE TO ONE および TRANSLATE ONE TO TWO ではダブルワード境界上にあり、TRANSLATE TWO TO ONE および TRANSLATE TWO TO TWO では 4K バイト境界上にあるものとして扱われます。レジスタのビットのうち、アドレスを形成するために使用されない右端ビット (ダブルワードの場合はビット 61-63 で、4K バイトの場合はビット 52-63) は、無視されます。

汎用レジスタ R₁、R₂、および 1 内のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₁ および R₂ のビット位置 40-63 と、汎用レジスタ 1 のビット位置 40-60 または 40-51 の内容がアドレスを形成し、ビット位置 0-39 の内容は無視されます。31 ビット・アドレッシング・モードでは、レジスタ R₁ および R₂ のビット位置 33-63 と、レジスタ 1 のビット位置 33-60 または 33-51 の内容がアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、レジスタ R₁ および R₂ のビット位置 0-63 と、レジスタ 1 のビット位置 0-60 または 0-51 の内容がアドレスを形成します。

7-151ページの図7-24 は、上記で説明したレジスタの内容を示しています。

アクセス・レジスタ・モードでは、アクセス・レジスタ R₁、R₂、および 1 の内容が、それぞれ、第 1 オペランド、第 2 オペランド、および変換テーブルをアクセスするために使用されます。

汎用レジスタ 1 に含まれるアドレスが指す変換テーブルの長さは、以下のとおりです。

- TRANSLATE ONE TO ONE の場合は、変換テーブル長は 256 バイトで、256 個の関数文字のそれぞれが 1 バイトずつを占めます。
- TRANSLATE ONE TO TWO の場合は、変換テーブル長は 512 バイトで、256 個の関数文字のそれぞれが 2 バイトずつを占めます。
- TRANSLATE TWO TO ONE の場合は、変換テーブル長は 65,536 (64K) バイトで、64K 個の関数文字のそれぞれが 1 バイトずつを占めます。
- TRANSLATE TWO TO TWO の場合は、変換テーブル長は 131,072 (128K) バイトで、128K 個の関数文字のそれぞれが 2 バイトずつを占めます。

第 2 オペランドの文字は、変換のために、左から右への順に 1 つずつ選択されます。各引き数文字が、変換テーブルの初期アドレスに加算されます。この加算はアドレス演算の規則に従って行われ、引き数文字は以下のように扱われます。

- TRANSLATE ONE TO ONE では、引き数文字は、8 ビットの符号なし 2 進整数の左に 56 個の 0 を付加したものととして扱われます。

- TRANSLATE ONE TO TWO では、引き数文字は、8 ビットの符号なし 2 進整数の右側に 1 個の 0、そして左側に 55 個の 0 を付加したものととして扱われます。
- TRANSLATE TWO TO ONE では、引き数文字は、16 ビットの符号なし 2 進整数の左に 48 個の 0 を付加したものととして扱われます。
- TRANSLATE TWO TO TWO では、引き数文字は、16 ビットの符号なし 2 進整数の右側に 1 個の 0、そして左側に 47 個の 0 を付加したものととして扱われます。

変換テーブル・アドレス内の無視される右端ビット (61-63 または 52-63) は、この加算では 0 として扱われます。

加算結果の和は、関数文字のアドレスとして使用されません。

上記の方法で選択された各関数文字は、まず、汎用レジスタ 0 中のテスト文字と比較されます。比較結果が等しい場合は、操作は完了します。関数文字がテスト文字と等しくない場合は、第 1 オペランド内の次の選択可能な文字位置に、その関数文字が入れられます。つまり、最初の関数文字は、第 1 オペランド位置の先頭に入れられ、後続の各関数文字は、それぞれ直前の文字の右側に入れられます。オーバーラップが生じない限り、第 2 オペランドおよび変換テーブルは変更されません。

この操作は、テスト文字に等しい関数文字が選択されるか、第 2 オペランド位置の終わりに達するか、または CPU で決定された数の第 1 オペランド文字の処理が完了するまで続けられます。

テスト文字に等しい関数文字が見つかったときは、条件コード 1 が設定されます。第 2 オペランド位置の終わりに達しても、テスト文字に等しい関数文字が見つからないときは、条件コード 0 が設定されます。CPU で決定される文字数の処理が完了した場合は、条件コード 3 が設定されます。次の文字を処理した結果、テスト文字に等しい関数文字が検出されることになる場合、または第 2 オペランド位置の終わりに達した場合にも、条件コード 3 が設定されることがあります。これらの場合は、命令が再度実行されるときに、それぞれ条件コード 1 または 0 が設定されます。

条件コード 0 で操作が完了した場合は、汎用レジスタ R₂ の内容が汎用レジスタ R₁ + 1 の内容分だけ増加し、汎用レジスタ R₁ の内容が以下のように増加します。

- TRANSLATE ONE TO ONE および TRANSLATE TWO TO TWO では、汎用レジスタ R₂ の場合と同じ。
- TRANSLATE ONE TO TWO では、汎用レジスタ R₂ の場合の 2 倍の量。
- TRANSLATE TWO TO ONE では、汎用レジスタ R₂ の場合の 2 分の 1 の量。

上記の操作の後で、汎用レジスタ R₁ + 1 の内容は 0 に設定されます。

条件コード 1 で操作が完了した場合は、汎用レジスタ R₁ + 1 の内容から、テスト文字に等しい関数文字が見つかる前に処理された第 2 オペランドバイト数が減算され、汎用レジスタ R₂ にそれと同じ数が加算され、その結果、汎用レジスタ R₂ には、テスト文字と等しい関数文字を選択した文字のアドレスが入ることになります。汎用レジスタ R₁ の内容は、上記で条件コード 0 について述べたときと同じ分類に従って、同一、2 倍、または 2 分の 1 の量だけ増加します。

条件コード 3 で操作が完了した場合は、汎用レジスタ R₁ + 1 の内容から処理された第 2 オペランド・バイト数が減算され、汎用レジスタ R₂ にそれと同じ数が加算されるので、命令の再実行では、次に処理すべき文字のアドレスが使用されることとなります。汎用レジスタ R₁ の内容は、上記で条件コード 0 について述べたときと同じ分類に従って、同一、2 倍、または 2 分の 1 の量だけ増加します。

24 ビットまたは 31 ビット・アドレッシング・モードで汎用レジスタ R₁ および R₂ が更新される時、これらのレジスタのビット位置 32-39 のうちアドレスの一部となっていない部分は、0 に設定される場合と、または元の値のまま変更されない場合があります。24 ビットまたは 31 ビット・アドレッシング・モードでは、汎用レジスタ R₁、R₁ + 1、および R₂ のビット位置 0-31 は、常に無変更のままになります。

汎用レジスタ 0 および 1 の内容は変更されません。

条件コード 3 が設定される結果になる処理量は、システム・パフォーマンスの向上という観点から CPU が決定するもので、この量は命令を実行するたびに異なる場合があります。

命令の実行中に CPU の再試行が起きると、条件コード 3 が設定されることがあります。その場合は、第 1 オペ

ランド位置の中で、汎用レジスタ R₁ 内の最終アドレスが指している位置またはその右側の位置には、誤ったデータが格納されている可能性があります。格納されているデータの量は、操作と、CPU 再試行が起きた時点によって異なります。どの場合も、命令が再実行されて同じオペランドの処理が続行されると、正しい格納データを用いて、再度格納が行われます。

R₁ レジスタと R₂ レジスタと同じレジスタである場合、R₁ または R₂ レジスタがレジスタ 0 である場合、または R₂ レジスタがレジスタ 1 である場合は、結果は予測不能です。

第 1 オペランド、第 2 オペランド、および変換テーブルのいずれかが他のいずれかとオーバーラップしている場合は、結果は予測不能です。

第 1 オペランドまたは第 2 オペランドの中の、処理された最後の文字より右にある部分については、アクセス例外が認識される場合とされない場合があります。4K バイトより長いオペランドの場合は、処理された最後の文字を 4K バイト超える位置より後にある記憶位置については、アクセス例外は認識されません。

変換テーブルについては、テーブル内のすべての文字が使用されていない場合であっても、すべての文字についてアクセス例外が認識されることがあります。

R₁ フィールドが奇数の場合は、アクセス例外は認識されません。第 2 オペランドの長さが 0 の場合は、第 1 オペランドまたは第 2 オペランドについてはアクセス例外は認識されず、変換テーブルについては、アクセス例外は認識されることもされないこともあります。

結果の条件コード:

- 0 第 2 オペランド全体が処理されましたが、テスト文字に等しい関数文字は見つかりませんでした。
- 1 結果の関数文字がテスト文字と等しくなるような第 2 オペランド文字が見つかりました。
- 2 --
- 3 CPU で決定された文字数が処理されました。

プログラム例外:

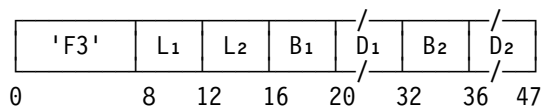
- アクセス例外 (取り出し: オペランド 2 および変換テーブル。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

- これらの命令は、以下の属性を備えているという点で、TRANSLATE EXTENDED 命令と異なっています。
 - 使用する命令に応じて、引き数文字と関数文字のセットには、1 バイト文字が含まれる場合と 2 バイト文字が含まれる場合があります。
 - テスト文字は、引き数文字ではなく結果の関数文字と比較されます。
 - 引き数 (ソース) オペランドと関数 (宛先) オペランドは、異なるオペランドです。
- 条件コード 3 が設定されたときは、プログラムはブランチに従ってこの命令に戻り、変換処理を継続できます。変換済みの文字数をプログラムで判別する必要はありません。
- これらの命令の記憶域オペランド参照は、複数アクセス参照となることがあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

UNPACK

UNPK D₁(L₁,B₁),D₂(L₂,B₂) [SS]



第 2 オペランドのフォーマットがパックからゾーンに変更され、結果が第 1 オペランド位置に入れられます。パック・フォーマットおよびゾーン・フォーマットについては、第 8 章、『10 進数命令』で説明されています。

第 2 オペランドは、パック・フォーマットであるものとして扱われます。第 2 オペランドの数字と符号は、無変更のまま、ゾーン・フォーマットで第 1 オペランド位置に入れられます。最右端の 1 バイトを除くすべてのバイトに、1111 のコードが入るゾーン・ビットがあります。最右端のバイトのゾーンには、第 2 オペランドの符号が入ります。符号および数字が有効なコードかどうかの検査は行われません。

結果は、両方のオペランドを右から左へ処理したものとして求められます。必要があれば、第 2 オペランドは左側が 0 で拡張されたものとして扱われます。第 1 オペランド・フィールドが短すぎて第 2 オペランドのすべての数字を収容できない場合は、第 2 オペランドの左端の残り部分は無視されます。第 2 オペランドの使用されな

かった部分については、アクセス例外が示されることもあり、示されないこともあります。

オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理され、最初のオペランド・バイトを取り出した直後に最初の結果バイトが格納されるものとして、結果が求められます。第 2 オペランドの右端の 1 バイト全体が、最初の結果バイトを形成するために使用されます。フィールドの残りの部分については、第 2 オペランドの単一のバイトから 2 個の結果バイトの情報が取得されます。そして、そのバイトの左端 4 ビットは、次の結果バイト用として使用可能なままになっており、再取り出しは不要なものとして、実行が進められます。したがって、結果は、1 個のオペランド・バイトの取り出しの直後に 2 個の結果バイトが格納されているかのように見えます。

条件コード: コードは変更されません。

プログラム例外:

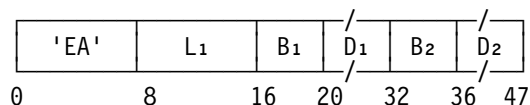
- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)

プログラミング上の注意:

- UNPACK 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
- 不適切なオーバーラップがあると、アンパックするフィールドが破壊されることがあります。オペランドをオーバーラップさせて記憶域スペースを節約するには、第 1 オペランドの右端バイトが、第 2 オペランドの右端バイトより、第 2 オペランドのバイト数から 2 を引いたバイト数だけ右になければなりません。1 バイトまたは 2 バイトのみをアンパックする場合は、両方のオペランドの右端バイトが重なっていても構いません。
- UNPACK の記憶域オペランド参照は、複数アクセス参照となることがあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

UNPACK ASCII

UNPKA D₁(L₁,B₁),D₂(B₂) [SS]



第 2 オペランドのフォーマットがパックから ASCII に変更され、結果が第 1 オペランド位置に入れられます。

パック・フォーマットについては、第8章、『10 進数命令』の説明を参照してください。

第 2 オペランドは、パック・フォーマットであるものとして扱われます。第 2 オペランドの数字は、左側に 2 進数 0011 を付加することにより ASCII 文字に変換され、変換後の ASCII 文字が第 1 オペランド位置に入れます。数字が有効なコードかどうかの検査は行われません。

第 2 オペランドの符号は、第 1 オペランドに転送されずに、妥当性が検査され、その結果に基づいて条件コードが決定されます。符号が 2 進数 1010、1100、1110、または 1111 (正) であれば、条件コード 0 が設定されます。符号が 2 進数 1011 または 1101 (負) であれば、条件コード 1 が設定されます。符号が、正または負を表すコードのいずれにも該当しない場合は、条件コード 3 が設定されます。

変換された最後の数字は、結果フィールドの右端のバイト位置に入れられ、その他の変換された数字は、その最後の数字に隣接し、また相互に隣接する形で、結果フィールドの残りの部分に入れられます。

結果は、両方のオペランドを右から左へ処理したものとして求められます。

第 2 オペランドの長さは 16 バイトです。第 2 オペランドは、31 個の数字と 1 個の符号から成ります。

第 1 オペランドの長さは、L₁ フィールドの内容によって指定されます。第 1 オペランドの長さは 32 バイトを超えてはなりません (L₁ が 31 以下でなければなりません)。さもないと、指定例外が認識されます。

第 1 オペランドが短すぎて第 2 オペランドのすべての数字を収容できない場合は、第 2 オペランドの左端の残り部分は無視されます。第 2 オペランドの使用されなかった部分については、アクセス例外が示されることもあり、示されないこともあります。

第 1 オペランドの長さが 32 バイトの場合は、左端バイトは ASCII の 0 (16 進数 30) に設定されます。

第 1 オペランドと第 2 オペランドがどのような形であれオーバーラップしている場合、結果は予測不能です。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランドには、必ずしも特定の順序で格納されてはいないように見えます。

結果の条件コード:

- 0 符号は正です。
- 1 符号は負です。
- 2 --
- 3 符号は無効です。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

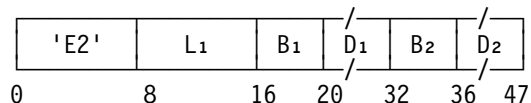
1. 次の例は、この命令を使用して ASCII 数字をアンパックする方法を示しています。

```
ASDIGITS DS CL31
PKDIGITS DS 0PL16
          DC X'1234567890'
          DC X'1234567890'
          DC X'1234567890'
          DC X'1C'
          ...
UNPKA ASDIGITS(31),PKDIGITS
```

2. UNPACK ASCII の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

UNPACK UNICODE

UNPKU D₁(L₁,B₁),D₂(B₂) [SS]



第 2 オペランドのフォーマットがパックから Unicode Basic Latin に変更され、結果が第 1 オペランド位置に入れます。パック・フォーマットについては、第8章、『10 進数命令』の説明を参照してください。

第 2 オペランドは、パック・フォーマットであるものとして扱われます。第 2 オペランドの数字は、左側に 2 進数 000000000011 (16 進数 003) を付加することにより 2 バイトの Unicode 文字に変換され、変換後の Unicode 文字が第 1 オペランド位置に入れます。数字が有効なコードかどうかの検査は行われません。第

2 オペランドの符号は、第 1 オペランドに転送されずに、妥当性が検査され、その結果に基づいて条件コードが決定されます。符号が 2 進数 1010、1100、1110、または 1111 (正) であれば、条件コード 0 が設定されます。符号が 2 進数 1011 または 1101 (負) であれば、条件コード 1 が設定されます。符号が、正または負を表すコードのいずれにも該当しない場合は、条件コード 3 が設定されます。

変換された最後の数字は、結果フィールドの右端の文字位置に入れられ、その他の変換された数字は、その最後の数字に隣接し、また相互に隣接する形で、結果フィールドの残りの部分に入れられます。

結果は、両方のオペランドを右から左へ処理したものとして求められます。

第 2 オペランドの長さは 16 バイトです。第 2 オペランドは、31 個の数字と 1 個の符号から成ります。

第 1 オペランドの長さは、L1 フィールドの内容によって指定されます。第 1 オペランドの長さは、32 文字つまり 64 バイトを超えてはなりません (L1 は 63 以下で、かつ奇数でなければなりません)。さもないと、指定例外が認識されます。

第 1 オペランドが短すぎて第 2 オペランドのすべての数字を収容できない場合は、第 2 オペランドの左端の残り部分は無視されます。第 2 オペランドの使用されなかった部分については、アクセス例外が示されることもあり、示されないこともあります。

第 1 オペランドの長さが 32 文字の場合は、左端の文字は Unicode Basic Latin の 0 (16 進数 0030) に設定されます。

第 1 オペランドと第 2 オペランドがどのような形であれオーバーラップしている場合、結果は予測不能です。

チャンネル・プログラムおよび他の CPU から見た場合、第 1 オペランドには、必ずしも特定の順序で格納されていないように見えます。

結果の条件コード:

- 0 符号は正です。
- 1 符号は負です。
- 2 --
- 3 符号は無効です。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. 次の例は、この命令を使用してヨーロッパ表記の数をアンパックする方法を示しています。

```

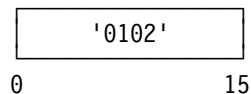
UNDIGITS DS CL62
PKDIGITS DS 0PL16
DC X'1234567890'
DC X'1234567890'
DC X'1234567890'
DC X'1C'
...
UNPKU UNDIGITS(62),PKDIGITS

```

2. UNPACK UNICODE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86 ページの『記憶域オペランドの一貫性』を参照してください。)

UPDATE TREE

UPT [E]



記憶域内にあるツリーのノードが、ツリーの基底に向かうパス上で連続して検査されます。そして、汎用レジスター 0 の内容と、概念上その右側に続いている汎用レジスター 1 の内容が、条件付きでノードの内容で置き換えられて、汎用レジスター 0 に固有の最大論理値が入れられます。ノードおよび汎用レジスター 0 の前半部分には、ソート/マージ・アルゴリズムで使用されるコードワードが含まれています。

汎用レジスター 0 の中のコードワードがノード内のコードワードに等しい場合は、ノード内容が汎用レジスター 2 および 3 に入れられます。

汎用レジスター 4 にはツリーの基底アドレスが入り、汎用レジスター 5 には、最初に検査される親ノードの子であるノードの指標が入っています。

アクセス・レジスター・モードでは、アクセス・レジスター 4 は、ツリーを含むアドレス・スペースを指定します。

この命令は、操作単位と操作単位の間で割り込まれることがあります。命令が割り込まれたときの条件コードは予測不能です。

ノードのサイズ、コードワードのサイズ、および、汎用レジスター 1-5 のビット 0-31 が操作に関与するかどうかは、アドレッシング・モードによって異なります。24 ビットまたは 31 ビット・アドレッシング・モードでは、ノードは 8 バイト、コードワードは 4 バイトで、ビット 0-31 は無視されて無変更のままになります。64 ビット・アドレッシング・モードでは、ノードは 16 バイト、コードワードは 8 バイトで、ビット 0-31 は操作に使用され、変更されることがあります。

24 ビットまたは 31 ビット・アドレッシング・モードの操作

24 ビットまたは 31 ビット・アドレッシング・モードでは、記憶域内にあるツリーのダブルワード・ノードが、ツリーの基底に向かうパス上で連続して検査されます。そして、汎用レジスター 0 のビット位置 32-63 の内容と、概念上その右側に続いている汎用レジスター 1 のビット位置 32-63 の内容が、条件付きでノードの内容で置き換えられて、汎用レジスター 0 のビット位置 32-63 に固有の最大論理値が入れられます。

汎用レジスター 4 のビット位置 32-63 にはツリーの基底アドレスが入り、汎用レジスター 5 のビット位置 32-63 には、最初に検査される親ノードの子であるノードの指標が入っています。基底アドレスは、ツリーのルート・ノードのアドレスより 8 だけ小さいアドレスです。汎用レジスター 4 および 5 のビット位置 32-63 の初期内容は、8 の倍数でなければなりません。さもないと、指定例外が認識されます。

操作単位は、汎用レジスター 5 のビット位置 32-63 の内容を右に論理上 1 桁シフトし、ビット 61 を 0 に設定することにより、開始されます。ただし、操作単位の実行が無効化または抑止された場合は、汎用レジスター 5 のビット 32-63 は変更されません。シフトしてビット 61 を 0 に設定した後で、汎用レジスター 5 のビット 32-63 がすべて 0 である場合は、命令は完了し、条件コード 1 が設定されます。そうでない場合は、操作単位は続行されます。

汎用レジスター 0 のビット 32 がテストされます。汎用レジスター 0 のビット 32 が 1 のときは、命令は完了し、条件コード 3 が設定されます。

汎用レジスター 0 のビット 32 が 0 の場合は、汎用レジスター 4 および 5 のビット 32-63 の和が、通常のオペランド・アドレス生成のための中間値として使用されます。生成されるアドレスは、記憶域内のノードのアドレスです。

汎用レジスター 0 のビット 32-63 が、現在アドレッシングされているノードの最初のワードの内容と論理的に比較されます。レジスター・オペランドの方が小さい場合は、汎用レジスター 0 および 1 のビット位置 32-63 の内容がノードの内容で置き換えられ、操作単位は完了します。レジスター・オペランドの方が大きい場合は、それ以上何もアクションは行われずに、操作単位は完了します。2 つの比較値が等しい場合は、現在アドレッシングされているノードから、汎用レジスター 2 のビット位置 32-63 と、論理上その右側に続く汎用レジスター 3 のビット位置 32-63 がロードされ、命令は完了し、条件コード 0 が設定されます。

ノードの最初のワードが、レジスターのビット位置 32-63 の中の値と同じかまたはそれより小さい場合は、ノードの内容は変更されません。ただし、一部のモデルでは、いったん内容が取り出された後で、その内容がそのまま再び格納されることもあります。

64 ビット・アドレッシング・モードの操作

64 ビット・アドレッシング・モードでは、記憶域内にあるツリーのクワッドワード・ノードが、ツリーの基底に向かうパス上で連続して検査されます。そして、汎用レジスター 1 の内容と、概念上その右側に続いている汎用レジスター 0 の内容が、条件付きでノードの内容で置き換えられて、汎用レジスター 0 に固有の最大論理値が入れられます。

汎用レジスター 4 にはツリーの基底アドレスが入り、汎用レジスター 5 には、最初に検査される親ノードの子であるノードの指標が入っています。基底アドレスは、ツリーのルート・ノードのアドレスより 16 だけ小さいアドレスです。汎用レジスター 4 および 5 の初期内容は、16 の倍数でなければなりません。さもないと、指定例外が認識されます。

操作単位は、汎用レジスター 5 の内容を右に論理上 1 桁シフトし、ビット 60 を 0 に設定することにより、開始されます。ただし、操作単位の実行が無効化または抑止された場合は、汎用レジスター 5 は変更されません。

シフトしてビット 60 を 0 に設定した後で、汎用レジスタ 5 の内容が 0 である場合は、命令は完了し、条件コード 1 が設定されます。そうでない場合は、操作単位は続行されます。

汎用レジスタ 0 のビット 0 がテストされます。汎用レジスタのビット 0 が 0 のときは、命令は完了し、条件コード 3 が設定されます。

汎用レジスタ 0 のビット 0 が 0 の場合は、汎用レジスタ 4 および 5 の内容の和が、通常のオペランド・アドレス生成のための中間値として使用されます。生成されるアドレスは、記憶域内のノードのアドレスです。

汎用レジスタ 0 の内容が、現在アドレッシングされているノードの最初のダブルワードの内容と論理的に比較されます。レジスタ・オペランドの方が小さい場合は、汎用レジスタ 0 および 1 の内容がノードの内容で置き換えられ、操作単位は完了します。レジスタ・オペランドの方が大きい場合は、それ以上何もアクションは行われずに、操作単位は完了します。2 つの比較値が等しい場合は、現在アドレッシングされているノードから汎用レジスタ 2 および 3 がロードされ、命令は完了し、条件コード 0 が設定されます。

ノードの最初のダブルワードが、レジスタの値と同じかまたはそれより小さい場合は、ノードの内容は変更されません。ただし、一部のモデルでは、いったん内容が取り出された後で、その内容がそのまま再び格納されることもあります。

アドレッシング・モードからは独立している指定

アクセス例外は、一度に 1 つのノードについてのみ認識されます。後続のノードについては、前のノードの比較と更新が正常に完了するまでは、アクセス例外、変更ビット・アクション、および PER 記憶域変更は起きません。また、指定例外条件が存在するときも、これらは起きません。

結果の条件コード:

- 0 現在アドレッシングされているノードで比較値が等しい。
- 1 パス上に等しい比較値が見つからないか、または比較が行われませんでした。
- 2 --
- 3 24 ビットまたは 31 ビット・モードでは、汎用レジスタ 5 のビット 32-63 が 0 以外で、汎用レジスタ 0 のビット 32-63 が負です。64 ビット・モードでは、汎用レジスタ 5 が 0 以外で、汎用レジスタ 0 が負です。

プログラム例外:

- アクセス例外 (取り出しおよび格納: ツリーのノード)
- 指定例外

プログラミング上の注意:

1. UPDATE TREE の使用例は、付録A、『数の表現と命令の使用例』の中の『ソート命令』に示されています。
2. 24 ビットまたは 31 ビット・アドレッシング・モードでソートに使用する場合に、等しい比較値が検出されたときは、汎用レジスタ 1 および 3 のビット位置 32-63 の内容は、(ツリーの内容に応じて) 後続の COMPARE AND FORM CODEWORD の実行に使用することができます。同様に、汎用レジスタ 2 のビット位置 32-63 を右に 16 ビット位置だけシフトしたのも使用でき、これにより、部分的に等しいキーの最小限の再比較を行うことができます。64 ビット・アドレッシング・モードについても同じことが言えますが、対象が両方のレジスタのビット位置 0-63 であり、汎用レジスタ 2 のビット位置 0-63 がビット位置 48 個分だけ右にシフトされるという点が異なります。ソートにおけるツリーおよびそれぞれの使用方法については、A-51ページの『ソート命令』を参照してください。
3. 24 ビット・アドレッシング・モードのときは、プログラムでは、汎用レジスタ 5 のビット位置 32-38 には 0 以外の値は入れないようにしてください。ビット位置 32-38 のどこかに 1 があると、ツリーのノードが連続して検査されなくなります。
4. 24 ビットまたは 31 ビット・アドレッシング・モードで、汎用レジスタ 0 のビット 32-63 が負であるか、または 64 ビット・モードでビット 0-63 が負であるときに、ツリーが事前に正しく更新されていれば、汎用レジスタ 0 および 1 が表すノードは、操作単位が続行される場合に選択されることになるノードであるか、またはそのノードに等しいノード (等しいキーを持つ) です。この場合は、操作単位を終了し条件コード 3 を設定する方が、それ以上ツリーの検査と更新を行う必要がないため、適切なノードをどれか選択するより効率的です。
5. 置換レコードが旧勝者に等しいとき、そしてさらに重要なのは、更新パスに、旧勝者と等しいことを示す負のコードワードが含まれているときは (最初のケースは CFC の条件コードにより検出可能なので)、条件コード 3 の設定によりパフォーマンスが向上します。

6. モデルによっては、ノード内のコードワードが汎用レジスタ 0 中のコードワードに等しいかそれより小さいときは、ノードがいったん取り出された後で、再び格納されることがあります。その結果、そのノードを含む記憶位置について、PER 記憶域変更イベントが認識されるか、格納に対する記憶保護例外が認識されるか、または、アクセス例外がないときに変更ビットが 1 に設定されることがあります。記憶域の内容は無変更のままになるので、PER 記憶域変更イベントが認識された場合、変更ビットは 1 になる場合もあり、ならない場合もあります。
7. UPDATE TREE を EXECUTE のターゲットとして使用する場合は、特別な注意が必要です。EXECUTE の項の『プログラミング上の注意』の、割り込み可能命令についての説明を参照してください。
8. 割り込み可能命令に関する詳しいプログラミング上の注意は、5-20ページの『割り込み可能命令』に示されています。
9. UPDATE TREE の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)
10. 7-161ページの図7-25 に、24 ビットまたは 31 ビット・アドレッシング・モードでの UPDATE TREE の操作の要約を示し、7-162ページの図7-26 に、64 ビット・アドレッシング・モードでの操作の要約を示します。

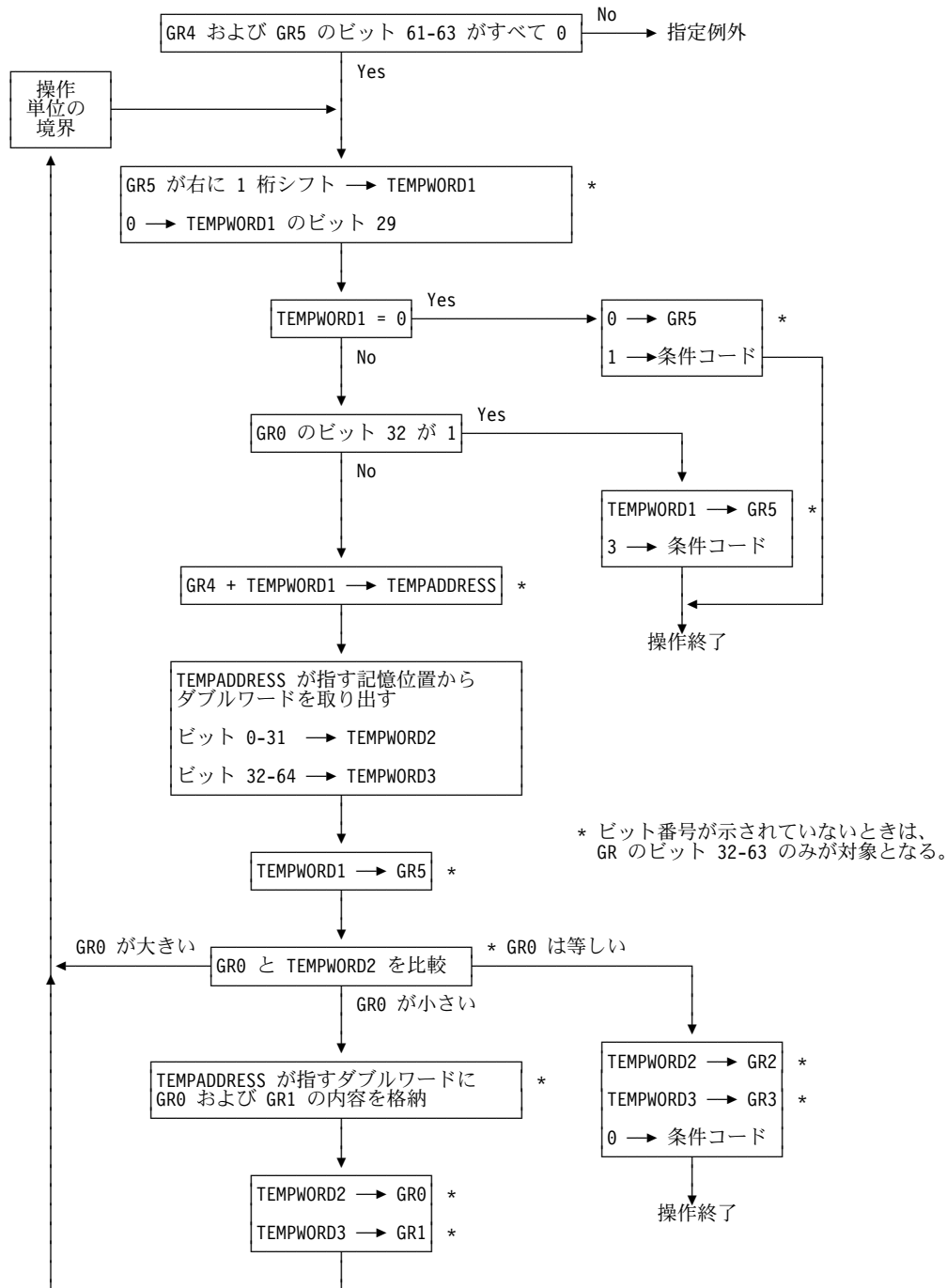


図 7-25. 24 ビットまたは 31 ビット・アドレッシング・モードでの UPDATE TREE の実行

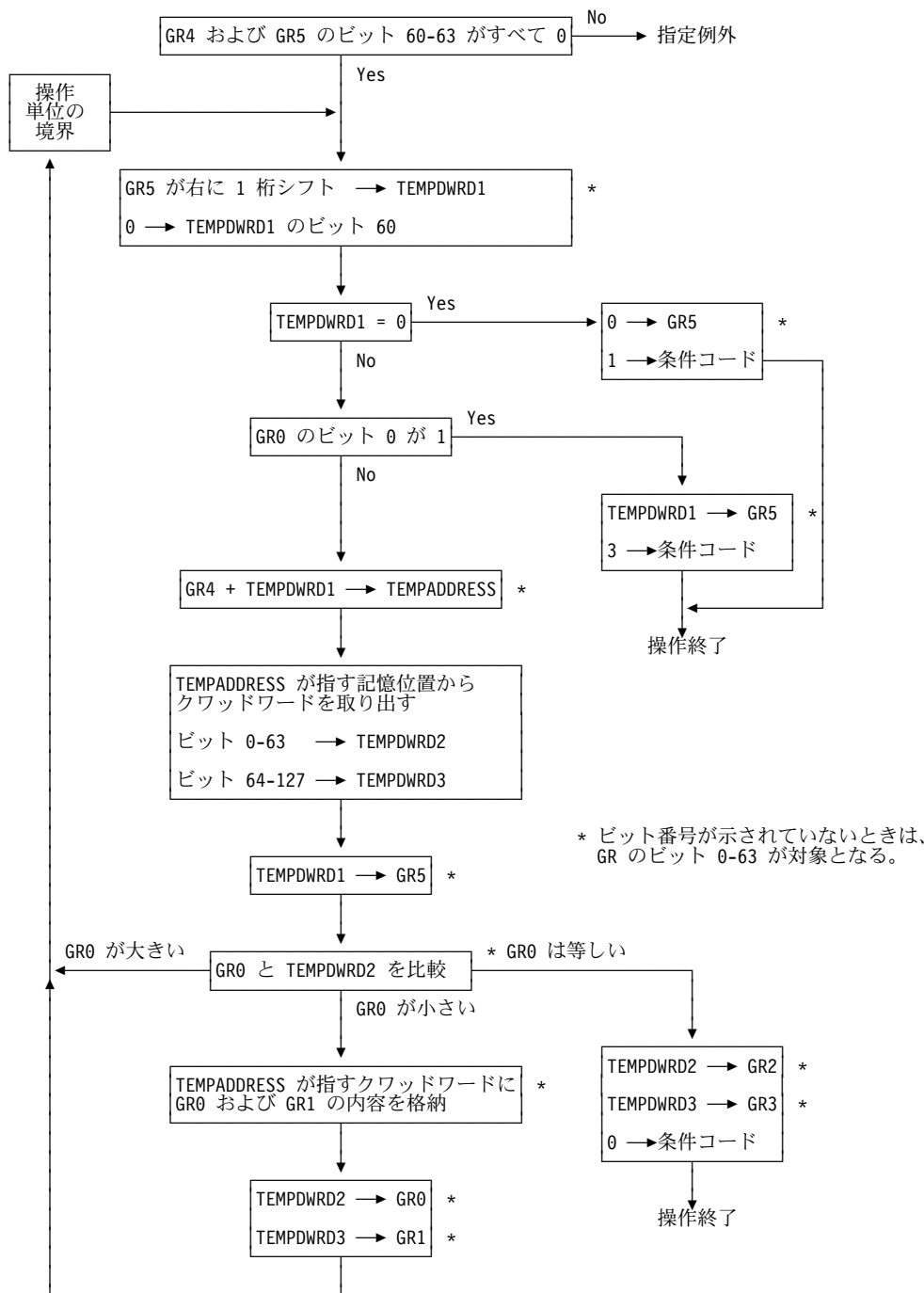


図 7-26. 64 ビット・アドレッシング・モードでの UPDATE TREE の実行

第8章 10 進数命令

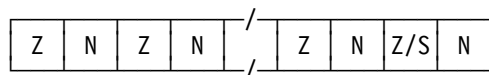
10 進数のフォーマット	8-1	ADD DECIMAL	8-5
ゾーン・フォーマット	8-1	COMPARE DECIMAL	8-6
パック・フォーマット	8-1	DIVIDE DECIMAL	8-6
10 進コード	8-2	EDIT	8-7
10 進数操作	8-2	EDIT AND MARK	8-9
10 進数演算命令	8-2	MULTIPLY DECIMAL	8-11
編集命令	8-3	SHIFT AND ROUND DECIMAL	8-11
10 進数命令の実行	8-3	SUBTRACT DECIMAL	8-12
10 進オペランドを対象とするその他の命令	8-3	TEST DECIMAL	8-12
10 進オペランド・データ例外	8-4	ZERO AND ADD	8-13
命令	8-4		

この章で紹介する 10 進数命令は、10 進数データに対する算術演算と編集操作を行います。そのほかに、第7章、『一般命令』で述べた幾つかの命令も、10 進数データに対する操作を行います。10 進オペランドは常に記憶域内に置かれており、すべての 10 進数命令は SS 命令フォーマットを使用します。10 進オペランドが占める記憶域フィールドは、どのバイト境界から始まっても構いません。

10 進数のフォーマット

10 進数を表すフォーマットには、ゾーン・フォーマットとパック・フォーマットがあります。どちらの 10 進数フォーマットも可変長です。オペランドと結果の長さは、10 進数データを操作する個々の命令で指定されます。どちらのフォーマットでも、各バイトは一對の 4 ビット・コードから成っています。4 ビット・コードには、10 進数字コード、符号コード、およびゾーン・コードが含まれています。

ゾーン・フォーマット



ゾーン・フォーマットでは、1 つのバイトの右端 4 ビットを数字ビット (N) と呼び、通常これは 1 個の 10 進数字を表すコードです。バイトの左端 4 ビットはゾーン・ビット (Z) と呼ばれますが、10 進オペランドの最右端の 1 バイトの場合に限り、左端 4 ビットはゾーンまたは符号 (S) として扱われることがあります。

ゾーン・フォーマットの中の 10 進数字は、英字や特殊文字も含むもっと大きい文字セットの一部となることもあります。したがって、ゾーン・フォーマットは、人間が理解できる形式での数値データの入力、編集、出力に適しています。しかし、ゾーン・フォーマットの 10 進数を直接操作できる 10 進数演算命令はありません。このような数値は、まずパック・フォーマットに変換する必要があります。

編集命令は、最大 256 バイトの結果を生成します。結果の各バイトは、ゾーン・フォーマットの 10 進数字、メッセージ・バイト、または充てんバイトです。

パック・フォーマット



パック・フォーマットでは、各バイトにそれぞれ 2 個の 10 進数字 (D) が含まれていますが、最右端の 1 バイトでは、10 進数字の右側に符号が入っています。10 進数演算は、パック・フォーマットのオペランドを使用して行われ、パック・フォーマットの結果を生成します。

10 進数演算命令に使用されるパック・フォーマットのオペランドおよび結果は、最大 16 バイト (31 桁の数字と符号) ですが、乗数および除数の最大長は 8 バイト (15 桁の数字と符号) です。除算では、商の長さとの余りの長さの合計は、2 バイトから 16 バイトまでです。編集命令は、それぞれパック・フォーマットで可変長の 1 つまたは複数の 10 進数から、最大 256 個の 10 進数字を取り出すことができます。

10 進コード

10 進数字 0-9 に対応する 2 進数エンコードは 0000-1001 です。

優先符号コードは、正が 1100 で、負が 1101 です。これらは、10 進数演算命令および CONVERT TO DECIMAL 命令の結果として生成される符号コードです。

代替符号コードとして、正を表す 1010、1110、および 1111、そして負を表す 1011 があり、これらのコードも、符号の位置にあれば有効な符号として認識されます。代替コードは、どの 10 進数ソース・オペランドにも使用可能ですが、10 進数演算命令または CONVERT TO DECIMAL の完了後の結果の中に現れることはありません。これは、例えばある数値に 0 を加算する場合のように、オペランドが変更されないような場合であっても同じです。ただし、MOVE NUMERICS、MOVE WITH OFFSET、MOVE ZONES、PACK、および UNPACK では、代替符号コードは変更されないままとなります。

無効な符号または数字コードが検出された場合は、データ例外が認識されます。10 進数演算命令および CONVERT TO BINARY の場合は、操作は抑止されません。

編集命令 EDIT および EDIT AND MARK の場合は、無効な符号コードは認識されません。無効な数字コードが原因でデータ例外が起きたときは、操作は中止されません。MOVE NUMERICS、MOVE WITH OFFSET、MOVE ZONES、PACK、および UNPACK では、妥当性検査は行われません。

ゾーン・フォーマットの結果の中では、1 個のゾーン数字および 10 進数字を表す各バイトの左側 4 個のビット位置には、ゾーン・コード 1111 が生成されます。ゾーン・フォーマットの結果を生成するのは、EDIT、EDIT AND MARK、および UNPACK です。EDIT および EDIT AND MARK の場合は、ゾーン・フォーマットの 10 進数字を表す各結果バイトには、左側 4 個のビット位置にゾーン・コード 1111 が含まれ、右側 4 個のビット位置には 10 進数字コードが含まれています。UNPACK の場合は、最右端の 1 バイトを除くすべてのバイトに、1111 のコードが入るゾーン・ビットがあります。最右端のバイトのゾーンには符号が入ります。

図8-1 は、10 進コードの意味を要約して示しています。

コード (2 進数)	認識	
	数字	符号
0000	0	無効
0001	1	無効
0010	2	無効
0011	3	無効
0100	4	無効
0101	5	無効
0110	6	無効
0111	7	無効
1000	8	無効
1001	9	無効
1010	無効	正
1011	無効	負
1100	無効	正 (優先)
1101	無効	負 (優先)
1110	無効	正
1111	無効	正 (ゾーン)

図 8-1. 数字コードと符号コードの要約

プログラミング上の注意: 1111 は、ゾーン・コードであると同時に正符号の代替コードでもあるので、符号なし (正) の 10 進数は、すべてのバイト位置にゾーン・コード 1111 が入ったゾーン・フォーマットで表されることがあります。このような数値をパック・フォーマットに変換する PACK 命令の結果は、10 進数命令のオペランドとしてそのまま使用できます。

10 進数操作

この章で説明する 10 進数命令には、10 進数演算命令と編集命令の 2 つのクラスがあります。

10 進数演算命令

10 進数演算命令は、加算、減算、乗算、除算、比較、およびシフトを行います。

10 進数演算命令のオペランドはパック・フォーマットであり、符号付き 10 進整数として扱われます。10 進整数は、絶対値に独立した正符号または負符号を付けた真の形式で表されます。10 進整数には、1 個から 31 個までの奇数個の 10 進数字が含まれています。これは、1 バイトから 16 バイトまでのオペランド長に対応しています。

通常は、10 進数の 0 は正符号を持っていますが、乗算、除算、およびオーバーフローの結果、負符号を持つ 0 の値が生成されることもあります。このような負の 0 も有効なオペランドであり、COMPARE DECIMAL では正の 0 に等しいものとして扱われます。

命令の中で指定する 2 つのオペランドの長さは、同じでなくても構いません。必要があれば、短い方のオペランドは、左側が 0 で拡張されたものとして扱われます。ただし、結果は、命令で指定されている第 1 オペランドの長さを超えることはできません。

第 1 オペランド・フィールドが短すぎるために、結果の繰り上がりまたは 0 以外の左端数字が失われる場合は、オーバーフロー数字を無視して結果が求められ、条件コード 3 が設定されます。さらに、10 進オーバーフロー・マスク・ビットが 1 のときは、10 進オーバーフローに関するプログラム割り込みが起きます。オペランドの長さのみが、オーバーフローを示すものではありません。オーバーフローが起きた場合、操作中に 0 以外の数字が失われています。

10 進数演算命令の 2 つのオペランドは、互いにオーバーラップしないようにするか、または両方の右端バイトが一致するようにする必要があります。ZERO AND ADD では、第 1 オペランド (結果になる) の最右端バイトが第 2 オペランドの最右端バイトの右にくるような形で、オペランドがオーバーラップすることがあります。このような適正なオーバーラップの場合は、オペランドを右から左へ処理したもとして結果が求められます。演算の実行中に数字コードと符号コードが検査されるので、不適切にオーバーラップしているオペランドはデータ例外として認識されます。ただし、ZERO AND ADD において、第 1 オペランドの最右端バイトが第 2 オペランドの最右端バイトの左側にあるときは、モデルによっては、格納が行われる前に第 2 オペランド全体が取り出されることがあり、その場合はデータ例外は認識されません。アクセス・レジスター・モードで、オーバーラップがどのように検出されるかについては、5-80ページの『単一命令内のインターロック』を参照してください。

プログラミング上の注意: ADD DECIMAL、COMPARE DECIMAL、DIVIDE DECIMAL、MULTIPLY DECIMAL、SUBTRACT DECIMAL、または ZERO AND ADD では、第 1 オペランドと第 2 オペランドの両方に、記憶域内の同じパック 10 進数を指定することができます。このようにすれば、ある 10 進数に同じ 10 進数を加算したり、同じ 10 進数同士を比較したりできます。例えば、SUBTRACT DECIMAL を使用して、記憶域内の特定の 10 進数フィールドを 0 に設定することや、MULTIPLY DECIMAL を使用して、特定の 10 進数をその場で 2 乗することができます。このような場合、2 つのオペランドの長さは必ずしも同じでなくてもよく、命令によっては、同じであってもいけない場合もあります。

編集命令

編集命令には、EDIT と EDIT AND MARK があります。この種の命令では、長さが明示指定されるのは第 1 オペランド (パターン) のみです。第 2 オペランド (ソース) は、操作を完了するために必要なだけの桁数を持つものと見なされます。

編集命令でオペランドがオーバーラップしている場合の結果は、予測不能です。

10 進数命令の実行

10 進数命令の実行では、必ずしも、オペランドのすべてのバイトが同時にアクセスされるとは限らず、また、1 つの記憶位置に対する取り出しアクセスと格納アクセスが連続して発生するとは限りません。さらに、10 進数命令では、ソース・フィールド内のデータは 2 回以上アクセスされることがあり、元のオペランドの値とも最終結果の値とも異なる中間値が、結果フィールドに入れられることがあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。) したがって、マルチプロセッシング構成において、他の CPU も共用記憶位置を更新する可能性があるときは、ADD DECIMAL のような命令を使用して共用記憶位置を更新するのは、安全ではありません。

10 進オペランドを対象とするその他の命令

この章で説明する 10 進数命令のほかに、ゾーン・フォーマットで長さが最大 256 バイトまでのデータを操作する命令として、MOVE NUMERICS と MOVE ZONES があります。また、ゾーン・フォーマットとパック・フォーマットの間でデータを変換するための命令が 2 つあります。その 1 つである PACK は、長さが最大 16 バイトまでのゾーン・データをパック・データに変換し、もう 1 つの UNPACK はその逆の変換を行います。MOVE WITH OFFSET は、長さが最大 16 バイトのパック・データを取り扱うことができます。さらに、パック 10 進数フォーマットと符号付き 2 進整数フォーマットの間で変換を行うための命令が、2 つあります。その 1 つの CONVERT TO BINARY はパック 10 進数を 2 進数に変換し、もう 1 つの CONVERT TO DECIMAL は 2 進数をパック 10 進数に変換します。これらの命令のパック 10 進オペランドの長さは、CONVERT TO BINARY (CVB) および CONVERT TO DECIMAL (CVD) の場合は 8 バイト (15 個の数字および符号) で、CONVERT TO BINARY (CVBG) お

よび CONVERT TO DECIMAL (CVDG) の場合は 16 バイト (31 個の数字および符号) です。上記 7 つの命令は 10 進数命令とは見なされないものであり、これらについては、第 7 章、『一般命令』で説明されています。データをパック・フォーマットからゾーン・フォーマットに変更するには、この章で説明する編集命令も使用できます。

10 進オペランド・データ例外

10 進オペランド・データ例外が認識されるのは、以下のいずれかが真の場合です。

1. 10 進数命令または CONVERT TO BINARY 命令 (第 7 章、『一般命令』を参照) で、オペランドの符号コードまたは数字コードが無効である。
2. ADD DECIMAL、COMPARE DECIMAL、DIVIDE DECIMAL、MULTIPLY DECIMAL、および SUBTRACT DECIMAL において、2 つのオペランド・フィールドが、両者の最右端バイトが一致している以外の形でオーバーラップしている。または、ZERO AND ADD の 2 つのオペランド・フィールドがオーバーラップしており、第 2 オペランドの最右端バイトが、第 1 オペランドの最右端バイトの右側にある。モデルによっては、ZERO AND ADD で不適切な形でオペランドがオーバーラップしていても、10 進オペランド・データ例外として認識されないことがあります。その場合は、結果のどのバイトも格納されないうちに、第 2 オペランド

全体が取り出されるものとして、操作が行われません。

3. MULTIPLY DECIMAL において、被乗数の左端に十分な数の 0 がない。

10 進オペランド・データ例外が起きると、操作は抑止されます。ただし、EDIT および EDIT AND MARK の場合は、操作は抑止されるときと中止されるときがあります。EDIT および EDIT AND MARK の場合は、無効な符号コードが発生することはありません。

命令

8-5 ページの図 8-2 に、10 進数命令とそれぞれのニーモニック、フォーマット、および命令コードを示します。この図には、どのようなときに条件コードが設定されるか、アクセス・レジスターを指定する命令フィールド、そして、オペランド指定、データ、または結果の中のような例外条件があるときにプログラム割り込みが起きるかも示してあります。

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、ADD DECIMAL の場合、AP がニーモニックで、 $D_1(L_1, B_1), D_2(L_2, B_2)$ がオペランド指定です。

プログラミング上の注意: 10 進数命令 TEST DECIMAL が使用できるのは、拡張変換機能 2 がインストールされているときです。

名前	二 モ ニ ック	特性						命 令 コ ード	
ADD DECIMAL COMPARE DECIMAL DIVIDE DECIMAL EDIT EDIT AND MARK	AP CP DP ED EDMK	SS SS SS SS SS	C C C C C	A A A A A	SP	Dd DF Dd Dd DK Dd G1	ST ST ST ST ST	B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₂	FA F9 FD DE DF
MULTIPLY DECIMAL SHIFT AND ROUND DECIMAL SUBTRACT DECIMAL TEST DECIMAL ZERO AND ADD	MP SRP SP TP ZAP	SS SS SS RSL SS	C C C C C	A A A A A	SP E2	Dd Dd DF Dd DF Dd DF	ST ST ST ST ST	B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₁ B ₂	FC F0 FB EBC0 F8

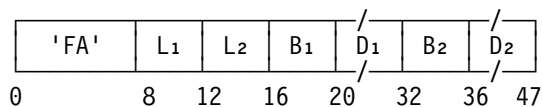
説明:

A 論理アドレスについてのアクセス例外。
B₁ アクセス・レジスタ・モードでは、B₁ フィールドはアクセス・レジスタを指定します。
B₂ アクセス・レジスタ・モードでは、B₂ フィールドはアクセス・レジスタを指定します。
C 条件コードが設定されます。
Dd 10 進オペランド・データ例外。
DF 10 進オーバーフロー例外。
DK 10 進除算例外。
E2 拡張変換機能 2。
G1 命令の実行時に、汎用レジスタ 1 が暗黙的に使用されます。
SP 指定例外。
SS SS 命令フォーマット。
ST PER 記憶域変更イベント。

図 8-2. 10 進数命令の要約

ADD DECIMAL

AP D₁(L₁,B₁),D₂(L₂,B₂) [SS]



第 2 オペランドが第 1 オペランドに加算され、その結果の和が第 1 オペランドの位置に入れます。オペランドおよび結果はパック・フォーマットです。

加算は、両方のオペランドの符号とすべての数字を計算に含めて、代数的に行われます。すべての符号コードおよび数字コードについて、妥当性検査が行われます。

第 1 オペランドが短すぎて、和の 0 以外の左端数字をすべて収容できない場合は、10 進オーバーフローが生じます。操作は完了します。オーバーフロー数字を無視することにより演算結果が求められ、条件コード 3 が設定されます。10 進オーバーフロー・マスクが 1 のときは、10 進オーバーフローに関するプログラム割り込みが起きます。

和の符号は代数の規則に従って決定されます。オーバーフローがないときは、結果が 0 の場合の符号は正になります。オーバーフローが起きた場合は、結果が 0 のときの符号が正か負かは、正しい和が得られたと仮定した場合の符号によって決まります。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

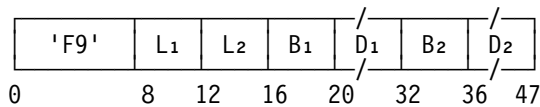
プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)
- データ例外
- 10 進オーバーフロー例外

プログラミング上の注意: ADD DECIMAL 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。

COMPARE DECIMAL

CP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



第 1 オペランドが第 2 オペランドと比較され、その結果が条件コードで示されます。オペランドはパック・フォーマットです。

比較は、10 進数減算の場合と同じ手順に従って代数的に行われますが、どちらのオペランドも変更されずにそのまま残されます。差が 0 であれば、両オペランドは等しいと見なされます。差が 0 以外の場合は、差が正であれば第 1 オペランドの方が大きく、差が負であれば第 1 オペランドの方が小さいことになります。

差は破棄されるので、オーバーフローが起きることはありません。

すべての符号コードおよび数字コードについて、妥当性検査が行われます。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

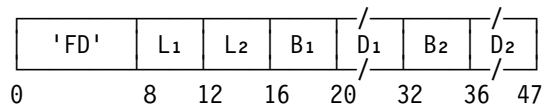
- アクセス例外 (取り出し: オペランド 1 および 2)
- データ例外

プログラミング上の注意:

1. COMPARE DECIMAL 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. 比較では、特定の符号について優先符号コードと代替符号コードは、どちらも等価であると見なされます。
3. 負の 0 と正の 0 は等価であると見なされます。

DIVIDE DECIMAL

DP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



第 1 オペランド (被除数) が第 2 オペランド (除数) で除算されます。結果の商と余りは、第 1 オペランドの位置に入れられます。オペランドおよび結果はパック・フォーマットです。

商は、第 1 オペランド位置の左寄りに入れられます。商フィールドのバイト数は、被除数と除数の長さの差 ($L_1 - L_2$) に等しくなります。余りは、第 1 オペランド位置の右寄りに入れられ、その長さは除数の長さと同じです。商フィールドと余りフィールドを合わせたものが、第 1 オペランド全体を占めます。したがって、商のアドレスは第 1 オペランドのアドレスです。

除数の長さは、15 桁の数字に符号を加えた長さを超えることはできません (L_2 が 7 より大きくてはいけません)。また、被除数の長さより小さくなければなりません (L_2 が L_1 より小さくなければなりません)。そうでない場合は、指定例外が認識されます。

被除数、除数、商、および余りは、どれもパック・フォーマットの符号付き 10 進整数で、フィールド内で右寄せされます。被除数および除数のすべての符号コードおよび数字コードについて、妥当性検査が行われます。

商の符号は、被除数と除数の符号に基づき代数の規則に従って決定されます。余りの符号は被除数の符号と同じになります。商または余りが 0 の場合も、この規則が適用されます。

オーバーフローが起きることはありません。除数が 0 の場合、または商が大きくなりすぎて指定された桁数では表せない場合は、10 進除算例外が認識されます。これには、0 を 0 で割る場合も含まれます。10 進除算例外が示されるのは、被除数と除数の両方の符号コードが有効であり、かつ、その例外を確立するために使用された数字が有効である場合のみです。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)
- データ例外

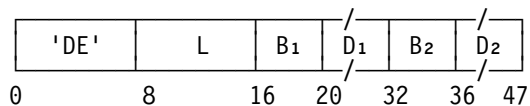
- 10 進除算例外
- 指定例外

プログラミング上の注意:

1. DIVIDE DECIMAL 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 被除数は、31 桁と符号を合わせた長さを超えることはできません。余りは、1 桁と符号より短くなることはできないので、商は 29 桁と符号を合わせた長さを超えることはできません。
3. 10 進除算例外の条件があるかどうかは、試験比較により判別できます。まず、除数の左端の数字を、被除数の左端の数字の 1 桁右の桁に合わせ、被除数と同じ長さになるまで右側に 0 を付加します。符号は無視して、このようにそろえて並べた除数が被除数に等しいかそれより小さい場合は、除算例外が示されます。
4. データ例外が存在しない場合に、被除数の左端の数字が 0 でないときは、10 進除算例外が起きます。

EDIT

ED D₁(L,B₁),D₂(B₂) [SS]



第 2 オペランド (ソース) には、通常 1 つ以上のパック・フォーマットの 10 進数が含まれており、その 10 進数がゾーン・フォーマットに変更され、第 1 オペランド (パターン) に従って修正されます。編集済みの結果は第 1 オペランドと置き換わります。

長さフィールドは、第 1 オペランドの長さを指定します。第 1 オペランドには、どのような値のバイトが含まれていても構いません。

ソースの長さは、パターンに基づく操作によって決まります。通常、ソースは、パック・フォーマットの 1 つ以上の 10 進数から成っています。ソースの各バイトの左端 4 ビットは、10 進数字コード (0000-1001) を指定していなければなりません。この位置に符号コード (1010-1111) がある場合は、データ例外が認識されません。右端 4 ビットには、符号コードまたは 10 進数字コードのどちらでも指定できます。アクセス例外およびデータ例外は、第 2 オペランド内の実際に必要なバイトについてのみ認識されます。

結果は、両方のオペランドが、左から右へ一度に 1 バイトずつ処理されたものとして求められます。パターン・フィールドとソース・フィールドがオーバーラップしている場合は、結果は予測不能です。

編集プロセスでは、パターンの各バイトが次の 3 つのうちいずれかの取り扱いを受けます。

1. 変更されない。
2. ゾーン・フォーマットに展開されたソースの数字で置き換えられる。
3. パターンの最初のバイト (充てんバイトと呼ばれる) で置き換えられる。

この 3 つのアクションのどれが行われるかは、パターン・バイトのタイプ、有効数字標識の状態、および、検査されたソースの数字が 0 かどうかによって決まります。

パターン・バイト: パターン・バイトには、数字選択文字、有効数字開始文字、フィールド分離文字、およびメッセージ・バイトの 4 種類があります。それぞれのコードは以下のとおりです。

名前	コード (2 進数)
数字選択文字	0010 0000
有効数字開始文字	0010 0001
フィールド分離文字	0010 0010
メッセージ・バイト	上記以外

パターンの中に数字選択文字または有効数字開始文字が検出された場合は、有効数字標識およびソース数字が検査されます。その結果、展開されたソース数字または充てんバイトのいずれか該当する方が選択されて、パターン・バイトと置き換わります。さらに、数字選択文字または有効数字開始文字が見つかったときは、有効数字標識が変更されます。

フィールド分離文字は、複数フィールド編集操作で個々のフィールドを識別します。結果の中では、フィールド分離文字は常に充てんバイトで置き換えられ、フィールド分離文字が検出された後は、有効数字標識は常にオフになります。

パターンの中のメッセージ・バイトは、結果の中では、有効数字標識の状態に応じて、充てんバイトで置き換えられるかまたは未変更のままになります。したがって、メッセージ・バイトは、フィールドの有効数字部分の中で埋め込み、句読点、またはテキスト用に使用されるほか、符号に応じた記号を挿入するためにも使用されます。

充てんバイト: パターンの最初の 1 バイトは充てんバイトとして使用されます。充てんバイトはどのようなコードでもとることができ、また同時に制御機能を指定することもあります。このバイトが数字選択文字または有効数字開始文字である場合は、そのコードが充てんバイトに割り当てられた後で、指示されている編集アクションがとられます。

ソース数字: パターン内で数字選択文字または有効数字開始文字が検出されるたびに、新しいソース数字が検査され、その数字をパターン・フィールドに入れるかどうかが決まります。ソース数字は無視されるか、または、左側にゾーン・コード 1111 を付加してゾーン・フォーマットに展開されて、パターン・バイトの該当位置に格納されます。

実行は、ソース数字が一度に 1 バイトずつ選択され、各ソース・バイトが 1 回の編集操作で 1 回だけ検査のために取り出されるものとして、行われます。各ソース数字は、値が 0 かどうかを 1 回だけ検査されます。まず各バイトの左端 4 ビットが検査され、右端 4 ビットは、10 進数字コードを表していれば、数字検査を必要とする次のパターン・バイト用に使用できます。右端 4 ビットに無効な数字コードが含まれているときは、データ例外が認識され、操作は中止されます。

ソース・バイトの左部分の数が検査されるときに、右端 4 ビットについて符号コードの有無が検査されます。右端 4 ビットの中に符号コードが見つかったときは、それらのビットは 10 進数字コードとは見なされず、次のパターン・バイトがソース数字検査を必要としているときは、新しいソース・バイトが記憶域から取り出されません。

パターンに数字選択文字または有効数字開始文字が含まれているときは、ソース・バイトの取り出しも検査も行われません。

有効数字標識: 有効数字標識は、後続のソース数字またはメッセージ・バイトが有意であればオン、非有意であればオフになっています。有意なソース数字は、結果の中の対応する数字選択文字または有効数字開始文字と置き換わります。有意なメッセージ・バイトは、変更されずにそのまま結果の中に残されます。

有効数字標識は、オンまたはオフの状態により、それぞれ、完了したソース・フィールドの値が負であるか正であるかも示し、また、条件コードを設定するときの要因の 1 つとしても使用されます。

編集操作の開始時点、フィールド分離文字が見つかった後、または右端 4 ビットが正符号コードであるソース・

バイトが見つかった後は、有効数字標識はオフに設定されます。

有効数字標識がオンに設定されるのは、有効数字開始文字が検出され、そのソース数字が有効な 10 進数字であるか、または、数字選択文字が検出され、そのソース数字が 0 以外の 10 進数字であり、かつ、いずれの場合もソース・バイトの右端 4 ビットに正符号コードが含まれていない場合です。

上記以外の場合は、有効数字標識は変更されません。負符号コードは、有効数字標識には影響を与えません。

結果バイト: 編集操作の結果はパターンと置き換わります。結果の長さはパターンと同じです。結果は、パターン・バイト、充てんバイト、およびゾーン化されたソース数字で構成されています。

パターン・バイトがメッセージ・バイトであり、有効数字標識がオンになっているときは、そのメッセージ・バイトは変更なしで結果に残されます。パターン・バイトがフィールド分離文字である場合、または、パターン内にメッセージ・バイトが検出されたときに有効数字標識がオフになっている場合は、結果の中では、そのパターン・バイトは充てんバイトで置き換えられます。

パターン内に数字選択文字または有効数字開始文字が検出されたときに、有効数字標識がオフで、ソース数字が 0 である場合は、そのソース数字は非有意と見なされ、そのパターン・バイトは充てんバイトで置き換えられます。数字選択文字または有効数字開始文字が検出されたときに、有効数字標識がオンか、またはソース数字が 0 以外の 10 進数である場合は、そのソース数字は有意と見なされ、ゾーン・フォーマットに変換され、結果の中で該当のパターン・バイトと置き換わります。

条件コード: 編集された最後のフィールドの符号と絶対値を使用して、条件コードが設定されます。「最後のフィールド」とは、最後のフィールド分離文字の後で数字選択文字または有効数字開始文字により選択された、第 2 オペランド内のソース数字を指します。パターンにフィールド分離文字が含まれていない場合は、フィールドは 1 つだけであり、したがってそれが最後のフィールドと見なされます。この条件に該当するソース数字が選択されない場合は、最後のフィールドは長さが 0 と見なされます。

最後に編集されたフィールドが 0 であるか、またはその長さが 0 であるときは、条件コード 0 が設定されません。

編集された最後のフィールドが 0 以外であり、有効数字標識がオンになっているときは、条件コード 1 が設定されます。(これは、検査される最後のソース・バイトの右端 4 ビットに符号コードが含まれている場合は、結果が 0 より小さいことを示します。)

編集された最後のフィールドが 0 以外であり、有効数字標識がオフになっているときは、条件コード 2 が設定されます。(これは、検査される最後のソース・バイトの右端 4 ビットに符号コードが含まれている場合は、結果が 0 より大きいことを示します。)

8-10ページの図8-3は、EDIT および EDIT AND MARK 操作の機能を要約して示しています。左側の 4 つの欄には、編集操作の実行中に生じる可能性がある 4 つの条件の有効なすべての組み合わせを示してあります。右側の 2 つの欄には、それぞれの場合にとられるアクション (結果フィールドに入れられるバイトのタイプ、および有効数字標識の新しい設定) を示してあります。

結果の条件コード:

- 0 最後のフィールドが 0、またはその長さが 0。
- 1 最後のフィールドが 0 より小さい。
- 2 最後のフィールドが 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)
- データ例外

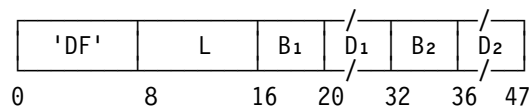
プログラミング上の注意:

1. EDIT 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. 編集には、句読点制御、および、先行ゼロの抑止や保護 (先行ゼロを空白またはアスタリスクで置き換える) が含まれます。また、編集により、プログラムで全桁ゼロのフィールドを空白にすることも容易になります。1 回の操作で複数のフィールドを編集することも、数値情報をテキストと組み合わせることもできます。
3. ほとんどの場合、結果の中では 4 ビットのソース数字ごとに 8 ビットのバイトが生成されるため、ソースの方がパターンより短くなります。

4. パターンの中の数字選択文字と有効数字開始文字の合計数は、常に、編集されるソース数字の数に等しくなります。
5. 充てんバイトが空白で、パターン内に有効数字開始文字が存在せず、各数字選択文字について検査されるソース数字が 0 である場合は、編集結果のフィールドは空白になります。
6. 結果の条件コードは、最後のフィールドが全桁 0 であるかどうかを示し、0 でない場合は有効数字標識の状態を示します。有効数字標識は、検査された最後のソース・バイトの右端 4 ビットに符号コードが含まれている場合のみ、ソース・フィールドの符号を示します。複数フィールド編集操作の場合は、条件コードは、最後のフィールド分離文字の次のフィールドのみの符号と値を示します。
7. DAT がオンのときに、EDIT 命令の第 2 オペランド・アドレスが指す記憶域位置が、その命令の第 1 オペランドの長さに対して、4K バイト境界の左に近い場合は、パフォーマンスが大幅に低下することがあります。これは、マシンが、第 2 オペランドが実際にその境界を超えることになるかどうかを判断するために、命令の試行実行を行うことがあるためです。通常は、EDIT の第 2 オペランドは第 1 オペランドより短いのですが、極端なケースでは第 1 オペランドと同じ長さになることもあります。

EDIT AND MARK

EDMK D₁(L,B₁),D₂(B₂) [SS]



第 2 オペランド (ソース) には、通常 1 つ以上のパック・フォーマットの 10 進数が含まれており、その 10 進数がゾーン・フォーマットに変更され、第 1 オペランド (パターン) に従って修正されます。結果の中の最初の有意バイトが、汎用レジスター 1 に挿入されます。編集済みの結果はパターンと置き換わります。

EDIT AND MARK は、追加の機能が 1 つある点を除けば、EDIT と同じです。この追加機能は、結果のバイトがゾーン化されたソース数字で、検査の前に有効数字標識がオフであった場合に、結果のバイトのアドレスを汎用レジスター 1 に挿入します。この基準を満たす結果バイトがない場合は、汎用レジスター 1 は変更されません。基準を満たす結果バイトが複数あるときは、それらの中で最も右側にあるバイトが挿入されます。

条件				結果	
パターン・バイト	有効数字標識の前の状態	ソース数字	ソースの右端 4 ビットが正符号コードか	結果バイト	数字検査後の有効数字標識の状態
数字選択文字	オフ	0 1-9	* いいえ	充てんバイト ソース数字#	オフ オン
	オン	0-9 0-9	はい いいえ	ソース数字# ソース数字	オフ オン
有効数字開始文字	オフ	0 0 1-9	いいえ はい いいえ	充てんバイト 充てんバイト ソース数字#	オン オフ オン
	オン	1-9 0-9 0-9	はい いいえ はい	ソース数字# ソース数字 ソース数字	オフ オン オフ
フィールド分離文字	*	**	**	充てんバイト	オフ
メッセージ・バイト	オフ オン	** **	** **	充てんバイト メッセージ・バイト	オフ オン

説明:

- * 結果バイトにも有効数字標識の新しい状態にも影響はありません。
- ** ソースが検査されないため、適用されません。
- # EDIT AND MARK の場合に限り、右端の結果バイトが汎用レジスタ 1 に入れます。

図 8-3. 編集機能の要約

24 ビット・アドレッシング・モードでは、アドレスは汎用レジスタ 1 のビット 40-63 と置き換わり、このレジスタのビット 0-39 は変更されません。31 ビット・アドレッシング・モードでは、アドレスは汎用レジスタ 1 のビット 33-63 と置き換わり、ビット 32 は 0 に設定され、ビット 0-31 は変更されません。64 ビット・アドレッシング・モードでは、アドレスは汎用レジスタ 1 のビット 0-63 と置き換わります。

アクセス・レジスタ 1 の内容は変更されません。

EDIT および EDIT AND MARK 操作の要約については、図8-3 を参照してください。

結果の条件コード:

- 0 最後のフィールドが 0、またはその長さが 0。
- 1 最後のフィールドが 0 より小さい。
- 2 最後のフィールドが 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)

- データ例外

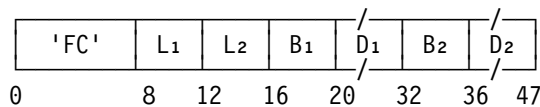
プログラミング上の注意:

1. EDIT AND MARK 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. EDIT AND MARK は、浮動通貨記号挿入のプログラミングをしやすくします。適切なソース・データとパターン・データを使用すれば、浮動通貨記号を挿入するアドレスより 1 だけ大きいアドレスが、汎用レジスタ 1 に挿入されます。そして、R₂ フィールドに 0 を指定した BRANCH ON COUNT (BCTR, BCTGR) を使用することにより、挿入するアドレスを 1 だけ小さくすることができます。
3. 有効数字開始文字が検出されたときに、それに対応するソース数字が 0 であったために、有効数字標識がオンになった場合は、汎用レジスタ 1 にはアドレスは何も挿入されません。この場合に、汎用レジスタ 1 に適切なアドレスが含まれるようにするには、該当の有効数字開始文字の直後にあるパターン・バイトのアドレスを、あらかじめこのレジスタ 1 に入れておきます。

- EDIT AND MARK 命令の 1 回の実行で複数のフィールドを編集する場合は、汎用レジスタ 1 に挿入されるアドレスは、編集されたフィールドのうち、基準を満たしている最右端のフィールドのアドレスです。
- 試行実行が原因でパフォーマンスが低下することがありますが、これについては、EDIT のセクションの『プログラミング上の注意』を参照してください。

MULTIPLY DECIMAL

MP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



第 1 オペランド (被乗数) と第 2 オペランド (乗数) の積が、第 1 オペランドの位置に入れられます。オペランドおよび結果はパック・フォーマットです。

乗数の長さは、15 桁の数字に符号を加えた長さを超えることはできません (L₂ が 7 より大きくてはいけません)。また、被乗数の長さより小さくなければなりません (L₂ が L₁ より小さくなければなりません)。そうでない場合は、指定例外が認識されます。

被乗数の左部分には、少なくとも乗数のバイト数と同数の 0 バイトがなければなりません。さもないと、データ例外が認識されます。この制約条件があるのは、積のオーバーフローが起きないようにするためです。

被乗数、除数、および積は、どれもパック・フォーマットの符号付き 10 進整数で、フィールド内で右寄せされます。被乗数および乗数のすべての符号コードおよび数字コードについて、妥当性検査が行われます。積の符号は、乗数および被乗数の符号に基づき、代数規則に従って決定されます。これは、一方または両方のオペランドが 0 の場合も同じです。

条件コード: コードは変更されません。

プログラム例外:

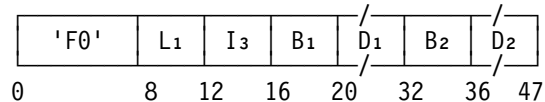
- アクセス例外 (取り出し: オペランド 2。取り出しおよび格納: オペランド 1)
- データ例外
- 指定例外

プログラミング上の注意:

- MULTIPLY DECIMAL の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
- 積は、31 桁と符号を合わせた長さを超えることはできません。積の最左端の数字は常に 0 です。

SHIFT AND ROUND DECIMAL

SRP $D_1(L_1, B_1), D_2(B_2), I_3$ [SS]



第 1 オペランドが、第 2 オペランド・アドレスの方向に、第 2 オペランド・アドレスで指定されている 10 進数字桁の数だけシフトされます。右方向へのシフトが指定されている場合は、第 1 オペランドの絶対値が、丸め数字 I₃ を使用して丸められます。第 1 オペランドおよび結果はパック・フォーマットです。

第 1 オペランドは、パック 10 進数フォーマットであると見なされます。数字部分のみがシフトされ、符号部分はシフトの対象にはなりません。シフトにより空になった数字位置には 0 が入れられます。結果は第 1 オペランドと置き換わります。指定された第 1 オペランド位置以外のところには、何も格納されません。

B₂ フィールドと D₂ フィールドに指定される第 2 オペランド・アドレスは、データのアドレッシングには使用されません。このアドレスのビット 58-63 はシフト値であり、このアドレスの左寄りのビットは無視されます。

シフト値は 6 ビットの 2 進整数で、シフトする方向と 10 進数字桁の数を示します。正のシフト値は、左方向へのシフトを指定します。負のシフト値は 2 の補数表記で表され、右方向へのシフトを指定します。以下に示すのは、シフト値の解釈の例です。

シフト値 (2 進数)	桁数と方向
011111	左へ 31 桁
000001	左へ 1 桁
000000	シフトなし
111111	右へ 1 桁
100000	右へ 32 桁

右方向のシフトの場合は、I₃ フィールド (命令のビット 12-15) が 10 進丸め数字として使用されます。丸めが行われるときは、第 1 オペランドは符号を無視して正の値として扱われ、シフトにより範囲を逸脱することになる

最左端の数字に丸め数字が 10 進法で加算され、繰り上がりが生じる場合は、左方向に繰り上がりが波及します。そして、この加算の結果が右方向にシフトされます。最右端の 10 進数字位置から外れてシフトされる数字は、妥当性検査を受けることと丸めにも関与することを除いては無視され、そして失われます。

左方向の移動で、1 つ以上の 0 でない数字が範囲を逸脱してシフトされる場合は、10 進オーバーフローが起きます。操作は完了します。オーバーフロー数字を無視することにより演算結果が求められ、条件コード 3 が設定されます。10 進オーバーフロー・マスクが 1 のときは、10 進オーバーフローに関するプログラム割り込みが起きます。右方向のシフトの場合は、丸めの有無にかかわらずオーバーフローが起きることはありません。シフトが指定されていない場合も、オーバーフローは起きません。

オーバーフローがないときは、結果が 0 の場合の符号は正になります。オーバーフローが起きるときは、結果の符号は元の符号と同じですが、優先符号コードが使用されます。

第 1 オペランドの符号コードおよび数字コードが有効でないか、または丸め数字が有効な数字コードでないときは、データ例外が認識されます。第 1 オペランドのコードの妥当性は、シフトが指定されていない場合でも検査され、丸め数字の妥当性は、丸めのための加算が行われない場合でも検査されます。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

- アクセス例外 (取り出しおよび格納: オペランド 1)
- データ例外
- 10 進オーバーフロー例外

プログラミング上の注意:

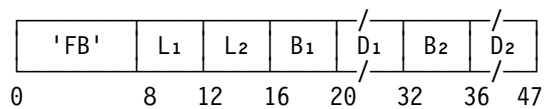
1. SHIFT AND ROUND DECIMAL 命令の使用例は、付録A、『数の表現と命令の使用例』に示されています。
2. SHIFT AND ROUND DECIMAL を使用して、左方向に最大 31 桁、右方向に最大 32 桁のシフトを

行うことができます。丸めがある場合も含めて、どのような 10 進数でも、このシフトによりすべての数字を消去できます。

3. 右方向のシフトの場合、丸め数字として 5 を使用すれば、一般的な四捨五入の結果が得られます。丸め数字 0 は、丸めなしで切り捨てを行うことを指定します。
4. B₂ フィールドが 0 のときは、6 ビットのシフト値は命令のビット 42-47 から直接取得されます。

SUBTRACT DECIMAL

SP D₁(L₁,B₁),D₂(L₂,B₂) [SS]



第 1 オペランドから第 2 オペランドが減算され、その結果の差が第 1 オペランドの位置に入れます。オペランドおよび結果はパック・フォーマットです。

SUBTRACT DECIMAL の実行は、第 2 オペランドが記憶域内の符号とは逆の符号を持つものと見なされるという点以外は、ADD DECIMAL の場合と同じです。記憶域内の第 2 オペランドは変更されません。

結果の条件コード:

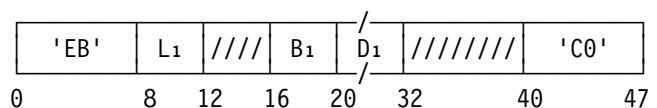
- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2. 取り出しおよび格納: オペランド 1)
- データ例外
- 10 進オーバーフロー例外

TEST DECIMAL

TP D₁(L₁,B₁) [RSL]



第 1 オペランドの 10 進数字および符号コードが有効かどうかテストされ、条件コードで結果が示されます。オペランドはパック・フォーマットです。

結果の条件コード:

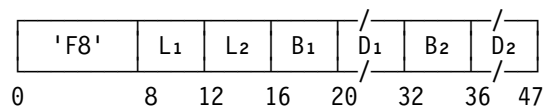
- 0 すべての数字コードおよび符号が有効です。
- 1 符号が無効です。
- 2 少なくとも 1 つの数字コードが無効です。
- 3 符号および少なくとも 1 つの数字コードが無効です。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1)
- 操作例外 (拡張変換機能 2 がインストールされていない場合)

ZERO AND ADD

ZAP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



第 2 オペランドが第 1 オペランドの位置に入れられません。この操作は 0 への加算と同じです。オペランドおよび結果はパック・フォーマットです。

符号コードおよび数字コードの妥当性検査は、第 2 オペランドについてのみ行われます。必要があれば、短い方のオペランドの左側に余分な 0 が付加されます。

第 1 オペランドが短すぎて、第 2 オペランドの 0 以外の左端数字をすべて収容できない場合は、10 進オーバーフローが起きます。操作は完了します。オーバーフロー数字を無視することにより演算結果が求められ、条件コード 3 が設定されます。10 進オーバーフロー・マス

クが 1 のときは、10 進オーバーフローに関するプログラム割り込みが起きます。

オーバーフローがないときは、結果が 0 の場合の符号は正になります。オーバーフローが起きたときは、結果が 0 の場合の符号は第 2 オペランドの符号と同じですが、優先符号コードが使用されます。

第 1 オペランドの最右端バイトが、第 2 オペランドの最右端バイトと同じかまたはそれより右側にあれば、2 つのオペランドがオーバーラップしていても構いません。その場合は、オペランドを右から左へ処理したものとして結果が求められます。2 つのオペランドがオーバーラップしており、第 1 オペランドの最右端バイトが第 2 オペランドの最右端バイトの左側にあるときは、モデルによっては、データ例外が認識される場合と、結果のどのバイトも格納されないうちに第 2 オペランド全体が取り出されたものとして結果が求められる場合があります。

結果の条件コード:

- 0 結果は 0 で、オーバーフローは起きません。
- 1 結果は 0 より小さく、オーバーフローは起きません。
- 2 結果は 0 より大きく、オーバーフローは起きません。
- 3 オーバーフロー。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- データ例外
- 10 進オーバーフロー例外

プログラミング上の注意: ZERO AND ADD 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。

第9章 浮動小数点の概要とサポート命令

レジスターと制御	9-2	BFP と HFP の数の範囲	9-4
浮動小数点レジスター	9-2	BFP と HFP の間の等価の数表現	9-4
追加浮動小数点 (AFP) レジスター	9-2	命令	9-6
有効な浮動小数点レジスター指定	9-2	CONVERT BFP TO HFP	9-8
浮動小数点制御 (FPC) レジスター	9-2	CONVERT HFP TO BFP	9-9
AFP レジスター制御ビット	9-2	LOAD	9-11
明示的な丸め方式	9-3	LOAD ZERO	9-11
丸めアクションの要約	9-3	STORE	9-11
BFP と HFP の数表現の比較	9-4	すべての浮動小数点命令の要約	9-12

浮動小数点命令は、絶対値が広範囲にわたるオペランドに対する計算を行い、精度保持のための位取りをした結果を取得するために使用されます。

浮動小数点オペランドのフォーマットは、基数 16 または基数 2 のいずれかに基づいています。基数値 16 および 2 は、「16 進」浮動小数点 (HFP) および「2 進」浮動小数点 (BFP) という用語につながっています。さらに、オペランドのフォーマットは、短精度 (32 ビット)、長精度 (64 ビット)、および拡張精度 (128 ビット) の 3 種類のオペランド長にも基づいています。短精度オペランドは、長精度オペランドまたは拡張精度オペランドに比べて所要記憶域の量が少なくてすみます。しかし、長精度オペランドおよび拡張精度オペランドを使用すれば、計算の精度が高くなります。

浮動小数点オペランドに使用できるのは数値ですが、BFP の場合のみ、非数値 (NaN = not-a-number) も使用できます。数値オペランド (浮動小数点数と呼ばれる) のコンポーネントには、符号ビット、符号付き 2 進指数、および仮数の 3 種類があります。仮数は、暗黙の小数点の左にある 1 個の暗黙の単位数字と、右にある明示的な小数部フィールドから成っています。仮数の数字は基数 2 または 16 をベースとしています。数値の絶対値 (符号なしの値) は、仮数と、基数を指数に従って累乗した値との積です。この数値は、符号ビットが 0 なら正、1 なら負になります。非数値 BFP オペランドにも、符号ビット、符号付き指数、および小数部フィールドがあります。

16 進浮動小数点 (HFP) オペランドのフォーマットでは、基数 16 の累乗を指定する指数と、16 進数の仮数を使用できます。指数の範囲は、短精度、長精度、拡張精度のどのフォーマットの場合も同じです。HFP データに対するほとんどの操作の結果は、ターゲット・フォーマットに収まるように切り捨てられますが、狭いフォー

マットに変換するときに結果を丸めるための命令も使用できます。HFP オペランドの場合、仮数の暗黙の単位数字は常に 0 です。仮数の値と小数部とは同じなので、HFP 操作については小数部という用語を使用して説明し、仮数という用語は使用しないことにします。

2 進浮動小数点 (BFP) オペランドのフォーマットでは、基数 2 の累乗を指定する指数と、2 進数の仮数を使用できます。指数の範囲はフォーマット間で異なり、長いフォーマットほど範囲は大きくなります。長精度フォーマットおよび拡張精度フォーマットでは、BFP データの指数範囲は HFP データに比べてはるかに大きくなります。BFP データに対する操作の結果は、ターゲット・フォーマットに収まるように自動的に丸められます。丸めのしかたは、プログラムで設定する丸めモードによって決まります。

どのような HFP 操作においても、正規化された数 (正規化数) または正規化されていない数 (非正規化数) のどちらでもオペランドとして使用できます。正規化数とは、最左端の小数桁が 0 でない値のことです。ほとんどの HFP 命令は、最大限の精度を達成できるように正規化された結果を生成します。正規化されていない結果を生成する HFP 加算命令および減算命令も使用できません。

BFP 操作の場合は、正規化されないオペランドはありません。正規化された BFP 数値の場合、仮数の暗黙の単位数字は 1 です。値の絶対値が小さすぎて正規化された形式で表せない場合は、暗黙の単位数字は 0 です。このような数値を「非正規化」数と呼びます。HFP フォーマットでは、正規化されていない数を使用できるため、同じ値が所定のフォーマットで複数の表現をとることがありますが、BFP フォーマットにはこのような冗長性は許されません。

BFP および HFP のどちらのデータ・フォーマットも、他のすべてのデータ・フォーマットと同様に、記憶域内では左から右への順序で現れます。データ・フォーマットの中では、0-7 の番号が付けられたビットが、記憶域内の最左端の (最も番号の小さい) バイト位置にあるバイトを形成し、ビット 8-15 がその次の順番に当たる位置のバイトを形成します (以下同様です)。(3-2ページの『記憶機構のアドレッシング』も参照してください。)

ほとんどの浮動小数点命令については、本書の第18章、『16 進浮動小数点命令』および第19章、『2 進浮動小数点命令』で詳しく定義されています。この章 (第9章) では、浮動小数点サポート (FPS) 命令と呼ばれる命令について詳しく説明します。FPS 命令には、BFP または HFP のどちらのフォーマットのオペランドでも使用できるものと、この 2 つのフォーマット間で変換する機能を備えているものがあります。この章では、すべての浮動小数点命令の要約情報も提供します。

レジスターと制御

浮動小数点レジスター

浮動小数点命令 (FPS、BFP、HFP) は、すべて同じ 16 個の浮動小数点レジスターを使用します。浮動小数点レジスターは 0 から 15 までの番号で識別され、浮動小数点命令の中の 4 ビットの R フィールドにより指定されます。浮動小数点レジスターは、どれも長さは 64 ビットで、短精度 (32 ビット) または長精度 (64 ビット) のどちらの浮動小数点オペランドでも格納できます。

短精度浮動小数点数は、浮動小数点レジスターの左端 32 個のビット位置のみです。レジスターの右端 32 個のビット位置は、レジスターが短精度フォーマットのオペランドのソースであるときは無視されます。また、特に他の指定がない限り、短精度の結果がレジスターに入られるときは、これらのビットは変更されません。

拡張フォーマット (128 ビット) の数値は、1 つの数値が 1 つのレジスター・ペアを占めます。レジスター・ペアは、16 個のレジスターを、0 と 2、4 と 6、8 と 10、12 と 14、1 と 3、5 と 7、9 と 11、および 13 と 15 のように組み合わせで作られます。

8 つのペアは、それぞれ、番号の小さい方のレジスターの番号で参照されます。

追加浮動小数点 (AFP) レジスター

浮動小数点レジスター 0、2、4、および 6 は、始めから ESA/390 の各モデルで使用できるものでした。残りの 12 個の浮動小数点レジスター (1、3、5、および 7-15) は、後で ESA/390 に追加されたもので、追加浮動小数点 (AFP) レジスターと呼ばれています。AFP レジスターを使用できるのは、制御レジスター 0 のビット 45 (AFP レジスター制御ビット) が 1 の場合のみです。AFP レジスター制御ビットが 0 のときに AFP レジスター制御ビットを使用しようとすると、AFP レジスター・データ例外 (DXC 1) が起きます。

有効な浮動小数点レジスター指定

命令では、短精度または長精度の浮動小数点オペランドのレジスター位置を指定するために、インストールされているどのレジスターでも使用できます。

拡張精度フォーマットで浮動小数点オペランドを指定する命令では、レジスター 0、1、4、5、8、9、12、または 13 のいずれかを指定する必要があります。さもないと、指定例外が認識されます。

浮動小数点制御 (FPC) レジスター

浮動小数点制御 (FPC) レジスターは、マスク・ビット、フラグ・ビット、データ例外コード、および丸めモード・ビットを含む 32 ビット・レジスターです。FPC レジスターについては、19-2ページの『浮動小数点制御 (FPC) レジスター』で説明します。

AFP レジスター制御ビット

制御レジスター 0 のビット 45 は、AFP レジスター制御ビットです。AFP レジスターおよび BFP 命令を正常に使用できるのは、AFP レジスター制御ビットが 1 の場合のみです。AFP レジスター制御ビットが 0 のときに、12 個の追加浮動小数点レジスターのどれかを使用しようとすると、AFP レジスター・データ例外 (DXC 1) が起きます。AFP レジスター制御ビットが 0 のときにいずれかの BFP 命令を実行しようとすると、BFP 命令データ例外 (DXC 2) が起きます。DXC 1 と DXC 2 の両方の条件が存在する場合は、DXC 1 が報告されず、データ例外および指定例外の両方の条件が存在する場合は、どちらの例外が報告されるかは予測不能です。

AFP レジスター制御ビットの初期値は、0 です。

明示的な丸め方式

浮動小数点サポート命令 CONVERT HFP TO BFP には M_3 修飾子フィールドがあり、そこに 5 つの丸め方式のいずれかを指定できます。1 つの HFP 命令 (CONVERT TO FIXED) および 3 つの BFP 命令 (CONVERT TO FIXED、DIVIDE TO INTEGER、および LOAD FP INTEGER) にも、 M_3 修飾子フィールドか、またはそれに似た M_4 修飾子フィールドがあります。5 つの丸め方式は、以下のとおりです。

M_3

または

M_4 丸め方式

- 1 バイアス付き最近値丸め
中間結果を、上位または下位の最も近い表現可能な値に丸めます。つまり、符号を無視して、保持すべき最後の結果ビットの直後のビットに 1 を加算し、繰り上がりを波及させ、保持すべき最後のビットを超えるビットを破棄します。
- 4 最近値丸め
中間結果を、上位または下位の最も近い表現可能な値に丸めます。つまり、符号を無視して、保持すべき最後の結果ビットの直後のビットに 1 を加算し、繰り上がりを波及させ、保持すべき最後のビットを超えるビットを破棄します。差が正確に 0.5 ulp である (最後の位置の直後のビット位置が 1 で、その後のビット位置がすべて 0 である) 場合は、最も近い偶数値が選択されます。つまり、丸めによ

る加算後、保持される最後の結果ビットは 0 に設定されます。

- 5 0 方向への丸め:
保持すべき最後の中間結果ビットより右側にあるすべてのビットを破棄します。
- 6 $+\infty$ 方向への丸め:
中間結果が正であり、保持すべき最後の結果ビットの右側のどこかに 1 がある場合は、そのビットに 1 を加算します。そして、正負どちらの場合も、保持すべき最後のビットを超えるビットを破棄します。
- 7 $-\infty$ 方向への丸め:
中間結果が負であり、保持すべき最後の結果ビットの右側のどこかに 1 がある場合は、そのビットから 1 を減算します (つまり、絶対値に 1 を加算します)。そして、正負どちらの場合も、保持すべき最後のビットを超えるビットを破棄します。

M_3 または M_4 の値が 0 の場合にどのような処理になるかは、命令のタイプによって異なります。BFP 命令では、 M_3 または M_4 の値が 0 のときは、FPC レジスターに指定されている現行の丸めモードに従って丸めが行われます。浮動小数点サポート命令および HFP 命令では、 M_3 または M_4 が 0 のときは、上記の 5 (0 方向への丸め) と同じ処理が行われます。

丸めアクションの要約

9-4 ページの図 9-1 は、浮動小数点 (FPS) 命令、BFP 命令、および HFP 命令の場合の丸めアクションを要約しています。

命令	丸めアクションの対象		
	FPS 命令	HFP 命令	BFP 命令
ADD	-	-	CRM
ADD NORMALIZED	-	GD	-
ADD UNNORMALIZED	-	GD	-
CONVERT BFP TO HFP	E	-	-
CONVERT FROM FIXED	-	RTZ	CRM
CONVERT HFP TO BFP	M	-	-
CONVERT TO FIXED	-	M	M
DIVIDE	-	RTZ	CRM
DIVIDE TO INTEGER	-	-	M
HALVE	-	RTZ	-
LOAD FP INTEGER	-	RTZ	M
LOAD ROUNDED	-	BR	CRM
MULTIPLY	-	RTZ	CRM
MULTIPLY AND ADD	-	-	CRM
MULTIPLY AND SUBTRACT	-	-	CRM
SQUARE ROOT	-	BR	CRM
SUBTRACT	-	-	CRM
SUBTRACT NORMALIZED	-	GD	-
SUBTRACT UNNORMALIZED	-	GD	-

説明:

BR バイアス付き最近値丸め。
CRM 現行丸めモードに従った丸め。
E 結果は正確であり、丸めは不要。
GD 保護桁を使用した丸め。該当の命令の定義を参照。これは、すべてではありませんが、ほとんどの場合 0 方向に丸められます。
M 命令の修飾子フィールドに指定されている丸め。
RTZ 0 方向への丸め。

図 9-1. 丸めアクションの比較

BFP と HFP の数表現の比較

BFP と HFP の数の範囲

図9-2 は、各浮動小数点フォーマットで表せる数値の範囲を 10 進数形式で示しています。

	タイプ	短精度	長精度	拡張精度
Nmax	BFP	$\pm 3.4 \times 10^{+38}$	$\pm 1.8 \times 10^{+308}$	$\pm 1.2 \times 10^{+4932}$
	HFP	$\pm 7.2 \times 10^{+75}$	$\pm 7.2 \times 10^{+75}$	$\pm 7.2 \times 10^{+75}$
Nmin	BFP	$\pm 1.2 \times 10^{-38}$	$\pm 2.2 \times 10^{-308}$	$\pm 3.4 \times 10^{-4932}$
	HFP	$\pm 5.5 \times 10^{-79}$	$\pm 5.5 \times 10^{-79}$	$\pm 5.5 \times 10^{-79}$
Dmin	BFP	$\pm 1.4 \times 10^{-45}$	$\pm 4.9 \times 10^{-324}$	$\pm 6.5 \times 10^{-4966}$
	HFP	$\pm 5.2 \times 10^{-85}$	$\pm 1.2 \times 10^{-94}$	$\pm 1.7 \times 10^{-111}$

説明:

Dmin 表現可能な非正規化数 (BFP) または 0 以外の正規化されていない数 (HFP) のうちで、絶対値が最小の値。
Nmax 表現可能な数値のうち絶対値が最大の値。
Nmin 表現可能な正規化数のうち絶対値が最小の値。
示されている値は、10 進数に換算した概算値です。

図 9-2. BFP および HFP フォーマットの数値範囲

BFP と HFP の間の等価の数表現

HFP 数値の指数は、指数部と呼ばれる符号なしの 7 ビットの 2 進整数で表されます。指数部は、指数値に 64 を加算することにより求められます (64 増しコード)。指数部の範囲は 0 から 127 までで、これは -64 から +63 までの指数範囲に対応します。

BFP 数値の指数は、バイアス指数と呼ばれる符号なしの 2 進整数で表されます。バイアス指数は、指数値にバイアスを加算することにより求められます。バイアス指数を含むビット位置の数、バイアスの値、および指数範囲は、数値フォーマット (短精度、長精度、または拡張精度) によって異なります。19-5 ページの図 19-7 は、フォーマット別のこれらの値を示しています。バイアス指数は、HFP フォーマットの指数部に似ていますが、全桁 0 および全桁 1 のバイアス指数には特殊な意味があるという点が異なります。これについては、19-6 ページの『BFP データのクラス』で説明します。

BFP または HFP の 3 つのフォーマットのどれにおいても、数値の 2 進小数点 (BFP の場合) または 16 進小数点 (HFP の場合) は、左端の小数桁の左にあるものと見なされます。小数点の左には暗黙の単位数字があり、これは、HFP 数値の場合は 0 と見なされ、BFP 数値の場合は、正規化された数値については 1 で、0 および非正規化数値については 0 と見なされます。

図9-3 および 9-6ページの図9-4 は、BFP フォーマットと HFP フォーマットでの同じ数値の最も近い表現の例を示しています。BFP の値は最近値に丸められ、HFP の値は切り捨てられています。

これらの図は、必ずしも、BFP/HFP 変換の結果を正確に示しているわけではありません。丸め誤差によりわずかな相違が生じていることがあります。また、図9-3 に示されているのは、丸められた短精度フォーマットの数値であり、短精度 BFP オペランドを変換した結果の長精度 HFP ではありません。

値		S BE または C	小数部
1.0	B	0	01111111 000000000000000000000000
	H	0	1000001 000100000000000000000000
0.5	B	0	01111110 000000000000000000000000
	H	0	1000000 100000000000000000000000
1/64	B	0	01111001 000000000000000000000000
	H	0	0111111 010000000000000000000000
+0	B	0	00000000 000000000000000000000000
	H	0	0000000 000000000000000000000000
-0	B	1	00000000 000000000000000000000000
	H	1	0000000 000000000000000000000000
-15.0	B	1	10000010 111000000000000000000000
	H	1	1000001 111100000000000000000000
20/7	B	0	10000000 01101101101101101101101101101101
	H	0	1000001 001011011011011011011011011011
2-126	B	0	00000001 000000000000000000000000
	H	0	0100001 010000000000000000000000
2-149	B	0	00000000 0000000000000000000000001
	H	0	0011011 100000000000000000000000
2128xF F=1-2-24	B	0	11111110 11111111111111111111111111111111
	H	0	1100000 11111111111111111111111111111111
2-260	B	0	(数値が小さすぎる)
	H	0	0000000 00010000000000000000000000
2248xF F=1-2-24	B	表現不能	
	H	0	11111110 11111111111111111111111111111111
説明:			
B		BFP。	
BE または C		BFP 数値のバイアス指数または HFP 数値の指数部。	
H		HFP。	
S		符号。	

図 9-3. 短精度フォーマットでの BFP および HFP の数値の例

値		S BE または C	小数部
1.0	B	0	0111111111 00000000000000000000 00000000000000000000000000000000
	H	0	1000001 000100000000000000000000 00000000000000000000000000000000
0.5	B	0	01111111110 00000000000000000000 00000000000000000000000000000000
	H	0	1000000 100000000000000000000000 00000000000000000000000000000000
1/64	B	0	011111111001 00000000000000000000 00000000000000000000000000000000
	H	0	01111111 010000000000000000000000 00000000000000000000000000000000
+0	B	0	0000000000 00000000000000000000 00000000000000000000000000000000
	H	0	0000000 000000000000000000000000 00000000000000000000000000000000
-0	B	1	0000000000 00000000000000000000 00000000000000000000000000000000
	H	1	0000000 000000000000000000000000 00000000000000000000000000000000
-15.0	B	1	1000000010 11100000000000000000 00000000000000000000000000000000
	H	1	1000001 111100000000000000000000 00000000000000000000000000000000
20/7	B	0	1000000000 01101101101101101101 1011011011011011011011011011011011
	H	0	1000001 001011011011011011011011 01101101101101101101101101101101
2-1022	B	0	0000000001 00000000000000000000 00000000000000000000000000000000
	H	0	(数値が小さすぎる)
2-1074	B	0	0000000000 00000000000000000000 00000000000000000000000000000001
	H	0	(数値が小さすぎる)
21024 _x F F=1-2 ⁻⁵³	B	0	11111111110 11111111111111111111 11111111111111111111111111111111
	H		表現不能
2-260	B	0	01011111011 00000000000000000000 00000000000000000000000000000000
	H	0	0000000 000100000000000000000000 00000000000000000000000000000000
2 ²⁴⁸ _x F F=1-2 ⁻⁵⁶	B	0	10011110111 00000000000000000000 00000000000000000000000000000000
	H	0	11111110 111111111111111111111111 11111111111111111111111111111111
説明:			
B		BFP。	
BE または C		BFP 数値のバイアス指数または HFP 数値の指数部。	
H		HFP。	
S		符号。	

図 9-4. 長精度フォーマットでの BFP および HFP の数値の例

命令

9-7ページの図9-5 に、浮動小数点サポート命令と、それぞれのニーモニックおよび命令コードを示します。この図の「特性」の欄には、命令のフォーマット、どのようなときに条件コードが設定されるか、アクセス・レジスタを指定する命令フィールド、そして、オペランド指定、データ、または結果の中にどのような例外条件があるときにプログラム割り込みが起きるかを示してあります。

すべての浮動小数点サポート命令は、AFP レジスタ制御ビット (制御レジスタ 0 のビット 45) により制御されます。オペランド位置の 1 つとして AFP レジスタが指定されている場合は、AFP レジスタ制御ビットは 1 でなければなりません。さもないと、AFP レジスタ・データ例外 (DXC 1) が認識されます。

RR、RRE、または RRF フォーマットの浮動小数点命令では、ニーモニックの最後の 1 文字は R です。ま

た、浮動小数点命令については、オペランド・フォーマットの長さを表すための以下の文字を使用しています。

- D 長精度
- E 短精度
- X 拡張精度

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、LOAD (短精度) を使用したレジスタ間操作では、LER がニーモニックで、R1,R2 がオペランド指定です。

名前	ニーモニック	特性					命令コード
CONVERT BFP TO HFP (長精度)	THDR	RRE	C			Da	B359
CONVERT BFP TO HFP (短精度から長精度へ)	THDER	RRE	C			Da	B358
CONVERT HFP TO BFP (長精度)	TBDR	RRF	C		SP	Da	B351
CONVERT HFP TO BFP (長精度から短精度へ)	TBEDR	RRF	C		SP	Da	B350
LOAD (拡張精度)	LXR	RRE			SP	Da	B365
LOAD (長精度)	LDR	RR				Da	28
LOAD (長精度)	LD	RX		A		Da	B ₂ 68
LOAD (短精度)	LER	RR				Da	38
LOAD (短精度)	LE	RX		A		Da	B ₂ 78
LOAD ZERO (拡張精度)	LZXR	RRE			SP	Da	B376
LOAD ZERO (長精度)	LZDR	RRE				Da	B375
LOAD ZERO (短精度)	LZER	RRE				Da	B374
STORE (長精度)	STD	RX		A		Da	ST B ₂ 60
STORE (短精度)	STE	RX		A		Da	ST B ₂ 70
説明:							
A 論理アドレスについてのアクセス例外。							
B ₂ アクセス・レジスタ・モードでは、B ₂ フィールドはアクセス・レジスタを指定します。							
C 条件コードが設定されます。							
Da AFP レジスタ・データ例外。							
RR RR 命令フォーマット。							
RRE RRE 命令フォーマット。							
RRF RRF 命令フォーマット。							
RX RX 命令フォーマット。							
SP 指定例外。							
ST PER 記憶域変更イベント。							

図 9-5. 浮動小数点サポート命令の要約

CONVERT BFP TO HFP

ニーモニック R₁, R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
THDER	'B358'	短精度 BFP オペ ランド、
THDR	'B359'	長精度 HFP 結果 長精度 BFP オペ ランド、 長精度 HFP 結果

第 2 オペランド (ソース・オペランド) が、2 進浮動小数点 (BFP) フォーマットから 16 進浮動小数点 (HFP) フォーマットに変換され、正規化された結果が第 1 オペランドの位置に入れます。ソース・オペランドの符号と絶対値がテストされ、条件コードの設定が決定されます。

数値オペランドの場合は、結果の符号はソース・オペランドの符号です。ソース・オペランドの符号ビットが 1 で、他のビットがすべて 0 である場合は、結果も 1 の後にすべて 0 を付加したものになります。

THDR において結果の指数部が負になる場合は、結果はすべて 0 になり、ソース・オペランドと同じ符号が付けられ、条件コードは、ソース・オペランドの符号を示す 1 または 2 に設定されます。

THDR において 16 進数の中間結果の指数部が大きすぎてターゲット・フォーマットに収まらない場合は、結果は、すべて 0 (つまり表現可能な最大絶対値の数) に設定されますが、符号はソース・オペランドと同じになり、条件コード 3 が設定されます。

この命令の結果についての詳細説明は、図9-6 を参照してください。

結果の条件コード:

- 0 ソースが 0。
- 1 ソースが 0 より小さい。
- 2 ソースが 0 より大きい。
- 3 特殊ケース。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外

プログラミング上の注意:

1. BFP 対 HFP 変換命令の要約を、9-9ページの図9-7 に示してあります。
2. CONVERT BFP TO HFP (THDER) は、完全精度を維持するために、短精度フォーマットの BFP オペランドを長精度フォーマットの HFP オペランドに変換します (短精度から短精度への変換はしません)。後でこの長精度の HFP 結果を短精度オペランドとして使用する際には、余分な変換ステップは必要ありません。

ソース・オペランド (a)	結果
$-\infty \leq a < -H_{max}$	T(-Hmax), cc3
$-H_{max} \leq a \leq -H_{min}$	T(r), cc1
$-H_{min} < a < 0$	T(-0) ¹ , cc1
-0	T(-0), cc0
+0	T(+0), cc0
$0 < a < +H_{min}$	T(+0) ² , cc2
$+H_{min} \leq a \leq +H_{max}$	T(r), cc2
$+H_{max} < a \leq +\infty$	T(+Hmax), cc3
NaN	T(+Hmax), cc3

説明:

- 1 ソースが 0 より小さいことを示す条件コード 1 が設定されます。
- 2 ソースが 0 より大きいことを示す条件コード 2 が設定されます。
- ccn 条件コードは n に設定されます。
- r BFP ソース値 a を HFP フォーマットに変換したときに得られる値。この結果は常に正確です。
- Hmax ターゲット HFP フォーマットで表現可能な数値のうち絶対値が最大の値。
- Hmin ターゲット HFP フォーマットで表現可能な正規化された数値のうち絶対値が最小の値。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。

図 9-6. 結果: CONVERT BFP TO HFP

命令	ニーモニック	ソース		ターゲット			オーバーフロー、アンダーフローの可能性
		フォーマット	有効数字ビット	フォーマット	有効数字ビット	結果	
CONVERT BFP TO HFP	THDER	BFP 短精度	24	HFP 長精度	53-56	正確	なし
	THDR	BFP 長精度	53	HFP 長精度	53-56	正確	あり
CONVERT HFP TO BFP	TBEDR	HFP 長精度	53-56	BFP 短精度	24	丸め	あり
	TBDR	HFP 長精度	53-56	BFP 長精度	53	丸め	なし

図 9-7. BFP と HFP の間の変換命令の要約

CONVERT HFP TO BFP

ニーモニック R_1, M_3, R_2 [RRF]

Op Code	M ₃	////	R ₁	R ₂
0	16	20	24	28 31

ニーモニック 命令コード オペランド
 TBEDR 'B350' 長精度 HFP オペ
 ンド、
 短精度 BFP 結果
 TBDR 'B351' 長精度 HFP オペ
 ンド、
 長精度 BFP 結果

第 2 オペランド (ソース・オペランド) が 16 進浮動小数点 (HFP) フォーマットから 2 進浮動小数点 (BFP) フォーマットに変換され、M₃ フィールドに指定されている丸め方式に従って丸められた結果が、第 1 オペランドの位置に入れられます。ソース・オペランドの符号と絶対値がテストされ、条件コードの設定が決定されます。

M₃ フィールドには、以下のいずれかの丸め方式を指定する修飾子が入っています。

M₃ 丸め方式

- 0 0 方向への丸め
- 1 バイアス付き最近値丸め
- 4 最近値丸め
- 5 0 方向への丸め
- 6 +∞ 方向への丸め
- 7 -∞ 方向への丸め

0、1、または 4-7 以外の修飾子は無効です。

結果の符号は第 2 オペランドの符号です。第 2 オペランドの符号ビットが 1 で、他のビットがすべて 0 である場合は、結果も 1 の後にすべて 0 を付加したものになります。

この命令の結果についての詳細説明は、9-10ページの図 9-8 を参照してください。

M₃ フィールドには有効な修飾子を指定しなければなりません。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 ソースが 0。
- 1 ソースが 0 より小さい。
- 2 ソースが 0 より大きい。
- 3 特殊ケース。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外

プログラミング上の注意:

1. HFP 対 BFP 変換命令の要約を、図9-7 に示してあります。
2. 短精度 BFP 数への変換を行うには、オペランドが長精度フォーマットであることが必要です。短精度 HFP オペランドは、必ずレジスターの右半分をクリアして、長精度に拡張する必要があります。これにより、短精度 HFP オペランドを BFP への変換のためにレジスターにロードする前に、レジスター全体がクリアされます。これで、前回の使用時から残されているデータが原因で、BFP 内に再現不能な丸め誤差が起きるのを防ぐことができます。

ソース・オペランド (a)	結果
$a < -N_{max}$	この図のパート 2 を参照。
$-N_{max} \leq a \leq -N_{min}$	T(r), cc1
$-N_{min} < a \leq -D_{min}$	T(d), cc1
$-D_{min} < a < 0$	T(d) ¹ , cc1
-0	T(-0), cc0
+0	T(+0), cc0
$0 < a < +D_{min}$	T(d) ² , cc2
$+D_{min} \leq a < +N_{min}$	T(d), cc2
$+N_{min} \leq a \leq +N_{max}$	T(r), cc2
$+N_{max} < a$	この図のパート 2 を参照。

図 9-8 (1/2). 結果: CONVERT HFP TO BFP

ソース・オペ ランド (a)	M3 に指定されている丸め方式の場合の結果				
	バイアス付き最近 値丸め	最近値丸め	0 方向への丸め	+∞ 方向への丸め	-∞ 方向への丸め
$a < -N_{max}$	T(-∞), cc3	T(-∞), cc3	T(-N _{max}), cc3	T(-N _{max}), cc3	T(-∞), cc3
$+N_{max} < a$	T(+∞), cc3	T(+∞), cc3	T(+N _{max}), cc3	T(+∞), cc3	T(+N _{max}), cc3

説明:

1 この場合は、丸めの結果が 0 であっても、条件コード 1 が設定されます。

2 この場合は、丸めの結果が 0 であっても、条件コード 2 が設定されます。

ccn 条件コードは n に設定されます。

d HFP ソース値が、M3 フィールドに指定されている丸め方式を使用してターゲットのフォーマットに丸められたときに得られる非正規化値。

r HFP ソース値が、M3 フィールドに指定されている丸め方式を使用してターゲットのフォーマットに丸められたときに得られる値。

Dmin ターゲット BFP フォーマットで表現可能な非正規化数のうち絶対値が最小の値。

Nmax ターゲット BFP フォーマットで表現可能な有限数のうち絶対値が最大の値。

Nmin ターゲット BFP フォーマットで表現可能な正規化数のうち絶対値が最小の値。

T(x) ターゲット・オペランドの位置に値 x が入れられます。

図 9-8 (2/2). 結果: CONVERT HFP TO BFP

LOAD

ニーモニク 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニク 1 命令コード オペランド
LER '38' 短精度
LDR '28' 長精度

ニーモニク 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニク 2 命令コード オペランド
LXR 'B365' 拡張精度

ニーモニク 3 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20 31

ニーモニク 3 命令コード オペランド
LE '78' 短精度
LD '68' 長精度

第2オペランドが、変更なしで第1オペランドの位置に入れられます。

この操作は、第2オペランドの内容を検査せずに行われます。したがって、演算例外が認識されることはありません。

LXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: LE および LD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外 (LXR のみ)

LOAD ZERO

ニーモニク R₁ [RRE]

Op Code	////////	R ₁	////
0	16	24	28 31

ニーモニク 命令コード オペランド
LZER 'B374' 短精度
LZDR 'B375' 長精度
LZXR 'B376' 拡張精度

第1オペランドのすべてのビットが0に設定されます。

LZXR の場合は、R₁ フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外 (LZXR のみ)

プログラミング上の注意: LOAD ZERO は、レジスターのすべてのビットを0に設定し、その結果は、HFP フォーマットでも BFP フォーマットでも正の0になります。

STORE

ニーモニク R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20 31

ニーモニク 命令コード オペランド
STE '70' 短精度
STD '60' 長精度

第1オペランドが、無変更のままで記憶域内の第2オペランド位置に入れられます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- データ例外 (DXC 1)、AFP レジスター例外

すべての浮動小数点命令の要約

以後のページの図 9-9 から 9-13 までには、オペランド・フォーマットのカテゴリおよび操作のタイプ (主としてレジスター対レジスター操作およびレジスター対記憶域操作) 別に分けて、すべての浮動小数点命令を示してあります。9-12ページの図9-9 は、浮動小数点サポート命令を示しています。9-13ページの図9-10 は、すべてのオペランドが同じ長さを持つ BFP 命令および HFP 命令を示しています。9-14ページの図9-11 は、結果がソース・オペランドより長くなる BFP 命令および

HFP 命令を示しています。9-14ページの図9-12 は、結果がソース・オペランドより短くなる BFP 命令および HFP 命令を示しています。9-14ページの図9-13 は、その他の命令 (FPC レジスターに対して働くものも含む) を示しています。CONVERT FROM FIXED および CONVERT TO FIXED 命令は、固定小数点フォーマットと浮動小数点フォーマットの間の変換を行います。図では、32 ビット固定小数点オペランドに関する項目と 32 ビット短精度オペランドに関する項目を同じ欄の中に組み合わせて示してあり、64 ビット固定小数点オペランドに関する項目と 64 ビット長精度オペランドに関する項目を組み合わせて示してあります。

命令名	短精度 (32)		長精度 (64)		拡張精度 (128)	32 から 64 へ	64 から 32 へ
	R-R	R-S	R-R	R-S	R-R	R-R	R-R
CONVERT BFP TO HFP			THDR			THDER	
CONVERT HFP TO BFP			TBDR				TBEDR
LOAD	LER	LE	LDR	LD	LXR		
LOAD ZERO	LZER		LZDR		LZXR		
STORE		STE		STD			
説明: R-R レジスターとレジスターの間の操作。 R-S レジスターと記憶域の間の操作。							

図 9-9. 浮動小数点サポート命令

命令名	HFP 命令					BFP 命令				
	短精度 (32)		長精度 (64)		拡張精度 (128)	短精度 (32)		長精度 (64)		拡張精度 (128)
	R-R	R-S	R-R	R-S	R-R	R-R	R-S	R-R	R-S	R-R
ADD						AEBR	AEB	ADBR	ADB	AXBR
ADD NORMALIZED	AER	AE	ADR	AD	AXR					
ADD UNNORMALIZED	AUR	AU	AWR	AW						
COMPARE	CER	CE	CDR	CD	CXR	CEBR	CEB	CDBR	CDB	CXBR
COMPARE AND SIGNAL						KEBR	KEB	KDBR	KDB	KXBR
CONVERT FROM FIXED ¹	CEFR		CDGR			CEFBR		CDGBR		
CONVERT TO FIXED ¹	CFER		CGDR			CFEBR		CGDBR		
DIVIDE	DER	DE	DDR	DD	DXR	DEBR	DEB	DDBR	DDB	DXBR
DIVIDE TO INTEGER						DIEBR		DIDBR		
HALVE	HER		HDR							
LOAD AND TEST	LTER		LTDR		LTXR	LTEBR		LTDBR		LTXBR
LOAD COMPLEMENT	LCER		LCDR		LCXR	LCEBR		LCDBR		LCXBR
LOAD FP INTEGER	FIER		FIDR		FIXR	FIEBR		FIDBR		FIXBR
LOAD NEGATIVE	LNER		LNDR		LNXR	LNEBR		LNDBR		LNXBR
LOAD POSITIVE	LPER		LPDR		LPXR	LPEBR		LPDBR		LPXBR
MULTIPLY	MEER	MEE	MDR	MD	MXR	MEEBR	MEEB	MDBR	MDB	MXBR
MULTIPLY AND ADD						MAEBR	MAEB	MADBR	MADB	
MULTIPLY AND SUBTRACT						MSEBR	MSEB	MSDBR	MSDB	
SQUARE ROOT	SQER	SQE	SQDR	SQD	SQXR	SQEBR	SQEB	SQDBR	SQDB	SQXBR
SUBTRACT						SEBR	SEB	SDBR	SDB	SXBR
SUBTRACT NORMALIZED	SER	SE	SDR	SD	SXR					
SUBTRACT UNNORMALIZED	SUR	SU	SWR	SW						
TEST DATA CLASS						TCEB		TCDB		TCXB
説明: ¹ この命令は混合長オペランドも持っています。 R-R レジスターとレジスターの間の操作。 R-S レジスターと記憶域の間の操作。										

図 9-10. すべてのオペランドが同じ長さである BFP および HFP 命令

命令名	HFP 命令						BFP 命令					
	32 から 64 へ		64 から 128 へ		32 から 128 へ		32 から 64 へ		64 から 128 へ		32 から 128 へ	
	R-R	R-S	R-R	R-S	R-R	R-S	R-R	R-S	R-R	R-S	R-R	R-S
CONVERT FROM FIXED	CDFR		CXGR		CXFR		CDFBR		CXGBR		CXFBR	
CONVERT TO FIXED	CGER						CGEBR					
LOAD LENGTHENED	LDER	LDE	LXDR	LXD	LXER	LXE	LDEBR	LDEB	LXDDBR	LXDB	LXEGBR	LXEB
MULTIPLY	MDER	MDE	MXDR	MXD			MDEBR	MDEB	MXDBR	MXDB		
説明: R-R レジスターとレジスターの間の操作。 R-S レジスターと記憶域の間の操作。												

図 9-11. 結果がソースより長い BFP および HFP 命令

命令名	HFP 命令			BFP 命令		
	64 から 32 へ	128 から 64 へ	128 から 32 へ	64 から 32 へ	128 から 64 へ	128 から 32 へ
	R-R	R-R	R-R	R-R	R-R	R-R
CONVERT FROM FIXED	CEGR			CEGBR	CGXBR	CFXBR
CONVERT TO FIXED	CFDR	CGXR	CFXR	CFDBR	CGXBR	CFXBR
LOAD ROUNDED	LEDR	LDXR	LEXR	LEDBR	LDXBR	LEXBR
説明: R-R レジスターとレジスターの間の操作。						

図 9-12. 結果がソースより短い BFP および HFP 命令

命令名	ニーモニック
EXTRACT FPC	EFPC
LOAD FPC	LFPC
SET FPC	SFPC
SET ROUNDING MODE	SRNM
STORE FPC	STFPC

図 9-13. その他の BFP 命令

第10章 制御命令

BRANCH AND SET AUTHORITY	10-5	PROGRAM TRANSFER	10-72
BRANCH AND STACK	10-9	PURGE ALB	10-78
BRANCH IN SUBSPACE GROUP	10-12	PURGE TLB	10-78
COMPARE AND SWAP AND PURGE	10-17	RESET REFERENCE BIT EXTENDED	10-78
DIAGNOSE	10-18	RESUME PROGRAM	10-79
EXTRACT AND SET EXTENDED		SET ADDRESS SPACE CONTROL FAST	10-81
AUTHORITY	10-19	SET CLOCK	10-83
EXTRACT PRIMARY ASN	10-19	SET CLOCK COMPARATOR	10-84
EXTRACT SECONDARY ASN	10-20	SET CLOCK PROGRAMMABLE FIELD	10-84
EXTRACT STACKED REGISTERS	10-20	SET CPU TIMER	10-84
EXTRACT STACKED STATE	10-22	SET PREFIX	10-85
INSERT ADDRESS SPACE CONTROL	10-24	SET PSW KEY FROM ADDRESS	10-85
INSERT PSW KEY	10-25	SET SECONDARY ASN	10-86
INSERT STORAGE KEY EXTENDED	10-25	SET STORAGE KEY EXTENDED	10-89
INSERT VIRTUAL STORAGE KEY	10-26	SET SYSTEM MASK	10-89
INVALIDATE PAGE TABLE ENTRY	10-27	SIGNAL PROCESSOR	10-89
LOAD ADDRESS SPACE PARAMETERS	10-28	STORE CLOCK COMPARATOR	10-91
LOAD CONTROL	10-37	STORE CONTROL	10-91
LOAD PSW	10-37	STORE CPU ADDRESS	10-91
LOAD PSW EXTENDED	10-38	STORE CPU ID	10-92
LOAD REAL ADDRESS	10-39	STORE CPU TIMER	10-93
LOAD USING REAL ADDRESS	10-44	STORE FACILITY LIST	10-93
MODIFY STACKED STATE	10-44	STORE PREFIX	10-93
MOVE PAGE	10-46	STORE REAL ADDRESS	10-94
MOVE TO PRIMARY	10-48	STORE SYSTEM INFORMATION	10-95
MOVE TO SECONDARY	10-48	STORE THEN AND SYSTEM MASK	10-105
MOVE WITH DESTINATION KEY	10-50	STORE THEN OR SYSTEM MASK	10-105
MOVE WITH KEY	10-50	STORE USING REAL ADDRESS	10-105
MOVE WITH SOURCE KEY	10-52	TEST ACCESS	10-106
PAGE IN	10-53	TEST BLOCK	10-108
PAGE OUT	10-54	TEST PROTECTION	10-110
PROGRAM CALL	10-55	TRACE	10-113
PROGRAM RETURN	10-68	TRAP	10-114

この章には、入出力命令を除き、本書で説明しているすべての特権命令と準特権命令を収めてあります。入出力命令については、第14章、『入出力命令』で説明します。

特権命令は、CPU が監視プログラム状態にあるときに限り実行できます。問題プログラム状態にあるときに特権命令を実行しようとする、特権命令例外が起きます。

準特権命令は、所定の権限要件を満たしていれば、問題プログラム状態にあるときでも実行できる命令です。権限要件が満たされていないときに、問題プログラム状態で準特権命令を実行しようとする、どの要件に違反しているかに応じて、特権命令例外か、または他のプログラム割り込み条件が起きます。問題プログラム状態のときに特権命令例外を起こす原因となる要件は、監視プログラム状態で実行する場合には適用されません。

10-2ページの図10-1 に、制御命令とそれぞれのニーモニック、フォーマット、および命令コードを示します。

この図には、ESA/390 に比べて z/Architecture で新しく追加された命令、条件コードが設定される場合、アクセス・レジスタを指定する命令フィールド、および、オペランド指定、データ、または結果でプログラム割り込みを起こすような例外条件も示してあります。

図10-1 では、z/Architecture で新規に加わった命令は「N」で示されています。

命令のオペランドが両方とも 32 ビット・オペランドの場合は、その命令のニーモニックにはオペランドの長さを示す文字は含まれていません。名前が同じで 64 ビット・オペランドの命令の場合は、ニーモニックに「G」の文字が含まれています。図10-1 では、32 ビット・オペランドを持つ命令と、それと同名でニーモニックに

「G」が追加されている別の命令がある場合は、最初の命令には名前の後に「(32)」が付き、もう 1 つの命令には名前の後に「(64)」が付いています。

例外的な状況に対する処理について特別な規則がある制御命令については、「特別な条件」というセクションを設けてあります。そのセクションには、終了のしかたが変化する命令についてのみ、終了のタイプ (抑止、無効化、または完了) が示されています。

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、LOAD PSW では、LPSW がニーモニックで、D₂(B₂) がオペランド指定です。

名前	ニーモニック	特性						命令コード
		RRE	Q	SO	B	R ₂	MD	
BRANCH AND SET AUTHORITY BRANCH AND STACK BRANCH IN SUBSPACE GROUP COMPARE AND SWAP AND PURGE DIAGNOSE	BSA BAKR BSG CSP	RRE RRE RRE RRE C DM	Q A ¹ A ¹ A ¹ P A ¹ SP P DM	SO T Z ⁵ T SO T \$	B B ST ST	R ₂ R ₂ MD	B25A B240 B258 B250 83	
EXTRACT AND SET EXTENDED AUTHORITY EXTRACT PRIMARY ASN EXTRACT STACKED REGISTERS (32) EXTRACT STACKED REGISTERS (64) EXTRACT STACKED STATE	ESEA EPAR EREG EREGG ESTA	RRE N RRE RRE RRE N RRE C	P Q A ¹ A ¹ A ¹ SP	SO SE SE SE		U ₁ U ₂ U ₁ U ₂	B99D B226 B249 B90E B24A	
EXTRACT SECONDARY ASN INSERT ADDRESS SPACE CONTROL INSERT PSW KEY INSERT STORAGE KEY EXTENDED INSERT VIRTUAL STORAGE KEY	ESAR IAC IPK ISKE IVSK	RRE RRE C S RRE RRE	Q Q Q P A ¹ Q A ¹	SO SO G2 SO		R ₂	B227 B224 B20B B229 B223	
INVALIDATE PAGE TABLE ENTRY LOAD ADDRESS SPACE PARAMETERS LOAD CONTROL (32) LOAD CONTROL (64) LOAD PSW	IPTE LASP LCTL LCTLG LPSW	RRE SSE C RS RSE N S L	P A ¹ P A ¹ SP P A SP P A SP P A SP	SO \$ ¢		B ₁ B ₂ B ₂	B221 E500 B7 EB2F 82	
LOAD PSW EXTENDED LOAD REAL ADDRESS (32) LOAD REAL ADDRESS (64) LOAD USING REAL ADDRESS (32) LOAD USING REAL ADDRESS (64)	LPSWE LRA LRAG LURA LURAG	S L N RX C RXE C N RRE RRE N	P A SP P A ¹ P A ¹ P A ¹ SP P A ¹ SP	SO ¢		B ₂ BP BP	B2B2 B1 E303 B24B B905	
MODIFY STACKED STATE MOVE PAGE MOVE TO PRIMARY MOVE TO SECONDARY MOVE WITH DESTINATION KEY	MSTA MVPG MVCP MVCS MVCDK	RRE RRE C SS C SS C SSE	A ¹ SP Q A SP Q A Q A Q A	SE G0 ¢ ¢ GM	ST ST ST ST ST	R ₁ R ₂ B ₁ B ₂	B247 B254 DA DB E50F	
MOVE WITH KEY MOVE WITH SOURCE KEY PAGE IN PAGE OUT PROGRAM CALL	MVCK MVCSK PGIN PGOUT PC	SS C SSE RRE C ES RRE C ES S	Q A Q A P A ¹ P A ¹ Q A ¹	GM ¢ ¢ ¢ GM	ST ST ST ST	B ₁ B ₂ B ₁ B ₂	D9 E50E B22E B22F B218	

図 10-1 (1/4). 制御命令の要約

名前	二 モ ニ ック	特性						命 令 コ ード
PROGRAM RETURN PROGRAM TRANSFER PURGE ALB PURGE TLB RESET REFERENCE BIT EXTENDED	PR PT PALB PTLB RRBE	E L RRE RRE S RRE C	Q A ¹ SP Q A ¹ SP P P P A ¹	Z ⁴ T Z ² T ¢ ¢ \$ \$		B ST B	0101 B228 B248 B20D B22A	
RESUME PROGRAM SET ADDRESS SPACE CONTROL SET ADDRESS SPACE CONTROL FAST SET CLOCK SET CLOCK COMPARATOR	RP SAC SACF SCK SCKC	S L S S S C S	Q A SP Q SP Q SP P A SP P A SP	WE T SW ¢ SW		B B ₂ B ₂	B277 B219 B279 B204 B206	
SET CLOCK PROGRAMMABLE FIELD SET CPU TIMER SET PREFIX SET PSW KEY FROM ADDRESS SET SECONDARY ASN	SCKPF SPT SPX SPKA SSAR	E S S S RRE	P SP P A SP P A SP Q A ¹	G0 \$ Z ³ T ¢		B ₂ B ₂	0107 B208 B210 B20A B225	
SET STORAGE KEY EXTENDED SET SYSTEM MASK SIGNAL PROCESSOR STORE CLOCK COMPARATOR STORE CONTROL (32)	SSKE SSM SIGP STCKC STCTL	RRE S RS C S RS	P A ¹ P A SP P P A SP P A SP	¢ SO \$		B ₂ ST B ₂ B ₂	B22B 80 AE B207 B6	
STORE CONTROL (64) STORE CPU ADDRESS STORE CPU ID STORE CPU TIMER STORE FACILITY LIST	STCTG STAP STIDP STPT STFL	RSE N S S S S N3	P A SP P A SP P A SP P A SP P			ST ST ST ST	B ₂ B ₂ B ₂ B ₂ B2B1	
STORE PREFIX STORE REAL ADDRESS STORE SYSTEM INFORMATION STORE THEN AND SYSTEM MASK STORE THEN OR SYSTEM MASK	STPX STRAG STSI STNSM STOSM	S SSE N S C SI SI	P A SP P A ¹ P A SP P A P A SP	GM		ST ST ST ST ST B ₁ B ₂ B ₁	B ₂ B ₁ BP B ₂ AC AD	
STORE USING REAL ADDRESS (32) STORE USING REAL ADDRESS (64) TEST ACCESS TEST BLOCK TEST PROTECTION	STURA STURG TAR TB TPROT	RRE RRE N RRE C RRE C SSE C	P A ¹ SP P A ¹ SP A ¹ P A ¹ P A ¹	II \$ G0		SU SU U ₁ B ₁	B246 B925 B24C B22C E501	
TRACE (32) TRACE (64) TRAP TRAP	TRACE TRACG TRAP2 TRAP4	RS RSE N E S	P A SP P A SP A A	T ¢ T ¢ SO T SO T		B ₂ B ₂ B ST B ST	99 EB0F 01FF B2FF	

図 10-1 (2/4). 制御命令の要約

説明:

¢	逐次化とチェックポイント同期化を起こします。
¢2	アンスタックする状態エントリーがプログラム呼び出し状態エントリーであるときに、逐次化とチェックポイント同期化を起こします。
\$	逐次化を起こします。
A	論理アドレスについてのアクセス例外。
A1	アクセス例外。すべてのアクセス例外が起こるわけではありません。詳細については、命令の説明を参照してください。
B	PER ブランチ・イベント。
B1	アクセス・レジスター・モードでは、B1 フィールドはアクセス・レジスターを指定します。
B2	アクセス・レジスター・モードでは、B2 フィールドはアクセス・レジスターを指定します。
BP	PSW ビット 16 および 17 の値が 2 進数 01 のとき、B2 フィールドはアクセス・レジスターを指定します。
C	条件コードが設定されます。
DM	モデルによっては、DIAGNOSE は各種のプログラム例外を生成し、条件コードを変更することがあります。
ES	拡張記憶機構。
FC	アクセス・レジスターの指定は、命令の機能コードに応じて決まります。
G0	命令の実行時に、汎用レジスター 0 が暗黙的に使用されます。
G2	命令の実行時に、汎用レジスター 2 が暗黙的に使用されます。
GM	命令の実行時に、複数の汎用レジスターが暗黙的に使用されます。 MOVE WITH DESTINATION KEY、MOVE WITH SOURCE KEY、および STORE SYSTEM INFORMATION の場合は、汎用レジスター 0 および 1。PROGRAM CALL の場合は、汎用レジスター 3、4、および 14。
II	割り込み可能命令
L	新しい条件コードがロードされます。
MD	アクセス・レジスター・モードのときにアクセス・レジスターを指すかどうかは、モデルによって異なります。
N	z/Architecture の新規の命令で、ESA/390 にはないものです。
N3	z/Architecture の新規の命令で、ESA/390 にも追加されたものです。
P	特権命令例外。
Q	準特権命令についての特権命令例外。
R1	アクセス・レジスター・モードでは、R1 フィールドはアクセス・レジスターを指定します。
R2	アクセス・レジスター・モードでは、R2 フィールドはアクセス・レジスターを指定します。
RRE	RRE 命令フォーマット。
RS	RS 命令フォーマット。
RSE	RSE 命令フォーマット。
RX	RX 命令フォーマット。
S	S 命令フォーマット。
SE	特殊操作例外、スタック空例外、スタック指定例外、およびスタック・タイプ例外。
SF	特殊操作例外、スタック一杯例外、およびスタティック指定例外。
SI	SI 命令フォーマット。
SO	特殊操作例外。
SP	指定例外。
SS	SS 命令フォーマット。
SSE	SSE 命令フォーマット。
ST	PER 記憶域変更イベント。
SU	PER 実アドレスを使用した格納イベント。
SW	特殊操作例外およびスペース切り替えイベント。
T	トレース例外 (トレース・テーブル、アドレッシング、および低アドレス保護を含む)。

図 10-1 (3/4). 制御命令の要約

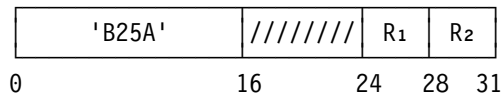
説明 (続き):

- U₁ R₁ フィールドは無条件にアクセス・レジスターを指します。
- U₂ R₂ フィールドは無条件にアクセス・レジスターを指します。
- WE スペース切り替えイベント。
- Z¹ PROGRAM CALL に関するその他の例外およびイベント (これには、ASX 変換例外、EX 変換例外、LX 変換例外、PC 変換指定例外、特殊操作例外、スタック一杯例外、スタック指定例外、およびスペース切り替えイベントが含まれます)。
- Z² PROGRAM TRANSFER に関するその他の例外およびイベント (これには、AFX 変換例外、ASX 変換例外、1 次権限例外、特殊操作例外、およびスペース切り替えイベントが含まれます)。
- Z³ SET SECONDARY ASN に関するその他の例外 (これには、AFX 変換例外、ASX 変換例外、2 次権限例外、および特殊操作例外が含まれます)。
- Z⁴ PROGRAM RETURN に関するその他の例外およびイベント (これには、AFX 変換例外、ASX 変換例外、2 次権限例外、特殊操作例外、スタック空例外、スタック操作例外、スタック指定例外、スタック・タイプ例外、およびスペース切り替えイベントが含まれます)。
- Z⁵ BRANCH AND STACK に関するその他の例外 (これには、特殊操作例外、スタック一杯例外、およびスタック指定例外が含まれます)。

図 10-1 (4/4). 制御命令の要約

BRANCH AND SET AUTHORITY

BSA R₁,R₂ [RRE]

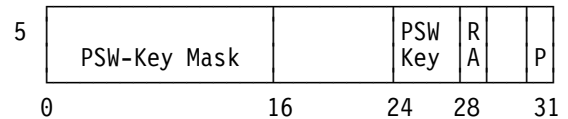


ディスパッチ可能単位が、基本権限状態にあり、24 ビットまたは 31 ビットのアドレッシング・モードである場合: 現 PSW のビット 32 および 97-127、基本アドレッシング・モード・ビット、および更新済み命令アドレスのビット 33-63 が、ディスパッチ可能単位制御テーブル (DUCT) に保管されます。PSW キー・マスク (PKM)、PSW キー、および問題プログラム状態ビットも DUCT に保管されます。PKM および PSW キーが、汎用レジスター R₁ の内容を使用して置き換えられます。問題プログラム状態ビットが 1 に設定されます。PSW のビット 32 および 97-127 が汎用レジスター R₂ の内容を使用して置き換えられます。そして、ディスパッチ可能単位は縮小権限状態になります。64 ビット・アドレッシング・モードの場合のアクションは、現 PSW のビット 64-127 が DUCT に保管され、汎用レジスター R₂ を使用して置き換えられ、PSW のビット 32 は保管も置き換えもされないという点以外は、同じです。

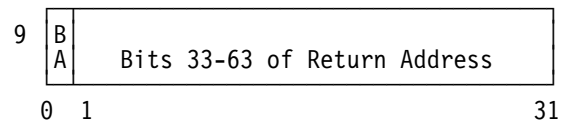
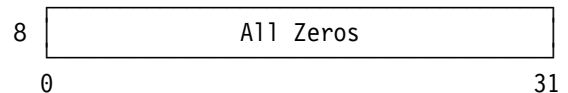
ディスパッチ可能単位が、縮小権限状態で、24 ビットまたは 31 ビットのアドレッシング・モードである場合: R₁ が 0 以外であれば、現 PSW のビット 32 および 97-127 が汎用レジスター R₁ に保管されます。PSW および PKM のビット 32 および 97-127、PSW キー、および問題プログラム状態ビットが、DUCT に保管されている値で置き換えられます。そして、ディスパッチ可

能単位は基本権限状態になります。64 ビット・アドレッシング・モードの場合のアクションは、R₁ が 0 でないときに現 PSW のビット 64-127 が汎用レジスター R₁ に保管され、PSW のビット 32 は保管も置き換えもされないという点以外は、同じです。

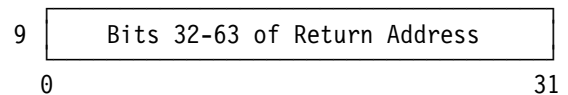
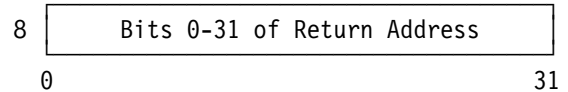
この命令では、DUCT のワード 5、8、および 9 が使用されます。これらのワードの内容は以下のとおりです。



24 ビットまたは 31 ビットのアドレッシング・モードのとき



64 ビット・アドレッシング・モードのとき



DUCT のワード 5、8、および 9 のフィールドは、以下のように割り振られています。

PSW キー・マスク: 割り込み 5 のビット位置 0-15 には、PSW キー・マスク (PKM)、つまり制御レジスター 3 のビット 32-47 が入ります。この値は、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されているものです。PKM は、縮小権限状態で実行される BRANCH AND SET AUTHORITY により、制御レジスター 3 に復元されます。

PSW キー: ワード 5 のビット位置 24-27 には、PSW キー (PSW のビット 8-11) が入ります。これは、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されている値です。PSW キーは、縮小権限状態で実行される BRANCH AND SET AUTHORITY により、PSW に復元されます。

縮小権限 (RA): ワード 5 のビット 28 が 0 のときは、DUCT に関連したディスパッチ可能単位が基本権限状態にあることを示し、1 のときは、ディスパッチ可能単位が縮小権限状態にあることを示します。ビット 28 は、基本権限状態で実行される BRANCH AND SET AUTHORITY では 1 に設定され、縮小権限状態で実行される BRANCH AND SET AUTHORITY では 0 に設定されます。

問題プログラム状態 (P): ワード 5 のビット 31 には、問題プログラム状態ビット (PSW のビット 15) が入ります。これは、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されている値です。問題プログラム状態ビットは、縮小権限状態で実行される BRANCH AND SET AUTHORITY により、PSW に復元されます。

基本アドレッシング・モード (BA): 24 ビットまたは 31 ビットのアドレッシング・モードでは、ワード 9 のビット位置 0 には、基本アドレッシング・モード・ビット (PSW のビット 32) が入ります。これは、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されている値です。基本アドレッシング・モード・ビットは、縮小権限状態で実行される BRANCH AND SET AUTHORITY により、DUCT から PSW に復元されます。

戻りアドレス: 24 ビットまたは 31 ビットのアドレッシング・モードでは、ワード 9 のビット位置 1-31 には、更新済み命令アドレスのビット 33-63、つまり PSW のビット 97-127 が入ります。これは、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されている値です。DUCT のワード 9 の 1-31 は、縮小権限状態で実行される BRANCH AND

SET AUTHORITY により、PSW のビット位置 97-127 に復元されます。64 ビット・アドレッシング・モードでは、ワード 8 および 9 には、基本権限状態で実行された BRANCH AND SET AUTHORITY により保管されている更新済み命令アドレスが入ります。ワード 8 および 9 の内容は、縮小権限状態で実行される BRANCH AND SET AUTHORITY により、PSW のビット位置 64-127 に復元されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、基本権限状態で保管が行われるときは、ワード 8 にはすべて 0 が格納されます。どのアドレッシング・モードでも、基本権限状態で保管が行われるときは、ワード 5 のビット位置 16-23、29、および 30 には、すべて 0 が格納されます。

縮小権限状態でワード 5 のビット 28 が 0 に設定されているときは、ワード 5、8、および 9 のその他のフィールドはどれも変更されません。

DUCT に対する取り出し参照、格納参照、および更新参照は単一アクセス参照であり、他の CPU からはワード単位のアクセスのように見えます。DUCT のワードがアクセスされる順序は一定していません。

基本権限操作

BRANCH AND SET AUTHORITY が基本権限状態で実行される時、つまり、DUCT 中の縮小権限ビット (RA) が 0 のときは、R₂ は 0 以外でなければなりません。さもないと、特殊操作例外が認識されます。R₁ は、0 でも 0 以外でも構いません。

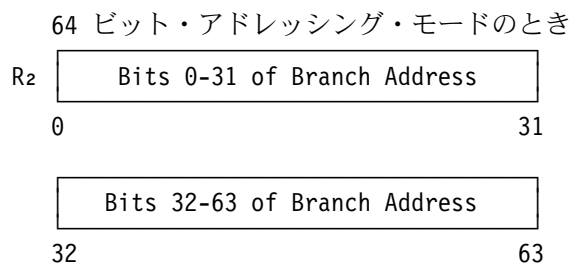
基本権限状態でこの命令の実行が開始されたときの、汎用レジスター R₁ および R₂ のビット位置 32-63 の内容は、以下のようになります。

R ₁	Key Mask		Key	
	32	48	56	60 63

24 ビットまたは 31 ビットのアドレッシング・モードのとき

R ₂	Ignored
	0 31

B A	Bits 33-63 of Branch Address
	32 63



どのアドレッシング・モードでも、汎用レジスタ R1 のビット位置 0-31 の内容は無視されます。24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R2 のビット位置 0-31 の内容は無視されま

す。

24 ビットまたは 31 ビットのアドレッシング・モードでは、DUCT のワード 9 には PSW ビット 32 および 97-127 が保管され、ワード 8 には 0 が格納されます。64 ビット・アドレッシング・モードでは、DUCT のワード 8 および 9 には、PSW ビット 64-127 が保管されます。どのアドレッシング・モードでも、DUCT のワード 5 には、PKM、PSW キー、および問題プログラム状態ビットが保管され、ワード 5 の RA ビットは 1 に設定され、ワード 5 のビット 16-23、29、および 30 は 0 に設定されます。

PSW のビット位置 8-11 には、汎用レジスタ R1 のビット 56-59 が、新 PSW キーとして格納されます。問題プログラム状態では、新 PSW キーは PKM により許可されることが必要です。新 PSW が許可されていない場合は、特権命令例外が認識されます。

PSW に新 PSW キーが入れた後で、汎用レジスタ R1 のビット 32-47 と制御レジスタ 3 の中の PKM との論理積が求められ、その結果が制御レジスタ 3 の中の PKM と置き換わります。

PSW 内の問題プログラム状態ビットは 1 に設定されま

す。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R2 のビット 32 が、新しい基本アドレッシング・モード・ビットとして、PSW のビット位置 32 に格納されます。新しい基本アドレッシング・モードの制御下で、汎用レジスタ R2 のビット 33-63 からブランチ・アドレスが生成され、その結果が、新しい命令アドレスとして PSW のビット位置 64-127 に格納されます。

64 ビット・アドレッシング・モードでは、汎用レジスタ R2 のビット 0-63 からブランチ・アドレスが生成され、新しい命令アドレスとして PSW のビット位置 64-127 に入れます。PSW のビット 32 は変更されません。

汎用レジスタ R1 のビット 48-55 および 60-63 は、将来の拡張に備えて予約されているもので、これらは 0 でなければなりません。さもないと、将来プログラムの動作の互換性が失われることがあります。

縮小権限操作

BRANCH AND SET AUTHORITY が縮小権限状態で実行されるとき、つまり、DUCT の中の縮小権限 (RA) ビットが 1 のときは、R2 は 0 でなければなりません。さもないと、特殊操作例外が認識されます。R1 は、0 でも 0 以外でも構いません。汎用レジスタ R1 および R2 の初期内容は無視されます。

24 ビットまたは 31 ビットのアドレッシング・モードで、R1 が 0 以外である場合は、現 PSW のビット 32 および 97-127、つまり更新済み命令アドレスの基本アドレッシング・モード・ビットとビット 33-63 が、それぞれ、汎用レジスタ R1 のビット位置 32 および 33-63 に入れられ、レジスタのビット 0-31 は変更されません。64 ビット・アドレッシング・モードで R1 が 0 以外であるときは、現 PSW のビット 64-127 が、汎用レジスタ R1 のビット位置 0-63 に入れます。R1 が 0 の場合は、汎用レジスタ 0 は変更されま

せん。

24 ビットまたは 31 ビットのアドレッシング・モードでは、DUCT のワード 9 のビット 0 が PSW のビット位置 32 に入れられ、ワード 9 のビット 1-31 の左側に 33 個の 0 を付加したものが、PSW のビット位置 64-127 に入れます。

64 ビット・アドレッシング・モードでは、DUCT のワード 8 および 9 の内容が PSW のビット位置 64-127 に入れられ、PSW のビット 32 は変更されません。

どのアドレッシング・モードでも、前に述べたように、DUCT から PKM、PSW キー、および問題プログラム状態ビットが復元され、RA ビットは 0 に設定されます。復元された PSW キーが復元された PKM により許可されているかどうかのテストは行われません。

特別な条件

R2 は、基本権限状態では 0 以外で、縮小権限状態では 0 でなければなりません。この規則に違反している場合は、特殊操作例外が認識され、操作は抑止されます。

問題プログラム状態では、基本権限状態でのこの命令の実行は、制御レジスタ 3 の中の PSW キー・マスクにより制御されます。PSW キー・マスク内の、設定すべき PSW キー値に対応するビットが 1 であれば、命令は正常に実行されます。PSW キー・マスク内の選択されたビットが 0 のときは、特権命令例外が認識されます。監視プログラム状態では、PSW キーの値は何であっても有効です。

この命令で行われるアクセスには、キー制御保護は適用されません。低アドレス保護は適用されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、DUCT のワード 9 の内容が PSW にロードされる前に、その内容の妥当性検査は行われません。ただし、新たにロードした PSW のビット位置 32 が 0 であり、ビット位置 97-103 がすべて 0 ではない場合は、ロード後に指定例外が認識され、プログラム割り込みが起きます。この場合は、操作は完了し、結果の命令長コードは 0 になります。この場合の指定例外は、この命令の中でプログラム例外としてリストされますが、これについては、6-9 ページの『早期例外認識』で説明されています。この例外は、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-9 ページの図 10-2 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (ディスパッチ可能単位制御テーブル)
- 特権命令例外 (基本権限操作において、問題プログラム状態のときに選択された PSW キー・マスク・ビットが 0)
- 保護例外 (低アドレス保護、ディスパッチ可能単位制御テーブル)
- 特殊操作例外
- 指定例外
- トレース例外

プログラミング上の注意:

1. BRANCH AND SET AUTHORITY を使用すると、「現 1 次へ」形式の PROGRAM TRANSFER (PT-cp) および基本 (非スタッキング) PROGRAM CALL (PC-cp) が置き換えられるため、パフォーマンスが向上します。PT-cp および PC-cp は、PSW キー・マスク (PKM) の権限を縮小するため、または、PT-cp により作られた呼び出し側リネージのときに監視プログラム状態から問題プログラム状態に変更し、そして PC-cp により作られた戻りリネージのときに PKM 権限または監視プログラム状態を復元するために、単一のアドレス・スペース内でしばしば使用されます。さらに、BRANCH AND SET AUTHORITY の PSW キー設定操作を、SET PSW KEY FROM ADDRESS 命令の代わりに使用することができます。BRANCH AND SET AUTHORITY は、ブランチと PSW キー設定を組み合わせた操作を行うので、この命令を使用して、取り出し保護プログラムからの (またはそこへの) ブランチを行うときに、PSW キーを変更することができます。
2. 1 つのディスパッチ可能単位では、基本権限状態と縮小権限状態をそれぞれ 1 つだけ使用できます。BRANCH AND SET AUTHORITY をネストして使用すること、つまり複数のサブルーチン・レベルで使用することはできません。R2 が、基本権限状態では 0 以外で、縮小権限状態では 0 でなければならないという要件があるので、前回この命令を基本権限状態で使用したためにディスパッチ可能単位がすでに縮小権限状態になっているときに、BRANCH AND SET AUTHORITY を基本権限状態で使用しようとした場合に、その状況が検出されます。
3. 基本権限状態での BRANCH AND SET AUTHORITY は、現在のアドレッシング・モードが拡張アドレッシング・モード (64 ビット) か基本アドレッシング・モード (24 ビットまたは 31 ビット) かの指示を、DUCT に保管しません。この命令では、基本権限状態または縮小権限状態のいずれの場合も、拡張アドレッシング・モードと基本アドレッシング・モードとの間の切り替えは行われません。縮小権限状態では、DUCT のワード 8 および 9 の内容は、現アドレッシング・モードのみに基づいて解釈されます。31 ビット・アドレッシング・モードで保管が行われ、次に 64 ビット・アドレッシング・モードで復元が行われる場合は、DUCT のワード 9 のビット 0 は、基本アドレッシング・モード・ビットではなくアドレス・ビットとして使用されます。64 ビット・アドレッシング・モード

で保管が行われ、次に 24 ビット・アドレッシング・モードで復元が行われる場合は、命令実行の完了後に、PSW のビット 32 が 0 のときにビット

97-103 が 0 以外になることがあるため、早期指定例外が認識されることがあります。

- | | |
|-------|---|
| 1.-6. | 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7. | 命令の第 2 ハーフワードについてのアクセス例外。 |
| 8.A | トレース例外。 |
| 8.B | ディスパッチ可能単位制御テーブルへのアクセスについての記憶保護例外 (低アドレス保護)。 |
| 8.C.1 | ディスパッチ可能単位制御テーブルへのアクセスについてのアドレッシング例外。 |
| 8.C.2 | 基本権限状態で R_2 が 0 であるため、または縮小権限状態で R_2 が 0 以外であるために起きる特殊操作例外。 |
| 8.C.3 | 選択された PSW キー・マスク・ビットが 0 であるために起きる特権命令例外 (基本権限操作の場合のみ)。 |
| 9. | 新たにロードされた PSW のビット 32 が 0 であるのに対し、ビット 97-103 がすべて 0 ではないために起きる指定例外 (縮小権限操作の場合のみ)。 |

図 10-2. 実行の優先順位: BRANCH AND SET AUTHORITY

4. この命令は、基本権限状態で実行されるか縮小権限状態で実行されるかに応じて、それぞれ BSA-ba または BSA-ra と呼ばれることがあります。

R_1 フィールドが 0 以外である場合は、汎用レジスター R_1 の内容は戻りアドレスと呼ばれるアドレスを指定しています。

BRANCH AND STACK

BAKR R_1, R_2 [RRE]

'B240'	////////	R_1	R_2
0	16	24	28 31

リンケージ・スタック・ブランチ状態エントリーが形成され、現 PSW がその状態エントリーに入れられます。ただし、PER マスクは予測不能であり、アドレッシング・モード・ビットと命令アドレスは、第 1 オペランドに基づいて置き換えられます。次に、現 PSW 内の更新済み命令アドレスが、第 2 オペランドに従って置き換えられます。さらに、現アドレッシング・モード・ビットおよび新命令アドレスを示す値が状態エントリーに入れられ、また、PSW キー・マスク、PASN、SASN、EAX、および、汎用レジスター 0-15 およびアクセス・レジスター 0-15 の内容も、状態エントリーに入れられます。いずれかのオペランドを指している R フィールドが 0 の場合は、そのオペランドに関連付けられているアクションは実行されません。

R_1 が 0 以外であり、汎用レジスター R_1 のビット 63 が 0 であるときは、このレジスターのビット 32 に指定されている基本アドレッシング・モードの制御下で、このレジスターの内容から戻りアドレスが生成されます。アドレッシング・モードは、ビット 32 が 0 なら 24 ビット・モードで、ビット 32 が 1 なら 31 ビット・モードです。PSW が状態エントリーに入れられるときに、レジスターのビット 32 および戻りアドレスが、それぞれ、現 PSW の基本アドレッシング・モード・ビット (ビット 32) および更新済み命令アドレスと置き換わります。状態エントリーに入れられる PSW の拡張アドレッシング・モード・ビット (ビット 31) は、0 に設定されます。現 PSW の内容は変更されません。

R_1 が 0 以外で、汎用レジスター R_1 のビット 63 が 1 であるときは、64 ビット・アドレッシング・モードの制御下で、そのレジスターの内容から戻りアドレスが生成されます。PSW が状態エントリーに入れられるときに、戻りアドレスのビット 0-62 の右側に 0 を 1 個付加したものが、現 PSW 内の更新済み命令アドレスと置き換わります。状態エントリーに入れられる PSW 内のビット 31 および 32 は、1 に設定されます。現 PSW の内容は変更されません。

R1 フィールドが 0 のときは、現 **PSW** が変更されないまま状態エントリーに入れられます (ただし、**PER** マスクは予測不能です)。

次に、**R2** フィールドが 0 以外であるときは、現 **PSW** 内の命令アドレスがブランチ・アドレスで置き換えられます。このブランチ・アドレスは、現アドレッシング・モードの制御下で、汎用レジスター **R2** の内容に基づいて生成されます。**R2** フィールドが 0 のときは、ブランチなしで操作が実行されます。

ブランチ状態エントリーが形成され、5-72ページの『スタッキング・プロセス』の説明に従って、そのエントリーに情報が入れられます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、ブランチ・アドレス (ブランチなしで命令が実行される場合は更新済み命令アドレス) のビット 33-63 が、状態エントリーのバイト 144-151 のビット位置 33-63 に入れられ、現 **PSW** のビット 32 がこれらのバイトのビット位置 32 に入れられ、これらのバイトのビット位置 0-31 には 0 が入れます。

64 ビット・アドレッシング・モードでは、ブランチ・アドレス (ブランチなしで命令が実行される場合は更新済み命令アドレス) のビット 0-62 が、状態エントリーのバイト 144-151 のビット位置 0-62 に入れられ、これらのバイトのビット位置 63 には 1 が入ります。

状態エントリー内のエントリー・タイプ・コードは、2 進数 0001100 です。

リンケージ・スタックへのアクセスには、キー制御保護は適用されませんが、低アドレス保護およびページ保護は適用されます。

特別な条件

CPU は、1 次スペース・モードまたはアクセス・レジスター・モードになっていなければなりません。さもないと、特殊操作例外が認識されます。

スタッキング・プロセスでは、スタック一杯例外またはスタック指定例外が認識されることがあります。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-11ページの図10-3 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (キー制御保護の場合を除き、リンケージ・スタック・エントリーの取り出しまたは格納)
- 特殊操作例外
- スタック一杯例外
- スタック指定例外
- トレース例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 ハーフワードについてのアクセス例外。
- 7.B DAT がオフのため、または CPU が 2 次スペース・モードまたはホーム・スペース・モードであるために起きる特殊操作例外。
- 8.A トレース例外 (R_2 が 0 以外の場合のみ)。
- 8.B.1 現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。
注: 8.B.2 から 8.B.7 までの例外が起きることがあるのは、現リンケージ・スタック・セクション内に十分なフリー・スペースが残っていない場合のみです。
- 8.B.2 現リンケージ・スタック・エントリー内の残余フリー・スペース値が 8 の倍数でないために起きるスタック指定例外。
- 8.B.3 現セクションのトレーラー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。このエントリーはトレーラー・エントリーであるものと見なされ、エントリー・タイプ・フィールドは検査されません。
- 8.B.4 トレーラー・エントリー内の正方向セクション妥当性ビットが 0 であるために起きるスタック一杯例外。
- 8.B.5 次セクションのヘッダー・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。このエントリーはヘッダー・エントリーであるものと見なされ、エントリー・タイプ・フィールドは検査されません。
- 8.B.6 次セクション内に十分なフリー・スペースが残っていないために起きるスタック指定例外。
- 8.B.7 次セクションのヘッダー・エントリーの第 2 ワードについてのアクセス例外 (格納)。例外が何も存在しない場合は、このヘッダーが現エントリーと呼ばれるようになります。
- 8.B.8 現エントリーのエントリー記述子について、および新状態エントリーについてのアクセス例外 (格納)。

図 10-3. 実行の優先順位: BRANCH AND STACK

プログラミング上の注意:

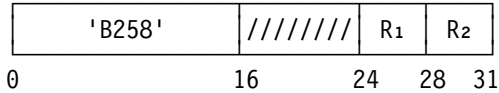
1. BRANCH AND STACK 命令の使用例は、付録 A、『数の表現と命令の使用例』に示されています。
2. どのような場合でも、BRANCH AND STACK が現アドレッシング・モードを変更することはありません。
3. R_1 フィールドが 0 のときは、戻りアドレスは、 R_1 汎用レジスタに指定されているアドレスではなく、次の順番に当たる命令のアドレスになります。その場合、BRANCH AND STACK は、BRANCH AND SAVE の機能と同等のプログラム・リンケージ機能を提供します。
4. R_1 フィールドが 0 でない BRANCH AND STACK は、呼び出し先プログラムのエントリー・

ポイントまたはその近くで使用するものです。呼び出し先プログラムは、24 ビットまたは 31 ビットのアドレッシング・モードで実行されているプログラムからの BRANCH AND LINK (BALR) または BRANCH AND SAVE (BAS または BASR)、任意のアドレッシング・モードで実行されているプログラムからの BRANCH AND SAVE AND SET MODE、または、「グルー・モジュール」内にあって任意のアドレッシング・モードで実行されている BRANCH AND SET MODE 命令により呼び出されます。いずれの場合も、呼び出し側命令の 0 でない R_1 フィールドが BRANCH AND STACK の R_1 フィールドと同じであれば、呼び出し側リンケージでアドレッシング・モードが変更された場合でも BRANCH AND STACK は呼び出し側プログラムのアドレッシング・モードと戻りアドレスを正しく保管するので、後続の PROGRAM

RETURN の実行時には、正しく呼び出し側プログラムに戻ります。

BRANCH IN SUBSPACE GROUP

BSG R₁,R₂ [RRE]



現ディスパッチ可能単位に関連したサブスペース・グループがあり、現 1 次アドレス・スペースがそのグループ内にある場合に、宛先 ASN 第 2 テーブル・エントリー (DASTE) を見つけるために、特殊形式のアクセス・レジスタ変換 (ART) を使用してアクセス・レジスタ R₂ 中のアクセス・リスト・エントリー・トークン (ALET) が変換されます。その DASTE がサブスペース・グループの基本スペースを指定している場合は、制御レジスタ 1 中の 1 次 ASCE (PASCE) が、DASTE 内の ASCE で置き換えられます。DASTE がそのグループのサブスペースの 1 つを指定している場合は、PASCE のビット 0-55 および 58-63 が DASTE 内の ASCE の同じビットで置き換えられます。いずれの場合も、同時に以下のアクションも発生します。

24 ビットまたは 31 ビットのアドレッシング・モードでは、現 PSW のビット 32 および 97-127、つまり更新済み命令アドレスの基本アドレッシング・モード・ビットおよびビット 33-63 が、それぞれ、汎用レジスタ R₁ のビット位置 32 および 33-63 に保管され、レジスタのビット 0-31 は変更されません。続いて、現 PSW 内の命令アドレスの基本アドレッシング・モード・ビットおよびビット 33-63 が、汎用レジスタ R₂ のビット位置 32-63 に従って置き換えられ、レジスタのビット 0-31 は無視されます。

64 ビット・アドレッシング・モードでは、現 PSW のビット 64-127、つまり更新済み命令アドレスが、汎用レジスタ R₁ のビット位置 0-63 に保管されます。続いて、現 PSW 内の命令アドレスが、汎用レジスタ R₂ のビット位置 0-63 に従って置き換えられます。PSW のビット 32 は変更されません。

どのアドレッシング・モードでも、制御レジスタ 7 中の 2 次 ASCE (SASCE) が、制御レジスタ 1 中の新しい PASCE と同じ値に設定されます。さらに、制御レジスタ 3 のビット 48-68 にある 2 次 ASN (SASN) が、制御レジスタ 4 のビット 48-63 にある 1 次 ASN (PASN) と同じに設定されます。R₁ フィー

ルドが 0 の場合は、汎用レジスタ 0 は変更されません。

制御レジスタ 5 のビット 33-57 にある現 1 次 ASTE 起点 (PASTE0) が、ディスパッチ可能単位のサブスペース・グループの基本スペースの ASTE を指している場合は、現 1 次アドレス・スペースはそのサブスペース・グループの中に含まれています。PASTE0 が基本スペース ASTE を指すのは、PASTE0 が、基本 ASTE 起点 (BASTE0)、つまりディスパッチ可能単位制御テーブル (DUCT) のワード 0 のビット 1-25 に等しいときです。PASTE0 が BASTE0 に等しいかどうかを判別するには、PASTE0 を BASTE0 と比較するか、または、制御レジスタ 5 のビット位置 32-63 を DUCT のワード 0 の内容全体と比較することができます。

標準 ART については、5-48 ページの『アクセス・レジスタ変換プロセス』で説明されています。この命令が行う特殊 ART には、標準 ART と比べて以下のような相違点があります。

1. 特殊 ART は、CPU がアクセス・レジスタ・モードになっているかどうかに関係なく実行されません。
2. 変換されている ALET が 16 進数 00000000 (ALET 0 と呼ばれる) である場合は、DASTE は基本スペースの ASTE です。DASTE のビット 0 は無視されます。
3. ALET が 16 進数 00000001 (ALET 1 と呼ばれる) である場合は、DASTE は、ディスパッチ可能単位が BRANCH IN SUBSPACE GROUP を使用して最後に入ったサブスペースの ASTE です。その ASTE は、サブスペース ASTE 起点 (SSASTE0)、つまり DUCT のワード 1 のビット 1-25 に示されているものです。前にどのサブスペースにも入ったことがない場合、つまり SSASTE0 がすべて 0 の場合は、特殊操作例外が認識されません。DASTE のビット 0 が 1 の場合は、ASTE 妥当性例外が認識されます。DASTE 内の ASTE シーケンス番号 (ASTESN) が、DUCT のワード 3 中のサブスペース ASTESN (SSASTESN) に等しくない場合は、ASTE シーケンス例外が認識されます。ALET 1 が原因で見つかった DASTE は、エラーのためにその DASTE が基本スペースの ASTE を示していたとしても、サブスペースを指定しているものと見なされます。つまり、SSASTE0 と BASTE0 の比較は行われません。
4. ALET が ALET 0 および ALET 1 のどちらでもない場合は、標準 ART の場合と同じ方法で、アクセ

ス・リスト・エンタリー (ALE) から起点を求めることにより ASTE が突き止められ、その ASTE が DASTE です。この場合は、標準 ART の場合と同様に、以下のことが行われます。

- ALET のビット 0-6 が 0 でない場合は、ALET 指定例外が認識されます。
- ALE が有効アクセス・リストの外部にあるか、ALE のビット 0 が 1 である場合は、ALEN 変換例外が認識されます。
- DASTE のビット 0 が 1 の場合は、ASTE 妥当性例外が認識されます。
- DASTE 内の ASTE シーケンス番号 (ASTESN) が ALE 内の ASTESN に等しくない場合は、ASTE シーケンス例外が認識されません。

標準 ART の操作と違うのは、ALE 内の ALE シーケンス番号 (ALESN) が ALET 内の ALESN と比較されず、ALE 内の専用ビットは 0 と見なされるという点です。したがって、ALE シーケンス例外および拡張権限例外が起きることはありません。

ALE 内の取り出し専用ビットは無視されます。

ALET が ALET 0 と ALET 1 のどちらでもないときは、ART 索引バッファ (ALB) を使用して特殊 ART を行うことができます。

ALET が ALET 0 および ALET 1 のいずれでもない場合に突き止められる DASTE は、ディスパッチ可能単位に関連したサブスペース・グループの基本スペースの ASTE である可能性があります。ART により ALE から求められた DASTE 起点 (DASTE0) が DUCT 内の BASTE0 に等しい場合は、DASTE は基本スペース ASTE です。DASTE0 が BASTE0 と等しいかどうかを判別するには、DASTE0 を BASTE0 と比較する方法と、DASTE0 の左側に 1 個右側に 6 個の 0 を付加したものを、DUCT のワード 0 の内容全体と比較する方法があります。DASTE が基本スペース ASTE でない場合は、(1) DASTE 内の ASCE のサブスペース・グループ・ビット (ビット 54) が 1 であり、かつ、(2) DASTE が他のサブスペース・グループの基本スペースを指定していなければ、DASTE は、ディスパッチ可能単位のサブスペース・グループのいずれかのサブスペースの ASTE として扱われます。DASTE のワード 0 の基本スペース・ビット (ビット 31) が 1 である場合は、DASTE は他のサブスペースの基本スペースを指定しています。上記の 2 つの条件のいずれかが満たされていない場合は、特殊操作例外が認識されます。

DASTE がサブスペース・グループの基本スペースを指定している場合は、制御レジスタ 1 中の PASCE が、DASTE 内の ASCE で置き換えられます。DASTE がサブスペースの 1 つを指定している場合は、PASCE のビット 0-55 および 58-63 が DASTE 内の ASCE の同じビットで置き換えられ、PASCE の 56 および 57 (記憶域変更イベント・ビットおよびスペース切り替えイベント・ビット) は変更されません。

24 ビットまたは 31 ビットのアドレッシング・モードで、R₁ が 0 以外である場合は、現 PSW のビット 32 および 97-127、つまり更新済み命令アドレスの基本アドレッシング・モード・ビットとビット 33-63 が、それぞれ、汎用レジスタ R₁ のビット位置 32 および 33-63 に入れられ、レジスタのビット 0-31 は変更されません。64 ビット・アドレッシング・モードで R₁ が 0 以外であるときは、現 PSW のビット 64-127 (更新済み命令アドレス) が、汎用レジスタ R₁ のビット位置 0-63 に入れられます。R₁ が 0 の場合は、汎用レジスタ 0 は変更されません。

24 ビットまたは 31 ビットのアドレッシング・モードでは、R₂ が 0 でも 0 以外でも、汎用レジスタ R₂ のビット 32-63 は、新しい基本アドレッシング・モードを示し、ブランチ・アドレスを指しています。レジスタのビット 32 は新しい基本アドレッシング・モードを示し、現 PSW のビット 32 と置き換わります。そして、新しい基本アドレッシング・モードの制御下で、レジスタのビット位置 33-63 の内容からブランチ・アドレスが生成されます。

64 ビット・アドレッシング・モードでは、R₂ が 0 でも 0 以外でも、汎用レジスタ R₂ の内容はブランチ・アドレスを指しています。このブランチ・アドレスは、64 ビット・アドレッシング・モードの制御下で、レジスタの内容に基づいて生成されます。PSW のビット 32 は変更されません。

PSW の新しい値は、アドレッシング・モードに関係なく、汎用レジスタ R₁ の変更前に計算されます。

制御レジスタ 7 中の 2 次 ASCE (SASCE) は、制御レジスタ 1 中の新しい PASCE と同じ値に設定されます。制御レジスタ 3 のビット 48-68 にある 2 次 ASN (SASN) は、制御レジスタ 4 のビット 48-63 にある 1 次 ASN (PASN) と同じに設定されます。

DASTE が基本スペースを指定している場合は、サブスペース・アクティブ・ビット (DUCT のワード 1 のビ

ット 0) は 0 に設定され、ワード 1 のビット 1-31 は変更されません。DASTE が ALET 1 によりサブスペースを指定している場合は、(1) サブスペース・アクティブ・ビットは 1 に設定され、(2) ワード 1 のビット位置 1-25 にある SSASTE0 は変更されず、(3) ワード 1 のビット 26-31 は、0 に設定されるかまたは変更されません。DASTE が ALET 以外の ALET によりサブスペースを指定している場合は、(1) サブスペース・アクティブ・ビットは 1 に設定され、(2) DASTE0 がワード 1 のビット位置 1-25 に SSASTE0 として格納され、(3) ワード 1 のビット位置 26-31 には 0 が格納され、(4) DASTE 内の ASTESN が DUCT のワード 3 に SSASTESN として格納されます。

DUCT に対する取り出し参照、格納参照、および更新参照は単一アクセス参照であり、他の CPU からはワード単位のアクセスのように見えます。DUCT のワードがアクセスされる順序は一定していません。

この操作では、制御レジスター 1 の中の変換パラメーターが変更されるため、ホーム・スペース・モードの場合を除き、事前取り出しされた命令のコピーはすべて破棄されます。

特別な条件

DAT がオンになっていなければなりません。さもないと、特殊操作例外が認識されます。また、現 1 次アドレス・スペースが現ディスパッチ可能単位に関連したサブスペース・グループ内でない場合、または、アクセス・レジスター R₂ 中の ALET が ALET 1 であるのに、これまでにディスパッチ可能単位が BRANCH IN SUBSPACE GROUP を使用してどのサブスペースにも入ったことがない場合、または、使用されている ALET

が ALET 0 および ALET 1 のいずれでもなく、宛先 ASTE がサブスペース・グループの基本スペースまたはサブスペースを指定していない場合も、特殊操作例外が認識されます。

操作の前または後の 1 次スペース切り替えイベント制御ビット (制御レジスター 1 のビット 57) が原因で、スペース切り替えイベント・プログラム割り込みが起きることはありません。

この命令で行われるアクセスには、キー制御保護は適用されません。低アドレス保護は適用されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

『実行の優先順位: BRANCH IN SUBSPACE GROUP』の図に、この命令に関するプログラム例外の認識の優先順位を示してあります。

条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (ディスパッチ可能単位制御テーブル、有効アクセス・リスト指定、アクセス・リスト・エントリー、宛先 ASN 第 2 テーブル・エントリー)
- ALET 指定例外
- ALEN 変換例外
- ASTE シーケンス例外
- ASTE 妥当性例外
- 保護例外 (低アドレス保護、ディスパッチ可能単位制御テーブル)
- 特殊操作例外
- トレース例外

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	DAT がオフであるために起きる特殊操作例外。
8.A	トレース例外。
8.B	ディスパッチ可能単位制御テーブルへのアクセスについての記憶保護例外 (低アドレス保護)。
8.C.1	ディスパッチ可能単位制御テーブルへのアクセスについてのアドレッシング例外。
8.C.2	現 1 次アドレス・スペースが、現ディスパッチ可能単位に関連したサブスペース・グループ内でない (つまり、制御レジスター 5 の中の 1 次 ASTE 起点が、ディスパッチ可能単位テーブル内の基本 ASTE 起点に等しくない) ために起きる特殊操作例外。 注: 例外 8.C.3.A が起きることがあるのは、アクセス・レジスター R ₂ 中のアクセス・リスト・エントリー・トークン (ALET) が ALET 0 である場合のみです。
8.C.3.A	基本 ASTE (ディスパッチ可能単位制御テーブル内の基本 ASTE 起点が指している ASTE) へのアクセスについてのアドレッシング例外。 注: 8.C.3.B.1 - 8.C.3.B.4 の例外が起きることがあるのは、アクセス・レジスター R ₂ 中のアクセス・リスト・エントリー・トークン (ALET) が ALET 1 である場合のみです。
8.C.3.B.1	ディスパッチ可能単位制御テーブル内のサブスペース ASTE 起点が 0 であるために起きる特殊操作例外。
8.C.3.B.2	サブスペース ASTE へのアクセスについてのアドレッシング例外。
8.C.3.B.3	サブスペース ASTE 内のビット 0 が 1 であるために起きる ASTE 妥当性例外。
8.C.3.B.4	サブスペース ASTE 内の ASTE シーケンス番号が、ディスパッチ可能単位制御テーブル内のサブスペース ASTE シーケンス番号に等しくないために起きる ASTE シーケンス例外。 注: 8.C.3.C.1 - 8.C.3.C.9 の例外が起きることがあるのは、アクセス・レジスター R ₂ 中のアクセス・レジスター・エントリー・トークン (ALET) が ALET 0 および ALET 1 以外である場合のみです。
8.C.3.C.1	ALET のビット 0-6 がすべて 0 ではないために起きる ALET 例外。
8.C.3.C.2	有効アクセス・リスト指定へのアクセスについてのアドレッシング例外。
8.C.3.C.3	アクセス・リスト・エントリーがリストの外部にあるために起きる ALEN 変換例外。

図 10-4 (1/2). 実行の優先順位: BRANCH IN SUBSPACE GROUP

- 8.C.3.C.4 アクセス・リスト・エントリーへのアクセスについてのアドレッシング例外。
- 8.C.3.C.5 アクセス・リスト・エントリー内の I ビットが 1 であるために起きる ALEN 変換例外。
- 8.C.3.C.6 宛先 ASTE へのアクセスについてのアドレッシング例外。
- 8.C.3.C.7 宛先 ASTE のビット 0 が 1 であるために起きる ASTE 妥当性例外。
- 8.C.3.C.8 アクセス・リスト・エントリー内の ASTE シーケンス番号 (ASTESN) が、宛先 ASTE 内の ASTESN に等しくないために起きる ASTE シーケンス例外。
- 8.C.3.C.9 宛先 ASTE 起点がディスパッチ可能単位制御テーブル内の基本 ASTE 起点に等しくなく、かつ、(1) 宛先 ASTE 内のアドレス・スペース制御エレメントの中のサブスペース・グループ・ビット (ビット 54) が 0 であるか、または (2) 宛先 ASTE 内の基本スペース・ビット (ビット 31) が 1 であるために起きる特殊操作例外。

図 10-4 (2/2). 実行の優先順位: *BRANCH IN SUBSPACE GROUP*

プログラミング上の注意:

1. 5-10ページの『リンケージ・スタックなしのサブチェーン・リンケージ』の中の **BRANCH IN SUBSPACE GROUP** に関する説明を参照してください。1 つのサブスペース・グループの基本スペースおよび個々のサブスペースのそれぞれについて、別々の ASN 第 2 テーブル・エントリー (ASTE) があることが前提とされます。3-16ページの『アドレス・スペース番号』の中の『プログラミング上の注意』で述べたように、サブスペース用の ASTE は「疑似」ASTE とすることができます。サブスペース用の DAT テーブルで、基本スペース用の DAT テーブルが指しているページのサブセットを指定することにより、そのサブスペースに基本スペースの記憶域のサブセットが含まれるようにすることができます。あるサブスペースの ASTE を指しているアクセス・リスト・エントリーが、ディスパッチ可能単位の 1 次スペースまたはディスパッチ可能単位アクセス・リスト内にあれば、そのディスパッチ可能単位はそのサブスペースにアクセスできません。
2. **BRANCH IN SUBSPACE GROUP** は、基本スペースからサブスペースに、サブスペースから別のサブスペースに、そしてサブスペースから基本スペースに制御を渡すために使用できます。また、この命令は、基本スペースから基本スペースに、またはサブスペースから同じサブスペースに制御を渡すためにも使用できます。
3. **BRANCH IN SUBSPACE GROUP** は、制御レジスター 7 の中の 2 次アドレス・スペース制御エレメント (ASCE) を、制御レジスター 1 の中の新し

い 1 次 ASCE と同じに設定する (そして、制御レジスター 3 の中の 2 次 ASN を制御レジスター 4 の中の 1 次 ASN と同じに設定する) ので、**BRANCH IN SUBSPACE GROUP** により制御が与えられたアドレス・スペース内のプログラムは、そのアドレス・スペースを 2 次アドレス・スペースとすることによって、呼び出し側プログラムのアドレス・スペースにアクセスすることはできません。

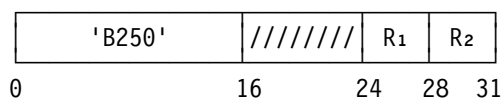
4. あるディスパッチ可能単位が、**BRANCH IN SUBSPACE GROUP** を使用していずれかのサブスペースに入った後、**BRANCH IN SUBSPACE GROUP** を使用して基本スペースに戻っていないとき、そのディスパッチ可能単位を「サブスペース・アクティブ」であると言います。**PROGRAM CALL**、**PROGRAM TRANSFER**、**PROGRAM RETURN**、**SET SECONDARY ASN**、または **LOAD ADDRESS SPACE PARAMETERS** が、ある ASCE を制御レジスター 1 には 1 次 ASCE として格納し、制御レジスター 7 には同じ ASCE を 2 次 ASCE として格納するときに、(1) その ASCE のサブスペース・グループ・ビットがオンであり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) その ASCE が、現ディスパッチ可能単位の基本スペース用の ASN 第 2 テーブル・エントリー (ASTE) から取得したものである場合は、命令 (上記 5 つの命令のいずれか) は、制御レジスター内の ASCE のビット 0-55 および 58-63 を、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えます。上記 5 つの命令に対するサブスペース・グループ機能の影響に関する詳細は、5-58ページの『サブスペース置き換

え操作』のほか、個々の命令の定義の中で説明されています。

5. **BRANCH IN SUBSPACE GROUP (BSG)** を、**PROGRAM CALL (PC)** および **PROGRAM TRANSFER (PT)** または **PROGRAM RETURN (PR)** と一緒に使用すると、予期しない結果が生じることがあります。以下の操作順序を考慮してください。
 - a. 基本スペースでの開始。
 - b. いずれかのサブスペースへの **BSG**。
 - c. サブスペース・グループ内にないアドレス・スペースへの **PC** (第 1 の **PC**)。
 - d. 基本スペースへの **PC** (第 2 の **PC**)。ディスパッチ可能単位はサブスペース・アクティブなので、制御はそのサブスペースに渡されます。
 - e. 基本スペースに戻る **BSG**。
 - f. サブスペース・グループ内にないアドレス・スペースに戻る **PT** または **PR** (第 2 の **PC** とペアで使用)。
 - g. サブスペース・グループに戻る **PT** または **PR** (第 1 の **PC** とペアで使用)。ディスパッチ可能単位はサブスペース・アクティブではなくなっているため、第 1 の **PC** がサブスペース内で発行されていた場合でも、制御は基本スペースに渡されます。
6. **BRANCH IN SUBSPACE GROUP** は、逐次化機能またはチェックポイント同期化機能は実行せずに、ホーム・スペース・モードにある場合を除き、事前取り出しされた命令のすべてのコピーを破棄させます。
7. **R2** フィールドがアクセス・レジスタ 0 を指しているときは、このアクセス・レジスタは、実際の内容に関係なく、**ALET 0** を含むものと見なされません。

COMPARE AND SWAP AND PURGE

CSP R_1, R_2 [RRE]



第 1 オペランドと第 2 オペランドが比較されます。両者が等しいときは、汎用レジスタ $R_1 + 1$ のビット位置 32-63 の内容が第 2 オペランドの位置に格納され、ページ操作が行われます。等しくない場合は、第 2 オペランドが第 1 オペランド位置にロードされます。比較の結果は条件コードで示されます。

第 1 オペランドは、汎用レジスタ R_1 のビット位置 32-63 の内容です。第 2 オペランドは記憶域内の 1 ワードです。第 2 オペランドの左端バイトの位置は、汎用レジスタ R_2 の内容によって指定されます。

ページ操作は、構成に含まれているすべての CPU 内の ART 索引バッファ (ALB) および変換索引バッファ (TLB) に適用されます。ページの対象として、ALB または TLB のいずれか一方、またはその両方を選択できます。選択したバッファ内のすべてのエントリが消去されます。

ページ操作は、汎用レジスタ R_2 のビット 62 および 63 に指定します。ビット 62 が 1 であれば、ALB からエントリが消去されます。ビット 63 が 1 であれば、TLB からエントリが消去されます。ビット 62 と 63 が両方とも 1 であれば、ALB および TLB からエントリが消去されます。ビット 62 と 63 が両方とも 0 のときは、エントリは消去されません。

汎用レジスタ R_2 の中のアドレスの取り扱いは、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスタ R_2 のビット位置 40-61 の内容の右側に 2 個の 0 を付加したものがアドレスを形成し、ビット位置 0-39 は無視されます。31 ビット・アドレッシング・モードでは、このレジスタのビット位置 33-61 の内容の右側に 2 個の 0 を付加したものがアドレスを形成し、ビット位置 0-32 の内容は無視されます。64 ビット・アドレッシング・モードでは、このレジスタのビット位置 0-61 の内容の右側に 2 個の 0 を付加したものが、アドレスを形成します。

10-18ページの図10-5 は、上記で挙げたレジスタの内容を示しています。

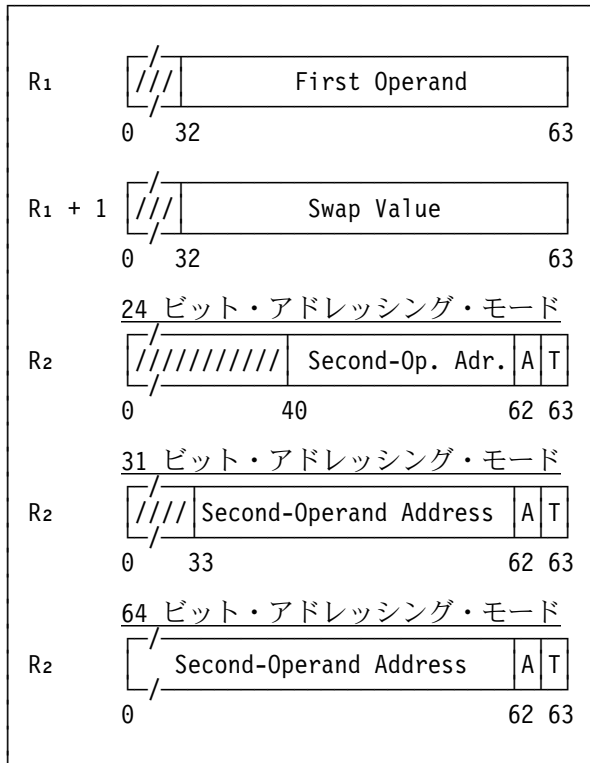


図 10-5. COMPARE AND SWAP AND PURGE で使用するレジスタの内容

比較の結果が等しいときは、汎用レジスタ $R_1 + 1$ のビット位置 32-63 の内容が第 2 オペランドの位置に格納されます。比較のための第 2 オペランドの取り出しと第 2 オペランド位置への格納は、他の CPU からは、ブロック単位のインターロック更新参照のように見えます。

比較の結果が等しくないときは、第 2 オペランドが第 1 オペランド位置にロードされ、汎用レジスタ R_1 のビット 0-31 は変更されず、第 2 オペランド位置は未変更のままになります。ただし、一部のモデルでは、第 2 オペランドが取り出された後、変更なしで再び第 2 オペランド位置に格納されることもあります。この更新は、他の CPU からはブロック単位のインターロック更新参照のように見えます。

オペランドが取り出される前、および操作が完了した後で、逐次化機能が実行されます。

比較の結果が等しい場合は、この CPU は、汎用レジスタ R_2 のビット 62 および 63 の指定に従って、ALB および TLB からエントリーを消去し、さらに、構成内のすべての CPU に対して、それぞれの ALB および TLB から同じエントリーを消去するよう通知します。ALB から消去されるエントリーは、すべての ALB ア

クセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、および権限テーブル・エントリーです。TLB から消去されるエントリーは、すべての領域テーブル/セグメント・テーブル結合エントリー、ページ・テーブル・エントリー、および実スペース・エントリーです。

CPU 上で COMPARE AND SWAP AND PURGE の実行が完了するのは、(1) その CPU の ALB および TLB から指定されたすべてのエントリーが消去され、そして、(2) 構成内の他のすべての CPU が、指定された ALB エントリーおよび TLB エントリーを使用して、すべての記憶域アクセス (変更ビットおよび参照ビットの更新を含む) を完了した時点です。

特別な条件

R_1 フィールドは、偶数番号のレジスタを指定していなければなりません。さもないと、指定例外が認識されます。

結果の条件コード:

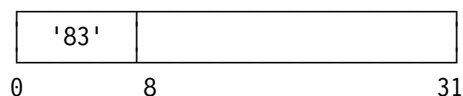
- 0 第 1 オペランドと第 2 オペランドが等しく、第 2 オペランドは汎用レジスタ $R_1 + 1$ のビット 32-63 で置き換えられました。
- 1 第 1 オペランドと第 2 オペランドが等しく、第 1 オペランドは第 2 オペランドで置き換えられました。
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (取り出しおよび格納: オペランド 2)
- 特権命令例外
- 指定例外

プログラミング上の注意: COMPARE AND SWAP AND PURGE は、ブロードキャスト形式の PURGE ALB 命令および PURGE TLB 命令と同じ働きをするので、SIGNAL PROCESSOR を使用をしなくて済みます。

DIAGNOSE



CPU は、組み込みの診断機能、またはモデルによって異なるその他の機能を実行します。診断機能の目的は、

装置が正しく機能しているかどうかを検査し、欠陥のあるコンポーネントを突き止めることにあります。モデルによって異なるその他の機能には、障害のあるバッファを使用不可にする機能、CPU、記憶機構、およびチャネル・パスを再構成する機能、および制御記憶機構を変更する機能などがあります。

ビット 8-31 は、SI フォーマットまたは RS フォーマットに見られる方法、またはその他の方法により、特定の診断機能を指定するために使用できます。この使用方法はモデルによって異なります。

この命令を実行すると、CPU の状態、レジスターまたは記憶位置の内容、そして入出力命令の進行に影響が現れることがあります。また、一部の診断機能では、テスト標識ライトが点灯することがあります。

結果の条件コード: このコードは予測不能です。

プログラム例外:

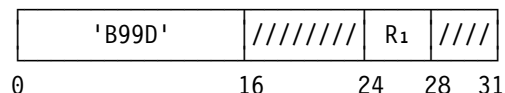
- 特権命令例外
- モデルに応じて、その他の例外が認識されることがあります。

プログラミング上の注意:

1. DIAGNOSE 命令は、問題プログラム状態プログラムや制御プログラムで使用するためのものではないので、この命令にはモニタリングはありません。
2. 他の命令では、プログラミング・エラーと装置エラーが区別されますが、DIAGNOSE はこの原則に従っていません。DIAGNOSE の使い方を誤ると、誤ったマシン・チェック標識が現れたり、実際に起きているマシンの誤動作が無視されたりすることがあります。また、この命令の結果、システムの動作の他の局面 (命令実行やチャネル・プログラムの動作など) が影響を受け、システムの動作が本書で述べる定義から逸脱したものになることもあります。DIAGNOSE を不適切に使用したために、システムが、パワーオン・リセットまたは初期マイクロプログラム・ローディング (IML) 機能を実行しなければならないような状態になることがあります。DIAGNOSE が果たす機能はモデル間で異なり、またモデルのバージョン間でも異なることがあるので、STORE CPU ID により格納されたモデル番号とバージョン・コードの両方を認識していない限り、プログラムで DIAGNOSE を発行しないでください。

EXTRACT AND SET EXTENDED AUTHORITY

ESEA R₁ [RRE]



制御レジスター 8 のビット 32-47 にある拡張許可指標 (EAX) が、第 1 オペランドのビット位置 32-47 に保管され、さらに、制御レジスター 8 の中の EAX が、第 1 オペランドのビット位置 48-63 の内容で置き換えられます。第 1 オペランドのビット 0-31 は無視され、変更されません。

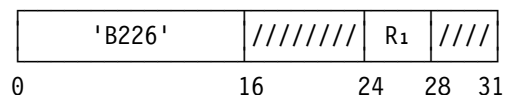
条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外

EXTRACT PRIMARY ASN

EPAR R₁ [RRE]



16 ビットの PASN (制御レジスター 4 のビット 48-63) が、汎用レジスター R₁ のビット位置 48-63 に入れます。この汎用レジスターのビット 32-47 は 0 に設定され、ビット 0-31 は変更されません。

特別な条件

この命令を実行するときは、DAT がオンになっていなければならないかもしれません。さもないと、特殊操作例外が認識されます。

問題プログラム状態では、抽出権限制御 (制御レジスター 0 のビット 36) が 1 でなければなりません。さもないと、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御ビットは検査されません。

10-20ページの図10-6 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

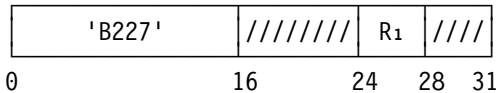
- 特権命令例外 (問題プログラム状態で抽出権限制御が 0)
- 特殊操作例外

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	DAT がオフであるために起きる特殊操作例外。
8.	問題プログラム状態で、抽出権限制御 (制御レジスター 0 のビット 36) が 0 であるために起きる特権命令例外。

図 10-6. 実行の優先順位: EXTRACT PRIMARY ASN

EXTRACT SECONDARY ASN

ESAR R1 [RRE]



16 ビットの SASN (制御レジスター 3 のビット 48-63) が、汎用レジスター R1 のビット位置 48-63 に入れます。この汎用レジスターのビット 32-47 は 0 に設定され、ビット 0-31 は変更されません。

特別な条件

この命令を実行するときは、DAT がオンになっていなければなりません。さもないと、特殊操作例外が認識されます。

問題プログラム状態では、抽出権限制御 (制御レジスター 0 のビット 36) が 1 でなければなりません。さもないと、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御ビットは検査されません。

図10-7 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外 (問題プログラム状態で抽出権限制御が 0)

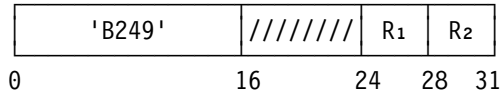
- 特殊操作例外

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	DAT がオフであるために起きる特殊操作例外。
8.	問題プログラム状態で、抽出権限制御 (制御レジスター 0 のビット 36) が 0 であるために起きる特権命令例外。

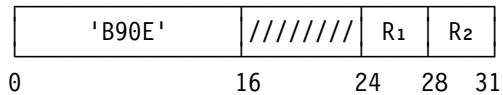
図 10-7. 実行の優先順位: EXTRACT SECONDARY ASN

EXTRACT STACKED REGISTERS

EREG R1,R2 [RRE]



EREGG R1,R2 [RRE]



リンケージ・スタック内の最後の状態エントリーに保管されている一組の汎用レジスターと一組のアクセス・レジスターの内容が、それぞれのレジスターに復元されます。これらのレジスターの組は、どちらも R1 で始まり R2 で終わるレジスターです。

EXTRACT STACKED REGISTERS (EREG) の場合は、汎用レジスターのビット位置 32-63 の内容が復元され、ビット位置 0-31 の内容は変更されません。

EXTRACT STACKED REGISTERS (EREGG) の場合は、汎用レジスターのビット位置 0-63 の内容が復元されます。どちらの場合も、アクセス・レジスターのビット位置 0-31 の内容が復元されます。

レジスター R1 から始まりレジスター R2 まで (このレジスターを含む) の汎用レジスターとアクセス・レジスターが、それぞれのレジスター番号の昇順に従ってロードされます。ただし、レジスター 0 はレジスター 15 の後に続くものと見なされます。各レジスターのビット位置には、状態エントリー内の対応する位置の内容がロードされます。状態エントリーには、その作成時に、レジスターの各ビット位置の内容が保管されています。状態エントリーの内容は変更されません。

最後の状態エントリーがどのようにして見つけ出されるかについては、5-75ページの『アンスタッキング・プロセス』の説明を参照してください。状態エントリーはリンケージ・スタック内に残され、制御レジスタ 15 中のリンケージ・スタック・エントリー・アドレスは変更されません。

リンケージ・スタックへの参照には、キー制御保護は適用されません。

特別な条件

CPU は、1 次スペース・モード、アクセス・レジスタ・モード、またはホーム・スペース・モードになっていなければなりません。さもないと、特殊操作例外が認識されます。

アンスタッキング・プロセスで、スタック空例外、スタック指定例外、またはスタック・タイプ例外が認識されることがあります。

何らかのアドレッシング例外が起きたときは、操作は中止されます。

図10-8 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (保護がある場合を除き、リンケージ・スタック・エントリーの取り出し)
- 特殊操作例外
- スタック空例外
- スタック指定例外
- スタック・タイプ例外

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	DAT がオフのため、または CPU が 2 次スペース・モードであるために起きる特殊操作例外。
8.	現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。
9.	現エントリーが状態エントリーまたはヘッダー・エントリーでないために起きるスタック・タイプ例外。 注: 例外 10-14 が起きることがあるのは、現エントリーがヘッダー・エントリーである場合のみです。
10.	ヘッダー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。
11.	ヘッダー・エントリー内の逆方向スタック・エントリー妥当性ビットが 0 であるために起きるスタック空例外。
12.	前のエントリーのエントリー記述子についてのアクセス例外 (取り出し)。前のエントリーとは、現 (ヘッダー) エントリー内の逆方向スタック・エントリー・アドレスが指しているエントリーです。
13.	前のエントリーがヘッダー・エントリーであるために起きるスタック指定例外。
14.	前のエントリーが状態エントリーでないために起きるスタック・タイプ例外。
15.	状態エントリーの選択された内容についてのアクセス例外 (取り出し)。

図 10-8. 実行の優先順位: EXTRACT STACKED REGISTERS

EXTRACT STACKED STATE

ESTA R₁,R₂ [RRE]

'B24A'				///////	R ₁	R ₂
0	16	24	28	31		

リンケージ・スタックに含まれている最後の状態エントリー内の 64 ビットまたは 128 ビットの状況情報が、R₁ フィールドに指定されている一対の汎用レジスタに入れられます。そして、その状態エントリーがブランチ状態エントリーかプログラム呼び出し状態エントリーを示す条件コードが設定されます。

R₁ フィールドは、汎用レジスタの奇偶ペアのうちの偶数番号のレジスタを指定します。

汎用レジスタ R₂ のビット 56-63 は符号なしの 2 進整数で、これは、以下に示すように、状態エントリーのどのバイト位置、またはどのバイト位置のどのビット位置から情報を抽出するかを選択するためのコードとして使用されます。

コード (汎用レジスタ R ₂ のビット 56-63)	状態エントリー内の選択されるバイト位置、またはバイトとビット位置
0	128-135
1	136-139、140.0、および 168-175.33-63 (説明を参照)
2	144-151
3	152-159
4	136-143 および 168-175

汎用レジスタ R₂ のビット位置 56-63 に入っているコードが 0、2、または 3 である場合は、8 バイトの状況情報のうちの左端 4 バイトが、汎用レジスタ R₁ のビット位置 32-63 に入れられ、右端 4 バイトは汎用レジスタ R₁ + 1 のビット位置 32-63 に入れられません。汎用レジスタ R₁ および R₁ + 1 のビット位置 0-31 は、変更されません。

汎用レジスタ R₂ のビット位置 56-63 にあるコードが 1 である場合は、状態エントリーのバイト 136-139 の内容 (状態エントリー内の PSW のビット 0-31) が、汎用レジスタ R₁ のビット位置 32-63 に入れられません。状態エントリーのバイト 140 のビット位置 0 の内容 (PSW のビット 32) が、汎用レジスタ R₁ + 1 のビット位置 32 に入れられます。そして、状態エントリ

一のバイト 168-175 のビット位置 33-63 の内容 (PSW のビット 97-127) が、汎用レジスタ R₁ + 1 のビット位置 33-63 に入れられます。ただし、汎用レジスタ R₁ のビット 44 (状態エントリー内の PSW のビット 12 に対応) は、ESA/390 モードを示す 1 に設定されます。また、状態エントリーのバイト 168-175 のビット 0-32 がすべて 0 ではない場合は、汎用レジスタ R₁ + 1 のビット 63 は 1 に設定されます。そうでない場合は、ビット 63 は、状態エントリーのバイト 168-175 のビット位置 63 からロードされた値のままになります。汎用レジスタ R₁ および R₁ + 1 のビット位置 0-31 は、変更されません。

汎用レジスタ R₂ のビット位置 56-63 にあるコードが 4 である場合は、状態エントリーのバイト 136-143 の内容 (状態エントリー内の PSW のビット 0-63) が、汎用レジスタ R₁ のビット位置 0-63 に入れられ、状態エントリーのバイト 168-175 の内容 (PSW のビット 64-127) が、汎用レジスタ R₁ + 1 のビット位置 0-63 に入れられます。

状態エントリーのバイト位置 128-175 のフォーマットは、以下のとおりです。

PKM	SASN	EAX	PASN
128	130	132	134 135

PSW Bits 0-63	
136	143

24 ビットまたは 31 ビット・モードで作成されるブランチ状態エントリーの場合

A		Bits 33-63 of Branch Address
144	148	151

64 ビット・モードで作成されるブランチ状態エントリーの場合

Bits 0-62 of Branch Address		1
144		151

結果のモードが 24 ビットまたは 31 ビットになるプログラム呼び出し状態エントリーの場合

Called-Space Id.	0	PC Number
144	148	151

結果のモードが 64 ビットになる
プログラム呼び出し状態エントリーの場合

Called-Space Id.	1	PC Number
144	148	151

Modifiable Area	
152	159

All Zeros	
160	167

PSW Bits 64-127	
168	175

状態エントリーの内容は変更されません。

最後の状態エントリーがどのようにして見つけ出されるかについては、5-75ページの『アンスタッキング・プロセス』の説明を参照してください。状態エントリーはリンケージ・スタック内に残され、制御レジスター 15 中のリンケージ・スタック・エントリー・アドレスは変更されません。

状態エントリーのエントリー記述子内のエントリー・タイプ・コードが 2 進数 0001100 (ブランチ状態エントリーを示す) であるときは、条件コードは 0 に設定されます。エントリー・タイプ・コードが 2 進数 0001101 (プログラム呼び出し状態エントリーを示す) であるときは、条件コードは 1 に設定されます。

リンケージ・スタックへの参照には、キー制御保護は適用されません。

汎用レジスター R₂ のビット 0-55 は無視されます。

特別な条件

R₁ が奇数である場合、または汎用レジスター R₂ のビット位置 56-63 のコードが 4 より大きい場合は、指定例外が認識されます。

CPU は、1 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードになっていなければなりません。さもないと、特殊操作例外が認識されます。

アンスタッキング・プロセスで、スタック空例外、スタック指定例外、またはスタック・タイプ例外が認識されることがあります。

何らかのアドレッシング例外が起きたときは、操作は抑止されます。

10-24ページの図10-9 に、この命令に関するプログラム例外の認識の優先順位を示します。

結果の条件コード:

- 0 ブランチ状態エントリー
- 1 プログラム呼び出し状態エントリー
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (保護がある場合を除き、リンケージ・スタック・エントリーの取り出し)
- 特殊操作例外
- 指定例外
- スタック空例外
- スタック指定例外
- スタック・タイプ例外

プログラミング上の注意: 汎用レジスター R₂ のビット位置 56-63 のコードが 1 であるときに生じる結果は、ESA/390 との互換性を維持するためのものです。(PSW のビット位置 0-31 のビット値のみが必要とされる場合もあります。)状態エントリーの中の PSW 内の命令アドレスが 31 ビット・アドレスより大きい場合は、汎用レジスター R₁ + 1 のビット 63 は 1 に設定されます。

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 ハーフワードについてのアクセス例外。
- 7.B DAT がオフのため、または CPU が 2 次スペース・モードであるために起きる特殊操作例外。
- 8.A R₁ が奇数であるか、または汎用レジスタ R₂ のビット 56-63 の値が 4 より大きいため起きる指定例外。
- 8.B.1 現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。
- 8.B.2 現エントリーが状態エントリーまたはヘッダー・エントリーでないために起きるスタック・タイプ例外。
注: 例外 8.B.3 - 8.B.7 が起きることがあるのは、現エントリーがヘッダー・エントリーである場合のみです。
- 8.B.3 ヘッダー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。
- 8.B.4 ヘッダー・エントリー内の逆方向スタック・エントリー妥当性ビットが 0 であるために起きるスタック空例外。
- 8.B.5 前のエントリーのエントリー記述子についてのアクセス例外 (取り出し)。前のエントリーとは、現 (ヘッダー) エントリー内の逆方向スタック・エントリー・アドレスが指しているエントリーです。
- 8.B.6 前のエントリーがヘッダー・エントリーであるために起きるスタック指定例外。
- 8.B.7 前のエントリーが状態エントリーでないために起きるスタック・タイプ例外。
- 8.B.8 状態エントリーの選択された内容についてのアクセス例外 (取り出し)。

図 10-9. 実行の優先順位: EXTRACT STACKED STATE

INSERT ADDRESS SPACE CONTROL

IAC R₁ [RRE]

0	'B224'	////////	R ₁	////	31
---	--------	----------	----------------	------	----

アドレス・スペース制御ビット (現 PSW のビット 16 および 17) が、汎用レジスタ R₁ のビット位置 54 および 55 に逆の順序で入れられます。つまり、ビット 16 はビット位置 55 に入れられ、ビット 17 はビット位置 54 に入れられます。このレジスタのビット 48-53 は 0 に設定され、ビット 0-47 および 56-63 は変更されません。アドレス・スペース制御ビットは、条件コードの設定にも使用されます。

特別な条件

この命令を実行するときは、DAT がオンになっていなければなりません。さもないと、特殊操作例外が認識されます。

問題プログラム状態では、抽出権限制御 (制御レジスタ 0 のビット 36) が 1 でなければなりません。さもないと、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御ビットは検査されません。

10-25ページの図10-10 に、この命令に関するプログラム例外の認識の優先順位を示します。

結果の条件コード:

- 0 PSW のビット 16 および 17 が 0 である (1 次スペース・モードを示す)。
- 1 PSW のビット 16 が 1 で、ビット 17 が 0 である (2 次スペース・モードであることを示す)。

- 2 PSW のビット 16 が 0 で、ビット 17 が 1 である (アクセス・レジスター・モードを示す)。
- 3 PSW のビット 16 および 17 が 1 である (ホーム・スペース・モードを示す)。

プログラム例外:

- 特権命令例外 (問題プログラム状態で抽出権限制御が 0)
- 特殊操作例外

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	DAT がオフであるために起きる特殊操作例外。
8.	問題プログラム状態で、抽出権限制御 (制御レジスター 0 のビット 36) が 0 であるために起きる特権命令例外。

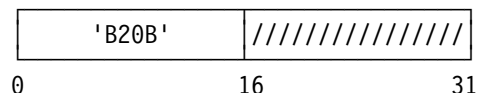
図 10-10. 実行の優先順位: INSERT ADDRESS SPACE CONTROL

プログラミング上の注意:

1. 汎用レジスター R₁ のビット 48-53 は、将来の機能拡張に備えて予約されています。プログラムでは、これらのビットが 0 に設定されていることを前提にはいけません。
2. INSERT ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL は、汎用レジスターの第 7 バイトに作用するように定義されており、したがって、アドレス・スペース制御ビットを、INSERT PSW KEY により汎用レジスター 2 の第 8 バイトに入れられた PSW キーと同じ汎用レジスターに保管することができます。

INSERT PSW KEY

IPK [S]



4 ビットの PSW キー (現 PSW のビット 8-11) が、汎用レジスター 2 のビット位置 56-59 に挿入され、そのレジスターのビット 60-63 は 0 に設定されます。汎用レジスター 2 のビット 0-55 は変更されません。

特別な条件

問題プログラム状態では、抽出権限制御 (制御レジスター 0 のビット 36) が 1 でなければなりません。さもないと、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御ビットは検査されません。

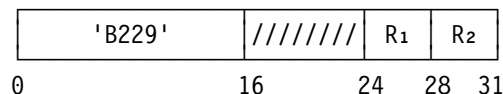
条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外 (問題プログラム状態で抽出権限制御が 0)

INSERT STORAGE KEY EXTENDED

ISKE R₁,R₂ [RRE]



汎用レジスター R₂ の内容が示すアドレスにあるブロックの記憶キーが、汎用レジスター R₁ に挿入されます。

24 ビット・アドレッシング・モードでは、汎用レジスター R₂ のビット 40-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-39 および 52-63 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスター R₂ のビット 33-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスター R₂ のビット 0-51 が、実記憶域内の 4K バイト・ブロックを指定し、ビット 52-63 は無視されます。

記憶域ブロックを指定するアドレスは、実アドレスなので、動的アドレス変換の対象にはなりません。また、記憶キーに対する参照には記憶保護例外は適用されません。

7 ビットの記憶キーが、汎用レジスター R₁ のビット位置 56-62 に挿入され、ビット 63 は 0 に設定されます。レジスターのビット位置 0-55 は変更されません。

条件コード: コードは変更されません。

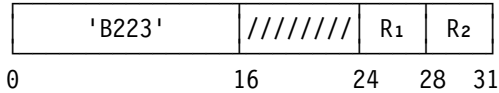
プログラム例外:

- アドレッシング例外 (汎用レジスター R₂ に指定されているアドレス)

- 特権命令例外

INSERT VIRTUAL STORAGE KEY

IVSK R₁,R₂ [RRE]



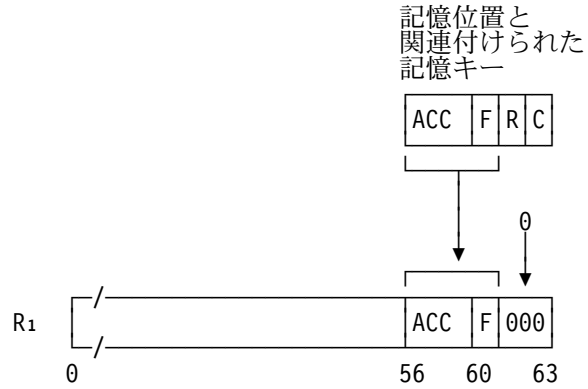
汎用レジスタ R₂ 中の仮想アドレスが示す位置の記憶キーが、汎用レジスタ R₁ に挿入されます。

汎用レジスタ R₂ の選択されたビットが、仮想アドレスとして使用されます。24 ビット・アドレッシング・モードでは、このアドレスは汎用レジスタのビット 40-63 と置き換わり、0-39 は無視されます。31 ビット・アドレッシング・モードでは、アドレスはビット 33-63 に示され、ビット 0-32 は無視されます。64 ビット・アドレッシング・モードでは、アドレスはレジスタのビット 0-63 に示されます。

このアドレスは仮想アドレスであり、アドレス・スペース制御ビット (現 PSW のビット 16 および 17) により制御されます。このアドレスは、1 次スペース・モードでは 1 次仮想アドレスとして、2 次スペース・モードでは 2 次仮想アドレスとして、アクセス・レジスタ・モードでは AR 指定仮想アドレスとして、そしてホーム・スペース・モードではホーム仮想アドレスとして扱われます。また、記憶キーに対する参照には記憶保護例外は適用されません。

記憶キーのビット 0-4 (アクセス制御ビットと取り出し保護ビット) が、汎用レジスタ R₁ のビット位置 56-60 に入れられ、ビット 61-63 は 0 に設定されます。レジスタのビット位置 0-55 は変更されません。記憶キーの中の変更ビットおよび参照ビットは検査されません。変更ビットは、この操作の影響を受けません。参照ビットは、モデルに応じて、この操作の結果 1 に設定される場合とされない場合があります。

次の図は、上記で述べた記憶キーおよびレジスタ位置を示しています。



特別な条件

この命令を実行するときは、DAT がオンになっていなければなりません。さもないと、特殊操作例外が認識されます。

問題プログラム状態では、抽出権限制御 (制御レジスタ 0 のビット 36) が 1 でなければなりません。さもないと、特権命令例外が認識されます。監視プログラム状態では、抽出権限制御ビットは検査されません。

図10-11 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (記憶保護例外の場合を除き、汎用レジスタ R₂ に指定されているアドレス)
- 特権命令例外 (問題プログラム状態で抽出権限制御が 0)
- 特殊操作例外

- | | |
|-------|--|
| 1.-6. | 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7.A | 命令の第 2 ハーフワードについてのアクセス例外。 |
| 7.B | DAT がオフであるために起きる特殊操作例外。 |
| 8. | 抽出権限制御 (制御レジスタ 0 のビット 36) が 0 であるために起きる特権命令例外。 |
| 9. | 汎用レジスタ R ₂ に指定されているアドレスについてのアクセス例外 (記憶保護例外は除く)。 |

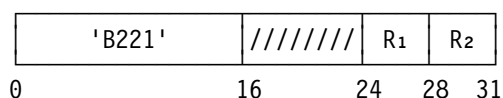
図 10-11. 実行の優先順位: INSERT VIRTUAL STORAGE KEY

プログラミング上の注意:

1. 1つの4Kバイト・ブロックの中のすべてのバイトは、同じページおよび同じ記憶キーに関連付けられるので、汎用レジスタ R₂ のビット 52-63 は事実上無視されます。
2. アクセス・レジスタ・モードでは、アクセス・レジスタ 0 は、アクセス・レジスタ 0 の内容に関係なく、1 次アドレス・スペースを示します。

INVALIDATE PAGE TABLE ENTRY

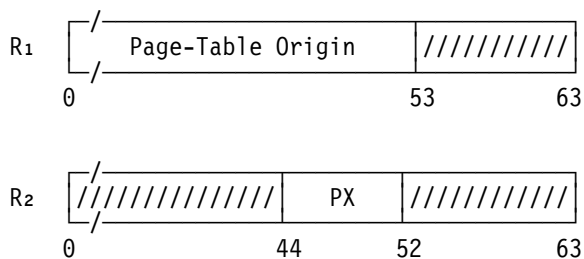
IPTE R₁,R₂ [RRE]



指定されたページ・テーブル・エントリが無効化され、構成内のすべての CPU の変換索引バッファ (TLB) から、そのエントリに関連したエントリが消去されます。

汎用レジスタ R₁ の内容はセグメント・テーブル・エントリのフォーマットをとり、その中のページ・テーブル起点のみが使用されます。汎用レジスタ R₂ は仮想アドレスのフォーマットをとり、その中のページ指標のみが使用されます。ページ・テーブル起点またはページ指標以外のフィールド内容は無視されます。

上記で述べた汎用レジスタの内容は、以下のとおりです。



ページ・テーブル起点とページ指標は、ページ・テーブル検索のための動的アドレス変換の規則に従って、1つのページ・テーブル・エントリを指定します。ページ・テーブル起点は64ビット・アドレスとして扱われ、現PSWのビット31および32に指定されている現アドレッシング・モードに関係なく、64ビット・アドレス演算の規則に従って加算が行われます。ページ指

標およびページ・テーブル起点の加算の結果として、ビット位置0からの繰り上がりが生じることはありません。この2つのコンポーネントから形成されるアドレスは、実アドレスまたは絶対アドレスです。このページ・テーブル・エントリのページ無効ビットは、1に設定されます。このプロシージャでは、このページ・テーブル・エントリについて、該当ページが使用可能かどうか、またはフォーマット・エラーがないかどうかは検査されません。また、このエントリに含まれているページ・フレーム実アドレスについて、アドレッシング例外があるかどうかも検査されません。

他のCPUからは、ページ・テーブル・エントリの全体が同時に記憶域から取り出されているように見えます。次に、ページ無効ビットを含むバイトが格納されます。ページ・テーブル・エントリへの取り出しアクセスは、キー制御保護の対象となり、格納アクセスは、キー制御保護と低アドレス保護の対象となります。

この操作の開始前と完了後に、逐次化機能が実行されます。他のすべての逐次化操作の場合と同様に、この逐次化もこのCPUのみに適用され、他のCPUは必ずしも逐次化されません。

ページ無効ビットが1に設定されると、このCPUは、選択されたエントリを自身のTLBから消去し、構成内のすべてのCPUに、それぞれのTLBから同じ選択されたエントリを消去するよう通知します。各TLBから、少なくとも、下記のすべてを使用して形成されているエントリが消去されます。

- 第1オペランドが指定しているページ・テーブル起点
- 第2オペランドが指定しているページ指標
- 指定されたページ・テーブル・エントリに含まれているページ・フレーム実アドレス

CPU上でINVALIDATE PAGE TABLE ENTRYの実行が完了するのは、(1) 指定されたパラメータに対応するすべてのエントリがこのCPUのTLBから消去され、そして、(2) 構成内の他のすべてCPUが、指定されたパラメータに対応するTLBエントリを使用して、すべての記憶域アクセス(変更ビットと参照ビットの更新を含む)を完了した時点です。

特別な条件

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

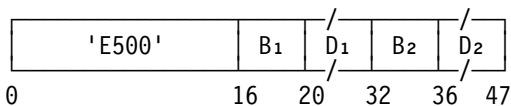
- アドレッシング例外 (ページ・テーブル・エントリー)
- 特権命令例外
- 記憶保護例外 (取り出しおよび格納、ページ・テーブル・エントリー、キー制御保護、および低アドレス保護)

プログラミング上の注意:

1. エントリーの選択的除去の方法は、モデルによって異なり、一般に、必要最小数より多くのエントリーが除去されます。一部のモデルでは、該当のページ・テーブル・エントリーから得られたページ・フレーム実アドレスを含むすべてのエントリーが、記憶域から除去されます。他のモデルでは、指定されたページ指標を含むすべてのエントリーが削除されることもあり、またインプリメンテーションによっては、正確に必要な最小数のみのエントリーが除去されることもあります。したがって、プログラムがすべてのモデルで稼働できるようにするためには、特定のモデルで選択度の低い除去を行うことにより得られる効果を利用することは避けるようにしてください。
2. TLB エントリーの除去では、ページ・テーブル・エントリー内のページ・フレーム実アドレスが利用されることがあります。ページ・テーブル・エントリーが、連結された状態にあるときに、現在の値とは異なるページ・フレーム実アドレスを含んでいたとすれば、前の値を含むエントリーのコピーが TLB 内に残されていることがあります。
3. 他の CPU またはチャネル・プログラムが主記憶域内の共用記憶位置を更新している可能性があるときは、INVALIDATE PAGE TABLE ENTRY を使用してその共用記憶位置を更新するのは、安全とは言えない場合があります。

LOAD ADDRESS SPACE PARAMETERS

LASP D₁(B₁), D₂(B₂) [SSE]



第 1 オペランド位置にあるダブルワードの内容には、制御レジスター 3 および 4 にロードする値が含まれています。これには、2 次 ASN (SASN) および 1 次 ASN (PASN) も含まれます。この命令の実行は、PASN 変換、SASN 変換、SASN 許可、および制御レジスター・ロードという 4 つの主要ステップに分けられます。どのステップも、所定のテストの結果と第 2 オペランド・アドレスのビット 61-63 の設定に応じて、実行され場合とされない場合があります。これらのステップが成功すると、新たな値が導き出されて、制御レジスター 1、5、および 7 にロードされます。これらのステップが成功しなかったときは、制御レジスターは変更されず、条件コードにその理由が示されます。

ダブルワードの第 1 オペランドには、PSW キー・マスク (PKM)、2 次 ASN (SASN)、許可指標 (AX)、および 1 次 ASN (PASN) が含まれています。1 次 ASN は ASN 変換テーブルを使用して変換されて、1 次 ASTE (PASTE) 起点 (PASTE0) が求められ、PASTE からは 1 次 ASCE (PASCE) が導き出されます。また、必要に応じて、PASTE から AX が導き出されます。2 次 ASN は ASN 変換テーブルを使用して変換されて、2 次 ASCE (SASCE) が求められ、必要があれば、新しい AX が新しい SASN を設定することを許可されているかどうかを確認するための権限検査が行われます。

第 1 オペランド位置にあるダブルワードのフォーマットは、以下のとおりです。

PKM-d	SASN-d	AX-d	PASN-d	
0	16	32	48	63

「d」は指定されたダブルワードを表しており、定義の中で参照している類似の名前の他のフィールドから、これらのフィールドを区別するために使用されています。制御レジスター内の対応するフィールドの現在の内容は、PKM-old、SASN-old のように表記します。制御レジスターの更新後の内容は、PKM-new、SASN-new のように表記します。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。代わりに、右端の 3 ビットが、操作の各部分を制御するために使用されます。第 2 オペランド・アドレスのその他の部分は無視されます。第 2 オペランド・アドレスのビット 61-63 は、以下のように使用されます。

ビット	第 2 オペランド・アドレス で指定された機能	
	ビットが 0 のとき	ビットが 1 のとき
61	新 ASN と旧 ASN が異なる場合のみ ASN 変更が行われる。	ASN 変換が行われる。*
62	PASN に関連した AX が使用される。	第 1 オペランドからの AX が使用される。
63	SASN 許可が行われる。*	SASN 許可は行われない。
* SASN 変換および SASN 許可は、SASN-d が PASN-d と等しくない場合のみ行われます。SASN-d が PASN-d と等しいときは、SASCE は PASCE からロードされ、許可は行われません。		

LOAD ADDRESS SPACE PARAMETERS の操作を、10-36ページの図10-15 に示してあります。

PASN 変換

PASN 変換プロセスでは、ASN 第 1 テーブルおよび ASN 第 2 テーブルを使用して、PASN-d が変換されます。ASTE から ASCE (および必要があれば AX フィールド) が取得されて、後で対応する制御レジスターを更新するために使用されます。PASN 変換の結果の PASTE0 は、制御レジスター 5 を更新するために使用されます。

第 2 オペランド・アドレスのビット 61 が 1 のときは、常に PASN 変換が行われます。ビット 61 が 0 のときは、PASN-d が PASN-old に等しくない場合のみ PASN 変換が行われます。ビット 61 が 0 で、PASN-d が PASN-old に等しいときは、制御レジスター内の PASCE-old および PASTE0-old は変更されずに残され、それぞれ PASCE-new および PASTE0-new となります。この場合に、ビット 62 が 0 であれば、制御レジスター内の AX-old は変更されず、AX-new となります。

PASN 変換は、通常の ASN 変換の規則に従って行われます。ただし、この場合は、無効ビット (ASN 第 1 テーブル・エントリーのビット 0 および ASTE のビット 0) が 1 であっても、ASN 変換例外にはならず、ASTE のスペース切り替えイベント制御ビットが 1 であっても、スペース切り替えイベントは生じません。無効ビットのいずれかが 1 のときは、条件コード 1 が設定されます。ASTE が有効であり、かつ、制御レジスター 1 の中の現 1 次スペース切り替えイベント制御ビットが 1 であるか、ASTE 内のスペース切り替えイベント制御ビ

ットが 1 である場合は、条件コード 3 が設定されます。条件コード 1 または 3 が設定された場合は制御レジスターは変更されません。

PASN 変換の結果としてアクセスされる ASTE の中の AX フィールドと ASCE フィールドの内容を、それぞれ AX-p および ASCE-p と呼びます。また、ASTE の起点を PASTE0-p と呼びます。

このパラグラフで述べる内容は、サブスペース・グループ機能の使用に適用されます。ASCE-p が取得された後で、(1) サブスペース・グループ制御ビット (ASCE-p のビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) PASTE0-p がディスパッチ可能単位の基本スペースの ASTE を示している場合は、ASCE-p のコピー (ASCE-rp と呼ばれる) が作成されます。そのとき、ASCE-rp のビット 0-55 および 58-63 は、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。サブスペース ASTE 内のビット 0 が 1 である場合、またはサブスペース ASTE 内の ASTE シーケンス番号 (ASTESN) がディスパッチ可能単位制御テーブル内のサブスペース ASTESN に等しくない場合でも、例外は認識されません。代わりに、条件コード 1 が設定され、制御レジスターは変更されません。

SASN 変換

SASN 変換プロセスでは、ASN 第 1 テーブルおよび ASN 第 2 テーブルを使用して、SASN-d が変換されます。ASTE から取得された ASCE フィールドは、後で制御レジスター 7 の中の 2 次 ASCE (SASCE) を更新するために使用されます。SASN 許可が行われる場合は、ここで取得された ATO フィールドと ATL フィールドが使用されます。

SASN 変換が行われるのは、SASN-d が PASN-d に等しくない場合のみです。SASN-d が PASN-d に等しいときは、SASCE-new は PASCE-new と同じ値に設定されます。SASN-d が SASN-old に等しく、ビット 61 (ASN 変換強制) が 0 で、かつビット 63 (SASN 許可スキップ) が 1 である場合は、SASN 変換は行われず、SASCE-old は SASCE-new になります。

SASN 変換は、通常の ASN 変換の規則に従って行われます。ただし、無効ビット (ASN 第 1 テーブル・エントリーのビット 0 および ASTE のビット 0) が 1 であ

っても、ASN 変換例外にはなりません。無効ビットのいずれかが 1 のときは、条件コード 2 が設定され、制御レジスターは変更されません。

SASN 変換の結果としてアクセスされる ASTE の中の ASCE、ATO、および ATL フィールドの内容を、それぞれ、ASCE-s、ATO-s、および ATL-s と呼びます。また、ASTE の起点を SASTEO-s と呼びます。

このパラグラフで述べる内容は、サブスペース・グループ機能の使用に適用されます。ASCE-s が取得された後で、(1) サブスペース・グループ制御ビット (ASCE-s のビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) SASTEO-s がディスパッチ可能単位の基本スペースの ASTE を示している場合は、ASCE-s のコピー (ASCE-rs と呼ばれる) が作成されます。そのとき、ASCE-rs のビット 0-55 および 58-63 は、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。サブスペース ASTE 内のビット 0 が 1 である場合、またはサブスペース ASTE 内の ASTE シーケンス番号 (ASTESN) がディスパッチ可能単位制御テーブル内のサブスペース ASTESN に等しくない場合でも、例外は認識されません。代わりに、条件コード 2 が設定され、制御レジスターは変更されません。

SASN 許可

SASN 許可が行われるのは、第 2 オペランド・アドレスのビット 63 が 0 で、SASN-d が PASN-d に等しくない場合です。SASN-d が PASN-d に等しいか、または第 2 オペランド・アドレスのビット 63 が 1 のときは、SASN 許可は行われません。

SASN 許可は、ATO-s、ATL-s、および意図されている AX-new の値を使用して行われます。第 2 オペランド・アドレスのビット 62 が 0 で、PASN 変換が行われた場合は、意図されている AX-new の値は AX-p です。そのアドレスのビット 62 が 0 で、PASN 変換が行われていない場合は、AX は変更されず、AX-new は AX-old と同じになります。そのアドレスのビット 62 が 1 のときは、意図されている AX-new の値は AX-d です。SASN 許可は、3-23ページの『ASN 許可プロセス』で説明されている 2 次許可に関する規則に従って行われます。SASN が許可されない場合 (つまり、権限テーブル長を超過しているか、選択されたビットが 0 の場

合) は、条件コード 2 が設定され、制御レジスターはどれも更新されません。

制御レジスター・ロード

PASN 変換、SASN 変換、および SASN 許可機能およびサブスペース置換操作が、命令の実行中に呼び出されて実行されたときに、例外または例外状況が発生しなかった場合は、実行は完了し、制御レジスター 1、3、4、5、および 7 が新しい値で置き換えられ、条件コード 0 が設定されます。制御レジスターは以下のようにロードされます。

制御レジスター 3 の PSW キー・マスク・フィールドおよび SASN フィールドは、第 1 オペランド位置にある PKM-d フィールドおよび SASN-d フィールドで置き換えられます。

PASN (制御レジスター 4 のビット 48-63) は、第 1 オペランド位置の PASN-d で置き換えられます。

許可指標 (制御レジスター 4 のビット 32-47) は、以下のように置き換えられます。

- 第 2 オペランド・アドレスのビット 62 が 1 のときは、AX-d で置き換えられます。
- 第 2 オペランド・アドレスのビット 62 が 0 で、PASN 変換が行われた場合は、AX-p で置き換えられます。
- 第 2 オペランド・アドレスのビット 62 が 0 で、PASN 変換が行われていない場合は、許可指標は変更されません。

制御レジスター 1 の中の 1 次アドレス・スペース制御エレメント (PASCE) および制御レジスター 5 の中の 1 次 ASN 第 2 テーブル・エントリー起点 (PASTEO) は、以下のように置き換えられます。

- PASN 変換が行われた場合は、制御レジスター 1 の中の PASCE は、PASN 変換で得られた ASCE-p フィールドで置き換えられます。ただし、ASCE-p に対してサブスペース変換操作が行われた場合は、PASCE が ASCE-rp で置き換えられます。さらに、制御レジスター 5 の中の PASTEO は PASTEO-p で置き換えられます。

PASTEO-p は、制御レジスター 5 のビット位置 33-57 に入れられ、ビット位置 32 および 58-63 には 0 が入れられます。このレジスターのビット 0-31 は変更されません。

- PASN 変換が行われていない場合は、制御レジスター 1 および 5 は変更されません。

制御レジスター 7 の中の 2 次アドレス・スペース制御エレメント (SASCE) は、以下のように置き換えられます。

- SASN-d が PASN-d に等しいときは、制御レジスター 1 の新しい内容、つまり PASCE で置き換えられます。新しい内容は、PASCE-old、ASCE-p、または ASCE-rp のいずれかです。
- SASN が行われた場合は、ASCE-s で置き換えられます。ただし、ASCE-s に対してサブスペース置換操作が行われた場合は、ASCE-rs で置き換えられます。

SASN-d が PASN-d に等しくなく、かつ SASN 変換が行われていない場合は、SASCE は変更されません。

その他の条件コードの設定

PASN 変換が要求されたときに、ASN 第 1 テーブル・エントリーまたは ASTE のビット 0 が 1 であるために変換を完了できない場合、または、完了はできるが、ASCE-p に対するサブスペース置換操作中に、サブスペース ASTE 内のビット 0 または ASTE シーケンス番号が原因でサブスペース置換例外条件が存在する場合は、条件コード 1 が設定され、制御レジスターは変更されません。

要求された PASN 変換が完了し、ASCE-p に対して必要なサブスペース置換操作も完了したときに、(1) 1 次スペース切り替えイベント制御ビット (制御レジスター 1 のビット 57) が 1 であるか、または、(2) PASTEOP に指定されている ASTE のスペース切り替えイベント制御ビットが 1 である場合は、条件コード 3 が設定され、制御レジスターは変更されません。

SASN 変換が要求されているときに、(1) ASN 第 1 テーブル・エントリーまたは ASTE のビット 0 が 1 であるか、(2) SASN 許可が要求されたが SASN が許可されないか、または、(3) ASCE-s に対するサブスペース置換操作中に、サブスペース ASTE のビット 0 または ASTE シーケンス番号が原因でサブスペース置換例外条件が存在するために、変換を完了できない場合は、条件コード 2 が設定されます。制御レジスターは変更されません。

特別な条件

この命令を実行できるのは、ASN 変換制御 (制御レジスター 14 のビット 44) が 1 のときだけです。ASN 変換制御ビットが 0 のときは、特殊操作例外が認識されません。

第 1 オペランドは、ダブルワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-33ページの図10-13 および 10-32ページの図10-12 は、この命令の機能と、例外と条件コードの認識の優先順位を要約して示しています。

結果の条件コード:

- 0 変換と許可は完了し、パラメーターがロードされました。
- 1 1 次 ASN またはサブスペースが使用できません。パラメーターはロードされていません。
- 2 2 次 ASN が使用可能でないか許可されていないか、または 2 次サブスペースが使用できません。パラメーターはロードされていません。
- 3 スペース切り替えイベントが指定されました。パラメーターはロードされていません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 1)
- アドレッシング例外 (ASN 第 1 テーブル・エントリー、ASN 第 2 テーブル・エントリー、権限テーブル・エントリー、ディスパッチ可能単位制御テーブル)
- 特権命令例外
- 特殊操作例外
- 指定例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 および第 3 ハーフワードについてのアクセス例外。
- 7.B.1 特権命令例外。
- 7.B.2 ASN 変換制御 (制御レジスター 14 のビット 34) が 0 であるために起きる特殊操作例外。
- 8. 指定例外。
- 9. 第 1 オペランドについてのアクセス例外。
- 10. PASN 変換の実行 (実行される場合)。
- 10.1 ASN 第 1 テーブル・エンタリーへのアクセスについてのアドレッシング例外。
- 10.2 ASN 第 1 テーブル・エンタリーの I ビット (ビット 0) が 1 であるために設定される条件コード 1。
- 10.3 ASN 第 2 テーブル・エンタリーへのアクセスについてのアドレッシング例外。
- 10.4 ASN 第 2 テーブル・エンタリーの I ビット (ビット 0) が 1 であるために設定される条件コード 1。
- 10.5 ディスパッチ可能単位制御テーブルへのアクセスについてのアドレッシング例外。
- 10.6 サブスペース ASN 第 2 テーブル・エンタリーへのアクセスについてのアドレッシング例外。
- 10.7 サブスペース ASN 第 2 テーブル・エンタリーの I ビット (ビット 0) が 1 であるために設定される条件コード 1。
- 10.8 ディスパッチ可能単位制御テーブル内の ASN 第 2 テーブル・エンタリー・シーケンス番号 (SSASTESN) が、サブスペース ASN 第 2 テーブル・エンタリー内の ASTESN に等しくないために設定される条件コード 1。
- 10.9 新旧いずれかのスペース切り替えイベント制御ビットが 1 であるために設定される条件コード 3。
- 11. SASN 変換の実行 (実行される場合)。
- 11.1 ASN 第 1 テーブル・エンタリーへのアクセスに関するアドレッシング例外。
- 11.2 ASN 第 1 テーブル・エンタリー内の I ビット (ビット 0) が 1 であるために設定される条件コード 2。
- 11.3 ASN 第 2 テーブル・エンタリーへのアクセスに関するアドレッシング例外。
- 11.4 ASN 第 2 テーブル・エンタリー内の I ビット (ビット 0) が 1 であるために設定される条件コード 2。

図 10-12 (1/2). 実行の優先順位: LOAD ADDRESS SPACE PARAMETERS

12.A	2 次許可の実行 (実行される場合)。
12.A.1	権限テーブル・エントリーがテーブルの外部にあるために設定される条件コード 2。
12.A.2	権限テーブル・エントリーへのアクセスについてのアドレッシング例外。
12.A.3	権限テーブル・エントリーの S ビットが 0 であるために設定される条件コード 2。
12.B.1	ディスパッチ可能単位制御テーブルへのアクセスについてのアドレッシング例外。
12.B.2	サブスペース ASN 第 2 テーブル・エントリーへのアクセスについてのアドレッシング例外。
12.B.3	サブスペース ASN 第 2 テーブル・エントリー内の I ビット (ビット 0) が 1 であるために設定される条件コード 2。
12.B.4	ディスパッチ可能単位制御テーブル内のサブスペース ASN 第 2 テーブル・エントリー・シーケンス番号 (SSASTESN) が、サブスペース ASN 第 2 テーブル・エントリー内の ASTESN に等しくないために設定される条件コード 1。

図 10-12 (2/2). 実行の優先順位: LOAD ADDRESS SPACE PARAMETERS

PASN-d は PASN-old に等しい か	第 2 オペ ランド・ア ドレスの ビット ¹		PASN 変換の実行 の有無	結果フィールド					
	61	62		PASCE-new	AX-new	PASTE0-new	PKM-new	SASN-new	PASN-new
等	0	0	なし	PASCE-old	AX-old	PASTE0-old	PKM-d	SASN-d	PASN-d
等	0	1	なし	PASCE-old	AX-d	PASTE0-old	PKM-d	SASN-d	PASN-d
等	1	0	あり	ASCE-p ²	AX-p	PASTE0-p	PKM-d	SASN-d	PASN-d
等	1	1	あり	ASCE-p ²	AX-d	PASTE0-p	PKM-d	SASN-d	PASN-d
不等	-	0	あり	ASCE-p ²	AX-p	PASTE0-p	PKM-d	SASN-d	PASN-d
不等	-	1	あり	ASCE-p ²	AX-d	PASTE0-p	PKM-d	SASN-d	PASN-d

図 10-13 (1/2). アクションの要約: LOAD ADDRESS SPACE PARAMETERS

SASN-d は PASN-d に等し いか	SASN-d は SASN-old に等し いか	第 2 オペラ ンド・アドレ スの ビット ¹		SASN 変換の実行 の有無	SASN 許可の実行 の有無 ³	結果 フィールド SASCE-new
		61	63			
等 不等 不等 不等 不等 不等	- 等 等 等 不等 不等	- 0 1 - - -	- 1 1 0 1 0	なし なし あり あり あり あり	なし なし なし なし なし あり	PASCE-new SASCE-old ASCE-s ⁴ ASCE-s ⁴ ASCE-s ⁴ ASCE-s ⁴

説明:

- この場合のアクションは、この比較の結果またはこのビットの設定に関係なく同じです。

¹ 第 2 オペランド・アドレスのビット:

- 61 ASN 変換の強制。
- 62 第 1 オペランドからの AX の使用。
- 63 2 次権限テストをスキップ。

² サブスペース置換が行われた場合は、PASCE-new は ASCE-rp です (ビット 0-55 および 58-63 をサブスペース ASCE 内の ASCE で置き換えた、ASCE-p のコピー)。

³ ATO-s、ATL-s、および AX-new を使用して SASN 許可が行われています。

⁴ サブスペース置換が行われた場合は、SASCE-new は ASCE-rs です (ビット 0-55 および 58-63 をサブスペース ASCE の ASCE で置き換えた、ASCE-s のコピー)。

図 10-13 (2/2). アクションの要約: LOAD ADDRESS SPACE PARAMETERS

プログラミング上の注意:

1. 第 2 オペランド・アドレスのビット 61 および 63 は、主として、関連のアクションが不要な場合のパフォーマンスを高めることを目的としています。

ビット 61 が 0 に設定されているときは、PASCE-old、PASTE0-old、および AX-old の現在の値と PASN-old の間に整合性があり、かつ SASCE-old と SASN-old の間に整合性があるという前提に基づいて、命令のアクションが実行されます。これに該当しない場合は、ビット 61 は 1 に設定する必要があります。

ビット 63 が 1 のときは、SASN 許可テストは行われません。場合によっては、前回の SASN の使用状況に基づき、またはロードされている AX が

すべてのアドレス・スペースへのアクセスを許可されているという理由で、SASN が許可されていることをプログラムが判別できることがあります。

2. SASN-d が PASN-d に等しいときは、SASN 変換および SASN 許可のステップは実行されません。これは、現 1 次への SET SECONDARY ASN (SSAR-cp) のアクションで、変換または ASN 許可が行われない点と一致しています。

3. LOAD ADDRESS SPACE PARAMETERS の記憶域オペランド参照は、複数アクセス参照の場合があります。(5-86ページの『記憶域オペランドの一貫性』を参照。)

4. この命令の説明の中で使用している省略形のリストについては、10-35ページの図10-14を参照してください。

制御レジスタ番号とビット位置	省略形	
	前の内容	後の内容
1.0-63	PASCE-old	PASCE-new
3.32-47	PKM-old	PKM-new
3.48-63	SASN-old	SASN-new
4.32-47	AX-old	AX-new
4.48-63	PASN-old	PASN-new
5.33-57	PASTE0-old	PASTE0-new
7.0-63	SASCE-old	SASCE-new

第 1 オペランドのビット位置	省略形
0-15	PKM-d
16-31	SASN-d
32-47	AX-d
48-63	PASN-d

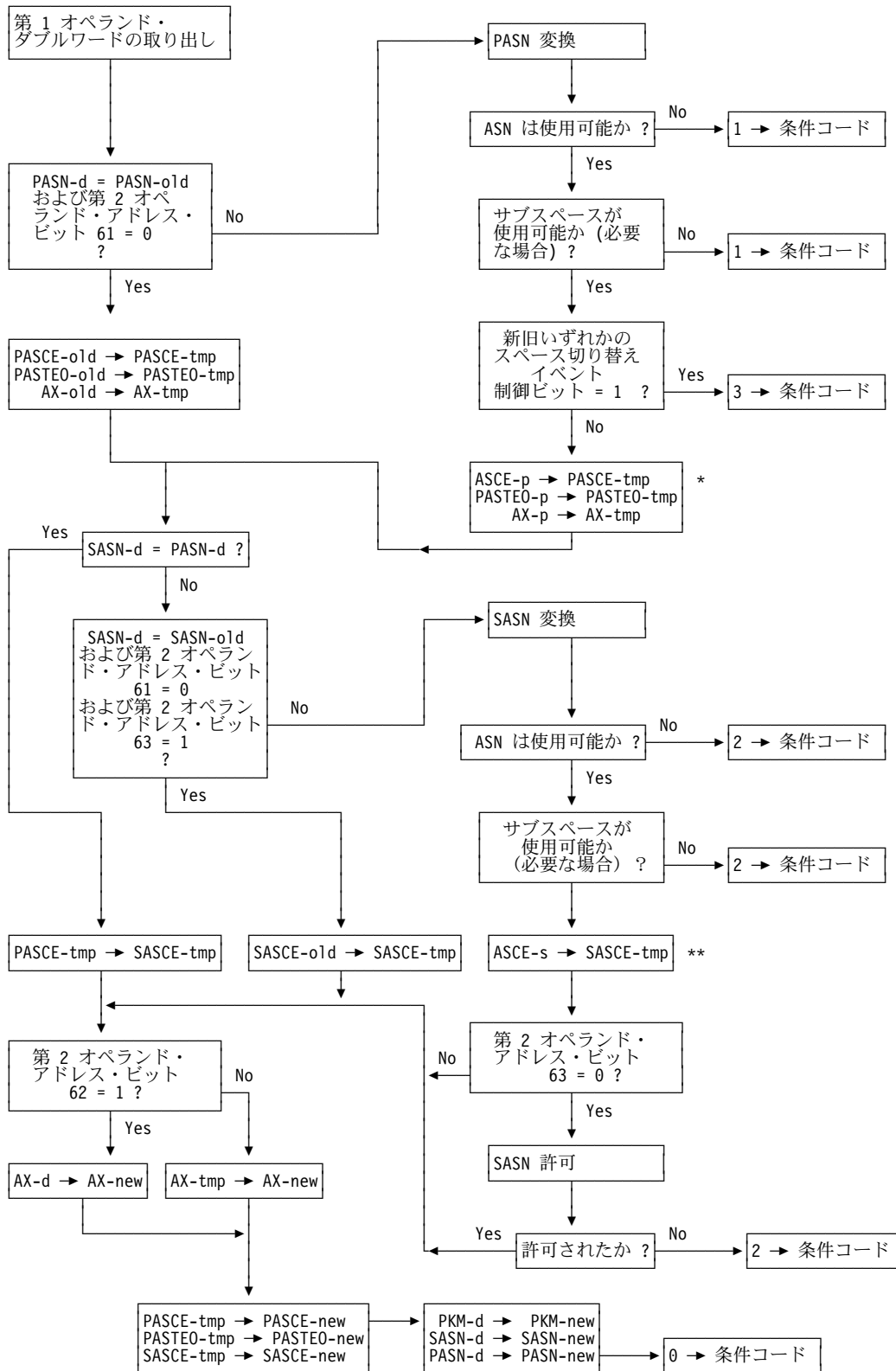
ASN 第 2 テーブル・エントリーのフィールド	以下の一部としてアクセスされるときにフィールドに使用される省略形	
	PASN 変換	SASN 変換
1-29	-	ATO-s
32-47	AX-p	-
48-59	-	ATL-s
64-127	ASCE-p ¹	ASCE-s ¹

説明:

- この場合はこのフィールドは使用されません。

¹ サブスペース置換操作により、ASCE-p から ASCE-rp が形成され、ASCE-s から ASCE-rs が形成されます。

図 10-14. LOAD ADDRESS SPACE PARAMETERS について使用される省略形の要約

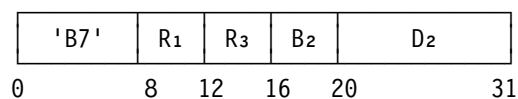


*: サブスペース置換が行われた場合は、PASCE-tmp は ASCE-rp です。
 **: サブスペース置換が行われた場合は、SASCE-tmp は ASCE-rs です。

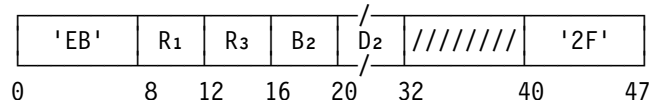
図 10-15. LOAD ADDRESS SPACE PARAMETERS の実行

LOAD CONTROL

LCTL R₁,R₃,D₂(B₂) [RS]



LCTLG R₁,R₃,D₂(B₂) [RSE]



制御レジスタ R₁ から始まり、制御レジスタ R₃ で終わる一連のレジスタのビット位置に、第 2 オペランド・アドレスで指定される記憶位置から始まる記憶域の、必要数の記憶位置の内容がロードされます。

LOAD CONTROL (LCTL) の場合は、制御レジスタのビット位置 32-63 に、第 2 オペランド・アドレスから始まる連続したワードがロードされ、レジスタのビット 0-31 は変更されません。LOAD CONTROL (LCTLG) では、制御レジスタのビット位置 0-63 に、第 2 オペランド・アドレスから始まる連続したダブルワードがロードされます。これらの制御レジスタは、制御レジスタ R₁ から始まり制御レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従ってロードされます。制御レジスタ 0 は、制御レジスタ 15 の後に続きます。

制御レジスタにロードされる情報は、この命令の実行が終了した時点で有効になります。

特別な条件

第 2 オペランドは、LCTL の場合はワード境界に、そして LCTLG の場合はダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外

- 指定例外

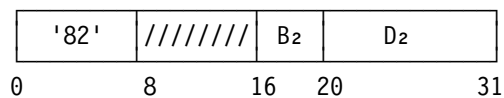
プログラミング上の注意:

1. これまで使用されていなかった制御レジスタ位置を使用する新しい機能を定義した場合、既存のプログラムを正しく動作させるためには、未割り当ての位置に 0 のみがロードされるようにしてください。
2. モデルによっては、制御レジスタのロードにかなり時間がかかることがあります。特に、重要なパラメータに変更がある場合は、所要時間が長くなります。

例えば、制御レジスタ 9-11 の中のプログラム・イベント記録パラメータを変更したり使用可能にした結果として、TLB のエントリが消去されることがあります。したがって、プログラムでは、可能な限り不要な制御レジスタ・ロードは回避するようにしてください。制御レジスタ 9-11 をロードするときに、ほとんどのモデルは、制御レジスタ 9 のビットが 0 である場合を想定して最適化を行います。

LOAD PSW

LPSW D₂(B₂) [S]



現 PSW が、第 2 オペランド・アドレスが指す記憶位置にあるダブルワードの内容から形成された 16 バイトの PSW で置き換えられます。

このダブルワードのビット 12 は 1 でなければなりません。さもないと、モデルによっては指定例外が認識されることがあります。

ダブルワードのビット 0-32 が、ビット 12 の値を反転させて、現 PSW のビット位置 0-32 に入れられます。ダブルワードのビット 33-63 は、現 PSW のビット位置 97-127 に入れられます。現 PSW のビット 33-96 は 0 に設定されます。

オペランドが取り出される前と後、さらに操作の完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。モデルによっては、オペランドのビット 12 が 0 の場合に、指定例外が認識されることがあります。

この命令によりロードする PSW フィールドについては、オプションとしてビット 12 が検査できる以外は、ロードの前に妥当性検査は行われません。しかし、新たにロードされた PSW について以下のいずれかが該当する場合は、指定例外が認識され、プログラム割り込みが発生します。

- ビット 0、2-4、12、または 24-30 のいずれかが 1 である。
- ビット 31 と 32 が両方とも 0 で、ビット 97-103 がすべて 0 ではない。
- ビット 31 が 1 で、ビット 32 が 0 である。

これらの場合は、操作は完了し、結果の命令長コードは 0 になります。

PSW がロードされた後の指定例外に関するテストについては、6-9ページの『早期例外認識』で説明されています。この例外は、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

結果の条件コード: コードは、ロードされた新しい PSW で指定されているとおりに設定されます。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外

プログラミング上の注意: 第 2 オペランドは、ESA/390 PSW のフォーマットになっていることが必要です。オペランドのビット 12 が 0 の場合は、LOAD PSW の実行中または実行後に指定例外が認識されません。

LOAD PSW EXTENDED

LPSWE D₂(B₂) [S]

'B2B2'	B ₂	D ₂
0	16 20	31

現 PSW が、16 バイトの第 2 オペランドの内容で置き換えられます。

オペランドが取り出される前と後、さらに操作の完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

この命令によりロードする値について、ロードの前に妥当性検査は行われません。しかし、新たにロードされた PSW について以下いずれかが該当する場合は、指定例外が認識され、プログラム割り込みが起きます。

- 未割り当てのビット (0、2-4、24-30、または 33-63) のいずれかが 1 である。
- ビット 12 が 1 である。
- ビット 31 が 0、ビット 32 が 1 であり、ビット 64-96 がすべて 0 ではない。
- ビット 31 および 32 が両方とも 0 であり、ビット 64-103 がすべて 0 ではない。
- ビット 31 が 1 で、ビット 32 が 0 である。

これらの場合は、操作は完了し、結果の命令長コードは 0 になります。

PSW がロードされた後の指定例外に関するテストについては、6-9ページの『早期例外認識』で説明されています。この例外は、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

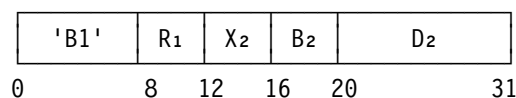
結果の条件コード: コードは、ロードされた新しい PSW で指定されているとおりに設定されます。

プログラム例外:

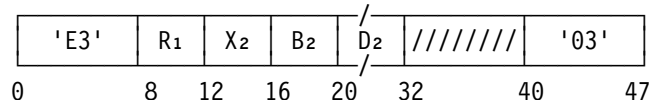
- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外

LOAD REAL ADDRESS

LRA R₁, D₂ (X₂, B₂) [RX]



LRAG R₁, D₂ (X₂, B₂) [RXE]



24 ビットまたは 31 ビットのアドレッシング・モードの LOAD REAL ADDRESS (LRA) では、第 2 オペランドの仮想アドレスに対応する 64 ビットの実アドレスのビット 0-32 がすべて 0 である場合は、その実アドレスのビット 32-63 が汎用レジスタ R₁ のビット位置 32-63 に入れられ、レジスタのビット 0-31 は変更されません。この実アドレスのビット 0-32 がすべて 0 ではない場合は、特殊操作例外が認識されます。

64 ビット・アドレッシング・モードの LRA の場合、およびアドレッシング・モードに関係なく LOAD REAL ADDRESS (LRAG) の場合は、第 2 オペランドの仮想アドレスに対応する 64 ビットの実アドレスが、汎用レジスタ R₁ に入れられます。

X₂、B₂、および D₂ フィールドに指定されている仮想アドレスは、DAT がオンかオフに関係なく、動的アドレス変換機能を使用して変換されます。

DAT は、アドレス・スペース制御エレメントを使用して実行されます。使用するエレメントは、次の表に示すように、アドレス・スペース制御ビット (PSW のビット 16 および 17) の現在の値によって決まります。

PSW の

ビット 16 および 17 DAT により使用されるアドレス・スペース制御エレメント

00	制御レジスタ 1 の内容
10	制御レジスタ 7 の内容
01	B ₂ フィールドに指定されているアクセス・レジスタに、アクセス・レジスタ変換 (ART) プロセスを適用することによって得られるアドレス・スペース制御エレメント
11	制御レジスタ 13 の内容

ART および DAT は、それぞれ、ART 索引バッファ (ALB) および変換索引バッファ (TLB) を利用して実行することができます。

仮想アドレスの計算は、現 PSW のビット 31 および 32 に指定されている現アドレッシング・モードに従って行われます。

領域テーブル・エントリ (使用される場合) のアドレス、および、セグメント・テーブル・エントリとページ・テーブル・エントリのアドレスは、現アドレッシング・モードに関係なく、64 ビット・アドレスとして扱われます。これらのエントリのアドレスが実アドレスとして扱われるか絶対アドレスとして扱われるかは、予測不能です。

ART (適用される場合) および DAT の両方が完了可能であり、特殊操作例外が認識されていない場合、つまり、アドレス・スペース制御エレメントが入手可能であり、各 DAT テーブル内のエントリがそのテーブルの範囲内にあり、そのエントリの I ビットが 0 であり、かつ、24 ビットまたは 31 ビットのアドレッシング・モードの LRA において、結果の実アドレスのビット 0-32 が 0 である場合は、条件コード 0 が設定されます。変換済みアドレスについて、境界合わせが適正かどうか、またはアドレッシング例外や記憶保護例外がないかどうかは、検査されません。

PSW ビット 16 および 17 が 2 進数 01 であるときに、通常は次の表に示す例外のいずれかを引き起こす条件が原因でアドレス・スペース制御エレメントを取得できない場合は、(1) その例外に割り当てられている割り込みコードが汎用レジスタ R₁ のビット位置 48-63 に入れられ、そのレジスタのビット 32 は 1 に設定され、ビット 33-47 は 0 に設定され、ビット 0-31 は無変更のまま残され、そして、(2) この命令は完了し、条件コード 3 が設定されます。

例外名	原因	コード (16進数)
ALET 指定例外	アクセス・リスト・エントリー・トークン (ALET) のビット 0-6 がすべて 0 ではない。	0028
ALEN 変換例外	アクセス・リスト・エントリー (ALE) がリストの外部にあるか無効 (ビット 0 が 1) である。	0029
ALE シーケンス例外	ALET 内の ALE シーケンス番号 (ALESN) が ALE 内の ALESN に等しくない。	002A
ASTE 妥当性例外	ASN 第 2 テーブル・エントリー (ASTE) が無効 (ビット 0 が 1)。	002B
ASTE シーケンス例外	ALE 内の ASTE シーケンス番号 (ASTESN) が ASTE 内の ASTESN に等しくない。	002C
拡張権限例外	ALE 専用ビットが 0 ではなく、ALE 許可指標 (ALEAX) が拡張許可指標 (EAX) に等しくなく、EAX により選択された 2 次ビットが権限テーブルの外部にあるかまたは 0 である。	002D

ART が正常に完了したときは、DAT の実行を介して操作は続行されます。

セグメント・テーブル・エントリーがテーブルの外部にあり、そのエントリーの実アドレスまたは絶対アドレスのビット 0-32 がすべて 0 である場合は、条件コード 3 が設定され、エントリー・アドレスのビット 32-63 が汎用レジスタ R₁ のビット位置 32-63 に入れられ、レジスタのビット 0-31 は変更されません。アドレスのビット 0-32 がすべて 0 ではない場合は、結果は以下の表のようになります。

64 ビット・アドレッシング・モードの LRA の場合、または任意のアドレッシング・モードの LRAG の場合、セグメント・テーブル・エントリーの I ビットが 1 のときは、条件コード 1 が設定され、セグメント・テーブル・エントリーの 64 ビットの実アドレスまたは絶対アドレスが、汎用レジスタ R₁ に入れられます。LRA

が 24 ビットまたは 31 ビットのアドレッシング・モードのときは、セグメント・テーブル・エントリーのアドレスのビット 0-32 がすべて 0 であれば、汎用レジスタ R₁ のビット 0-31 が変更されない点を除けば、結果は同じです。アドレスのビット 0-32 がすべて 0 ではない場合は、結果は以下の表のようになります。

64 ビット・アドレッシング・モードの LRA の場合、または任意のアドレッシング・モードの LRAG の場合、ページ・テーブル・エントリーの I ビットが 1 のときは、条件コード 2 が設定され、ページ・テーブル・エントリーの 64 ビットの実アドレスまたは絶対アドレスが、汎用レジスタ R₁ に入れられます。LRA が 24 ビットまたは 31 ビットのアドレッシング・モードのときは、ページ・テーブル・エントリーのアドレスのビット 0-32 がすべて 0 であれば、汎用レジスタ R₁ のビット 0-31 が変更されない点を除けば、結果は同じです。アドレスのビット 0-32 がすべて 0 ではない場合は、結果は以下の表のようになります。

汎用レジスタ R₁ に入れられるセグメント・テーブル・エントリーまたはページ・テーブル・エントリーのアドレスは、変換のときに使用されたアドレスのタイプに応じて、実アドレスまたは絶対アドレスになります。

通常では次の表に示す例外のいずれかを起こす原因となる条件が存在する場合は、(1) その例外に割り当てられている割り込みが汎用レジスタ R₁ のビット 48-63 に入れられ、そのレジスタのビット 32 は 1 に設定され、ビット 33-47 は 0 に設定され、ビット 0-31 は無変更のまま残され、(2) 命令は完了し、条件コード 3 が設定されます。

例外名	原因	コード (16進数)
ASCE タイプ例外	使用されているアドレス・スペース制御エレメント (ASCE) が領域第 2 テーブル指定であり、仮想アドレスのビット 0-10 がすべて 0 ではない。または、ASCE が領域第 3 テーブル指定であり、仮想アドレスのビット 0-21 がすべて 0 ではない。または、ASCE がセグメント・テーブル指定であり、仮想アドレスのビット 0-32 がすべて 0 ではない。	0038

例外名	原因	コード (16 進 数)
領域第 1 変換 例外	仮想アドレスの領域第 1 指標により選択される領域第 1 テーブル・エントリーが、テーブルの外部にあるか、または無効である。	0039
領域第 2 変換 例外	仮想アドレスの領域第 2 指標により選択される領域第 2 テーブル・エントリーが、テーブルの外部にあるか、または無効である。	003A
領域第 3 変換 例外	仮想アドレスの領域第 3 指標により選択される領域第 3 テーブル・エントリーが、テーブルの外部にあるか、または無効である。	003B
セグメント変 換例外	仮想アドレスのセグメント指標部分により選択されるセグメント・テーブル・エントリーが、テーブルの外部にある (エントリー・アドレスのビット 0-32 がすべて 0 ではない場合のみ)。または、そのセグメント・テーブル・エントリーが無効である (LRA の場合のみ、および、24 ビットまたは 31 ビットのアドレッシング・モードで、エントリー・アドレスのビット 0-32 がすべて 0 ではない場合のみ)。	0010
ページ変換例 外	仮想アドレスのセグメント指標部分により選択されるページ・テーブル・エントリーが無効である (LRA の場合のみ、および、24 ビットまたは 31 ビットのアドレッシング・モードで、エントリー・アドレスのビット 0-32 がすべて 0 ではない場合のみ)。	0011

特別な条件

24 ビットまたは 31 ビットのアドレッシング・モードの LRA において、結果の 64 ビット実アドレスのビット 0-32 がすべて 0 ではない場合は、特殊操作例外が認識されます。

有効アクセス・リスト指定を取り出すために ART により使用されるアドレス、あるいは、ALE、ASTE、または権限テーブル・エントリーが、構成内で使用可能でない記憶位置を指定している場合、または、領域テーブル・エントリー (存在する場合)、セグメント・テーブル・エントリー、またはページ・テーブル・エントリーを取り出すために使用されるアドレスが、構成内で使用可能でない記憶位置を指定している場合は、アドレッシング例外が認識されます。

アクセスされた領域テーブル・エントリー、あるいはセグメント・テーブル・エントリーまたはページ・テーブル・エントリーにおいて、I ビットが 0 でありフォーマット・エラーがある場合、つまり、6-34 ページの『変換指定例外』にリストされている理由のいずれかに該当する場合は、変換指定例外が認識されます。

領域テーブル・エントリーまたはセグメント・テーブル・エントリーのアドレスを計算するための加算の結果としてビット位置 0 からの繰り上がりが生じる場合、繰り上がりは無視されるか、またはアドレッシング例外が起きます。

何らかのアドレッシング例外が起きたときは、操作は抑止されます。

結果の条件コード:

- 0 変換は使用可能です。
- 1 セグメント・テーブル・エントリーが無効 (I ビットが 1) です。
- 2 ページ・テーブル・エントリーが無効 (I ビットが 1) です。
- 3 アドレス・スペース制御エレメントが使用可能でないか、領域テーブル・エントリーがテーブルの外部にあるかまたは無効 (I ビットが 1) であるか、またはセグメント・テーブル・エントリーがテーブルの外部にあります。あるいは、24 ビットまたは 31 ビットのアドレッシング・モードの LRA の場合のみ、エントリー・アドレスのビット 0-32 がすべて 0 ではないときに、セグメント・テーブル・エントリーまたはページ・テーブル・エントリーが無効 (I ビットが 1) です。

プログラム例外:

- アドレッシング例外 (有効アクセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、権限テーブル・エントリー、領域テーブル・エントリー、セグメント・テーブル・エントリー、またはページ・テーブル・エントリー)

- 特権命令例外
- 指定例外 (LRA のみ)
- 変換指定例外

プログラミング上の注意:

1. マルチプロセッシング構成で LOAD REAL ADDRESS を使用するときは、注意が必要です。INVALIDATE PAGE TABLE ENTRY は、他の CPU の TLB 内の対応するエントリーが消去される前に、記憶域内で I ビットを 1 に設定することがあるので、この CPU での LOAD REAL ADDRESS の実行と同時に他の CPU で INVALIDATE PAGE TABLE ENTRY を実行すると、矛盾した結果が生じることがあります。LOAD

REAL ADDRESS は記憶域内のテーブルにアクセスすることがあるので、対応する TLB エントリーがまだ消去されていなくても、ページ・テーブル・エントリーが無効 (条件コード 2) に見えることがあり、他の CPU で INVALIDATE PAGE TABLE ENTRY が完了するまで、その TLB エントリーが TLB 内に残されることがあります。LOAD REAL ADDRESS が完了してから、TLB 内の該当エントリーが消去されるまでの間に実行される可能性のある命令の数については、保証された上限はありません。

2. 10-43ページの図10-16 に、結果の汎用レジスタ R1 の内容および条件コードの要約を示してあります。

例外/原因/ エントリー・ アドレス・サイズ または結果の 実アドレス・サイズ	汎用レジスタ R ₁ の内容と条件コード									
	24 ビットまたは 31 ビット・ アドレッシング・モードの LRA					64 ビット・アドレッシング・ モードの LRA または任意の アドレッシング・モードの LRA				
	0-31	32	33-47	48-63	CC	0-31	32	33-47	48-63	CC
ALET 指定	U	1	0	0028	3	U	1	0	0028	3
ALEN 変換	U	1	0	0029	3	U	1	0	0029	3
ALE シーケンス	U	1	0	002A	3	U	1	0	002A	3
ASTE 妥当性	U	1	0	002B	3	U	1	0	002B	3
ASTE シーケンス	U	1	0	002C	3	U	1	0	002C	3
拡張権限	U	1	0	002D	3	U	1	0	002D	3
ASCE タイプ	U	1	0	0038	3	U	1	0	0038	3
領域第 1 変換	U	1	0	0039	3	U	1	0	0039	3
領域第 2 変換	U	1	0	003A	3	U	1	0	003A	3
領域第 3 変換	U	1	0	003B	3	U	1	0	003B	3
セグメント変換/テ ーブル外エントリー/ エントリー・アドレス <2GB	U	0	EA3	EA4	3	U	0	EA3	EA4	3
セグメント変換/テ ーブル外エントリー/ エントリー・アドレス >= 2GB	U	1	0	0010	3	U	1	0	0010	3
セグメント変換/ I ビットが 1/ エントリー・アドレス < 2GB	U	0	EA3	EA4	1	EA1	EA2	EA3	EA4	1
セグメント変換/ I ビットが 1/ エントリー・アドレス >= 2GB	U	1	0	0010	3	EA1	EA2	EA3	EA4	1
ページ変換/ I ビットが 1/ エントリー・アドレス < 2GB	U	0	EA3	EA4	2	EA1	EA2	EA3	EA4	2
ページ変換/ I ビットが 1/ エントリー・アドレス >= 2GB	U	1	0	0011	3	EA1	EA2	EA3	EA4	2
実アドレス < 2GB	U	0	RA3	RA4	0	RA1	RA2	RA3	RA4	0
実アドレス >= 2GB			特殊操作例外			RA1	RA2	RA3	RA4	0

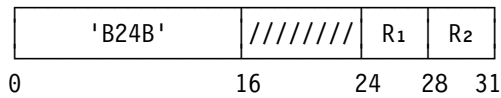
説明:

EA1 エントリー・アドレスのビット 0-31。
EA2 エントリー・アドレスのビット 32。
EA3 エントリー・アドレスのビット 33-47。
EA4 エントリー・アドレスのビット 48-63。
RA1 結果の実アドレスのビット 0-31。
RA2 結果の実アドレスのビット 32。
RA3 結果の実アドレスのビット 33-47。
RA4 結果の実アドレスのビット 48-63。
U 変更されません。

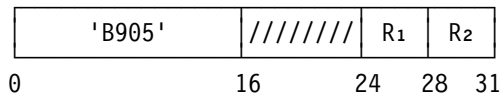
図 10-16. 結果の要約: LOAD REAL ADDRESS

LOAD USING REAL ADDRESS

LURA R₁,R₂ [RRE]



LURAG R₁,R₂ [RRE]



LOAD USING REAL ADDRESS (LURA) では、汎用レジスタ R₂ の内容が指す実記憶位置にあるワードが汎用レジスタ R₁ のビット位置 32-63 に入れられ、ビット位置 0-31 は変更されません。LOAD USING REAL ADDRESS (LURAG) では、その実記憶位置にあるダブルワードが、汎用レジスタ R₁ のビット位置 0-63 に入れられます。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 40-63 が実記憶位置を指定し、ビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 33-63 が実記憶位置を指定し、ビット 0-33 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 0-63 が実記憶位置を指定します。

記憶域ワードまたは記憶域ダブルワードを指定するアドレスは実アドレスなので、動的アドレス変換は行われません。

特別な条件

汎用レジスタ R₂ の内容は、LURA の場合はワード境界上の位置を、そして LURAG の場合はダブルワード境界上の位置を指定していなければなりません。さもないと、指定例外が認識されます。

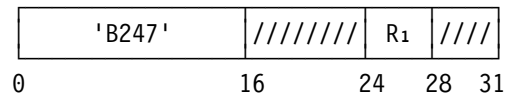
条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (汎用レジスタ R₂ に指定されているアドレス)
- 特権命令例外
- 記憶保護例外 (取り出し: オペランド 2、キー制御保護)
- 指定例外

MODIFY STACKED STATE

MSTA R₁ [RRE]



R₁ フィールドに指定されている一対の汎用レジスタのビット位置 32-63 の内容が、リンケージ・スタック内の最後の状態エントリーの変更可能域 (バイト位置 152-159) に入れられます。

R₁ フィールドは、汎用レジスタの奇偶ペアのうちの偶数番号のレジスタを指定します。

最後の状態エントリーがどのようにして見つけ出されるかについては、5-75ページの『アンスタッキング・プロセス』で説明されています。状態エントリーはリンケージ・スタック内に残され、制御レジスタ 15 の中のリンケージ・スタック・エントリー・アドレスは変更されません。

リンケージ・スタックに対する参照には、キー制御保護は適用されませんが、低アドレス保護およびページ保護は適用されます。

特別な条件

R₁ が奇数の場合は、指定例外が認識されます。

CPU は、1 次スペース・モード、アクセス・レジスタ・モード、またはホーム・スペース・モードになっていなければなりません。さもないと、特殊操作例外が認識されます。

アンスタッキング・プロセスで、スタック空例外、スタック指定例外、またはスタック・タイプ例外が認識されることがあります。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-45ページの図10-17 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (キー制御保護の場合を除き、リンケージ・スタック・エントリーの取り出しおよび格納)

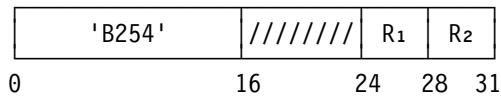
- 特殊操作例外
- 指定例外
- スタック空例外
- スタック指定例外
- スタック・タイプ例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 ハーフワードについてのアクセス例外。
- 7.B DAT がオフのため、または CPU が 2 次スペース・モードであるために起きる特殊操作例外。
- 8.A R₁ が奇数のために起きる指定例外。
- 8.B.1 現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。
- 8.B.2 現エントリーが状態エントリーまたはヘッダー・エントリーでないために起きるスタック・タイプ例外。
- 注: 例外 8.B.3 - 8.B.7 が起きることがあるのは、現エントリーがヘッダー・エントリーである場合のみです。
- 8.B.3 ヘッダー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。
- 8.B.4 ヘッダー・エントリー内の逆方向スタック・エントリー妥当性ビットが 0 であるために起きるスタック空例外。
- 8.B.5 前のエントリーのエントリー記述子についてのアクセス例外 (取り出し)。前のエントリーとは、現 (ヘッダー) エントリー内の逆方向スタック・エントリー・アドレスが指しているエントリーです。
- 8.B.6 前のエントリーがヘッダー・エントリーであるために起きるスタック指定例外。
- 8.B.7 前のエントリーが状態エントリーでないために起きるスタック・タイプ例外。
- 8.B.8 状態エントリーの変更可能域についてのアクセス例外 (格納)。

図 10-17. 実行の優先順位: MODIFY STACKED STATE

MOVE PAGE

MVPG R₁,R₂ [RRE]



第 1 オペランドが第 2 オペランドで置き換えられます。第 1 オペランドと第 2 オペランドは、どちらも 4K バイトで、4K バイト境界に位置合わせされます。結果は条件コードで示されます。汎用レジスター 0 に指定されているキーを使用して、第 1 オペランド位置または第 2 オペランド位置のいずれか一方 (両方ではなく) へのアクセスが行われます。さもなければ、PSW キーを使用して、いずれかのオプション位置へのアクセスが行われます。

第 1 オペランドおよび第 2 オペランドの左端バイトの位置は、それぞれ、汎用レジスター R₁ および R₂ の内容によって指定されます。

汎用レジスター R₁ および R₂ の中のアドレスの取り扱い、アドレッシング・モードに応じて異なります。24 ビット・アドレッシング・モードでは、汎用レジスターのビット位置 40-51 の内容の右側に 12 個の 0 を付加したものがアドレスであり、ビット 0-39 および 52-63 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスターのビット位置 33-51 の内容の右側に 12 個の 0 を付加したものがアドレスであり、ビット 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスターのビット位置 0-51 の内容の右側に 12 個の 0 を付加したものがアドレスであり、ビット 52-63 は無視されます。

汎用レジスター 0 のビット 56-59 は、指定されたアクセス・キーとして使用されます。汎用レジスター 0 のビット 52 が 1 のときは、指定されたアクセス・キーを第 1 オペランドのアクセスのための使用することを指定します。ビット 53 は、第 2 オペランドについて同じことを指定します。ビット 52 と 53 が両方とも 1 の場合は、指定例外が認識されます。汎用レジスター 0 のビット 54 は宛先参照意図ビットで、ビット 55 は条件コード・オプション・ビットです。汎用レジスター 0 のビット 48-51 はすべて 0 でなければなりません。さもないと、指定例外が認識されます。汎用レジスター 0 のビット 0-47 および 60-63 は無視されます。

10-47 ページの図 10-18 は、上記で説明したレジスターの内容を示しています。

汎用レジスター 0 のビット 52 が 1 のときは、PSW キーを使用して第 2 オペランド位置への取り出しアクセスが行われ、汎用レジスター 0 に指定されているキーを使用して第 1 オペランド位置への格納アクセスが行われ、汎用レジスター 0 のビット 53 が 1 のときは、汎用レジスターに指定されているキーを使用して第 2 オペランド位置への取り出しアクセスが行われ、PSW キーを使用して第 1 オペランド位置への格納アクセスが行われます。ビット 52 と 53 が両方とも 0 のときは、PSW キーを使用して両方のオペランドがアクセスされます。

4K バイトが移動されてしまうと、条件コード 0 が設定されます。

ページ変換例外条件が存在していても、条件コード・オプション・ビット (汎用レジスター 0 のビット 55) が 1 の場合は、例外は認識されません。代わりに、条件コード 1 または 2 が設定されます。第 1 オペランドについてページ変換例外条件が存在し、第 2 オペランドについてはその条件が存在しない場合は、条件コード 1 が設定されます。第 2 オペランドについてページ変換例外条件が存在する場合は、第 1 オペランドについてその条件が存在するかどうかに関係なく、条件コード 2 が設定されます。

両方のオペランドについてアクセス例外が認識される可能性がある場合は、どちらのオペランドについて例外が認識されるかは、予測不能です。例外の 1 つが、条件コード 1 または 2 が設定される原因となるようなページ変換例外である場合は、他方のオペランドについて、アクセス例外が認識されるか、それとも条件コード 1 または 2 が設定されるかは、予測不能です。

チャンネル・プログラムまたは他の CPU から見た場合、主記憶機構への参照は必ずしも単一アクセス参照ではなく、また、必ずしも左から右への方向に実行されるわけではありません。

特別な条件

問題プログラム状態では、汎用レジスターのビット 52 またはビット 53 のどちらかが 1 のときは、汎用レジスター 0 に指定されているアクセス・キーが有効である場合、つまり、制御レジスター 3 の中の対応する PSW キー・マスク・ビットが 1 である場合に限り、操作が行われます。そうでないときは、特権命令例外が認識されます。監視プログラム状態では、指定されたアクセス・キーの値は何であっても有効です。ビット 52 と 53 が両方とも 0 のときは、汎用レジスター 0 の中のアクセス・キーに対する妥当性検査は行われません。

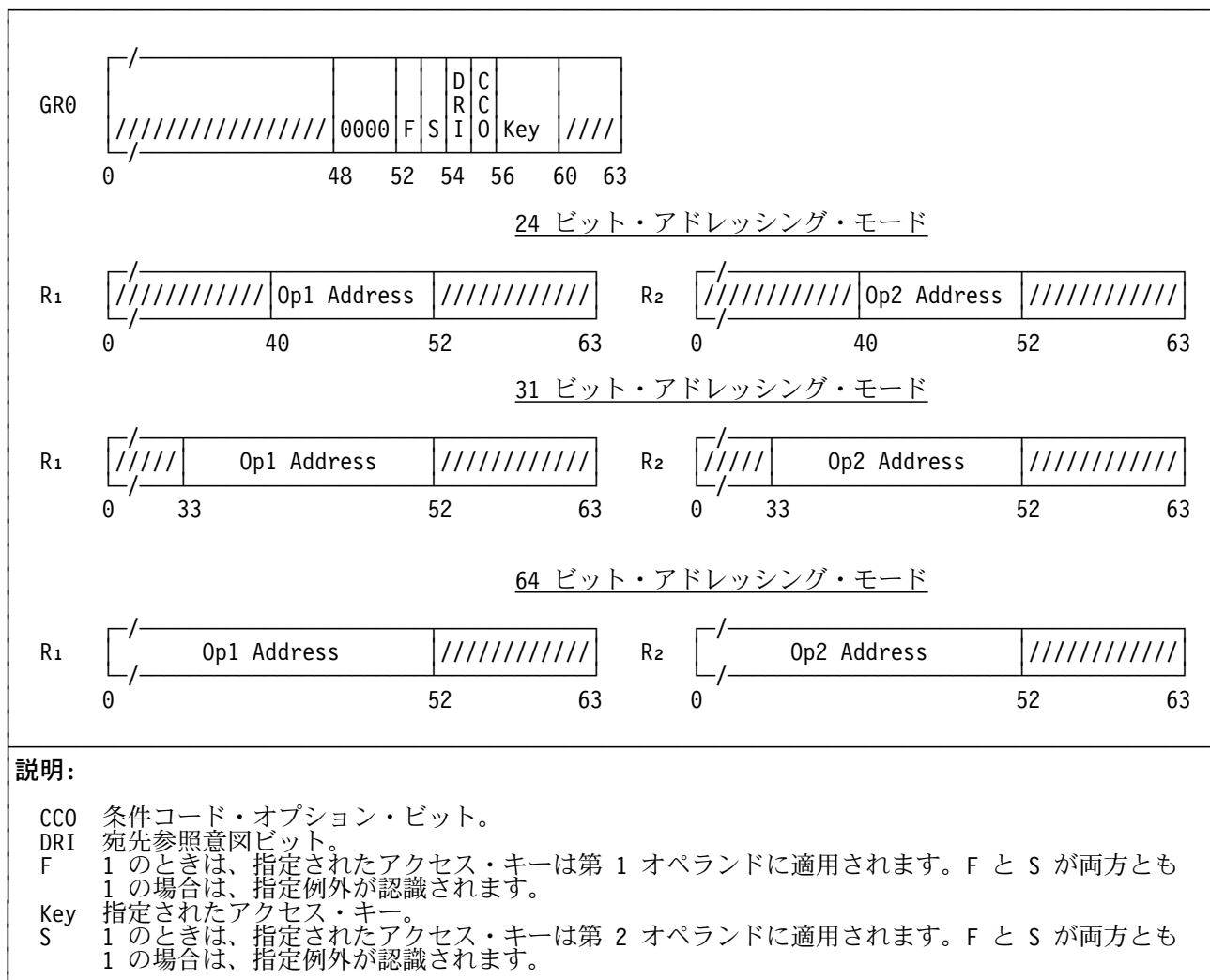


図 10-18. MOVE PAGE で使用するレジスタの内容

問題プログラム状態では、汎用レジスタ 0 のビット 52 と 53 が両方とも 1 で、汎用レジスタ 0 の中のアクセス・キーが PSW キー・マスクにより許可されていない場合は、指定例外または特権命令例外のどちらが認識されるかは、予測不能です。

結果の条件コード:

- 0 データが移動されました。
- 1 条件コード・オプション・ビットが 1 で、第 1 オペランドが指すページ・テーブル・エントリが無効で、第 2 オペランドが指すページ・テーブル・エントリは有効です。
- 2 条件コード・オプション・ビットが 1 で、第 2 オペランドが指すページ・テーブル・エントリが無効です。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 特権命令例外 (指定されたアクセス・キー、および、問題プログラム状態で選択された PSW キー・マスク・ビットが 0)
- 指定例外

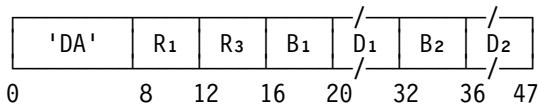
プログラミング上の注意:

1. ほとんどのモデルでは、MOVE PAGE (または複数のページを移動する MOVE PAGE 命令のループ) の方が、MOVE LONG 命令 (または同じ機能を実行する MOVE (MVC) 命令のループ) より優れたパフォーマンスを達成できます。MOVE PAGE の方が優れたパフォーマンスを提供するかどうかは、制御プログラムの仕様と、制御プログラムがページ変換例外を処理する方式によって決まります。

2. MOVE PAGE 以外の命令を使用して第 1 オペランドを参照する意図がある場合は、宛先参照意図ビットを 1 に設定する必要があります。このビットを設定することにより、制御プログラムがページ変換例外を効率的に処理できるようになります。
3. 条件コード・オプション・ビットは、ESA/390 ページ移動機能 1 の MOVE PAGE 命令との互換性を確保します。このビットは、MVS/ESA HSPSERV マクロ展開で使用するためのものです。
4. 条件コード・オプション・ビットが 0 の場合は、通常、この命令により設定される条件コードを調べる必要はありません。
5. MOVE PAGE により認識されたページ変換例外を原因とするプログラム割り込みで格納される情報については、第 3 章、『記憶機構』の中の『割り当て済みの記憶位置』のセクションの実記憶位置 162 および 168-175 についての定義を参照してください。

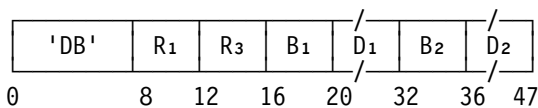
MOVE TO PRIMARY

MVCP D₁(R₁,B₁),D₂(B₂),R₃ [SS]



MOVE TO SECONDARY

MVCS D₁(R₁,B₁),D₂(B₂),R₃ [SS]



第 1 オペランドが第 2 オペランドで置き換えられます。一方のオペランドは 1 次アドレス・スペース内にあり、他方は 2 次アドレス・スペース内にあります。1 次スペース内のオペランドへのアクセスは、PSW キーを使用して行われます。2 次スペース内のオペランドへのアクセスは、第 3 オペランドに指定されているキーを使用して行われます。

第 1 オペランドおよび第 2 オペランドのアドレスは仮想アドレスで、一方のオペランド・アドレスは 1 次アドレス・スペース制御エレメントを使用して変換され、もう一方は 2 次アドレス・スペース制御エレメントを使用して変換されます。現 PSW 内のアドレス・スペース制御ビットが、1 次スペース・モードまたは 2 次スペース・

モードのどちらを指定していても、オペランド・アドレス変換は同じ方法で行われます。

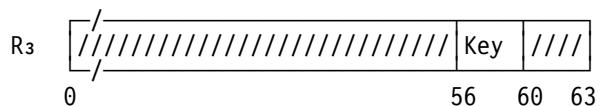
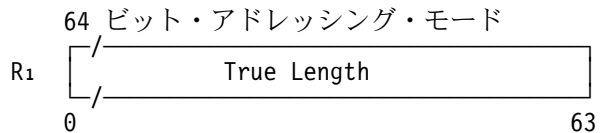
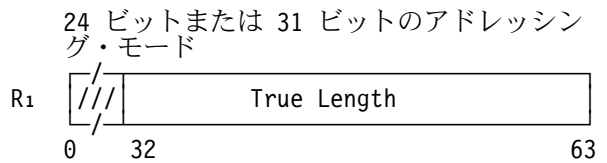
MOVE TO PRIMARY では、2 次スペースから 1 次スペースへの移動が行われます。第 1 オペランド・アドレスは、1 次アドレス・スペース制御エレメントを使用して変換され、第 2 オペランド・アドレスは、2 次アドレス・スペース制御エレメントを使用して変換されます。

MOVE TO SECONDARY では、1 次スペースから 2 次スペースへの移動が行われます。第 1 オペランド・アドレスは、2 次アドレス・スペース制御エレメントを使用して変換され、第 2 オペランド・アドレスは、1 次アドレス・スペース制御エレメントを使用して変換されます。

汎用レジスター R₃ のビット位置 56-59 は、2 次スペース・アクセス・キーとして使用されます。このレジスターのビット位置 0-55 および 60-63 は無視されます。

汎用レジスター R₁ には、真の長さと呼ばれる符号なしの 2 進整数が含まれています。24 ビットまたは 31 ビットのアドレッシング・モードでは、このレジスターのビット位置 32-63 が真の長さを示し、ビット位置 0-31 の内容は無視されます。64 ビット・アドレッシング・モードでは、このレジスターのビット位置 0-63 に真の長さが入っています。

上記で述べた汎用レジスターの内容は、以下のとおりです。



第 1 オペランドと第 2 オペランドは同じ長さであり、それは有効長と呼ばれます。有効長は、真の長さとの、どちらか小さい方の長さです。第 1 オペランドおよび第 2 オペランドに関するアクセス例外は、オペランドの中の有効長の範囲内の部分についてのみ認識されま

す。有効長が 0 のときは、第 1 オペランドおよび第 2 オペランドのどちらについてもアクセス例外は認識されず、移動は行われません。

各記憶域オペランドは、左から右へ処理されます。記憶域オペランド一貫性の規則は、MOVE (MVC) と同じです。ただし、実記憶域内でオペランドがオーバーラップする場合は、共通の実記憶域位置の使用は必ずしも認識されません。

命令の実行の一環として、真の長さの値に従って条件コードが設定されます。真の長さが 256 またはそれ以下 (0 も含む) である場合は、真の長さと同様であり、条件コード 0 が設定されます。真の長さが 256 より大きい場合は、有効長は 256 で、条件コード 3 が設定されます。

MOVE TO PRIMARY と MOVE TO SECONDARY のどちらの場合も、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

2 次スペースがアクセス対象になるので、2 次スペース制御 (制御レジスター 0 の ビット 37) が 1 で、DAT がオンの場合のみ、操作が行われます。2 次スペース制御が 0 か、または DAT がオフのときは、特殊操作例外が認識されます。また、現 PSW 中のアドレス・スペース制御ビットがアクセス・レジスター・モードまたはホーム・スペース・モードを指定している場合も、特殊操作例外が認識されます。

問題プログラム状態では、この操作が行われるのは、2 次スペース・アクセス・キーが有効な場合、つまり、制御レジスター 3 中の対応する PSW キー・マスク・ビットが 1 の場合のみです。そうでないときは、特権命令例外が認識されます。監視プログラム状態では、2 次スペース・アクセス・キーの値は何であっても有効です。

図 10-19 に、例外と条件コードの認識の優先順位を示してあります。

結果の条件コード:

- 0 真の長さが 256 またはそれ以下。
- 1 --
- 2 --
- 3 真の長さが 256 より大きい。

プログラム例外:

- アクセス例外 (取り出し: 1 次仮想アドレス、オペランド 2、MVCS。取り出し: 2 次仮想アドレス、オペランド 2、MVCP。格納: 2 次仮想アドレス、オペランド 1、MVCS。格納: 1 次仮想アドレス、オペランド 1、MVCP)
- 特権命令例外 (問題プログラム状態で、選択された PSW キー・マスク・ビットが 0)
- 特殊操作例外

- | |
|--|
| 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7.A 命令の第 2 および第 3 ハーフワードについてのアクセス例外。 |
| 7.B 2 次スペース制御 (制御レジスター 0 のビット 37) が 0 か、DAT がオフであるか、または CPU がアクセス・レジスター・モードまたはホーム・スペース・モードであるために起きる特殊操作例外。 |
| 8. 問題プログラム状態で、選択された PSW キー・マスク・ビットが 0 であるために起きる特権命令例外。 |
| 9. 長さが 0 であるための完了。 |
| 10. オペランドについてのアクセス例外。 |

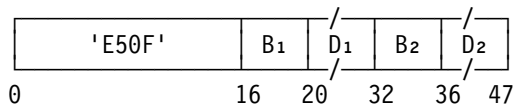
図 10-19. 実行の優先順位: MOVE TO PRIMARY および MOVE TO SECONDARY

プログラミング上の注意:

1. MOVE TO PRIMARY および MOVE TO SECONDARY をループの中で使用することにより、任意の長さの可変数のバイトを移動することができません。MOVE WITH KEY の項の『プログラミング上の注意』を参照してください。
2. MOVE TO PRIMARY および MOVE TO SECONDARY は、異なるアドレス・スペース間での移動のみに使用してください。ほとんどのモデルでは、この 2 つの命令は、MOVE WITH KEY、MOVE (MVC)、または MOVE LONG に比べて著しくパフォーマンスが低下することがあります。さらに、MOVE TO PRIMARY および MOVE TO SECONDARY の場合のオーバーラップ・オペランドについての定義と、MOVE (MVC)、MOVE WITH KEY、および MOVE LONG の場合のさらに厳密な定義との間には、互換性がありません。

MOVE WITH DESTINATION KEY

MVCDK D₁(B₁), D₂(B₂) [SSE]

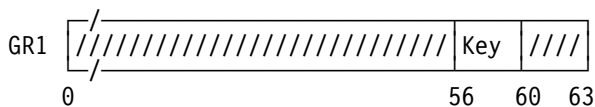
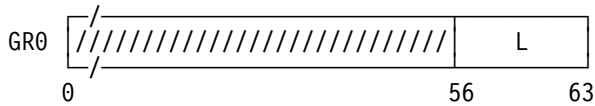


第 1 オペランドが第 2 オペランドで置き換えられます。汎用レジスタ 1 に指定されているキーを使用して、宛先オペランド位置へのアクセスが行われ、PSW キーを使用して、ソース・オペランド位置へのアクセスが行われます。

第 1 オペランドと第 2 オペランドは同じ長さで、この長さは汎用レジスタ 0 のビット 56-63 で指定されます。汎用レジスタ 0 のビット 0-55 は無視されます。

汎用レジスタ 1 のビット 56-59 は、指定されたアクセス・キーとして使用されます。汎用レジスタ 1 のビット 0-55 および 60-63 は無視されます。

汎用レジスタ 0 および 1 の内容は、以下のとおりです。



L は、各オペランドの第 1 バイトの右側にあるバイト数を示します。長さコード L の値の範囲は 0-255 であり、したがって各オペランドの長さ (バイト数) の範囲は 1-256 です。

第 2 オペランド位置への取り出しアクセスは PSW キーを使用して行われ、第 1 オペランド位置への格納アクセスは、汎用レジスタ 1 に指定されているキーを使用して行われます。

各オペランドは、左から右へ処理されます。実記憶域内にオペランドの破壊的オーバーラップがある場合は、第 1 オペランド位置に生じる結果は予測不能です。このように、破壊的オーバーラップがあるときに結果が予測不能である点を除けば、記憶域オペランド一貫性の規則は、MOVE (MVC) 命令の場合と同じです。

特別な条件

問題プログラム状態では、汎用レジスタ 1 に指定されているアクセス・キーが有効である場合、つまり、制御レジスタ 3 の中の対応する PSW キー・マスク・ビットが 1 である場合に限り、操作が行われます。そうでないときは、特権命令例外が認識されます。監視プログラム状態では、指定されたアクセス・キーの値は何であっても有効です。

条件コード: コードは変更されません。

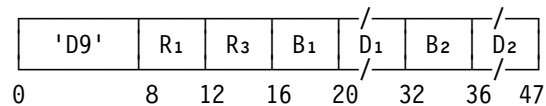
プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 特権命令例外 (問題プログラム状態で、選択された PSW キー・マスク・ビットが 0)

プログラミング上の注意: MOVE WITH SOURCE KEY 命令の『プログラミング上の注意』を参照してください。

MOVE WITH KEY

MVCK D₁(R₁, B₁), D₂(B₂), R₃ [SS]

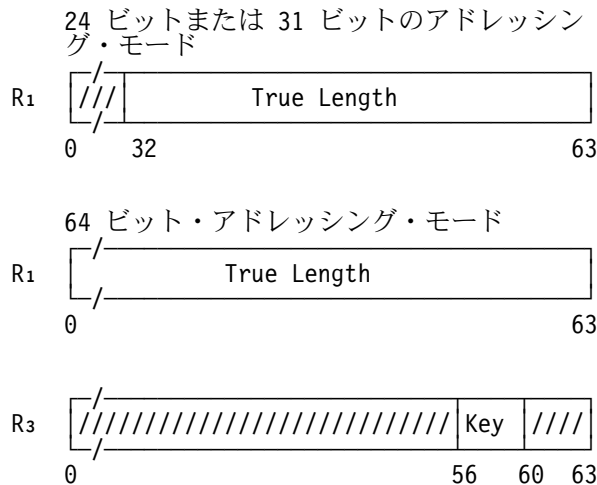


第 1 オペランドが第 2 オペランドで置き換えられます。第 3 オペランドに指定されているキーを使用して、第 2 オペランド位置への取り出しアクセスが行われ、PSW キーを使用して、第 1 オペランド位置への格納アクセスが行われます。

汎用レジスタ R₃ のビット位置 56-59 は、ソース・アクセス・キーとして使用されます。このレジスタのビット位置 0-55 および 60-63 は無視されます。

汎用レジスタ R₁ には、真の長さと呼ばれる符号なしの 2 進整数が含まれています。24 ビットまたは 31 ビットのアドレッシング・モードでは、このレジスタのビット位置 32-63 が真の長さを示し、ビット位置 0-31 の内容は無視されます。64 ビット・アドレッシング・モードでは、このレジスタのビット位置 0-63 に真の長さが入っています。

上記で述べた汎用レジスタの内容は、以下のとおりです。



第 1 オペランドと第 2 オペランドは同じ長さであり、それは有効長と呼ばれます。有効長は、真の長さ 256 の、どちらか小さい方の長さです。第 1 オペランドおよび第 2 オペランドに関するアクセス例外は、オペランドの中の有効長の範囲内の部分についてのみ認識されません。有効長が 0 のときは、第 1 オペランドおよび第 2 オペランドのどちらについてもアクセス例外は認識されず、移動は行われません。

各記憶域オペランドは、左から右へ処理されます。記憶域オペランドがオーバーラップしているときには、オペランドが一度に 1 バイトずつ処理されて必要なオペランド・バイトを取り出した直後に各結果バイトが格納される場合と同様の方法で、結果が取得されます。記憶域オペランド一貫性の規則は、MOVE (MVC) 命令の場合と同じです。

命令の実行の一環として、真の長さの値に従って条件コードが設定されます。真の長さが 256 またはそれ以下 (0 も含む) である場合は、真の長さと同じであり、条件コード 0 が設定されます。真の長さが 256 より大きい場合は、有効長は 256 で、条件コード 3 が設定されます。

特別な条件

問題プログラム状態では、この操作が行われるのは、ソース・アクセス・キーが有効な場合、つまり、制御レジスター 3 中の対応する PSW キー・マスク・ビットが 1 の場合のみです。そうでないときは、特権命令例外が認識されます。監視プログラム状態では、ソース・アクセス・キーの値は何であっても有効です。

図 10-20 に、例外と条件コードの認識の優先順位を示してあります。

結果の条件コード:

- 0 真の長さが 256 またはそれ以下。
- 1 --
- 2 --
- 3 真の長さが 256 より大きい。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 特権命令例外 (問題プログラム状態で、選択された PSW キー・マスク・ビットが 0)

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 および第 3 ハーフワードについてのアクセス例外。
- 8. 問題プログラム状態で、選択された PSW キー・マスク・ビットが 0 であるために起きる特権命令例外。
- 9. 長さが 0 のための完了。
- 10. オペランドについてのアクセス例外。

図 10-20. 実行の優先順位: MOVE WITH KEY

プログラミング上の注意:

- 1. MOVE WITH KEY をループの中で使用することにより、次のように、任意の長さの可変数のバイトを移動することができます。

```

LOOP   MVCK   D1(R1,B1),D2(B2),R3
        BC     8,END
        AHI   B1,256
        AHI   B2,256
        AHI   R1,-256
        B     LOOP
END     [Any instruction]

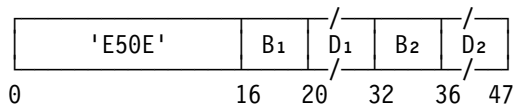
```

上記のプログラムは、24 ビットまたは 31 ビットのアドレッシング・モードで実行するためのものです。64 ビット・アドレッシング・モードでは、上記の AHI 命令の代わりに AGHI 命令を使用してください。

- 2. ほとんどのモデルでは、MOVE WITH KEY のパフォーマンスは、MOVE (MVC) および MOVE LONG 命令に比べて著しく低下することがあります。したがって、ソースとターゲットのキーが同じである場合は、MOVE WITH KEY は使用しないでください。

MOVE WITH SOURCE KEY

MVCSK D₁(B₁), D₂(B₂) [SSE]

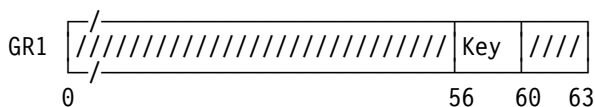
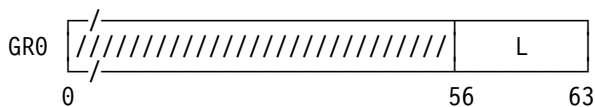


第 1 オペランドが第 2 オペランドで置き換えられます。汎用レジスタ 1 に指定されているキーを使用して、ソース・オペランド位置へのアクセスが行われ、PSW キーを使用して、宛先オペランド位置へのアクセスが行われます。

第 1 オペランドと第 2 オペランドは同じ長さで、この長さは汎用レジスタ 0 のビット 56-63 に指定されます。汎用レジスタ 0 のビット 0-55 は無視されます。

汎用レジスタ 1 のビット 56-59 は、指定されたアクセス・キーとして使用されます。汎用レジスタ 1 のビット 0-55 および 60-63 は無視されます。

汎用レジスタ 0 および 1 の内容は、以下のとおりです。



L は、各オペランドの第 1 バイトの右側にあるバイト数を示します。長さコード L の値の範囲は 0-255 であり、したがって各オペランドの長さ (バイト数) の範囲は 1-256 です。

汎用レジスタ 1 に指定されているキーを使用して、第 2 オペランド位置への取り出しアクセスが行われ、PSW キーを使用して、第 1 オペランド位置への格納アクセスが行われます。

各オペランドは、左から右へ処理されます。実記憶域内にオペランドの破壊的オーバーラップがある場合は、第 1 オペランド位置に生じる結果は予測不能です。このように、破壊的オーバーラップがあるときに結果が予測不能である点を除けば、記憶域オペランド一貫性の規則は、MOVE (MVC) 命令の場合と同じです。

特別な条件

問題プログラム状態では、汎用レジスタ 1 に指定されているアクセス・キーが有効である場合、つまり、制御レジスタ 3 の中の対応する PSW キー・マスク・ビットが 1 である場合に限り、操作が行われます。そうでないときは、特権命令例外が認識されます。監視プログラム状態では、指定されたアクセス・キーの値は何であっても有効です。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2。格納: オペランド 1)
- 特権命令例外 (問題プログラム状態で、選択された PSW キー・マスク・ビットが 0)

プログラミング上の注意:

1. 別々のキーを使用して取り出し保護されている 2 つの記憶域間で、両方向に交互にデータを移動する場合は、PSW キーを変更することなく、MOVE WITH SOURCE KEY および MOVE WITH DESTINATION KEY を使用することができます。ほとんどのモデルでは、この方法を使用する方が、PSW キーを変更するための SET PSW KEY FROM ADDRESS を伴う MOVE WITH KEY を使用するより、処理がはるかに高速になります。
2. MOVE WITH SOURCE KEY および MOVE WITH DESTINATION KEY は、異なるキーを持つ記憶域間で移動を行う場合のみ使用してください。ほとんどのモデルでは、この 2 つの命令は、MOVE (MVC) に比べて著しくパフォーマンスが低下することがあります。
3. MOVE WITH SOURCE KEY または MOVE WITH DESTINATION KEY をループの中で使用することにより、以下の例に示すように、可変数のバイトを移動することができます。この例では、開始の時点で、指定されたアクセス・キー、第 1 オペランド・アドレス、第 2 オペランド・アドレス、および各オペランドの長さが、それぞれ、汎用レジスタ 1-4 に入っているものと想定しています。各オペランドの長さは 32 ビットの符号付きの値として扱われ、負の値は 0 と見なされます。

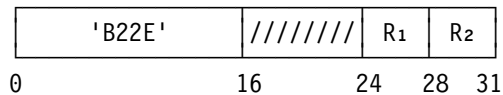

```

LTR    4,4
BC     12,END
AHI    4,-256
BC     12,LAST
LA     0,255
LOOP   MVCSK 0(2),0(3)
      LA     2,256(2)
      LA     3,256(3)
      AHI    4,-256
      BC     2,LOOP
LAST   LA     0,255(4)
      MVCSK 0(2),0(3)
END    [Any instruction]

```

PAGE IN

PGIN R₁,R₂ [RRE]



ページイン操作により、汎用レジスタ R₂ に指定されている拡張記憶ブロックから、汎用レジスタ R₁ に指定されている実記憶位置へ、4K バイトのブロックが転送されます。

汎用レジスタ R₂ のビット 32-63 は、拡張記憶域ブロック番号と呼ばれる 32 ビットの符号なし 2 進整数です。この番号は、拡張記憶域内の転送すべき 4K バイト・ブロックを指定します。この拡張記憶域ブロック番号が拡張記憶域内のアクセス不能ブロックを指している場合は、条件コード 3 が設定されます。

汎用レジスタ R₁ の内容は、主記憶域内の特定の 4K ブロックを指す実アドレスです。24 ビット・アドレッシング・モードでは、ビット 40-51 が目的のブロックを指定し、ビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、ビット 33-51 が目的のブロックを指定し、ビット 0-32 は無視されます。64 ビット・アドレッシング・モードでは、ビット 0-51 が目的のブロックを指定します。どのモードの場合も、アドレスのビット 52-63 は無視されます。

主記憶域ブロックを指すアドレスは実アドレスなので、動的アドレス変換は行われません。PAGE IN には、キ

ー制御記憶保護は適用されますが、低アドレス保護は適用されません。PAGE IN には、記憶域変更を記録するプログラム・イベント記録は適用されません。

操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

ページイン操作がエラーなしで完了した場合は、条件コード 0 が設定されます。

ページイン操作で拡張記憶域データ・エラーが検出された場合は、条件コード 1 が設定されます。拡張記憶域データ・エラー条件が存在する場合は、実記憶域内の 4K バイト・ブロック全体の内容は予測不能ですが、この条件が原因で、実記憶域内に無効な検査ブロック・コードが生成されることはありません。

目的の拡張記憶域ブロックが使用可能でない場合、つまり、そのブロックが用意されていないか、または現在構成内に存在しない場合は、条件コード 3 が設定され、その他のアクションは何も行われません。

複数 CPU 構成での PAGE IN の動作

チャンネル・プログラムまたは他の CPU から見た場合、PAGE IN による主記憶域および拡張記憶域へのアクセスは必ずしも単一アクセス参照ではなく、また、必ずしも左から右への方向に実行されません。

PAGE OUT の説明も参照してください。

結果の条件コード:

- 0 ページイン操作が完了しました。
- 1 拡張記憶域データ・エラー。
- 2 --
- 3 拡張記憶域ブロックが使用可能ではありません。

プログラム例外:

- アドレッシング例外 (汎用レジスタ R₁ に指定されているブロック)
- 操作例外 (拡張記憶域機能がインストールされていない場合)
- 特権命令例外
- 記憶保護例外 (汎用レジスタ R₁ に指定されているブロック、低アドレス保護)

PAGE OUT

PGOUT R₁,R₂ [RRE]

0	'B22F'	////////	R ₁	R ₂	31
			24	28	

ページアウト操作により、汎用レジスター R₁ に指定されている実記憶位置から、汎用レジスター R₂ に指定されている拡張記憶域ブロックへ、4K バイトのブロックが転送されます。

汎用レジスター R₂ のビット 32-63 は、拡張記憶域ブロック番号と呼ばれる 32 ビットの符号なし 2 進整数です。この番号は、拡張記憶域内の置き換えられる 4K バイト・ブロックを指定します。この拡張記憶域ブロック番号が拡張記憶域内のアクセス不能ブロックを指している場合は、条件コード 3 が設定されます。

汎用レジスター R₁ の内容は、主記憶域内の特定の 4K ブロックを指す実アドレスです。24 ビット・アドレッシング・モードでは、ビット 40-51 が目的のブロックを指定し、ビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、ビット 33-51 が目的のブロックを指定し、ビット 0-32 は無視されます。64 ビット・アドレッシング・モードでは、ビット 0-51 が目的のブロックを指定します。どのモードの場合も、アドレスのビット 52-63 は無視されます。

主記憶域ブロックを指すアドレスは実アドレスなので、動的アドレス変換は行われません。PAGE OUT は、キー制御保護の対象にはなりません。

操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

モデルによっては、拡張記憶域ブロックにデータが書き込まれた後で、データが正しく書き込まれたかどうかを確認するために、リードバック検査操作が行われることがあります。リードバック検査操作の結果、データが正しく書き込まれたことが確認された場合は、条件コード 0 が設定されます。リードバック検査操作で拡張記憶域データ・エラーが検出された場合は、条件コード 1 が設定されます。

ほとんどのモデルでは、リードバック検査操作は行われず、ページアウト操作が完了した時点で条件コード 0 が設定されます。

条件コード 0 または 1 のどちらが設定された場合も、4K バイト・ブロック全体が書き込まれています。ブロックの書き込み後に、そのブロック内にエラーがある場合、そのエラーはそのまま保持されます。したがって、後で同じ拡張記憶域ブロックを指定して PAGE IN を実行した場合は、拡張記憶域データ・エラーが検出され、条件コード 1 が示されることとなります。

目的の拡張記憶域ブロックが使用可能でない場合、つまり、そのブロックが用意されていないか、または現在構成内に存在しない場合は、条件コード 3 が設定され、その他のアクションは何も行われません。

複数 CPU 構成での PAGE OUT の動作

チャンネル・プログラムまたは他の CPU から見た場合、PAGE OUT による主記憶域および拡張記憶域へのアクセスは必ずしも単一アクセス参照ではなく、また、必ずしも左から右への方向に実行されるわけではありません。

複数の CPU がほとんど同時に PAGE IN または PAGE OUT 命令を発行した場合、モデルに応じて、2 つの操作が一度に 1 つずつ行われる場合と、同時に行われる場合があります。複数の命令が同じ拡張記憶域ブロックを指定している場合でも、同時に操作が行われることがあります。

同じ拡張記憶域ブロックをアドレッシングする複数の PAGE OUT 命令が同時に実行された場合は、転送されたバイトの各グループ単位で拡張記憶域ブロック内に生じる結果の値は、同時に実行されている命令のいずれかにより転送されたものとなります。1 グループとして転送されるバイトの数は、モデルによって異なります。

同様に、同じ拡張記憶域を対象として同時に PAGE IN 命令および PAGE OUT 命令を実行した場合は、PAGE IN 命令の実行結果として転送されるバイトの各グループの値は、その拡張記憶域ブロック内の古い値のこともあり、新しい値のこともあります。

複数のページング命令を同時に実行しても、拡張記憶域データ・エラーが起きることはありません。

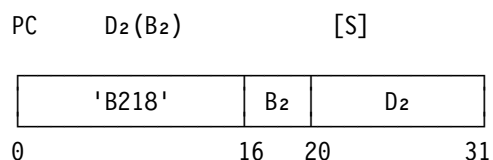
結果の条件コード:

- 0 ページアウト操作が完了しました。
- 1 拡張記憶域データ・エラー。
- 2 --
- 3 拡張記憶域ブロックが使用可能ではありません。

プログラム例外:

- アドレッシング例外 (汎用レジスター R₁ に指定されているブロック)
- 操作例外 (拡張記憶域機能がインストールされていない場合)
- 特権命令例外

PROGRAM CALL



第 2 オペランド・アドレスに指定されているプログラム呼び出し番号を使用して、特定のエンタリー・テーブル・エンタリー (ETE) を見つけるための 2 レベル検索が行われます。プログラムが ETE の使用を許可されるのは、制御レジスター 3 の中の PSW キー・マスクと、ETE 内の許可キー・マスクとの論理積が 0 以外であるか、または CPU が監視プログラム状態にあるときです。

PC タイプ・ビット (ETE のビット 128) が 0 のときは、基本 PROGRAM CALL と呼ばれる操作が行われます。PC タイプ・ビットが 1 のときは、スタッキング PROGRAM CALL と呼ばれる操作が行われます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、基本 PROGRAM CALL は、基本アドレッシング・モード・ビット (更新済み命令アドレスのビット 33-62) と、PSW の問題プログラム状態ビットを、汎用レジスター 14 のビット位置 32-63 にロードし、このレジスターのビット 0-31 は無変更のまま残します。64 ビット・アドレッシング・モードでは更新済み命令アドレスのビット 0-62、および問題プログラム状態ビットが、汎用レジスター 14 のビット位置 0-63 に入れられます。どのアドレッシング・モードでも、PSW キー・マスクと PASN が汎用レジスター 3 のビット位置 32-63 に入れられ、このレジスターのビット 0-31 は変更されません。

スタッキング PROGRAM CALL は、リンケージ・スタック・プログラム呼び出し状態エンタリーを作成して、PSW の内容全体 (ただし予測不能の PER マスクは除く) と、PSW キー・マスク、PASN、SASN、および EAX を、そのエンタリーに入れます。この状態エンタリーには、呼び出し先スペース ID、結果のアドレッシング・モードが 64 ビット・モードかどうかについての

指示、プログラム呼び出し番号、および、汎用レジスター 0-15 とアクセス・レジスター 0-15 の内容も入れられます。

基本 PROGRAM CALL では、拡張アドレッシング・モード・ビット (PSW のビット 31) の値は、エンタリー拡張アドレッシング・モード・ビット (ETE のビット 129) と同じでなければなりません。さもないと、特殊操作例外が認識されます。基本 PROGRAM CALL は、PSW のビット 31 を変更することなく、したがって、基本アドレッシング・モード (24 ビットまたは 31 ビット・モード) と拡張アドレッシング・モード (64 ビット・モード) の間の切り替えは行いません。24 ビットまたは 31 ビットのアドレッシング・モードでは、基本 PROGRAM CALL は、基本アドレッシング・モード・ビット (PSW のビット 32) を、エンタリー基本アドレッシング・モード・ビット (ETE のビット 32) の値に設定するので、アドレッシング・モードを 24 ビットと 31 ビットの間で切り替えることがあります。64 ビット・アドレッシング・モードでは、PSW のビット 32 は変更されません。

スタッキング PROGRAM CALL は、ETE のビット 129 が 0 のときは、PSW のビット 31 を 0 に設定し、PSW のビット 32 を ETE のビット 32 の値に設定します。ETE のビット 129 が 1 のときは、スタッキング PROGRAM CALL は、PSW のビット 31 および 32 を 1 に設定します。したがって、スタッキング PROGRAM CALL は、24 ビット、31 ビット、または 64 ビットのどのアドレッシング・モードでも設定できます。

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードである場合は、基本およびスタッキングのどちらの PROGRAM CALL も、ETE 内のエンタリー命令アドレスのビット 33-62 (つまり ETE のビット 33-62) の左側に 33 個、右側に 1 個の 0 を付加したものを、新しい命令アドレスとして PSW のビット位置 64-127 に入れ、そして、エンタリー問題プログラム状態ビット (ETE のビット 63) を、新しい問題プログラム状態ビットとして PSW のビット位置 15 に入れます。ETE 内のエンタリー・パラメーターのビット 32-63 は、汎用レジスター 4 のビット位置 32-63 に入れられ、このレジスターのビット 0-31 は変更されません。

結果のアドレッシング・モードが 64 ビット・モードである場合は、基本およびスタッキングのどちらの PROGRAM CALL も、エンタリー命令アドレスのビット 0-62 (ETE のビット 0-62) の右側に 1 個の 0 を付加

したものを、PSW のビット位置 64-127 に入れ、そして、ETE のビット 63 を PSW のビット位置 15 に入れます。ETE 内のエンタリー・パラメーターのビット 0-63 は、汎用レジスター 4 に入れられます。

基本 PROGRAM CALL は、ETE からのエンタリー・キー・マスクと制御レジスター 3 中の PSW キー・マスクの論理和を求め、その結果で PSW キー・マスクを置き換えます。スタッキング PROGRAM CALL は、これと同じことを行うか、または、PSW キー・マスクをエンタリー・キー・マスクで置き換えます。どちらを行うかは、ETE 内の PSW キー・マスク制御によって決まります。

スタッキング PROGRAM CALL は、必要に応じて、PSW 内の PSW キーおよび制御レジスター 8 中の EAX を ETE からの情報で置き換え、PSW 内のアドレス・スペース制御ビットを、ETE 内の制御ビットに従って設定します。

ETE に 0 以外の ASN が含まれている場合は、スペース切り替え操作が行われます。ETE に 0 の ASN が含まれている場合の操作を、現 1 次への PROGRAM CALL (PC-cp) と呼びます。ETE に 0 以外の ASN が含まれている場合の操作を、スペース切り替えを伴う PROGRAM CALL (PC-ss) と呼びます。スペース切り替えが指定されているときは、新しい PASN が ETE から制御レジスター 4 にロードされ、新しい 1 次 ASTE 起点が (PASTE0) が同じく ETE から制御レジスター 5 にロードされます。そして、PASTE からは、新しい 1 次 ASCE (PASCE) および AX が、それぞれ制御レジスター 1 および 4 にロードされます。

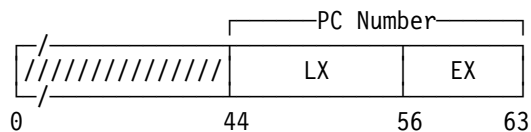
PC-cp と PC-ss のどちらの場合も、SASN および 2 次 ASCE (SASCE) は、それぞれ、元の PASN および PASCE と同じに設定されます。ただし、スペース切り替えを伴うスタッキング PROGRAM CALL 操作では、ETE 内の制御ビットの値によっては、SASN および SASCE はそれぞれ新しい PASN および PASCE と同じに設定されることがあります。

ディスパッチ可能単位がサブスペース・アクティブであるときに、ディスパッチ可能単位の基本スペースに対する PC-ss を実行した場合は、新しい PASCE のビット 0-55 および 58-63 は、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。これは、SASCE が PASCE と同じに設定される前に行われま

PROGRAM CALL の PC 番号変換

第 2 オペランド・アドレスは、データのアドレスを指定するために使用されるものではありません。代わりに、このアドレスの右端 20 ビットが PC 番号として使用されます。このアドレスのフォーマットは以下のとおりです。

第 2 オペランド・アドレス



リンケージ指標 (LX): 第 2 オペランド・アドレスのビット 44-55 はリンケージ指標であり、これは、1 次 ASTE 内のリンケージ・テーブル指定で指定されているリンケージ・テーブルからエンタリーを選択するために使用されます。

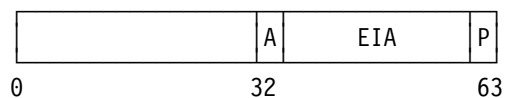
エンタリー指標 (EX): 第 2 オペランド・アドレスのビット 56-63 はエンタリー指標で、リンケージ・テーブル・エンタリーで指定されているエンタリー・テーブルからエンタリーを選択するために使用されます。

第 2 オペランド・アドレスのビット 0-43 は無視されます。

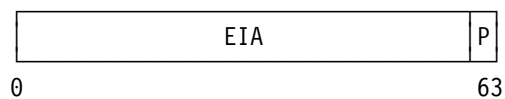
リンケージ・テーブルおよびエンタリー・テーブルの検索プロセスは、10-63 ページの図 10-22 のパート 1 に示してあります。このテーブル検索プロセスの詳細な定義については、5-28 ページの『PC 番号の変換』で説明されています。

32 バイトのエンタリー・テーブル・エンタリー (ETE) のフォーマットは、以下のとおりです。

結果のアドレッシング・モードが
24 ビット・モードまたは
31 ビット・モードのとき



結果のアドレッシング・モードが
64 ビット・モードのとき



AKM	ASN	EKM	
64	80	96	112 127

Cntrl/EK	EEAX	ASTE Adr	
128	144	160	186 191

Entry Parameter	
192	255

ETE のビット 128-143 の詳細なフォーマットは、以下のとおりです。

T	G	K	M	E	C	S	EK	
128	131			136			143	

24 ビットまたは 31 ビットのアドレッシング・モードにおいて、ETE のビット 32 (A) が 0 (24 ビット・モードを指定) である場合の基本 PROGRAM CALL、および、ビット 32 (A) および 129 (G) が 0 (24 ビット・モードを指定) である場合のスタッキング PROGRAM CALL では、ビット 33-39 は 0 でなければなりません。さもないと、PC 変換指定例外が認識されます。

ETE が取り出された後で、現 PSW が問題プログラム状態を指定している場合は、制御レジスター 3 の中の現 PSW キー・マスクが、ETE 内の AKM フィールドに照らしてテストされ、プログラムがこのエントリーへのアクセスを許可されているかどうかを判別されます。AKM と PSW キー・マスクの論理積が求められ、その結果が 0 であれば、特権命令例外が認識されます。制御レジスター 3 の中の PSW キー・マスクは、変更されません。PROGRAM CALL が監視プログラム状態で実行されている場合は、AKM フィールドは無視されます。

AKM と PSW キー・マスクの論理積演算の結果が 0 ではない場合、または CPU が監視プログラム状態にある場合は、命令の実行は続行されます。

ETE のビット 128 (T) が 0 の場合は、基本 PROGRAM CALL 操作が指定されます。ETE のビット 128 が 1 の場合は、スタッキング PROGRAM CALL 操作が指定されます。

基本 PROGRAM CALL

基本 PROGRAM CALL が指定されているときは、以下の操作が行われます。

現 PSW のビット 31 (拡張アドレッシング・モード・ビット) は、ETE のビット 129 (G) に等しくなければなりません。さもないと、特殊操作例外が認識されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、PSW のビット 97-126 (更新済み命令アドレスのビット 33-62) が、汎用レジスター 14 のビット位置 33-62 に入れられ、PSW のビット 32 (基本アドレッシング・モード・ビット) がこのレジスターのビット位置 32 に入れられ、PSW のビット 15 (問題プログラム状態ビット) がこのレジスターのビット位置 63 に入れられます。このレジスターのビット 0-31 は変更されません。

64 ビット・アドレッシング・モードでは、PSW のビット 64-126 (更新済み命令アドレスのビット 0-62) が汎用レジスター 14 のビット位置 0-62 に入れられ、PSW のビット 15 (問題プログラム状態ビット) がこのレジスターのビット位置 63 に入れられます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、ETE のビット 32 (A)、およびビット 33-62 (EIA) の右側に 1 個の 0 を付加したものが、それぞれ、PSW のビット位置 32 および 97-127 (命令アドレスの基本アドレッシング・モードおよびビット 33-63) に入れられます。64 ビット・アドレッシング・モードでは、ETE のビット 0-62 の右側に 1 個の 0 を付加したものが、PSW のビット位置 64-127 (命令アドレス) に入れられ、PSW のビット 32 は変更されません。どのアドレッシング・モードでも、ETE のビット 63 (P) は PSW のビット位置 15 (問題プログラム状態ビット) に入れられます。

PSW キー・マスク (制御レジスター 3 のビット 32-47) が、汎用レジスター 3 のビット位置 32-47 に入れられ、現 PASN (制御レジスター 4 のビット 48-63) が、汎用レジスター 3 のビット位置 48-63 に入れられます。汎用レジスター 3 のビット 0-31 は変更されません。

ETE のビット 96-111 (EKM) と PSW キー・マスク (制御レジスター 3 のビット 32-47) の論理和が求められ、その結果が制御レジスター 3 の中の PSW キー・マスクと置き換わります。

24 ビットまたは 31 ビットのアドレッシング・モードでは、ETE のビット 224-255 (エントリー・パラメーターのビット 32-63) が、汎用レジスター 4 のビット位置 32-63 にロードされ、このレジスターのビット 0-31 は変更されません。64 ビット・アドレッシング・モードでは、ETE のビット 192-255 (エントリー・パラメーター) が、汎用レジスター 4 のビット位置 0-63 にロードされます。

スタッキング PROGRAM CALL

スタッキング PROGRAM CALL が指定されているときは、以下の操作が行われます。

スタッキング・プロセスでは、リンケージ・スタック・プログラム呼び出し状態エントリーが形成され、その状態エントリーに情報が入れます。ここに入られる情報は、現 PSW (予測不能の PER マスクも含む)、PSW キー・マスク、PASN、SASN、EAX、呼び出し先スペース ID、結果のアドレッシング・モードが 64 ビット・モードかどうかの指示、プログラム呼び出し番号、汎用レジスター 0-15 の内容、およびアクセス・レジスター 0-15 の内容です。これについては、5-72 ページの『スタッキング・プロセス』で説明されています。状態エントリー内のエントリー・タイプ・コードは、2 進数 0001101 です。

ETE のビット 129 (G) が 0 のときは、PSW のビット 31 (拡張アドレッシング・モード・ビット) が 0 に設定され、ETE のビット 32 (A) が PSW のビット位置 32 (基本アドレッシング・モード・ビット) に入れられます。(アドレッシング・モードは、ビット 32 が 0 の場合は 24 ビット・モードに設定され、ビット 32 が 1 の場合は 31 ビット・モードに設定されます。) ETE のビット 129 が 1 のときは、PSW のビット 31 および 32 は 1 に設定されます。(64 ビット・アドレッシング・モードが設定されます。)

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードである場合は、ETE のビット 33-62 (EIA) の右側に 33 個の 0 を付加したものが、PSW のビット位置 64-127 (命令アドレス) に入れられます。結果のアドレッシング・モードが 64 ビット・モードである場合は、ETE のビット 0-62 (EIA) の右側に 1 個の 0 を付加したものが、PSW のビット位置 64-127 に入れられます。

ETE の 63 (P) が、PSW のビット位置 15 (問題プログラム状態ビット) に入れられます。

ETE のビット 131 (K) が 0 のときは、PSW のビット 8-11 (PSW キー) は変更されません。ETE のビット 131 が 1 のときは、ETE のビット 136-139 (EK) が PSW 内の PSW キーと置き換わります。

ETE のビット 132 (M) が 0 のときは、ETE のビット 96-111 (EKM) と PSW キー・マスク (制御レジスター 3 のビット 32-47) の論理和が求められ、その結果が制御レジスター 3 中の PSW キー・マスクと置き換わります。ETE のビット 132 が 1 のときは、ETE のビット 96-111 が、制御レジスター 3 中の PSW キー・マスクと置き換わります。

ETE のビット 133 (E) が 0 のときは、EAX (制御レジスター 8 のビット 32-47) は変更されません。ETE のビット 133 が 1 のときは、ETE のビット 144-159 (EEAX) が、制御レジスター 8 中の EAX と置き換わります。

ETE のビット 134 (C) が 0 のときは、PSW のビット 16 および 17 (アドレス・スペース制御ビット) は、2 進数 00 (1 次スペース・モード) に設定されます。ETE のビット 134 が 1 のときは、PSW のアドレス・スペース制御ビットは、2 進数 01 (アクセス・レジスター・モード) に設定されます。

結果のアドレッシング・モードが 24 ビットまたは 31 ビット・モードである場合は、ETE のビット 224-255 (エントリー・パラメーターのビット 32-63) が、汎用レジスター 4 のビット位置 32-63 にロードされ、このレジスターのビット 0-31 は変更されません。結果のアドレッシング・モードが 64 ビット・モードである場合は、ETE のビット 192-255 (エントリー・パラメーター) が、汎用レジスター 4 のビット位置 0-63 にロードされます。

リンケージ・スタックへの参照には、キー制御保護は適用されませんが、低アドレス保護およびページ保護は適用されます。

現 1 次への PROGRAM CALL (PC-cp)

ETE のビット 80-95 (ASN) が 0 の場合は、現 1 次への PROGRAM CALL (PC-cp) が指定されています。この場合、『PROGRAM CALL の PC 番号変換』で説明した操作の後で「基本 PROGRAM CALL」または「スタッキング PROGRAM CALL」が実行され、さらに以下の操作が終わった時点で、命令の実行は完了します。

現 PASN (制御レジスター 4 のビット 48-63) が、制御レジスター 3 のビット位置 48-63 に入れられて、現 SASN になります。

制御レジスター 1 の中の現 PASCE が制御レジスター 7 に入れられて、現 SASCE になります。

基本 PC-cp の動作は、10-63ページの図10-22 のパート 1 から 3 に示してあります。スタッキング PC-cp の動作は、同じ図のパート 1、4、および 5 に示してあります。

スペース切り替えを伴う PROGRAM CALL (PC-ss)

ETE 内の ASN が 0 でない場合は、スペース切り替えを伴う PROGRAM CALL (PC-ss) が指定されています。『PROGRAM CALL の PC 番号変換』で説明した操作の後で「基本 PROGRAM CALL」または「スタッキング PROGRAM CALL」が実行され、さらに以下の操作が終わった時点で、命令の実行は完了します。

ETE のビット 80-95 (ASN) が、制御レジスター 4 のビット位置 48-63 に新 PASN として入れられます。

ETE のビット 161-185 の右側に 6 個の 0 を付加したものが、新しい PASN で指定されている ASTE の実アドレスとして使用されます。ASTE のビット 0 が 1 の場合は、ASX 変換例外が認識されます。

ASTE のビット 64-127 (ASCE) が、制御レジスター 1 に新 PASCE として入れられます。

ASTE のビット 32-47 (AX) が、制御レジスター 4 のビット位置 32-47 に新しい許可指標として入れられます。

ASTE アドレスのビット 33-57 が、制御レジスター 5 のビット位置 33-57 に新しい 1 次 ASTE 起点として入れられ、ビット位置 32 および 58-63 にはすべて 0 が入れられます。このレジスターのビット 0-31 は変更されません。

基本 PROGRAM CALL、または、ETE のビット 135 (S) が 0 の場合のスタッキング PROGRAM CALL では、ETE からの情報で置き換えられる前の PASN が、制御レジスター 3 のビット位置 48-63 に入れられて現 SASN となり、ASTE からの情報で置き換えられる前の PASCE が、制御レジスター 7 に入れられて現 SASCE となります。(つまり、SASN および SASCE

は、それぞれ旧 PASN および旧 PASCE と同じに設定されます。)

ETE のビット 135 (S) が 1 の場合のスタッキング PROGRAM CALL では、SASN は、ETE からの情報で置き換えられた後の PASN で置き換えられ、SASCE は、ASTE からの情報で置き換えられた後の PASCE で置き換えられます。(つまり、SASN および SASCE は、それぞれ新 PASN および新 PASCE と同じに設定されます。)

このパラグラフで述べる内容は、サブスペース・グループ機能の使用に適用されます。新 PASCE が制御レジスター 1 に入れられ、新 1 次 ASTE 起点が制御レジスター 5 に入れられた後で、(1) PASCE 内のサブスペース・グループ制御ビット (ビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) 1 次 ASTE 起点がディスパッチ可能単位の基本スペース用の ASTE を指している場合は、PASCE のビット 0-55 および 58-63 が、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。この置き換えは、制御レジスター 7 の中の SASCE が PASCE で置き換えられる前に行われます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。

PC-ss の動作は、10-63ページの図10-22のパート 1 および 4-6 に示してあります。

PROGRAM CALL の逐次化

PC-cp と PC-ss のどちらの場合も、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

基本 PROGRAM CALL 操作が正常に実行されるのは、(1) 操作の開始時点で CPU が 1 次スペース・モードにあり、(2) サブシステム・リンケージ制御 (リンケージ・テーブル指定のビット 0) が 1 であり、かつ、(3) 拡張アドレッシング・モード・ビット (現 PSW のビット 31) がエンタリー拡張アドレッシング・モード・ビット (エンタリー・テーブル・エンタリーのビット 129) に等しい場合のみです。スタッキング PROGRAM CALL が正常に実行されるのは、操作の開始時点で CPU が 1 次スペース・モードまたはアクセス・レジスター・モードにあり、サブシステム・リンケージ制御が 1 である場合のみです。さらに、PC-ss を正常に実行で

きるのは、ASN 変換制御 (制御レジスター 14 のビット 44) が 1 のときだけです。これらの規則のいずれかに違反している場合は、特殊操作例外が認識されます。

スタッキング・プロセスでは、スタック一杯例外またはスタック指定例外が認識されることがあります。

PC-ss において、命令の実行前または実行後に、1 次スペース切り替えイベント制御ビット (制御レジスター 1 のビット 57) が 1 のときは、操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。また、PER イベントが報告された場合も、PC-ss 操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-61ページの図10-21 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (キー制御保護の場合を除き、リンケージ・スタック・エントリーの取り出しまたは格納)
- アドレッシング例外 (1 次 ASN 第 2 テーブル・エントリー内のリンケージ・テーブル指定、リンケージ・テーブル・エントリー、エントリー・テーブル・エントリー、ASN 第 2 テーブル・エントリー。PC-ss のみ)
- ASX 変換例外 (PC-ss のみ)
- EX 変換例外
- LX 変換例外
- PC 変換指定例外
- 特権命令例外 (問題プログラム状態で、AKM と PSW キー・マスクの論理積が 0)
- スペース切り替えイベント例外 (PC のみ)
- 特殊操作例外
- スタック一杯例外 (スタッキング PC のみ)
- スタック指定例外 (スタッキング PC のみ)
- サブスペース置換例外 (PC-ss のみ)
- トレース例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 ハーフワードについてのアクセス例外。
- 7.B DAT がオフのため、または CPU が 2 次スペース・モードまたはホーム・スペース・モードであるために起きる特殊操作例外。
- 8.A トレース例外。
- 8.B.1 1 次 ASN 第 2 テーブル・エントリー内のリンケージ・テーブル指定へのアクセスについてのアドレッシング例外。
- 8.B.2 リンケージ・テーブル指定内のサブシステム・リンケージ制御が 0 であるために起きる特殊操作例外。
- 8.B.3 リンケージ・テーブル・エントリーがテーブルの外部にあるために起きる LX 変換例外。
- 8.B.4 リンケージ・テーブル・エントリーへのアクセスについてのアドレッシング例外。
- 8.B.5 リンケージ・テーブル・エントリーの I ビット (ビット 0) が 1 であるために起きる LX 変換例外。
- 8.B.6 エントリー・テーブル・エントリーがテーブルの外部にあるために起きる EX 変換例外。
- 8.B.7 エントリー・テーブル・エントリーへのアクセスについてのアドレッシング例外。
- 8.B.8 CPU がアクセス・レジスター・モードであるため、または、拡張アドレッシング・モード・ビット (PSW のビット 31) が、エントリー拡張アドレッシング・モード・ビット (エントリー・テーブル・エントリーのビット 129) に等しくないために起きる特殊操作例外 (基本 PC の場合のみ)。
- 8.B.9 エントリー・テーブル・エントリー内に、無効な組み合わせ (結果のアドレッシング・モードが 24 ビットのときにビット 33-39 が 0 以外) があるために起きる PC 変換指定例外。
- 8.B.10 問題プログラム状態で、PSW キー・マスクと AKM の論理積が 0 であるために起きる特権命令例外。
- 8.B.11 ASN 変換制御 (制御レジスター 14 のビット 44) が 0 であるために起きる特殊操作例外 (PC-ss の場合のみ)。
- 8.B.12 ASN 第 2 テーブル・エントリーへのアクセスについてのアドレッシング例外 (PC-ss の場合のみ)。
- 8.B.13 ASN 第 2 テーブル・エントリーの I ビット (ビット 0) が 1 であるために起きる ASX 変換例外 (PC-ss の場合のみ)。

図 10-21 (1/2). 実行の優先順位: PROGRAM CALL

注: この図では詳細を示しませんが、8.B.13 と 9 の間の優先順位を持つサブスペース置換例外が起きることがあります。

8.B.14 現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)(スタッキング PC の場合のみ)。

注: 例外 8.B.15 - 8.B.20 が起きることがあるのは、現リンケージ・スタック・セクション内に十分なフリー・スペースが残っていない場合のみです。

8.B.15 現リンケージ・スタック・エントリー内の残余フリー・スペース値が 8 の倍数でないために起きるスタック指定例外。

8.B.16 現セクションのトレーラー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。このエントリーはトレーラー・エントリーであるものと見なされ、エントリー・タイプ・フィールドは検査されません。(スタッキング PC の場合のみ)

8.B.17 トレーラー・エントリー内の正方向セクション妥当性ビットが 0 であるために起きるスタック一杯例外 (スタッキング PC の場合のみ)。

8.B.18 次セクションのヘッダー・エントリーのエントリー記述子についてのアクセス例外 (取り出し)(スタッキング PC の場合のみ)。このエントリーはヘッダー・エントリーであるものと見なされ、エントリー・タイプ・フィールドは検査されません。

8.B.19 次セクション内に十分なフリー・スペースが残っていないために起きるスタック指定例外 (スタッキング PC の場合のみ)。

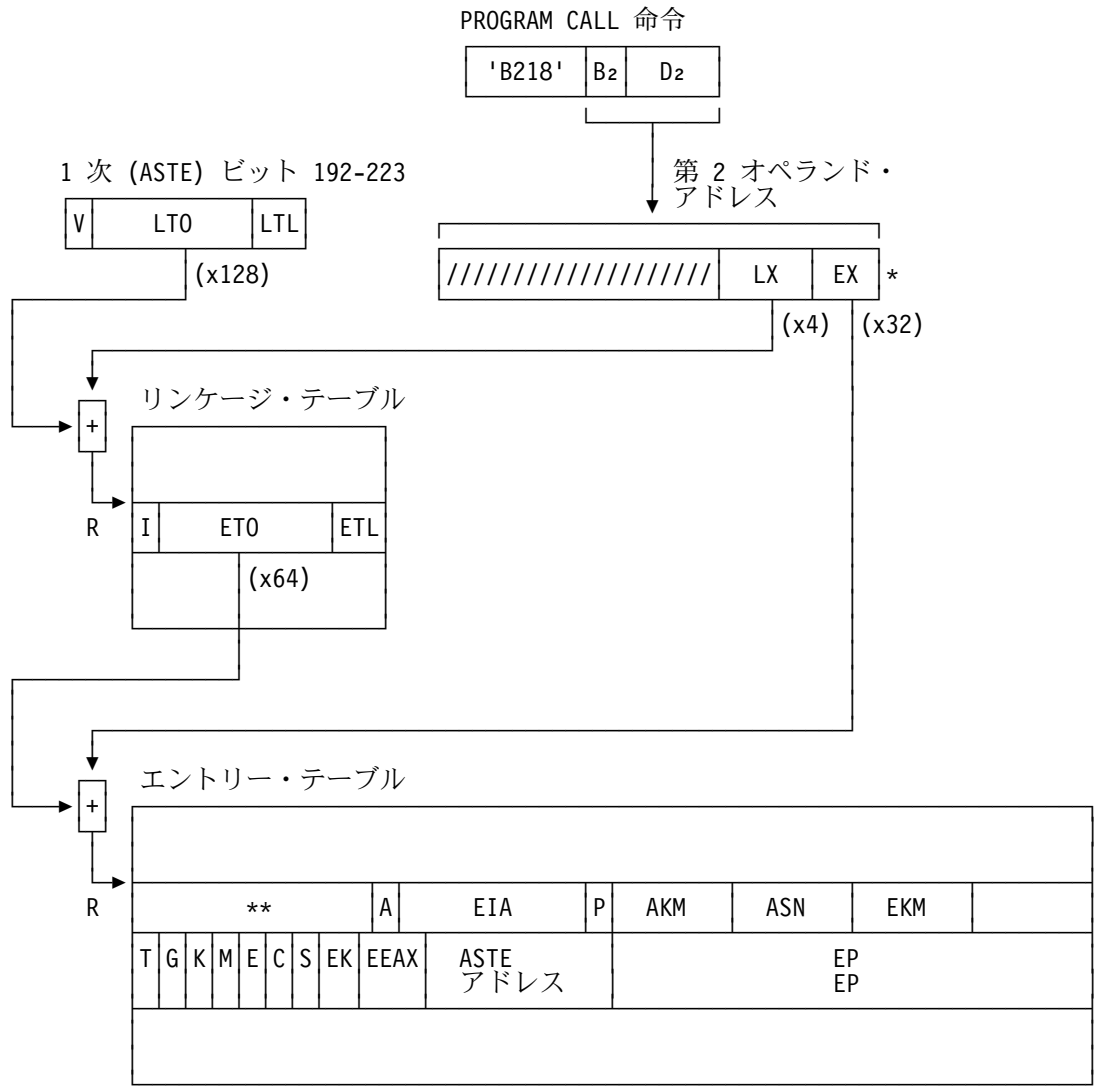
8.B.20 次セクションのヘッダー・エントリーの第 2 ワードについてのアクセス例外 (格納)。例外が何も存在しない場合は、このヘッダーが現エントリーと呼ばれるようになります。

8.B.21 現エントリーのエントリー記述子について、および新状態エントリーについてのアクセス例外 (格納)(スタッキング PC の場合のみ)。

9. スペース切り替えイベント (PC-ss の場合のみ)。

図 10-21 (2/2). 実行の優先順位: PROGRAM CALL

PC 番号変換



- R: アドレスは実アドレスです。
- *: スタッキング PC では、PC 番号がリンケージ・スタック内に入れます。
- ** : 結果のアドレッシング・モードが 64 ビット・モードである場合は、ETE の第 1 ワードおよび A は EIA のビット 0-32 です。

図 10-22 (1/6). PROGRAM CALL の実行

24 ビットまたは 31 ビットのアドレッシング・モードでの基本 PC-cp および PC-ss

エントリー・テーブル・エントリー

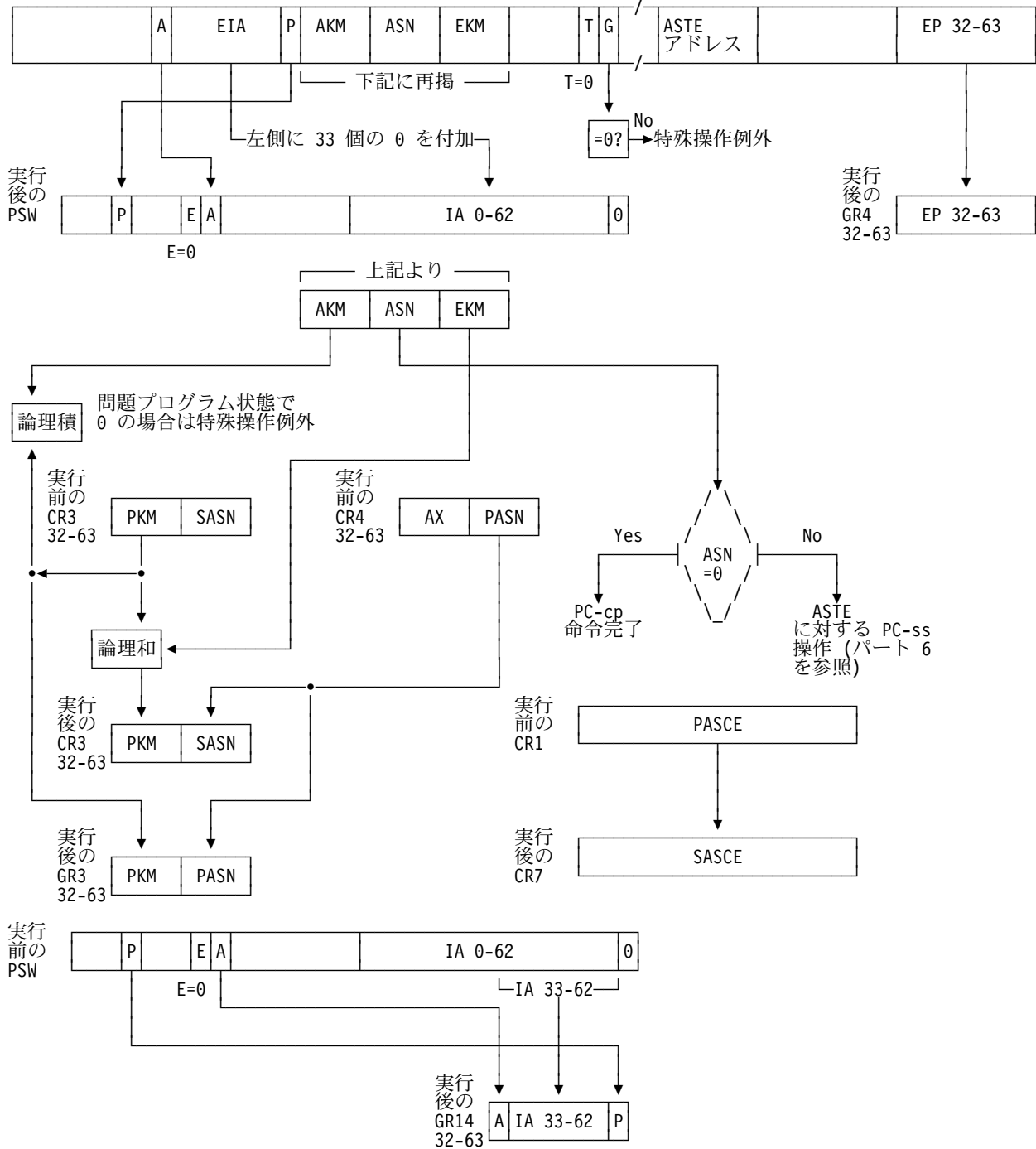


図 10-22 (2/6). PROGRAM CALL の実行

64 ビット・アドレッシング・モードでの基本 PC-cp および PC-ss

エンタリー・テーブル・エンタリー

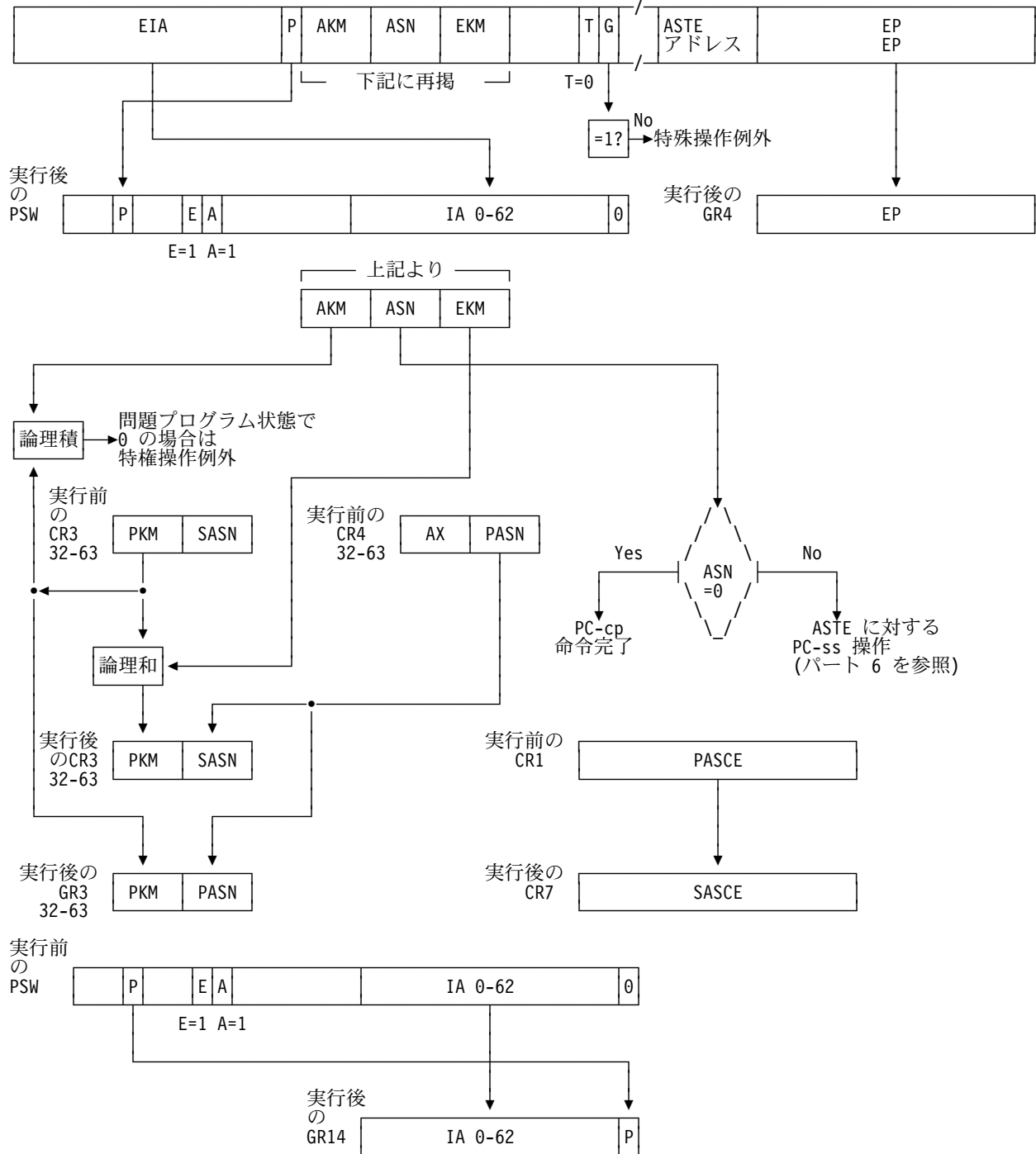
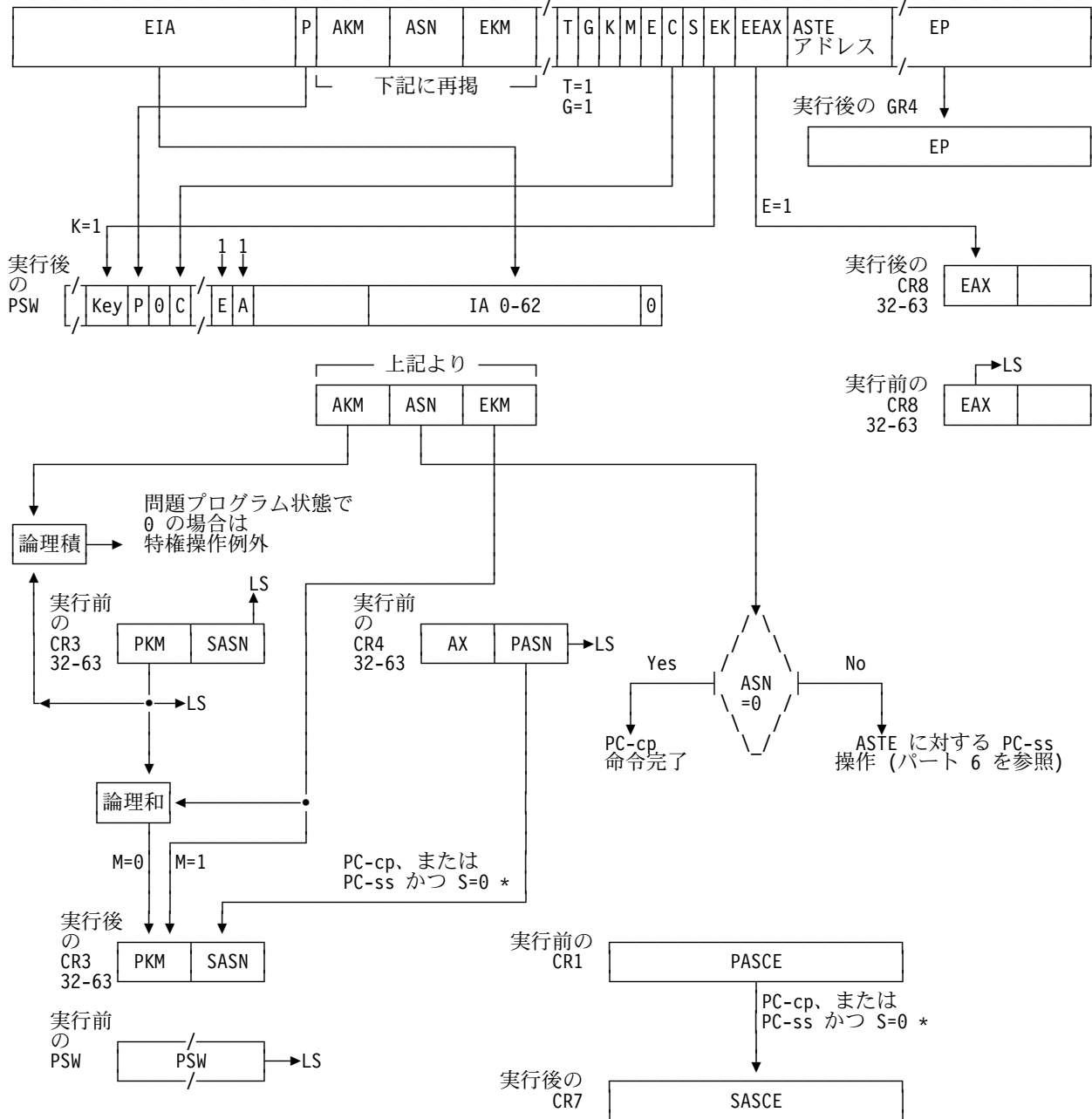


図 10-22 (3/6). PROGRAM CALL の実行

24 ビットまたは 31 ビットのアドレッシング・モードから 64 ビット・モードへのスタッキング PC-cp および PC-ss
 エントリー・テーブル・エントリー

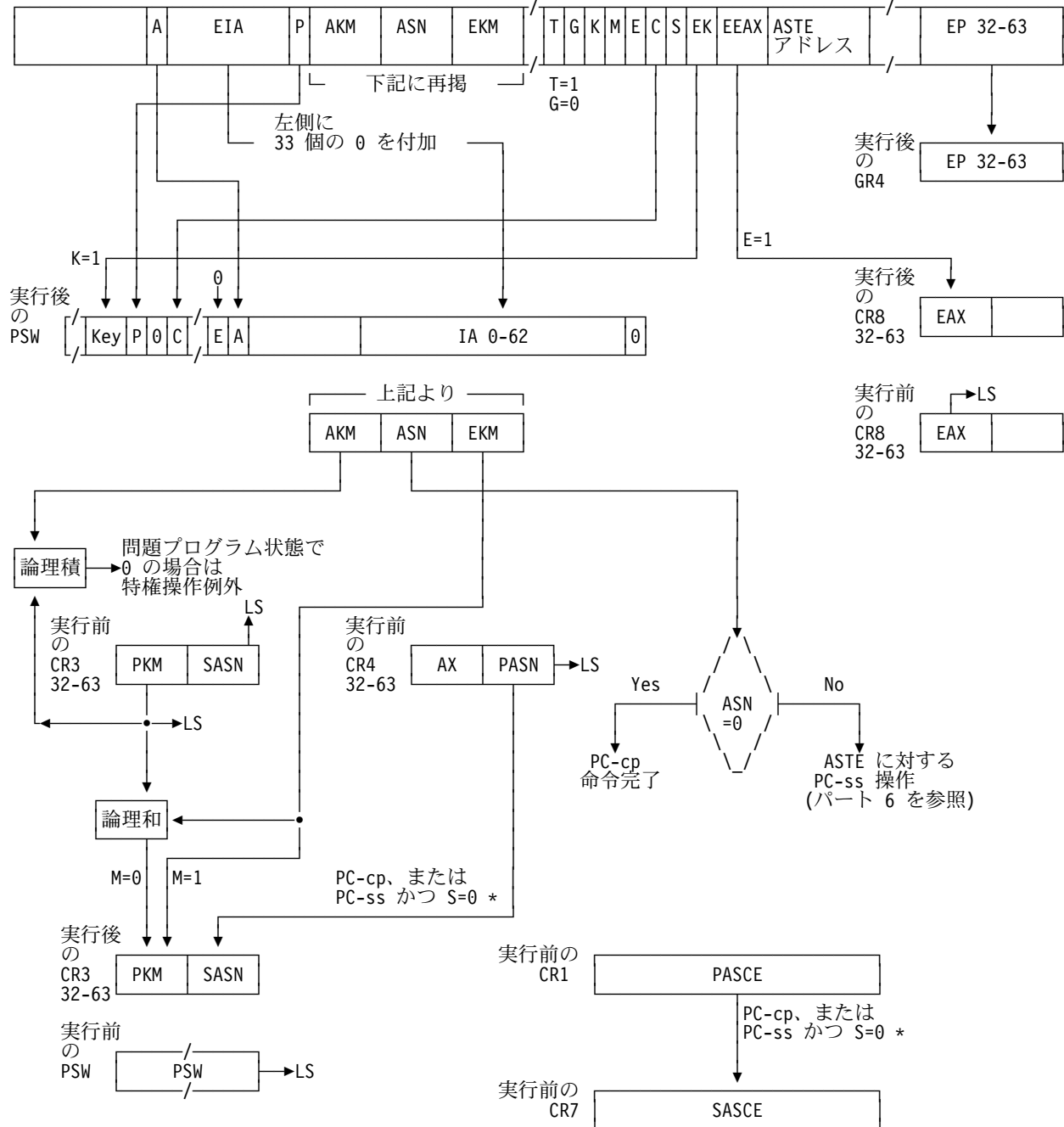


*: PC-ss で S=1 の場合は、SASN は新 PASN で置き換えられ、SASCE は新 PASCE で置き換えられます。

図 10-22 (4/6). PROGRAM CALL の実行

64 ビットから 24 ビットまたは 31 ビットのアドレッシング・モードへのスタッキング PC-cp および PC-ss

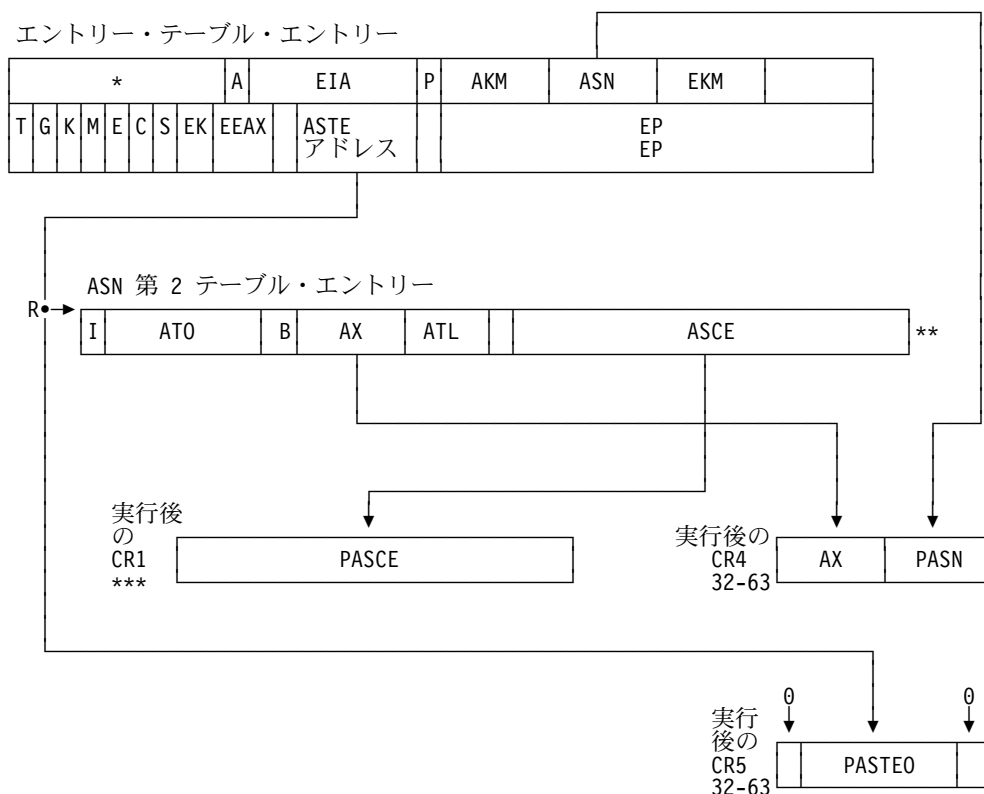
エントリー・テーブル・エントリー



*: PC-ss で S=1 の場合は、SASN は新 PASN で置き換えられ、SASCE は新 PASCE で置き換えられます。

図 10-22 (5/6). PROGRAM CALL の実行

PC-ss の場合の ASN 第 2 テーブル・エントリーに対する操作



- R: アドレスは実アドレスです。
 *: 結果のアドレッシング・モードが 64 ビット・モードである場合は、ETE の第 1 ワードおよび A は EIA のビット 0-32 です。
 **: ASTE は 64 バイトですが、最後の 48 バイトは示されていません。
 ***: PASCE のビット 0-55 および 58-63 は、サブスペース ASCE からの情報で置き換えられることがあります。

図 10-22 (6/6). PROGRAM CALL の実行

PROGRAM RETURN

PR [E]

'0101'
0 15

最新のリンケージ・スタック状態エントリーに保管されている PSW (PER マスク・ビットを除く) が、現 PSW として復元されます。現 PSW 中の PER マスクは、無変更のまま残されます。さらに、この状態エントリーから、汎用レジスター 2-14 およびアクセス・レジスター 2-14 の内容も復元されます。状態エントリーのエントリー記述子内のエントリー・タイプ・コードが 2 進数 0001101 (プログラム呼び出し状態エントリーを表す) である場合は、制御レジスター内の 1 次 ASN (PASN)、2 次 ASN (SASN)、PSW キー・マスク (PKM)、および拡張許可指標 (EAX) も、この状態エン

トリーから復元されます。エントリー・タイプ・コードが 2 進数 0001100 (ブランチ状態エントリーを表す) である場合は、現在の PASN、SASN、PKM、および EAX は変更されません。

最新の状態エントリーがどのように見つけ出され、その中の情報がどのように復元されるかについては、5-75ページの『アンスタッキング・プロセス』で説明されています。この状態エントリーはリンケージ・スタックから論理的に削除され、制御レジスター 15 中のリンケージ・スタック・エントリー・アドレスが、その直前の状態エントリーまたはヘッダー・エントリーのアドレスで置き換えられます。これについても、『アンスタッキング・プロセス』で説明されています。

状態エントリーがプログラム呼び出しエントリーであり、そのエントリーに含まれている PASN が現 PASN に等しくない場合は、スペース切り替え操作が行われます。状態エントリーに、現 PASN に等しい PASN が含

まれている場合の操作は、現 1 次への PROGRAM RETURN (PR-cp) と呼ばれます。状態エントリーに含まれている PASN が現 PASN に等しくない場合の操作は、スペース切り替えを伴う PROGRAM RETURN (PR-ss) と呼ばれます。PR-ss では、PASN 変換が行われます。SASN 変換および許可は、PR-cp または PR-ss のどちらでも行われることがあります。状態エントリーがブランチ状態エントリーである場合は、PR-cp および PR-ss という用語は適用されません。

リンケージ・スタックへのアクセスには、キー制御保護は適用されませんが、低アドレス保護およびページ保護は適用されます。

『PASN 変換』、『SASN 変換』、『SASN 許可』、および『PROGRAM RETURN の逐次化』のセクションの内容が適用されるのは、アンスタックされる状態エントリーがプログラム呼び出し状態エントリーである場合のみです。状態エントリーがブランチ・エントリーである場合は、これらのセクションで説明する機能は実行されません。

PASN 変換

新 PASN が、制御レジスター 4 のビット位置 48-63 にある旧 PASN に等しい場合は、PASN 変換は行われず、制御レジスター内の許可指標 (AX)、PASN、PASCE、および 1 次 ASN 第 2 テーブル・エントリー (1 次 ASTE) 起点は変更されません。

新 PASN が旧 PASN に等しくない場合は、新 PASN が変換されて、64 バイトの ASTE が見つけ出されます。ASN テーブル検索プロセスについては、3-17ページの『ASN 変換』で説明されています。ASN 変換に関連した例外を、総称的に ASN 変換例外と呼びます。これらの例外とそれぞれの優先順位については、第 6 章、『割り込み』で説明されています。

ASTE のビット 64-127 が、制御レジスター 1 に新 PASCE として入れられます。

ASTE のビット 32-47 が、制御レジスター 4 のビット位置 32-47 に新しい AX として入れられます。

ASTE アドレスのビット 33-57 が、制御レジスター 5 のビット位置 33-57 に新しい 1 次 ASTE 起点として入れられ、ビット位置 32 および 58-63 にはすべて 0 が入れられます。このレジスターのビット 0-31 は変更されません。

このパラグラフで述べる内容は、PASN 変換が行われた後のサブスペース・グループ機能の使用に適用されません。(1) 新 PASCE のサブスペース・グループ制御ビット (ビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) 新しい 1 次 ASTE 起点がディスパッチ可能単位の基本サブスペースの ASTE を指している場合は、レジスター 1 の中の新 PASCE のビット 0-55 および 58-63 が、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。新 SASN が新 PASN に等しい場合、SASCE が PASCE と同じに設定される前に、この置き換えが行われます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。

SASN 変換

新 SASN が新 PASN に等しい場合は、制御レジスター 7 の中の SASCE が、制御レジスター 1 の中の新 PASCE と同じに設定されます。新 SASN が新 PASN に等しくない場合は、新 SASN が変換されて、64 バイトの ASTE が見つけ出されます。その ASTE のビット 64-127 が、制御レジスター 7 に新 SASCE として入れられます。

SASN 許可

新 SASN が新 PASN に等しくない場合は、新 SASN 用の ASTE から取得した権限テーブル起点 (ATO) が、第 3 テーブル検索の基礎として使用されます。新しい許可指標 (制御レジスター 4 のビット 32-47) は、権限テーブルの長さと比較して検査された後で、権限テーブル内のエントリーを見つけるための指標として使用されます。権限テーブル検索については、3-22ページの『ASN 許可』で説明されています。

このパラグラフで述べる内容は、SASN 変換および許可が行われた後のサブスペース・グループ機能の使用に適用されます。(1) 新 SASCE のサブスペース・グループ制御ビット (ビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) SASN 変換により得られた ASTE 起点がディスパッチ可能単位の基本スペースの ASTE を指している場合は、ビット 0-55 および 58-63 が、ディスパッチ可能単位が最後に制御権を持っていたサポート用の ASTE 内の ASCE の同じビットで置き換えられます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。

PROGRAM RETURN の逐次化

アンスタックされた状態エントリーがプログラム呼び出し状態エントリーである場合は、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

この命令が正常に実行されるのは、操作の開始時点で、CPU が 1 次スペース・モードまたはアクセス・レジスター・モードにある場合のみです。さらに、PASN または SASN のいずれの場合も、ASN 変換プロセスを実行できるのは、ASN 変換制御 (制御レジスター 14 のビット 44) が 1 の場合のみです。これらの規則のいずれかに違反している場合は、特殊操作例外が認識されます。

アンスタッキング・プロセスで、スタック空例外、スタック操作例外、スタック指定例外、またはスタック・タイプ例外が認識されることがあります。

PR-ss において、命令の実行前または実行後に、1 次スペース切り替えイベント制御 (制御レジスター 1 のビット 57) が 1 のときは、操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。また、PER イベントが報告された場合も、PR-ss 操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。

この命令によりロードする PSW について、ロードの前に妥当性検査は行われません。ただし、ロードの後で、PSW のビット 0、2-4、12、24-30、および 33-63 のいずれかが 1 の場合、ビット 31 が 0、ビット 32 が 1 で、かつビット 64-96 がすべて 0 ではない場合、ビット 31 と 32 が両方とも 0 で、ビット 64-103 がすべて 0 ではない場合、または ビット 31 が 1 でビット 32 が 0 である場合は、指定例外が認識され、プログラム割り込みが起きます。これらの場合は、操作は完了し、結果の命令長コードは 0 になります。この場合の指定例外は、この命令の中でプログラム例外としてリストされま

すが、これについては、6-9ページの『早期例外認識』で説明されています。この例外は、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

スペース切り替えイベントが示されているときに、前のパラグラフで述べた理由のいずれかにより、命令によりロードされた PSW が無効である場合は、結果の命令長コードが 0 または 1 (EXECUTE が実行された場合は 0 または 2) のいずれになるかは、予測不能です。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-71ページの図10-23 に、この命令に関するプログラム例外の認識の優先順位を示します。

結果の条件コード: コードは、ロードされた新しい PSW で指定されているとおりに設定されます。

プログラム例外:

- アクセス例外 (キー制御保護の場合を除き、リンケージ・スタック・エントリーの取り出しおよび格納)
- アドレッシング例外 (SASN 変換が行われた場合、権限テーブル・エントリー)
- ASN 変換例外 (PASN または SASN 変換が行われた場合)
- 2 次権限例外 (SASN 変換が行われた場合)
- スペース切り替えイベント例外
- 特殊操作例外
- 指定例外
- スタック空例外
- スタック操作例外
- スタック指定例外
- スタック・タイプ例外
- サブスペース置換例外 (PASN または SASN 変換が行われた場合)
- トレース例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7. DAT がオフのため、または CPU が 2 次スペース・モードまたはホーム・スペース・モードであるために起きる特殊操作例外。
- 8.A トレース例外。
- 8.B.1 現リンケージ・スタック・エントリーのエントリー記述子についてのアクセス例外 (取り出し)。
- 8.B.2 現エントリーが状態エントリーまたはヘッダー・エントリーでないために起きるスタック・タイプ例外。
注: 例外 8.B.3 - 8.B.7 が起きることがあるのは、
現エントリーがヘッダー・エントリーである場合のみです。
- 8.B.3 ヘッダー・エントリー内のアンスタック抑止ビットが 1 であるために起きるスタック操作例外。
- 8.B.4 ヘッダー・エントリーの第 2 ワードについてのアクセス例外 (取り出し)。
- 8.B.5 ヘッダー・エントリー内の逆方向スタック・エントリー妥当性ビットが 0 であるために起きるスタック空例外。
- 8.B.6 前のエントリーのエントリー記述子についてのアクセス例外 (取り出し)。前のエントリーとは、現 (ヘッダー) エントリー内の逆方向スタック・エントリー・アドレスが指しているエントリーです。
- 8.B.7 前のエントリーがヘッダー・エントリーであるために起きるスタック指定例外。
- 8.B.8 前のエントリーが状態エントリーではないために起きるスタック・タイプ例外。
- 8.B.9 状態エントリー内のアンスタック抑止ビットが 1 であるために起きるスタック操作例外。
- 8.B.10 状態エントリーについてのアクセス例外 (取り出し)、および、状態エントリーの前のエントリーのエントリー記述子についてのアクセス例外 (格納)。

図 10-23 (1/2). 実行の優先順位: PROGRAM RETURN

注: 例外 8.B.11 - 8.B.15 およびイベント 9 が起きることがあるのは、状態エントリーがプログラム呼び出し状態エントリーの場合のみです。

8.B.11 ASN 変換制御 (制御レジスター 14 のビット 54) が 0 であるために起きる特殊操作例外 (PASN または SASN 変換が行われる場合)。

8.B.12 ASN 変換例外 (PASN または SASN 変換が行われる場合)。

注: この図では詳細を示しませんが、PASCE または SASCE の中のビットの置換について、8.B.12 と 9 の間の優先順位を持つサブスペース置換例外が起きることがあります。

8.B.13 権限テーブル・エントリーがテーブルの外部にあるために起きる 2 次権限例外 (SASN 変換が行われる場合)。

8.B.14 権限テーブル・エントリーへのアクセスについてのアドレッシング例外 (SASN 変換が行われる場合)。

8.B.15 権限テーブル・エントリー内の S ビットが 0 であるために起きる 2 次権限例外 (SASN 変換が行われる場合)。

9. スペース切り替えイベント (PR-ss の場合のみ)。

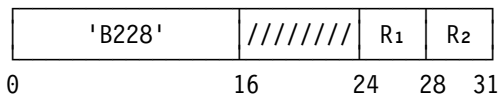
10. 即時割り込みを起こすタイプの PSW に起因する指定例外。

図 10-23 (2/2). 実行の優先順位: PROGRAM RETURN

プログラミング上の注意: 2 次スペース・モードまたはホーム・スペース・モードでは、PROGRAM CALL は正常に実行できないので、PROGRAM RETURN には、これらの変換モードのいずれかを指定する PSW をロードする機能はありません。SET ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL FAST と異なり、PROGRAM RETURN では、ホーム・スペース・モードを指定している PSW をロードすることが原因で、スペース切り替えイベントが認識されることはありません。

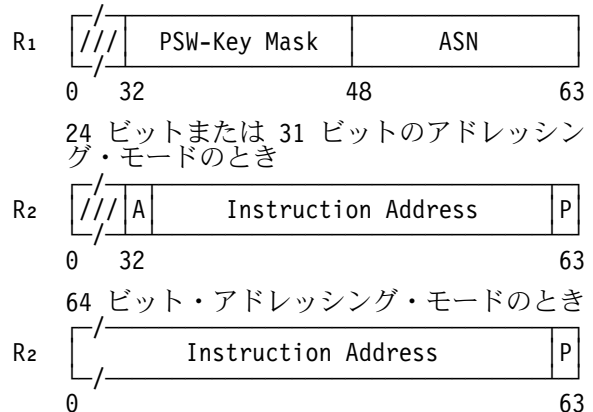
PROGRAM TRANSFER

PT R₁,R₂ [RRE]



汎用レジスター R₁ の内容が、PSW キー・マスク、PASN、および SASN の新しい値として使用されます。また、汎用レジスター R₂ の内容が、現 PSW 内の問題プログラム状態ビット、基本アドレッシング・モード・ビット、および命令アドレスの、新しい値として使用されます。

汎用レジスター R₁ および R₂ のフォーマットは、以下のとおりです。



汎用レジスター R₁ のビット位置 48-63 の内容が現 PASN に等しい場合は、この操作は現 1 次への PROGRAM TRANSFER (PT-cp) と呼ばれます。この 2 つのフィールドが等しくない場合は、この操作はスペース切り替えを伴う PROGRAM TRANSFER (PT-ss) と呼ばれます。

汎用レジスター R₂ の内容を使用して、現 PSW 内の問題プログラム状態ビットおよび命令アドレスが更新されます。また、24 ビットまたは 31 ビットのアドレッシング・モードでは、現 PSW 内の基本アドレッシング・モード・ビットも更新されます。汎用レジスター R₂ のビット 63 が、問題プログラム状態ビット (PSW のビット位置 15) に入られます。ただし、この操作の結果、PSW のビット 15 が 1 から 0 へ (問題プログラム状態から監視プログラム状態へ) 変更されてしまう場合は除

きます。このような変更が起きる場合は、特権命令例外が認識されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ R₂ のビット 32 が、基本アドレッシング・モード・ビット (現 PSW のビット 32) と置き換わり、このレジスタのビット 33-62 の右側に 1 個の 0 を付加したものが、PSW 内の命令アドレスのビット 33-63 (PSW のビット 97-127) と置き換わります。64 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 0-62 の右側に 1 個の 0 を付加したものが命令アドレスと置き換わり、基本アドレッシング・モード・ビットは変更されません。

R₁ のビット 32-47 と PSW キー・マスク (制御レジスタ 3 のビット 32-47) との論理積が求められ、その結果が PSW キー・マスクと置き換わります。汎用レジスタ R₁ のビット 0-31 は無視されます。

PT-ss および PT-cp のどちらの操作でも、汎用レジスタ R₁ のビット 48-63 が指す ASN が、制御レジスタ 3 の中の SASN と置き換わり、制御レジスタ 7 の中の SASCE が、制御レジスタ 1 の最終的な内容で置き換えられます。

現 1 次への PROGRAM TRANSFER (PT-cp)

現 1 次への PROGRAM TRANSFER (PT-cp) の操作は、10-76 ページの図 10-25 のパート 1 に示してあります。PT-cp 操作が完了するのは、上記で述べた PROGRAM TRANSFER 操作の共通部分が完了した時点です。PT-cp では、許可指標、PASN、1 次 ASCE、および、制御レジスタ 5 の内容 (1 次 ASN 第 2 テーブル・エン트리起点) は変更されません。

スペース切り替えを伴う PROGRAM TRANSFER (PT-ss)

汎用レジスタ R₁ のビット位置 48-63 にある ASN が現 PASN に等しくない場合は、スペース切り替えを伴う PROGRAM TRANSFER (PT-ss) が指定されており、2 段階のテーブル検索により ASN が変換されます。

PT-ss の動作は、10-76 ページの図 10-25 のパート 1 および 2 に示してあります。PT-ss 操作は以下に進みます。

PT-ss では、汎用レジスタ R₁ のビット位置 48-63 の内容が ASN として使用され、2 段階のテーブル検索によりその ASN が変換されます。

汎用レジスタ R₁ のビット 48-57 は 10 ビットの AFX で、これは ASN 第 1 テーブルからエントリーを選択するために使用されます。ビット 58-63 は 6 ビットの ASX で、これは ASN 第 2 テーブルからエントリーを選択するために使用されます。ASN テーブル検索プロセスについては、3-17 ページの『ASN 変換』で説明されています。ASN 変換に関連した例外を、総称的に「ASN 変換例外」と呼びます。これらの例外とそれぞれの優先順位については、第 6 章、『割り込み』で説明されています。

ASN 第 2 テーブル・エントリー (ASTE) から取得した権限テーブル起点が、第 3 テーブル検索の基礎として使用されます。現行の許可指標 (制御レジスタ 4 のビット 32-47) が、権限テーブルの長さと比較して検査された後で、権限テーブル内のエントリーを見つけるための指標として使用されます。権限テーブル検索については、3-22 ページの『ASN 許可』で説明されています。

制御レジスタ 1 の中の ASTE のビット 64-127 が、制御レジスタ 1 に新 PASCE として入れられ、制御レジスタ 7 に新 SASCE として入れられて、PT-ss 操作は完了します。ASTE のビット位置 32-47 が、制御レジスタ 4 のビット位置 32-47 にある許可指標と置き換わります。ASTE アドレスのビット 33-57 が、制御レジスタ 5 のビット位置 33-57 に新しい 1 次 ASTE 起点として入れられ、ビット位置 32 および 58-63 にはすべて 0 が入れられます。このレジスタのビット 0-31 は変更されません。ASN (制御レジスタ R₁ のビット 48-63) が、制御レジスタ 3 および 4 のビット位置 48-63 にある SASN および PASN と置き換わります。

このパラグラフで述べる内容は、サブスペース・グループ機能の使用に適用されます。新 PASCE が制御レジスタ 1 に入れられ、新 1 次 ASTE 起点が制御レジスタ 5 に入れられた後で、(1) PASCE 内のサブスペース・グループ制御ビット (ビット 54) が 1 であり、(2) ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) 1 次 ASTE 起点がディスパッチ可能単位の基本スペース用の ASTE を指している場合は、PASCE のビット 0-55 および 58-63 が、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。この置き換えは、制御レジスタ 7 の中の SASCE が PASCE で置き換えられる前に行われます。詳細につい

ては、5-58ページの『サブスペース置き換え操作』で説明されています。

PROGRAM TRANSFER の逐次化

PT-cp と PT-ss のどちらの場合も、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

この命令を実行できるのは、CPU が 1 次スペース・モードにあり、サブシステム・リンケージ制御 (リンケージ・テーブル指定のビット 0) が 1 である場合のみです。CPU が、実モード、2 次スペース・モード、アクセス・レジスター・モード、またはホーム・スペース・モードにある場合、またはサブシステム・リンケージ制御が 0 の場合は、特殊操作例外が認識されます。

汎用レジスター R₂ のビット 63 が、問題プログラム状態ビット (PSW のビット位置 15) に入れられます。ただし、この操作の結果、PSW のビット 15 が 1 から 0 へ (問題プログラム状態から監視プログラム状態へ) 変更されてしまう場合は除きます。このような変更が起きる場合は、特権命令例外が認識されます。

24 ビットまたは 31 ビットのアドレッシング・モードでは、この命令が完了するためには、汎用レジスター R₂ のビット 32-39 が、PSW のビット 32 および 97-103 の有効な組み合わせを指定していなければなりません。汎用レジスター R₂ のビット 32 が 0 で、ビット 33-39 がすべて 0 ではない場合は、指定例外が認識されます。

PT-ss 命令が指定されている場合は、上記の要件に加えて、ASN 変換制御 (制御レジスター 14 のビット 44)

が 1 でなければなりません。さもないと、特殊操作例外が認識されます。

PT-ss において、命令の実行前または実行後に、1 次スペース切り替えイベント制御ビット (レジスター 1 のビット 57) が 1 のときは、操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。また、PER イベントが報告された場合も、PT-ss 操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。

何らかのアドレッシング例外が起きたときは、操作は抑止されます。

10-75ページの図10-24 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (1 次 ASN 第 2 テーブル・エントリー内のリンケージ・テーブル指定; 権限テーブル・エントリー、PT-ss のみ)
- ASN 変換例外 (PT-ss のみ)
- 1 次権限例外 (PT-ss)
- 特権命令例外 (問題プログラム状態あるときに監視プログラム状態を設定しようとした場合)
- スペース切り替えイベント例外 (PT-ss のみ)
- 特殊操作例外
- 指定例外
- サブスペース置換例外 (PT-ss のみ)
- トレース例外

- | | |
|-------|--|
| 1.-6. | 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7.A | 命令の第 2 ハーフワードについてのアクセス例外。 |
| 7.B | DAT がオフのため、または CPU が 2 次スペース・モード、アクセス・レジスタ・モード、またはホーム・スペース・モードであるために起きる特殊操作例外。 |
| 8.A | トレース例外。 |
| 8.B.1 | 1 次 ASN 第 2 テーブル・エンタリー内のリンケージ・テーブル指定へのアクセスについてのアドレッシング例外。 |
| 8.B.2 | リンケージ・テーブル指定内のサブシステム・リンケージ制御が 0 であるために起きる特殊操作例外。 |
| 8.B.3 | 問題プログラム状態にあるときに監視プログラム状態を設定しようとしたために起きる特権命令例外。 |
| 8.B.4 | 24 ビットまたは 31 ビットのアドレッシング・モードで、汎用レジスタ R ₂ 内に無効な組み合わせ (ビット 32 が 0 で、ビット 33-39 がすべて 0 ではない) があるために起きる指定例外。 |
| 8.B.5 | ASN 変換制御 (制御レジスタ 14 のビット 44) が 0 であるために起きる特殊操作例外 (PT-ss の場合のみ)。 |
| 8.B.6 | ASN 変換例外 (PT-ss の場合のみ)。
注: この図では詳細を示しませんが、8.B.6 と 9 の間の優先順位を持つサブスペース置換例外が起きることがあります。 |
| 8.B.7 | 権限テーブル・エンタリーがテーブルの外部にあるために起きる 1 次権限例外 (PT-ss の場合のみ)。 |
| 8.B.8 | 権限テーブル・エンタリーへのアクセスについてのアドレッシング例外 (PT-ss の場合のみ)。 |
| 8.B.9 | 権限テーブル・エンタリーの P ビットが 0 であるために起きる 1 次権限例外 (PT-ss の場合のみ)。 |
| 9. | スペース切り替えイベント (PT-ss の場合のみ)。 |

図 10-24. 実行の優先順位: PROGRAM TRANSFER

プログラミング上の注意:

- PROGRAM TRANSFER (PT) の操作は、CPU を以前の PROGRAM CALL 操作で保管されている状態に復元するために使用できます。この復元を行うには、PT 3,14 を実行します。汎用レジスタ 3 および 14 はそれぞれの元の値に復元されませんが、PASN、PSW キー・マスク、問題プログラム状態ビット、および命令アドレスは復元され、許可指標、PASCE、および 1 次 ASN 第 2 テーブル・エンタリー起点は、復元された PASN と一貫性のある値にされます。24 ビットおよび 31 ビット・ア

ドレッシング・モードでは、基本アドレッシング・モード・ビットも復元されます。

- 適正な権限を持ち、共通域内に実行している場合は、PROGRAM TRANSFER を使用して、1 次アドレス・スペースを任意のスペースに変更できます。2 次アドレス・スペースも、新しい 1 次アドレス・スペースと同じに設定されます。
- RR フォーマットのブランチ命令とは異なり、PROGRAM TRANSFER の R₂ フィールドの値が 0 である場合、それは汎用レジスタ 0 を指し、したがってブランチが行われます。

PT-cp および PT-ss

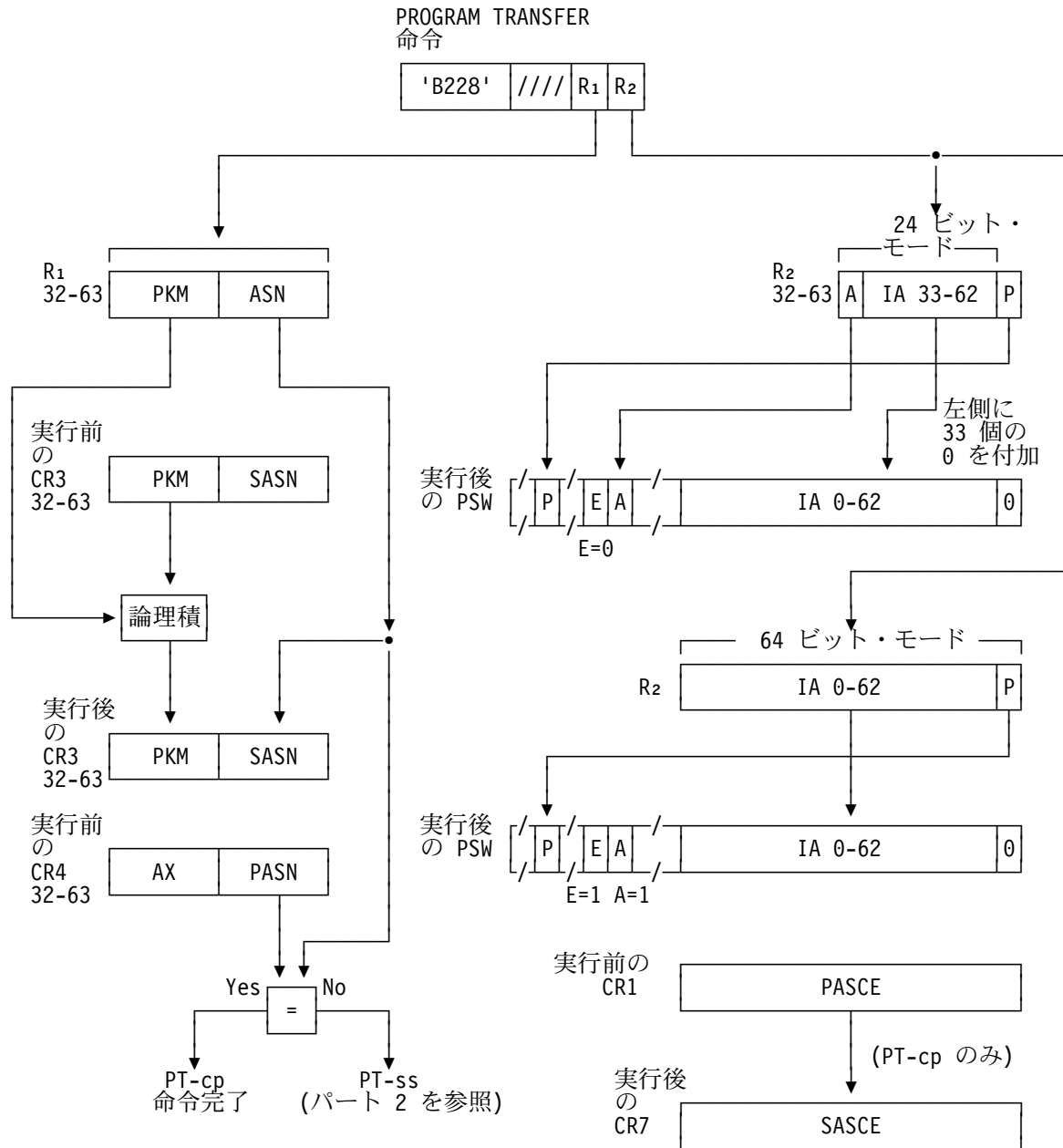
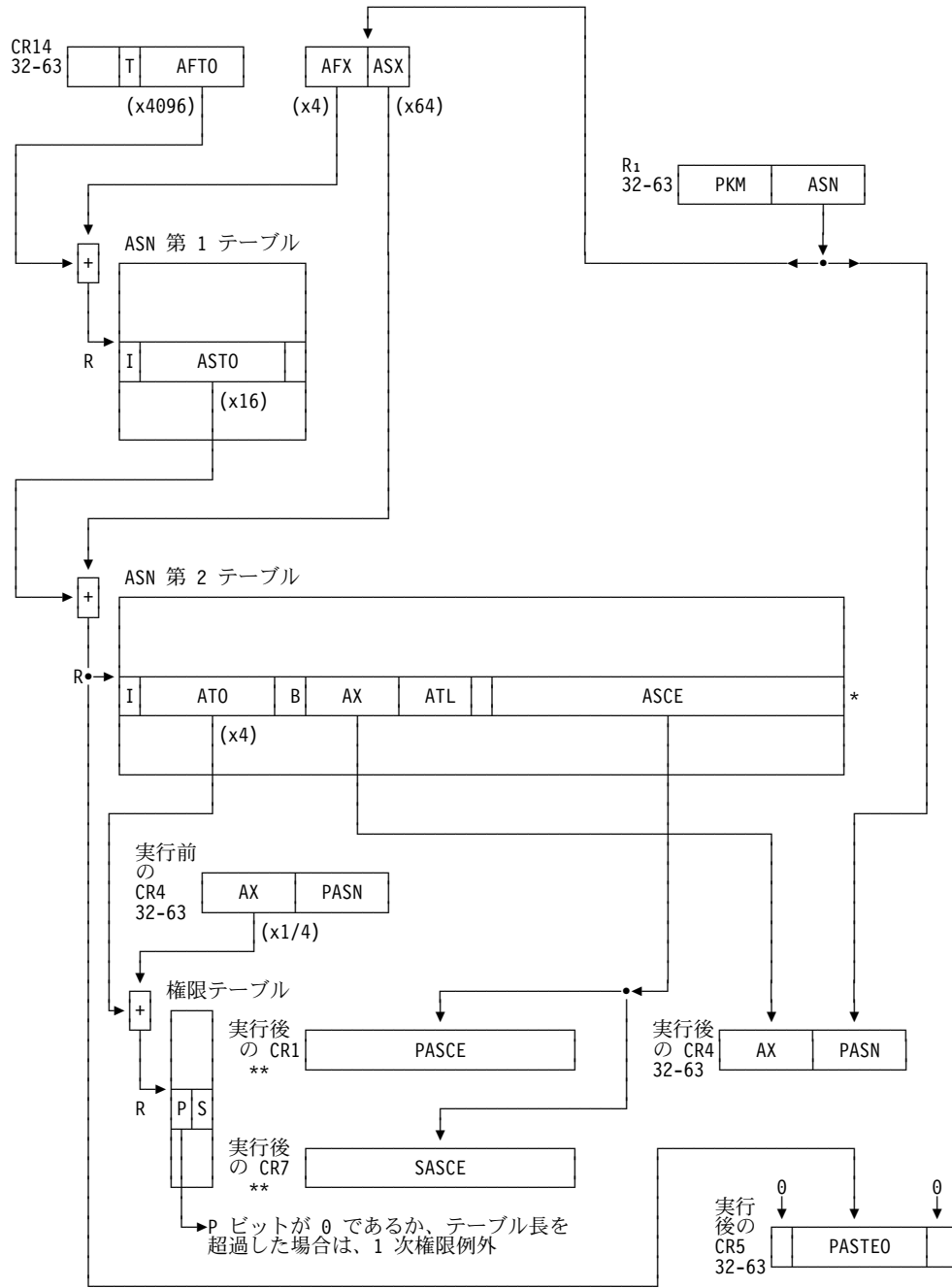


図 10-25 (1/2). PROGRAM TRANSFER の実行

PT-ss

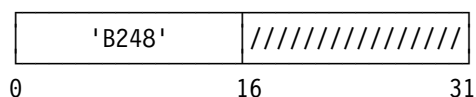


- R: アドレスは実アドレスです。
 *: ASTE は 64 バイトですが、最後の 48 バイトは示されていません。
 **: PASCE および SASCE のビット 0-55 および 58-63 は、サブスペース ASCE からの情報で置き換えられることがあります。

図 10-25 (2/2). PROGRAM TRANSFER の実行

PURGE ALB

PALB [RRE]



この CPU の ART 索引バッファ (ALB) 内のエントリが消去されます。アドレス可能な記憶域またはレジスタの内容は変更されません。

次の順番に当たっている命令の実行開始時点で、ALB の元の内容が消去されるように見えます。この操作は、他の CPU には通知されません。

逐次化機能が実行されます。

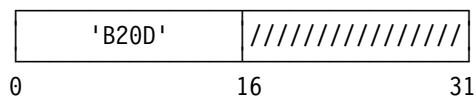
条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外

PURGE TLB

PTLB [S]



この CPU の変換索引バッファ (TLB) 内のエントリが消去されます。アドレス可能な記憶域またはレジスタの内容は変更されません。

次の順番に当たっている命令の取り出しの開始時点で、TLB の元の内容が消去されるように見えます。この操作は、他の CPU には通知されません。

逐次化機能が実行されます。

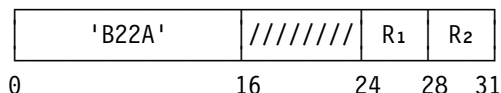
条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外

RESET REFERENCE BIT EXTENDED

RRBE R₁, R₂ [RRE]



汎用レジスタ R₂ の内容が示すアドレスにある 4K バイト・ブロックの記憶キー内の参照ビットが、0 に設定されます。汎用レジスタ R₁ の内容は無視されます。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 40-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-39 および 52-63 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 33-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 0-51 が、実記憶域内の 4K バイト・ブロックを指定し、ビット 52-63 は無視されます。

記憶域ブロックを指すアドレスは実アドレスなので、動的アドレス変換は行われません。また、記憶キーに対する参照には記憶保護例外は適用されません。

変更ビットも含めて、記憶キーのその他のビットは変更されません。

条件コードは、参照ビットが 0 に設定される前の参照ビットおよび変更ビットを表す値に設定されます。

結果の条件コード:

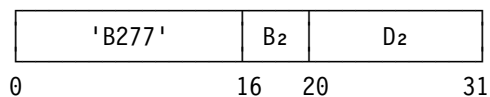
- 0 参照ビットは 0、変更ビットは 0 です。
- 1 参照ビットは 0、変更ビットは 1 です。
- 2 参照ビットは 1、変更ビットは 0 です。
- 3 参照ビットは 1、変更ビットは 1 です。

プログラム例外:

- アドレッシング例外 (汎用レジスタ R₂ に指定されているアドレス)
- 特権命令例外

RESUME PROGRAM

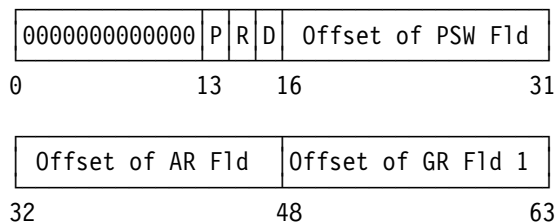
RP D₂(B₂) [S]



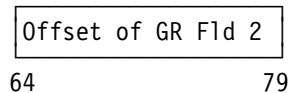
現 PSW とアクセス・レジスターおよび汎用レジスター B₂ の中の特定の内容が、第 2 オペランド内の対応する 3 つまたは 4 つのフィールドの内容で置き換えられます。命令アドレス・スペース内でこの命令の後に続くパラメーター・リストに、第 2 オペランド内の PSW フィールドのサイズ、第 2 オペランド内の汎用レジスター・フィールドのサイズまたは数、および第 2 オペランド内のフィールドのオフセットが指定されます。

命令アドレス・スペースは、命令が取り出されるアドレス・スペースです。DAT がオフの場合は、このアドレス・スペースは実アドレスから成っています。

パラメーター・リストの最初の 64 ビットのフォーマットは、以下のとおりです。



パラメーター・リストのビット 14 (R) および 15 (D) が両方とも 1 のときは、リストは、以下に示す追加の 16 ビットを持つ長さになります。



パラメーター・リストのビット 13 (P) は、第 2 オペランド内の PSW フィールドのサイズを指定します。このフィールドは、ビット 13 が 0 なら 8 バイトで、ビット 13 が 1 なら 16 バイトです。

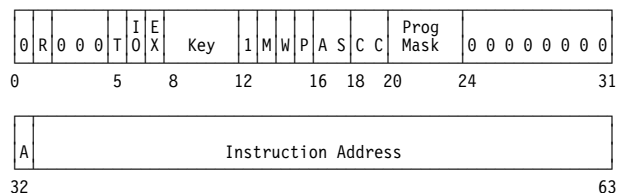
パラメーター・リストのビット 14 および 15 (R および D) は、第 2 オペランド内の 1 つまたは 2 つの汎用レジスター・フィールドを以下のように指定します。

- ビット 14 が 0 のときは、ビット 15 は無視され、第 2 オペランド内の汎用レジスター・フィールド 1 は 4 バイトであり、汎用レジスター B₂ のビット 32-63 がその 4 バイトで置き換えられます。第 2 オペランドには汎用レジスター・フィールド 2 はなく、汎用レジスター B₂ のビット 0-31 は変更されません。
- ビット 14 が 1 でビット 15 が 0 のときは、汎用レジスター・フィールド 1 は 8 バイトで、汎用レジスター B₂ のビット 0-63 がその 8 バイトで置き換えられます。汎用レジスター・フィールド 2 はありません。
- ビット 14 および 15 が両方とも 1 のときは、汎用レジスター・フィールド 1 および 2 は両方とも 4 バイトで、汎用レジスター B₂ のビット 32-63 はフィールド 1 で置き換えられ、ビット 0-31 はフィールド 2 で置き換えられます。(“D” の文字は、分離 (disjoint) を意味します。)

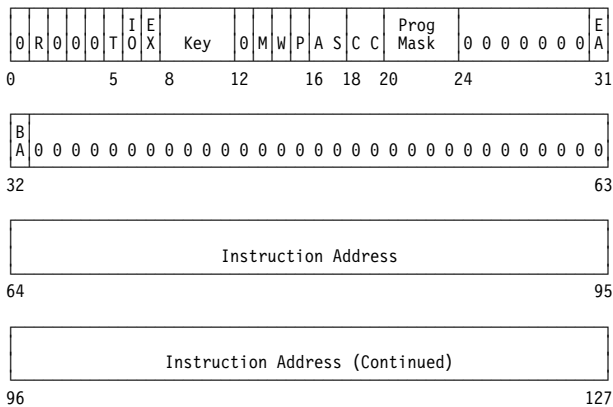
パラメーター・リストのビット 16-31 は符号なしの整数で、これは、第 2 オペランドの先頭から、ビット 13 の値に応じて 8 バイトまたは 16 バイトの PSW のフォーマットを持つフィールドまでのオフセットを表します。現 PSW 中のフィールドが、このフィールドで置き換えられます。同様に、ビット 32-47 は 4 バイト・フィールドへのオフセットで、アクセス・レジスター B₂ の内容がそのフィールドで置き換えられます。同様に、ビット 48-63 は、ビット 14 および 15 に応じて 4 バイトまたは 8 バイトの長さを持つフィールドへのオフセットで、汎用レジスター B₂ のビット 32-63 または 0-63 がそのフィールドで置き換えられます。パラメーター・リストのビット 64-79 が存在する場合、これらのビットも 4 バイト・フィールドへのオフセットを表し、汎用レジスター B₂ のビット 0-31 がそのフィールドで置き換えられます。

パラメーター・リストのビット 0-12 は、すべて 0 でなければなりません。さもないと、指定例外が認識されます。

8 バイトの第 2 オペランド PSW フィールドは、以下のような ESA/390 PSW フォーマットになります。



16 バイトの第 2 オペランド PSW フィールドは、以下のような z/Architecture PSW フォーマットになります。



現 PSW 中のフィールドが、第 2 オペランドの PSW フィールド中の対応するフィールドで置き換えられます。置き換えられる PSW フィールドは、以下のとおりです。

PSW ビット	フィールド名
16, 17 18, 19 20-23 31	アドレス・スペース制御 (AS) 条件コード (CC) プログラム・マスク 拡張アドレッシング・モード (EA)
32	基本アドレッシング・モード (BA)
64-127	命令アドレス

第 2 オペランドの PSW フィールドのその他のフィールドは、無視されます。特に、8 バイト PSW の場合にビット 12 が 1 で、16 バイト PSW の場合にビット 12 が 0 であるかどうかのテストは行われません。また、8 バイト PSW でビット 31 が 0 かどうかのテストも行われません。

PSW 内の未割り当てのフィールドは、将来割り当てられる可能性があります。RESUME PROGRAM により復元されることになる場合があります。したがって、第 2 オペランドの PSW フィールド中のこれらのフィールドには、0 が含まれていなければなりません。さもないと、将来、プログラムの動作の互換性が失われることがあります。

PSW のビット 64-127 が第 2 オペランド内の 8 バイトの PSW フィールドで置き換えられるときは、そのフィ

ールドのビット 33-63 の左側に 33 個の 0 を付加したもので置き換えられます。

第 2 オペランド内のフィールドは、アクセス・レジスタ B₂ および汎用レジスタ B₂ が変更される前に、取り出されます。

RESUME PROGRAM が EXECUTE 命令のターゲットである場合、パラメーター・リストは、EXECUTE 命令ではなく RESUME PROGRAM 命令の直後に置かれます。

パラメーター・リストに対する参照は、命令取り出しではなく記憶域オペランド取り出しです。

特別な条件

この命令が完了するためには、現 PSW に入れるビット 31、32、および 64-127 が、PSW に入れる値として有効なものでなければなりません。ビット 31 および 32 が両方とも 0 で、ビット 64-103 がすべて 0 ではない場合、ビット 31 が 0、ビット 32 が 1 で、ビット 64-96 がすべて 0 ではない場合、ビット 31 が 1 でビット 32 が 0 である場合、またはビット 127 が 1 である場合は、指定例外が認識されます。

この操作でホーム・スペース・モードが設定される場合は、CPU は監視プログラム状態でなければなりません。さもないと、特権命令例外が認識されます。DAT がオフのときは、第 2 オペランド内の PSW フィールドのビット 16 および 17 の値はテストされません。

操作の前または後のいずれか一方で (両方ではなく)、CPU がホーム・スペース・モードにあるときは、次のいずれかが真であれば、操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。(1) レジスタ 1 中の 1 次アドレス・スペース制御エレメント (ASCE) の 1 次スペース切り替えイベント制御 (ビット 57) が 1 であるか、(2) 制御レジスタ 13 中のホーム ASCE のホーム・スペース切り替えイベント制御 (ビット 57) が 1 であるか、または、(3) PER イベントが示される。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-81ページの図10-26に、この命令に関するプログラム例外の認識の優先順位を示します。

結果の条件コード: コードは、ロードされた新しい条件コードで指定されているとおりに設定されます。

プログラム例外:

- アクセス例外 (取り出し: パラメーター・リストとオペランド 2)
- 特権命令例外 (問題プログラム状態にあるときにホーム・スペース・モードを設定しようとした場合)
- スペース切り替えイベント例外
- 指定例外
- トレース例外

プログラミング上の注意:

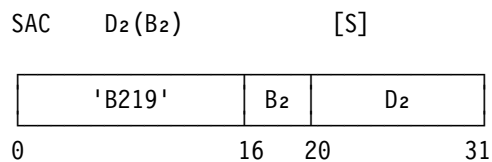
1. 5-81ページの『命令取り出し』で述べたように、命令に含まれるバイトは1つずつばらばらに取り出されることがあり、また、命令は1回の実行で複数回取り出されることがあります。したがって、チャンネル・プログラムまたは他のCPUが記憶域を変更し

ているときに、その記憶域から命令を取り出して実行した場合は、結果は予測不能です。この警告は、特に **RESUME PROGRAM** が **EXECUTE** のターゲットになっているときに重大な意味を持ちます。なぜなら、**RESUME PROGRAM** が使用するパラメーター・リストを、**B**、**X**、および **D** フィールドから生成するために、**EXECUTE** 命令が再取り出しされることがあるからです。**EXECUTE** が再取り出しされた場合、記憶域内にまだ **EXECUTE** 命令または **RESUME PROGRAM** が含まれているかどうかのテストが、必ず行われるとは限りません。

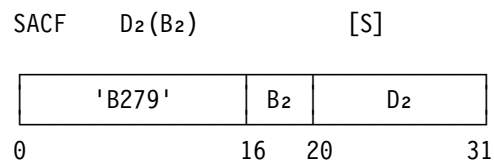
2. **RESUME PROGRAM** の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照。)

- | | |
|-------|---|
| 1.-6. | 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7. | 命令の第2ハーフワードについてのアクセス例外。 |
| 8.A | トレース例外。 |
| 8.B.1 | パラメーター・リストのビット 0-63 についてのアクセス例外。 |
| 8.B.2 | パラメーター・リストのビット 0-12 がすべて 0 ではないために起きる指定例外。 |
| 8.B.3 | パラメーター・リストのビット 64-79 が存在する場合に、これらのビットについてのアクセス例外。 |
| 8.B.4 | 第2オペランドについてのアクセス例外。 |
| 8.B.5 | 問題プログラム状態にあるときにホーム・スペース・モードを設定しようとしたために起きる特権命令例外。 |
| 8.B.6 | 第2オペランド内のPSWのビット位置 31、32、および 64-127 に無効な値があるために起きる指定例外。 |
| 9. | スペース切り替えイベント。 |

図 10-26. 実行の優先順位: **RESUME PROGRAM**



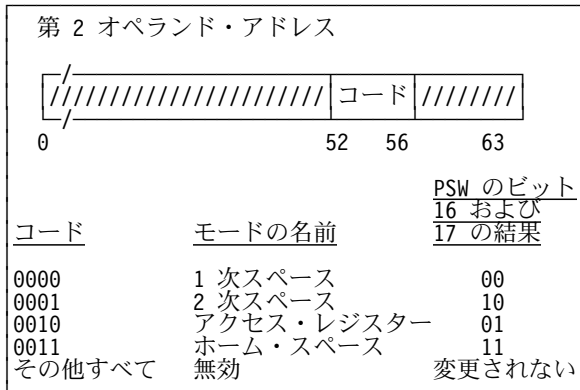
SET ADDRESS SPACE CONTROL FAST



第2オペランド・アドレスのビット 52-55 が、PSW 内のアドレス・スペース制御ビットを設定するためのコードとして使用されます。第2オペランド・アドレスは、データのアドレッシングには使用されません。代わり

に、ビット 52-55 がコードを形成します。第 2 オペランドのビット 0-51 および 56-63 は無視されます。第 2 オペランド・アドレスのビット 52 および 53 は 0 でなければなりません。さもないと、指定例外が認識されません。

次の図は、SET ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL FAST の操作を要約して示しています。



この操作がホーム・スペース・モードを設定する場合は、CPU は監視プログラム状態でなければなりません。さもないと、特権命令例外が認識されます。

SET ADDRESS SPACE CONTROL では、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。SET ADDRESS SPACE CONTROL FAST の場合は、この機能は実行されません。

特別な条件

SET ADDRESS SPACE CONTROL の操作が行われるのは、2 次スペース制御 (制御レジスター 0 のビット 37) が 1 で、DAT がオンの場合のみです。2 次スペース制御が 0 か、または DAT がオフのときは、特殊操作例外が認識されます。SET ADDRESS SPACE CONTROL FAST にもこれと同じ規則が適用されますが、2 次スペース制御がテストされるかどうかは予測不能です。

操作の前または後のいずれか一方で (両方ではなく)、CPU がホーム・スペース・モードにあるときは、次のいずれかが真であれば、操作の完了後にスペース切り替えイベント・プログラム割り込みが起きます。(1) レジスター 1 中の 1 次アドレス・スペース制御エレメント (ASCE) の 1 次スペース切り替えイベント制御 (ビット 57) が 1 であるか、(2) 制御レジスター 13 中のホーム ASCE のホーム・スペース切り替えイベント制

御 (ビット 57) が 1 であるか、または、(3) PER イベントが示される。

図10-27 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外 (問題プログラム状態で、ホーム・スペース・モードを設定しようとした場合)
- スペース切り替えイベント例外
- 特殊操作例外
- 指定例外

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
- 7.A 命令の第 2 ハーフワードについてのアクセス例外。
- 7.B DAT がオフであるために起きる特殊操作例外。
- 7.C 2 次スペース制御 (制御レジスター 0 のビット 37) が 0 であるために起きる特殊操作例外。
SET ADDRESS SPACE CONTROL FAST の場合は省略されることがあります。
8. 問題プログラム状態でホーム・スペース・モードを設定しようとしたために起きる特権命令例外。
9. 第 2 オペランド・アドレスのビット位置 52 および 53 の値が 0 でないために起きる指定例外。
10. スペース切り替えイベント。

図 10-27. 実行の優先順位: SET ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL FAST

プログラミング上の注意:

1. SET ADDRESS SPACE CONTROL および SET ADDRESS SPACE CONTROL FAST の定義によれば、設定すべきモードは、命令の変位フィールドに直接置くことも、また、INSERT ADDRESS SPACE CONTROL によりモードが保管される汎用レジスターの同じビット位置によって指定することもできます。
2. モデルによっては、SET ADDRESS SPACE CONTROL FAST の方が SET ADDRESS SPACE CONTROL より高いパフォーマンスを達成することがあります。

3. SET ADDRESS SPACE CONTROL FAST は、逐次化機能を実行しないので、事前取り出しされた命令のコピーが破棄されることはありません。SET ADDRESS SPACE CONTROL FAST を使用して、ホーム・スペース・モードに (またはホーム・スペース・モードから) 切り替えた後で、必ず予測可能な結果が得られるようにするには、1 次アドレス・スペースとホーム・アドレス・スペースの両方に同じ命令を含んでいない記憶位置でその命令を実行する前に、事前取り出しされた命令が破棄されるように、プログラムを設定する必要があります。事前取り出しされた命令を破棄するための操作については、5-81ページの『命令取り出し』で説明されています。
4. プログラムが、後続の SET ADDRESS SPACE CONTROL FAST 命令の後の記憶位置にある命令ストリームへの格納を行い、その SET ADDRESS SPACE CONTROL FAST 命令が、アクセス・レジスタ・モードまたはホーム・スペース・モードから (またはそのモードへ) 変換モードを変更した場合は、格納された値ではなく、事前取り出しされている命令が実行されることがあります。この状況を回避するためには、SET ADDRESS SPACE CONTROL FAST の代わりに SET ADDRESS SPACE CONTROL を使用するか、または、概念上の格納が行われた後で、事前取り出しされた命令を破棄させるための何らかの手段を用いる必要があります。

SET CLOCK

SCK D₂ (B₂) [S]

'B204'	B ₂	D ₂	
0	16	20	31

TOD クロックの現在値が、第 2 オペランド・アドレスに指定されているダブルワードの内容で置き換えられ、クロックは停止状態になります。

ダブルワード・オペランドは、クロックの分解能に応じて、クロックの内容と置き換わります。このオペランドのビットのうち、クロックにより更新されるビット位置に対応するビットのみが、クロック内で設定されます。オペランドの残りの右端ビット位置は無視され、クロック内には保存されません。モデルによっては、第 2 オペランドのビット 52 またはそれより右側のいずれかのビットを先頭とするすべての右端ビットが無視され、クロ

ック内の対応する位置はすべて 0 に設定されることがあります。また、クロックのビット位置 63 の右側の位置にも 0 が入れられます。

クロック値が設定されると、クロックは停止状態になります。クロックは、最後にクロックを停止状態にした CPU の TOD クロック同期 (制御レジスタ 0 のビット 34) の制御に従って、停止状態から離れて設定状態に入り、増加を再開します。このビットが 0 のときは、クロックはこの命令の完了時に設定状態になります。このビットが 1 のときは、このビットが 0 に設定されるか、または、他の CPU がクロックに影響を与える SET CLOCK を実行するまで、クロックは停止状態のままになります。外部時刻参照 (ETR) がインストールされている場合は、ETR からの信号を使用して、停止状態から設定状態に設定することができます。

クロックの値が変更され、クロックが停止状態に置かれるのは、構成内のいずれかの CPU の手動 TOD クロック制御が設定許可の位置に設定されているか、または、TOD クロック制御オーバーライド制御 (制御レジスタ 14 のビット 42) が 1 である場合のみです。すべての CPU の TOD クロック制御が保護位置に設定されており、TOD クロック制御オーバーライド制御が 0 のときは、クロックの値および状態は変更されません。クロックが設定されたか、無変更のままになっているかは、それぞれ、条件コードが 0 か 1 かで判別できます。

クロックが稼働していないときは、TOD クロック制御および TOD クロック制御オーバーライド制御の設定に関係なく、クロックの値および状態は変更されず、条件コード 3 が設定されます。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

結果の条件コード:

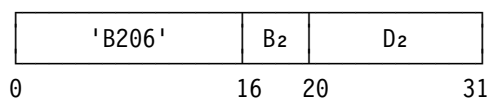
- 0 クロック値が設定された。
- 1 クロック値は保護状態にある。
- 2 --
- 3 クロックは非稼働状態にある。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外

SET CLOCK COMPARATOR

SCKC D₂(B₂) [S]



クロック・コンパレーターの現在値が、第 2 オペランド・アドレスに指定されているダブルワードの内容で置き換えられます。

オペランドのビットのうち、TOD クロックと比較するビット位置に対応するビットのみが、クロック・コンパレーター内に設定されます。オペランドの残りの右端ビット位置の内容は無視され、クロック・コンパレーター内には保存されません。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

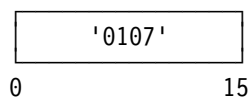
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外

SET CLOCK PROGRAMMABLE FIELD

SCKPF [E]



汎用レジスター 0 のビット 48-63 が、TOD プログラマブル・レジスターのビット位置 16-31 に入れられます。TOD プログラマブル・レジスターのビット位置 0-15 には、すべて 0 が入れられます。

特別な条件

汎用レジスター 0 のビット 32-47 はすべて 0 でなければなりません。さもないと、指定例外が認識されます。汎用レジスター 0 のビット 0-31 は無視されます。

条件コード: コードは変更されません。

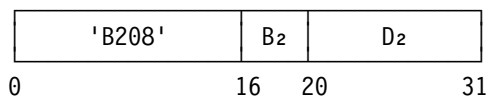
プログラム例外:

- 特権命令例外
- 指定例外

プログラミング上の注意: 同一構成内では、TOD プログラマブル・レジスター内の値はすべて同じでなければならない。また複数構成システム内では各構成ごとに固有の値でなければなりません。

SET CPU TIMER

SPT D₂(B₂) [S]



CPU タイマーの現在値が、第 2 オペランド・アドレスに指定されているダブルワードの内容で置き換えられます。

オペランドのビットのうち、更新するビット位置に対応するビットのみが CPU タイマー内で設定されます。オペランドの残りの右端ビット位置は無視され、CPU タイマー内には保存されません。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外

SET PREFIX

SPX D₂(B₂) [S]

'B210'	B ₂	D ₂	
0	16	20	31

プレフィックス・レジスタのビット位置 33-50 の内容が、第 2 オペランド・アドレスが指している位置にあるワードのビット位置 1-18 の内容で置き換えられます。この CPU の ART 索引バッファ (ALB) および変換索引バッファ (TLB) から、エントリが消去されます。

第 2 オペランドが取り出された後で、その値の妥当性が検査されてから、プレフィックス・レジスタの内容がその値で置き換えられます。オペランドのビット 1-18 に右側に 13 個、左側に 33 個の 0 を付加したものが、記憶域内の 8K バイトの新しいプレフィックス域の絶対アドレスとして使用されます。このアドレスは、現 PSW に指定されているアドレッシング・モードに関係なく、64 ビット・アドレスとして扱われます。新しいプレフィックス域内にある 2 つの 4K バイト・ブロックがアクセスされます。どちらかが構成内で使用可能でない場合は、アドレッシング例外が認識され、操作は抑止されます。これらのブロックへのアクセスは保護の対象にはなりませんが、アクセスの結果として、ブロックに対する参照ビットが 1 に設定されることがあります。

操作が完了すると、この命令の実行後のすべての割り込み、および後続の命令の実行に、新しいプレフィックスが使用されます。第 2 オペランドのビット位置 0 および 19-31 の内容は、無視されます。

ART 索引バッファ (ALB) および変換索引バッファ (TLB) から、エントリが消去されます。次の順番に当たっている命令の取り出し開始の時点から、ALB および TLB の元の内容が消去されているものと見なされず。

第 2 オペランドが取り出される前と後、さらに操作の完了後に、逐次化機能が実行されます。

特別な条件

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- アドレッシング例外 (新しいプレフィックス域)
- 特権命令例外
- 指定例外

SET PSW KEY FROM ADDRESS

SPKA D₂(B₂) [S]

'B20A'	B ₂	D ₂	
0	16	20	31

4 ビットの PSW キー (現 PSW のビット 8-11) が、第 2 オペランド・アドレスのビット 56-59 で置き換えられます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。代わりに、アドレスのビット 56-59 が新しい PSW キーを形成します。第 2 オペランド・アドレスのビット 0-55 および 60-63 は無視されます。

特別な条件

問題プログラム状態では、この命令の実行は、制御レジスタ 3 の中の PSW キー・マスクにより制御されません。PSW キー・マスク内の、設定すべき PSW キー値に対応するビットが 1 であれば、命令は正常に実行されます。PSW キー・マスク内の選択されたビットが 0 のときは、特権命令例外が認識されます。監視プログラム状態では、PSW キーの値は何であっても有効です。

条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外 (問題プログラム状態で、選択された PSW キー・マスク・ビットが 0)

プログラミング上の注意:

1. SET PSW KEY FROM ADDRESS のフォーマットは、プログラムが、B₂ フィールドに指定されている汎用レジスターからも、命令自体の中の D₂ フィールドからも、PSW キーを設定できるように定義されています。
2. あるプログラムが他のプログラムに対して、指定した記憶位置にアクセスすることを要求した場合、その呼び出し先プログラムの記憶位置が取り出し保護の対象になっていないものであれば、呼び出し先プログラムは、SET PSW KEY FROM ADDRESS を使用して、要求側のプログラムがこのアクセスを許可されているかどうかを検査することができます。呼び出し先プログラムは、このアクセスを行う前に、PSW キーを要求側のプログラムの PSW キーで置き換え、続いて呼び出し先プログラムの PSW キーを元の値に復元することにより、この検査を行うことができます。ただし、結果の保護例外が原因で操作が中止されることがあるので、このような保護例外の取り扱いには注意が必要です。呼び出し側プログラムから渡されるアドレスをテストするための別の方法については、TEST PROTECTION およびそれに関連した『プログラミング上の注意』を参照してください。

SET SECONDARY ASN

SSAR R₁ [RRE]

0	'B225'	////////	R ₁	////	31
		16	24	28	

汎用レジスター R₁ のビット位置 48-63 に指定されている ASN が、制御レジスター 3 の中の 2 次 ASN と置き換わり、その ASN に対応するアドレス・スペース制御エレメントが、制御レジスター 7 の中の SASCE と置き換わります。

汎用レジスター R₁ のビット位置 48-63 の内容が、新しい ASN と呼ばれます。このレジスターのビット位置 0-47 は無視されます。

最初に、新 ASN が現 PASN と比較されます。新 ASN が PASN に等しい場合の操作は、現 1 次への SET SECONDARY ASN (SSAR-cp) と呼ばれます。新 ASN が PASN に等しくない場合の操作は、スペース切り替えを伴う SET SECONDARY ASN (SSAR-ss) と呼ばれます。SSAR-cp および SSAR-ss 操作の手順は、10-88 ページの図 10-29 に示されています。

現 1 次への SET SECONDARY ASN (SSAR-cp)

新しい ASN が、SASN (制御レジスター 3 のビット 48-63) と置き換わり、制御レジスター 1 の中の PASCE が、制御レジスター 7 の中の SASCE と置き換わり、そして操作は完了します。

スペース切り替えを伴う SET SECONDARY ASN (SSAR-ss)

新 ASN が ASN 変換テーブルを使用して変換され、次に、現許可指標 (制御レジスター 4 のビット 32-47) を使用して、プログラムがその指定された ASN を使用することを許可されているかどうかテストされます。

新 ASN は、2 段階のテーブル検索により変換されません。新 ASN のビット 0-9 (レジスターの 48-57) は 10 ビットの AFX で、これは ASN 第 1 テーブルからエントリーを選択するために使用されます。新 ASN のビット 10-15 (レジスターのビット 58-63) は 6 ビットの ASX で、これは ASN 第 2 テーブルからエントリーを選択するために使用されます。この 2 段階検索については、3-17 ページの『ASN 変換』で説明されています。ASN 変換に関連した例外を、総称的に「ASN 変換例外」と呼びます。これらの例外とそれぞれの優先順位については、第 6 章、『割り込み』で説明されています。

第 2 の検索結果として得られる ASN 第 2 テーブル・エントリー (ASTE) には、アドレス・スペース制御エレメントと、ASN に関連した権限テーブルの起点と長さが含まれています。

ASTE から取得した権限テーブル起点が、第 3 テーブル検索の基礎として使用されます。現行の許可指標 (制御レジスター 4 のビット 32-47) が、権限テーブルの長さと比較して検査された後で、権限テーブル内のエントリーを見つけるための指標として使用されます。権限テーブル検索については、3-22 ページの『ASN 許可』で説明されています。

新 ASN (汎用レジスター R₁ のビット 48-63) が、SASN (制御レジスター 3 のビット 48-63) と置き換わります。ASTE 内のアドレス・スペース制御エレメントが、制御レジスター 7 の中の SASCE と置き換わります。

このパラグラフで述べる内容は、サブスペース・グループ機能の使用に適用されます。新 SASCE が制御レジスター 7 に入れられた後で、(1) SASCE 内のサブスペース・グループ制御ビット (ビット 54) が 1 であり、(2)

ディスパッチ可能単位がサブスペース・アクティブであり、かつ、(3) ASN 変換の結果得られた ASTE がディスパッチ可能単位の基本スペース用の ASTE である場合は、SASCE のビット 0-55 および 58-63 が、ディスパッチ可能単位が最後に制御権を持っていたサブスペース用の ASTE 内の ASCE の同じビットで置き換えられます。詳細については、5-58ページの『サブスペース置き換え操作』で説明されています。

SET SECONDARY ASN の逐次化

SSAR-cp and SSAR-ss のどちらの場合も、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

この操作が行われるのは、ASN 変換制御 (制御レジスター 14 のビット 44) が 1 で、かつ DAT がオンの場合

のみです。ASN 変換制御ビットが 0 か、または DAT がオフのときは、特殊操作例外が認識されます。

図10-28 に、この命令に関するプログラム例外の認識の優先順位を示します。

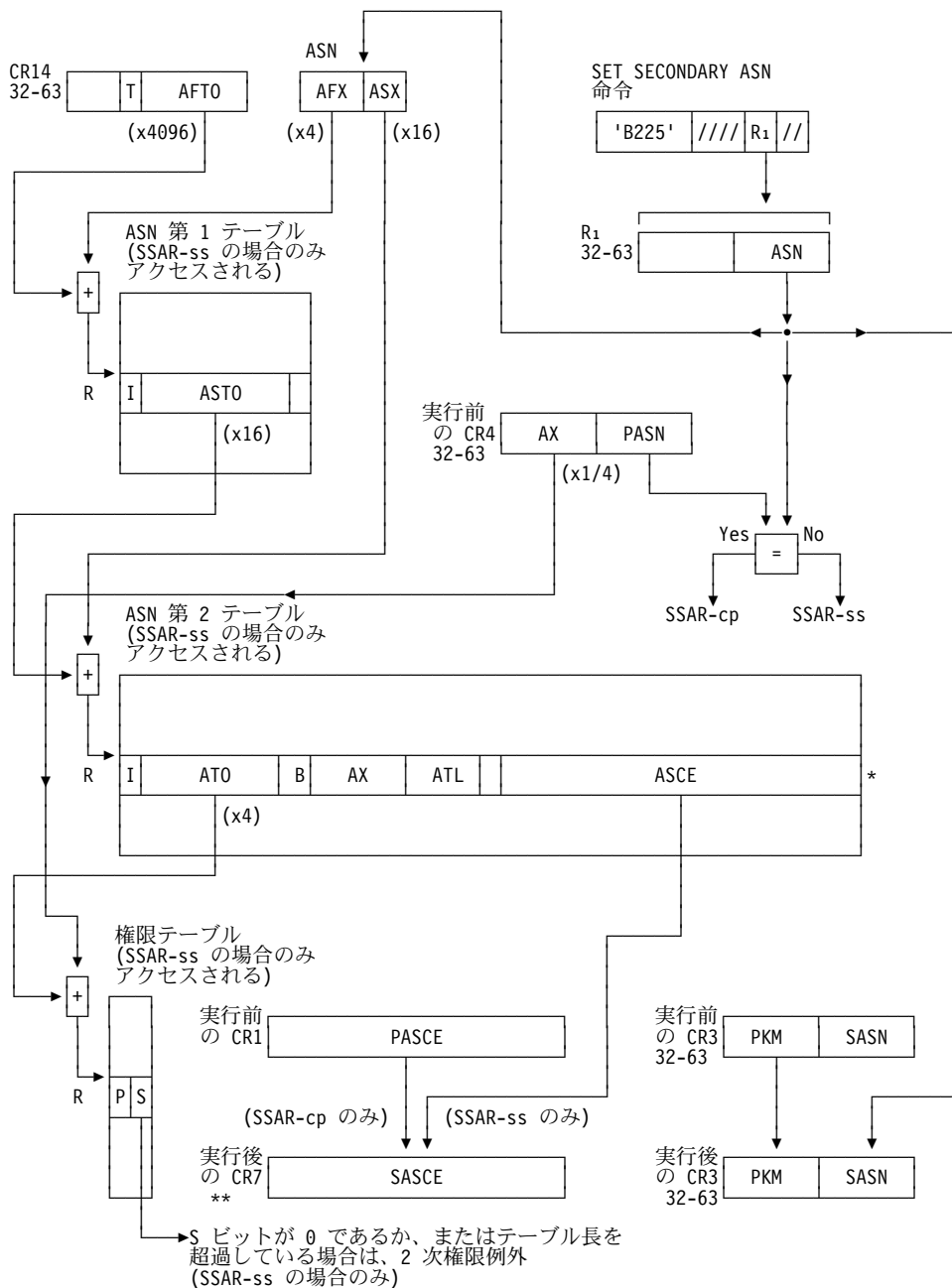
条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (権限テーブル・エントリー、SSAR-ss のみ)
- ASN 変換例外 (SSAR-ss のみ)
- 2 次権限例外 (SSAR-ss のみ)
- 特殊操作例外
- サブスペース置換例外 (SSAR-ss のみ)
- トレース例外

- | |
|---|
| <p>1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。</p> <p>7.A 命令の第 2 ハーフワードについてのアクセス例外。</p> <p>7.B DAT がオフであるか、または ASN 変換制御 (制御レジスター 14 のビット 44) が 0 であるために起きる特殊操作例外。</p> <p>8.A トレース例外。</p> <p>8.B.1 ASN 変換例外 (SSAR-ss のみ)。</p> <p>注: この図では詳細を示しませんが、8.B.1 の後では優先順位に関係なくサブスペース置換例外が起きることがあります。</p> <p>8.B.2 権限テーブル・エントリーがテーブルの外にあるために起きる 2 次権限例外 (SSAR-ss の場合のみ)。</p> <p>8.B.3 権限テーブル・エントリーへのアクセスについてのアドレッシング例外 (SSAR-ss の場合のみ)。</p> <p>8.B.4 権限テーブル・エントリー内の S ビットが 0 であるために起きる 2 次権限例外 (SSAR-ss の場合のみ)。</p> |
|---|

図 10-28. 実行の優先順位: SET SECONDARY ASN



R: アドレスは実アドレスです。
 *: ASTE は 64 バイトですが、最後の 48 バイトは示されていません。
 **: SSAR-ss の場合のみ、SASCE のビット 0-55 および 58-63 がサブスペース ASCE からの情報で置き換えられることがあります。

図 10-29. SET SECONDARY ASN の実行

SET STORAGE KEY EXTENDED

SSKE R₁,R₂ [RRE]

'B22B'	////////	R ₁	R ₂
0	16	24	28 31

汎用レジスタ R₂ の内容が指しているアドレスにある 4K バイト・ブロックの記憶キーが、汎用レジスタ R₁ のビットで置き換えられます。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 40-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-39 および 52-63 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 33-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 0-51 が、実記憶域内の 4K バイト・ブロックを指定し、ビット 52-63 は無視されます。

記憶域ブロックを指すアドレスは実アドレスなので、動的アドレス変換は行われません。また、記憶キーに対する参照には記憶保護例外は適用されません。

汎用レジスタ R₁ のビット位置 56-62 から、新しい 7 ビットの記憶キー値が取得されます。このレジスタのビット位置 0-55 および 63 の内容は、無視されます。

操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (汎用レジスタ R₂ に指定されているアドレス)
- 特権命令例外

SET SYSTEM MASK

SSM D₂(B₂) [S]

'80'	////////	B ₂	D ₂
0	8	16	20 31

現 PSW のビット0-7 が、第 2 オペランド・アドレスが指している記憶位置のバイトで置き換えられます。

特別な条件

SSM 抑止制御ビット (レジスタ 0 のビット 33) が 1 で、CPU が監視プログラム状態にあるときは、特殊操作例外が認識されます。

PSW にロードする値については、ロードの前に妥当性検査は行われません。ただし、PSW のビット位置 0 および 2-4 の内容がすべて 0 でない場合は、ロードの直後に、指定例外が認識され、プログラム割り込みが起きます。この場合は、命令は完了し、命令長コードは 2 に設定されます。この指定例外は、この命令ではプログラム例外としてリストされますが、これについては、6-9 ページの『早期例外認識』で説明されています。この例外は、この命令の実行が原因で発生するか、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 特殊操作例外
- 指定例外

SIGNAL PROCESSOR

SIGP R₁,R₃,D₂(B₂) [RS]

'AE'	R ₁	R ₃	B ₂	D ₂
0	8	12	16 20	31

第 3 オペランドに含まれている CPU アドレスが指す CPU に、8 ビットの指令コード、および 32 ビットのパラメータ (要求された場合) が伝送されます。結果は条件コードで示され、また、第 1 オペランド位置のビット位置 32-63 に集められた状況情報により詳細が示されることもあります。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。代わりに、アドレスのビット 56-63 に 8 ビットの指令コードが含まれています。第 2 オペランド・アドレスのビット 0-55 は無視されます。

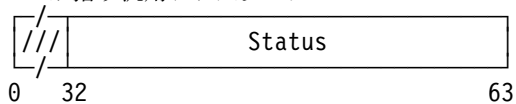
指令コードは、アドレッシングされた CPU が行う機能を指定します。指令コードの割り当てと定義は、4-48ページの『CPU の信号発信と応答』に示してあります。

汎用レジスタ R₃ のビット位置 48-63 に含まれている 16 ビットの 2 進数が、CPU アドレスを形成します。このレジスタのビット 0-47 は無視されます。指定された指令がアーキテクチャ設定指令である場合は、CPU アドレスは無視されます。構成内の他のすべての CPU がアドレッシングされるものと見なされます。

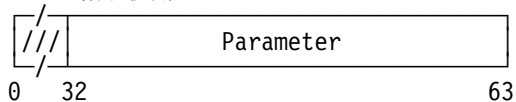
ビット位置 32-63 に 32 ビット・パラメータを含むレジスタは、R₁ または R₁+1 のどちらか、奇数番号を持つ方のレジスタです。パラメータが提供されるかどうかおよびどのような目的に使用されるかは、指令コードによって異なります。

上記で述べたオペランドのフォーマットは、以下のとおりです。

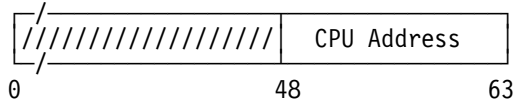
R₁ が指す汎用レジスタ:



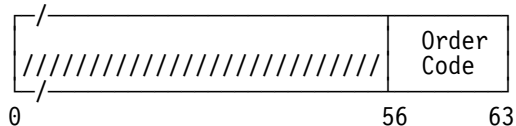
R₁ または R₁ + 1 のうち、奇数番号を持つ方のレジスタが指す汎用レジスタ:



R₃ が指す汎用レジスタ:



第 2 オペランド・アドレス:



この操作の開始前と完了後に、逐次化機能が実行されません。

指令コードが受け入れられ、0 以外の状況が戻されなかった場合は、条件コード 0 が設定されます。状況情報がこの CPU で生成されるか、またはアドレスを指定した CPU から戻されると、その状況が汎用レジスタ R₁

のビット位置 32-63 に入れられ、ビット 0-31 は変更されず、そして条件コード 1 が設定されます。

アドレッシングされた CPU へのアクセス・パスが使用中である場合、またはアドレッシングされた CPU は稼働中であるが、この指令コードに回答できる状態にはない場合は、条件コード 2 が設定されます。

アドレッシングされた CPU が稼働状態にない場合 (つまり、その CPU がインストール・システム内にないか、構成内にないか、幾つかの技術員テスト・モードのいずれかになっているか、または電源がオフになっている場合) は、条件コード 3 が設定されます。

結果の条件コード:

- 0 指令コードは受理されました。
- 1 状況が格納されました。
- 2 使用中。
- 3 非稼働。

プログラム例外:

- 特権命令例外

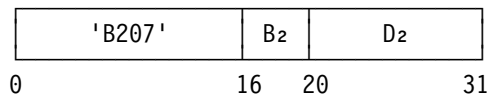
プログラミング上の注意:

1. SIGNAL PROCESSOR の場合の条件コード設定については、4-48ページの『CPU の信号発信と応答』で詳しく説明されています。
2. 将来、追加ビットを使用する新機能がインストールされたときに、現在のプログラムが正しく実行されるようにするために、第 2 オペランド・アドレスの未使用ビット位置、および制御レジスタ R₃ のビット位置 32-47 は、必ず 0 にしておくようにしてください。
3. ある種の SIGNAL PROCESSOR 指令は、主として特殊な状況下で使用することを目的として用意されています。このような指令は、補助の保守またはサービス・プロセッサの助けを借りてインプリメントされていることがあり、したがって、実行に数秒を要する場合があります。特定の指令が提供するすべての機能を必要とする場合以外は、他の指令との組み合わせを適切なプログラミング・サポートとともに利用する方が、特定機能をもっと速く実行できる場合があります。頻繁に使用される指令は、緊急信号指令、外部呼び出し指令、およびセンス指令だけです。再始動、プレフィックス設定、アドレス指定状況記憶、開始、停止、停止/状況記憶、アーキテクチャ設定、およびその他のすべてのリセット指令は、頻繁な使用を想定しているものではなく、したがって頻繁に使用される指令よりはるかにパフォ

パフォーマンスが低下します。受け入れ側の CPU がまだ停止していないときに、プレフィックス設定指令に代わるパフォーマンスの高い方法として、緊急信号指令または外部呼び出し指令を使用し、続いて、アドレッシングされた CPU で SET PREFIX 命令を実行する方法があります。通常、緊急信号指令または外部呼び出し指令を使用し、次に、アドレッシングされた CPU で PURGE TLB 命令を実行する方が、プレフィックス設定指令を使用した場合より、TLB からのエントリーの消去が迅速に行われます。

STORE CLOCK COMPARATOR

STCKC D₂(B₂) [S]



クロック・コンパレーターの現在値が、第 2 オペランド・アドレスに指定されているダブルワードの位置に格納されます。

クロック・コンパレーターの中の、TOD クロックと比較されない右端ビット位置には、すべて 0 が入れられます。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

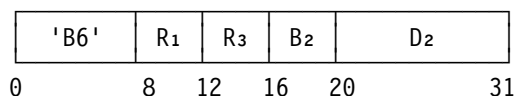
条件コード: コードは変更されません。

プログラム例外:

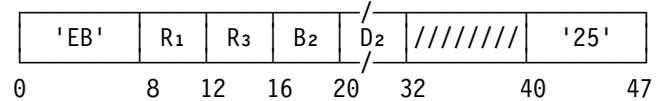
- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

STORE CONTROL

STCTL R₁,R₃,D₂(B₂) [RS]



STCTG R₁,R₃,D₂(B₂) [RSE]



制御レジスター R₁ から制御レジスター R₃ までの一連のレジスターのビットが、第 2 オペランド・アドレスが指す記憶位置に格納されます。

STORE CONTROL (STCTL) の場合は、制御レジスターのビット 32-63 が、第 2 オペランド・アドレスから始まる連続したワードに格納され、レジスターのビット 0-31 は無視されます。STORE CONTROL (STCTG) では、制御レジスターのビット位置 0-63 が、第 2 オペランド・アドレスから始まる連続したダブルワードに格納されます。

制御レジスターの内容が格納される記憶域は、第 2 オペランド・アドレスが指す位置から始まり、指定された制御レジスターの数と同数の記憶域ワード (STCTL の場合) または記憶域ダブルワード (STCTG の場合) に達するまで続きます。制御レジスターの内容は、制御レジスター R₁ から始まり、制御レジスター R₃ に至るまで、レジスター番号の昇順に従って格納されます。制御レジスター 15 の次は制御レジスター 0 になります。制御レジスターの内容は変更されません。

特別な条件

第 2 オペランドは、STCTL の場合はワード境界に、そして STCTG の場合はダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

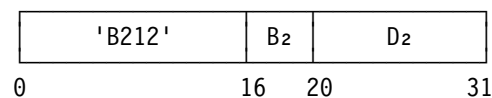
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

STORE CPU ADDRESS

STAP D₂(B₂) [S]



マルチプロセッシング構成において、この CPU を識別する CPU アドレスが、第 2 オペランド・アドレスが指すハーフワード位置に格納されます。

特別な条件

オペランドは、ハーフワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

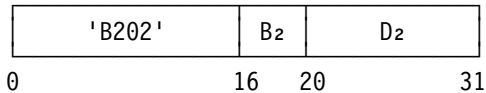
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

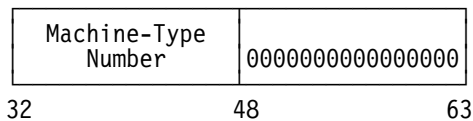
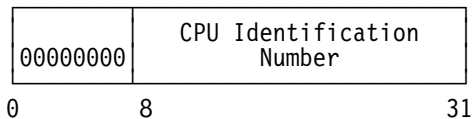
STORE CPU ID

STIDP D₂(B₂) [S]



CPU を識別する情報が、第 2 オペランド・アドレスが指すダブルワード位置に格納されます。

格納される情報のフォーマットは、以下のとおりです。



ビット位置 8-31 には、4 ビットの数字 6 桁から成る CPU 識別番号が入ります。これらの数字は、一部またはすべてが、CPU に付けられている物理シリアル番号から選択されます。CPU 識別番号フィールドの内容とマシン・タイプ番号の組み合わせにより、CPU を一意的に識別することができます。

ビット位置 32-47 には、CPU のマシン・タイプ番号が入ります。ビット位置 48-63 はすべて 0 になります。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

プログラミング上の注意:

1. プログラムでは、CPU 識別番号には、0-9 の数字のほかに A-F の数字も含まれる可能性があることを考慮に入れる必要があります。
2. CPU 識別番号とマシン・タイプ番号の組み合わせは固有の CPU 識別を形成するので、これを使用して、結果を個々のマシンに関連付けることができます。
3. 「Enterprise Systems Architecture/390 Principles of Operation」の SA22-7201-03 より前のバージョンでは、マシン・タイプ番号フィールドは、モデル番号フィールドと呼ばれていました。
4. ESA/390 では、格納される情報のビット位置 0-7 にはバージョン・コードが含まれています。バージョン・コードが 0 以外である場合は、通常それは、モデルのモデル番号およびモデルに含まれている CPU の番号を示します。ESA/390 および z/Architecture のどちらでも、STORE SYSTEM INFORMATION 命令を使用して、モデル番号およびモデル内の CPU の番号を判別できます。
5. 現在のマシン・タイプの場合、CPU 識別番号は以下の 16 進数フォーマットで表記されます。
 - 基本モードでは「Annnnn」
 - LPAR (論理分割) モードでは「LPnnnn」

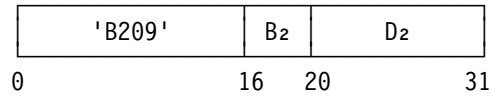
ここで:

- A は CPU の CPU アドレスです。
- L は論理 CPU アドレスです。
- P は論理区画 ID です。
- n は、CPU のシリアル番号から導き出された数字です。

上記の用語のうち本書で定義されていないものについては、マシンのマニュアルの中で定義されています。

STORE CPU TIMER

STPT D₂(B₂) [S]



CPU タイマーの現在値が、第 2 オペランド・アドレスが指しているダブルワード位置に格納されます。

CPU タイマーにより更新されない右端のビット位置には、すべて 0 が入ります。

特別な条件

オペランドは、ダブルワード境界に合わせて指定しなければなりません。さもないと、指定例外が認識されません。

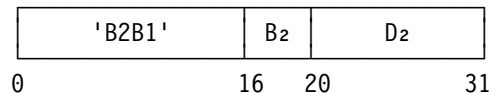
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

STORE FACILITY LIST

STFL D₂(B₂) [S]



機能に関する情報を示す一連のビットが、実アドレス 200 にあるワードに格納されます。

ビット 0 が 1 のときは、第 7 章、『一般命令』および第 10 章、『制御命令』の中の命令要約図の中で「N3」のマークが付けられた z/Architecture 命令が、ESA/390 モードの CPU で使用できることを示します。

ビット 1 が 1 のときは、z/Architecture アーキテクチャー・モードが CPU にインストールされていることを示します。

ビット 2 が 1 のときは、z/Architecture アーキテクチャー・モードがアクティブであることを意味します。

ビット 16 が 1 のときは、拡張変換機能 2 が CPU にインストールされていることを示します。

3-15 および 17-31 は未割り当てであり、0 が格納されます。

第 2 オペランド・アドレスは無視されますが、将来の拡張に備えて 0 にしておく必要があります。

キー制御保護および低アドレス保護は適用されません。

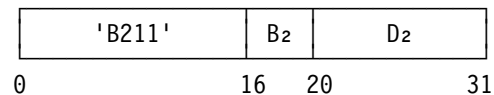
条件コード: コードは変更されません。

プログラム例外:

- 特権命令例外

STORE PREFIX

STPX D₂(B₂) [S]



プレフィックス・レジスターのビット位置 33-50 の内容が、第 2 オペランド・アドレスが指すワード位置のビット位置 1-18 に格納され、そのワードのビット位置 0 および 19-31 には 0 が格納されます。

特別な条件

オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されません。

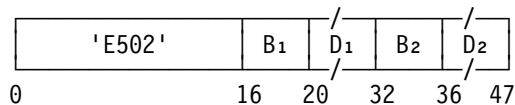
条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

STORE REAL ADDRESS

STRAG D₁(B₁), D₂(B₂) [SSE]



第 2 オペランドの仮想アドレスに対応する 64 ビット実アドレスが、第 1 オペランド・アドレスが指す位置のダブルワードに格納されます。

DAT がオンかオフに関係なく、B₂ および D₂ フィールドに指定されている仮想アドレスが、動的アドレス変換機能により変換されます。

DAT は、アドレス・スペース制御エレメントを使用して実行されます。使用するエレメントは、次の表に示すように、アドレス・スペース制御ビット (PSW のビット 16 および 17) の現在の値によって決まります。

PSW のビット

ビット	DAT により使用されるアドレス・スペース制御エレメント
00	制御レジスター 1 の内容
10	制御レジスター 7 の内容
01	B ₂ フィールドに指定されているアクセス・レジスターに、アクセス・レジスター変換 (ART) プロセスを適用することによって得られるアドレス・スペース制御エレメント
11	制御レジスター 13 の内容

ART および DAT は、それぞれ、ART 索引バッファ (ALB) および変換索引バッファ (TLB) を利用して実行することができます。

結果の 64 ビット・アドレスが、第 1 オペランド位置に格納されます。

変換済みアドレスについて、境界合わせが適正かどうか、またはアドレッシング例外や記憶保護例外がないかどうかは、検査されません。

オペランドのアドレスの計算は、現 PSW のビット 31 および 32 に指定されている現アドレッシング・モードに従って行われます。

領域テーブル・エントリ (使用される場合) のアドレス、および、セグメント・テーブル・エントリとページ・テーブル・エントリのアドレスは、現アドレッシング・モードに関係なく、64 ビット・アドレスとして扱われます。これらのエントリのアドレスが実アドレスとして扱われるか絶対アドレスとして扱われるかは、予測不能です。

特別な条件

第 1 オペランドは、ダブルワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

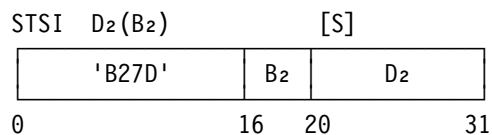
プログラム例外:

- アクセス例外 (指定された位置についてアドレッシング例外または記憶保護例外がある場合を除き、取り出し: オペランド 2。格納: オペランド 1)
- 特権命令例外
- 指定例外

プログラミング上の注意: STORE REAL ADDRESS と LOAD REAL ADDRESS の間には、以下のような違いがあります。

- 24 ビットまたは 31 ビットのアドレッシング・モードの LOAD REAL ADDRESS (LRA) では、実アドレスのビット 0-32 がすべて 0 であれば、そのアドレスのビット 33-63 がロードされ、ビット 0-32 がすべて 0 ではない場合は、特殊操作例外が認識されます。64 ビット・アドレッシング・モードの LRA、およびアドレッシング・モードに関係なく LOAD REAL ADDRESS (LRAG) では、実アドレスのビット 0-63 がロードされます。これに対して、STORE REAL ADDRESS は、どのアドレッシング・モードでも、実アドレスのビット 0-63 がロードされます。
- LOAD REAL ADDRESS では、ほとんどのアクセス例外条件は例外として認識されず、代わりに、その条件が発生したことを示す条件コードが設定されます。STORE REAL ADDRESS では、すべてのアクセス例外条件が例外として認識され、その結果プログラム割り込みが起きます。

STORE SYSTEM INFORMATION



汎用レジスター 0 中の機能コードに応じて、プログラムを実行している構成のレベルを示す識別が汎用レジスター 0 に入れられるか、または、構成の 1 つ以上のコンポーネントに関する情報が、システム情報ブロック (SYSIB) に格納されます。コンポーネントに関する情報を要求する場合は、汎用レジスター 0 の内容と汎用レジスター 1 の内容により、必要な情報を指定します。SYSIB がある場合は、第 2 オペランド・アドレスにより SYSIB を指定します。

マシンには、1 レベル、2 レベル、または 3 レベルの構成があるものと考えられます。各レベルは以下のとおりです。

1. 基本マシン。これは、基本モードで稼働しているときと同じ状態のマシンです。
2. 論理区画。これは、マシンが LPAR (論理分割) モードで稼働しているときに提供されます。論理区画は、マシンの一部となっている LPAR ハイパーバイザーにより提供されます。マシンが LPAR モードで稼働しているときでも、基本マシンは存在します。
3. 仮想計算機。これは、基本マシンが実行しているかまたは論理区画内で実行されている仮想計算機 (VM) 制御プログラムにより提供されます。仮想計算機は、それ自身が VM 制御プログラムを実行でき、この制御プログラムは、さらに高いレベルの (基本マシンからの隔たりがさらに大きい) 仮想計算機を提供します。これはレベル 3 の構成と見なされます。

基本モード、LPAR モード、論理区画、ハイパーバイザー、および仮想計算機という用語、および特にこれらの用語に関連した用語については、本書では定義しません。これらの用語は、マシンのマニュアルで定義されています。

レベル 1 構成 (基本マシン) により実行されているプログラムは、その構成に関する情報を要求することができます。レベル 2 構成 (論理区画内) で実行されているプログラムは、論理区画とその基礎となっている基本マシンに関する情報を要求することができます。レベル 3 構成 (仮想計算機) により実行されているプログラム

は、その仮想計算機とその基礎となっている 1 つまたは 2 つのレベルに関する情報を要求することができます。基本マシンは常に基礎となっており、基本マシンと仮想計算機の間で論理区画がある場合とない場合があります。仮想計算機に関する情報が要求された場合は、プログラムを実行している構成と、仮想計算機の基礎となっているすべてのレベルに関する情報が提供されます。いずれの場合も、この命令をインプリメントしているレベルについてのみ情報が提供されます。

どの操作を行うかを決定する機能コードは、汎用レジスター 0 のビット位置 32-35 にあるビット符号なし 2 進整数で、その値は以下のとおりです。

機能

コード 要求される情報

0	現構成レベル番号
1	レベル 1 (基本マシンに関する情報)
2	レベル 2 (論理区画) に関する情報
3	レベル 3 (仮想計算機) に関する情報
4-15	なし。これらのコードは予約済みです。

無効な機能コード

プログラムを実行している構成のレベルを、現レベルと呼びます。0 以外の機能コードにより指定されている構成レベルを、指定されたレベルと呼びます。指定されたレベルの機能コード番号が現レベルより大きいとき、その機能コードは無効であり、条件コードは 3 に設定され、他のアクション (検査も含む) は行われません。

有効な機能コード

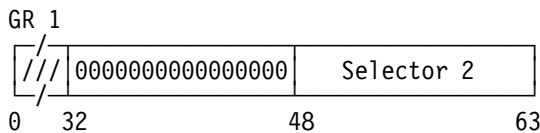
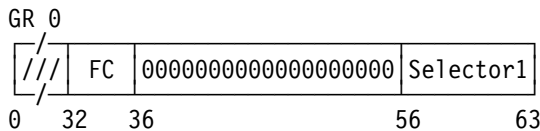
機能コードが現レベルの番号と同じかまたはそれ以下である場合、そのコードは有効であると見なされます。この場合は、汎用レジスター 0 のビット 36-55 および汎用レジスター 1 のビット 32-47 が 1 でなければなりません。さもないと、指定例外が認識されます。汎用レジスター 0 および 1 のビット 0-31 は、常に無視されます。

機能コードが 0 のときは、現構成レベル (基本マシンの場合 1、論理区画の場合は 2、仮想計算機の場合は 3) を識別する符号なしの 2 進整数が、汎用レジスター 0 のビット位置 32-35 に置き換えられ、条件コードは 0 に設定され、その他のアクションは行われません。

機能コードが有効で、かつ 0 以外である場合は、汎用レジスター 0 および 1 には、以下のように、要求された情報に関する追加の指定が含まれます。

- 汎用レジスタ 0 のビット位置 56-63 には、セクター 1 と呼ばれる符号なしの 2 進整数が含まれます。これは、指定された構成の 1 つまたは複数のコンポーネントを指定します。
- 汎用レジスタのビット 48-63 には、セクター 2 と呼ばれる符号なしの 2 進整数が含まれます。これは、要求された情報のタイプを指定します。

汎用レジスタ 0 および 1 の内容は、以下のとおりです。



機能コードが有効で、かつ 0 以外である場合は、システム情報ブロック内の第 2 オペランド位置を先頭とする場所に情報が格納されます。SYSIB は 4K バイトで、4K バイト境界で始まっていなければなりません。さもないと、セクター 1 および セクター 2 に応じて、また SYSIB に対する参照が原因でアクセス例外が認識されたかどうかに応じて、指定例外が認識されることがあります (『特別な条件』を参照)。

セクター 1 がとる値は次のいずれかです。

セクタ

ター 1 要求される情報

- | | |
|-------|---------------------------------|
| 0 | なし。このセクターは予約済みです。 |
| 1 | 指定された構成レベルに関する情報。 |
| 2 | 指定された構成レベル内の 1 つ以上の CPU に関する情報。 |
| 3-255 | なし。これらのセクターは予約済みです。 |

セクター 1 が 1 のときは、セクター 2 がとる値は次のいずれかです。

セクタ

ター 1 が 1

のときの

セクタ

ター 2

要求される情報

- | | |
|-----------|---------------------|
| 0 | なし。このセクターは予約済みです。 |
| 1 | 指定された構成レベルに関する情報。 |
| 2-65, 535 | なし。これらのセクターは予約済みです。 |

セクター 1 が 2 のときは、セクター 2 がとる値は次のいずれかです。

セクタ

ター 1 が 2

のときの

セクタ

ター 2

要求される情報

- | | |
|-----------|--------------------------------------|
| 0 | なし。このセクターは予約済みです。 |
| 1 | 指定された構成レベル内でプログラムを実行している CPU に関する情報。 |
| 2 | 指定された構成レベル内のすべての CPU に関する情報。 |
| 3-65, 535 | なし。これらのセクターは予約済みです。 |

機能コード、セクター 1、およびセクター 2 の特定の組み合わせのみが有効です。有効な組み合わせは、図 10-30 に示すとおりです。

機能コード	セクタ 1	セクタ 2	要求される情報
0	-	-	現構成レベル番号
1	1	1	基本マシン構成
1	2	1	基本マシン CPU
1	2	2	基本マシン CPU
2	2	1	論理区画 CPU
2	2	2	論理区画 CPU
3	2	2	仮想計算機 CPU
説明:			
- 無視されます。			

図 10-30. STORE SYSTEM INFORMATION の場合の機能コード、セクター 1、およびセクター 2 の有効な組み合わせ

指定された機能コード、セクター 1、およびセクター 2 の組み合わせが無効な場合 (図 10-30 に示されてい

るもの以外の組み合わせの場合)、あるいは、組み合わせは有効であっても、指定されたレベルがこの命令をインプリメント (または完全インプリメント) していないか、またはそのレベルの必須部分がまだインストールされていないか初期化されていないため、要求された情報が使用可能でない場合は、例外が認識されていなければ (『特別な条件』を参照)、条件コードが 3 に設定されます。機能コードが 0 以外であり、組み合わせが有効で、要求された情報が使用可能で、かつ例外が生じていない場合は、要求された情報が、第 2 オペランド・アドレスの位置にあるシステム情報ブロック (SYSIB) に格納されます。

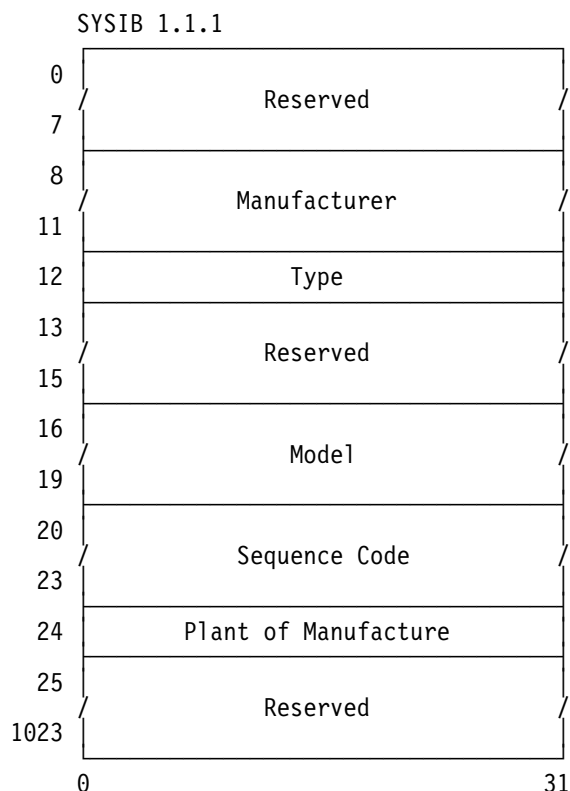
格納の前に、SYSIB の一部が取り出される場合も、全部が取り出される場合もあります。

参照の中では、SYSIB は「SYSIB fc.s1.s2」を使用して識別できます。ここで、fc、s1、および s2 は、それぞれ、機能コード、セレクター 1、およびセレクター 2 の値です。

以下のセクションでは、図および関連のテキストを使用して、定義済みの各 SYSIB について説明します。図の中で、左側に示されているオフセットはワード値を表しています。「構成」は、機能コードが指定している構成レベル (情報が要求されている構成レベル) を表します。

SYSIB 1.1.1 (基本マシン構成)

SYSIB 1.1.1 のフォーマットは以下のとおりです。



予約済み (Reserved): ワード 0-7、13-15、および 25-63 の内容は予約済みであり、0 が格納されています。ワード 64-1023 の内容は予約済みであり、0 が格納されるか、または変更されません。

製造元 (Manufacturer): ワード 8-11 には、構成の製造元を示す 16 文字 (0-9 または大文字の A-Z) の EBCDIC 名が含まれます。この名前は左寄せにされ、必要に応じて末尾ブランクが付加されます。

タイプ (Type): ワード 12 には、構成のタイプ番号を表す 4 文字 (0-9) の EBCDIC が含まれます。(これは、STORE CPU ID の定義ではマシン・タイプ番号と呼ばれています。)

モデル (Model): ワード 16-19 には、構成のモデル ID を示す 16 文字 (0-9 または大文字の A-Z) の EBCDIC が含まれます。モデル ID は左寄せにされ、必要に応じて末尾ブランクが付加されます。(これは、STORE CPU ID の『プログラミング上の注意 4 (10-92 ページ)』では、モデル番号と呼ばれています。)

シーケンス・コード (Sequence Code): ワード 20-23 には、構成のシーケンス・コードを示す 16 文字 (0-9 または大文字の A-Z) の EBCDIC が含まれます。

シーケンス・コードは右寄せにされ、必要に応じて先行ゼロ (EBCDIC) が付加されます。

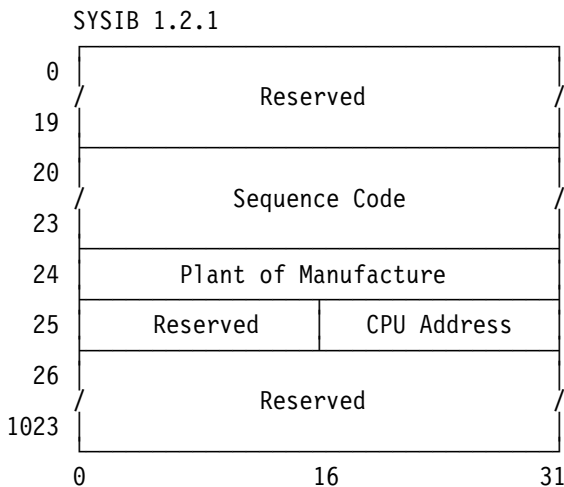
製造工場 (Plant of Manufacture): ワード 24 には、構成の製造工場を識別する 4 文字 (0-9 または大文字の A-Z) の EBCDIC コードが含まれます。このコードは左寄せにされ、必要に応じて末尾ブランクが付加されます。

プログラミング上の注意: SYSIB 1.1.1 のフィールドは、「*Common I/O-Device Commands and Self Description*」(SA22-7204) で説明されているノード記述子のフィールドに似ています。ただし、SYSIB のフィールドの内容は、対応するノード記述子のフィールドの内容とは必ずしも同じではない場合があります。これは、SYSIB フィールドには次のような特性があるためです。

- 収容できる文字数が多い。
- 使用できる文字のタイプについての柔軟性が高い。
- フィールド内での情報の位置調整のしかたが異なる。
- シーケンス・コード・フィールドなどのように、フィールド内容を判別するための方式が異なっていることがある。

SYSIB 1.2.1 (基本マシン CPU)

SYSIB 1.2.1 のフォーマットは以下のとおりです。



予約済み (Reserved): ワード 0-19、ワード 25 のバイト 0 と 1、およびワード 26-63 は予約済みであり、0 が格納されます。ワード 64-1023 の内容は予約済みであり、0 が格納されるか、または変更されません。

シーケンス・コード (Sequence Code): ワード 20-23 には、CPU のシーケンス・コードを示す 16 文字 (0-9 または大文字の A-Z) の EBCDIC が含まれます。このコードは右寄せにされ、必要に応じて先行ゼロ (EBCDIC) が付加されます。

このシーケンス・コードは、STORE CPU ID により格納される CPU 識別番号と同じものではありません。シーケンス・コードは、CPU シリアル番号から、製造工場識別部分を除外した残りの部分です。

製造工場 (Plant of Manufacture): ワード 24 には、CPU の製造工場を識別する 4 文字 (0-9 または大文字の A-Z) の EBCDIC コードが含まれます。このコードは左寄せにされ、必要に応じて末尾ブランクが付加されます。

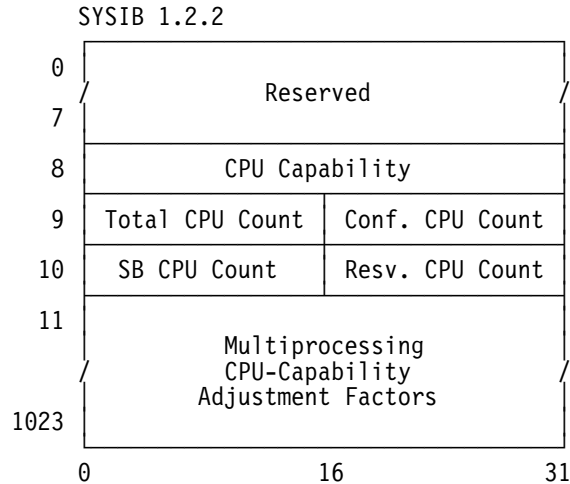
CPU アドレス (CPU Address): ワード 25 のバイト 2 および 3 には、マルチプロセッシング構成の中でこの CPU を識別するために使用される CPU アドレスが含まれます。CPU アドレスは、16 ビットの符号なし 2 進整数です。

基本モードで稼働しているマシンによりプログラムが実行されている場合は、CPU アドレスは STORE CPU ADDRESS により格納されるアドレスと同じです。

プログラミング上の注意: 同一構成内の複数の CPU が同じシーケンス・コードを持つことができますが、その場合、固有の CPU 識別を確立するために他の情報 (CPU アドレスなど) を使用する必要があります。これに対して、STORE CPU ID により格納される CPU 識別番号は、シリアル番号の未記述の部分と、その番号を特定の CPU に固有のものにする値を組み合わせ導き出されます。

SYSIB 1.2.2 (基本マシン CPU)

SYSIB 1.2.2 のフォーマットは以下のとおりです。



予約済み (Reserved): ワードの 0-7 と、この SYSIB の Adjustment Factors (調整係数) リストの後からワード 64 までの内容は、予約済みであり、0 が格納されています。ワード 64-1023 の内容は予約済みであり、0 が格納されるか、または変更されません。

CPU 能力 (CPU Capability): ワード 8 には、構成内の 1 つの CPU の能力を示す 32 ビットの符号なし 2 進整数が含まれます。この整数を生成するために使用されるアルゴリズムについての、正式な記述はありません。この整数は、この CPU を他の CPU モデルと比較した場合の相対的な能力を示す値として使用されます。

能力値は、構成内の各 CPU に適用されます。つまり、構成内のすべての CPU が同じ能力を備えています。

総 CPU 数 (Total CPU Count): ワード 9 のバイト 0 および 1 には、構成内の CPU の合計数を示す 16 ビットの符号なし 2 進整数が含まれます。この数には、構成済み状態、スタンバイ状態、または予約済み状態にあるすべての CPU が含まれます。

構成済み CPU 数 (Configured CPU Count): ワード 9 のバイト 2 および 3 には、構成済み状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。CPU が構成済み状態になるのは、構成内に含まれていて、かつプログラムの実行のために使用できるからです。

スタンバイ CPU 数 (Standby CPU Count): ワード 10 のバイト 0 および 1 には、スタンバイ状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。CPU がスタンバイ状態になるのは、その

CPU が構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその CPU を構成済み状態に変更できることです。

予約済み CPU 数 (Reserved CPU Count): ワード 10 のバイト 2 および 3 には、予約済み状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。CPU が予約済み状態になるのは、その CPU が構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその CPU を構成済み状態にすることができないときです。(手動アクションによって、予約済み CPU をスタンバイ状態または構成済み状態にすることはできる場合があります。)

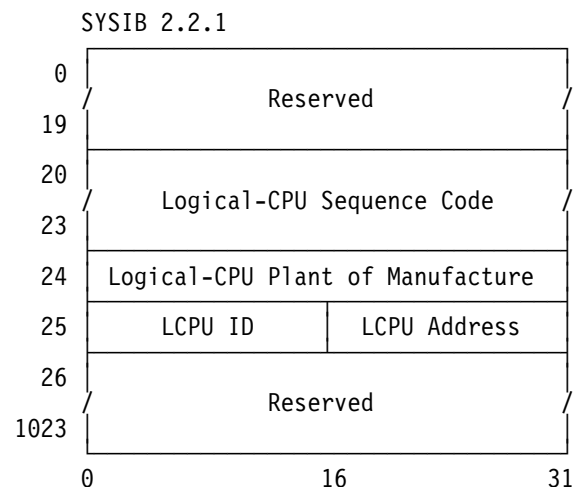
マルチプロセッシング CPU 能力調整係数 (Multiprocessing CPU-Capability Adjustment Factors):

この SYSIB のワード 11 のバイト 0 および 1 を先頭とする連続した 2 バイト・フィールドには、それぞれ、CPU 能力フィールドに含まれている値の調整係数 (パーセンテージ) を表す 16 ビットの符号なし 2 進整数が含まれます。

調整係数フィールドの数は、総 CPU 数フィールドに指定されている CPU 数から 1 を引いた数です。調整係数フィールドは、構成内の構成済み状態にある CPU の数に対応しています。つまり、最初の調整係数フィールドは、構成済み状態の CPU が 2 つある構成に対応します。以後の連続した調整係数フィールドは、それぞれ、構成済み状態の CPU の数が直前のフィールドより 1 つだけ多い構成に対応します。

SYSIB 2.2.1 (論理区画 CPU)

SYSIB 2.2.1 のフォーマットは以下のとおりです。



予約済み (Reserved): ワード 0-19 および 26-63 の内容は予約済みであり、0 が格納されています。ワード 64-1023 の内容は予約済みであり、0 が格納されるか、または変更されません。

論理 CPU シーケンス・コード (Logical-CPU Sequence Code): ワード 20-23 には、論理 CPU のシーケンス・コードを示す 16 文字 (0-9 または大文字の A-Z) の EBCDIC が含まれます。このコードは右寄せにされ、必要に応じて先行ゼロ (EBCDIC) が付加されます。

論理 CPU シーケンス・コード・フィールドの内容は、STORE CPU ID により格納される論理 CPU 識別番号と同じものではありません。論理 CPU シーケンス・コードは、論理 CPU シリアル番号から論理 CPU 製造工場識別部分を除外した残りの部分です。

論理 CPU 製造工場 (Logical-CPU Plant of Manufacture): ワード 24 には、論理 CPU の製造工場を識別する 4 文字 (0-9 または大文字の A-Z) の EBCDIC コードが含まれます。このコードは左寄せにされ、必要に応じて末尾ブランクが付加されます。

論理 CPU ID (Logical-CPU ID): ワード 25 のバイト 0 および 1 には、16 バイトの符号なし 2 進整数が含まれます。この整数と論理 CPU アドレスを組み合わせることで使用することにより、この論理 CPU が、同じ LPAR ハイパーバイザーから提供される他の論理 CPU と区別されます。

論理 CPU アドレス (Logical-CPU Address): ワード 25 のバイト 2 および 3 には、レベル 2 構成の中でこの論理 CPU を識別するために使用される論理 CPU アドレスが含まれます。論理 CPU アドレスは、16 ビットの符号なし 2 進整数です。

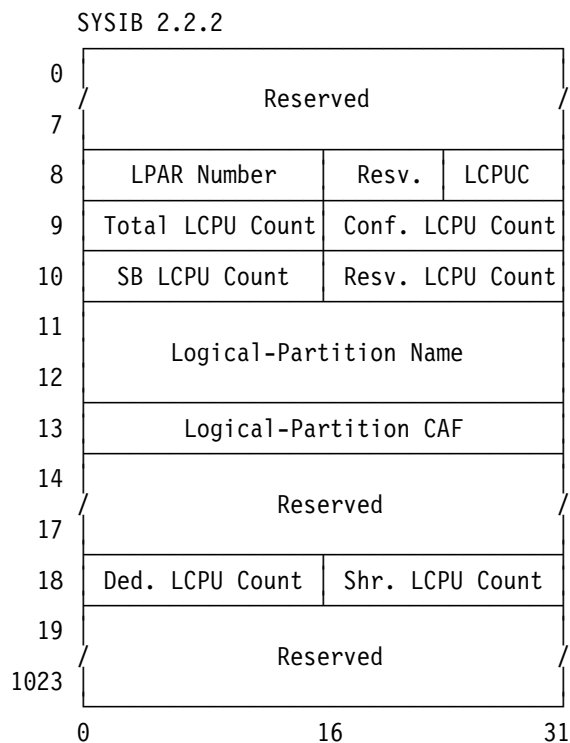
論理 CPU アドレス・フィールドには、マシンが LPAR モードで稼働しているときに、STORE CPU ADDRESS により格納されるものと同じ情報が含まれます。

プログラミング上の注意: 同じレベル 2 構成内の複数の論理 CPU が同じ論理 CPU シーケンス・コードを持つことができるため、固有の論理 CPU 識別を確立するために、他の情報 (論理 CPU アドレスなど) を使用することが必要になる場合があります。これに対して、

STORE CPU ID により格納される論理 CPU 識別番号は、論理 CPU シリアル番号の未記述の部分と、その番号を特定の論理 CPU に固有のものにする値を組み合わせることで導き出されます。

SYSIB 2.2.2 (論理区画 CPU)

SYSIB 2.2.2 のフォーマットは以下のとおりです。



予約済み (Reserved): ワード 0-7、ワード 8 のバイト 2、ワード 14-17、およびワード 19-63 は予約済みであり、0 が格納されています。ワード 64-1023 の内容は予約済みであり、0 が格納されるか、または変更されません。

Logical-Partition Number (論理区画番号): ワード 8 のバイト 0 および 1 には、レベル 2 構成の番号を示す 16 ビットの符号なし 2 進整数が含まれます。この番号は、この構成を、同じ LPAR ハイパーバイザーが提供する他のすべてのレベル 2 構成と区別します。

論理 CPU 特性 (Logical-CPU Characteristics) (LCPUC): ワード 8 のバイト 3 の内容は、このレベル 2 構成用として用意されている論理 CPU の特性を記述します。各ビットの意味は以下のとおりです。

ビット 意味

0 専用: ビット 0 が 1 のときは、このレベル 2 構成用の論理 CPU のうち 1 つまたは幾つか、このレベル 2 構成専用であるレベル 1 の CPU を使用して用意されており、これらの CPU は他のレベル 2 構成用の論理 CPU には使用されないことを意味します。専用のレベル 1 CPU を使用して用意される論理 CPU の数は、ワード 18 のバイト 0 および 1 にある専用 LCPU 数の値で指定されます。

ビット 0 が 0 のときは、このレベル 2 構成用の論理 CPU は、どれも、このレベル 2 構成専用のレベル 1 CPU を使用して用意されるものではないことを示します。

1 共用: ビット 1 が 1 のときは、このレベル 2 構成用の論理 CPU のうちの 1 つ以上が、他のレベル 2 構成用の論理 CPU にも使用できるレベル 1 CPU を使用して用意されることを示します。共用のレベル 1 CPU を使用して用意される論理 CPU の数は、ワード 18 のバイト 2 および 3 にある共用 LCPU 数の値で指定されます。

ビット 1 が 0 のときは、このレベル 2 構成用の論理 CPU は、どれも、共用のレベル 1 CPU を使用して用意されるものではないことを示します。

2 使用制限: ビット 2 が 1 のときは、このレベル 2 構成用の論理 CPU を用意するために使用できるレベル 1 CPU の数に、制限があることを示します。

ビット 2 が 0 のときは、このレベル 2 構成用の論理 CPU を用意するために使用できるレベル 1 CPU の数に、制限がないことを示します。

3-7 予約済み。

総論理 CPU 数 (Total Logical-CPU Count): ワード 9 のバイト 0 および 1 には、このレベル 2 構成用に用意されている論理 CPU の合計数を示す 16 ビットの符号なし 2 進整数が含まれます。この数には、構成済み状態、スタンバイ状態、または予約済み状態にあるすべての論理 CPU が含まれます。

構成済み論理 CPU 数 (Configured Logical-CPU Count): ワード 9 のバイト 2 および 3 には、このレベル 2 構成用の論理 CPU のうち、構成済み状態にある

CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU が構成済み状態になるのは、このレベル 2 構成内に含まれていて、かつプログラムの実行のために使用できるときです。

スタンバイ論理 CPU 数 (Standby Logical-CPU Count): ワード 10 のバイト 0 および 1 には、このレベル 2 構成用の論理 CPU のうち、スタンバイ状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU がスタンバイ状態になるのは、その論理 CPU がこのレベル 2 構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその論理 CPU を構成済み状態に変更できるときです。

予約済み論理 CPU 数 (Reserved Logical-CPU Count): ワード 10 のバイト 2 および 3 には、このレベル 2 構成用の論理 CPU のうち、予約済み状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU が予約済み状態になるのは、その論理 CPU がこのレベル 2 構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその CPU を構成済み状態にすることができないときです。(手動アクションによって、予約済み CPU をスタンバイ状態または構成済み状態にすることはできる場合があります。)

論理区画名 (Logical-Partition Name): ワード 11-12 には、このレベル 2 構成の名前を示す 8 文字の EBCDIC が含まれます。この名前は左寄せにされ、必要に応じて末尾ブランクが付加されます。

論理区画能力調整係数 (CAF) (Logical-Partition Capability Adjustment Factor): ワード 13 には、調整係数と呼ばれる 32 ビットの符号なし 2 進整数が含まれます。これは、1000 以下の値です。調整係数は、LPAR ハイパーバイザーがこのレベル 2 構成用に使用することのできる、基礎となるレベル 1 構成の能力を指定します。レベル 1 構成の能力を表す分数値は、CAF の値を 1000 で除算することにより求められます。

専用論理 CPU 数 (Dedicated Logical-CPU Count): ワード 18 のバイト 0 および 1 には、このレベル 2 構成用の構成済み状態の論理 CPU のうち、専用のレベル 1 CPU を使用して用意されている論理

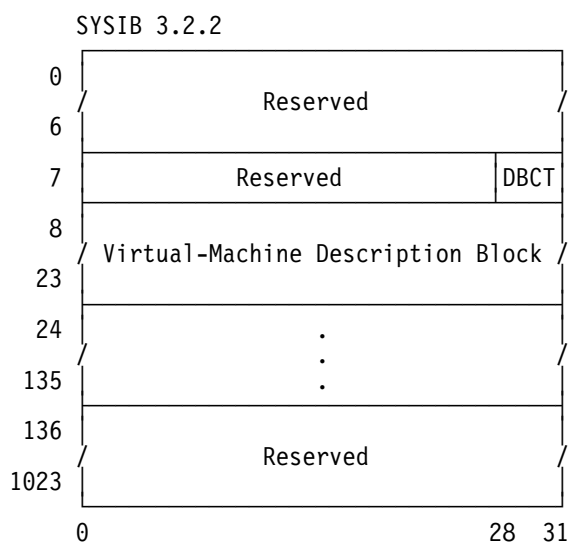
CPU の数を示す、16 ビットの符号なし 2 進整数が含まれます。(論理 CPU 特性フィールドのビット 0 の説明を参照。)

共用論理 CPU 数 (Shared Logical-CPU Count):

ワード 18 のバイト 2 および 3 には、このレベル 2 構成用の構成済み状態の論理 CPU のうち、共用のレベル 1 CPU を使用して用意されている論理 CPU の数を示す、16 ビットの符号なし 2 進整数が含まれます。(論理 CPU 特性フィールドのビット 1 の説明を参照。)

SYSIB 3.2.2 (仮想計算機 CPU)

SYSIB 3.2.2 のフォーマットは以下のとおりです。



予約済み (Reserved): ワード 0-6、ワード 7 のビット 0-27、およびワード 136-1023 は予約済みであり、0 が格納されています。

記述ブロック数 (DBCT) (Description-Block

Count): ワード 7 のビット 28-31 には、この SYSIB のワード 8 およびそれ以降に格納されている仮想計算機記述ブロックの数 (最大 8) を示す、4 ビットの符号なし 2 進整数が含まれます。

仮想計算機記述ブロック (Virtual-Machine

Description Blocks): ワード 8-135 には、ネストされているレベル 3 構成の数 (存在する場合) およびその

処理特性に応じて、1 個から 8 個の 64 バイトの仮想計算機記述ブロックが含まれます。

仮想計算機制御プログラムがレベル 3 構成を提供しており、その制御プログラムが、他の仮想計算機制御プログラムが提供するレベル 3 構成により実行されている場合、これらのレベル 3 構成は「ネストされた」構成と呼ばれます。レベル 3 構成は、幾つかのレベルにわたってネストすることができます。

レベル 3 構成により実行されるプログラムと基本マシンの間のパス内にあるネストされたレベル 3 構成の集合を、「レベル 3 構成スタック」と呼びます。1 つのスタック内のレベル 3 構成には、連続した番号が付けられません。レベル 2 構成またはレベル 1 構成により実行されている仮想計算機制御プログラムが提供するレベル 3 構成は、スタック内の最も小さい番号 (0) のレベル 3 構成です。この命令を含むプログラムを実行しているレベル 3 構成は、スタック内の最も大きい番号 (N) のレベル 3 構成です。

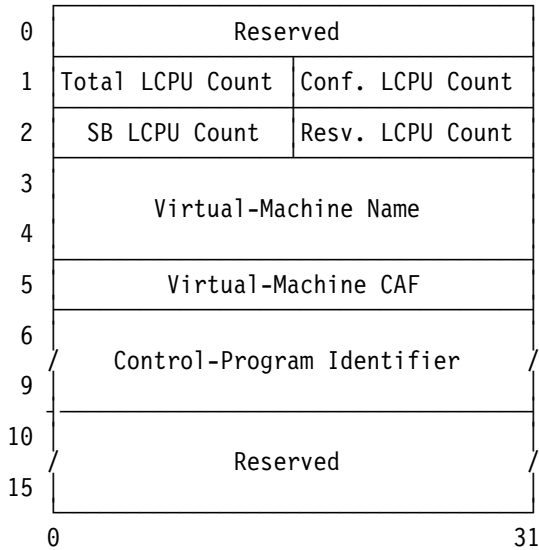
この SYSIB のワード 8-135 に複数の仮想計算機記述ブロックが格納される場合は、それらのブロックは以下の規則に従って格納されます。

- 記述されるレベル 3 構成の集合は、レベル 3 構成スタック内のレベル 3 構成の集合全体の中の連続したサブセットです。このサブセットには、必ず、スタック内の最も大きい番号のレベル 3 構成が含まれています。スタックの最下部にある幾つかのレベル 3 構成については、その中の最高位の構成により STORE SYSTEM INFORMATION がインプリメントされていない場合、または最大 8 個の記述ブロックという制限を超えた場合に、1 つまたは複数の構成が記述されないことがあります。
- レベル 3 構成スタック内の最大番号のレベル 3 構成は、必ず、SYSIB 内の最初の記述ブロックにより記述されます。記述されるサブセットに含まれるレベル 3 構成のうち、スタック内の最小番号の構成は、SYSIB 内の最後の記述ブロックにより記述されます。

SYSIB 内の仮想計算機記述ブロックの後からワード 136 の前までの内容は予約済みであり、0 が格納されません。

仮想計算機記述ブロックのフォーマットは、以下のとおりです。

仮想計算機記述ブロック



Reserved (予約済み): ワード 0 および 10-15 は予約済みであり、0 が格納されています。

Total Logical-CPU Count (総論理 CPU 数): ワード 1 のビット 0 および 1 には、このレベル 3 構成用に用意されている論理 CPU の合計数を示す 16 ビットの符号なし 2 進整数が含まれます。この数には、構成済み状態、スタンバイ状態、および予約済み状態にあるすべての論理 CPU が含まれます。

Configured Logical-CPU Count (構成済み論理 CPU 数): ワード 1 のバイト 2 および 3 には、このレベル 3 構成用の論理 CPU のうち、構成済み状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU が構成済み状態になるのは、このレベル 3 構成内に含まれていて、かつプログラムの実行のために使用できるときです。

Standby Logical-CPU Count (スタンバイ論理 CPU 数): ワード 2 のバイト 0 および 1 には、このレベル 3 構成用の論理 CPU のうち、スタンバイ状態に

ある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU がスタンバイ状態になるのは、その論理 CPU がこのレベル 3 構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその論理 CPU を構成済み状態に変更できるときです。

Reserved Logical-CPU Count (予約済み論理 CPU 数): ワード 2 のバイト 2 および 3 には、このレベル 3 構成用の論理 CPU のうち、予約済み状態にある CPU の数を示す 16 ビットの符号なし 2 進整数が含まれます。

論理 CPU が予約済み状態になるのは、その論理 CPU がこのレベル 3 構成内にあり、プログラムの実行に使用できる状態ではなく、かつ、命令を発行してその CPU を構成済み状態にすることができないときです。(手動アクションによって、論理 CPU をスタンバイ状態または構成済み状態にすることはできる場合があります。)

Virtual-Machine Name (仮想計算機名): ワード 3-4 には、このレベル 3 構成の名前を示す 8 文字の EBCDIC が含まれます。この名前は左寄せにされ、必要に応じて末尾ブランクが付加されます。

Virtual-Machine Capability Adjustment Factor (CAF)(仮想計算機能力調整係数): ワード 5 には、調整係数と呼ばれる 32 ビットの符号なし 2 進整数が含まれます。これは、1000 以下の値です。調整係数は、仮想計算機制御プログラムがこのレベル 3 構成用に使用することのできる、基礎となるレベル 1、レベル 2、またはレベル 3 構成の能力を指定します。基礎となる構成の能力を表す分数値は、CAF の値を 1000 で除算することにより求められます。

Control-Program Identifier (制御プログラム ID): ワード 6-9 には、このレベル 3 構成を提供する仮想計算機制御プログラムの 16 文字の EBCDIC ID が含まれます。この ID には、バージョン番号やリリース・レベルなどの修飾子が含まれることがあります。ID は左寄せにされ、必要に応じて末尾ブランクが付加されます。

特別な条件

汎用レジスター 0 のビット位置 32-35 にある機能コードが現レベル番号より大きい場合は、条件コードは 3 に設定されます。

汎用レジスター 0 のビット 36-55 および汎用レジスター 1 のビット 32-47 は、すべて 0 でなければなりません。さもないと、指定例外が認識されます。

機能コードが有効であり、0 以外である場合は、以下の特別な条件が適用されますが、適用される順序は予測不能です。

- 第 2 オペランドは、4K バイト境界に合わせて指定されていないければなりません。さもないと、指定例外が認識されます。
- 機能コード、セクター 1、およびセクター 2 の組み合わせが無効であるか、または、有効であって

も、要求された情報が使用可能でない場合は、条件コードは 3 に設定されます。

図10-31 に、例外と条件コードの認識の優先順位を示してあります。

結果の条件コード:

- 0 要求された構成レベル番号が汎用レジスター 0 に入れられたか、または要求された SYSIB 情報が格納されました。
- 1 --
- 2 --
- 3 要求された SYSIB 情報は使用できません。

プログラム例外:

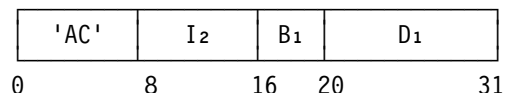
- アクセス例外 (取り出し: オペランド 2。機能コードが 0 以外の場合のみ)
- 特権命令例外
- 指定例外

- | | |
|-------|--|
| 1.-6. | 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。 |
| 7.A | 命令の第 2 ハーフワードについてのアクセス例外。 |
| 7.B.1 | システム情報格納機能がインストールされていない場合の操作例外。 |
| 7.B.2 | 特権命令についての特権命令例外。 |
| 8. | 機能コードが現レベル番号より大きいために起きる条件コード 3。 |
| 9. | 汎用レジスター 0 のビット 36-55 または汎用レジスター 1 のビット 32-47 が 0 ではないために起きる指定例外。 |
| 10. | 機能コードが 0 であるために起きる条件コード 0。 |
| 11.A | 第 2 オペランド・アドレスが 4K バイト境界を示していないために起きる指定例外。 |
| 11.B | 機能コード、セクター 1、およびセクター 2 の組み合わせが無効であるため、または要求された情報が使用可能でないために起きる条件コード 3。 |
| 12. | システム情報ブロックについてのアクセス例外 (格納)。 |
| 13. | システム情報ブロックに格納されている情報が原因で起きる条件コード 0。 |

図 10-31. 実行の優先順位: STORE SYSTEM INFORMATION

STORE THEN AND SYSTEM MASK

STNSM $D_1(B_1), I_2$ [SI]



現 PSW のビット 0-7 が、第 1 オペランド位置に格納されます。次に、現 PSW のビット位置 0-7 が、その内容と第 2 オペランドとの論理積で置き換えられます。

特別な条件

アドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

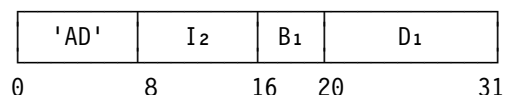
プログラム例外:

- アクセス例外 (格納: オペランド 1)
- 特権命令例外

プログラミング上の注意: プログラムは、STORE THEN AND SYSTEM MASK を使用することにより、後で復元できるように元の内容を残したままで、システム・マスク内の選択したビットを 0 に設定することができます。例えば、現在の状況についての記録を持たないプログラムが、幾つかの命令についてプログラム・イベント記録を使用不可にすることが必要になる場合があります。

STORE THEN OR SYSTEM MASK

STOSM $D_1(B_1), I_2$ [SI]



現 PSW のビット 0-7 が、第 1 オペランド位置に格納されます。次に、現 PSW のビット位置 0-7 が、その内容と第 2 オペランドとの論理和で置き換えられます。

特別な条件

PSW にロードする値については、ロードの前に妥当性検査は行われません。ただし、PSW のビット位置 0 および 2-4 の内容がすべて 0 でない場合は、ロードの直後に、指定例外が認識され、プログラム割り込みが起き

ます。この場合は、命令は完了し、命令長コードは 2 に設定されます。この指定例外は、この命令ではプログラム例外としてリストされますが、これについては、6-9 ページの『早期例外認識』で説明されています。この例外は、後続の命令の実行準備を行うプロセスで早期に発生するものと考えられます。

アドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

条件コード: コードは変更されません。

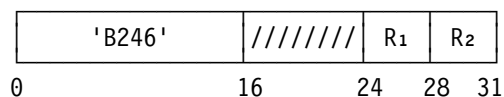
プログラム例外:

- アクセス例外 (格納: オペランド 1)
- 特権命令例外
- 指定例外

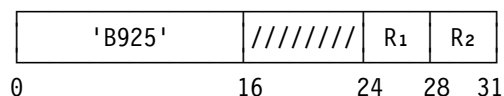
プログラミング上の注意: プログラムは、STORE THEN OR SYSTEM MASK を使用することにより、後で復元できるように元の内容を残したままで、システム・マスク内の選択したビットを 1 に設定することができます。例えば、プログラムは、外部マスク・ビットの現在の状況が分からないままで、入出力割り込みに対して CPU を使用可能にすることができます。

STORE USING REAL ADDRESS

STURA R_1, R_2 [RRE]



STURG R_1, R_2 [RRE]



STORE USING REAL ADDRESS (STURA) の場合は、汎用レジスタ R_1 のビット 32-63 が、汎用レジスタ R_2 の内容が指しているアドレスの実記憶位置にあるワードに格納されます。STORE USING REAL ADDRESS (STURG) の場合は、汎用レジスタ R_1 のビット 0-63 が、上記と同じ実記憶位置にあるダブルワードに格納されます。

24 ビット・アドレッシング・モードでは、汎用レジスタ R_2 のビット 40-63 が実記憶位置を指定し、ビット 0-39 は無視されます。31 ビット・アドレッシング・モードでは、汎用レジスタ R_2 のビット 33-63 が実

記憶位置を指定し、ビット 0-33 は無視されます。64 ビット・アドレッシング・モードでは、汎用レジスタ R2 のビット 0-63 が実記憶位置を指定します。

記憶域ワードまたは記憶域ダブルワードを指定するアドレスは実アドレスなので、動的アドレス変換は行われません。

特別な条件

汎用レジスタ R2 の内容は、STURA の場合はワード境界上の位置を、そして STURG の場合はダブルワード境界上の位置を指定していなければなりません。さもないと、指定例外が認識されます。

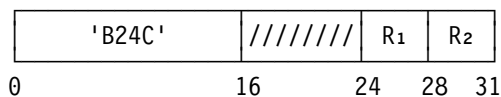
条件コード: コードは変更されません。

プログラム例外:

- アドレッシング例外 (汎用レジスタ R2 に指定されているアドレス)
- 特権命令例外
- 記憶保護例外 (格納: オペランド 2、キー制御保護および低アドレス保護)
- 指定例外

TEST ACCESS

TAR R1, R2 [RRE]



アクセス・レジスタ R1 内のアクセス・リスト・エントリー・トークン (ALET) について、アクセス・レジスタ変換 (ART) 時に認識される例外がないかどうかを検査されます。使用される拡張許可指標 (EAX) は、汎用レジスタ R2 のビット 32-47 です。また、ALET が、ディスパッチ可能単位アクセス・リストまたは 1 次スペース・アクセス・リストを指しているか、および、その値が 16 進数 00000000 または 00000001 であるかどうかも検査されます。

R1 が 0 のときは、ART では、通常使用される 16 進数 00000000 の代わりに、アクセス・レジスタ 0 の実際の内容が使用されます。

汎用レジスタ R2 のビット 0-31 および 48-63 は無視されます。

この操作は、どの変換モードの場合も同じです。つまり、PSW のビット 5、16、および 17 は無視されます。

R1 フィールドが指定する ALET が 16 進数 00000000 または 00000001 のいずれでもない場合は、その ALET には ART プロセスが適用されます。R2 フィールドが指定する EAX は有効 EAX と呼ばれ、この EAX が ART で使用されます。次の表に示す例外のいずれかを引き起こすような条件が存在するときは、条件コード 3 が設定されて、命令が完了します。

例外名	原因
ALET 指定例外	ALET のビット 0-6 がすべて 0 ではない。
ALEN 変換例外	アクセス・リスト・エントリー (ALE) がリストの外部にあるか無効 (ビット 0 が 1) である。
ALE シーケンス例外	ALET 内の ALE シーケンス番号 (ALESN) が ALE 内の ALESN に等しくない。
ASTE 妥当性例外	ASN 第 2 テーブル・エントリー (ASTE) が無効 (ビット 0 が 1)。
ASTE シーケンス例外	ALE 内の ASTE シーケンス番号 (ASTESN) が ALE 内の ASTESN に等しくない。
拡張権限例外	ALE 専用ビットが 0 ではなく、ALE 許可指標 (ALEAX) が有効 EAX に等しくなく、有効 EAX により選択された 2 次ビットが権限テーブルの外部にあるかまたは 0 である。

ART が完了したときに、上記のいずれの条件も存在していない場合は、命令は完了し、有効アクセス・リストがディスパッチ可能単位アクセス・リストであれば条件コード 1 が設定され、1 次スペース・アクセス・リストであれば条件コード 2 が設定されます。有効アクセス・リストは、ALET のビット 7 が 0 ならディスパッチ可能単位アクセス・リストであり、ビット 7 が 1 なら 1 次スペース・アクセス・リストです。ART については、有効なアクセス・リスト指定の取得方法も含めて、5-48 ページの『アクセス・レジスタ変換プロセス』で説明されています。

ALET が 16 進数 00000000 である場合は、条件コード 0 が設定されて、命令が完了します。ALET が 16 進数 00000001 である場合は、条件コード 3 が設定されて、命令が完了します。

特別な条件

有効なアクセス・リスト指定を取り出すために ART で使用されるアドレス、あるいは、ALE、ASTE、または権限テーブル・エントリーが、構成内で使用可能でない記憶位置を示しているときは、アドレッシング例外が認識されます。

何らかのアドレッシング例外が起きたときは、操作は抑止されます。

10-108ページの図10-32に、この命令に関するプログラム例外の認識の優先順位を示します。

結果の条件コード:

- 0 アクセス・リスト・エントリー・トークン (ALET) が 16 進数 00000000 です。
- 1 ALET はディスパッチ可能単位アクセス・リストを指しており、アクセス・レジスタ変換 (ART) で例外を引き起こしません。
- 2 ALET は 1 次スペース・アクセス・リストを指しており、ART で例外を引き起こしません。
- 3 ALET は 16 進数 00000001 であるか、または ART で例外を引き起こします。

プログラム例外:

- アドレッシング例外 (有効なアクセス・リスト指定、アクセス・リスト・エントリー、ASN 第 2 テーブル・エントリー、または権限テーブル・エントリー)

プログラミング上の注意:

1. TEST ACCESS を使用すると、呼び出し先プログラムは、呼び出し側プログラムの EAX を使用して、呼び出し側プログラムから渡された ALET が使用を許可されているかどうかを検査することができます。呼び出し側プログラムの EAX は、EXTRACT STACKED STATE を使用して最後のリンケージ・スタック状態エントリーから取得できます。この方法により、呼び出し先プログラムが、呼び出し側プログラムの EAX では許可されていない操作を、呼び出し先プログラム自身の EAX を使用して実行するのを防止することができます。

2. スペース切り替えを伴う PROGRAM CALL (PC-ss) が行うプログラム・リンケージの際に、16 進数 00000000 に等しい ALET が渡され、かつ、その ALET が、概念上、呼び出し側プログラムの 1 次アドレス・スペースおよび呼び出し先プログラムの 2 次アドレス・スペースを指している場合は、呼び出し先プログラムがその ALET を使用する前に、16 進数 00000001 に変更する必要があります。TEST ACCESS の条件コード 0 は、ALET が 16 進数 00000000 であることを示しているので、呼び出し先プログラムは ALET を 16 進数 00000001 に変更することができます。
3. 現 1 次への PROGRAM CALL (PC-cp) は、2 次アドレス・スペースを 1 次アドレス・スペースに等しく設定します。PC-ss は、2 次アドレス・スペースを、呼び出し側プログラムの 1 次アドレス・スペースに等しく設定します。ただし、スタッキング PC-ss は、使用されているエントリー・テーブル・エントリー内の 2 次 ASN 制御が 1 のときは、2 次アドレス・スペースを呼び出し先プログラムの 1 次・アドレス・スペースに等しく設定します。これらいずれの場合も、渡された ALET が 16 進数 00000001 であれば、それは概念上呼び出し側プログラムの 2 次アドレス・スペースを指しており、呼び出し先プログラムは、変換が行われた後であってもそのアドレス・スペースを使用できません (ただし、操作が PC-cp であり、かつ呼び出し側プログラムの PASN と SASN が等しい場合を除きます)。テストされた ALET が 16 進数 00000001 であるときに、TEST ACCESS が条件コード 3 を設定するのは、このためです。
4. PC-ss の実行後に、渡された ALET が、概念上、呼び出し側プログラムの 1 次スペース・アクセス・リスト内のエントリーを指している場合、呼び出し先プログラムはその ALET を使用できません。テストされた ALET が 1 次スペース・アクセス・リストを指しているときに、TEST ACCESS が条件コード 2 を設定するのは、このためです。
5. 制御プログラムは、ASTE 妥当性例外状況または ASTE シーケンス例外状況が訂正可能になるように、ASN 第 2 テーブル・エントリーを管理することができます。つまり、本来ならアクセス・レジスタ変換時にプログラム割り込みを起こすような状況が生じたときに、制御プログラムがその状況を訂正し、アクセス・レジスタ変換が正常に完了できるようにすることができます。そのためには、プログラムでは、TEST ACCESS を直接使用するのではなく、代わりに TEST ACCESS を使用する制御

- 1.-6. 一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7. 命令の第 2 ハーフワードについてのアクセス例外。
8. アクセス・リスト・エントリー・トークン (ALET) が 16 進数 00000000 があるために起きる条件コード 0。
9. ALET が 16 進数 00000001 であるか、ALET のビット 0-6 がすべて 0 ではないために起きる条件コード 3。
10. 有効なアクセス・リスト指定へのアクセスについてのアドレッシング例外。
11. アクセス・リスト・エントリー (ALE) がリストの外部にあるために起きる条件コード 3。
12. ALE へのアクセスについてのアドレッシング例外。
13. ALE が無効 (ビット 0 が 1) であるか、ALET 内のアクセス・リスト・エントリー・シーケンス番号 (ALESN) が ALE 内の ALESN に等しくないために起きる条件コード 3。
14. ASN 第 2 テーブル・エントリー (ASTE) へのアクセスについてのアドレッシング例外。
15. ASTE が無効 (ビット 0 が 1) であるか、ALE 内の ASTE シーケンス番号 (ASTESN) が ASTE 内の ASTESN に等しくないために起きる条件コード 3。
16. 権限テーブル・エントリーがテーブルの外部にあるために起きる条件コード 3。
17. 権限テーブル・エントリーへのアクセスについてのアドレッシング例外。
18. ALE 専用ビットが 0 でなく、ALE 許可指標 (ALEAX) が有効な拡張許可指標 (EAX) に等しくなく、有効 EAX により選択された 2 次ビットが 0 であるために起きる条件コード 3。
19. ALET ビット 7 が 0 の場合の条件コード 1。その他の場合は条件コード 2。

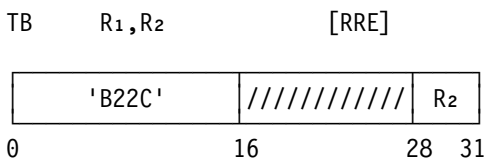
図 10-32. 実行の優先順位: TEST ACCESS

プログラム・サービスを利用して、条件コード 3 が設定されたときにその状況が訂正されるようにする必要があります。MVS/ESA には、TEST ACCESS を直接使用する代わりに使用できる TESTART マクロ命令が用意されています。

ロック・コードが発生する可能性の程度に基づいて行われます。

テストするブロックのアドレスは、汎用レジスター R2 の内容により指定されます。

TEST BLOCK



4K バイト・ブロックの記憶位置および記憶キーが使用可能かどうかテストされ、条件コードによりテスト結果が表示されます。使用可能性のテストは、無効な検査ブ

完全なテスト操作が必ず行われるのは、24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスター 0 のビット位置 32-63 の初期内容が 0 の場合のみであり、64 ビット・アドレッシング・モードでは、そのレジスターのビット位置 0-63 の初期内容が 0 の場合のみです。24 ビットまたは 31 ビットのアドレッシング・モードでは、この操作の完了時に、汎用レジスター 0 のビット位置 32-63 が 0 に設定され、ビット 0-31 は常に無視され、未変更のままとなります。64 ビット・アドレッシング・モードでは、操作の完了時に、レジスターのビット位置 0-63 の内容が 0 に設定されます。

ブロックが使用可能であると認められた場合は、そのブロックの 4K バイトが 0 にクリアされ、記憶キーの内容は予測不能であり、そして条件コード 0 が設定されず。ブロックが使用可能ではないと認められた場合は、データおよび記憶キーは、該当モデルで可能な限り、後でその領域に対する取り出しを行ってもマシン・チェック条件を起こさないような値に設定され、条件コード 1 が設定されます。

24 ビット・アドレッシング・モードでは、汎用レジスタ R₂ のビット 40-51 が実記憶域内の 4K バイト・ブロックを指定し、ビット 0-39 および 52-63 は無視されます。31 ビット・アドレッシング・モードでは、そのレジスタのビット 33-51 が目的のブロックを指定し、ビット 0-32 および 52-63 は無視されます。64 ビット・アドレッシング・モードでは、そのレジスタのビット 0-51 が目的のブロックを指定し、ビット 52-63 は無視されます。

ブロックのアドレスは実アドレスなので、第 2 オペランド・アドレスにより指定されているブロックへのアクセスには、キー制御保護、アクセス・リスト制御保護、およびページ保護は適用されません。低アドレス保護は適用されます。アドレッシング例外または記憶保護例外が起きたときは、操作は中止されます。中止された場合は、24 ビットまたは 31 ビットのアドレッシング・モードでは、条件コードおよび汎用レジスタ 0 のビット 32-63 の内容は予測不能であり、64 ビット・アドレッシング・モードでは、条件コードおよびそのレジスタのビット 0-63 は予測不能です。これらの例外が起きたときは、記憶域ブロックおよびそれに関連した記憶キーの内容は変更されません。

使用可能性テストの方法はモデルによって異なり、(1) ブロック内のデータおよび記憶キーを対象として、テスト・パターンの格納と読み取りを交互に行う方法、(2) 構成内のブロックの使用可能度を記録した内部レコードを参照する方法、および、(3) その両方のメカニズムを組み合わせた方法があります。

内部レコードを使用するモデルでは、それまでに固定的な障害が検出されているか、または、ブロック内の断続的障害がそのモデルで決められているしきい値を超えていると、ブロックは使用不可能と示されます。この種のモデルでは、基準に応じて、格納が行われる場合と行われない場合があります。したがって、ブロック 0 が使用可能でないときに、格納が行われない場合は、低アドレス保護は示される場合と示されない場合があります。

テスト・パターンを使用するモデルでは、TEST BLOCK は割り込み可能な場合があります。最後の操作

単位以外の操作単位の後で割り込みが起きたときは、条件コードは予測不能であり、24 ビットまたは 31 ビットのアドレッシング・モードでは、汎用レジスタ 0 のビット位置 32-63 には中間ステップの状態の記録が含まれることがあり、64 ビット・アドレッシング・モードでは、ビット 0-63 の内容にその記録が含まれることがあります。割り込みの後で実行が再開されるときは、条件コードは無視されますが、汎用レジスタ 0 の中の記録は、再開点を決定するために使用されることがあります。

If (1) TEST BLOCK を実行するときに、24 ビットまたは 31 ビットのアドレッシング・モードで汎用レジスタ 0 のビット位置 32-63 の初期値が 0 であるか、64 ビット・モードでビット位置 0-63 の初期値が 0 以外である場合、または、(2) 割り込みが生じ、汎用レジスタ 0 のビット位置 32-63 または 0-63 (該当アドレッシング・モードに対応) の値、あるいは記憶域ブロックまたはそれに関連した記憶キーの値が、割り込み発生時点での対応する値と異なっているときに、割り込まれた命令が再開された場合、または、(3) 命令の実行中に、ブロックまたはそれに関連した記憶キーがチャンネル・サブシステムまたは他の CPU によりアクセスされた場合は、記憶域ブロック、それに関連した記憶キー、および汎用レジスタ 0 のビット位置 32-63、および結果の条件コード設定は、予測不能です。

ブロック内で最初から無効検査ブロック・コード・エラーが見つかった場合、またはテストの時点でこの種のエラーが検出された場合、通常、マシン・チェック条件が生じることはありません。ブロック・テスト機能は、この命令の実行が原因でマシン・チェック割り込みが過剰に発生するのを防止する設計になっています。ただし、使用不可能なブロックを対象とした TEST BLOCK の実行中に、他の CPU (またはチャンネル・サブシステム) がそのブロックにアクセスした場合は、この CPU と他の CPU (またはチャンネル・サブシステム) の両方に対して、エラー条件が報告されることがあります。

ブロックがアクセスされる前と、操作の完了後 (または部分的な完了後) に、逐次化機能が実行されます。

10-110ページの図10-33 に、例外と条件コードの認識の優先順位を示してあります。

結果の条件コード:

- 0 ブロックは使用可能です。
- 1 ブロックは使用不可能です。
- 2 --
- 3 --

プログラム例外:

- アドレッシング例外 (取り出しおよび格納: オペランド 2)
- 特権命令例外
- 記憶保護例外 (格納: オペランド 2。低アドレス保護のみ)

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外。
7.B	特権命令例外。
8.	ブロックが構成内で使用可能でないために起きるアドレッシング例外。*
9.A	条件コード 1。ブロックが使用可能ではありません。
9.B	低アドレス保護が原因で起きる記憶保護例外。*
10.	条件コード 0。ブロックは使用可能で、0 に設定されます。
説明:	
* アドレッシング例外または記憶保護例外のときは、操作は中止され、条件コードは予測不能です。	

図 10-33. 実行の優先順位: TEST BLOCK

プログラミング上の注意:

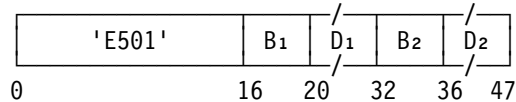
1. ほとんどのモデルでは、TEST BLOCK の実行には、埋め込みを伴う MOVE LONG 命令の場合よりはるかに時間がかかります。したがって、通常の場合の記憶域消去には、この命令は使用しないでください。
2. プログラムでは、初期プログラム・ローディングのとき、および記憶機構オンライン変更プロシージャの一環として、TEST BLOCK を使用して、使用すべきではないブロックが存在していないかどうかを判別してください。
3. ブロックのデータまたは記憶キーについて、訂正されていないエラーが報告されたときは、プログラムで TEST BLOCK を使用してください。TEST BLOCK ブロックを実行すると、以後の同じブロックの事前取り出しまたは意図しない参照が原因でマシン・チェック条件が生じることがないように、該当モデルで可能な限り、ブロックの内容が維持されます。この場合、プログラムは、結果の条件コードを使用して、そのブロックの再利用が可能かどうか

を判別できます。(例えば、エラーが外部で生成されたエラーまたは間接記憶域エラーである場合は、ブロックは使用可能として示されることがあります。)間接記憶域エラー指示が報告されたかどうかに関係なく、この手順に従ってください。

4. TEST BLOCK を実行したときに、モデルによって、ブロックのエラーを除去できる場合とできない場合があります。したがって、プログラムでは、使用不可能なブロックを参照するのを避けるために、妥当なすべての予防措置を講じる必要があります。例えば、使用不可能なブロックのページ・フレーム実アドレスを、連結された有効なページ・テーブル・エントリに入れないようにしてください。
5. 一部のモデルでは、プログラムが参照していないブロックについても、マシン・チェックが報告されることがあります。プログラムにより使用不可能のマークが付けられているブロック内の記憶キー・エラーについてマシン・チェックが報告されたときは、TEST BLOCK を使用するより、SET STORAGE KEY EXTENDED を使用して記憶キーの妥当性を検査する方が、効率的な場合があります。
6. TEST BLOCK の記憶域オペランド参照は、複数アクセス参照になることがあります。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

TEST PROTECTION

TPROT D₁(B₁), D₂(B₂) [SSE]



第 2 オペランド・アドレスのビット 56-59 に指定されているアクセス・キーを使用して、第 1 オペランド・アドレスが指す記憶機構について記憶保護例外の有無がテストされます。

第 2 オペランド・アドレスは、データのアドレッシングには使用されません。代わりに、このアドレスのビット 56-59 がテストに使用するアドレス・キーを形成します。第 2 オペランド・アドレスのビット 0-55 および 60-63 は無視されます。

第 1 オペランド・アドレスは、論理アドレスです。CPU がアクセス・レジスター・モードにあるとき(つまり、DAT がオンで、PSW のビット 16 および 17 が 2 進数 01 であるとき)は、第 1 オペランド・アドレスは、アクセス・レジスター変換 (ART) プロセスと動的

アドレス変換 (DAT) プロセスの両方により変換されま
す。ART は、B1 フィールドに指定されているアクセ
ス・レジスターに適用され、DAT で使用するアドレ
ス・スペース制御エレメントが取得されます。DAT が
オンであり、CPU がアクセス・レジスター・モードで
ないときは、第 1 オペランド・アドレスは DAT により
変換されます。この場合、DAT は、CPU が 1 次スペ
ース・モードのときは制御レジスター 1、2 次スペ
ース・モードのときは制御レジスター 7、そしてホーム・
スペース・モードのときは 制御レジスター 13 に含まれ
ているアドレス・スペース制御エレメントを使用しま
す。DAT がオフのときは、第 1 オペランド・アドレス
は実アドレスであり、ART または DAT のいずれの変
換も受けません。

CPU がアクセス・レジスター・モードであるときに、
次の表に示す例外のいずれかを引き起こすような条件が
原因で、ART によりアドレス・スペース制御エレメン
トを取得できない場合は、条件コード 3 が設定されて、
命令は完了します。

例外名	原因
ALET 指定例外	アクセス・リスト・エントリー・トークン (ALET) のビット 0-6 がすべて 0 ではない。
ALEN 変換例外	アクセス・リスト・エントリー (ALE) がリストの外部にあるか無効 (ビット 0 が 1) である。
ALE シーケンス例外	ALET 内の ALE シーケンス番号 (ALESN) が ALE 内の ALESN に等しくない。
ASTE 妥当性例外	ASN 第 2 テーブル・エントリー (ASTE) が無効 (ビット 0 が 1)。
ASTE シーケンス例外	ALE 内の ASTE シーケンス番号 (ASTESN) が ASTE 内の ASTESN に等しくない。
拡張権限例外	ALE 専用ビットが 0 ではなく、ALE 許可指標 (ALEAX) が拡張許可指標 (EAX) に等しくなく、EAX により選択された 2 次ビットが権限テーブルの外部にあるかまたは 0 である。

アクセス・レジスターに 16 進数 00000000 または
00000001 が含まれているときは、ART では、アクセ
ス・リストにアクセスせずに、それぞれ、制御レジス
ター 1 または 7 からアドレス・スペース制御エレメン
トが取得されます。B1 フィールドがアクセス・レジス
ター 0 を指定している場合は、ART では、このアクセ
ス・レジスターは 16 進数 00000000 を含むものと見な
され、アクセス・レジスターの実際の内容は検査されま
せん。

ART が正常に完了したときは、DAT の実行を介して操
作は続行されます。

DAT がオンのときに、次の表に示す例外のいずれかを
引き起こすような条件が原因で、第 1 オペランド・アド
レスを変換できない場合は、条件コード 3 が設定され
て、命令は完了します。

例外名	原因
ASCE タイプ例外	使用されているアドレス・スペース制御エレメント (ASCE) が領域第 2 テーブル指定であり、第 1 オペランド・アドレスのビット 0-10 がすべて 0 ではない。または、ASCE が領域第 3 テーブル指定であり、第 1 オペランド・アドレスのビット 0-21 がすべて 0 ではない。または、ASCE がセグメント・テーブル指定であり、仮想アドレスのビット 0-32 がすべて 0 ではない。
領域第 1 変換例外	領域第 1 テーブル・エントリーがテーブルの外部にあるか、または無効である。
領域第 2 変換例外	領域第 2 テーブル・エントリーがテーブルの外部にあるか、または無効である。
領域第 3 変換例外	領域第 3 テーブル・エントリーがテーブルの外部にあるか、または無効である。
セグメント変換例外	セグメント・テーブル・エントリーがテーブルの外部にあるか、または無効である。
ページ変換例外	ページ・テーブル・エントリーが無効である。

第 1 オペランド・アドレスの変換が完了可能な場合、または DAT がオフの場合は、第 1 オペランド・アドレスに指定されているブロックの記憶キーが、第 2 オペランド・アドレスのビット 56-59 に指定されているアクセスと比較して検査され、適用可能なすべての保護メカニズムを考慮に入れて、格納アクセスおよび取り出しアクセスが許可されるかどうかを示す条件コードが設定されます。したがって、低アドレス保護がアクティブで、第 1 オペランドの有効アドレスが 0-511 または 4096-4607 の範囲内にあるときは、格納アクセスは許可されません。アクセス・リスト制御保護、ページ保護、記憶保護オーバーライド、および取り出し保護オーバーライドも、考慮に入れられます。

変更ビットも含めて、記憶域の内容は変更されません。モデルによっては、目的の記憶位置が取り出し保護の対象となっている場合でも、第 1 オペランドに対する参照ビットが 1 に設定されることがあります。

特別な条件

CPU がアクセス・レジスター・モードのときに、有効なアクセス・リスト指定を取り出すために ART で使用されるアドレス、あるいは、ALE、ASTE、または権限テーブル・エントリが、構成内で使用可能でない記憶位置を示している場合は、アドレッシング例外が認識されます。

DAT がオンのときに、領域テーブル・エントリ、セグメント・テーブル・エントリ、ページ・テーブル・エントリ、または、変換後のオペランドの実アドレスが、構成内で使用可能でない記憶位置を指している場合は、アドレッシング例外が認識されます。さらに、領域テーブル・エントリ、セグメント・テーブル・エントリ、またはページ・テーブル・エントリにフォーマット・エラーがある場合、つまり、6-34ページの『変換指定例外』に示す理由のいずれかに該当する場合は、変換指定例外が認識されます。DAT がオフのときは、オペランドの実アドレスが原因で起きるアドレッシング例外のみが適用されます。

上記いずれの場合も、操作は抑止されます。

結果の条件コード:

- 0 取り出しと格納が許可されます。
- 1 取り出しは許可され、格納は許可されません。
- 2 取り出しは許可されず、格納は許可されません。
- 3 変換は使用可能ではありません。

プログラム例外:

- アドレッシング例外 (有効アクセス・リスト指定、アクセス・リスト・エントリ、ASN 第 2 テーブル・エントリ、権限テーブル・エントリ、領域テーブル・エントリ、セグメント・テーブル・エントリ、ページ・テーブル・エントリ、またはオペランド 1)
- 特権命令例外
- 変換指定例外

プログラミング上の注意:

1. プログラムでは、TEST PROTECTION を使用することにより、プログラム例外を起こすことなく、呼び出し側プログラムから渡されたアドレスの妥当性を検査することができます。この命令は、第 1 オペランド・アドレスに指定されている記憶位置での取り出しまたは格納が許可されるかどうかを示す条件コードを設定します。この命令では、マシン内のすべての保護メカニズム (アクセス・リスト制御保護、ページ保護、キー制御保護、低アドレス保護、記憶保護オーバーライド、および取り出し保護オーバーライド) が考慮に入れられます。さらに、ASCE タイプ例外条件、領域変換例外条件、セグメント変換例外条件、およびページ変換例外条件は、プログラムで保護違反を取り扱う場合の代替手段としての役割を果たすものなので、これらの条件は、プログラム例外の発生を回避し、条件コードを設定するために使用されます。

CPU がアクセス・レジスター・モードのときは、TEST PROTECTION により、プログラムは、プログラム例外を引き起こすことなく、アクセス・レジスター内のアクセス・リスト・エントリ・トークン (ALET) の使用可能性を検査することができます。ALET について、その妥当性 (ALET 指定例外条件、ALEN 変換例外条件、ALE シーケンス例外条件)、および、プログラムによる使用が許可されているかどうか (ASTE 妥当性例外条件、ASTE シーケンス例外条件、および拡張権限例外条件がないこと) が検査されます。

ASCE タイプ例外条件がある場合は、条件コードも設定されます。

2. 呼び出し側プログラムから渡されたアドレスの妥当性を検査する方法の詳細について、またその代替方法については、『SET PSW KEY FROM ADDRESS』の項の『プログラミング上の注意』を参照してください。TEST PROTECTION を用いたテスト方法には、割り込みが生じないという利点があります。しかし、テストと実際の使用との間に時間的な差があるため、テストから使用までの間に間

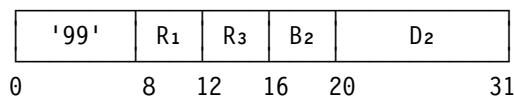
題の記憶位置の記憶キーが変化する可能性がある場合は、テストの結果が正確ではない場合があります。

3. 動的アドレス変換の処理については、TEST PROTECTION は、ASCE タイプ例外、領域変換例外、セグメント変換例外、およびページ変換例外を引き起こさないという点で、LOAD REAL ADDRESS に似ています。代わりに、これらの例外条件は条件コード設定により示されます。同様に、この 2 つのいずれかの命令の中でアクセス・レジスタ変換が行われた場合、一定の例外条件については条件コードが設定されます。LOAD REAL ADDRESS で条件コード 1、2、および 3 が設定される条件については、TEST PROTECTION では条件コード 3 が設定されます。この 2 つの命令には、その他幾つかの相違点があります。TEST PROTECTION の第 1 オペランドは論理アドレスであり、したがって、DAT がオフのときは動的アドレス変換の対象となりません。LOAD REAL ADDRESS の第 2 オペランドは仮想アドレスであり、これは常に変換されます。

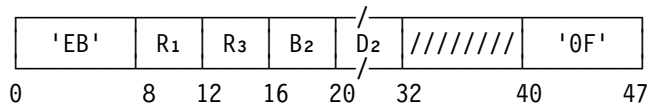
TEST PROTECTION にアクセス・レジスタ変換が適用されるのは、CPU がアクセス・レジスタ・モードにあるとき (DAT がオンするとき) だけですが、LOAD REAL ADDRESS の場合は、DAT がオンかオフかに関係なく、PSW のビット 16 および 17 が 2 進数 01 のときは、アクセス・レジスタ変換が適用されます。アクセス・レジスタ変換時の例外条件が原因で条件コード 3 が設定される場合は、LOAD REAL ADDRESS は、その例外に割り当てられているプログラム割り込みコードを汎用レジスタに戻しますが、TEST PROTECTION はそれをしません。

TRACE

TRACE R₁,R₃,D₂(B₂) [RS]



TRACG R₁,R₃,D₂(B₂) [RSE]



明示トレースがオンのとき (制御レジスタ 12 のビット 63 が 1 のとき) は、第 2 オペランド (記憶域内の

32 ビット・ワード) が取り出され、そのワードのビット 0 が検査されます。第 2 オペランドのビット 0 が 0 の場合は、制御レジスタ 12 に指定されている実記憶位置にトレース・エントリーが形成されます。

明示トレースがオフ (制御レジスタ 12 のビット 63 が 0) の場合、または第 2 オペランドのビット 0 が 1 の場合は、トレース・エントリーは形成されず、トレース例外も認識されません。

トレース・エントリーを構成する要素は、エントリー・タイプ ID、エントリー内に内容の一部または全部が取り込まれている汎用レジスタの数、エントリーが TRACE (TRACE) または TRACE (TRACG) のどちらにより形成されたかを示すフィールド、TOD クロックの選択されたビット、第 2 オペランド、そして、一連の汎用レジスタの内容の一部または全部です。TRACE (TRACE) の場合は、TOD クロックのビット 16-63 および汎用レジスタのビット 32-63 が、トレース・エントリーに入れられます。TRACE (TRACG) の場合は、クロックのビット 0-79 およびレジスタのビット 0-63 が、エントリーに入れられます。

これらの汎用レジスタは、汎用レジスタ R₁ から始まり汎用レジスタ R₃ まで (このレジスタも含む)、レジスタ番号の昇順に従って格納されます。汎用レジスタ 0 は、汎用レジスタ 15 の後に続きます。トレース・テーブルおよびトレース・エントリー・フォーマットについては、4-10 ページの『トレース』で説明されています。

トレース・エントリーが作成される時、操作の開始前と完了後に、逐次化およびチェックポイント同期化機能が実行されます。

特別な条件

問題プログラム状態では、明示トレースがオフまたは第 2 オペランドのビット 0 が 1 であっても、特権命令例外が認識されます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。明示トレースがオフのときは、指定例外が認識されるかどうかは予測不能です。

明示トレースがオフのときに、第 2 オペランドについてアクセス例外が認識されるかどうかは予測不能です。

条件コード: コードは変更されません。

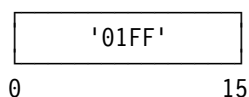
プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- 特権命令例外
- 指定例外
- トレース例外

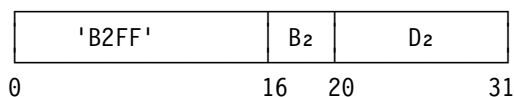
プログラミング上の注意: 第 2 オペランドのビット 1-15 は、モデル依存の機能のために予約されているので、0 に設定する必要があります。

TRAP

TRAP2 [E]



TRAP4 D₂(B₂) [S]



トラップ操作が行われるのは、CPU が 1 次スペース・モードまたはアクセス・レジスター・モードにあり、ディスパッチ可能単位制御テーブル (DUCT) のバイト 47 の TRAP 使用可能ビットが 1 である場合です。そうでない場合は、特殊操作例外が認識されます。

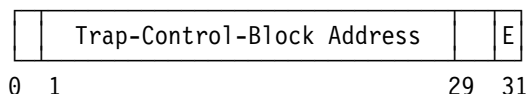
トラップ操作では、DUCT からトラップ制御ブロック・アドレスが取得され、次にトラップ制御ブロックからトラップ保管域アドレスおよびトラップ・プログラム・アドレスが取得されます。トラップ保管域には、状態情報が格納されます。次に、トラップ制御ブロック・アドレスが汎用レジスター 15 にロードされます。最後に、現 PSW の基本アドレッシング・モード・ビットが 1 に設定され (その結果、アドレッシング・モードは、31 ビット・モードまたは 64 ビット・モードのときはそのまま変更されず、24 ビット・モードは 31 ビット・モードに変更されます)、アドレス・スペース制御ビットは 0 (1 次スペース・モード) に設定され、そして、命令アドレスがトラップ・プログラム・アドレスで置き換えられます。オプションとして、ESA/390 での TRAP 操作との互換性が用意されています。

TRAP4 では、第 2 オペランド・アドレスはデータのアドレスを指定するために使用されるものではありません。代わりに、そのアドレスのビット 33-63 がトラップ保管域に格納されます。

ディスパッチ可能単位制御テーブル

この命令では、ディスパッチ可能単位制御テーブル (DUCT) のバイト 44-47 (ワード 10) が使用されます。これらのバイトの内容は以下のとおりです。

DUCT バイト 44-47



DUCT のバイト 44-47 のフィールドは以下のように割り振られています。

トラップ制御ブロック・アドレス (Trap-Control-Block Address):

ビット 1-28 の右側に 3 個の 0 を付加したものが、トラップ制御ブロックの 31 ビット・ホーム仮想アドレスを形成します。このアドレスは、現アドレッシング・モードにも、アドレス・スペース制御ビットの現在値にも関係なく、31 ビットのホーム仮想アドレスとして扱われます。汎用レジスター 15 の内容がトラップ保管域に保管された後で、このアドレスの左側に 1 個の 0 を付加したものが、そのレジスターのビット位置 32-63 に入れます。現アドレッシング・モードが 64 ビット・モードである場合は、汎用レジスター 15 のビット 0-31 は 0 に設定されます。

TRAP 使用可能ビット (E): ビット 31 が 1 のときは、トラップ操作が行われることを示します。ビット 31 が 0 の場合は、TRAP では特殊操作例外が認識されません。

バイト 44-47 のビット 0、29、および 30 は無視されますが、将来の機能拡張に備えて 0 にしておく必要があります。

トラップ制御ブロック

トラップ制御ブロックは、ダブルワード境界に位置合わせされた 64 バイトです。トラップ制御ブロックのフォーマットは、以下のとおりです。

16 進数 10 進数

0	0		P	R	
4	4				
8	8				
C	12	Trap-Save-Area Address			
10	16				
14	20	Trap-Program Address			
18	24	////////////////////////////////////			
1C	28	////////////////////////////////////			
20	32				
3C	60				

(P はビット13)

トラップ制御ブロックのフィールドは、以下のように割り振られています。

PSW 制御 (P): バイト 0 および 1 のビット13 は、現 PSW のビット 31 がとることのできる値と、現 PSW のビット 12 および 33-127 がトラップ保管域の PSW 値フィールドにどのように保管されるかを制御します。ビット 13 が 0 のときは、次のようになります。

- 現 PSW のビット 31 (拡張アドレッシング・モード・ビット) は 0 でなければなりません。さもないと、特殊操作例外が認識されます。
- 現 PSW のビット 12 が 0 であっても、PSW 値フィールドのビット位置 12 には 1 が格納されます。
- 現 PSW のビット 97-127 が PSW 値フィールドのビット位置 33-63 に格納され、現 PSW のビット 33-96 は格納されず、PSW 値フィールドのビット位置 64-127 には 0 が格納されます。

ビット 13 が 1 のときは、次のようになります。

- 現 PSW のビット 31 は、0 または 1 のいずれでも構いません。
- 現 PSW のビット 12 が、PSW 値フィールドのビット位置 12 に格納されます。

- 現 PSW のビット 64-127 が、PSW 値フィールドのビット位置 64-127 に格納されます。

汎用レジスター制御 (R): バイト 0 および 1 のビット 14 は、汎用レジスターの内容を、トラップ保管域内の汎用レジスター 0-15 フィールドにどのように格納するかを制御します。R が 0 のときは、汎用レジスターのビット 32-63 が、汎用レジスター 0-15 フィールドの先頭から始まる連続した 4 バイトの記憶位置に格納され、レジスターのビット 0-31 は格納されず、汎用レジスター 0-15 フィールドの最後の 64 バイトは変更されません。R が 1 のときは、汎用レジスターのビット 0-63 が、汎用レジスター 0-15 フィールド内の連続した 8 バイトの記憶位置に格納されます。

トラップ保管域アドレス (Trap-Save-Area Address)

Address): バイト 12-15 のビット 1-28 の右側に 3 個の 0 を付加したものが、トラップ保管域の 31 ビットのホーム仮想アドレスを形成します。このアドレスは、現アドレッシング・モードにも、アドレス・スペース制御ビットの現在値にも関係なく、31 ビットのホーム仮想アドレスとして扱われます。バイト 12-15 のビット 0 および 29-31 は無視されます。

トラップ・プログラム・アドレス (Trap-Program Address)

Address): バイト 20-23 のビット 1-31 は、トラップ・プログラムの 31 ビットの 1 次仮想アドレスを形成します。このアドレスは、現アドレッシング・モードに関係なく、31 ビットの 1 次仮想アドレスとして扱われます。

トラップ制御ブロックのバイト 0-3 のビット位置 0-12 と 15-31、およびバイト 4-11、16-19、および 32-63 は予約済みであり、すべて 0 にしておく必要があります。バイト 24-31 はプログラミング用に使用できます。

トラップ保管域

トラップ保管域は、ダブルワード境界に位置合わせされた 256 バイトです。

トラップ操作では、トラップ保管域には次のように情報が格納されます。

16 進数
10 進数

0	0	Trap Flags
4	4	Reserved (Zeros Stored)
8	8	Bits 33-63 of Second-Op Address of TRAP4
C	12	Access Register 15
10 14 18 1C	16 20 24 28	PSW Values
20 24	32 36	General Registers 0-15
98 9C	152 156	
A0 A4	160 164	
A8 AC	168 172	Reserved (Unchanged)
F8 FC	248 252	

トラップ保管域のフィールドは、次のように割り振られています。

トラップ・フラグ (Trap Flags): バイト位置 0-3 には、トラップ操作の原因となった命令を識別する情報が格納されます。バイト 0-3 の詳細なフォーマットは以下のとおりです。

フラグ・ビット	意味
0	TRAP は EXECUTE のターゲットでした。
1	TRAP は TRAP4 です (TRAP2 ではない)。
2-12	予約済み。0 が格納されます。
13-14	命令長コード (ILC)。
15-31	予約済み。0 が格納されます。

TRAP が EXECUTE 命令のターゲットであった場合は、バイト 0-3 のビット 0 が 1 に設定されます。

TRAP が TRAP4 である (TRAP2 ではない) 場合は、バイト 0-3 のビット 1 が 1 に設定されます。

ビット 13 および 14 は命令長コード (ILC) で、これは、TRAP 命令の長さ、または、TRAP が EXECUTE

のターゲットであった場合は EXECUTE の長さを指定します。

ビット 2-12 および 15-31 予約済みであり、0 が格納されます。

TRAP4 の第 2 オペランド・アドレスのビット 33-63 (Bits 33-63 of Second-Operand Address of TRAP4):

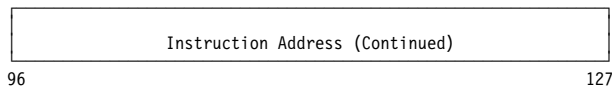
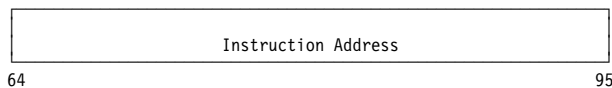
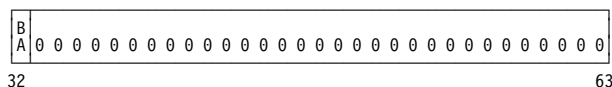
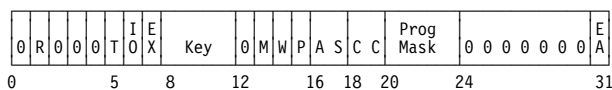
TRAP4 の場合は、第 2 オペランド・アドレスのビット 33-63 は現アドレッシング・モードの制御下で生成され、これらのビットの左側に 1 個の 0 を付加したものが、バイト位置 8-11 に格納されます。現アドレッシング・モードが 64 ビット・モードであっても、第 2 オペランド・アドレスのビット 33-63 のみが格納されます。TRAP2 の場合は、バイト位置 8-11 にはすべて 0 が格納されます。

アクセス・レジスター 15 (Access Register 15):

バイト位置 12-15 には、アクセス・レジスター 15 の内容が格納されます。

PSW 値 (PSW Values): 以下の説明が適用されるのは、PSW 制御 (トラップ制御ブロックのバイト 0 および 1 のビット 13) が 1 である場合です。

バイト位置 16-31 には、現 PSW 内の一定の情報が格納されます。PSW のフォーマットは以下のとおりです。



バイト 16-31 のビット 0-127 は、PSW のビット 0-127 に 1 対 1 で対応します。バイト 16-31 の一部のビット位置については、それぞれ対応する PSW ビットが格納されます。バイト 16-31 のその他のビット位置については、格納される値は予測不能です。バイト 16-31 には、以下のように情報が格納されます。

ビット	値
0	0

ビット	値
1	予測不能
2-4	0
5-11	予測不能
12	0
13	予測不能
14	待ち状態 (W)
15	問題プログラム状態 (P)
16-17	アドレス・スペース制御 (AS)
18-19	条件コード (CC)
20-23	プログラム・マスク
24-30	0
31	拡張アドレッシング・モード (EA)
32	基本アドレッシング・モード (BA)
33-63	0
64-127	命令アドレス

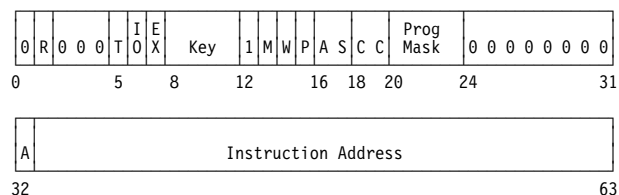
要約すれば、ビット 0、2-4、12、24-30、および 33-63 は 0、ビット 1、5-11、および 13 は予測不能、そしてその他のビットは PSW からの可変情報に設定されます。

待ち状態、問題プログラム状態、アドレス・スペース制御、条件コード、プログラム・マスク、拡張アドレッシング・モード、および基本アドレッシング・モードの値は、TRAP 命令が実行される前の CPU の状態を示します。命令アドレスの値は更新後の命令アドレスであり、これは、TRAP の次の命令のアドレスか、または、TRAP が EXECUTE のターゲットであった場合は EXECUTE の次の命令のアドレスです。

トラップ制御ブロック内の PSW 制御が 0 のときは、以下の点を除き、上記で述べたのと同じ操作が行われます。

- 現 PSW のビット 31 は 0 でなければなりません。さもないと、特殊操作例外が認識されます。
- バイト 16-31 のビット位置 12 には 1 が格納されます。
- 現 PSW のビット 97-127 がバイト 16-31 のビット位置 33-63 に格納され、現 PSW のビット 33-96 は格納されず、バイト 16-31 のビット位置 64-127 (バイト 24-31) には 0 が格納されます。

この場合は、バイト 16-23 は、次のような ESA/390 PSW のフォーマットになります。



汎用レジスター (General Registers 0-15): バイト位置 32-159 には、10-115ページの『汎用レジスター制御 (R)』で説明したように、汎用レジスター 0-15 の内容が格納されます。汎用レジスターのビット 32-63 または 0-63 が格納されるときは、レジスター 0 から始まりレジスター 15 に至るまで、レジスター番号の昇順に従って格納されます。

バイト 160-255 は常に無変更のままとなります。バイト 168-255 は予約済みです。バイト 160-167 はプログラミング用に使用できます。

特別な条件

CPU は、1 次スペース・モードまたはアクセス・レジスター・モードでなければならず、ディスパッチ可能単位制御テーブルのバイト 44-47 のビット 31 は 1 でなければなりません。さもないと、特殊操作例外が認識されます。また、PSW 制御 (トラップ制御ブロックのバイト 1 のビット 6) が 0 で、現 PSW のビット 31 (拡張アドレッシング・モード・ビット) が 1 である場合も、特殊操作例外が認識されます。

トラップ制御ブロックおよびトラップ保管域へのアクセスには、すべての保護メカニズムが通常通りに適用されます。

トラップ制御ブロック内のトラップ・プログラム・アドレスは、PSW 内の命令アドレスと置き換わる前には、テストされません。このアドレスが奇数の場合は、次の命令の実行時に指定例外が認識されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、操作は抑止されます。

10-118ページの図10-34 に、この命令に関するプログラム例外の認識の優先順位を示します。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: トラップ制御ブロック。格納: トラップ保管域)

- アドレッシング例外 (ディスパッチ可能単位制御テーブル)
- 特殊操作例外
- トレース例外

プログラミング上の注意:

1. TRAP 命令は、アプリケーション・プログラム内の命令をオーバーレイして、トラップ・プログラムに制御を渡すことを目的とするものです。トラップ・プログラムは、例えば、アプリケーション・プログラムが使用するデータの修正 (「2000 年」問題の原因となる日付の修正など) を行います。TRAP2 は 2 バイトの命令をオーバーレイでき、TRAP4 は、4 バイトの命令、または 6 バイトの命令の最初の 4 バイトをオーバーレイすることができます。トラップ・プログラムは、オーバーレイされた命令をシミュレートし、必要に応じて修正を行い、そしてアプリケーション・プログラムに制御を戻します。
2. トラップ・プログラムは、RESUME PROGRAM 命令を使用して、アプリケーション・プログラムに制御を戻すことができます。例えば、トラップ・プログラムは、アクセス・レジスター 15 と汎用レジスター 15 を除くすべてのレジスターの内容を復元した後、それらのレジスター (または少なくとも汎用レジスター) を使用してトラップ保管域をアドレ

ッシングすることにより、トラップ保管域から、これらのレジスターの内容および PSW のフィールドを復元することができます。RESUME

PROGRAM は、パラメーター・リスト内の制御ビットを使用することにより、ESA/390 PSW フォーマットまたは z/Architecture PSW フォーマットのいずれのフィールドからでも PSW のフィールドを復元し、汎用レジスターのビット 32-63 または 0-63 を復元することができます。

3. トラップ制御ブロックとトラップ保管域はホーム・アドレス・スペース内にあり、トラップ・プログラムは 1 次アドレス・スペース内にあります。トラップ・プログラムにとって、TRAP が汎用レジスター 15 に入れるトラップ制御ブロック・アドレスが役立つのは、(1) 1 次アドレス・スペースとホーム・アドレス・スペースが同じアドレス・スペースであるか、(2) 1 次アドレス・スペース内において、トラップ制御ブロックおよびトラップ保管域がホーム・アドレス・スペースの中と同じ位置にあるか、または、(3) トラップ・プログラムが、アクセス・レジスターを使用してホーム・アドレス・スペースにアクセスできる場合です。
4. TRAP の記憶域オペランド参照は、複数アクセス参照となることもあります。(5-86ページの『記憶域オペランドの一貫性』を参照。)

1.-6.	一般的な場合のプログラム割り込み条件と同じ優先順位を持つ例外。
7.A	命令の第 2 ハーフワードについてのアクセス例外 (TRAP4 の場合のみ)。
7.B	CPU が 1 次スペース・モードまたはアクセス・レジスター・モードになっていないために起きる特殊操作例外。
7.C.1	ディスパッチ可能単位制御テーブルへのアクセスについてのアドレッシング例外。
7.C.2	ディスパッチ可能単位制御テーブルのバイト 44-47 のビット 31 が 0 であるために起きる特殊操作例外。
8.A	トレース例外。
8.B.1	トラップ制御ブロックについてのアクセス例外 (取り出し)。
8.B.2	トラップ制御ブロック内の PSW 制御が 0 で、PSW のビット 31 が 1 であるために起きる特殊操作例外。
8.B.3	トラップ保管域についてのアクセス例外 (格納)。

図 10-34. 実行の優先順位: TRAP

第11章 マシン・チェックの取り扱い

マシン・チェックの検出	11-2	処理損傷	11-19
マシン誤動作の訂正	11-2	記憶機構エラー	11-19
エラー検査/訂正	11-2	未訂正の記憶機構エラー	11-19
CPU 再試行	11-2	訂正済みの記憶機構エラー	11-19
CPU 再試行の影響	11-3	未訂正の記憶キー・エラー	11-19
チェックポイント同期化	11-3	記憶機構性能低下	11-19
チェックポイント同期化の際のマシン・チェックの取り扱い	11-3	間接記憶機構エラー	11-20
チェックポイント同期化操作	11-3	マシン・チェック割り込みコードの妥当性ビット	11-20
チェックポイント同期化アクション	11-4	PSW-MWP の妥当性	11-20
チャンネル・サブシステム・リカバリー	11-4	PSW のマスクとキーの妥当性	11-20
ユニット削除	11-4	PSW プログラム・マスクと条件コードの妥当性	11-21
マシン・チェックの取り扱い	11-4	PSW 命令アドレスの妥当性	11-21
有効化	11-5	故障記憶機構アドレスの妥当性	11-21
記憶機構内の無効 CBC	11-6	外部損傷コードの妥当性	11-21
プログラムによる記憶機構の有効化	11-6	浮動小数点レジスターの妥当性	11-21
記憶キー内の無効 CBC	11-7	汎用レジスターの妥当性	11-21
レジスター内の無効 CBC	11-9	制御レジスターの妥当性	11-21
チェック停止状態	11-10	記憶機構の論理的妥当性	11-21
システム・チェック停止	11-11	アクセス・レジスターの妥当性	11-21
マシン・チェック割り込み	11-11	TOD プログラマブル・レジスターの妥当性	11-21
緊急条件	11-11	浮動小数点レジスターの妥当性	11-21
抑制可能条件	11-11	CPU タイマーの妥当性	11-21
割り込みアクション	11-12	クロック・コンパレーターの妥当性	11-21
割り込み点	11-13	マシン・チェック拡張割り込み情報	11-22
マシン・チェック割り込みコード	11-14	レジスター保管域	11-22
サブクラス	11-15	外部損傷コード	11-22
システム損傷	11-15	故障記憶機構アドレス	11-22
命令処理損傷	11-15	マシン・チェック条件の取り扱い	11-23
システム・リカバリー	11-16	浮動割り込み条件	11-23
タイミング機構損傷	11-16	浮動マシン・チェック割り込み条件	11-23
外部損傷	11-16	浮動入出力割り込み	11-23
性能低下	11-17	マシン・チェック・マスキング	11-23
警告	11-17	チャンネル報告保留サブクラス・マスク	11-24
チャンネル報告保留	11-17	リカバリー・サブクラス・マスク	11-24
サービス・プロセッサ損傷	11-17	性能低下サブクラス・マスク	11-24
チャンネル・サブシステム損傷	11-17	外部損傷サブクラス・マスク	11-24
サブクラス修飾子	11-17	警告サブクラス・マスク	11-24
バックアップ	11-18	マシン・チェック・ログアウト	11-24
遅延アクセス例外	11-18	マシン・チェックのマスキングの要約	11-24
補助報告	11-18		
同期マシン・チェック割り込み条件	11-18		
処理バックアップ	11-18		

マシン・チェック取り扱いメカニズムは、システム動作の保全性を確保し、一部の誤動作からの自動リカバリーを可能にする、広範な装置誤動作検出機能を提供します。プログラム損傷の影響評価とリカバリーを助けるために、マシン・チェック割り込みを使用して、装置誤動作とある種の外部障害が報告されます。この割り込みは、損傷の程度と、原因の位置と性質に関する情報を、プログラムに提供します。装置誤動作、エラー、および、マシン・チェック割り込みを起こす可能性のあるその他の状況を、マシン・チェックと呼びます。

マシン・チェックの検出

マシン・チェック検出メカニズムにはさまざまな形態があり、特に、算術演算や論理演算の処理、アドレッシング、順序付け、および実行のための制御機能の中では、多数の検出メカニズムが働きます。プログラムでアドレッシングできる情報については、通常、情報に冗長コードを組み込んでおくことにより、この検出が行われます。これにより、情報の保存または伝送に障害が起きると、ほとんどの場合は、無効コードが生じることになります。通常、このエンコードの形式は、検査ビットと呼ばれる 1 つ以上の冗長ビットに、1 グループのデータ・ビットを付加したものです。このようなデータ・ビットのグループとそれに関連した検査ビットを、検査ブロックと呼びます。検査ブロックのサイズは、モデルによって異なります。

検査ブロックに 1 個の検査ビットを組み込んでおくことにより、検査ブロック内の単一ビット誤りをすべて検出できます。この方法で使用する検査ビットを、「パリティ・ビット」と呼ぶことがあります。ほかに、1 グループの検査ビットを組み込んで、複数のエラーの検出、またはエラーの訂正、またはその両方ができるようにする方法もあります。

検査を目的として、冗長コードも含めた検査ブロック全体の内容を検査ブロック・コード (CBC) と呼びます。CBC が検査要件を完全に満たしている (つまりまったく誤りがない) 場合、それを有効と言います。検出と訂正の両方の機能があるときに、CBC が有効ではないが訂正のための検査要件は満たしているという場合 (つまり訂正可能な誤りの場合) は、それを近似的有効と言います。CBC が検査要件を満たしていない場合 (つまり訂正不能の誤りの場合) は、それを無効と言います。

マシン誤動作の訂正

マシンが検出した誤動作からのリカバリーのために使用できるメカニズムには、エラー検査/訂正、CPU 再試行、チャンネル・サブシステム・リカバリー、およびユニット削除の 4 つがあります。

正常に訂正されたマシン障害は、マシン・チェック割り込みとして報告される場合とされない場合があります。報告されれば、それはシステム・リカバリー条件であり、プログラムは、CPU 遅延の原因を認識し、この種の問題のログを維持することができます。

エラー検査/訂正

回路または検査ブロックに十分な冗長コードが組み込まれていれば、誤りは訂正可能です。例えば、回路を 3 重にすることにより、3 つのうち結果が一致する 2 つを選んで正しい値を決定するような回路構成にしておけば、単一誤りを訂正することができます。ある次元の誤りを訂正し、それより高い次元の誤りを検出する仕組みを、エラー検査/訂正 (ECC) と呼びます。通常、ECC では、単一ビット誤りの訂正と 2 ビット誤りの検出が可能です。

モデル、および ECC が適用されたマシンの部分に応じて、訂正はシステム・リカバリーとして報告されることもあり、また何も報告されないこともあります。

記憶域および記憶キーの中の訂正されなかったエラーは、エラーの発生場所を示す故障記憶機構アドレスと共に報告されることがあります。この種のエラーは、状況に応じて、システム条件と共に報告されることもあり、エラーの結果生じた損傷やバックアップ条件と共に報告されることもあります。

CPU 再試行

一部のモデルでは、マシンの状態の一部に関する情報が定期的に保管されます。処理の中でこの情報が保管される時点を、チェックポイントと呼びます。保管される情報をチェックポイント情報と呼びます。情報を保管する操作を、チェックポイントの設定と言います。保管されている情報を破棄する処置を、チェックポイント情報の無効化と呼びます。チェックポイント設定の間隔の長さは、モデルによって異なります。チェックポイントは、各命令の始めに設定することや、1 つの命令内で複数回設定することも、もっと低い頻度で設定することもできます。

この保管された情報は、後で、チェックポイントが設定された時点の状態にマシンを復元するために使用できま

す。マシン状態の該当部分を復元した後で、チェックポイントから処理が続行されます。チェックポイントの時点で復元し、そして処理を続行するプロセスを、CPU 再試行と呼びます。

CPU 再試行は、ある種のプログラム割り込みが生じたとき、およびモデルによって異なるその他の特定の状況の中で、マシン・チェック・リカバリーを目的として、命令実行の無効化と抑止を実施するために使用されます。

CPU 再試行の影響

通常は、CPU 再試行によりプログラムが何らかの影響を受けることはありません。ただし、0 から 1 に変更された変更ビットは、必ずしも 0 に戻されないことがあります。その結果、チェックポイントへの復元が発生しなければアクセスされるはずだったブロックについて、変更ビットが 1 に設定されているように見ることがあります。プログラムがとるパスが他の CPU またはチャンネル・プログラムにより変更される可能性のある情報に依存している場合、または割り込みが起きた場合は、プログラムが最終的にとるパスは初期のパスと異なることがあります。したがって、これまでとられたことがないとみられるパスを使用して格納が行われるため、変更ビットが 1 になることがあります。

チェックポイント同期化

チェックポイント同期化は、以下のステップに従って進められます。

1. 概念上現時点より前であると認められる、この CPU からのすべての記憶域アクセスが、マシン・チェック検出の目的から見ても、また他の CPU およびチャンネル・プログラムから見ても完了するまでは、この CPU の操作は延期されます。
2. 前のチェックポイント (存在する場合) は、すべて取り消されます。
3. 必要があれば、新しいチェックポイントが設定されます。

他の CPU およびチャンネル・プログラムから見て、上記のすべてのアクションが完了したと認められるまで、CPU 操作は延期されます。

チェックポイント同期化の際のマシン・チェックの取り扱い

チェックポイント同期化アクションの一環として、以前のすべての格納を完了するプロセスにおいて、マシンがすべての格納を正常に完了することはできないが、以前のチェックポイントにまで正常に復元できるという場合は、処理バックアップが報告されます。

チェックポイント同期化アクションの一環としてすべての格納を完了するプロセスにおいて、マシンがすべての格納を正常に完了することができず、また以前のチェックポイントにまで正常に復元することもできないときは、どのタイプのマシン・チェック割り込み条件が報告されるかは、格納の起点によって異なります。命令実行に関連した格納を正常に完了できないときは、命令処理損傷として報告されることがありますが、記憶域論理妥当性ビットを 0 に設定することにより、比較的軽度のマシン・チェック割り込み条件が報告されることもあります。プログラム割り込みまたは監視プログラム呼び出し割り込み以外の割り込みの実行に関連した格納を正常に完了できない場合は、システム損傷が報告されます。

ある命令の実行前に、チェックポイント同期化アクションの一環としてマシン・チェックが起きたときは、その命令の実行は無効化されます。割り込みの実行前にマシン・チェックが起きたときは、その割り込み条件が外部割り込み、入出力割り込み、または再始動割り込みであれば、その割り込み条件は保留にされます。チェックポイント同期化操作がマシン・チェック割り込みであった場合は、元の条件と共に、記憶域論理妥当性ビットが 0 に設定されるか、命令処理損傷が報告されます。プログラム割り込み (存在する場合) は失われます。

チェックポイント同期化操作

すべての割り込み、および一部の命令の実行は、チェックポイント同期化アクションを引き起こします。チェックポイント同期化アクションを引き起こす操作を、チェックポイント同期化操作と呼びます。このような操作には以下のものがあります。

- CPU リセット
- すべての割り込み: 外部、入出力、マシン・チェック、プログラム、再始動、および監視プログラム呼び出し
- M1 フィールドがすべて 1 で、R2 フィールドがすべて 0 である BRANCH ON CONDITION (BCR) 命令
- LOAD PSW、LOAD PSW EXTENDED、SET STORAGE KEY EXTENDED、および SUPERVISOR CALL 命令
- すべての入出力命令
- MOVE TO PRIMARY、MOVE TO SECONDARY、PROGRAM CALL、PROGRAM TRANSFER、SET ADDRESS SPACE CONTROL、および SET SECONDARY ASN 命令、および、アンスタックする状態エントリーがプログラム呼び出し状態エントリーである場合の PROGRAM RETURN 命令
- 4 つのトレース機能: プランチ・トレース、ASN トレース、モード・トレース、および明示トレース

- PAGE IN および PAGE OUT

プログラミング上の注意: チェックポイント同期化アクションを引き起こすように定義されている命令は、チェックポイント情報を無効化しますが、必ずしも新しいチェックポイントを設定しません。さらに、CPU は、任意の 2 つの命令または操作単位の間、または 1 つの操作単位の内部に、チェックポイントを設定することがあります。したがって、マシン・チェックのための割り込み点は、必ずしも、チェックポイント同期化アクションを引き起こすものとして定義されている命令にあるとは限りません。

チェックポイント同期化アクション

入出力割り込みを除くすべての割り込みでは、割り込みの完了時にチェックポイント同期化アクションが行われます。入出力割り込みの場合は、割り込みの完了時に、チェックポイント同期化アクションが行われるときと行われないうちがあります。プログラム割り込み、監視プログラム呼び出し割り込み、および緊急マシン・チェック割り込みを除くすべての割り込みでは、割り込みの前にもチェックポイント同期化アクションが行われます。最初のチェックポイント同期化アクションの前または後に、新 PSW に対する取り出しアクセスが行われることがあります。また、最初のチェックポイント同期化アクションと 2 回目の同期化アクションの間には、割り込みに関連した現 PSW の格納アクセスおよび変更が行われます。

BRANCH ON CONDITION (BCR)、入出力命令、および SUPERVISOR CALL を除くすべてのチェックポイント同期化命令では、命令実行の前と後に、チェックポイント同期化アクションが行われます。BCR の場合は、必ず 1 回だけチェックポイント同期化アクションが行われますが、それは、命令アドレスが更新される前のこともあり後のこともあります。SUPERVISOR CALL の場合は、PSW 内の命令アドレスの更新も含めて、命令の実行前に、チェックポイント同期化アクションが行われます。監視プログラム呼び出し割り込みの後でとられるチェックポイント同期化アクションは、命令実行の一部ではなく、割り込みアクションの一部であると見なされます。入出力命令の場合は、チェックポイント同期化アクションは、命令実行の前には必ず行われ、命令実行の後には行われることも行われないうちがあります。

4 つのトレース機能 (ブランチ・トレース、ASN トレース、モード・トレース、および明示トレース) では、トレース・アクションの開始前と完了後に、チェックポイント同期化アクションが行われます。

チャンネル・サブシステム・リカバリー

チャンネル・サブシステム内にエラーが検出されたときは、チャンネル・サブシステムは、各種のチャンネル・サブシステム機能に関連した内部状態、およびチャンネル・サブシステムと各種のサブチャンネルの状態を、分析してリカバリーしようとしています。これはチャンネル・サブシステム・リカバリーと呼ばれるプロセスで、その結果として、完全なリカバリーが行われることも、また、1 つ以上の入出力操作が中止され、関連するサブチャンネルがクリアされることもあります。プログラムにチャンネル・サブシステム・リカバリーの状況を知らせるために、特殊なチャンネル報告保留マシン・チェック割り込み条件が生成されることがあります。

入出力操作に関連した誤動作は、その重大度に応じて、入出力割り込みメカニズムにより報告される場合と、チャンネル報告保留マシン・チェック割り込み条件およびチャンネル・サブシステム損傷マシン・チェック割り込み条件により報告される場合があります。

ユニット削除

一部のモデルでは、システムの特定制ユニット内の誤動作は、そのユニットの使用を中止することで回避できます。このようなユニット削除が行われる場合としては、例えば、キャッシュの全体または一部、または変換索引バッファ (TLB) を使用不可にした場合などが考えられます。ユニット削除は、性能低下マシン・チェック割り込み条件として報告されることがあります。

マシン・チェックの取り扱い

マシン・チェックは、データや命令の誤りではなくマシンの誤動作が原因で生じます。この条件が確実に満たされるようにするために、パワーオン・シーケンスの時点で、マシン制御が有効状態に初期化され、CPU レジスター、記憶キー、および主記憶機構に有効な CBC が入れられます。

使用可能でないコンポーネント (記憶位置、サブチャンネル、入出力装置など) が指定されていても、それによってマシン・チェックが示されることはありません。このような条件は、該当のプログラム割り込みまたは入出力割り込み、または条件コード設定により示されます。特に、構成内に存在しないかまたは記憶装置で電源がオフになっている記憶位置にアクセスしようとした場合は、CPU がそれを検出するとアドレッシング例外が起きますが、その記憶位置またはそれに関連した記憶キーの CBC が無効であったとしても、マシン・チェック条件は生成されません。同様に、チャンネル・サブシステムがこのような記憶位置にアクセスしようとした場合も、マ

シン・チェック条件が生じるのではなく、プログラム・チェックを示す入出力割り込み条件が生成されます。

ある操作の結果が無効な CBC を持つ情報に影響される可能性がある場合、または、他の何らかの誤動作が原因で、ある操作を正しく実行できる (または実行した) ことを確実に判断できない場合は、必ずマシン・チェックが示されます。無効な CBC を持つ情報が取り出されたとしても、その情報が使用されなければ、その条件は示されることも示されないこともあり、その無効な CBC はそのまま保存されます。

マシンの誤動作が検出されたときにとられるアクションは、モデル、誤動作の性質、および誤動作が起きた状況によって異なります。オペレーター機能のアクションに影響を与える誤動作は、マシン・チェックを引き起こす場合と、オペレーターに通知される場合があります。他の特定の操作 (SIGNAL PROCESSOR など) に影響を与える誤動作は、マシン・チェック割り込み条件を引き起こすことがあります。

入出力操作の一部として誤動作が検出された場合は、マシン・チェック割り込み条件、入出力エラー条件、またはその両方が起きることがあります。入出力エラー条件は、入出力割り込みにより示されるか、または、入出力命令の実行時に該当の条件コード設定により示されません。マシンが、入出力操作中に検出された故障記憶機構を報告した場合は、入出力エラー条件とマシン・チェック条件の両方が示されることがあります。プログラムにとっての主要な指示は、入出力エラー条件です。マシン・チェック条件は 2 次的な指示であり、これは、故障記憶機構アドレスと一緒にシステム・リカバリーとして提示されます。

入出力命令および入出力操作の一部として検出される一部の誤動作は、入出力マシン・チェック条件と呼ばれる特殊なマシン・チェック条件により報告されます。つまり、入出力関連の操作の一部として検出された誤動作は、エラーの種類に応じて、入出力エラー条件、入出力マシン・チェック条件、または非入出力マシン・チェック条件の 3 通りの手段のいずれかにより報告されます。場合によっては、これらのメカニズムのうち特定の 1 つのみによりエラーを報告する必要があるものとして、定義されていることがあります。また、どれか 1 つまたは複数の条件が示されることもあります。

プログラミング上の注意: マシン・チェック条件は、データや命令が原因で起こるものではなくマシンの誤動作が原因で起こるものと定義されていますが、マシンの誤動作以外のイベントが原因でマシン・チェック条件が生じる例外的な状況があります。次に 2 つの例を挙げます。

1. 場合によっては、チャンネル報告保留マシン・チェック割り込み条件が、エラーではない状況を示していることがあります。例えば、RESET CHANNEL PATH で指定されている機能の完了時に、この条件が生成されます。
2. DIAGNOSE の使い方に誤りがあると、マシン・チェック条件が生じる場合があります。

有効化

マシン・エラーは、一般に、誤動作の持続性に応じて、固的なものと断続的なものに分類されます。持続的なマシン・エラーを固定エラーと呼び、非持続的なエラーを断続的なエラーと呼びます。レジスターまたは記憶位置の場合は、さらに、外部生成エラーと呼ばれる第 3 のタイプのエラーも考慮に入れる必要があります。外部生成エラーが起きるのは、該当のレジスターまたは記憶位置自体には障害がなく、その位置の外部でのアクションが原因で、無効な CBC がその位置に持ち込まれた場合です。例えば、一時的な電源変動によりその値が影響を受けたり、情報がその位置に置かれるときに誤った値が持ち込まれたりすることがあります。

無効な CBC を持つ情報が取り出されるか、または、検査ブロックの一部のみの更新が試みられたときは、無効な CBC は無効なままで保存されます。検査ブロックに無効な CBC が含まれているときに、そのブロック全体の内容を置き換えようとする操作が行われた場合は、そのブロックが無効な CBC を持ったままになるか、それとも置き換えられるかは、操作とモデルによって異なります。現在の内容を無視して、無効な CBC を持つ検査ブロックの内容を置き換える操作を、有効化操作と呼びます。有効化は、断続的なエラーまたは外部生成エラーがあるレジスターまたは記憶位置に、有効な CBC を入れるために使用されます。

検査ブロックを有効化しても、次回にその検査ブロックがアクセスされたときに、必ず有効な CBC が観察されるとは限りません。固定エラーの場合は、有効化が効力を持つのは、失敗したビットが、そのビットが取り損なった値に設定されるような形で、検索ブロックに情報が入れられた場合に限られます。ビットが取り損なった状態と反対の状態にそのビットを設定しようとした場合は、有効化には効果はありません。したがって、固定エラーの場合に有効化が役に立つのは、根底には障害が残されていても、エラー条件だけを排除して、それ以上そのエラーが報告に示されないようにしたいときのみです。ただし、この記憶位置に他の値を格納しようとするとき、無効な CBC が生じるため、この記憶位置は使用できません。しかし、断続的なエラーの場合は、有効化により有効な CBC が復元され、以後は検査ブロックへの部

分的な格納ができるようになります。(部分的な格納とは、検査ブロック全体を置き換えずに、そのブロックに格納することです。)

検査ブロックが、記憶域内の複数バイトまたは CPU レジスター内の複数ビットから成っている場合は、無効な CBC を有効化できるのは、それらのバイトまたはビットが同時に置き換えられる場合のみです。

システム内の各フィールド・タイプについて、それぞれのフィールドを有効化するための特定の命令が定義されています。モデルによっては、有効化を行うための命令がほかにも用意されていることがあります。また、一部のモデルでは、レジスターの元の内容が該当の保管域に入れられた後で、マシン・チェック割り込みシーケンスの一環として、そのレジスターが自動的に有効化されません。

検査ブロック内でエラーが起きたときは、有効化が行われた後であっても、その検査ブロックに含まれていた元の情報は失われているものと見なすべきです。レジスターの自動有効化が行われた後の内容は、予測不能です。プログラムまたは手動で検査ブロックを有効化した場合は、内容は明示的に変更されます。

プログラミング上の注意: マシン・チェック割り込みハンドラーは、レジスターが有効化を必要とすることを想定していなければなりません。したがって、各レジスターが使用または格納される前に、有効化を行うものとして定義されている命令を使用して、レジスターをロードする必要があります。

記憶機構内の無効 CBC

記憶機構内の検査ブロックのサイズはモデルによって異なりますが、4K バイトを超えることはありません。

記憶機構内に無効 CBC が検出されると、マシン・チェック条件が起きることがあります。このマシン・チェック条件は、状況に応じて、システム損傷、命令処理損傷、またはシステム・リカバリーのいずれかです。チャンネル・プログラムの実行の一部として無効 CBC が検出された場合は、そのエラーは入出力エラー条件として報告されます。記憶機構から事前取り出しされた CCW、間接データ・アドレス・ワード、またはデータが、無効 CBC を持つことが判明したとしても、チャンネル・プログラムで使用されない場合は、通常その条件は入出力エラー条件として報告されません。この条件は、マシン・チェック割り込み条件として報告されることもあり、されないこともあります。CPU 関連のアクセス以外の記憶機構アクセス中に検出された無効 CBC は、他の何らかの手段で主要エラー指示が報告されるために、システ

ム・リカバリーとして報告され、訂正されていない記憶機構エラーが示されることがあります。

記憶機構検査ブロックが複数バイトから成り、無効な CBC を含んでいる場合は、通常、検査ブロックを復元するため、または検査ブロックに新しい情報を入れるために、特殊な記憶機構有効化手順が必要になります。記憶機構の有効化の手段としては、手動のロード消去操作とシステム・リセット・クリア操作があり、またプログラム機能を使用する方法もあります。プログラムによる記憶機構の有効化は、特権命令 TEST BLOCK を実行することにより、一度に 1 ブロックずつ行うことができます。クリア・リセットによる手動の記憶機構有効化では、構成内で使用可能なすべてのブロックが有効化されます。

無効 CBC を持つ検査ブロックは、その検査ブロックの内容全体が置き換えられるまでは、有効化されません。無効 CBC を持つ検査ブロックに、そのブロック全体を置き換えずに格納しようとした場合は、そのブロック内のデータ (検査ビットも含む) は変更されません。命令またはチャンネル・プログラム入力操作で、検査ブロックの内容全体を置き換えることが指定されていても、操作とモデルに応じて、有効化が行われることも行われなこともあります。

プログラミング上の注意: 事前取り出しされており、しかも使用されていないデータについて、マシン・チェック条件が報告されることがあります。このような状況で再試行が成功するかどうかは、モデルによって異なります。例えば、R₂ フィールドに 0 が指定された BRANCH AND LINK (BALR) 命令は、ブランチすることはありませんが、一部のモデルでは、レジスター 0 に指定されている記憶位置の事前取り出しが起きることがあります。この事前取り出しに関連したアクセス例外は、報告されません。ただし、無効な検査ブロック・コードが検出された場合は、CPU の再試行が行われることがあります。モデルによっては、この再試行の一環として再度事前取り出しが行われることがあり、その場合は再試行は成功しません。CPU 再試行が成功したとしても、パフォーマンスが著しく低下し、通常は故障記憶機構アドレスとともにシステム・リカバリーが提示されます。パフォーマンス低下が継続するのを回避するには、プログラムは、その記憶位置の使用を中止して有効化するための手続きを開始する必要があります。

プログラムによる記憶機構の有効化

ある 4K バイト・ブロックに関連した記憶キー内に無効 CBC が存在しない場合は、TEST BLOCK 命令を実行すると、記憶機構の現在の内容に関係なく、その 4K バイト・ブロック全体が、有効 CBC を持つ 0 に設定されます。つまり、TEST BLOCK は、断続的障害または一回限りの障害がある記憶機構内の記憶位置から無効

CBC を取り除きます。しかし、記憶機構の一部に永続障害がある場合は、以後の取り出しで無効 CBC が見つかることがあります。

記憶キー内の無効 CBC

モデルによっては、各記憶キーがそれぞれ 1 つの検査ブロックに含まれていることや、アクセス制御ビットと取り出し保護ビットおよび参照ビットと変更ビットがそれぞれ独立した検査ブロックに含まれていることがあります。

11-8ページの図11-1 は、記憶キーが無効 CBC を持っている場合のアクションを示しています。この図に示さ

れているのは、アクセス制御ビットと取り出し保護ビットが 1 つの検査ブロック内にあり、参照ビットと変更ビットが別の検査ブロック内にある場合にとられるアクションです。両方のフィールドが 1 つの検査ブロックに含まれているマシンでは、エラーのある各フィールドのアクションを組み合わせたアクションになりますが、完了が可能なのは、影響を受けたすべてのフィールドのエラーが完了を許すものである場合のみです。キー制御保護が適用されない主記憶機構に対する参照は、その参照用に 0 のアクセス・キーが使用されるものとして扱われます。これには、初期プログラム・ローディング中のチャンネル・プログラム参照などの参照、および、割り込みアクションや DAT テーブル・アクセスなどの暗黙参照が含まれます。

参照のタイプ	無効 CBC に対してとられるアクション	
	アクセス制御ビットと 取り出し保護ビット	参照ビットと 変更ビット
SET STORAGE KEY EXTENDED	完了: 有効化	完了: 有効化
INSERT STORAGE KEY EXTENDED	PD: 保存	PD: 保存
RESET REFERENCE BIT EXTENDED	PD または完了: 保存	PD: 保存
INSERT VIRTUAL STORAGE KEY or TEST PROTEC- TION	PD: 保存	CPF: 保存
CPU 事前取り出し (情報は使用されない)	CPF: 保存	CPF: 保存
チャンネル・プログラム 事前取り出し (情報は 使用されない)	IPF: 保存	IPF: 保存
取り出し、0 以外の アクセス・キー	MC: 保存	MC または完了: 保存
格納 ¹ 、0 以外の アクセス・キー	MC ² : 保存	MC および保存: または 完了 ³ および訂正
取り出し、0 以外の アクセス・キー ⁴	MC または完了: 保存	MC または完了: 保存
取り出し ¹ 、0 の アクセス・キー ²	MC または完了: 保存	MC および保存: または 完了 ³ および訂正
説明: ¹ CPU 仮想/論理アドレス格納アクセスは、ページ保護の対象 となります。ページ保護ビットが 1 のときは、位置は変更され ません。ただし、記憶キーまたはデータ自体に無効 CBC がある 場合は、マシンはマシン・チェック条件を示すことがあります。 ² 主記憶位置の内容は変更されません。 ³ 「完了」アクションがとられたときは、参照ビットおよび変更 ビットの内容は 1 に設定されます。 ⁴ 0 のアクセス・キーについて示されているアクションは、 キー制御保護の対象にならない参照にも適用されます。		

図 11-1 (1/2). 記憶キー内の無効 CBC

説明 (続き):

完了	この条件は命令実行を中止させることはなく、無関係の条件により禁止されない限り、エラー条件を無視して命令実行は完了します。マシン・チェック損傷条件は報告されませんが、システム・リカバリーは報告されることがあります。
訂正	参照ビットと変更ビットは、有効 CBC を持つ 1 に設定されま す。
保存	無効な CBC を持つ検査ブロック全体が、無変更のまま残されま す。
有効化	キー全体が、有効な CBC を持つ新しい値に設定されます。
CPF	使用されない CPU 事前取り出し、または参照ビットおよび変更 ビットを検査しない命令の場合、記憶キー内に無効 CBC がある と、以下のいずれかの状況が生じることがあります。 ・ 操作が完了する。マシン・チェック条件は報告されない。 ・ 操作が完了する。訂正されていない記憶キー・エラーと ともに、システム・リカバリーが報告される。 ・ バックアップ付きまたはバックアップなしで、訂正されて いない記憶キー・エラー・キーと共に、命令処理損傷が 報告される。
IPF	使用されなかったチャンネル・プログラム事前取り出しについて、 記憶キー内に無効 CBC がある場合は、以下のいずれかの状況が 生じることがあります。 ・ 入出力操作が完了する。マシン・チェック条件は報告され ない。 ・ 入出力操作が完了する。訂正されていない記憶キー・ エラーと共に、システム・リカバリーが報告される。
MC	CPU 参照の場合は PD と同じですが、チャンネル・サブシステム 参照では、入出力エラー条件とマシン・チェック条件を次のよ うに組み合わせた結果が生じることがあります。 ・ 入出力エラー条件が報告される。マシン・チェック条件は 報告されない。 ・ 入出力エラー条件が報告される。訂正されない記憶キー・ エラーを伴うかまたは伴わずに、システム・リカバリーが 報告される。
PD	バックアップ付きまたはバックアップなしで、訂正されていな い記憶キー・エラーを伴うかまたは伴わずに、命令処理損傷が 報告される。

注: 訂正されていない記憶キー・エラーが報告された場合は、故障記憶
機構アドレスが報告されることも、されないこともあります。

図 11-1 (2/2). 記憶キー内の無効 CBC

レジスター内の無効 CBC

CPU レジスター内に無効 CBC が検出された場合は、
マシン・チェック条件が認識されることがあります。
CPU レジスターには、汎用レジスター、浮動小数点レ
ジスター、浮動小数点制御レジスター、アクセス・レジ
スター、制御レジスター、および TOD プログラムマ
ル・レジスターのほか、現 PSW、プレフィックス・レ
ジスター、TOD クロック、CPU タイマー、およびクロ
ック・コンパレーターがあります。

マシン・チェック割り込みが起きたときは、その原因が
CPU レジスター内の無効 CBC にあるかどうかに関係
なく、割り込みの一環として、プレフィックス・レジ
スターと TOD クロックを除く CPU レジスターに影響を
与える以下のアクションがとられます。

1. レジスターの内容が、割り当てられている記憶位置
に保管されます。エラーのあるレジスターについ
ては、マシン・チェック割り込みコード内の対応する
妥当性ビットが 0 に設定されます。レジスターの保
管時に誤動作が検出されても、追加のマシン・チェ
ック割り込み条件が生じることはありません。代わ

りに、妥当性ビットの該当の設定により、格納されたすべての情報の正確さが示されます。

- 一部のモデルでは、無効な CBC を持つレジスターがその後に有効化され、その実際の内容は予測不能なものになります。他のモデルでは、プログラムによる有効化が必要です。

プレフィックス・レジスターおよび TOD クロックは、マシン・チェック割り込み時には格納されず、また、対応する妥当性ビットもなく、有効化もされません。

マシン・チェック割り込みの一環として自動的にレジスターが有効化されないモデルでは、無効 CBC を持つレジスターがあっても、そのレジスターの内容が実際に使用されない限り、マシン・チェック割り込み条件は生じません。このようなモデルでは、各レジスターに 1 つ以上の検査ブロックが含まれていることがありますが、1 つの検査ブロックに複数のレジスターが含まれることはありません。レジスターの一部のみがアクセスされるときに、同じレジスターの使用されない部分にある無効 CBC が原因で、マシン・チェック割り込み条件が起きることがあります。例えば、LOAD (LE) 命令が、浮動小数点レジスターの左半分 (つまり短精度形式) を置き換えようとしたときに、右半分に無効 CBC があるために、マシン・チェック割り込み条件が生じることがあります。

プレフィックス・レジスターに関連した無効 CBC は、マシン・チェック割り込みにより確実に報告することは不可能です。これは、その割り込み自体が、実アドレスを対応する絶対アドレスに変換するために、そのプレフィックス値を適用する必要があるからです。プレフィックス・レジスター内に無効 CBC があると、CPU はただちにチェック停止状態になります。

マシン・チェック割り込み時にレジスターを有効化しないモデルでは、以下に示す命令によりレジスターの有効化が行われます。ただし、これは、有効化の前にレジスター内の情報が使用されていない場合に限られます。その他の命令では、レジスターの内容全体が置き換えられますが、必ずしも有効化は行われません。

汎用レジスターを有効化する命令は、64 ビット・アドレッシング・モードで実行されている BRANCH AND SAVE と LOAD ADDRESS、および LOAD (LGR)、およびオペランドがダブルワード境界上にある場合の LOAD (LG) と LOAD MULTIPLE (LMG) です。

浮動小数点レジスターを有効化する命令は、LOAD (LDR) と、オペランドがダブルワード境界上にある場合の LOAD (LD) です。

浮動小数点制御レジスターを有効化する命令は、LOAD FLOATING POINT CONTROL REGISTER です。

アクセス・レジスターを有効化する命令は、LOAD ACCESS MULTIPLE です。有効化されるのは、LOAD ACCESS MULTIPLE の操作対象として指定されている一組のアクセス・レジスターに含まれる、奇偶レジスター・ペアのみです。つまり、単一のレジスターを指定した場合、または奇数番号から始まるアクセス・レジスターのペアを指定した場合は、有効化は行われません。

制御レジスターは、LOAD CONTROL (LCTLG) を使用して、単独またはグループで有効化することができます。

TOD プログラマブル・レジスター、CPU タイマー、クロック・コンパレーター、およびプレフィックス・レジスターを有効化する命令は、それぞれ、SET CLOCK PROGRAMMABLE REGISTER、SET CPU TIMER、SET CLOCK COMPARATOR、および SET PREFIX です。

TOD クロックは、クロックを設定する SET CLOCK 命令により有効化されます。

プログラミング上の注意: レジスターとモデルによっては、マシン・チェック割り込みによりレジスターの内容が有効化されることもあります。また、マシン・チェック割り込みが起きた後で、プログラムが有効化命令を実行しなければならないモデルもあります。CPU タイマーの場合は、モデルによっては、CPU タイマーをフル稼働状態に復元するために、マシン・チェック割り込みと有効化命令の両方が必要になることがあります。

チェック停止状態

マシン・エラーが起きたときに、状況によっては、操作を続行することが不可能または望ましくない場合があります。このような場合は、CPU はチェック停止状態になることがあり、その状態はチェック停止標識により示されます。

一般に、訂正不能のエラーまたはその他の誤動作が起き、マシンが特定のマシン・チェック割り込み条件を認識できないときは、CPU はチェック停止状態になることがあります。

以下のいずれかの条件が存在する場合は、CPU は必ずチェック停止状態になります。

- PSW のビット 13 が 0 であり、緊急マシン・チェック条件が生成された。
- 1 つの緊急マシン・チェック条件が原因で生じた割り込みの実行中に、別の緊急マシン・チェック条件が検出された。
- マシン・チェック割り込み時に、マシン・チェック割り込みコードが正常に格納できないか、または新 PSW を正常に取り出せない。
- プレフィックス・レジスター内で無効 CBC が検出された。
- SIGNAL PROCESSOR の指令を受け入れた後で、受信側 CPU で起きた誤動作が原因でその指令が正常に完了できなくなり、かつ、その指令がリセットであったか、または受信側 CPU がその指令が何であったかを判別できない。この場合は、受信側 CPU がチェック停止状態になります。

モデルによっては、エラーがチェック停止を引き起こすような条件が、ほかにも多数あります。

CPU がチェック停止状態にあるときは、命令および割り込みは実行されません。TOD クロックは、通常、チェック停止状態の影響は受けません。CPU タイマーは、チェック停止状態のときは、エラーとモデルに応じて、稼働を続ける場合とそうでない場合があります。この状態では、開始キーと停止キーは働きません。

CPU リセットにより、CPU をチェック停止状態から抜け出させることができます。

マルチプロセッシング構成では、チェック停止状態に入る CPU は、構成内のすべての CPU に対して、誤動作警報外部割り込みを要求します。他の CPU および入出力システムは、通常、誤動作警報を受け取る以外には、CPU の 1 つがチェック停止状態にあることで何らかの影響を受けることはありません。ただし、チェック停止を引き起こした条件の性質によっては、他の CPU も遅延または停止することがあり、チャンネル・サブシステムおよび入出力活動が影響を受けることがあります。

システム・チェック停止

マルチプロセッシング構成では、エラー、誤動作、および損傷条件によっては、構成内のすべての CPU がチェック停止状態になるほど重大な結果をもたらすことがあります。このような条件をシステム・チェック停止と呼びます。この場合のチャンネル・サブシステムおよび入出力活動の状態は、予測不能です。

マシン・チェック割り込み

マシン・チェックの結果として保留にされているマシン・チェック割り込みの要求を、マシン・チェック割り込み条件と呼びます。マシン・チェック割り込み条件には、緊急条件と抑制可能条件の 2 種類があります。

緊急条件

緊急マシン・チェック割り込み条件は、現在の命令または割り込みシーケンスの実行を安全に続行できないような損傷が起きた（または起きたはずであると認められる）場合に生じる条件です。緊急条件には、命令処理損傷およびシステム損傷という 2 つのサブクラスがあります。システム損傷は、特定の緊急条件を示すほかに、それ以上重大度の低い報告に切り離すことができないような誤動作やエラーを報告するために使用されます。

命令シーケンスに関する緊急条件は、影響を受ける命令が無効化されるか中止されるかに従って、無効化緊急条件の場合と中止緊急条件の場合があります。割り込みシーケンスについての緊急条件は、中止緊急条件です。ここで言う「無効化」および「中止」という用語は、第 5 章、『プログラムの実行』で使用されている同じ用語の意味と同じです。ただし、ここでは複数の命令が対象となることがあります。つまり、無効化緊急条件は、CPU が、エラーが起きる前の操作単位の始めに戻ったことを示します。中止緊急条件は、1 つ以上の命令の結果が予測不能な値になる可能性があることを意味します。

抑制可能条件

抑制可能マシン・チェック割り込み条件は、命令処理シーケンスの結果に影響を与えないような条件です。抑制可能条件は、CPU 操作の安全性を損なうことなく、現在の命令が完了するか、または場合によってはもっと先まで、遅延させることができます。抑制可能条件は、リカバリー、警報、および抑制可能損傷の 3 つのグループに分けられます。各グループには、それぞれ 1 つ以上のサブクラスがあります。

CPU、記憶機構、またはオペレーター機能に誤動作が生じ、それが正しく訂正されたか、または論理的障害を残さずに内部的に回避された場合、その状況をリカバリー条件と呼びます。モデルおよび誤動作のタイプによっては、一部またはすべてのリカバリー条件が破棄され、報告されないことがあります。報告されるリカバリー条件は、システム・リカバリーという 1 つのサブクラスにまとめられています。

マシン誤動作に直接関係ないマシン・チェック割り込み条件を、警報条件と呼びます。警報条件は、性能低下および警告の2つのサブクラスに分類されます。

システムの一部に誤った状態を生じさせるが、順次 CPU 操作には直接影響を与えることのない誤動作を、抑制可能損傷条件と呼びます。抑制可能損傷条件は、影響を受ける機能に従って、タイミング機構損傷、外部損傷、チャネル報告保留、チャネル・サブシステム損傷、およびサービス・プロセッサ損傷の5つのサブクラスに分類されています。

プログラミング上の注意:

1. 抑制可能条件は、通常は正常な割り込み点でのみ報告されるものですが、緊急マシン・チェック条件とともに報告されることもあります。つまり、ある緊急マシン・チェック条件が原因で命令が異常終了し、その緊急条件を報告するためのマシン・チェック割り込みが起きた場合は、保留中の抑制可能条件も同時に報告されることがあります。妥当性ビットが意味を持つかどうかは、どの緊急条件が報告されるかによって異なります。
2. 損傷が緊急または抑制可能のどちらに分類されるかは、損傷の重大度を意味するものではありません。両者の違いは、損傷が検出されたときにただちにアクションを行う必要があるか(緊急)、それとも CPU が処理を続行できるか(抑制可能)という点にあります。抑制可能条件の場合は、CPU がマシン・チェックに対して使用可能にされていれば、現在の命令が完了してから、マシン・チェック割り込みを起こさせることができます。CPU がマシン・チェックに対して使用不可にされているときは、CPU が再びマシン・チェックに対して使用可能にされるまでその条件を保留にしておいても、安全性が損なわれることはありません。

例えば、CPU は、前の命令処理損傷割り込みを処理しているために、マシン・チェック割り込みに対して使用不可にされていることがあります。このようなときに、入出力操作で記憶機構エラーが検出された場合、その条件は、現在のマシン・チェック処理を妨害するものとは考えられないので、保留にしておくことができます。しかし、マシン・チェック割り込みを再度使用可能にする前に CPU がエラーのある記憶域を参照したとすれば、新たな命令処理損傷条件が発生します。この条件は緊急条件として扱われるため、CPU はチェック停止状態になります。

3. 抑制可能条件は、浮動条件の場合があります。浮動抑制可能条件は、構成内のどの CPU に対しても割

り込みを起こさせる可能性があります。浮動抑制可能条件に対して CPU が割り込みを行った時点で、その条件は、構成内の他の CPU で割り込みを起こさせることはできなくなります。

割り込みアクション

マシン・チェック割り込みは、以下のアクションを引き起こします。アーキテクチャー・モード ID として、16進数値 01 が実記憶位置 163 に格納されます。割り込み点を反映した PSW が、マシン・チェック旧 PSW として、実記憶位置 352 にあるクワッドワードに格納されます。他のレジスタの内容が、実記憶位置 4608-4863、4892-4895、4900-4911、4913-4919、および 4928-5119 にあるレジスタ保管域に格納されます。モデルによっては、レジスタの内容がレジスタ保管域に格納された後で、レジスタが有効化され、その内容が予測不能になることがあります。8 バイトのマシン・チェック割り込みコード (MCIC) が、実記憶位置 232-239 に格納されます。外部損傷コードが実記憶位置 244-247 に格納されることがあり、また、故障記憶機構アドレスが実記憶位置 248-255 に格納されることがあります。実記憶位置 480-495 から新 PSW が取り出されます。さらに、マシン・チェック・ログアウトが起きていることがあります。

旧 PSW と新 PSW、MCIC、拡張割り込み情報、および固定ログアウト域にアクセスするための使用されるマシン生成のアドレスは、すべて実アドレスです。

図11-2 は、マシン・チェック割り込み時にアクセスされる割り当て済み記憶位置のフィールドを要約して示しています。

格納される (取り出される) 情報	開始位置 *	長さ (バイト)
アーキテクチャー・モード ID	163	1
旧 PSW	352	16
新 PSW (取り出された)	480	16
マシン・チェック割り込みコード	232	8
レジスタ保管域		
浮動小数点レジスタ 0-15	4608	128
汎用レジスタ 0-15	4736	128
浮動小数点制御レジスタ	4892	4
TOD プログラマブル・レジスタ	4900	4
CPU タイマー	4904	8
クロック・コンパレータ	4913	7
アクセス・レジスタ 0-15	4928	64
制御レジスタ 0-15	4992	128
拡張割り込み情報		
外部損傷コード	244	4
故障記憶機構アドレス	248	8
固定ログアウト域	4864	16

説明:
* 位置はすべて実記憶位置です。

図 11-2. マシン・チェック割り込みの記憶位置

マシン・チェック割り込みコードを正しく格納できない場合、または新 PSW を正しく取り出せない場合は、CPU はチェック停止状態になります。

抑制可能マシン・チェック条件がマシン・チェック割り込みを引き起こすことがあるのは、PSW のビット 13 が 1 で、かつ、制御レジスター 14 の中の対応するサブクラス・マスク・ビット (存在する場合) も 1 である場合のみです。この割り込みが起きても、現在の命令の実行は中止されません。この割り込みは通常の割り込み点で発生し、プログラム割り込みまたは監視プログラム呼び出し割り込みが排除されることはありません。CPU タイマー更新など、何らかのマシン機能の実行中にマシン・チェックが生じた場合は、そのマシン機能が完了した後で、マシン・チェック割り込みが起きます。

特定の抑制可能マシン・チェック条件に対して CPU が使用不可にされている場合は、その条件は保留状態のままになります。モデルと条件によっては、特定のサブクラスについて検出された条件の数に関係なく、そのサブクラスに属する複数の抑制可能条件が保留にされることも、そのクラスに属する条件が 1 つだけ保留にされることもあります。

割り込み条件が、CPU が使用可能にされているサブクラス内にあるために、抑制可能マシン・チェック割り込みが起きた場合、CPU が他のサブクラスに対して使用不可にされていても、同じ割り込みコードにより他のサブクラス内の保留中の条件も示されることがあります。そして、示されたすべての条件がクリアされます。

前のマシン・チェック条件が原因で生じた割り込みプロシージャの実行中に、システム・リカバリーとして報告すべきマシン・チェックが検出された場合は、そのシステム・リカバリー条件は、他の条件と組み合わせられることも、破棄されることも、保留にされることもあります。

緊急マシン・チェック条件がマシン・チェック割り込みを引き起こすことがあるのは、PSW のビット 13 が 1 のときだけです。無効化緊急条件がマシン・チェック割り込みを起こす場合は、割り込みは通常の割り込み点で発生します。中止緊急条件がマシン・チェック割り込みを起こす場合は、その割り込みにより現在の命令の実行は中止されます。また、実行が続行されていたとすれば起こったはずのプログラム割り込みおよび監視プログラム割り込みがある場合は、それらの割り込みは除去されます。旧 PSW やその他の情報の格納も含めて、割り込みシーケンスが正しく実行されるかどうかは、誤動作の性質によって異なります。CPU タイマーの更新など、何らかのマシン機能の実行中に緊急マシン・チェック条

件が発生したときは、割り込みシーケンスは必ず完了するとは限りません。

ある緊急マシン・チェック条件が原因で割り込みが実行されているときに、別の緊急マシン・チェックが検出された場合は、CPU はチェック停止状態に入ります。抑制可能マシン・チェック条件が原因で割り込みが実行されているときに、緊急マシン・チェックが検出された場合は、システム損傷が報告されます。

PSW のビット 13 が 0 のときに緊急マシン・チェック条件が起きたときは、CPU はチェック停止状態に入ります。

マシン・チェック割り込み条件の取り扱いは、PSW の待ち状態ビットが 1 であっても 0 であっても同じです。マシン・チェック条件が生じたときに、CPU がその条件に対して使用可能にされていれば、割り込みが起きます。

速度制御が命令ステップ位置に設定されているときに起きたマシン・チェックの取り扱いは、速度制御が処理位置に設定されているときと同じです。つまり、リカバリー・メカニズムはアクティブであり、マシン・チェック割り込みは、可能な状態にあれば発生します。手動操作時にマシン・チェックが起きた場合は、モデルに応じて、オペレーターに知らされるか、システム・リカバリー条件が生成されるか、システム損傷が起きるか、またはチェック停止になります。

マシン・チェックおよびそれに関連した割り込みによる副次作業を制限するために、妥当と認められるすべての試みが行われます。通常、割り込みも、入出力操作の進行も影響を受けません。ただし、誤動作によりこれらの活動が影響を受けることがあり、現在アクティブな PSW のビット 13 が 1 に設定されている場合は、マシン・チェック割り込みは、この条件の原因になった損傷だけでなく、発生した全範囲損傷を示します。

割り込み点

処理の中で、割り込みにより示され、マシンが格納された状況の妥当性を判別して示すために参照点として使用する点を、割り込み点と呼びます。

CPU 再試行機能を持つモデルでは、チェックポイント能力があるため、緊急マシン・チェック割り込み条件を原因とする割り込みにより、CPU 処理シーケンス内で論理上はエラーより前にある点が示されることがあります。さらに、この種のモデルでは、CPU 処理シーケンスのどの時点で割り込みを示すかについてある程度選択できることがあり、有効として指示可能な状況は、選択された時点によって異なることもあります。

割り込み点として使用されるのは、処理の中の所定の点のみです。抑制マシン・チェック割り込みの場合は、割り込み点は、1つの操作単位が完了し、関連のプログラム割り込みまたは監視プログラム呼び出し割り込みがとられた後で、かつ次の操作単位が始まる前でなければなりません。

命令シーケンスに関する緊急マシン・チェック条件は、命令ストリームに対して損傷が起きた(または起きるはずだった)ときの条件です。したがって、通常、損傷は1つの命令の進行途中にある特定の点に関連付けられるもので、この点を損傷点と呼びます。場合によっては、損傷点と割り込み点を切り離す命令が1つ以上存在することがあり、また、1つ以上の命令に関連した処理が損傷を受けることがあります。割り込み点が、無効化可能緊急マシン・チェック条件を原因とする損傷点より前にある場合は、その割り込み点は、抑制可能マシン・チェック条件の場合と同じ点に限られます。

抑制可能マシン・チェック条件の場合に許される割り込み点に加えて、中止緊急マシン・チェック条件の場合の割り込み点も、操作単位の完了後、関連のプログラム割り込みまたは監視プログラム呼び出し割り込みが起きる前に、設定することができます。この場合、有効な PSW 命令アドレスと見なされるのは、そのプログラム割り込みまたは監視プログラム呼び出し割り込みのアドレスとして旧 PSW に格納されるはずであったアドレスです。操作は中止されているので、命令アドレス以外の結果フィールドの値は、予測不能です。したがって、中止緊急マシン・チェック条件が報告された場合は、命令

ストリームにより変更されるはずのフィールドに関連した妥当性ビットは、無意味です。

緊急マシン・チェック条件を原因とする損傷点と割り込み点が、チェックポイント同期化機能により切り離されている場合は、損傷は特定のプログラムに対応付けられておらず、システム損傷が示されます。

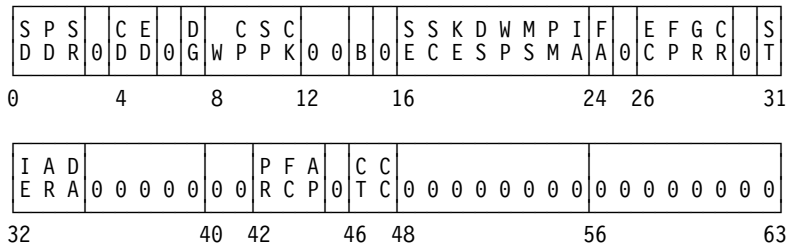
緊急マシン・チェック割り込み条件が起きたときは、どの割り込み点を選択されているかによって、指示する必要がある損傷の量が変化します。可能であれば、最小限の損傷の提示ですむような割り込み点を選択されます。一般に、エラーの直前の割り込み点が優先的に選択されます。

緊急マシン・チェック割り込み条件の結果として格納されたすべての状況情報が、同一地点の状況を反映しているものではない場合は、可能な限り、マシン・チェック旧 PSW に格納される命令アドレスが有効になるように、割り込み点を選択されます。

マシン・チェック割り込みコード

どのマシン・チェック割り込みでも、マシン・チェック割り込みコード (MCIC) が、実記憶位置 232 から始まるダブルワードに格納されます。このコードのフォーマットは、11-15ページの図11-3 に示すとおりです。

MCIC 内の未割り当てのビット、または特定モデルで使用されていないビットは、0 として格納されます。



ビット 名称

- 0 システム損傷 (SD)
- 1 命令処理損傷 (PD)
- 2 システム・リカバリー (SR)
- 4 タイミング機構損傷 (CD)
- 5 外部損傷 (ED)
- 7 性能低下 (DG)
- 8 警告 (W)
- 9 チャンネル報告保留 (CP)
- 10 サービス・プロセッサ損傷 (SP)
- 11 チャンネル・サブシステム損傷 (CK)
- 14 バックアップ (B)
- 16 未訂正の記憶機構エラー (SE)
- 17 訂正済み記憶機構エラー (SC)
- 18 未訂正の記憶キー・エラー (KE)
- 19 記憶機構性能低下 (DS)
- 20 PSW-MWP 妥当性 (WP)
- 21 PSW マスクおよびキーの妥当性 (MS)
- 22 PSW プログラム・マスクと条件コードの妥当性 (PM)
- 23 PSW 命令アドレスの妥当性 (IA)
- 24 故障記憶機構アドレスの妥当性 (FA)
- 26 外部損傷コードの妥当性 (EC)
- 27 浮動小数点レジスタの妥当性 (FP)
- 28 汎用レジスタの妥当性 (GR)
- 29 制御レジスタの妥当性 (CR)
- 31 記憶機構の論理的妥当性 (ST)
- 32 間接記憶機構エラー (IE)
- 33 アクセス・レジスタの妥当性 (AR)
- 34 遅延アクセス例外 (DA)
- 42 TOD プログラマブル・レジスタの妥当性 (PR)
- 43 浮動小数点制御レジスタの妥当性 (FC)
- 44 補助報告 (AP)
- 46 CPU タイマーの妥当性 (CT)
- 47 クロック・コンパレーターの妥当性 (CC)

注: MCIC の他のすべてのビットは未割り当てであり、0 として格納されます。

図 11-3. マシン・チェック割り込みコードのフォーマット

サブクラス

ビット 0-2 および 4-11 は、割り込みを起こすマシン・チェック条件のタイプを示すサブクラス・ビットです。サブクラス・ビットの少なくとも 1 つは、1 として格納されます。複数のエラーが起きたときは、幾つかのサブクラス・ビットが 1 に設定されることがあります。

システム損傷

ビット 0 (SD) が 1 のときは、もっと重大度の低い 1 または複数のマシン・チェック・サブクラスに特定できない損傷が起きたことを示します。システム損傷が示されているときは、補助報告ビット (ビット 44) が意味を持ち、マシン・チェック割り込みコードのその他のビットは無意味であり、また、レジスタ保管域およびマシ

ン・チェック拡張割り込みフィールドに格納されている情報も無意味です。

システム損傷は中止緊急条件であり、サブクラス・マスク・ビットはありません。

命令処理損傷

ビット 1 (PD) が 1 のときは、CPU の命令処理に損傷が起きたことを示します。

ビット 1 の正確な意味は、バックアップ・ビット (ビット 14) の設定によって異なります。バックアップ・ビットが 1 のときは、この条件は処理バックアップと呼ばれます。バックアップ・ビットが 0 のときは、この条件は処理損傷と呼ばれます。この 2 つの条件については、11-18 ページの『同期マシン・チェック割り込み条件』で説明します。

命令処理損傷は、無効化緊急条件の場合と中止緊急条件の場合があり、サブクラス・マスク・ビットはありません。

システム・リカバリー

ビット 2 (SR) が 1 のときは、誤動作が検出されたが、損傷には至らなかったか、または正常に訂正されたことを示します。入出力操作の一部として誤動作が検出された場合は、入出力エラー条件に加えてシステム・リカバリー条件が生じることがあります。システム・リカバリー能力の有無とその範囲は、モデルによって異なります。

システム・リカバリーは、抑制可能条件です。この条件は、リカバリー・サブクラス・マスク・ビット (制御レジスター 14 のビット位置 36) によりマスクされます。

プログラミング上の注意:

1. システム・リカバリーは、CPU 事前取り出しまたは入出力操作で検出された故障記憶機構アドレスを報告するために使用されることがあります。
2. システム・リカバリーが示されていても、対応する妥当性ビットが 1 でない限り、記憶機構が論理的に有効であること、またはマシン・チェック割り込みの結果として格納されたフィールドが有効であることを意味するわけではありません。

タイミング機構損傷

ビット 4 (CD) が 1 のときは、TOD クロック、CPU タイマー、クロック・コンパレーター、または TOD プログラマブル・レジスター、あるいは、CPU タイマーまたはクロック・コンパレーターの外部割り込み条件に、損傷が起きたことを示します。以下のいずれかが起きたときは、必ずタイミング機構損傷マシン・チェック条件が設定されます。

1. TOD クロックが、エラー状態または非稼働状態になる。
2. CPU タイマーが損傷を受け、CPU は CPU タイマー外部割り込みに対して使用可能にされている。一部のモデルでは、CPU が CPU タイマー割り込みに対して使用可能にされていないときでも、この条件が認識されることがあります。モデルによっては、CPU タイマーがエラー状態になったときのみ、このマシン・チェック条件が生成される場合もあります。また、CPU が CPU タイマー割り込みに対して使用可能にされている限り、CPU タイマーが有効化されるまで、継続的にこのマシン・チェック条件が生成されることもあります。

3. クロック・コンパレーターが損傷を受け、CPU がクロック・コンパレーター外部割り込みに対して使用可能にされている。一部のモデルでは、CPU がクロック・コンパレーター割り込みに対して使用可能にされていないときでも、この条件が認識されることがあります。

TOD クロック、CPU タイマー、またはクロック・コンパレーターにアクセスする命令の結果が正しくない場合は、タイミング機構損傷は、命令処理損傷と一緒に設定されることもあります。モデルによっては、TOD プログラマブル・レジスター、CPU タイマー、またはクロック・コンパレーターは、その TOD クロック、CPU タイマー、またはクロック・コンパレーターを無効として報告する割り込みによって、有効化されることがあります。

タイミング機構損傷は、抑制可能条件です。この条件は、外部損傷サブクラス・マスク・ビット (制御レジスター 14 のビット位置 38) によりマスクされます。

ほとんどのモデルでは、CPU タイマーおよびクロック・コンパレーターに関するタイミング機構損傷条件は、これらの機構が使用されていないときは認識されません。これらの機構が使用されていないと見なされるのは、CPU が対応する外部割り込みに対して使用不可にされているとき (PSW のビット 7、または制御レジスター 0 のビット 52 および 53 にあるサブクラス・マスク・ビットが 0 であるとき)、および対応する SET または STORE 命令が実行されていないときです。しかし、すでに保留にされているタイミング機構損傷条件は、CPU が対応する外部割り込みに対して使用不可にされているときは、保留のままになります。

TOD クロックに対する損傷が原因で生じるタイミング機構損傷条件は、常に認識されます。

外部損傷

ビット 5 (ED) が 1 のときは、現在の命令の処理には直接関連していない操作で損傷が起きたことを示します。

ビット 5 (外部損傷) が 1 で、ビット 26 (外部損傷コードの妥当性) も 1 のときは、外部損傷マシン・チェック割り込みの原因を詳細に示す外部損傷コードが格納されています。外部損傷を、外部損傷コードで定義されている 1 つまたは複数の条件に限定できないとき、または、モデルでこの条件を詳細に示す手段がインプリメントされていないときは、ビット 26 を 0 に設定することにより外部損傷が示されます。外部損傷の報告の有無とその範囲は、モデルによって異なります。

外部損傷は、抑制可能条件です。この条件は、外部損傷サブクラス・マスク・ビット (制御レジスター 14 のビット位置 38) によりマスクされます。

性能低下

ビット 7 (DG) が 1 のときは、システム・リカバリーが示す程度より重大度の高い、継続的なシステム・パフォーマンスの低下が生じていることを示します。性能低下が報告されることがあるのは、システム・リカバリー条件の発生頻度が、マシンで事前設定されているしきい値を超えたときや、ユニット削除が起きたときです。性能低下報告能力の有無とその範囲は、モデルによって異なります。

性能低下は、抑制可能条件です。この条件は、性能低下サブクラス・マスク・ビット (制御レジスター 14 のビット位置 37) によりマスクされます。

警告

ビット 8 (W) が 1 のときは、システムのどこかで損傷が起きそうな状態にあること (例えば、電源障害や冷却能力低下が起きようとしている) を示します。警告条件が認識されるかどうかは、モデルによって異なります。

割り込み要求が受理される前に、差し迫っている損傷の原因となっている条件が取り除かれた場合 (例えば、電源が回復した場合) は、その要求は保留されず、割り込みは起きません。逆に言えば、その要求は割り込みによって消去されないため、その条件が持続する場合は、同じ条件の結果として複数の割り込みが起きることがあります。

警告は、抑制可能条件です。この条件は、警告サブクラス・マスク・ビット (制御レジスター 14 のビット位置 39) によりマスクされます。

チャンネル報告保留

ビット 9 (CP) が 1 のときは、1 つ以上のチャンネル報告ワードから成るチャンネル報告が保留されていることを示します。これらのチャンネル報告ワードの内容は、誤動作の影響、および分析の結果や行われたアクションについての詳細を示しています。チャンネル報告が保留されるのは、以下のいずれかの条件が起きたときです。

1. チャンネル・サブシステム・リカバリーが完了した。チャンネル・サブシステム・リカバリーは、あらかじめプログラムに通知することなく開始されていることもあり、また、すでにプログラムに報告されてい

る条件の結果として開始されていることもあります。

2. RESET CHANNEL PATH に指定されている機能が完了した。

チャンネル報告は、他の条件下でも保留されることがあります。

チャンネル報告を形成するチャンネル報告ワードは、STORE CHANNEL REPORT WORD を実行することにより、一度に 1 つずつ消去されます。この命令については、第14章、『入出力命令』で説明します。

チャンネル・サブシステム損傷が報告されたときは、ビット 9 は無意味です。

チャンネル報告保留は、浮動抑制可能条件です。この条件は、チャンネル報告保留サブクラス・マスク・ビット (制御レジスター 14 のビット位置 35) によりマスクされません。

サービス・プロセッサ損傷

ビット 10 (SP) が 1 のときは、サービス・プロセッサに損傷が起きたことを示します。サービス・プロセッサ損傷は、構成内のすべての CPU で保留されることもあり、各 CPU ごとに別々に検出されることもあります。サービス・プロセッサ損傷の報告の有無とその範囲は、モデルによって異なります。

サービス・プロセッサ損傷は抑制可能条件であり、サブクラス・マスク・ビットはありません。

チャンネル・サブシステム損傷

ビット 11 (CK) が 1 のときは、チャンネル・サブシステムでエラーまたは誤動作が起きたか、あるいはチャンネル・サブシステムがチェック停止状態にあることを示します。チャンネル・サブシステムがチェック停止状態になるのは、チャンネル・サブシステムの処理が続行不能なほど重大な誤動作が起きたとき、またはチャンネル・サブシステムの電源が失われた場合です。

チャンネル・サブシステム損傷は浮動抑制可能条件であり、サブクラス・マスク・ビットはありません。

サブクラス修飾子

マシン・チェック割り込みコードのビット 14 (B)、34 (DA)、および 44 (AP) は、サブクラス・ビットに対する修飾子の役割を果たします。

バックアップ

ビット 14 (B) が 1 のときは、割り込み点がエラー地点の前のチェックポイントにあることを示します。このビットが意味を持つのは、命令処理損傷ビット (ビット 1) も 1 に設定されている場合のみです。バックアップ条件を示す能力の有無とその範囲は、モデルによって異なります。

遅延アクセス例外

ビット 34 (DA) が 1 のときは、以前のアクセス例外テストではアクセス例外は検出されなかったか、DAT を使用した記憶機構アクセス時にアクセス例外が検出されたことを示します。

マシン・チェック割り込みコードのビット 34 は命令処理損傷 (ビット 1) の修飾子であり、ビット 1 が 1 の場合のみ意味を持ちます。ビット 1 が 0 のときは、ビット 34 は無意味です。遅延アクセス例外の報告の有無とその範囲は、モデルによって異なります。

プログラミング上の注意: 遅延アクセス例外の発生は、通常、プログラムが DAT テーブルを更新するために使用している手順が正しくないことを示しています。

補助報告

ビット 44 (AP) が 1 のときは、システム・コンポーネントの誤動作が起きたこと、そして、その誤動作は、以前に認識されているか、または CPU やサブチャネルなどの複数のシステム・エレメントに影響を与えたものであることを示します。誤動作が複数のエレメントの活動に影響を与えている場合は、影響を受けたエレメントのうち 1 つを除くすべてについて補助報告条件が認識されます。このビットが 0 のときは、システム・コンポーネントのこの誤動作が、以前に認識されているものではないことを示します。このビットは、マシン・チェック割り込みコードまたは外部損傷コードがどの条件を示している場合でも、意味を持ちます。

モデルによっては、補助報告条件を認識する能力がない場合や、必ずしもすべてのシステム誤動作についてこの条件が認識されない場合があります。補助報告の認識能力がないときは、ビット 44 は 0 に設定されます。

同期マシン・チェック割り込み条件

マシン・チェック割り込みコードの命令処理損傷ビットとバックアップ・ビット (ビット 1 と 14) は、組み合わせに応じて 2 つの条件を示します。

ビット	ビット	条件の名称
1	14	
1	0	処理損傷
1	1	処理バックアップ

処理バックアップ

処理バックアップ条件は、割り込み点がエラーの地点 (1 つまたは複数) より前であることを示します。これは、無効化緊急条件です。マシン・チェック割り込みコードの他のすべての CPU 関連損傷サブクラスおよび修飾子が 0 であり、かつ、CPU 状況に関連した所定の妥当性ビットが有効であることが示されている場合は、マシンは誤動作より前のチェックポイントに正しく戻っており、CPU にはまだ損傷は起きていません。

何も損傷が起きていない状態にあるためには、以下のサブクラスビットが 0 でなければなりません。

MCIC

ビット	名称
0	システム損傷
4	タイミング機構損傷

何も損傷が起きていない状態にあるためには、遅延アクセス例外サブクラス修飾子ビット (MCIC のビット 34) が 0 でなければなりません。

何も損傷が起きていない状態にあるためには、マシン・チェック割り込みコードの以下の妥当性ビットが 1 でなければなりません。

MCIC

ビット	名称
20	PSW MWP ビット
21	PSW のマスクとキー
22	PSW のプログラム・マスクと条件コード
23	PSW の命令アドレス
27	浮動小数点レジスター
28	汎用レジスター
29	制御レジスター
31	記憶機構の論理的妥当性 (現在のチェックポイント間隔内の結果フィールド)
33	アクセス・レジスター
42	TOD プログラムブル・レジスター
43	浮動小数点制御レジスター
46	CPU タイマー
47	クロック・コンパレーター

プログラミング上の注意: CPU 操作の続行中に誤動作または障害が起きたことを示すには、システム・リカバリではなく処理バックアップ条件が報告されます。こ

の場合は、誤動作は回避されていないため、命令の処理が続行されていたとすれば、損傷が起きています。

処理損傷

処理損傷条件は、CPU の命令処理に損傷が起きたことを示します。割り込み点は、一部またはすべての損傷点より前にあります。処理損傷は中止緊急条件であり、したがって、結果フィールドの内容は予測不能な場合がありますが、それでも有効として示されることもあります。

処理損傷には、プログラム・イベント記録、モニター呼び出し、トレース、アクセス・レジスター変換、および動的アドレス変換における誤動作などが含まれます。処理損傷が起きたときは、監視プログラム呼び出し割り込み条件およびプログラム割り込み条件はすべて破棄されます。ただし、これらの割り込みに関する旧 PSW および割り込みコード記憶位置の内容は、予測不能な値に設定されることがあります。

記憶機構エラー

マシン・チェック割り込みコードのビット 16-18 は、主記憶機構内で検出された無効 CBC または近似的有効 CBC、あるいは記憶キー内の無効 CBC を示すために使用されます。ビット 17 と同時に、ビット 19 (記憶機構性能低下) が示されることもあります。故障記憶機構アドレス・フィールドが有効として示されている場合は、そのフィールドの内容は、記憶機構検査ブロック内のエラーのある位置を示すか、または、訂正されていない記憶キー・エラーの場合は、記憶キーに関連したブロック内の位置を示します。ビット 32 (間接記憶機構エラー) が 1 に設定されているときは、故障記憶機構アドレスが指している記憶位置がエラーの本来の発生源ではないことを意味します。

検出されたエラーは結果に影響を与えていないこともあるため、未訂正記憶機構エラー・ビットおよび未訂正記憶キー・エラー・ビットは、それだけでは損傷の発生を示すことにはなりません。無効 CBC の影響を受けた構成内の部分は、マシン・チェック割り込みコードのサブクラス・フィールドに示されます。

チャンネル・プログラムについて検出されたエラーが入出力エラー条件として示される場合は、そのエラーはシステム・リカバリーとしても報告されることがあります。記憶機構内または記憶キー内で起きた CBC エラーのうち、CPU プログラム用として事前取り出しされたデータまたは未使用のデータについて検出されたエラーは、モデルによって、報告される場合とされない場合があります。

未訂正の記憶機構エラー

ビット 16 (SE) が 1 のときは、主記憶機構内の検査ブロックの 1 つに無効 CBC が含まれており、その誤った情報を訂正できなかったことを示します。主記憶機構内のその検査ブロックの内容は、変更されていません。報告された記憶位置は、この CPU または他の CPU、あるいはチャンネル・プログラム用としてアクセスまたは事前取り出しされたか、モデルによって異なるアクセスの結果としてアクセスされた可能性があります。

訂正済みの記憶機構エラー

ビット 17 (SC) が 1 のときは、主記憶機構内の検査ブロックの 1 つに近似的有効 CBC が含まれており、その誤った情報が訂正されたことを示します。モデルによっては、主記憶機構内のこの検査ブロックの内容が、有効な CBC に復元されていることもされていないこともあります。報告された記憶位置は、この CPU または他の CPU、あるいはチャンネル・プログラム用としてアクセスまたは事前取り出しされたか、モデルによって異なるアクセスの結果としてアクセスされた可能性があります。記憶機構エラー訂正機能の有無とその範囲は、モデルによって異なります。この指示に伴って、ビット 19 (DS) に記憶機構の性能低下が示されることもあり、示されないこともあります。

未訂正の記憶キー・エラー

ビット 18 (KE) が 1 のときは、無効な CBC を含む記憶キーがあり、その誤った情報を訂正できなかったことを示します。この記憶キー内の検査ブロックの内容は、変更されていません。この記憶キーは、この CPU または他の CPU、あるいはチャンネル・プログラム用としてアクセスまたは事前取り出しされたか、モデルによって異なるアクセスの結果としてアクセスされた可能性があります。

記憶機構性能低下

ビット 19 (DS) が 1 のときは、故障記憶機構アドレスに示されている 4K バイト・ブロックについて、リカバリー特性の性能低下が起きたことを示します。

記憶機構性能低下は、関連の記憶機構エラーは訂正されたが、該当の記憶ブロック (またはそれに関連したキー) に関連した固定的障害があるため、訂正プロセスに相当の時間がかかること、そして、そのブロック内でさらに新たなエラーが起きた場合は、そのエラーは訂正不能かまたは検出されないままになる可能性があることを示します。つまり、このビットは、できるだけこの記憶ブロックを使用しない方がよいということを示しています。

この記憶機構性能低下の指示が意味を持つのは、故障記憶機構アドレス妥当性 (MCIC のビット 24) も 1 である場合のみです。記憶機構性能低下の報告の有無とその範囲は、モデルによって異なります。

プログラミング上の注意: 通常、記憶機構性能低下はシステム・リカバリーと一緒に報告されるので、記憶機構性能低下が示されるようにするには、リカバリー・サブクラス・マスク (制御レジスター 14 のビット 36) を 1 に設定しておく必要があります。

間接記憶機構エラー

ビット 32 (IE) が 1 のときは、故障記憶機構アドレスが指している物理主記憶位置が、エラーの本来の発生源ではないことを示します。つまり、このエラーは、記憶機構階層の別のレベルで発生したものであり、それが記憶機構階層の現在の物理記憶機構部分に波及しています。ビット 32 が意味を持つのは、マシン・チェック割り込みコードのビット 16 (未訂正記憶機構エラー) またはビット 18 (未訂正記憶キー・エラー) が 1 のときだけです。ビット 16 と 18 が両方とも 0 のときは、ビット 32 は無意味です。

記憶機構階層の外部で発生したエラーの場合は、格納の試みは拒否され、該当のエラー指示が提示されます。記憶機構階層の内部ででの暗黙的な情報移動中にエラーが検出されたときは、そのアクションは拒否されず、この方法で報告されます。これは、移動は非同期で行われていることもあり、また、まったく無関係の情報にアクセスしようとした結果開始されていることもあるからです。記憶機構階層の 1 つの部分から別の部分への情報の暗黙移動中に、異動元の内容にエラーが検出された場合、移動先の記憶位置に関連した検査ブロック内に特別な無効 CBC を入れられ、結果としてそのエラーが移動先の領域に保存されてしまうことがあります。このように波及したエラーが後で検出された場合、それは間接記憶機構エラーとして報告されます。この種のエラーは、入出力プロセッサまたは CPU に関連したキャッシュ内で発生していることもあり、また、記憶機構階層のある部分から別の部分へのデータの転送中に、データ・パス障害の結果として発生していることもあります。さらに、波及したエラーは、記憶機構再構成の結果として、記憶機構のある物理部分から別の物理部部へのデータの移動中に、発生していることもあります。

間接記憶機構エラーの報告の有無とその範囲は、モデルによって異なります。

プログラミング上の注意: 記憶機構エラーが報告された後でとるべきアクションについては、第10章、『制御命令』の TEST BLOCK の『プログラミング上の注意』の項を参照してください。

マシン・チェック割り込みコードの妥当性ビット

マシン・チェック割り込みコードのビット 20-24、26-29、31、33、42、43、46、および 47 は、妥当性ビットです。これらのビットは、それぞれ記憶機構内の特定フィールドの妥当性を示します。記憶機構論理妥当性ビット (ビット 31) を除き、どのビットも、マシン・チェック割り込み時に格納される 1 つのフィールドに関連しています。妥当性ビットが 1 のときは、示されている割り込み点に関して、そのビットに対応するフィールドに入れられた保管値が有効であり、そのデータが格納されるときにエラーが検出されなかったことを示します。

妥当性ビットが 0 のときは、元の情報に誤りがあったか、元の情報に無効な CBC が含まれていたか、情報を格納するときに新たな誤動作が検出されたか、あるいは、情報がまったくまたは部分的に格納されなかったことを示します。マシンは、情報が予測不能であったとしても、可能な限り、記憶機構フィールドに有効な CBC を入れて、新たなマシン・チェックが起きる可能性を抑えようとしています。

浮動小数点レジスター、汎用レジスター、制御レジスター、アクセス・レジスター、TOD プログラマブル・レジスター、浮動小数点制御レジスター、CPU タイマー、およびクロック・コンパレーターに関する妥当性ビットは、それぞれに対応する保管域に入れられた保管値を妥当性を示します。マシン・チェック割り込みの後のこれらのレジスター内の情報は、保管域に正しい値が入れられ、該当の妥当性ビットが 1 に設定されていたとしても、必ずしも正しいとは限りません。これらのレジスターが使用されるかどうか、および、制御レジスター、浮動小数点制御レジスター、TOD プログラマブル・レジスター、CPU タイマー、およびクロック・コンパレーターに関連する機能がどのように働くかは、これらのレジスターが有効化されるまでは予測不能です。(11-9 ページの『レジスター内の無効 CBC』を参照してください。)

PSW-MWP の妥当性

ビット 20 (WP) が 1 のときは、マシン・チェック旧 PSW のビット 12-15 が正しいことを示します。

PSW のマスクとキーの妥当性

ビット 21 (MS) が 1 のときは、マシン・チェック旧 PSW のシステム・マスク、PSW キー、およびその他の各種ビットが正しいことを示します。このビットは、特に、PSW のビット 0-11、16、17、24-30、および 33-63 を対象としています。

PSW プログラム・マスクと条件コードの妥当性

ビット 22 (PM) が 1 のときは、マシン・チェック旧 PSW のプログラム・マスクおよび条件コードが正しいことを示します。

PSW 命令アドレスの妥当性

ビット 23 (IA) が 1 のときは、マシン・チェック旧 PSW のアドレッシング・モード・ビットと命令アドレス・ビット (ビット 31、32、および 64-127) が正しいことを示します。

故障記憶機構アドレスの妥当性

ビット 24 (FA) が 1 のときは、未訂正記憶機構エラー条件、未訂正記憶キー・エラー条件、または訂正済み記憶機構エラーが起きた後で、実記憶位置 248-255 に正しい故障記憶機構アドレスが格納されたことを示します。故障記憶機構位置を識別する機能の有無とその範囲は、モデルによって異なります。この種のエラーが報告されていない場合、つまり、マシン・チェック割り込みコードのビット 16-18 が 0 である場合は、故障記憶機構アドレスは、有効と示されていたとしても、無意味です。

外部損傷コードの妥当性

ビット 26 (EC) が 1 で、ビット 5 (外部損傷) も 1 であるときは、実記憶位置 244 にあるワードに有効な外部損傷コードが格納されたことを示します。ビット 5 が 0 のときは、ビット 26 は無意味です。

浮動小数点レジスタの妥当性

ビット 27 (FP) が 1 のときは、実記憶位置 4608-4735 にある浮動小数点レジスタ保管域の内容が、割り込み点での浮動小数点レジスタの正しい状態を反映していることを示します。

汎用レジスタの妥当性

ビット 28 (GR) が 1 のときは、実記憶位置 4736-4863 にある汎用レジスタ保管域の内容が、割り込み点での汎用レジスタの正しい状態を反映していることを示します。

制御レジスタの妥当性

ビット 29 (CR) が 1 のときは、実記憶位置 4992-5119 にある制御レジスタ保管域の内容が、割り込み点での制御レジスタの正しい状態を反映していることを示します。

記憶機構の論理的妥当性

ビット 31 (ST) が 1 のときは、実行中の命令により内容が変更される記憶位置に、割り込みとの関係において正しい情報が含まれていることを示します。つまり、割り込み点より前の格納はすべて完了し、割り込み点より後の格納 (存在する場合) はすべて抑止されます。無効な CBC が原因で割り込み点より前の格納が抑止された場合は、その無効 CBC が無効として保存されていれば、記憶機構論理妥当性ビットは 1 として示されることがあります。

命令処理損傷が示され、処理バックアップは示されていない場合は、記憶機構論理妥当性ビットは無意味です。

記憶機構論理妥当性は、命令処理活動を反映するのみであり、入出力操作の結果や、旧 PSW およびその他の割り込み情報の格納結果として生じる、記憶機構の状態のエラーを反映するものではありません。

アクセス・レジスタの妥当性

ビット 33 (AR) が 1 のときは、実記憶位置 4928-4991 にあるアクセス・レジスタ保管域の内容が、割り込み点におけるアクセス・レジスタの正しい状態を反映していることを示します。

TOD プログラマブル・レジスタの妥当性

ビット 42 (PR) が 1 のときは、実記憶位置 4900-4903 にある TOD プログラマブル・レジスタ保管域の内容が、割り込み点での TOD プログラマブル・レジスタの正しい状態を反映していることを示します。

浮動小数点レジスタの妥当性

ビット 43 (FC) が 1 のときは、実記憶位置 4892-4895 にある浮動小数点制御レジスタ保管域の内容が、割り込み点での浮動小数点制御レジスタの正しい状態を反映していることを示します。

CPU タイマーの妥当性

ビット 46 (CT) が 1 のときは、CPU タイマーにエラーがないこと、および、実記憶位置 4904-4911 にある CPU タイマー保管域の内容が、割り込みが起きた時点での CPU タイマーの正しい状態を反映していることを示します。

クロック・コンパレーターの妥当性

ビット 47 (CC) が 1 のときは、クロック・コンパレーターにエラーがないこと、および、実記憶位置 4913-4919 にあるクロック・コンパレーター保管域の内容が、割り込みが起きた時点でのクロック・コンパレーターの正しい状態を反映していること、および、実記憶位置 4912 に 0 が格納されていることを示します。

プログラミング上の注意: マシン・チェック条件により生じた損傷の範囲を判別するには、妥当性ビットを、サブクラス・ビットおよびバックアップ・ビットと組み合わせる必要があります。下記事項のすべてが真であれば、システムには何も損傷が起きていません。

- 4つの PSW 妥当性ビット、6つのレジスター妥当性ビット、2つのタイミング機構妥当性ビット、および記憶機構論理妥当性ビットが、すべて 1 であること。
- サブクラス・ビット 0、4、5、10、および 11 がすべて 0 であること。
- 命令処理損傷ビットが 0 であること。または、このビットが 1 のときは、バックアップ・ビットも 1 であること。
- 遅延アクセス例外ビットが 0 であること。

マシン・チェック拡張割り込み情報

場合によっては、マシン・チェック割り込みの一環として、記憶機構内に割り当てられている固定域に拡張割り込み情報が入れます。レジスター保管域には、CPU に関連したレジスターの内容が入れます。外部損傷の場合は、一部のモデルでは、外部損傷コードを格納することにより追加情報が提供されます。未訂正記憶機構エラー、訂正済み記憶機構エラー、または未訂正記憶キー・エラーが示された場合は、故障記憶機構アドレスが保管されます。

これらのフィールドには、それぞれ、マシン・チェック割り込みコード内の妥当性ビットが 1 つずつ関連付けられています。何かの理由で、マシンがこれらのフィールドに適切な情報を格納できない場合は、そのフィールドに対応する妥当性ビットが 0 に設定されます。

レジスター保管域

マシン・チェック割り込みの一環として、プレフィックス・レジスターと TOD クロックを除く CPU レジスターの現在の内容が、記憶機構内に割り当てられている 8 つのレジスター保管域に格納されます。これらの保管域には、それぞれ、マシン・チェック割り込みコード内の妥当性ビットが 1 つずつ関連付けられています。何かの理由で、マシンがこれらのフィールドに適切な情報を格納できない場合は、そのフィールドに対応する妥当性ビットが 0 に設定されます。

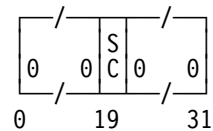
レジスターと、マシン・チェック割り込み時に各レジスターの内容が保管される実記憶位置には、次の 8 つの組み合わせがあります。

記憶位置	レジスター
4608-4735	浮動小数点レジスター 0-15
4736-4863	汎用レジスター 0-15
4892-4895	浮動小数点制御レジスター
4900-4903	TOD プログラマブル・レジスター
4904-4911	CPU タイマー
4913-4919	クロック・コンパレーター
4928-4991	アクセス・レジスター 0-15
4992-5119	制御レジスター 0-15

外部損傷コード

実記憶位置 244 にあるワードは、外部損傷コードです。このフィールドは、インプリメント済みで有効として示されているときは、外部損傷の原因を示します。このフィールドが有効と見なされるのは、外部損傷ビットと外部損傷コード妥当性ビット (マシン・チェック割り込みコードのビット 5 と 26) が、両方とも 1 の場合のみです。外部損傷コードの報告の有無とその範囲は、モデルによって異なります。

外部損傷コードのフォーマットは以下のとおりです。



ETR 同期検査 (SC): ビット 19 が 1 のときは、ビット 32 から、走行中のクロックの増分された最右端ビットまでが、ETR の同じビットと同期していないことを示します。

割り込みが起きる前に、この条件が複数回発生しても、この条件が生成されるのは 1 回だけです。この条件は、構成内のすべての CPU について生成され、1 つの CPU についての条件は、その CPU で割り込みが行われた時点でクリアされます。

予約済み: ビット 0-7、10-16、および 21-31 は将来の拡張に備えて予約されており、常に 0 に設定されます。

故障記憶機構アドレス

マシン・チェック割り込みコードに、未訂正記憶機構エラー、訂正済み記憶機構エラー、または未訂正記憶キー・エラーが示されているときは、それに関連したアドレス (故障記憶機構アドレスと呼ばれる) が、実記憶位置 248-255 に格納されます。このフィールドが有効と見なされるのは、故障記憶機構アドレス妥当性ビット

(マシン・チェック割り込みコードのビット 24) が 1 の場合のみです。

記憶機構エラーが起きたときは、故障記憶機構アドレスは、検査ブロック内のいずれかのバイトを指していることがあります。未訂正の記憶キー・エラーの場合は、故障記憶機構アドレスは、エラーのある記憶キーに関連した記憶機構ブロック内のいずれかのアドレスを指していることがあります。割り込みの前の複数の位置でエラーが検出された場合は、故障記憶機構アドレスは、障害位置のいずれかを指していることがあります。格納されるアドレスは絶対アドレスです。つまり、格納される値は、動的アドレス変換およびプレフィックシングが適用された後で、記憶機構の参照に使用されるアドレスです。

マシン・チェック条件の取り扱い

浮動割り込み条件

マルチプロセッシング構成内のどの CPU に対しても使用可能にされる割り込み条件を、浮動割り込み条件と呼びます。この割り込みを最初に受け入れた CPU がその割り込み条件を消去するので、構成内の他の CPU は、その割り込み条件を使用できなくなります。

浮動割り込み条件には、サービス信号外部割り込み条件と入出力割り込み条件が含まれます。チャンネル報告保留およびチャンネル・サブシステム損傷の 2 つのマシン・チェック割り込み条件も、浮動割り込み条件です。モデルによっては、システム・リカバリーと警告に関連した一部のマシン・チェック割り込み条件も、浮動割り込み条件です。

浮動割り込みは、構成内の CPU のうち、該当の割り込み条件に対して使用可能にされていて、その割り込みを受け入れることのできる最初の CPU に提示されます。チェック停止状態にある CPU、無効なプレフィックスを持っている CPU、早期認識されるタイプの PSW フォーマット・エラーが原因で終わりのない割り込みストリングを実行している CPU、または停止状態にある CPU は、割り込みを受け入れることはできません。ただし、速度制御が命令ステップに設定されている CPU は、開始キーが活動化されると、割り込みを受け入れることができます。

プログラミング上の注意: マルチプロセッシング構成において、ある CPU がチェック停止状態になったときは、他の CPU 上のプログラムは、その障害のある CPU に対して浮動割り込みが報告された後で失われているかどうかを、判別することができます。これを達成

するには、割り込みプログラムは、割り込みが処理された後で (または割り込みが後で処理するために別の領域に移された後で)、旧 PSW と割り込みコードを含む実記憶位置に 0 を入れる必要があります。ある CPU がチェック停止状態になった後で、他の CPU 上のプログラムは、障害のある CPU の旧 PSW および割り込みコードの記憶位置を検査することができます。旧 PSW または割り込みコード内の 0 以外の値は、CPU が割り込まれたが、プログラムがその割り込みの処理を完了しなかったことを示します。

浮動マシン・チェック割り込み条件

浮動マシン・チェック割り込み条件をリセットするには、オペレーター機能を使用して手動でリセットを開始する必要があります。浮動マシン・チェック割り込みの完了を妨げるマシン・チェックが起きたときは、その割り込み条件は浮動割り込み条件とは見なされなくなり、システム損傷が示されます。

浮動入出力割り込み

浮動入出力割り込みを求める入出力割り込みを提示する過程で、チャンネル・プログラムがマシン誤動作を検出した場合は、チャンネル報告保留またはチャンネル・サブシステム損傷が報告されます。浮動入出力割り込みを受け入れる過程で、CPU がマシン誤動作を検出した場合は、システム損傷が報告されます。

マシン・チェック・マスキング

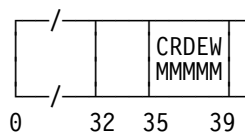
すべてのマシン・チェック割り込みは、マシン・チェック・マスク (PSW のビット 13) により制御されます。さらに、一部のマシン・チェック条件は、制御レジスター 14 の中のサブクラス・マスクにより制御されます。

緊急マシン・チェック条件 (システム損傷および命令処理損傷) は、マシン・チェック・マスク (PSW のビット 13) のみにより制御されます。PSW のビット 13 が 1 のときは、緊急条件が生じるとマシン・チェック割り込みが起きます。PSW のビット 13 が 0 のときに緊急マシン・チェック条件が起きると、CPU はチェック停止状態に入ります。

抑制可能マシン・チェック条件 (チャンネル・サブシステム損傷およびサービス・プロセッサ損傷を除く) は、マシン・チェック・マスク (PSW のビット 13) と、制御レジスター 14 の 5 つのサブクラス・マスク・ビットの両方により制御されます。PSW のビット 13 が 1 で、サブクラス・マスク・ビットの 1 つが 1 である場合は、それに対応する条件により、マシン・チェック割り込みが開始されます。サブクラス・マスク・ビットが

0 の場合は、それに対応する条件は、割り込みを開始せずに保留状態にします。ただし、CPU が使用可能にされている条件が原因でマシン・チェック割り込みが開始された場合は、その割り込みを開始した条件と一緒に、CPU が使用可能にされていない条件も提示されることがあります。その後で、提示された条件はすべて消去されます。

制御レジスター 14 には、特定の条件がマシン・チェック割り込みを引き起こすかどうかを指定するマスク・ビットが含まれています。このレジスターのフォーマットは以下のとおりです。



制御レジスター 14 のビット 35-39 は、抑制可能マシン・チェック条件用のサブクラス・マスクです。さらに、制御レジスター 14 のビット 32 は 1 に初期設定されますが、そうでないときは、マシンはこのビットを無視します。

プログラミング上の注意: PSW のビット 13 (マシン・チェック・マスク) が 0 に設定された状態でプログラムを稼働させるのは、極力避けるようにしてください。この状況で何らかの緊急マシン・チェック条件が認識されると、CPU がチェック停止状態になります。特に、PSW のビット 13 が 0 のときに、プログラムで入出力命令を実行すること、または入出力割り込みを許すことは、避けてください。

チャンネル報告保留サブクラス・マスク

制御レジスター 14 のビット 35 (CM) は、チャンネル報告保留割り込み条件を制御します。このビットは 0 に初期化されます。

リカバリー・サブクラス・マスク

制御レジスター 14 のビット 36 (RM) は、システム・リカバリー割り込み条件を制御します。このビットは 0 に初期化されます。

性能低下サブクラス・マスク

制御レジスター 14 のビット 37 (DM) は、性能低下割り込み条件を制御します。このビットは 0 に初期化されます。

外部損傷サブクラス・マスク

制御レジスター 14 のビット 38 (EM) は、タイミング機構損傷割り込み条件および外部損傷割り込み条件を制御します。このビットは 1 に初期化されます。

警告サブクラス・マスク

レジスター 14 のビット 39 (WM) は、警告割り込み条件を制御します。このビットは 0 に初期化されます。

マシン・チェック・ログアウト

一部のモデルでは、マシン・チェック割り込みの一環として、固定ログアウト域にモデル固有の情報が入れられることがあります。この領域は 16 バイトの長さで、実記憶位置 4864 から始まります。

マシン・チェックのマスクングの要約

11-25ページの図11-4 および 11-25ページの図11-5 に、マシン・チェックのマスクングの要約を示します。

マシン・チェック条件		サブクラス・マスク	CPU がサブクラスに対して使用不可にされているときのアクション
MCIC ビット	サブクラス		
0	システム損傷	-	チェック停止
1	命令処理損傷	-	チェック停止
2	システム・リカバリー	RM	Y
4	タイミング機構損傷	EM	P
5	外部損傷	EM	P
7	性能低下	DM	P
8	警告	WM	P
9	チャンネル報告保留	CM	P
10	サービス・プロセッサ損傷	-	P
11	チャンネル・サブシステム損傷	-	P

説明:

- この条件にはサブクラス・マスクはありません。
- P この指示は保留にされます。
- Y この指示は保留にされるか破棄されます。
- CM チャンネル報告保留サブクラス・マスク (CR14 のビット 35)。
- DM 性能低下サブクラス・マスク (CR14 のビット 37)。
- EM 外部損傷サブクラス・マスク (CR14 のビット 38)。
- RM リカバリー・サブクラス・マスク (CR14 のビット 36)。
- WM 警告サブクラス・マスク (CR14 のビット 39)。

図 11-4. マシン・チェック条件のマスク

ビットの名称	制御レジスタ 14 のビット位置	初期 CPU リセット時のビットの状態
チャンネル報告保留サブクラス・マスク	35	0
リカバリー・サブクラス・マスク	36	0
性能低下サブクラス・マスク	37	0
外部損傷サブクラス・マスク	38	1
警告サブクラス・マスク	39	0

図 11-5. マシン・チェック制御レジスタ・ビット

第12章 オペレーター機能

手動操作	12-1	手動標識ライト	12-3
基本オペレーター機能	12-1	電源制御	12-4
アドレス比較制御	12-1	速度制御	12-4
変更/表示制御	12-2	再始動キー	12-4
アーキテクチャー・モード標識ライト	12-2	開始キー	12-4
アーキテクチャー・モード選択制御	12-2	停止キー	12-4
チェック停止標識ライト	12-3	状況記憶キー	12-5
IML 制御	12-3	システム・リセット消去キー	12-5
割り込みキー	12-3	システム・リセット通常キー	12-5
ロード標識ライト	12-3	テスト標識ライト	12-5
ロード消去キー	12-3	TOD クロック制御	12-5
ロード通常キー	12-3	待ち状態ライト	12-6
ロード装置アドレス制御	12-3	マルチプロセッシング構成	12-6

手動操作

オペレーター機能は、マシンを手動で操作および制御するための機能です。これらの機能には、オペレーター/マシン間通信、マシン状況の表示、TOD クロック設定の制御、初期プログラム・ローディング、リセットのほか、通常のマシン操作にオペレーターが介入するためのその他の手動制御が含まれます。

モデルによっては、この章で説明していない追加のオペレーター機能を提供するものもあります。この種の機能には、例えば、装置内の特定のエラー条件を示したり、装置構成を変更したり、保守を容易にしたりする手段が含まれます。さらに、この章に挙げる制御機能について、この章では説明していない設定が可能な場合もあります。このような追加の機能および設定については、システム・ライブラリーの該当資料に説明が出ています。

ほとんどのモデルは、オペレーター機能に関連したコンソール装置を備えており、これは、オペレーターがプログラムと通信するための入出力装置として使用できます。また、コンソール装置は、この章で述べる機能の一部またはすべてをインプリメントするためにも使用できます。

オペレーター機能は、モデルによってさまざまなテクノロジーと構成を使用してインプリメントされています。一部のモデルでは、例えば複数のローカル操作ステーションまたはリモート操作ステーションにおいて、ある種のキー、制御機能、および標識が2組以上備わっていて、それらが同時に働くこともあります。

マシン誤動作が原因で、手動操作を定義どおりに正しく行うことができない場合は、CPU がチェック停止状態になるか、または、操作が失敗したことが別の方法でオペレーターに知らされることがあります。また、マシン誤動作の結果、マシン・チェック割り込み条件が認識されることもあります。

基本オペレーター機能

アドレス比較制御

アドレス比較制御機能を使用すると、指定したタイプの主記憶域参照で使用されたアドレスが、事前設定されているアドレスと一致したときに、CPU を停止させることができます。

アドレス比較制御の1つは、記憶域アドレスを比較するアドレスを設定するために使用します。

もう1つの制御では、次のように、アドレスが一致したときにとるべきアクションを指定するための位置を、少なくとも2つ設定することができます。

1. 通常位置に設定した場合は、アドレス比較操作は使用不可になります。
2. 停止位置に設定した場合は、アドレスが一致するとCPU は停止状態になります。この制御がこの位置に設定されているときは、テスト標識ライトが点灯しています。モデルと参照タイプにより、停止状態に入る前に、保留中の入出力割り込み、外部割り込

み、およびマシン・チェック割り込みが行われる場合と行われない場合があります。

第 3 のアドレス比較制御では、アドレス比較を行う記憶域参照のタイプを指定できます。モデルには、以下の位置の 1 つまたは幾つかを備えているものや、さらに別の位置を備えているものがあります。

1. 任意 (ANY) 位置に設定すると、すべての記憶域参照についてアドレス比較が行われます。
2. データ格納位置に設定すると、データの格納のために記憶域がアドレッシングされたときに、アドレス比較が行われます。
3. 入出力位置に設定すると、データの転送のため、またはチャンネル・コマンド・ワードまたは間接データ・アドレス・ワードの取り出しのために、チャンネル・サブシステムが記憶域をアドレッシングしたときに、アドレス比較が行われます。測定ブロック、割り込み応答ブロック、チャンネル・パス状況ワード、チャンネル報告ワード、サブチャンネル状況ワード、サブチャンネル情報ブロック、および操作要求ブロックに対する参照について、一致が示されるかどうかは、モデルによって異なります。
4. 命令アドレス位置に設定すると、命令の取り出しのために記憶域がアドレッシングされたときに、アドレス比較が行われます。アドレス設定の右端ビットは、無視される場合と無視されない場合があります。一致が示されるのは、命令の最初のバイトが選択した位置から取り出された場合のみです。
EXECUTE のターゲット命令を取り出すときにも一致が示されるかどうかは、モデルによって異なります。

モデルと参照タイプによっては、仮想アドレス、実アドレス、または絶対アドレスのすべてについてアドレス比較が行われることもあり、また、アドレスのタイプを指定できる場合もあります。

マルチプロセッシング構成において、アドレス設定が構成内の 1 つの CPU のみに適用されるのか、それともすべての CPU に適用されるのか、および、アドレスが一致したときに構成内の 1 つの CPU のみが停止するのか、それともすべての CPU が停止するのかは、モデルによって異なります。

変更/表示制御

オペレーター機能が提供する制御と手順を利用することにより、オペレーターは、記憶位置、記憶キー、レジスター (汎用、浮動小数点、浮動小数点制御、アクセス、および制御)、プレフィックス、および PSW の内容を、変更および表示することができます。

変更/表示操作を行うには、その前に CPU を停止状態にしておく必要があります。変更/表示操作中は、手動標識が一時的にオフになることがあり、また開始キーおよび再始動キーは機能しなくなることがあります。

変更/表示操作の対象とする記憶位置を選択するために使用するアドレスは、実アドレスです。論理アドレス、仮想アドレス、または絶対アドレスを指定できる場合もあります。

アーキテクチャー・モード標識ライト

アーキテクチャー・モード標識ライトは、最後のアーキテクチャー・モード選択操作と、最後の SIGNAL PROCESSOR のアーキテクチャー設定指令、またはモードを決定した最後のリセットにより選択された、操作のアーキテクチャー・モード (ESA/390 モード、z/Architecture モード、またはその他のモード) を示します。

アーキテクチャー・モード選択制御

アーキテクチャー・モード選択制御を使用すると、操作モードとして、ESA/390 アーキテクチャー・モードまたはその他のアーキテクチャー・モードを選択できます。(z/Architecture モードを選択するには、ESA/390 モードを選択してから、SIGNAL PROCESSOR のアーキテクチャー設定指令を使用します。)アーキテクチャー・モード選択は、モデルにより、IML 操作の一部となっている場合と、独立した操作となっている場合があります。

アーキテクチャー・モード選択プロセスでは、特定の構成内のすべての CPU および関連のチャンネル・サブシステム・コンポーネントが、同じアーキテクチャー・モードに設定されます。

チェック停止標識ライト

CPU がチェック停止状態にあるときは、チェック停止標識ライトが点灯しています。一般に、リセット操作を行うと、CPU はチェック停止状態から抜け出し、したがってこの標識ライトは消えます。また、チェック停止状態では、手動標識ライトも点灯していることがあります。

IML 制御

一部のモデルが備えている IML 制御は、初期マシン・ローディング (IML) を行って、ライセンス内部コードをマシンにロードします。IML 操作の一環として、ESA/390 モードまたはその他の操作モードを選択できる場合があります。

IML 操作が完了した後は、対象となった CPU、チャンネル・サブシステム、主記憶機構、およびオペレーター機能は、パワーオン・リセットが行われたときと同じ状態になりますが、TOD クロックの値と状態は変わりません。拡張記憶機構の内容は、モデルにより、0 にクリアされて有効な検査ブロック・コードが設定される場合と、変更されないままになる場合があります。

IML 制御は、電源がオンのときに有効です。

割り込みキー

割り込みキーが活動化されると、割り込みキーを示す外部割り込み条件が生成されます。(6-12ページの『割り込みキー』を参照してください。)

割り込みキーが働くのは、CPU が作動状態または停止状態にあるときです。CPU がロード状態のときに割り込みキーが働くかどうかは、モデルによって異なります。

ロード標識ライト

初期プログラム・ローディング中は、ロード標識ライトが点灯し、CPU がロード状態にあることを示します。特定の CPU について、ロード消去キーまたはロード通常キーが活動化され、それに該当する操作が開始されると、その CPU についてこの標識ライトが点灯します。そして、新 PSW が正常にロードされてしまうと、この

標識ライトは消えます。詳しくは、4-46ページの『初期プログラム・ローディング』を参照してください。

ロード消去キー

ロード消去キーを活動化すると、リセット操作が行われ、ロード装置アドレス制御で指定した入出力装置を使用して初期プログラム・ローディングが開始されます。構成に対して、クリア・リセットが行われます。詳しくは、4-40ページの『リセット』、および 4-46ページの『初期プログラム・ローディング』を参照してください。

ロード消去キーが働くのは、CPU が作動状態、停止状態、ロード状態、またはチェック停止状態にあるときです。

ロード通常キー

ロード通常キーを活動化すると、リセット操作が行われ、ロード装置アドレス制御で指定した入出力装置を使用して初期プログラム・ローディングが開始されます。最初に、ロード通常キーが活動化された CPU について CPU リセットが行われ、構成内の他のすべての CPU に CPU リセットが波及し、そして、構成のその他の部分についてはサブシステム・リセットが行われます。詳しくは、4-40ページの『リセット』、および 4-46ページの『初期プログラム・ローディング』を参照してください。

ロード通常キーが働くのは、CPU が作動状態、停止状態、ロード状態、またはチェック停止状態にあるときです。

ロード装置アドレス制御

ロード装置アドレス制御は、初期プログラム・ローディングに使用する装置番号を示す 4 つの 16 進数字を指定します。詳しくは、4-46ページの『初期プログラム・ローディング』を参照してください。

手動標識ライト

CPU が停止状態にあるときは、手動標識ライトが点灯しています。一部の機能および幾つかの手動制御は、CPU が停止状態にあるときのみ有効になります。

電源制御

電源制御は、電源をオンまたはオフに切り替えるために使用します。

CPU、記憶機構、チャンネル・サブシステム、オペレータ一機能、および入出力装置は、共通の制御を使用して全部の電源を一緒にオンまたはオフに切り替えることも、それぞれ単独に電源制御を行うこともできます。ある装置の電源をオンにすると、その装置はリセットされます。このリセット・シーケンスが行われるのは、明示的に指定するまでは、どの命令や入出力操作も行われないうようにするためです。電源制御により、電源を段階的にオンにすることもできますが、電源が完全にオンになるまでは、マシンは稼働可能になりません。

電源が完全にオンになると、IML 機能を持つモデルでは IML 操作が行われます。そして、パワーオン・リセットが開始されます (4-40ページの『リセット』を参照)。電源をオンにするときに操作のアーキテクチャー・モードを選択できるか、後でモード選択制御を使用してモードを変更する必要があるかは、モデルによって異なります。

速度制御

速度制御の設定により、開始機能の効果と命令の実行方法が決まります。

速度制御には、少なくとも 2 つの設定位置があります。通常の設定は、処理位置です。もう 1 つは、命令ステップ位置です。速度制御が処理位置に設定されているときに開始機能が実行された場合は、CPU は通常で動作を開始します。速度制御が命令ステップ位置に設定されており、待ち状態ビットが 0 のときは、1 つの命令 (割り込み可能命令の場合は 1 つの操作単位) が実行され、保留中のすべての可能な可能割り込みは、CPU が停止状態に戻る前に発生します。速度制御が命令ステップ位置に設定されており、待ち状態ビットが 1 のときは、命令はどれも実行されませんが、保留中のすべての可能な割り込みは、CPU が停止状態に戻る前に発生します。詳しくは、4-1ページの『停止、作動、ロード、およびチェック停止の各状態』を参照してください。

速度制御が処理位置に設定されていないときは、テスト標識ライトが点灯しています。

CPU が作動状態またはロード状態にあるときに速度制御の設定が変更された場合は、結果は予測不能です。

再始動キー

再始動キーを活動化すると、再始動割り込みが開始されます。(6-44ページの『再始動割り込み』を参照してください。)

再始動キーが働くのは、CPU が作動状態または停止状態にあるときです。CPU がチェック停止状態にあるときは、このキーは働きません。構成内のいずれかの CPU がロード状態にあるときに再始動キーが働くかどうかは、モデルによって異なります。

構成内のいずれかの CPU が活動化された場合の結果は、予測不能です。特に、いずれかの CPU がロード状態にあるときに、別の CPU が再始動割り込みを実行して作動状態に入った場合は、入出力命令、**SIGNAL PROCESSOR** 命令、および **INVALIDATE PAGE TABLE ENTRY** 命令は、本書に示されている定義どおりには機能しないことがあります。

開始キー

開始キーを活動化すると、CPU は開始機能を実行します。(4-1ページの『停止、作動、ロード、およびチェック停止の各状態』を参照してください。)

開始キーが働くのは、CPU が停止状態にあるときのみです。リセットにより停止状態になっているときは、結果は予測不能です。

停止キー

停止キーを活動化すると、CPU は停止機能を実行します。(4-1ページの『停止、作動、ロード、およびチェック停止の各状態』を参照してください。)

停止キーが働くのは、CPU が作動状態にあるときのみです。

操作上の注意: 以下の場合は、停止キーを活動化しても効果はありません。

- ある種のプログラム割り込みまたは外部割り込みのストリングがエンドレスに発生している場合。
- プレフィックス・レジスターに無効アドレスが含まれている場合。
- CPU がロード状態またはチェック停止状態にある。

状況記憶キー

状況記憶キーを活動化すると、状況記憶操作が開始されます。(4-46ページの『状況記憶』を参照してください。)

状況記憶キーが働くのは、CPU が停止状態にあるときのみです。

操作上の注意: 状況記憶操作とスタンドアロン・ダンプ・プログラムを一緒に使用することにより、主要なプログラム誤動作の分析を行うことができます。そのためには、以下のような手順が必要になります。

1. 停止キーまたはシステム・リセット通常キーを活動化する。
2. 状況記憶キーを活動化する。
3. ロード通常キーを活動化して、スタンドアロン・ダンプ・プログラムに入る。

ステップ 1 でシステム・リセット通常キーを活動化する必要があるのは、(1) 連続的な割り込みのストリングが起きているため、停止キーが機能しない場合、(2) プレフィックス・レジスターに無効なアドレスが含まれている場合、または、(3) CPU がチェック停止状態にある場合です。

システム・リセット消去キー

システム・リセット消去キーを活動化すると、構成に対してクリア・リセット操作が行われます。詳しくは、4-40ページの『リセット』を参照してください。

システム・リセット消去キーが働くのは、CPU が作動状態、停止状態、ロード状態、またはチェック停止状態にあるときです。

システム・リセット通常キー

システム・リセット通常キーを活動化すると、CPU リセット操作およびサブシステム・リセット操作が行われます。マルチプロセッシング構成では、CPU リセットは構成内のすべての CPU に波及します。詳しくは、第 4 章、『制御』の中の『リセット』のセクションを参照してください。

システム・リセット通常キーが働くのは、CPU が作動状態、停止状態、ロード状態、またはチェック停止状態にあるときです。

テスト標識ライト

操作または保守のための手動制御が、プログラムの通常の操作に影響を与えるような異常な位置にあるときは、テスト標識ライトが点灯しています。

アドレス比較制御を停止位置に設定するか、または速度制御を命令ステップ位置に設定すると、テスト標識ライトが点灯します。

テスト標識ライトは、DIAGNOSE の制御下にある 1 つ以上の診断機能が活動化されたとき、またはその他の異常条件が起きたときに、テスト標識ライトが点灯することがあります。

手動制御の異常設定の結果として、影響を受ける CPU のテスト標識ライトが点灯します。しかし、マルチプロセッシング構成では、他の CPU のテスト標識ライトが点灯していなくても、それらの CPU の操作も影響を受けることがあります。

操作上の注意: 手動制御が、例えば異常設定など、保守を目的とする設定のままになっていると、さまざまな問題が起きることがありますが、特に、間違っただマシン・チェックが示されたり、実際に起きたマシン誤動作が無視されたりすることがあります。さらに、命令の実行、チャンネル・サブシステムの動作、およびオペレーター制御機能や標識ライトの働きなど、マシンの動作の他の局面に影響を受け、マシンの動作が本書の記述に合わなくなることがあります。

TOD クロック制御

TOD クロック制御が活動化されていない場合、つまりこの制御が保護位置に設定されている場合は、SET CLOCK または DIAGNOSE 命令による TOD クロックの状態または値の変更が禁止され、したがって、その状態と値を無許可または不注意による変更から保護することができます。

TOD クロック制御が活動化されている場合、つまりこの制御が設定可能位置に設定されている場合は、SET CLOCK または DIAGNOSE によるクロックの状態または値の変更が許されます。この設定は一時的なもので、この制御は自動的に保護位置に戻ります。

TOD クロック制御が物理的に複数ある場合は、TOD クロックが保護されるのは、構成内のすべての TOD クロック制御が保護位置に設定されている場合に限られません。

待ち状態ライト

現 PSW 内の待ち状態ビットが 1 のときは、待ち状態ライトが点灯しています。待ち状態ライトの代わりに、待ち状態ビットの平均時間値を示す手段を備えているモデルもあります。

マルチプロセッシング構成

マルチプロセッシング構成では、各 CPU が、各種のキーと制御 (変更/表示、割り込み、速度、再始動、開始、停止、および状況記憶) をそれぞれ備えています。また、入出力操作を行うことのできる各 CPU には、ロード消去キー、ロード通常キー、およびロード装置アドレス制御があります。代替の方法として、必要とする CPU を選択するための制御と一緒に、一組の初期プロ

グラム・ローディング・キーおよび制御を使用することもできます。

マルチプロセッシング構成では、一組だけあればよいキーと制御が幾つかあります。それは、アドレス比較、IML、電源、システム・リセット・クリア、システム・リセット通常、および TOD クロックです。

チェック停止標識、手動標識、テスト標識、および待ち標識は、各 CPU ごとに備わっています。ロード標識はを備えているのは、入出力操作を行うことのできる CPU のみです。代わりに、複数の CPU 間で一組の標識を切り替えて使用できます。

マルチプロセッシング構成では、アーキテクチャー・モード標識は複数個ある必要はありません。

再構成が可能なシステムでは、各構成内に、それぞれ専用のキー、制御、および標識のセットが必要です。

第13章 入出力の概要

入出力 (I/O)	13-1	サブチャンネル番号	13-5
チャンネル・サブシステム	13-1	装置番号	13-5
サブチャンネル	13-2	装置 ID	13-5
入出力装置の接続	13-2	入出力操作の実行	13-5
チャンネル・パス	13-2	開始機能の開始	13-6
制御装置	13-3	パス管理	13-6
入出力装置	13-4	チャンネル・プログラムの実行	13-6
入出力アドレッシング	13-4	入出力操作の終結	13-7
チャンネル・パス ID	13-4	入出力割り込み	13-8

入出力 (I/O)

「入力」および「出力」という用語は、入出力装置と主記憶機構の間のデータの転送を意味します。この種の転送を含む操作を、入出力操作と呼びます。入出力操作を制御するために使用する各種の機構を、総称的にチャンネル・サブシステムと呼びます。(入出力装置とそれぞれの制御装置は、チャンネル・サブシステムに接続されます。)この章では、チャンネル・サブシステムの基本的なコンポーネントと操作について、簡単に説明します。

チャンネル・サブシステム

チャンネル・サブシステムは、入出力装置と主記憶機構の間の情報の流れを制御します。これにより、CPU は入出力装置と直接通信する作業から解放され、入出力処理と並行してデータ処理を進めることができるようになります。チャンネル・サブシステムは、入出力装置との間の情報の流れを管理するために、1 つ以上のチャンネル・パスを通信リンクとして使用します。また、チャンネル・サブシステムは、入出力処理の一環として、チャンネル・パスの可用性の検査、使用可能なチャンネル・パスの選択、および装置での入出力操作の実行開始などのパス管理操作も行います。

チャンネル・サブシステムにはサブチャンネルが含まれています。チャンネル・サブシステムからアクセスできる各入出力装置に、それぞれ専用のサブチャンネルが 1 つずつ用意されます。各サブチャンネルは、関連する入出力装置およびその装置とチャンネル・サブシステムの接続に関する情報を提供します。また、サブチャンネルは、入出力操作に関する情報、および関連入出力装置を必要とするその他の機能に関する情報も提供します。サブチャンネルは、チャンネル・サブシステムが、関連の入出力装置に関する情報を CPU に提供するための手段です。CPU は、入

出力命令を実行することによりこの情報を入手します。実際に用意されるサブチャンネルの数は、モデルと構成によって異なります。最大のアドレス可能数は、65,536 です。

入出力装置は、制御装置を介し、チャンネル・パスを経由してチャンネル・サブシステムに接続されます。制御装置は複数のチャンネル・パスを介してチャンネル・サブシステムに接続でき、また、1 つの入出力装置を複数の制御装置に接続することもできます。モデルと構成によって異なりますが、チャンネル・サブシステムは、最大 8 つの異なるチャンネル・パスを使用して 1 つの入出力装置にアクセスできます。1 つのチャンネル・サブシステムが提供するチャンネル・パスの合計数は、モデルと構成によって異なります。最大アドレス可能数は、256 です。

チャンネル・サブシステムのパフォーマンスは、その使用方法と、それがインプリメントされているシステム・モデルによって異なります。チャンネル・パスはそれぞれデータ転送能力が異なり、特定の速度でのみデータを転送するように設計されている入出力装置 (磁気テープ装置やディスク装置など) には、最低限そのデータ速度をサポートするチャンネル・パスを使用する必要があります。

チャンネル・サブシステムは、入出力操作を制御するための共通の機能を備えています。これらの機能が、特に入出力装置を制御するように設計された独立した自律型装置として提供されていれば、入出力操作は、CPU 内の活動と完全に並行して行われます。入出力操作の際にチャンネル・サブシステムが必要とする主記憶機構サイクルは、主記憶機構内の最終宛先記憶位置との間でデータと制御情報を送受信するためのサイクル、および、アドレス不能主記憶機構の一部としてサブチャンネルがインプリメントされている場合にチャンネル・サブシステムがサブチャンネルにアクセスするために必要となるサイクルです。CPU とチャンネル・サブシステムが同時に同じ主記

憶域を参照しようとしないう限り、これらのサイクルが CPU プログラムを遅らせることはありません。

サブチャネル

サブチャネルは、プログラムに対して装置の論理的な外観を提供するものであり、単一の入出力操作の持続に必要な情報が含まれています。サブチャネルを構成する内部記憶機構には、CCW アドレス、チャンネル・パス ID、入出力装置番号、カウント、状況標識、入出力割り込みサブクラス・コードのほか、パスの可用性、および保留中または実行中の機能に関する情報が含まれています。特定の入出力装置に関連したサブチャネルを指定した入出力命令が実行されると、その装置での入出力操作が開始されます。

個々の入出力装置には、1 チャンネル・サブシステムあたり 1 つのサブチャネル (インストール時に割り当てられているもの) を介してアクセス可能です。入出力装置には、物理的に識別可能なユニットの場合と、制御装置の内部に組み込まれている場合があります。例えば、一部のディスク装置では、データをリトリブするときに使用される各アクチュエーターが、それぞれ 1 つの入出力装置と見なされます。いずれの場合も、チャンネル・サブシステムから見て、入出力装置は 1 つのサブチャネルのみに関連付けられている 1 つのエンティティーであって、アクセスに使用するチャンネル・パスのタイプについて定義されている通信プロトコルを使用してチャンネル・サブシステムが行った選択に応答するものです。

一部のモデルでは、サブチャネルはブロック単位で提供されます。この種のモデルでは、接続されている装置の数より多くのサブチャネルが用意されていることがあります。用意はされているが装置が割り当てられていないサブチャネルは、チャンネル・サブシステムが何らかの機能を実行するためには使用されません。この種のサブチャネルについては、サブチャネル情報ブロック内の関連の装置番号有効ビットに 0 が保管されています。

チャンネル・サブシステムが提供するサブチャネルの数は、関連の入出力装置へのチャンネル・パスの数とは関係ありません。例えば、幾つかの代替チャンネル・パスを介してアクセス可能な装置であっても、1 つのサブチャネルで表されます。各サブチャネルのアドレスは、それぞれ 16 ビットの 2 進数サブチャネル番号で表されます。

START SUBCHANNEL の実行によってサブチャネルでの入出力処理が要求された後は、CPU は解放されて他の作業を実行できるようになります。チャンネル・サブシステムは、データの組み立てや分解を行い、入出力装置と主記憶機構間のデータ・バイトの転送を同期化します。この作業のために、チャンネル・サブシステムは、

主記憶機構内でのデータの宛先とソースを記述するアドレスおよびカウントを維持し、更新しています。同様に、入出力装置がプログラムに通知する必要のある信号を出したときは、チャンネル・サブシステムは、その信号を状況情報に変換してサブチャネルに格納し、プログラムはそこから情報を検索することができます。

入出力装置の接続

チャンネル・パス

チャンネル・サブシステムは、チャンネル・サブシステムと制御装置間のチャンネル・パスを使用して、入出力装置と通信します。チャンネル・サブシステムは、複数のチャンネル・パスを介して同じ制御装置にアクセスできます。同様に、チャンネル・サブシステムは、それぞれがチャンネル・サブシステムへの複数のチャンネル・パスを持つ複数の制御装置を介して、1 つの入出力装置にアクセスできます。

チャンネル・サブシステムは、複数のチャンネル・パスによりチャンネル・サブシステムに接続されている装置には、使用可能なチャンネル・パスをどれでも使用してアクセスできます。同様に、動的再接続機能を持ち、マルチパス・モードで稼働している装置は、一連の入出力操作を続けるためにチャンネル・サブシステムに論理的に再接続するときに、接続されているチャンネル・パスをどれでも任意に選択できるように、初期化することができます。

チャンネル・サブシステムには、複数のタイプのチャンネル・パスが含まれていることもあります。チャンネル・サブシステムが使用するチャンネル・パスのタイプには、例えば、ESCON 入出力インターフェース、FICON 入出力インターフェース、FICON 変換入出力インターフェース、IBM システム/360 およびシステム/370 入出力インターフェースなどがあります。「直列入出力インターフェース」という用語は、ESCON 入出力インターフェース、FICON 入出力インターフェース、および FICON 変換入出力インターフェースの総称として使用されています。「並列入出力インターフェース」という用語は、IBM システム/360 およびシステム/370 入出力インターフェースを表します。

ESCON 入出力インターフェースについては、システム・ライブラリー資料の「*IBM Enterprise Systems Architecture/390 ESCON I/O Interface*」(SA22-7202) で説明されています。FICON 入出力インターフェースについては、ANSI 規格文書「*Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)*」で説明されています。IBM システム/360 およびシステム/370 入出力インターフェースについては、システム・ライブラリー

資料の「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974)で説明されています。

入出力操作は、チャンネル・バスのタイプ、チャンネル・バスが提供する機能、および入出力装置に応じて、フレーム多重モード、バースト・モード、またはバイト多重モードの3モードのいずれかで行われます。

フレーム多重モードでは、入出力デバイスは、チャンネル・プログラムが実行されている間、論理的にチャンネル・バスに接続されたままの状態を維持することができます。フレーム多重モードで稼働できるチャンネル・バスの機能は、並行して稼働している複数の入出力装置間で共用することができます。このモードでは、1つの入出力操作を完了するために必要な情報を複数のフレームに分割して、他の入出力装置での入出力操作から発生するフレームと互いにインターリーブさせることができます。その間は、チャンネル・バスに複数の論理的に入出力装置が接続しているものと見なされます。

バースト・モードでは、入出力装置は1つのチャンネル・バスを占有し、1バーストの情報が転送されている間は、論理上そのチャンネル・バスに接続されたままになっています。バーストの転送中は、他の装置はそのチャンネル・バスを介して通信を行うことはできません。バーストは、数バイトのことも、1ブロックのデータのことも、関連の制御情報と状況情報を伴うブロックのシーケンスのことも(ブロック長は0のこともあります)、また、チャンネル・バスを占有する状況情報のこともあります。バースト・モードで稼働できるチャンネル・バスの機能は、並行して稼働している複数の入出力装置間で共用することができます。

チャンネル・バスには、バースト・モードでの操作において、例えば磁気テープの長いギャップを読み取る時のように、データ転送が存在しない状態を30秒程度容認するものもあります。データ転送が存在しない時間が所定の限度を超えると、装置誤動作が示されることがあります。

バイト多重モードでは、入出力装置が論理的にチャンネル・バスに接続されているのは短い間だけです。バイト多重モードで稼働できるチャンネル・バスの機能は、並行して稼働している複数の入出力装置間で共用することができます。このモードでは、すべての入出力操作が短い時間間隔に分割され、1つの時間間隔の期間中は1セグメントの情報のみがチャンネル・バスを介して転送されます。この期間中は、1つの入出力装置とそれに関連したサブチャンネルのみが、論理的にチャンネル・バスに接続されています。複数の入出力装置の並行操作に関連した時間間隔は、それらの装置からの要求に応答する形で順序付けられます。1つのサブチャンネルに関連したチャネ

ル・サブシステム機能が、1つの操作に対して制御機能を働かせるのは、1セグメントの情報を転送するために必要な時間だけです。セグメントを形成するのは、1バイトのデータ、何バイトかのデータ、装置からの状況報告、または、新しい操作を開始するために使用される制御シーケンスです。

一般に、高いデータ転送速度を必要とする装置はフレーム多重モードのチャンネル・バスと連動し、それより遅い装置はバースト・モードで働き、そして最も遅い装置はバイト多重モードで稼働します。一部の制御装置には、任意の操作モードを設定できる手動スイッチが備わっています。

並列入出力入出力インターフェース・タイプのチャンネル・バス上で発生する入出力操作は、そのチャンネル・バスと入出力装置が備えている機能に応じて、バースト・モードまたはバイト多重モードで行われます。一部のチャンネル・バスおよび制御装置には、パフォーマンスを高めるために、高速転送およびデータ・ストリーミングのための機構が備わっています。この2つの機構の説明については、システム・ライブラリー資料「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974)を参照してください。

直列入出力インターフェース・タイプのチャンネル・バス上で発生する入出力操作は、フレーム多重モードまたはバースト・モードで行うことができます。パフォーマンス向上のために、直列入出力インターフェースに接続される一部の制御装置には、プログラムで許されていれば装置検査状況を表示すると同時に、センス・データもプログラムに提供する機能が備わっています。(17-17ページの『並行センス』を参照してください。)

制御装置およびチャンネル・サブシステムによっては、サブチャンネルを介した装置へのアクセスが、1つのチャンネル・バス・タイプのみに制限される場合があります。

上記で述べたモードと機構は、チャンネル・バスを介して情報を転送するためのプロトコルと、伝送の速度に影響を与えるだけです。CPUやチャンネル・プログラムから見た、これらのプログラムの実行のしかたについては何の影響もありません。

制御装置

制御装置は、入出力装置を操作および制御するために必要な論理機能を提供し、各入出力装置の特性を、チャンネル・サブシステムが提供する標準形式の制御に合わせて調整します。

制御装置とチャンネル・サブシステム間の通信は、チャ

ネル・パスを介して行われます。制御装置は、チャンネル・サブシステムから制御信号を受け取り、チャンネル・パスを介したデータ転送のタイミングを制御し、入出力装置の状況を知らせる指示を送ります。

制御装置に接続されている入出力装置は、限定された特定の操作のみを行うように設計されていることも、多種の異なる操作を行うように設計されていることもあります。代表的な操作の 1 つとして、記録メディアおよび記録データの移動があります。入出力装置が操作を行うためには、その装置タイプに特有の詳細な信号シーケンスが必要です。制御装置は、チャンネル・サブシステムから受け取ったコマンドをデコードし、該当の装置タイプに合わせてそのコマンドを解釈し、操作を実行するために必要な信号シーケンスを準備します。

制御装置は、独立した形で設置することも、入出力装置、チャンネル・サブシステム、または CPU に、物理的または論理的に組み込むこともできます。ほとんどの電気機械的装置の場合、制御装置と入出力装置に必要な機構のタイプが異なるために、入出力装置と制御装置の間には明確に定義されたインターフェースが存在します。これらの電気機械的装置の場合、一時点でデータ転送に必要なのは、1 つの制御装置に接続されている装置のグループのうち 1 つの装置のみであることが多く（例えば、磁気テープ装置やディスク・アクセス機構など）、制御装置は複数の入出力装置間で共用されることとなります。これに対して、一部の電子入出力装置（例えばチャンネル間アダプター）では、制御装置は独自の識別を持っていない場合があります。

プログラマーの視点から見れば、制御装置が行うほとんどの機能は、入出力装置が行う機能と混然一体となっているように見えます。したがって、本書では、一般に制御装置の機能について特別に言及することはしません。つまり、入出力操作の実行については、入出力装置がチャンネル・サブシステムと直接通信しているものとして説明します。制御装置について言及するのは、制御装置が実行する特定の機能を特に強調して示すとき、または、複数の入出力装置が制御装置を共用する場合に、入出力操作の実行にどのような影響が現れるかを説明するときのみです。

入出力装置

入出力 (I/O) 装置は、外部記憶機構、データ処理システム間の通信手段、またはシステムとその環境の間の通信手段を提供します。入出力装置には、磁気テープ装置、直接アクセス記憶装置（ディスクなど）、ディスプレイ装置、タイプライター型キーボード、プリンター、テレプロセッシング装置、センサー・ベース装置などがあります。入出力装置は、物理的に独立した装置のこともあ

り、他の入出力装置との間で同一機器を共用することもあります。

プリンターや磁気テープ装置など、ほとんどのタイプの入出力装置は、外部メディアを使用し、また、この種の装置は物理的に区別および識別できます。ほかに、純粋に電子的な装置タイプもあり、この種の装置は物理的な記録メディアを直接取り扱うことはありません。例えば、チャンネル間アダプターは、2 つのチャンネル・パス間のデータ転送を行うためのものであり、この転送により、データが主記憶機構の外部にある物理記録メディアにまで到達することはありません。同様に、通信コントローラーは、データ処理システムとリモート・ステーションの間の情報の伝送を取り扱うことができますが、その入力と出力は伝送回線上の信号です。

最も単純な形態では、1 つの入出力装置が 1 つの制御装置に接続され、1 つのチャンネル・パスからのみアクセスされます。切り替え機構を使用すれば、個々の入出力装置を制御装置間で切り替える、または個々の制御装置をチャンネル・パス間で切り替えることにより、幾つかの入出力装置に複数のチャンネル・パスからアクセス可能にすることができます。このような切り替え機構は、入出力装置にアクセスするための複数のパスを提供します。このように、1 つの入出力装置への複数のチャンネル・パスを用意するのは、システム内でのパフォーマンスまたは入出力可用性、またはその両方を高めるためです。入出力装置への複数チャンネル・パスの管理は、チャンネル・サブシステムと入出力装置の制御下で行われますが、チャンネル・パスは、プログラムにより間接的に制御することができます。

入出力アドレッシング

チャンネル・サブシステムは、各種コンポーネントのアドレッシング要件を満たすために、4 つのタイプの入出力アドレッシング手段を提供します。それは、チャンネル・パス ID、サブチャンネル番号、入出力装置番号、そして、チャンネル・パス・タイプに依存するアドレスです（このアドレスはプログラムからは見えません）。

チャンネル・パス ID

チャンネル・パス ID (CHPID) は、システムにインストールされている各チャンネル・パスに割り当てられている、システム固有の 8 ビット値です。CHPID は、個々の物理チャンネル・パスを識別します。CHPID は、RESET CHANNEL PATH の第 2 オペランド・アドレスに指定され、リセットすべき物理チャンネル・パスを指示します。特定の入出力装置にアクセスするために使用できるチャンネル・パスは、STORE SUBCHANNEL の

実行時に、サブチャンネル情報ブロック (SCHIB) 内に、それぞれの CHPID で指定されます。CHPID は、オペレーター・メッセージの中で、特定のチャンネル・パスを識別する必要がある場合にも使用できます。1 つのシステム・モデルでのチャンネル・パスの最大数は、256 個です。チャンネル・パスと CHPID 割り当ての最大数は、システム・モデルによって異なります。

サブチャンネル番号

サブチャンネル番号は、サブチャンネルのアドレスを指定するために使用される、システム固有の 16 ビット値です。サブチャンネル・アドレスを指定する入出力命令には、CANCEL SUBCHANNEL、CLEAR SUBCHANNEL、HALT SUBCHANNEL、MODIFY SUBCHANNEL、RESUME SUBCHANNEL、START SUBCHANNEL、STORE SUBCHANNEL、TEST SUBCHANNEL の 8 つがあります。チャンネル・サブシステムからアクセスできる各入出力装置には、インストール時にそれぞれ専用のサブチャンネルが割り当てられます。プログラムは、特定の入出力装置に割り当てられているサブチャンネルを指示することによって、その入出力装置に関連したすべての入出力機能を指定します。サブチャンネルには、必ず、1 つの連続した番号範囲の中から、サブチャンネル番号が割り当てられます。最も番号の小さいサブチャンネルは、サブチャンネル 0 です。チャンネル・サブシステムの最も番号の大きいサブチャンネルは、用意されているサブチャンネル数から 1 を引いたサブチャンネル番号を持つサブチャンネルです。用意できるサブチャンネルの最大数は、65,536 です。通常、サブチャンネル番号は、CPU プログラムとチャンネル・サブシステム間の通信に使用されるだけです。

装置番号

入出力装置が割り当てられている各サブチャンネルには、装置番号と呼ばれるシステム固有のパラメーターも含まれています。装置番号は、装置がサブチャンネルに割り当てられる時点で、サブチャンネルのパラメーターの 1 つとして割り当てられる 16 ビット値です。

装置番号は、システム・モデル、構成、またはチャンネル・パス・プロトコルにより課せられた、あらゆる制限に制約されずに、特定装置を識別するための手段として使用されます。装置番号は、システムとシステム・オペレーターの間で、該当装置に関連した通信を行うときに使用されます。例えば、システム・オペレーターは、初期プログラム・ローディングに使用する入力装置を指定するために、装置番号を入力します。

装置 ID

装置 ID は、チャンネル・サブシステムが入出力装置と通信するために使用するアドレスであり、これはプログラムからは見えません。使用される装置 ID のタイプは、特定のチャンネル・パス・タイプと、提供されているプロトコルによって異なります。1 つのサブチャンネルに、1 つまたは複数の装置 ID が含まれています。

並列入出力インターフェース・タイプのチャンネル・パスの場合は、装置 ID は、装置アドレスと呼ばれる 8 ビット値です。ESCON インターフェースの場合は、装置 ID は、4 ビットの制御装置アドレスと 8 ビットの装置アドレスから成っています。FICON 入出力インターフェースの場合は、装置 ID は、8 ビットの制御装置イメージ ID と 8 ビットの装置アドレスから成っています。FICON 変換入出力インターフェースの場合は、装置 ID は、4 ビットの制御装置アドレスと 8 ビットの装置アドレスから成っています。

装置アドレスは、サブチャンネルに関連付けられている特定の入出力装置 (および、並列入出力インターフェースでは、さらに制御装置) を識別します。例えば、装置アドレスは、特定の磁気テープ・ドライブ、ディスク・アクセス機構、伝送回線などを識別します。装置アドレスとして割り当てることができるのは、0-255 の範囲内の番号です。

特定のチャンネル・パス・タイプに使用できる装置 ID についての詳細は、そのチャンネル・パス・タイプに該当する資料を参照してください。

プログラミング上の注意: 装置番号は装置のインストール時に割り当てられるもので、任意の値をとることができます。ただし、装置番号割り当てについて、制御プログラム、サポート・プログラム、または特定の制御装置または入出力装置で必要な制約条件があれば、ユーザーはそれに従う必要があります。

入出力操作の実行

入出力操作は、START SUBCHANNEL 命令、チャンネル・コマンド・ワード (CCW)、および指令の 3 種類のフォーマットの情報に基づいて、開始され制御されません。START SUBCHANNEL 命令は、入出力データを管理または処理する他のプログラムからの入出力操作要求の流れを監視する CPU プログラムの一部として、CPU により実行されます。

START SUBCHANNEL が実行されると、ターゲット・サブチャンネルに関連した入出力装置で開始機能を実

行することをチャンネル・サブシステムに要求するパラメーターが、ターゲット・サブチャンネルに渡されます。チャンネル・サブシステムは、そのサブチャンネルにある情報(これには START SUBCHANNEL の実行により渡された情報も含まれます)を使用して開始機能を実行し、該当の入出力装置へのアクセス可能なチャンネル・パスを見つけてます。装置が選択されると、チャンネル・サブシステムは、CCW をデコードし実行することにより、入出力操作を行います。順次実行用に配列された 1 つ以上の CCW が 1 つのチャンネル・プログラムを形成し、1 つ以上の入出力操作として実行されます。命令と CCW はどちらも主記憶機構から取り出され、それぞれのフォーマットはすべてのタイプの入出力装置にわたり共通ですが、CCW のコマンド・コード内の修飾子ビットでは、特定装置での操作を実行するために、その装置固有の条件を指定することができます。

特定の装置に固有の操作(磁気テープの巻き戻しや、ディスク・ドライブでのアクセス機構の位置付けなど)は指令により指定され、入出力装置は、その指令をデコードして実行します。指令は、制御コマンドのコマンド・コード内の修飾子ビットとして装置に転送されるか、制御操作または書き込み操作の実行時にデータとして装置に転送されるか、または、その他の手段で装置に対して使用可能にされます。

開始機能の開始

CPU プログラムは、START SUBCHANNEL 命令により入出力操作を開始します。この命令は、操作要求ブロック(ORB)の内容をサブチャンネルに渡します。ORBの内容には、サブチャンネル・キー、最初に実行する CCW のアドレス、および、CCW のフォーマットの指定が含まれています。CCW は、実行するコマンドと、使用する記憶域(存在する場合)を指定します。

ORB の内容がサブチャンネルに渡された時点で、START SUBCHANNEL の実行は完了です。命令の実行結果は、プログラム状況ワードで設定される条件コードにより示されます。

必要な機能が使用可能になると、チャンネル・サブシステムは、最初の CCW を取り出し、ORB に指定されているフォーマット・ビットに従ってその CCW をデコードします。フォーマット・ビットが 0 のときは、フォーマット 0 の CCW が指定されています。フォーマット・ビットが 1 のときは、フォーマット 1 の CCW が指定されています。フォーマット 0 とフォーマット 1 の CCW には同じ情報が含まれていますが、フォーマット 1 の CCW では、CCW 内で 31 ビット・アドレスを直接指定できるように、フィールドが配列されています。

パス管理

最初の CCW が妥当性検査に合格し、延期フラグに 1 が指定されていない場合は、チャンネル・サブシステムは、選択可能なチャンネル・パスのグループからいずれかのチャンネル・パスを選択する方法により、装置を選択しようとしています。その装置 ID を認識する制御装置が、論理的にそのチャンネル・パスに接続し、選択に対する応答を返します。チャンネル・サブシステムは、そのチャンネル・パスを介して、CCW のコマンド・コード部分を送り、入出力装置は、そのコマンドを実行できるかどうかを示す状況バイトを、応答として返します。この時点で、制御装置は、チャンネル・パスから論理的に切断することも、接続したままでデータ転送を開始することもできます。

使用中表示またはパス非稼働条件のために選択が受け入れられなかった場合は、チャンネル・サブシステムは、代替チャンネル・パスがあれば、そのパスにより装置を選択しようとしています。選択可能なすべてのパスについて選択を試み、それでもなお使用中条件が続いているときは、いずれかのパスが解放されるまで操作は保留されます。装置選択を試みたチャンネル・パスの 1 つまたは複数についてパス非稼働条件が検出された場合は、入出力割り込みによりプログラムに警報が渡されます。この入出力割り込みは、チャンネル・プログラムの実行時点で起きるか(代替チャンネル・パスで装置が選択された場合)、または、装置選択が試みられたすべてのチャンネル・パスでパス非稼働条件が検出されたために、実行が中止された結果として起きます。

チャンネル・プログラムの実行

入力装置でコマンドが開始され、コマンドの実行のために装置との間でデータのやりとりをする必要がない場合は、装置は、コマンド・コードを受信した時点でただちに操作終了の信号を送ることがあります。データの転送を必要とする操作では、サブチャンネルは、チャンネル・サブシステムが装置からのサービス要求に回答して以後の操作の制御を引き受けるように、セットアップされません。

入出力操作には、1 つの CCW で指定された 1 つの記憶域との間でのデータ転送、または、連続していない複数の記憶域との間でのデータ転送が含まれることがあります。後者の場合は、一般に、入出力操作を実行するために複数の CCW のリストが使用されます。このリストでは、各 CCW がそれぞれ 1 つの連続した記憶域を指定し、データ・チェーニングによりすべての CCW が連結されています。データ・チェーニングは CCW 内のフラグにより指定され、これにより、チャンネル・サブシ

テムは、現 CCW が指している記憶域を使い尽くすかまたはいっぱいにしてしまうと、別の CCW を取り出します。データ・チェーニング上で取り出された CCW が指す記憶域は、入出力装置ですでに進行中の入出力操作に使用されるものであるため、新しい CCW が取り出されても、入出力装置には通知されません。

プログラマーは CCW のフォーマットを利用して、CCW がデコードされたときに、チャンネル・サブシステムができるだけ早く入出力割り込み要求を発行するように指定することにより、少なくともチャンネル・プログラム内のその CCW まではチェーニングが進行したことを CPU プログラムに通知することができます。

CPU での動的アドレス変換を補完するために、CCW の間接データ・アドレッシング機能が用意されています。そのために、CCW には、その CCW 用の記憶域を指定するために間接データ・アドレス・リストを使用することを指定するフラグが含まれています。1 つの記憶域ブロックの境界に達するたびに、そのリストを参照して、次に使用する記憶域ブロックが決定されます。これらの記憶域ブロックのサイズは、2K バイトまたは 4K バイトのいずれかで、これは ORB で指定します。CCW 間接データ・アドレッシングを使用することにより、動的アドレス変換がアクティブのときに CPU 内で実行されるプログラムにおいて、CPU が同じ連続した実記憶域を操作している場合と実質的に同じ CCW シーケンスを利用することができます。CCW 間接データ・アドレッシングを使用するプログラムは、ORB で指定されている CCW がフォーマット 0 かフォーマット 1 に関係なく、最大 $2^{64} - 1$ までの絶対記憶アドレスを持つデータ・ブロックを指定できます。

一般に、1 つの入出力操作または一連の入出力操作の実行では、最大 3 段階で各種の構成要素が操作に関与します。

1. CPU とチャンネル・サブシステム装置の統合による影響を除き、CPU は START SUBCHANNEL の実行中は使用中の状態になり、アドレッシングされたサブチャンネルに ORB の内容が渡されるまで、その状態が続きます。
2. サブチャンネルは、ORB の内容を受け取ってから、サブチャンネルで 1 次割り込み条件がクリアされるまでは、新しい START SUBCHANNEL に対して使用中状態になります。
3. 入出力装置は、その装置での最初の操作の開始時点から、サブチャンネルが延期されるか、またはサブチャンネルに 2 次割り込み条件が起きるまで、使用中になります。サブチャンネル延期の場合は、延期されて

いたチャンネル・プログラムの実行が再開されると、入出力装置は再び使用中になります。

入出力操作の終結

入出力操作の終結は、通常、チャンネル終了および装置終了の 2 つの状況条件により示されます。チャンネル終了条件は、入出力装置が、操作に関連したすべてのデータを受信または送信してしまい、これ以上チャンネル・サブシステムの機能を必要としなくなっていることを示します。この条件は 1 次割り込み条件と呼ばれるもので、この場合はチャンネル終了が 1 次状況となります。一般に、1 次割り込み条件とは、1 つの入出力操作に関係し、かつサブチャンネルにおいて入出力操作または入出力操作のチェーンが終結したことを知らせる割り込み条件を指します。

装置終了信号は、入出力装置が実行を完了し、別の操作を実行できる状態になったことを示します。この条件は 2 次割り込み条件と呼ばれるもので、この場合は装置終了が 2 次状況となります。一般に、2 次割り込み条件とは、1 つの入出力操作に関係し、かつ入出力装置において入出力操作または入出力操作のチェーンが終結したことを知らせる割り込み条件を指します。2 次割り込み条件は、1 次割り込み条件と同時に、またはその後で起きます。

1 次または 2 次割り込み条件と同時に、チャンネル・サブシステムと入出力装置の両方が、異常状況を示すことがあります。

入出力操作の終結を知らせる条件は、入出力割り込みによりプログラムに認識させることができます。また、CPU が入出力割り込みに対して使用不可にされているときは、プログラムからチャンネル・サブシステムに、終結の問い合わせを出すようにすることもできます。前者の場合は、これらの条件の結果として、割り込みのソースに関する情報を含む入出力割り込みコードが格納されます。後者の場合は、TEST PENDING INTERRUPTION の実行結果として、割り込みコードが格納されます。

1 次割り込み条件が認識されると、チャンネル・サブシステムは、割り込み要求を使用して、サブチャンネルで入出力操作が終結したことを示す情報がサブチャンネルに含まれていることを、プログラムに通知しようとします。この情報は、最後に使用された CCW を識別するほか、残余バイト・カウントを示す場合もあり、これにより使用された主記憶域の範囲が分かります。チャンネル・サブシステムと入出力装置は、どちらも、1 次または 2 次割り込み条件の中で、追加の指示として異常条件を示すことがあります。サブチャンネルに含まれている情報は、

TEST SUBCHANNEL または STORE SUBCHANNEL を実行して格納することができます。格納後のこの情報は、サブチャンネル状況ワード (SCSW) と呼ばれます。

プログラムは、START SUBCHANNEL 命令を 1 回使用するだけで、入出力操作のチェーンの実行を開始することができます。現 CCW がコマンド・チェーニングを指定しており、操作中に異常条件が検出されなかった場合は、装置終了信号が受信されると、サブシステムは新しい CCW を取り出します。新しい CCW が所定の妥当性検査に合格し、その CCW 内で延期フラグに 1 が指定されていない場合は、入出力装置で新しいコマンドの実行が開始されます。その CCW が妥当性検査に合格しなかった場合は、新しいコマンドは開始されず、コマンド・チェーニングは抑止され、その新しい CCW に関連する状況に応じた割り込み条件が生成されます。延期フラグに 1 が指定されていて、かつ、延期制御 (関連 ORB のワード 1 のビット 4) が 1 であるために延期フラグの 1 の値が有効である場合は、新しいコマンドの実行は開始されず、コマンド・チェーニングは終了します。

新しいコマンドの実行は、前の操作と同じ方法でチャンネル・サブシステムにより開始されます。コマンド・チェーニングを指定している CCW によって操作が終了したときに発生する終了信号は、プログラムでは使用できません。コマンド・チェーニングにより別の入出力操作が開始されると、チャンネル・サブシステムはチャンネル・プログラムの実行を続けます。ただし、異常条件が検出された場合は、コマンド・チェーニングは抑止され、チャンネル・プログラムは中止され、割り込み条件が生成され、中止を引き起こした終了信号がプログラムに通知されます。

プログラムは、延期/再開機能を使用して、チャンネル・プログラムの実行を制御することができます。延期機能の開始は、ORB の延期制御ビットの設定に従って制御されます。ORB の延期制御ビットが 1 の場合、延期機能は、チャンネル・プログラムの実行時に、最初の CCW またはコマンド・チェーニングで取り出された CCW 内で延期フラグに 1 を指定することにより、チャンネル・サブシステムに通知されます。

延期が起きるのは、ORB 内の延期制御ビットが 1 であるために延期フラグ値 1 が有効であるときに、チャンネル・サブシステムが、延期フラグに 1 が指定されている CCW を取り出した場合です。この CCW 内のコマンドは入出力装置には送られず、コマンドのチェーンが終了したことが装置に通知されます。後続の RESUME SUBCHANNEL 命令では、延期を引き起こした CCW

が変更されている可能性があり、したがって、チャンネル・サブシステムは CCW を再取り出しし、延期フラグの現在の設定を検査する必要があることが、チャンネル・サブシステムに通知されます。その CCW 内の延期フラグが 0 であることが確認されると、チャンネル・サブシステムは、入出力装置でコマンド・チェーンの実行を再開します。

チャンネル・プログラムの実行は、CANCEL SUBCHANNEL、HALT SUBCHANNEL、または CLEAR SUBCHANNEL により途中で中止することができます。CANCEL SUBCHANNEL を実行すると、チャンネル・サブシステムは、入出力装置でまだチャンネル・プログラムが開始されていない場合は、サブチャンネルでの開始機能を中止します。CANCEL SUBCHANNEL の実行により開始機能が中止されたときは、チャンネル・サブシステムは、その CANCEL SUBCHANNEL 命令に対する応答として、条件コード 0 を設定します。HALT SUBCHANNEL を実行すると、チャンネル・サブシステムは、入出力装置に停止信号を送り、サブチャンネルでのチャンネル・プログラムの実行を中止します。HALT SUBCHANNEL の実行によりチャンネル・プログラムの実行が中止されたときは、入出力割り込み要求を使用して、中止がプログラムに通知されます。この割り込み要求が生成されるのは、入出力装置が中止された操作の状況を提示したときです。ただし、コマンド・チェーニングの実行中に、装置終了を受信した後で、次のコマンドが装置に転送される前に、装置に対して停止信号が出された場合は、装置への通知の後で割り込み要求が生成されます。後者の場合は、SCSW の装置状況フィールドには 0 が入れられます。CLEAR SUBCHANNEL を実行すると、チャンネル・プログラム実行中の表示がサブチャンネルから消去され、チャンネル・サブシステムは入出力装置に消去信号を送り、また、消去機能の完了をプログラムに知らせる入出力割り込み要求を生成します。

入出力割り込み

入出力割り込み要求を起こす条件は、CPU 内の活動とは非同期に行われ、また、同時に複数の条件が生じることもあります。これらの条件は、TEST SUBCHANNEL または CLEAR SUBCHANNEL により消去されるか、入出力システム・リセットによりリセットされるまで、サブチャンネルに保持されます。

チャンネル・サブシステムが入出力割り込み条件を認識し、サブチャンネルでその条件が示されると、サブチャンネルで指定されている入出力割り込みサブクラスについては、入出力割り込み要求は保留にされます。割り込みが保留にされている入出力割り込みサブクラスは、プログラムの下で MODIFY SUBCHANNEL を使用して制御されます。保留中の入出力割り込みの入出力割り込みサ

ブクラスからの割り込みに対して使用可能にされている CPU であれば、どの CPU でもその割り込みを受け入れることができます。各 CPU について、制御レジスタ内に、8 つの入出力サブクラスのそれぞれに対して CPU を使用可能にするかどうかを制御する、8 つのマスク・ビットがあり、入出力マスク (PSW のビット 6) が CPU のマスター入出力割り込みマスクです。

ある CPU で入出力割り込みが起きると、その CPU の入出力通信域に入出力割り込みコードが格納され、入出力割り込み要求は消去されます。入出力割り込みコードは、割り込みが保留にされているサブチャネルを識別します。その後で、TEST SUBCHANNEL または STORE SUBCHANNEL を使用することより、割り込み要求を生成させた条件をサブチャネルから明示的に検索することができます。

保留中の入出力割り込み要求は、TEST PENDING INTERRUPTION により消去することも (対応する入出力割り込みサブクラスが使用可能にされているが、PSW で入出力割り込みが使用不可にされている場合)、または TEST SUBCHANNEL により消去することもできます (対応する入出力割り込みサブクラスからの入出力割り込みに対して CPU が使用不可にされている場合)。さらに、保留中の入出力割り込み要求は、CLEAR SUBCHANNEL によっても消去できます。CLEAR SUBCHANNEL と TEST SUBCHANNEL は、どちらも、サブチャネルに保持されている割り込みも消去します。

通常は、CLEAR SUBCHANNEL を使用して割り込み要求を消去しない限り、プログラムは、TEST SUBCHANNEL を実行して操作の実行に関する情報を取得します。

第14章 入出力命令

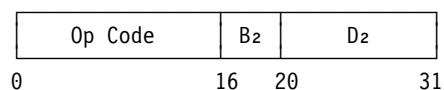
入出力命令のフォーマット	14-1	RESET CHANNEL PATH	14-8
入出力命令の実行	14-1	RESUME SUBCHANNEL	14-10
逐次化	14-1	SET ADDRESS LIMIT	14-11
オペランドのアクセス	14-1	SET CHANNEL MONITOR	14-12
条件コード	14-2	START SUBCHANNEL	14-14
プログラム例外	14-2	STORE CHANNEL PATH STATUS	14-15
命令	14-2	STORE CHANNEL REPORT WORD	14-16
CANCEL SUBCHANNEL	14-4	STORE SUBCHANNEL	14-17
CLEAR SUBCHANNEL	14-4	TEST PENDING INTERRUPTION	14-17
HALT SUBCHANNEL	14-6	TEST SUBCHANNEL	14-19
MODIFY SUBCHANNEL	14-7		

入出力命令には、チャンネル・サブシステムの動作を制御するすべての命令が含まれます。14-3ページの図14-1に、入出力命令のリストを示してあります。入出力命令は、すべて特権命令です。

一部の命令では、その命令の実行とは非同期に機能を実行するように、チャンネル・サブシステムに対して指示が送られます。このタイプの各命令については、非同期機能を要約する『関連機能』のセクションで説明されています。

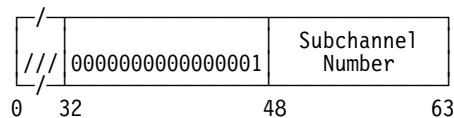
入出力命令のフォーマット

すべての入出力命令は S フォーマットをとります。



第 2 オペランドおよび汎用レジスター 1 と 2 (暗黙オペランド) を使用するかどうかは、入出力命令によって異なります。各入出力オペランドを実行するために使用するオペランドは、14-3ページの図14-1で定義されています。さらに、オペランドの詳細な使用方法については、各入出力命令の項で説明します。

サブチャンネルを参照するすべての入出力命令は、汎用レジスター 1 の内容を暗黙オペランドとして使用します。この種の入出力命令の場合、汎用レジスター 1 のビット位置 32-63 にサブシステム識別ワードが含まれていません。サブシステム識別ワードのフォーマットは、以下のとおりです。



汎用レジスター 1 のビット 48-63 は、命令で指定されている機能に使用するサブチャンネルの 2 進数を形成します。汎用レジスター 1 のビット 0-31 は無視され、ビット 32-47 は 2 進数 1 を指定します。

入出力命令の実行

逐次化

どの入出力命令を実行した場合も、逐次化とチェックポイント同期化が行われます。CPU 操作に対する逐次化の定義については、5-89ページの『CPU の逐次化』を参照してください。

オペランドのアクセス

入出力命令の実行時に、オペランドのフィールドおよびサブチャンネルのフィールド (適用される場合) がアクセスされる順序は、予測不能です。また、オペランドまたはサブチャンネルのフィールドが、入出力命令の実行を完了するために必要ない場合に、それらのフィールドに対する取り出しアクセスが行われるかどうか、予測不能です。(5-88ページの『オペランド・アクセス間の関係』を参照してください。)

条件コード

一部の命令の実行では、特定のテストの結果を使用して、PSW 内の 4 つの条件コードの 1 つが設定されず、実行結果として条件コードが設定される入出力命令は、14-3ページの図14-1 に示されています。条件コードは、入出力命令の実行結果を示します。入出力命令の条件コードの一般的な意味は、以下に示すとおりです。特定の命令での条件コードの意味については、該当の命令の項で説明してあります。

条件コード 0: 命令の実行により、予期した結果または最も予測される結果が得られました。(条件コード 0 が提示された後で生じることがあり、かつ 0 以外の据え置き条件コードを生じる条件の説明については、16-8ページの『据え置き条件コード (CC)』を参照してください。)

条件コード 1: 命令の実行により、代替的な結果または 2 番目に予測される結果が得られたか、または状況条件が提示されました。この状況条件は、予期した結果を妨げている場合とそうでない場合があります。

条件コード 2: 指定されたサブチャネルまたはチャネル・サブシステム機構が、すでに開始されている機能により使用中のため、命令の実行は無効となりました。

条件コード 3: 指定されたエレメントが非稼働状態であるか、または正常な機能の開始を妨げる何らかの条件が存在するため、命令の実行は無効となりました。

0 以外の条件コードが 2 つ以上設定されるような条件が存在する場合、条件コードの優先順位は以下のとおりです。

条件コード 3 は、条件コード 1 および 2 に優先しません。

条件コード 1 は、条件コード 2 に優先します。

プログラム例外

入出力命令で起きうるプログラム例外には、アクセス例外、オペランド例外、特権操作例外、および指定例外があります。14-3ページの図14-1 に、各入出力命令に適用される例外を示してあります。特権命令例外、オペランド例外、および指定例外の場合は、命令の実行は抑止されます。アクセス例外が起きた場合に命令がどのように終了するかは、各命令の『特別な条件』の項に特に明記されていない限り、6-35ページの『アクセス例外の認識』の説明のとおりになります。

命令

14-3ページの図14-1 は、入出力命令のニーモニック、フォーマット、および命令コードを示しています。また、この図には、プログラム割り込みの原因となる条件、および、条件コードが設定されるかどうかを示されています。

個々の命令の詳細記述では、命令ごとに、アセンブラー言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、START SUB-CHANNEL では、SSCH がニーモニックで、D₂(B₂) がオペランド指定です。

名前	ニー モニク	特性							命令 コード
		S	C	P	OP	¢	GS	B ₂	
CANCEL SUBCHANNEL	XSCH	S	C	P	OP	¢	GS		B276
CLEAR SUBCHANNEL	CSCH	S	C	P	OP	¢	GS		B230
HALT SUBCHANNEL	HSCH	S	C	P	OP	¢	GS		B231
MODIFY SUBCHANNEL	MSCH	S	C	P A SP	OP	¢	GS	B ₂	B232
RESET CHANNEL PATH	RCHP	S	C	P	OP	¢	G1		B23B
RESUME SUBCHANNEL	RSCH	S	C	P	OP	¢	GS		B238
SET ADDRESS LIMIT	SAL	S		P	OP	¢	G1		B237
SET CHANNEL MONITOR	SCHM	S		P	OP	¢	GM		B23C
START SUBCHANNEL	SSCH	S	C	P A SP	OP	¢	GS	B ₂	B233
STORE CHANNEL PATH STATUS	STCPS	S		P A SP		¢		ST B ₂	B23A
STORE CHANNEL REPORT WORD	STCRW	S	C	P A SP		¢		ST B ₂	B239
STORE SUBCHANNEL	STSCH	S	C	P A SP	OP	¢	GS	ST B ₂	B234
TEST PENDING INTERRUPTION	TPI	S	C	P A ¹ SP		¢		ST B ₂	B236
TEST SUBCHANNEL	TSCH	S	C	P A SP	OP	¢	GS	ST B ₂	B235

説明:

¢ 逐次化とチェックポイント同期化を起こします。
A 論理アドレスについてのアクセス例外。
A¹ 有効アドレスが 0 のときは、そのアドレスは記憶機構アクセスには使用されず、アクセス例外は起きません。ただし、アクセス・レジスターの変換時にアクセス例外が起きることがあります。
B₂ アクセス・レジスター・モードでは、B₂ フィールドはアクセス・レジスターを指定します。
C 条件コードが設定されます。
G1 命令の実行時に、汎用レジスター 1 が暗黙的にパラメーターとして使用されます。
GM 命令の実行時に、複数のレジスターが暗黙的に使用されます。
GS 命令の実行時に、汎用レジスター 1 が暗黙的にサブシステム識別ワードとして使用されます。
P 特権命令例外。
S S 命令フォーマット。
SP 指定例外。
ST PER 記憶域変更イベント。

図 14-1. 入出力命令の要約

CANCEL SUBCHANNEL

XSCH	[S]	
0	16	31
'B276'	////////////////	

CANCEL SUBCHANNEL が適用可能な場合に、指定されたサブチャンネルにおいて、現在の開始機能 (存在する場合) が中止されます。

汎用レジスター 1 には、現在の開始機能 (存在する場合) が中止されるサブチャンネルを指定するサブシステム識別ワードが含まれています。

サブチャンネルが、(1) サブチャンネル・アクティブではなく、(2) 開始保留中、再開保留中、または延期状態にあり、かつ、(3) 開始機能のみを実行している場合は、サブチャンネルで開始機能が中止され、サブチャンネルは、開始保留中、再開保留中、または延期状態ではなくなります。さらに、サブチャンネルについて、使用中を示す内部表示がリセットされます。

条件コード 0 が設定されて、上記で述べたアクションが行われたことを示します。

すでに開始されている開始機能について、無効な ORB フィールドまたは使用可能パスなし条件が存在し、START SUBCHANNEL の実行時にその条件が報告されなかった場合、サブチャンネルがまだこのエラー条件を報告するための状況保留状態になっていないときは、CANCEL SUBCHANNEL について条件コード 0 が示されることがあります。

特別な条件

サブチャンネルが、何らかの状況を伴う状況保留状態にあるときは、**条件コード 1** が設定され、その他のアクションはとられません。

CANCEL SUBCHANNEL が適用可能でなく、サブチャンネルが状況保留中でないときは、**条件コード 2** が設定され、その他のアクションはとられません。CANCEL SUBCHANNEL が適用可能でないのは、サブチャンネルが、(1) 指定された機能を持っていないか、(2) 指定された開始機能以外の機能も持っているか、(3) 再開保留中でも、開始保留中でも、延期でもないか、または (4) サブチャンネル・アクティブのときです。

CANCEL SUBCHANNEL に対してサブチャンネルが非稼働状態のときは、**条件コード 3** が設定され、その他の

アクションはとられません。CANCEL SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、そのサブチャンネルがチャンネル・サブシステムから提供されていないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。

CANCEL SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

結果の条件コード:

- 0 開始機能が取り消されました。
- 1 状況保留中。
- 2 CANCEL SUBCHANNEL が適用不能。
- 3 非稼働。

プログラム例外:

- オペランド例外
- 特権命令例外

プログラミング上の注意:

1. CANCEL SUBCHANNEL により行われるアクションは、命令の実行中に完了します。条件コード 0 が提示された場合は、中止された入出力操作の結果として生じる後続の入出力割り込みはありません。ただし、取り消された操作が開始保留状態にある間は、装置は使用中条件の信号を送っていることがあります。その場合は、装置はチャンネル・サブシステムに使用中解除信号を送る必要があります。その結果として、装置で次の操作が開始される前に、送信請求されていない装置終了状況が送信されることがあります。
2. CANCEL SUBCHANNEL が条件コード 0 を伴って終了すると、サブチャンネルは、START SUBCHANNEL により開始される新しい開始機能を受け入れられる状態になっています。

CLEAR SUBCHANNEL

CSCH	[S]	
0	16	31
'B230'	////////////////	

指定したサブチャンネルが消去され、そのサブチャンネルでの現在の開始機能または停止機能 (存在する場合) が中

止され、チャンネル・サブシステムには、そのサブチャンネルおよび関連の装置において非同期に消去機能を実行するよう指示する信号が送られます。

汎用レジスター 1 には、消去するサブチャンネルを指定するサブシステム識別ワードが含まれています。

開始機能または停止機能が進行中である場合は、サブチャンネルでその機能が中止されます。

サブチャンネルは、状況保留中ではなくなります。サブチャンネルにおいて、SCSW の活動制御フィールドに示されているすべての活動が消去されますが、サブチャンネルは消去保留状態にされます。サブチャンネルにおいて、この命令の実行結果として実行される消去機能を除き、SCSW の機能制御フィールドに示されている進行中のすべての機能が消去されます。

チャンネル・サブシステムには、消去機能を非同期に実行するよう指示する信号が送られます。消去機能については、『関連機能』のセクションに要約を示してあり、15-13ページの『消去機能』で詳しく説明してあります。

条件コード 0 が設定されて、上記で述べたアクションが行われたことを示します。

関連機能

チャンネル・サブシステムは、CLEAR SUBCHANNEL の実行後に、非同期に消去機能を実行します。条件が許せば、チャンネル・サブシステムは、チャンネル・パスを選択し、そして、実行中の入出力操作があればそれを中止するように指示する消去信号を、入出力装置に送ろうとします。そして、サブチャンネルは状況保留状態になります。チャンネル・サブシステムが、入出力装置に対する消去信号の発信を妨げるような条件が生じて、サブチャ

ネルは状況保留状態になります (15-13ページの『消去機能』を参照)。

消去機能を実行した結果、サブチャンネルが状況保留状態になったときは、関連の入出力装置との間のデータ転送 (存在する場合) は中止されています。TEST SUBCHANNEL により結果の状況が消去されるときに格納される SCSW では、消去機能ビットに 1 が格納されます。チャンネル・サブシステムが、入出力装置に対して消去信号が発信されたことを確認できた場合は、SCSW の消去保留ビットに 0 が格納されます。そうでない場合は、消去保留ビットには 1 が格納され、検出された条件を詳細に記述する他の表示が提供されます。(16-6ページの『割り込み応答ブロック』を参照してください。)

CLEAR SUBCHANNEL により中止された開始機能については、測定データは累算されず、サブチャンネルの拡張状況ワードに装置接続時間は格納されません。

特別な条件

CLEAR SUBCHANNEL に対してサブチャンネルが非稼働状態のときは、条件コード 3 が設定され、その他のアクションはとられません。CLEAR SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、そのサブチャンネルがチャンネル・サブシステム内にないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。

CLEAR SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

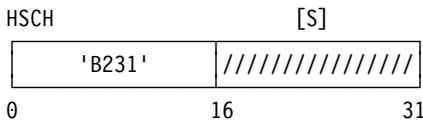
結果の条件コード:

- 0 機能が開始されました。
- 1 --
- 2 --
- 3 非稼働。

プログラム例外:

- オペランド例外
- 特権命令例外

HALT SUBCHANNEL



指定したサブチャンネルでの現在の開始機能 (存在する場合) が中止され、チャンネル・サブシステムには、そのサブチャンネルおよび関連の装置において非同期に停止機能を実行するよう指示する信号が送られます。

汎用レジスター 1 には、停止するサブチャンネルを指定するサブシステム識別ワードが含まれています。

開始機能が進行中である場合は、サブチャンネルでその機能が中止されます。

サブチャンネルは停止保留状態にされ、そのサブチャンネルで停止機能が示されます。

HALT SUBCHANNEL が実行されるときに、指定したサブチャンネルがサブチャンネル/装置アクティブの状態にあり、かつ中間状況について状況保留状態にあるときは、状況保留表示は除去されます (16-13ページの『活動制御 (AC)』の中のビット 24、25、および 28 に関する説明を参照)。状況保留条件は、停止機能の一環として再設定されます (下記の『関連機能』のセクションを参照)。

チャンネル・サブシステムには、停止機能を非同期に実行するよう指示する信号が送られます。停止機能については、『関連機能』のセクションに要約を示してあり、15-14ページの『停止機能』で詳しく説明してあります。

条件コード 0 が設定されて、上記で述べたアクションが行われたことを示します。

関連機能

チャンネル・サブシステムは、HALT SUBCHANNEL の実行後に、非同期に停止機能を実行します。条件が許せば、チャンネル・サブシステムは、チャンネル・パスを選択し、そして、実行中の入出力操作があればそれを中止するように指示する停止信号を、入出力装置に送ろうとします。そして、サブチャンネルは状況保留状態になります。

停止機能を実行した結果、サブチャンネルが状況保留状態になったときは、関連の入出力装置との間のデータ転送 (存在する場合) は中止されています。TEST SUBCHANNEL により結果の状況が消去されるときに格納される SCSW では、停止機能ビットに 1 が格納されません。入出力装置に対して停止信号が発信された場合は、停止保留ビットには 0 が格納されます。そうでない場合は、停止保留ビットには 1 が格納され、検出された条件を詳細に記述する他の表示が提供されます。(16-6ページの『割り込み応答ブロック』および 15-14ページの『停止機能』を参照してください。)

一部のモデルでは、この命令の実行の一環としてではなく、停止機能の一環として、バスが使用可能かどうかを検査されます。この種のモデルでは、選択可能なチャンネル・パスがないときは、停止信号は発信されず、サブチャンネルは状況保留状態にされます。その後、TEST SUBCHANNEL により状況保留条件が消去された場合は、SCSW の停止保留ビットに 1 が格納されます。

HALT SUBCHANNEL の実行中に状況保留条件が除去された場合は、プログラムに停止機能の完了が通知された時点で、この状況が他の状況条件とともに再設定されます。

状況保留条件が生成されている場合は、チャンネル・サブシステムは停止保留条件を認識しないことがあります。このような状況が生じるのは、例えば、サブチャンネルがすでに開始保留中または再開保留中であるときに、警報状況が提示または生成された場合、あるいは、開始機能に指定されているかまたは再開機能で暗黙に指定されている最初のコマンドについて、入出力操作を開始しようとしたときに、1 次状況が提示された場合です。チャンネル・サブシステムが状況保留条件を認識した時点が、論理上、停止保留条件を認識した時点より前である場合は、TEST SUBCHANNEL により消去されるときに、SCSW の停止保留ビットに 1 が格納されます。

HALT SUBCHANNEL により開始機能が中止されるときに、測定データが累算されていた場合は、引き続き測定データが累算され、その測定データは、現在中止されている開始機能を実行するときに必要な、サブチャンネル

および装置の使用範囲を反映します。測定データ (存在する場合) は、サブチャンネルの測定ブロック内に累積されるか、または、サブチャンネルが 1 次状況または 2 次状況で状況保留状態になる場合は、拡張状況ワードに入れられます。(17-1ページの『チャンネル・サブシステム・モニター』を参照してください。)

特別な条件

サブチャンネルが、状況保留のみの状態か、または警報状況、1 次状況、または 2 次状況のいずれかの組み合わせを伴う状況保留状態にあるときは、**条件コード 1** が設定され、その他のアクションはとられません。

HALT SUBCHANNEL に対してサブチャンネルが使用中であるときは、**条件コード 2** が設定され、その他のアクションはとられません。HALT SUBCHANNEL に対してサブチャンネルが使用中になるのは、サブチャンネルにおいて停止機能または消去機能がすでに進行中の場合です。

HALT SUBCHANNEL に対してサブチャンネルが非稼働状態のときは、**条件コード 3** が設定され、その他のアクションはとられません。HALT SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、そのサブチャンネルがチャンネル・サブシステム内にないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。一部のモデルでは、入出力装置により選択可能なチャンネル・パスがないときも、HALT SUBCHANNEL に対してサブチャンネルが非稼働状態になることがあります。(選択可能なチャンネル・パスの説明については、15-11ページの『チャンネル・パスの可用性』を参照してください。)

HALT SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

結果の条件コード:

- 0 機能が開始されました。
- 1 中間状況以外の状況を伴う状況保留。
- 2 使用中。
- 3 非稼働。

プログラム例外:

- オペランド例外
- 特権命令例外

プログラミング上の注意: 例えば、入出力装置が巻き戻しコマンドを実行中の磁気テープ装置などである場合、HALT SUBCHANNEL の実行後に、停止機能の完了を示す状況保留条件が示されるまでにかかり時間がかかることがあります。

MODIFY SUBCHANNEL

MSCH	D ₂ (B ₂)	[S]
	'B232'	B ₂ D ₂
0	16	20 31

サブチャンネル情報ブロック (SCHIB) に含まれている情報が、サブチャンネルのプログラム変更可能フィールドに入れられます。その結果、プログラムは、そのサブチャンネルに対して、消去、停止、再開、および開始機能、および特定の入出力処理機能に関連した幾つかの局面について、影響を与えるようになります。

汎用レジスター 1 に含まれているサブシステム識別ワードが指しているサブチャンネルが、SCHIB の特定フィールドの指定に従って変更されます。第 2 オペランド・アドレスは SCHIB の論理アドレスであり、これはワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

SCHIB 情報をサブチャンネルに入れることにより影響を受けるチャンネル・サブシステム操作には、以下のものがあります。

- 入出力処理 (E フィールド)
- 割り込み処理 (入出力割り込みパラメーターと ISC フィールド)
- パス管理 (D、LPM、および POM フィールド)
- モニターおよびアドレス限界検査 (測定ブロック指標、LM、および MM フィールド)
- 並行センス機能 (S フィールド)

SCHIB オペランドのワード 1 のビット 0、1、6、および 7、およびワード 6 のビット 0-30 は、すべて 0 でなければならず、ワード 1 の 9 および 10 は両方が同時に 1 であってはなりません。SCHIB のその他のフィールドは無視され、MODIFY SUBCHANNEL の処理には影響を与えません。(詳しくは、15-1ページの『サブチャンネル情報ブロック』を参照してください。)

条件コード 0 が設定されて、SCHIB からの情報がサブチャンネルのプログラム変更可能フィールドに入れられたことを示します。ただし、指定したサブチャンネルの装置番号有効ビット (V) が 0 であるときは、条件コード 0

が設定され、SCHIB からの情報はプログラム変更可能フィールドには入れられません。

特別な条件

サブチャンネルが状況保留状態にあるときは、**条件コード 1** が設定され、その他のアクションはとられません。(16-16ページの『状況制御 (SC)』を参照してください。)

サブチャンネルで、消去、停止、または開始機能が進行中の場合は、**条件コード 2** が設定され、その他のアクションはとられません。(16-12ページの『機能制御 (FC)』を参照してください。)

MODIFY SUBCHANNEL に対してサブチャンネルが非稼働状態のときは、**条件コード 3** が設定され、その他のアクションはとられません。MODIFY SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、チャンネル・サブシステム内でそのサブチャンネルが提供されていないときです。

MODIFY SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

SCHIB のワード 1 のビット 0、1、6、および 7 は 0 でなければならず、ビット 9 および 10 は両方が同時に 1 であってはなりません。SCHIB のワード 6 のビット 0-30 は、すべて 0 でなければなりません。さもないと、オペランド例外が認識されます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外および保護例外が起きたときは、MODIFY SUBCHANNEL の実行は抑止されません。

結果の条件コード:

- 0 機能は完了しました。
- 1 状況保留中。
- 2 使用中。
- 3 非稼働。

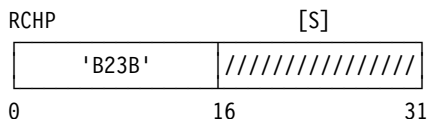
プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- オペランド例外
- 特権命令例外
- 指定例外

プログラミング上の注意:

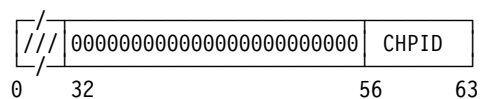
1. 関連のサブチャンネルが使用不可にされているときに、入出力装置が入出力エラー警報の信号を出した場合は、チャンネル・サブシステムは、その装置に対して消去信号を出し、入出力割り込み条件を生成せずにその入出力エラー警報指示を破棄します。
2. 関連のサブチャンネルが使用不可にされているときに、入出力装置が非送信請求状況を提示した場合は、チャンネル・サブシステムは、入出力割り込み条件を生成せずに、その状況を破棄します。ただし、提示された状況に装置チェックが含まれている場合は、チャンネル・サブシステムは、入出力割り込み条件を生成せずに、関連のサブチャンネルに対して消去信号を出します。プログラムで MODIFY SUBCHANNEL を使用してサブチャンネルを使用可能にするときは、この点に注意する必要があります。例えば、サブチャンネルが使用不可にされたときに関連装置上に存在していたメディアが置き換えられていることがあるので、プログラムはそのメディアの健全性を検査する必要があります。
3. MODIFY SUBCHANNEL で条件コード 0 が設定されたときは、その後でプログラムで STORE SUBCHANNEL を実行して、サブチャンネルの内容を検査することをお勧めします。STORE SUBCHANNEL は、指定したサブチャンネルが変更されているかどうかを判別するための手段として使用できます。MODIFY SUBCHANNEL により条件コード 0 が設定された後でサブチャンネルを検査しておかないと、プログラムが予期していないような条件が起きることがあります。

RESET CHANNEL PATH



汎用レジスター 1 の内容が指しているチャンネル・パスに対するチャンネル・パス・リセット機能を実行するよう指示する信号が、チャンネル・パス・リセット機構に送られます。

汎用レジスタ 1 のフォーマットは、以下のとおりです。



チャンネル・パス ID (CHPID): 汎用レジスタ 1 のビット位置 56-63 には、チャンネル・パス・リセット機能の実行対象とするチャンネル・パスを指定する、符号なしの 2 進整数が含まれています。

汎用レジスタ 1 のビット位置 32-55 は予約済みであり、すべて 0 でなければなりません。さもないと、オペランド例外が認識されます。汎用レジスタ 1 のビット 0-31 は無視されます。

条件が許せば、指定したチャンネル・パスに対して非同期にチャンネル・パス・リセット機能を実行するよう指示する信号が、チャンネル・パス・リセット機構に送られます。チャンネル・パス・リセット機能については、『関連機能』のセクションに要約を示してあり、17-9ページの『チャンネル・パス・リセット』で詳しく説明してあります。

条件コード 0 が設定されて、チャンネル・パス・リセット機構に信号が送られたことを示します。

関連機能

RESET CHANNEL PATH の実行後は、チャンネル・パス・リセット機構は、非同期にチャンネル・パス・リセット機能を実行します。指定したチャンネル・パスにアクセスできるすべてのサブチャンネルで、特定の標識がリセットされ、そのチャンネル・パスに対してリセット信号が出されます。リセット信号が受信されたチャンネル・パスについてのみ、入出力装置で進行中だった入出力機能がすべてリセットされます。マルチパス・モードで実行されている入出力操作または一連の入出力操作は、マルチパス・グループ内の他のチャンネル・パスで続行できる場合があります。(15-42ページの『チャンネル・パス・リセット機能』を参照してください。)

指定されたチャンネル・パスに対するチャンネル・パス・リセット機能の実行結果は、チャンネル報告を使用してプログラムに通知されます(17-18ページの『チャンネル報告』を参照)。

特別な条件

一部のモデルでは、前の RESET CHANNEL PATH の実行についてチャンネル・パス・リセット機能を実行しているために使用中である場合は、**条件コード 2** が設定され、その他のアクションはとられません。

一部のモデルでは、RESET CHANNEL PATH の実行に対して、指定されたチャンネル・パスが非稼働状態のときは、**条件コード 3** が設定され、その他のアクションはとられません。これらのモデルにおいて、チャンネル・パスが RESET CHANNEL PATH の実行に対して非稼働状態になるのは、指定されたチャンネル・パスが物理的に使用可能でないときです。

チャンネル・パス・リセット機構が使用中であり、しかも指定されたチャンネル・パスが物理的に使用可能でない場合に、条件コード 2 または 3 のいずれが設定されるかは、モデルによって異なります。

RESET CHANNEL PATH では、以下に述べるプログラム例外が起きることがあります。

汎用レジスタ 1 のビット位置 32-55 は、すべて 0 でなければなりません。さもないと、オペランド例外が認識されます。

結果の条件コード:

- 0 機能が開始されました。
- 1 --
- 2 使用中。
- 3 非稼働。

プログラム例外:

- オペランド例外
- 特権命令例外

プログラミング上の注意:

1. 送信請求されていない装置終了状況を生成することができる入出力装置については、データの安全性が損なわれる可能性を回避するために、RESET CHANNEL PATH を実行するチャンネル・パス上のその種の装置で進行中の入出力操作は、HALT SUBCHANNEL または CLEAR SUBCHANNEL のいずれかで終了させる必要があります。そうしないと、入出力装置はリセット信号を受信した後で送信請求されていない装置終了を提示することがあり、チャンネル・サブシステムがそれを送信請求された装置終了と解釈してしまう結果、コマンド・チェーニングが生じることがあります。

2. 状況検査機能が使用されているときは、リセットしようとしているチャンネル・パスに関連した進行中の操作をすべて停止してから **RESET CHANNEL PATH** を実行しないと、誤った装置状況チェック条件が検出されることがあります。

RESUME SUBCHANNEL

RSCH	[S]
0	31
16	

チャンネル・サブシステムに、指定したサブチャンネルで再開機能を実行するよう指示する信号が送られます。

汎用レジスター 1 には、どのサブチャンネルで再開機能を実行するかを指定するサブシステム識別ワードが含まれています。

指定したサブチャンネルは、再開保留にされます。

論理上、条件コード 0 が設定される前であって、かつサブチャンネルが現在延期状態にある場合のみ、サブチャンネルのパス非稼働条件が消去されます。

チャンネル・サブシステムには、非同期に再開機能を行うよう指示する信号が送られます。再開機能については、『関連機能』のセクションに要約を示してあり、15-17 ページの『開始機能と再開機能』で詳しく説明してあります。

条件コード 0 が設定されて、上記で述べたアクションが行われたことを示します。

関連機能

チャンネル・サブシステムは、RESUME SUBCHANNEL の実行後に、非同期に再開機能を実行します。サブチャンネルがサブチャンネル・アクティブである場合を除き、RESUME SUBCHANNEL の実行により条件コード 0 が設定されたときは、現在延期されているチャンネル・プログラムにより現 CCW の延期フラグが 0 に設定されていれば、そのチャンネル・プログラムの実行が再開されます。延期フラグが 1 のままになっている場合は、チャンネル・プログラムの実行は延期されたままになります。しかし、RESUME SUBCHANNEL の結果条件コード 0 が設定されたときに、サブチャンネルがサブチャンネル・アクティブである場合は、現プログラムの実行が再開されるか、サブチャンネルが一時的に延期されていることを再開機能が検出するかは、予測不能です。再開機能が、サブチャンネルが延期されていると判断するの

は、チャンネル・サブシステムが再開保留条件を認識したときに、サブチャンネルが中間状況を伴って状況保留状態にある場合のみです。(15-17 ページの『開始機能と再開機能』を参照してください。)

特別な条件

サブチャンネルが状況保留状態にあるときは、条件コード 1 が設定され、その他のアクションはとられません。

再開機能が適用されない場合は、条件コード 2 が設定され、その他のアクションはとられません。再開機能が適用されないのは、サブチャンネルにおいて、(1) 開始機能のみでなく他の機能も指定されているか、(2) 機能が何も指定されていないか、(3) 再開保留状態になっているか、または、(4) 進行中の開始機能について延期制御が指定されている場合です。

再開機能に対してサブチャンネルが非稼働状態のときは、条件コード 3 が設定され、その他のアクションはとられません。サブチャンネルが再開機能に対して非稼働状態になるのは、サブチャンネル・サブシステム内に該当のサブチャンネルが提供されていないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。

RESUME SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

結果の条件コード:

- 0 機能が開始されました。
- 1 状況保留中。
- 2 機能は適用されません。
- 3 非稼働。

プログラム例外:

- オペランド例外
- 特権命令例外

プログラミング上の注意:

1. 延期状態にあったチャンネル・プログラムの実行が再開された場合、入出力装置は、その再開を新しいコマンド・チェーンの始めと見なします。チャンネル・プログラムの実行の延期が起きたときに、入出力装置において、特定のコマンドが、コマンド・チェーンの最初になければならないかまたは 1 回だけ実行

されなければならない場合は (例えば直接アクセス記憶装置の場合など)、チャンネル・プログラム実行の再開後に、チャンネル・サブシステムが適切なコマンドを適切な順序で取り出すようにプログラムで指定する必要があります。コマンドの正しい取り出し順序をプログラムで確保するための 1 つの方法は、延期を原因とする中間割り込み条件について入出力割り込みが起きるようにすることです。

S フラグが含まれている CCW 内の PCI フラグを使用して、サブチャンネルの延期をプログラムに知らせるのは、確実な方法とは言えません。これは、サブチャンネルが延期される前に、PCI 割り込みが起きることがあるためです。その場合、SCSW は、サブチャンネルおよび装置において入出力操作が進行中であることを示します。

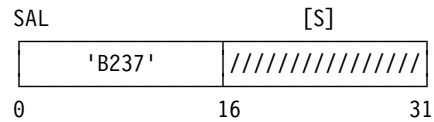
RESUME SUBCHANNEL を実行する前に、ターゲット CCW の延期フラグが 0 に設定されていることが必要です。さもないと、再開保留条件が認識され、延期フラグがまだ 1 である間に CCW が再取り出しされることがあります。その場合、再開保留条件がリセットされ、チャンネル・プログラムの実行が延期されてしまうことになります。RESUME SUBCHANNEL の実行前にターゲットの CCW の延期フラグが 0 に設定されている場合は、RESUME SUBCHANNEL の実行結果として条件コード 0 が設定されるときにサブチャンネルがサブチャンネル・アクティブでない限り、チャンネル・プログラムは延期されません。サブチャンネルがまだサブチャンネル・アクティブである間に条件コード 0 が設定された場合は、チャンネル・サブシステムが再開保留条件を認識するか、または、再開機能が、サブチャンネルが一時的に延期状態になっているものと見なすかは、予測不能です。再開機能が、サブチャンネルが延期されていると判断するのは、再開保留条件が認識されたときに、サブチャンネルが中間状況を伴って状況保留状態にある場合のみです。サブチャンネルが延期されているときは、TEST SUBCHANNEL を実行すると、中間割り込み条件が消去されるほかに、再開保留の標識も消去されます。

- 一部のモデルでは、S フラグが有効に 1 に設定されている CCW が取り出された後でのみ、再開保留条件が認識されます。したがって、あるサブチャンネルが再開保留状態にあるときに、チャンネル・プログラムの実行時に、S フラグが有効に 1 に設定されている CCW が取り出されなかった場合は、TEST SUBCHANNEL により 1 次割り込み条件が消去されるまで、そのサブチャンネルは再開保留のままになります。
- RESUME SUBCHANNEL の実行時には、パスの可用性はテストされません。代わりに、チャンネル・

サブシステムが再開機能の実行を開始した時点で、パスの可用性がテストされます。

- 再開機能の実行時に取り出される CCW の内容は、前に取り出されて、S フラグが 1 に設定されている同じ CCW の内容とは異なることがあります。

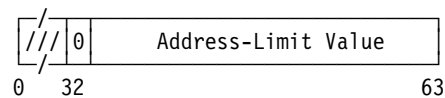
SET ADDRESS LIMIT



アドレス限界検査機能に、指定したアドレスをアドレス限界値として使用するよう指示する信号が送られ、その指定したアドレスが渡されます。

汎用レジスター 1 には、アドレス限界値として使用する絶対アドレスが含まれています。CPU が、24 ビット、31 ビット、または 64 ビットのいずれのアドレッシング・モードで稼働しているときも、指定するアドレスは 64K バイトの境界に合わせる必要があり、また、指定できる最大の絶対記憶機構アドレスは、2,147,418,112 (16 進数 7FFF0000) です。汎用レジスター 1 のビット 0-31 は無視され、ビット 32 は 0 でなければなりません。

汎用レジスター 1 のフォーマットは、以下のとおりです。



関連機能

アドレス限界検査機能が、データ・アクセスを許可するか禁止するかを決定するために使用する値を、アドレス限界値と呼びます。アドレス限界値の初期値は、0 です。アドレス限界検査機能は、SET ADDRESS LIMIT の実行により、指定されたアドレスを使用するよう指示する信号を認識するまで、この初期アドレス限界値を使用します。この指定したアドレスを新しいアドレス限界値として認識する操作は、SET ADDRESS LIMIT の実行とは非同期に行われます。

あるサブチャンネルについてアドレス限界検査を指定した場合に、アドレス検査機能が、データ・アクセスを許可するか禁止するかを決定するときに、指定されたアドレスを使用するかどうかは、SET ADDRESS LIMIT が、そのサブチャンネルに対する START SUBCHANNEL の

実行前、実行中、または実行後のどの時点で実行されたかによって異なります。SET ADDRESS LIMIT が START SUBCHANNEL の前に実行されている場合は、アドレス限界検査機能は指定されたアドレスを使用します。SET ADDRESS LIMIT が START SUBCHANNEL の実行中または実行後に実行された場合は、その特定の開始機能についてアドレス限界検査機能が指定のアドレスを使用するかどうかは、予測不能です。アドレス限界検査がどのように行われるかについては、17-16ページの『アドレス限界検査』の説明を参照してください。

特別な条件

SET ADDRESS LIMIT では、以下に述べるプログラム例外が起きることがあります。

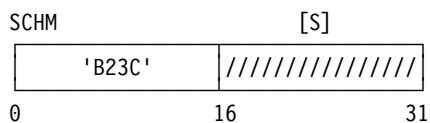
汎用レジスタ 1 のアドレスは、64K バイトの境界上にあるものでなければならず、汎用レジスタ 1 のビット 32 は 0 でなければなりません。さもないと、オペランド例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

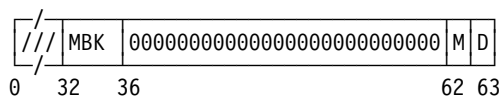
- オペランド例外
- 特権命令例外

SET CHANNEL MONITOR



チャンネル・サブシステムの測定ブロック更新モードおよび装置接続時間測定モードが、それぞれ、汎用レジスタ 1 の測定モード制御ビットの値に従って、アクティブまたは非アクティブにされます。測定ブロック更新に関する測定モード制御ビットが 1 である場合は、測定ブロック起点と測定ブロック・キーがチャンネル・サブシステムに渡されます。

汎用レジスタ 1 のフォーマットは、以下のとおりです。



無視: 汎用レジスタ 1 のビット位置 0-31 は無視されます。

測定ブロック・キー (MBK): 汎用レジスタ 1 のビット位置 32-35 には、測定ブロック・キーが含まれています。ビット 62 が 1 のときは、MBK は、チャンネル・サブシステムが測定ブロック域にアクセスするときに使用するアクセス・キーを指定します。その他の場合は、MBK は無視されます。

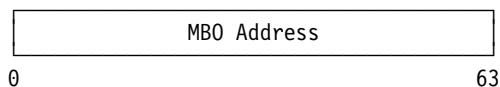
測定ブロック更新制御 (M): 汎用レジスタ 1 のビット 62 は、測定ブロック更新モードを制御する測定モード制御ビットです。汎用レジスタ 1 のビット 62 が 1 の場合は、条件が許せば、測定ブロック更新モードを非同期にアクティブにするように指示する信号が、測定ブロック更新機能に送られます。そして、汎用レジスタ 2 の中の測定ブロック起点 (MBO) アドレスと、汎用レジスタ 1 の中の測定ブロック・キー (MBK) が、測定ブロック更新機能に渡されます。測定ブロック更新機能が行う非同期機能については、下記の『関連機能』の項に要約を示し、17-1ページの『チャンネル・サブシステム・モニター』で詳しく説明します。

汎用レジスタ 1 のビット 62 が 0 の場合は、条件が許せば、測定ブロック更新モードは、アクティブのときは非アクティブにされ、非アクティブのときは非アクティブのままにされます。汎用レジスタ 1 のビット位置 32-35 (MBK) の内容と、汎用レジスタ 2 の内容は無視されます。

装置接続時間測定制御 (D): 汎用レジスタ 1 のビット 63 は、装置接続時間測定モードを制御する測定モード制御ビットです。ビット 63 で、条件が許せば、装置接続時間測定モードは、非アクティブのときはアクティブにされ、アクティブのときはアクティブのままにされます。ビット 63 が 0 で、条件が許せば、装置接続時間測定モードは、アクティブのときは非アクティブにされ、非アクティブのときは非アクティブのままにされます。

汎用レジスタ 1 のビット位置 36-61 は予約済みであり、すべて 0 でなければなりません。さもないと、オペランド例外が認識されます。

汎用レジスタ 2 のフォーマットは、以下のとおりです。



測定ブロック起点 (MBO) アドレス: 汎用レジスタ 1 のビット 62 (M) が 1 のときは、汎用レジスタ 2 のビット位置 0-63 に測定ブロック起点 (MBO) の絶対アドレスが含まれており、これが測定ブロック域の始めとなります。チャンネル・サブシステムは、MBO アドレスを使用して、測定ブロックを見つけます。測定ブロック域の起点は、32 バイト境界に合わせて指定する必要があります。さもないと、オペランド例外が認識されます。汎用レジスタ 1 のビット 62 が 0 のときは、汎用レジスタ 2 の内容は無視されます。

チャンネル・サブシステム・モニター機能が使用するチャンネル・サブシステム・タイマーがエラー状態にあるときは、その状態はリセットされます。これは、2 つの測定モード制御ビットの設定に関係なく行われます。(タイミング機能の説明については、17-2 ページの『チャンネル・サブシステム・タイミング』を参照してください。)

関連機能

測定ブロック更新機能が、SET CHANNEL MONITOR から、測定ブロック更新モードをアクティブにするように指示する信号を受け取ったときに、どのような機能を実行するかは、その信号が生成されたときにモードがすでにアクティブであったかどうかにより異なります。

信号が生成されたときに測定ブロック更新モードが非アクティブだった場合は、測定ブロック更新機能がその信号を認識するまで、モードは非アクティブのままです。測定ブロック更新機能がその信号を認識すると、測定ブロック更新モードはアクティブになり、その信号に関連した MBK および MBO (つまり、信号が生成されたときに渡された MBK と MBO) を使用して、測定データの格納が制御されます。

信号が生成されたときに測定ブロック更新モードがアクティブだった場合は、測定ブロック更新機能が信号を認識するまでは、モードはアクティブのままになり、引き続き前の SET CHANNEL MONITOR の実行に関連した MBK と MBO を使用して、測定データの格納が制御されます。測定ブロック更新機能が信号を認識すると、前の SET CHANNEL MONITOR 命令の実行に関連した MBK と MBO の代わりに、その信号に関連した MBK と MBO が使用されるようになります。

上記いずれの場合も、測定ブロック更新機能が信号を認識するのは、信号を生成させた SET CHANNEL MONITOR 命令の実行前または実行後であって、かつ、論理的に、この機能による測定の対象として使用可

能にされているサブチャンネルに対する後続の START SUBCHANNEL が開始する開始機能が実行される前です。この機能による測定の対象として使用可能にされているサブチャンネルにおいて、シグナル生成の時点ですでに開始機能が進行中である場合は、そのサブチャンネルに関する測定データが、その信号に関連した MBK と MBO を使用して格納されるかどうかは、予測不能です。

測定ブロック更新モードがアクティブのときは、測定ブロック更新の対象として使用可能にされているサブチャンネルについて、パフォーマンス測定が累積されます。1 つのサブチャンネルに関する測定データが、測定ブロック域内の 1 つの 32 バイト測定ブロックの中に累積されません。サブチャンネルを測定ブロック更新モードの対象として使用可能にするには、SCHIB で測定ブロック更新使用可能ビットを 1 に設定し、次にそのサブチャンネルに対して MODIFY SUBCHANNEL を実行します。サブチャンネルに関する測定データを累積するために使用される測定ブロックは、そのサブチャンネルに含まれている測定ブロック指標によって決まります。

装置接続時間測定モードがアクティブのときは、装置接続時間測定の対象として使用可能にされているサブチャンネルについて、チャンネル・プログラムの実行中に入出力装置がアクティブな状態でチャンネル・サブシステムと通信している時間が累積されます。サブチャンネルの測定値は、IRB の拡張状況ワード (ESW) に示されます。サブチャンネルを装置接続時間測定モードに対して使用可能にするには、SCHIB の装置接続時間測定使用可能ビットを 1 に設定し、次にそのサブチャンネルに対して MODIFY SUBCHANNEL を実行します。

測定ブロック更新モード、測定ブロックのフォーマットと内容、および装置接続時間測定モードについての詳細は、17-1 ページの『チャンネル・サブシステム・モニター』を参照してください。

特別な条件

SET CHANNEL MONITOR では、以下に述べるプログラム例外が起きることがあります。

汎用レジスタ 1 のビット 36-61 は、すべて 0 でなければなりません。汎用レジスタ 1 のビット 62 (M) が 1 のときは、汎用レジスタ 2 の中の MBO アドレスを 32 バイト境界に合わせて指定する必要があります。さもないと、オペランド例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- オペランド例外
- 特権命令例外

プログラミング上の注意: チャンネル・サブシステムが初期化される時点では、測定ブロック更新モードおよび装置接続時間測定モードは、非アクティブにされています。

START SUBCHANNEL

SSCH	D ₂ (B ₂)	[S]
	'B233'	B ₂ D ₂
0	16	20 31

関連の入出力装置について非同期に開始機能を実行するよう指示する信号が、チャンネル・サブシステムに送られ、指定した ORB に含まれている実行パラメーターが指定したサブチャンネルに置かれます。(15-20ページの『操作要求ブロック』を参照してください。)

汎用レジスター 1 には、開始するサブチャンネルを指定するサブシステム識別ワードが含まれています。第 2 オペランド・アドレスは ORB の論理アドレスであり、これはワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

ORB に含まれている実行パラメーターが、指定したサブチャンネルに入れられます。

START SUBCHANNEL が実行されるときは、サブチャンネルは 2 次状況のみを伴って状況保留状態になり、拡張状況ワード・フォーマット・ビット (L) は 0 であり、そのサブチャンネルでの状況保留条件は破棄されません。

サブチャンネルは開始保留にされ、そのサブチャンネルで開始機能が示されます。

サブチャンネルに、条件コード 0 の設定より論理上は前の時点のパス非稼働条件が存在する場合は、その条件は消去されます。

チャンネル・サブシステムに、開始機能を非同期に行うよう指示する信号が送られます。開始機能については、下記の『関連機能』のセクションに要約を示し、15-17ページの『開始機能と再開機能』で詳しく説明してあります。

条件コード 0 が設定されて、上記で述べたアクションが行われたことを示します。

関連機能

START SUBCHANNEL の実行後に、チャンネル・サブシステムは非同期に開始機能を実行します。

すべて 0 でなければならぬフィールドを除き、ORB の内容の妥当性が検査されます。一部のモデルでは、0 でなければならぬ ORB のフィールドが、命令の実行時ではなく、非同期に検査されます。無効なフィールドが非同期に検出された場合は、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴って状況保留になり、据え置き条件コード 1 およびプログラム・チェックが示されます。(16-23ページの『プログラム・チェック』を参照。)その場合は、装置では入出力操作または入出力操作のチェーンは開始されず、TEST SUBCHANNEL の実行により SCSW が消去されるときに、開始保留ビットに 1 が格納されて、この条件を示します。(16-6ページの『サブチャンネル状況ワード』を参照してください。)

一部のモデルでは、この命令の実行時ではなく、非同期にパスの可用性が検査されます。選択可能なチャンネル・パスがないときは、サブチャンネルは、1 次状況および 2 次状況を伴って状況保留になり、据え置き条件コード 3 が示されます。その場合は、装置では入出力操作または入出力操作のチェーンは開始されず、TEST SUBCHANNEL の実行により SCSW が消去されるときに、開始保留ビットに 1 が格納されて、この条件を示します。

条件が許せば、いずれかのチャンネル・パスが選択され、ORB に指定されているチャンネル・プログラムの実行が開始されます。(15-17ページの『開始機能と再開機能』を参照してください。)

特別な条件

START SUBCHANNEL の実行時にサブチャンネルが状況保留状態にあるときは、**条件コード 1** が設定され、その他のアクションはとられません。一部のモデルでは、サブチャンネルが 2 次状況のみと伴って状況保留状態にあるときは、条件コード 1 は設定されず、代わりに状況保留状況は破棄されます。

サブチャンネルで、開始機能、停止機能、または消去機能が現在進行中である場合は、**条件コード 2** が設定され、その他のアクションはとられません (16-12ページの『機能制御 (FC)』を参照)。

START SUBCHANNEL に対してサブチャンネルが非稼働状態のときは、条件コード 3 が設定され、その他のアクションはとられません。START SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、そのサブチャンネルがチャンネル・サブシステム内にはないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。

また、一部のモデルでは、選択可能なチャンネル・パスがないときも、サブチャンネルは START SUBCHANNEL に対して非稼働状態になります。この種のモデルでは、START SUBCHANNEL の実行の一環として、使用可能なチャンネル・パスが存在しないことが検出されます。その他のモデルでは、チャンネル・パスの可用性は、非同期の開始機能の一部として検査されるだけです。

START SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスタ 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

ORB のワード 1 のビット 13 および 25-30、および ORB のワード 2 のビット 0 は、すべて 0 でなければなりません。そうでない場合は、一部のモデルではオペランド例外が認識されます。他のモデルでは、非同期の開始機能の一環として、プログラム・チェックを示す入出力割り込み条件が生成されます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識され、START SUBCHANNEL の実行は抑止されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、START SUBCHANNEL の実行は抑止されます。

結果の条件コード:

- 0 機能が開始されました。
- 1 状況保留中。
- 2 使用中。
- 3 非稼働。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- オペランド例外
- 特権命令例外
- 指定例外

STORE CHANNEL PATH STATUS

STCPS	D ₂ (B ₂)	[S]
	'B23A'	B ₂ D ₂
0	16	20 31

256 ビットのチャンネル・パス状況ワードが、指定した位置に格納されます。

第 2 オペランド・アドレスは、チャンネル・パス状況ワードを格納する位置の論理アドレスです。このアドレスは、32 バイト境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

チャンネル・パス状況ワードは、STORE CHANNEL PATH STATUS の実行時点で、いずれかの入出力装置とアクティブに通信しているチャンネル・パスを示します。ビット位置 0-255 は、それぞれ、チャンネル・パス ID 0-255 を持つチャンネル・パスに対応しています。指定した位置にある 256 ビットのそれぞれが、次の条件に従って、1 に設定されるか、0 に設定されるか、または無変更のままとなります。

- 構成内のチャンネル・パスのうち、STORE CHANNEL PATH STATUS の実行時点で入出力装置とアクティブに通信しているすべてのチャンネル・パスについて、それぞれ対応するビットに 1 が格納されます。
- (1) システムで提供されているチャンネル・パス (PMCW の PIM ビットが 1) のうち、(2) 構成には含まれているが、現在は、チャンネル・サブシステムにより入出力装置とのアクティブな通信に使用されていないすべてのチャンネル・パスについては、それぞれ対応するビットに 0 が格納されます。
- システムで提供されていないすべてのチャンネル・パス (PMCW の PIM ビットが 0 のもの) については、対応するビットには、何も格納されないかまたは 0 が格納されます。
- 構成内のチャンネル・パスのうち、チャンネル・パス終了状態にあるか、または物理的に使用可能でない (PMCW の対応する PAM ビットが 0 である) すべてのチャンネル・パスについては、対応するビットに 0 が格納されます。

特別な条件

STORE CHANNEL PATH STATUS では、以下に述べるプログラム例外が起きることがあります。

第 2 オペランドは、32 バイト境界に合わせて指定する必要があります。さもないと、指定例外が認識されません。

何らかのアドレッシング例外および保護例外が起きたときは、STORE CHANNEL PATH STATUS の実行は抑止されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

プログラミング上の注意:

1. チャンネル・パス状況ワードのビットが一貫性をもって解釈されるようにするためには、プログラムは、その区域を最初に使用する前に、チャンネル・パス状況ワードを格納する位置に 0 を格納する必要があります。

STORE CHANNEL REPORT WORD

STCRW	D ₂ (B ₂)	[S]	
0	16	20	31
'B239'			
B ₂		D ₂	

チャンネル・サブシステムに影響を与える情報を含む CRW が、指定した位置に格納されます。

第 2 オペランド・アドレスは、CRW を格納する位置の論理アドレスです。このアドレスは、ワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されます。

チャンネル・サブシステム操作に影響を与える誤動作またはその他の条件が認識された場合は、その条件を記述するチャンネル報告 (1 つ以上の CRW から成る) が、プログラムによる検索および分析のために保留にされます。チャンネル報告には、誤動作またはその他の条件が検出された後の、チャンネル・サブシステムの機構の識別と状態に関する情報が含まれています。チャンネル報告、CRW、および、チャンネル・サブシステムに関連したプログラム回復アクションの説明については、17-17ページの『チャンネル・サブシステム・リカバリー』を参照してください。

1 つ以上のチャンネル報告が保留にされているときは、この命令により、CRW が指定の位置に格納され、条件コード 0 が設定されます。保留中の CRW は、STORE CHANNEL REPORT WORD を実行することによってのみ格納でき、格納後は保留状態ではなくなります。したがって、保留中の各 CRW がプログラムに提示されるのは、それぞれ 1 回のみです。

チャンネル・サブシステム内の保留中のチャンネル報告がないときは、STORE CHANNEL REPORT WORD を実行すると、指定した位置に 0 が格納され、条件コード 1 が設定されます。

特別な条件

STORE CHANNEL REPORT WORD では、以下に述べるプログラム例外が起きることがあります。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外および保護例外が起きたときは、STORE CHANNEL REPORT WORD の実行は抑止されます。

結果の条件コード:

- 0 CRW が格納されました。
- 1 0 が格納されました。
- 2 --
- 3 --

プログラム例外:

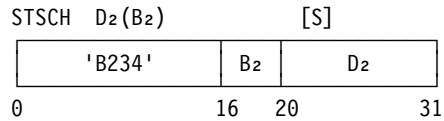
- アクセス例外 (格納: オペランド 2)
- 特権命令例外
- 指定例外

プログラミング上の注意:

1. 保留中のチャンネル報告を消去するための STORE CHANNEL REPORT WORD が実行されないと、CRW オーバーフロー条件が起きることがあります。オーバーフロー条件が検出された場合は、1 つ以上のチャンネル報告ワードが失われることがあります。(詳細については、17-17ページの『チャンネル・サブシステム・リカバリー』を参照してください。)
2. 保留中の CRW は、いずれかの CPU でマシン・チェック割り込みが起きたかどうかに関係なく、構成内の任意の CPU で STORE CHANNEL REPORT

WORD を実行することにより、消去することができます。

STORE SUBCHANNEL



指定したサブチャンネルに関する制御情報と状況情報が、指定した SCHIB に格納されます。

汎用レジスタ 1 には、どのサブチャンネルに関する情報を格納するかを指定するサブシステム識別ワードが含まれています。第 2 オペランド・アドレスは SCHIB の論理アドレスであり、これはワード境界に合わせて指定する必要があります。さもないと、指定例外が認識されません。

SCHIB に格納される情報は、バス管理制御ワード、SCSW、および、3 ワードのモデル固有情報から成っています。(15-1ページの『サブチャンネル情報ブロック』を参照してください。)

STORE SUBCHANNEL を実行することによって、サブチャンネル内の情報が変更されることはありません。

指定したサブチャンネルに関する制御情報と状況情報が SCHIB に格納されると、条件コード 0 が設定されます。STORE SUBCHANNEL の実行結果として条件コード 0 が設定されたときは、SCHIB 内の情報はサブチャンネルの一貫性のある状態を示しています。

特別な条件

STORE SUBCHANNEL に対して、指定したサブチャンネルが非稼働状態のときは、**条件コード 3** が設定され、その他のアクションはとられません。STORE SUBCHANNEL に対してサブチャンネルが非稼働状態になるのは、チャンネル・サブシステム内でそのサブチャンネルが提供されていない場合です。

STORE SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスタ 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 SCHIB が格納されました。
- 1 --
- 2 --
- 3 非稼働。

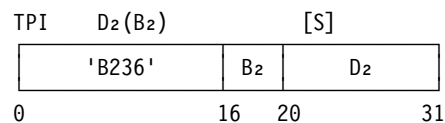
プログラム例外:

- アクセス例外 (格納: オペランド 2)
- オペランド例外
- 特権命令例外
- 指定例外

プログラミング上の注意:

1. SCSW に格納される装置状況には、装置使用中、制御装置使用中、または制御装置終了の表示が含まれることがあります。
2. SCHIB に格納される情報は、該当のサブチャンネルから取得されます。STORE SUBCHANNEL 命令の結果として、チャンネル・サブシステムが、アドレスリングされた装置に対して問い合わせを出すことはありません。
3. STORE SUBCHANNEL は、サブチャンネルに存在する条件をサンプリングするためにいつでも実行できます。その実行により保留中の状況条件が消去されることはありません。
4. CPU が 1 つのサブチャンネルに反復的にアクセスすると、チャンネル・サブシステムがそのサブチャンネルを更新するために行うアクセスが遅延または禁止されることがあるため、間を置かずに (例えばサブチャンネルの状態がいつ変化するかを判別するためなどに)、反復的に STORE SUBCHANNEL を実行することは避けてください。

TEST PENDING INTERRUPTION



サブチャンネルで保留中の入出力割り込みについての入出力割り込みコードが、第 2 オペランドに指定する位置に格納され、保留中の入出力割り込み要求が消去されます。

第 2 オペランド・アドレス (0 以外のとき) は、2 ワードの割り込みコード (ワード 0 および 1) を格納する位置の論理アドレスです。第 2 オペランド・アドレスは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

第 2 オペランド・アドレスが 0 の場合は、3 ワードの入出力割り込みコード (ワード 0-2 から成る) が、実記憶位置 184-195 に格納されます。その場合は、低アドレス保護およびキー制御保護は適用されません。

アクセス・レジスタ・モードでは、第 2 オペランド・アドレスが 0 のときは、アクセス・レジスタ B₂ についてアクセス・レジスタ変換が行われるかどうかは予測不能です。変換が行われても、結果のアドレス・スペース制御エレメントは使用されません。つまり、この場合も、割り込みコードが実記憶位置 184-195 に格納されます。

保留中の入出力割り込み要求が受理されるのは、この命令を実行している CPU の制御レジスタ 6 の中の入出力割り込みサブクラス・マスクにより許可される入出力割り込みサブクラスについてのみです。制御レジスタ 6 により許可されている入出力割り込み要求が存在しない場合は、入出力割り込みコードは格納されず、第 2 オペランド位置は変更されず、条件コード 0 が設定されます。

保留中の入出力割り込み要求が受理された場合は、入出力割り込みコードが格納され、保留中の入出力割り込み要求が消去され、条件コード 1 が設定されます。格納される入出力割り込みコードは、入出力割り込みが起きた場合に格納されるコードと同じものです。ただし、入出力割り込みが起きた場合と異なり、PSW はスワップされません。

この命令の実行により格納される入出力割り込みコードの定義は、以下のとおりです。

Word 0	Subsystem-Identification Word
1	I/O-Interruption Parameter
2	I/O-Interruption-Identification Word
0	31

サブシステム識別ワード (Subsystem-Identification Word: SID): 14-1 ページの『入出力命令のフォーマット』を参照してください。

入出力割り込みパラメータ (I/O-Interruption Parameter): ワード 1 には、プログラムにより指定された 4 バイトのパラメータが入ります。このパラメータは、ORB のワード 0 かまたは PMCW に入れて、サブチャンネルに渡されたものです。装置が警報状況を示しており、START SUBCHANNEL または MODIFY SUBCHANNEL の実行により前もって割り込みパラメータがサブチャンネルに渡されていない場合は、このフィールドには 0 が入ります。

入出力割り込み識別ワード (I/O-Interruption-Identification Word): ワード 2 (格納される場合) には、入出力割り込みのソースをさらに厳密に識別する入出力割り込み識別ワードが入ります。ワード 2 が格納されるのは、第 2 オペランド・アドレスが 0 の場合のみです。入出力割り込み識別ワードの定義は、以下のとおりです。

00	ISC	00000000000000000000000000000000	
0	2	5	31

入出力割り込みサブクラス (ISC): 入出力割り込み識別ワードのビット位置 2-4 には、0-7 の範囲内の符号なしの 2 進整数が入ります。これは、保留中の割り込み要求が消去されたサブチャンネルに関連した入出力割り込みサブクラスを指定します。

特別な条件

TEST PENDING INTERRUPTION では、以下に述べるプログラム例外が起きることがあります。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

何らかのアドレッシング例外または記憶保護例外が起きたときは、TEST PENDING INTERRUPTION の実行は抑止されます。

結果の条件コード:

- 0 割り込みコードは格納されていません。
- 1 割り込みコードが格納されました。
- 2 --
- 3 --

プログラム例外:

- アクセス例外 (格納: オペランド 2。第 2 オペランド・アドレスが 0 以外の場合のみ)

- 特権命令例外
- 指定例外

プログラミング上の注意:

1. 入出力割り込みのマスクがオフになっているときに、TEST PENDING INTERRUPTION を実行する場合は、必ず第 2 オペランド・アドレスを 0 にしてください。さもないと、入出力割り込みが起きた場合に、この命令により格納された入出力割り込みコードが失われることがあります。つまり、TEST PENDING INTERRUPTION の後で起きた入出力割り込みのソースを識別する割り込みコードも、実記憶位置 184-195 に格納されるため、この命令により格納された入出力割り込みコードは置き換えられてしまいます。
2. アクセス・レジスター・モードにおいて、第 2 オペランド・アドレスが 0 のときに、アクセス・レジスター変換が行われ、アクセス・レジスターにエラーがある場合は、アクセス例外が認識されます。この例外を防止するには、B₂ フィールドを 0 にするか、または、16 進数 00000000、16 進数 00000001、またはその他の有効な内容をアクセス・レジスターに入れます。

TEST SUBCHANNEL

TSCH	D ₂ (B ₂)	[S]
0	'B235'	31
	B ₂	20
	D ₂	16

サブチャンネルに関する制御情報と状況情報が、指定した IRB に格納されます。

汎用レジスター 1 には、どのサブチャンネルに関する情報を格納するかを指定するサブシステム識別ワードが含まれています。第 2 オペランドは IRB の論理アドレスで、これはワード境界に合わせて指定されていなければなりません。さもないと、指令例外が認識されます。

IRB に格納される情報は、SCSW、拡張状況ワード、および拡張制御ワードから成っています。(16-6ページの『割り込み応答ブロック』を参照してください。)

サブチャンネルが状況保留状態にある場合は、状況制御フィールドの状況保留ビットには 1 が格納されます。サブチャンネルが保留状態にあるかどうかによって、TEST SUBCHANNEL の実行時に行われる機能が異なります。

サブチャンネルが状況保留状態にあるときに TEST SUBCHANNEL が実行された場合は、上記の情報が IRB に格納された後、14-20ページの図14-2 に示すように、サブチャンネルに存在する特定の条件および標識が消去されます。そのサブチャンネルについて入出力割り込み要求が保留状態にある場合は、その要求は消去されます。条件コード 0 が設定されて、これらのアクションが行われたことを示します。

サブチャンネルが保留状態でないときに TEST SUBCHANNEL が実行された場合は、上記の情報が IRB に格納されるだけで、条件や標識は消去されません。条件コード 1 が設定されて、これらのアクションが行われたことを示します。

14-20ページの図14-2 は、サブチャンネルが状況保留状態にあるときに、TEST SUBCHANNEL によりどの条件および標識が消去されるかを示しています。サブチャンネルに存在するその他の条件および標識は、変更されません。

フィールド	サブチャネル条件*				
	警報 状況 保留	中間 状況 保留	1 次 状況 保留	2 次 状況 保留	状況 保留 のみ
機能制御	C	Nc	C	C	C
活動制御	Cp	Nr	Cp	Cp	Cp
状況制御	Cs	Cs	Cs	Cs	Cs
N 条件	C	Nr	C	C	C
説明: * 右端の列は、状況保留のみが起きている場合に適用されるという点に注意してください。その他の 4 つの状況保留条件の場合は、消去アクションがとられます。これらのアクションは、単一の状況保留条件が起きたときと、4 つの保留条件の幾つかが組み合わされて起きたときの、どちらの場合にも適用されます。複数の条件が起きている場合は、個々の条件に該当するすべての消去アクションが適用されます。 C 消去されます。 Cp 再開保留条件、開始保留条件、停止保留条件、消去保留条件、および延期保留条件が消去されます。 Cs 状況保留条件が消去されます。 Nc 機能制御が停止機能を示し、活動制御が延期を示している場合以外は、変更されません。停止機能と延期の両方が示されている場合は、状況保留のみの場合と同様に条件が消去されます。 Nr 活動制御が延期を示し、機能制御が開始機能（停止機能を伴うか伴わないもの）を示している場合以外は、変更されません。停止機能が示されている場合は、状況保留のみの場合と同様に条件が消去されます。開始機能のみが示されている場合は、再開保留条件と N 条件が消去されます。					

図 14-2. TEST SUBCHANNEL によりサブチャネルで消去される条件と標識

特別な条件

TEST SUBCHANNEL に対してサブチャネルが非稼働状態のときは、**条件コード 3** が設定され、その他のアクションはとられません。TEST SUBCHANNEL に対してサブチャネルが非稼働状態になるのは、そのサブチャネルが提供されていないか、有効な装置番号が割り当てられていないか、または使用可能にされていない場合です。

TEST SUBCHANNEL では、以下に述べるプログラム例外が起きることがあります。

汎用レジスター 1 のビット位置 32-47 には、16 進数値 0001 が含まれていなければなりません。さもないと、オペランド例外が認識されます。

第 2 オペランドは、ワード境界に合わせて指定されていなければなりません。さもないと、指定例外が認識されます。

アドレッシング例外および記憶保護例外により TEST SUBCHANNEL の実行が中止されたときは、サブチャネルの状態は変更されません。

結果の条件コード:

- 0 IRB が格納されます。サブチャネルは状況保留状態です。
- 1 IRB が格納されます。サブチャネルは状況保留状態ではありません。
- 2 --
- 3 非稼働。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- オペランド例外
- 特権命令例外
- 指定例外

プログラミング上の注意:

1. SCSW に格納される装置状況には、装置使用中、制御装置使用中、または制御装置終了の表示が含まれることがあります。
2. IRB に格納される情報は、該当のサブチャネルから取得されます。TEST SUBCHANNEL 命令の結果として、チャンネル・サブシステムが、アドレッシングされた装置に対して問い合わせを出すことはありません。
3. 入出力割り込みが起きた場合、それはサブチャネルにおける状況保留条件の結果なので、通常、その状況を消去するために TEST SUBCHANNEL を実行します。そのほかに、サブチャネルに存在する条件をサンプリングするためにも、TEST SUBCHANNEL を実行することができます。
4. 条件によっては、開始機能の完了が示される場合と示されない場合があります。したがって、開始機能が完了したかどうかを判別する目的で TEST SUBCHANNEL を繰り返し実行することは避けてください。例えば、別のサブチャネルがすでに状況保留状態にあるために、チャンネル・サブシステムにおいていずれかのサブチャネルについてインターフェース制御チェック (IFCC) が未決状態にあり、TEST SUBCHANNEL でテストしようとしている開始機能用として選択できるパスが、IFCC 条件を持つチャンネル・パスのみである場合は、その別のサブチャ

ネルにおける状況保留条件が消去されて、IFCC 条件が適用されるサブチャンネルで IFCC 条件が示されるようになるまでは、開始機能は開始されないことがあります。

5. CPU が 1 つのサブチャンネルに反復的にアクセスすると、チャンネル・サブシステムによるそのサブチャンネルへのアクセスが遅延または禁止されることがあります。したがって、例えばサブチャンネルの状態がいつ変化するかを判別するなどの目的で、間を置かずに反復的に TEST SUBCHANNEL を実行することは避けてください。複数の CPU で、同じサブチャンネルを対象としてほとんど同時に TEST SUB-

CHANNEL が実行された場合も、同じ影響が生じるので、このような操作も避けるようにしてください。

6. TEST SUBCHANNEL を実行することにより、1 つの CPU による入出力割り込みの処理優先順位を変更することができます。TEST SUBCHANNEL を実行するときに、指定したサブチャンネルで入出力割り込み要求が保留状態になっている場合は、前に設定されている優先順位に関係なく、その入出力割り込み要求は消去され、SCSW が格納されます。残りの入出力割り込み要求の相対優先順位は変わりません。

第15章 基本入出力機能

基本入出力機能の制御	15-1	データのブロック化	15-20
サブチャンネル情報ブロック	15-1	操作要求ブロック	15-20
パス管理制御ワード		チャンネル・コマンド・ワード	15-25
(Path-Management-Control Word)	15-2	コマンド・コード	15-27
サブチャンネル状況ワード		記憶域の指定	15-27
(Subchannel-Status Word)	15-7	チェーニング	15-29
モデル依存区域 (Model-Dependent Area)	15-7	データ・チェーニング	15-31
変更可能フィールドの要約	15-7	コマンド・チェーニング	15-32
チャンネル・パス従属関係	15-10	スキップ	15-33
作業従属関係	15-10	プログラム制御割り込み	15-33
アクティブ従属関係	15-11	CCW 間接データ・アドレッシング	15-34
専用従属関係	15-11	チャンネル・プログラム実行の延期	15-36
チャンネル・パスの可用性	15-11	コマンドとフラグ	15-38
制御装置タイプ	15-12	チャンネル・プログラム内でのブランチ	15-38
消去機能	15-13	チャンネル内転送	15-39
消去機能: パス管理	15-13	コマンド再試行	15-39
消去機能: サブチャンネル変更	15-13	開始前の入出力操作の終結	15-39
消去機能: 信号発信と完了	15-13	開始時の入出力操作の終結	15-40
停止機能	15-14	入出力操作の即時終結	15-40
停止機能: パス管理	15-14	データ転送時の入出力操作の終結	15-41
停止機能: 信号発信と完了	15-15	チャンネル・パス・リセット機能	15-42
開始機能と再開機能	15-17	チャンネル・パス・リセット機能の信号発信	15-42
開始機能と再開機能: パス管理	15-17	チャンネル・パス・リセット機能完了信号の発信	15-43
入出力操作の実行	15-19		

入出力命令には、チャンネル・サブシステムに対して、特定の機能を実行するように指定するものがあります。このような機能を、総称的に基本入出力機能と呼びます。基本入出力機能には、消去機能、停止機能、開始機能、再開機能、およびチャンネル・パス・リセット機能があります。

基本入出力機能の制御

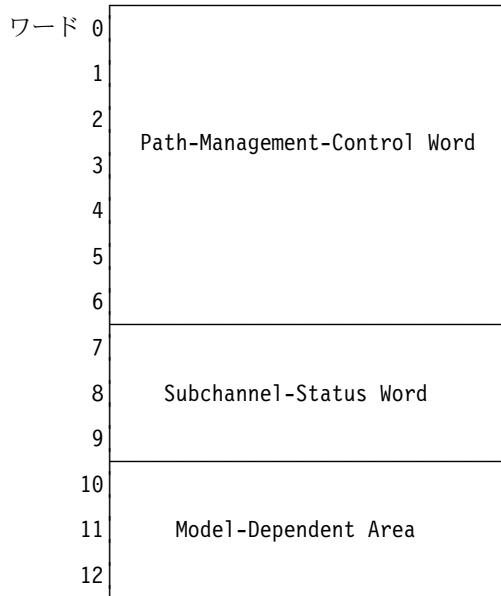
消去機能、停止機能、再開機能、および開始機能をどのように実行するかは、サブチャンネルが持つ情報に従って制御されます。この情報は、STORE SUBCHANNELの実行時に、サブチャンネル情報ブロックに入れてプログラムに渡されます。

サブチャンネル情報ブロック

サブチャンネル情報ブロック (SCHIB) は、MODIFY SUBCHANNEL 命令および STORE SUBCHANNEL 命令のオペランドです。SCHIB アドレスの右端 2 ビットは 0 で、これは SCHIB をワード境界上に指定します。SCHIB の主要フィールドには、パス管理制御ワード (PMCW)、サブチャンネル状況ワード (SCSW)、モデル依存区域の 3 つがあります。(15-2ページの図15-1 は PMCW のフォーマットを示し、16-7ページの図16-2 は SCSW のフォーマットを示しています。)

指定したサブチャンネルの現在の PMCW、SCSW、およびモデル依存データを格納するには、STORE SUBCHANNEL を使用します。MODIFY SUBCHANNEL は、サブチャンネルの特定の PMCW フィールドを変更します。プログラムで 1 つ以上の PMCW フィールドの内容を変更する必要がある場合、一般的な手順は、(1) STORE SUBCHANNEL 命令を実行して現在の内

容を取得し、(2) 主記憶機構内で PMCW に対して必要な変更を行い、そして、(3) MODIFY SUBCHANNEL を実行してサブチャンネルに新しい情報を渡すことです。SCHIB のフォーマットは以下のとおりです。



パス管理制御ワード (Path-Management-Control Word)

SCHIB のワード 0-6 には、パス管理制御ワード (PMCW) が入ります。サブチャンネルが有効なとき (15-4 ページの『装置番号有効 (V)』を参照)、PMCW のフォーマットは、図15-1 に示すとおりです。

PMCW のフォーマットは以下のとおりです。

0	Interruption Parameter										
1	00	ISC	000	E	LM	MM	D	T	V	Device Number	
2	LPM			PNOM			LPUM		PIM		
3	MBI					POM		PAM			
4	CHPID-0			CHPID-1			CHPID-2		CHPID-3		
5	CHPID-4			CHPID-5			CHPID-6		CHPID-7		
6	00000000			00000000			00000000		00000000 S		
	0	8	16	24	31						

図 15-1. PMCW のフォーマット

割り込みパラメーター (Interruption Parameter):

ワード 0 のビット位置 0-31 には、割り込みコードのワード 1 に格納されている割り込みパラメーターが入ります。割り込みパラメーターは、START SUB-

CHANNEL および MODIFY SUBCHANNEL を使用して任意の値に設定できます。割り込みパラメーターの初期値は 0 です。

入出力割り込みサブクラス・コード (ISC): ワード 1 のビット 2-4 は 0-7 の範囲内の符号なしの 2 進整数で、これは、構成内の各 CPU の制御レジスタ 6 のサブクラス・マスク・ビットのビット位置に対応します。CPU の制御レジスタ 6 の該当するマスク・ビットの設定によって、その CPU がこのサブチャンネルに関連した割り込み要求を認識するかどうか制御されます (16-4 ページの『割り込みの優先順位』を参照)。ISC は、MODIFY SUBCHANNEL を使用して任意の値に設定できます。ISC の初期値は 0 です。

予約済み: ワード 1 のビット 0、1、および 5-7 は予約済みであり、STORE SUBCHANNEL はこれらのビットに 0 を格納します。MODIFY SUBCHANNEL を実行するときには、ビット 0、1、6、および 7 が 0 でなければなりません。さもないと、オペランド例外が認識されます。ワード 1 のビット 5 は、MODIFY SUBCHANNEL の実行時には無視されます。

使用可能 (E): ワード 1 のビット 8 が 1 のときは、サブチャンネルがすべての入出力機能に対して使用可能になっていることを示します。E ビットが 0 のときは、装置が提示する状況はプログラムで使用できません。また、指定したサブチャンネルに対して MODIFY SUBCHANNEL または STORE SUBCHANNEL 以外の入出力命令を実行すると、条件コード 3 が設定されます。MODIFY SUBCHANNEL の実行時には、E ビットは 0 でも 1 でも構いません。初期時点では、どのサブチャンネルも使用可能ではありません。IPL が実行されると、IPL 入出力装置が使用可能にされます。

限界モード (LM): ワード 1 のビット 9 および 10 は、サブチャンネルの限界モード (LM) を定義します。チャンネル・サブシステムは、入出力操作のためにアドレス限界検査が呼び出されたときに、限界モードを使用します。(17-16 ページの『アドレス限界検査』を参照してください。)アドレス限界検査は、アドレス限界検査制御ビットにより制御されます。このビットは、START SUBCHANNEL の実行時に、操作要求ブロック (ORB) に入れてサブチャンネルに渡されます。(15-23 ページの『アドレス限界検査制御 (A)』を参照してください。) LM ビットの値は、データ転送時に使用されます。LM ビットの定義は下のとおりです。

ビット	ビット	機能
9	10	機能
0	0	初期設定値。このサブチャネルについては限界検査は行われません。
0	1	データ・アドレスは、現在のアドレス限界と同じかそれより大きくなければなりません。
1	0	データ・アドレスは、現在のアドレス限界より小さくなければなりません。
1	1	予約済み。

MODIFY SUBCHANNEL の実行時には、ビット位置 9 および 10 は、上記のビットの組み合わせのうち、最初の 3 つのどれであっても構いません。オペランドで予約済みのビットの組み合わせ指定すると、MODIFY SUBCHANNEL の実行時にオペランド例外が認識されます。

測定モード使用可能 (MM): ワード 1 のビット 11 およびビット 12 は、それぞれ、サブチャネルの測定ブロック更新モードおよび装置接続時間測定モードを使用可能にします。MODIFY SUBCHANNEL の実行時には、これらのビットはどのような値であっても構いません。初期時点では、どちらの測定モードも使用可能にされていません。これらのビットの定義は以下のとおりです。

ビット	機能
11	測定ブロック更新可能:
0	初期設定値。サブチャネルでは、測定ブロック更新は使用可能になっていません。測定ブロック・データの格納は行われません。
1	サブチャネルでは、測定ブロック更新が使用可能になっています。測定ブロック更新モードがアクティブである場合は、サブチャネル・ログアウトにより記述される何らかのエラー条件が検出されない限り、チャネル・プログラムの実行が、サブチャネルで完了するかまたは延期された時点、または入出力装置でチャネル・プログラムの実行が完了した時点で、測定ブロック内に測定データが累積されます。(17-2 ページの『測定ブロック更新』を参照してください。)測定ブロック更新モードが非アクティブのときは、測定ブロック・データは格納されません。

ビット	機能
12	装置接続時間測定可能:
0	初期設定値。サブチャネルでは、装置接続時間測定は使用可能になっていません。拡張状況ワード (ESW) への装置接続時間間隔 (DCTI) の格納は行われません。
1	サブチャネルでは、装置接続時間測定が使用可能になっています。該当のサブチャネルについて、装置接続時間測定モードがアクティブであり、かつタイミング機構が提供されている場合は、サブチャネル・ログアウトにより記述されるようなエラー条件が検出されない限り、そのサブチャネルでチャネル・プログラムの実行が完了するか延期された後で、TEST SUBCHANNEL が実行されると、DCTI の値が ESW に格納されます。装置接続時間測定モードが非アクティブの場合は、測定値は ESW に格納されません。

上記で述べた測定モード使用可能ビット (MM) の意味が適用されるのは、サブチャネルのタイミング機構ビットが 1 であるときです。タイミング機構ビットが 0 のときは、下記の『タイミング機構』の項に示すように、MM ビットの効果が変化します。(測定モードについての詳細は、17-2 ページの『測定ブロック更新』および 17-7 ページの『装置接続時間測定』を参照してください。)

マルチパス・モード (D): ワード 1 のビット 13 が 1 のときは、サブチャネルが、1 つまたは一連の入出力操作を行うときにマルチパス・モードで稼働することを示します。選択可能なチャネル・パスが複数あるときには、マルチパス・モードで正しく稼働できるようにするには、関連の装置に動的再接続機構がインストールされていること、およびその装置がマルチパス・モード操作ができるようにセットアップされていることが必要です。マルチパス・モードで開始機能が実行されているときは、装置は、チャネル・サブシステムから、サブチャネルで選択可能として示されている任意のチャネル・パスを介して、チャネル・サービスにサービスを要求することができます (15-4 ページの『論理パス・マスク (LPM)』および 15-7 ページの『パス使用可能マスク (PAM)』を参照)。ビット 13 が 0 のときは、サブチャネルが、1 つまたは一連の入出力操作を行うときにシングル・パス・モードで稼働することを示します。シングル・パス・モードでは、装置が 1 つまたは一連の入出力操作の最初のコマンドを受け入れるときに使用されたチャネル・パスを使用して、開始機能の全体が実行されます。MODIFY SUBCHANNEL の実行時点では、D ビットは 0 でも 1 でも構いません。初期時点では、サブチャネルはシングル・パス・モードになっています。

タイミング機構 (T): ワード 1 のビット 14 が 1 のときは、目的のサブチャンネルについてチャンネル・サブシステム・タイミング機構が使用可能であり、2 つの測定モード使用可能ビット (MM) および SET CHANNEL MONITOR により制御されることを示します。ビット 14 が 0 のときは、そのサブチャンネルについてチャンネル・サブシステム・タイミング機能が使用可能でないことを示します。ビット 14 が 0 の場合は、サブチャンネルについての測定ブロック内に累積される測定データは、START SUBCHANNEL カウントのみです。上記の『測定モード使用可能』の項で述べたように、START SUBCHANNEL カウントの格納は、SET CHANNEL MONITOR のビット 11 により制御されます。同様に、T ビットが 0 のときは、サブチャンネルについて装置接続時間間隔 (DCTI) の値を測定することはできません。(17-2ページの『測定ブロック更新』および 17-7ページの『装置接続時間測定』を参照してください。)

装置番号有効 (V): ワード 1 のビット 15 が 1 のときは、装置番号フィールド (下記を参照) に有効な装置番号が含まれており、このサブチャンネルに関連した装置が物理的にインストールできることを示します。ビット 15 が 0 のときは、このサブチャンネルは有効ではなく、現在このサブチャンネルに関連付けられている入出力装置はなく、そして、SCHIB の他の定義済みのフィールドの内容はすべて予測不能であることを示します。

装置番号 (Device Number): ワード 1 のビット位置 16-31 には、このサブチャンネルに関連付けられている装置の装置番号を表す 4 桁の 16 進数を 2 進数で表した値が入ります。装置番号は、サブチャンネルに関連した装置がインストールされるときに、サブチャンネルとその装置に割り当てられるシステム固有のパラメータです。

論理パス・マスク (LPM): ワード 2 のビット 0-7 は、関連装置へのチャンネル・パスが論理的に使用可能かどうかを示します。LPM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。これらのビットの 1 つが 1 に設定されている場合、それに対応するチャンネル・パスが論理的に使用可能であることを示します。0 の場合は、対応するチャンネル・パスが論理的に使用可能でないことを示します。あるチャンネル・パスが論理的に使用可能でない場合は、そのチャンネル・パス用の専用従属関係が存在していない限り、チャンネル・サブシステムは、そのチャンネル・パスを使用して、消去、停止、再開、または開始機能の実行を開始することはできません。サブチャンネルに特定のチャンネル・パス用の専用従属関係が存在する場合は、消去、停止、再開、または開始機能を実行するときに、そのチャンネル・パスの論理的な可用性は無視されます。(15-10ページの『チャネ

ル・パス従属関係』を参照してください。) サブチャンネルがアイドル状態にある場合は、制御装置がチャンネル・サブシステムに警報状況を提示するための要求を開始する際には、常に、チャンネル・パスの論理的な可用性は無視されます。サブチャンネルに関連したチャンネル・パスの論理的な可用性を変更するには、SCHIB の中の対応する LPM ビットを設定してから、MODIFY SUBCHANNEL を実行するか、または、ORB の中の対応する LPM ビットを設定してから、START SUBCHANNEL を実行します。初期時点では、インストール済みのチャンネル・パスはすべて論理的に使用可能です。

パス非稼働マスク (PNOM): ワード 2 のビット 8-15 のいずれかが 1 のときは、そのビットに対応するチャンネル・パス上でパス非稼働条件が認識されたことを示します。PNOM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。チャンネル・サブシステムがパス非稼働条件を認識するのは、消去、停止、再開、または開始機能を実行するために装置を選択しようとしたときに、サブチャンネルで使用できるチャンネル・パス上で、サブチャンネルに関連した装置が稼働状態にないと認められた場合です。パス非稼働条件が認識された場合は、チャンネル・パスの状態は、サブチャンネルにとって稼働状態から非稼働状態に変わります。チャンネル・パスがサブチャンネルにとって稼働状態にあるのは、チャンネル・サブシステムが、消去、停止、再開、または開始機能を実行するために最後に行った装置選択の際に、そのチャンネル・パス上で関連装置が稼働状態であると認められたときです。チャンネル・パス上で装置が稼働状態にあると認められるのは、その装置が装置選択に応答した場合です。チャンネル・パスがサブチャンネルにとって非稼働状態にあるのは、チャンネル・サブシステムが、消去、停止、再開、または開始機能を実行するために最後に行った装置選択の際に、そのチャンネル・パス上で関連装置が非稼働状態であると認められたときです。ワード 2 のビット 8-15 のいずれかが 0 のときは、そのビットに対応するチャンネル・パス上でパス非稼働条件が認識されていないことを示します。

初期時点では、各サブチャンネルに関連した 8 個のチャンネル・パスのそれぞれは、対応するチャンネル・パスがインストール済みまたは使用可能かどうかに関係なく、いずれも稼働状態にあると見なされます。したがって、初期プログラム・ローディングの時点でパス非稼働条件が認識されていない限り、CLEAR SUBCHANNEL、HALT SUBCHANNEL、RESUME SUBCHANNEL、または START SUBCHANNEL の実行前に PMCW が格納された場合、その PMCW 内の PNOM はすべて 0 になります。

プログラミング上の注意: PNOM は、最新の消去、停止、再開、または開始機能の実行時に、パス非稼働条件が認識されたチャンネル・パスを示します。つまり、

PNOM は、最新の消去、停止、再開、または開始機能の実行時に、サブチャンネルに関連したチャンネル・パスのどれが、サブチャンネルにとって稼働状態から非稼働状態に変化したかを示します。ただし、チャンネル・パスがサブチャンネルにとって非稼働状態から稼働状態に変化したことは、POM に示されます。したがって、指定したサブチャンネルに関連したチャンネル・パスのどれかが、そのサブチャンネルにとって稼働状態にあるかどうかを判別するには、POM を調べる必要があります。

さらに、開始機能または再開機能の実行時に、チャンネル・パスがサブチャンネルにとって非稼働状態から稼働状態に変わったことをチャンネル・サブシステムが認識するのは、開始機能で指定されているかまたは再開機能で暗黙指定される最初のコマンドの開始シーケンスのときのみです。したがって、サブチャンネルに関連した装置は、コマンド・チューニングを続行するためにチャンネル・サブシステムに再接続したときに、現在サブチャンネルにとって稼働状態にないチャンネル・パスを使用することがあります。しかし、チャンネル・サブシステムは、そのチャンネル・パスがサブチャンネルにとって非稼働状態から稼働状態に変化したことを、POM 内に示すことはしません。

選択試行の前の POM の値と装置の 状態		選択試行の後の 指定ビットの値		
装置の 状態 ¹	POM	POM	PNOM ²	SCSW ビット N
OP	0	1	0	0
NOP	0	0	0	0
OP	1	1	0	0
NOP	1	0	1	1 ³

説明:

- 1 対応するチャンネル・パス上で認められる装置の状態。
- 2 サブチャンネルが延期されているときに開始機能または再開機能が実行された場合は、その実行時に装置選択が試行される前に、指定のサブチャンネルに既存のパス非稼働条件があれば、チャンネル・サブシステムはその条件をすべて消去します。
- 3 次回にサブチャンネルが中間状況のみ以外の場合状況保留状態になったときに TEST SUBCHANNEL が実行されると、サブチャンネルが延期されていない限り、プログラムに対して N ビット (SCSW のワード 0 のビット 15) が示され、サブチャンネルでは N 条件が消去されます。

NOP 対応するチャンネル・パス上で装置が非稼働状態にあります。

OP 対応するチャンネル・パス上で装置が稼働状態にあります。

図 15-2. 選択試行の後の POM、PNOM、および N ビットの値

最終使用パス・マスク (LPUM): ワード 2 のビット 16-23 は、チャンネル・サブシステムと装置間の通信または情報転送のための最後に使用されたチャンネル・パスを示します。LPUM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。次のいずれかに該当する場合は、LPUM のビットには、最後に使用されたチャンネル・パスを除き、すべて 0 が格納されます。

1. 装置が、開始機能または再開機能の最初のコマンドを受け入れた (16-13 ページの『活動制御 (AC)』を参照)。
2. 装置とチャンネル・サブシステムがアクティブに通信しているときに、実行中のチャンネル・プログラムに対して延期機能が実行された。
3. 装置から受け入れられた状況が割り込み条件として認識されたか、コマンド・チューニングを抑止する条件が認識された (16-2 ページの『割り込み条件』を参照)。
4. インターフェース制御チェック条件が認識され (16-27 ページの『インターフェース制御チェック』

を参照)、現在サブチャンネルにはサブチャンネル・ログアウト情報がない。

PMCW の LPUM フィールドには、最新の設定が入っています。LPUM の初期値は 0 です。

インストール済みパス・マスク (PIM): ワード 2 のビット 24-31 は、入出力装置へのチャンネル・パス 0-7 のどれが物理的にインストールされているかを示します。つまり、PIM は、物理的にインストールされているチャンネル・パスのチャンネル・パス ID (下記を参照) の妥当性を示します。PIM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。1 が格納されている PIM ビットは、それに対応するチャンネル・パスがインストールされていることを示します。0 が格納されている PIM ビットは、対応するチャンネル・パスがインストールされていないことを示します。システムがどのように構成されているかに関係なく、PIM には、常に、該当の入出力装置へのすべてのインストール済みパスが反映されます。したがって、PIM に示されるチャンネル・パスの中には、パス使用可能マスク (下記を参照) のビット設定に基づき、構成内では物理的に使用可能でないものもあります。PIM の初期値は、入出力装置への物理的にインストール済みのすべてのチャンネル・パスを示します。

測定ブロック指標 (MBI): ワード 3 のビット 0-15 は、測定ブロック更新モードがアクティブで (14-12ページの『SET CHANNEL MONITOR』を参照)、サブチャンネルでそのモードが使用可能にされているときに (15-3ページの『測定モード使用可能 (MM)』を参照)、測定ブロック更新機能を使用する指標値を形成します。測定ブロック指標が使用されるときは、右側に 5 個の 0 ビットが付加され、その結果が、SET CHANNEL MONITOR で指定されている測定ブロック起点アドレスに加算されます。このようにして計算されたアドレスは測定ブロック・アドレスと呼ばれ、測定データが格納される 32 バイト記憶域の先頭を示します。(17-3ページの『測定ブロック』を参照してください。) MODIFY SUBCHANNEL の実行時には、MBI にどのような値が入っていても構いません。初期値は 0 です。

パス稼働マスク (POM): ワード 3 のビット 16-23 は、対応するチャンネル・パス上の装置について最後に認識された稼働状態を示します。POM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。チャンネル・サブシステムが、消去、停止、再開、または開始機能を実行するために、最後に装置選択を試行したときに、あるチャンネル・パス上で関連装置が稼働状

態にあると認められた場合は、そのチャンネル・パスはサブチャンネルにとって稼働状態にあり、POM 内のそのチャンネル・パスに対応するビットは 1 になります。チャンネル・パス上で装置が稼働状態にあると認められるのは、その装置が装置選択に応答した場合です。また、MODIFY SUBCHANNEL が実行され、POM 内のそのチャンネル・パスに対応するビットに 1 が指定された場合も、そのチャンネル・パスはサブチャンネルにとって稼働状態にあります。

チャンネル・サブシステムが、消去、停止、再開、または開始機能を実行するために、最後に装置選択を試行したときに、あるチャンネル・パス上で関連装置が非稼働状態にあると認められた場合は、そのチャンネル・パスはサブチャンネルにとって非稼働状態にあり、POM 内のそのチャンネル・パスに対応するビットは 0 になります。また、MODIFY SUBCHANNEL が実行され、POM 内のそのチャンネル・パスに対応するビットに 0 が指定された場合も、そのチャンネル・パスはサブチャンネルにとって非稼働状態にあります。

消去、停止、再開、または開始機能を実行するために装置選択が試行されたときに、サブチャンネルにとって稼働状態にあるチャンネル・パス上で、サブチャンネルに関連した装置が非稼働状態にあると認められた場合は、チャンネル・サブシステムは、パス非稼働条件を認識します。その後で SCSW が格納された場合は、ワード 0 のビット 15 は、パス非稼働条件を示す 1 に設定されます。パス非稼働条件が認識された場合は、チャンネル・パスの状態は、サブチャンネルにとって稼働状態から非稼働状態に変わります。

チャンネル・パスがサブチャンネルにとって非稼働状態にあるときは、パス非稼働条件は認識できません。さらに、サブチャンネルにとって非稼働状態にあるチャンネル・パスが選択可能な場合もあります。チャンネル・サブシステムが、パス管理操作を行うときにそのチャンネル・パスを選択し、そして、装置選択を試行した時点でそのチャンネル・パス上で該当の装置が再び稼働状態にあると認められた場合は、そのチャンネル・パスの状態はサブチャンネルにとって非稼働状態から稼働状態に変わります。

MODIFY SUBCHANNEL の実行時には、POM にはどのような値が入っていても構いません。初期時点では、各サブチャンネルに関連した 8 個のチャンネル・パスのそれぞれは、対応するチャンネル・パスがインストール済みまたは使用可能かどうかに関係なく、いずれも稼働状態にあると見なされます。したがって、初期プログラム・ローディングの時点でパス非稼働条件が認識されていない限り、CLEAR SUBCHANNEL、HALT SUBCHANNEL、RESUME SUBCHANNEL、または START SUBCHANNEL の実行前に PMCW が格納さ

れた場合、その PMCW 内の POM は全桁 1 になります。

パス使用可能マスク (PAM): ワード 3 のビット 24-31 は、インストール済みのチャンネル・パスの物理的な可用性を示します。PAM の各ビットは、SCHIB のワード 4 および 5 の関連バイト内にある CHPID の相対ビット位置と、1 対 1 で対応しています。PAM ビットが 1 のときは、それに対応するチャンネル・パスが、目的の装置にアクセスするために物理的に使用可能であることを示します。PAM ビットが 0 のときは、それに対応するチャンネル・パスが、目的の装置にアクセスするために物理的に使用可能でないことを示します。あるチャンネル・パスが物理的に使用可能でないときでも、モデルや障害の範囲によっては、チャンネル・パス・リセット機能の実行時に、そのチャンネル・パスが使用されることがあります。物理的に使用可能なチャンネル・パスは、システムの再構成の結果として、またはチャンネル・パス・リセット機能の実行結果として、物理的に使用不可能になることがあります。PAM の初期値は、初期化の時点で入出力装置が物理的にアクセスできるチャンネル・パスのセットを反映しています。

注: チャンネル・パスの可用性が変化すると、そのチャンネル・パスにアクセスできるすべてのサブチャンネルが影響を受けます。この章では、チャンネル・パスの可用性状況に関連して PAM ビットの設定について触れるときは、説明を完結にするために、影響を受けるすべてのサブチャンネルの対応する PAM ビットではなく、単一の PAM ビットのみを取り上げることにします。

チャンネル・パス ID (CHPID): ワード 4 および 5 には、PIM のチャンネル・パス 0-7 に対応する 8 個の 1 バイトのチャンネル・パス ID が入ります。対応する PIM ビットが 1 であれば、CHPID は有効です。有効な各 CHPID には、関連の入出力装置にアクセスするために使用できる制御装置への物理チャンネル・パスの ID が入ります。システム内の各物理チャンネル・パスに、それぞれ固有の CHPID が割り当てられます。

同一の物理チャンネル・パスにより複数の異なる装置にアクセスできる場合、それらの装置は、それぞれに対応するサブチャンネル内に同じ CHPID 値を持つこととなります。ただし、各サブチャンネル内では、CHPID 値は CHPID フィールド 0-7 の中の異なる位置に現れることがあります。

同じチャンネル・パスのセットを共用するサブチャンネルについては、それぞれに対応する PIM ビットが 1 に設定されます。これらのチャンネル・パスのチャンネル・パス ID (CHPID) はすべて同じであり、SCHIB 内の同じ対応位置を占めます。

予約済み: ワード 6 のビット 0-30 は予約済みであり、STORE SUBCHANNEL はこれらのビットに 0 を格納します。MODIFY SUBCHANNEL の実行時には、これらのビットは 0 でなければなりません。さもないと、オペランド例外が認識されます。

並行センス (S): ワード 6 のビット 31 が 1 のときは、サブチャンネルが並行センス・モードになっていることを示します。サブチャンネルが並行センス・モードになっているときは、サブチャンネルが警報状況を伴う状況保留状態になり、装置が受け取った状況バイトに装置チェック標識が含まれていると、チャンネル・サブシステムは、関連装置からセンス情報を検索し、その情報を拡張制御ワードに入れようとしています。

並行センス機能がインストールされていない場合は、MODIFY SUBCHANNEL の実行時には、SCHIB オペランドのワード 6 のビット 31 は 0 でなければなりません。さもないと、オペランド例外が認識されます。

サブチャンネル状況ワード (Subchannel-Status Word)

SCHIB のワード 7-9 には、SCSW のコピーが入ります。SCSW のフォーマットについては、16-6 ページの『サブチャンネル状況ワード』で説明します。SCSW は、STORE SUBCHANNEL または TEST SUBCHANNEL の実行時に格納されます (14-17 ページの『STORE SUBCHANNEL』および 14-19 ページの『TEST SUBCHANNEL』を参照)。

モデル依存区域 (Model-Dependent Area)

SCHIB のワード 10-12 には、モデル依存情報が入ります。

変更可能フィールドの要約

15-8 ページの図 15-3 は、装置番号有効ビットが 1 であるサブチャンネル内のフィールドの初期設定値と、各フィールドが何により変更されるかを示しています。

STORE SUBCHANNEL が実行され、指定のサブチャンネルがアイドル状態にあるときは、PMCW のすべてのフィールドに意味のある情報が含まれています。

STORE SUBCHANNEL が実行されるときに、装置番号有効ビットが 1 であれば、チャンネル・サブシステムが変更しないサブチャンネル・フィールドには、有効な情報が含まれています。チャンネル・サブシステムが変更できるサブチャンネル・フィールドの妥当性は、STORE SUBCHANNEL の実行時点でのサブチャンネルの状態によって決まります。

サブチャンネル・フィールド	初期値 ¹	変更のためにプログラムが使用する命令	チャンネル・サブシステムによる変更 ²
割り込みパラメータ	0	MSCH、SSCH	なし
入出力割り込みサブクラス・コード	0	MSCH	なし
使用可能 (E)	0	MSCH	なし
限界モード (LM)	0	MSCH	なし
測定モード (MM)	0	MSCH	あり ³
マルチパス・モード (D)	0	MSCH	なし
タイミング機構 (T)	インストール時の値 ⁴	なし	なし
装置番号有効 (V)	インストール時の値 ⁴	なし	なし
装置番号 (Device Number)	インストール時の値 ⁴	なし	なし
論理パス・マスク (LPM)	インストール済みパス・マスクの値	MSCH、SSCH	なし
パス非稼働マスク (PNOM)	0	CSCH、SSCH、RSCH ⁵	あり
最終使用パス・マスク (LPUM)	0	CSCH	あり
インストール済みパス・マスク (PIM)	インストール時の値 ⁴	なし	なし
測定ブロック指標 (MBI)	0	MSCH	なし
パス稼働マスク (POM)	1	CSCH、MSCH、RSCH ⁵	あり
パス使用可能マスク (PAM)	インストール時の値 ^{4,6}	なし	あり ⁶
チャンネル・パス ID 0-7 (CHPID)	インストール時の値 ⁴	なし	なし
並行センス (S)	0	MSCH	なし
サブチャンネル状況ワード (SCSW)	0	TSCH	あり
モデル依存区域	*	なし	*

図 15-3 (1/2). サブチャンネル・フィールドの変更

説明:

- * モデルにより異なります。
- 1 サブチャンネルが有効でない場合は、これらのフィールドは意味を持ちません。入出力システム・リセットが起きたときは、サブチャンネルの初期化が行われます。(第 17 章、『入出力サポート機能』の『入出力システム・リセット』のセクションを参照してください。) サブチャンネルがアイドル状態にあるときは、プログラムでは変更できないインストール済み値パラメーターの 1 つまたは幾つかが設定されることがあります。その場合は、プログラム変更可能フィールドは、すべてそれぞれの初期設定値に設定され、チャンネル報告によりその変更がプログラムに通知されます。(第 17 章、『入出力サポート機能』の『チャンネル報告ワード』のセクションを参照してください。)
- 2 動的構成変更または外部アクションの結果として、通常はチャンネル・サブシステムにより変更できないサブチャンネル・フィールドが、変更されることがあります。その場合は、その変更の時点で保留にされているチャンネル報告により、変更がプログラムに通知されます。
- 3 測定ブロック更新モードに関連した以下のエラー条件のいずれかが検出されたときは、チャンネル・サブシステムは、影響を受けるサブチャンネル内で測定ブロック更新モードを使用不可 (SCHIB のワード 1 のビット 11 が 0) にします。装置接続時間測定可能ビット (SCHIB のワード 1 のビット 12) が、チャンネル・サブシステムにより変更されることはありません。
 - 測定プログラム・チェック
 - 測定記憶保護チェック
 - 測定データ・チェック
 - 測定キー・チェック
- 4 この情報が入力されるのは、チャンネル・サブシステム構成が確立されるときです。
- 5 再開機能によりこのマスクが変更されるのは、RESUME SUBCHANNEL の実行時に、サブチャンネルが延期状態にあるときのみです。
- 6 チャンネル・サブシステムは、入出力システムの再構成または入出力システムの一部に生じた永続障害が原因でチャンネル・パスを分割または統合した結果としてシステム内に生じる変更を反映して、PAM を変更することがあります。

図 15-3 (2/2). サブチャンネル・フィールドの変更

プログラミング上の注意:

1. PNOM にバス非稼働条件が示されているチャンネル・パスがある場合、LPM を使用してそのようなバスを論理的に使用不可能にしておかないと、システムのパフォーマンスが低下することがあります。
2. 開始機能の実行時に、チャンネル・バス障害が認識されたため、チャンネル・バスが物理的に使用可能でなくなった場合は、開始機能が続行できなくなることがあります。つまり、プログラムは通知を受けることも受けないこともあり、サブチャンネルは、消去機能の実行により消去されるまで、サブチャンネル/装置アクティブ状態のままになっていることがあります。
3. プログラムが複数のサブチャンネルに同じ MBI を入れた場合は、チャンネル・サブシステム・モニター機構は、複数のサブチャンネルに関連した測定データで同じ記憶位置を更新します。その場合に測定データに格納される値は、予測不能です。(17-2ページの『測定ブロック更新』を参照してください。)
4. 入出力構成の変更(再構成)には、モデルによってさまざまな方法があります。再構成手順がチャンネル・バスの物理的な可用性に影響を与える場合は、そのチャンネル・バスにアクセスできるサブチャンネルを対象として STORE SUBCHANNEL を実行し、その後で SCHIB の PAM ビットを調べることにより、可用性の変化を確認することができます。
5. PNOM、POM、および N ビットの定義によれば、バス非稼働条件がプログラムに報告されるのは、対応する POM ビットが 1 に設定された後で、チャンネル・サブシステムが初めてその条件を検出したときのみです。

例えば、チャンネル・サブシステムが入出力装置で開始機能を開始しようとしているときに、選択可能なすべてのチャンネル・パスについて POM ビットが 1 であり、対応するすべてのチャンネル・パス上でその装置が非稼働状態にあると認められた場合は、チャンネル・サブシステムはサブチャンネルを状況保留にし、据え置き条件コード 3 を設定し、N ビットに 1 を格納します。SCHIB 内の PNOM は、非稼働と認められたチャンネル・パスを示します(対応する POM ビットが 0 に設定されています)。次回に START SUBCHANNEL が実行されると、チャンネル・サブシステムは、選択可能なすべてのチャンネル・パスからいずれかのチャンネル・パスを選択することにより、再度装置選択を試みます。この装置選択が失敗し、選択可能なすべてのチャンネル・パスを再度選択し終えた場合は、据え置き条件コード 3 が設定されますが、SCSW の N ビットは 0 になります。POM 内では、少なくとも、選択可能なチャネ

ル・パスに対応するビット位置は 0 になります。(「選択可能」という用語の意味については、15-11 ページの『チャンネル・パスの可用性』を参照してください。) SCSW の N ビットが 0 のときは、PNOM も 0 です。

6. プログラムでバス非稼働条件を検出する必要がある場合は、TEST SUBCHANNEL の実行した後で(この実行結果として、条件コード 0 が設定され、N ビットに 1 が格納される)、そして、サブチャンネルで新たな開始、再開、停止、または消去機能を実行する前に、PNOM を検査する必要があります。

チャンネル・パス従属関係

チャンネル・サブシステムは、サブチャンネルとチャンネル・パスの間の従属関係条件を確立します。サブチャンネルにおいて、1 つまたは一組のチャンネル・パスについて確立される従属関係の種類は、サブチャンネルと装置の状態、および、チャンネル・サブシステムと装置の間で転送される情報があればその情報の状態によって異なります。消去、停止、再開、または開始機能の実行時にバス管理がどのように扱われるかは、現在サブチャンネルとチャンネル・パスの間で認識されている従属関係(存在する場合)の種類によって決まります。

サブチャンネルで消去機能を実行すると、すべてのチャンネル・パスについて、現在そのサブチャンネル内に存在する従属関係がすべて消去されます。

チャンネル・パス・リセット機能を実行すると、すべてのサブチャンネル内で、現在そのチャンネル・パスについて存在している従属関係がすべて消去されます。

チャンネル・パスが物理的に使用可能でなくなると、そのチャンネル・パスにアクセスできるすべてのサブチャンネルの中で、以前の従属関係条件を示すすべての内部標識が消去されます。

作業従属関係

サブチャンネルがチャンネル・パスとの作業従属関係を持つのは、そのチャンネル・パス上でサブチャンネルが装置アクティブになったときです。作業従属関係が確立されると、チャンネル・サブシステムは、サブチャンネルが装置アクティブでなくなるか、または専用従属関係が認識されるまで(いずれか早い方)、サブチャンネルにそのチャンネル・パスについて作業従属関係を維持します。専用従属関係が認識されない限り、装置がマルチパス・モードで稼働するように指定されていれば(つまり、SCHIB のマルチパス・モード・ビットに 1 が格納されていれば)、1 つのチャンネル・パスに対する作業従属関係が、選

択可能なチャンネル・パスのセットにまで拡張されます。そうでない場合は、開始機能がチャンネル・パスについてのみ作業従属関係が維持されます。

1 つまたは一組のチャンネル・パスに対する作業従属関係が確立されると、サブチャンネルが装置アクティブでなくなるか、または専用従属関係が確立されるまで、その作業従属関係は変更されません。サブチャンネルがシングル・パス・モードで稼働している場合は、単一のパスについてのみ作業従属関係が維持されます。

サブチャンネルに作業従属関係が存在している間は、装置がマルチパス・モードで稼働するものとして指定されていない限り、その作業従属関係が維持されている 1 つのチャンネル・パスについてのみ、アクティブ従属関係が発生します。装置がマルチパス・モードで稼働するものとして指定されているときは、装置が選択可能ではないチャンネル・パス上にあることを示す状況表示が原因で警報割り込み条件が認識された場合、その選択可能ではないチャンネル・パスについてもアクティブ従属関係が発生することがあります。

あるチャンネル・パスが物理的に使用可能でなくなった場合は、そのチャンネル・パスにアクセスできるすべてのサブチャンネルにおいて、作業従属関係が消去されます。

アクティブ従属関係

サブチャンネルとチャンネル・パスの間にアクティブ従属関係が確立されるのは、そのチャンネル・パスと入出力装置との間のアクティブ通信が開始される時点またはそれより前です。サブチャンネルは、一度に 1 つのチャンネル・パスに対してのみアクティブ従属関係を持つことができます。サブチャンネルが 1 つのチャンネル・パスに対するアクティブ従属関係を持っている間は、チャンネル・サブシステムは、他のチャンネル・パス上でその装置とアクティブに通信することはありません。チャンネル・サブシステムが、装置から割り込み条件を引き起こさない使用中終了標識を受理しても、その状況によってアクティブ通信が開始されることにはなりません。サブチャンネルにおいてチャンネル・パスに対するアクティブ従属関係が終了するのは、チャンネル・サブシステムが、そのチャンネル・パス上で入出力装置とアクティブに通信を行わなくなったときです。

作業従属関係は、アクティブ従属関係になることができます。

専用従属関係

あるチャンネル・パスが物理的に使用可能（つまり対応する PAM ビットが 1）である場合は、そのチャンネル・パスについて専用従属関係が認識されることがあります。チャンネル・パスが物理的に使用可能でない場合は、そのチャンネル・パスについては専用従属関係は認識されません。チャンネル・サブシステムが、サブチャンネルにおいてチャンネル・パスについての専用従属関係を確立するのは、(1) サブチャンネルが警報状況を伴う状況保留状態になり、かつ装置チェック標識を含む装置状況が存在するが、(2) サブチャンネルに並行センス情報がないときです。専用従属関係は、次の開始機能、消去機能、またはチャンネル・パスリセット機能、あるいは次の再開機能（該当する場合）の実行後に、サブチャンネルが開始保留（延期になる場合を除く）または再開保留ではなくなるまで、維持されます。サブチャンネルが延期になった場合は、専用従属関係は、再開機能が開始され、サブチャンネルが再開保留ではなくなるまで、維持されます。消去機能またはチャンネル・パス・リセット機能が実行される場合を除き、サブチャンネルは、専用従属関係が終了した時点で作業従属関係を確立します。これが行われるのは、サブチャンネルが装置アクティブになったときです。サブチャンネルに、あるチャンネル・パスに対する専用従属関係が存在している間は、その専用従属関係条件が消去されるまでは、そのチャンネル・パスのみが選択可能です。

専用従属関係は、アクティブ従属関係になることができます。専用従属関係が存在している間は、同じチャンネル・パスについてのみアクティブ従属関係を確立できません。

あるチャンネル・パスにアクセスできるサブチャンネルで、現在存在している専用従属関係が消去されるのは、そのチャンネル・パスが物理的に使用可能でなくなるか、または、専用従属関係が存在するチャンネル・パス上で装置が非稼働状態にあると認められたときです。

チャンネル・パスの可用性

あるチャンネル・パスが物理的に使用可能でないときは、チャンネル・サブシステムは、基本入出力機能を実行するためにそのチャンネル・パスを使用せず、そのチャンネル・パス上で制御装置により開始される要求には応答しません（ただし、場合によっては、チャンネル・パス・リセット機能にはそのチャンネル・パスが使用されることがあります）。チャンネル・パスが物理的に使用可能でない場合は、STORE SUBCHANNEL の実行時に、該当の条件を示すために、PAM の対応するパス使用可能マスク・ビットが 0 に設定されます（15-7 ページの『パス使用可能マスク (PAM)』を参照）。さらに、チャンネル・パスが、STORE SUBCHANNEL に指定されているサブチ

チャンネルにとって物理的に使用可能でない場合は、そのチャンネル・パスを介して入出力装置にアクセスできるどのサブチャンネルにとっても、そのチャンネル・パスは物理的に使用可能ではありません。

サブチャンネルに、チャンネル・パスに対する専用従属関係が存在していない場合は、そのチャンネル・パスが論理的かつ物理的に使用可能であれば、つまり、STORE SUBCHANNEL の実行時に、LPM および PAM の中のそのチャンネル・パスに対応するビットに 1 が格納されていれば、そのチャンネル・パスは選択可能になります。サブチャンネル、そのチャンネル・パスに対する専用従属関係が存在している場合は、そのチャンネル・パスのみが選択可能であり、対応する LPM ビットの設定は無視されます。チャンネル・パスが現在使用中であり、そのチャンネル・パスに対する専用従属関係がサブチャンネルに存在する場合は、そのチャンネル・パスが使用されなくなるまで、装置の選択は遅延されます。

図15-4 に示すように、関連装置への 8 つの論理パスの可用性状況は、PIM、PAM、および LPM の中の対応するビットの階層的な配列と、チャンネル・サブシステムが認識した既存の条件 (存在する場合) によって決まります。

ビット「n」の値			チャンネル・パス条件 ¹	チャンネル・パス状態
PIM	PAM	LPM		
0	0 ²	-	X	未インストール
1	0	-	X	物理的に使用可能でない
1	1	0 ³	X	論理的に使用可能でない
1	1	1 ³	アクティブ	選択可能 ⁴
1	1	1	非アクティブ	選択可能

説明:

- このビット値には意味はありません。
- ¹ チャンネル・パスが、入出力装置とのアクティブな通信で使用されているものとして認識された場合は、チャンネル・パス条件はアクティブと見なされます。そうでない場合は、条件は非アクティブと見なされます。
- ² PAM のビットは、それに対応する PIM のビットの値が 0 のときは、1 の値をとることはできません。
- ³ サブチャンネルにおいてチャンネル・パスへの専用従属関係が存在する場合は、ビットの状態は無視され、そのチャンネル・パスは選択可能と見なされます。
- ⁴ チャンネル・パス終了条件が認識されたときは、チャンネル・パスはアクティブと見なされることがあります。
- X この条件には意味はありません。

図 15-4. PIM、PAM、および LPM の値のパス条件とパス可用性状況

制御装置タイプ

15-13ページの『消去機能』、15-14ページの『停止機能』、および 15-17ページの『開始機能と再開機能』の説明には、タイプ 1、タイプ 2、およびタイプ 3 の制御装置が出てきます。これらの制御装置タイプの説明については、システム・ライブラリー資料の「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974) を参照してください。この定義の目的上、直列入出力インターフェースに接続される制御装置は、すべてタイプ 2 の制御装置と見なされます。

消去機能

CLEAR SUBCHANNEL の実行後に、チャンネル・サブシステムは消去機能を実行します。消去機能が実行されると、(1) パス管理操作が行われ、(2) サブチャンネルでフィールドが変更され、(3) 関連の装置に消去信号が送られ、そして、(4) サブチャンネルは、消去機能の完了を示す状況保留状態にされます。

消去機能: パス管理

消去機能の一環として、関連のサブチャンネルに関するチャンネル・パス条件を調べるため、および関連の装置に消去信号を出すために使用できるチャンネル・パスを選択するために、パス管理操作が行われます。

チャンネル・パス条件は以下の順序で検査されます。

1. チャンネル・サブシステムが、信号の送り先の装置とアクティブに通信しているか、またはその装置との間のアクティブ通信を確立しようとしている場合は、現在使用中のチャンネル・パスが選択されます。
2. チャンネル・サブシステムが、信号の送り先の装置から使用中終了標識 (これは割り込み条件の認識を引き起こしません) を受け入れている途中であり、かつ関連のサブチャンネルがどのチャンネル・パスに対しても従属関係を持っていない場合は、使用中のチャンネル・パスが選択されます。
3. 関連のサブチャンネルが、いずれかのチャンネル・パスに対する専用従属関係を持っている場合は、そのチャンネル・パスが選択されます。
4. 関連のサブチャンネルが、1 つまたは複数のチャンネル・パスに対する作業従属関係を持っている場合は、それらのチャンネル・パスの 1 つが選択されません。
5. 関連のサブチャンネルがどのチャンネル・パスに対する従属関係も持っておらず、最後に使用されたチャンネル・パスが示されており、そのチャンネル・パスが選択可能である場合は、そのチャンネル・パスが選択されます。そのチャンネル・パスが選択可能でない場合は、どのチャンネル・パスも選択されないか、または、選択可能なチャンネル・パスのセットがあれば、その中の 1 つが選択されます (最後に使用されたチャンネル・パスが示されていない場合と同じです)。
6. 関連のサブチャンネルがどのチャンネル・パスに対する従属関係も持っておらず、最後に使用されたチャンネル・パスが示されており、かつ、選択可能なチャンネル・パスが 1 つ以上存在する場合は、それらのチャンネル・パスの 1 つが選択されます。

上記で述べたチャンネル・パス条件のどれも該当しない場合は、どのチャンネル・パスも選択されません。

項目 4、所定の条件下での項目 5、および項目 6 の場合は、チャンネル・サブシステムは、一組のチャンネル・パスの中の 1 つを選択します。これらの場合、チャンネル・サブシステムは、以下の条件に該当しないチャンネル・パスを選択しようとします。

1. そのチャンネル・パスについてチャンネル・パス終了条件が存在する。
2. そのチャンネル・パスに対して他のサブチャンネルがアクティブ従属関係を持っている。
3. 信号の送り先の装置がタイプ 1 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する従属関係を持っている。ただし、その従属関係が作業従属関係であり、そのサブチャンネルで 1 次状況が受理されている場合を除きます。
4. 信号の送り先の装置がタイプ 3 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する専用従属関係を持っている。

消去機能: サブチャンネル変更

消去機能の実行時に、サブチャンネルにおいてパス管理制御標識が変更されます。実際にこの変更が行われるのは、チャンネル・パスの選択を試みた後で、かつ消去信号の送り先の装置を選択しようとする前です。変更されるパス管理制御標識は以下のとおりです。

1. サブチャンネルにおける 8 個すべてのチャンネル・パスの状態が、サブチャンネルにとって稼働状態に設定されます。
2. 最終使用パス標識がリセットされ、最終使用チャンネル・パスがないことが示されます。
3. パス非稼働条件があれば、リセットされます。

消去機能: 信号発信と完了

チャンネル・パスの選択を試み、パス管理制御フィールドを変更した後で、条件が許せば、チャンネル・サブシステムは、消去信号の送り先の装置を選択しようとします。(17-8ページの『消去信号』を参照してください。) サブチャンネルおよび選択されたチャンネル・パス (選択されている場合) に関連した条件によって、(1) 消去信号の発信を試みるかどうか、および、(2) 消去信号発信の試みが成功するかどうかが決まります。これらの条件とは無関係に、サブチャンネルは、次に状況保留状態に設定さ

れ、消去機能の実行は完了します。以下、これらの条件と、消去機能へのそれぞれの影響について説明します。

消去信号の発信が試みられない: 以下のいずれかの条件に該当する場合は、チャンネル・サブシステムは装置への消去信号の発信を試みません。

1. どのチャンネル・パスも選択されなかった。(15-13ページの『消去機能: パス管理』を参照してください。)
2. 選択されたチャンネル・パスが、もはや選択可能ではない。
3. 選択されたチャンネル・パスについてチャンネル・パス終了条件が存在する。
4. 選択されたチャンネル・パスは、現在別の装置とのアクティブ通信のために使用中である。
5. 信号の送り先の装置がタイプ 1 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する従属関係を持っている。ただし、その従属関係が作業従属関係であり、そのサブチャンネルで 1 次状況が受理されている場合を除きます。
6. 信号の送り先の装置がタイプ 3 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する専用従属関係を持っている。

上記のいずれかの条件が存在する場合は、サブチャンネルは消去保留のままになり、状況保留が設定され、そして消去機能の実行は完了します。

消去信号発信の試みが成功しない: チャンネル・サブシステムが装置に対して消去信号を発信しようとしたときに、以下の条件が原因でその試みが成功しないことがあります。

1. チャンネル・サブシステムが消去信号を発信する入出力装置を選択しようとしたときに、制御装置または入出力装置が使用中条件を示す信号を発信した。
2. チャンネル・サブシステムが消去信号を発信する入出力装置を選択しようとしたときに、パス非稼働条件が認識された。
3. チャンネル・サブシステムが消去信号を発信しようとしたときに、エラー条件が検出された。

上記のいずれかの条件が存在するときに、チャンネル・サブシステムが、消去信号発信の試みが成功しなかったと判断するか、またはその試みが成功したかどうかを判断できない場合は、サブチャンネルは消去保留のままに

り、状況保留が設定され、そして消去機能の実行は完了します。

消去信号発信の試みが成功した: チャンネル・サブシステムが、消去信号発信の試みが成功したと判断したときは、サブチャンネルは消去保留ではなくなり、状況保留が設定され、消去機能の実行は完了します。サブチャンネルが状況保留になったときは、関連の装置との間の入出力操作 (もしあれば) は終了しています。

プログラミング上の注意: 消去機能の実行が終わると、装置からチャンネル・サブシステムに提示されたゼロ以外の状況 (制御装置終了のみの状況を除く) が、非送信請求警報状況としてプログラムに渡されます。制御装置終了のみまたはゼロの状況から成る非送信請求状況は、プログラムに渡されません。

停止機能

HALT SUBCHANNEL が実行されると、チャンネル・サブシステムは停止機能を実行します。停止機能が実行されると、(1) パス管理操作が行われ、(2) 関連装置に停止信号が送られ、(3) サブチャンネルは、停止機能の完了を示す状況保留状態にされます。

停止機能: パス管理

停止機能の一環として、関連のサブチャンネルに関するチャンネル・パス条件を調べるため、および関連の装置に停止信号を出すために使用できるチャンネル・パスを選択するために、パス管理操作が行われます。

チャンネル・パス条件は以下の順序で検査されます。

1. チャンネル・サブシステムが、信号の送り先の装置とアクティブに通信しているか、またはその装置との間のアクティブ通信を確立しようとしている場合は、現在使用中のチャンネル・パスが選択されます。
2. チャンネル・サブシステムが、信号の送り先の装置から使用中終了標識 (これは割り込み条件の認識を引き起こしません) を受け入れている途中であり、かつ関連のサブチャンネルがどのチャンネル・パスに対しても従属関係を持っていない場合は、使用中のチャンネル・パスが選択されます。
3. 関連のサブチャンネルが、いずれかのチャンネル・パスに対する専用従属関係を持っている場合は、そのチャンネル・パスが選択されます。
4. 関連のサブチャンネルが、1 つまたは複数のチャンネル・パスに対する作業従属関係を持っている場合は、それらのチャンネル・パスの 1 つが選択されます。

5. 関連のサブチャンネルがどのチャンネル・パスに対する従属関係も持っておらず、最後に使用されたチャンネル・パスが示されており、そのチャンネル・パスが選択可能である場合は、そのチャンネル・パスが選択されます。そのチャンネル・パスが選択可能でない場合は、どのチャンネル・パスも選択されないか、または、選択可能なチャンネル・パスのセットがあれば、その中の 1 つが選択されます (最後に使用されたチャンネル・パスが示されていない場合と同じです)。
6. 関連のサブチャンネルがどのチャンネル・パスに対する従属関係も持っておらず、最後に使用されたチャンネル・パスが示されておらず、かつ、選択可能なチャンネル・パスが 1 つ以上存在する場合は、それらのチャンネル・パスの 1 つが選択されます。

上記で述べたチャンネル・パス条件のどれも該当しない場合は、どのチャンネル・パスも選択されません。

項目 4 (15-14ページ)、所定の条件下での項目 5、および項目 6 の場合は、チャンネル・サブシステムは、一組のチャンネル・パスの中の 1 つを選択します。これらの場合、チャンネル・サブシステムは、以下の条件に該当しないチャンネル・パスを選択しようとします。

1. そのチャンネル・パスについてチャンネル・パス終了条件が存在する。
2. そのチャンネル・パスに対して他のサブチャンネルがアクティブ従属関係を持っている。
3. 信号の送り先の装置がタイプ 1 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する従属関係を持っている。ただし、その従属関係が作業従属関係であり、そのサブチャンネルで 1 次状況が受理されている場合を除きます。
4. 信号の送り先の装置がタイプ 3 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する専用従属関係を持っている。

停止機能: 信号発信と完了

チャンネル・パスの選択を試みた後で、条件が許せば、チャンネル・サブシステムは、停止信号の送り先の装置を選択しようとします。(17-8ページの『停止信号』を参照してください。) サブチャンネルおよび選択されたチャンネル・パス (選択されている場合) に関連した条件によって、(1) 停止信号の発信を試みるかどうか、(2) 停止信号発信の試みが成功するかどうか、および、(3) 停止機能を完了するためにサブチャンネルが状況保留にされるかどうかが決まります。以下、これらの条件と、停止機能へのそれぞれの影響について説明します。

停止信号の発信が試みられない: 以下のいずれかの条件に該当する場合は、チャンネル・サブシステムは装置への停止信号の発信を試みません。

1. どのチャンネル・パスも選択されなかった。(15-14ページの『停止機能: パス管理』を参照してください。)
2. 選択されたチャンネル・パスが、もはや選択可能ではない。
3. 選択されたチャンネル・パスについてチャンネル・パス終了条件が存在する。
4. 関連のサブチャンネルが、中間状況のみ以外の状況を伴って状況保留になっている。
5. 信号の送り先の装置がタイプ 1 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する従属関係を持っている。ただし、その従属関係が作業従属関係であり、そのサブチャンネルで 1 次状況が受理されている場合を除きます。
6. 信号の送り先の装置がタイプ 3 の制御装置に接続されており、かつ、同じ制御装置に接続されている別の装置用のサブチャンネルが同じチャンネル・パスに対する専用従属関係を持っている。

項目 3、5、または 6 で述べた条件が存在する場合は、その条件が解消されるまで、関連のサブチャンネルは停止保留のままになります。条件が解消されると (チャンネル・パス終了条件の場合は、RESET CHANNEL PATH の実行結果として条件が解消されると)、チャンネル・サブシステムは、装置に対して停止信号を発信しようとします。

前記の条件のうち上記以外の条件が存在する場合は、サブチャンネルは停止保留のままになり、状況保留が設定され、そして停止機能は完了します。

停止信号発信の試みが成功しない: チャンネル・サブシステムが装置に対して停止信号を発信しようとしたときに、以下の条件が原因でその試みが成功しないことがあります。

1. チャンネル・サブシステムが停止信号の送り先の入出力装置を選択しようとしたときに、制御装置または入出力装置が使用中条件を示す信号を発信した。
2. チャンネル・サブシステムが停止信号の送り先の入出力装置を選択しようとしたときに、パス非稼働条件が認識された。
3. チャンネル・サブシステムが停止信号を発信しようとしたときに、エラー条件が検出された。

制御装置または入出力装置が使用中条件の信号を発信した場合は (項目 1)、サブチャンネルは、使用中を示す内部標識がリセットされるまで停止保留のままになります。その場合は、チャンネル・サブシステムは、装置に対する停止信号を再度発信しようとしません。

前記の条件のうち上記以外のいずれかの条件が存在するときに、チャンネル・サブシステムが、停止信号発信の試みが成功しなかったと判断するか、またはその試みが成功したかどうかを判別できない場合は、サブチャンネルは停止保留のままになり、状況保留が設定され、そして停止機能は完了します。

停止信号発信の試みが成功した: チャンネル・サブシステムが、停止信号発信の試みが成功し、終了状況 (該当する場合) がサブチャンネルで受信されていると判断したときは、サブチャンネルは停止保留ではなくなり、状況保留が設定され、停止機能は完了します。サブチャンネルが状況保留になったときは、関連の装置との間の入出力操作 (もしあれば) は終了しています。以下、サブチャンネルで終了状況が受信されるかどうかを決定する条件と、入出力装置における停止信号の影響について説明します。

停止機能の実行時に、サブチャンネルがサブチャンネル/装置アクティブであるか、または装置アクティブのみである場合は、(1) 装置が終了状況を提示したか、または (2) チャンネル・サブシステムが終了状況が無効であると判断したために、サブチャンネルが状況保留にされるまで、その状態が続きます。サブチャンネルが、アイドル、開始保留、開始保留と再開保留、延期、または延期と再開保留の状態にあるとき、あるいは、コマンド・チューニングの実行中に、装置終了が受信された後で、次のコマンドが装置に転送される前に停止信号が発信されたときは、装置では何の操作も進行せず、したがって、装置は停止信号を受信しても何の状況も生成しません。サブチャンネルが、サブチャンネル・アクセスでも、中間状況を伴う状況保留状態でもなく、かつ、装置への停止信号を発信しようとしたときにエラーが検出されていない場合は、停止信号を発信した後で、状況保留のみを示す割り込み条件が生成されます。

停止信号が装置に与える影響には、装置のタイプや状態によって異なる部分もあります。アクティブでない装置や、チャンネル・パスを介したデータ転送を行わない機械的な操作 (テープの巻き戻しやディスク・アクセス・メカニズムの位置決めなど) を行っている装置に対する停止信号の影響は、制御装置または入出力装置のモデルによって異なります。装置が、所要時間を予測できないタイプの操作や、チャンネル・パスを介したデータ転送を伴う操作を行っている場合は、制御装置は、停止信号を、操作を終了させる指示と解釈します。装置での保留状況条件はリセットされません。制御装置は、停止信号を認

識すると、ただちに、正常な終了点に達するまで、チャンネル・サブシステムとのすべての通信を終了します。次に、制御装置は、生成された状況を提示するための選択を行うよう、チャンネル・サブシステムに要求します。

サブチャンネルが入出力操作のデータ転送部分に関与している場合は、停止機能の実行時点でデータ転送は終了し、装置は論理的にチャンネル・パスから切断されます。一連の入出力操作を実行しているサブチャンネルに対して停止機能が実行される場合に、装置がすでに現在の入出力操作についてチャンネル終了を提示しているときは、チャンネル・サブシステムは、装置を切断させ、コマンド・チューニングまたはコマンド再試行を抑制させます。サブチャンネルが装置との間で一連の入出力操作を行っており、コマンド・チューニングの実行中に、前の入出力操作についての装置終了が受信された後で、かつ次のコマンドが装置に転送される前に、停止信号が発信された場合は、その発信直後に、サブチャンネルは 1 次状況と 2 次状況を伴う状況保留にされます。この場合は、SCSW の装置状況フィールドには 0 が入ります。開始保留状態のサブチャンネルに対して停止機能が実行され、開始機能が開始される前に停止保留条件が認識された場合は、開始機能の開始は試みられず、サブチャンネルは、装置への信号が発信された後で状況保留になります。

サブチャンネルが関連装置との間で入出力操作を行っていないときは、その装置が選択され、その装置からの応答があると、停止信号の発信が試みられます。サブチャンネルが装置アクティブ状態にある場合は、停止された装置から装置終了状況を受信するまでは、サブチャンネルは状況保留になりません。サブチャンネルが、サブチャンネル/装置アクティブでも装置アクティブでもない場合は、サブチャンネルは、装置を選択し停止信号を発信した後で、ただちに状況保留になります。後者の場合、SCSW の状況保留ビットは 1 に設定されます (16-17 ページの『状況保留 (ビット 31)』を参照)。

停止機能の実行によって入出力操作を中止すると、2 つの異なる割り込み条件が起きることがあります。

第 1 の割り込み条件が起きるのは、装置がチャンネル終了条件を生成したときです。チャンネル・サブシステムは、この条件を装置からの他の割り込み条件と同様に取り扱います。ただし、関連の SCSW の中のコマンド・アドレスが、入出力操作が終了した位置を指している点と、サブチャンネル状況ビットが、検出された異常条件を反映していることがあるという点が異なります。操作の対象として指定されたすべてのデータが転送されてしまう前に停止信号が発信された場合は、現 CCW 内の SLI フラグの制御に従って、誤った長さが示されます。関連の SCSW のカウント・フィールド内の値は、予測不能です。

第 2 の割り込み条件が起きるのは、チャンネル終了割り込み条件とともに装置終了状況が提示されなかった場合です。その場合は、関連の SCSW 中のサブチャンネル・キー・フィールド、コマンド・アドレス・フィールド、およびカウント・フィールドは、意味を持ちません。

HALT SUBCHANNEL が入出力操作を終了させるときの終了方法は、カウントが使い尽くされたときやプログラム・エラーが検出されたときの終了方法と異なり、関連装置からサービス要求を受信されたかどうか依存することはありません。

プログラミング上の注意:

1. HALT SUBCHANNEL により操作が中止された後で、サブチャンネルが、1 次状況、1 次状況と 2 次状況、または 2 次状況を伴う状況保留状態にあるときは、カウント・フィールドに示される転送済みデータの範囲は、予測不能です。
2. パス管理操作で選択されたパスについて、チャンネル・パス終了条件が存在しているときは、その条件が解消されるまで、停止機能は保留のままになります。条件が解消されるまでは、他のチャンネル・パスが選択可能になっていても、関連のサブチャンネルを使用して入出力操作を行うことはできません。
CLEAR SUBCHANNEL を実行すれば、停止保留条件を終了させて、サブチャンネルを使用可能にすることができます。

開始機能と再開機能

START SUBCHANNEL および RESUME SUBCHANNEL の実行後に、チャンネル・サブシステムは、それぞれ開始機能と再開機能を実行して、関連装置との入出力操作を開始します。開始機能または再開機能が実行されると、(1) パス管理操作が行われ、(2) 関連装置との間で 1 つまたは一連の入出力操作が行われ、(3) サブチャンネルは、開始機能の完了を示す状況保留状態にされます。(開始機能の完了については、16-1 ページの第 16 章、『入出力割り込み』で説明します。) 開始機能は、START SUBCHANNEL のオペランドとして指定された ORB の中に示されているチャンネル・プログラムの実行を開始します。これに対して、再開機能は、延期されているチャンネル・プログラムがある場合に、延期の原因となった CCW からそのプログラムの実行を開始します。その他の点では、再開機能は開始機能と同じことを行います (16-13 ページの『再開保留 (ビット 20)』を参照)。

開始機能と再開機能: パス管理

チャンネル・サブシステムは、開始機能または再開機能の実行中にパス管理操作を行います。この操作では、装置選択に使用できるチャンネル・パスが選択され、その装置で入出力操作が開始されます。実行されるアクションは以下のとおりです。

1. 現在、サブチャンネルが開始保留および装置アクティブの状態にある場合は、前の開始機能についての 2 次状況が関連装置から受理され、サブチャンネルが開始保留のみの状態になるまで、開始機能はサブチャンネルで保留のままになります。2 次状況が受理され、その状況が警報割り込み条件を示していなければ、サブチャンネルは状況保留にはされず、保留中の開始機能の実行が開始されます。状況が警報割り込み条件を示している場合は、サブチャンネルは、2 次状況および警報状況を伴う状況保留になり、保留中の開始機能は開始されず、据え置き条件コード 1 が設定され、開始保留ビットは 1 のままになります。現在サブチャンネルが開始保留のみの状態にある場合は、以下に述べるように開始機能の実行が開始されます。
2. サブチャンネルに、いずれかのチャンネル・パスに対する専用従属関係が存在する場合は、チャンネル・サブシステムは装置選択のためにそのパスを選択します。装置選択しようとしたときに使用中条件が検出され、サブチャンネルに専用従属関係が存在している場合は、そのチャンネル・パスについて使用中を示す内部標識がリセットされるまで、開始機能は保留のままになります。使用中の内部標識がリセットされると、そのチャンネル・パス上で、保留中の開始機能の実行が開始されます。
3. 選択可能なチャンネル・パスがなく、また、サブチャンネル内にいずれかのチャンネル・パスに対する専用従属関係も存在しない場合は、どのチャンネル・パスも選択されません。
4. 選択可能なすべてのチャンネル・パスが試みられた結果、それらのチャンネル・パスの 1 つ以上が他の装置とのアクティブな通信に使用されているか、あるいは、チャンネル・サブシステムが、それらのチャンネル・パスの 1 つ以上について制御装置使用中条件または装置使用中条件を検出するか、またはその両方の条件を検出した場合は、いずれかのチャンネル・パス、制御装置、または入出力装置 (いずれか該当するもの) が使用可能になるまで、開始機能はサブチャンネルで保留のままになります。
5. (1) タイプ 1 の制御装置に接続されている装置へのチャンネル・パス上で開始機能を開始するときに、
(2) 同じチャンネル・パスに対する専用従属関係を持

つサブチャンネルに対応する他の装置、または同じチャンネル・パスに対する作業従属関係を持つサブチャンネルに対応する他の装置 (そのサブチャンネルについての 1 次状況がまだ受信されていないもの) が、同じ制御装置に接続されていない場合は、そのチャンネル・パスが選択可能であればそれが選択され、選択可能でなければ選択されません。ただし、その装置への他のチャンネル・パスが選択可能であり、上記で述べたいずれの従属関係も存在していない場合は、そのチャンネル・パスが選択されます。選択可能な他のチャンネル・パスがない場合は、開始機能または再開機能 (いずれか該当する方) は、チャンネル・パスがどれか使用可能になるまで保留のままになります。

6. 装置がタイプ 3 の制御装置に接続されており、同じチャンネル・パスに対する専用従属関係を持つサブチャンネルに対応する他の装置が少なくとも 1 つは同じ制御装置に接続されている場合は、選択可能な別のチャンネル・パスが選択されるか、または、別の装置に対する専用従属関係が消去されるまで、開始機能が保留のままになります。
7. チャンネル・パスが選択された後、保留中のチャンネル・プログラムの最初のコマンドの実行を開始するために装置を選択しているときに、使用中標識が受信された場合は、使用中の内部標識がリセットされるまでは、使用中標識の受信するときに使用していたチャンネル・パスは、その装置または制御装置 (受信したのが装置使用中標識か制御装置使用中標識かによって異なる) に再び使用されることはありません。
8. 開始機能で指定されたかまたは再開機能で暗黙に指定されている最初のコマンドの実行を開始するために装置を選択しようとしているときに (7 を参照)、使用中標識を受信した場合は、チャンネル・サブシステムは次のいずれかの処置をとります。
 - a. 装置がマルチパス・モードで稼働するものとして指定されており、受信したのが入出力装置使用中標識である場合は、使用中を示す内部標識がリセットされるまで、開始機能または再開機能は保留のままになります。(15-3 ページの『マルチパス・モード (D)』を参照してください。)
 - b. 装置がマルチパス・モードで稼働するように指定されており、受信したのが制御装置使用中標識である場合、または、装置がシングル・パス・モードで稼働するものとして指定されている場合は、チャンネル・サブシステムは、選択可能な代替チャンネル・パスを選んで装置の選択を試み、そして、開始または再開機能が開始されるか、または選択可能なすべてのチャンネル・パス

スで装置選択を試みてしまうまで、パス管理操作を続けます。選択可能なすべてのチャンネル・パスを選択し終わってもチャンネル・サブシステムにより開始または再開機能が開始されない場合は、使用中を示す内部標識がリセットされるまで、開始または再開機能は保留のままになります。

- c. サブチャンネルが専用従属関係を持っている場合は、アクション 2 (15-17 ページ) がとられません。
9. 最初のコマンドを転送するための選択を試みるときに、装置が非稼働と見なされ、対応するチャンネル・パスがサブチャンネルにとって稼働状態である場合は、パス非稼働条件が認識され、サブチャンネルにおけるそのチャンネル・パスの状態は、サブチャンネルにとって稼働状態から非稼働状態に変わります (15-4 ページの『パス非稼働マスク (PNOM)』を参照)。サブチャンネルにパス非稼働条件が存在する場合、その条件は、サブチャンネルが次に消去保留、開始保留、または再開保留 (サブチャンネルが延期されていた場合) のいずれかの状態になるまで持続し、その時点でその条件は消去されます。しかし、対応するチャンネル・パスがサブチャンネルにとって非稼働状態にある場合は、パス非稼働条件は認識されません。選択可能ないずれかのチャンネル・パス上で最初のコマンドを転送するための選択を試みるときに、装置が非稼働と見なされた場合は、次のいずれかの処置がとられます。
 - a. そのチャンネル・パスに対する専用従属関係が存在する場合は、そのチャンネル・パスが選択可能な唯一のチャンネル・パスです。したがって、開始機能または再開機能を開始しようとするそれ以上の試みは中止され、割り込み条件が認識されます。
 - b. 専用従属関係が存在せず、まだ試していない選択可能な代替チャンネル・パスが残っている場合は、それらのチャンネル・パスの 1 つが選択されて、装置選択が試みられ、最初のコマンドが転送されます。
 - c. 専用従属関係が存在せず、まだ試していない選択可能な代替チャンネル・パスもなく、そして試し済みのチャンネル・パスの少なくとも 1 つで装置が稼働状態にあると見なされる場合は、いずれかのチャンネル・パス、制御装置、または入出力装置 (いずれか該当するもの) が使用可能になるまで、開始または再開機能はサブチャンネルで保留状態のままになります。
 - d. 専用従属関係が存在せず、まだ試していない選択可能な代替チャンネル・パスもなく、そして、

試し済みのすべてのチャンネル・パスにおいて装置が非稼働と見なされた場合は、開始機能または再開機能を開始しようとするそれ以上の試みは中止され、割り込み条件が認識されます。

10. サブチャンネルがアクティブであり、ある装置で入出力操作を開始しようとしているときに、サブチャンネルでマルチパス・モードが指定されていない場合は、すべての装置選択は LPUM 標識に従って行われます。例えば、コマンド・チューニングが指定されている場合は、チャンネル・サブシステムは、一連の入出力操作を記述する最初および後続のすべてのコマンドを、同じチャンネル・パスを介して転送しません。

入出力操作の実行

チャンネル・パスが選択されると、チャンネル・サブシステムは、条件が許せば、関連の装置との入出力操作の実行を開始します。最初の入出力操作の開始と実行に続いて、さらに追加の入出力操作が実行されることがあります。チャンネル・サブシステムが実行できるコマンドには、書き込み (write)、読み取り (read)、逆方向読み取り (read backward)、制御 (control)、センス (sense)、センス ID (sense ID)、およびチャンネル内転送 (transfer in channel) の 7 つがあります。チャンネル内転送以外の各コマンドは、それぞれ対応する入出力操作を開始します。サブチャンネルでチャンネル・プログラムの実行が延期されている期間 (15-36 ページの『チャンネル・プログラム実行の延期』を参照) を除き、サブチャンネルは、最初のコマンドを受理してから、サブチャンネルで 1 次割り込み条件が認識されるまで、アクティブの状態になっています。最初のコマンドを受理する前に 1 次割り込み条件が認識された場合は、サブチャンネルはアクティブになりません。通常、1 次割り込み条件を起こすのは、チャンネル終了信号か、または、コマンド・チューニングの場合はチェーン内の最後の CCW を表すチャンネル終了信号です。(16-4 ページの『1 次割り込み条件』を参照してください。) 装置は、サブチャンネルで 2 次割り込み条件が認識されるまでアクティブになっています。通常、2 次割り込み条件を起こすのは、装置終了信号か、または、コマンド・チューニングの場合はチェーン内の最後の CCW を表す装置終了信号です。(16-4 ページの『2 次割り込み条件』を参照してください。)

プログラミング上の注意:

シングル・パス・モードでは、1 つまたは一連の入出力操作で使用するコマンド、データ、および状況の転送は、すべて、最初のコマンドが装置に転送されるときに経由したチャンネル・パスを使用して行われます。

装置に動的再接続機構がインストールされていれば、1 つまたは一連の入出力操作をマルチパス・モードで行うことができます。マルチパス・モードで実行するには、SCHIB のワード 1 のビット 13 に 1 を指定して、サブチャンネルに対して事前に MODIFY SUBCHANNEL を実行しておく必要があります。(15-3 ページの『マルチパス・モード (D)』を参照してください。) さらに、使用する装置のタイプに該当する幾つかのモデル依存コマンドを実行して、マルチパス・モードで稼働できるように装置をセットアップしておくことも必要です。マルチパス・モード操作を取り扱うための一般的な手順は、以下のとおりです。

1. セットアップ

- a. セットアップしようとしているマルチパス・グループのメンバーとなる各チャンネル・パスについて、装置でマルチパス・モード設定タイプのコマンドを正常に実行することが必要です。このようにしておかないと、マルチパス・モードの操作により、サブチャンネルで予測不能の結果が生じることがあります。何らかの理由により、装置への物理的に使用可能なチャンネル・パスの 1 つまたは幾つかをマルチパス・グループから除外する場合は、サブチャンネルがマルチパス・モードで稼働している間は、それらのチャンネル・パスは選択可能であってはなりません。特定のチャンネル・パスを選択不可能にするには、MODIFY SUBCHANNEL の実行前に SCHIB 内の対応する LPM ビットを 0 に設定するか、START SUBCHANNEL の実行前に ORB 内の対応する LPM ビットを 0 に設定します。
- b. マルチパス・モード設定タイプのコマンドが装置に転送されるときは、開始機能の実行のために代替チャンネル・パス選択が行われるのを防ぐために、1 つのチャンネル・パスのみを論理的に使用可能にしておく必要があります。このようにしておかないと、チャンネル・サブシステムが複数のチャンネル・パス上で入出力装置使用中条件を検出し、その結果、後続のマルチパス・モード操作について予測不能の結果が生じることがあります。このタイプのセットアップ手順は、マルチパス・グループのメンバーを変更する場合にも使用する必要があります。

2. マルチパス・モードの終了

マルチパス・モードを終了し、シングル・パス・モードで処理を続けるには、次の 2 つの手順のいずれかを使用できます。

- a. マルチパス・グループ内の任意のチャンネル・パスに対して、マルチパス・モード解除タイプの

コマンドを実行することができます。このコマンドの後で、(1) SCHIB のワード 1 のビット 13 に 0 を指定して MODIFY SUBCHANNEL を実行するか、または、(2) LPM の中で、1 つのチャンネル・パスのみが論理的に使用可能であることを指定する必要があります。装置がシングル・パス・モードで稼働している間は、複数の選択可能なチャンネル・パスを持ちマルチパス・モードで稼働しているサブチャンネルで開始機能を実行してはなりません。これを行うと、その機能または後続の開始機能について、サブチャンネルで予測不能の結果が生じることがあります。

- b. マルチパス・グループ内の各チャンネル・パス上で、マルチパス・モード停止タイプのコマンドを実行します (項目 1 (15-19ページ) で述べたセットアップの逆)。このコマンドの後で、(1) SCHIB のワード 1 のビット 13 に 0 を指定して MODIFY SUBCHANNEL を実行するか、または、(2) LPM の中で、1 つのチャンネル・パスのみが論理的に使用可能であることを指定する必要があります。装置がシングル・パス・モードで稼働している間は、複数の選択可能なチャンネル・パスを持ちマルチパス・モードで稼働しているサブチャンネルで、開始機能を実行することはできません。これを行うと、その開始機能または後続の開始機能について、サブチャンネルで予測不能の結果が生じることがあります。

データのブロック化

入出力装置が記録するデータは、ブロックに分割されます。1 ブロックの長さは装置によって異なります。例えば、1 ブロックは、1 枚のカードや 1 印刷行のこともあり、また磁気テープ上の 2 つの連続したギャップの間に記録される情報のこともあります。

1 回の入出力操作で転送できる情報の最大量は、1 ブロックです。関連の主記憶域が使い尽くされるか、またはブロックの終わりに達するか、いずれかの条件が先に発生した時点で、入出力操作は終了します。一部の操作、例えば磁気テープ装置や照会端末での書き込みなどでは、ブロックは定義されず、転送する情報の量はプログラムによってのみ制御されます。

操作要求ブロック

操作要求ブロック (ORB) は、START SUBCHANNEL のオペランドです。ORB は、特定の開始機能を制御するために使用するパラメーターを指定します。これらのパラメーターには、割り込みパラメーター、サブチャンネル・キー、最初の CCW のアドレス、操作制御ビット、優先順位を指定する番号、および、チャンネル・パスの論理的可用性の指定などがあります。

START SUBCHANNEL の実行時には、ORB の内容が指定のサブチャンネルに入れられてから、条件コード 0 が設定されます。実行の結果が 0 以外の条件コードになる場合は、ORB の内容は指定のサブチャンネルに入れられません。

ORB がワード境界に合わせて置かれるようにするために、ORB アドレスの右端 2 ビットは 0 でなければなりません。さもないと、指定例外が認識されます。ORB のフォーマットは以下のとおりです。

ワード

0	Interruption Parameter																					
1	Key	S	C	M	Y	F	P	I	A	U	0	H	T	LPM	L	0	0	0	0	0	0	X
2	Channel-Program Address																					
3	CSS Priority				Reserved				CU Priority				Reserved									
4	Reserved																					
5	Reserved																					
6	Reserved																					
7	Reserved																					
	0	8				16				24				31								

ORB のフィールドは以下のように定義されています。

割り込みパラメーター (Interruption Parameter):

ワード 0 のビット 0-31 は、後続の START SUBCHANNEL または MODIFY SUBCHANNEL 命令により置き換えられるまでは、変更されないままサブチャンネル内に保持されます。入出力割り込みが起きるか、TEST PENDING INTERRUPTION の実行により割り込み要求が消去されると、割り込みコードのワード 1 にこれらのビットが入れます。

サブチャンネル・キー (Subchannel Key): ワード 1 のビット 0-3 は、CCW、IDAW、および出力データのすべての取り出し、および START SUBCHANNEL により開始される開始機能に関連した入力データの格納に使用するサブチャンネル・キーを形成します。これらの記憶域参照を行うときに、このキーが記憶キーと突き合わされます。詳しくは、3-8ページの『キー制御保護』を参照してください。

延期制御 (S): ワード 1 のビット 4 は、ORB 内で指定されているチャンネル・プログラムに対する延期機能の実行を制御します。S ビットの設定は、ORB 内に指定されているチャンネル・プログラムのすべての CCW に適用されます (15-38ページの『コマンドとフラグ』を参照)。ビット 4 が 1 のときは、延期制御が指定され、CCW 内で 1 に設定された延期フラグが検出されるとチャンネル・プログラムの延期が起きます。ビット 4 が 0 のときは、延期制御は指定されず、チャンネル・プログラムのいずれかの CCW の中で延期フラグのどれかが 1 に設定されていると、プログラム・チェック条件が認識されます。

ストリーム・モード制御 (C): ワード 1 のビット 5 は、指定した開始機能の実行時に、FICON 変換入出力インターフェース・チャンネル・パスに対して構成されているサブチャンネルについて、ストリーム・モードを使用可能にするかどうかを制御します。ビット 5 が 0 のときは、サブチャンネルでストリーム・モードが使用可能にされます。ビット 5 が 1 のときは、サブチャンネルでストリーム・モードが使用不可にされます。ビット 5 が意味を持つのは、FICON 変換入出力インターフェース・チャンネル・パスに対して構成されているサブチャンネルの場合のみであり、他のタイプのチャンネル・パスに対して構成されているサブチャンネルの場合は無視されます。

ストリーム・モードが使用可能になっているときは、チャンネル・パスは、指定されたチャンネル・プログラムの最初のコマンドが関連装置で受理されたことを示す指示を受け取った時点で、その装置でそのコマンドが進行中であるものと見なします。さらに、一定の条件下では、装置から状況を受理したことを示す肯定応答を受け取っていないくても、チャンネル・パスは状況を受理したことを認識します。

ストリーム・モードが使用可能になっていないときは、チャンネル・パスで装置コマンド応答または状況が受理されたことを示す適切なチャンネル・パス応答が装置に送られるまでは、チャンネル・パスは、指定されたチャンネル・プログラムの最初のコマンドが関連の装置で進行中であるとは見なしません。さらに、装置がチャンネル・パスに状況を送信した場合、装置がチャンネル・パスによる受理確認を受信し、それに対する肯定応答を返すまでは、チャンネル・パスではその状況の受理は認識されません。

変更制御 (M): ワード 1 のビット 6 は、チャンネル・プログラムが変更制御を必要とするかどうかを指定します。ビット 6 が 0 のときは、変更制御が指定されません。ビット 6 が 1 のときは、変更制御は指定されません。

変更制御が指定されているときは、コマンドが 1 つ実行されるたびに、その前に実行されたコマンドで PCI フラグとコマンド・チェーニング・フラグが 1 に設定されていて、データ・チェーニング・フラグと延期フラグが 0 に設定されていれば、チャンネル・サブシステムは、強制的にそのコマンドと指定された入出力装置を同期させます。この条件が認識されたときは、チャンネル・サブシステムは、現コマンドについて同期要求を入出力装置に送ります。この場合、チャンネル・サブシステムは一時的にコマンド・チェーニングを延期し、同期しようとしているコマンドについて正常終了状況が受信されるまでは、次のコマンド・チェーン CCW の取り出し (または再取り出し) を行いません。

変更制御が指定されていないときは、コマンドの同期は不要であり、チャンネル・サブシステムは、状況の受信を待たずに入出力装置にコマンドを転送することができます。

M ビットが意味を持つのは、FICON 入出力インターフェース・チャンネル・パスまたは FICON 変換入出力インターフェース・チャンネル・パスに対して構成されたサブチャンネルの場合のみであり、その他のタイプのチャンネル・パスに対して構成されているサブチャンネルの場合は無視されます。

プログラミング上の注意: FICON 変換入出力インターフェース・チャンネル・パスの場合は、変更制御を利用することにより、CCW 内の PCI フラグを使用してチャンネル・プログラム変更を開始して、動的に変更されるチャンネル・プログラムを最適化することができます。特に、プログラムは、PCI ビットが設定されているコマンド次のコマンドについての状況が受信されるまで、チャンネル・サブシステムが後続のコマンドの取り出しと転送を行うのを遅延させることができます。これにより、CPU プログラム制御の割り込みを受理する可能性が高くなり、プログラムは、同期するコマンドの後に続く 1 つまたは複数のコマンド・チェーン CCW を動的に変更することができます。

動的に変更される CCW が、変更前ではなく変更後に取り出される可能性を高めるためには、変更を行うプログラムは、CPU がプログラム制御の割り込みを受理した後で、できるだけ早く実行することが重要です。さらに、プログラムは、構成済みの CPU が入出力割り込みに対して使用不可にされている時間を最小限に抑えるようにすることも重要です。

FICON インターフェース・チャンネル・パスまたは FICON 変換入出力インターフェース・チャンネル・パス以外のチャンネル・パスの場合は、コマンド同期は、チャンネル・サブシステムと入出力装置の間のシグナル・プロ

トコルの中で暗黙的に行われます。したがって、コマンド同期の強制実施するための明示的なプログラミング・アクションは必要ありません。ただし、これらのタイプのチャンネル・パスの場合も、上記と同じ理由により、プログラムは、プログラム制御の割り込みをできるだけ早く受理して処理することが重要です。

FICON 入出力インターフェース・チャンネル・パスまたは FICON 変換入出力インターフェース・チャンネル・パスの場合に、チャンネル・サブシステムがチャンネル・プログラムの実行を最適化できるようにするには、チャンネル・プログラムが動的可変を必要としない限り、変更制御機能はできるだけ使用しないようにしてください。

同期制御 (Y): ワード 1 のビット 7 は、チャンネル・プログラムが同期制御を必要とするかどうかを指定します。ビット 7 が 0 で、事前取り出し制御ビット (ワード 1 のビット 9) が 1 のときは、同期制御が指定されません。ビット 7 が 1 でビット 9 が 1 のときは、同期制御は指定されません。

同期制御が指定されている場合は、現在実行中のコマンドが入力操作を指定していて、次に取り出す CCW が出力操作を指定しているとき、チャンネル・サブシステムは、コマンドと指定された入出力装置の強制同期を行います。この条件が認識されると、チャンネル・サブシステムは、入力コマンドを転送するときに同期要求を入出力装置に送ります。正常終了状況 (入出力装置での入力操作の実行が完了したことを通知する) が受信されるまでは、サブチャンネルでは出力コマンドの転送が保留のままにされます。終了状況を受信すると、チャンネル・サブシステムは、出力コマンドに関連したデータの取り出し (または再取り出し) を行い、そのデータを入出力装置に転送します。

同期制御が指定されていないときは、チャンネル・サブシステムは、各コマンドごとに入出力操作の完了を知らせる状況を待たずに、チャンネル・プログラムのコマンドを転送することができます。

Y ビットが意味を持つのは、サブチャンネルが、FICON 入出力インターフェース・チャンネル・パスまたは FICON 変換入出力インターフェース・チャンネル・パスに対して構成されており、かつ事前取り出し制御ビット (ワード 1 のビット 9) が 1 のときだけです。サブチャンネルがその他のタイプのチャンネル・パスに対して構成されているとき、または事前取り出し制御ビットが 0 のときは、Y ビットは無視されます。

フォーマット制御 (F): ワード 1 のビット 8 は、チャンネル・プログラム・アドレス・フィールドに指定されているチャンネル・プログラムを形成するチャンネル・コマンド・ワード (CCW) のフォーマットを指定します。ワード 1 のビット 8 が 0 のときは、フォーマット 0 の CCW が指定されます。ビット 8 が 1 のときは、フォーマット 1 の CCW が指定されます。(CCW フォーマットの定義については、15-25 ページの『チャンネル・コマンド・ワード』を参照してください。)

事前取り出し制御 (P): ワード 1 のビット 9 は、チャンネル・プログラムで CCW の事前取り出しが無制限に許されるかどうかを指定します。ビット 9 が 1 のときは、CCW の無制限事前取り出しが許されます。(現 CCW および事前取り出し済み CCW に関連したデータと IDAW の無制限事前取り出しは、常に許されます。) 実際に事前取り出しが行われるかどうかは、モデルによって異なります。

ワード 0 のビット 9 が 0 のときは、事前取り出しは許されません。ただし、出力でのデータ・チェーニングの場合は、データ域を記述する 1 つの CCW の事前取り出しが許されます。ワード 0 のビット 9 が 0 のときは、同期制御ビット (ワード 1 のビット 7) は無視されます。

事前取り出しの有効範囲は、その他の制御機能により制限されることもあります。

初期状況割り込み制御 (I): ワード 1 のビット 10 は、チャンネル・サブシステムが、開始または再開機能に関連した最初のコマンドを装置が受理したかどうかを確認し、プログラムに通知する必要があるかどうかを指定します。ORB の I ビットに 1 が指定されているときは、サブチャンネルがアクティブになると (これはこの開始または再開機能のための最初のコマンドが受理されたことを示します)、このサブチャンネルで Z ビット (16-11 ページの『0 条件コード (Z)』を参照) が 1 に設定され、サブチャンネルは中間状況を伴う状況保留になります。

サブチャンネルがアクティブにならない場合は (例えば、装置が最初のコマンドを受け取った直後にチャンネル終了を通知し、CCW 内でコマンド・チェーニングが指定されておらず、コマンド再試行が通知されていない場合など)、コマンド受理条件 (Z ビットが 1 に設定される) は生成されず、代わりに、サブチャンネルは 1 次状況を伴う状況保留になります。この場合、最初の CCW の中で PCI フラグが 1 に設定されていれば、コマンドが受理されたときに中間状況も示されることがあります。

アドレス限界検査制御 (A): ワード 1 のビット 11 は、チャンネル・プログラムでアドレス限界検査を行うかどうかを指定します。このビットが 0 のときは、サブチャンネルでの限界モード・ビット (15-2ページの『限界モード (LM)』を参照) の設定には関係なく、チャンネル・プログラムの実行時にアドレス限界検査は行われません。このビットが 1 のときは、サブチャンネルでの限界モード・ビットの設定に応じて、チャンネル・プログラムに関するアドレス限界検査が許可されます。

延期割り込み抑止制御 (U): ワード 1 のビット 12 が 1 のときは、サブチャンネルが延期された場合に、チャンネル・サブシステムは、延期を原因とする中間割り込み条件の生成を抑止します。ビット 12 が 0 のときは、チャンネル・サブシステムは、チャンネル・プログラムの実行中にサブチャンネルが延期されるたびに、中間割り込み条件を生成します。

フォーマット 2 IDAW 制御 (H): ワード 1 のビット 14 は、間接データ・アドレッシングを指定する CCW 用の IDAW のフォーマットを指定します。ワード 1 のビット 14 が 1 のときは、IDAW フラグが 1 に設定されているすべての CCW について、フォーマット 2 (64 ビット・データ・アドレス) の IDAW が用意されます。ワード 1 のビット 14 が 0 のときは、IDAW フラグが 1 に設定されているすべての CCW について、フォーマット 1 (31 ビット・データ・アドレス) の IDAW が用意されます。

プログラミング上の注意: 64 ビット IDAW は、入出力装置と、2G バイトより大きいアドレスを持つ記憶位置との間で、データを直接転送するための手段として使用されるだけです。

2K IDAW 制御 (T): ワード 1 のビット 15 は、フォーマット 2 IDAW データ域の主記憶域ブロック・サイズを指定します。ビット 15 が意味を持つのは、ビット 14 (フォーマット 2 IDAW 制御) が 1 のときのみであり、ビット 14 が 0 のときは無視されます。ワード 1 のビット 15 が 1 のときは、すべてのフォーマット 2 IDAW が 2K バイトの記憶域ブロックを指定します。ワード 1 のビット 15 が 0 のときは、すべてのフォーマット 2 IDAW が 4K バイトの記憶域ブロックを指定します。

ワード 1 のビット 13 は将来の使用に備えて予約済みであり、このビットは 0 でなければなりません。さもないと、オペランド例外またはプログラム・チェック条件が認識されます。

論理パス・マスク (LPM): ワード 1 のビット 16-23 は、サブチャンネル内に無変更のまま残され、論理パス 0-7 のどれがプログラムから見て論理的に使用可能と認められるかを、チャンネル・サブシステムに指示します。これらのビットの 1 つが 1 に設定されている場合、それに対応するチャンネル・パスが論理的に使用可能であることを示します。0 の場合は、対応するチャンネル・パスが論理的に使用可能でないことを示します。プログラムが、あるチャンネル・パスを論理的に使用可能でないものとして指定している場合は、そのチャンネル・パス用の専用従属関係が存在していない限り、チャンネル・サブシステムは、そのチャンネル・パスを使用して、プログラムから要求された消去、停止、再開、または開始機能の実行を開始することはありません。専用従属関係条件が存在する場合は、LPM の設定は無視され、専用従属関係を持つチャンネル・パスを使用して、再開、開始、停止、または消去機能が実行されます。

誤長抑止モード (L): ワード 1 のビット 8 が 1 のときに、ワード 1 のビット 24 が 1 であれば、誤長抑止モードであることを示します。即時操作が行われ (つまり、装置がコマンドの開始時にチャンネル終了条件を通知し)、そして現 CCW のビット位置 16-31 に 0 以外の値が含まれているときに、サブチャンネルがこのモードにある場合は、誤長条件の表示が抑止されます。

ワード 1 のビット 8 が 1 のときに、ワード 1 のビット 24 が 0 であれば、誤長表示モードであることを示します。即時操作が行われ (つまり、装置がコマンドの開始時にチャンネル終了条件を通知し)、そして現 CCW のビット位置 16-31 に 0 以外の値が含まれているときに、サブチャンネルがこのモードにある場合は、誤長条件の表示が認識されます。この場合、CCW 内の SLI フラグが 1 で、かつデータ・チェーン・フラグが 0 でない限り、コマンド・チェーニングは抑止されます。

ワード 1 のビット 8 が 0 のときは、チャンネル・サブシステムはビット 24 の値を無視し、サブチャンネルは誤長抑止モードになっています。

ORB 拡張制御 (X): ワード 1 のビット 31 は、ORB を拡張するかどうかを指定します。ワード 1 のビット 31 が 0 のときは、ORB はワード 0-2 のみで形成され、ワード 3-7 は無視されます。ワード 1 のビット 31 が 1 のときは、ORB はワード 0-7 により形成されます。ワード 0 および 1 については、上記で説明しました。ワード 2-7 については、これから説明します。

予約済み (Reserved): ワード 1 のビット 25-30 は将来の使用に備えて予約済みであり、すべて 0 に設定されていなければなりません。ORB 拡張機能がインストー

ルされていない場合は、ワード 1 のビット 31 は 0 でなければなりません。さもないと、オペランド例外またはプログラム・チェック条件が認識されます。

チャンネル・プログラム・アドレス (Channel-Program Address):

ワード 2 のビット 1-31 は、主記憶機構内での最初の CCW の絶対アドレスを指定します。ワード 2 のビット 0 は 0 でなければなりません。さもないと、オペランド例外またはプログラム・チェック条件が認識されます。ワード 1 のビット 8 でフォーマット 0 CCW が指定されている場合は、ワード 2 のビット 1-7 も 0 でなければなりません。さもないと、プログラム・チェック条件が認識されます。

チャンネル・プログラム・アドレスの右端 3 ビットは、すべて 0 (CCW をダブルワード境界に合わせるため) でなければなりません。さもないと、プログラム・チェック条件が認識されます。

チャンネル・プログラム・アドレスが、取り出しに対して保護されている記憶位置を指している場合、または特定のインストール・システムの記憶域の外部にある記憶位置を指している場合は、装置で開始機能は開始されません。その場合は、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態になります。

チャンネル・サブシステム優先順位 (CSS

Priority): ORB のワード 1 のビット 31 (X) が 1 のときは、ワード 3 のバイト 0 には、チャンネル・サブシステム優先順位番号と呼ばれる符号なしの 2 進整数が入ります。この優先順位番号は指定されたサブチャンネルに割り当てられているもので、開始保留または再開保留状態にある 1 つ以上のサブチャンネルを対象として開始機能または再開機能を開始するときに、サブチャンネルを選択する順序を決定するために使用されます。

指定されるチャンネル・サブシステム優先順位番号は、0 から 255 までの任意の値です。0 は最低、255 は最高の優先順位を示します。

モデルと構成に応じて、次のような違いがあります。

1. 提供される優先順位レベルが 256 より少ない場合があります。このようなモデルでは、ORB に指定された優先順位番号が無視されて、サブチャンネルが開始保留状態になったときに、暗黙的に代替優先順位番号がサブチャンネルに割り当てられることがあります。
2. ORB のワード 1 のビット 31 (X) が 0 のときは、暗黙の優先順位番号がサブチャンネルに割り当てられます。

上記 2 つの場合に、優先順位番号がどのように割り当てられるかについての詳細は、17-21 ページの『チャンネル・サブシステム入出力優先順位決定機能』を参照してください。

制御装置優先順位 (CU Priority) ORB のワード 1 のビット 31 (X) が 1 のときは、ワード 3 のバイト 2 には、制御装置優先順位番号と呼ばれる符号なしの 2 進整数が入ります。この番号は、FICON チャンネル・パスにより接続されている関連制御装置について、開始機能に関連したすべての入出力操作を行うときにその制御装置に適用される優先順位レベルを指定します。

指定される制御装置優先順位番号は、1 から 255 までの任意の整数です。1 は最低、255 は最高の優先順位を示します。番号 0 は、開始機能に関連した入出力操作に優先順位が割り当てられていないことを示します。優先順位番号が 0 のときに入出力操作がどのように取り扱われるかは、制御装置のモデルによって異なります。

さらに、制御装置のモデルによっては、制御装置がサポートする優先順位レベルの数が 255 より少ないこともあります。サポートされる優先順位番号の範囲と、優先順位番号がどのように使用されるかについての詳細は、該当の制御装置のシステム・ライブラリー資料を参照してください。

次のいずれかの条件に該当する場合は、指定された制御装置優先順位番号は無視されます。

1. ワード 1 のビット 31 (X) が 0 である。この場合は、制御装置優先順位番号 0 が、関連のアウトバウンド・フレームに入れて伝送されます。
2. 指定されたサブチャンネルが、FICON チャンネル・パスに対して構成されている制御装置に関連していない。
3. 関連の制御装置が、優先順位付けどおりの入出力操作を提供していない。この場合は、制御装置では、関連のアウトバウンド・フレームに含まれている制御装置優先順位番号は無視されます。
4. チャンネル・サブシステム・モデルが、制御装置優先順位番号を伝送する能力を備えていない。
5. オペレーターのアクションが原因で、チャンネル・サブシステム入出力優先順位機能が操作可能な状態にない。

予約済み (Reserved): ORB の中で「0」または「Reserved」として定義されているすべてのフィールドは、START SUBCHANNEL の実行時点で 0 になっていなければなりません。さもないと、オペランド例外またはプログラム・チェック条件が認識されます。

プログラミング上の注意:

- ORB の中で現在 0 を含むものとして指定されているビット位置は、将来、新しい機能を制御するために割り当てられる可能性があります。
- 割り込みパラメーターにはどのような情報でも含めることができますが、通常は、入出力割り込みを取り扱うプログラムにとって重要な意味を持つ情報です。

チャンネル・コマンド・ワード

チャンネル・コマンド・ワード (CCW) は、実行するコマンドを指定します。また、ある種の入出力操作を開始するコマンドについては、操作に関連した記憶域、その記憶域との間の転送が完了したときにとるアクション、およびその他のオプションを指定します。

チャンネル・プログラムは、1 つまたは複数の CCW から成っており、それらの CCW は、チャンネル・サブシステムにより順次または非順次の順序で取り出されて実行されるように、論理的にリンクされています。順次 (連続) CCW は、データ・チェーン・フラグおよびコマンド・チェーン・フラグを使用してリンクされ、非順序 (不連続) CCW は、チャンネル内転送コマンドを指定する CCW によりリンクされます。

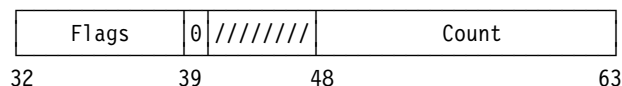
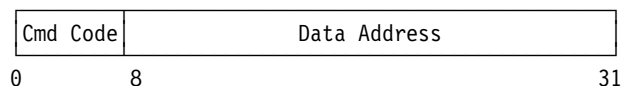
各 CCW が実行されているとき、それは現 CCW として認識されます。CCW が現 CCW となるのは、(1) チャンネル・プログラムの最初の CCW として取り出されたとき、(2) コマンド・チェーニングの実行中に新たな CCW が論理的に取り出されたとき、または、(3) データ・チェーニングの実行中に新たな CCW が入出力操作の制御を引き継いだときです (15-31 ページの『データ・チェーニング』を参照)。チェーニングが指定されていないときは、TEST SUBCHANNEL がサブチャンネル内の開始機能ビットを消去してしまうと、CCW は現 CCW ではなくなります。

チャンネル・プログラムの最初の CCW の位置は、START SUBCHANNEL のオペランドである ORB の中で指定されます。最初の CCW は、この命令の実行後に取り出されます。チャンネル・サブシステムが取り出す CCW のフォーマットは、ORB のワード 1 のビット 8 に指定されます。チャンネル・プログラム内のその他の CCW は、それぞれ必要になったときに取得されます。チャンネル・サブシステムが CCW を取り出すことで、主記憶機構内での CCW の位置が変化することはありません。

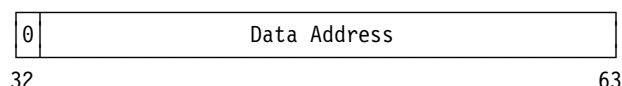
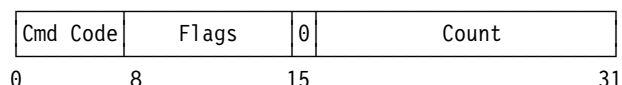
CCW には、フォーマット 0 とフォーマット 1 の 2 つのフォーマットがあります。この 2 つのフォーマットの間では、CCW に含まれる情報が違うのではなく、CCW 内のアドレスのサイズとフィールドの配列が異なるだけです。

これらのフォーマットは以下のように定義されています。

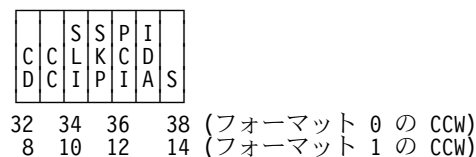
フォーマット 0 の CCW



フォーマット 1 の CCW



フラグ



フォーマット 0 の CCW は、絶対記憶域の最初の 2^{24} (16M) バイトのどの位置にでも置くことができ、フォーマット 1 の CCW は、絶対記憶域の最初の 2^{31} (2G) バイトのどの位置にでも置くことができます。

チャンネル内転送を指定するフォーマット 0 の CCW を除くすべての CCW では、ビット 39 (フォーマット 0 の場合) またはビット 15 (フォーマット 1 の場合) は 0 でなければなりません。間接データ・アドレッシングが指定されていて、CCW に関連した ORB のフォーマット 2 IDAW 制御ビットが 0 である場合は、次の条件が満たされていることが必要です。

- CCW のビット 30 および 31 (フォーマット 0 の場合) またはビット 62 および 63 (フォーマット 1 の場合) は、ワード境界を指定するために 0 でなければなりません。
- 間接データ・アドレス・リストの最初のエントリーのビット 0 は 0 でなければなりません。

間接データ・アドレッシングが指定されていて、CCWに関連した ORB のフォーマット 2 IDAW 制御ビットが 1 である場合は、CCW のビット 29-31 (フォーマット 0 の場合) またはビット 61-63 (フォーマット 1 の場合) は、ダブルワード境界を指定する 0 でなければなりません。これらの条件のいずれかが満たされていないと、プログラム・チェック条件が認識されることがあります (15-34ページの『CCW 間接データ・アドレッシング』を参照)。データ・チェーニングの実行中にこの条件が検出されると、操作を終結するよう指示する信号が入出力装置に送られます。コマンド・チェーニングの実行中または START SUBCHANNEL の実行後に、上記で述べた必須の 0 が存在しないことが検出された場合は、新しい操作は開始されず、割り込み条件が生成されます。

フォーマット 0 の CCW のビット位置 40-47 の内容は無視されます。

CCW のフィールドは以下のように定義されています。

コマンド・コード (Cmd Code): ビット 0-7 (両フォーマット共通) は、実行する操作を指定します。

データ・アドレス (Data Address): ビット 8-31 (フォーマット 0) またはビット 33-63 (フォーマット 1) は、絶対記憶域内の位置を指定します。ここに指定されるのは、CCW で指定されている区域内で最初に参照される位置です。フォーマット 1 の CCW のビット 32 は、0 でなければなりません。さもないと、プログラム・チェック条件が認識されます。バイト・カウント 0 が指定されている場合は、このフィールドは検査されません。

2G バイトを超えるデータ・アドレスの指定については、15-34ページの『CCW 間接データ・アドレッシング』を参照してください。

データ・チェーン (CD) フラグ: ビット 32 (フォーマット 0) またはビット 8 (フォーマット 1) が 1 のときは、データをチェーニングすることを指定します。その場合、現在の入出力操作で、次の CCW に指定されている記憶域が使用されます。CCW 内の CD フラグが 1 のときは、コマンド・チェーン・フラグおよび誤長表示抑止フラグ (下記を参照) は無視されます。

コマンド・チェーン (CC) フラグ: ビット 33 (フォーマット 0) またはビット 9 (フォーマット 1) が 1 で、CD フラグと S フラグが両方とも 0 のときは、コマン

ドをチェーニングすることを指定します。その場合、現在の操作が正常に完了した時点で、次の CCW の中のコマンド・コードで指定されている操作が開始されます。

誤長表示抑止 (SLI) フラグ: ビット 34 (フォーマット 0) またはビット 10 (フォーマット 1) は、誤長条件をプログラムに提示するかどうかを制御します。このビットが 1 で、CD フラグが 0 のときは、誤長表示は抑止されます。CC フラグと SLI フラグが両方とも 1 で、CD フラグが 0 のときは、誤長条件が存在するかどうかに関係なく、コマンド・チェーニングが行われます。誤長表示を抑止したい CCW では、必ずこのビットを指定するようにしてください。

スキップ (SKP) フラグ: ビット 35 (フォーマット 0) またはビット 11 (フォーマット 1) が 1 のときは、読み取り、逆方向読み取り、センス ID、またはセンス操作で、記憶機構への情報の転送を抑止することを指定します。

プログラム制御割り込み (PCI) フラグ: ビット 36 (フォーマット 0) またはビット 12 (フォーマット 1) が 1 のときは、チャンネル・サブシステムは、このビットを含む CCW が入出力操作の制御権を獲得したときに、中間割り込み条件を生成します。PCI フラグ・ビットが 0 のときは、通常の操作が行われます。

間接データ・アドレス (IDA) フラグ: ビット 37 (フォーマット 0) またはビット 13 (フォーマット 1) が 1 のときは、間接データ・アドレッシングを指定します。

延期 (S) フラグ: ビット 38 (フォーマット 0) またはビット 14 (フォーマット 1) が 1 のときは、チャンネル・プログラムの実行を延期することを指定します。この値が有効な場合、S フラグを含む CCW の実行の前に、チャンネル・プログラムの実行が延期されます。S フラグの 1 の値が有効なのは、関連の ORB のワード 1 のビット 4 が 1 のときです。

カウント (Count): ビット 48-63 (フォーマット 0) またはビット 16-31 (フォーマット 1) は、CCW により指定されている記憶域のバイト数を指定します。

プログラミング上の注意: フォーマット 0 の CCW のビット 39 およびフォーマット 1 の CCW のビット 15 は、現在は 0 でなければなりません。将来は、新しい機能を制御するために割り当てられる可能性があります。したがって、故意にプログラム・チェック表示を起こさせる目的で、このビットを 1 に設定することは避けてください。

コマンド・コード

コマンド・コード (CCW のビット位置 0-7) は、実行する操作をチャンネル・サブシステムおよび入出力装置に対して指定します。

コマンド・コードの右端の 2 ビット (これらのビットが 0 のときは右端の 4 ビット) は、実行する操作をチャンネル・サブシステムに対して指示します。チャンネル・サブシステムは、次の 4 種類の操作を識別します。

- 正方向出力 (書き込み、制御)
- 正方向入力 (読み取り、センス、センス ID)
- 逆方向入力 (逆方向読み取り)
- ブランチ (チャンネル内転送)

チャンネル・サブシステムは、チャンネル内転送を指定するフォーマット 1 の CCW の場合を除き、コマンド・コードの左寄りのビットを無視します。チャンネル内転送を指定するフォーマット 1 の CCW の場合は、チャンネル・サブシステムはコマンド・コードのすべてのビットをデコードします。

入出力操作 (書き込み、読み取り、逆方向読み取り、制御、センス、センス ID) を開始するコマンドでは、コマンド・コードの 8 ビットすべてが制御装置に転送されます。これらのコマンド・コードの左寄りのビット位置には、修飾ビットが含まれています。修飾ビットは、装置に対して、コマンドをどのように実行するかを指定します。例えば、修飾ビットにより、書き込み操作の際に受け取ったデータを以前に記録されているデータと比較することを装置に指示したり、記録密度やパリティなどの属性を指定したりできます。制御コマンドの場合は、修飾ビットには、実行する制御機能を指定する指令コードを含めることができます。修飾ビットの意味は、入出力装置のタイプによって異なり、該当装置のシステム・ライブラリ資料の中で定義されています。

コマンド・コードの割り当ては、図15-5に示すとおりです。x 記号は、そのビット位置が無視されることを示します。m は修飾ビットを表します。

コード	コマンド
x x x x 0 0 0 0	無効
m m m m m m 0 1	書き込み
m m m m m m 1 0	読み取り
m m m m 1 1 0 0	逆方向読み取り
m m m m m m 1 1	制御
m m m m 0 1 0 0	センス
1 1 1 0 0 1 0 0	センス ID
x x x x 1 0 0 0	チャンネル内転送 ¹
0 0 0 0 1 0 0 0	チャンネル内転送 ²
m m m m 1 0 0 0	無効 ³

説明:

- m 修飾ビット
- x 無視
- ¹ フォーマット 0 の CCW
- ² フォーマット 1 の CCW
- ³ ビット 0-3 のいずれかが 0 以外であるフォーマット 1 の CCW

図 15-5. コマンド・コードの割り当て

チャンネル・サブシステムが、コマンド実行の開始時に無効なコマンド・コードを検出すると、プログラム・チェック割り込み条件が生成され、チャンネル・プログラムの実行は中止されます。データ・チェーニングの際には、チャンネル内転送を指定している場合以外は、コマンド・コードは無視されます。

記憶域の指定

入出力操作に関連した主記憶域は、1 つ以上の CCW により定義されます。CCW は、転送する最初のバイトのアドレスと、対象区域に含まれる連続したバイト数を指定することにより、その区域を定義します。転送する最初のバイトのアドレスは、CCW のデータ・アドレス・フィールド内で直接指定されるか、または、CCW のデータ・アドレス・フィールドが示す間接データ・アドレス・ワード (IDAW) で間接的に指定されます。記憶域に含まれるバイト数は、カウント・フィールドに指定されます。

書き込み、読み取り、制御、およびセンス操作では、記憶位置はアドレスの昇順に従って使用されます。主記憶機構との間での情報の転送が進むにつれて、アドレス・フィールド内のアドレスは増加し、カウント・フィールド内のカウントは減少します。逆方向読み取り操作では、データがアドレスの降順に従って記憶域に格納され、カウントとアドレスは両方とも減少します。カウントが 0 に達すると、CCW で定義された記憶域は使い尽くされたこととなります。

開始機能で使用できる主記憶位置は、該当タイプの参照に対して保護されていない限り、すべて、入出力装置へ

のデータ転送または入出力装置からのデータ転送に使用できます。フォーマット 0 CCW は、絶対記憶域の最初の 2²⁴ (16M) バイトのうちの使用可能部分の中であれば、どこにでも置くことができます。フォーマット 1 の CCW は、絶対記憶域の最初の 2³¹ (2G) バイトのうちの使用可能部分の中であれば、どこにでも置くことができます。ただし、いずれの場合も、該当の位置が取り出しタイプの参照に対して保護されていないことが条件です。チャンネル・サブシステムが保護されている位置を参照しようとする、記憶保護チェック条件が生成され、操作を終了するよう指示する信号が装置に送られます。

ある主記憶位置が使用可能と見なされるのは、構成内でその位置が使用可能であり、さらに、データ記憶位置 (CCW または IDAW ではない) の場合は、アドレス限界検査機能によりその位置へのアクセスが阻止されないときです。ある主記憶位置が使用可能でない状態を、無効アドレスを持っている状態と呼びます。

チャンネル・サブシステムが使用可能でない記憶位置を参照すると、プログラム・チェック条件が生成されます。チャンネル・プログラム・アドレスに指定されている最初の CCW が使用不可の位置にあるときは、装置では開始機能は開始されず、SCSW の状況部分がプログラム・チェック標識で更新され、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態になり、据え置き条件コード 1 が示されます。無効なデータ・アドレスがある場合、または、チェーニングの時点または START SUBCHANNEL の実行後に無効な CCW アドレスが検出された場合は、チャンネル・サブシステムは、装置が次に 1 バイトのデータまたは状況を要求または供給するときに、操作を終結するよう指示する信号を装置に送ります。この場合は、サブチャンネルは状況保留にされ、サブチャンネル状況にはプログラム・チェックが示され、装置状況は装置から受信された状況によって決まります。プログラム・チェック条件が起きると、コマンド・チェーニングおよびコマンド再試行は抑止されます。

出力操作では、入出力装置がデータを要求する前に、チャンネル・サブシステムがそのデータを主記憶機構から取り出すことがあります。現 CCW で指定されているバイト数のうち、任意の数のバイトを事前に取り出してバッファに入れることができます。出力操作でデータ・チェーニングを行っているときは、チャンネル・サブシステムは、現 CCW の実行中の任意の時点で、1 つのデータ域を記述する別の CCW を取り出すことがあります。

ORB の事前取り出し制御ビットの設定により無制限の事前取り出しが許されている場合は、チャンネル・サブシステムは、任意の数の CCW と IDAW および関連のデータを事前取り出しすることができます。入出力操作で、使用可能な記憶域の終わりに近い位置にあるデータ

および CCW が使用される場合、そのための事前取り出しの際に、チャンネル・サブシステムが存在しない記憶位置を参照することがあります。事前取り出しで無効アドレスが検出されても、入出力操作の実行に影響はなく、また、その操作で実際に該当位置の情報を使用しようとするまでは、エラーも示されません。無効な情報が実際に必要になる前に、入出力装置により、または HALT SUBCHANNEL または CLEAR SUBCHANNEL の実行により、操作が終結された場合は、プログラムにはその条件は通知されません。

CCW のカウント・フィールドには、最大 65,535 までの任意のバイト数を指定できます。フォーマット 0 の CCW では、カウント・フィールドは、コマンド・コードがチャンネル内転送を指定していない限り常に 0 以外の値であり、チャンネル内転送が指定されている場合は、カウント・フィールドは無視されます。フォーマット 1 の CCW では、データ・チェーニングが指定されているか、またはデータ・チェーニング中にその CCW が取り出される場合を除き、カウント・フィールドに 0 の値を含めることができます。(1) フォーマット 1 の CCW 内のカウント・フィールドが 0 であり、(2) データ・チェーニングが指定されていないかまたは有効でなく、かつ、(3) 装置がデータを要求した場合は、停止を指示する信号が装置に送られ、入出力操作は中止されます。CCW 内の SLI フラグが 1 でない場合は、チャンネル・サブシステムは誤長条件を設定します。その場合、データは転送されません。装置がデータ転送を要求していない場合は、操作は正常な終了点まで進められます。

チャンネル内転送を指定していないフォーマット 0 の CCW でバイト・カウントが 0 の場合、あるいは、データ・チェーニングを指定しているかまたはデータ・チェーニング中に取り出されたフォーマット 1 の CCW で、バイト・カウントが 0 の場合は、プログラム・チェック条件が認識され、サブチャンネルは、サブチャンネルの状態と装置から受信した状況に応じて、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを伴う状況保留にされます。

注: 間接データ・アドレッシングが使用されているときの、CCW に関連した記憶域の説明については、15-34ページの『CCW 間接データ・アドレッシング』を参照してください。

プログラミング上の注意:

1. カウントが 0 であるフォーマット 1 の CCW は有効なので、プログラムで CCW のカウント・フィールドを使用して、入出力装置にデータを転送しないことを指定することができます。その場合、装置がデータ転送を要求すると、データ転送を中止するよ

う指示する信号が装置に送られます。そのときに、SLI フラグとコマンド・チェーン・フラグに 1 が指定されていて、装置にデータ転送中止の信号を送った後で異常な条件が検出されなかった場合は、装置から装置終了を受信した時点で新しい操作が開始されます。

2. サブチャンネルが誤長表示抑止モードにあり、現 CCW 内のデータ・チェーン・フラグが 0 であり、かつ操作が即時操作として実行される場合は、SLI フラグの設定に関係なく、誤長は示されません。

サブチャンネルが誤長表示モードにあり、現 CCW のデータ・チェーン・フラグが 0 であり、かつ、操作が即時操作として実行される場合は、現 CCW のカウント・フィールドに 0 以外の値が指定されていれば、CCW の SLI フラグで抑止されていない限り誤長が示されます。しかし、CCW のカウント・フィールドに 0 の値が指定されている場合は、誤長は示されません。

データ・チェーン中、カウント・フィールドが 0 である新しい CCW が取り出された場合、または、データ・チェーン・フラグが 1 に設定され、カウント・フィールドが 0 に設定されている CCW が取り出された場合は、チャンネル・サブシステムはプログラム・チェック条件を認識します。

チェーンニング

チャンネル・サブシステムは、CCW で指定されている情報の転送を完了した後で、新しい CCW を取り出して開始機能の実行を続けることができます。このように新しい CCW を取り出す操作をチェーンニングと呼び、このようなシーケンスに属している CCW をチェーンニングされた CCW と呼びます。

チェーンニングは、記憶域内の連続したダブルワード位置にある CCW の間で行われます。チェーンニングは、アドレスの昇順に従って進められます。つまり、新しい CCW のアドレスは、現 CCW のアドレスに 8 を加算することによって取得されます。チャンネル内転送コマンドを使用すれば、不連続な記憶域にある 2 つの CCW チェーンを結合することができます。1 つのチェーン内のすべての CCW が、オリジナルの START SUB-CHANNEL 命令で指定されているサブチャンネルに関連した入出力装置に適用されます。

チェーンニングのタイプには、データ・チェーンニングとコマンド・チェーンニングの 2 つがあります。チェーンニングは、CCW 中のデータ・チェーン (CD) フラグとコマンド・チェーン (CC) フラグ、および誤長表示抑止 (SLI) フラグにより制御されます。これらのフラグは、現 CCW が使い尽くされたとき、および装置から終了状況が受信されたときにチャンネル・サブシステムがとるアクションを指定します。これらのアクションは、15-30 ページの図15-6 に示すとおりです。

チェーンニングの指定は、実際にはチャンネル内転送コマンドを介して伝搬されます。チェーンニングの過程でチャンネル内転送コマンドが取り出されると、チャンネル内転送コマンドに指定されている CCW が、チャンネル内転送コマンドに先行する CCW で指定されているタイプのチェーンニングに使用されます。

チャンネル内転送を指定するフォーマット 0 の CCW では、CD フラグと CC フラグは無視されます。チャンネル内転送コマンドを指定するフォーマット 1 の CCW では、CD フラグと CC フラグは 0 でなければなりません。さもないと、プログラム・チェック条件が認識されません。

現 CCW 内の フラグ			カウントが使い尽くされたとき、またはチャンネル終了を受信したときの、 サブチャンネルでのアクション						
			即時操作				非即時操作		
			誤長 抑止モード ¹		誤長 表示モード		カウントが使い 尽くされたとき		カウント が使い尽 くされて おらず ^{CE} を受信 したとき
CD	CC	SLI	CCW カウント ≠0	CCW カウント =0	CCW カウント ≠0	CCW カウント =0	CE を受信し ていない	CE を受信	
0	0	0	End, NIL	End, NIL	End, IL	End, NIL	Stop, IL	End, NIL	End, IL
0	0	1	End, NIL	End, NIL	End, NIL	End, NIL	Stop, NIL	End, NIL	End, NIL
0	1	0	CC	CC	End, IL	CC	Stop, IL	CC	End, IL
0	1	1	CC	CC	CC	CC	Stop, CC	CC	CC
1	-	-	End, NIL	PC	End, IL	PC	CD	*	End, IL

説明:

- 選択されたビットは無視され、0 または 1 のいずれかになります。
- * この状況は有効な形で生じることはありません。データ・チェーニングが指定されているときは、新しい CCW が操作の制御権を得るのは、現 CCW に指定されているデータの最後のバイトが転送された後で、かつ、装置からデータまたは状況の転送を求める次の要求が出される前です。新しい CCW (プログラム・チェック条件も認識されない限り、この CCW 内のカウントは 0 になることはありません) が、操作の制御権を獲得しています。
- ¹ フォーマット 0 の CCW が指定されているときは、カウント・フィールドには 0 以外の値が含まれていなければなりません。さもないと、プログラム・チェック条件により操作は中止されます。
- CC チャンネル・サブシステムは、装置終了を受信するとコマンド・チェーニングを行います。
- CD データ・チェーン・フラグが指定されていると、チャンネル・サブシステムは、同じ操作に使用する新しい CCW をただちに取り出します。このようにして取り出された CCW のカウント・フィールドが 0 でない限り、操作が続行されます。カウントが 0 の場合は、プログラム・チェック条件が生じて、操作は中止されます。
- CE 装置から送られてくるチャンネル終了で、ブロックの終わりを示します。
- End 操作は中止されます。
- IL サブチャンネルで生成される後続の割り込み条件とともに誤長が示されます。
- NIL サブチャンネルで生成される後続の割り込み条件とともに誤長は示されません。
- PC この状況は有効な形で生じることはありません。データ・チェーン・フラグが 1 に設定され、カウント・フィールドが 0 である CCW が取り出されると、チャンネル・サブシステムはプログラム・チェック条件を認識します。
- Stop 装置にはデータ転送を中止するよう指示する信号が送られますが、サブチャンネルは、チャンネル終了を受信するまではサブチャンネル・アクティブのままになっています。

図 15-6. サブチャンネルのチェーニング・アクション

プログラミング上の注意: ORB のワード 1 のビット 9 が 1 のときは、チャンネル・サブシステムは、チューニングされた CCW (チャンネル内転送コマンドによりリンクされた CCW も含む) を無制限に事前取り出しすることができます。ORB で事前取り出しが許されているときは、サブチャンネルが現在延期されていて、かつ再開保留状態でない場合を除き、START SUBCHANNEL の実行後、操作について 1 次割り込み条件が受信されるまでの間は、チャンネル・プログラムの変更は行わないでください。

データ・チューニング

データ・チューニング中にチャンネル・サブシステムが取り出す新しい CCW は、オリジナルの入出力操作に使用する新しい記憶域を定義します。チャンネル・パスが並列入出力インターフェース・タイプのものである場合は、入出力装置での操作の実行が影響を受けることはありません。チャンネル・パスが直列入出力インターフェース・タイプのものである場合は、入出力装置での操作の実行は、まったく影響を受けないこともあり、また装置モデルによっては、装置チェック状況を伴って中止されることもあります。入出力装置での操作が影響を受けず、現 CCW で指定されているすべてのデータが主記憶機構または入出力装置に転送されると、データ・チューニングにより、新しい CCW に指定されている記憶域を使用して操作が続行されます。新しい CCW のコマンド・コード・フィールドの内容は、チャンネル内転送を指定している場合以外は無視されます。

データ・チューニングは、現 CCW で指定されているデータの最後のバイトが、主記憶機構または入出力装置に転送された直後に起きるものと見なされます。データ転送の最後のバイトが、主記憶機構に入れられるかまたは入出力装置により受理されると、新 CCW が操作の制御を引き継ぎます。現 CCW のカウントが使い尽くされた後で、新 CCW に指定されている記憶域への (またはそこからの) データ転送がまだ始まっていないときに、装置がチャンネル終了を送信した場合は、終了した操作に関連した SCSW は新 CCW に関する情報を示していません。

新 CCW の中、またはその取り出しの時点でプログラミング・エラーが検出された場合は、そのエラーが示され、そして、装置が新 CCW で指定されたデータを転送しようとしたときに、操作を終結するよう指示する信号が装置に送られます。装置が、新 CCW に指定されているデータを転送する前にチャンネル終了条件を示す信号を発信した場合は、終了に関連した SCSW の中にプログラム・チェックまたは記憶保護チェックが示されます。新 CCW のアドレスが無効であるか、その記憶位置が取り出しに対して保護されているか、または介在するチャンネル内転送コマンドでプログラミング・エラーが検出さ

れた場合を除き、SCSW の内容は新 CCW に関するものです。存在しないかまたは保護されている区域を指すデータ・アドレスがあった場合、エラーが示されるのは、入出力装置が、その無効な記憶位置への (またはそこからの) データの転送を試みた後です。

入力操作でデータ・チューニングが行われると、現 CCW に指定されているすべてのデータが主記憶機構に入れられた時点で、新 CCW が取り出されます。出力操作では、チャンネル・サブシステムは、データ・チューニングが起きる前に、主記憶機構から新しい CCW を取り出すことがあります。ただし、このように事前取り出した CCW の中にプログラミング・エラーがあったとしても、現 CCW に指定されているすべてのデータが入出力装置に転送されてしまうまでは、操作の実行がそのエラーの影響を受けることはありません。入出力装置が、現 CCW に指定されているすべてのデータが転送される前に操作を終結した場合は、事前取り出しされた CCW に関連した条件は、プログラムには示されません。ORB 内の事前取り出しビットの設定によっては、無制限の事前取り出しが許可されます。(15-22ページの『事前取り出し制御 (P)』を参照してください。) 無制限の事前取り出しが許可されていないときに、出力操作が指定されている場合は、事前取り出しできるのは、1 つのデータ域を記述する 1 つの CCW のみです。事前取り出しされた CCW がチャンネル内転送を指定している場合は、現 CCW が使い尽くされる前に、さらにもう 1 つだけ CCW が取り出されることがあります。

プログラミング上の注意:

1. ORB で無制限事前取り出しが指定されていない場合は、CCW の事前取り出しは行われません。ただし、出力操作でのデータ・チューニングが指定されているときは、1 つのデータ域を記述する CCW が一度に 1 つだけ事前取り出しできます。

入出力操作に関する ORB で、事前取り出しが許されることが指定されている場合は、CCW と IDAW および関連のデータ域を、幾つでも事前取り出しして、チャンネル・サブシステムのバッファに入れることができます。

ORB で無制限の事前取り出しが許可されているときもそうでないときも、エラーを通知して操作を中止するためのアクションは同じです。ただし、無制限の事前取り出しが指定されているときにエラー条件が検出された場合は、エラーの時点で実行されていたチャンネル・コマンドに関して、チャンネル・サブシステムでの中止の地点と入出力装置での中止の地点が異なる可能性があるということを、チャンネル・サブシステムとプログラムの両方が認識する必要があります。チャンネル・サブシステムは、該当の

CCW アドレスをサブチャンネル状況ワードのワード 1 に格納することにより、チャンネル・サブシステムでの中止の地点を示し、2 次 CCW アドレスをフォーマット 0 拡張状況ワードのワード 4 に格納することにより、入出力装置での中止の地点を示します。

ORB で事前取り出しが指定されているときは、START SUBCHANNEL が実行された後、または自己規定チャンネル・プログラムが使用された後で、CCW に変更を加えた場合の結果は、予測不能です。(自己規定チャンネル・プログラムの定義については、注意 2 を参照してください。)

2. データ・チェーニングを使用することにより、主記憶機構と入出力装置の間で転送される情報を、転送の時点で再編成することができます。つまり、データ・チェーニングでは、2 つの不連続記憶域間で情報のブロックを転送でき、またスキップ機能と一緒にデータ・チェーニングを使用すれば、プログラムは、1 ブロックのデータの指定された部分を主記憶機構に格納することができます。

入力操作中に、プログラムが、現 CCW の制御下でデータが格納された位置へのデータ・チェーニングを指定しているときは、チャンネル・サブシステムは、次の CCW を取り出すときに、その位置の新しい内容を取り出します。これは、その位置に、現 CCW の制御下で転送された最後のバイトが含まれている場合でも、同じです。チャンネル・プログラムが、データ・チェーニングを指定している CCW により記憶域に格納されている CCW へのデータ・チェーニングを行うとき、入力ブロックを自己規定ブロックと呼びます。自己規定ブロックには、同じ入力ブロック内の後続データの記憶位置とカウントを指定する 1 つ以上の CCW が含まれています。

自己規定ブロックを使用するのは、未検査データを使用するのと同じことです。つまり、入出力データ転送で情報のブロックの妥当性に影響を与えるような誤動作が起きても、それが通知されるのはデータ転送の完了時です。通常、この種のエラー条件により、操作の実行が途中で中止されたり、また実行にその他の影響が現れることはありません。したがって、操作が完了するまでは、データとして読み取られた CCW が有効であるという保証はありません。このようにして読み取られた CCW にエラーがある場合に、現在の操作でその CCW が使用されると、後続のデータが主記憶機構内の誤った位置に格納されることがあり、その結果、記憶保護キーおよびアドレス限界検査機能(使用されている場合)の制御に基づいて、その CCW の内容が破壊されてしまうことがあります。

3. データ・チェーニング時に、入出力装置がデータ・ストリーミング機能を使用してデータを転送する場合は、CCW に小さいバイト・カウント値が指定されていると、オーバーランまたはチェーニング・チェック条件が認識されることがあります。指定できる妥当な最小バイト数は、システム・モデルとシステム・アクティビティによって異なります。

コマンド・チェーニング

コマンド・チェーニング中にチャンネル・サブシステムが取り出す新しい CCW は、新しい入出力操作を指定します。チャンネル・サブシステムは、現在の操作について装置終了信号を受け取った時点で、新しい CCW を取り出します。新しい CCW で S フラグが 1 に設定されておらず、かつ異常な条件が検出されていなければ、チャンネル・サブシステムは新しい操作を開始します。S フラグが 1 に設定されているか、または異常な条件が存在する場合は、コマンド・チェーニングは抑止されます。コマンド・チェーニングが行われる場合は、現在の操作が完了しても入出力割り込みは起きず、現在の操作中に転送されたデータの量を示すカウントは、プログラムに提示されません。データ転送を含む操作の場合は、新しいコマンドは、常に装置上の次のブロックのデータに適用されます。

コマンド・チェーニングが行われ、新しい操作が開始されるのは、現在の操作で異常な条件が検出されていない場合に限られます。具体的には、チャンネル・サブシステムがコマンド・チェーニングの結果として新しい入出力操作を開始するのは、(1) 装置終了、(2) 装置終了と状況修飾、(3) 装置終了とチャンネル終了、または、(4) 装置終了とチャンネル終了と状況修飾の、いずれかの組み合わせのビットのみを含む状況バイトを受け取ったときです。最初の 2 つの場合は、装置終了の前にチャンネル終了が通知され、他の状況ビットはすべて 0 に設定されます。アテンション、装置チェック、装置例外、誤長、プログラム・チェック、または記憶保護チェックなどの条件が発生した場合は、操作シーケンスは終結し、現在の操作に関連した状況の結果として割り込み条件が生成されます。この場合は、新しい CCW は取り出されません。現 CCW の SLI フラグが 1 に設定されている場合は、誤長条件が起きてもコマンド・チェーニングは抑止されません。

入出力装置が、装置終了信号、またはチャンネル終了および装置終了信号とともに、状況修飾条件を提示した場合は、CCW の順次チェーニングに例外が生じます。コマンド・チェーニングが指定されていて、異常な条件が何も検出されなかった場合、または、前にコマンド再試行が通知されていて、即時再試行ができなかった場合は、状況修飾ビットと装置終了ビットの両方があると、チャンネル・サブシステムは、CCW の順次実行を変更しま

す。コマンド・チェーニングが指定されているときに、状況修飾ビットと装置終了日があると、チャンネル・サブシステムは、チェーニングを指定している CCW より 16 だけ大きい主記憶機構アドレスを持つ CCW を取り出して、その CCW にチェーニングします。前にコマンド再試行を指示する信号が出されていて、即時再試行ができなかったときに、これらの状況が存在する場合は、チャンネル・サブシステムは、最初に再試行が指示された CCW より 8 だけ大きい記憶アドレスを持つ CCW へのコマンド・チェーニングを行います。

コマンド・チェーニングとデータ・チェーニングが両方とも指定されているときは、操作に関連した最初の CCW は実行する操作を指定し、最後の CCW は別の操作が続いているかどうかを指定します。

プログラミング上の注意: プログラムでコマンド・チェーニングを使用することにより、1 つの START SUBCHANNEL 命令で複数ブロックのデータの転送を開始することができます。さらに、他のコマンド (ディスク・アクセス・メカニズムの位置決めなど) を実行できるように、また各操作が終了するたびにプログラムによる介入を必要とせずに複数のデータ転送操作を実行できるように、サブチャンネルを設定することもできます。状況修飾条件が存在するときにコマンド・チェーニングを使用すれば、チャンネル・サブシステムは、入出力装置から提供される信号に応じて通常の操作シーケンスを変更することができます。

スキップ

スキップは、入出力操作の実行中に主記憶機構参照を抑制させます。スキップは、読み取り、逆方向読み取り、センス ID、およびセンス操作についてのみ定義されており、スキップ・フラグにより制御されます。スキップ・フラグは、各 CCW ごとに個別に指定できます。スキップ・フラグが 1 のときは、スキップが発生します。0 のときは、通常の操作が行われます。上記以外のすべての操作では、スキップ・フラグの設定は無視されません。

スキップの影響を受けるのは、チャンネル・サブシステムでの情報の取り扱いのみです。入出力装置での操作は通常どおりに進行し、情報が転送されます。チャンネル・サブシステムは、カウンタの更新を続けますが、主記憶機構には情報を格納しません。スキップによりチェーニングが妨げられることはありません。データ・チェーニングの場合は、新 CCW のスキップ・フラグが 0 であれば、通常の操作が再開されます。

スキップの実行時には、データ・アドレスが無効または保護されているかどうかは、検査されません。

プログラミング上の注意: プログラムでスキップとデータ・チェーニングを併用することにより、1 ブロックの情報のうちの指定した部分を、入出力装置から主記憶機構に格納することができます。

プログラム制御割り込み

プログラムでプログラム制御割り込み (PCI) 機能を使用することにより、入出力操作の実行中に入出力割り込みを起こさせることができます。この機能は、CCW の PCI フラグにより制御されます。PCI フラグの値も関連の割り込み要求も、現在の操作の実行には影響を与えません。

現在の開始または再開機能用として指定されている最初の CCW でも、またチェーニング時に取り出される CCW でも、PCI フラグは 1 の値をとることができます。新たに現 CCW となった CCW の PCI フラグが 1 である場合は、サブチャンネルは中間状況を伴う状況保留状態になり、入出力割り込み要求が生成されます。サブチャンネルがどの時点で状況保留になるかは、以下に示すように、現在の開始機能または再開機能の進行状況によって異なります。

1. 開始機能または再開機能に関連した最初の CCW の PCI フラグが 1 である場合は、サブチャンネルは、コマンドが受理された後でのみ、中間状況を伴う状況保留状態になります。
2. データ・チェーニング時に現 CCW となった CCW の PCI フラグが 1 である場合は、サブチャンネルは、その前の CCW に指定されているすべてのデータが転送された時点で、中間状態を伴う状況保留状態になります。
3. コマンド・チェーニング中に現 CCW となった CCW の PCI フラグが 1 である場合は、サブチャンネルは、その CCW が現 CCW になった時点で、中間状況を伴う状況保留状態になります。

上記いずれの場合も、サブチャンネルが入出力割り込み可能状態にあるときに、どの時点で割り込みが起きるかは、システム内での現在の活動によって異なり、場合によっては遅延が生じることもあります。PCI フラグが原因で割り込み要求が生成される時点と、CCW に指定されている区域への (またはそこから) データ転送が完了する程度との間には、予測可能な関係はありません。ただし、SCSW の中のすべてのフィールドは、同じ時点に関する情報を含んでいます。

PCI フラグが原因で保留された中間割り込み条件は、TEST SUBCHANNEL または CLEAR SUBCHANNEL により消去されない限り、チェーニング中

は保留のままになっています。この中間割り込み条件が消去される前に、PCI フラグが 1 である別の CCW が現 CCW になっても、保持される割り込み条件は 1 つだけです。

中間割り込みが起きることがあるのは、中間割り込み条件の原因となった CCW に指定されている操作、または、その後で現 CCW になった CCW に指定されている操作で、サブチャンネルがサブチャンネル/装置アクティブであるときです。該当の操作または操作チェーンが終結する前に中間割り込み条件が消去されていない場合は、その操作または操作チェーンの終結時に、その条件が 1 次割り込み条件とともに示されます。中間割り込み条件は、サブチャンネルがサブチャンネル・アクティブであるときに TEST SUBCHANNEL により消去することができます。

TEST SUBCHANNEL により格納された SCSW が、中間状況を伴う状況保留状態にあり、操作または操作チェーンがまだ終結していない (つまり、活動制御フィールドがサブチャンネル/装置アクティブまたは延期を示している) 場合は、CCW アドレス・フィールドには、PCI フラグが 1 であって次に現 CCW になる最新の CCW のアドレスより 8 だけ大きいアドレスが含まれているか、または、続いて現 CCW になった CCW のアドレスより 8 だけ大きいアドレスが含まれています。SCSW の中で 1 次状況ビットも 1 に設定されている場合を除き、装置状況フィールドに 0 が入っており、カウントは予測不能です。

SCSW が格納されるときに、PCI 以外のサブチャンネル状況条件が示されることがあります。サブチャンネルが 1 次状況を伴う状況保留状態ではない場合は、これらの条件は再度示されることも示されないこともあります。事前取り出しの際にそのサブチャンネル状況条件が検出され、その条件が操作に影響を与える前に操作または操作チェーンが終結した場合は、その条件はリセットされ、後でサブチャンネルが 1 次状況を伴う状況保留状態になったときは、その条件は示されません。サブチャンネル状況条件が操作に影響を与えた場合は、サブチャンネルが 1 次状況を伴う状況保留状態になったときに、その条件が示されます。

プログラム割り込み条件が、サブチャンネルで操作または操作チェーンが終結するまで保留のままになっている場合は、単一の割り込み要求が存在しています。その後で TEST SUBCHANNEL が実行されると、格納された SCSW は 1 次割り込み条件と中間割り込み条件の両方

を示し、サブチャンネル状況フィールドの PCI ビットは 1 になっています。

チャンネル内転送コマンドを指定している CCW を除くすべての CCW で、PCI フラグの値が検査されます。初期プログラム・ローディングの際には、PCI フラグは無視されます。

プログラミング上の注意:

1. プログラム制御割り込みは、入出力操作中のチェーニングの進行状況をプログラムに知らせるための手段として利用できます。これにより、プログラム式の動的主記憶域割り振りが可能になります。
2. コマンド再試行が原因で、PCI フラグが 1 に設定されている CCW が再試行されると、複数の PCI 割り込みが起きることがあります。(15-39ページの『コマンド再試行』を参照してください。)

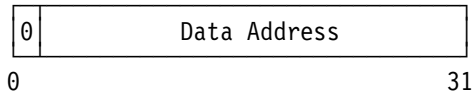
CCW 間接データ・アドレッシング

CCW 間接データ・アドレッシングを使用すると、主記憶機構内の不連続の 2K バイトまたは 4 K バイトのブロックにまたがるデータの転送を、1 つのチャンネル・コマンド・ワードで制御することができます。さらに、プログラムで CCW 間接データ・アドレッシングを使用して、フォーマット 0 の CCW では 16M バイトを超えるデータ・アドレス、そしてフォーマット 1 の CCW では 2G バイトを超えるデータ・アドレスを指定することもできます。

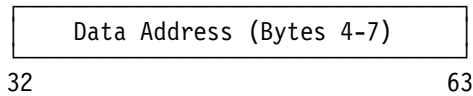
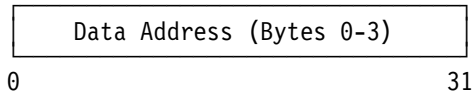
CCW 間接データ・アドレッシングは、CCW の中のフラグの 1 つに指定されます。このフラグが 1 のときは、データ・アドレスを直接データ・アドレッシングには使用しないことを示します。代わりに、アドレスは、間接データ・アドレス・ワード (IDAW) と呼ばれるワードまたはダブルワードのリストを指し示します。これらのワードには、主記憶機構内の 1 つのデータ域を示す絶対アドレスが 1 つずつ入っています。

IDAWには、フォーマット 1 およびフォーマット 2 と呼ばれる 2 つのフォーマットがあり、これは、実行されているチャンネル・プログラムに関連した ORB のフォーマット 2 IDAW 制御 (ビット 14) によって決まります。フォーマット 2 IDAW 制御が 0 のときは、IDAW はフォーマット 1 で、31 ビット・アドレスを含むワードです。この制御が 1 のときは、IDAW はフォーマット 2 で、64 ビット・アドレスを含むダブルワードです。IDAW のフォーマットは以下のとおりです。

フォーマット 1 の IDAW



フォーマット 2 の IDAW



ビット 0 (フォーマット 1) は将来の使用に備えて予約済みであり、0 でなければなりません。さもないと、以下に述べるようにプログラム・チェック条件が認識されることがあります。

フォーマット 1 の IDAW は、2K バイトの主記憶域ブロックに含まれるデータ域を指定するもので、0 から $2^{31} - 1$ までの範囲内の記憶域をアドレッシングすることができます。

フォーマット 2 の IDAW は、2K バイトまたは 4K バイトの主記憶域ブロックに含まれるデータ域を指定するもので、0 から $2^{64} - 1$ までの範囲内の記憶域をアドレッシングできます。2K か 4K かは、実行されているチャンネル・プログラムに関連した ORB の 2K IDAW 制御 (ビット 15) によって決まります。2K IDAW 制御ビットが 0 のときは、指定されたチャンネル・プログラムのフォーマット 2 IDAW は、4K バイトの主記憶域ブロックを示します。2K IDAW 制御ビットが 1 のときは、フォーマット 2 IDAW は 2K バイトのデータ域ブロックを示します。指定したチャンネル・プログラムに関連する IDAW は、すべて同じ IDAW フォーマットでなければならず、すべての IDAW が同じサイズの記憶域ブロックを指定していなければなりません。

CCW の間接データ・アドレッシング・ビットが 1 のときは、CCW のデータ・アドレス・フィールドは、コマンドでデータ転送に使用する最初の IDAW の位置を指定します。その CCW に対応するデータ転送を完了するためにさらに IDAW が必要であれば、それらの IDAW は記憶域内の連続した位置に置かれます。1 つの CCW に必要な IDAW の数は、ORB に指定されている IDAW フォーマットと、CCW のカウント・フィールドと、最初の IDAW の中のデータ・アドレスによって決まります。例えば、(1) ORB が 4K バイト・ブロックのフォーマット 2 IDAW を指定しており、(2) CCW カウント・フィールドが 8K バイトを指定しており、か

つ、(3) 最初の IDAW が 4K バイト・ブロックの途中の位置を指定しているとするれば、3 個の IDAW が必要です。

CCW により指定される IDAW は、任意の位置を指定できます。その後は、2K バイト・ブロック境界 (フォーマット 1 またはフォーマット 2 の IDAW) または 4K バイト・ブロック境界 (フォーマット 2 の IDAW) に達するまで、読み取り、書き込み、制御、センス ID、およびセンス・コマンドの場合は順に大きい記憶位置との間でデータが転送され、逆方向読み取りコマンドの場合は順に小さい記憶位置にデータが転送されます。そして、データ転送の制御が次の IDAW に渡されません。コマンドに応じて、2 番目およびそれ以降の IDAW は、2K バイト・ブロック (フォーマット 1 またはフォーマット 2 の IDAW) または 4K バイト・ブロック (フォーマット 2 の IDAW) の最初のバイト (逆方向読み取りの場合は最後のバイト) を指定していなければなりません。したがって、読み取り、書き込み、制御、センス ID、およびセンス・コマンドの場合は、この種のフォーマット 1 IDAW ではビット位置 21-31 がすべて 0 でなければならず、この種のフォーマット 2 IDAW ではビット位置 53-63 (2K バイト・ブロック) または 52-63 (4K バイト・ブロック) がすべて 0 でなければなりません。逆方向読み取りコマンドの場合は、この種のフォーマット 1 IDAW ではビット位置 21-31 がすべて 1 でなければならず、この種のフォーマット 2 IDAW ではビット位置 53-63 (2K バイト・ブロック) または 52-63 (4K バイト・ブロック) がすべて 1 でなければなりません。これらの規則のいずれかに違反している場合は、プログラム・チェック条件が認識されます。

IDAW によるデータ・アドレスの指定に特有の制限事項がある点以外は、データ・アドレスについてとられるアクション (例えば保護記憶域や無効アドレスに対するアクション)、およびデータの事前取り出しについてとられるアクションは、すべて、間接データ・アドレッシングを使用していない場合と同じです。

現 CCW または事前取り出しされた CCW に関する IDAW は、事前取り出しが可能です。事前取り出しのできる IDAW の数は、IDAW を指す CCW 内のカウントを満たすために必要な数を超えることはできません。1 つの IDAW について最後のバイトが転送されると、次の IDAW がデータ転送の制御権を獲得します。IDAW が入出力操作中のどの時点でデータ転送の制御権を得るかについては、データ・チェーニングの場合と同じアクションがとられます。つまり、まだ CCW のカウントが 0 に達していないときは、その CCW に関する前の IDAW について最後のバイトが転送されると、

新しい IDAW がデータ転送の制御権を得ます。これは、新しい IDAW に関するデータ・バイトを転送する前に、(1) チャンネル終了、(2) チャンネル終了と装置終了、または、(3) チャンネル終了、装置終了、および状況修飾が受信された場合でも、同じです。

ある CCW の先行 IDAW について最後のデータ・バイトを転送したときに、CCW 内のカウントが 0 に達した場合は、事前取り出しされた IDAW には入出力操作の制御権は与えられません。事前取り出しされた IDAW の中で検出されたプログラム・エラーまたはアクセス・エラーは、その IDAW がデータ転送の制御権を得るまでは、プログラムに提示されません。しかし、チャンネル・サブシステムが、事前取り出しされた IDAW またはそれに関連したキーの内容について無効な CBC を検出した場合は、その条件は、検出された時点で、つまりその IDAW がデータ転送の制御権を得る前に、プログラムに提示されることがあります。IDAW またはそれに関連したキーの内容について、無効な CBC が検出されたときに示される標識の説明については、16-27 ページの『チャンネル制御チェック』を参照してください。

ビット 1-31 (フォーマット 1) またはビット 0-63 (フォーマット 2) は、データ転送で使用する最初のバイトの絶対記憶位置を指定します。フォーマット 1 の IDAW が指定されているときは、チャンネル・サブシステムは、ビット 1 の左側に 33 個の 0 を付加して、64 ビットの絶対主記憶アドレスを形成します。

CCW の IDAW フラグが 1 に設定されているときに、次のいずれかの条件が起きたとします。

1. ORB 内でフォーマット 1 の IDAW が指定されており、CCW 内のアドレスはワード境界上に最初の IDAW を指定していない。
2. ORB 内でフォーマット 2 の IDAW が指定されており、CCW 内のアドレスはダブルワード境界上に最初の IDAW を指定していない。
3. CCW 内のアドレスは、物理的に使用可能でない記憶位置を指定している。
4. CCW 内のアドレスにより指定されている記憶位置へのアクセスが、記憶保護により禁止されている。
5. 最初の IDAW のビット 0 が 0 でない (フォーマット 1 の場合のみ)。

この場合は、スキップ・フラグの設定に関係なく、モデルに応じて次の 2 つのアクションのいずれかがとられます (上記の条件 5 が真の場合は、アクション 2 が取られます)。

1. 装置で操作を開始する前に、上記の条件が存在するかどうかを検査されます。いずれかの条件が認識された場合は、入出力操作は開始されず、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態にされます。
2. 上記の条件についての検査を行う前に、装置で操作が開始されます。装置がデータを転送しようとする時、入出力操作を中止するよう指示する信号が装置に送られ、サブチャンネルは、サブチャンネルの状態および装置から提示された状況に応じて、1 次状況、2 次状況、および警報状況を伴う状況保留状態にされます。

チャンネル・プログラム実行の延期

プログラムで、RESUME SUBCHANNEL と一緒に延期機能を使用することにより、チャンネル・プログラムの実行をいったん停止し、そして再開することができません。延期機能の開始は、延期制御 (ORB のワード 1 のビット 4) の設定に従って制御されます。このサブチャンネルについて ORB 内で延期制御が指定されているときに、S フラグが 1 に設定されている CCW が現 CCW になると、延期機能に信号が送られます。このフラグは、チャンネル・プログラムの最初の CCW 内で指定することも、コマンド・チェーニング中に取り出される CCW 内で指定することもできます。(1) ORB の中で延期制御ビットが 0 に設定されているか、または、(2) CCW がデータ・チェーニング中に取り出された場合は、S フラグは無効であり、プログラム・チェック条件が認識されます (データ・チェーニング中に検出されたプログラミング・エラーの取り扱いについては、15-31 ページの『データ・チェーニング』を参照)。

延期機能が認識されると、該当の CCW が現 CCW になった時点で、チャンネル・プログラムの実行が延期されます (いつ CCW が現 CCW になるかの定義については、15-25 ページの『チャンネル・コマンド・ワード』を参照)。コマンド・チェーニング時に延期が起きた場合は、コマンド・チェーニングが効力を失ったことを示す信号が装置に送られます。

RESUME SUBCHANNEL は、チャンネル・プログラム延期を引き起こした CCW は変更されている可能性があること、その CCW を再取り出しする必要があること、そして、その CCW の内容を調べてフラグの設定を判別する必要があることを、通知します。S フラグが 1 であれば、その CCW は実行されません。CCW が有効で、その中の S フラグが 0 であれば、実行が開始されます (14-10 ページの『RESUME SUBCHANNEL』および 15-17 ページの『開始機能と再開機能』を参照)。

コマンド・チューニング時に、S フラグが有効に 1 に設定されている有効な CCW が現 CCW になり、再開保留条件が認識されていない場合は、延期機能が実行され、以下のアクションが以下の順序で行われます。

1. 装置に、操作のチェーンが終結したことが通知されます。
2. サブチャンネルで、チャンネル・プログラムの実行が延期されます。事前取り出しされている IDAW、CCW、およびデータは、すべて破棄されます。そして、次回にサブチャンネルが再開保留であると認識されたときに再開機能を実行できるように、サブチャンネルがセットアップされます。
3. 測定ブロック更新モードがアクティブであり、サブチャンネルがそのモードで稼働できるように設定されている場合は、測定データの累加値(サブチャンネル開始カウントやサンプル・カウントを含む)が、サブチャンネルの測定ブロック内の累算値に加算されます。サブチャンネルについてチャンネル・サブシステム・タイミング機構が使用可能でない場合は、サブチャンネル開始カウントは、測定ブロック内で更新される唯一の測定データです。(詳細については、17-1ページの『チャンネル・サブシステム・モニター』を参照してください。)

測定ブロックの更新時に測定チェック条件が検出された場合は、サブチャンネルでチャンネル・プログラムが終了します。サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態にされ、装置状況フィールドとサブチャンネル状況フィールドは 0 に設定され、フォーマット 0 の ESW の拡張状況フラグに測定チェック条件の 1 つが示されます。サブチャンネルは延期状態にはなりません。(16-11ページの『サブチャンネル制御フィールド』を参照してください。)

4. サブチャンネルは延期状態にされます。
5. この時点でサブチャンネルが再開保留状態にない場合は、ORB の延期割り込み抑止ビットが 0 であれば、延期を原因とする中間割り込み条件が認識されます。そうでない場合は、再開機能が実行されます。

コマンド・チューニング時に、S フラグが有効に 1 に設定されている有効な CCW が現 CCW になり、再開保留条件が認識された場合は、延期機能の代わりに再開機能が実行されます。

チャンネル・プログラムの最初の CCW で S フラグが有効に 1 に設定されていて、再開保留条件が認識されていない場合は、延期機能が実行され、以下のアクションが以下の順序で行われます。

1. 装置を選択する前に、チャンネル・プログラムの実行が延期されます。
2. 次回にサブチャンネルが再開保留状態にあると認識されたときに再開機能を実行できるように、サブチャンネルがセットアップされます。
3. 測定ブロック更新モードがアクティブであり、サブチャンネルがこのモードで稼働できるように設定されている場合は、SSCH+RSCH カウントが増加し、測定結果の機能保留時間(タイミング機構ビットの設定によって決まる)が、サブチャンネルの測定ブロック内の累算値に加算されます。

測定ブロックの更新時に測定チェック条件が検出された場合は、サブチャンネルでチャンネル・プログラムは開始されません。サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態にされます。据え置き条件コード 1 が設定され、開始保留ビットは 1 に設定されたままになります。装置状況フィールドとサブチャンネル状況フィールドは 0 に設定され、フォーマット 0 の ESW の拡張状況フラグに測定チェック条件の 1 つが示されます。サブチャンネルは延期状態にはなりません。(16-11ページの『サブチャンネル制御フィールド』を参照してください。)
4. サブチャンネルは延期状態にされます。
5. この時点でサブチャンネルが再開保留状態にない場合は、ORB の延期割り込み抑止制御ビットが 0 であれば、延期が原因でサブチャンネルは中間割り込み条件を伴う状況保留状態にされます。そうでない場合は、再開機能が実行されます。

チャンネル・プログラムの最初の CCW で S フラグが有効に 1 に設定されていて、再開保留条件が認識された場合は、延期機能の代わりに再開機能が実行されます。

プログラミング上の注意:

1. 指定されたサブチャンネルが延期されている場合は、MODIFY SUBCHANNEL および START SUBCHANNEL の実行完了時に条件コード 2 が設定されます。サブチャンネルが延期状態にあるときは、サブチャンネルでは開始機能が示されています。
2. 状況によっては、延期されているチャンネル・プログラムの実行を通常どおりに再開することが望ましくない場合があります。延期されているチャンネル・プログラムの実行を正常に終了させるには、次のようにすることができます。
 - a. 該当のサブチャンネルを指定して HALT SUBCHANNEL を実行する。

- b. チャンネル・プログラムの実行が再開されるときに装置に転送されるコマンドが、すべての修飾ビットに 0 が指定され (ノーオペレーション)、コマンド・チェーン・フラグに 0 が指定されている制御コマンドになるように、記憶域内で CCW を変更する。そして、SUB-CHANNEL を実行します。
- c. IRB が、ゼロ装置状況、ゼロ・サブチャンネル状況、および、1 次、2 次、および警報状況を伴う状況保留とともに、測定チェックを示しているときは、それは、サブチャンネルを延期状態にしようとしたときに測定チェックが検出されたことを示している場合があります。

3. 延期割り込みが抑止された場合は、前のサブチャンネル・アクティブ期間に適用される N 条件および DCTI 値は、プログラムには提供されません。サブチャンネルが延期状態にあるときに RESUME SUB-CHANNEL を実行すると、パス非稼働条件および N 条件は 0 にリセットされます。RESUME SUB-CHANNEL が実行されるときに、指定されたサブチャンネルが延期状態でない場合は、パス非稼働条件および N 条件はリセットされません。

コマンドとフラグ

図15-7 は、7 つのコマンドのコマンド・コードと、各コマンドについて定義されているフラグを示しています。チャンネル内転送を指定しているフォーマット 1 の CCW の場合を除き、どのコマンドでも、そのコマンドについて定義されていないフラグは無視されます。チャンネル内転送を指定するフォーマット 1 の CCW では、フラグはすべて予約済みであり、0 でなければなりません。

名称	コード	フラグ
書き込み	M M M M M M 0 1	CD CC SLI PCI IDA S
読み取り	M M M M M M 1 0	CD CC SLI SK PCI IDA S
逆方向読み取り	M M M M 1 1 0 0	CD CC SLI SK PCI IDA S
制御	M M M M M M 1 1	CD CC SLI PCI IDA S
センス	M M M M 0 1 0 0	CD CC SLI SK PCI IDA S
センス ID	1 1 1 0 0 1 0 0	CD CC SLI SK PCI IDA S
チャンネル内転送	X X X X 1 0 0 0	(See note below)

説明:

CC コマンド・チェーン
 CD データ・チェーン
 IDA 間接データ・アドレッシング
 M 修飾ビット
 PCI プログラム制御割り込み
 S 延期
 SK スキップ
 SLI 誤長表示抑止
 X フォーマット 0 の CCW では無視されます。フォーマット 1 の CCW では 0 でなければなりません。

注: フォーマット 0 のチャンネル内転送 CCW ではフラグは無視され、フォーマット 1 のチャンネル内転送 CCW ではフラグは 0 でなければなりません。

図 15-7. コマンド・コードとフラグ

すべてのフラグはそれぞれ個別に意味を持っていますが、CD フラグが 1 に設定されているときは CC フラグと SLI フラグは無視され、また出力転送操作では SK フラグは無視されます。フォーマット 0 の CCW に関連した即時操作では SLI フラグは無視され、その設定値に関係なく誤長表示は抑止されます。フォーマット 1 の CCW を実行しているときは、誤長表示抑止モードに応じて、即時操作で誤長表示が抑止されることがあります。初期プログラム・ローディングの際には、PCI フラグは無視されます。S フラグが 1 のときは、PCI フラグを除くすべてのフラグが無視されます。

プログラミング上の注意:

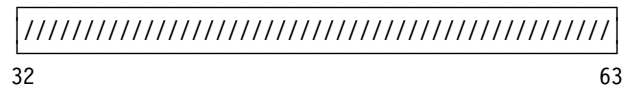
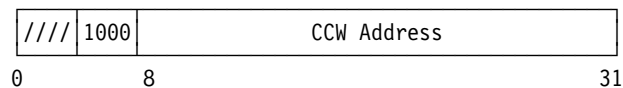
1. 入出力操作で転送されるデータの妥当性に影響を与える誤動作が起きた場合は、入出力装置 (制御装置) とチャンネル・サブシステムのどちらがエラーを検出したかに応じて、装置チェックまたはチャンネル・データ・チェックを使用して、操作の終わりにエラーが通知されます。システムが提供する検査機能を十分に活用するには、操作の終わりに達し、データの妥当性検査が完了するまでは、入力操作で読み取られたデータが使用されないようにしてください。同様に、書き込み操作では、データの転送と記録に影響を与える誤動作が検出されなかったことをプログラムが確認するまでは、主記憶機構内のデータのコピーが破壊されないようにしてください。
2. 1 つの装置で 256 個以上のチェーニングされたコマンドが実行され、どのコマンドでもデータがまったく転送されなかった場合は、エラー条件が認識され、入出力操作が中止されることがあります。この条件が認識されたときは、プログラム・チェックが示されます。
3. 誤長表示の抑止を必要とする CCW では、必ず SLI フラグを使用する必要があります。

チャンネル・プログラム内でのブランチ

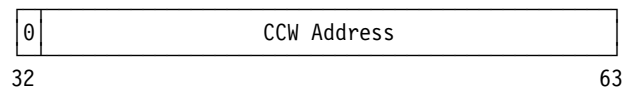
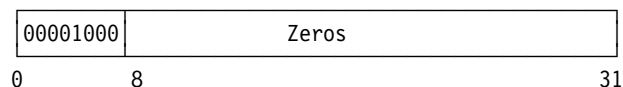
チャンネル・プログラム内では、通常は CCW が順次に行われますが、チャンネル・サブシステムはこの順次実行を変更するための 2 つの方法を提供しています。その 1 つはチャンネル内転送 (TIC) コマンド (15-39 ページの『チャンネル内転送』の説明を参照) で、これを使用すると、前に実行した CCW にループバックすること、またはチャンネル・プログラムの非連続セグメントに接続することができます。もう 1 つの方法は状況修飾装置状況ビット (『ESA/390 Common I/O-Device Commands』(SA22-7204) を参照) を使用するもので、これにより、チャンネル・プログラム内の次の CCW をバイパスするような条件を装置に生じさせることができます。

チャンネル内転送

フォーマット 0 の TIC CCW



フォーマット 1 の TIC CCW



チャンネル内転送を指定する CCW のデータ・アドレス・フィールドに指定されている絶対主記憶位置から、次の CCW が取り出されます。チャンネル内転送コマンドは何らかの入出力操作を開始するものではなく、したがってこのコマンドの実行は入出力装置には通知されません。チャンネル内転送コマンドの目的は、隣接するダブルワード位置に置かれていない 2 つの CCW を、アドレスの昇順に従ってチェーニングすることにあります。このコマンドは、データ・チェーニングとコマンド・チェーニングのどちらにも使用できます。

チャンネル内転送を指定する CCW のビット 29-31 (フォーマット 0) またはビット 61-63 (フォーマット 1) は、ダブルワード境界上の CCW を指定するために、すべて 0 でなければなりません。また、チャンネル内転送を指定する CCW は、その直前のチャンネル内転送に指定されている位置から取り出すことはできません。これらのエラーのいずれかが検出された場合、またはチャンネル内転送コマンドで無効なアドレスが指定されている場合は、プログラム・チェック条件が生成されます。チャンネル内転送コマンドを指定する CCW が、取り出しに対して保護されている位置にある CCW を指定している場合は、記憶保護チェック条件が生成されます。データ・チェーニング中にこれらのエラーが検出されると、入出力装置での操作は中止され、割り込み条件が生成されます。コマンド・チェーニング中のエラーの場合は、割り込み条件は生成されません。

フォーマット 0 の CCW の後半部分 (ビット 32-63) の内容は無視されます。同様に、フォーマット 0 の CCW のビット位置 0-3 の内容も、無視されます。

フォーマット 1 の CCW のビット位置 0-3 および 8-32 は、すべて 0 でなければなりません。さもないと、プログラム・チェック条件が生成されます。

コマンド再試行

チャンネル・サブシステムには、コマンド再試行を行う能力があります。コマンド再試行とは、入出力割り込みを使用せずにコマンドを実行しなおす手順です。制御装置は、特殊なシーケンスを使用して、2 通りの状況ビットの組み合わせを提示することにより、この再試行を開始します。即時再試行が可能なときは、制御装置は、装置終了と一緒に、チャンネル終了、装置チェック、および状況修飾状況ビットの特定の組み合わせを提示します。即時再試行ができないときは、制御装置の準備が整うまで、装置終了の提示が遅れます。装置終了のみが提示されたときは、前のコマンドが再度転送されます。装置終了に状況修飾が付随している場合は、コマンド再試行は行われず、チャンネル・サブシステムは、コマンド再試行の信号が出された CCW の次の CCW のコマンド・チェーニングします (状況修飾については、「*ESA/390 Common I/O-Device Commands*」(SA22-7204)を参照)。エラー条件が原因で、チャンネル・サブシステムがコマンド再試行を行うことができないとき、または、要求されたコマンド再試行開始に、装置終了または装置終了と状況修飾以外の状況ビットが付随しているときは、再試行は抑止され、サブチャンネルは状況保留になります。TEST SUBCHANNEL により格納される SCSW には、入出力装置が提示する状況が入っています。

プログラミング上の注意: プログラムでは、コマンド再試行により次のような結果が生じる可能性があることを予期しておく必要があります。

1. コマンド再試行が原因で、PCI を含む CCW が再試行された場合は、複数の PCI 割り込みが起きることがあります。
2. ある操作で使用している CCW が、その操作が正常に完了する前に変更された場合は、結果は予測不能です。

開始前の入出力操作の終結

START SUBCHANNEL または RESUME SUBCHANNEL が実行された後で、最初のコマンドが受理される前に、CANCEL SUBCHANNEL (この命令がインストールされている場合)、CLEAR SUBCHANNEL、または HALT SUBCHANNEL を使用して、サブチャンネルで開始機能を終了することができます。CANCEL SUBCHANNEL により入出力操作が終了した場合は、以後は入出力操作からの割り込み条件が起きることはなく、サブチャンネルで新たな開始機能を

開始できます。ただし、取り消された操作が開始保留状態にある間は、装置は使用中条件の信号を送っていることがあります。その場合は、装置はチャンネル・サブシステムに使用中解除信号を送る必要があります。その結果として、装置で次の操作が開始される前に、送信請求されていない装置終了状況が送信されることがあります。(15-13ページの『消去機能』および 15-14ページの『停止機能』も参照してください。)

開始時の入出力操作の終結

指定されたサブチャンネルが **START SUBCHANNEL** の実行が可能な状態にあると確認された後で、プログラムが指定する情報の妥当性と、関連装置の論理的可用性について、所定のテストが行われます。このテストが行われるのは、**START SUBCHANNEL** の実行中または実行後、およびコマンド・チェーニングとコマンド再試行の時点です。

サブチャンネルおよび入出力装置でデータ転送操作が開始されるのは、チャンネル・サブシステムがプログラミン
グ・エラーも装置エラーも検出せず、開始シーケンスで装置が応答として 0 状況を戻したときです。入出力操作の開始時に、チャンネル・サブシステムが異常条件を検出するか、または入出力装置が異常条件を示す信号を発信したときは、コマンドは受理されていません。その場合は、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留状態になります。据え置き条件コード 1 が設定され、開始保留ビットは 1 に設定されたままになります。

入出力操作の開始を妨げる条件については、**TEST SUBCHANNEL** により格納される **SCSW** に詳細情報が示されています。この場合、入出力装置は開始されず、**TEST SUBCHANNEL** の後で割り込み条件は生成されず、サブチャンネルはアイドル状態になります。コマンドが、使用中条件または非稼働条件が原因で拒否されたのではない限り、入出力装置は他の操作を開始するために即時に使用できるようになります。

コマンド・チェーニングまたはコマンド再試行により入出力操作が開始されるときに、異常条件が原因でコマンドが受理されなかったときは、割り込み条件が生成され、サブチャンネルは、装置から通知される状況に応じて、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを伴う状況保留状態になります。この条件を表す状況は、**TEST SUBCHANNEL** により消去されるまで、サブチャンネルに残されています。条件は、**SCSW** の対応するビットを使用してプログラムに提示されません。コマンド・チェーニング時に認識されたパス非稼働条件は、インターフェース制御チェック標識を使用して

プログラムに通知されます。入出力装置では、新たな入出力操作は開始されません。

START SUBCHANNEL は、関連装置から独立して実行されます。**START SUBCHANNEL** の実行の後で、ほとんどのプログラム指定の情報、装置可用性と装置状況、およびほとんどのエラー条件について、テストが行われます。開始機能の実行を妨げるような何らかの条件が検出されたときは、チャンネル・サブシステムにより割り込み条件が生成されてサブチャンネルに置かれ、その結果サブチャンネルは状況保留状態になります。

入出力操作の即時終結

入出力操作の開始時に、入出力装置は、コマンド・コードを受け取ったときに、ただちにコマンドを受理し、チャンネル終了条件を示す信号を発信することができます。このように、開始シーケンス中にチャンネル終了条件を通知させるような入出力操作を、*即時操作*と呼びます。コマンド・チェーニングが指定されておらず、コマンド再試行の信号も出されていないときに、入出力装置で即時コマンドを表す状況が生成されると、サブチャンネルは、**ORB** と **CCW** に指定されている情報および装置が提示した状況に応じて、1 次状況、2 次状況、中間状況、および警報状況のいずれかの組み合わせを伴う状況保留状態になります。チャンネル・プログラムの最初の操作が即時操作である場合は、据え置き条件コード 1 が設定され、状況標識と一緒に示されます。中間状況が示されるのは、**CCW** の **PCI** フラグが 1 に設定されている場合のみです (15-33ページの『プログラム制御割り込み』を参照)。

即時操作の後にコマンド・チェーニングが指定されており、その操作中に異常条件が検出されていない場合、または即時操作に対してコマンド再試行が行われる場合は、割り込み条件は生成されません。チェーン内の後続のコマンドは通常どおりに取り扱われ、一般に、最後の **CCW** についてチャンネル終了条件が生じると、1 次割り込み条件が生成されます。チャンネル終了とともに装置終了も通知された場合は、2 次割り込み条件も生成されません。

入出力操作の即時完了を示す信号が出されたときは、入出力装置との間のデータ転送は行われておらず、**CCW** 中のデータ・アドレスの妥当性は検査されません。サブチャンネルが誤長抑止モードにある場合は、誤長はプログラムに提示されず、コマンド・チェーニング (指定されている場合) が実行されます。サブチャンネルが誤長表示モードにある場合は、**SLI** フラグとコマンド・チェーン・フラグにより誤長とコマンド・チェーニングが制御されます。どのような場合に誤長表示が抑止されるかに

については、15-30ページの図15-6 に要約して示してあります。

プログラミング上の注意: コマンド・コードの中で操作全体が指定されている入出力操作は、即時操作として実行されることがあります。コマンドが即時操作として実行されるかどうかは、操作と装置タイプによって決まります。

データ転送時の入出力操作の終結

サブチャンネルに ORB の内容が渡されると、サブチャンネルは開始保留状態になります。入出力操作が開始され、コマンドが受理された後は、サブチャンネルはサブチャンネル/装置アクティブの状態になり、(1) チャンネル・サブシステムが装置誤動作を検出するか、(2) CLEAR SUBCHANNEL または HALT SUBCHANNEL の実行により操作が終結するか、または、(3) 装置から、1 次割り込み条件が認識される原因となる状況 (通常チャンネル終了) が受理されない限り、その状態のままになっています。コマンド・チェーニングおよびコマンド再試行が指定されていないか、または異常条件が原因でチェーニングが抑止されているときに、1 次状況として認識される状況が生じると、サブチャンネルでの操作が終結し、割り込み条件が生成されます。関連の SCSW の状況ビットに、1 次状況が示されるほか、異常条件があればそれも示されています。入出力装置は、1 次状況として認識される状況を、入出力操作の開始後いつでも提示することができ、その状況提示はまだデータ転送が始まっていない時点で生じることもあります。

データ転送を伴わない操作の場合は、通常、入出力装置がチャンネル終了条件のタイミングを制御します。データ転送操作の存続時間は可変であり、入出力装置により制御されることも、チャンネル・サブシステムにより制御されることもあります。

装置エラー、および CLEAR SUBCHANNEL、HALT SUBCHANNEL、および RESET CHANNEL の実行の場合を除き、以下のいずれかの条件に該当する場合は、チャンネル・サブシステムは、データ転送時に、入出力操作の実行を終結するよう指示する信号を装置に送ります。

- その操作用として指定されている記憶域が使い尽くされたか一杯になった。
- プログラム・チェック条件が検出された。
- 記憶保護チェック条件が検出された。
- チェーニング・チェック条件が検出された。

- 入出力操作の制御に影響を与えないチャンネル制御チェック条件が検出された。

上記の最初の条件が起きるのは、チャンネル・サブシステムが、この操作に関連した最後の CCW でカウントを 0 に減少させたときです。0 のカウントは、チャンネル・サブシステムが入出力操作で指定されているすべての情報を転送したことを示します。その他の 4 つの条件はエラーが原因で起きるものであり、いずれもデータ転送を未完のまま終結させます。いずれの場合も、装置からのサービス要求に対する応答として終結が通知され、その時点でデータ転送は中止されます。この操作のためのブロックが定義されていない場合は (例えば磁気テープへの書き込みの場合など)、装置は操作を終結し、チャンネル終了状況を提示します。

装置は、データをブロック化することにより、操作の存続時間とチャンネル終了のタイミングを制御します。ブロックが定義されているある種の操作 (例えば磁気テープの読み取りなど) では、装置は、データ転送の終結を指示する信号を受けているかどうかに関係なく、ブロックの終わりに達するまでチャンネル終了状況を提示しません。

データ・アドレスの妥当性検査が行われるのは、データが主記憶機構へ (またはそこから) 伝送されるときだけです。CCW 内の初期データ・アドレスが無効である場合は、入出力操作ではデータは転送されず、最初のサービス要求に対する応答として、操作の終結を指示する信号が装置に送られます。書き込みでは、磁気テープ装置などの入出力装置は、機械的動作が開始される前にデータの最初のバイトを要求し、その最初のデータ・アドレスが無効な場合は、チャンネル・サブシステムは、記録メディアが先へ進まないうちに操作を中止します。しかし、入出力装置ではすでに操作が開始されているので、装置はチャンネル終了状況を提示し、その結果チャンネル・サブシステムは 1 次割り込み条件を認識します。続いて、装置は装置終了状況も提示し、その結果チャンネル・サブシステムは 2 次割り込み条件を認識します。データが転送されないときに、装置でブロックが 1 つ進められるかどうかは、装置のタイプによって異なります。

コマンド・チェーニングが行われる場合は、装置で最初の入出力操作が開始されてから、装置がチェーンの最後の入出力操作についてチャンネル終了状況を提示するまで、サブチャンネルはサブチャンネル/装置アクティブ状態になっています。そして、装置がチェーンの最後の入出力操作について装置終了状況を提示するまで、サブチャンネルは装置アクティブ状態のままになっています。

何らかの異常条件が生じると、コマンド・チェーニングは抑止され、1 次割り込み条件が生成されます。異常条件は、チャンネル・サブシステムにより検出されることも

装置により検出されることもあり、装置は、チャンネル終了、制御装置終了、または装置終了により異常条件を示すことができます。チャンネル・サブシステムが、操作に関するチャンネル終了状況を受理する前に異常条件を認識したときは、その条件が生じた操作がチェーンの最後の操作であるものとして、チェーンが終了します。装置終了状況は、チャンネル終了状況と一緒に提示される場合も単独で提示される場合も、2次割り込み条件として認識されます。装置が、制御装置終了または装置終了とともに装置チェックまたは装置例外を提示し、その結果チャンネル・サブシステムが1次割り込み条件を認識することになる場合は、サブチャンネルのサブチャンネル/装置アクティブ状態は終了し、サブチャンネルは、1次状況、2次状況、および警報状況を伴う状況保留状態にされます。初期状況割り込み条件またはPCI条件について、すでにサブチャンネルに中間割り込み条件が存在しており、その条件がサブチャンネルでまだ保留状態にある場合は、中間状況も示されることがあります。以前に、コマンド・チェーンの信号が出されたときに、チャンネル・サブシステムに提示されているチャンネル終了状況は、プログラムには提示されません。

チャンネル・パス・リセット機能

RESET CHANNEL PATH が実行されると、続いてチャンネル・パス・リセット機能が実行されます。この機能が実行されると、(1) 指定したチャンネル・パスでリセット信号が出され、その結果、(2) チャンネル報告が保留にされます (これはチャンネル・パス・リセット機能の完了を示します)。

チャンネル・パス・リセット機能の信号発信

チャンネル・サブシステムは、指定されたチャンネル・パスでリセット信号を発信します。この操作の一環として、以下のアクションがとられます。

1. 指定されたチャンネル・パスについての制御装置使用中、入出力装置使用中、および従属関係条件に関連したすべての内部標識が、リセットされます。指定されたチャンネル・パスにアクセスできるすべてのサブチャンネルで、これらの標識がリセットされます。リセット機能は、入出力操作が進行中のサブチャンネルも含めて、これ以外の影響をサブチャンネルに与えることはありません。

2. チャンネル・パスがリセット信号に対して正しく応答できない場合 (詳しくは、17-10ページの『入出力システム・リセット』を参照)、または、誤動作が原因でリセット信号を発信できない場合は、該当する各サブチャンネルでチャンネル・パスが物理的に使用不可能になります。
3. チャンネル・パスでリセット信号が受信されたときに、入出力装置で入出力操作が進行中であり、その装置がその入出力操作を実行するためにそのチャンネル・パスとアクティブに通信している場合は、その入出力操作はリセットされ、制御装置と入出力装置は、チャンネル・サブシステムとの間の現在の通信をただちに中止します。(非送信請求の装置終了状況が誤って解釈されるのを避けるために、『プログラミング上の注意 2 (15-43ページ)』で述べるプログラミング手法を使用することができます。)
4. リセット信号が受信されたときに、入出力装置において入出力操作がマルチパス・モードで進行中であり、その装置がその入出力操作の実行のために現在チャンネル・パスと通信していない場合は、入出力操作がリセットされるかどうかは、同じマルチパス・グループ内に、その装置用として選択可能な別のチャンネル・パスがあるかどうかによって決まります。マルチパス・グループ内に、その装置用として選択可能な別のチャンネル・パスが1つでもあれば、入出力操作はリセットされません。ただし、システム・リセットが受信されたチャンネル・パスは、マルチパス・グループを形成するチャンネル・パスの現行セットから除去されます。リセット信号が受信されたチャンネル・パスが、マルチパス・グループの唯一のチャンネル・パスである場合、または、装置がシングル・パス・モードで稼働している場合は、入出力操作はリセットされます。
5. チャンネル・パス・リセット機能を実行すると、入出力装置では、上記に述べたように入出力操作が中止されます。しかし、サブチャンネルでは、チャンネル・パス・リセット機能により入出力操作が中止されることはありません。

サブチャンネルで入出力操作が進行中であり、チャンネル・パス・リセット機能の実行の対象として指定されているチャンネル・パスがその入出力操作のために使用されている場合は、サブチャンネルは、その時点までの入出力操作の進行状況を正確に反映していることもあり、そうでないこともあります。サブチャンネルは、プログラムまたは装置による何らかのアクションの結果として状態が変化するまでは、チャンネル・パス・リセット機能の実行時点での状態のままになります。

チャンネル・パス・リセット機能完了信号の発信

チャンネル・パス・リセット機能が完了するのは、リセット信号が出されるか、リセット信号を出す試みが行われた後、または、リセット信号を出すことができないと判断された後です。(17-9ページの『リセット信号』を参照してください。)

チャンネル・パス・リセット機能が実行されると、その結果を報告するためのチャンネル報告が保留にされます(17-17ページの『チャンネル・サブシステム・リカバリ』を参照)。チャンネル・パスがシステム・リセット信号に正しく応答した場合は、チャンネル報告には、チャンネル・パスが初期化され、物理的に使用可能になっていることが示されます。リセット信号が発信されたが、チャンネル・パスが正しく応答できないか、または、チャンネル・パスにアクセスできる各サブチャンネルにおいてすでにチャンネル・パスが物理的に使用可能でなくなっている場合は、チャンネル報告には、チャンネル・パスが初期化されているが、物理的に使用可能ではないことが示されます。誤動作のため、または指定されたチャンネル・パスが構成内に存在しないために、リセット信号を発信できない場合は、チャンネル報告には、チャンネル・パスが初期化されておらず、物理的に使用可能になっていないことが示されます。

プログラミング上の注意:

1. マルチパス・グループのチャンネル・パスでチャンネル・パス・リセット機能が実行されたときに、入出力操作がマルチパス・モードで進行中である場合は、その入出力操作は、グループ内の別のチャンネル・パスで続行できることがあります。
2. チャンネル・パス・リセット機能の実行により、入出力装置で入出力操作がリセットされるときに、装置が提示する非請求の装置終了状況(存在する場合)が、チャンネル・サブシステムによりチェーニング状況と誤って解釈されることがあり、その結果チャンネル・サブシステムがコマンドのチェーンを続行することがあります。このような状況が生じた場合は、装置終了状況はプログラムに提示されないため、チャンネル・サブシステムは再度その装置を選択します。しかし、装置は、開始シーケンスを、コマンド・チェーニングではなく新しいチャンネル・プログラムの始めと解釈することがあります。このような事態を回避するには、**RESET CHANNEL PATH**を発行する前に、影響を受けるサブチャンネルを指定して **CLEAR SUBCHANNEL** または **HALT SUBCHANNEL** を発行します。
3. モデルによっては、チャンネル・パス・リセット機能を実行した結果、他のチャンネル・パスでオーバーランが起きることがあります。
4. 指定したチャンネル・パス上にリセット信号が送られても、制御装置の誤動作または制御装置への物理チャンネル・パスの誤動作が原因で、1 つまたは複数の装置からそのチャンネル・パスへの従属関係がリセットされていないことがあります。

第16章 入出力割り込み

割り込み条件	16-2	CCW アドレス・フィールド	16-18
中間割り込み条件	16-4	装置状況フィールド	16-22
1 次割り込み条件	16-4	サブチャンネル状況フィールド	16-23
2 次割り込み条件	16-4	プログラム制御割り込み	16-23
警報割り込み条件	16-4	誤長	16-23
割り込みの優先順位	16-4	プログラム・チェック	16-23
割り込みアクション	16-5	記憶保護チェック	16-26
割り込み応答ブロック	16-6	チャンネル・データ・チェック	16-26
サブチャンネル状況ワード	16-6	チャンネル制御チェック	16-27
サブチャンネル・キー	16-7	インターフェース制御チェック	16-27
延期制御 (S)	16-8	チェーニング・チェック	16-28
拡張状況ワード・フォーマット (L)	16-8	カウント・フィールド	16-28
据え置き条件コード (CC)	16-8	拡張状況ワード	16-31
フォーマット (F)	16-10	拡張状況フォーマット 0	16-31
事前取り出し (P)	16-10	サブチャンネル・ログアウト	
初期状況割り込み制御 (I)	16-11	(Subchannel Logout)	16-31
アドレス限界検査制御 (A)	16-11	拡張報告ワード (Extended-Report Word)	16-35
延期割り込み抑止 (U)	16-11	故障記憶機構アドレス	
サブチャンネル制御フィールド	16-11	(Failing-Storage Address)	16-36
0 条件コード (Z)	16-11	2 次 CCW アドレス	
拡張制御 (E)	16-11	(Secundary-CCW Address)	16-36
パス非稼働 (N)	16-11	拡張状況フォーマット 1	16-37
機能制御 (FC)	16-12	拡張状況フォーマット 2	16-37
活動制御 (AC)	16-13	拡張状況フォーマット 3	16-38
状況制御 (SC)	16-16	拡張制御ワード	16-39

START SUBCHANNEL の実行により開始された 1 つまたは一連の入出力操作が終了すると、チャンネル・サブシステムおよび入出力装置は状況条件を生成します。入出力割り込みを使用するか、または TEST PENDING INTERRUPTION 命令を実行することにより、これらの条件が生成されたことをプログラムに通知できます。(ある種の異常な状況下では、マシン・チェック割り込みによって、これらの条件がプログラムに通知されることがあります。詳細については、17-17ページの『チャンネル・サブシステム・リカバリー』を参照してください。) 状況条件は、アドレスや、操作シーケンスの範囲を示すカウントと同様に、サブチャンネル状況ワード (SCSW) の形でプログラムに提示されます。SCSW は、TEST SUBCHANNEL の実行時に、割り込み応答ブロック (IRB) に格納されます。

通常、入出力操作は、入出力装置が 1 次割り込み状況を示す信号を出すまで実行されます。1 次割り込み状況の

信号は、入出力操作の開始時に出されることも、それより後で出されることもあります。チャンネル・サブシステムは、装置の誤動作、プログラム・チェック、チェーニング・チェック、記憶保護チェック、または誤長条件を検出したときに、消去機能または停止機能を実行して、入出力操作を中止することができます。また、CLEAR SUBCHANNEL、HALT SUBCHANNEL、または RESET CHANNEL PATH の結果として、それぞれ消去、停止、またはチャンネル・パス・リセット機能が実行されて、入出力操作が中止されることもあります。

CPU は、入出力割り込みを使用して、入出力装置またはサブチャンネルで発生した条件に合わせて CPU 自体の状態を変化させることができます。これらの条件を引き起こすものには、プログラム、チャンネル・サブシステム、および入出力装置での外部イベントがあります。

割り込み条件

入出力割り込み要求を開始させる条件を、入出力割り込み条件と呼びます。チャンネル・サブシステムが割り込み条件を認識すると、該当のサブチャンネルでその条件が示されます。このときのサブチャンネルは、状況保留状態にあると言われます。サブチャンネルが状況保留状態になると、チャンネル・サブシステムは入出力割り込み要求を生成します。入出力割り込み要求は、プログラムに 1 回だけ通知されます。

入出力割り込み要求は、構成内のいずれかの CPU により受理されるか、チャンネル・サブシステムにより撤回されるか、TEST PENDING INTERRUPTION、TEST SUBCHANNEL、または CLEAR SUBCHANNEL の実行により消去されるか、またはサブシステム・リセットにより消去されるまで、保留状態のままになっています。CPU が割り込み要求を受理し、関連の割り込みコードを格納すると、割り込み要求は消去されます。代わりに、TEST PENDING INTERRUPTION を実行することにより入出力割り込み要求を消去することもできます。いずれの場合も、サブチャンネルは、TEST SUBCHANNEL または CLEAR SUBCHANNEL が実行されるか、あるいはサブチャンネルがリセットされて、関連の割り込み条件が消去されるまで、状況保留状態のままになっています。

入出力割り込み条件は、通常、TEST SUBCHANNEL を実行することにより消去されます。入出力割り込み要求が保留されているサブチャンネルを指定して TEST SUBCHANNEL を実行すると、そのサブチャンネルでは、割り込み要求と割り込み条件の両方が消去されます。また、CLEAR SUBCHANNEL により割り込み要求および割り込み条件を消去することもできます。

入出力装置により生成され、開始機能の最後の入出力操作の終結後に提示される装置終了状況が、サブチャンネルでチャンネル・サブシステムによりリセットされたときに、そのサブチャンネルが現在開始保留状態にあり、かつ、その装置終了が、装置終了のみかまたは制御装置終了を伴う装置終了である場合は、入出力割り込み条件または入出力割り込み要求は生成されません。装置終了状況ビットにその他の状況ビットが付随している場合は、チャンネル・サブシステムは、入出力割り込み要求を生成し、据え置き条件コード 1 を示します。

コマンド開始シーケンスにおいてチャンネル・サブシステムが検出した異常条件が原因で入出力操作が中止された場合は、割り込み条件を表す状況がサブチャンネルに置かれ、サブチャンネルは状況保留になります。入出力装置が異常条件を検出した場合は、関連の SCSW の装置状況フィールドにその条件が示されます。

コマンド・チューニングが行われているときは、装置が状況を生成しても割り込みは起きず、プログラムにはその状況は通知されません。

チャンネル・サブシステムは、以下のいずれかの割り込み条件を検出すると、必ずしも入出力装置と通信していても、またその装置から状況バイトを受け取っていても、入出力割り込み要求を開始します。

- 前の START SUBCHANNEL の実行によりサブチャンネルに渡された ORB の内容に関連したプログラミング・エラー
- START SUBCHANNEL または RESUME SUBCHANNEL でチャンネル・プログラムの実行を開始するために取り出された CCW の延期フラグが 1 に設定されており、ORB では延期割り込み抑止が指定されていない。
- 最初の CCW または最初の IDAW に関連したプログラミング・エラー

延期条件を除き、サブチャンネルからのこれらの割り込み条件には、他のサブチャンネル状況標識が付随していることがあります。装置状況標識にはすべて 0 が格納されます。

使用不可にされているサブチャンネルに、装置チェックを含む状況が提示された場合、または装置がどのサブチャンネルにも関連付けられていない場合は、チャンネル・サブシステムはその装置に対して消去信号を出します。しかし、提示された状況に装置チェックが含まれていない場合は、チャンネル・サブシステムはその状況を受け入れ、サブチャンネルを状況保留にすることなく、その状況を破棄します。

入出力装置が原因で生じる割り込み条件には、複数の装置状況条件が付随していることがあります。さらに、チャンネル・サブシステムは、入出力割り込みの介入なしに、同じ装置に関連した複数の割り込み条件を受理することがあります。例えば、装置終了までに装置でチャンネル終了条件が消去されないときは、装置で両方の条件が同時に消去され、SCSW の中に一緒に示されることがあります。また、サブチャンネルですでにチャンネル終了状況が受理されていることや、入出力割り込みが起きていることもあります。しかし、サブチャンネルで装置終了状況が受理されるまでは、TEST SUBCHANNEL により関連の状況保留条件が消去されない場合があります。このような場合は、装置終了状況とチャンネル終了状況が組み合わせられて、追加の入出力割り込みは生じないことがあります。サブチャンネルにおいて、割り込み条件が他の既存の割り込み条件と組み合わせられるかどうかは、その割り込み条件が非請求型か請求型かによって決まります。

非請求割り込み条件: 非請求割り込み条件は、消去、停止、再開、または開始機能の実行に関係のないすべての割り込み条件です。非請求割り込み条件は、サブチャネルでは警報状況として識別されます。非請求割り込み条件が生成されるのは、サブチャネルが装置アクティブのときのみです。

非請求割り込み条件に関連したサブチャネル状況および装置状況は、既存の割り込み条件に関連した状況と組み合わせられることはありません。サブチャネルが現在、状況保留状態にある場合は、その状況保留条件が消去されるまでは、非請求割り込み条件はチャンネル・サブシステムまたは装置のいずれか該当する方で休止状態になりま

す。サブチャネルがアイドル状態にあり、装置が 0 状況を提示しているときは、その状況は破棄されます。

請求割り込み条件: 請求割り込み条件は、消去、停止、再開、または開始機能を実行または実行しようとした結果として、直接生成されるすべての割り込み条件です。請求割り込み条件には、サブチャネルがサブチャネル/装置アクティブまたは装置アクティブにあるときに生成される割り込み条件がすべて含まれます。請求割り込み条件に関連したサブチャネル状況および装置状況は、他の既存の請求割り込み条件と組み合わせられることがあります。図16-1 は、SCSW の状況制御フィールド内のビットの各組み合わせの結果として生じる割り込み条件を示しています。

状況制御フィールド	状況制御ビットの組み合わせ															
警報	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1 次	0	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0
2 次	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0
中間	0	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0
状況保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
結果の割り込み条件	E	S	S	S	S	S	-	S	S	S	S	S	S	-	S	S
説明:																
- この組み合わせは生じません。																
E 非請求または請求割り込み条件。																
S 請求割り込み条件。																
0 ビットに 0 が格納されることを示します。																
1 ビットに 1 が格納されることを示します。																

図 16-1. 状況制御ビットの組み合わせ別の割り込み条件

中間割り込み条件

中間割り込み条件は、前にプログラムから通知要求が出されているイベントが起きたことを示す請求割り込み条件です。中間割り込み条件は、請求サブチャンネル状況、Z ビット、サブチャンネル延期条件、またはこれら 3 つのいずれかの組み合わせで表されます。中間割り込み条件が起きるのは、プログラムが ORB または CCW のフラグを使用してその条件を要求していた場合のみです。中間状況ビットの設定に関係なく、サブチャンネルの状況に応じて、入出力操作の実行または延期が続けられます。

中間割り込み条件は、必ず次の標識の 1 つと一緒に示されます。

1. サブチャンネル・アクティブ
2. 1 次状況のみをとまなう状況保留
3. 警報状況または 2 次状況またはその両方が付随する 1 次状況をとまなう状況保留
4. 延期

TEST SUBCHANNEL の実行時に、状況制御フィールドの中間状況ビットと状況保留ビットのみが 1 である場合は、装置状況フィールドは 0 です。

1 次割り込み条件

1 次割り込み条件は、サブチャンネルで開始機能の実行が完了したことを示す請求割り込み条件です。1 次割り込み条件は、サブチャンネルが 1 次状況を伴う状況保留状態にあるときに、TEST SUBCHANNEL の実行結果として格納される SCSW で表されます。サブチャンネルで 1 次割り込み条件が示されたときは、チャンネル・サブシステムは、もはや、コマンドまたはデータの転送により入出力操作にアクティブに関与している状態にはありません。サブチャンネルが 1 次割り込み条件を伴う状況保留状態にあるときは、HALT SUBCHANNEL、MODIFY SUBCHANNEL、RESUME SUBCHANNEL、および START SUBCHANNEL 命令のいずれかを実行すると、0 以外の条件コードが設定されます。TEST SUBCHANNEL の実行により 1 次割り込み条件が消去されると、サブチャンネルは START SUBCHANNEL 命令を受理します。(14-14 ページの

『START SUBCHANNEL』を参照してください。)

2 次割り込み条件

2 次割り込み条件は、通常、入出力装置での入出力操作の完了を示す請求割り込み条件です。また、入出力装置で最初の入出力操作を開始する前に請求警報割り込み条件が認識されたために、開始機能が中止された場合にも、チャンネル・サブシステムにより 2 次割り込み条件が生成されることがあります。2 次割り込み条件は、サブチャンネルが 2 次状況を伴う状況保留状態にあるときに、TEST SUBCHANNEL の実行結果として格納される SCSW で表されます。チャンネル・サブシステムが、2 次割り込み条件が認識される原因となった装置からの状況を受理すると、装置での開始機能は完了します。

警報割り込み条件

警報割り込み条件は、停止、再開、または開始機能で異常条件が発生したことを示す請求割り込み条件か、または、停止、再開、または開始機能の実行とは無関係の条件を表す非請求割り込み条件です。警報割り込み条件は、サブチャンネルが警報状況を伴う状況保留状態にあるときに、TEST SUBCHANNEL の実行結果として格納される SCSW で表されます。警報割り込み条件は、チャンネル・サブシステムまたは入出力装置により生成されます。0 以外の警報状況は、必ずプログラムに通知されます。

割り込みの優先順位

すべての入出力割り込み要求は、どの CPU 内の活動とも非同期に行われ、同時に複数のサブチャンネルに関連した割り込み要求が存在することができます。割り込みの優先順位は、2 種類のメカニズムにより制御されます。その 1 つは、チャンネル・サブシステム内で、同じ入出力割り込みサブクラスに関連したサブチャンネルからの割り込み要求間での優先順位を決定し、もう 1 つは、特定の CPU 内で、異なる入出力割り込みサブクラスに関連したサブチャンネルからの要求間での優先順位を決定します。チャンネル・サブシステムが入出力割り込みを要求するのは、各サブチャンネルからの要求間の優先順位を決定した後です。要求の原因となる条件は、CPU での TEST SUBCHANNEL または CLEAR SUBCHANNEL の実行により消去されるか、または入出力システム・リセットが行われるまで、サブチャンネルに保持されています。

同じ入出力割り込みサブクラスのサブチャンネルからの割り込み要求間の優先順位は、チャンネル・サブシステムが割り込みの必要性を認識した順序で割り当てられます。チャンネル・サブシステムが必要性を認識する順序は、割

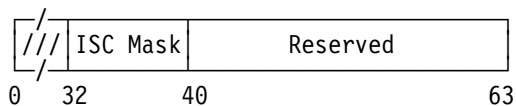
り込み条件のタイプとチャンネル・パスのタイプによって決まります。

入出力サブクラスの異なるサブチャンネルからの割り込み要求間の優先順位は、入出力割り込みサブクラス・コードの数値 (0 が最高優先順位) と、制御レジスタ 6 中の入出力割り込みサブクラス・マスクに従って、CPU が割り当てます。入出力割り込みサブクラス・コードの数値は、CPU の制御レジスタ 6 中の入出力割り込みサブクラス・マスクのビット位置に、直接対応しています。ある CPU において入出力割り込みサブクラス・マスク・ビットが 0 である場合は、マスク・レジスタ内の関連の位置に等しい数値の入出力割り込みサブクラスを持つすべてのサブチャンネルは、その CPU 内で隠されていると言われます。したがって、CPU は、その CPU の制御レジスタ 6 中の対応するビットにより隠されていない最小番号の入出力割り込みサブクラス・コードを持つサブチャンネルから、最高優先順位の入出力割り込み要求を受理することになります。CPU が最高優先順位の割り込み要求を受理すると、その要求は、構成内の他の CPU が受理しないようにするために消去されます。

割り込み処理の優先順位は、TEST SUBCHANNEL または CLEAR SUBCHANNEL を実行して変更することができます。これらの命令のいずれかを実行したときに、指定したサブチャンネルで割り込み要求が保留状態にある場合は、その割り込み要求は、事前に設定されている優先順位に関係なく消去されます。残りの割り込み要求の相対優先順位は変わりません。

プログラミング上の注意:

1. 入出力割り込みサブクラス・マスクは制御レジスタ 6 中にあり、そのフォーマットは以下のとおりです。



2. 初期 CPU リセット時には、制御レジスタ 6 はすべて 0 に設定されます。

割り込みアクション

入出力割り込みが起きるのは、該当のサブチャンネルに関連した入出力割り込みサブクラス・マスク・ビットが 1 で、CPU が入出力割り込み可能になっている場合に限られます。

割り込みが起きるのは、1 つの操作単位が完了した時点です (5-20 ページの『割り込み点』を参照)。CPU が入出力割り込み不可にされているときに、チャンネル・サブシステムがサブチャンネルからの要求間の優先順位を決定する場合、サブチャンネルに関連した入出力割り込みサブクラス・マスク・ビットが 1 であれば、CPU を使用可能にする命令の実行後ただちに (そして次の命令が実行される前に)、割り込みが起きます。また、状況保留中の各サブチャンネルについて入出力割り込みサブクラス・マスク・ビットが 0 のときに、チャンネル・サブシステムがサブチャンネルからの割り込み要求間の優先順位を決定する場合は、CPU が入出力割り込み可能にされていれば、入出力割り込みサブクラス・マスク・ビットの少なくとも 1 つを設定する命令が完了した後で、ただちに割り込みが起きます。この割り込みは、CPU が決定した最高優先順位の入出力割り込み要求に対応しています。

割り込みが許可されたときまでに、チャンネル・サブシステムがサブチャンネルからの割り込み要求間の優先順位を決定していない場合は、必ずしも、CPU を使用可能にする命令の完了直後に割り込みが起きるとは限りません。割り込み条件がサブチャンネルに存在していた時間に関係なく、遅延が生じることがあります。

割り込みが起きると、現 PSW が入出力旧 PSW として実記憶位置 368-383 に格納され、割り込みに関連した入出力割り込みコードが CPU の実記憶位置 184-195 に格納されて、割り込みが可能になります。次に、新しい入出力 PSW が実記憶位置 496-511 からロードされ、その PSW に示されている CPU 状態で処理が再開されます。割り込みの原因となっているサブチャンネルは、割り込みコードにより識別されます。

入出力割り込みコードは、以下のフォーマットで格納されます。このコードについては、14-17 ページの

『TEST PENDING INTERRUPTION』で説明されています。

16 進数 10 進数

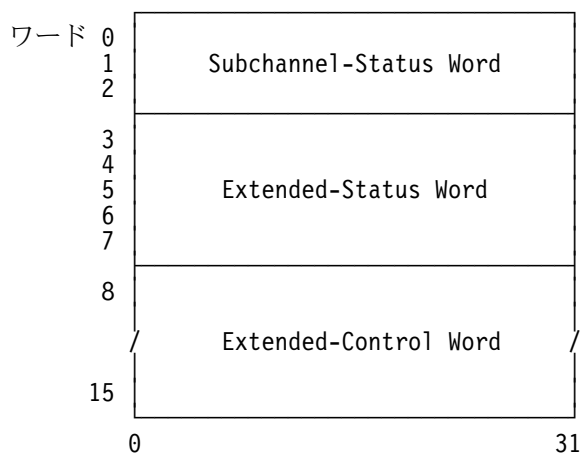
B8	184	Subsystem-Identification Word
BC	188	I/O-Interruption Parameter
C0	192	I/O-Interruption-Identification Word
	0	31

プログラミング上の注意: 入出力システム・リセット時には、すべてのサブチャンネルについて入出力割り込みサブクラス・コードが 0 に設定されます。このコードは、MODIFY SUBCHANNEL を実行することによ

り、0-7 の任意の値に変更できます。(この命令の働きについては、14-7ページの『MODIFY SUBCHANNEL』で説明してあります。)

割り込み応答ブロック

割り込み応答ブロック (IRB) は、TEST SUBCHANNEL のオペランドです。IRB アドレスの右端 2 ビットは 0 で、これは IRB をワード境界上に指定します。IRB の主なフィールドには、サブチャネル状況ワード、拡張状況ワード、拡張制御ワードの 3 つがあります。IRB のフォーマットは以下のとおりです。



サブチャネル状況ワードと拡張状況ワードの長さは、それぞれ 12 バイトと 20 バイトです。拡張制御ワードの長さは 32 バイトです。拡張制御ビット (SCSW のワード 0 のビット 14) が 0 のときは、割り込み応答ブロックのワード 8-15 は格納されることもあり、されないこともあります。

サブチャネル状況ワード

サブチャネル状況ワード (SCSW) は、プログラムに、サブチャネルとそれに関連した装置の状況を示します。停止、再開、または開始機能が実行された場合は、SCSW は、そのような条件下で操作が終了したかを示していることがあります。

TEST SUBCHANNEL が実行されたときに、指定したサブチャネルが稼働状態にあれば、SCSW が格納されます。SCSW が格納される場所は、TEST SUBCHANNEL のオペランドとして指定されている IRB のワード 0-2 です。STORE SUBCHANNEL が実行されると、サブチャネル情報ブロックのワード 7-9 に SCSW が格納されます (15-1ページの『サブチャネル情報ブロック』を参照)。16-7ページの図16-2 は、SCSW のフォーマットとその内容の要約を示しています。

ワード 0	Key	S	L	CC	F	P	I	A	U	Z	E	N	0	FC	AC	SC
1	CCW Address															
2	Device Status				Subch. Status				Count							
	0	4	8	16	20	27	31									

ビット 名称

ワード 0

0-3	サブチャンネル・キー
4	延期制御 (S)
5	ESW フォーマット (L)
6-7	据え置き条件コード (CC)
8	フォーマット (F)
9	事前取り出し (P)
10	初期状況割り込み制御 (I)
11	アドレス限界検査制御 (A)
12	延期割り込み抑止 (U)
13	0 条件コード (Z)
14	拡張制御 (E)
15	パス非稼働 (N)
16	予約済み
17-19	機能制御 (FC) (ビット 17 は開始機能、ビット 18 は停止機能、ビット 19 は消去機能)
20-26	活動制御 (AC) (ビット 20 は再開保留、ビット 21 は開始保留、ビット 22 は停止保留、 ビット 23 は消去保留、ビット 24 はサブチャンネル・アクティブ、 ビット 25 は装置アクティブ、ビット 26 は延期)
27-31	状況制御 (SC) (ビット 27 は警報状況、ビット 28 は中間状況、ビット 29 は 1 次状況、 ビット 30 は 2 次状況、ビット 31 は状況保留)

ワード 1

0-31 CCW アドレス

ワード 2

0-7	装置状況 (ビット 0 はアテンション、ビット 1 は状況修飾、ビット 2 は制御装置終了、 ビット 3 は使用中、ビット 4 はチャンネル終了、ビット 5 は装置終了、 ビット 6 は装置チェック、ビット 7 は装置例外)
8-15	サブチャンネル状況 (ビット 8 はプログラム制御割り込み、ビット 9 は誤長、ビット 10 はプログラム・ チェック、ビット 11 は記憶保護チェック、ビット 12 はチャンネル・データ・チェック、 ビット 13 はチャンネル制御チェック、ビット 14 はインターフェース制御チェック、 ビット 15 はチェーニング・チェック)
16-31	カウント

図 16-2. SCSW のフォーマット

サブチャンネル状況ワード (SCSW) の内容は、SCSW が格納されるときのサブチャンネルの状態によって異なります。SCSW の各フィールドには、サブチャンネルと装置の状態に応じて、(1) 最後の操作に関する情報、(2) 操作の実行には関係のない情報、(3) 0、または、(4) 意味のない値が含まれます。以下の説明は、該当の SCSW フィールドに意味のある情報が含まれる場合を想定しています。

サブチャンネル・キー

開始機能ビット (ワード 0 のビット 17) が 1 のときは、ワード 0 のビット 0-3 には、関連の開始機能の実行時に使用するアクセス・キーが入っています。これらのビットは、ORB のワード 1 のビット位置 0-3 に指定されているキーと同じです。サブチャンネル・キーが意味を持つのは、開始機能ビット (ワード 0 のビット 17) が 1 のときだけです。

延期制御 (S)

開始機能ビット (ワード 0 のビット 17) が 1 で、ワード 0 のビット 4 が 1 のときは、サブチャンネルで延期機能を開始できることを示します。ビット 4 が意味を持つのは、ビット 17 が 1 のときのみです。ビット 17 が 1 でビット 4 が 1 のとき、CCW の S フラグが 1 に設定されていることをチャンネル・サブシステムが認識すると、チャンネル・プログラムの実行を延期できます。ビット 4 が 0 のときは、チャンネル・プログラムの実行を延期することはできず、CCW 内で S フラグの 1 つが 1 に設定されていることが検出されると、プログラム・チェック条件が認識されます。

拡張状況ワード・フォーマット (L)

状況保留ビット (ワード 0 のビット 31) が 1 で、ワード 0 のビット 5 が 1 のときは、フォーマット 0 の ESW が格納されていることを示します。フォーマット 0 の ESW が格納されるのは、以下のいずれかの標識を含む割り込み条件が、TEST SUBCHANNEL により消去されたときです。

- チャンネル・データ・チェック
- チャンネル制御チェック
- インターフェース制御チェック
- 測定ブロック・プログラム・チェック
- 測定ブロック・データ・チェック
- 測定ブロック記憶保護チェック
- 要パス検査
- 許可検査

拡張状況ワード・フォーマット・ビットは、サブチャンネルが状況保留状態のときは常に意味を持ちます。TEST SUBCHANNEL または CLEAR SUBCHANNEL を実行すると、フォーマット 0 の ESW を形成するために使用される拡張状況情報が、サブチャンネルで消去されます。

据え置き条件コード (CC)

開始機能ビット (ワード 0 のビット 17) が 1 で、状況保留ビット (ワード 0 のビット 31) も 1 のときは、ワード 0 のビット 6 および 7 には、TEST SUBCHANNEL または STORE SUBCHANNEL が実行されたときにサブチャンネルが状況保留状態だったことについての一般的な理由が示されています。据え置き条件コードが意味を持つのは、サブチャンネルが何らかの組み合わせの状況を伴う状況保留状態にあり、SCSW の機能制御フィールドの開始機能ビットが 1 である場合のみです。サブチャンネルが状況保留状態にあるときの、据え置き条件コードの値の意味については、16-10ページの図 16-3 を参照してください。

据え置き条件コード (0 以外の場合) は、サブチャンネルが開始保留または延期状態にあるときに、サブチャンネルがサブチャンネル/装置アクティブになるのを妨げるような条件が検出されたかどうかを示すために使用されます。

据え置き条件コード 0: 通常の入出力割り込みが行われました。

据え置き条件コード 1: START SUBCHANNEL または RESUME SUBCHANNEL について条件コード 0 が設定された後で、関連装置が提示した状況またはチャンネル・サブシステムが生成した状況が、SCSW に示されています。SCSW の状況制御フィールドの警報状況ビットと状況保留ビットのみが 1 である場合は、提示された状況はチャンネル・プログラムの実行とは無関係です。中間状況ビット、1 次状況ビット、またはその両方が 1 である場合は、提示された状況は、最後に実行された START SUBCHANNEL 命令で指定されているチャンネル・プログラム、または最後に実行された RESUME SUBCHANNEL 命令で暗黙指定されているチャンネル・プログラムに関係しています。(15-40ページの『入出力操作の即時終結』を参照してください。) 2 次状況ビットが 1 で、1 次状況ビットが 0 の場合は、提示された状況は、START SUBCHANNEL 命令で指定されているチャンネル・プログラム、または、最後に実行された START SUBCHANNEL の前の RESUME SUBCHANNEL 命令で暗黙指定されているチャンネル・プログラムに関係しています。

据え置き条件コード 2: このコードは将来の使用に備えて予約済みであり、これが設定されることはありません。

据え置き条件コード 3: 装置選択が試みられましたが、その装置は、その装置への選択可能なすべてのチャンネル・パス上で非稼働状態にあると見なされました。

装置が非稼働状態にあると見なされるのは、チャンネル・サブシステムによる装置選択にその装置が応答しない場合です。この状況が生じるのは、システム内に制御装置が提供されていないか、制御装置の電源がオフになっているか、また、制御装置でチャンネル・パスが論理的にオフにされているときです。制御装置が提供されていて、装置の接続が可能であっても、選択対象の装置がインストールされておらず、制御装置がその装置を接続されている装置の 1 つとして認識できるように設計されていない場合も、非稼働状態が示されます。(13-4ページの『入出力アドレッシング』も参照してください。)

チャンネル・サブシステムは、装置への選択可能なチャンネル・パスがない場合も、据え置き条件コード 3 を設定す

ることがあります。(16-10ページの図16-3を参照してください。)

プログラミング上の注意:

1. 開始機能の実行時に、選択しようとしている入出力装置がインストールされていないか、または制御装置から論理的に除去されているが、その関連制御装置が稼働状態にあり、その入出力装置が自身に接続されている装置の1つとして選択されていると認識している場合は、制御装置のモデルによっては、その入出力装置のアドレスを認識できないか、またはその入出力装置が作動不能であると見なすことがあります。前者の場合は、パス稼働マスクの設定に従って、パス非稼働条件が認識されます。(15-6ページの『パス稼働マスク (POM)』を参照してください。) 後者の場合は、制御装置が選択に応答すると作動不能条件が示され、作動不能状態のために入出力装置での操作の正常な開始が妨げられているときは常に、装置チェックが示されます。この場合は、

SCSW に装置チェック状況が示され、サブチャネルは、1次状況、2次状況、および警報状況を伴う状況保留になり、据え置き条件コード1が示されま

す。(装置チェック状況の説明については、

「*ESA/390 Common I/O-Device Commands*」(SA22-7204)を参照してください。)この条件がどのように示されるかについては、該当の制御装置に関するシステム・ライブラリー資料を参照してください。

2. HALT SUBCHANNEL が実行されたときに、指定したサブチャネルが延期されていて、中間状況を伴う状況保留状態にある場合は、据え置き条件コード1が設定され、状況制御フィールドの状況保留ビットおよび中間状況ビットか、または状況保留ビット、中間状況ビット、および警報状況ビットが、1に設定されます。警報状況ビットが1である場合は、装置に対して停止信号を出そうとした結果として、サブチャネル・ログアウト情報が生成されます。

ビット 6	ビット 7	状況制御 ¹	意味
0	0	A I P S X A I P - X A - P S X A - P - X - I P S X - I P - X - I - - X - - P S X - - P - X	通常の入出力割り込み
0	1	A I P S X A I P - X A I - - X ² A - P S X A - P - X A - - S X A - - - X - I P S X - I P - X - I - - X ² - - P S X - - P - X - - - S X ³ - - - - X ^{3 2}	チェーニングが指定されていない即時操作で正常に終了したか、または、入出力装置において、何らかの状況条件の設定が原因で、要求された入出力操作の開始または再開が防げられた。
1	0	予約済み	予約済み
1	1	- - P S X - I P S X	入出力装置が、使用可能などのパス上でも操作可能になっていない。あるいは、専用従属関係条件が存在する場合は、その従属関係が適用されるパス上で装置が操作可能になっていない。
説明: <ul style="list-style-type: none"> - このビットは 0 です。 ¹ 機能制御フィールド内の開始機能ビットが 1 のときに許される状況制御ビット設定の組み合わせ。 ² この条件は、HALT SUBCHANNEL の実行後に、サブチャネルが現在延期されているときに検出されます。 ³ この条件は、HALT SUBCHANNEL の実行後に、サブチャネルが現在開始保留状態にあるときに検出されます。 <p>A 警報状況。 I 中間状況。 P 1 次状況。 S 2 次状況。 X 状況保留。</p>			

図 16-3. 状況保留状態のサブチャネルに関する据え置き条件コードの意味

フォーマット (F)

開始機能ビット (ワード 0 のビット 17) が 1 のときは、ワード 0 のビット 8 は、入出力操作に関連した CCW のフォーマットを示します。フォーマット・ビットが意味を持つのは、ビット 17 が 1 のときのみです。ワード 1 のビット 8 が 0 のときは、フォーマット 0 の CCW が指定されます。このビットが 1 のときは、フォーマット 1 の CCW が指定されます。(2 つの CCW フォーマットの説明については、15-25 ページの『チャンネル・コマンド・ワード』を参照してください。)

事前取り出し (P)

開始機能ビット (ワード 0 のビット 17) が 1 のときは、ワード 0 のビット 9 は、CCW、IDAW、および関連データの無制限の事前取り出しが許されるかどうかを示します。事前取り出しビットが意味を持つのは、ビット 17 が 1 のときのみです。ビット 9 が 0 のときは、出力データ・チェーニング操作では、1 つのデータ域を記述する 1 つの CCW の事前取り出しができますが、その他の操作では、事前取り出しは許されません。ビット 9 が 1 のときは、CCW、IDAW、および関連データの無制限の事前取り出しが許されます。チャンネル・プログラムを形成する一部またはすべての CCW、IDAW、

および関連データが、実際に事前取り出しされるかどうかは、モデルによって異なります。

初期状況割り込み制御 (I)

開始機能ビット (ワード 0 のビット 17) が 1 で、ワード 0 のビット 10 が 1 のときは、サブチャンネルがサブチャンネル・アクティブになった場合に、チャンネル・サブシステムが中間割り込み条件を生成することを指定します (15-22ページの『初期状況割り込み制御 (I)』を参照)。ワード 0 のビット 10 が 0 のときは、サブチャンネルがサブチャンネル・アクティブになっても、中間割り込み条件を生成しないことを示します。

プログラムは、ORB を使用して中間割り込み条件を要求します。その要求の結果として起きる入出力割り込みは、開始機能または再開機能を実行しているチャンネル・サブシステムが原因で起きることがあります。(TEST SUBCHANNEL により中間割り込み条件が消去されたときにチャンネル・サブシステムが提示する表示についての詳細は、『0 条件コード (Z)』を参照してください。)

アドレス限界検査制御 (A)

開始機能ビット (ワード 0 のビット 17) が 1 で、割り込み 0 のビット 11 が 1 のときは、プログラムがチャンネル・サブシステムに、サブチャンネルでの限界モードの設定に従ってアドレス限界検査を行うよう要求したことを示します (15-23ページの『アドレス限界検査制御 (A)』を参照)。アドレス限界検査制御ビットが意味を持つのは、ビット 17 が 1 のときのみです。

延期割り込み抑止 (U)

開始機能ビット (ワード 0 のビット 17) が 1 で、ワード 0 のビット 12 が 1 のときは、プログラムがチャンネル・サブシステムに、サブチャンネルが延期されているときにサブチャンネル延期割り込み条件の生成を抑止するよう要求したことを示します (15-23ページの『延期割り込み抑止制御 (U)』を参照)。ビット 12 が 0 のときは、チャンネル・サブシステムは、関連のチャンネル・プログラムの実行中にサブチャンネルが延期されるたびに、中間割り込み条件を生成します。延期割り込み抑止ビットが意味を持つのは、ビット 17 が 1 のときのみです。

サブチャンネル制御フィールド

サブチャンネル制御情報に関する以下の説明は、SCSW のサブチャンネル制御フィールド (ワード 0 のビット 13-31) に適用されます。

0 条件コード (Z)

ワード 0 のビット 13 が 1 のときは、サブチャンネルがサブチャンネル・アクティブになっていること、および、チャンネル・サブシステムがサブチャンネルで初期状況割り込み条件を認識していることを示します。Z ビットが意味を持つのは、中間状況ビット (ワード 0 のビット 28) と、開始機能ビット (ワード 0 のビット 17) が、両方とも 1 の場合のみです。

START SUBCHANNEL が実行されるときに、初期状況割り込み制御ビット (ORB のワード 1 のビット 10) が 1 であるときに、サブチャンネルがサブチャンネル・アクティブになると、サブチャンネルは、初期状況割り込み条件を示す中間状況を伴う状況保留にされます。TEST SUBCHANNEL または CLEAR SUBCHANNEL の実行により中間割り込み条件が消去されるまでは、サブチャンネルには初期状況割り込み条件が残されています。START SUBCHANNEL の実行時に ORB の初期状況割り込み制御ビットが 0 である場合は、サブチャンネルがサブチャンネル・アクティブになっても、中間割り込み条件は生成されず、初期状況割り込み条件は認識されません。

拡張制御 (E)

ワード 0 のビット 14 が 1 のときは、モデル依存情報または並行センス情報が拡張制御ワード (ECW) に格納されます。ビット 14 が 0 のときは、ECW のワード 0-7 が格納される場合、その内容は予測不能です。E ビットが意味を持つのは、サブチャンネルが、警報状況のみを伴うか、または 1 次状況、2 次状況、またはその両方が付随した警報状況を伴う状況保留状態にあるときです。

プログラミング上の注意: TEST SUBCHANNEL の実行時に行われる ECW のワード 0-7 の格納は、上記で述べたビット 14 の設定値に従ったモデル依存の機能です。したがって、プログラムでは、常に、64 バイトの IRB を収容するための十分な記憶域を用意しておく必要があります。

パス非稼働 (N)

ワード 0 のビット 15 が 1 のときは、チャンネル・サブシステムが N 条件を認識したことを示します。N 条件は、パス非稼働条件が 1 つ以上認識されていることを示します。チャンネル・サブシステムがパス非稼働条件を認識するのは、消去、停止、再開、または開始機能を実行するために装置を選択しようとしたときに、サブチャンネルで使用できるチャンネル・パス上で、サブチャンネルに関連した装置が稼働状態にないと認められた場合です。チャンネル・パスがサブチャンネルにとって稼働状態にあるのは、チャンネル・サブシステムが、消去、停止、再開、ま

たは開始機能を実行するために最後に行った装置選択の際に、そのチャンネル・パス上で関連装置が稼働状態であると認められたときです。チャンネル・パスがサブチャンネルにとって非稼働状態にあるのは、チャンネル・サブシステムが、消去、停止、再開、または開始機能を実行するために最後に行った装置選択の際に、そのチャンネル・パス上で関連装置が非稼働状態であると認められたときです。チャンネル・パス上で装置が稼働状態であると認められるのは、その装置が装置選択に応答した場合です。

N ビットが意味を持つのは、状況制御フィールドに以下のいずれかの表示が含まれており、かつサブチャンネルで基本入出力機能が少なくとも 1 つは示されているときです。

- 1 次状況、2 次状況、または警報状況のいずれかの組み合わせを伴う状況保留
- 状況保留のみ
- サブチャンネルが延期もされているときの、中間状況を伴う状況保留

N 条件がリセットされるのは、TEST SUBCHANNEL の実行結果として条件コード 0 が設定され、上記の記述に従って N ビットが意味を持っているときです。

注:

1. パス非稼働条件は、チャンネル・パスに誤動作が生じていることを意味するわけではありません。チャンネル・パスに誤動作が生じているときは、インターフェース制御チェックなどのエラー表示が生成されません。
2. パス非稼働条件が認識されており、その後サブチャンネルが中間状況のみを伴う状況保留状態になったときは、(a) サブチャンネルが 1 次状況と伴う状況保留になるかまたは延期状態になるまで、パス非活動条件は引き続き認識され、そして、(b) TEST SUBCHANNEL の実行時に、パス非稼働条件を示すためにパス非稼働ビットに 1 が格納されます。パス非稼働条件が認識された後でチャンネル・プログラムの実行が延期された場合は、その後チャンネル・プログラムが再開されれば、パス非稼働条件は保留状態ではなくなります。代わりに、前の表示は消去され、パス非稼働の表示 (存在する場合) は、チャンネル・サブシステムがチャンネル・プログラムの実行を再開しようとしたときのパス非稼働を表しています。

機能制御 (FC)

機能制御フィールドは、サブチャンネルで示されている基本入出力機能を示します。このフィールドは、最大 2 つの機能の受理を示します。このフィールドは、SCSW の最初のワードのビット位置 17-19 にあります。インストールされているサブチャンネルについてこのフィールドが意味を持つのは、そのサブチャンネルが有効なときです (15-4 ページの『装置番号有効 (V)』を参照)。活動制御フィールドと状況制御フィールドの両方がすべて 0 であるときは、機能制御フィールドにはすべて 0 が含まれます。個々のビットの意味は以下のとおりです。

開始機能 (ビット 17): ビット 17 が 1 のときは、開始機能が要求されており、それがサブチャンネルで保留中または進行中であることを示します。開始機能を要求するには、START SUBCHANNEL を実行します。サブチャンネルで開始機能が示されるのは、START SUBCHANNEL の実行中に条件コード 0 が設定されたときです。サブチャンネルが、状況保留のみか、あるいは警報状況、1 次状況、または 2 次状況を任意の組み合わせで伴う状況保留状態にあるときに、TEST SUBCHANNEL が実行されると、サブチャンネルでの開始機能表示は消去されます。また、CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの開始機能表示が消去されます。

停止機能 (ビット 18): ビット 18 が 1 のときは、停止機能が要求されており、それがサブチャンネルで保留中または進行中であることを示します。停止機能を要求するには、HALT SUBCHANNEL を実行します。サブチャンネルで停止機能が示されるのは、HALT SUBCHANNEL で条件コード 0 が設定されたときです。TEST SUBCHANNEL の実行により次に発生した状況保留条件が消去されると、サブチャンネルでの停止機能表示は消去されます。どの状況保留条件が次に発生するかは、HALT SUBCHANNEL が実行される際のサブチャンネルの状態によって決まります。HALT SUBCHANNEL の実行時に、サブチャンネルがサブチャンネル・アクティブである場合は、次に起きる状況保留条件は、少なくとも 1 次状況の表示を伴う状況保留です。HALT SUBCHANNEL の実行時に、サブチャンネルが装置アクティブである場合は、次に起きる状況保留条件は、少なくとも 2 次状況の表示を伴う状況保留です。HALT SUBCHANNEL の実行時に、サブチャンネルが延期されており、中間状況を伴う状況保留状態にある場合は、次に起きる状況保留条件は、中間状況を伴う状況保留です。HALT SUBCHANNEL の実行時に、サブチャンネルがアイドル状態になっている場合は、次に起きる状況保留条件は状況保留のみです。CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの停止機能表示は消去されます。通常の操作では、この機能

はビット 17 とともに示されます。つまり、保留中または進行中の開始機能があり、それが停止されることを意味します。

消去機能 (ビット 19): ビット 19 が 1 のときは、消去機能が要求されており、それがサブチャンネルで保留中または進行中であることを示します。消去機能を要求するには、CLEAR SUBCHANNEL を実行します。サブチャンネルで停止機能が示されるのは、CLEAR SUBCHANNEL で条件コード 0 が設定されたときです (14-4ページの『CLEAR SUBCHANNEL』を参照)。結果の状況保留条件が TEST SUBCHANNEL により消去されると、消去機能表示は消去されます。

活動制御 (AC)

活動制御フィールドは、SCSW の最初のワードのビット位置 20-26 にあります。このフィールドは、すでにサブチャンネルで受理されている基本入出力機能の現在の進行状況を示します。プログラムは、このフィールドの内容を使用して、基本入出力機能の完了の程度を判断することができます。インストールされているサブチャンネルで活動制御フィールドが意味を持つのは、そのサブチャンネルが有効なときです (15-4ページの『装置番号有効 (V)』を参照)。ただし、基本入出力機能の実行中に IFCC または CCC 条件が検出され、その機能が保留中であるものとして示されている場合は、入出力装置では入出力操作が行われていることも行われていないこともあります。活動制御ビットは以下のように定義されています。

ビット	意味
20	再開保留
21	開始保留
22	停止保留
23	消去保留
24	サブチャンネル・アクティブ
25	装置アクティブ
26	延期

状況制御フィールドの状況保留ビットが 0 で、活動制御フィールドがすべて 0 である SCSW が格納されたとき、サブチャンネルはアイドルであるかまたはアイドル状態になっていると言われます。

注: 機能制御フィールド内のビットと、活動制御フィールド内の再開保留ビット、開始保留ビット、停止保留ビット、消去保留ビット、サブチャンネル・アクティブ・ビット、および延期ビットが表すすべてのビットがサブチャンネルでリセットされるのは、サブチャンネルが、(1) 状況保留のみのとき、(2) 1 次状況を伴う状況保留のとき、(3) 警報状況を伴う状況保留のとき、または、(4) 中間状況を伴う状況保留であってかつ延期もされている

ときに、TEST SUBCHANNEL が実行された場合です。

再開保留 (ビット 20): ビット 20 が 1 のときは、サブチャンネルが再開保留中であることを示します。チャンネル・サブシステムは、開始機能を実行中であることもそうでないこともあります。RESUME SUBCHANNEL で条件コード 0 が設定されると、サブチャンネルは再開保留になります。サブチャンネルがどの時点で再開保留ではなくなるかは、再開保留条件が認識されたときのサブチャンネルの状態と、装置の状態 (チャンネル・プログラムの実行が再開される場合) によって決まります。

再開保留条件が認識されたときに、サブチャンネルが延期状態にある場合は、その延期の原因となった CCW が再取り出しされ、延期フラグの設定が調べられ、チャンネル・サブシステムにより以下のいずれかのアクションがとられます。

1. CCW 延期フラグが 1 である場合は、装置は選択されず、サブチャンネルは再開保留ではなくなり、チャンネル・プログラムの実行は延期状態のままとなります。
2. CCW 延期フラグが 0 の場合は、チャンネル・サブシステムは、変更された開始機能を実行することにより、チャンネル・プログラムの実行を再開しようとし、装置は、チャンネル・プログラム実行の再開を、新しいチャンネル・プログラム実行の開始と見なします。再開機能が実行されると、チャンネル・サブシステムは、START SUBCHANNEL によりすでにサブチャンネルに渡されている ORB パラメータを使用して、新規の開始機能が開始された場合と同様にパス管理操作を行います。ただし、この場合のチャンネル・プログラム・アドレスは、チャンネル・プログラム実行を延期させた CCW のアドレスです。

開始機能の実行時に、チャンネル・サブシステムが、(1) 最初のコマンドに指定されている入出力操作の開始を試みることができないと判断するか、(2) 最初のコマンドに指定されている入出力操作を開始しようとしたが、そのコマンドが受理されないと判断するか、または、(3) IFCC または CCC 条件を検出し、最初のコマンドが受理されたかどうかを判断できないときは、サブチャンネルは再開保留のままになります。(15-17ページの『開始機能と再開機能』を参照してください。)

以下のいずれかのイベントが起きると、サブチャンネルは再開保留ではなくなります。

- a. 開始機能の実行中に、サブチャンネルがサブチャンネル/装置アクティブまたは装置アクティブのみになるか、または、最初のコマンドが受理されたときに、初期状況としてチャンネル終了と装置

終了が示され、CCW はコマンド・チェーニングを指定していない。

- b. CLEAR SUBCHANNEL が実行された。
- c. TEST SUBCHANNEL が、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを消去したか、または状況保留条件のみを消去した。
- d. サブチャンネルが延期されているときに、TEST SUBCHANNEL により中間状況が消去された。
- e. CANCEL SUBCHANNEL が実行された結果、条件コード 0 が設定された。

再開保留条件が認識されたときに、サブチャンネルが延期状態にないときは、最後に取り出された CCW (取り出された CCW がある場合) の CCW 延期フラグが調べられ、チャンネル・サブシステムにより以下のいずれかのアクションがとられます。

1. CCW が 1 つも取り出されていないか、または最後に取り出された CCW の延期フラグが 0 である場合は、サブチャンネルは再開保留ではなくなり、再開機能は実行されません。
2. 最後に取り出された CCW の延期フラグが 1 の場合は、サブチャンネルは再開保留ではなくなり、その CCW が再取り出しされます。再取り出しされた CCW の延期フラグが 0 の場合は、サブチャンネルはチャンネル・プログラムの実行を続けます。再取り出しされた CCW の延期フラグが 1 の場合は、サブチャンネルはチャンネル・プログラムの実行を延期します。

一部のモデルでは、S フラグが有効に 1 に設定されている CCW が取り出された後でのみ、再開保留条件が認識されます。したがって、あるサブチャンネルが再開保留状態にあるときに、チャンネル・プログラムの実行時に、S フラグが有効に 1 に設定された CCW が取り出されなかった場合は、TEST SUBCHANNEL が 1 次割り込み条件を消去するまで、そのサブチャンネルは再開保留のままになります。

開始保留 (ビット 21): ビット 21 が 1 のときは、サブチャンネルが開始保留中であることを示します。チャンネル・サブシステムは、開始機能を実行中であることもそうでないこともあります。START SUBCHANNEL で条件コード 0 が設定されると、サブチャンネルは開始保留になります。開始機能の実行時に、チャンネル・サブシステムが、(1) 最初のコマンドに指定されている入出力操作の開始を試みることができないと判断するか、(2) 最初のコマンドに指定されている入出力操作を開始しようとしたが、そのコマンドが受理されないと判断するか、

または、(3) IFCC または CCC 条件を検出し、最初のコマンドが受理されたかどうかを判別できないときは、サブチャンネルは開始保留のままになります。(15-17ページの『開始機能と再開機能』を参照してください。)

以下のいずれかが起きると、サブチャンネルは開始保留ではなくなります。

1. 開始機能の実行中に、サブチャンネルがサブチャンネル/装置アクティブまたは装置アクティブのみになるか、または、最初のコマンドが受理されたときに、初期状況としてチャンネル終了と装置終了が示され、CCW はコマンド・チェーニングを指定していない。
2. CCW 内で延期フラグが有効に 1 に設定されているために、サブチャンネルが延期状態になった。
3. CLEAR SUBCHANNEL が実行された。
4. TEST SUBCHANNEL が、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを消去したか、または状況保留条件のみを消去した。
5. CANCEL SUBCHANNEL が実行された結果、条件コード 0 が設定された。

停止保留 (ビット 22): ビット 22 が 1 のときは、サブチャンネルが停止保留中であることを示します。チャンネル・サブシステムは、停止機能を実行中であることもそうでないこともあります。HALT SUBCHANNEL で条件コード 0 が設定されると、サブチャンネルは停止保留になります。停止機能の実行時に、チャンネル・サブシステムが、(1) 装置への停止信号の発信を試みることができないと判断するか、(2) 装置への停止信号の発信の試みが成功していないと判断するか、または、(3) IFCC または CCC 条件を検出し、装置に対して停止信号が発信されたかどうかを判別できないときは、サブチャンネルは停止保留のままになります。(15-14ページの『停止機能』を参照してください。)

以下のいずれかが起きると、サブチャンネルは停止保留ではなくなります。

1. 停止機能を実行しているときに、チャンネル・サブシステムが、装置に停止信号がすでに発信されていると判断した。
2. CLEAR SUBCHANNEL が実行された。
3. TEST SUBCHANNEL が、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを消去したか、または状況保留条件のみを消去した。
4. サブチャンネルが延期されているときに、TEST SUBCHANNEL により中間状況が消去された。

消去保留 (ビット 23): ビット 23 が 1 のときは、サブチャンネルが消去保留中であることを示します。チャンネル・サブシステムは、消去機能を実行中であることもそうでないこともあります。CLEAR SUBCHANNEL で条件コード 0 が設定されると、サブチャンネルは消去保留になります。消去機能の実行時に、チャンネル・サブシステムが、(1) 装置への消去信号の発信を試みることができないと判断するか、(2) 装置への消去信号の発信の試みが成功していないと判断するか、または、(3) IFCC または CCC 条件を検出し、装置に対して消去信号が発信されたかどうかを判別できないときは、サブチャンネルは消去保留のままになります。(15-13ページの『消去機能』を参照してください。)

以下のいずれかが起きると、サブチャンネルは消去保留ではなくなります。

1. 消去機能を実行しているときに、チャンネル・サブシステムが、装置に消去信号がすでに発信されていると判断した。
2. TEST SUBCHANNEL により状況保留条件のみが消去された。

サブチャンネル・アクティブ (ビット 24): ビット 24 が 1 のときは、サブチャンネルがサブチャンネル・アクティブであることを示します。サブチャンネルがサブチャンネル・アクティブであると見なされるのは、現在そのサブチャンネルで入出力操作が行われているときです。サブチャンネルがサブチャンネル・アクティブになるのは、サブチャンネルで最初のコマンドが受理され、開始機能または再開機能がただちに終結しないときです。(15-40ページの『入出力操作の即時終結』を参照してください。) 以下のいずれかが起きると、サブチャンネルはサブチャンネル・アクティブではなくなります。

1. サブチャンネルが延期状態になった。
2. サブチャンネルが、1 次状況を伴う状況保留状態になった。
3. CLEAR SUBCHANNEL が実行された。
4. 停止機能の実行時に、装置が非稼働状態と見なされた。

HALT SUBCHANNEL 命令または CLEAR SUBCHANNEL 命令で指定された機能を実行しているときには、サブチャンネルはサブチャンネル・アクティブにはなりません。

装置アクティブ (ビット 25): ビット 25 が 1 のときは、サブチャンネルが装置アクティブであることを示します。サブチャンネルが装置アクティブであると見なされるのは、現在そのサブチャンネルの関連装置で入出力操作が

行われているときです。サブチャンネルが装置アクティブになるのは、最初のコマンドが受理された時点です。以下のいずれかが起きると、サブチャンネルは装置アクティブではなくなります。

1. サブチャンネルが延期状態になった。
2. サブチャンネルが、2 次状況を伴う状況保留状態になった。
3. CLEAR SUBCHANNEL が実行された。
4. 停止機能の実行時に、装置が非稼働状態と見なされた。

サブチャンネルが開始保留でない場合、または装置から受け取った状況が警報条件も示している場合は、サブチャンネルは 2 次状況を伴う状況保留になります。装置からの状況が受理されると、装置は、新たな入出力操作を行うためのコマンドを受理できるようになります。サブチャンネルが開始保留状態にあり、状況が、装置終了か、制御装置終了を伴う装置終了である場合は、チャンネル・サブシステムはその状況を破棄し、新しいチャンネル・プログラムのための開始機能を実行します(15-17ページの『開始機能と再開機能』を参照)。この場合は、サブチャンネルは 2 次割り込み条件を伴う状況保留にはならず、プログラムには状況は提示されません。

HALT SUBCHANNEL または CLEAR SUBCHANNEL 命令で指定された機能を実行しているときには、サブチャンネルは装置アクティブにはなりません。

延期 (ビット 26): ビット 26 が 1 のときは、サブチャンネルが延期状態にあることを示します。サブチャンネルが延期状態にあると見なされるのは、現在チャンネル・プログラムの実行が延期されているときです。サブチャンネルは、延期機能の一環として延期されます。(15-36ページの『チャンネル・プログラム実行の延期』を参照してください。)

以下のいずれかが起きると、サブチャンネルは延期状態ではなくなります。

1. RESUME SUBCHANNEL の実行後の再開機能の一環として、サブチャンネルがサブチャンネル/装置アクティブまたは装置アクティブのみになるか、または、最初のコマンドが受理されたときに、初期状況としてチャンネル終了と装置終了(状況修飾を伴う場合と伴わない場合がある)が示され、CCW はコマンド・チェーニングを指定していない。
2. CLEAR SUBCHANNEL が実行された。
3. TEST SUBCHANNEL が、1 次状況、2 次状況、および警報状況のいずれかの組み合わせを消去したか、または状況保留条件のみを消去した。

4. 停止機能が指定されているときに、TEST SUBCHANNEL により中間状況が消去された。
5. CANCEL SUBCHANNEL が実行された結果、条件コード 0 が設定された。

プログラミング上の注意: CLEAR SUBCHANNEL の実行後、サブチャンネルが状況保留になる前に、STORE SUBCHANNEL または TEST SUBCHANNEL により SCSW が格納されたときに、サブチャンネル・アクティブ・ビット (ワード 0 のビット 24) に 0 が格納されていても、それは、装置との間のデータ転送が停止したことを意味するわけではありません。消去機能の実行結果としてサブチャンネルが状況保留になるまでは、プログラムは、データ転送が停止したかどうかを判断することはできません。

状況制御 (SC)

状況制御フィールドは、SCSW の最初のワードのビット位置 27-31 にあります。このフィールドは、サブチャンネルまたは装置の状況、Z ビット、または、サブチャンネル延期割り込みの場合は延期ビット (ビット 26) で表した、割り込み条件の要約レベルの表示を、プログラムに提示します。サブチャンネルでの既存の条件の結果として、複数の要約表示が通知されることもあります。サブチャンネルが使用可能にされていて (15-2 ページの『使用可能 (E)』を参照)、かつ少なくともビット 31 が 1 であれば、そのサブチャンネルは状況保留状態にあると見なされます。サブチャンネルが使用不可にされているときは、サブチャンネルは状況保留にはされません。インストールされているサブチャンネルにおいて SCSW のワード 0 のビット 31 が意味を持つのは、そのサブチャンネルが有効なときです (15-4 ページの『装置番号有効 (V)』を参照)。ビット 27-30 が意味を持つのは、ビット 31 が 1 のときです。状況制御ビットは以下のように定義されています。

警報状況 (ビット 27): ビット 27 が 1 のとき (かつ状況保留ビットも 1 のとき) は、警報割り込み条件が存在することを示します。このような場合、サブチャンネルは、警報状況を伴う状況保留状態にあると言われます。警報割り込み条件が認識されるのは、サブチャンネルに警報状況が存在するときです。警報状況は、サブチャンネル状況の場合も装置状況の場合もあります。警報状況は、以下のいずれかの条件に該当する場合に、チャンネル・サブシステムまたは装置により生成される状況です。

- サブチャンネルがアイドル状態になっている (活動制御ビット 20-26 および状況制御ビット 31 がすべて 0)。
- サブチャンネルが開始保留状態にあり、状況条件によって入出力操作の開始が妨げられている。

- サブチャンネルがサブチャンネル/装置アクティブであり、状況条件によってコマンド・チェーニングが抑止されたか、または、チェーニングが指定されているとすれば抑止されることになる (15-29 ページの『チェーニング』を参照)。
- サブチャンネルがサブチャンネル/装置アクティブであり、コマンド・チェーニングは指定されておらず、チャンネル・プログラムの実行が終了したばかりであり、装置が提示した状況は、コマンドの順次実行を変更しようとする試みを示している (状況修飾を使用してコマンドの順次実行を変更する方法についての詳細は、「ESA/390 Common I/O-Device Commands」(SA22-7204) を参照)。
- サブチャンネルが装置アクティブのみであり、装置が提示した状況は、装置終了、制御装置終了、または装置終了と制御装置終了以外の状況である。
- サブチャンネルは延期されている (ビット 26 が 1)。

警報割り込み条件が認識されたときに、サブチャンネルが開始保留状態にある場合は、サブチャンネルは警報状況を伴う状況保留になり、据え置き条件コード 1 が設定され、開始保留ビットは 1 のままになり、保留されている入出力操作の実行は開始されません。

TEST SUBCHANNEL の実行により、警報状況ビットと状況保留ビットがそれぞれ 1 に設定された SCSW が IRB に格納されると、サブチャンネルでの警報割り込み条件は消去されます。また、CLEAR SUBCHANNEL が実行された場合も、警報割り込み条件は消去されます。

サブチャンネルに警報状況が存在するときは、その状況は必ずプログラムに提示されます。警報状況には、例えば、アテンション、装置終了 (これは作動不能状態から作動可能状態への変化を通知します)、誤長、プログラム・チェック、および装置チェックなどがあります。

中間状況 (ビット 28): ビット 28 が 1 のとき (かつ状況保留ビットも 1 のとき) は、中間割り込み条件が存在することを示します。このような場合、サブチャンネルは、中間状況を伴う状況保留状態にあると言われます。中間条件が示されることがあるのは、Z ビット (サブチャンネル制御フィールドの)、延期ビット (活動制御フィールドの)、または PCI ビット (サブチャンネル状況フィールドの) が 1 のときです。

ORB 内の初期状況割り込み制御ビットが 1 のときは、サブチャンネルがサブチャンネル・アクティブになったときのみ、サブチャンネルは中間状況を伴う状況保留になります (Z ビットが示されます)。サブチャンネルがサブチャネ

ル・アクティブにならない場合は、Z 条件は生成されません。

延期制御が指定されており、ORB の中で延期を原因とする中間割り込み条件の生成が抑止されていないときは、延期フラグが 1 に設定されている CCW が現 CCW になると、延期が原因でサブチャンネルが中間状況を伴う状況保留状態になることがあります。チャンネル・プログラムの最初の CCW で延期フラグが指定されているときは、最初の CCW の中のコマンドが装置に転送される前に、チャンネル・プログラムの実行は延期され、サブチャンネルは中間状況を伴う状況保留状態になります (延期ビットが示されます)。コマンド・チェーニング時に取り出された CCW の中で延期フラグが指定されているときは、先行する CCW の実行が完了した後で、チャンネル・プログラムの実行が延期され、サブチャンネルは中間状況を伴う状況保留状態になります (延期ビットが示されます)。

CCW の中で PCI フラグが指定されているときに、PCI が原因で中間割り込み条件が生成されるかどうかは、その CCW がチャンネル・プログラムの最初の CCW かどうかによって異なります。チャンネル・プログラムの最初の CCW で PCI フラグが指定されているときは、サブチャンネルが中間状況を伴う状況保留状態になる (PCI ビットが示される) のは、その CCW のコマンドが受理されたことを示す初期状況が受信された後です。チェーニング時に取り出された CCW の中で PCI フラグが指定されているときは、サブチャンネルが中間状況を伴う状況保留状態になる (PCI ビットが示される) のは、先行する CCW の実行が完了した後です。PCI を含む割り込み条件が TEST SUBCHANNEL により消去される前にチェーニングが起きた場合は、その条件は次の CCW に継承されます。この継承はデータ・チェーニングとコマンド・チェーニングの両方で行われ、いずれの場合も、条件はチャンネル内転送コマンドにより伝搬されます。

HALT SUBCHANNEL の実行時に、サブチャンネルが中間割り込み条件を伴う状況保留状態にある場合は、その中間割り込み条件はサブチャンネルに残されますが、割り込み要求があった場合その要求は撤回され、サブチャンネルは状況保留ではなくなります。サブチャンネルは、停止機能の実行が終了するまで、状況保留ではなくなります。その後で、サブチャンネルは状況保留状態になり、同時に中間状況が示されます (任意の組み合わせの 1 次状況、2 次状況、および警報状況と一緒に示されることもあります)。

TEST SUBCHANNEL の実行により、中間状況ビットと状況保留ビットがそれぞれ 1 に設定された SCSW が IRB に格納されると、サブチャンネルでの中間割り込み条

件は消去されます。また、CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの中間割り込み条件は消去されます。

1 次状況 (ビット 29): ビット 29 が 1 のとき (かつ状況保留ビットも 1 のとき) は、1 次割り込み条件が存在することを示します。このような場合、サブチャンネルは、1 次状況を伴う状況保留状態にあると言われます。1 次割り込み条件は、サブチャンネルで開始機能の実行が完了したことを示す請求割り込み条件です。1 次割り込み条件は、格納された SCSW に記述されています。HALT SUBCHANNEL により入出力操作が中止されたときに装置が非稼働状態と見なされたために、装置に対して停止信号が出されていない場合は、サブチャンネルは、1 次状況 (および 2 次状況) を伴う状況保留にされ、サブチャンネル状況フィールドと装置状況フィールドはどちらも 0 に設定されます。

TEST SUBCHANNEL の実行により、1 次状況ビットと状況保留ビットがそれぞれ 1 に設定された SCSW が IRB に格納されると、サブチャンネルでの 1 次割り込み条件は消去されます。また、CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの 1 次割り込み条件は消去されます。

2 次状況 (ビット 30): ビット 30 が 1 のとき (かつ状況保留ビットも 1 のとき) は、2 次割り込み条件が存在することを示します。このような場合、サブチャンネルは、2 次状況を伴う状況保留状態にあると言われます。2 次割り込み条件は、通常、装置で入出力が完了したことを示す請求割り込み条件です。2 次割り込み条件は、格納された SCSW に記述されています。

HALT SUBCHANNEL により入出力操作が中止されたときに装置が非稼働状態と見なされたために、装置に対して停止信号が出されていない場合は、サブチャンネルは、2 次状況 (および、サブチャンネルがサブチャンネル・アクティブでもある場合は 1 次状況) を伴う状況保留にされ、サブチャンネル状況と装置状況はどちらも 0 に設定されます。

TEST SUBCHANNEL の実行により、2 次状況ビットが 1 に設定された SCSW が IRB に格納されると、サブチャンネルでの 2 次割り込み条件は消去されます。また、CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの 2 次割り込み条件は消去されます。

状況保留 (ビット 31): ビット 31 が 1 のときは、サブチャンネルが状況保留状態にあり、割り込み条件の原因を示す情報がプログラムに提示されることを示します。中間状況、1 次状況、2 次状況、または警報状況が生成

されると、サブチャンネルは必ず状況保留になります。アイドル状態にあるサブチャンネルを指定して HALT SUBCHANNEL を実行すると、停止機能の実行後にサブチャンネルは状況保留になり、停止機能が完了したことがプログラムに通知されます。TEST SUBCHANNEL の実行により、状況保留ビットが 1 に設定された SCSW が IRB に格納されると、サブチャンネルでの状況保留条件は消去されます。また、CLEAR SUBCHANNEL が実行されたときも、サブチャンネルでの状況保留条件は消去されます。CLEAR SUBCHANNEL が実行されたときに、指定したサブチャンネルが稼働状態であれば、消去機能の実行後にサブチャンネルは状況保留になり、消去機能が完了したことがプログラムに通知されます。

注: 状況保留ビットと、状況制御フィールドのその他のビットとの組み合わせにより、状況条件のタイプが示されます。例えば、ビット 29 と 31 が 1 であれば、サブチャンネルは 1 次状況を伴う状況保留状態にあります。しかし、ビット 31 のみが 1 である場合は、サブチャンネルは、状況保留状態または消去保留のみの状態にあると言われます。状況制御フィールド内でビット 31 のみが 1 である場合は、サブチャンネル状況フィールドと装置状況フィールド内のビットの設定は、すべて予測不能です。状況制御フィールドのビット 31 が 1 でない場合は、そのフィールド内の残りのビットには意味がありません。

CCW アドレス・フィールド

ワード 1 のビット 1-31 は絶対アドレスを形成します。ここに示されるアドレスは、16-19ページの図16-4に示すとおり、SCSW が格納されたときのサブチャンネルの状態によって決まります。サブチャンネル状況フィールドが、チャンネル制御チェック、チャンネル・データ・チェック、またはインターフェース制御チェックを示しているときは、ESW 内の CCW アドレス・フィールド妥当性フラグが 1 であれば、回復のために CCW アドレス・フィールドを使用できます。

プログラミング上の注意: 操作用としてフォーマット 0 の CCW が指定されているときに、チャンネル・プログラム・アドレスの中で検出された CCW アドレス (15-24ページの『チャンネル・プログラム・アドレス (Channel-Program Address)』を参照) またはチェーニング時に生成された CCW アドレスが原因で、チャンネル・サブシステムが $2^{24} - 1$ より大きい記憶位置から CCW を取り出すことになる場合は、その無効なアドレスは、切り捨てられずに SCSW の CCW アドレス・フィールドに格納されます。操作用としてフォーマット 1 の CCW が指定されているときに、無効なアドレスが原因で、チャンネル・サブシステムがチェーニング時に $2^{31} - 1$ より大きい記憶位置から CCW を取り出した場合は、その無効なアドレスの右端 31 ビットが CCW アドレス・フィールドに格納されます。

サブチャネルの状態 ¹	CCW アドレス ²
開始保留 (UUUU0/AIP SX) ³	予測不能
開始保留および装置アクティブ (UUUU0/AIP SX) ³	予測不能
サブチャネル/装置アクティブ (UUUU0/AIP SX) ³	予測不能
装置アクティブのみ (UUUU0/AIP SX)	予測不能
延期 (YYYYY/AIP SX) ³	注 1 を参照
サブチャネルが開始保留状態にあるときに装置から非請求警報状況が通知されたことに起因する状況保留 (10001/AIP SX) ³	チャンネル・プログラム・アドレス + 8
すべてのパスで装置が非稼働と見なされたことに起因する状況保留 (0Y111/AIP SX) ³	チャンネル・プログラム・アドレス + 8
サブチャネルが開始保留および装置アクティブであるときに装置から非請求警報状況が通知されたことに起因する状況保留 (10011/AIP SX) ³	チャンネル・プログラム・アドレス + 8
サブチャネルが開始保留 ³ または開始保留で装置アクティブ ³ であるときに、チャンネル・サブシステムが非請求警報状況を生じたことに起因する状況保留 (10111/AIP SX)	注 2 を参照
サブチャネルがサブチャネル/装置アクティブであるときに、プログラム制御割り込み条件が起きた場合の状況保留 (01001/AIP SX) ³	最後に認識された PCI を含んでいる CCW の CCW + 8、または続いて現 CCW になった CCW より 8 だけ大きいアドレス
サブチャネルがサブチャネル/装置アクティブにあるときに、初期状況割り込み条件が起きた場合の状況保留 (01001/AIP SX) ³	中間割り込み条件の原因となった CCW の CCW + 8、または続いて現 CCW になった CCW
状況保留 (1Y1Y1/AIP SX)：以下のいずれかの条件に起因するプログラム・チェックが原因で中止が生じた。 ³	
ORB のワード 1 のビット 24 が 1 に設定されている。誤長表示抑止機能がインストールされていない。	チャンネル・プログラム・アドレス + 8
ORB 内の使用されないビットが 0 に設定されていない。	チャンネル・プログラム・アドレス + 8
チャンネル内転送 (TIC) で無効な CCW アドレスが指定された。	TIC のアドレス + 8
ORB のチャンネル・プログラム・アドレス内で無効な CCW アドレスが指定された。	チャンネル・プログラム・アドレス + 8 ⁴

図 16-4 (1/4). サブチャネルの状態に基づく CCW アドレス

サブチャネル状態 ¹	CCW アドレス ²
TIC 内の無効な CCW アドレス	TIC のアドレス + 8
ORB 内のチャネル・プログラム・アドレスの中の無効な CCW アドレス	チャネル・プログラム・アドレス + 8 ⁴
チューニング時の無効な CCW アドレス	無効な CCW アドレス + 8
無効なコマンド・コード	無効な CCW のアドレス + 8 ⁵
無効なカウント	無効な CCW のアドレス + 8 ⁵
無効な IDAW アドレス指定	無効な CCW のアドレス + 8 ⁵
CCW 内の無効な IDAW アドレス	無効な CCW のアドレス + 8 ⁵
IDAW の順次取り出し時の無効な IDAW アドレス	現 CCW のアドレス + 8
無効なデータ・アドレス指定 (フォーマット 1)	無効な CCW のアドレス + 8 ⁵
CCW 内の無効なデータ・アドレス	無効な CCW のアドレス + 8 ⁵
記憶域の順次アクセス時の無効なデータ・アドレス	現 CCW のアドレス + 8
IDAW 内の無効なデータ・アドレス	現 CCW のアドレス + 8
無効な IDAW 指定	現 CCW のアドレス + 8
TIC 以外の CCW の場合の無効 CCW (フォーマット 0 または 1)	無効な CCW のアドレス + 8 ⁵
無効な延期フラグ: データ・チューニング時に取り出された CCW で延期フラグが 1 に設定されている	無効な CCW のアドレス + 8
無効な延期フラグ: CCW で延期フラグが 1 に設定されているが、ORB 内で延期制御が指定されていない	無効な CCW のアドレス + 8
TIC の場合の無効な CCW (フォーマット 1)	TIC のアドレス + 8
無効なシーケンス: 2 つの TIC	2 番目の TIC のアドレス + 8
無効なシーケンス: データ転送を伴わない 256 個以上の CCW	256 番目の CCW のアドレス + 8
状況保留 (1Y1Y1/AIP SX): 以下のように、記憶保護チェックが原因で中止が生じた。 ³	
CCW アクセス時	保護されている CCW のアドレス + 8 ⁵
データ・アクセス時または IDAW アクセス時	現 CCW のアドレス + 8

図 16-4 (2/4). サブチャネルの状態に基づく CCW アドレス

サブチャネルの状態 ¹	CCW アドレス ²
状況保留 (1Y1Y1/AIP SX): チューニング・チェックが原因で中止が生じた ³	現 CCW のアドレス + 8
状況保留 (YY1Y1/AIP SX): カウント制御の下で中止が生じた ³	現 CCW のアドレス + 8 ⁶
状況保留 (1Y1Y1/AIP SX): 警報状況が原因で、操作が未完了のまま装置により中止された ³	現 CCW のアドレス + 8 ⁶
HALT SUBCHANNEL による中止が生じ、以下に示す活動制御フィールド・ビットが 1 に設定された後の状況保留 (YYYY1/AIP SX)	
状況保留のみ	予測不能
開始保留 ³	予測不能
装置アクティブおよび開始保留 ³	予測不能
装置アクティブ	予測不能
サブチャネル/装置アクティブ ³	最後に実行された CCW のアドレス + 8
延期	延期の原因となった CCW のアドレス + 8
延期および再開保留	予測不能
CLEAR SUBCHANNEL による中止後の状況保留 (00001/AIP SX)	予測不能
状況保留 (YY1Y1/AIP SX): サブチャネルで操作が正常に完了した ³	最後に実行された CCW のアドレス + 8 ⁶
状況保留 (00011/AIP SX)	予測不能
状況保留 (10001/AIP SX)	予測不能
状況保留 (00001/AIP SX)	予測不能
状況保留 (1Y111/AIP SX): チャンネル制御チェックまたはインターフェース制御チェック以外の警報状況が原因でコマンド・チューニングが抑止された ³	現 CCW のアドレス + 8 ⁶
チャンネル制御チェックまたはインターフェース制御チェックによる警報状況に起因する状況保留 (1YYYY1/AIP SX) ³	注 3 を参照 ⁶
チャンネル・データ・チェックに起因する状況保留 (1Y1Y1/AIP SX) ³	現 CCW のアドレス + 8 ⁶

図 16-4 (3/4). サブチャネルの状態に基づく CCW アドレス

説明:

- この欄で使用されている記号の意味は、以下のとおりです。
 - A 警報状況
 - I 中間状況
 - P 1 次状況
 - S 2 次状況
 - X 状況保留状況制御ビット設定の可能な組み合わせは、以下の記号を使用して「/」記号の左側に示してあります。
 - 0 対応する条件は示されません。
 - 1 対応する条件は示されます。
 - U 予測不能。サブチャンネルが状況保留でないときは、対応する条件は意味を持ちません。
 - Y 対応する条件は重要ではなく、サブチャンネルの状態に応じて示されます。
- CCW が現 CCW となるのは、(1) チャンネル・プログラムの最初の CCW であって、すでに取り出されているとき、(2) コマンド・チューニング時に、直前の CCW がすでに現 CCW ではなく、すでに新しい CCW が取り出されているとき、または、(3) データ・チューニングの場合に、新しい CCW が入出力操作の制御を引き継いだときです (第 15 章、『基本入出力機能』の『データ・チューニング』を参照)。チューニングが指定されていないかまたは抑止されている場合は、CCW は、チャンネル・サブシステムが 2 次状況を受理した時点で、現 CCW ではなく、最後に実行された CCW となります。コマンド・チューニングでは、CCW は、装置終了状況が受理された時点で現 CCW ではなく、データ・チューニングでは、CCW のデータの最後のバイトが受理された時点で現 CCW ではなくとなります。
- サブチャンネルは再開保留状態の場合もあります。
- 格納されるアドレスは、該当アドレスが無効であるか保護されているかに関係なく、チャンネル・プログラム・アドレス (ORB 内の) + 8 です。
- 格納されるアドレスは、該当アドレスが無効であるか保護されているかに関係なく、現 CCW のアドレス + 8 です。
- 誤長が示されるかどうかは、現 CCW 内の誤長表示抑止フラグの設定によって決まります (第 15 章、『基本入出力機能』の『チャンネル・コマンド・ワード』を参照)。

注:

- サブチャンネルが再開保留でもある場合を除き、格納されるアドレスは、延期の原因となった CCW のアドレス + 8 です。その他の場合は、格納されるアドレスは予測不能です。
- 示される CCW のアドレスは、示される警報状況によって決まります。例えば、プログラム・チェックまたは記憶保護チェック条件が認識された場合は、格納される CCW アドレスは、それぞれ、この表でプログラム・チェックまたは記憶保護チェックについての項に示されているアドレスと同じです。また、インターフェース制御チェックまたはチャンネル制御チェックについて警報状況が示されている場合は、格納される CCW アドレスは、チャンネル・プログラム・アドレス (ORB 内の) + 8 となるか、または、サブチャンネル・ログアウト内のフィールド妥当性フラグに指定により無効となります。
- サブチャンネル・ログアウト情報のビット 21 に 1 が格納されているときは、最後に取り出された CCW のコマンドがまだ装置で受理されていなければ、アドレスはその CCW のアドレス + 8 です。エラー条件が認識されたときに、すでに装置でコマンドが受理されている場合は、格納されるアドレスは、最後に実行された CCW のアドレス + 8 です。

図 16-4 (4/4). サブチャンネルの状態に基づく CCW アドレス

装置状況フィールド

装置状況条件は、入出力装置により生成され、チャンネル・バスを介してチャンネル・サブシステムに提示されます。各装置タイプにおけるこれらの条件の発生時期と原因は、該当装置のシステム・ライブラリー資料に示されています。装置状況フィールドが意味を持つのは、サブチャンネルが、1 次状況、2 次状況、中間状況、または警報状況のいずれかの組み合わせを伴う状況保留状態にあ

るときです。サブチャンネルが中間状況のみを伴う状況保留状態にあるときは、装置状況フィールドは 0 です。サブチャンネル状況フィールドが、チャンネル制御チェック、チャンネル・データ・チェック、またはインターフェース制御チェックを示しているときは、ESW 内の装置状況フィールド妥当性フラグが 1 であれば、回復のために装置状況フィールドを使用できます。サブチャンネルが状況保留状態にあり、据え置き条件コード 3 が示されているときは、装置状況フィールドの内容は意味を持ちません。

1 つのシステムの中で、複数のチャンネル・パスから入出力装置にアクセスできる場合は、シングル・パス・モードでチャンネル・サブシステムにより開始された操作に関する状況（請求状況）は、その操作を開始したチャンネル・パスを介して通知されます。マルチパス・モードで稼働している装置は、開始に使用されたチャンネル・パスと同じパス・グループに属している任意のチャンネル・パスを介して、請求状況を通知することができます。アテンション、装置例外、および、作動不能から作動可能への移行に伴う装置終了などのような、入出力操作に関連していない条件（非請求警報状況）の取り扱い、装置のタイプと、その装置に関するシステム・ライブラリー資料に示されている条件によって、異なります。

チャンネル・サブシステムは、入出力装置から受け取った状況ビットを変更しません。これらのビットは、チャンネル・パスを介して受信されたままの状態です。SCSW に示されます。入出力装置から受信される状況ビットについての詳細は、「ESA/390 Common I/O-Device Commands」(SA22-7204) を参照してください。

サブチャンネル状況フィールド

サブチャンネル状況条件は、チャンネル・サブシステムにより検出され、SCSW に示されます。装置の誤動作が原因で起きる条件を除き、サブチャンネル状況条件が起きるのは、チャンネル・サブシステムが、停止、再開、または開始機能を実行している間だけです。サブチャンネル状況フィールドが意味を持つのは、サブチャンネルが、1 次状況、2 次状況、中間状況、または警報状況のいずれかの組み合わせを伴う状況保留状態にあるときです。サブチャンネル状況フィールドが意味を持つ場合でも、サブチャンネル状況フィールドに含まれる個々のビットは予測不能な場合があります。サブチャンネルが状況保留状態にあり、据え置き条件コード 3 が示されているときは、サブチャンネル状況フィールドの内容は意味を持ちません。

プログラム制御割り込み

プログラム制御割り込み (PCI) フラグが 1 に設定されている CCW が現 CCW になると、中間割り込み条件が生成されます。PCI フラグが原因で生じる入出力割り込みは、割り込み要求のマスキングやシステム内のその他の活動が原因で遅延されることがあります（遅延時間は予測不能です）。(15-33 ページの『プログラム制御割り込み』を参照してください。) チャンネル・サブシステムが、チャンネル制御チェック条件またはインターフェース制御チェック条件に起因する警報割り込み条件を認識した場合、CCW 内の PCI フラグを原因とする既存の中間割り込み条件は、チャンネル・サブシステムにより認識されるときと、されないときがあります。

PCI 条件が検出されても、入出力操作の進行には影響はありません。

誤長

入出力操作として割り当てられている記憶域に含まれるバイト数が、入出力装置が要求または提供するバイト数に一致していないと、誤長が発生します。誤長が示される原因は、次のいずれかです。

入力のブロックが長い: 読み取り、逆方向読み取り、またはセンス操作で、割り当てられた主記憶域がいっぱいになった後で、装置がさらに 1 つ以上のバイトを主記憶域に転送しようとしたか、または、装置が、カウントがもっと大きければもっと多くのデータを転送できたことを示した。残りのバイトは主記憶域に格納されていません。SCSW 内のカウントは 0 です。

出力のブロックが長い: 書き込みまたは制御操作で、割り当てられている主記憶域を使い尽くしてしまった後で、装置がチャンネル・サブシステムに対して 1 つ以上のバイトを要求したか、または、装置が、カウントがもっと大きければもっと多くのデータを転送できたことを示した。SCSW 内のカウントは 0 です。

入力のブロックが短い: 読み取り、逆方向読み取り、またはセンス操作で、転送バイト数不足のため、操作に割り当てられている主記憶域がいっぱいにならない。SCSW 内のカウントは 0 ではありません。

出力のブロックが短い: 割り当てられている主記憶域に含まれているすべての情報が装置に転送される前に、装置が書き込みまたは制御操作を中止した。SCSW 内のカウントは 0 ではありません。

現 CCW で、SLI フラグが 1 に設定されていて、CD フラグが 0 に設定されていると、誤長表示は抑止されません。開始シーケンスで拒否された操作については、誤長は表示されません。カウント・フィールドが 0 以外で、サブチャンネルが誤長表示抑止モードである場合は、即時操作についても誤長表示は行われません。SCSW のカウント・フィールドが意味を持たないときは、誤長表示も意味を持ちません。

CCW の SLI フラグが 1 である場合、またはサブチャンネルが誤長抑止モードのときに即時操作で誤長条件が起きた場合以外は、誤長条件があるとコマンド・チェーニングは抑止されます。

プログラム・チェック

チャンネル・サブシステムがプログラミング・エラーを検出すると、プログラム・チェックが起きます。この条件の原因には以下のようなものがあります。

無効な CCW アドレス指定: チャンネル・プログラム・アドレス (CPA) またはチャンネル内転送コマンドが CCW をダブルワード境界上に指定していないか、あるいは、CPA のビット 0 またはチャンネル内転送コマンドを指定するフォーマット 1 の CCW のビット 32 が 0 でない。

無効な CCW アドレス: チャンネル・サブシステムが、使用可能でない主記憶位置から CCW を取り出そうとした。無効な CCW アドレスが生じるのは、例えば、プログラムが、ORB のチャンネル・プログラム・アドレス・フィールドまたはチャンネル内転送コマンドの中で無効なアドレスを指定した場合や、チューニング時に、チャンネル・サブシステムが使用不可の記憶位置から CCW を取り出そうとした場合などです。主記憶位置が使用不可と見なされるのは、以下のいずれかの条件が検出されたときです。

1. 絶対 CCW アドレスに対応する物理記憶位置がない。
2. ORB 内でフォーマット 0 の CCW が指定されているときに、絶対 CCW アドレスが $2^{24} - 1$ より大きい。
3. ORB 内でフォーマット 1 の CCW が指定されているときに、絶対 CCW アドレスが $2^{31} - 1$ より大きい。

無効なコマンド・コード: CPA で指定されている CCW またはコマンド・チューニング時に取り出された CCW の中で、コマンド・コードの右端 4 個のビット位置に 0 が含まれている。データ・チューニング時には、コマンド・コードの妥当性はテストされません。

無効なカウント (フォーマット 0): チャンネル内転送を指定する CCW 以外の CCW の中で、ビット位置 48-63 に 0 が含まれている。

無効なカウント (フォーマット 1): データ・チューニングを指定する CCW またはデータ・チューニング時に取り出された CCW の中で、ビット位置 16-31 に 0 が含まれている。

無効な IDAW アドレス指定: 間接的なデータ・アドレッシングが指定されており、次のいずれかの条件が検出された。

1. ORB がフォーマット 1 の IDAW を指定している、CCW 内のデータ・アドレス・フィールドの内容が最初の IDAW をワード境界上に指定していない。つまり、ビット 30 および 31 (フォーマット 0

の CCW の場合) またはビット 62 および 63 (フォーマット 1 の CCW の場合) が 0 でない。

2. ORB がフォーマット 2 の IDAW を指定している、CCW 内のデータ・アドレス・フィールドの内容が最初の IDAW をダブルワード境界上に指定していない。つまり、ビット 29-31 (フォーマット 0 の CCW の場合) またはビット 61-63 (フォーマット 1 の CCW の場合) が 0 でない。

無効な IDAW アドレス: チャンネル・サブシステムが、使用可能でない主記憶位置から IDAW を取り出そうとした。無効な IDAW アドレスが生じるのは、例えば、プログラムが、間接データ・アドレッシングを指定する CCW の中で無効なアドレスを指定した場合や、チャンネル・サブシステムが、IDAW の順次取り出しを行っているときに、使用不可の記憶位置から取り出そうとした場合などです。主記憶位置が使用不可と見なされるのは、以下のいずれかの条件が検出されたときです。

1. 絶対 IDAW アドレスに対応する物理記憶位置がない。
2. ORB 内でフォーマット 0 の CCW が指定されているときに、絶対 IDAW アドレスが $2^{24} - 1$ より大きい。
3. ORB 内でフォーマット 1 の CCW が指定されているときに、絶対 IDAW アドレスが $2^{31} - 1$ より大きい。

無効なデータ・アドレス指定: フォーマット 1 の CCW のビット 32 が 0 でない。

無効なデータ・アドレス: 以下のいずれかの条件が検出されると、チャンネル・サブシステムは無効なデータ・アドレスを認識します。

1. 指定されたデータ・アドレスを使用した結果、チャンネル・サブシステムは最大記憶域アドレスから 0 に循環しようとした。
2. 逆方向読み取り操作において、指定したデータ・アドレスを使用した結果、チャンネル・サブシステムは 0 から最大記憶域アドレスに循環しようとした。
3. チャンネル・サブシステムが、使用不可の記憶位置にデータを転送しようとした。

無効なデータ・アドレスが生じるのは、例えば、プログラムが、CCW または IDAW の中で使用不可の記憶位置を指定した場合や、チャンネル・サブシステムが、記憶域の順次アクセスにおいて使用不可の記憶位置にアクセスしようとした場合などです。主記憶位置が使用不可と

見なされるのは、以下のいずれかの条件が検出されたときです。

1. その記憶位置の絶対アドレスに対応する物理記憶位置がない。
2. ORB 内でフォーマット 0 の CCW が指定されていて、CCW 内で間接データ・アドレッシングが指定されていないときに、絶対アドレスが $2^{24} - 1$ より大きい。
3. ORB 内でフォーマット 1 の CCW が指定されていて、CCW 内で間接データ・アドレッシングが指定されていないときに、絶対アドレスが $2^{31} - 1$ より大きい。
4. ORB 内でフォーマット 1 の IDAW が指定されていて、CCW 内で間接データ・アドレッシングが指定されているときに、絶対アドレスが $2^{31} - 1$ より大きい。
5. サブチャンネルで限界モードがアクティブのときに、絶対アドレスが SET ADDRESS LIMIT に指定されているアドレッシング範囲を逸脱している。

注: 最大記憶域アドレスは、使用されている CCW と IDAW のフォーマットによって決まります。IDAW が使用されていない場合は、最大記憶域アドレスは、指定されている CCW のフォーマットに応じて以下のように決定されます。

1. 24 ビット (フォーマット 0) の CCW が指定されている場合は、チャンネル・サブシステムが認識する最大記憶域アドレスは $2^{24} - 1$ です。
2. 31 ビット (フォーマット 1) の CCW が指定されている場合は、チャンネル・サブシステムが認識する最大記憶域アドレスは $2^{31} - 1$ です。

IDAW が使用されている場合は、最大記憶域アドレスは、指定されている IDAW のフォーマットに応じて以下のように決定されます。

1. 31 ビット (フォーマット 1) の IDAW が指定されている場合は、チャンネル・サブシステムが認識する最大記憶域アドレスは $2^{31} - 1$ です。
2. 64 ビット (フォーマット 2 の IDAW が指定されている場合は、チャンネル・サブシステムが認識する最大記憶域アドレスは $2^{64} - 1$ です。

無効な IDAW 指定: 以下のいずれかの条件が検出されると、チャンネル・サブシステムは無効な IDAW 指定を認識します。

1. フォーマット 1 の IDAW のビット 0 が 0 でない。

2. 2 番目またはそれ以降のフォーマット 1 の IDAW が、2K バイト・ブロックの先頭バイト (逆方向読み取り操作の場合は、2K バイト・ブロックの最終バイト) の位置を指定していない。
3. 2 番目またはそれ以降のフォーマット 2 の IDAW が、2K バイト・ブロックまたは 4 K バイト・ブロック (ORB 内の 2K-IDAW 制御で指定されている方) の先頭バイト (逆方向読み取りの場合はブロックの最終バイト) の位置を指定していない。

無効な CCW (フォーマット 0): チャンネル内転送を指定している CCW 以外の CCW の中で、ビット位置 39 に 0 が含まれていない。

無効な CCW (フォーマット 1): チャンネル内転送を指定する CCW 以外の CCW の中で、ビット位置 15 に 0 が含まれていないか、または、チャンネル内転送を指定する CCW の中で、ビット位置 0-3 および 8-31 に 0 が含まれていない。

無効な延期フラグ: データ・チューニング時に取り出されたフォーマット 0 またはフォーマット 1 の CCW (チャンネル内転送を指定している CCW 以外) で、ビット位置 38 (フォーマット 0) または 14 (フォーマット 1) に 0 が含まれていない。チャンネル内転送を指定する CCW 以外の CCW で、ビット位置 38 (フォーマット 0 の CCW の場合) またはビット位置 14 (フォーマット 1 の CCW の場合) に 0 が含まれておらず、ORB のワード 1 のビット 4 に延期制御が指定されていない。

無効な ORB フォーマット: 操作要求ブロック (ORB) の中の 1 つまたは複数の予約ビット位置が 0 でない。(詳細については、15-20ページの『操作要求ブロック』を参照してください。) 誤長表示抑止機能がインストールされていない場合は、ORB のワード 1 のビット 24 も 0 でなければなりません。

無効なシーケンス: チャンネル・サブシステムが、2 つの連続した CCW を取り出し、その両方がチャンネル内転送を指定しているか、あるいは、モデルによっては、チャンネル・サブシステムが、コマンド・チューニングが指定されている 256 個以上の CCW のシーケンスを実行したが、入出力装置との間でデータの転送がまったく行われなかった。

装置での操作の開始時にプログラム・チェック条件が検出されると、操作は抑止され、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留にされます。装置で入出力操作が開始された後でこの条件が検出されたときは、次回に装置がデータ・バイトまたは状況

を要求するかまたは提供したときに、操作を終結するよう指示する信号が装置に送られます。その場合は、サブチャンネルは、装置から受信した状況に応じた状況保留状態になります。プログラム・チェック条件が起きると、コマンド・チューニングおよびコマンド再試行は抑止されます。

記憶保護チェック

チャンネル・サブシステムが記憶保護メカニズムにより禁止されている記憶域アクセスを行おうとすると、記憶保護チェックが起きます。記憶保護は、CCW、IDAW、および出力データの取り出しと、入力データの格納に適用されます。入出力操作に関連した記憶域アクセスでは、ORB 内で指定されているサブチャンネルがアクセス・キーとして使用されます。

装置で最初の CCW または IDAW を取り出すときに記憶保護チェック条件が検出されると、操作は抑止され、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留にされます。装置で入出力操作が開始された後で記憶保護チェックが検出されたときは、その記憶保護チェックより論理的に前にある使用可能なデータが転送された後で、操作を終結するよう指示する信号が装置に送られます。ただし、データ・チューニング時に、チャンネル・サブシステムが新しい IDAW または新しい CCW を取り出しているときにアクセス違反が起き、その新しい CCW または IDAW で指定されているデータが転送される前に装置がチャンネル終了条件を示す信号を出した場合は、その状況が受理され、サブチャンネルは 1 次状況と警報状況を伴う状況保留にされ、記憶保護チェックが示されます。CCW で指定されている操作、装置から受信した状況、およびサブチャンネルの現在の状態によっては、記憶保護チェック表示に他の表示が付随していることがあります。記憶保護チェック条件が起きると、コマンド・チューニングおよびコマンド再試行は抑止されます。

チャンネル・データ・チェック

チャンネル・データ・チェックは、主記憶域内に含まれていて現在入出力操作の実行に使用されているデータについて、訂正されていない記憶域エラーが検出されたことを示します。該当データが事前取り出しされたものであって、まだ使用されていない場合でも、この条件が示されることがあります。チャンネル・データ・チェックが示されるのは、チャンネル・サブシステムがデータを参照したときに、そのデータまたは関連のキーについて、主記憶域内に無効な検査ブロック・コード (CBC) がある場合です。

入力操作で、チャンネル・サブシステムが 1 つの完全な検査ブロックに満たない量のデータを格納しようとしたときに、その検査ブロックについて記憶域内に無効な CBC が検出された場合は、その記憶位置の内容は変更されず、無効な CBC を持ったままになります。出力操作では、チャンネル・データ・チェックが示されたときは、無効な CBC を持つ検査ブロックからのバイトは装置に転送されません。

記憶域アクセス時に転送できる最大バイト数は、モデルによって異なります。記憶域アクセスでチャンネル・データ・チェック条件が認識された場合は、チャンネル・サブシステムは、記憶機構との間で転送されたバイト数を検出できないことがあります。したがって、記憶機構との間で転送されたバイト数が正しく残余カウンタに反映されないことがあります。ただし、SCSW に格納されている残余カウンタは、記憶域アクセス・コードおよび CCW アドレスと共に使用されて、チャンネル・データ・チェック条件が認識されたページ内のバイト位置を示します。

チャンネル・データ・チェックとして示される条件が起きると、現在進行中の操作がある場合はその操作が中止されます。サブチャンネルは、装置から受け取った状況に応じて、1 次状況と警報状況を伴う状況保留か、または 1 次状況、2 次状況、および警報状況を伴う状況保留になります。TEST SUBCHANNEL により格納される SCSW のカウンタ・フィールドとアドレス・フィールドの内容は、中止された操作に関する情報を示しています。TEST SUBCHANNEL が実行されると、拡張状況ワード・フォーマット・ビットは 1 で、サブチャンネル・ログアウト情報が ESW に格納されます。

チャンネル・データ・チェック条件が事前取り出しされたデータに関連するものである場合は、故障記憶機構アドレス妥当性フラグ (ERW のビット 6) は常に 1 です。チャンネル・データ・チェック条件が生成される原因となった検査ブロック内の記憶位置のアドレスが、ESW の故障記憶機構アドレス・フィールドに格納されます。

サブチャンネルが開始保留状態にあるときに、事前取り出しされたデータについて、訂正不能の記憶域エラーまたはキー・エラーが検出されると、操作は装置で開始される前に取り消されます。その場合は、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留にされ、チャンネル・データ・チェックが示され、ESW に故障記憶機構アドレスが格納されます。

チャンネル・データ・チェックが示されたときは、サブチャンネルに関する測定データは格納されません。

チャンネル制御チェック

チャンネル・サブシステムの制御に影響を与えるような何らかのマシン誤動作があると、チャンネル制御チェックが起きます。CCW、IDAW、またはそれぞれに関連したキーについて無効な CBC がある場合は、この条件に該当します。また、事前取り出しされた CCW、IDAW、またはそれぞれに関連したキーについて無効な CBC が検出された場合、その CCW または IDAW が使用されていなくても、この条件が示されることがあります。

チャンネル制御チェックは、入出力操作で、主記憶機構との間で転送された情報にエラーが検出されたことを示すこともあります。ただし、この条件が検出されたときは、エラーは、チャンネル・パスの内部、つまり、チャンネル・サブシステムの内部、またはチャンネル・サブシステムと主記憶機構の間のパスで起きています。

チャンネル制御チェック条件が検出されると、現在進行中の操作がある場合はその操作がただちに中止されます。サブチャンネルは、中止のタイプ、現在のサブチャンネルの状態、および提示された装置状況（もしあれば）に応じて、1 次状況と警報状況を伴う状況保留か、または 1 次状況、2 次状況、および警報状況を伴う状況保留にされます。チャンネル・サブシステムがチャンネル制御チェック条件を認識した場合、CCW 内の PCI フラグが原因で生じている既存の中間割り込み条件は、チャンネル・サブシステムにより認識されるときと、されないことがあります。TEST SUBCHANNEL により格納される SCSW のカウント・フィールドとデータ・アドレス・フィールドの内容は、中止された操作に関する情報を示しています。TEST SUBCHANNEL が実行されると、拡張状況ワード・フォーマット・ビットは 1 で、サブチャンネル・ログアウト情報が ESW に格納されます。

チャンネル制御チェック条件が、事前取り出しされた CCW、事前取り出しされた IDAW、あるいは事前取り出しされた CCW または事前取り出しされた IDAW に関連したキーについて検出された無効 CBC に関するものであるときは、TEST SUBCHANNEL が実行されると、ビット 6 が 1 に設定された拡張報告ワードが、ESW に格納されます。

サブチャンネルが開始保留状態にあるときに、事前取り出しの際にチャンネル制御チェック条件が検出されると、操作は装置で開始される前に取り消されます。その場合は、サブチャンネルは、1 次状況、2 次状況、および警報状況を伴う状況保留にされ、チャンネル制御チェックが示され、故障記憶機構アドレスが ESW に格納されます。

サブチャンネルが停止保留中であり、そのサブチャンネルについて停止機能を実行しているときにチャンネル・サブシステムがチャンネル制御チェック条件を検出した場合は、

停止信号が出たことをチャンネル・サブシステムが確認できない限り、サブチャンネルは停止保留状態のままになります。チャンネル・サブシステムが停止信号を出そうとしていた場合であっても、停止信号が出たことを確認できなければ、サブチャンネルは停止保留のままになります。

サブチャンネルが開始保留中または再開保留中であり、そのサブチャンネルについて開始機能を実行しているときにチャンネル・サブシステムがチャンネル制御チェック条件を検出した場合は、最初のコマンドが受理されたことをチャンネル・サブシステムが確認できない限り、サブチャンネルは開始保留または再開保留状態のままになります。チャンネル・サブシステムが最初のコマンドについて入出力操作を開始しようとしていた場合であっても、そのコマンドが受理されたことを確認できなければ、サブチャンネルは開始保留または再開保留のままになります。チャンネル・サブシステムが、最初のコマンドが受理されたかどうかを判別できない場合は、サブチャンネルは、少なくとも警報状況と 1 次状況を伴う状況保留にされます。

チャンネル・サブシステムの誤動作があるときは、場合によっては、チャンネル制御チェック条件はマシン・チェック条件として報告されることがあります。

チャンネル制御チェックが示されたときは、サブチャンネルに関する測定データは格納されません。

プログラミング上の注意: SCSW の状況制御フィールドが、サブチャンネルが警報状況を伴う状況保留状態にあることを示していて、SCSW のフィールド妥当性フラグが、装置状況フィールドをエラー回復の目的に使用できないことを示している場合は、プログラムでは、(1) チャンネル・サブシステムが装置から警報状況を受理しようとしたときにチャンネル制御チェック条件が起きたものと想定して、(2) その警報状況自体は消滅していても、警報状況に対する適切な処置をとるようにしてください。

インターフェース制御チェック

インターフェース制御チェックは、チャンネル・パス上で無効な信号が発生したことを示します。この条件はチャンネル・サブシステムにより検出されるもので、通常、入出力装置の誤動作を示します。インターフェース制御チェックが起きる原因には、次のようなものがあります。

1. サブチャンネルがサブチャンネル/装置アクティブまたは装置アクティブであるときに、装置から受け取ったデータ・バイトまたは状況バイトについて、無効な検査ブロック・コードがある。
2. サブチャンネルが、アイドル、開始保留、延期、または停止保留状態にあるときに装置から受け取った状況バイトについて、無効な検査ブロック・コードがある。

3. 装置が、操作の開始時にチャンネル・サブシステムが指定したアドレスとは異なるアドレスで応答した。
4. コマンド・チェーニング時に、装置が非稼働状態にあると見なされた。
5. 入出力装置から信号が出されなかったか、無効な時点で出されたか、または出されていた時間の長さが無効であった。
6. チャンネル・サブシステムが、入出力エラー警報条件を認識した (16-34ページの『入出力エラー警報 (A)』を参照)。
7. ESW のビット 26 (装置状況チェックを示す) が 1 に設定されている。

インターフェース制御チェック条件が検出されると、現在進行中の操作があれば即時に中止され、サブチャンネルは、中止のタイプ、現在のサブチャンネルの状態、および装置の状況 (提示されている場合) に応じて、警報状況を伴う状況保留、1 状況と警報状況を伴う状況保留、2 次状況と警報状況を伴う状況保留、または、1 次状況、2 次状況、および警報状況を伴う状況保留にされます。チャンネル・サブシステムがインターフェース制御チェック条件を認識した場合、CCW 内の PCI フラグが原因で生じている既存の中間割り込み条件は、チャンネル・サブシステムにより認識されるときと、されないときがあります。TEST SUBCHANNEL が実行されると、拡張状況ワード・フォーマット・ビットは 1 になり、サブチャンネル・ログアウト情報が ESW に格納されます。

サブチャンネルが停止保留中であり、そのサブチャンネルについて停止機能を実行しているときにチャンネル・サブシステムがインターフェース制御チェック条件を検出した場合は、停止信号が出たことをチャンネル・サブシステムが確認できない限り、サブチャンネルは停止保留状態のままになります。チャンネル・サブシステムが停止信号を出そうとしていた場合であっても、停止信号が出たことを確認できなければ、サブチャンネルは停止保留のままになります。

サブチャンネルが開始保留中または再開保留中であり、そのサブチャンネルについて開始機能を実行しているときにチャンネル・サブシステムがインターフェース制御チェック条件を検出した場合は、最初のコマンドが受理されたことをチャンネル・サブシステムが確認できない限り、サブチャンネルは開始保留または再開保留状態のままになります。チャンネル・サブシステムが最初のコマンドについて入出力操作を開始しようとしていた場合であっても、そのコマンドが受理されたことを確認できなければ、サブチャンネルは開始保留または再開保留のままになります。チャンネル・サブシステムが、最初のコマンドが受理

されたかどうかを判別できない場合は、サブチャンネルは、少なくとも警報状況と 1 次状況を伴う状況保留にされます。

状況の提示またはデータの転送を目的としてチャンネル・サブシステムとの信号発信シーケンスを開始しているときに、装置が無効なパリティを持つアドレスを提示した場合は、装置および関連のサブチャンネルを識別できないため、このエラー条件はプログラムには提示されません。

インターフェース制御チェックが示されたときは、サブチャンネルに関する測定データは格納されません。

プログラミング上の注意: SCSW の状況制御フィールドが、サブチャンネルが警報状況を伴う状況保留状態にあることを示している、SCSW のフィールド妥当性フラグが、装置状況フィールドをエラー回復の目的に使用できないことを示している場合は、プログラムでは、(1) チャンネル・サブシステムが装置から警報状況を受理しようとしたときにインターフェース制御チェック条件が起きたものと想定して、(2) その警報状況自体は消滅していても、警報状況に対する適切な処置をとるようにしてください。

チェーニング・チェック

チェーニング・チェックは、入力操作でのデータ・チェーニング時のチャンネル・サブシステムのオーバーランが原因で起きます。この条件が起きるのは、入出力データ速度が速すぎてデータ・アドレスの解決ができないときです。出力操作では、チェーニング・チェックが起きることはありません。

チェーニング・チェック条件が検出されると、操作を終結するよう指示する信号が入出力装置に送られます。この条件が起きるとは、コマンド・チェーニングは抑止されます。

カウント・フィールド

ワード 2 のビット位置 16-31 には、残余カウントが入っています。このカウントは、最後の CCW に指定されているオリジナル・カウントと一緒に使用すべきもので、既存の条件 (16-19ページの図16-4を参照) に応じて、その CCW で指定されている記憶域へ (またはその記憶域から) 転送されたバイトの数を示します。カウント・フィールドが意味を持つのは、サブチャンネルが、(1) 装置状況のみから成る 1 次状況を伴う状況保留か、または、(2) 誤長のみ、PCI のみ、またはその両方のサブチャンネル状況が付随した装置状況から成る 1 次状況を伴う情報保留状態にあるときです。

16-29ページの図16-5 は、サブチャンネルが、開始保留、サブチャンネル/装置アクティブ、装置アクティブ、延期、

または状況保留状態にあるときのすべての場合について、カウント・フィールドの内容を示しています。

サブチャンネルの状態 ¹	カウント
開始保留 (UUUU0/AIP SX) ²	意味なし ³
開始保留および状況保留 (10YY1/AIP SX) ²	意味なし ³
すべてのバスで装置が非稼働と見なされたことに起因する開始保留および状況保留 (00111/AIP SX) ²	意味なし ³
開始保留および装置アクティブ (UUUU0/AIP SX) ²	意味なし ³
延期 (YYYYY/AIP SX) ²	意味なし ³
サブチャンネル/装置アクティブ (UUUU0/AIP SX) ²	意味なし ³
装置アクティブ (UUUU0/AIP SX)	意味なし ³
プログラム制御取り込み条件または初期状況割り込みに起因する状況保留 (01001/AIP SX)。	意味なし ³
状況保留 (1Y1Y1/AIP SX)：下記のいずれかが原因で中止が生じた ²	
プログラム・チェック	意味なし ³
記憶保護チェック	意味なし ³
チェーニング・チェック	意味なし ³
チャンネル制御チェック	注 1 を参照
インターフェース制御チェック	意味なし ³
チャンネル・データ・チェック	注 2 を参照
状況保留 (YY1Y1/AIP SX)：カウント制御の下で中止が生じた ²	正しい
状況保留 (Y0011/AIP SX) ²	意味なし ³
状況保留 (1Y1Y1/AIP SX) ²	正しい。最後に使用された CCW の残余カウント
状況保留 (1Y111/AIP SX)：警報状況が原因でコマンド・チェーニングが抑止された ²	正しい。最後に使用された CCW の残余カウント
状況保留 (YYYY1/AIP SX)：HALT SUBCHANNEL による中止後 ²	予測不能
状況保留 (00001/AIP SX)：CLEAR SUBCHANNEL による中止後	意味なし ³
状況保留 (YY1Y1/AIP SX)：サブチャンネルで操作が正常に完了した ²	正しい。残余カウントを示す

図 16-5 (1/2). SCSW 内のカウント・フィールドの内容

サブチャネルの状態 ¹	カウント
状況保留 (1Y111/AIP SX)：警報状況が原因でコマンド・チェーニングが中止された ²	正しい。新しい入出力操作を指定する CCW のオリジナル・カウント
警報状況に起因する状況保留 (10001/AIP SX)	意味なし ³
説明：	
<p>¹ 例えばプログラム・チェックと装置チェックで表される警報状況などが原因で複数の条件が存在する場合は、表の中で最初に記載されている条件が優先されます。</p> <p>この欄で使用されている記号の意味は、以下のとおりです。</p> <p>A 警報状況 I 中間状況 P 1 次状況 S 2 次状況 X 状況保留</p> <p>状況制御ビット設定の可能な組み合わせは、「/」記号の左側に示してあります。</p> <p>ビット設定は以下のとおりです。</p> <p>0 対応する条件は示されません。 1 対応する条件は示されます。 U 予測不能。サブチャネルが状況保留でないときは、対応する条件は意味を持ちません。 Y 対応する条件は重要ではなく、サブチャネルの状態に応じて示されます。</p> <p>² サブチャネルは再開保留状態の場合もあります。</p> <p>³ SCSW が格納され、サブチャネルが記載されている状態にあるときは、カウント・フィールドは無効なので、カウント・フィールドの内容は意味を持ちません。</p>	
注：	
<p>1. IDAW チェックが示されていない場合は、カウントは予測不能です。IDAW チェックが示されている場合は、カウントは、主記憶機構との間で転送されたバイト数を正確に反映していないことがあります。CCW アドレスと一緒にカウントを使用することにより、チャンネル制御チェックが認識されたページ内のバイト位置が分かります。</p> <p>2. 記憶域アクセス時にチャンネル・サブシステムが格納できる最大バイト数は、モデルによって異なります。記憶域アクセスでチャンネル・データ・チェック条件が認識された場合は、チャンネル・サブシステムは、記憶機構との間で転送されたバイト数を検出できないことがあります。したがって、記憶機構との間で転送されたバイト数が正しく残余カウントに反映されないことがあります。ただし、格納された残余カウントは、記憶域アクセス・コードおよび CCW と一緒に使用されて、チャンネル・データ・チェック条件が認識されたページ内のバイト位置を示します。</p>	

図 16-5 (2/2). SCSW 内のカウント・フィールドの内容

拡張状況ワード

拡張状況ワード (ESW) は、サブチャンネルとその関連装置に関する追加情報を、プログラムに提供します。

TEST SUBCHANNEL が実行され、指定されたサブチャンネルが稼働状態にあれば、TEST SUBCHANNEL の第 2 オペランドに指定されている IRB のワード 3-7 に ESW が置かれます。TEST SUBCHANNEL が実行されるたびに、サブチャンネルが、状況保留状態にあるか、または、1 次状況、2 次状況、中間状況、または警報状況を任意の組み合わせで伴う状況保留状態にある場合は (次のパラグラフに示す例外の場合を除く)、ESW は以下のいずれかのタイプの拡張状況フォーマットをとりま

フォーマット 0 ワード 0 にサブチャンネル・ログアウト、ワード 1 に ERW、ワード 2 および 3 に故障記憶機構アドレスまたは 0、そしてワード 4 に 2 次 CCW アドレスが入ります。

フォーマット 1 ワード 0 のバイト 0、2、および 3 に 0、ワード 0 のバイト 1 に LPUM、ワード 1 に ERW、そしてワード 2-4 に 0 が入ります。

フォーマット 2 ワード 0 のバイト 0 に 0、バイト 1 に LPUM、バイト 2 および 3 に装置接続時間、ワード 1 に ERW、そしてワード 2-4 に 0 が入ります。

フォーマット 3 ワード 0 のバイト 0 に 0、バイト 1 に LPUM、バイト 2 および 3 に予測不能の値、ワード 1 に ERW、そしてワード 2-4 に 0 が入ります。

以下のいずれかの条件に該当する場合は、ESW のワード 0-4 の内容は予測不能です。

1. サブチャンネルが状況保留でない。
2. サブチャンネルが状況保留のみで、拡張状況ワード・フォーマット・ビットが 0 である。
3. サブチャンネルが、延期に起因する中間割り込み条件以外の中間状況のみを伴う状況保留状態にある。

格納される拡張状況フォーマットのタイプは、TEST SUBCHANNEL の実行時点でサブチャンネルに存在する条件によって決まります。以下このセクションでは、どのような条件の場合にどのタイプのフォーマットが格納されるかについて説明します。

拡張状況フォーマット 0

TEST SUBCHANNEL により格納される ESW がフォーマット 0 の ESW となるのは、拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) が 1 で、サブチャンネルが、16-35ページの図16-6 で定義されているいずれかの組み合わせの状況を伴う状況保留状態にあるときです。この場合は、サブチャンネル・ログアウト情報および ERW が拡張状況ワードに格納されま

サブチャンネル・ログアウトは、サブチャンネルに関連し、チャンネル・サブシステムが検出した装置エラーを記述するモデル依存の詳細情報を提供します。この情報は、入出力操作、装置、またはその両方の回復に役立ちます。サブチャンネル・ログアウトが提供される場合、エラー条件は、エラーを報告しているサブチャンネルにのみ関係のあるものだけです。他のサブチャンネルも関与している入出力操作がエラー条件の影響を受けている場合は、それらのサブチャンネルも同様のサブチャンネル・ログアウト情報を提供します。拡張報告ワードは、誤動作の原因に関する追加情報を提供します。

フォーマット 0 の ESW のフォーマットは以下のとおりです。

0	Subchannel Logout
1	Extended-Report Word
2	Failing-Storage Address
3	
4	Secondary-CCW Address
0	31

サブチャンネル・ログアウト (Subchannel Logout)

サブチャンネル・ログアウトのフォーマットは以下のとおりです。

0	ESF	LPUM	R	FVF	SA	TC	D	E	A	SC
0	1	8	16	22	24	26				31

拡張状況フラグ (ESF): ビット 1-7 のいずれかが 1 のときは、チャンネル・サブシステムがエラー・チェック条件を検出したことを示します。ESF フィールドには、以下の標識が示されます。

キー・チェック。ビット 1 が 1 のときは、データにアクセスするとき、測定ブロックを更新しようとしているとき、または CCW または IDAW を取り出そうとしているときに、チャンネル・サブシステムが、関連の記憶キーについて無効な検査ブロック・

コード (CBC) を検出したことを示します。キー・エラーのソースは、チャンネル・データ・チェック・ビット (SCSW のワード 2 のビット 12)、測定ブロック・データ・チェック・ビット (ESW のワード 0 のビット 3)、CCW チェック・ビット (ESW のワード 0 のビット 5)、または IDAW チェック・ビット (ESW のワード 0 のビット 6) に示されます。

注: あるキーについて無効な検査ブロック・コードが検出されたが、データ、CCW、または IDAW が事前取り出しの後でまだ使用されていないという場合にも、この条件が示されることがあります。その場合は、故障記憶機構アドレス妥当性ビット (ERW のビット 6) は 1 になります。これは、記憶域ブロック内の該当キーを持つ記憶位置のアドレスが、ESW のワード 2 および 3 に格納されていることを示します。

測定ブロック・プログラム・チェック。 ビット 2 が 1 のときは、測定ブロックを更新しようとしているチャンネル・サブシステムが、このサブチャンネルについて測定ブロック起点と測定ブロック指標を結合するときに、無効な絶対アドレスを検出したことを示します。

測定ブロック・データ・チェック。 ビット 3 が 1 のときは、主記憶機構内の測定ブロックのデータに関する誤動作が検出されたことを示します。(17-3 ページの『測定ブロック』を参照してください。) 測定ブロック・データ・チェックが示されるのは、測定ブロックを更新するときに、測定データを入れる記憶域または関連のキーについて、無効な検査ブロック・コード (CBC) が検出されたときです。関連キーについて無効な CBC が検出されたときは、キー・チェック・ビット (ESF フィールドのビット 1) にも 1 が格納されます。

測定ブロック記憶保護チェック。 ビット 4 が 1 のときは、チャンネル・サブシステムが測定ブロックを更新しようとしたときに、記憶キーが測定ブロック・キーと一致しないために、測定ブロックへのアクセスを禁止されたことを示します (17-3 ページの『測定ブロック』を参照)。測定ブロック更新操作に関連した記憶域アクセスには、SET CHANNEL MONITOR で与えられたキーが使用されます (14-12 ページの『SET CHANNEL MONITOR』を参照)。

注: ビット 2-4 に測定チェック条件のいずれかが示されると、チャンネル・サブシステムは、サブチャンネル測定ブロック更新可能ビットを 0 に設定して、サブチャンネルについて測定データの格納を使用不可

にします (15-3 ページの『測定モード使用可能 (MM)』を参照)。

CCW チェック。 ビット 5 が 1 のときは、CCW またはそれに関連したキーの内容について、無効な CBC が検出されたことを示します。これらの条件のいずれかが検出されると、該当の装置での入出力操作は中止され、サブチャンネルは 1 次状況と警報状況を伴う状況保留になり、SCSW の拡張状況ワード・フォーマット・ビットに 1 が格納され、サブチャンネル状況フィールドにチャンネル制御チェックが示されます。中止のタイプまたは装置から受け取った状況によっては、サブチャンネルは 2 次状況を伴う状況保留にもなります。関連キーについて無効な CBC が検出されたときは、キー・チェック・ビット (ESF フィールドのビット 1) にも 1 が格納されます。

注: 事前取り出しされた CCW について無効な検査ブロック・コードが検出された場合、その CCW がまだ使用されていないとしても、この条件がプログラムに提示されることがあります。その場合は、故障記憶機構アドレス妥当性ビット (ERW のビット 6) は 1 になります。これは、無効 CBC を持つ記憶位置のアドレスが、ESW のワード 2 および 3 に格納されていることを示します。

IDAW チェック。 ビット 6 が 1 のときは、IDAW またはそれに関連したキーの内容について、無効な CBC が検出されたことを示します。これらの条件のいずれかが検出されると、該当の装置での入出力操作は中止され、サブチャンネルは 1 次状況と警報状況を伴う状況保留になり、SCSW の拡張状況ワード・フォーマット・ビットは 1 になり、サブチャンネル状況フィールドにチャンネル制御チェックが示されます。中止のタイプまたは装置から受け取った状況によっては、サブチャンネルは 2 次状況を伴う状況保留にもなります。関連キーについて無効な CBC が検出されたときは、キー・チェック・ビット (ESF フィールドのビット 1) も 1 になります。

注: 事前取り出しされた IDAW の内容について無効な検査ブロック・コードが検出された場合、その IDAW がまだ使用されていないとしても、この条件がプログラムに提示されることがあります。その場合は、故障記憶機構アドレス妥当性ビット (ERW のビット 6) は 1 になります。これは、無効 CBC を持つ記憶位置のアドレスが、ESW のワード 2 および 3 に格納されていることを示します。チャンネル・データ・チェック条件が検出されても、CCW チェック・ビットおよび IDAW チェック・ビットには 1 は格納されません。

予約済み。ビット 7 には 0 が格納されます。

最終使用パス・マスク (LPUM): ビット 8-15 は、チャンネル・サブシステムと装置間の通信または情報転送のための最後に使用されたチャンネル・パスを示します。次のいずれかが起きると、使用中のチャンネル・パスに対応するビットが設定されます。

1. サブチャンネル開始機能の最初のコマンドが装置で受理された (16-13ページの『活動制御 (AC)』を参照)。
2. 装置とチャンネル・サブシステムがアクティブに通信しているときに、実行中のチャンネル・プログラムに対してサブチャンネルが延期機能を実行した。
3. チャンネル・サブシステムが装置から受理した状況が割り込み条件として認識されたか、コマンド・チェーニングを抑止する条件が認識された (16-2ページの『割り込み条件』を参照)。
4. チャンネル・サブシステムがインターフェース制御チェック条件を認識したが (16-27ページの『インターフェース制御チェック』を参照)、現在サブチャンネルにはサブチャンネル・ログアウト情報がない。

LPUM フィールドには、最新の設定値が入っています。このフィールドが有効なのは、ESW にフォーマット 0-3 のいずれかのフォーマットで情報が含まれており (16-31ページの『拡張状況ワード』を参照)、SCSW が格納されている場合です。ESW にサブチャンネル・ログアウト情報が示されているときに、LPUM フィールド妥当性フラグが 0 であれば、LPUM の設定が他のサブチャンネル・ログアウト表示と矛盾していることを意味します。

補助報告 (R): ビット 16 が 1 のときは、システム・コンポーネントの誤動作が起きたこと、そして、その誤動作は、以前に認識されているか、または複数のサブチャンネルの活動に影響を与えたものであることを示します。誤動作が複数のサブチャンネルの活動に影響を与えている場合は、影響を受けたサブチャンネルのうち 1 つを除くすべてについて、補助報告条件が認識されます。このビットが 0 のときは、システム・コンポーネントのこの誤動作が、以前に認識されているものではないことを示します。このビットが意味を持つのは、SCSW のワード 2 のビット位置 12-14 に、チャンネル制御チェック、チャンネル・データ・チェック、またはインターフェース制御チェックが示されている場合のみです。

モデルによっては、補助報告条件を認識する機能がない場合や、必ずしもサブチャンネル活動に影響を与えるすべてのシステム誤動作についてこの条件が認識されない場

合があります。補助報告の認識機能がないときは、ビット 16 は 0 に設定されます。

フィールド妥当性フラグ (FVF): ビット 17-21 は、SCSW または拡張状況ワードの対応するフィールドに格納されている情報の妥当性を示します。妥当性ビットが 1 のときは、対応するフィールドは格納されており、回復のために使用できます。妥当性ビットが 0 のときは、対応するフィールドは使用できません。

このビット妥当性フィールドが意味を持つのは、SCSW に、チャンネル・データ・チェック、チャンネル制御チェック、またはインターフェース制御チェックが示されているときです。これらのチェックがどれも示されていないときは、このフィールド (および終了コード・フィールドとシーケンス・コード・フィールド) は意味を持ちません。さらに、これらのチェックが示されていないときは、最終使用パス・マスク・フィールド、装置状況フィールド、および CCW アドレス・フィールドは、すべて有効です。各ビットの定義は以下のとおりです。

- 17 最終使用パス・マスク
- 18 終了コード
- 19 シーケンス・コード
- 20 装置状況
- 21 CCW アドレス

記憶域アクセス・コード (SA): ビット 22-23 は、エラーが起きたときにチャンネル・サブシステムが行っていた記憶域アクセスのタイプを示します。SA フィールドは、入出力操作の実行中に、データの取り出しまたは格納を目的として行う記憶域アクセスのみに関連しています。このエンコードされたフィールドが意味を持つのは、サブチャンネル状況に、チャンネル・データ・チェック、チャンネル制御チェック、またはインターフェース制御チェックが示されている場合のみです。アクセス・コードの割り当ては以下のとおりです。

- 00 アクセス・タイプは不明
- 01 読み取り
- 10 書き込み
- 11 逆方向読み取り

終了コード (TC): ビット 24 および 25 は、発生した終了のタイプを示します。このエンコードされたフィールドが意味を持つのは、SCSW に、チャンネル・データ・チェック、チャンネル制御チェック、またはインターフェース制御チェックが示されている場合のみです。終了のタイプは以下のとおりです。

- 00 停止信号が出された。
- 01 停止、スタック、または正常終了。
- 10 消去信号が出された。

11 予約済み。

SCSW に少なくとも 1 つのチャンネル・チェックが示されており、終了コード・フィールド妥当性フラグが 0 のときは、どのタイプの終了が装置に通知されたかは、予測不能です。SCSW に複数のチャンネル・チェック条件が示されている場合は、同一または異なる 1 つ以上の終了コードが装置に通知されていることがあります。その場合は、終了コード・フィールド妥当性フラグが 1 であれば、終了コードは、装置に通知された終了のうち最も重大度の高いものを示します。終了コードは、重大度の低いものから、「停止、スタック、または正常終了」(01)、「停止信号が出された」(00)、「消去信号が出された」(10)の順になります。

装置状況チェック (D): 状況検査機能がインストールされている場合に、ビット 26 が 1 であれば、ESW 中のサブチャンネル・ログアウトは、チャンネル・サブシステムが有効な CBC を持つ装置状況を検出したが、チャンネル・サブシステムに状況バイトが提示されたときに不適切な組み合わせのビットが含まれていた結果であることを示しています。装置状況チェック・ビットが 1 のときは、インターフェース制御チェック状況ビットは 1 に設定されます。さらに、サブチャンネル・ログアウト・フィールドのビット 20 に 1 が格納されている場合は、SCSW の装置状況フィールドに、エラーのある状況バイトが格納されています。状況検査機能がインストールされていない場合は、ビット 26 には 0 が格納されません。

2 次エラー (E): ビット 27 が 1 のときは、システム・コンポーネントの誤動作が起きたこと、そして、その誤動作は、サブチャンネルまたは入出力装置の何らかの活動に直接関係していることもそうでないこともあることを示しています。この誤動作の結果、このサブチャンネルおよび関連の入出力装置に関係した活動が影響を受けており、サブチャンネルは、チャンネル制御チェックまたはインターフェース制御チェックを伴う状況保留に設定されています。

入出力エラー警報 (A): ビット 28 が 1 のときは、ESW 中のサブチャンネル・ログアウトが、入出力エラー警報の信号が出された結果であることを示しています。入出力エラー警報信号は、制御装置または入出力装置が、チャンネル・サブシステムに報告する必要がある誤動作を検出したことを示します。これに対して、チャンネル・サブシステムは、消去信号を出し、次のパラグラフで述べる場合を除き、インターフェース制御チェックを設定し、拡張状況フォーマット 0 (ログアウト) 情報を ESW に格納します。

入出力エラー警報信号が出されたときに、サブチャンネルがすでに使用不可に設定されているか、または入出力装

置に関連するサブチャンネルがない場合は、装置に消去信号が出され、チャンネル・サブシステムは入出力エラー警報表示を無視します。

シーケンス・コード (SC): ビット 29-31 は、エラーが起きた時点で進行中だった入出力シーケンスを示します。シーケンス・コードは、START SUBCHANNEL または RESUME SUBCHANNEL の実行により開始された入出力操作のみに関係しています。このエンコードされたフィールドが意味を持つのは、SCSW に、チャンネル・データ・チェック、チャンネル制御チェック、またはインターフェース制御チェックが示されている場合のみです。

シーケンス・コードの割り当ては以下のとおりです。

000 予約済み。

001 チャンネル・サブシステムは、0 以外のコマンド・バイトを送信したが、まだそれに対する応答を分析していない。このコードは開始シーケンスで設定されます。

010 装置はコマンドを受理したが、データはまだ転送されていない。

011 チャンネル・サブシステムと装置の間で、少なくとも 1 バイトのデータが転送された。このコードは、チャンネル・パスがアイドル状態またはポーリング状態にあるときにも使用されることがあります。

100 現 CCW 内のコマンドが、(1) まだ装置に送信されていないか、(2) 送信されたが、まだ装置で受理されていないか、または、(3) 送信され受理されたが、コマンド再試行状況が提示された。このコードが設定されるのは、以下のいずれかの条件に該当する場合です。

1. コマンド・チューニングの時点、または装置での開始機能または再開機能の開始時点で、コマンド・アドレスが更新された。
2. 開始シーケンスにおける状況が、アテンション、制御装置終了、装置チェック、装置例外、使用中、状況修飾 (チャンネル終了および装置終了はない)、または装置終了 (チャンネル終了はない) を示している。
3. コマンド再試行が通知された。
4. チャンネル・サブシステムが、割り込み条件を消去している途中で、装置に問い合わせを出した。
5. チャンネル・サブシステムが、延期機能を実行しているときのコマンド・チューニング時

に、操作の終結を指示する信号を装置に送った。

101 現 CCW 内のコマンドが受理されたが、データが転送されたかどうかは判別できない。このコードが適用されるのは、装置が論理的にチャンネル・パスに接続されてから、新しいシーケンス・コードを適用することが決定される時点までです。このコードは、チャンネル・サブシステムがチャンネル・パスをポーリング状態またはアイドル状態にしたときに、コード 010 と 011 のどちらを適用するかを決定できない場合にも、使用されることがあります。そのほかに、チャンネル・パスがコード 010 と 011 を区別できないときも、このコードが使用されることがあります。

110 予約済み。

111 予約済み。

図16-6 は、サブチャンネル・ログアウト・データとして提供される標識と、それぞれに該当する SCSW ビットとの関係を示しています。

表示されるサブチャンネル・ログアウト条件	SCSW のログアウト条件 ¹		
	CDC	CCC	IFCC
キー・チェック	V	V	-
測定ブロック・プログラム・チェック ²	-	-	-
測定ブロック・データ・チェック ²	-	-	-
測定ブロック保護チェック ²	-	-	-
CCW チェック	-	V	-
IDAW チェック	-	V	-
最終使用パス・マスク ³	V	V	V
フィールド妥当性フラグ	V	V	V
終了コード ³	V	V	V
装置状況チェック	-	-	V
2 次エラー	-	V	V
入出力エラー・警報	-	-	V
シーケンス・コード ³	V	V	V

説明:

- 関係はありません。
- ¹ 複数の SCSW 表示が通知された場合は、関連する各 SCSW 表示の論理和が、有効なサブチャンネル・ログアウト条件となります。
- ² 1つのサブチャンネル・ログアウト内に示される測定ブロック・チェックは1つだけです。
- ³ このフィールドにはフィールド妥当性フラグがあります。

CCC チャンネル制御チェック。
 CDC チャンネル・データ・チェック。
 IFCC インターフェース制御チェック。
 V ビット設定は有効です。

図 16-6. サブチャンネル・ログアウト・データと SCSW ビットの間関係

拡張報告ワード (Extended-Report Word)

拡張報告ワード (ERW) は、装置、サブチャンネル、またはチャンネル・サブシステムに存在している可能性のある特定の条件を表す情報を、プログラムに提供します。拡張状況ワードが格納される時は、常に ERW も格納されます。拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) と、拡張制御ビット (SCSW のワード 0 のビット 14) が両方とも 0 であるときは、ERW にはすべて 0 が入ります。拡張状況ワード・フォーマット・ビットまたは拡張制御ビットの一方または両方が 1 のときは、ERW のフォーマットは以下ようになります。

000	A	P	T	F	S	C	R	SCNT	00000	000000000000	
0	3					8	10		16	21	31

許可検査 (A): ビット 3 が 1 のときは、開始機能または再開機能が中止されたことを示します。中止された理由は、チャンネル・サブシステムが隔離された状態にされたため、保留中の入出力操作は開始されず、現在実行中の入出力操作が中止されようとしているか、またはすでに中止されていることにあります。

要パス検査 (P): ビット 4 が 1 のときは、プログラムが装置の識別を検査する必要があることを示します。LPUM が有効であれば、装置検査を行う必要があるチャンネル・パスは LPUM に示されています。有効な LPUM がない場合は、使用可能な各チャンネル・パスごとに、装置の識別を検査する必要があります。

チャンネル・パス・タイムアウト (T): ビット 5 が 1 のときは、信号発信シーケンスにおいて、事前に設定されている時間内に装置から適切な信号が発信されなかったことを示します。ビット 5 が意味を持つのは、拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5)、およびインターフェース制御チェック・ビット (SCSW のワード 2 のビット 14) が、両方とも 1 のときです。

故障記憶機構アドレス妥当性 (F): ビット 6 が 1 で、拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) も 1 であるときは、チャンネル・サブシステムが、CCW、データ位置、IDAW、またはそれぞれに関連したキーについて、無効な CBC を検出したことを示します。チャンネル・サブシステムは、その無効な CBC に関連した記憶位置の絶対アドレスを、ESW のワード 2 および 3 に格納しています。ビット 6 が 0 に設定された ERW が格納されると、ESW のワード 2 および 3 には 0 が格納されます。

並行センス (S): ビット 7 が 1 のときは、並行センス機能が、装置から受け取ったセンス情報を拡張制御ワードに格納し、拡張制御ワードに格納されているセンス・バイトの数を示す値を、ERW のビット位置 10-15 に格納したことを示します。ビット 7 が 1 のときは、SCSW のワード 0 のビット 14 も 1 です。

並行センス・カウント (SCNT): ビット 7 が 1 のときは、ビット位置 10-15 には、並行センス機能が拡張制御ワードに格納したセンス・バイトの数を示す 1-32 の範囲内の値が入ります。ビット 7 が 0 のときは、ビット位置 10-15 はすべて 0 になります。

2 次 CCW アドレス妥当性 (C): ビット 8 が 1 で、拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) も 1 であるときは、チャンネル・サブシステムが、入出力操作実行の継続を妨げるようなエラー条件を検出したことを示します。事前取り出しが適用される (ORB のワード 1 のビット 9 が 1 である) ときに、チャンネル・サブシステムが、チャンネル制御チェック、チャンネル・データ・チェック、またはインターフェース制御チェックにより識別される特定のエラー条件を認識した場合は、チャンネル・プログラムの実行終了点が、チャンネル・サブシステムと入出力装置の間で異なるような状況が生じることがあります。終了点を適切に識別するために、ビット 8 が 1 に設定され、第 2 CCW アドレス (2 次 CCW アドレス) が ESW に入れられます。このアドレスは、装置で最後に実行された CCW を示します。上記で述べたエラーが原因で妥当性ビットが 0 である場合は、チャンネル・サブシステムは制御装置の実行の終了点を判別できなかったため、2 次 CCW アドレス・フィールドには 0 が入っています。

実行中のチャンネル・プログラムについて、プログラムで事前取り出し制御ビット (ORB のワード 1 のビット 9) を 1 に設定して、事前取り出しを許可しておかないと、ビット 8 は 1 に設定されません。

故障記憶機構アドレス・フォーマット (R): ビット 9 は、故障記憶機構アドレス妥当性ビット (ERW のビット 6) が 1 のときに、故障記憶機構アドレスのフォーマットを示します。ビット 6 が 0 のときは、ビット 9 は意味を持たず、0 が格納されます。ビット 6 が 1 でビット 9 が 0 のときは、フォーマット 1 の故障記憶機構アドレスが、ESW のワード 2 および 3 に格納されます。ビット 6 が 1 でビット 9 が 1 のときは、フォーマット 2 の故障記憶機構アドレスが、ワード 2 および 3 に格納されます。フォーマット 1 およびフォーマット 2 のアドレスの説明については、下記の『故障記憶機構アドレス』を参照してください。

ERW のその他のビットは現在予約済みであり、ERW が格納されるときにこれらのビットには 0 が格納されません。

故障記憶機構アドレス (Failing-Storage Address)

拡張状況のワード 2 および 3 には、24 ビット、31 ビット、または 64 ビットの絶対アドレスが入ります。故障記憶機構アドレス妥当性フラグ (ERW のビット 6) が 1 のときは、ワード 2 および 3 には、フォーマット 1 の故障記憶機構アドレスまたはフォーマット 2 の故障記憶機構アドレスが入ります。ビット 6 が 0 のときは、ワード 2 および 3 には 0 が格納されます。ビット 6 が 1 のときは、故障記憶機構アドレス・フィールドに、CCW、データ記憶位置、IDAW、またはそれぞれに関連したキーについての無効な CBC に関連した、無効な検査ブロック内のバイト位置が示されています。

ワード 2 および 3 に格納されるアドレスの形式は、エラー条件が検出されたときの、フォーマット 2 IDAW 制御 (ORB のワード 1 のビット 14) によって異なります。この制御が 0 (31 ビット・アドレスを含むフルワードの IDAW を使用することを指定する) であるときは、フォーマット 1 のアドレスが格納され、故障記憶機構アドレス・フォーマット・ビット (ERW のビット 9) には 0 が格納されます。フォーマット 2 IDAW 制御が 1 (64 ビット・アドレスを含むダブルワード IDAW を使用することを指定する) であるときは、フォーマット 2 のアドレスが格納され、故障記憶機構アドレス・フォーマット・ビットには 1 が格納されます。

フォーマット 1 のアドレスが格納されるときは、ワード 2 のビット 1-31 が、報告されたエラー条件に関連したアドレスを形成し、ワード 2 のビット 0 とワード 3 の全ビットに 0 が格納されます。フォーマット 2 のアドレスが格納されるときは、ワード 2 のビット 0-31 の後にワード 3 のビット 0-31 を付加したものが、報告されたエラー条件に関連した 64 ビット・アドレスを形成します。

2 次 CCW アドレス (Secondary-CCW Address)

チャンネル状況フィールドが、チャンネル制御チェック、チャンネル・データ・チェック、またはインターフェース制御チェックを示していて、2 次 CCW アドレス妥当性フラグ (ワード 1 のビット 8) が 1 であるときは、ワード 4 のビット 1-31 が、報告された条件によりチャンネル・プログラムが中止された時点で、入出力装置で実行された最後の CCW の絶対アドレスを形成します。2 次

CCW アドレス (提供されている場合) は、回復操作のために利用できます。2 次 CCW アドレス妥当性フラグが 0 のときは、このフィールドは 0 になります。

拡張状況フォーマット 1

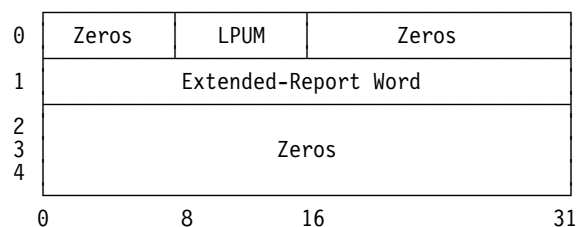
以下の条件がすべて満たされているときは、TEST SUBCHANNEL により格納される ESW はフォーマット 1 の ESW です。

1. 拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) が 0 である。
2. サブチャネル状況制御フィールドで、状況保留ビット (SCSW のワード 0 のビット 31) と、次のいずれかのビットが 1 に設定されている。
 - a. 1 次状況ビット (SCSW のワード 0 のビット 29) のみ。
 - b. 1 次状況ビットおよびその他の状況制御ビット。
 - c. 中間状況ビット (SCSW のワード 0 のビット 28)、および延期ビット (SCSW のワード 0 のビット 26)。
3. 以下の条件の少なくとも 1 つが示されている。
 - a. 装置接続時間測定モードが非アクティブになっている。
 - b. サブチャネルでチャンネル・サブシステム・タイミング機構が使用可能でない。
 - c. サブチャネルが装置接続時間測定モードで稼働できるように設定されていない。

ワード 0 のバイト 0、2、および 3 には 0 が格納され、ワード 0 のバイト 1 には LPUM が格納されます。ワード 1 には ERW、そしてワード 2-4 には 0 が格納されます。

装置接続時間測定モードが非アクティブになるのは、SET CHANNEL MONITOR が実行され、汎用レジスタ 1 のビット 31 が 0 である場合です。

フォーマット 1 の ESW のフォーマットは、以下のとおりです。



最終使用パス・マスク (LPUM): LPUM の定義については、16-33ページの『最終使用パス・マスク (LPUM)』を参照してください。

拡張報告ワード (ERW): ERW の定義については、16-35ページの『拡張報告ワード (Extended-Report Word)』を参照してください。

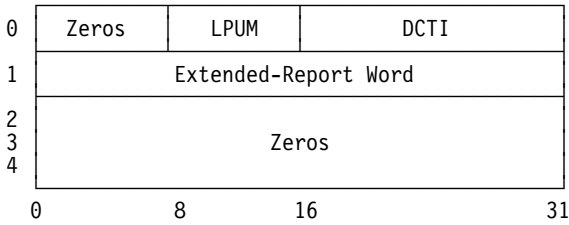
拡張状況フォーマット 2

以下の条件がすべて満たされているときは、TEST SUBCHANNEL により格納される ESW はフォーマット 2 の ESW です。

1. 拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) が 0 である。
2. サブチャネルでチャンネル・サブシステム・タイミング機構が使用可能である。
3. サブチャネルが装置接続時間測定モードで稼働できるように設定されている。
4. 装置接続時間測定モードがアクティブになっている。
5. サブチャネル状況制御フィールドで、状況保留ビット (SCSW のワード 0 のビット 31) と、次のいずれかのビットが 1 に設定されている。
 - a. 1 次状況ビット (SCSW のワード 0 のビット 29) のみ。
 - b. 1 次状況ビットおよびその他の状況制御ビット。
 - c. 中間状況ビット (SCSW のワード 0 のビット 28)、および延期ビット (SCSW のワード 0 のビット 26)。

ワード 0 のバイト 0 には 0 が格納され、ワード 0 のバイト 1 には LPUM が格納され、ワード 0 のバイト 2 および 3 には装置接続時間が格納されます。ワード 1 には ERW、そしてワード 2-4 には 0 が格納されます。

フォーマット 2 の ESW のフォーマットは以下のとおりです。



最終使用パス・マスク (LPUM): LPUM の定義については、16-33ページの『最終使用パス・マスク (LPUM)』を参照してください。

装置接続時間間隔 (DCTI): ビット 16-31 には、チャンネル・サブシステムと入出力装置がアクティブに通信しており、サブチャンネルがサブチャンネル・アクティブにあった間に、チャンネル・サブシステムにより累積された時間増分のカウントを示す 2 進数が入ります。DCTI の時間増分値は 128 マイクロ秒です。

ESW への DCTI 値の格納に関する上記の条件が満たされていても、このサブチャンネルに対して START SUB-CHANNEL の実行後に、SET CHANNEL MONITOR により装置接続時間測定モードがアクティブにされたという場合は、格納される DCTI 値は、0 以上で、かつ正しい DCTI 値以下になります。

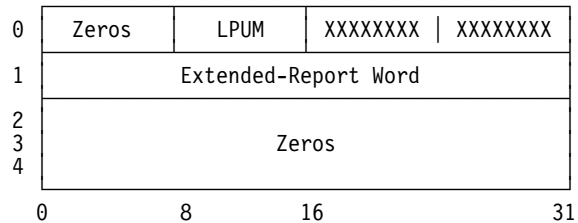
注: サブチャンネルに対して測定ブロック更新モードが使用されている場合は、ESW に格納される DCTI 値は、そのサブチャンネル用の対応する測定ブロック・データを更新するために使用された値と同じです。チャンネル・サブシステムについて測定ブロック・モードがアクティブであり、サブチャンネルが装置接続時間測定モードで稼働できるように設定されているが、ESW に DCTI 値が格納されなかったか (サブチャンネル・ログアウト情報が提示されたため)、または DCTI が 0 であるという場合は、対応する測定ブロック・データには何も加算されません。

拡張報告ワード (ERW): ERW の定義については、16-35ページの『拡張報告ワード (Extended-Report Word)』を参照してください。

拡張状況フォーマット 3

拡張状況ワード・フォーマット・ビット (SCSW のワード 0 のビット 5) が 0 であり、サブチャンネルが、(1) 1 次状況が示されていないときに 2 次状況、警報状況、またはその両方を伴う状況保留状態にあるか、または、(2) サブチャンネルが延期されていないときに中間状況を伴う状況保留状態にある場合は、TEST SUB-CHANNEL により格納される ESW は、フォーマット 3 の ESW です。ワード 0 のバイト 0 には 0 が格納され、ワード 0 のバイト 1 には LPUM が格納されます。ワード 0 の 2 および 3 に入る値は予測不能です。ワード 1 には ERW が格納され、ワード 2-4 には 0 が格納されます。

フォーマット 3 の ESW のフォーマットは以下のとおりです。



最終使用パス・マスク (LPUM): LPUM の定義については、16-33ページの『最終使用パス・マスク (LPUM)』を参照してください。

フォーマットの中の「X」は、該当ビットが 0 または 1 のいずれかの値をとることを示しています。

拡張報告ワード (ERW): ERW の定義については、16-35ページの『拡張報告ワード (Extended-Report Word)』を参照してください。

16-39ページの図16-7 は、サブチャンネルがどのような条件のときに、どのタイプの情報が ESW に格納されるかを示しています。

IRB が格納されるときサブチャネルの条件						拡張状況ワード (ESW) ワード 0	
サブチャネル状況ワード			装置接続時間測定モード	バス管理制御ワード		フォーマット	内容 バイト 0、1、 2、3
状況制御フィールド AIPXS	Lビット	延期ビット		タイミング機構ビット	装置接続時間測定モード可能ビット		
----	0	////	////	////	////	U	****
00001	0	////	////	////	////	U	****
01001	0	1	非アクティブ	////	////	1	ZMZZ
			アクティブ	0	////		
			0	0	////	2	ZMDD
			1	1	0		
**1*1	0	////	非アクティブ	////	////	1	ZMZZ
			アクティブ	0	////		
			0	0	////	2	ZMDD
			1	1	0		
011	0	////	////	////	////	3	ZM
1*001	0	////	////	////	////	3	ZM**
****1	1	////	////	////	////	0	RRRR

説明:

- X が 0 のときは意味がありません。
- * ビットは 0 または 1 です。
- / この場合には関係のない情報。
- A 警報状況。
- D 累算された装置接続時間間隔 (DCTI) 値がバイト 2 および 3 に格納されます。
- I 中間状況。
- L 拡張状況ワード・フォーマット。
- M 最終使用バス・マスク (LPUM) はバイト 1 に格納されます。
- P 1 次状況。
- R サブチャネル・ログアウト情報がワード 0 に格納されません。
- S 2 次状況。
- U フォーマットは定義されていません。
- X 状況保留。
- Z ビットには 0 が格納されます。

図 16-7. ESW に格納される情報

拡張制御ワード

割り込み応答ブロックのワード 8-15 にある拡張制御ワード (16-6 ページの『割り込み応答ブロック』を参照) は、チャンネル・サブシステム、サブチャネル、または装置に存在している可能性がある条件を記述する追加情報を、プログラムに提供します。拡張制御 (E) ビット (SCSW のワード 0 のビット 14) が 1 のときは、モデル依存情報または並行センス情報が拡張制御ワードに格納されていることを示します。

拡張制御ワードが提供する情報は、以下のとおりです。

SCSW ビット 5 14	ERW ビット 7	ERW ビット 10-15	ECW ワード 0-7
0 0	0	0	予測不能 ¹
0 1	0	(5)	(5)
0 1	1	(3)	変更センス情報 ⁴
1 0	0	0	予測不能 ¹
1 1	0	0	モデル依存情報 ²
1 1	1	(3)	並行センス情報 ⁴

説明:

- 1 これらのワードが格納される場合、値は予測不能です。
- 2 モデル依存情報の中の使用されないビットには 0 が格納されます。
- 3 ビット位置 10-15 には、戻されたセンス・バイトの数に等しい値が入ります。
- 4 並行センス情報の中の使用されないビットには 0 が格納されます。
- 5 SCSW ビット 5 が 0、SCSW ビット 14 が 1、ERW ビット 7 が 0 という組み合わせは生じません。

第17章 入出力サポート機能

チャンネル・サブシステム・モニター	17-1	停止信号	17-8
チャンネル・サブシステム・タイミング	17-2	消去信号	17-8
チャンネル・サブシステム・タイマー	17-2	リセット信号	17-9
測定ブロック更新	17-2	リセット	17-9
測定ブロック	17-3	チャンネル・パス・リセット	17-9
測定ブロック起点	17-5	入出力システム・リセット	17-10
測定ブロック・キー	17-5	外部開始機能	17-13
測定ブロック指標	17-5	初期プログラム・ローディング	17-13
測定ブロック更新モード	17-6	入出力システムの再構成	17-16
測定ブロック更新可能	17-6	状況検査	17-16
制御装置キューイング測定	17-6	アドレス限界検査	17-16
制御装置据え置き時間	17-6	構成警報	17-17
装置アクティブ測定	17-6	誤長表示抑止	17-17
時間間隔測定精度	17-7	並行センス	17-17
装置接続時間測定	17-7	チャンネル・サブシステム・リカバリー	17-17
装置接続時間測定モード	17-7	チャンネル報告	17-18
装置接続時間測定可能	17-8	チャンネル報告ワード	17-19
信号およびリセット	17-8	チャンネル・サブシステム入出力優先順位決定機能	17-21
信号	17-8	チャンネル・サブシステム優先順位レベルの 数	17-21

入出力サポート機能は、チャンネル・サブシステムの機能のうち、入出力操作の開始または制御に直接関係のない機能です。この章では、以下の入出力サポート機能について説明します。

- チャンネル・サブシステム・モニター
- 信号およびリセット
- 外部開始機能
- 状況検査
- アドレス限界検査
- 構成警報
- 誤長表示抑止
- 並行センス
- チャンネル・サブシステム回復
- チャンネル・サブシステム入出力優先順位決定機能

チャンネル・サブシステム・モニター

プログラムは、チャンネル・サブシステムが提供するモニター機能を利用して、指定したサブチャンネルのパフォーマンスに関する測定値を検索することができます。モニター機能を使用するには、プログラム制御の下で、SET CHANNEL MONITOR 命令を実行します。さらに、

MODIFY SUBCHANNEL 命令を実行することによって、各サブチャンネルごとに、モニター機能を使用するかどうかを個別に選択することができます。

チャンネル・サブシステム・モニター機能の主要コンポーネントには、チャンネル・サブシステム・タイミング機能、測定ブロック更新機能、および装置接続時間測定機能があります。測定ブロック更新機能と装置接続時間測定機能は、どちらもチャンネル・サブシステム・タイミング機能を使用しますが、その他の点ではそれぞれ別個のものであり、互いに独立して働きます。

チャンネル・サブシステム・モニター機能のその他のコンポーネントには、制御装置キューイング測定機能、制御装置据え置き時間機能、および装置アクティブ時間測定機能があります。使用している制御装置でこれらの機能が使用可能であれば、測定ブロック更新機能の測定能力が拡張されます。

このセクションでは、チャンネル・サブシステム・モニター機能を構成する個々の機能について説明します。

チャンネル・サブシステム・タイミング

チャンネル・サブシステムは、チャンネル・サブシステム・タイミング機能を利用して、START SUBCHANNELにより開始される開始機能の処理に含まれる幾つかの異なるフェーズを実行するために必要な経過時間を測定することができます。このようにして得られる経過時間測定値は、サブチャンネル・パフォーマンス状況をプログラムに提供するために、測定ブロック更新機能と装置接続時間測定機能で使用されます。

どのチャンネル・サブシステムもチャンネル・サブシステム・タイミング機能を備えています。すべてのサブチャンネルについてこの機能が使用できるとは限りません。タイミング機能が提供されているサブチャンネルについては、関連のサブチャンネル情報ブロック内のタイミング機能ビット (ワード 1 のビット 14) に 1 が格納されています。(15-4ページの『タイミング機構 (T)』を参照してください。) チャンネル・サブシステム・タイミング機能が提供されていないサブチャンネルについては、タイミング機能ビットには 0 が格納されています。

チャンネル・サブシステム・タイミング機能が提供されないサブチャンネルとは、関連装置の特性、チャンネル・サブシステムへの装置の接続方法、または装置をサポートするために必要なチャンネル・サブシステム・リソースが原因で、チャンネル・サブシステム・タイミング機能の使用が妨げられるようなサブチャンネルです。

チャンネル・サブシステム・タイミング機能は、少なくとも 1 つのチャンネル・サブシステム・タイマーと、2 つの測定機能で使用される経過時間を計算および記録するために必要な関連ロジックと記憶域から成っています。以下のセクションでは、プログラムにとって重要なチャンネル・サブシステム・タイミング機能の局面について説明します。

チャンネル・サブシステム・タイマー

各チャンネル・サブシステム・タイマーは、プログラムからはアクセスできない 2 進数カウンターです。チャンネル・サブシステム・タイマーは、128 マイクロ秒ごとに最右端ビット位置に 1 が加算される方法で増加します。チャンネル・サブシステム・タイマーの増加により最左端ビット位置からの繰り上がりが生じる場合は、繰り上がりは無視され、カウントは 0 から続けられます。このようなオーバーフローを示す標識は生成されません。

各 CPU が TOD クロックにアクセスするように、各チャンネル・サブシステムは、少なくとも 1 つのチャンネル・サブシステム・タイマーにアクセスします。複数のチャンネル・サブシステム・タイマーがあるときは、それらのタイマー間で同期がとられるので、チャンネル・サブシ

テムのすべてのタイミング機能が単一のタイマーを共用しているのと同じ結果になります。これらのタイマー間の同期化は、何らかの TOD クロックを介して行われるか、またはチャンネル・サブシステムにより独自に行われます。

TOD クロックが同期されていない場合は、モデルとチャンネル・サブシステム・タイマー、そして関連装置が物理的にシステムに接続されている形態によっては、チャンネル・サブシステム・タイミング機能が測定する経過時間は、一部またはすべてのサブチャンネルについて予測不能な値になることがあります。予測不能な値になるのは、別々に構成可能なチャンネル・バスを介して装置がシステムに接続されており、関連の CPU TOD クロックが同期化されていない場合です。

同期化: 測定ブロック更新モードまたは装置接続時間測定モードがアクティブであるときに、いずれかのチャンネル・サブシステム・タイマーの同期がとられていないことが判明すると、チャンネル・サブシステム・タイマー同期チェックが認識され、プログラムに対して警報を出すためのチャンネル報告が生成されます (17-17ページの『チャンネル・サブシステム・リカバリー』を参照)。これらのモードがどちらもアクティブでない場合は、同期が失われている状態は認識されません。

測定ブロック更新

プログラムは、測定ブロック更新機能を使用して、測定ブロック更新モードがアクティブになっているときに、測定ブロック更新モードが使用可能にされているサブチャンネルについて、パフォーマンス情報を累積することができます。サブチャンネルに対して測定ブロック更新モードを使用可能にするには、SCHIB オペランドのワード 1 のビット 11 を 1 に設定して、MODIFY SUBCHANNEL を実行します。測定ブロック更新モードをアクティブにするには、汎用レジスタ 1 のビット 62 が 1 のときに、SET CHANNEL MONITOR を実行します。

測定ブロック更新モードがアクティブで、サブチャンネルで測定ブロック更新モードが使用可能にされているときは、サブチャンネルに関連した測定ブロック内に情報が累積されます。測定ブロックは、測定データを累積するためにサブチャンネルに割り当てられている、主記憶域内の 32 バイトの区域です。プログラムは、測定ブロック域と呼ばれる連続した絶対記憶域を指定し、それを 32 バイトずつのブロックに分割して、測定データを累積する必要がある各サブチャンネルに 1 ブロックずつ割り当てます。測定ブロック更新機能は、サブチャンネルに含まれている測定ブロック指標と、SET CHANNEL MONITOR の実行により設定される測定ブロック起点を使用して、

サブチャンネルに関連した測定ブロックの絶対アドレスを計算します。

START SUBCHANNEL の実行により開始された 1 つまたは一連の操作が延期されるかまたは完了するたびに、サブチャンネルに関連した測定ブロックに測定データが格納されます。1 つまたは一連の入出力操作が完了したことは、通常、1 次割り込み条件により通知されます。測定ブロック更新機能により測定データが累積される測定ブロック内には、7 つのフィールドが定義されています。それは、SSCH+RSCH カウント、サンプル・カウント、装置接続時間、機能保留時間、装置切断時間、制御装置キューイング時間、および装置アクティブ時間です。

測定ブロック

測定ブロックは、プログラムが、測定ブロック起点アドレスと測定ブロック指標を使用して指定する位置にある、32 バイトの区域です。測定ブロックには、以下に述べる測定データの累算値が含まれています。測定ブロック更新モードがアクティブで、サブチャンネルで測定ブロック更新が使用可能にされているときは、測定ブロック更新機能は、START SUBCHANNEL の実行により開始された 1 つまたは一連の入出力操作の実行時に得られた測定データの値を累積します。

1 つまたは一連の入出力操作が延期されたかまたは完了し、エラー条件が検出されなかった場合は、入手した値が、サブチャンネルの測定ブロック内の累算値に加算されます。エラー条件が検出され、拡張状況ワード (ESW) にサブチャンネル・ログアウト情報が格納されている場合は、入手した値は、サブチャンネルの測定ブロック内の累算値に加算されず、測定ブロック内の 2 つのカウント・フィールドは増加しません。

(1) 入手した時間値のいずれかが、その値を入れるために用意されている内部記憶域を超過していることが検出されるか、(2) 制御装置が、現在の操作について正確なキューイング時間または据え置き時間を提示できないか、または、(3) チャンネル・サブシステムが特定のエラー条件から正常に回復した場合は、入手した値はサブチャンネルの測定ブロックに加算されず、測定ブロック内のサンプル・カウントは増加しませんが、ブロック内の SSCH+RSCH カウントは増加します。

入出力機能の延期または完了時に、測定ブロック更新機能が正確なデータを累積するために行う測定ブロック参照は、単一アクセス参照であり、CPU からはワード単位で行われているように見えます。

測定ブロックのフォーマットは以下のとおりです。

ワード 0	SSCH+RSCH Count	Sample Count
1	Device-Connect Time	
2	Function-Pending Time	
3	Device-Disconnect Time	
4	Control-Unit-Queuing Time	
5	Device-Active-Only Time	
6	Reserved	
7	Reserved	
	0	16
		31

SSCH+RSCH カウント (SSCH+RSCH Count): ワード 0 のビット 0-15 は、2 進数カウンターとして使用されます。測定ブロック更新がアクティブになっている開始機能の実行時に、(1) 1 次割り込み条件または 2 次割り込み条件 (いずれか該当する方法) が認識されるか、または、(2) 延期機能が実行されたときは、ビット位置 15 に 1 が加算されてこのカウンターが増加し、測定データが格納されます。カウンターは、最大値の 65,535 に達すると 0 に循環します。カウンター・オーバーフローが起きても、プログラムに対して警報は出されません。

測定ブロック更新モードがアクティブであり、サブチャンネルで測定が使用可能にされているときは、特定の開始機能についての測定値がないために測定ブロックの残りのフィールドの更新ができない場合や、サブチャンネルのタイミング機能ビットが 0 である場合であっても、SSCH+RSCH カウントは増加します。測定ブロック更新モードが非アクティブの場合、サブチャンネルで測定ブロック更新が使用可能にされていない場合、または開始機能についてサブチャンネル・ログアウト情報が生成されている場合は、SSCH+RSCH カウントは増加しません。

サンプル・カウント (Sample Count): ワード 0 のビット 16-31 は、2 進数カウンターとして使用されません。測定ブロックのワード 0 の後に続く各時間累積フィールドが更新され、ビット位置 31 に 1 が加算されてこのカウンターが増加します。一部のモデルでは、測定ブロック更新モードがアクティブであり、サブチャンネルでそのモードが使用可能にされていても、ある種の条件が起きると、測定ブロック更新機能が、個々の開始機能に関する測定データの値を取得できないことがあります。制御装置が、正確なキューイング時間を累算できなかったことを通知することもあります。その場合は、サンプル・カウント・フィールドは増加しません。

カウンターは、最大値の 65,535 に達すると 0 に循環します。カウンター・オーバーフローが起きても、プログラムに対して警報は出されません。サブチャンネルに対してチャンネル・サブシステム・タイミング機能が適用されていない場合は、このフィールドは更新されません。

測定ブロックのサンプル・カウント・フィールドおよび時間累積フィールドの更新を妨げる条件がある場合は、該当のシステム・モデル用のシステム・ライブラリー資料にそれらの条件が示されています。

装置接続時間 (Device-Connect Time): ワード 1 のビット位置 0-31 には、測定された装置接続時間間隔の累算値が入ります。装置接続時間間隔 (DCTI) は、装置が論理的にチャンネル・パスに接続されているときに、サブチャンネルがサブチャンネル・アクティブであり、装置がチャンネル・パスとアクティブに通信している間に測定された合計時間間隔です。装置が論理的にチャンネル・パスに接続されていても、チャンネルとアクティブに通信していない時間は、装置接続時間には含まれません。装置が論理的に接続されていて、チャンネル・パスとアクティブに通信していない時間間隔の累算値は、制御装置据え置き時間として報告されます。制御装置据え置き時間は、装置接続時間の測定値には含まれず、操作の装置切断時間の測定値に累算されます。

時間間隔は、128 マイクロ秒の分解能を使用して測定されます。累算値は、最大約 152.71 時間までであり、オーバーフローが起きてもプログラムに警報は通知されません。(1) サブチャンネルに対してチャンネル・サブシステム・タイミング機能が提供されていないか、(2) 測定ブロック更新モードが非アクティブであるか、または、(3) 現在の開始機能について入手した時間値のいずれかが該当の内部記憶域を超過していることが検出された場合は、このフィールドは更新されません。

サブチャンネルについて装置接続時間間隔が累積されるかどうか、およびそのデータが ESW に格納されるかどうかは、測定ブロック更新モードがアクティブかどうかには関係ありません。(17-7ページの『装置接続時間測定』を参照してください。)

機能保留時間 (Function-Pending Time): ワード 2 のビット位置 0-31 には、SSCH 機能保留時間および RSCH 機能保留時間の累算値が入ります。機能保留時間は、開始機能 (サブチャンネルが延期状態にあるときは再開機能) がサブチャンネルで受理されてから、チャンネル・プログラムの実行の開始または再開に関連した最初のコマンドが装置で受理されるまでの時間間隔です。

チャンネル・プログラムの最初の CCW 内の延期フラグが原因でチャンネル・プログラムの実行が延期される場合

は、最初のコマンドが装置に転送される前に延期が起きます。その場合は、その時点までに累算された機能保留時間が、測定ブロックの機能保留時間フィールド内の値に累算されます。サブチャンネルが延期されているときは、機能保留時間は発生しません。その場合は、指定したサブチャンネルが延期状態にあるときに RESUME SUBCHANNEL が実行されると、機能保留時間の測定が再び開始されます。

機能保留時間間隔は、128 マイクロ秒の分解能を使用して測定されます。累算値は、最大約 152.71 時間までであり、オーバーフローが起きてもプログラムに警報は通知されません。サブチャンネルに対してチャンネル・サブシステム・タイミング機能が適用されていない場合は、このフィールドは更新されません。

装置切断時間 (Device-Disconnect Time): ワード 3 のビット位置 0-31 には、装置切断時間の累算値が入ります。装置切断時間は、サブチャンネルがサブチャンネル・アクティブのときに、装置がチャンネル・サブシステムから論理的に切断されている時間間隔の合計です。装置切断時間には、入出力操作中に装置から報告される制御装置据え置き時間間隔の合計値も含まれます。

サブチャンネルが延期状態にあるときは、装置切断時間は発生しません。その場合は、装置でチャンネル・プログラムの実行が再開されたあとで (サブチャンネルは再びサブチャンネル・アクティブになる)、最初の装置切断が起きると、再び装置切断時間の測定が開始されます。

装置切断時間間隔は、128 マイクロ秒の分解能を使用して測定されます。累算値は、最大約 152.71 時間までであり、オーバーフローが起きてもプログラムに警報は通知されません。サブチャンネルに対してチャンネル・サブシステム・タイミング機能が適用されていない場合は、このフィールドは更新されません。

サブチャンネルがサブチャンネル・アクティブではなく、入出力装置がアクティブであるとき、入出力装置の終わりの 1 次状況条件から 2 次状況条件までの間の時間間隔は、装置切断時間は含まれません。その時間間隔は、測定ブロックの装置アクティブ時間フィールドに累積されます。

制御装置キューイング時間 (Control-Unit-Queuing Time): ワード 4 のビット位置 0-31 には、制御装置キューイング時間の累算値が入ります。制御装置キューイング時間は、別のシステムで開始された操作により装置が使用中状態にあるときに、入出力操作中に装置がチャンネル・サブシステムから論理的に切断されている間に制御装置により測定された時間間隔の合計です。

サブチャンネルが延期状態にある間は、制御装置キューイング時間は発生しません。再開が成功し、サブチャンネルがサブチャンネル・アクティブになった後は、チャンネル・プログラムにとって制御装置キューイング時間が発生することがあります。

制御装置キューイング時間フィールドは、ビット 31 が 128 マイクロ秒を指定しているものとして更新されません。累算値は、最大約 152.71 時間までであり、オーバーフローが起きてもプログラムに警報は通知されません。サブチャンネルに対してチャンネル・サブシステム・タイミング機能が提供されていないか、制御装置がキューイング時間を提示しない場合は、このフィールドは更新されません。

装置アクティブ時間 (Device-Active-Only Time):

ワード 5 のビット位置 0-31 には、装置アクティブ時間の累算値が入ります。装置アクティブ時間は、開始機能または再開機能により開始された 1 つまたは一連の入出力操作の終わりの時点で、サブチャンネルが装置アクティブではあるが、サブチャンネル・アクティブではない時間の合計です。

サブチャンネルがアクティブである期間中のサブチャンネルが装置アクティブである間は、装置アクティブ時間は累算されません。このような時間は、装置接続時間または装置切断時間 (いずれか該当する方) として累算されません。

装置アクティブ時間フィールドは、ビット 31 が 128 マイクロ秒を指定しているものとして更新されます。累算値は、最大約 152.71 時間までであり、オーバーフローが起きてもプログラムに警報は通知されません。サブチャンネルに対してチャンネル・サブシステム・タイミング機能が適用されていない場合は、このフィールドは更新されません。

制御装置据え置き時間 (Control-Unit-Defer Time):

制御装置据え置き時間は、入出力操作で装置がチャンネル・サブシステムに論理的に接続されているが、チャンネル・プログラムの実行で装置依存の遅延が原因で、装置がチャンネルとアクティブに通信していない時間間隔の合計です。制御装置据え置き時間は、独立した測定フィールドとして測定ブロック内に格納されるのではなく、操作の装置接続時間測定値および装置切断時間測定値を計算するために使用されます。

制御装置が制御装置据え置き時間を提示する場合、その値は装置がチャンネルに論理的に接続している間に取得されます。装置切断を引き起こすかまたは入出力操作を中止させるようなチャンネル終了状況が提示されると、制御装置据え置き時間がチャンネルに報告されます。制御装置

据え置き時間は、操作について報告されている装置接続時間測定値から減算され、装置切断時間測定値に加算されます。

予約済み (Reserved): 測定ブロックの上記以外のワードと、チャンネル・サブシステムまたはサブチャンネルが提供していない機能に関連したワードは、将来の使用に備えて予約済みです。これらのワードは、測定ブロック更新機能により更新されません。

測定ブロック起点

測定ブロック起点は、主記憶機構内の測定ブロック域の始めです。汎用レジスター 1 のビット 62 が 1 に設定されている状態で SET CHANNEL MONITOR が実行されると、測定ブロック起点の絶対アドレス (32 バイト境界上に指定される) が、汎用レジスター 2 に入れて測定ブロック更新機能に渡されます。

測定ブロック・キー

汎用レジスター 1 のビット 32-35 は、SET CHANNEL MONITOR により測定ブロック更新モードがアクティブにされたときに以後の測定ブロック更新のために使用される、4 ビットのアクセス・キーを形成します。測定ブロック・キーは、測定ブロック起点が渡されるたびに、測定ブロック更新機能に渡されます。

測定ブロック指標

測定ブロック指標は、MODIFY SUBCHANNEL の実行によりサブチャンネル内に設定されます。測定ブロック指標は、サブチャンネルの測定ブロック・パラメーターを累算するために、測定ブロック起点を基準としてどの位置にある 32 バイトの測定ブロックを使用するかを指定します。測定ブロック更新機能は、サブチャンネルの測定ブロック指標の右側に 5 個の 0 を付加し、その結果を測定ブロック起点に加算することにより、測定ブロックの位置を計算します。その結果が、そのサブチャンネル用の 32 バイトの測定ブロックの絶対アドレスです。計算結果の測定ブロック・アドレスが $2^{31} - 1$ を超えている場合は、測定ブロック・プログラム・チェック条件が認識され、それ以前のサブチャンネル・アクティブ期間については、測定ブロック更新は行われません。

プログラミング上の注意: 測定ブロック指標の初期値は 0 です。サブチャンネルで測定ブロック更新モードを使用可能にし、そのモードをアクティブにする前に、測定ブロック指標を適切な値に設定するのは、プログラム側で行う必要があります。測定ブロック内の測定対象パラメーターについて、確実に予測可能な結果が得られるようにするためには、測定対象パラメーターが累算される各サブチャンネルの測定ブロック指標が、それぞれ異なる値を持っていることが必要です。

測定ブロック更新モード

測定ブロック更新モードをアクティブにするには、汎用レジスター 1 のビット 62 を 1 に設定して、SET CHANNEL MONITOR を実行します。SET CHANNEL MONITOR を実行するときに、汎用レジスター 1 のビット 62 が 0 であれば、このモードは非アクティブにされます。測定ブロック更新モードが非アクティブのときは、主記憶機構内にどの測定値も累積されません。測定ブロック更新モードがアクティブにされると、汎用レジスター 2 の内容が、測定ブロック起点の絶対アドレスとして、測定ブロック更新機能に渡されます。また、MBK が、各サブチャンネルの測定ブロックを更新するときに使用するアクセス・キーとして、測定ブロック更新機能に渡されます。測定ブロック更新モードがアクティブのときは、測定ブロック更新機能は、測定ブロック更新可能ビットが 1 であるサブチャンネルについて、測定ブロック域の中の個々の測定ブロック内に測定値を累算します。(測定対象パラメーターの説明については、17-3 ページの『測定ブロック』を参照してください。)

SET CHANNEL MONITOR を実行するときに、すでに測定ブロック更新モードがアクティブになっている場合は、測定ブロック更新機能による測定が可能にされているサブチャンネルに使用される測定ブロック起点および測定ブロック・キーの値は、SET CHANNEL MONITOR が、そのサブチャンネルに対する START SUBCHANNEL の実行前、実行時、または実行後のどの時点で実行されるかにより異なります。SET CHANNEL MONITOR が START SUBCHANNEL の前に実行される場合は、現在の測定ブロック起点および測定ブロック・キーが制御権を持ちます。SET CHANNEL MONITOR が START SUBCHANNEL の実行時または実行後に実行される場合は、制御権を持つ測定ブロック起点および測定ブロック・キーが前のものか現在のものかは、予測不能です。

測定ブロック更新可能

SCHIB のワード 1 のビット 11 は、測定ブロック更新可能ビットです。このビットを使用して、測定ブロック・パラメーターの累算をサブチャンネル単位で制御することができます。この測定ブロック更新可能ビットの初期値は 0 です。SCHIB 内で更新可能ビットを 1 に設定して MODIFY SUBCHANNEL を実行すると、サブチャンネルで測定ブロック更新モードが使用可能になります。測定ブロック更新モードがアクティブの場合は、測定ブロック更新機能は、該当のサブチャンネルに対して次の START SUBCHANNEL が出された時点から、そのサブチャンネルについて測定ブロック・パラメーターの累算を開始します。これに対して、プログラムで SCHIB

オペランドのワード 1 のビット 11 を 0 に設定して、MODIFY SUBCHANNEL を実行した場合は、サブチャンネルでは測定ブロック更新モードは使用不可にされ、そのサブチャンネルについては、それ以上測定ブロック・パラメーターの累算は行われません。

制御装置キューイング測定

制御装置キューイング測定機能を使用すると、チャンネル・サブシステムは、制御装置から提示されたキューイング時間を受理し、測定ブロック更新機能と共同して、測定ブロック内でキューイング時間を累算することができます。

各制御装置モデルがキューイング時間を提示する能力については、該当モデルに関するシステム・ライブラリー資料を参照してください。キューイング時間を提示する機能を備えている制御装置モデルについては、どのような条件が起きたときに、制御装置が正確な制御装置キューイング時間を累算できなくなるかが、資料に示されています。

制御装置据え置き時間

制御装置据え置き時間機能を使用すると、チャンネル・サブシステムは、制御装置が提示する据え置き時間を受理し、測定ブロック更新機能と共同して、据え置き時間を反映して、報告された装置接続時間および装置切断時間を測定ブロック内で変更します。制御装置据え置き時間は、操作について報告されている装置接続時間測定値から減算され、装置切断時間測定値に加算されます。

各制御装置モデルが据え置き時間を提示する能力については、該当モデルに関するシステム・ライブラリー資料を参照してください。据え置き時間を提示する機能を備えている制御装置モデルについては、どのような条件が起きたときに、制御装置が正確な制御装置据え置き時間を累算できないかが、資料に示されています。

装置アクティブ測定

装置アクティブ測定機能を使用すると、チャンネル・サブシステムは、1 つまたは一連の入出力操作の終わりに、1 次状況と 2 次状況の間で装置が切断されていた時間を報告することができます。装置アクティブ時間は、32 バイトの測定ブロックのワード 5 に累算されます。この時間は、この機能を使用しなかった場合には、測定データで表すことはできません。

測定ブロック更新は、入出力装置からの 2 次状況が受理された時点で行われます。これは、1 次状況と 2 次状況の間の装置アクティブ時間を報告できるようにするためです。

入出力装置からの 2 次状況を受理した時点でサブチャンネルが開始保留状態にあり、測定ブロック更新を行う必要がある場合は、開始機能の実行前に測定ブロック更新が行われます。測定ブロック・エラーが起きた場合は、開始機能は実行されずに、2 次状況と共にそのエラーがプログラムに報告されます。

時間間隔測定精度

一部のモデルでは、時間間隔を測定することになっているときに、START SUBCHANNEL (延期されているサブチャンネルの場合は RESUME SUBCHANNEL) で条件コード 0 が設定された場合は、機能保留時間測定が開始される前に待ち時間が生じることがあります。該当のシステム・モデルに関するシステム・ライブラリー資料に、測定される各時間間隔の平均待ち時間と分散値が示されています。

プログラミング上の注意:

1. チャンネル・サブシステムが測定データを更新しようとしているときに、同時にプログラムが同じ測定ブロック域にアクセスしていると、過剰な遅延が生じることがあります。プログラミングの際には、測定ブロック更新モードがアクティブのときは、SET CHANNEL MONITOR で指定する記憶域ブロックが読み取り専用になるような手段を講じてください。
2. 測定機能をサポートするように書かれたプログラムを適切に実行するためには、そのプログラムで、測定ブロック更新モードをアクティブにする前に、すべての測定ブロックを 0 に初期化してください。測定ブロックの予約済みまたは使用されないワードは、すべて 0 でなければなりません。
3. 累算値が増加した結果、ビット位置 0 から波及的な繰り上がりが生じる場合は、繰り上がりは無視され、累算は 0 から続けられます。

装置接続時間測定

装置接続時間測定機能を使用すると、プログラムは、チャンネル・プログラムの実行中に、装置がチャンネル・サブシステムとアクティブに通信している時間の長さを検索することができます。チャンネル・プログラムの実行中に、装置がチャンネル・パス上でアクティブに通信するために費やした時間の測定値を、装置接続時間間隔 (DCTI) と呼びます。制御装置据え置き時間は、DCTI の中に含まれません。

該当のサブチャンネルについてタイミング機構が使用可能な場合は、操作の完了時に、TEST SUBCHANNEL が 1 次状況条件を消去したとき、および中間状況条件のみ

(サブチャンネルが延期されている場合) を消去したときに、DCTI 値が拡張状況ワード (ESW) に入れてプログラムに渡されます。ESW に入れて渡される DCTI 値は、前のサブチャンネル・アクティブ期間に関連するものです。ESW を使用した DCTI の受け渡しは、プログラムが、SET CHANNEL MONITOR で装置接続時間測定モード制御ビットと、サブチャンネル内のそれに対応する使用可能ビットを使用して制御します。ただし、サブチャンネル・ログアウトに記述されるエラー条件が原因で、START SUBCHANNEL の実行により開始された入出力機能が中止された場合は、DCTI 値は ESW に格納されません (16-31 ページの『拡張状況フォーマット 0』を参照)。その場合は、SCSW の拡張状況ビット (L) に 1 が格納されます。これは、エラー条件を記述するログアウト情報が ESW に含まれていることを示します。ログアウト情報の説明については、16-31 ページの『拡張状況ワード』を参照してください。前のサブチャンネル・アクティブ期間について取得された DCTI 値が 8.388608 秒を超えている場合は、最大値 (16 進数 FFFF) が ESW に入れて渡されます。

装置接続時間測定モード

装置接続時間測定モードをアクティブにするには、汎用レジスター 1 のビット 63 を 1 に設定して、SET CHANNEL MONITOR を実行します。SET CHANNEL MONITOR を実行するときに、汎用レジスター 1 のビット 63 が 0 であれば、このモードは非アクティブにされます。該当のサブチャンネルについてタイミング機構が使用可能であり、装置接続時間測定モードがアクティブであり、サブチャンネルでこのモードが使用可能にされているときは、TEST SUBCHANNEL が、(1) SCSW 内にログアウト情報が示されていない (拡張状況ワード・フォーマット・ビットが 0 になっている) 状態で、1 次割り込み条件を消去したとき、または、(2) サブチャンネルが延期されている状態で拡張状況条件のみを消去したときに、格納される ESW に入れて DCTI 値がプログラムに渡されます。

サブチャンネルで装置接続時間測定モードが使用可能にされている状態で、現在開始機能が実行されているときに、SET CHANNEL MONITOR がチャンネル・サブシステムに対してこのモードをアクティブにした場合は、適切な条件下で格納される DCTI の値は、モデルと、このモードがアクティブにされた時点での開始機能の進行状況に応じて、0 か、部分結果か、または完全に正しい値になります。

測定ブロック域に DCTI 値が格納されるかどうかは、装置接続時間測定モードがアクティブであるかどうかには関係ありません。

装置接続時間測定可能

SCHIB のワード 1 のビット 12 は、装置接続時間測定モード使用可能ビットです。プログラムは、このビットを使用することにより、装置接続時間測定モードがアクティブのときに、特定のサブチャンネルについて DCTI 値を格納するかどうかを、選択的に制御することができます。この使用可能ビットの初期値は 0 です。SCHIB 内でこの使用可能ビットが 1 のときに MODIFY SUBCHANNEL が実行されると、サブチャンネルで装置接続時間測定モードが使用可能にされます。装置接続時間測定モードがアクティブの場合は、装置接続時間測定機能は、該当のサブチャンネルに対して次の START SUBCHANNEL が出された時点から、そのサブチャンネルについて DCTI 値の提供を開始します。その場合、DCTI 値は ESW に入れて提示されます (16-37 ページの『拡張状況フォーマット 2』を参照)。同様に、プログラムで SCHIB オペランドのワード 1 のビット 12 を 0 に設定して、MODIFY SUBCHANNEL を実行した場合は、サブチャンネルでは装置接続時間測定モードは使用不可にされ、そのサブチャンネルについては、それ以上 DCTI 値はプログラムに渡されません。

信号およびリセット

システム操作時に、入出力操作を中止したり、入出力システム自体または入出力システムの一部をリセットしたりすることが必要になることがあります。(入出力システムは、チャンネル・サブシステムと、それに接続されているすべての制御装置および入出力装置から成っています。) その目的のために、さまざまな信号およびリセット機能が用意されています。信号は 3 つあり、チャンネル・サブシステムはこれらの信号を使用して、操作の中止、リセット機能の実行、またはその両方を入出力装置に通知します。リセットは 2 つあり、チャンネル・サブシステムはこれらのリセットに従って、入出力装置またはチャンネル・サブシステムに含まれている所定の情報を再初期化します。

信号

チャンネル・サブシステムに信号送信シーケンスを開始させる要求は、次のいずれかにより出されます。

1. プログラムによる CLEAR SUBCHANNEL、HALT SUBCHANNEL、または RESET CHANNEL PATH 命令の実行。
2. 入出力装置からの入出力エラー警報の信号送信。
3. チャンネル・サブシステム自体 (特定のエラー条件または装置誤動作を検出したとき)。

3 つの信号とは、停止信号、消去信号、およびリセット信号です。

停止信号

停止信号は、チャンネル・サブシステムが入出力操作を中止するために使用できます。チャンネル・サブシステムは、HALT SUBCHANNEL の実行に続いて実行される停止機能の一部として、停止信号を発信します。また、ある種のエラーが検出された場合も、チャンネル・サブシステムは停止信号を発信します。

並列入出力インターフェース・タイプのチャンネル・パスの場合は、停止信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974) の中で定義されているインターフェース切断シーケンス制御を使用します。

ESCON 入出力インターフェース・タイプのチャンネル・パスの場合は、停止信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「*IBM Enterprise Systems Architecture/390 ESCON I/O Interface*」(SA22-7202) の中で定義されている取り消し機能を使用します。

FICON 入出力インターフェース・タイプまたは FICON 変換入出力インターフェース・タイプのチャンネル・パスの場合は、停止信号が出されると、チャンネル・サブシステムは、ANSI 標準文書「*Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)*」の中で定義されている取り消し機能を使用します。

消去信号

チャンネル・サブシステムは、消去信号を使用して、入出力操作を中止し、装置に含まれている状況情報と制御情報をリセットすることができます。消去信号は、CLEAR SUBCHANNEL の実行に続いて実行される消去機能の一部として発信されます。また、入出力装置またはチャンネル・サブシステムが特定のエラー条件または装置誤動作を検出したときも、チャンネル・サブシステムは消去信号を発信します。

並列入出力インターフェース・タイプのチャンネル・パスの場合は、消去信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974) の中で定義されている選択的リセット・シーケンス制御を使用します。

ESCON 入出力インターフェース・タイプのチャンネル・パスの場合は、消去信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「*IBM Enter-*

prise Systems Architecture/390 ESCON I/O Interface」(SA22-7202)の中で定義されている選択的リセット機能を使用します。

FICON 入出力インターフェース・タイプまたは FICON 変換入出力インターフェース・タイプのチャンネル・パスの場合は、消去信号が出されると、チャンネル・サブシステムは、ANSI 標準文書「Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)」の中で定義されている選択的リセット機能を使用します。

装置で入出力操作が進行中であり、装置がその入出力操作の実行のためにチャンネル・パスを介してアクティブに通信しているときに、そのチャンネル・パス上で消去信号が受信された場合は、その受信と同時に、装置はそのチャンネル・パスから切断されます。データ転送のほか、制御装置の機能を使用する操作はすべて即時に終了します。入出力装置は、必ずしもブロックの始めに位置付けられるとは限りません。制御装置を使用しない機械的な動作(磁気テープの巻き戻しやディスク・アクセス機構の位置付けなど)は、可能であれば、通常の停止位置に達するまで続けられます。装置は、機械的な動作または固有の操作サイクル(もしあれば)が終了するまで、使用中状態のように見えることがありますが、それが終われば使用可能な状態になります。入出力装置および制御装置内の状況情報はリセットされますが、機械的動作が完了したときに割り込み条件が生成されることがあります。

リセット信号

チャンネル・サブシステムは、リセット信号を使用して、1つのチャンネル・パス上のすべての入出力装置をリセットすることができます。チャンネル・サブシステムは、RESET CHANNEL の実行に続いて実行されるチャンネル・パス・リセット機能の一部として、停止信号を発信します。また、チャンネル・サブシステムは、入出力システム・リセット機能の一部としてもリセット信号を発信します。

並列入出力インターフェース・タイプのチャンネル・パスの場合は、リセット信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI」(GA22-6974)の中で定義されているシステム・リセット・シーケンス制御を使用します。

ESCON 入出力インターフェース・タイプのチャンネル・パスの場合は、リセット信号が出されると、チャンネル・サブシステムは、システム・ライブラリー資料「IBM Enterprise Systems Architecture/390 ESCON I/O Interface」(SA22-7202)の中で定義されているシステム・リセット機能を使用します。

FICON 入出力インターフェース・タイプまたは FICON 変換入出力インターフェース・タイプのチャンネル・パスの場合は、リセット信号が出されると、チャンネル・サブシステムは、ANSI 標準文書「Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)」の中で定義されているシステム・リセット機能を使用します。

リセット

リセット機能は2つあり、チャンネル・サブシステムはこれらのリセット機能を使用して、入出力装置またはチャンネル・サブシステムに含まれている所定の情報を初期化し直すことができます。チャンネル・サブシステムにリセット機能の1つを開始させる要求は、次のいずれかにより出されます。

1. プログラムによる RESET CHANNEL PATH 命令の発行。
2. オペレーターによる、システム・リセット消去キーまたはシステム・リセット通常キー、あるいはロード消去キーまたはロード通常キーの活動化。
3. チャンネル・サブシステム自体(特定のエラー条件または装置誤動作を検出したとき)。

リセット機能には、チャンネル・パス・リセットと入出力システム・リセットがあります。

チャンネル・パス・リセット

チャンネル・パス・リセット機能は、すべての関連サブチャンネルにおいて、指定したチャンネル・パスに関連した特定の標識をリセットする手段を提供します。チャンネル・パス・リセットが行われるのは、チャンネル・サブシステムが、RESET CHANNEL PATH により開始されたチャンネル・パス・リセット機能を実行した時点です。

(14-8ページの『RESET CHANNEL PATH』を参照してください。)すべてのサブチャンネルで、指定したチャンネル・パスに関連した専用従属関係、制御装置使用中、および入出力装置使用中の内部標識がすべて消去され、そのチャンネル・パス上でリセット信号が送信されます。そのチャンネル・パスに接続されている制御装置がリセット信号を受信すると、その制御装置およびそれに接続されている入出力装置で進行中のすべての操作、すべての状況、モード設定、および従属関係(そのチャンネル・パスに関連したもの)が、リセットされます。(17-10ページの『入出力システム・リセット』の中のシステム・リセット信号アクションの説明も参照してください。)

指定したチャンネル・パスに対するチャンネル・パス・リセット機能の実行結果は、チャンネル・サブシステムが生成するマシン・チェック割り込み条件を使用してプログラ

ムに通知されます (17-17ページの『チャンネル・サブシステム・リカバリー』を参照)。

入出力システム・リセット

入出力システム・リセット機能が実行されるのは、チャンネル・サブシステムの電源が入れられたとき、初期プログラム・ローディングが手動で開始されたとき (17-13ページの『初期プログラム・ローディング』を参照)、および、システム・リセット消去キーまたはシステム・リセット通常キーが活性化されたときです。入出力システム・リセット機能は手動で開始すべきものであり、プログラム制御の下でこの機能を開始することはできません。チャンネル・サブシステムまたはチャンネル・パス上で誤動作が検出されると、入出力システム・リセットは完了できないことがあります。入出力システム・リセットは、サブシステム・リセットの一環として行われます。サブシステム・リセットでは、保留中の入出力割り込みも含めてすべての浮動割り込み要求もリセットされます。(4-44ページの『サブシステム・リセット』を参照してください。) 使用システムの各種コンポーネントに対する入出力システム・リセットの影響については、この章の後の方で詳しく説明します。

入出力システム・リセットは、チャンネル・サブシステムとそれに接続している入出力装置を初期化された状態に戻すための手段です。入出力システム・リセットの影響を受けるのは、リセットが実行されるチャンネル・サブシステム構成のみですが、これには、そのチャンネル・サブシステムに対して構成されているすべてのチャンネル・サブシステム・コンポーネントが含まれます。入出力システム・リセットは、リセットするチャンネル・サブシステム構成に含まれていないシステム・コンポーネントには、影響を与えません。目的のチャンネル・サブシステムの構成済みコンポーネントに対する入出力システム・リセットの影響については、以下のセクションで説明します。

チャンネル・サブシステムの状態: 入出力システム・リセットによって、チャンネル・サブシステムは初期化された状態になり、すべてのチャンネル・サブシステム・コンポーネントが以下のセクションで述べる状態になります。進行中の操作はすべて中止されてリセットされ、前の条件の標識はすべてリセットされます。これらの標識には、状況情報、割り込み条件 (保留中の割り込みは除く)、専用従属関係条件、保留中のチャンネル報告、および、前の条件と操作に関するすべての内部情報が含まれます。初期化された状態のチャンネル・サブシステムは、進行中の活動がなく、初期プログラム・ローディング (IPL) を行うこと、または入出力命令 (第14章、『入出力命令』を参照) に応答することが可能な状態になっています。

制御装置と入出力装置: 入出力システム・リセットを行うと、構成に含まれるすべてのチャンネル・パス上にリセット信号が送信されます。これには、前に検出されている永続エラーが原因で物理的に使用可能ではなくなっているチャンネル・パス (PAM ビットが 0 に設定されているもの) も含まれます。制御装置がリセット信号を受信すると、リセットされるチャンネル・パスについて、進行中の制御装置機能、制御装置状況、制御装置従属関係、および制御装置モードがリセットされます。また、リセットするチャンネル・パスについて、入出力装置で進行中の操作、装置状況、装置従属関係、および装置モードもリセットされます。他のチャンネル・パスに関連した制御装置モードと入出力装置モード、従属関係、状況、および進行中の入出力機能は、影響を受けません。

シングル・パス・モードで稼働している装置の場合は、操作は 1 つのチャンネル・パス上で進行中になるだけです。したがって、そのチャンネル・パスでリセット信号が受信されれば、その進行中の操作はリセットされます。しかし、動的再接続機構を持ち、マルチパス・モードで稼働している装置は、入出力操作時に、1 グループのチャンネル・パスに対する従属関係を確立することができ、その場合は、そのパス・グループのすべてのチャンネル・パスが同じチャンネル・サブシステムに属するものとして構成されます。マルチパス・モードで稼働している装置に対する操作が進行中のときに、パス・グループ内のチャンネル・パスの 1 つでリセット信号が受信された場合は、そのリセットされるチャンネル・パスについてのみ、進行中の操作がリセットされます。進行中の操作は、リセットされるチャンネル・パスでは続行できなくなりますが、パス・グループ内の他のチャンネル・パスでは、以下の制約条件の下で続行が可能です。

1. 装置が 1 つのチャンネル・パス上でリセット信号を受信したときに、装置がそのチャンネル・パス上でアクティブにチャンネル・サブシステムと通信していた場合は、パス・グループに関係なく、操作は無条件にリセットされます。
2. マルチパス・モードで操作が進行中であっても、パス・グループに含まれているのがリセットしようとしているパスのみである場合は、操作はリセットされます。
3. 上記の 2 で述べた場合を除き、進行中の操作が、現在切断状態にある (装置がチャンネル・サブシステムとアクティブに通信していない) 場合、またはパス・グループ内の他のチャンネル・パス上でアクティブになっている場合は、システム・リセットにより操作の続行が影響を受けることはありません。

制御装置のすべてのチャンネル・パスでリセット信号が受信されると、最初のリセット信号が受信されてから最後のリセット信号が受信されるまでの間に制御装置で新しい活動が開始されていない限り、制御装置は完全にリセ

ットされます。「完全にリセットされる」というのは、制御装置で現在操作が行われているときはその操作が中止され、制御装置従属関係、制御装置状況、および制御装置モードがリセットされることを意味します。

ある入出力装置にアクセスできるすべての制御装置のすべてのチャンネル・パスでリセット信号が受信されると、最初のリセット信号を受信してから最後のリセット信号を受信するまでの間にその装置で新たな活動が開始されていない限り、その装置は完全にリセットされます。

「完全にリセットされる」というのは、装置で現在操作が行われているときはその操作が中止され、装置従属関係、装置状況、および装置モードがリセットされることを意味します。

以上を要約すれば、システム・リセットが行われると、リセット信号が受信されたチャンネル・パスについては、進行中の操作は必ずリセットされるということになります。リセットされるチャンネル・パスが、進行中の操作が実行されている唯一のチャンネル・パスである場合は、操作は完全にリセットされます。装置が、リセット信号が受信されたチャンネル・パス上でアクティブに通信している場合は、進行中の操作は無条件かつ完全にリセットされます。

該当の制御装置から隔離されているチャンネル・パス上にある制御装置および入出力装置は、リセット信号を受信しません。制御装置を特定のチャンネル・パスから隔離するには、その制御装置にアクセスするために使用できる各チャンネル・パスごとに、制御装置の使用可能/使用不可スイッチを使用します。その場合、制御装置にそのチャンネル・パスについてマルチタグ付きの非請求状況がある場合は、その状況は保留のままになります。しかし、制御装置が、現在アクセス可能なすべてのチャンネル・パス上でリセット信号を受信している場合は、プログラムの視点から見れば、制御装置および入出力装置は完全にリセットされているように見えます。

個々の制御装置および入出力装置におけるリセット結果の状態については、該当の制御装置に関するシステム・ライブラリ資料の中で説明されています。

チャンネル・パス: 入出力システム・リセットが行われると、上記のセクションで述べたように、構成に含まれるすべてのチャンネル・パスでリセット信号が送信され、チャンネル・サブシステムはリセットされて初期化された状態になります。これらのアクションの結果として、チャンネル・サブシステムとそれに接続している制御装置および入出力装置の間の通信はすべて中止され、それらのコンポーネントがリセットされ、そして、構成内のすべてのチャンネル・パスは、静止状態にされるか構成解除されます。

サブチャンネル: 入出力システム・リセットが行われると、すべてのサブチャンネル上のすべての操作が終了します。すべてのサブチャンネル内で、状況情報、すべての割り込み条件 (保留中の割り込みを除く)、専用従属関係条件、および、前の条件と操作に関する内部標識がリセットされ、有効なすべてのサブチャンネルが初期化された状態になります。

初期化された状態では、有効なすべてのサブチャンネルのサブチャンネル・パラメーターが、それぞれの初期値に設定されます。以下のサブチャンネル・パラメーターの初期値は 0 です。

- 割り込みパラメーター
- 入出力割り込みサブクラス・コード (ISC)
- 使用可能
- 限界モード
- 測定モード
- マルチパス・モード
- パス非稼働マスク
- 最終使用パス・マスク
- 測定ブロック指標
- 並行センス

以下のサブチャンネル・パラメーターの初期値は、個々の有効なサブチャンネルに関連した装置のインストール手順の一環として割り当てられます。

- タイミング機構
- 装置番号
- 論理パス・マスク (パス・インストール・マスクと同じ値)
- パス・インストール・マスク
- パス使用可能マスク
- チャンネル・パス ID 0-7

割り当てられる値は、システム・モデルと構成によって異なることがあります。どのように異なるかについては、該当のシステム・モデルに関するシステム・ライブラリ資料で説明されています。また、プログラミング上の考慮事項により、割り当てられる値がさらに制約されることがあります。

パス稼働マスクの初期値は、すべて 1 です。

装置番号有効ビットは、入出力装置が割り当てられているすべてのサブチャンネルについて 1 に設定されます。

サブチャンネル情報ブロック内のモデル依存区域の初期値については、該当システム・モデルに関するシステム・ライブラリ資料の中で説明されています。

サブチャンネル状況ワードと拡張状況ワードの初期値は、すべて 0 です。

サブチャンネルの初期化された状態は、上記で述べたサブチャンネル・パラメーターがすべて初期値であるときの状態です。サブチャンネル・パラメーターについては、15-1ページの『サブチャンネル情報ブロック』、16-6ページの『サブチャンネル状況ワード』、および 16-31ページの『拡張状況ワード』で説明されています。

チャンネル・パス・リセット機能: 入出力システム・リセットを行うと、チャンネル・パス・リセット機能がリセットされます。RESET CHANNEL PATH により開始されたチャンネル・パス・リセット機能 (保留中のものも進行中のものも含む) は、入出力システム・リセットによりオーバーライドされます。通常は、チャンネル・パス・リセット機能が完了するとマシン・チェック割り込み条件が起きますが、入出力システム・リセットが起きたときに保留中または進行中だったチャンネル・パス・リセット機能については、この割り込み条件は生成されません。

アドレス限界検査機能: 入出力システム・リセットを行うと、アドレス限界検査機能がリセットされます。アドレス限界値は、すべて 0 に初期化され、有効化されません。

チャンネル・サブシステム・モニター機能: 入出力システム・リセットを行うと、チャンネル・サブシステム・モニター機能がリセットされます。測定ブロック更新モードおよび装置接続時間測定モードがアクティブのときは、非アクティブにされます。測定ブロック起点と測定ブロック・キーは、どちらも 0 に初期化され、有効化されます。

保留チャンネル報告: 入出力システム・リセットを行うと、保留チャンネル報告がリセットされます。

チャンネル・サブシステム・タイマー: 入出力システム・リセットは、チャンネル・サブシステム・タイマーの内容には、必ずしも影響を与えません。チャンネル・サブシステム・タイマー検査の機能を備えたモデルでは、入出力システム・リセットの結果、チャンネル・サブシステム・タイマーが有効化されることがあります。

保留入出力割り込み: 入出力システム・リセットは、保留入出力割り込みには影響を与えません。ただし、サブシステム・リセット時には、入出力システム・リセットが実行されると同時に、入出力割り込みが消去されます。(4-44ページの『サブシステム・リセット』を参照してください。)

影響を受ける部分	入出力システム・リセットによる影響 ¹
チャンネル・サブシステム状態	リセットされ初期化される
制御装置および入出力装置	リセットされる
チャンネル・パス	静止する
サブチャンネル	リセットされ初期化される
割り込みパラメーター	0 ²
入出力割り込みサブクラス・コード (ISC)	0 ²
使用可能ビット	0 ²
アドレス限界モード・ビット	0 ²
タイミング機構ビット	インストール時の値 ²
マルチパス・モード・ビット	0 ²
測定モード・ビット	0 ²
装置番号有効ビット	インストール時の値 ²
装置番号	インストール時の値 ²
論理パス・マスク	パス・インストール・マスク値と同じ ²
パス非稼働マスク	0 ²
最終使用パス・マスク	0 ²
パス・インストール・マスク	インストール時の値 ²
測定ブロック指標	0 ²
パス稼働マスク	1 ²
パス使用可能マスク	インストール時の値 ^{2, 3}
チャンネル・パス ID 0-7	インストール時の値 ²
並行センス・ビット	0 ²
サブチャンネル状況ワード	0 ²
拡張状況ワード	0 ²
モデル依存区域	モデルにより異なる ²
チャンネル・パス・リセット機能	リセットされる
アドレス限界検査機能	リセットされ初期化される
アドレス限界値	0 ²
チャンネル・サブシステム・モニター機能	リセットされ初期化される
測定ブロック更新モード	非アクティブ ²
装置接続時間測定モード	非アクティブ ²
測定ブロック起点	0 ²
測定ブロック・キー	0 ²
保留チャンネル報告ワード	消去される
チャンネル・サブシステム・タイマー	変更されない/有効化される
説明:	
1 入出力システム・リセットが各分野に与える影響の詳細説明については、本文を参照してください。	
2 初期化値。	
3 モデル固有の構成制御機能があれば、それにも依存します。	

図 17-1. 入出力システム・リセット・アクションの要約

外部開始機能

入出力システム・リセットは外部開始機能の 1 つですが、これについては、17-10 ページの『入出力システム・リセット』を参照してください。

初期プログラム・ローディング

初期プログラム・ローディング (IPL) は、手動により指定の装置からプログラムを読み取り、そのプログラムの実行を開始するための手段を提供します。

モデルによっては、IPL に関連した追加の制御機能および標識を備えているものもあります。この追加情報については、該当モデル用のシステム・ライブラリー資料を参照してください。

手動で IPL を開始するには、ロード装置アドレス制御を、入力装置を指示する 4 桁番号に設定し、次にロード消去キーまたはロード通常キーを活動化します。ロード消去キーまたはロード通常キーを活動化すると、アーキテクチャー・モードが ESA/390 モードに設定されます。参考のために、ESA/390 の IPL の説明に対する追加要素を挙げておきます。

ロード消去キーを活動化すると、構成に対してクリア・リセットが行われます。

ロード通常キーを活動化すると、この CPU で初期 CPU リセットが行われ、構成内の他のすべての CPU に CPU リセットが伝搬され、構成のその他の部分についてはサブシステム・リセットが行われます。

操作のローディング部分の中で、リセットが行われた後は、この CPU はロード状態に入ります。この CPU は、リセット操作の実行中に必ず停止状態に入るとは限りません。CPU がロード状態にあるときは、ロード表示ライトが点灯しています。

条件が許せば、次に、指定した入力装置と関連のサブチャンネルからの読み取り操作が開始されます。この読み取り操作が行われる方法は、(1) ロード装置アドレス制御により指定されている装置番号に対応するサブチャンネルと、(2) 論理パス・マスク・フィールドのみがすべて 1 で、その他はすべて 0 である ORB を指定して、START SUBCHANNEL 命令を実行した場合と同じです。チャンネル・サブシステムは、ORB のパラメーターを以下のように解釈します。

- 割り込みパラメーター: すべて 0
- サブチャンネル・キー: すべて 0
- 延期制御: 0 (延期は許されない)
- CCW フォーマット: 0
- CCW 事前取り出し: 0 (事前取り出しは許されない)
- 初期状況割り込み制御: 0 (要求なし)
- アドレス限界検査制御: 0 (検査なし)
- 延期割り込み抑制: 0 (抑制は許されない)
- 論理パス・マスク: 1 (すべてのチャンネル・パスが論理的に使用可能)
- 誤長抑制モード: 0 (フォーマット 0 の CCW が指定されているので無視される)
- チャンネル・プログラム・アドレス: 絶対アドレス 0

最初に実行する CCW は、記憶機構からは取り出されません。代わりに、絶対記憶位置 0 から始まり、以下の詳細フォーマットを持つフォーマット 0 の CCW が暗黙指定されたものとして、実行されます。

記憶位置

0	00000010	00000000	0000000000000000
4	01100000	////////	0000000000011000
	0	8	16
			31

上記の図に示した CCW は、修飾ビットが 0、データ・アドレスが 0、バイト・カウントが 24、コマンド・チェーン・フラグが 1、誤長表示抑止ビットが 1、データ・チェーン・フラグが 0、スキップ・フラグが 0、プログラム制御割り込み (PCI) フラグが 0、間接データ・アドレス (IDA) フラグが 0、そして延期フラグが 0 の読み取りコマンドを指定しています。コマンド・チェーンの結果として記憶位置 8 または 16 から取り出される CCW は、IPL シーケンスにおける後続の CCW と同様に、一般的な入出力操作での CCW と同じに解釈されます。ただし、IPL チャンネル・プログラムで指定された PCI フラグは無視されます。

サブチャンネルが IPL 読み取りに対して開始保留にされた時点で、サブチャンネルは使用可能にもされます。これは、チャンネル・サブシステムが装置からの後続の状況を適切に処理できるようにするとともに、IPL 装置を使用する後続の入出力操作を容易にするためです。(IPL 入出力操作に使用するサブチャンネルを除き、各サブチャンネルは、まず MODIFY SUBCHANNEL により使用可能にしておかないと、開始機能または装置からの状況を受理することができません。)

IPL チャンネル・プログラムの最後の操作について IPL サブチャンネルが状況保留になったときは、入出力割り込み条件は生成されません。代わりに、TEST SUBCHANNEL が実行された場合と同様に、絶対記憶位置 184-187 にサブシステム ID が格納され、絶対記憶位置 188-191 に 0 が格納され、そしてサブチャンネルから保留状況が消去されます。ただし、通常の場合であれば IRB に格納される情報は、格納されません。通常の場合であれば格納されているはずのサブチャンネル状況フィールドにすべて 0 が格納されており、また、通常の場合であれば格納されているはずの装置状況フィールドに、チャンネル終了標識のみ (装置終了標識を伴っていてもいなくてもよい) が格納されている場合は、IPL 入出力操作は正常に完了しているものと見なされます。IPL 入出力操作について、チャンネル終了状況の後で別個に装置終了状況が提示された場合は、入出力割り込み条件が生成されます。IPL 入出力操作が正常に完了すると、絶対記憶位置 0-7 から新しい PSW がロードされます。PSW のロードが成功し、IPL の完了を妨げるような誤動作が認識されなければ、CPU はロード状態から抜けて、ロード・ライトは消えます。速度制御が処理の位置に設定

されている場合は、CPU は稼働状態になり、新しい PSW の制御下で CPU 操作が進行します。速度制御が命令ステップの位置に設定されている場合は、CPU は、新しい PSW がロードされた後で停止状態に入り、手動ライトが点灯します。

IPL 入出力操作または PSW ローディングが正常に完了しなかった場合は、CPU はロード状態のままになり、ロード・ライトは点灯したままになります。

以下のいずれかが起きた場合は、IPL は完了しません。

- ロード装置アドレス制御で指定されている IPL 装置番号に等しい有効な装置番号を含むサブチャンネルがない。
- CPU、主記憶機構、またはチャンネル・サブシステムで、IPL の完了を妨げる誤動作が検出された。
- IPL 読み取りに対してサブチャンネルが開始保留になった後、IPL サブチャンネルがサブチャンネル・アクティブになる前に、装置が非請求の警報状況を提示した。この場合は、IPL 読み取り操作は開始されません。
- IPL 装置へのすべての使用可能なチャンネル・パス上で、その装置が稼働状態にないと見なされたか、または使用可能なチャンネル・パスがない。
- IPL 入出力操作時に、IPL 装置が、チャンネル終了、装置終了、状況修飾、制御装置終了、制御装置使用中、装置使用中、または再試行状況以外の標識を含む状況バイトを提示した。状況バイトに、制御装置終了、制御装置使用中、または装置使用中が示されているときは、通常のパス管理アクションがとられません。
- IPL 入出力操作時に、PCI 以外のサブチャンネル状況標識が生成された。
- 絶対記憶位置 0-7 からロードされた PSW に、早期認識されるタイプの PSW フォーマット・エラーがある。

入力された装置番号に対応するサブチャンネルがないか、またはマシン誤動作が生じた場合を除き、IPL 装置のサブシステム ID が絶対記憶位置 184-187 に格納されます。上記いずれかの条件が生じた場合は、これらの記憶位置の内容は予測不能です。また、どのような場合も、IPL が成功しなかったときは、絶対記憶位置 0-7 の内容は予測不能です。

IPL が成功した後は、サブチャンネル・パラメーターには、上記で述べたとおり ORB を指定して実際に START SUBCHANNEL を実行した場合と同様の、通常の値が入っています。

プログラミング上の注意:

1. 読み取られて絶対記憶位置 8-15 および 16-23 に格納された情報は、IPL 入出力操作時に追加情報を読み取るために使用することができます。例えば、記憶位置 8 の CCW で、記憶機構内の他の場所にある追加の CCW の読み取りを指定し、記憶位置 16 の CCW で、チャンネル内転送コマンドを指定して追加の CCW への転送を行うことができます。
2. 状況修飾ビットは、IPL 入出力操作の際にも通常と同じ働きをします。つまり、チャンネル・サブシステムは、現 CCW より 16 だけ大きいアドレスを持つ CCW を取り出して、その CCW にチェーニングします。これは、暗黙 CCW で指定されている読み取り操作の完了後に起きる初期チェーニングにも適用されます。
3. IPL 操作が完了したときにロードされる PSW は、IPL 入出力操作の最初の 8 バイトに指定されているものでも、後続の CCW によって絶対記憶位置 0-7 に格納されたものでも構いません。
4. ロード通常キーを活動化すると、主記憶機構の最初の 24 バイトと、絶対記憶位置 184-191 の 8 バイトを使用することが、暗黙的に指定されます。IPL プログラムの残りの部分は記憶機構内のどの位置にでも置くことができるので、記憶機構内のデバッグや回復に役立つ領域をそのまま残しておくことができます。1-15 ページの『z/Architecture と ESA/390 の間の互換性』の『プログラミング上の注意』で述べたように、記憶機構の低位 512 バイトは予約済みなので、この領域には IPL プログラムを入れないでください。ロード消去キーが活動化されると、IPL プログラムは、クリアされたマシンが既知の状態にされ、IPL プログラムはその状態から始動します。ただし、外部記憶装置の情報は変わりません。
5. 絶対記憶位置 0 にある PSW のビット 14 が 1 に設定されているときは、IPL 操作の完了後に CPU は待ち状態にされます。その時点で、ロード・ライトおよび手動ライトは消え、待ち状態ライトが点灯します。

入出力システムの再構成

入出力システムの再構成の方法はモデルによって異なります。例えば、構成変更を行うには、プログラムの制御下でモデル依存の **DIAGNOSE** 命令を使用する方法、システム・オペレーター構成制御を使用する方法、あるいは、**DIAGNOSE** と手動制御を組み合わせる方法があります。どの方法を使用するかは、システム・モデルによって異なります。どの方法で変更を行うかについては、該当のシステム・モデルに関するシステム・ライブラリー資料を参照してください。再構成に伴うチャンネル・パスの分割は、**STORE SUBCHANNEL** が実行されたときに格納される **SCHIB** 内の **PAM** ビットの設定値により示されます (15-7ページの『パス使用可能マスク (PAM)』を参照)。

状況検査

状況検査機能を使用することにより、チャンネル・サブシステムは、装置が有効な **CBC** を持つ装置状況バイトを提示したが、その状況バイトがチャンネル・サブシステムに提示された時点では不適切であったビットの組み合わせが含まれていた、ということを示すことができます。チャンネル・サブシステムが **ESW** に入れてプログラムに渡す標識を、装置状況チェックと言います。チャンネル・サブシステムが装置状況チェック条件を認識すると、インターフェース制御チェック条件も認識されます。適切と見なされる状況の組み合わせと、不適切と見なされる状況の組み合わせに関する要約説明については、システム・ライブラリー資料「*IBM Enterprise Systems Architecture/390 ESCON I/O Interface*」(SA22-7202) および「*IBM System/360 and System/370 I/O Interface Channel to Control Unit OEMI*」(GA22-6974)、および ANSI 標準文書「*Fibre Channel - Single-Byte Command Code Sets-2 (FC-SB-2)*」を参照してください。

アドレス限界検査

アドレス限界検査機能は、記憶機構への入出力データ・アクセスに対する記憶機構保護メカニズムを提供するもので、キー制御記憶保護をさらに強化します。アドレス限界検査が使用されているときは、プログラム制御のアドレス限界値に従って、絶対記憶域が 2 つの部分に分割されます。オプションにより、各サブチャンネルの限界モードに基づき、入出力データ・アクセスをこの絶対記憶域の 2 つの部分のいずれか一方のみに制限することができます。アドレス限界による制限は、キー制御記憶保護に関係なく、つまり、サブチャンネル・キーが 0 であるかまたは記憶域内のキーと一致していても、適用されま

す。CCW と IDAW の取り出しには、アドレス限界検査は適用されません。

アドレス限界検査機能は以下の要素から成っています。

- 入出力命令 **SET ADDRESS LIMIT**。
- 各サブチャンネルでの限界モード。
- **ORB** 内のアドレス限界検査制御ビット。

SET ADDRESS LIMIT を実行すると、汎用レジスター 1 の内容がアドレス限界検査機能に渡されて、アドレス限界値として使用されます。汎用レジスター 1 のビット 32 と 48-63 は、すべて 0 (64K バイト境界に合わされた有効な絶対アドレスを指定する) でなければなりません。さもないと、オペランド例外が認識され、命令の実行は抑止されます。

各サブチャンネルの限界モードは、アドレス限界検査をどのように行うかを示します。限界モードを設定するには、**SCHIB** のワード 1 のビット位置 9 および 10 に所要の値を入れて、**MODIFY SUBCHANNEL** を実行します。これらの **SCHIB** ビットの設定値の意味は、以下のとおりです。

- 00 限界検査は行われぬ (初期設定値)。
- 01 データ・アドレスは、現在のアドレス限界と同じかそれより大きくなければならない。
- 10 データ・アドレスは、現在のアドレス限界より小さくなければならない。
- 11 予約済み。この組み合わせの限界モード・ビットを使用すると、**MODIFY SUBCHANNEL** が実行されたときにオペランド例外が認識されます。

アドレス限界検査制御ビット (**ORB** のワード 1 のビット 11) は、**START SUBCHANNEL** の実行により **ORB** の内容がサブチャンネルに渡されたときに受理された開始機能について、アドレス限界検査を使用するかどうかを指定します。**ORB** の内容が渡されるときにアドレス限界検査制御ビットが 0 であれば、その開始機能にはアドレス限界検査は適用されません。このビットが 1 のときは、アドレス限界検査が指定され、現在のアドレス限界と、サブチャンネルでの現在の限界モードの設定に基づいて、アドレス限界検査が行われます。

開始機能の実行時に、アドレス限界 (高または低) により保護されている絶対記憶位置にあるデータにアクセスしようとする、それはアドレス限界違反として認識され、アクセスは許可されません。その場合は、無効なアドレスにアクセスしようとした場合と同様に、プログラ

ム・チェック条件が認識され、チャンネル・プログラムの実行は中止されます。

構成警報

構成警報機能は、構成内のどのサブチャンネルにも関連付けられていない装置を検出するメカニズムを提供します。構成警報機能は、どのサブチャンネルにも関連付けられていない装置がプログラムと通信しようとしたことを、チャンネル報告を使用してプログラムに通知します。

インストール手順の実行時に、各装置をいずれかのサブチャンネルに割り当てる必要があります。割り当てられていない装置については、チャンネル・サブシステムは入出力割り込み条件を生成することができません。なぜなら、入出力割り込みコードには、入出力割り込み条件を引き起こした特定の装置を識別するサブチャンネル番号が含まれていなければならないからです。どのサブチャンネルにも関連付けられていない装置がチャンネル・サブシステムと通信しようとする、構成警報機能は、その関連付けられていない装置を示すチャンネル報告を生成しません。保留中のチャンネル報告がどのようにプログラムに通知されるか、およびチャンネル報告内の情報がどのように検索されるかについては、17-18ページの『チャンネル報告』を参照してください。

誤長表示抑止

誤長表示抑止機能を使用すると、フォーマット 1 の CCW を使用しているときに、フォーマット 0 の CCW を使用しているときと同様に、即時操作についての誤長の表示を抑止することができます。誤長表示抑止機能がインストールされているときは、ORB のワード 1 のビット 24 を使用して、フォーマット 1 の CCW を使用しているときに即時操作について誤長表示を抑止するか、それとも誤長表示を現 CCW の SLI フラグの制御下においたままにするか (即時操作として実行されていない CCW の場合と同じ) を、チャンネル・サブシステムに指示します。このビットの働きにより、チャンネル・プログラムは、フォーマット 0 またはフォーマット 1 のどちらの CCW が使用されているときでも、誤長表示について同じ動作をすることができます。

並行センス

サブチャンネルが並行センス・モードになっているときに、並行センス機能が提供するメカニズムを使用することにより、チャンネル・サブシステムは、装置から渡されたセンス情報を、装置チェック表示が含まれているものと同じ IRB に入れてプログラムに提示することができます。

まず、並行センス機能を適用できるサブチャンネルに対して、SCHIB オペランドのワード 6 のビット 31 を 1 に設定して MODIFY SUBCHANNEL を実行すると、そのサブチャンネルで並行センス・モードがアクティブになります。並行センス機能が適用されるのは、チャンネル・サブシステムがプログラムの介入を必要とせず、装置からセンス情報を検索するために使用できるチャンネル・パスに関連したサブチャンネルです。

チャンネル・サブシステム・リカバリー

チャンネル・サブシステムは、広範囲にわたる誤動作やその他の条件を検出するリカバリー・メカニズムを備えており、これによりチャンネル・サブシステムの安全性が確保されると同時に、一部の誤動作からの自動リカバリーも達成されます。チャンネル・サブシステムのリカバリー・メカニズムは、さまざまな報告手段を使用して、プログラムのリカバリー、保守、および修復を助けます。

特定の誤動作またはその他の条件を報告するために使用される方法は、その誤動作またはその他の条件の重大度と、その誤動作またはその他の条件をどの程度分離できるかによって異なります。チャンネル・サブシステム内の誤動作またはその他の条件は、次のいずれかの方法で格納される情報によって、プログラムに示されます。

1. チャンネル・サブシステムまたは装置により認識され、プログラムに知らせる必要がある条件については、その状況を記述する情報が IRB に入れて提示されます。一般に、TEST SUBCHANNEL 命令を実行すると、プログラムでこの情報を使用できるようになります。この命令は、通常、入出力割り込みの発生に対する応答として実行されます。(格納される情報の定義については、16-5ページの『割り込みアクション』、および第6章、『割り込み』を参照してください。)
2. チャンネル・サブシステム内の特定の機能に影響を与えるマシン誤動作については、チャンネル報告により情報が提供されます。STORE CHANNEL REPORT WORD 命令を実行すると、プログラムでこの情報を使用できるようになります。この命令は、通常、マシン・チェック割り込みの発生に対する応答として実行されます。(マシン・チェック割り込みのメカニズムと、マシン・チェック割り込みコードの内容については、第11章、『マシン・チェックの取り扱い』を参照してください。)
3. 一連のチャンネル・サブシステム機能に影響を与えるような誤動作またはその他の条件については、チャンネル報告により情報が提供されます。この情報は、項目 2 に示したのと同じ方法により、プログラムで使用できるようになります。

4. チャンネル・サブシステムの操作安全性の維持に影響を与えるような誤動作については、マシン・チェック割り込みコード (MCIC) により情報が提供されます。(11-17ページの『チャンネル・サブシステム損傷』を参照してください。)
5. プロセスまたはシステムの操作安全性の維持に影響を与えるような誤動作については、MCIC を使用して情報が提供されます。(11-15ページの『命令処理損傷』、および 11-15ページの『システム損傷』を参照してください。)

誤動作またはその他の条件を報告するためにチャンネル報告が使用されるのは、入出力割り込み機能の使用が適切でない場合のみであり、その場合は、チャンネル・サブシステム損傷、命令処理損傷、またはシステム損傷の報告より優先して使用されます。

チャンネル報告

チャンネル・サブシステムのエレメントに影響を与えるような誤動作またはその他の条件が認識されると、チャンネル報告が生成されます。エラー条件からリカバリーするためには、プログラムまたは外部的な手段によりリカバリー・アクションを実行することが必要な場合があります。チャンネル報告は、適切なリカバリー・アクションを決定するために必要と認められる程度で、チャンネル報告の発生源とリカバリー状態を示します。チャンネル報告は、誤動作またはその他の条件の分析結果に基づいて生成された 1 つ以上のチャンネル報告ワード (CRW) から成っています。チャンネル報告に複数の CRW が含まれている場合は、そのことを示すために、チャンネル報告内の最後の 1 つを除くすべての CRW のチェーンニング・フラグに 1 が格納されています。

チャンネル・プログラムが、プログラムによる検索と分析 (STORE CHANNEL REPORT WORD の実行による) のためにチャンネル報告を保留にしているときは、1 つ以上のチャンネル・サブシステム機能の正常な動作に影響を与えるような誤動作またはその他の条件が認識されています。保留にされているチャンネル報告が初期チャンネル報告である場合は、チャンネル・サブシステムで 1 つ以上の CRW が保留中であることを示すマシン・チェック割り込み条件が生成されます。チャンネル報告が初期報告であると見なされるのは、その報告が、最も新しい入出力リセット後に生成された最初のチャンネル報告である場合、または、これまでに生成されて保留状態になっている報告がなく、かつ、最後に実行された STORE CHANNEL REPORT WORD の結果として、保留中のチャンネル・レポートがないことを示す条件コード 1 が設定されている場合です。マシン・チェック割り込みが起き、マシン・チェック割り込みコードのビット 9 (チャ

ネル報告保留) が 1 であるときは、保留中のチャンネル報告があります。関連のマシン・チェック割り込みが起きる前に、プログラムがチャンネル報告の最初の CRW を消去した場合、一部のモデルでは、マシン・チェック割り込み条件がリセットされ、関連のマシン・チェック割り込みが起きないことがあります。チャンネル報告が保留中であることを示すマシン・チェック割り込みが起きるのは、マシン・チェック・マスク (PSW のビット 13) およびチャンネル報告保留サブクラス・マスク (制御レジスター 14 のビット 3) が、両方とも 1 の場合のみです。

保留にされているチャンネル報告が初期チャンネル報告ではない場合は、マシン・チェック割り込み条件は生成されません。マシン・チェック割り込み後に実行される最初の STORE CHANNEL REPORT WORD 命令に対する応答としてプログラムに提示される CRW は、マシン・チェック条件が生成される原因となった初期チャンネル報告に含まれていることも、そうでないこともあります。マシン・チェック割り込みがどの CPU で起きたかに関係なく、いずれかの CPU で STORE CHANNEL REPORT WORD が実行されると、保留中のチャンネル報告ワードは消去されます。保留中の CRW がいないときに STORE CHANNEL REPORT WORD が実行された場合は、条件コード 1 が設定され、第 2 オペランド・アドレスが示す位置には 0 が格納されます。マシン・チェック割り込みを受信した結果として STORE CHANNEL REPORT WORD が実行されたときに、条件コード 1 が設定され、0 が格納されることがあります。これは、(1) 他の CPU により関連のチャンネル報告が消去されたためか、または、(2) チャンネル報告を生成しているときに誤動作が起きたためです。後者の場合は、後続の試行の際に有効なチャンネル報告を保留にできるときは、さらに新たなマシン・チェック割り込み条件が生成されます。

チャンネル報告が、チェーンニングされた複数の CRW から成っているときは、それらの CRW は、連続した STORE CHANNEL REPORT WORD の実行結果として、チャンネル・サブシステムがチェーンに格納したときと同じ順序で、プログラムに提示されます。例えば、STORE CHANNEL REPORT WORD の実行結果として、チェーン内の最初の CRW がプログラムに提示されたとすれば、次回に STORE CHANNEL REPORT WORD が実行されたときに提示される CRW は、同じチェーン内の 2 番目の CRW であり、別のチャンネル報告内の CRW ではありません。

複数のチャンネル報告がプログラムに提示される順序については、特定の規則はありません。ただし、複数のチャンネル報告において、最初または唯一の CRW が同じ報告ソース・コードおよび同じ報告ソース ID を示している

場合は例外です。このようなチャンネル報告は、チャンネル・サブシステムにより生成されたときと同じ順序でプログラムに提示されますが、必ずしも連続して提示されるとは限りません。例えば、チャンネル・サブシステムが、チャンネル報告 A、B、および C をこの順序で生成したとします。そして、チャンネル報告 B および C の最初の CRW が、同じ報告ソース・コードおよび同じ報告ソース ID を示しているとします。この場合、チャンネル報告 B はチャンネル報告 C より前にプログラムに提示されますが、チャンネル報告 A は、チャンネル報告 B とチャンネル報告 C の間に提示されることもあります。

プログラミング上の注意:

- 1 つの CRW に入れて提供された情報が、その後同じチャンネル・サブシステム機能について生成された別の CRW によって無効になることがあります。したがって、1 つのチャンネル報告で提供された情報は、その時点で保留されているすべてのチャンネル報告で提供されている情報を考慮に入れて、解釈する必要があります。
- 保留されているチャンネル報告があるときに、必ずマシン・チェック割り込み条件が生成されるとは限りません。マシン・チェック割り込み条件が生成される原因となる条件については、このセクションで前述しました。
- マシン・チェック割り込みが起き、マシン・チェック割り込みコードのビット 9 が 1 に設定された場合は、保留中のチャンネル報告がすべて消去され、条件コード 1 が設定されるまで、STORE CHANNEL REPORT WORD を繰り返し実行してください。
- マシン・チェック割り込みが起きた後で、プログラムがタイムリーな方法で連続して STORE CHANNEL REPORT WORD 命令を実行しないと、CRW オーバーフロー条件が発生することがあります。
- 同時に保留にできる CRW の数は、モデルによって異なります。オーバーフロー条件が発生しているときは、本来なら保留にされるはずの CRW が失われ、プログラムに提示されません。

チャンネル報告ワード

チャンネル報告ワード (CRW) は、入出力操作、入出力装置、またはチャンネル・サブシステムの一部のエLEMENT (チャンネル・パスやサブチャンネルなど) のリカバリーに役立つ情報を、プログラムに提供します。CRW のフォーマットは以下のとおりです。

0	S	R	C	RSC	A	0	ERC	Reporting-Source ID
0	1	2	3	4	8	10	16	31

請求 CRW (S): ビット 1 が 1 のときは、請求 CRW を示します。チャンネル・サブシステムが CRW を請求 CRW と見なすのは、プログラムによる何らかのアクションの直接の結果として、その CRW が保留にされている場合です。ビット 1 が 0 のときは、CRW は非請求であり、チャンネル・サブシステムがプログラムから独立して行ったアクションの結果として保留にされていません。

オーバーフロー (R): ビット 2 が 1 のときは、この CRW が保留になった後で CRW オーバーフロー条件が認識されていること、および 1 つまたは複数の CRW が失われていることを示します。このビットが 1 である CRW は、オーバーフローが認識されたときに保留にされていた最新の CRW です。ビット 2 が 0 のときは、CRW オーバーフロー条件は認識されていません。

オーバーフロー条件が存在していなくても、チャンネル報告の一部となっている CRW が保留されないことがあります。それは、オーバーフロー条件が原因で、その報告内の前の CRW が保留されなかった場合です。

チェーニング (C): ビット 3 が 1 で、オーバーフロー・フラグが 0 のときは、関連 CRW のチェーニングを示します。1 つの誤動作またはその他の条件が複数の CRW によって記述されているときは、常に CRW のチェーニングが示されます。チャンネル報告が単一の CRW で記述されているか、またはこの CRW がチャンネル報告の最後の CRW である場合は、チェーニング・フラグは 0 です。

オーバーフロー・ビット (ビット 2) が 1 のときは、チェーニング・フラグには意味はありません。

報告ソース・コード (RSC): ビット 4-7 は、チャンネル報告に関連したチャンネル・サブシステム機能を示します。一部の機能は、報告ソース識別フィールド (下記を参照) によりさらに詳細に識別されます。機能は、以下に示すビットの組み合わせにより識別されます。

ビット	機能
4 5 6 7	モニター機能
0 0 1 0	サブチャンネル
0 1 0 0	チャンネル・パス
1 0 0 1	構成警報機能
1 0 1 1	チャンネル・サブシステム

報告ソース・コード・フィールド内での上記以外のビットの組み合わせは、すべて予約済みです。

補助報告 (A): ビット 8 が 1 のときは、システム・コンポーネントの誤動作が起きたこと、そして、その誤動作は、以前に認識されているか、または複数のチャンネル・サブシステム機能の活動に影響を与えたものであることを示します。誤動作が複数のチャンネル・サブシステム機能の活動に影響を与えている場合は、影響を受けた機能のうち 1 つを除くすべてについて補助報告条件が認識されます。このビットが 0 のときは、システム・コンポーネントのこの誤動作が、以前に認識されたものではないことを示します。このビットはすべてのチャンネル報告で意味を持ちます。

モデルによっては、補助報告条件を認識する機能がない場合や、チャンネル・サブシステム機能に影響を与えるすべてのシステム誤動作についてこの条件が認識されるとは限らない場合があります。補助報告認識の機能がないときは、ビット 8 は 0 に設定されます。

エラー・リカバリー・コード (ERC): ビット 10-15 が 0 のときは、チャンネル・サブシステムに、報告ソース・コードに示されているチャンネル・サブシステム機能に関するエラー情報があり、プログラムがその情報を要求できることを示します。そうでない場合は、ビット位置 10-15 には、報告ソース・コードに示されているチャンネル・サブシステム機能のリカバリー状態を定義するエラー・リカバリー・コードが含まれています。プログラムは、このフィールドと報告ソース・コードを組み合わせることで使用することにより、識別されている機能がすでにリカバリー済みで使用可能であるか、それともまだリカバリー・アクションを必要としているかを判別することができます。以下のエラー・リカバリー・コードが定義されています。

ビット	状態
10 11 12 13 14 15	イベント情報保留中
0 0 0 0 0 0	使用可能
0 0 0 0 1 0	初期設定済み
0 0 0 0 1 1	一時エラー
0 0 0 1 0 0	インストール済みパラメーターの初期設定済み
0 0 0 1 0 1	終了
0 0 0 1 1 0	機能が初期設定されていない永続エラー
0 0 0 1 1 1	機能が初期設定されている永続エラー
0 0 1 0 0 0	インストール済みパラメーターが変更された

エラー・リカバリー・コード・フィールド内での上記以外のビットの組み合わせは、すべて予約済みです。

各エラー・リカバリー・コードの意味は、CRW 内に一緒に含まれている報告ソース・コードによって異なります。エラー・リカバリー・コードは、以下のように定義されています。

イベント情報保留中: プログラムは、示されている機能に関するイベント情報を検索できます。この CRW は、示されている機能の状態を示していません。

使用可能: 示されている機能は、CRW が生成されていない場合にプログラムが予想するおりの状態になっています。

初期設定済み: 示されている機能は、最新のシステム IPL の一環として行われた入出力システム・リセットの直後に存在していた状態と同じ状態になっています。

一時エラー: 示されている機能は、通常どおりに作動していないか、異常イベントの発生を認識しています。続いて行われるアクションとしては、この機能を正常な動作に復元すること、または異常イベントを記述する適切な情報を記録することが考えられます。

インストール済みパラメーターの初期設定済み: この状態は、この機能に関連するパラメーターのうち、プログラムでは変更できない 1 つまたは複数のパラメーターが変更されている場合があることを除けば、初期設定された状態と同じです。

終了: 示されている機能は、進行中だった操作が正常に完了または終了できないような状態になっています。

機能が初期設定されていない永続エラー: 示されている機能は誤動作状態にあり、チャンネル・サブシステムは、その機能に対してリセット機能を実行していません。

機能が初期設定されている永続エラー: 示されている機能は誤動作状態にあり、チャンネル・サブシステムは、その機能に対してリセット機能を実行したか、または実行した可能性があります。

インストール済みパラメーターが変更された: 示されている機能の 1 つまたは複数のパラメーターが変更されています。

報告ソース ID (RSID): ビット位置 16-31 には、報告ソース ID が含まれています。報告ソース ID は、チャンネル報告の原因となった条件および報告ソース・コードに応じて、影響を受けたチャンネル・サブシステム機能をさらに詳しく識別するか、または、チャンネル報告の原因となった条件を記述する追加情報を提供します。RSID フィールドのフォーマットは、報告ソース・コードのビット設定に応じて以下のようになります。

報告ソース・コード				報告ソース ID			
4	5	6	7	ビット 16-31			
0	0	1	0	0000	0000	0000	0000
0	0	1	1	xxxx	xxxx	xxxx	xxxx
0	1	0	0	0000	0000	yyyy	yyyy
1	0	0	1	0000	0000	yyyy	yyyy
1	0	1	1	0000	0000	0000	0000

注:

xxxx xxxx xxxx xxxx
yyyy yyyy

サブチャンネル番号
チャンネル・パス ID
(CHPID)

チャンネル・サブシステム入出力優先順位決定機能

チャンネル・サブシステム入出力優先順位機能を使用することにより、プログラムは、チャンネル・サブシステムにおいて、START SUBCHANNEL が実行されて条件コード 0 が示されたときに開始保留状態にするサブチャンネル間の、優先順位関係を確立することができます。また、ファイバー・チャンネル・チャンネル・パス (FICON チャンネル・パスおよび FICON 変換チャンネル・パス) に対して構成されている入出力サブチャンネルの場合は、プログラムは、この機能を使用して、ファイバー・チャンネル接続の制御装置における入出力操作の優先順位関係も確立することができます。

プログラムで、所要のチャンネル・サブシステム優先順位および制御装置優先順位を割り当てるには、START SUBCHANNEL を実行するときに、ORB の拡張部分の中に所要の優先順位番号を指定します。

チャンネル・サブシステムは、開始機能または再開機能を開始しようとするときに、ORB の中で指定されているチャンネル・サブシステム優先順位番号を使用して、開始保留中または再開保留中のサブチャンネルを選択する順序を決定します。これらの機能についての詳細は、15-17 ページの『開始機能と再開機能』を参照してください。一般に、チャンネル・サブシステムは、開始保留または再開保留状態にある入出力サブチャンネルを、開始機能または再開機能の開始のために選択するときに、優先順位番号の高いものから低いものへの順に選択していきます。チャンネル・サブシステムがこの目的に使用する詳細な優先順位選択アルゴリズムは、モデルによって異なります。さらに、チャンネル・サブシステムは、入出力サブチャンネルを選択するときに、優先順位選択アルゴリズムに加えて公平選択アルゴリズムも適用します。公平選択アルゴリズムも、モデルによって異なります。どのモデルでも、開始保留中または再開保留中の入出力サブチャンネルには、常に、チャンネル・サブシステムの優先順位選択アルゴリズムと公平選択アルゴリズムが適用されます。一部のモデルでは、消去保留中または停止保留中のサブチャンネルにも、両方のアルゴリズムが適用されることがあります。各モデルが提供する優先順位選択アルゴリズムと公平選択アルゴリズムの説明、および、これらのアルゴリズムが消去保留中または停止保留中のサブチャンネルにも適用されるかどうかについては、該当モデルのシステム・ライブラリー資料を参照してください。

ORB 内に指定されている制御装置優先順位番号は、ファイバー・チャンネル・チャンネル・パスに接続されている制御装置で CCW を実行する優先順位を決定するために、その制御装置により使用されます。詳しくは、15-24 ページの『制御装置優先順位 (CU Priority)』を参照してください。

チャンネル・サブシステム優先順位レベルの数

モデルによっては、チャンネル・サブシステムが提供するチャンネル・サブシステム優先順位レベルの数が 256 より少ないことがあります。モデルが提供する各優先順位レベルは、8 ビットの符号なし 2 進整数で示されます。最も低いチャンネル・サブシステム優先順位レベルは整数 0 で示され、以後は、優先順位レベルが 1 つ高くなるたびにこの整数の値が 1 ずつ増加します。例えば、モデルが 16 個の優先順位レベルを提供しているとすれば、最低優先順位レベルから最高優先順位レベルへの順に、0-15 の番号が付けられます。

第18章 16 進浮動小数点命令

HFP 演算	18-1	LOAD AND TEST	18-12
HFP 数の表現	18-1	LOAD COMPLEMENT	18-13
正規化	18-3	LOAD FP INTEGER	18-13
HFP データのフォーマット	18-3	LOAD LENGTHENED	18-14
命令	18-4	LOAD NEGATIVE	18-15
ADD NORMALIZED	18-7	LOAD POSITIVE	18-15
ADD UNNORMALIZED	18-8	LOAD ROUNDED	18-16
COMPARE	18-9	MULTIPLY	18-16
CONVERT FROM FIXED	18-10	SQUARE ROOT	18-18
CONVERT TO FIXED	18-10	SUBTRACT NORMALIZED	18-19
DIVIDE	18-11	SUBTRACT UNNORMALIZED	18-20
HALVE	18-12		

HFP 演算

HFP 数の表現

16 進浮動小数点 (HFP) 数は、符号ビット、16 進小数部、および、指数部と呼ばれる 7 ビットの符号なし 2 進整数から成っています。指数部は、指数値に 64 を加算することにより求められた符号付き指数を表します (64 増しコード表記)。指数部の範囲は 0 から 127 まで、これは -64 から +63 までの指数範囲に対応します。HFP の大きさは、小数部の値と、指数部で表されている指数だけ 16 を累乗した値との積です。この数値は、符号ビットが 0 なら正、1 なら負になります。

HFP 数の小数部は、16 の累乗数である値が乗算されるものと見なされるので、16 進数として扱われます。小数部という名称は、左端の小数桁のすぐ左に小数点があるものとするということを示しています。

HFP 演算の結果、指数が 63 を超えることになる場合は、指数部は 127 から 0 に循環し、HFP 指数オーバーフロー条件が起きます。そのときの結果の指数部は、128 だけ小さすぎることとなります。演算の結果、指数が -64 より小さくなる場合は、指数部は 0 から 127 に循環し、HFP アンダーフロー条件が起きます。そのときの結果の指数部は、128 だけ大きすぎることとなります。ただし、真の 0 が強制適用されるときは、0 の指数部が生成されます。

真の 0 は、0 の指数部と 0 の小数部を持つ HFP 数です。真の 0 は、オペランドの特定の絶対値が原因で、算

術演算の正常な結果として生じることがあります。HFP 演算では、以下の場合には結果が強制的に正の真の 0 にされます。

1. HFP 指数アンダーフローが起き、PSW の HFP 指数アンダーフロー・マスク・ビットが 0 である。
2. 加算または減算操作の結果の小数部が 0 で、PSW の HFP 有効数字マスク・ビットが 0 である。
3. CONVERT FROM FIXED 命令のオペランドが 0 である。
4. DIVIDE 命令の被除数の小数部が 0 である。
5. HALVE、LOAD FP INTEGER、または SQUARE ROOT 命令のオペランドの小数部が 0 である。
6. 乗算操作の 1 つまたは両方のオペランドの小数部が 0 である。

上記の項目 2 は、正規化された命令にも正規化されていない命令にも適用されます。

HFP 指数アンダーフローに関するプログラム割り込みが起きたときは、真の 0 は強制適用されず、代わりに、小数部と符号は正しいままで、指数部は 128 だけ大きすぎることとなります。HFP 有効数字に関するプログラム割り込みが起きたときは、小数部は 0 のままで、符号は正で、指数部は正しいままとなります。

小数部が 0 の和、差、積、商、平方根、CONVERT FROM FIXED の結果、または LOAD FP INTEGER の結果は、正になります。その他の HFP 操作の結果と

命令	0 以外の 結果の 正規化の有無	0 の結果を強制的に 真の 0 にする処置の有無		0 の結果を 正にする処置の有無	
		短精度および 長精度	拡張精度	短精度および 長精度	拡張精度
ADD NORMALIZED	あり	あり/なし	あり/なし	あり	あり
ADD UNNORMALIZED	なし	あり/なし	-	あり	-
CONVERT BFP TO HFP ¹	あり	あり	-	なし	-
CONVERT FROM FIXED	あり	あり	あり	あり	あり
DIVIDE	あり	あり	あり	あり	あり
HALVE	あり	あり	-	あり	-
LOAD ¹	なし	なし	なし	なし	なし
LOAD AND TEST	なし	なし	あり	なし	なし
LOAD COMPLEMENT	なし	なし	あり	なし	なし
LOAD FP INTEGER	あり	あり	あり	あり	あり
LOAD LENGTHENED	なし	なし	あり	なし	なし
LOAD NEGATIVE	なし	なし	あり	なし	なし
LOAD POSITIVE	なし	なし	あり	あり	あり
LOAD ROUNDED	なし	なし	-	なし	-
LOAD ZERO ¹	-	あり	あり	あり	あり
MULTIPLY	あり	あり	あり	あり	あり
SQUARE ROOT	あり	あり	あり	あり	あり
STORE ¹	なし	なし	-	なし	-
SUBTRACT NORMALIZED	あり	あり/なし	あり/なし	あり	あり
SUBTRACT UNNORMALIZED	なし	あり/なし	-	あり	-

説明:

- 適用されません。

¹ 浮動小数点サポート命令。

あり/なし HFP 有効数字マスク・ビット (PSW のビット 23) が 0 のときは、強制的に真の 0 にされます。HFP 有効数字マスク・ビットが 1 のときは、指数部は変更されず、HFP 有効数字に関するプログラム割り込みが起きます。

図 18-1. 結果が HFP になる命令の場合の正規化と 0 の取り扱い

して 0 の小数部が生じたときの符号は、小数部が 0 以外である場合と同様に、オペランドの符号に基づいて決定されます。

正規化

HFP 数で数量を表すときに正規化を行えば、与えられた小数部の長さの範囲内で最大限の精度が達成されます。正規化された HFP 数では、左端の 16 進小数桁が 0 以外の値になります。小数部の左端の 1 桁以上に 0 があるとき、その数は正規化されていないと言われます。

非正規化数を正規化するときは、小数部が、左端の 16 進数字が 0 以外の数字になるまで、一度に 1 桁ずつ左にシフトされ、シフトした 16 進数字の数だけ指数部が小さくされます。小数部が 0 である数は、正規化することはできません。このような数の指数部は無変更のままになるか、または、結果が強制的に真の 0 にされるときは 0 になります。

拡張精度オペランドを使用した加算と減算、そして、MULTIPLY、DIVIDE、CONVERT FROM FIXED、HALVE、LOAD FP INTEGER、および SQUARE ROOT 演算を行うときは、必ず正規化が行われます。短精度または長精度のオペランドを使用した加算と減算については、正規化または非正規化のどちらも指定できます。その他のすべての操作では、正規化なしで結果が生成されます。

非正規化操作では、結果の小数部の左端にある 0 は除去されません。結果は、元のオペランドに応じて、正規化された形になることも、ならないこともあります。

正規化および非正規化のどちらの操作の場合も、初期オペランドが正規化された形になっている必要はありません。乗算、除算、および平方根演算では、演算の前にオペランドが正規化されます。正規化を伴うその他の操作では、中間演算結果が最終結果に変更される前に正規化が行われます。

加算、減算、または丸めの中間結果により小数部にオーバーフローが起きる場合は、小数部が 16 進数 1 桁だけ右にシフトされ、空いた左端の数字位置に 1 の値が入れます。その後で、小数部は最終結果の長さに合わせて切り捨てられ、指数部は 1 だけ増加します。正規化および非正規化のどちらの操作の場合も、この調整が行われます。

18-2 ページの図 18-1 は、HFP の結果を生じさせるすべての命令について、結果が 0 の場合の取り扱い、および結果が 0 以外の場合に正規化が行われるかどうかを、要約して示しています。

プログラミング上の注意: 非ゼロ・テストは左端の 16 進数字全体に適用されるので、正規化された数の小数部

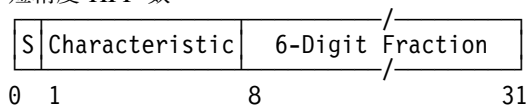
の左端ビットは、最大 3 ビットまでは 0 でもよいこととなります。

HFP データのフォーマット

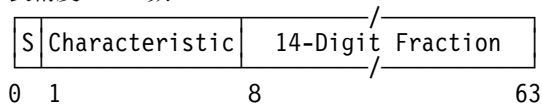
HFP 数には、32 ビット (短精度) フォーマット、64 ビット (長精度) フォーマット、または 128 ビット (拡張精度) フォーマットがあります。短精度および長精度フォーマットの数は、記憶域および浮動小数点レジスタのどちらの中でもオペランドとして指定できますが、拡張フォーマットのオペランドは浮動小数点レジスタの中でしか指定できません。

どのフォーマットでも、最初の ビット (ビット 0) は符号ビット (0) です。その次の 7 ビットは指数部です。短精度および長精度フォーマットでは、残りのビットは小数部を形成し、短精度の場合は 6 個の 16 進数字、そして長精度の場合は 14 個の 16 進数字です。

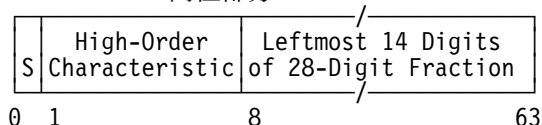
短精度 HFP 数



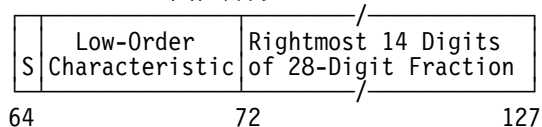
長精度 HFP 数



拡張精度 HFP 数
高位部分



低位部分



拡張精度の HFP 数は、28 桁の小数部を持ち、高位部分および低位部分と呼ばれる 2 つの長精度 HFP 数から成っています。高位部分は、任意の長精度 HFP 数です。高位部分の小数部には、28 桁の小数部の左端 14 個の 16 進数字が含まれます。高位部分の指数部と符号は、拡張精度 HFP 数の指数部と符号です。高位部分が正規化されていれば、拡張精度数は正規化されているものと見なされます。低位部分の小数部には、28 桁の小数部

の右端 14 個の数字が含まれます。拡張精度オペランドの低位部分の符号と指数部は無視されます。

結果が拡張精度フォーマットで生成され、レジスター・ペアに入れられるときは、低位部分の符号は高位部分の符号と同じにされ、そして、結果が真の 0 でない限り、低位指数部は高位指数部より 14 だけ小さくされます。このように 14 を減算した結果、低位指数部が 0 より小さくなる場合は、指数部はその正しい値より 128 だけ大きくなります。(したがって、減算はモジュロ 128 をとって行われます。) HFP の指数アンダーフローが示されるのは、高位指数部でアンダーフローが起きたときのみです。

拡張精度の結果が真の 0 にされるときは、高位部分と低位部分がどちらも真の 0 にされます。

正規化された HFP 数の大きさ (M) の範囲は、フォーマットによって異なります。

短精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-6}) \times 16^{63}$$

長精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-14}) \times 16^{63}$$

拡張精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-28}) \times 16^{63}$$

すべてのフォーマットの場合のおよその大きさ:

$$5.4 \times 10^{-79} \leq M \leq 7.2 \times 10^{75}$$

HFP 演算の最終結果では、小数部の 16 進桁数は、短精度フォーマットの場合は 6 桁、長精度フォーマットの場合は 14 桁、そして拡張精度フォーマットの場合は 28 桁ですが、中間結果では、それぞれの右側に余分な 16 進数字が 1 個付加されます。この数字は保護桁と呼ばれています。保護桁は、加算、減算、および比較操作、そして正規化の際の左へのシフトに関与するもので、最終結果の精度を高める働きをします。

レジスター対レジスター・バージョンの短精度、長精度、および拡張精度オペランドの場合、ならびにレジスター対記憶域バージョンの短精度および長精度オペランドについては、正規化されたされた結果に対して HFP 操作の全セットが使用可能です。ほとんどの命令では、ソース・オペランドと同じフォーマットの結果が生成されますが、一部の乗算操作では、短精度のオペランドから長精度の積が生成され、長精度のオペランドから拡張精度の積が生成されることがあります。その他の例外として、ある浮動小数点フォーマットから別の浮動小数点フォーマットへ、あるいは、浮動小数点フォーマットと

固定小数点 (2 進整数) フォーマットの間でオペランドを変換する命令があります。

プログラミング上の注意:

1. HFP 指数オーバーフローまたは HFP 指数アンダーフローがなく、拡張精度の結果の高位部分の指数部が 14 以上であれば、拡張精度の結果の低位部分を形成する長精度 HFP 数は、低位部分の値を正しく表しています。結果が真の 0 である場合も、同じことが言えます。高位指数部が 14 未満で、数値が真の 0 ではないときは、長精度 HFP 数と見なされる低位部分は、正しい指数部の値を表していません。
2. 拡張精度の結果の小数部は、その全体が正規化の対象とされます。低位部分のみをとってみれば、正規化された 28 桁の小数部の 15 桁目が 0 以外であるか 0 であるかに応じて、正規化された長精度 HFP 数と見なされる場合とそうでない場合があります。

命令

18-5 ページの図 18-2 に、HFP 命令と、それぞれのニーモニックおよび命令コードを示します。この図の「特性」の欄には、命令のフォーマット、どのようなときに条件コードが設定されるか、アクセス・レジスターを指定する命令フィールド、そして、オペランド指定、データ、または結果の中にどのような例外条件があるときにプログラム割り込みが起きるかを示してあります。

すべての HFP 命令は、AFP レジスター制御ビット (制御レジスター 0 のビット 45) により制御されます。オペランド位置の 1 つとして AFP レジスターが指定されている場合は、AFP レジスター制御ビットは 1 でなければなりません。さもないと、AFP レジスター・データ例外 (DXC 1) が認識されます。

RR、RRE、または RRF フォーマットの HFP 命令では、ニーモニックの最後の 1 文字は R です。また、HFP 命令については、オペランド・フォーマットの長さで正規化を表すための以下の文字を使用しています。

- F 32 ビット固定小数点
- G 64 ビット固定小数点
- D 長精度正規化
- E 短精度正規化
- U 短精度非正規化
- W 長精度非正規化
- X 拡張精度正規化

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペ

ランド指定が示されています。例えば、COMPARE (短精度) を使用したレジスタ間操作では、CER がニーモニックで、R1,R2 がオペランド指定です。

名称	ニーモニック	特性						命令コード	
ADD NORMALIZED (拡張精度 HFP)	AXR	RR	C	SP	Da	EU	EO	LS	B ₂ 36 2A 6A 3A 7A
ADD NORMALIZED (長精度 HFP)	ADR	RR	C		Da	EU	EO	LS	
ADD NORMALIZED (長精度 HFP)	AD	RX	C	A	Da	EU	EO	LS	
ADD NORMALIZED (短精度 HFP)	AER	RR	C		Da	EU	EO	LS	
ADD NORMALIZED (短精度 HFP)	AE	RX	C	A	Da	EU	EO	LS	
ADD UNNORMALIZED (長精度 HFP)	AWR	RR	C		Da	EO	LS	B ₂ 2E 6E 3E 7E B369	
ADD UNNORMALIZED (長精度 HFP)	AW	RX	C	A	Da	EO	LS		
ADD UNNORMALIZED (短精度 HFP)	AUR	RR	C		Da	EO	LS		
ADD UNNORMALIZED (短精度 HFP)	AU	RX	C	A	Da	EO	LS		
COMPARE (拡張精度 HFP)	CXR	RRE	C	SP	Da				
COMPARE (長精度 HFP)	CDR	RR	C		Da			B ₂ 29 69 39 79 B3B6	
COMPARE (長精度 HFP)	CD	RX	C	A	Da				
COMPARE (短精度 HFP)	CER	RR	C		Da				
COMPARE (短精度 HFP)	CE	RX	C	A	Da				
CONVERT FROM FIXED (32 から拡張精度 HFP へ)	CXFR	RRE		SP	Da				
CONVERT FROM FIXED (32 から長精度 HFP へ)	CDFR	RRE			Da			B3B5 B3B4 B3C6 B3C5 B3C4	
CONVERT FROM FIXED (32 から短精度 HFP へ)	CEFR	RRE			Da				
CONVERT FROM FIXED (64 から拡張精度 HFP へ)	CXGR	RRE	N	SP	Da				
CONVERT FROM FIXED (64 から長精度 HFP へ)	CDGR	RRE	N		Da				
CONVERT FROM FIXED (64 から短精度 HFP へ)	CEGR	RRE	N		Da				
CONVERT TO FIXED (拡張精度 HFP から 32 へ)	CFXR	RRF	C	SP	Da			B3BA B3B9 B3B8 B3CA B3C9	
CONVERT TO FIXED (長精度 HFP から 32 へ)	CFDR	RRF	C	SP	Da				
CONVERT TO FIXED (短精度 HFP から 32 へ)	CFER	RRF	C	SP	Da				
CONVERT TO FIXED (拡張精度 HFP から 64 へ)	CGXR	RRF	C N	SP	Da				
CONVERT TO FIXED (長精度 HFP から 64 へ)	CGDR	RRF	C N	SP	Da				
CONVERT TO FIXED (短精度 HFP から 64 へ)	CGER	RRF	C N	SP	Da			B3C8 B22D 2D 6D 3D	
DIVIDE (拡張精度 HFP)	DXR	RRE		SP	Da	EU	EO		FK
DIVIDE (長精度 HFP)	DDR	RR			Da	EU	EO		FK
DIVIDE (長精度 HFP)	DD	RX		A	Da	EU	EO		FK
DIVIDE (短精度 HFP)	DER	RR			Da	EU	EO		FK
DIVIDE (短精度 HFP)	DE	RX		A	Da	EU	EO	FK	B ₂ 7D 24 34 B362 22
HALVE (長精度 HFP)	HDR	RR			Da	EU			
HALVE (短精度 HFP)	HER	RR			Da	EU			
LOAD AND TEST (拡張精度 HFP)	LTXR	RRE	C	SP	Da				
LOAD AND TEST (長精度 HFP)	LTDR	RR	C		Da				
LOAD AND TEST (短精度 HFP)	LTER	RR	C		Da			32 B363 23 33 B367	
LOAD COMPLEMENT (拡張精度 HFP)	LCXR	RRE	C	SP	Da				
LOAD COMPLEMENT (長精度 HFP)	LCDR	RR	C		Da				
LOAD COMPLEMENT (短精度 HFP)	LCER	RR	C		Da				
LOAD FP INTEGER (拡張精度 HFP)	FIXR	RRE		SP	Da				

図 18-2 (1/3). HFP 命令の要約

名称	二 モ ニ ツ ク	特性						命 令 コ ー ド
LOAD FP INTEGER (長精度 HFP) LOAD FP INTEGER (短精度 HFP) LOAD LENGTHENED (長精度から拡張精度 HFP ^) LOAD LENGTHENED (長精度から拡張精度 HFP ^) LOAD LENGTHENED (短精度から拡張精度 HFP ^)	FIDR FIER LXDR LXD LXER	RRE RRE RRE RXE RRE		SP SP SP	Da Da Da Da Da			B37F B377 B325 ED25 B326
LOAD LENGTHENED (短精度から拡張精度 HFP ^) LOAD LENGTHENED (短精度から長精度 HFP ^) LOAD LENGTHENED (短精度から長精度 HFP ^) LOAD NEGATIVE (拡張精度 HFP) LOAD NEGATIVE (長精度 HFP)	LXE LDER LDE LNXR LNDR	RXE RRE RXE RRE C RR C	A	SP SP	Da Da Da Da Da		B2 B2	ED26 B324 ED24 B361 21
LOAD NEGATIVE (短精度 HFP) LOAD POSITIVE (拡張精度 HFP) LOAD POSITIVE (長精度 HFP) LOAD POSITIVE (短精度 HFP) LOAD ROUNDED (拡張精度から長精度 HFP ^)	LNER LPXR LPDR LPER LDXR	RR C RRE C RR C RR C RR		SP SP	Da Da Da Da Da E0			31 B360 20 30 25
LOAD ROUNDED (拡張精度から長精度 HFP ^) LOAD ROUNDED (拡張精度から短精度 HFP ^) LOAD ROUNDED (長精度から短精度 HFP ^) LOAD ROUNDED (長精度から短精度 HFP ^) MULTIPLY (拡張精度 HFP)	LRDR LEXR LEDR LRER MXR	RR RRE RR RR RR		SP SP SP	Da E0 Da E0 Da E0 Da E0 Da EU E0			25 B366 35 35 26
MULTIPLY (長精度 HFP) MULTIPLY (長精度 HFP) MULTIPLY (長精度から拡張精度 HFP ^) MULTIPLY (長精度から拡張精度 HFP ^) MULTIPLY (短精度 HFP)	MDR MD MXDR MXD MEER	RR RX RR RX RRE	A	SP SP	Da EU E0 Da EU E0 Da EU E0 Da EU E0 Da EU E0		B2 B2	2C 6C 27 67 B337
MULTIPLY (短精度 HFP) MULTIPLY (短精度から長精度 HFP ^) MULTIPLY (短精度から長精度 HFP ^) MULTIPLY (短精度から長精度 HFP ^) MULTIPLY (短精度から長精度 HFP ^)	MEE MDER MER MDE ME	RXE RR RR RX RX	A A		Da EU E0 Da EU E0 Da EU E0 Da EU E0 Da EU E0		B2 B2 B2 B2	ED37 3C 3C 7C 7C
SQUARE ROOT (拡張精度 HFP) SQUARE ROOT (長精度 HFP) SQUARE ROOT (長精度 HFP) SQUARE ROOT (短精度 HFP) SQUARE ROOT (短精度 HFP)	SQXR SQDR SQD SQER SQE	RRE RRE RXE RRE RXE		SP	Da SQ Da SQ Da SQ Da SQ Da SQ			B336 B244 ED35 B245 ED34
SUBTRACT NORMALIZED (拡張精度 HFP) SUBTRACT NORMALIZED (長精度 HFP) SUBTRACT NORMALIZED (長精度 HFP) SUBTRACT NORMALIZED (短精度 HFP) SUBTRACT NORMALIZED (短精度 HFP)	SXR SDR SD SER SE	RR C RR C RX C RR C RX C		SP	Da EU E0 LS Da EU E0 LS Da EU E0 LS Da EU E0 LS Da EU E0 LS			37 2B B2 6B 3B B2 7B
SUBTRACT UNNORMALIZED (長精度 HFP) SUBTRACT UNNORMALIZED (長精度 HFP) SUBTRACT UNNORMALIZED (短精度 HFP) SUBTRACT UNNORMALIZED (短精度 HFP)	SWR SW SUR SU	RR C RX C RR C RX C	A		Da E0 LS Da E0 LS Da E0 LS Da E0 LS			2F B2 6F 3F B2 7F

図 18-2 (2/3). HFP 命令の要約

説明:

- A 論理アドレスについてのアクセス例外。
- B2 アクセス・レジスター・モードでは、B2 フィールドはアクセス・レジスターを指定します。
- C 条件コードが設定されます。
- Da AFP レジスター・データ例外。
- EO HFP 指数オーバーフロー例外。
- EU HFP 指数アンダーフロー例外。
- FK HFP 除算例外。
- HX HFP 拡張機能。
- LS HFP 有効数字例外。
- N z/Architecture の新規の命令で、ESA/390 にはないものです。
- QR 平方根機能。
- RR RR 命令フォーマット。
- RRE RRE 命令フォーマット。
- RRF RRF 命令フォーマット。
- RX RX 命令フォーマット。
- RXE RXE 命令フォーマット。
- SP 指定例外。
- SQ HFP 平方根例外。

図 18-2 (3/3). HFP 命令の要約

ADD NORMALIZED

ニーモニック 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

0 8 12 15

ニーモニック 1 命令コード オペランド

AER	'3A'	短精度 HFP
ADR	'2A'	長精度 HFP
AXR	'36'	拡張精度 HFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16 20	31

0 8 12 16 20 31

ニーモニック 2 命令コード オペランド

AE	'7A'	短精度 HFP
AD	'6A'	長精度 HFP

第 2 オペランドが第 1 オペランドに加算され、正規化された和が第 1 オペランド位置に入れられます。

2 つの HFP 番号の加算では、指数部が比較され、小数部が位置合わせされ、その後、符号付きの小数部の加算が行われます。2 つのオペランドの指数部が比較され、小さい方の指数部に付随している小数部が右にシフトされて、もう一方の小数部と位置合わせされます。1 つの 16 進数字がシフトされるたびに指数部は 1 ずつ増加し、2 つの指数部が一致するまでシフトが続けられます。

位置合わせのために小数部が右にシフトされるときに、外へ送り出された左端の 16 進数字は、保護桁として保存されます。シフトされなかった方の小数部は、保護桁

の位置に 0 を入れて拡張されたものと見なされます。位置合わせのためのシフトが行われなときは、どちらのオペランドも、保護桁の位置に 0 を入れて拡張されるものと見なされます。次に、2 つの符号付き小数部が代数的に加算されて、符号付きの中間和が求められます。

中間和の小数部は、保護桁と繰り上がりも含めて、7 桁 (短精度フォーマット)、15 桁 (長精度フォーマット)、または 29 桁 (拡張精度フォーマット) の 16 進数字から成っています。繰り上がりがある場合は、繰り上がりが小数部の左端の数字になるように、和は 1 桁右へシフトされ、指数部は 1 だけ増加します。

加算による繰り上がりがなく、中間和の小数部が 0 でない場合は、加算の結果生じる先行の 16 進 0 を除去するために、小数部が必要なだけ左にシフトされます。空になった右端桁には 0 が入れられ、指数部は、シフトされた 16 進数字の数だけ減少します。このようにして正規化された小数部は、右側が切り捨てられて、6 桁 (短精度フォーマット)、14 桁 (長精度フォーマット)、または 28 桁 (拡張精度フォーマット) の 16 進数にされます。拡張精度フォーマットでは、低位部分の指数部も生成されます。これは、高位部分の指数部より 14 だけ小さい値になります。

和の符号は、代数の規則に従って決定されます。ただし、中間和の小数部が全桁 0 である場合は、結果は正の真の 0 になります。

中間和の小数部の左端桁からの繰り上がりにより、正規化された和の指数部が 127 を超える場合は、HFP 指数オーバーフロー例外が起きます。この場合は、結果の指数部が正しい値より 128 だけ小さくされて操作は完了し、HFP 指数オーバーフローに関するプログラム割り

込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。拡張精度の結果の場合は、低位部分の指数部も正しい値を維持します。

正規化された和の指数部が 0 より小さくなり、小数部が 0 でないときは、HFP 指数アンダーフロー例外が生じています。この場合、PSW 内の HFP 指数アンダーフロー・マスク・ビットが 1 であれば、結果の指数部が正しい値より 128 だけ大きくされて操作は完了し、HFP 指数アンダーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。PSW 内の HFP 指数アンダーフロー・マスク・ビットが 0 の場合は、プログラム割り込みは起きず、代わりに、結果が正の真の 0 にされて操作は完了します。拡張精度の結果において、低位指数部が 0 より小さくても、高位指数部が 0 以上であれば、HFP 指数アンダーフローは認識されません。

中間和の小数部 (保護桁も含む) が 0 のときは、結果の小数部は 0 です。結果の小数部が 0 のときのアクションは、PSW の HFP 有効数字マスク・ビットの設定により異なります。PSW の HFP 有効数字マスク・ビットが 1 のときは、正規化は行われず、中間結果と最終結果の指数部は同じであり、HFP 有効数字に関するプログラム割り込みが起きます。PSW の HFP 有効数字マスク・ビットが 0 のときは、プログラム割り込みは起きず、代わりに、結果は正の真の 0 にされます。

AXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: AE および AD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- HFP 指数オーバーフロー例外
- HFP 指数アンダーフロー例外
- HFP 有効数字例外
- 指定例外 (AXR のみ)

プログラミング上の注意:

1. 付録Aに、ADD NORMALIZED 命令 (AE) の使用例を示してあります。
2. HFP の加算では、2 つのオペランドの値を入れ替えても和の値は変わりません。
3. ADD NORMALIZED 命令は、和を正規化しますが、オペランドは正規化しません。したがって、一方または両方のオペランドが正規化されていない場合は、小数部の位置合わせの際に精度が失われることがあります。

ADD UNNORMALIZED

ニーモニック 1 R₁,R₂ [RX]

Op Code	R ₁	R ₂
0	8	12 15

0 8 12 15

ニーモニック 1 命令コード オペランド
 AUR '3E' 短精度 HFP
 AWR '2E' 長精度 HFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16 20	31

0 8 12 16 20 31

ニーモニック 2 命令コード オペランド
 AU '7E' 短精度 HFP
 AW '6E' 長精度 HFP

第 2 オペランドが第 1 オペランドに加算され、正規化されないままの和が第 1 オペランド位置に入られます。

ADD UNNORMALIZED の実行は ADD NORMALIZED の実行とほぼ同じですが、以下の点が異なります。

1. 加算の結果繰り上がりが生じないときは、中間和の小数部は、結果の小数部の適切な長さに合わせて切り捨てられます。先行する 16 進 0 を除去するために左にシフトする処理も、それに対応して指数部を減少させる処理も行われません。
2. HFP 指数アンダーフローは起きません。
3. 結果の小数部が 0 であるかどうかの認識には、保護桁は考慮されません。小数部 (中間和の小数部から保護桁を除いたもの) が 0 であれば、結果の小数部は 0 であるものとして認識されます。

結果の条件コード:

- 0 結果の小数部が 0。
- 1 結果が 0 より小さい。
- 2 結果が 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: AU および AW のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスタ例外
- HFP 指数オーバーフロー例外
- HFP 有効数字例外

プログラミング上の注意:

1. 『付録A』に、ADD UNNORMALIZED 命令 (AU) の使用例を示してあります。
2. 結果が真の 0 にされる場合を除けば、ADD UNNORMALIZED の結果の指数部は、2 つのオペランドの指数部のいずれか大きい方に等しいか、あるいは、小数部の加算の結果繰り上がりがかかる場合は 1 だけ増加し、また、HFP 指数オーバーフローが起きたときは 0 に設定されます。

COMPARE

ニーモニック 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
---------	----------------	----------------

0 8 12 15

ニーモニック 1 命令コード オペランド
 CER '39' 短精度 HFP
 CDR '29' 長精度 HFP

ニーモニック 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
---------	----------	----------------	----------------

0 16 24 28 31

ニーモニック 2 命令コード オペランド
 CXR 'B369' 拡張精度 HFP

ニーモニック 3 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
---------	----------------	----------------	----------------	----------------

0 8 12 16 20 31

ニーモニック 3 命令コード オペランド
 CE '79' 短精度 HFP
 CD '69' 長精度 HFP

第 1 オペランドが第 2 オペランドと比較され、その結果を示す条件コードが設定されます。

比較は、正規化を伴う減算と同じ手順で代数的に行われますが、条件コードが設定された後は、差は廃棄され、オペランドはどちらも変更されません。保護桁も含めて差が 0 であれば、両オペランドは等しいと見なされます。差が 0 以外の場合は、差が正であれば第 1 オペランドの方が大きく、差が負であれば第 1 オペランドの方が小さいことになります。

HFP 指数オーバーフロー例外、HFP 指数アンダーフロー例外、または HFP 有効数字例外が起きることはありません。

CXR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: CE および CD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外 (CXR のみ)

プログラミング上の注意:

1. 付録Aに、COMPARE 命令 (CDR) の使用例を示してあります。
2. 指数が等しくないというだけでは、同じ符号を持つ 2 つのオペランドが等しいと断定することはできません。なぜなら、小数部に含まれる先行 16 進 0 の数が違っている場合があるからです。
3. 小数部が 0 である数値同士の比較結果は、符号または指数部が違っていても、等しいと見なされます。

CONVERT FROM FIXED

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
CEFR	'B3B4'	32 ビット 2 進整数オペランド、短精度 HFP 結果
CDFR	'B3B5'	32 ビット 2 進整数オペランド、長精度 HFP 結果
CXFR	'B3B6'	32 ビット 2 進整数オペランド、拡張精度 HFP 結果
CEGR	'B3C4'	64 ビット 2 進整数オペランド、短精度 HFP 結果
CDGR	'B3C5'	64 ビット 2 進整数オペランド、長精度 HFP 結果
CXGR	'B3C6'	64 ビット 2 進整数オペランド、拡張精度 HFP 結果

第 2 オペランドの固定小数点数が HFP フォーマットに変換され、その結果が正規化されて第 1 オペランドの位置に入れられます。

0 以外の結果は正規化されます。0 の結果は、正の真の 0 にされます。

第 2 オペランドは、R₂ に指定されている汎用レジスタ内にある符号付き 2 進整数です。このレジスタのビット位置 32-63 が、32 ビットのオペランドを表しています。

結果は正規化され、0 の方向へ丸められて (切り捨てられて) から、第 1 オペランドの位置に入れられます。

CXFR および CXGR の場合は、R₁ フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外 (CXFR および CXGR)

CONVERT TO FIXED

ニーモニック R₁,M₃,R₂ [RRF]

Op Code	M ₃	////	R ₁	R ₂
0	16	20	24	28 31

ニーモニック	命令コード	オペランド
CFER	'B3B8'	短精度 HFP オペランド、32 ビット 2 進整数結果
CFDR	'B3B9'	長精度 HFP オペランド、32 ビット 2 進整数結果
CFXR	'B3BA'	拡張精度 HFP オペランド、32 ビット 2 進整数結果
CGER	'B3C8'	短精度 HFP オペランド、64 ビット 2 進整数結果
CGDR	'B3C9'	長精度 HFP オペランド、64 ビット 2 進整数結果
CGXR	'B3CA'	拡張精度 HFP オペランド、64 ビット 2 進整数結果

第 2 オペランドの HFP が整数値に丸められてから、固定小数点フォーマットに変換されます。その結果が第 1 オペランドの位置に入れられます。

結果は符号付き 2 進整数で、R₁ に指定されている汎用レジスタに入れられます。このレジスタのビット 32-63 が 32 ビットの結果で置き換えられ、ビット 0-31 は変更されません。

第 2 オペランドは、M₃ フィールド内の修飾子の指定に従って、整数値に丸められます。

M₃ 丸め方式

- 0 0 方向への丸め
- 1 バイアス付き最近値丸め
- 4 最近値丸め
- 5 0 方向への丸め
- 6 +∞ 方向への丸め
- 7 -∞ 方向への丸め

0、1、または 4-7 以外の修飾子は無効です。

結果の符号は第 2 オペランドの符号と同じですが、結果が 0 のときは正符号になります。

丸めた結果の値が結果のフォーマットで表せる範囲を超えている場合は、表現可能な最大の絶対値にソースと同じ符号を付けた値がターゲット位置に入れられ、条件コード 3 が設定されます。

小数值は、丸めの方式に従って、1 (適切な符号付き) または 0 に丸められるので、HFP 指数アンダーフローが認識されることはありません。

M3 フィールドには有効な修飾子を指定しなければなりません。さもないと、指定例外が認識されます。CFXR および CGXR の場合は、R2 フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 ソースが 0。
- 1 ソースが 0 より小さい。
- 2 ソースが 0 より大きい。
- 3 特殊ケース。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外

DIVIDE

ニーモニク 1 R₁, R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニク 1 命令コード オペランド
 DER '3D' 短精度 HFP
 DDR '2D' 長精度 HFP

ニーモニク 2 R₁, R₂ [RRE]

'B22D'	////////	R ₁	R ₂
0	16	24	28 31

ニーモニク 2 命令コード オペランド
 DXR 'B22D' 拡張精度 HFP

ニーモニク 3 R₁, D₂ (X₂, B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16 20	31

ニーモニク 3 命令コード オペランド
 DE '7D' 短精度 HFP
 DD '6D' 長精度 HFP

第 1 オペランド (被除数) が第 2 オペランド (除数) で除算され、正規化された商が第 1 オペランドの位置に入れられます。剰余は保存されません。

HFP の除算では、指数部の減算と小数部の除算が行われます。オペランドは、先行の 16 進 0 を除去するために、最初に正規化されます。正規化されたオペランドの被除数の指数部と除数の指数部との差に 64 を加えた値が、中間結果の商として使用されます。

中間結果の商の小数部を形成するために、被除数および除数の小数部のすべての桁が使用されます。中間結果の商の小数部には先行 16 進 0 はありませんが、1 桁右へのシフトが必要になることがあり、その場合は指数部が 1 だけ増加します。その後で、小数部は結果の小数部の適切な長さに合わせて切り捨てられます。

最終の商の指数部が 127 を超えることになり、小数部が 0 でない場合は、HFP 指数オーバーフロー例外が起きます。この場合は、結果の指数部が正しい値より 128 だけ小さくされて操作は完了し、HFP 指数オーバーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。拡張精度の結果において、低位指数部も 127 を超える場合は、その指数部も 128 だけ小さくされます。

最終結果の商の指数部が 0 より小さくなり、小数部が 0 でないときは、HFP 指数アンダーフロー例外が起きます。この場合、PSW 内の HFP 指数アンダーフロー・マスク・ビットが 1 であれば、結果の指数部が正しい値より 128 だけ大きくされて操作は完了し、HFP 指数アンダーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。PSW 内の HFP 指数アンダーフロー・マスク・ビットが 0 の場合は、プログラム割り込みは起きず、代わりに、結果が正の真の 0 にされて操作は完了します。拡張精度の結果において、低位指数部が 0 より小さくても、高位指数部が 0 以上であれば、HFP 指数アンダーフローは認識されません。

オペランドを正規化するときオペランドの 1 つが 0 より小さくなる場合、または中間結果の商の指数部が 0 より小さい場合は、最終結果の商が正しい指数部により表現できるのであれば、HFP 指数アンダーフローは起きません。

除数の小数部が 0 のときは、HFP 除算例外が認識されます。これには、0 を 0 で割る場合も含まれます。

被除数の小数部が 0 で、除数の小数部が 0 でないときは、商は正の真の 0 にされます。HFP 指数オーバーフローまたは HFP 指数アンダーフローは起きません。

商の符号は、オペランドの符号の排他論理和ですが、商が正の真の 0 にされるときは、符号は常に正です。

DXR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: DE および DD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスタ例外
- HFP 除算例外
- HFP 指数オーバーフロー例外
- HFP 指数アンダーフロー例外
- 指定例外 (DXR のみ)

プログラミング上の注意:

付録Aに、DIVIDE 命令 (DER) の使用例を示してあります。

HALVE

ニーモニック R_1, R_2 [RR]

Op Code	R1	R2
0	8	12 15

ニーモニック	命令コード	オペランド
HER	'34'	短精度 HFP
HDR	'24'	長精度 HFP

第 2 オペランドが 2 で除算され、正規化された商が第 1 オペランドの位置に入られます。

第 2 オペランドの小数部が 1 ビット位置だけ右へシフトされ、右端のビット位置の内容が保護桁の左端のビット位置に入れられ、小数部の左端のビット位置には 0 が入ります。そして、中間結果 (保護桁も含む) が正規化され、最終結果は適正な長さに切り捨てられます。

最終結果の指数部が 0 より小さくなり、小数部が 0 でないときは、HFP 指数アンダーフロー例外が起きます。この場合、PSW 内の HFP 指数アンダーフロー・マスク・ビットが 1 であれば、結果の指数部が正しい値より 128 だけ大きくされて操作は完了し、HFP 指数アンダーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。PSW 内の HFP 指数アンダーフロー・マスク・ビットが 0 の場合は、プログラム割り込みは起きず、代わりに、結果が正の真の 0 にされて操作は完了します。

第 2 オペランドの小数部が 0 のときは、結果は正の真の 0 にされ、HFP 指数アンダーフローは起きません。

結果の符号は第 2 オペランドと同じですが、商が正の真の 0 にされるときは、符号は常に正です。

条件コード: コードは変更されません。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- HFP 指数アンダーフロー例外

プログラミング上の注意:

1. 付録Aに、HALVE 命令 (HDR) の使用例を示してあります。
2. 短精度および長精度のオペランドの場合は、HALVE 命令は、2 を除数とする除算操作と同じです。同様に、HDR の結果は、2/1 を乗数とする MD または MDR の結果に等しく、また、HER の結果は、1/2 を乗数とする MEE または MEER の結果に等しくなります。

LOAD AND TEST

ニーモニック 1 R_1, R_2 [RR]

Op Code	R1	R2
0	8	12 15

ニーモニック 1	命令コード	オペランド
LTER	'32'	短精度 HFP
LTDR	'22'	長精度 HFP

ニーモニック 2 R_1, R_2 [RRE]

Op Code	////////	R1	R2
0	16	24	28 31

ニーモニック 2	命令コード	オペランド
LTXR	'B362'	拡張精度 HFP

第 2 オペランドが第 1 オペランドの位置に入れられ、その符号と大きさがテストされ、その結果に基づいて条件コードが設定されます。条件コードは、第 2 オペランドを 0 として比較を行った場合と同じに設定されます。

短精度および長精度のオペランドの場合は、第 2 オペランドは無変更のまま第 1 オペランド位置に入れられます。

拡張精度のオペランドの場合は、ソースの高位部分の符号および小数部全体が無変更のまま結果に入れられ、低位部分の符号は高位部分の符号と同じに設定されます。拡張精度オペランドの小数部が 0 以外である場合は、高位指数部が無変更のまま結果の高位指数部に入れられ、低位指数部は、モジュロ 128 をとって、高位指数部より 14 だけ小さい値に設定されます。拡張精度オペランドの小数部が 0 の場合は、結果は、ソースと同じ符号を持つ真の 0 にされます (結果の高位符号ビットおよび低位符号ビットは、ソースの高位符号ビットと同じになります)。

LTXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 --

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外 (LTXR のみ)

プログラミング上の注意: LTER および LTDR で、第 1 オペランドと第 2 オペランドに同じレジスターを指定した場合は、データ移動を伴わないテストと同じ操作が行われることになります。

LOAD COMPLEMENT

ニーモニック 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニック 1 命令コード オペランド
 LCER '33' 短精度 HFP
 LCDR '23' 長精度 HFP

ニーモニック 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 2 命令コード オペランド
 LCXR 'B363' 拡張精度 HFP

第 2 オペランドが、符号ビットを反転して第 1 オペランド位置に入れられます。

オペランドが 0 の場合でも、符号ビットは反転されません。どのオペランド長の場合も、ソースの小数部は無変更のまま結果に入れられます。

短精度および長精度のオペランドの場合は、ソースの指数部は無変更のまま結果に入れられます。

拡張精度オペランドの場合は、低位符号は高位符号と同じに設定されます。拡張精度オペランドの小数部が 0 以外である場合は、高位指数部が無変更のまま結果の高位指数部に入れられ、低位指数部は、モジュロ 128 をとって、高位指数部より 14 だけ小さい値に設定されます。拡張精度オペランドの小数部が 0 の場合は、結果は、ソースと逆の符号を持つ真の 0 にされます (結果の高位符号ビットおよび低位符号ビットは、ソースの高位符号ビットの逆になります)。

LCXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 --

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外 (LCXR のみ)

LOAD FP INTEGER

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 命令コード オペランド
 FIER 'B377' 短精度 HFP
 FIDR 'B37F' 長精度 HFP
 FIXR 'B367' 拡張精度 HFP

第 2 オペランドが、同じ浮動小数点フォーマットの整数値に切り捨てられ (0 方向に丸められる)、正規化された結果が第 1 オペランド位置に入れられます。

0 以外の結果は正規化されます。0 の結果は、正の真の 0 にされます。

FIXR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外 (FIXR のみ)

プログラミング上の注意:

1. LOAD FP INTEGER は、HFP 数を整数値に切り捨てます (0 方向への丸め)。これらの整数 (HFP フォーマットのまま残される) を、2 進整数 (固定小数点フォーマットを使用する) と混同しないでください。
2. HFP オペランドが、すでに整数となるだけの十分な大きさの指数を持つ数値である場合は、正規化されていないオペランドが正規化され、小数部が 0 のオペランドが正の真の 0 に変更される点を除けば、結果の値は変わりません。

LOAD LENGTHENED

ニーモニック 1 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 1	命令コード	オペランド
LDER	'B324'	短精度 HFP オペランド 2、 長精度 HFP オペランド 1
LXDR	'B325'	長精度 HFP オペランド 2、 拡張精度 HFP オペランド 1
LXER	'B326'	短精度 HFP オペランド 2、 拡張精度 HFP オペランド 1

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	D ₂	////////	Op Code
0	8	12	16	20	32	40 47

ニーモニック 2	命令コード	オペランド
LDE	'ED24'	短精度 HFP オペランド 2、 長精度 HFP オペランド 1
LXD	'ED25'	長精度 HFP オペランド 2、 拡張精度 HFP オペランド 1
LXE	'ED26'	短精度 HFP オペランド 2、 拡張精度 HFP オペランド 1

第 2 オペランドが精度の高いフォーマットに拡張され、結果が第 1 オペランド位置に入れられます。

どのオペランド長の場合も、ソースの小数部が 0 で拡張されて結果に入れられます。結果の符号ビットは、結果が真の 0 にされる場合でも、ソースの符号と同じに設定されます。

長精度の結果の場合は、ソースの指数部は無変更のまま結果に入れられます。

拡張精度結果の場合は、低位符号は高位符号と同じに設定されます。小数部が 0 以外である場合は、ソースの指数部が無変更のまま結果の高位指数部に入れられ、低位指数部は、モジュロ 128 をとって、高位指数部より 14 だけ小さい値に設定されます。小数部が 0 の場合は、結果は、ソースと同じ符号を持つ真の 0 にされます (結果の高位符号ビットおよび低位符号ビットは、ソースの符号ビットと同じになります)。

LXD、LXDR、LXE、および LXER の場合は、R₁ フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: LDE、LXE、および LXD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外 (LXE、LXER、LXD、LXDR)

LOAD NEGATIVE

ニーモニク 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニク 1 命令コード オペランド
LNER '31' 短精度 HFP
LNDR '21' 長精度 HFP

ニーモニク 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニク 2 命令コード オペランド
LNXR 'B361' 拡張精度 HFP

第 2 オペランドが、符号ビットを 1 に設定して第 1 オペランド位置に入れられます。

オペランドが 0 の場合でも、符号ビットは 1 に設定されます。どのオペランド長の場合も、ソースの小数部は無変更のまま結果に入れられます。

短精度および長精度のオペランドの場合は、ソースの指数部は無変更のまま結果に入れられます。

拡張精度オペランドの場合は、低位符号は高位符号と同じに設定されます。拡張精度オペランドの小数部が 0 以外である場合は、高位指数部が無変更のまま結果の高位指数部に入れられ、低位指数部は、モジュロ 128 をとって、高位指数部より 14 だけ小さい値に設定されま
す。 拡張精度オペランドの小数部が 0 の場合は、結果は負の真の 0 にされます (結果の高位符号ビットおよび低位符号ビットは 1 に設定されます)。

LNXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 --
- 3 --

プログラム例外:

- データ例外 (DXC 1)、AFP レジスター例外
- 指定例外 (LNXR のみ)

LOAD POSITIVE

ニーモニク 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニク 1 命令コード オペランド
LPER '30' 短精度 HFP
LPDR '20' 長精度 HFP

ニーモニク 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニク 2 命令コード オペランド
LPXR 'B360' 拡張精度 HFP

第 2 オペランドが、符号ビットを 0 に設定して第 1 オペランド位置に入れられます。

どのオペランド長の場合も、符号ビットが 0 に設定されて、ソースの小数部が無変更のまま結果に入れられま
す。

短精度および長精度のオペランドの場合は、ソースの指数部は無変更のまま結果に入れられます。

拡張精度オペランドの場合は、低位符号は高位符号と同じに設定されます。拡張精度オペランドの小数部が 0 以外である場合は、高位指数部が無変更のまま結果の高位指数部に入れられ、低位指数部は、モジュロ 128 をとって、高位指数部より 14 だけ小さい値に設定されま
す。 拡張精度オペランドの小数部が 0 の場合は、結果は正の真の 0 にされます (結果の高位符号ビットおよび低位符号ビットは 0 に設定されます)。

LPXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 --
- 2 結果は 0 より大きい。
- 3 --

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- 指定例外 (LPXR のみ)

LOAD ROUNDED

ニーモニック 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニック 1 命令コード オペランド
LEDR '35' 長精度 HFP オペラ
LDXR '25' HFP オペランド 1
拡張精度 HFP オペラ
ランド 2、短精度
HFP オペランド 1

上記のニーモニックは、オペランド長がはっきり示されていない下記の古いニーモニックに取って代わるものです。

LRER '35' 長精度 HFP オペラ
LRDR '25' 拡張精度 HFP オペラ
ランド 2、短精度
HFP オペランド 1

ニーモニック 2 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 2 命令コード オペランド
LEXR 'B366' 拡張精度 HFP オペラ
ランド 2、短精度
HFP オペランド 1

第 2 オペランドが精度の低いフォーマットに丸められ、結果が第 1 オペランド位置に入れられます。

この丸めでは、第 2 オペランドの左端ビット位置 (除去するビット) に 1 が加えられ、小数部に繰り上がりがある場合はそれが反映されます。第 2 オペランドの符号は無視され、小数部が正であるものとして加算が行われます。

丸めにより、小数部の左端の 16 進数字位置からの繰り上がりが起きる場合は、小数部が 1 桁ずつ右にシフトさ

れて、繰り上がりが小数部の左端の桁になり、指数部は 1 だけ増加します。

次に、中間結果の小数部が、結果の小数部の適切な長さに合わせて切り捨てられます。

結果の符号は第 2 オペランドの符号と同じです。先行ゼロを除去するための正規化は行われません。

小数部を右へシフトすることにより、指数部が 127 を超えることになる場合は、HFP 指数オーバーフロー例外が起きます。この場合は、結果の指数部が正しい値より 128 だけ小さくされて操作は完了し、HFP 指数オーバーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。

HFP 指数アンダーフロー例外および HFP 有効数字例外が起きることはありません。

LDXR (または LRDR) および LEXR の場合は、R₂ フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- データ例外 (DXC 1)、AFP レジスタ例外
- HFP 指数オーバーフロー例外
- 指定例外 (LDXR、LEXR、LRDR)

プログラミング上の注意: 丸められた結果の符号は、結果が 0 であっても、オペランドの符号と同じになります。

MULTIPLY

ニーモニック 1 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 1 命令コード オペランド
MEER 'B337' 短精度 HFP

ニーモニック 2 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニック 2 命令コード オペランド
 MDR '2C' 長精度 HFP
 MXR '26' 拡張精度 HFP
 MDER '3C' 短精度 HFP 乗数
 および被乗数、
 長精度 HFP 積
 MXDR '27' 長精度 HFP 乗数
 および被乗数、
 拡張精度 HFP 積

上記のニーモニック MDER は、オペランド長がはっきり示されていない下記の古いニーモニックに取って代わるものです。

MER '3C' 短精度 HFP 乗数
 および被乗数、
 長精度 HFP 積

ニーモニック 3 R₁,D₂(X₂,B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	D ₂	////////	Op Code
0	8	12	16	20	32	40 47

ニーモニック 3 命令コード オペランド
 MEE 'ED37' 短精度 HFP

ニーモニック 4 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20 31

ニーモニック 4 命令コード オペランド
 MD '6C' 長精度 HFP
 MDE '7C' 短精度 HFP 乗数
 および被乗数、
 長精度 HFP 積
 MXD '67' 長精度 HFP 乗数
 および被乗数、
 拡張精度 HFP 積

ME '7C' 短精度 HFP 乗数
 および被乗数、
 長精度 HFP 積

第 2 オペランド (乗数) と第 1 オペランド (被乗数) の正規化された積が、第 1 オペランド位置に入れられます。

2 つの HFP 数の乗算では、指数の加算と小数部の乗算が行われます。オペランドは、先行の 16 進 0 を除去するために、最初に正規化されます。正規化された 2 つのオペランドの指数部の和から 64 を引いた値が、中間結果の指数部として使用されます。

中間積の小数部は、正規化されたオペランドの小数部の正確な積です。中間積の小数部に、先行 16 進 0 が 1 つ付いている場合は、小数部は左に 1 桁ずつシフトされ、保護桁位置の内容が結果の小数部の右端の位置に入れられ、中間積の指数部が 1 だけ小さくなります。その後で、小数部は結果の小数部の適切な長さに合わせて切り捨てられます。

MDE および MDER では、乗数と被乗数の小数部はそれぞれ 6 桁の 16 進数字です。積の小数部は、14 桁の完全な長精度フォーマットになり、右端の 2 つの小数桁は常に 0 です。MEE および MEER の場合は、乗数および被乗数の小数部はそれぞれ 6 桁で、最終結果の積の小数部は 6 桁に切り捨てられます。結果は、すべての短精度フォーマットの結果の場合と同様に、ターゲット・レジスターの左端 32 ビットと置き換わり、ターゲット・レジスターの右端 32 ビットは変更されません。

MD および MDR の場合は、乗数および被乗数の小数部はそれぞれ 14 桁で、最終結果の積の小数部は 14 桁に切り捨てられます。MXD および MXDR の場合は、乗数および被乗数の小数部はそれぞれ 14 桁で、乗数は第 1 オペランドの高位部分を占めます。最終結果の積の小数部には 28 桁が含まれ、2 つのオペランドの小数部の正確な積となります。MXR の場合は、乗数および被乗数の小数部はそれぞれ 28 桁で、最終結果の積の小数部は 28 桁に切り捨てられます。

最終の積の指数部が 127 を超えることになり、小数部が 0 でない場合は、HFP 指数オーバーフロー例外が起きます。この場合は、結果の指数部が正しい値より 128 だけ小さくされて操作は完了し、HFP 指数オーバーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。拡張精度の結果において、低位指数部も 127 を超える場合は、その指数部も 128 だけ小さくされます。

中間積の指数部が最初は 128 であっても、正規化によって許容範囲内に収まれば、HFP 指数オーバーフローは認識されません。

最終結果の積の指数部が 0 より小さくなり、小数部が 0 でないときは、HFP 指数アンダーフロー例外が起きます。この場合、PSW 内の HFP 指数アンダーフロー・マスク・ビットが 1 であれば、結果の指数部が正しい値より 128 だけ大きくされて操作は完了し、HFP 指数アンダーフローに関するプログラム割り込みが起きます。結果は正規化されて、符号と小数部は正しい値を維持します。PSW 内の HFP 指数アンダーフロー・マスク・ビットが 0 の場合は、プログラム割り込みは起きず、代

わりに、結果が正の真の 0 にされて操作は完了します。拡張精度の結果において、低位指数部が 0 より小さくても、高位指数部が 0 以上であれば、HFP 指数アンダーフローは認識されません。

オペランドの正規化により、いずれかのオペランドの指数部が 0 より小さくなくても、最終的な積が正しい指数部により表現できるのであれば、HFP 指数アンダーフローは起きません。

一方または両方のオペランドの小数部が 0 の場合は、結果は正の真の 0 にされ、HFP 指数オーバーフローまたは HFP 指数アンダーフローは起きません。

積の符号は、オペランドの符号の排他論理和ですが、積が真の 0 にされるときは、符号は常に正です。

MXD、MXDR、および MXR の R₁ フィールド、および MXR の R₂ フィールドには、有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: MDE、MEE、MD、および MXD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- HFP 指数オーバーフロー例外
- HFP 指数アンダーフロー例外
- 指定例外 (MXD、MXDR、MXR)

プログラミング上の注意:

1. 付録Aに、MULTIPLY 命令 (MDR) の使用例を示してあります。
2. HFP の乗算では、2 つのオペランドの値を入れ替えても積の値は変わりません。

SQUARE ROOT

ニーモニック 1 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 1	命令コード	オペランド
SQER	'B245'	短精度 HFP
SQDR	'B244'	長精度 HFP
SQXR	'B336'	拡張精度 HFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	D ₂	////////	Op Code	
0	8	12	16	20	32	40	47

ニーモニック 2	命令コード	オペランド
SQE	'ED34'	短精度 HFP
SQD	'ED35'	長精度 HFP

第 2 オペランドの平方根が、正規化され、丸められて、第 1 オペランド位置に入れられます。

第 2 オペランドの小数部が 0 のときは、第 2 オペランドの符号と指数部は無視され、第 1 オペランド位置に正の真の 0 が入れられて操作は完了します。

第 2 オペランドが 0 より小さい場合は、HFP 平方根例外が認識されます。

第 2 オペランドが正規化されて 0 より大きい場合は、結果の指数部、小数部、および符号は以下のように生成されます。

- 結果の指数部は、オペランドの指数部に、64 (オペランドの指数部が偶数の場合) または 65 (奇数の場合) を加えた和の 2 分の 1 です。
- オペランドの指数部が奇数の場合は、オペランドの小数部は右に 1 桁ずつシフトされ、右端の桁の数字が保護桁の位置に入ります。
- 中間結果の小数部は、上記の右シフトの後で、オペランドの小数部の平方根を丸めずに計算されます。中間結果の小数部は、拡張精度フォーマットでは、平方根結果の中の最上位 29 桁、長精度フォーマットでは最上位 15 桁、そして短精度フォーマットでは最上位 7 桁の 16 進数字から成り、どのフォーマットの場合も、右端に保護桁が含まれます。
- 中間結果の保護桁である左端ビットに 1 が加算され、繰り上がりがあれば左へ繰り上がりが波及し、保護桁が除去されて、結果の小数部が生成されます。
- 結果の符号は正にされます。

第 2 オペランドが正規化されておらず、0 より大きい場合は、まず正規化されます。そして、正規化されたオペランドの場合と同様に操作が進められます。

SQXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

プログラム例外:

- アクセス例外 (取り出し: SQE および SQD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- HFP 平方根例外
- 指定例外 (SQXR のみ)

プログラミング上の注意:

1. 次の表は、短精度オペランドを使用した場合の SQUARE ROOT 命令 (SQER) の使用例を示しています。

オペランド (16 進数)	10 進数値	結果 (16 進数)	10 進数値
42 190000	25.0	41 500000	5.0
40 400000	0.250	40 800000	0.50
40 800000	0.50	40 B504F3	0.7071...
41 800000	8.0	41 2D413D	2.8284...

2. 結果の小数部は、中間結果の小数部をそれ以上左右にシフトすることも、指数の調整を行うこともせず、正しく正規化されます。丸めによって、左端桁からの繰り上がりが起きることはありません。
3. 操作の途中で、指数部が一時的に 127 を超えたり 0 より小さくなることはありますが、結果の指数部は常に表現可能な範囲内に収まるので、HFP 指数オーバーフローまたは HFP 指数アンダーフローが起きることはありません。

具体的に言えば、長精度フォーマットにおける 0 以外の最小のオペランドは、値が 1 のビットの左に 63 個の 0 を付加したものです。このオペランドは、 16^{-78} の値を持つ正規化されていない数で、その平方根は 16^{-39} です。この結果を正規化すると、指数部は 26 (10 進数) になります。同様に、表現可能な最大のオペランドの平方根の指数部は、96 (10 進数) です。したがって、この命令は、26 から 96 までの範囲から外れた指数部を持つ 0 以外の結果を生成することはありません。

SUBTRACT NORMALIZED

ニーモニック 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニック 1 命令コード オペランド
 SER '3B' 短精度 HFP
 SDR '2B' 長精度 HFP
 SXR '37' 拡張精度 HFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16 20	31

ニーモニック 2 命令コード オペランド
 SE '7B' 短精度 HFP
 SD '6B' 長精度 HFP

第 1 オペランドから第 2 オペランドが減算され、その差が正規化されて、第 1 オペランド位置に入れられます。

SUBTRACT NORMALIZED の実行結果は、第 2 オペランドの符号ビットを逆にして操作に使用するという点を除けば、ADD NORMALIZED の場合と同じです。

SXR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: SE および SD のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- HFP 指数オーバーフロー例外
- HFP 指数アンダーフロー例外
- HFP 有効数字例外
- 指定例外 (SXR のみ)

SUBTRACT UNNORMALIZED

ニーモニク 1 R₁,R₂ [RR]

Op Code	R ₁	R ₂
0	8	12 15

ニーモニク 1 命令コード オペランド
SUR '3F' 短精度 HFP
SWR '2F' 長精度 HFP

ニーモニク 2 R₁,D₂(X₂,B₂) [RX]

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20 31

ニーモニク 2 命令コード オペランド
SU '7F' 短精度 HFP
SW '6F' 長精度 HFP

第 1 オペランドから第 2 オペランドが減算され、その差が正規化されずに第 1 オペランド位置に入れられます。

SUBTRACT UNNORMALIZED の実行結果は、第 2 オペランドの符号ビットを逆にして操作に使用するという点を除けば、ADD UNNORMALIZED の場合と同じです。

結果の条件コード:

- 0 結果の小数部が 0。
- 1 結果が 0 より小さい。
- 2 結果が 0 より大きい。
- 3 --

プログラム例外:

- アクセス例外 (取り出し: SU および SW のオペランド 2 のみ)
- データ例外 (DXC 1)、AFP レジスター例外
- HFP 指数オーバーフロー例外
- HFP 有効数字例外

第19章 2進浮動小数点命令

2 進浮動小数点機能	19-1	IEEE アンダーフロー	19-13
浮動小数点制御 (FPC) レジスター	19-2	IEEE 不正確	19-13
IEEE マスクおよびフラグ	19-3	結果の表	19-14
FPC DXC バイト	19-3	データ例外コード (DXC) とその略語	19-15
FPC レジスターに対する操作	19-4	命令	19-16
BFP 演算	19-4	ADD	19-19
BFP データ・フォーマット	19-4	COMPARE	19-24
BFP 短精度フォーマット	19-4	COMPARE AND SIGNAL	19-25
BFP 長精度フォーマット	19-4	CONVERT FROM FIXED	19-27
BFP 拡張精度フォーマット	19-4	CONVERT TO FIXED	19-28
バイアス指数	19-5	DIVIDE	19-30
仮数	19-5	DIVIDE TO INTEGER	19-32
非ゼロ数の値	19-5	EXTRACT FPC	19-37
BFP データのクラス	19-6	LOAD AND TEST	19-37
ゼロ	19-6	LOAD COMPLEMENT	19-37
非正規化数	19-6	LOAD FP INTEGER	19-38
正規化数	19-6	LOAD FPC	19-39
無限大	19-7	LOAD LENGTHENED	19-40
信号付き NaN とクワイエット NaN	19-7	LOAD NEGATIVE	19-40
BFP フォーマット変換	19-7	LOAD POSITIVE	19-41
BFP の丸め	19-8	LOAD ROUNDED	19-41
丸めモード	19-8	MULTIPLY	19-42
正規化と非正規化	19-8	MULTIPLY AND ADD	19-44
BFP の比較	19-9	MULTIPLY AND SUBTRACT	19-44
BFP 命令の条件コード	19-9	SET FPC	19-46
剰余	19-9	SET ROUNDING MODE	19-47
IEEE 例外条件	19-11	SQUARE ROOT	19-47
IEEE 無効操作	19-12	STORE FPC	19-47
IEEE ゼロ除算	19-12	SUBTRACT	19-48
IEEE オーバーフロー	19-12	TEST DATA CLASS	19-48

2 進浮動小数点機能

2 進浮動小数点 (BFP) 機能は、2 進 (基数 2) の浮動小数点データを操作する命令を提供します。

BFP には、16 進浮動小数点 (HFP) に比べて、重要な利点が幾つかあります。

- 精度と指数の範囲が大きい (ただし、短精度フォーマットの数の場合は、HFP の方が範囲が大きい)。
- すべての算術演算における最近値への丸めが自動的に行われる。代わりに、有向丸めオプションも使用できます。

- 算術演算において、「無限大」および「非数値」(NaN) という特殊エンティティが受け入れられ、妥当な方法で処理されます。これらのエンティティは、指数オーバーフローおよび無効な操作 (0 を 0 で除算など) の場合に対処するための便利なデフォルトとして使用できます。ほとんどのプログラムは、これらのエンティティを利用することにより、このようなエラーを隠したり、特殊化された例外ハンドラーを使用したりせずに、処理を続行できます。
- 指数アンダーフローが起きた場合、HFP 命令では結果は 0 になり、一貫性が失われますが、BFP で

はデフォルトとして「非正規化数」が生成されるため、一貫性が高くなります。

- 指数範囲が大きいため、プログラムが正しく書かれていれば、指数オーバーフローおよび指数アンダーフローが起きる可能性は極めて低いので、プログラマーは多くの場合これらの条件を無視することができます。
- すべての算術例外条件について、マスク・ビットとフラグ・ビットの両方が提供されています。マスク・ビットは、割り込みを使用可能または使用不可にします。割り込みが使用不可にされているときは、実行中の例外条件がフラグ・ビットに記録されるので、警告メッセージの発行が可能になります。
- 異なるアーキテクチャーを使用しているワークステーションおよび他のシステムとの間で、プログラムのマイグレーションができます。他のシステムにおける浮動小数点操作も IEEE 標準に準拠していれば、一貫性のある結果が得られます。ただし、この標準は、特に例外が存在するときのインプリメンテーションの柔軟性を提供するためのものなので、ビット単位で互換性のある結果が保証されるわけではありません。

プログラミング上の注意: 記憶機構内での BFP データ・フォーマットのビット表現は、z/Architecture アーキテクチャーでのすべての数値オペランドに共通する方式に従い、左から右への方向に進むものとして定義され

ています。IEEE 浮動小数点標準のフォーマット・ダイアグラムは、これと同じ左から右へのビット・シーケンスを使用しているように見えますが、この標準は単にビットの意味を定義しているだけであり、記憶機構内でビットがどのように配列されるかを指定するものではありません。記憶機構内での配列は、インプリメンテーションによって決まります。実際に、別のシーケンスを使用しているインプリメンテーションも幾つかあります。これは、記憶機構内での浮動小数点データのビット表現に依存するプログラムに影響を与えることがあります。

浮動小数点制御 (FPC) レジスター

浮動小数点制御 (FPC) レジスターは、マスク・ビット、フラグ・ビット、データ例外コード、および丸めモード・ビットを含む 32 ビット・レジスターです。図 19-1 は、FPC レジスターの概要を示しています。詳細は、19-3 ページの図 19-2 および 19-3 ページの図 19-3 に示されています。(図 19-2 中の略語「IM」および「SF」は、それぞれ、「割り込みマスク (Interruption Mask)」および「状況フラグ (Status Flag)」を表しています。

FPC レジスターのビットは、多くの場合、例えば FPC 1.0 のように表記されますが、これはこのレジスターのバイト 1 のビット 0 を表しています。

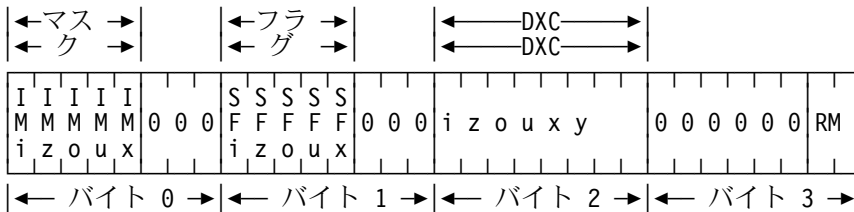


図 19-1. FPC レジスターの概要

バイト	ビット	名称	略語
0	0	IEEE 無効操作マスク	IMi
0	1	IEEE ゼロ除算マスク	IMz
0	2	IEEE オーバーフロー・マスク	IMo
0	3	IEEE アンダーフロー・マスク	IMu
0	4	IEEE 不正確マスク	IMx
0	5-7	(予約済み)	0
1	0	IEEE 無効操作フラグ	SFi
1	1	IEEE ゼロ除算フラグ	SFz
1	2	IEEE オーバーフロー・フラグ	SFo
1	3	IEEE アンダーフロー・フラグ	SFu
1	4	IEEE 不正確フラグ	SFx
1	5-7	(予約済み)	0
2	0-7	データ例外コード	DXC
3	0-5	(予約済み)	0
3	6-7	丸めモード	RM

図 19-2. FPC レジスターのビット割り当て

FPC バイト 3 ビット 6-7	丸めモード
00	最近値丸め
01	0 方向への丸め
10	+∞ 方向への丸め
11	-∞ 方向への丸め

図 19-3. 丸めモード

IEEE マスクおよびフラグ

FPC レジスターには、5 つの IEEE マスク・ビットと 5 つの IEEE フラグ・ビットが含まれており、これらは、それぞれ、BFP 命令を実行しているときに発生する可能性がある 5 つの演算例外の 1 つに対応しています。マスク・ビットが 1 のときは、例外条件が認識されると割り込みが起きます。ある例外条件についてのマスク・ビットが 0 の場合は、その条件が認識されると、それに対応するフラグ・ビットが 1 に設定されます。つまり、フラグ・ビットは、プログラムが最後にそのフラグ・ビットを設定してから後で、対応する例外条件が少なくとも 1 回は認識されているかどうかを示します。浮動小数点サポート (FPS) 命令および HFP 命令について演算例外が認識されたときは、マスク・ビットは無視され、フラグ・ビットは変更されません。

FPC レジスターの IEEE フラグ・ビットは、明示的なプログラム・アクション、クリア・リセット、またはパワーオン・リセットによってのみ、0 にリセットされます。

FPC DXC バイト

FPC レジスターのバイト 2 には、データ例外コード (DXC) が入ります。これは、データ例外の特定の理由を示す 8 ビットのコードです。AFP レジスター制御ビット (制御レジスター 0 のビット 45) が 1 のときに、プログラム割り込みが原因で DXC が実記憶位置 147 に入った場合は、FPC レジスターの DXC フィールドにも DXC が入れられます。AFP レジスター制御ビットが 0 であるか、または他のプログラム例外が報告された場合は、FPC レジスターの DXC フィールドは変更されません。DXC については、6-14 ページの『データ例外コード (DXC)』で説明します。

DXC はコードの一種であり、したがって、個別のビットとしてではなく 1 つの整数として取り扱うべきものです。ただし、ビット 6 および 7 が 0 のときは、ビット 0-5 はビット単位で意味を持ちます。この場合、ビット 0-4 (i, z, o, u, x) はトラップ・フラグで、FPC レジスターのバイト 0 および 1 の同じビット (IEEE マスクと IEEE フラグ) に対応します。ビット 5 (y) は、ビット 4 の不正確マスク (x) と一緒に使用されて、結果の大きさが増加したことを示します。ある例外に対応する IEEE マスクによりその例外についての割り込みが使用可能にされているときは、IEEE フラグではなく、その例外についてのトラップ・フラグが 1 に設定されます。

FPC レジスターに対する操作

問題プログラム状態のプログラムは、以下の非特権 BFP 命令を使用して、FPC レジスターに対する操作を行うことができます。

EXTRACT FPC
LOAD FPC
SET FPC
SET ROUNDING MODE
STORE FPC

これらの命令は、AFP レジスター制御ビット (制御レジスター 0 のビット 45) により制御されます。AFP レジスター制御ビットが 0 のときに上記のいずれかの命令を実行しようとする、BFP 命令データ例外 (DXC 2) が起きます。

BFP 演算

BFP データ・フォーマット

2 進浮動小数点数および NaN は、短精度、長精度、拡張精度の 3 つのフォーマットのどれでも表現できます。

BFP 短精度フォーマット

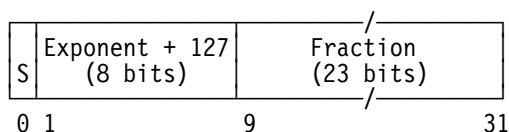


図 19-4. BFP 短精度フォーマット (4 バイト)

BFP 短精度フォーマットの数値または NaN が浮動小数点レジスターにロードされると、その値はレジスターの左半分を占め、右半分は変更されません。

BFP 長精度フォーマット

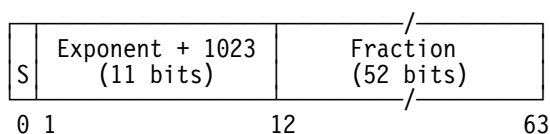


図 19-5. BFP 長精度フォーマット (8 バイト)

BFP 長精度フォーマットの数値または NaN が浮動小数点レジスターにロードされると、その値はレジスター全体を占めます。

BFP 拡張精度フォーマット

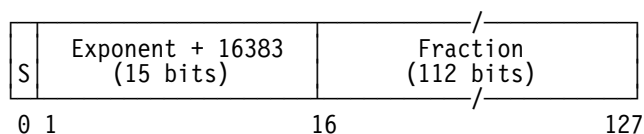


図 19-6. BFP 拡張精度フォーマット (16 バイト)

BFP 拡張精度フォーマットの数値または NaN は、一対のレジスターを占めます。左のレジスターには、左端 16 ビットに符号とバイアス指数が入り、その後小数部の左端 48 ビットが続きます。対を成す右のレジスターには、小数部の右端 64 ビットが入ります。

19-5 ページの図 19-7 に、3 つのフォーマットの特性を要約して示してあります。

プロパティ	フォーマット		
	短精度	長精度	拡張精度
フォーマット長 (ビット)	32	64	128
バイアス指数長 (ビット)	8	11	15
小数部長 (ビット)	23	52	112
精度 (p)	24	53	113
最大指数 (E _{max})	127	1023	16383
最小指数 (E _{min})	-126	-1022	-16382
指数バイアス	127	1023	16383
N _{max}	$(1-2^{-24}) \times 2^{128}$ $\approx 3.4 \times 10^{38}$	$(1-2^{-53}) \times 2^{1024}$ $\approx 1.8 \times 10^{308}$	$(1-2^{-113}) \times 2^{16384}$ $\approx 1.2 \times 10^{4932}$
N _{min}	1.0×2^{-126} $\approx 1.2 \times 10^{-38}$	1.0×2^{-1022} $\approx 2.2 \times 10^{-308}$	1.0×2^{-16382} $\approx 3.4 \times 10^{-4932}$
D _{min}	1.0×2^{-149} $\approx 1.4 \times 10^{-45}$	1.0×2^{-1074} $\approx 4.9 \times 10^{-324}$	1.0×2^{-16494} $\approx 6.5 \times 10^{-4966}$
説明: ≈ 値は概数です。 D _{min} 表現可能な最小の大きさの非正規化数。 N _{max} 表現可能な最大の大きさの数。 N _{min} 表現可能な最小の大きさの正規化数。			

図 19-7. BFP データ・フォーマットの要約

バイアス指数

図19-7 に、各フォーマットについて、すべての指数を符号なしの数で表せるようにするために使用されるバイアスを示してあります。バイアス指数は、HFP フォーマットの指数部に似ていますが、全桁 0 および全桁 1 のバイアス指数には特殊な意味があるという点が異なります。これについては、19-6ページの『BFP データのクラス』で説明します。

仮数

どのフォーマットでも、BFP 数の 2 進小数点は、左端の小数部ビットの左にあるものと見なされます。2 進小数点の左には暗黙の単位ビットがあり、これは正規化数の場合は 1 と見なされ、0 および非正規化数の場合は 0 と見なされます。小数部とその左に付加された暗黙の単位ビットが、数の仮数です。

正規化された BFP 数の値は、仮数に、基数 2 に非バイアス指数を累乗した値を掛けて得た値です。非正規化された BFP 数の値は、仮数に、基数 2 に最小指数を累乗した値を掛けて得た値です。

各フォーマットにおいて、仮数の右端のビット位置に 1 の値が入っているとき、それを 1 ulp (最終位置単位: unit in the last place) と呼ぶことがあります。

非ゼロ数の値

19-6ページの図19-8 は、各フォーマットでの非ゼロ数の値を示しています。

数のクラス	フォーマット	値
正規化	短精度	$\pm 2e-127x(1.f)$
	長精度	$\pm 2e-1023x(1.f)$
	拡張精度	$\pm 2e-16383x(1.f)$
非正規化	短精度	$\pm 2^{-126}x(0.f)$
	長精度	$\pm 2^{-1022}x(0.f)$
	拡張精度	$\pm 2^{-16382}x(0.f)$
説明: e バイアス指数 (10 進数)。 f 小数部 (2 進数)。		

図 19-8. 非ゼロ数の値

プログラミング上の注意: IEEE 標準では、拡張精度フォーマットの場合の最小要件は指定されていますが、詳細な規定はありません。BFP 拡張精度フォーマットはこの要件を満たすと共に、精度の点でこの要件をはるかに上回っています。

BFP データのクラス

BFP データには、数値エンティティおよび関連の非数値エンティティを含む 6 つのクラスがあります。各データ項目は、符号、指数、および仮数から成っています。指数はバイアスされています。すべてのバイアス指数は負でない符号なしの数で、最小バイアス指数は 0 です。仮数は、明示的な小数部と、2 進小数点の左の暗黙的な単位ビットから成っています。符号ビットは、正の場合は 0 で、負の場合は 1 です。

1 つのフォーマットで許される正規化範囲内のすべての有限非ゼロ数は、同じ固有の BFP 表現を持っています。同じ値について複数の表現が許される非正規化数はなく、また、非正規化算術演算はありません。1 つのフォーマットにおいて、最小正規化数に満たない大きさを持つ小さい数は、*非正規化数* として表されますが、この種の値も固有の形で表現されます。暗黙の単位ビットは、正規化数の場合は 1 で、非正規化数または 0 の場合は 0 です。

図 19-9 に、BFP データの 6 つのクラスを要約して示します。

データ・クラス	符号	バイアス指数	単位ビット*	小数部
ゼロ	±	0	0	0
非正規化数	±	0**	0	0 以外
正規化数	±	0 以外、すべて 1 でない	1	任意
無限大	±	すべて 1	-	0
クワイエット NaN	±	すべて 1	-	F0=1、Fr= 任意
信号付き NaN	±	すべて 1	-	F0=0、Fr≠0
説明: - 適用されません。 * 単位ビットは暗黙指定されます。 ** バイアス指数は、演算上は 1 の値を持つものとして扱われます。 F0 小数部の左端ビット。 Fr 小数部の残りのビット。 NaN 非数値。				

図 19-9. BFP データのクラス

BFP オペランドのクラスは、TEST DATA CLASS 命令を使用して判別できます。

ゼロ

ゼロ・クラスでは、バイアス指数および小数部は 0 です。暗黙の単位ビットは 0 です。+0 と -0 は区別されますが、比較演算ではこれらは等値と見なされます。

非正規化数

非正規化数は、最小の正規化数より小さく 0 より大きい大きさを持つ数です。非正規化数では、バイアス指数は 0 で、小数部は 0 以外です。バイアス指数は、演算上は 1 であるものとして扱われ、したがって指数は最小指数となります。暗黙の単位ビットは 0 です。

正規化数

正規化数では、バイアス指数は 0 より大きく、しかしすべて 1 より小さくなります。暗黙の単位ビットは 1 で、小数部は任意の値を持つことができます。

無限大

無限大は、すべて 1 のバイアス指数と 0 の小数部で表されます。無限大はほとんどの算術演算に使用でき、一貫性のある結果 (通常は無限大) をもたらします。比較演算では、 $+\infty$ はすべての有限数より大きく、 $-\infty$ はすべての有限数より小さいものと見なされます。

信号付き NaN とクワイエット NaN

NaN (非数値) エンティティは、すべて 1 のバイアス指数と 0 の小数部で表されます。NaN は、無効な演算が行われた後で、割り込みが起きなかった場合に、数値の結果の代わりに生成されます。また、プログラムが、未初期設定の記憶域の内容などの特殊オペランドをフラグで通知するために、NaN を使用することもあります。

NaN には、信号付き NaN とクワイエット NaN の 2 つのタイプがあります。信号付き NaN (SNaN) とそれに対応するクワイエット NaN (QNaN) は、小数部の左端ビットで区別されます。SNaN の場合は 0 で、QNaN の場合は 1 です。IEEE 無効操作条件の場合のデフォルトの結果として、特殊 QNaN が用意されています。特殊 QNaN は正符号を持ち、小数部の左端ビットが 1 で、小数部の残りのビットは 0 に設定されます。

通常、QNaN は計算中に単純に伝搬されるので、計算の終了時点まで可視状態のままになっています。SNaN オペランドは、IEEE 無効操作例外を発生させます。IEEE 無効操作マスク (FPC 0.0) が 0 の場合は、結果は対応する QNaN (小数部の左端ビットを 1 に設定される) となり、IEEE 無効操作フラグ (FPC 1.0) が 1 に設定されます。IEEE 無効操作マスク (FPC 0.0) が 1 の場合は、操作は抑止され、IEEE 無効操作に対応するデータ例外が起きます。

プログラミング上の注意:

1. プログラムにより、NaN の小数部について任意の非ゼロ値を生成し、それぞれに意味を割り当てることができます。CPU は、これらの値を無変更のまままで伝搬するため (ただし、IEEE 無効操作マスク・ビットが 0 の場合は、SNaN は対応する QNaN に変更されます)、精度の低いフォーマットへの変換が行われると、右側の有効数字ビットが切り捨てられることがあります。
2. IEEE 標準では、標準に含まれている算術演算、比較操作、および変換操作で SNaN が生じたときは、無効操作例外を通知することが必須条件とされていますが、フォーマットの変更なしで SNaN を

コピーしたときに例外を通知するかどうかについては、インプリメンテーションにより任意に選択できるものとされています。さらに、この標準の付録には、数値の符号のコピー、絶対値の使用、符号の反転、およびデータ・クラスのテストの推奨機能について SNaN が生じたときに、無効操作例外を通知すべきかどうかについても、インプリメンテーションにより任意に選択できると規定されています。

上記の機能は、一般に、LOAD、LOAD COMPLETE、LOAD NEGATIVE、LOAD POSITIVE、および TEST DATA CLASS 命令に対応しています。これらの命令は、無効操作例外を通知せずに、SNaN を他の一般的なデータと同様に扱います。SNaN を組み込むことを目的としているときに例外を起こさせるのは、処理を妨害することになるからです。TEST DATA CLASS は、SNaN の存在をテストするための命令なので、例外を通知しません。

3. LOAD AND TEST は、オペランドが SNaN であるときは、無効操作例外を通知します。上記の命令と共に、この命令の場合も、標準で許容されているどちらのオプションをとるかを、プログラムで選択することができます。
4. 精度を変更する LOAD タイプの命令は、オペランドが SNaN であるときは無効操作例外を通知します。これは、標準で規定されている必須条件です。

BFP フォーマット変換

LOAD LENGTHENED および LOAD ROUNDED

は、数値のフォーマットを、短精度、長精度、および拡張精度の間で変換します。BFP のフォーマット変換では、小数部と指数の両方の調整が行われます。正規化数を高い精度のフォーマットに変換するときは (短精度から長精度へ、長精度から拡張精度へ、または短精度から拡張精度へ)、小数部が、右側に必要数の 0 を付加することにより調整されます。低い精度への変換の場合は、右側の余剰ビットを除去する前に、小数部の丸めが必要です。その結果、IEEE 不正確条件が生じることがあります。

指数は、2 つのフォーマットのバイアス内の差を加算または減算することにより調整されます。低い精度のフォーマットへの変換の場合は、この調整の結果、バイアス指数が 1 より小さくなる場合は IEEE アンダーフローが起き、指数が新しいフォーマットの最大指数と同じかまたはそれより大きくなる場合は IEEE オーバーフローが起きます。

非正規化数が高い精度のフォーマットに変換されるときは、ソース・オペランドのバイアス指数は、値が 1 であるものとして扱われます。結果は正規化されます。

プログラミング上の注意:

1. NaN が低い精度のフォーマットに変換されるときは、小数部の右端ビットが必要数だけ除去され、それについての通知は出されません。これに対して、非ゼロ数の変換の場合は、小数部の 0 以外のビットが失われると、IEEE 不正確条件が発生します。したがって、特定の目的のために NaN の小数部ビットをエンコードするプログラムでは、必ず小数部の左部分に識別ビットを置くようにする必要があります。
2. SNaN は、割り込みを起こすかまたは QNaN に変更され、すべての QNaN の小数部の左端ビットは 1 なので、NaN を精度の高いフォーマットから低いフォーマットに変更することにより、NaN を無限大に変えることはできません。

BFP の丸め

算術演算および変換操作は、無限の精度と無制限の範囲を持つ正しい中間結果をまず最初に生成するものとして実行されます。この中間結果が、ターゲット・フォーマットでそのまま正確に表現できるものであれば、その値がそのまま使用されます。そうでない場合は、中間結果は、表現可能な 2 つの最近値のいずれかで置き換えられます。どちらの最近値が選択されるかは、丸めモードによって決まります。

丸めは、すべての算術演算および変換操作の一環として、自動的に行われます。ターゲットの精度 (短精度、長精度、または拡張精度) は、命令コードにより指定されます。

丸めモード

丸めモードは 4 種類あります。現行丸めモードは、FPC レジスター内の 2 つの丸めモード・ビットにより、以下のように指定されます。

- 00 最近値への丸め (デフォルト)。中間結果を、上位または下位の最も近い表現可能な値に丸めます。つまり、符号を無視して、保持すべき最後の結果ビットの直後のビットに 1 を加算し、繰り上がりを波及させ、保持すべき最後のビットを超えるビットを破棄します。差が正確に 0.5 ulp である (最後の位置の直後のビット位置が 1 で、その後のビット位置がすべて 0 である) 場合は、最も近い偶数値が選択されます。つまり、丸めによる加算

後、保持される最後の結果ビットは 0 に設定されます。

中間結果の絶対値が、表現可能な最大数に 0.5 ulp を加えた値に等しいかまたはそれより大きい場合、つまり、絶対値が $2^{E_{\max}}(2 - 2^{-p})$ に等しいかまたはそれより大きい場合は、結果は、中間結果と同じ符号を持つ無限大に丸められます。

- 01 0 方向への丸め。保持すべき最後の中間結果ビットより右側にあるすべてのビットを破棄します。
- 10 $+\infty$ 方向への丸め。中間結果が正であり、保持すべき最後の結果ビットの右側のどこかに 1 がある場合は、そのビットに 1 を加算します。そして、正負どちらの場合も、保持すべき最後のビットを超えるビットを破棄します。
- 11 $-\infty$ 方向への丸め。中間結果が負であり、保持すべき最後の結果ビットの右側のどこかに 1 がある場合は、そのビットから 1 を減算します (つまり、絶対値に 1 を加算します)。そして、正負どちらの場合も、保持すべき最後のビットを超えるビットを破棄します。

プログラミング上の注意:

1. 有限の結果を 0 方向に丸めても、無限大になることはありません。
2. 結果を $+\infty$ 方向に丸めた場合、 $+\infty$ になることはありませんが、 $-\infty$ になることはありません。
3. 結果を $-\infty$ 方向に丸めた場合、 $-\infty$ になることはありませんが、 $+\infty$ になることはありません。

正規化と非正規化

すべての算術演算または変換操作では、結果がゼロ、無限大、または NaN であるものとして定義されている場合を除き、精度および指数範囲にまったく制約がないものとして中間結果が生成されるものと見なされます。この中間結果を正規化してから丸めることにより、最終結果が生成されます。指数アンダーフローがある場合、つまり、正規化された中間結果のバイアス指数が 1 より小さい場合は、以下に述べるように、中間結果を非正規化して最終結果が生成されます。

非正規化では、仮数 (単位ビットも含む) が右にシフトされ、左側に 0 のビットが付加され、シフトされた各ビットごとに指数が 1 ずつ増加します。バイアス指数が +1 に達すると、現行丸めモードに従って仮数が丸められます。丸められた仮数のすべてのビットが 0 である場合は、結果は 0 にされます。丸めにより仮数の単位ビット位置への繰り上がりが生じる場合は、結果は正規化数

($\pm 2^{E_{min}}$) となるため、バイアス指数は +1 のままです。そうでない場合は、単位ビットは 0 のままで、バイアス指数は 0 に設定され、結果は非正規化数と見なされません。

非正規化オペランドに対する算術演算は、最初にオペランドが正規化されているものとして実行されます。

中間結果は、必要に応じて最初に正規化または非正規化されてから、丸められます。これは、1 つの操作で 2 回丸めが行われて、丸め誤差が増大するのを防ぐためです。(丸めによる繰り上がりの後で、結果を再正規化するために右シフトが必要になっても、シフトにより右に送り出されたビットは常に 0 なので、再度丸めが必要になることはありません。)

BFP の比較

比較は常に正確であり、IEEE 不正確条件が生じることはありません。

比較では、0 の符号は無視され、+0 と -0 は等値であると見なされます。

同種の符号を持つ無限大は等しいものと見なされます。つまり、 $+\infty$ と $+\infty$ は等値であり、 $-\infty$ と $-\infty$ も等値です。

比較では、NaN は、それ自体も含めて、他のどのオペランド (有限数、無限大、または他の NaN) に対しても *順序付け不能* と見なされます。

比較のための命令には、COMPARE と COMPARE AND SIGNAL の 2 つがあります。QNaN がいないときは、これらの命令の働きは同じです。これらの命令の働きが異なるのは、以下の条件が両方とも満たされている場合のみです。

- 命令のどちらのオペランドも SNaN ではない。
- 命令の少なくとも 1 つのオペランドが QNaN である。

この場合は、COMPARE は単に条件コード 3 を設定するだけですが、COMPARE AND SIGNAL は IEEE 無効操作条件を認識します。いずれかのオペランドが SNaN である場合は、どちらの命令も IEEE 無効操作条件を認識します。

IEEE 無効操作条件が認識されたときのアクションは、FPC レジスタの IEEE 無効操作マスク・ビットによ

って決まります。このマスク・ビットが 0 であれば、条件コード 3 を設定して命令の実行は完了し、FPC レジスタ内の IEEE 無効操作フラグが 1 に設定されます。このマスク・ビットが 1 の場合は、16 進数 80 の DXC を持つデータ例外 (IEEE 無効操作) に対応するプログラム割り込みの形で、この例外条件が報告されます。

プログラミング上の注意: コンパイラーは、比較を行うときに、IEEE 標準または関連の言語標準で、QNaN を例外条件として認識することが必須条件として規定されているかどうかに応じて、COMPARE または COMPARE AND SIGNAL のいずれかを選択します。

BFP 命令の条件コード

結果が有限または無限大の数値となる算術演算の場合は、条件コード 0、1、または 2 が設定されます。これらの条件コードは、それぞれ、結果が 0 (符号はどちらでもよい) か、0 より小さいか、または 0 より大きいことを示します。条件コードの設定は、丸められた結果を検査することによってのみ決定されます。比較操作では、条件コード 0、1、および 2 は、それぞれ、等しい、小さい、または大きいを示します。これらの設定は、HFP 命令の場合と同じです。

条件コード 3 が設定されることもあります。算術操作の後で条件コード 3 が設定された場合、それは結果がいずれかの符号の NaN であることを示します。比較の後の条件コード 3 は、比較の対象に NaN が含まれていること (順序付け不能条件) を示します。図 19-10 を参照してください。

CC	算術	比較
0	± 0	等しい
1	< 0	小さい
2	> 0	大きい
3	$\pm \text{NaN}$	順序付け不能

図 19-10. 条件コード

剰余

DIVIDE TO INTEGER は、2 つの浮動小数点結果を生成します。それは、正確な整数の商とそれに対応する剰余です。剰余は以下のように定義されます。

選択した浮動小数点フォーマットにおいて、次のように想定します。

a = 被除数
b = 除数
q = 正確な商 (a÷b)
r = 剰余

この場合、次の式が成立します。

$$r = a - b \cdot n$$

n は整数です。q が整数であれば、n と q は等値です。そうでなければ、n は、指定されている商の丸めモードに従って q を丸めることによって求められます。

指定されている商の丸めモードが、最近値への丸めまたは 0 方向への丸めであるときは、被除数が有限数で、除数が 0 以外であれば、剰余は正確です。剰余はオーバーフローすることはありません。

整数商の値がオペランド・フォーマットの範囲外にある場合は、循環された結果が提示されます。

整数商に含まれるビットの数が、オペランド・フォーマットの精度が提供する最大ビット数を超える (または超える可能性がある) 場合は、部分結果が生成され、最終結果を得るために命令を複数回実行することが必要になることがあります。これは、単純な命令ループを使用して実行できます。

部分結果が生成されるのは、厳密な商が整数ではなく、その厳密な商に最も近い 2 つの整数がどちらも商の精度では正確に表現できないときです。この状況が生じるの

は、厳密な商が 2^P より大きく (P はオペランド・フォーマットの精度)、かつ剰余が 0 ではないときです。剰余が 0 のときは、商は整数なので、商を表すために必要なビット数は、ターゲットの精度を超えることはありません。

プログラミング上の注意: 指定されている商の丸めモードが最近値への丸めである場合、DIVIDE TO INTEGER の結果の剰余は、IEEE 標準における *Remainder* (剰余) 関数に対応しています。この関数は、一部の言語に見られる *MOD* 関数および数学的 *modulo* 関数に似ていますが、同じではありません。つまり、次の示すように n の定義が異なります。

Remainder n は、q を最近値に丸めた値です。
modulo n は、q を $-\infty$ 方向に丸めた値です。
MOD n は、q を 0 方向に丸めた値です。

その他の重要な相違点は、*modulo* および *MOD* のインプリメンテーションでは単に *DIVIDE* 命令を使用してその範囲制限を受理するだけなので、結果の範囲について制約が課せられるということです。

MOD 定義は、*Remainder* の場合と同様に正確な結果を提供しますが、*modulo* 定義では丸めエラーが生じることがあります。

各種の方法の間の相違点は、a を b で割り、整数商 n を求めるという単純な計算例を使用して明確にすることができます。ここで、a は一連の整数であり、b は +4 または -4 です。19-11 ページの図 19-11 は、3 つの定義の結果を示しています。

	a																	
	-8	-7	-6	-5	-4	-3	-2	-1	-0	+0	+1	+2	+3	+4	+5	+6	+7	+8
	Remainder																	
b=+4: n	-2	-2	-2	-1	-1	-1	-0	-0	-0	+0	+0	+0	+1	+1	+1	+2	+2	+2
r	-0	+1	+2	-1	-0	1	-2	-1	-0	+0	+1	+2	-1	+0	+1	-2	-1	+0
b=-4: n	+2	+2	+2	+1	+1	+1	+0	+0	+0	-0	-0	-0	-1	-1	-1	-2	-2	-2
r	-0	+1	+2	-1	-0	+1	-2	-1	-0	+0	+1	+2	-1	+0	+1	-2	-1	+0
	MOD																	
b=+4: n	-2	-1	-1	-1	-1	0	0	0	0	0	0	0	+1	+1	+1	+1	+1	+2
r	0	-3	-2	-1	0	-3	-2	-1	0	+1	+2	+3	0	+1	+2	+3	0	0
b=-4: n	+2	+1	+1	+1	+1	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-2
r	0	-3	-2	-1	0	-3	-2	-1	0	+1	+2	+3	0	+1	+2	+3	0	0
	modulo																	
b=+4: n	-2	-2	-2	-2	-1	-1	-1	-1	0	0	0	0	+1	+1	+1	+1	+1	+2
r	0	+1	+2	+3	0	+1	+2	+3	0	+1	+2	+3	0	+1	+2	+3	0	0
b=-4: n	+2	+1	+1	+1	+1	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-2
r	0	-3	-2	-1	0	-3	-2	-1	0	-3	-2	-1	0	-3	-2	-1	0	0
説明:																		
a 被除数。																		
b 除数。																		
n 整数商。																		
r 結果 (Remainder、MOD、または modulo)。																		

図 19-11. Remainder と MOD および modulo との比較

Remainder の結果は、大きさが 0 から除数の 1/2 まで (両端の値を含む) の範囲内に収まります。結果が 0 のときの符号は、被除数の符号と同じであるものとして定義されています。0 の除数は無効です。

modulo と MOD の結果は、どちらも Remainder の結果から計算できます。ただし、その逆は成立しないことがあります。なぜなら、丸めエラーが起きることがあり、また、インプリメンテーションによっては範囲の制限が課せられることがあるからです。

以下に示すのは、modulo 定義について起きる可能性のある丸めエラーの極端な例です。この例では、結果は 2 個の有効 10 進数字に制限されます。

$\text{modulo}(0.01, -95) = -94.99$ (これは丸められて -95 になります)

$\text{Remainder}(0.01, -95) = 0.01$

modulo の結果が適切に丸められた場合、その値は除数に等しくなり、絶対値が除数より小さくならないため、完全な誤りとなります。Remainder の結果は正確であり、したがって modulo の理論的な結果を計算するために使用できます。

Remainder は、初等超越関数を計算するときの引き数選元に役立つため、算術演算の 1 つとして組み込まれています。したがって、 $\text{SIN}(X)$ は、 X がどのような次数の値であっても、最初に引き数を $\text{Remainder}(X, 360)$ に還元することにより、完全な精度で計算できます。

IEEE 例外条件

各 IEEE 例外条件の結果は、それぞれ、FPC レジスタ内の 1 つのマスク・ビットにより制御されます。

IEEE 例外条件が認識されると、次の 2 つのアクションのいずれかがとられます。

- FPC レジスタ内の対応するマスク・ビットが 0 である場合は、各条件について指定されているデフォルトのアクションがとられ、FPC レジスタ内の対応するフラグ・ビットが 1 に設定されます。そして、プログラムの実行が正常に続行されます。
- FPC レジスタ内の対応するマスク・ビットが 1 である場合は、データ例外に関するプログラム割り込みが起き、操作は、条件に応じて抑止されるかまたは完了し、その条件に割り当てられているデータ例外コード (DXC) が提示されます。

IEEE 無効操作

BFP 命令の実行中に以下のいずれかが起きると、IEEE 無効操作条件が認識されます。

1. BFP の算術、比較、または変換操作の中で、または LOAD AND TEST により、SNaN が検出された場合。
2. BFP 比較操作の中で、または COMPARE AND SIGNAL により、QNaN が検出された場合。
3. BFP の差が未定義の場合 (相反する符号を持つ無限大値の加算、または同種の符号を持つ無限大値の減算)。
4. BFP の積が未定義の場合 (0 と無限大の乗算)。
5. BFP の商が未定義の場合 (両オペランドに 0 または両オペランドに無限大を指定した DIVIDE 命令)。
6. BFP の剰余が未定義の場合 (被除数に無限大または除数に 0 を指定した DIVIDE TO INTEGER)。
7. BFP の平方根が未定義の場合 (0 以外の負のオペランド)。

FPC レジスター内の IEEE 無効操作マスク・ビットが 0 である場合は、FPC レジスター内の IEEE 無効操作フラグ・ビットが 1 に設定されます。操作の完了のしかたは、操作のタイプとオペランドによって異なります。

命令が比較を行い、プログラム割り込みが起きなかった場合は、比較の結果は順序付け不能になります。

命令が BFP の結果を生成するものであり、プログラム割り込みが起きず、どのオペランドも NaN ではない場合は、結果はデフォルトの QNaN になります。オペランドの 1 つが NaN である場合は、そのオペランドが無変更のまま結果となります。ただし、SNaN の場合は、最初に、小数部の左端ビットを 1 に設定することにより、対応する QNaN に変換されます。

FPC レジスター内の IEEE 無効操作マスク・ビットが 1 である場合は、操作は抑止され、16 進数 80 の DXC を持つデータ例外に関するプログラム割り込みの形でその条件が報告されます。

IEEE ゼロ除算

BFP の除算において、除数が 0 で被除数が 0 以外の有限数であると、IEEE 0 除算条件が認識されます。

FPC レジスター内の IEEE ゼロ除算マスク・ビットが 0 である場合は、FPC レジスター内の IEEE ゼロ除算フラグ・ビットが 1 に設定されます。操作は、被除数と除数の符号の排他論理和である符号を持つ無限大を結果として使用して、完了します。

FPC レジスター内の IEEE ゼロ除算マスク・ビットが 1 である場合は、操作は抑止され、16 進数 40 の DXC を持つデータ例外に関するプログラム割り込みの形でその条件が報告されます。

IEEE オーバーフロー

指数範囲が無制限であるときに、BFP 操作で丸められた結果の指数が、ターゲット・フォーマットの最大指数より大きくなる場合は、IEEE オーバーフロー条件が認識されます。

FPC レジスター内の IEEE オーバーフロー・マスク・ビットが 0 である場合は、FPC レジスター内の IEEE オーバーフロー・フラグ・ビットが 1 に設定されます。操作の結果は、次のように、中間結果の符号と現行丸めモードによって異なります。

1. 最近値への丸めの場合、結果は、中間結果と同じ符号の無限大です。
2. 0 方向への丸めの場合、結果は、中間結果と同じ符号を持つ、該当フォーマットの最大有限数です。
3. $+\infty$ 方向への丸めの場合、結果は、符号が正であれば $+\infty$ であり、符号が負であれば、最大の絶対値を持つ負の有限数です。
4. $-\infty$ 方向への丸めの場合、結果は、符号が正であれば、最大の正の有限数であり、符号が負であれば $-\infty$ です。

FPC レジスター内の IEEE オーバーフロー・マスク・ビットが 1 の場合は、操作は、循環した結果を生成して完了します。そして、16 進数 20、28、または 2C の DXC を持つデータ例外に関するプログラム割り込みの形で、その条件が報告されます。DXC は、循環された結果が正確であれば、20、不正確で切り捨て済みであれば 28、そして不正確で増加済みであれば 2C です。

IEEE アンダーフロー

BFP 操作の正確な結果の指数が、ターゲット・フォーマットの最小指数より小さくなるときは、IEEE アンダーフロー条件が認識されます。

FPC レジスター内の IEEE アンダーフロー・マスク・ビットが 0 である場合は、結果を正確に表現できるかどうかによって、また、正確に表現できないときは、FPC レジスターの IEEE 不正確マスク・ビットの設定によって、アクションが異なります。結果を正確に表現できる場合は、操作は、中間結果を非正規化して完了します。結果を正確に表現できず、FPC レジスター内の IEEE 不正確マスク・ビットが 0 である場合は、操作は、中間結果を非正規化して丸めることにより完了し、FPC レジスター内の IEEE アンダーフロー・フラグ・ビットおよび IEEE 不正確フラグ・ビットが 1 に設定されます。結果を正確に表現できず、FPC 不正確マスク・ビットが 1 である場合は、FPC レジスター内の IEEE アンダーフロー・フラグ・ビットが 1 に設定されます。そして、結果が不正確で切り捨て済みであるか、不正確で増加済みであるかに応じて、それぞれ 16 進数 08 または 0C の DXC を持つデータ例外に関するプログラム割り込みの形で、不正確条件が報告されます。

FPC レジスター内の IEEE アンダーフロー・マスク・ビットが 1 である場合は、結果が正確に表現できるかどうかに関係なく、操作は循環した結果を生成して完了します。そして、循環した結果が正確か、不正確で切り捨て済みか、または不正確で増加済みかに応じて、それぞれ 16 進数 10、18、または 1C の DXC を持つデータ例外に関するプログラム割り込みの形で、その条件が報告されます。

IEEE 不正確

BFP 操作の丸められた結果の値が、指数範囲と精度に制限がないものとして計算された中間結果の値と異なっていると、IEEE 不正確条件が認識されます。結果を丸めることにより IEEE オーバーフローが起き、IEEE オーバーフロー・マスク・ビットが 0 である場合も、この条件が認識されます。操作は、丸められた結果を使用するか、または、オーバーフローまたはアンダーフローが起きたときは IEEE オーバーフローまたは IEEE アンダーフローについて指定されている結果を使用して、完了します。

FPC レジスター内の IEEE 不正確マスク・ビットが 0 の場合は、FPC レジスター内の IEEE 不正確フラグ・ビットが 1 に設定されます。

FPC レジスター内の IEEE 不正確マスク・ビットが 1 である場合は、操作は完了します。そして、結果が正確

で切り捨て済みであるか、不正確で増加済みであるかに応じて、それぞれ 16 進数 08 または 0C の DXC を持つデータ例外に関するプログラム割り込みの形として、その条件が報告されます。

プログラミング上の注意:

- すべての IEEE トラップは、データ例外コードを持つデータ例外に関するプログラム割り込みを使用して報告されます。データ例外は、システム/360 アーキテクチャーに本来備わっている 15 種の例外の 1 つであり、ESA/390 アーキテクチャーをサポートするほとんどの制御プログラムでサポートされているので、アプリケーション・プログラムは、便利なインターフェースとしてデータ例外を利用できます。
- IEEE 標準には、トラップ・ハンドラーに関する推奨事項が含まれています。トラップ・ハンドラーは、システムがトラップしたときに、以下の事項を判別できることが望ましいとされています。
 - この操作でどの例外 (1 つまたは複数) が起きたか。
 - 実行されていた操作。
 - 宛先のフォーマット。
 - オーバーフロー例外、アンダーフロー例外、および不正確例外の場合に、正しく丸められた値 (宛先のフォーマットに適合しない情報も含む)。
 - 無効操作例外およびゼロ除算例外の場合に、オペランドの値。

上記の項目 a および d は、割り込みアクションの一環として提供されます。項目 b、c、および e は、旧 PSW 内の命令アドレスから開始して、命令 (操作とフォーマットを示す) を検出し、さらにオペランドを検出することにより、取得できます。

- アンダーフローについての記述は、IEEE 標準の中で最も理解するのが困難な部分の 1 つです。その理由は以下の点にあります。
 - 条件が「tininess (過小であること)」と「loss of accuracy (精度の消失)」の 2 つの「関連イベント」として記述される。
 - tininess の検出については、標準は「after rounding (丸め後)」と「after rounding (丸め前)」の 2 つのオプションを提供している。
 - loss of accuracy の検出については、標準は、「denormalization loss (非正規化消失)」と

「inexact result (不正確結果)」の 2 つのオプションを提供している。

- d. トラップのインプリメンテーションはオプションである。
- e. アンダーフローを通知する条件が、トラップがとられたかどうかによって異なる。

以下、上記の各項目について説明します。

- a. **tininess** は、厳密に $\pm 2^{E_{min}}$ の間にある 0 以外の数を表します。(非正規化数はすべてこの範囲に含まれます。) **loss of accuracy** は、結果を正確に表現できないことを意味します。
- b. 丸め後と丸め前で **tininess** の検出が異なるのは、「丸め」により、結果の絶対値が正確に $\pm 2^{E_{min}}$ に増加する場合のみです。ただし、ここで注意する必要があるのは、標準の中で「丸め」と称しているのは、最終結果を生成するための丸めではなく、「指数範囲に制限がない」ものとして、結果の精度を持つ中間値を計算するための丸めであるという点です。事実、「丸め後」の中間値が **tiny** (過小) であっても最終結果は **tiny** ではない可能性があります。

ESA/390 BFP アーキテクチャー (および RS/6000) では、丸め後の **tininess** の検出オプションが選択されています。

- c. **loss of accuracy** を **denormalization loss** として検出するか、**inexact result** として検出するかの違いは、2 つの中間値を考えてみるとよく理解できます。それは、(1) 精度と指数範囲の制限がない厳密な中間値、および (2) 厳密な中間値を、指数範囲に制限がないものとして、結果の精度に合わせて丸めることによって得られる、丸められた中間値です。**inexact result** は、最終結果が厳密な中間値と異なるときに生じるものとされています。**denormalization loss** は、最終結果が丸められた中間値と異なるときに生じるものとされています。この 2 つのオプション間に相違が生じるのは、最終結果の値が丸められた中間値に等しく、かつその 2 つの値が厳密な中間値に等しくない場合です。標準では「**denormalization loss (非正規化消失)**」という用語を使用していますが、この条件には、最終結果が正規化される場合も含まれていません。

ESA/390 BFP アーキテクチャー (および RS/6000) では、「**loss of accuracy**」を **inexact result** として検出するオプションが選択されています。

- d. 標準は、アンダーフローまたはその他の算術例外についてトラップをインプリメントすることを必須としていませんが、「ユーザー制御の下で各例外にトラップを関連付けることが望ましい」としています。標準は、「望ましい (should)」という用語を、「標準の意図に沿うために強く推奨する」と定義しているため、ESA/390 BFP アーキテクチャーは、プログラム割り込みを利用したトラップを提供していません。
- e. アンダーフロー・トラップが使用可能にされているときに、**tininess** が検出されると、**loss of accuracy** の有無に関係なく、アンダーフローが通知されます。アンダーフロー・トラップが使用可能にされていないときは、アンダーフロー・フラグ・ビットが設定されるのは、**tininess** および **loss of accuracy** の両方が検出された場合のみです。加算および減算では、**tininess** または **inexact result** のいずれかが生じることはありますが、両方も生じることはありません。したがって、アンダーフローが使用不可にされているときは、加算および減算でアンダーフロー・フラグ・ビットが設定されることはありません。

結果の表

多くの BFP 命令では、表を使用した簡潔な記述により結果が示されています。この表には、命令のソース・オペランドに関する BFP データ・クラスのすべての可能な組み合わせを示す列と行が含まれています。行と列の交点には、ソース・オペランドのデータ・クラスの該当の組み合わせの場合に生成される結果 (1 つまたは複数) を表す、1 つまたは複数の記号が示されています。各記号の説明は、それぞれの表の中に含まれています。多くの場合は、特定の結果についての説明は、他の表との相互参照形式で提供されています。多くの場合、交点に示される情報は、コンマで区切った幾つかの記号から成っています。結果の 1 つがプログラム割り込みでない限り、示されている結果がすべて生成されます。プログラム割り込みが起きた場合は、19-16 ページの図 19-13 に示すように、操作は抑止されるか、または完了します。

データ例外コード (DXC) とその略語

例外条件		FPC IEEE マスク・ビット	IEEE フラグ	
名称	略語	FPC ビット	略語	
IEEE 無効操作	Xi ¹	0.0	1.0	SFi
IEEE ゼロ除算	Xz ²	0.1	1.1	SFz
IEEE オーバーフロー	Xo	0.2	1.2	SFo
IEEE アンダーフロー	Xu	0.3	1.3	SFu
IEEE 不正確	Xx	0.4	1.4	SFx

説明:

- 図の中で、「Xi:」の記号の後に結果のリストが続いている場合、FPC 0.0 が 0 であれば、命令の実行が完了し、SFi (FPC 1.0) が 1 に設定され、提示されている結果が生成されたことを示します。FPC 0.0 が 1 であれば、命令の実行は抑止され、データ例外コード (DXC) が 16 進数 80 に設定され、データ例外に関するプログラム割り込みが起きたことを示します。
- 図の中で、「Xz:」の記号の後に結果のリストが続いている場合、FPC 0.1 が 0 であれば、命令の実行が完了し、SFz (FPC 1.1) が 1 に設定され、提示されている結果が生成されたことを示します。FPC 0.1 が 1 であれば、命令の実行は抑止され、データ例外コード (DXC) が 16 進数 40 に設定され、データ例外に関するプログラム割り込みが起きたことを示します。

図 19-12. IEEE 例外条件およびフラグの略語

図19-12 は、結果の図の中で使用される IEEE 例外条件とフラグの略語を示すとともに、図の中で使用される「Xi:」および「Xz:」という記号の意味を説明しています。FPC レジスタのバイト 2 にある 8 ビットのデータ例外コード (DXC) のうちのビット 0-4 (i, z, o, u, x) はトラップ・フラグであり、同じレジスタのバイト 0 および 1 の同じビット (IEEE マスクおよび IEEE フラグ) に対応しています。ある例外に対応する IEEE マスクによりその例外についての割り込みが使用可能にされているときは、IEEE フラグではなく、その例外についてのトラップ・フラグが 1 に設定されます。バイト 2 のビット 5 (y) は、不正確を示すビット 4 (x) と共に使用されて、結果が絶対値が増加したかどうかを示します。

19-16ページの図19-13 は、提示される可能性のある各種の DXC、それぞれに関連した命令終了の形態、および、結果の図の中で DXC を表すために使用される略語を示しています。(「PID」という略語は、「データ例外に関するプログラム割り込み (program interruption for a data exception)」を表しています。

略語	DXC (16 進数)	データ例外コードの名称	割り込み終了の形態
PIDx	08	IEEE 不正確および切り捨て	完了
PIDy	0C	IEEE 不正確および増加	完了
PIDu	10	IEEE アンダーフロー、正確	完了、指数循環
PIDux	18	IEEE アンダーフロー、不正確および切り捨て	完了、指数循環
PIDuy	1C	IEEE アンダーフロー、不正確および増加	完了、指数循環
PIDo	20	IEEE オーバーフロー、正確	完了、指数循環
PIDox	28	IEEE オーバーフロー、不正確および切り捨て	完了、指数循環
PIDoy	2C	IEEE オーバーフロー、不正確および増加	完了、指数循環
PIDz	40	IEEE ゼロ除算	抑止
PIDi	80	IEEE 無効操作	抑止

図 19-13. IEEE データ例外コード (DXC) とその略語

命令

19-17ページの図19-14 に、BFP 命令とそれぞれのニーモニックおよび命令コードを示してあります。この図の「特性」の欄には、命令のフォーマット、どのようなときに条件コードが設定されるか、アクセス・レジスタを指定する命令フィールド、そして、オペランド指定、データ、または結果の中にどのような例外条件があるときにプログラム割り込みが起きるかを示してあります。

すべての BFP 命令は、AFP レジスタ制御ビット (制御レジスタ 0 のビット 45) により制御されます。

BFP 命令が正常に実行されるためには、AFP レジスタ制御ビットが 1 でなければなりません。さもないと、BFP 命令データ例外 (DXC 2) が認識されます。

BFP 命令のニーモニックには、対応する HFP 命令と区別するために、B の文字が含まれています。RRE また

は RRF フォーマットの BFP 命令の場合は、ニーモニックの最後の文字として R が使用されています。また、BFP 命令については、次のように、オペランド・フォーマットの長さを表す特定の文字が使用されていません。

- F 32 ビット固定小数点
- G 64 ビット固定小数点
- D 長精度
- E 短精度
- X 拡張精度

注: 個々の命令の詳細記述では、命令ごとに、アセンブラ言語の場合のニーモニックとシンボリック・オペランド指定が示されています。例えば、COMPARE (短精度) を使用したレジスタ間操作では、CEBR がニーモニックで、R1,R2 がオペランド指定です。

名称	二 モ ニ ツ ク	特性						命 令 コ ド
ADD (拡張精度 BFP) ADD (長精度 BFP) ADD (長精度 BFP) ADD (短精度 BFP) ADD (短精度 BFP)	AXBR ADBR ADB AEBR AEB	RRE C RRE C RXE C RRE C RXE C		SP A A	Db Xi Db Xi Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx	B2 B2	B34A B31A ED1A B30A ED0A
COMPARE (拡張精度 BFP) COMPARE (長精度 BFP) COMPARE (長精度 BFP) COMPARE (短精度 BFP) COMPARE (短精度 BFP)	CXBR CDBR CDB CEBR CEB	RRE C RRE C RXE C RRE C RXE C		SP A A	Db Xi Db Xi Db Xi Db Xi Db Xi		B2 B2	B349 B319 ED19 B309 ED09
COMPARE AND SIGNAL (拡張精度 BFP) COMPARE AND SIGNAL (長精度 BFP) COMPARE AND SIGNAL (長精度 BFP) COMPARE AND SIGNAL (短精度 BFP) COMPARE AND SIGNAL (短精度 BFP)	KXBR KDBR KDB KEBR KEB	RRE C RRE C RXE C RRE C RXE C		SP A A	Db Xi Db Xi Db Xi Db Xi Db Xi		B2 B2	B348 B318 ED18 B308 ED08
CONVERT FROM FIXED (32 から拡張精度 BFP へ) CONVERT FROM FIXED (32 から長精度 BFP へ) CONVERT FROM FIXED (32 から短精度 BFP へ) CONVERT FROM FIXED (64 から拡張精度 BFP へ) CONVERT FROM FIXED (64 から長精度 BFP へ)	CXFBR CDFBR CEFBR CXGBR CDGBR	RRE RRE RRE RRE RRE		SP N N	Db Db Db Db Db	Xx Xx		B396 B395 B394 B3A6 B3A5
CONVERT FROM FIXED (64 から短精度 BFP へ) CONVERT TO FIXED (拡張精度 BFP から 32 へ) CONVERT TO FIXED (長精度 BFP から 32 へ) CONVERT TO FIXED (短精度 BFP から 32 へ) CONVERT TO FIXED (拡張精度 BFP から 64 へ)	CEGBR CFXBR CFDBR CFEBR CGXBR	RRE N RRF C RRF C RRF C RRF C	N	SP SP SP SP SP	Db Db Xi Db Xi Db Xi Db Xi	Xx Xx Xx Xx Xx		B3A4 B39A B399 B398 B3AA
CONVERT TO FIXED (長精度 BFP から 64 へ) CONVERT TO FIXED (短精度 BFP から 64 へ) DIVIDE (拡張精度 BFP) DIVIDE (長精度 BFP) DIVIDE (長精度 BFP)	CGDBR CGEBR DXBR DDBR DDB	RRF C N RRF C N RRE RRE RXE	N	SP SP SP A	Db Xi Db Xi Db Xi Db Xi Db Xi	Xx Xx Xz Xo Xu Xx Xz Xo Xu Xx Xz Xo Xu Xx	B2	B3A9 B3A8 B34D B31D ED1D
DIVIDE (短精度 BFP) DIVIDE (短精度 BFP) DIVIDE TO INTEGER (長精度 BFP) DIVIDE TO INTEGER (短精度 BFP) EXTRACT FPC	DEBR DEB DIDBR DIEBR EFPC	RRE RXE RRF C RRF C RRE		A SP SP	Db Xi Db Xi Db Xi Db Xi Db	Xz Xo Xu Xx Xz Xo Xu Xx Xu Xx Xu Xx	B2	B30D ED0D B35B B353 B38C
LOAD AND TEST (拡張精度 BFP) LOAD AND TEST (長精度 BFP) LOAD AND TEST (短精度 BFP) LOAD COMPLEMENT (拡張精度 BFP) LOAD COMPLEMENT (長精度 BFP)	LTXBR LTDBR LTEBR LCXBR LCDBR	RRE C RRE C RRE C RRE C RRE C		SP SP	Db Xi Db Xi Db Xi Db Db			B342 B312 B302 B343 B313

図 19-14 (1/3). BFP 命令の要約

名称	二 モ ニ ツ ク	特性					命 令 コ ー ド
LOAD COMPLEMENT (短精度 BFP) LOAD FP INTEGER (拡張精度 BFP) LOAD FP INTEGER (長精度 BFP) LOAD FP INTEGER (短精度 BFP) LOAD FPC	LCEBR FIXBR FIDBR FIEBR LFPC	RRE C RRF RRF RRF S		SP SP SP A SP	Db Db Xi Db Xi Db Xi Db		B303 B347 B35F B357 B29D
LOAD LENGTHENED (長精度から拡張精度 BFP へ) LOAD LENGTHENED (長精度から拡張精度 BFP へ) LOAD LENGTHENED (短精度から拡張精度 BFP へ) LOAD LENGTHENED (短精度から拡張精度 BFP へ) LOAD LENGTHENED (短精度から長精度 BFP へ)	LXDBR LXDB LXEBR LXEB LDEBR	RRE RXE RRE RXE RRE		SP A SP SP A SP	Db Xi Db Xi Db Xi Db Xi Db Xi		B305 ED05 B306 ED06 B304
LOAD LENGTHENED (短精度から長精度 BFP へ) LOAD NEGATIVE (拡張精度 BFP) LOAD NEGATIVE (長精度 BFP) LOAD NEGATIVE (短精度 BFP) LOAD POSITIVE (拡張精度 BFP)	LDEB LNXBR LNDBR LNEBR LPXBR	RXE RRE C RRE C RRE C RRE C		A SP	Db Xi Db Db Db Db		B2 ED04 B341 B311 B301 B340
LOAD POSITIVE (長精度 BFP) LOAD POSITIVE (短精度 BFP) LOAD ROUNDED (拡張精度から長精度 BFP へ) LOAD ROUNDED (拡張精度から短精度 BFP へ) LOAD ROUNDED (長精度から短精度 BFP へ)	LPDBR LPEBR LDXBR LEXBR LEDBR	RRE C RRE C RRE RRE RRE			Db Db Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx Xo Xu Xx	B310 B300 B345 B346 B344
MULTIPLY (拡張精度 BFP) MULTIPLY (長精度 BFP) MULTIPLY (長精度 BFP) MULTIPLY (長精度から拡張精度 BFP へ) MULTIPLY (長精度から拡張精度 BFP へ)	MXBR MDBR MDB MXDBR MXDB	RRE RRE RXE RRE RXE		SP A SP A SP	Db Xi Db Xi Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx Xo Xu Xx	B2 B34C B31C ED1C B307 ED07
MULTIPLY (短精度 BFP) MULTIPLY (短精度 BFP) MULTIPLY (短精度から長精度 BFP へ) MULTIPLY (短精度から長精度 BFP へ) MULTIPLY AND ADD (長精度 BFP)	MEEBR MEEB MDEBR MDEB MADBR	RRE RXE RRE RXE RRF		A A	Db Xi Db Xi Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx	B2 B317 ED17 B30C ED0C B31E
MULTIPLY AND ADD (長精度 BFP) MULTIPLY AND ADD (短精度 BFP) MULTIPLY AND ADD (短精度 BFP) MULTIPLY AND SUBTRACT (長精度 BFP) MULTIPLY AND SUBTRACT (長精度 BFP)	MADB MAEBR MAEB MSDBR MSDB	RXF RRF RFX RRF RFX		A A A	Db Xi Db Xi Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx	B2 ED1E B30E ED0E B31F ED1F
MULTIPLY AND SUBTRACT (短精度 BFP) MULTIPLY AND SUBTRACT (短精度 BFP) SET FPC SET ROUNDING MODE SQUARE ROOT (拡張精度 BFP)	MSEBR MSEB SFPC SRNM SQXBR	RRF RFX RRE S RRE		A SP	Db Xi Db Xi Db Db Db Xi	Xo Xu Xx Xo Xu Xx	B2 B30F ED0F B384 B299 B316

図 19-14 (2/3). BFP 命令の要約

名称	ニーモニツク	特性						命令コード	
SQUARE ROOT (長精度 BFP) SQUARE ROOT (長精度 BFP) SQUARE ROOT (短精度 BFP) SQUARE ROOT (短精度 BFP) STORE FPC	SQDBR SQDB SQEBR SQEB STFPC	RRE RXE RRE RXE S		A	Db Xi Db Xi Db Xi Db Xi Db	Xx Xx Xx Xx		B2 B2 B2 B2	B315 ED15 B314 ED14 B29C
SUBTRACT (拡張精度 BFP) SUBTRACT (長精度 BFP) SUBTRACT (長精度 BFP) SUBTRACT (短精度 BFP) SUBTRACT (短精度 BFP)	SXBR SDBR SDB SEBR SEB	RRE C RRE C RXE C RRE C RXE C		SP A	Db Xi Db Xi Db Xi Db Xi Db Xi	Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx Xo Xu Xx		B2 B2	B34B B31B ED1B B30B ED0B
TEST DATA CLASS (拡張精度 BFP) TEST DATA CLASS (長精度 BFP) TEST DATA CLASS (短精度 BFP)	TCXB TCDB TCEB	RXE C RXE C RXE C		SP	Db Db Db			ED12 ED11 ED10	
説明: A 論理アドレスについてのアクセス例外。 B2 アクセス・レジスター・モードでは、B2 フィールドはアクセス・レジスターを指定します。 BF BFP 機能。 C 条件コードが設定されます。 Db BFP 命令データ例外。 N z/Architecture の新規の命令で、ESA/390 にはないものです。 RRE RRE 命令フォーマット。 RRF RRF 命令フォーマット。 RXE RXE 命令フォーマット。 RXF RXF 命令フォーマット。 SP 指定例外。 ST PER 記憶域変更イベント。 Xi IEEE 無効操作条件。 Xo IEEE オーバーフロー条件。 Xu IEEE アンダーフロー条件。 Xx IEEE 不正確条件。 Xz IEEE ゼロ除算条件。									

図 19-14 (3/3). BFP 命令の要約

ADD

ニーモニツク 1 R₁, R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニツク 1 命令コード オペランド
AEBR 'B30A' 短精度 BFP
ADBR 'B31A' 長精度 BFP
AXBR 'B34A' 拡張精度 BFP

ニーモニツク 2 R₁, D₂(X₂, B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	D ₂	////////	Op Code
0	8	12	16	20	32	40 47

ニーモニツク 2 命令コード オペランド
AEB 'ED0A' 短精度 BFP
ADB 'ED1A' 長精度 BFP

第 2 オペランドが第 1 オペランドに加算され、和が第 1 オペランド位置に入られます。

両方のオペランドが有限の数値である場合は、両方が代数的に加算され、中間和が生成されます。中間和は、0 以外である場合は、正規化されて、現行丸めモードに従ってオペランド・フォーマットに丸められます。その結果の和が、結果の位置に入られます。

和の符号は代数の規則に従って決定されます。これは、次のように、結果が 0 の場合にも適用されます。

- 0 でない中間和を丸めた結果が 0 になった場合は、結果の 0 の符号は中間和の符号と同じです。
- 逆の符号を持つ 2 つのオペランドの和が正確に 0 になった場合は、丸めモードが $-\infty$ 方向への丸めであるときは負ですが、その他の丸めモードではすべて正です。
- $x + x$ の和の符号は、 x が 0 であっても x の符号です。

一方のオペランドが無限大であり、もう一方が有限数である場合は、結果はその無限大です。両方のオペランドが同じ符号の無限大である場合は、結果は同じ無限大です。2つのオペランドが逆の符号を持つ無限大である場合は、IEEE 無効操作条件が認識されます。

この命令の結果についての詳細説明は、19-21ページの図19-16を参照してください。(図19-16では、図19-15が参照されています。)

AXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 結果は NaN。

IEEE 例外条件:

- 無効操作例外
- オーバーフロー例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- アクセス例外 (取り出し: AEB および ADB のオペランド 2 のみ)

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (AXBR のみ)

プログラミング上の注意: BFP の加算では、結果が数値のときは、2つのオペランドの値を入れ替えても和の値は変わりません。ただし、オペランドが両方とも QNaN であるときは、結果は第 1 オペランドです。また、オペランドが両方とも SNaN であり、FPC レジスタ内の IEEE 無効操作マスク・ビットが 0 のときは、結果は第 1 オペランドから導き出された QNaN です。

結果の値 (r)	条件コード
r=0	cc0
r<0	cc1
r>0	cc2
説明: ccn 条件コードは n に設定されます。	

図 19-15. 結果の和に関する条件コード

第 1 オペランド (a) の値	ADD (a+b) の結果 第 2 オペランド (b) が以下の値であるとき									
	$-\infty$	-Nn	-Dn	-0	+0	+Dn	+Nn	$+\infty$	QNaN	SNaN
$-\infty$	T($-\infty$), cc1	T($-\infty$), cc1	T($-\infty$), cc1	T($-\infty$), cc1	T($-\infty$), cc1	T($-\infty$), cc1	T($-\infty$), cc1	Xi: T(dNaN), cc3	T(b), cc3	Xi: T(b*), cc3
-Nn	T($-\infty$), cc1	R(a+b), cc1	R(a+b), cc1	T(a), cc1	T(a), cc1	R(a+b), cc1	R(a+b), ccrs	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
-Dn	T($-\infty$), cc1	R(a+b), cc1	R(a+b), cc1	R(a), cc1	R(a), cc1	R(a+b), ccrs	R(a+b), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
-0	T($-\infty$), cc1	T(b), cc1	R(b), cc1	T(-0), cc0	Rezd, cc0	R(b), cc2	T(b), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
+0	T($-\infty$), cc1	T(b), cc1	R(b), cc1	Rezd, cc0	T(+0), cc0	R(b), cc2	T(b), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
+Dn	T($-\infty$), cc1	R(a+b), cc1	R(a+b), ccrs	R(a), cc2	R(a), cc2	R(a+b), cc2	R(a+b), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
+Nn	T($-\infty$), cc1	R(a+b), ccrs	R(a+b), cc2	T(a), cc2	T(a), cc2	R(a+b), cc2	R(a+b), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
$+\infty$	Xi: T(dNaN), cc3	T($+\infty$), cc2	T($+\infty$), cc2	T($+\infty$), cc2	T($+\infty$), cc2	T($+\infty$), cc2	T($+\infty$), cc2	T($+\infty$), cc2	T(b), cc3	Xi: T(b*), cc3
QNaN	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	T(a), cc3	Xi: T(b*), cc3
SNaN	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3	Xi: T(a*), cc3

説明:

- * SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。
- ccn 条件コードは n に設定されます。
- ccrs 条件コードは結果の和に従って設定されます。19-20 ページの図19-15 を参照してください。
- dNaN デフォルトのクワイエット NaN。
- Dn 非正規化数。
- Nn 正規化された 0 以外の数。
- R(v) 値 v に対して丸めおよび範囲アクションが行われます。19-22ページの図19-17 を参照してください。
- Rezd 正確なゼロ差結果。19-22ページの図19-17 を参照してください。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。
- Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-16. 結果: ADD

v の範囲	ケース	正常な結果 (r) 丸めモードが下記の場合			
		最近値 へ	0 方向 へ	+∞ 方向 へ	-∞ 方向 へ
$v < -Nmax, p < -Nmax$	オーバーフロー	$-\infty^1$	-Nmax	-Nmax	$-\infty^1$
$v < -Nmax, p = -Nmax$	正常	-Nmax	-Nmax	-Nmax	-
$-Nmax \leq v \leq -Nmin$	正常	p	p	p	p
$-Nmin < v \leq -Dmin$	過小 (Tiny)	d*	d	d	d*
$-Dmin < v < -Dmin/2$	過小	-Dmin	-0	-0	-Dmin
$-Dmin/2 \leq v < 0$	過小	-0	-0	-0	-Dmin
$v = 0$	正確なゼロ差 ²	+0	+0	+0	-0
$0 < v \leq +Dmin/2$	過小	+0	+0	+Dmin	+0
$+Dmin/2 < v < +Dmin$	過小	+Dmin	+0	+Dmin	+0
$+Dmin \leq v < +Nmin$	過小	d*	d	d*	d
$+Nmin \leq v \leq +Nmax$	正常	p	p	p	p
$+Nmax < v, p = +Nmax$	正常	+Nmax	+Nmax	-	+Nmax
$+Nmax < v, +Nmax < p$	オーバーフロー	$+\infty^1$	+Nmax	$+\infty^1$	+Nmax

説明:

- この状況が生じることはありません。
- * 極端な場合、丸められた値は Nmin になることがあります。その場合は、例外条件はアンダーフロー、不正確、および増加済みです。
- ¹ 正常な結果 r は、増加しているものと見なされます。
- ² 正確なゼロ差のケースが適用されるのは、ADD、SUBTRACT、MULTIPLY AND ADD、および MULTIPLY AND SUBTRACT のみです。その他の操作の場合は、R(v) 関数を使用せずに、ソース・オペランドの検査によりゼロ結果が検出されます。
- d 正確な結果 v がターゲットのフォーマット (精度および制限付きの指数範囲を含む) に丸められたときに導き出される値。* の注の説明を除き、これは非正規化数です。
- p 正確な結果 v が、指数範囲が無制限であるものとしてターゲットの精度に丸められたときに導き出される値。
- v 精度も指数範囲も無制限であるものとした場合の、丸めの前の正確な結果。LOAD ROUNDED の場合は、v はソース値 a です。
- Dmin ターゲット・フォーマットで表現できる、絶対値が最小の非正規化数。
- Nmax ターゲット・フォーマットで表現できる、絶対値が最小の有限数。
- Nmin ターゲット・フォーマットで表現できる、絶対値が最小の正規化数。

図 19-17 (1/2). R(v) の場合のアクション: 丸めおよび範囲機能

ケース	r は 不正確 ($r \neq v$) か	オーバーフ ロー・ マスク (FPC 0.2)	アンダー フロー・ マスク (FPC 0.3)	不正確 マスク (FPC 0.4)	r は 増加済み ($ r > v $) か	p は 不正確 ($p \neq v$) か	p は 増加済み ($ p > v $) か	結果
オーバーフ ロー	はい ¹	0	-	0	-	-	-	T(r), SFo←-1, SFx←-1
オーバーフ ロー	はい ¹	0	-	1	いいえ	-	-	T(r), SFo←-1, PIDx(08)
オーバーフ ロー	はい ¹	0	-	1	はい	-	-	T(r), SFo←-1, PIDy(0C)
オーバーフ ロー	はい ¹	1	-	-	-	いいえ	いいえ ¹	Tw(p+β), PIDo(20)
オーバーフ ロー	はい ¹	1	-	-	-	はい	いいえ	Tw(p+β), PIDox(28)
オーバーフ ロー	はい ¹	1	-	-	-	はい	はい	Tw(p+β), PIDoy(2C)
正常	いいえ	-	-	-	-	-	-	T(r)
正常	はい	-	-	0	-	-	-	T(r), SFx←-1
正常	はい	-	-	1	いいえ	-	-	T(r), PIDx(08)
正常	はい	-	-	1	はい	-	-	T(r), PIDy(0C)
過小 (Tiny)	いいえ	-	0	-	-	-	-	T(r)
過小	いいえ	-	1	-	-	いいえ ¹	いいえ ¹	Tw(p·β), PIDu(10)
過小	はい	-	0	0	-	-	-	T(r), SFu←-1, SFx←-1
過小	はい	-	0	1	いいえ	-	-	T(r), SFu←-1, PIDx(08)
過小	はい	-	0	1	はい	-	-	T(r), SFu←-1, PIDy(0C)
過小	はい	-	1	-	-	いいえ	いいえ ¹	Tw(p·β), PIDu(10)
過小	はい	-	1	-	-	はい	いいえ	Tw(p·β), PIDux(18)
過小	はい	-	1	-	-	はい	はい	Tw(p·β), PIDuy(1C)

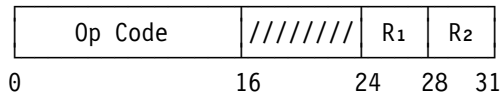
説明:

- 結果は、この条件またはマスク・ビットに応じて変化しません。
- ¹ この条件は、この列の左側にあるいずれかの条件の状態に応じて真になります。
- β 循環調整。これは、操作のタイプとオペランド・フォーマットによって異なります。LOAD ROUNDED を除くすべての操作の場合、循環調整はターゲット・フォーマットによって異なります。β = 2^α で、α は短精度の場合は 192、長精度の場合は 1536、拡張精度の場合は 24576 です。LOAD ROUNDED の場合は、循環調整はソース・フォーマットによって異なります。β = 2^κ で、κ は長精度の場合は 512、拡張精度の場合は 8192 です。
- p 正確な結果 v が、指数範囲が無制限であるものとしてターゲットの精度に丸められたときに導き出される値。
- r この図のパート 1 で定義されている正常な結果。
- v 精度も指数範囲も無制限であるものとした場合の、丸めの前の正確な結果。
- PIDc(h) 16 進数 h の DXC を持つデータ例外 (条件 c) に関するプログラム割り込み。19-16ページの図19-13 を参照してください。
- SFo IEEE オーバーフロー・フラグ (FPC 1.2)。
- SFu IEEE アンダーフロー・フラグ (FPC 1.3)。
- SFx IEEE 不正確フラグ (FPC 1.4)。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。
- Tw(x) ターゲット・オペランドの位置に循環結果 x が入れられます。LOAD ROUNDED を除くすべての操作の場合、循環結果のフォーマットと長さは、ターゲット位置での正常結果と同じになります。LOAD ROUNDED の場合は、循環結果のフォーマットと長さはソースと同じですが、ターゲットの精度に合わせて丸められます。

図 19-17 (2/2). R(v) の場合のアクション: 丸めおよび範囲機能

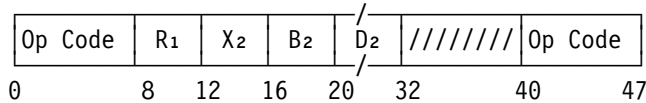
COMPARE

ニーモニク 1 R₁, R₂ [RRE]



ニーモニク 1	命令コード	オペランド
CEBR	'B309'	短精度 BFP
CDBR	'B319'	長精度 BFP
CXBR	'B349'	拡張精度 BFP

ニーモニク 2 R₁, D₂(X₂, B₂) [RXE]



ニーモニク 2	命令コード	オペランド
CEB	'ED09'	短精度 BFP
CDB	'ED19'	長精度 BFP

第 1 オペランドが第 2 オペランドと比較され、その結果を示す条件コードが設定されます。

オペランドが両方も有限数である場合は、比較は、BFP 減算と同じ手順に従って代数的に行われますが、条件コードが設定された後は、差は破棄され、オペランドはどちらも変更されません。差が正確に 0 であれば、符号に関係なく、両オペランドは等しいものと見なされます。これには 0 オペランドも含まれます (したがって、+0 と -0 は等値です)。差が 0 以外の場合は、差が正であれば第 1 オペランドの方が大きく、差が負であれば第 1 オペランドの方が小さいこととなります。

+∞ はすべての有限数より大きく、すべての有限数は -∞ より大きいものと見なされます。同種の符号を持つ 2 つの無限大オペランドは、等しいと見なされます。

数値比較は正確であり、有限数オペランドについての条件コードは、範囲と精度に制限がないものとして決定されます。オーバーフロー条件またはアンダーフロー条件が生じることはありません。

一方または両方のオペランドが QNaN であり、どちらのオペランドも SNaN ではない場合は、比較の結果は順序付け不能であり、条件コード 3 が設定されます。

オペランドの一方または両方が SNaN である場合は、IEEE 無効操作条件が認識されます。IEEE 無効操作マ

スク・ビットが 1 の場合は、16 進数 80 の DXC を持つデータ例外 (IEEE 無効操作) に対応するプログラム割り込みが起きます。IEEE 無効操作マスク・ビットが 0 である場合は、IEEE 無効操作フラグ・ビットが 1 に設定され、命令実行は、条件コード 3 を設定して完了します。

この命令の結果についての詳細説明は、19-25 ページの図 19-18 を参照してください。

CXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されません。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 オペランドは順序付け不能。

IEEE 例外条件:

- 無効操作例外

プログラム例外:

- アクセス例外 (取り出し: CEB および CDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (CXBR のみ)

プログラミング上の注意:

1. コンパイラーで COMPARE を使用することで、IEEE 標準により、QNaN が原因で結果が順序付け不能となっても例外条件を認識しないことが必須条件とされている比較をインプリメントすることができます。
2. IEEE 標準では、フォーマットの異なる BFP オペランドを比較できることが必須条件とされています。これを達成するためには、COMPARE の前に LOAD LENGTHENED を使用して、短い方のオペランドを長い方のオペランドと同じフォーマットに変換します。

第1オペランド (a) の値	COMPARE (a:b) の結果 第2オペランド (b) が以下の値であるとき							
	$-\infty$	-Fn	-0	+0	+Fn	$+\infty$	QNaN	SNaN
$-\infty$	cc0	cc1	cc1	cc1	cc1	cc1	cc3	Xi: cc3
-Fn	cc2	C(a:b)	cc1	cc1	cc1	cc1	cc3	Xi: cc3
-0	cc2	cc2	cc0	cc0	cc1	cc1	cc3	Xi: cc3
+0	cc2	cc2	cc0	cc0	cc1	cc1	cc3	Xi: cc3
+Fn	cc2	cc2	cc2	cc2	C(a:b)	cc1	cc3	Xi: cc3
$+\infty$	cc2	cc2	cc2	cc2	cc2	cc0	cc3	Xi: cc3
QNaN	cc3	cc3	cc3	cc3	cc3	cc3	cc3	Xi: cc3
SNaN	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3

説明:

ccn 条件コードは n に設定されます。

C(a:b) 基本的な比較結果。図19-19 を参照してください。

Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。

Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-18. 結果: COMPARE

値 (a) と 値 (b) の関係	C(a:b) の条件コード
a=b	cc0
a<b	cc1
a>b	cc2

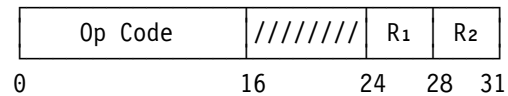
説明:

ccn 条件コードは n に設定されます。

図 19-19. 基本的な比較結果

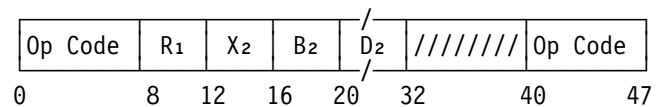
COMPARE AND SIGNAL

ニーモニック 1 R_1, R_2 [RRE]



ニーモニック 1 命令コード オペランド
 KEBR 'B308' 短精度 BFP
 KDBR 'B318' 長精度 BFP
 KXBR 'B348' 拡張精度 BFP

ニーモニック 2 $R_1, D_2 (X_2, B_2)$ [RXE]



ニーモニック 2 命令コード オペランド
 KEB 'ED08' 短精度 BFP
 KDB 'ED18' 長精度 BFP

第1オペランドが第2オペランドと比較され、その結果を示す条件コードが設定されます。QNaN オペランドがあると IEEE 無効操作条件が認識されるという点を除

けば、この操作は COMPARE と同じです。QNaN オペランドは SNaN であるものとして扱われます。

この命令の結果についての詳細説明は、図19-20 を参照してください。

KXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 両オペランドは等しい。
- 1 第 1 オペランドの方が小さい。
- 2 第 1 オペランドの方が大きい。
- 3 オペランドは順序付け不能。

IEEE 例外条件:

- 無効操作例外

プログラム例外:

- アクセス例外 (取り出し: KEB および KDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (KXBR のみ)

プログラミング上の注意:

1. コンパイラーで COMPARE AND SIGNAL を使用することで、IEEE 標準により、QNaN が原因で結果が順序付け不能となったときに例外条件を認識することが必須条件とされている比較をインプリメントすることができます。
2. IEEE 標準では、フォーマットの異なる BFP オペランドを比較できることが必須条件とされています。これを達成するためには、COMPARE AND SIGNAL の前に LOAD LENGTHENED を使用して、短い方のオペランドを長い方のオペランドと同じフォーマットに変換することができます。

第 1 オペランド (a) の値	COMPARE AND SIGNAL (a:b) の結果 第 2 オペランド (b) が以下の値であるとき						
	$-\infty$	-Fn	-0	+0	+Fn	$+\infty$	NaN
$-\infty$	cc0	cc1	cc1	cc1	cc1	cc1	Xi: cc3
-Fn	cc2	C(a:b)	cc1	cc1	cc1	cc1	Xi: cc3
-0	cc2	cc2	cc0	cc0	cc1	cc1	Xi: cc3
+0	cc2	cc2	cc0	cc0	cc1	cc1	Xi: cc3
+Fn	cc2	cc2	cc2	cc2	C(a:b)	cc1	Xi: cc3
$+\infty$	cc2	cc2	cc2	cc2	cc2	cc0	Xi: cc3
NaN	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3	Xi: cc3

説明:

ccn 条件コードは n に設定されます。

C(a:b) 基本的な比較結果。19-25ページの図19-19 を参照してください。

Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。

Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-20. 結果: COMPARE AND SIGNAL

CONVERT FROM FIXED

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31
ニーモニック	命令コード	オペランド	
CEFBR	'B394'	32 ビット 2 進 整数オペランド、 短精度 BFP 結果	
CDFBR	'B395'	32 ビット 2 進 整数オペランド、 長精度 BFP 結果	
CXFBR	'B396'	32 ビット 2 進 整数オペランド、 拡張精度 BFP 結果	
CEGBR	'B3A4'	64 ビット 2 進 整数オペランド、 短精度 BFP 結果	
CDGBR	'B3A5'	64 ビット 2 進 整数オペランド、 長精度 BFP 結果	
CXGBR	'B3A6'	64 ビット 2 進 整数オペランド、 拡張精度 BFP 結果	

第 2 オペランドの固定小数点数が BFP フォーマットに変換され、その結果が第 1 オペランドの位置に入れます。

第 2 オペランドは、R₂ に指定されている汎用レジスタ内にある符号付き 2 進整数です。このレジスタのビット位置 32-63 が、32 ビットのオペランドを表しています。

結果は、第 1 オペランド位置に入れられる前に、現行丸めモードに従って丸められます。

この命令の結果についての詳細説明は、19-28ページの図19-21 を参照してください。

CXFBR および CXGBR の場合は、R₁ フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 不正確例外 (CEFBR、CDGBR、CEGBR)

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (CXFBR および CXGBR)

命令	単一オペランド (a) の命令の結果 オペランド (a) が以下の値のとき							
	$-\infty$	$-Fn$	-0	$+0$	$+Fn$	$+\infty$	QNaN	SNaN
CONVERT FROM FIXED	-	Rf(a)	-	T(+0)	Rf(a)	-	-	-
LOAD AND TEST	T($-\infty$)	T(a)	T(-0)	T(+0)	T(a)	T(+ ∞)	T(a)	Xi: T(a*)
LOAD LENGTHENED	T($-\infty$)	T(a) ¹	T(-0)	T(+0)	T(a) ¹	T(+ ∞)	T(a) ¹	Xi: T(a*) ¹
LOAD ROUNDED	T($-\infty$)	R(a)	T(-0)	T(+0)	R(a)	T(+ ∞)	T(a) ²	Xi: T(a*) ²
SQUARE ROOT	Xi: T(dNaN)	Xi: T(dNaN)	T(-0)	T(+0)	R(\sqrt{a})	T(+ ∞)	T(a)	Xi: T(a*)

説明:

- この状況が生じることはありません。
- * SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。
- ¹ オペランドは、右側に 0 を付加して長い方のフォーマットに拡張されてから、ターゲット・オペランド位置に入れられます。
- ² NaN は、右端ビットを切り捨ててターゲット・フォーマットに短縮されます。
- dNaN デフォルトのクワイエット NaN。
- Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。
- R(v) 値 v に対して丸めおよび範囲アクションが行われます。19-22ページの図19-17 を参照してください。
- Rf(a) 値 a が正確な浮動小数点数 v に変換されてから、アクション R(v) が実行されます。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。
- Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-21. 結果: 単一オペランド命令

CONVERT TO FIXED

ニーモニック R_1, M_3, R_2 [RRF]

Op Code	M ₃	////	R ₁	R ₂
0	16	20	24	28 31
ニーモニック	命令コード		オペランド	
CFEBR	'B398'		短精度 BFP オペラ ンド、32 ビット 2 進 整数結果	
CFDBR	'B399'		長精度 BFP オペラ ンド、32 ビット 2 進 整数結果	
CFXBR	'B39A'		拡張精度 BFP オペラ ンド、32 ビット 2 進 整数結果	
CGEBR	'B3A8'		短精度 BFP オペラ ンド、64 ビット 2 進 整数結果	
CGDBR	'B3A9'		長精度 BFP オペラ ンド、64 ビット 2 進 整数結果	
CGXBR	'B3AA'		拡張精度 BFP オペラ ンド、64 ビット 2 進 整数結果	

第 2 オペランドの BFP が整数値に丸められてから、固定小数点フォーマットに変換されます。その結果が第 1 オペランドの位置に入れられます。

結果は符号付き 2 進整数で、R₁ に指定されている汎用レジスタに入れられます。このレジスタのビット 32-63 が 32 ビットの結果で置き換えられ、ビット 0-31 は変更されません。

第 2 オペランドが有限数である場合は、その数は、M₃ フィールド内の修飾子の指定に従って、整数値に丸められます。

M₃ 丸め方式

- 0 現行丸めモード
- 1 バイアス付き最近値丸め
- 4 最近値丸め
- 5 0 方向への丸め
- 6 $+\infty$ 方向への丸め
- 7 $-\infty$ 方向への丸め

0、1、または 4-7 以外の修飾子は無効です。

修飾子フィールドが 0 のときは、丸めは、FPC レジスタに指定されている現行丸めモードにより制御されます。このフィールドが 0 でないときは、現行丸めモードに関係なく、修飾子に指定されている丸めが行われます。修飾子 4-7 の丸めは、それぞれ、丸めモード 0-3 (2 進数 00-11) と同じです。バイアス付き最近値丸め (修飾子 1) は、最近値丸め (修飾子 4) とおおむね同じですが、第 2 オペランドが 2 つの整数間の正確に半分の位置にある場合に、バイアス付き丸めでは、結果が、絶対値が次に大きい整数になるという点が異なります。

結果の符号は第 2 オペランドの符号と同じですが、結果が 0 のときは正符号になります。

この命令の結果についての詳細説明は、19-30ページの図19-22 を参照してください。

M3 フィールドには有効な修飾子を指定しなければなりません。さもないと、指定例外が認識されます。CFXBR および CGXBR の場合は、R₂ フィールドには

有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 ソースが 0。
- 1 ソースが 0 より小さい。
- 2 ソースが 0 より大きい。
- 3 特殊ケース。

IEEE 例外条件:

- 無効操作例外
- 不正確例外

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外

オペランド (a)	n は 不正確 (n≠a) か	無効操作 マスク (FPC 0.0)	不正確 マスク (FPC 0.4)	n は 増加済み (n > a) か	結果
$-\infty \leq a < MN, p < MN$	-	0	0	-	T(MN), SFi←-1, SFx←-1, cc3
$-\infty \leq a < MN, p < MN$	-	0	1	-	T(MN), SFi←-1, cc3, PIDx(08)
$-\infty \leq a < MN, p < MN$	-	1	-	-	PIDi(80)
$-\infty < a < MN, p = MN$	-	-	0	-	T(MN), SFx←-1, cc1
$-\infty < a < MN, p = MN$	-	-	1	-	T(MN), cc1, PIDx(08)
$MN \leq a < 0$	いいえ	-	-	-	T(n), cc1
$MN \leq a < 0$	はい	-	0	-	T(n), SFx←-1, cc1
$MN \leq a < 0$	はい	-	1	いいえ	T(n), cc1, PIDx(08)
$MN \leq a < 0$	はい	-	1	はい	T(n), cc1, PIDy(0C)
-0	いいえ ¹	-	-	-	T(0), cc0
+0	いいえ ¹	-	-	-	T(0), cc0
$0 < a \leq MP$	いいえ	-	-	-	T(n), cc2
$0 < a \leq MP$	はい	-	0	-	T(n), SFx←-1, cc2
$0 < a \leq MP$	はい	-	1	いいえ	T(n), cc2, PIDx(08)
$0 < a \leq MP$	はい	-	1	はい	T(n), cc2, PIDy(0C)
$MP < a < +\infty, p = MP$	-	-	0	-	T(MP), SFx←-1, cc2
$MP < a < +\infty, p = MP$	-	-	1	-	T(MP), cc2, PIDx(08)
$MP < a \leq +\infty, p > MP$	-	0	0	-	T(MP), SFi←-1, SFx←-1, cc3
$MP < a \leq +\infty, p > MP$	-	0	1	-	T(MP), SFi←-1, cc3, PIDx(08)
$MP < a \leq +\infty, p > MP$	-	1	-	-	PIDi(80)
NaN	-	0	0	-	T(MN), SFi←-1, SFx←-1, cc3
NaN	-	0	1	-	T(MN), SFi←-1, cc3, PIDx(08)
NaN	-	1	-	-	PIDi(80)

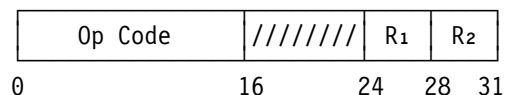
説明:

- 結果は、この条件またはマスク・ビットに応じて変化しません。
- ¹ この条件は、この列の左側にあるいずれかの条件の状態に応じて真になります。
- ccn 条件コードは n に設定されます。
- n 値 p は固定小数点結果に変換されます。
- p 指定されている丸めモードを使用してソース値 a を整数に丸めたときに得られる値。
- MN ターゲットの固定小数点フォーマットで表現できる最大の負の値。
- MP ターゲットの固定小数点フォーマットで表現できる最大の正の値。
- PIDc(h) 16 進数 h の DXC を持つデータ例外 (条件 c) に関するプログラム割り込み。19-16ページの図19-13 を参照してください。
- SFi IEEE 無効操作フラグ (FPC 1.0)。
- SFx IEEE 不正確フラグ (FPC 1.4)。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。

図 19-22. 結果: CONVERT TO FIXED

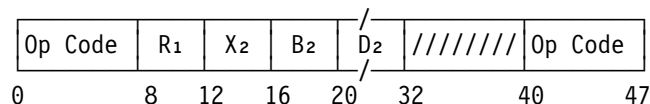
DIVIDE

ニーモニク 1 R₁,R₂ [RRE]



ニーモニク 1 命令コード オペランド
 DEBR 'B30D' 短精度 BFP
 DDBR 'B31D' 長精度 BFP
 DXBR 'B34D' 拡張精度 BFP

ニーモニク 2 R₁,D₂(X₂,B₂) [RXE]



ニーモニク 2 命令コード オペランド
 DEB 'ED0D' 短精度 BFP
 DDB 'ED1D' 長精度 BFP

第 1 オペランド (被除数) が第 2 オペランド (除数) で除算され、商が第 1 オペランド位置に入れられます。剰余は保存されません。

除数が 0 でなく、被除数と除数が両方とも有限数である場合は、第 1 オペランドを第 2 オペランドで除算して、中間商が求められます。この中間商は、0 以外であれば、正規化されて、現行丸めモードに従ってターゲット・フォーマットに丸められます。

商の符号は、両オペランドの符号の排他論理和です。これは、商が 0 のときの符号にも適用されます。

除数が 0 で、被除数が 0 以外の有限数である場合は、IEEE ゼロ除算条件が認識されます。被除数と除数が両方とも 0 であるか、または両方とも無限大である場合は、符号に関係なく、IEEE 無効操作条件が認識されません。

この命令の結果についての詳細説明は、19-32ページの図19-23 を参照してください。

DXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外
- ゼロ除算例外
- オーバーフロー例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- アクセス例外 (取り出し: DEB および DDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (DXBR のみ)

被除数 (a)	DIVIDE (a÷b) の結果 除数 (b) が以下の値のとき							
	-∞	-Fn	-0	+0	+Fn	+∞	QNaN	SNaN
-∞	Xi: T(dNaN)	T(+∞)	T(+∞)	T(-∞)	T(-∞)	Xi: T(dNaN)	T(b)	Xi: T(b*)
-Fn	T(+0)	R(a+b)	Xz: T(+∞)	Xz: T(-∞)	R(a+b)	T(-0)	T(b)	Xi: T(b*)
-0	T(+0)	T(+0)	Xi: T(dNaN)	Xi: T(dNaN)	T(-0)	T(-0)	T(b)	Xi: T(b*)
+0	T(-0)	T(-0)	Xi: T(dNaN)	Xi: T(dNaN)	T(+0)	T(+0)	T(b)	Xi: T(b*)
+Fn	T(-0)	R(a+b)	Xz: T(-∞)	Xz: T(+∞)	R(a+b)	T(+0)	T(b)	Xi: T(b*)
+∞	Xi: T(dNaN)	T(-∞)	T(-∞)	T(+∞)	T(+∞)	Xi: T(dNaN)	T(b)	Xi: T(b*)
QNaN	T(a)	T(a)	T(a)	T(a)	T(a)	T(a)	T(a)	Xi: T(b*)
SNaN	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)

説明:

- * SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。
- Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。
- R(v) 値 v に対して丸めおよび範囲アクションが行われます。19-22ページの図19-17 を参照してください。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。
- Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。
- Xz: IEEE ゼロ除算例外。示されている結果は、FPC 0.1 が 0 の場合のみ生成されます。

図 19-23. 結果: DIVIDE

DIVIDE TO INTEGER

ニーモニック R_1, R_3, R_2, M_4 [RRF]

Op Code	R3	M4	R1	R2
0	16	20	24	28 31

ニーモニック 命令コード オペランド
 DIEBR 'B353' 短精度 BFP
 DIDBR 'B35B' 長精度 BFP

第 1 オペランド (被除数) が第 2 オペランド (除数) で除算されます。BFP 形式の整数商が生成され、第 3 オペランド位置に入れられます。第 1 オペランド位置の被除数は剰余で置き換えられます。第 1、第 2、および第 3 のオペランドは、それぞれ異なるレジスター内になければなりません。条件コードは、部分結果または完全結

果のどちらが生成されたか、および商が有限数であるかどうかを示します。

剰余結果は次のようになります。

$$r = a - b \cdot n$$

ここで、a は被除数、b は除数、そして n は厳密な商を丸めて求められる整数です。

$$q = a \div b.$$

第 1 オペランドの結果は r で、符号は上記の式に基づいて決定されます。第 3 オペランドの結果は n で、符号は、被除数と除数の符号の排他論理和です。

厳密な商が整数ではなく、その厳密な商に最も近い 2 つの整数がどちらも商の精度では正確に表現できないときは、部分商および部分剰余が生成されます。部分商 n および以下に示す部分剰余が、結果として使用されます。

$$r = a - b \cdot n$$

部分剰余の符号は、被除数の符号と同じです。部分商の符号は、被除数と除数の符号の排他論理和です。

剰余が 0 の場合は、厳密な商は整数であり、したがって商の精度で正確に表現できます。

M4 フィールド (修飾子フィールド) には、最終商の丸め方法が指定されています。この丸めを「指定された商の丸めモード」と呼び、これに対して、FPC レジスタの丸めモード・ビットに指定されている丸めを「現行丸めモード」と呼びます。最終商は、指定された商の丸めモードに従って丸められます。指定された商の丸めモードが適用されるのは、最終商のみです。部分商は 0 方向に丸められます。

部分商は 0 方向に丸められるので、部分剰余は常に正確です。指定された商の丸めモードが、0 方向への丸め、最近値丸め、およびバイアス付き最近値丸めである場合は、最終剰余は正確です。指定された商の丸めモードが、 $+\infty$ 方向への丸めおよび $-\infty$ 方向への丸めである場合は、最終剰余は正確ではないことがあります。

最終商は、**M4** フィールド内の修飾子により指定されている丸め方法により、整数に丸められます。

M4 丸め方式

- 0 現行丸めモード
- 1 バイアス付き最近値丸め
- 4 最近値丸め
- 5 0 方向への丸め
- 6 $+\infty$ 方向への丸め
- 7 $-\infty$ 方向への丸め

0、1、または 4-7 以外の修飾子は無効です。

修飾子フィールドが 0 のときは、最終商の丸めは、FPC レジスタに指定されている現行丸めモードにより制御されます。このフィールドが 0 でないときは、現行丸めモードに関係なく、修飾子に指定されている丸めが行われます。修飾子 4-7 の丸めは、それぞれ、丸めモード 0-3 (2 進数 00-11) と同じです。バイアス付き最近値丸め (修飾子 1) は、最近値丸め (修飾子 4) とおおむね同じですが、最終商が 2 つの整数間の正確に半分のある位置にある場合に、バイアス付き丸めでは、結果が、絶対値が次に大きい整数になるという点が異なります。

アンダーフローが認識されるのは最終剰余のみで、部分剰余ではアンダーフローは認識されません。

指定された商の丸めモードが、 $+\infty$ 方向への丸めおよび $-\infty$ 方向への丸めである場合は、最終剰余は正確ではないことがあります。最終剰余が正確でないときは、FPC レジスタに指定されている現行丸めモードに従って丸められます。

商が 0 の場合の符号は、除数と被除数の符号の排他論理和です。

剰余が 0 の場合の符号は、被除数の符号と同じです。

この命令の結果についての詳細説明は、19-35 ページの図 19-24 を参照してください。

商の指数が、オペランド・フォーマットで表現できる最大の指数より大きい場合でも、正しい剰余または部分剰余が生成され、第 3 オペランドの結果は正しい値になりますが、指数は、短精度オペランドの場合は 192、そして長精度オペランドの場合は 1536 だけ減少します。そして、条件コードにこの範囲外条件が示されます。

M4 フィールドは有効な修飾子を指し、**R1**、**R2**、および **R3** フィールドはそれぞれ異なるレジスタを指していなければなりません。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 完全な剰余、正常な商
- 1 完全な剰余、商のオーバーフローまたは NaN
- 2 不完全な剰余、正常な商
- 3 不完全な剰余、商のオーバーフローまたは NaN

IEEE 例外条件:

- 無効操作例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. M4 フィールドを 4 に設定して、反復ループ内で **DIVIDE TO INTEGER** を実行すると、IEEE 標準で定義されている **Remainder** 操作が行われます。
2. 丸め方法として、最近値丸め、0 方向への丸め、または $-\infty$ 方向への丸めを指定してこの命令を実行すると、それぞれ、**Remainder**、**MOD**、および **modulo** 関数と同じ結果を生成することができます。
3. **DIVIDE TO INTEGER** 反復ループの中で使用すると、商はすべて **BFP** フォーマットで生成されますが、複精度の固定小数点数の一部と見なされることがあります。
4. 結果の剰余が非正規化される場合、IEEE 標準では、トラップがインプリメントされていて、アンダーフロー・マスクが 1 のときは、アンダーフロー・トラップが生じなければならないと規定されています。この条件を満たすために、**DIVIDE TO INTEGER** は、部分剰余ではなく最終剰余についてアンダーフローを認識します。部分剰余についてアンダーフローが起きたときは、最終剰余についても必ずアンダーフローが起きるので、最終剰余についてのみアンダーフローを認識するようにすることにより、標準により単一の **Remainder** と見なされる操作について、2 つのアンダーフロー・トラップが報告されるのを回避することができます。

被除数 (a)	DIVIDE TO INTEGER (a+b) の結果 除数 (b) が以下の値のとき							
	$-\infty$	$-Fn$	-0	$+0$	$+Fn$	$+\infty$	QNaN	SNaN
$-\infty$	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	T(b), cc1	Xi: T(b*), cc1
$-Fn$	T(a,+0), cc0	D(a,b)	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	D(a,b)	T(a,-0), cc0	T(b), cc1	Xi: T(b*), cc1
-0	T(-0,+0), cc0	T(-0,+0), cc0	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	T(-0,-0), cc0	T(-0,-0), cc0	T(b), cc1	Xi: T(b*), cc1
$+0$	T(+0,-0), cc0	T(+0,-0), cc0	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	T(+0,+0), cc0	T(+0,+0), cc0	T(b), cc1	Xi: T(b*), cc1
$+Fn$	T(a,-0), cc0	D(a,b)	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	D(a,b)	T(a,+0), cc0	T(b), cc1	Xi: T(b*), cc1
$+\infty$	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	Xi: T(dNaN), cc1	T(b), cc1	Xi: T(b*), cc1
QNaN	T(a), cc1	T(a), cc1	T(a), cc1	T(a), cc1	T(a), cc1	T(a), cc1	T(a), cc1	Xi: T(b*), cc1
SNaN	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1	Xi: T(a*), cc1

説明:

* SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。

ccn 条件コードは n に設定されます。

D(a,b) 基本的な整数への除算の結果。この図のパート 2 を参照。

Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。

T(r,q) 結果の r (剰余) および q (商) が、それぞれターゲット・オペランド 1 および 3 に入れられます。

T(x) 値 x がターゲット・オペランド 1 および 3 の両方に入れられます。

Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-24 (1/2). 結果: DIVIDE TO INTEGER

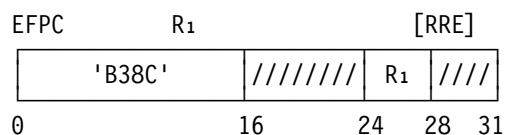
$ q < 2^P$	$r = 0$	ケース	r は過小 (Tiny) か	r は不正確か	アンダーフロー・マスク (FPC 0.3)	不正確マスク (FPC 0.4)	商オーバーフロー	r は増加済みか	D(a,b) の結果
はい	はい	最終	いいえ ¹	いいえ ¹	-	-	なし ¹	-	T(r,n), cc0
はい	いいえ	最終	いいえ	いいえ	-	-	なし ¹	-	T(r,n), cc0
はい	いいえ	最終	はい	いいえ ¹	0	-	なし ¹	-	T(r,n), cc0
はい	いいえ	最終	はい	いいえ ¹	1	-	なし ¹	いいえ ¹	T(r·β, n), cc0, PIDu(10)
はい	いいえ	最終	いいえ	はい	-	0	なし ¹	-	T(r,n), SFx←-1, cc0
はい	いいえ	最終	いいえ	はい	-	1	なし ¹	いいえ	T(r,n), cc0, PIDx(08)
はい	いいえ	最終	いいえ	はい	-	1	なし ¹	はい	T(r,n), cc0, PIDy(0C)
いいえ	はい	最終	いいえ ¹	いいえ ¹	-	-	なし	-	T(r,n), cc0
いいえ	はい	最終	いいえ ¹	いいえ ¹	-	-	あり	-	T(r,n+β), cc1
いいえ	いいえ	部分	- ²	いいえ ¹	-	-	なし	-	T(r,n), cc2
いいえ	なし	部分	- ²	いいえ ¹	-	-	あり	-	T(r, n+β), cc3

説明:

- 結果は、この条件またはマスク・ビットに応じて変化しません。
- 1 この条件は、この列の左側にあるいずれかの条件の状態に応じて真になります。つまり、 $|q| < 2^P$ のときは、商のオーバーフローが起きることはありません。剰余については、0、過小 (Tiny)、および不正確は互いに排他的です。また、r は、正確であるときは増加しません。
- 2 部分剰余については、アンダーフローは認識されません。
- β 循環調整 (これはフォーマットにより異なります)。β = 2^α で、α は、短精度の場合は 192、長精度の場合は 1536 です。
- lql q の絶対値。q は、精度と指数範囲に制限がないと想定した場合の、丸めの前の a+b の正確な結果です。
- cc0 条件コードは、0 (完全な剰余、正常な商) に設定されます。
- cc1 条件コードは、1 (完全な剰余、商のオーバーフロー) に設定されます。
- cc2 条件コードは、2 (不完全な剰余、正常な商) に設定されます。
- cc3 条件コードは、3 (不完全な剰余、商のオーバーフロー) に設定されます。
- n 整数商。n は、q を、部分結果の場合は 0 方向に丸め、最終結果の場合は指定された商の丸めモードに従って丸めて得た値です。整数商の符号は、部分と最終、循環オーバーフローと 0 の場合も含めて、被除数 (a) と除数 (b) の符号の排他論理和です。
- r 剰余。r = a-b·n。部分剰余は常に正確で、丸めは不要です。部分剰余の符号は、常に被除数 (a) の符号と同じです。最終剰余は、現行丸めモードに従って丸められます (必要な場合)。0 の剰余の符号は、被除数 (a) の符号と同じです。0 以外の最終剰余の符号は、代数の規則に従って決定されます。
- P オペランドの精度。ターゲットのフォーマットにより異なります。P は、短精度の場合は 24 で、長精度の場合は 53 です。
- PIDc(h) 16 進数 h の DXC を持つデータ例外 (条件 c) に関するプログラム割り込み。19-16ページの図19-13 を参照してください。
- SFi IEEE 無効操作フラグ (FPC 1.0)。
- SFu IEEE アンダーフロー・フラグ (FPC 1.3)。
- SFx IEEE 不正確フラグ (FPC 1.4)。
- T(r,n) 結果の r (剰余) および n (整数商) が、それぞれターゲット・オペランド 1 および 3 に入れられます。

図 19-24 (2/2). 結果: DIVIDE TO INTEGER

EXTRACT FPC



FPC (浮動小数点制御) レジスタの内容が、R₁ に指定されている汎用レジスタのビット位置 32-63 に入れます。この汎用レジスタのビット位置 0-31 は変更されません。

条件コード: コードは変更されません。

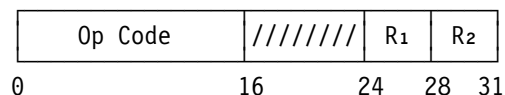
IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)

LOAD AND TEST

ニーモニック R₁,R₂ [RRE]



ニーモニック	命令コード	オペランド
LTEBR	'B302'	短精度 BFP
LTDBR	'B312'	長精度 BFP
LTXBR	'B342'	拡張精度 BFP

第 2 オペランドが第 1 オペランドの位置に入れられ、その符号と大きさがテストされ、その結果に基づいて条件コードが設定されます。条件コードは、第 2 オペランドを 0 として比較を行った場合と同じに設定されます。

第 2 オペランドが、無変更のまま第 1 オペランドの位置に入れられます。第 2 オペランドが SNaN である場合は、IEEE 無効操作条件が認識されます。割り込みがない場合は、結果は対応する QNaN です。

この命令の結果についての詳細説明は、19-28ページの図19-21 を参照してください。

LTXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 結果は NaN。

IEEE 例外条件:

- 無効操作例外

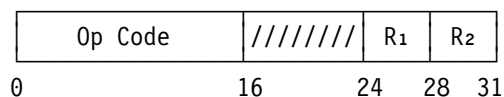
プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LTXBR のみ)

プログラミング上の注意: IEEE 標準では、オペランドが SNaN であるときに、LOAD AND TEST のような命令が無効操作を通知するかどうかは、任意とされています。通知を送ることが望ましくない場合にオペランドをテストするには、TEST DATA CLASS を使用できません。

LOAD COMPLEMENT

ニーモニック R₁,R₂ [RRE]



ニーモニック	命令コード	オペランド
LCEBR	'B303'	短精度 BFP
LCDBR	'B313'	長精度 BFP
LCXBR	'B343'	拡張精度 BFP

第 2 オペランドが、符号ビットを反転して第 1 オペランド位置に入れられます。

オペランドが 0 の場合でも、符号ビットは反転されません。第 2 オペランドのその他の部分は、変更なしで第 1 オペランド位置に入れられます。QNaN または SNaN も含めて、どのようなオペランドの場合も符号が反転され、算術例外が起きることはありません。

LCXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 結果は NaN。

IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LCXBR のみ)

プログラミング上の注意: IEEE 標準では、オペランドが SNaN であるときに、LOAD COMPLEMENT のような命令が無効操作を通知するかどうかは、任意とされています。通知が必要な場合は、この命令と一緒に LOAD AND TEST を使用することができます。

LOAD FP INTEGER

ニーモニック R₁, M₃, R₂ [RRF]

Op Code	M ₃	////	R ₁	R ₂
0	16	20	24	28 31

ニーモニック	命令コード	オペランド
FIEBR	'B357'	短精度 BFP
FIDBR	'B35F'	長精度 BFP
FIXBR	'B347'	拡張精度 BFP

第 2 オペランドが、同じ浮動小数点フォーマットの整数値に丸められ、その結果が第 1 オペランド位置に入れます。

第 2 オペランドが数値であれば、M₃ フィールド内の修飾子で指定されている方法で整数値に丸められます。

M₃ 丸め方式

- 0 現行丸めモード
- 1 バイアス付き最近値丸め
- 4 最近値丸め
- 5 0 方向への丸め
- 6 +∞ 方向への丸め
- 7 -∞ 方向への丸め

0、1、または 4-7 以外の修飾子は無効です。

修飾子フィールドが 0 のときは、丸めは、FPC レジスタに指定されている現行丸めモードにより制御されます。このフィールドが 0 でないときは、現行丸めモードに関係なく、修飾子に指定されている丸めが行われます。修飾子 4-7 の丸めは、それぞれ、丸めモード 0-3 (2 進数 00-11) と同じです。バイアス付き最近値丸め (修飾子 1) は、最近値丸め (修飾子 4) とおむね同じですが、第 2 オペランドが 2 つの整数間の正確に半分のある場合に、バイアス付き丸めでは、結果が、絶対値が次に大きい整数になるという点が異なります。

割り込みがない場合、第 2 オペランドが無限大または QNaN であるときは、結果はそのオペランドです。第 2 オペランドが SNaN であるときは、結果はそれに対応する QNaN です。

結果の符号は、結果が 0 の場合も含めて、第 2 オペランドの符号と同じです。

この命令の結果についての詳細説明は、19-39ページの図19-25 を参照してください。

M₃ フィールドは、有効な修飾子を指定していなければなりません。また、FIXBR の場合は、R フィールドは有効な浮動小数点レジスタのペアを指定していなければなりません。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外
- 不正確例外

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外

プログラミング上の注意:

1. LOAD FP INTEGER は、BFP 数を整数値に丸めます。これらの整数 (BFP フォーマットのまま残される) を、2 進整数 (固定小数点フォーマットをとる) と混同しないでください。
2. BFP オペランドが、すでに整数としての十分な大きさの指数を持つ数値である場合は、結果の値は変わりません。

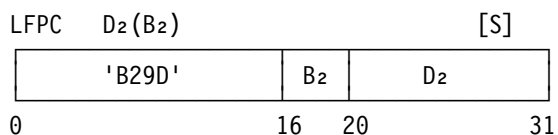
オペランド (a)	n は 不正確 (n≠a) か	無効操作 マスク (FPC 0.0)	不正確 マスク (FPC 0.4)	n は 増加済み (n > a) か	結果
-∞	いいえ ¹	-	-	-	T(-∞)
-Fn	いいえ	-	-	-	T(n)
-Fn	はい	-	0	-	T(n), SFx←1
-Fn	はい	-	1	いいえ	T(n), PIDx(08)
-Fn	はい	-	1	はい	T(n), PIDy(0C)
-0	いいえ ¹	-	-	-	T(-0)
+0	いいえ ¹	-	-	-	T(+0)
+Fn	いいえ	-	-	-	T(n)
+Fn	はい	-	0	-	T(n), SFx←1
+Fn	はい	-	1	いいえ	T(n), PIDx(08)
+Fn	はい	-	1	はい	T(n), PIDy(0C)
+∞	いいえ ¹	-	-	-	T(+∞)
QNaN	いいえ ¹	-	-	-	T(a)
SNaN	いいえ ¹	0	-	-	T(a*), SFi←1
SNaN	いいえ ¹	1	-	-	PIDi(80)

説明:

- 結果は、この条件またはマスク・ビットに応じて変化しません。
- * SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。
- ¹ この条件は、この列の左側にあるいずれかの条件の状態に応じて真になります。
- n 指定されている丸めモードを使用してソース値 a を整数に丸めたときに得られる値。
- Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。
- PIDc(h) 16 進数 h の DXC を持つデータ例外 (条件 c) に関するプログラム割り込み。19-16ページの図19-13 を参照してください。
- SFi IEEE 無効操作フラグ (FPC 1.0)。
- SFx IEEE 不正確フラグ (FPC 1.4)。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。

図 19-25. 結果: LOAD FP INTEGER

LOAD FPC



記憶域内にある 4 バイトの第 2 オペランドが、FPC (浮動小数点制御) レジスターにロードされます。

FPC レジスター内の未割り当てのビット位置に対応するビットは、すべて 0 でなければなりません。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件: なし。

プログラム例外:

- アクセス例外 (取り出し: オペランド 2)
- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外

LOAD LENGTHENED

ニーモニック 1 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
LDEBR	'B304'	短精度 BFP オペランド 2、長精度 BFP オペランド 1
LXDBR	'B305'	長精度 BFP オペランド 2、拡張精度 BFP オペランド 1
LXEBR	'B306'	短精度 BFP オペランド 2、拡張精度 BFP オペランド 1

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	/	D ₂	////////	Op Code
0	8	12	16	20	32	40	47

ニーモニック	命令コード	オペランド
LDEB	'ED04'	短精度 BFP オペランド 2、長精度 BFP オペランド 1
LXDB	'ED05'	長精度 BFP オペランド 2、拡張精度 BFP オペランド 1
LXEB	'ED06'	短精度 BFP オペランド 2、拡張精度 BFP オペランド 1

第 2 オペランドが長いフォーマットに拡張され、結果が第 1 オペランド位置に入れられます。

結果の符号はソースの符号と同じです。第 2 オペランドの指数は、結果のフォーマットでの対応する指数に変換され、小数部は、右側に 0 を付加して拡張されます。第 2 オペランドが無大である場合は、結果は同じ符号の無限大です。第 2 オペランドが SNaN である場合は、

IEEE 無効操作条件が認識されます。割り込みがない場合は、結果は、対応する QNaN の小数部を拡張した値です。

この命令の結果についての詳細説明は、19-28ページの図19-21 を参照してください。

LXDB、LXDBR、LXEB、および LXEBR の場合は、R₁ フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外

プログラム例外:

- アクセス例外 (取り出し: LDEB、LXEB、および LXDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LXEB、LXEBr、LXDB、LXDBR)

LOAD NEGATIVE

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
LNEBR	'B301'	短精度 BFP
LNDBR	'B311'	長精度 BFP
LNXBR	'B341'	拡張精度 BFP

第 2 オペランドが、符号ビットを 1 に設定して第 1 オペランド位置に入れられます。

オペランドが 0 の場合でも、符号ビットは 1 に設定されます。第 2 オペランドのその他の部分は、変更なしで第 1 オペランド位置に入れられます。QNaN または SNaN も含めて、どのようなオペランドの場合も符号が設定され、算術例外が起きることはありません。

LNXBR の場合は、R フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 --
- 3 結果は NaN。

IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LNXBR のみ)

プログラミング上の注意: IEEE 標準では、オペランドが SNaN であるときに、LOAD NEGATIVE のような命令が無効操作を通知するかどうかは、任意とされています。通知が必要な場合は、この命令と一緒に LOAD AND TEST を使用することができます。

LOAD POSITIVE

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
LPEBR	'B300'	短精度 BFP
LPDBR	'B310'	長精度 BFP
LPXBR	'B340'	拡張精度 BFP

第 2 オペランドが、符号ビットを 0 に設定して第 1 オペランド位置に入れます。

第 2 オペランドの符号ビットが 0 に設定され、その他の部分は変更なしで、第 1 オペランド位置に入れます。QNaN または SNaN も含めて、どのようなオペランドの場合も符号が設定され、算術例外が起きることはありません。

LPXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 --
- 2 結果は 0 より大きい。

3 結果は NaN。

IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LPXBR のみ)

プログラミング上の注意: IEEE 標準では、オペランドが SNaN であるときに、LOAD POSITIVE のような命令が無効操作を通知するかどうかは、任意とされています。通知が必要な場合は、この命令と一緒に LOAD AND TEST を使用することができます。

LOAD ROUNDED

ニーモニック R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック	命令コード	オペランド
LEDBR	'B344'	長精度 BFP ソース、 短精度 BFP ターゲット
LDXBR	'B345'	拡張精度 BFP ソース、 長精度 BFP ターゲット
LEXBR	'B346'	拡張精度 BFP ソース、 短精度 BFP ターゲット

ソースのフォーマットを持つ第 2 オペランドがターゲットの精度に丸められ、その結果が第 1 オペランド位置に入れます。結果の符号は第 2 オペランドの符号と同じです。

第 2 オペランドが数値であれば、現行丸めモードに従って、ターゲットの小数部の精度に丸められます。通常は、結果はターゲットのフォーマットと長さになります。ただし、IEEE オーバーフローまたは IEEE アンダーフローが起き、対応するマスク・ビットが 1 であるときは、フォーマットと長さはソースと同じで、ターゲットの精度に丸められた循環結果を生成して、操作は完了します。

この命令の結果についての詳細説明は、19-28ページの図19-21 を参照してください。

LDXBR および LEXBR の場合は、R₁ および R₂ フィールドには、それぞれ、有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外
- オーバーフロー例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (LDXBR および LEXBR)

プログラミング上の注意:

1. 丸められた結果の符号は、結果が 0 であっても、オペランドの符号と同じになります。
2. LDXBR および LEXBR の場合は、結果が拡張精度フォーマットになることがあるので、R1 フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。LDXBR および LEXBR の通常の操作では、結果のフォーマットはそれぞれ長精度と短精度であり、結果は、ターゲット・レジスター・ペアの左端 32 ビットまたは 64 ビットと置き換わります。ただし、IEEE オーバーフローまたは IEEE アンダーフローが起き、対応するマスク・ビットが 1 であるときは、結果を拡張精度フォーマットでターゲット位置に入れて、操作は完了します。したがって、プログラムでは、これらの命令が、ペアを成す両方のレジスターを更新することがあるという事実を考慮に入れる必要があります。

MULTIPLY

ニーモニック 1 R₁,R₂ [RRE]

Op Code	////////	R ₁	R ₂
0	16	24	28 31

ニーモニック 1	命令コード	オペランド
MEEBR	'B317'	短精度 BFP
MDBR	'B31C'	長精度 BFP
MXBR	'B34C'	拡張精度 BFP
MDEBR	'B30C'	短精度 BFP の乗数と被乗数、長精度 BFP の積
MXDBR	'B307'	長精度 BFP の乗数と被乗数、拡張精度 BFP の積

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]

Op Code	R ₁	X ₂	B ₂	D ₂	////////	Op Code	
0	8	12	16	20	32	40	47

ニーモニック 2	命令コード	オペランド
MEEB	'ED17'	短精度 BFP
MDB	'ED1C'	長精度 BFP
MDEB	'ED0C'	短精度 BFP の乗数と被乗数、長精度 BFP の積
MXDB	'ED07'	長精度 BFP の乗数と被乗数、拡張精度 BFP の積

第 2 オペランド (乗数) と第 1 オペランド (被乗数) の積が、第 1 オペランド位置に入れられます。

2 つの BFP オペランドが有限数である場合、両者が乗算されて、中間積が作られます。MDEB、MDEBR、MXDB、および MXDBR の場合は、中間積は長いターゲット・フォーマットに変換されます。この結果は正確であり、オーバーフローまたはアンダーフローが起きることはありません。MDB、MDBR、MEEB、MEEBR、および MXBR の場合は、結果は、現行丸めモードに従ってオペランド・フォーマットに丸められます。MEEB および MEEBR の場合は、結果は、すべての短精度フォーマットの結果の場合と同様に、ターゲット・レジスターの左端 32 ビットと置き換わり、ターゲット・レジスターの右端 32 ビットは変更されません。

積が数値の場合、その符号は、両オペランドの符号の排他論理和です。これは、積が 0 または無限大であるときの符号にも適用されます。

一方のオペランドが 0 で、もう一方が無限大である場合は、IEEE 無効操作条件が認識されます。

この命令の結果についての詳細説明は、19-43ページの図19-26を参照してください。

MXDB、MXDBR、およびMXBRのR₁フィールド、およびMXBRのR₂フィールドには、有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外
- オーバーフロー例外 (MDB、MDBR、MEEB、MEEBR、MXBR)
- アンダーフロー例外 (MDB、MDBR、MEEB、MEEBR、MXBR)
- 不正確例外 (MDB、MDBR、MEEB、MEEBR、MXBR)

プログラム例外:

- アクセス例外 (取り出し: MDEB、MEEB、MDB、およびMXDBのオペランド2のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示すDXCを伴うデータ例外
- 操作例外 (BFP機能がインストールされていない場合)
- 指定例外 (MXDB、MXDBR、MXBR)

プログラミング上の注意: BFPの乗算では、結果が数値のときは、2つのオペランドの値を入れ替えても積の値は変わりません。ただし、オペランドが両方ともQNaNであるときは、結果は第1オペランドです。また、オペランドが両方ともSNaNであり、FPCレジスタ内のIEEE無効操作マスク・ビットが0のときは、結果は第1オペランドから導き出されたQNaNです。

第1オペランド (a) の値	MULTIPLY (a·b) の結果 第2オペランド (b) が以下の値のとき							
	-∞	-Fn	-0	+0	+Fn	+∞	QNaN	SNaN
-∞	T(+∞)	T(+∞)	Xi: T(dNaN)	Xi: T(dNaN)	T(-∞)	T(-∞)	T(b)	Xi: T(b*)
-Fn	T(+∞)	R(a·b)	T(+0)	T(-0)	R(a·b)	T(-∞)	T(b)	Xi: T(b*)
-0	Xi: T(dNaN)	T(+0)	T(+0)	T(-0)	T(-0)	Xi: T(dNaN)	T(b)	Xi: T(b*)
+0	Xi: T(dNaN)	T(-0)	T(-0)	T(+0)	T(+0)	Xi: T(dNaN)	T(b)	Xi: T(b*)
+Fn	T(-∞)	R(a·b)	T(-0)	T(+0)	R(a·b)	T(+∞)	T(b)	Xi: T(b*)
+∞	T(-∞)	T(-∞)	Xi: T(dNaN)	Xi: T(dNaN)	T(+∞)	T(+∞)	T(b)	Xi: T(b*)
QNaN	T(a)	T(a)	T(a)	T(a)	T(a)	T(a)	T(a)	Xi: T(b*)
SNaN	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)

説明:

- * SNaNは、対応するQNaNに変換されてから、ターゲット・オペランド位置に入れられます。
- dNaN デフォルトのクワイエット NaN。
- Fn 0でない有限数 (正規化数と非正規化数の両方を含む)。
- R(v) 値 v に対して丸めおよび範囲アクションが行われます。19-22ページの図19-17を参照してください。
- T(x) ターゲット・オペランドの位置に値 x が入れられます。
- Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-26. 結果: MULTIPLY

MULTIPLY AND ADD

ニーモニック 1 R₁,R₃,R₂ [RRF]

Op Code	R ₁	////	R ₃	R ₂
0	16	20	24	28 31

ニーモニック 1 命令コード オペランド
 MAEBR 'B30E' 短精度 BFP
 MADBR 'B31E' 長精度 BFP

ニーモニック 2 R₁,R₃,D₂(X₂,B₂) [RXF]

Op Code	R ₃	X ₂	B ₂	D ₂	R ₁	////	Op Code
0	8	12	16	20	32	36	40 47

ニーモニック 2 命令コード オペランド
 MAEB 'ED0E' 短精度 BFP
 MADB 'ED1E' 長精度 BFP

MULTIPLY AND SUBTRACT

ニーモニック 1 R₁,R₃,R₂ [RRF]

Op Code	R ₁	////	R ₃	R ₂
0	16	20	24	28 31

ニーモニック 1 命令コード オペランド
 MSEBR 'B30F' 短精度 BFP
 MSDBR 'B31F' 長精度 BFP

ニーモニック 2 R₁,R₃,D₂(X₂,B₂) [RXF]

Op Code	R ₃	X ₂	B ₂	D ₂	R ₁	////	Op Code
0	8	12	16	20	32	36	40 47

ニーモニック 2 命令コード オペランド
 MSEB 'ED0F' 短精度 BFP
 MSDB 'ED1F' 長精度 BFP

第 3 オペランドに第 2 オペランドが乗算され、第 1 オペランドが積に加算、または積から減算されます。その和または差が第 1 オペランド位置に入れます。
MULTIPLY AND ADD および **MULTIPLY AND SUBTRACT** 操作は、以下のように要約することができます。

$$op1 = op3 \cdot op2 \pm op1$$

オペランドが有限数であるときは、第 2 および第 3 の **BFP** オペランドを乗算して中間積が作られ、その積に (またはその積から) 第 1 オペランドが代数的に加算 (または減算) されて中間和が作られます。中間和は、0 以外であれば、正規化され、現行丸めモードに従ってオペランド・フォーマットに丸められて、第 1 オペランド位置に入れます。中間積の指数および小数部はそのまま正確に維持されます。丸めおよび範囲検査は、中間和に対してのみ行われます。

MULTIPLY AND ADD の結果についての詳細説明は、19-45 ページの図 19-27 を参照してください。

MULTIPLY AND SUBTRACT の結果も、第 1 オペランドの符号を反転させて操作に使用するという点を除けば、**MULTIPLY AND ADD** の場合と同じです。

条件コード: コードは変更されません。

IEEE 例外条件:

- 無効操作例外
- オーバーフロー例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- アクセス例外 (取り出し: MAEB、MADB、MSEB、MSDB のオペランド 2)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)

プログラミング上の注意: **MULTIPLY AND ADD** および **MULTIPLY AND SUBTRACT** は、厳密な中間結果を生成し、加算または減算の後で 1 回だけ丸め操作が行われます。この定義は RS/6000 との整合性を備えており、ある種のアプリケーションにおいては、特に数学ライブラリーで使用するアルゴリズムで大きな利点をもたらします。

第 3 オペランド (a) の値	MULTIPLY AND ADD (a·b+c) の結果 (パート 1) 第 2 オペランド (b) が以下の値のとき							
	-∞	-Fn	-0	+0	+Fn	+∞	QNaN	SNaN
-∞	P(+∞)	P(+∞)	Xi: T(dNaN)	Xi: T(dNaN)	P(-∞)	P(-∞)	P(b)	Xi: T(b*)
-Fn	P(+∞)	P(a·b)	P(+0)	P(-0)	P(a·b)	P(-∞)	P(b)	Xi: T(b*)
-0	Xi: T(dNaN)	P(+0)	P(+0)	P(-0)	P(-0)	Xi: T(dNaN)	P(b)	Xi: T(b*)
+0	Xi: T(dNaN)	P(-0)	P(-0)	P(+0)	P(+0)	Xi: T(dNaN)	P(b)	Xi: T(b*)
+Fn	P(-∞)	P(a·b)	P(-0)	P(+0)	P(a·b)	P(+∞)	P(b)	Xi: T(b*)
+∞	P(-∞)	P(-∞)	Xi: T(dNaN)	Xi: T(dNaN)	P(+∞)	P(+∞)	P(b)	Xi: T(b*)
QNaN	P(a)	P(a)	P(a)	P(a)	P(a)	P(a)	P(a)	Xi: T(b*)
SNaN	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)	Xi: T(a*)

図 19-27 (1/2). 結果: MULTIPLY AND ADD

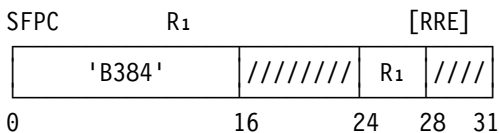
パート 1 の (P) の 値	MULTIPLY AND ADD (a•b+c) の 結果 (パート 2) 第 1 オペランド (c) が以下の値のとき							
	-∞	-Fn	-0	+0	+Fn	+∞	QNaN	SNaN
-∞	T(-∞)	T(-∞)	T(-∞)	T(-∞)	T(-∞)	Xi: T(dNaN)	T(c)	Xi: T(c*)
-Fn	T(-∞)	R(p+c)	R(p)	R(p)	R(p+c)	T(+∞)	T(c)	Xi: T(c*)
-0	T(-∞)	R(c)	T(-0)	Rezd	R(c)	T(+∞)	T(c)	Xi: T(c*)
+0	T(-∞)	R(c)	Rezd	T(+0)	R(c)	T(+∞)	T(c)	Xi: T(c*)
+Fn	T(-∞)	R(p+c)	R(p)	R(p)	R(p+c)	T(+∞)	T(c)	Xi: T(c*)
+∞	Xi: T(dNaN)	T(+∞)	T(+∞)	T(+∞)	T(+∞)	T(+∞)	T(c)	Xi: T(c*)
QNaN	T(p)	T(p)	T(p)	T(p)	T(p)	T(p)	T(p)	Xi: T(c*)

説明:

* SNaN は、対応する QNaN に変換されてから、ターゲット・オペランド位置に入れられます。
dNaN デフォルトのクワイエット NaN。
Fn 0 でない有限数 (正規化数と非正規化数の両方を含む)。
P(x) 値 x はこの図のパート 2 に適用されます。
R(v) 値 v に対して丸めおよび範囲アクションが行われます。19-22ページの図19-17 を参照してください。
Rezd 正確なゼロ差結果。19-22ページの図19-17 を参照してください。
T(x) ターゲット・オペランドの位置に値 x が入れられます。
Xi: IEEE 無効操作例外。示されている結果は、FPC 0.0 が 0 の場合のみ生成されます。

図 19-27 (2/2). 結果: MULTIPLY AND ADD

SET FPC



R₁ に指定されている汎用レジスタのビット位置
32-63 の内容が、FPC (浮動小数点制御) レジスタに
入れられます。

ビット 32-63 のうち、FPC 内の未割り当てのビット位
置に対応するビットは、すべて 0 でなければなりません。
さもないと、指定例外が認識されます。汎用レジス
ターのビット 0-31 は無視されます。

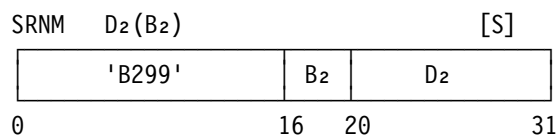
条件コード: コードは変更されません。

IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場
合)
- 指定例外

SET ROUNDING MODE



丸めモード・ビットが、第 2 オペランド・アドレスから取り出した値に設定されます。

第 2 オペランド・アドレスは、データをアドレッシングするために使用されるのではなく、FPC レジスタの丸めモード・ビットが、そのアドレスのビット 62 および 63 の値に設定されます。

第 2 オペランド・アドレスの 62 および 63 以外のビットは無視されます。

条件コード: コードは変更されません。

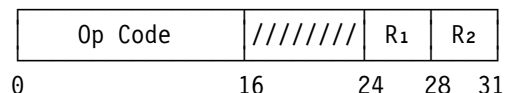
IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)

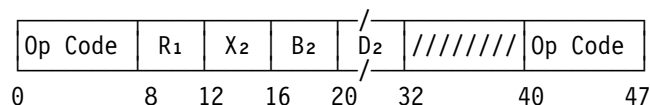
SQUARE ROOT

ニーモニック 1 R₁,R₂ [RRE]



ニーモニック 1 命令コード オペランド
 SQEBR 'B314' 短精度 BFP
 SQDBR 'B315' 長精度 BFP
 SQXBR 'B316' 拡張精度 BFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]



ニーモニック 2 命令コード オペランド
 SQEB 'ED14' 短精度 BFP
 SQDB 'ED15' 長精度 BFP

第 2 オペランドの平方根が第 1 オペランド位置に入れます。

結果は、現行丸めモードに従って丸められて、第 1 オペランド位置に入れます。

第 2 オペランドが正の有限数である場合は、結果は、その数の平方根に正符号を付けた値です。オペランドがいずれかの符号の 0 である場合は、結果はそれと同じ符号の 0 です。オペランドが +∞ である場合は、結果は +∞ です。

第 2 オペランドが 0 より小さい場合は、IEEE 無効操作条件が認識されます。

この命令の結果についての詳細説明は、19-28ページの図19-21を参照してください。

SQXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

条件コード: コードは変更されません。

IEEE 例外条件:

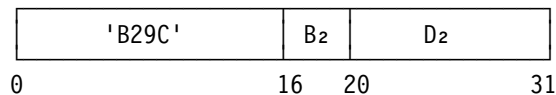
- 無効操作例外
- 不正確例外

プログラム例外:

- アクセス例外 (取り出し: SQEB および SQDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (SQXBR のみ)

STORE FPC

STFPC D₂(B₂) [S]



FPC (浮動小数点制御) レジスタの内容が、記憶域内の第 2 オペランド位置に入れます。

オペランドの長さは 4 バイトです。したがって、FPC レジスタの 32 ビットすべてが格納されます。

条件コード: コードは変更されません。

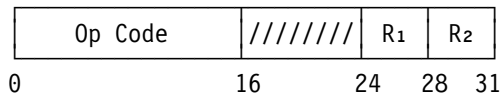
IEEE 例外条件: なし。

プログラム例外:

- アクセス例外 (格納: オペランド 2)
- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)

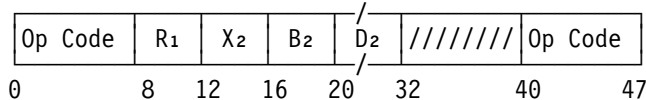
SUBTRACT

ニーモニック 1 R₁,R₂ [RRE]



ニーモニック 1 命令コード オペランド
SEBR 'B30B' 短精度 BFP
SDBR 'B31B' 長精度 BFP
SXBR 'B34B' 拡張精度 BFP

ニーモニック 2 R₁,D₂(X₂,B₂) [RXE]



ニーモニック 2 命令コード オペランド
SEB 'ED0B' 短精度 BFP
SDB 'ED1B' 長精度 BFP

第 1 オペランドから第 2 オペランドが減算され、差が第 1 オペランド位置に入られます。

SUBTRACT の実行は、第 2 オペランドの符号を逆にして演算が行われるという点を除けば、ADD と同じです。ADD の結果についての詳細説明は、19-21 ページの図19-16 を参照してください。

SXBR の場合は、R フィールドには有効な浮動小数点レジスタのペアを指定する必要があります。さもないと、指定例外が認識されます。

結果の条件コード:

- 0 結果は 0。
- 1 結果は 0 より小さい。
- 2 結果は 0 より大きい。
- 3 結果は NaN。

IEEE 例外条件:

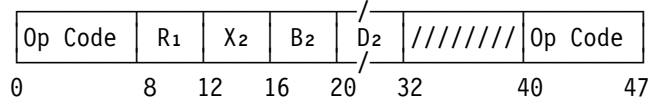
- 無効操作例外
- オーバーフロー例外
- アンダーフロー例外
- 不正確例外

プログラム例外:

- アクセス例外 (取り出し: SEB および SDB のオペランド 2 のみ)
- データ例外 (DXC 2)、BFP 命令
- IEEE 例外条件を示す DXC を伴うデータ例外
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (SXBR のみ)

TEST DATA CLASS

ニーモニック R₁,D₂(X₂,B₂) [RXE]



ニーモニック 命令コード オペランド
TCEB 'ED10' 短精度 BFP
TCDB 'ED11' 長精度 BFP
TCXB 'ED12' 拡張精度 BFP

第 1 オペランドのクラスと符号を調べて、第 2 オペランド・アドレスから 1 つのビットが選択されます。選択されたビットが 0 か 1 かに応じて、それぞれ条件コード 0 または 1 が設定されます。

第 2 オペランド・アドレスはデータをアドレッシングするために使用されるのではなく、そのアドレスの右端 12 ビット (ビット 20-31) が、オペランドのクラスと符号の 12 通りの組み合わせを指定するために使用されます。第 2 オペランド・アドレスのビット 0-19 は無視されます。

19-49 ページの図19-28 に示すように、BFP オペランドは、ゼロ、正規化数、非正規化数、無限大、クワイエット NaN、および信号付き NaN の 6 つのクラスに分類されます。

BFP オペランド・クラス	以下の符号のときに使用されるビット	
	+	-
ゼロ	20	21
正規化数	22	23
非正規化数	24	25
無限大	26	27
クワイエット NaN	28	29
信号付き NaN	30	31

図 19-28. TEST DATA CLASS で使用する第 2 オペランド・アドレスのビット

第 2 オペランド・アドレスの 1 つまたは複数のビットが 1 に設定されることがあります。第 1 オペランドのクラスと符号に対応する第 2 オペランド・アドレス・ビットが 1 であれば、条件コード 1 が設定されます。そうでない場合は、条件コード 0 が設定されます。

オペランド (SNaN および QNaN も含む) の検査により、算術例外が起きることはありません。

TCXB の場合は、R1 フィールドには有効な浮動小数点レジスターのペアを指定する必要があります。さもないと、指定例外が認識されず。

結果の条件コード:

- 0 選択されたビットは 0 (不一致)。
- 1 選択されたビットは 1 (一致)。
- 2 --
- 3 --

IEEE 例外条件: なし。

プログラム例外:

- データ例外 (DXC 2)、BFP 命令
- 操作例外 (BFP 機能がインストールされていない場合)
- 指定例外 (TCXB のみ)

プログラミング上の注意: TEST DATA CLASS は、例外が起きることも IEEE フラグが設定されることもなく、オペランドをテストできるようにするための手段として利用できます。

付録A. 数の表現と命令の使用例

数の表現	A-2	DIVIDE (D, DR)	A-19
2 進整数	A-2	EXCLUSIVE OR (X, XC, XI, XR)	A-19
符号付き 2 進整数	A-2	XC の例	A-20
符号なし 2 進整数	A-3	XI の例	A-20
10 進整数	A-4	EXECUTE (EX)	A-21
16 進浮動小数点数	A-5	INSERT CHARACTERS UNDER MASK	
変換の例	A-6	(ICM)	A-22
命令の使用例	A-6	LOAD (L, LR)	A-22
マシン・フォーマット	A-6	LOAD ADDRESS (LA)	A-23
アセンブラー言語フォーマット	A-7	LOAD HALFWORD (LH)	A-23
例で使用されているアドレッシング・モード	A-7	MOVE (MVC, MVI)	A-23
一般命令	A-7	MVC の例	A-23
ADD HALFWORD (AH)	A-7	MVI の例	A-24
AND (N, NC, NI, NR)	A-7	MOVE INVERSE (MVCIN)	A-24
NI の例	A-8	MOVE LONG (MVCL)	A-25
リンケージ命令 (BAL, BALR, BAS,		MOVE NUMERICS (MVN)	A-26
BASR, BASSM, BSM)	A-8	MOVE STRING (MVST)	A-26
BALR および BASR のその他の例	A-9	MOVE WITH OFFSET (MVO)	A-26
BRANCH AND STACK (BAKR)	A-10	MOVE ZONES (MVZ)	A-27
BAKR の例 1	A-10	MULTIPLY (M, MR)	A-27
BAKR の例 2	A-11	MULTIPLY HALFWORD (MH)	A-28
BAKR の例 3	A-11	OR (O, OC, OI, OR)	A-28
BRANCH ON CONDITION (BC, BCR)	A-11	OI の例	A-28
BRANCH ON COUNT (BCT, BCTR)	A-12	PACK (PACK)	A-28
BRANCH ON INDEX HIGH (BXH)	A-12	SEARCH STRING (SRST)	A-29
BXH の例 1	A-12	SRST の例 1	A-29
BXH の例 2	A-13	SRST の例 2	A-29
BRANCH ON INDEX LOW OR EQUAL		SHIFT LEFT DOUBLE (SLDA)	A-29
(BXLE)	A-13	SHIFT LEFT SINGLE (SLA)	A-30
BXLE の例 1	A-13	STORE CHARACTERS UNDER MASK	
BXLE の例 2	A-14	(STCM)	A-30
COMPARE AND FORM CODEWORD		STORE MULTIPLE (STM)	A-31
(CFC)	A-14	TEST UNDER MASK (TM)	A-31
COMPARE HALFWORD (CH)	A-14	TRANSLATE (TR)	A-32
COMPARE LOGICAL (CL, CLC, CLI,		TRANSLATE AND TEST (TRT)	A-32
CLR)	A-15	UNPACK (UNPK)	A-34
CLC の例	A-15	UPDATE TREE (UPT)	A-34
CLI の例	A-15	10 進数命令	A-34
CLR の例	A-15	ADD DECIMAL (AP)	A-34
COMPARE LOGICAL CHARACTERS		COMPARE DECIMAL (CP)	A-34
UNDER MASK (CLM)	A-16	DIVIDE DECIMAL (DP)	A-35
COMPARE LOGICAL LONG (CLCL)	A-16	EDIT (ED)	A-35
COMPARE LOGICAL STRING (CLST)	A-18	EDIT AND MARK (EDMK)	A-36
CONVERT TO BINARY (CVB)	A-18	MULTIPLY DECIMAL (MP)	A-37
CONVERT TO DECIMAL (CVD)	A-19	SHIFT AND ROUND DECIMAL (SRP)	A-37

10 進数の左シフト	A-37
10 進数の右シフト	A-37
10 進数の右シフトと丸め	A-38
10 の可変累乗による乗算	A-38
ZERO AND ADD (ZAP)	A-38
16 進浮動小数点命令	A-39
ADD NORMALIZED (AD, ADR, AE, AER, AXR)	A-39
ADD UNNORMALIZED (AU, AUR, AW, AWR)	A-39
COMPARE (CD, CDR, CE, CER)	A-40
DIVIDE (DD, DDR, DE, DER)	A-40
HALVE (HDR, HER)	A-41
MULTIPLY (MD, MDR, MDE, MDER, MXD, MXDR, MXR)	A-41
16 進浮動小数点数の変換	A-42
固定小数点から 16 進浮動小数点へ	A-42
16 進浮動小数点から固定小数点へ	A-42

マルチプログラミングとマルチプロセッシングの例	A-43
即値 OR を用いたプログラム障害の例	A-43
条件付きスワップ命令 (CS, CDS)	A-43
単一ビットの設定	A-44
カウンターの更新	A-45
POST と WAIT のバイパス	A-45
POST バイパス・ルーチン	A-45
WAIT バイパス・ルーチン	A-46
ロック/アンロック	A-46
競合に対する LIFO キューイングを使用したロック/アンロック	A-46
競合に対する FIFO キューイングを使用したロック/アンロック	A-47
フリー・プールの操作	A-49
PERFORM LOCKED OPERATION (PLO)	A-50
ソート命令	A-51
ツリー・フォーマット	A-51
ソート命令の使用例	A-53

この付録の内容は、「Enterprise Systems Architecture/390 Principles of Operation」(SA22-7201)に含まれているものと同じであり、レジスター・サイズの拡張や 64 ビット・アドレッシング・モードに合わせた改訂はされていません。

数の表現

2 進整数

符号付き 2 進整数

符号付き 2 進整数は、ほとんどの場合、ハーフワード (16 ビット) またはワード (32 ビット) で表されます。どちらの長さの場合も、左端のビット (ビット 0) は数の符号です。残りのビット (ハーフワードの場合はビット 1-15、ワードの場合は 1-31) は、数の絶対値を指定するために使用されます。2 進整数は、小数点 (2 進小数点) が右側の固定位置にあると見なされるため、固定小数点数とも呼ばれます。位取りが必要な場合はプログラマーが指定します。

正の 2 進整数は、符号ビットが 0 である真の 2 進表記で表されます。負の 2 進整数は、符号ビットが 1 である 2 の補数表記で表されます。いずれの場合も、整数の符号ビットと左端の有効ビットの間にあるビットは、す

べて符号ビットと同じ値になります (つまり、正の数の場合はすべて 0 で、負の数の場合はすべて 1 です)。

負の 2 進整数は、正の 2 進整数の各ビットを逆転し、1 を追加することにより、2 の補数の形で表されます。例えば、ハーフワード・フォーマットを使用した場合、10 進数値 +26 に相当する 2 進数は、次のようにして負の値 (-26) にされます。

```

+26      0 000 0000 0001 1010
逆転     1 111 1111 1110 0101
1 を加算      1
-----
-26      1 111 1111 1110 0110 (2 の補数形式)
(S は符号ビットです。)

```

これは、

```
00000000 00011010
```

上記の数値から下記の数値を引くと同じことです。

```
1 00000000 00000000
```

同じ方法で、負の 2 進整数を正の数に変更することもできます。

次の加算の例は、2 の補数の計算とオーバーフロー条件を示しています。8 個のビット位置のみが使用されています。

```

1.  +57 = 0011 1001
     +35 = 0010 0011
     -----
     +92 = 0101 1100

```

2.
$$\begin{array}{r} +57 = 0011\ 1001 \\ -35 = 1101\ 1101 \\ \hline +22 = 0001\ 0110 \end{array}$$
 オーバーフローなし。左端位置への繰り上がりがあり、左端位置からの繰り上がりもある。
3.
$$\begin{array}{r} +35 = 0010\ 0011 \\ -57 = 1100\ 0111 \\ \hline -22 = 1110\ 1010 \end{array}$$
 符号変更のみ。左端位置への繰り上がりも、左端位置からの繰り上がりもない。
4.
$$\begin{array}{r} -57 = 1100\ 0111 \\ -35 = 1101\ 1101 \\ \hline -92 = 1010\ 0100 \end{array}$$
 オーバーフローなし。左端位置への繰り上がりがあり、左端位置からの繰り上がりもある。
5.
$$\begin{array}{r} +57 = 0011\ 1001 \\ +92 = 0101\ 1100 \\ \hline +149 = *1001\ 0101 \end{array}$$
 *オーバーフロー。左端位置への繰り上がりがあり、左端位置からの繰り上がりはない。
6.
$$\begin{array}{r} -57 = 1100\ 0111 \\ -92 = 1010\ 0100 \\ \hline -149 = *0110\ 1011 \end{array}$$
 *オーバーフロー。左端位置への繰り上がりはないが、左端位置からの繰り上がりがある。

オーバーフローの有無は、繰り上がりから判別することができます。

$2^{31}-1 = 2\ 147\ 483\ 647$	$= 0\ 111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$
2^{16}	$= 65\ 536 = 0\ 000\ 0000\ 0000\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000$
2^0	$= 1 = 0\ 000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$
0	$= 0 = 0\ 000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$
-2^0	$= -1 = 1\ 111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$
-2^1	$= -2 = 1\ 111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110$
-2^{16}	$= -65\ 536 = 1\ 111\ 1111\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000$
$-2^{31}+1$	$= -2\ 147\ 483\ 647 = 1\ 000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$
-2^{31}	$= -2\ 147\ 483\ 648 = 1\ 000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

図 A-1. 32 ビット符号付き 2 進整数

符号なし 2 進整数

一部の命令 (例えば、ADD LOGICAL) では、2 進整数は符号付きではなく符号なしとして扱われます。符号なし 2 進整数のフォーマットは符号付き 2 進整数と同じですが、左端の 1 ビットが符号ビットではなく通常の数字ビットとして解釈される点が異なります。符号なし 2 進整数はすべて正と見なされるので、補数表記はありません。

以下の例は、符号なし 2 進整数の加算を示しています。8 個のビット位置のみが使用されています。これらの例

- 次の場合は、オーバーフローはありません。

1. 左端ビット位置への繰り上がりも、左端ビット位置からの繰り上がりもない (例 1 および 3)。
2. 左端位置への繰り上がりがあり、左端位置からの繰り上がりもある (例 2 および 4)。

- 次の場合は、オーバーフローがあります。

1. 左端位置への繰り上がりがあるが、左端位置からの繰り上がりはない (例 5)。
2. 左端位置への繰り上がりはないが、左端位置からの繰り上がりはある (例 6)。

以下に示すのは、どれも 16 ビットの符号付き 2 進整数です。最初の数値は、最大の正の 16 ビット 2 進整数です。最後の数値は、最大の負の 16 ビット 2 進整数 (最大の絶対値を持つ負の 16 ビット 2 進整数) です。

$$\begin{array}{r} 2^{15}-1 = 32,767 = 0\ 111\ 1111\ 1111\ 1111 \\ 2^0 = 1 = 0\ 000\ 0000\ 0000\ 0001 \\ 0 = 0 = 0\ 000\ 0000\ 0000\ 0000 \\ -2^0 = -1 = 1\ 111\ 1111\ 1111\ 1111 \\ -2^{15} = -32,768 = 1\ 000\ 0000\ 0000\ 0000 \end{array}$$

図A-1 は、幾つかの 32 ビット符号付き 2 進整数を降順に示しています。最初の数は、32 ビットで表現できる最大の正の 2 進整数であり、最後の数は、32 ビットで表現できる最大の負の 2 進整数です。

は、それぞれ、同じ番号を持つ符号付き 2 進整数の例と対応しています。

1.
$$\begin{array}{r} 57 = 0011\ 1001 \\ 35 = 0010\ 0011 \\ \hline 92 = 0101\ 1100 \end{array}$$
2.
$$\begin{array}{r} 57 = 0011\ 1001 \\ 221 = 1101\ 1101 \\ \hline 278 = *0001\ 0110 \end{array}$$
 *左端位置からの繰り上がり

$$\begin{array}{l} 3. \quad 35 = 0010 \ 0011 \\ \quad 199 = 1100 \ 0111 \end{array}$$

$$\hline 234 = 1110 \ 1010$$

$$\begin{array}{l} 4. \quad 199 = 1100 \ 0111 \\ \quad 221 = 1101 \ 1101 \end{array}$$

$$\hline 420 = *1010 \ 0100 \quad *左端位置からの繰り上がり$$

$$\begin{array}{l} 5. \quad 57 = 0011 \ 1001 \\ \quad 92 = 0101 \ 1100 \end{array}$$

$$\hline 149 = 1001 \ 0101$$

$$\begin{array}{l} 6. \quad 199 = 1100 \ 0111 \\ \quad 164 = 1010 \ 0100 \end{array}$$

$$\hline 363 = *0110 \ 1011 \quad *左端位置からの繰り上がり$$

左端ビット位置からの繰り上がりは、適用のしかたに応じて、オーバーフローを意味することもあり、意味しないこともあります。

図A-2は、幾つかの32ビット符号なし2進整数を降順に示しています。

$2^{32}-1$	=	4 294 967 295	=	1111 1111 1111 1111 1111 1111 1111 1111
2^{31}	=	2 147 483 648	=	1000 0000 0000 0000 0000 0000 0000 0000
$2^{31}-1$	=	2 147 483 647	=	0111 1111 1111 1111 1111 1111 1111 1111
2^{16}	=	65 536	=	0000 0000 0000 0000 0001 0000 0000 0000
2^0	=	1	=	0000 0000 0000 0000 0000 0000 0000 0001
0	=	0	=	0000 0000 0000 0000 0000 0000 0000 0000

図 A-2. 32 ビット符号なし 2 進整数

10 進整数

10 進整数は、1 個以上の 10 進数字と 1 個の符号から成っています。個々の数字および符号は、それぞれ 4 ビットのコードで表されます。10 進数字は 2 進化 10 進数 (BCD) 形式をとり、0-9 の値は 0000-1001 としてエンコードされます。符号は、通常、負の場合は 1100 (16 進数 C)、正の場合は 1101 (16 進数 D) で表されます。これらは優先符号コードであり、10 進算術演算の結果を表すためにマシンが生成するコードです。ほかに、幾つかの代替符号コードがあります (正符号を表す 1010、1110、および 1111、負符号を表す 1011)。マシンは、ソース・オペランド内の代替符号コードは有効なものとして受け入れますが、結果を表すときには代替コードは使用しません。

10 進整数は、1 バイトから 16 バイトの長さをとることができます。10 進数には、パックおよびゾーンの 2 つのフォーマットがあります。パック・フォーマットでは、各バイトにそれぞれ 2 個の 10 進数字が含まれていますが、右端の 1 バイトでは右側半分には符号が入っています。10 進数演算では、パック・フォーマットの 10 進数字の数は、1 から 31 までの範囲内で変化します。10 進整数は完全なバイトから成っていなければならないので、右端に符号コードがなければならぬので、10 進数字の数は常に奇数です。偶数の有効数字を必要とする場合は、左側に先行ゼロを挿入する必要があります。

ゾーン・フォーマットでは、各バイトには、右側に 1 個の 10 進数字と、左側にゾーン・コード 1111 (16 進数 F) が入りますが、右端の 1 バイトにはゾーン・コード

の代わりに符号コードが入ります。したがって、ゾーン・フォーマットの 10 進整数の数字の数は、1 個から 16 個までです。ゾーン・フォーマットは、拡張 2 進化 10 進コード (EBCDIC) として直接、入力および出力に使用できますが、符号は右端の数字から切り離して、別個の文字として扱う必要があります。ただし、正 (符号なし) の数の場合は、符号は単に右端の数字のゾーン・コードによってのみ表すことができます。これは、ゾーン・コードが、正符号を表すものとして受け入れられる代替コードの 1 つであるからです。

どちらのフォーマットでも、負の 10 進整数は、別個の符号を持つ真の表記で表されます。10 進整数の小数点 (10 進小数点) は、2 進整数の場合と同様に、右側の位置に固定されているものと見なされます。位取りが必要な場合はプログラマーが指定します。

以下の例は、10 進整数を 16 進数表記で表すとどうなるかを示しています。

10 進数	パック・フォーマット	ゾーン・フォーマット
+123	12 3C または 12 3F	F1 F2 C3 または F1 F2 F3
-4321	04 32 1D	F4 F3 F2 D1
+000050	00 00 05 0C または 00 00 05 0F	F0 F0 F0 F0 F5 C0 または F0 F0 F0 F0 F5 F0
-7	7D	D7
00000	00 00 0C または 00 00 0F	F0 F0 F0 F0 C0 または F0 F0 F0 F0 F0

特定の状況下では、負符号を持つ 0 (負の 0) が生成されることがあります。例えば、被乗数が次のとおりであるとします。

00 12 3D (-123)

これに次の乗数を乗算します。

0C (+0)

積は次のようになります。

00 00 0D (-0)

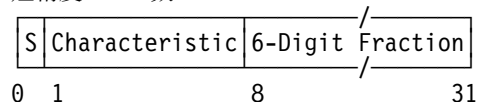
これは、積の値が 0 であっても、符号は代数の符号規則に従って設定されるからです。ただし、負の 0 は、10 進数比較において正の 0 と等しいものと見なされるという点で、正の 0 と等価です。

16 進浮動小数点数

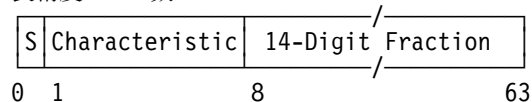
16 進浮動小数点 (HFP) 数は、16 進数の小数部に 16 の累乗数を掛けた積として表現されます。浮動小数点という言葉は、16 進小数点の位置決め、つまり位取りが、マシンにより自動的に維持されることを意味します。

HFP 数において、その数の有効数字を表す部分を小数部と呼びます。もう 1 つの部分は、16 を何乗するか (指数) を指定するもので、これは小数点の位置を示します。小数部と指数は、32 ビット (短精度フォーマット)、64 ビット (長精度フォーマット)、または 128 ビット (拡張精度フォーマット) で表すことができます。

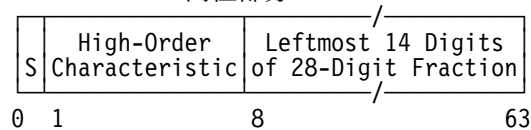
短精度 HFP 数



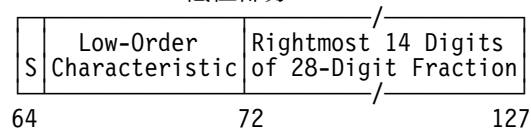
長精度 HFP 数



拡張精度 HFP 数
高位部分



低位部分



HFP 数には、小数部用と指数用の 2 つの符号があります。小数部の符号は数全体の符号でもあり、各フォーマットの左端ビットに入れられます (正符号は 0、負符号は 1)。小数部の数字部分は、符号に関係なく真の表記で表されます。数字部分は、短精度フォーマットではビット 8-31、長精度フォーマットではビット 8-63 に含まれ、拡張精度フォーマットでは、ビット 8-63 およびビット 72-127 に含まれています。

指数の符号は、指数を 64 増し表記で表すことによって得られます。つまり、指数を符号付き数として 64 に加算したものです。その結果の数値を指数部と呼びます。指数部は、どのフォーマットの場合もビット 1-7 に入れます。小数部は 0 から 127 までの範囲内の値をとることができ、したがって、指数部は -64 から 0、さらに +63 までの範囲内で変化します。したがって、位取り乗数の範囲は、 16^{-64} から $+63$ までです。0 でない小数部を正規化した場合、その値は 1 より小さく、 $1/16$ 以上になります。したがって、絶対値が M の正規化された浮動小数点数がとり得る値の範囲は次のとおりです。

$$16^{-65} \leq M < 16^{63}$$

これを 10 進数 に換算すると、次のようになります。

$$16^{-65} \text{ は約 } 5.4 \times 10^{-79}$$

$$16^{63} \text{ は約 } 7.2 \times 10^{75}$$

もっと厳密に言えば、以下のようになります。

短精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-6}) \times 16^{63}$$

長精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-14}) \times 16^{63}$$

拡張精度フォーマットの場合:

$$16^{-65} \leq M \leq (1 - 16^{-28}) \times 16^{63}$$

所定の小数部の長さ (6 桁、14 桁、または 28 桁) において、HFP 演算で最大の精度が達成されるのは、小数部が正規化された場合です。左端の数字 (ビット位置 8、9、10、および 11) が 0 以外であれば、小数部は正規化されています。左端の数字のビットがすべて 0 である場合は、小数部は正規化されていません。

オペランドの正規化が必要な場合は、自動正規化を行う HFP 命令を使用します。自動正規化では、左端の数字

1.0	= +1/16x16 ¹	= 0 100 0001 0001 0000 0000 0000 0000 0000 ₂
0.5	= +8/16x16 ⁰	= 0 100 0000 1000 0000 0000 0000 0000 0000 ₂
1/64	= +4/16x16 ⁻¹	= 0 011 1111 0100 0000 0000 0000 0000 0000 ₂
0.0	= +0 x16 ⁻⁶⁴	= 0 000 0000 0000 0000 0000 0000 0000 0000 ₂
-15.0	= -15/16x16 ¹	= 1 100 0001 1111 0000 0000 0000 0000 0000 ₂
5.4x10 ⁻⁷⁹	≈ +1/16x16 ⁻⁶⁴	= 0 000 0000 0001 0000 0000 0000 0000 0000 ₂
7.2x10 ⁷⁵	≈ (1-16 ⁻⁶)x16 ⁶³	= 0 111 1111 1111 1111 1111 1111 1111 1111 ₂

図 A-3. 正規化された短精度 16 進浮動小数点数

変換の例

10 進数 59.25 を短精度 HFP 数に変換します。(別の付録に、16 進および 10 進の整数および小数部を変換するための表を示してあります。)

1. 該当の数が、10 進整数と 10 進小数部に分けられます。

$$59.25 = 59 + 0.25$$

2. 10 進整数が 16 進表現に変換されます。

$$59_{10} = 3B_{16}$$

3. 10 進小数部が 16 進表現に変換されます。

$$0.25_{10} = 0.4_{16}$$

4. 整数部分と小数部分が組み合わせられ、小数部 x 16 の累乗 (指数) の形で表されます。

$$3B.4_{16} = 0.3B4_{16} \times 16^2$$

5. 指数から指数部が求められ、2 進数に変換されます。

$$\begin{array}{lcl} \text{基数} + \text{指数} & = & \text{指数部} \\ 64 + 2 & = & 66 = 1000010 \end{array}$$

6. 小数部が 2 進数に変換され、16 進表現用にグループ分けされます。

$$.3B4_{16} = .0011 1011 0100$$

7. 指数部と小数部が短精度フォーマットで格納されません。符号位置には、小数部の符号が入ります。

位置が 0 以外の数字になるまで、小数部が左にシフトされます (一度に 4 ビットずつシフトされます)。1 桁シフトされるたびに、指数部の値は 1 ずつ減少します。

図A-3 は、正規化された短精度 HFP 数の例を示しています。最後の 2 つの数は、最小および最大の正規化された正の数を表しています。

S	指数部	小数部
0	1000010	0011 1011 0100 0000 0000 0000

A-42ページの『16 進浮動小数点数の変換』に、符号付き 2 進整数と HFP 数の間の変換のために使用できる命令シーケンスの例を示してあります。

命令の使用例

以下のセクションでは、多くの非特権命令の使用例を示します。これらの例を考察する前に、必ず該当の命令に関する説明を読んでおくようにしてください。

命令の使用例は、主としてアセンブラ言語のプログラマーを対象として書かれているので、適宜のアセンブラ言語関連資料と併せて使用してください。

ほとんどの例では、1 つの特定の命令について、アセンブラ言語ステートメントで書いた形式と、記憶域内でアセンブルされた後の形式 (マシン・フォーマット) の両方を示してあります。

マシン・フォーマット

マシン・フォーマットの値は、特に断らない限り、すべて 16 進数表記で示してあります。記憶域アドレスも 16 進数で示されています。16 進数オペランドは、読者にとって例を分かりやすくするために必要と認められる場合は、2 進数、10 進数、またはその両方に変換して示してあります。

アセンブラー言語フォーマット

アセンブラー言語ステートメントでは、レジスターおよび長さは 10 進数で表されます。変位、即値オペランド、およびマスクは、10 進数、16 進数、または 2 進数のいずれかの表記で示されます。例えば、12、 $X'C'$ 、および $B'1100'$ は同じ値を表しています。レジスターまたは記憶位置内の値が「無意味」として示されているときは、その値は命令の実行時に置き換えられます。

SS フォーマットの命令がアセンブラー言語で書かれる場合、長さはフィールドに含まれるバイトの合計数で表されます。マシン定義はこれとは異なり、長さフィールドはフィールド・アドレスに加算するバイト数を指定します。この加算により、フィールドの最後のバイトのアドレスが求められます。したがって、マシンでの長さはアセンブラー言語での長さより 1 だけ小さくなります。アセンブラー・プログラムは、命令をアセンブルするときに、指定された長さから自動的に 1 を引きます。

一部の例では、例を単純化するためにシンボリック・アドレスが使用されています。アセンブラー言語ステートメントでは、シンボリック・アドレスは、例えば **FLAGS** などのように、すべて大文字で表記したニーモニックで表されます。これにより、データまたはプログラム制御情報を含む記憶位置のアドレスを示すことができます。シンボリック・アドレスが使用されていると、アセンブラーは、プログラマーの指定に従って、実際の基数値および変位値を供給します。したがって、アセンブラー言語フォーマットでもマシン言語フォーマットでも、基数および変位の実際の値は示されていません。アセンブラー言語フォーマットの場合は、命令のフィールドを指定するラベルの中で、オペランド・アドレスを指定する基数フィールドと変位フィールドの組み合わせを示すために「S」という文字が使用されています。(例えば、S2 は B2 と D2 の組み合わせを表します。) マシン言語フォーマットでは、基数アドレス・コンポーネントと変位アドレス・コンポーネントはアスタリスク(****)で示されます。

例で使用されているアドレッシング・モード

例では、特に断りがない限り、24 ビット・アドレッシング・モードを想定しています。

一般命令

(一般命令の詳しい説明については、第7章、『一般命令』を参照してください。)

ADD HALFWORD (AH)

ADD HALFWORD 命令は、記憶域内の 2 バイト・フィールドの内容を、レジスターの内容に代数的に加算します。記憶域オペランドは、取り出された後で 32 ビットに拡張されてから、加算操作に使用されます。この拡張では、左端ビット(符号)が左側 16 桁に入れられます。例えば、記憶位置 2000-2001 の内容をレジスター 5 に加算するとします。それぞれの初期値は以下のとおりであるとします。

レジスター 5 の内容は、 $00\ 00\ 00\ 19 = 25_{10}$ 。

記憶位置 2000-2001 の内容は、 $FF\ FE = -2_{10}$ 。

レジスター 12 の内容は、 $00\ 00\ 18\ 00$ 。

レジスター 13 の内容は、 $00\ 00\ 01\ 50$ 。

必要な命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令 コード	R ₁	X ₂	B ₂	D ₂
4A	5	D	C	6B0

アセンブラー・フォーマット
命令コード R₁,D₂(X₂,B₂)

AH 5,X'6B0'(13,12)

この命令を実行すると、レジスター 5 の内容は $00\ 00\ 00\ 17 = 23_{10}$ となります。結果が 0 より大きいことを示す条件コード 2 が設定されます。

AND (N, NC, NI, NR)

2 個のビットにブール演算子 AND を適用すると、両方のビットが 1 であれば結果は 1 となり、それ以外の場合は結果は 0 になります。2 個のバイトに対して AND 演算を行うと、対応するビットが一對ずつ別個に処理されます。1 つのビット位置と他のビット位置との間には関連性はありません。以下の例は、2 個のバイトに対する AND 演算の結果を示しています。

第 1 オペランドのバイト: 0011 0101₂

第 2 オペランドのバイト: 0101 1100₂

結果のバイト: 0001 0100₂

NI の例

AND 命令は、特定のビットを 0 に設定するためによく使用されます。例えば、記憶位置 4891 に 0100 0011₂ が含まれているとします。このバイトの右端のビットのみを 0 に設定し、他のビットは変わらないようにするには、次のような命令を使用できます (レジスター 8 の内容は、00 00 48 90 であるものとします)。

マシン・フォーマット

命令 コード	I ₂	B ₁	D ₁
94	FE	8	001

アセンブラー・フォーマット

命令 コード	D ₁ (B ₁), I ₂
NI	1(8), X'FE'

この命令が実行されると、記憶域内のバイトと即値バイト (命令の I₂ フィールド) との AND 演算が行われます。

記憶位置 4891:	0100 0011 ₂
即値バイト:	1111 1110 ₂
結果:	0100 0010 ₂

結果のバイト (ビット 7 が 0 に設定されたもの) が、記憶位置 4891 に戻されて格納されます。そして、条件コード 1 が設定されます。

リンケージ命令 (BAL、BALR、BAS、BASR、BASSM、BSM)

4 つの非特権命令 (BRANCH AND LINK、BRANCH AND SAVE、BRANCH AND SAVE AND SET MODE、および BRANCH AND SET MODE) を、無条件ブランチ (15 に設定されたマスクを持つ BRANCH ON CONDITION) と共に使用することにより、サブルーチン同士をリンクさせることができます。BRANCH AND LINK (BAL または BALR) は、主として、システム/370 用に書かれたプログラムとの互換性を確保するために提供されています。ESA/370 を使用して実行するプログラム用には、代わりに BRANCH AND SAVE (BAS または BASR) を使用することをお勧めします。BRANCH AND SAVE AND SET MODE (BASSM) および BRANCH AND SET MODE (BSM) 命令は、サブルーチン間のリンケージを提供するほか

に、24 ビット・アドレッシング・モードと 31 ビット・アドレッシング・モードの間の切り替えも行います。これらの命令の使用方法については、『リンケージ・スタックなしのサブルーチン・リンケージ』の末尾の『プログラミング上の注意』で説明されています。(準特権命令 BRANCH AND STACK も参照してください。)

次の例では、これらの命令の演算と、無条件ブランチ命令 BRANCH ON CONDITION (15 に設定されたマスクを持つ BC または BCR) の演算が比較されます。各命令が、それぞれ現命令アドレスの位置に入れられており、次に実行される準備ができているものとします。最初の一組の例では、アドレッシング・モード・ビット (PSW のビット 32) は、最初は 0 です (24 ビット・アドレッシングが有効になっています)。2 番目の一組では、PSW のビット 32 は、最初は 1 (31 ビット・アドレッシング) です。さらに、リンケージ情報は汎用レジスター 5 に格納され、ブランチ・アドレスは汎用レジスター 6 に入っているものとします。

BALR 命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令 コード	R ₁	R ₂
05	5	6

アセンブラー・フォーマット

命令 コード	R ₁ , R ₂
BALR	5, 6

RR フォーマットのその他のリンケージ命令も上記と同じフォーマットですが、命令コードが異なります。

BASR 0D
BASSM 0C
BSM 0B

RR フォーマットの命令との比較のために、2 つの RX フォーマットの命令の結果も示しておきます。

BAL 命令のフォーマットは、以下のとおりです。

マシン・フォーマット

命令 コード	R ₁	X ₂	B ₂	D ₂
45	5	0	6	000

アセンブラー・フォーマット

命令 コード	R ₁ , D ₂ (X ₂ , B ₂)
BAL	5,0(0,6)

BAS 命令も同じフォーマットですが、命令コードは 4D です。

BCR 命令では、レジスターを 1 つだけ指定します。

マシン・フォーマット

命令 コード	M ₁	R ₂
07	F	6

アセンブラー・フォーマット

命令 コード	M ₁ , R ₂
BCR	15,6

以下のように仮定します。

レジスター 5 の内容は、BB BB BB BB。

レジスター 6 の内容は、82 46 8A CE。

PSW のビット 32-63 の内容は、

00 00 10 D6 (24 ビット・アドレッシングの
場合)。

80 00 10 D6 (31 ビット・アドレッシングの
場合)。

条件コードは、01₂。

プログラム・マスクは、1100₂。

各命令が順に実行されると、結果は以下のようになります。

最初が 24 ビット・モードのとき

命令	レジスター 5	PSW (32-63)
実行前	BB BB BB BB	00 00 10 D6
BCR 15,6	BB BB BB BB	00 46 8A CE
BAL 5,0(0,6)	9C 00 10 DA	00 46 8A CE
BAS 5,0(0,6)	00 00 10 DA	00 46 8A CE
BALR 5,6	5C 00 10 D8	00 46 8A CE
BASR 5,6	00 00 10 D8	00 46 8A CE
BASSM 5,6	00 00 10 D8	82 46 8A CE
BSM 5,6	3B BB BB BB	82 46 8A CE

最初が 31 ビット・モードのとき

命令	レジスター 5	PSW (32-63)
実行前	BB BB BB BB	80 00 10 D6
BCR 15,6	BB BB BB BB	82 46 8A CE
BAL 5,0(0,6)	80 00 10 DA	82 46 8A CE
BAS 5,0(0,6)	80 00 10 DA	82 46 8A CE
BALR 5,6	80 00 10 D8	82 46 8A CE
BASR 5,6	80 00 10 D8	82 46 8A CE
BASSM 5,6	80 00 10 D8	82 46 8A CE
BSM 5,6	BB BB BB BB	82 46 8A CE

RR フォーマットのどの命令の場合も、R₂ フィールドの値が 0 であれば、ブランチ機能が実行されないことを意味します。レジスター 0 を指しているではありません。同様に、BSM 命令の R₁ フィールドの値が 0 であれば、それは、PSW のビット 32 の古い値が保管されず、レジスター 0 は変更されないことを意味します。ただし、BAL、BALR、BAS、BASR、および BASSM 命令の R₁ フィールドには、レジスター 0 を指定できません。RX フォーマットのブランチ命令では、B₂ フィールドまたは X₂ フィールドの値が 0 であるかどうかに関係なく、ブランチが発生します。ただし、このフィールドが 0 のときは、アドレス生成のコンポーネントについては、汎用レジスター 0 の内容の代わりに 0 の値が使用されます。

プログラミング上の注意: 24 ビット・アドレッシング・モードで BAL を実行すると、レジスター 5 のビット 0 が 1 に設定されるという点に注意してください。これは、RX フォーマットの命令では ILC が 10 だからです。これは、戻りレジスターのビット 0 が呼び出し側のアドレッシング・モードを正しく反映しない唯一のケースです。24 ビット・アドレッシング・モードと 31 ビット・アドレッシング・モードのどちらでも、BALR、BAS、BASR、または BASSM を使用した場合に、BSM を使用して戻ることができますが、24 ビット・アドレッシング・モードで BAL を使用してプログラムが呼び出された場合は、BSM を使用して戻ることはできません。

BALR および BASR のその他の例

R₂ を 0 に設定した BALR または BASR 命令を使用することにより、基底レジスターとして使用するレジスターをロードすることができます。例えば、アセンブラー言語で次の 2 つのステートメントを書いたとします。

```
BALR 15,0
USING *,15
```

または

```
BASR 15,0
USING *,15
```

上記のステートメントは、BALR または BASR 命令の次の順番に当たる命令がレジスター 15 に入れられること、および、特に別の指示がない限り、アセンブラーはレジスター 15 を基底レジスターとして使用できることを示しています。(USING は「アセンブラー命令」の 1 つであり、したがってオブジェクト・プログラムの一部ではありません。)

BRANCH AND STACK (BAKR)

準特権命令 BRANCH AND STACK は、リンケージ・スタック状態エントリー (プログラム呼び出し状態エントリーと区別するためにブランチ状態エントリーと呼ばれることもあります) に状況を保管することにより、サブルーチン間のリンケージを容易にします。BRANCH AND STACK が使用された場合は、呼び出し先プログラムからの戻りは、PROGRAM RETURN 命令を使用して行われます。PROGRAM RETURN は、状態エントリーに保管されている値を使用して、アクセス・レジスター 2-14、汎用レジスター 2-14、および PSW を復元します。ただし、PER マスクは変更されず、条件コードがどの値に設定されるかは予測不能です。

BRANCH AND STACK の使用方法については、5-62 ページの『リンケージ・スタックを使用したブランチ』で説明されています。

BRANCH AND STACK は、この命令の R₁ フィールドが 0 か 0 以外かに応じて、それぞれ、呼び出しリンケージを行うために使用すること、または、呼び出し先プログラムのエントリー・ポイントまたはその近くで使用することができます。R₁ フィールドが 0 の場合は、状態エントリーに保管されている PSW のビット 32-63 は、現在のアドレッシング・モード (24 ビットまたは 31 ビット) と、BRANCH AND STACK 命令または EXECUTE 命令の次の順番に当たる命令のアドレスを示しています。R₁ フィールドが 0 以外である場合は、状態エントリーに保管されている PSW のビット 32-63 は、汎用レジスター R₁ の内容に基づいて生成された値に設定されます。つまり、PSW のビット 32 はレジスターのビット 0 の値に設定され、PSW のビット 1-31 は、レジスターのビット 0 の設定に従ってレジスターのビット 1-31 から生成されたアドレスに設定されます。以下の例では、状態エントリーに保管されている PSW のビット 32-63 を戻り値と呼んでいます。

この命令で使用するブランチ・アドレスは、現在のアドレッシング・モードの制御下で、汎用レジスター R₂ の内容から生成されます。汎用レジスター R₂ のビット 0

は、演算に影響を与えません。この命令の R₂ フィールドが 0 である場合は、ブランチなしで演算が行われます。

BRANCH AND STACK は、完全な PSW (予測不能の PER マスクを除く) を状態エントリーに保管するほかに、現 PSW のビット 32-63 の新しい値も状態エントリーに保管します。以下の例では、ビット 32-63 をブランチ値と呼んでいます。

以下の例は、BRANCH AND STACK の実行前の現 PSW のビット 32 が 0 (24 ビット・アドレッシング) または 1 (31 ビット・アドレッシング) の場合、および、0 以外の R₁ または R₂ フィールドに指定されている汎用レジスターのビット 0 が 0 または 1 の場合を示しています。

BAKR の例 1

この例は、呼び出し側プログラムで使用される BAKR を示しています。BAKR はブランチを行い、戻り先は次の順番に当たる命令です。

この BAKR 命令のフォーマットは以下のとおりです。

マシン・フォーマット
命令コード

	R ₁	R ₂
B240	0	6

アセンブラー・フォーマット

命令コード	R ₁ , R ₂
BAKR	0, 6

次の 4 通りの初期値があるものとします。

PSW (32-63) レジスター 6

- 00 00 10 D6 02 46 8A CE
- 00 00 10 D6 82 46 8A CE
- 80 00 10 D6 02 46 8A CE
- 80 00 10 D6 82 46 8A CE

この 4 つの場合の結果は以下のとおりです。

戻り値 ブランチ値および PSW (32-63)

- 00 00 10 DA 00 46 8A CE
- 00 00 10 DA 00 46 8A CE
- 80 00 10 DA 82 46 8A CE
- 80 00 10 DA 82 46 8A CE

BAKR の例 2

この例は、呼び出し先プログラムで使用される BAKR を示しています。BAKR はブランチを行わず、汎用レジスター R₁ に指定されているとおりに戻りが行われま

す。

この BAKR 命令のフォーマットは以下のとおりです。

マシン・フォーマット
命令コード

	R ₁	R ₂
B240	5	0

アセンブラー・フォーマット
命令コード R₁,R₂

BAKR 5,0

次の 4 通りの初期値があるものとします。

レジスター 5 PSW (32-63)
5

1.	04 00 10 D6	00 46 8A CE
2.	04 00 10 D6	82 46 8A CE
3.	84 00 10 D6	00 46 8A CE
4.	84 00 10 D6	82 46 8A CE

この 4 つの場合の結果は以下のとおりです。

戻り値 ブランチ値および
 PSW (32-63)

1.	00 00 10 D6	00 46 8A D2
2.	00 00 10 D6	82 46 8A D2
3.	84 00 10 D6	00 46 8A D2
4.	84 00 10 D6	82 46 8A D2

BAKR の例 3

この例は、呼び出し先プログラムで使用される BAKR を示しています。BAKR はブランチを行い、汎用レジスター R₁ に指定されているとおりに戻りが行われま

す。

この BAKR 命令のフォーマットは以下のとおりです。

マシン・フォーマット
命令コード

	R ₁	R ₂
B240	5	6

アセンブラー・フォーマット
命令コード R₁,R₂

BAKR 5,6

次の 8 通りの初期値があるものとします。

	レジスター 5	レジスター 6	PSW (32-63)
1.	04 00 10 D6	06 99 99 00	00 46 8A CE
2.	04 00 10 D6	06 99 99 00	82 46 8A CE
3.	04 00 10 D6	86 99 99 00	00 46 8A CE
4.	04 00 10 D6	86 99 99 00	82 46 8A CE
5.	84 00 10 D6	06 99 99 00	00 46 8A CE
6.	84 00 10 D6	06 99 99 00	82 46 8A CE
7.	84 00 10 D6	86 99 99 00	00 46 8A CE
8.	84 00 10 D6	86 99 99 00	82 46 8A CE

この 8 つの場合の結果は以下のとおりです。

戻り値 ブランチ値および
 PSW (32-63)

1.	00 00 10 D6	00 99 99 00
2.	00 00 10 D6	86 99 99 00
3.	00 00 10 D6	00 99 99 00
4.	00 00 10 D6	86 99 99 00
5.	84 00 10 D6	00 99 99 00
6.	84 00 10 D6	86 99 99 00
7.	84 00 10 D6	00 99 99 00
8.	84 00 10 D6	86 99 99 00

BRANCH ON CONDITION (BC, BCR)

BRANCH ON CONDITION は、条件コードをテストして、ブランチを行うかどうかを判断します。現在の条件コードが、この命令で指定されているマスク内の 1 のビットのいずれかに対応している場合のみ、ブランチが起きます。

条件コード	命令 (マスク) ビット	マスク値
0	8	8
1	9	4
2	10	2
3	11	1

例えば、ADD (A または AR) 演算が行われた結果、和が 0 またはそれより小さいとき (条件コードが 0 または 1 のとき) は、アドレス 6050 にブランチされるようにしたいとします。また、以下のように仮定します。

レジスター 10 の内容は、00 00 50 00。

レジスター 11 の内容は、00 00 10 00。

この命令を、次のように **RX** 形式で書くことにより、必要なテスト（および、必要に応じてブランチ）を行うことができます。

マシン・フォーマット

命令コード	M ₁	X ₂	B ₂	D ₂
47	C	B	A	050

アセンブラー・フォーマット

命令コード	M ₁ ,D ₂ (X ₂ ,B ₂)
BC	12,X'50'(11,10)

12₁₀ というマスクは、命令のビット 8 と 9 が 1 で、ビット 10 と 11 が 0 であり、したがって、条件コードが 0 または 1 であればブランチが行われることを意味しています。

マスク値が 15 であれば、どのような条件の場合でもブランチが行われることを示します（無条件ブランチ）。マスク値が 0 であれば、ブランチを行わないことを示します（ノーオペレーション）。

(BCR 命令の例については、A-8ページの『リンケージ命令 (BAL, BALR, BAS, BASR, BASSM, BSM)』も参照してください。)

BRANCH ON COUNT (BCT, BCTR)

BRANCH ON COUNT 命令は、プログラム・ループを指定の回数だけ実行する目的でよく使用されます。例えば、アセンブラー言語プログラムに次のようなコーディング行が含まれているとします。

```

:
LUPE AR 8,1
:
BACK BCT 6,LUPE
:

```

ここで、レジスター 6 の内容は 00 00 00 03 で、LUPE のアドレスは 6826 です。また、この記憶位置をアドレッシングするために、レジスター 10 が基底レジスターとして使用され、その内容は 00 00 68 00 であるものとします。

この BCT 命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
46	6	0	A	026

アセンブラー・フォーマット

命令コード	R ₁ ,D ₂ (X ₂ ,B ₂)
BCT	6,X'26'(0,10)

このコーディングの結果として、LUPE のラベルが付いた命令から BACK のラベルが付いた命令までに定義されているループが 3 回実行され、その間にレジスター 6 が 3 から 0 に減少します。

BRANCH ON INDEX HIGH (BXH)

BXH の例 1

BRANCH ON INDEX HIGH 命令は、指標値を増加させ、ループを制御する命令です。指標値と増分値の和が特定の比較値より大きくなると、ブランチが起きます。例えば次のように仮定します。

レジスター 4 の内容は、00 00 00 8A = 138₁₀ = 指標。

レジスター 6 の内容は、00 00 00 02 = 2₁₀ = 増分。

レジスター 7 の内容は、00 00 00 AA = 170₁₀ = 比較値。

レジスター 10 の内容は、00 00 71 30 = ブランチ・アドレス。

この BXH 命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令コード	R ₁	R ₃	B ₂	D ₂
86	4	6	A	000

アセンブラー・フォーマット

命令コード	R ₁ ,R ₃ ,D ₂ (B ₂)
BXH	4,6,0(10)

この命令を実行すると、最初にレジスター 6 の内容がレジスター 4 に加算され、2 番目にその和がレジスター 7 の内容と比較され、3 番目にブランチを行うかどうかが決まります。実行結果は以下のようになります。

レジスター 4 の内容は、00 00 00 8C = 140₁₀ になります。

レジスター 6 および 7 は変更されません。

レジスター 4 の新しい値は、まだレジスター 7 の値より大きくなっていないので、アドレス 7130 へのブランチは行われません。この命令を繰り返し使用すると、最後にはレジスター 4 の値が 17210 に達し、その時点でブランチが行われます。

BXH の例 2

増分値を入れるために使用されるレジスターが奇数番号である場合は、そのレジスターは比較値レジスターとしても使用されます。以下のアセンブラ言語サブルーチンは、この機能を使用してテーブルを検索する方法を示しています。

テーブル	
2 バイト	2 バイト
ARG1	FUNCT1
ARG2	FUNCT2
ARG3	FUNCT3
ARG4	FUNCT4
ARG5	FUNCT5
ARG6	FUNCT6

以下のように仮定します。

レジスター 8 には、検索引き数が入っています。

レジスター 9 には、テーブルの幅のバイト数が入っています (00 00 00 04)。

レジスター 10 には、テーブルの長さのバイト数が入っています (00 00 00 18)。

レジスター 11 には、テーブルの開始アドレスが入っています。

レジスター 14 には、メイン・プログラムへの戻りアドレスが入っています。

以下のサブルーチンを実行すると、レジスター 8 に入っている引き数が、テーブル内の引き数と次々に比較されます。比較は、引き数 6 から始めて、引き数 1 の方へ戻る方向に進められます。一致する引き数が見つかる、それに対応する機能がレジスター 8 の中の引き数と置き換わります。一致する引き数が見つからなかった場合は、レジスター 8 の引き数は 0 で置き換えられます。

```
SEARCH  LNR  9,9
NOTEQUAL BXH 10,9,LOOP
NOTFOUND SR  8,8
        BCR 15,14
LOOP    CH  8,0(10,11)
        BC  7,NOTEQUAL
        LH  8,2(10,11)
        BCR 15,14
```

最初の命令 (LNR) では、レジスター 9 の値が負にされます。この命令の実行後、レジスター 9 の内容は、FF FF FC = -410 となります。一致する引き数が見つからない場合を想定すると、BXH 命令は 7 回実行されることとなります。BXH が 1 回実行されるたびに、レジスター 10 に -4 の値が加算され、レジスター 10 の値は 4 ずつ減少します。そして、レジスター 10 の新しい値がレジスター 9 の中の -4 の値と比較されます。レジスター 10 の値が -4 になるまで、毎回ブランチが行われます。-4 に達すると、ブランチは行われず、SR 命令によりレジスター 8 が 0 に設定されます。

BRANCH ON INDEX LOW OR EQUAL (BXLE)

BRANCH ON INDEX LOW OR EQUAL 命令は、BRANCH ON INDEX HIGH に似ていますが、和が比較値より大きいときではなく、等しいかまたは小さいときに、ブランチが行われるという点が異なります。どちらの命令も、プログラム・ループ内の指標値を増加させてテストするものですが、BXLE はループの終わりで使用し、BXH はループの始めで使用するのが効果的です。次のアセンブラ言語ルーチンは、BXLE を伴うループを示しています。

BXLE の例 1

10 個の 32 ビット符号付き 2 進整数のグループが、記憶位置 GROUP を先頭とする連続した記憶位置に格納されているとします。これらの整数を加算して、その和を記憶位置 SUM に格納します。

```
SR  5,5      和を 0 に設定する
LA  6,GROUP  最初のアドレスを
              ロードする
SR  7,7      指標を 0 に設定する
LA  8,4      増分値 4 をロードする
              する
LA  9,39     比較値をロードする
LOOP A  5,0(7,6) 整数を和に加算する
BXLE 7,8,LOOP ループの終わりかど
              うかをテストする
ST  5,SUM    和を格納する
```

上記の 2 つの命令から成るループには、各整数を汎用レジスター 5 の内容に加算する ADD (A) 命令が含まれています。この ADD 命令は、汎用レジスター 7 の内容を指標値として使用して、レジスター 6 から取得した開始アドレスを変更します。次に、BXLE は、指標値を 4 (前もってレジスター 8 にロードされている増分値) だけ増加させ、その値を、レジスター 9 (奇数/偶数の対を成すレジスターのうちの奇数番号の方) に入っている比較値と比較します。比較値は、前もって、39 (データ域のバイト数より 1 だけ小さい値) に設定されています。この値は、開始アドレスを基準とした、追加する最後の整数の右端バイトの相対アドレスでもあります。最後の整数が加算されると、BXLE は、指標値を次の相対アドレス (40) に増加させます。40 は比較値 (39) より大きいので、ブランチは行われません。

BXLE の例 2

例 1 で示した手法は、RX 命令フォーマットの命令を含むループに限定されます。RX フォーマットでは、基底レジスターと指標レジスターの両方を指定できます (二重指標付け)。

指標レジスターを指定できない他のフォーマットの命令の場合は、前記の手法を修正して利用することができません。つまり、アドレス自体が BXLE 命令の中で指標値として働くようにし、最後のバイトのアドレス (相対アドレスではなく) を比較値として使用するようになります。このようにすれば、ループの反復のたびに、基底レジスターからアドレスが直接供給されるので、指標値を保持するための第 2 のレジスターを指定する必要はありません (単一指標付け)。

以下の例では、SI 命令フォーマットの AND (NI) 命令は、例 1 の場合と同じグループに含まれる各整数の右端ビットを 0 に設定するので、これらの整数はすべて偶数になります。NI 命令の I₂ フィールドには、7 個の 1 と 1 個の 0 から成る X'FE' のバイトが含まれています。AND 演算により、このバイトと各整数のバイト 3 (右端バイト) との論理積が、順に求められます。

```

LA 6, GROUP    最初のアドレスを
                  ロードする
LA 8, 4        増分値 4 をロード
                  する
LA 9, GROUP+39 比較値をロードする
LOOP NI 3(6), X'FE'  即値 AND
BXLE 6, 8, LOOP  ループの終わりかど
                  うかをテストする

```

ただし、ESA/370 システムでは、31 ビット・アドレッシング・モードが使用されていて、データが 31 ビッ

ト・アドレス・スペースの最右端に入れられている場合は、例 2 に示した手法は使用できません。その場合は、比較値は、 $2^{31}-1$ 、つまり可能な最大の 32 ビット符号付き 2 進数値に設定されます。この手法が無効なのは、BXLE および BXH 命令では、それぞれのオペランドが 32 ビット符号付き 2 進整数として扱われるからです。汎用レジスター 6 中のアドレスが $2^{31}-4$ の値に達すると、BXLE は、その値を、 -2^{31} として解釈される値 (2^{31} として解釈される値ではなく) に増加させます。したがって、比較の結果は「より小」のままなので、無限にループが続くことになります。

この事態を回避するには、データ域が 31 ビット・アドレス・スペースの最右端の位置まで拡大しないようにするか、または別の手法を使用します。例えば、可能であれば例 1 に示した二重指標付けの手法を用いるか、あるいは、データ域の終わりから開始し、負の増分値を使用して下方へと操作を進めます。

COMPARE AND FORM CODEWORD (CFC)

A-51 ページの『ソート命令』を参照してください。

COMPARE HALFWORD (CH)

COMPARE HALFWORD 命令は、記憶域内の 16 ビット符号付き 2 進整数を特定のレジスターの内容と比較します。例えば次のように仮定します。

レジスター 4 の内容は、FF FF 80 00 =
-32,768₁₀。

レジスター 13 の内容は、00 01 60 50。

記憶位置 16080-16081 の内容は、8000 =
-32,768₁₀。

次の命令を実行します。

マシン・フォーマット
命令コード R₁ X₂ B₂ D₂

49	4	0	D	030
----	---	---	---	-----

アセンブラー・フォーマット
命令コード R₁, D₂(X₂, B₂)

```

CH      4, X'30' (0, 13)

```

記憶位置 16080-16081 の内容が取り出され、32 ビットに拡張されて (符号ビットが左へ波及)、レジスター 4

の内容と比較されます。2つの数値は等しいので、条件コード 0 が設定されます。

COMPARE LOGICAL (CL, CLC, CLI, CLR)

COMPARE LOGICAL 命令は、すべての数量を符号なし 2 進整数または非構造化データとして扱うという点で、符号付き 2 進比較命令 (C, CH, CR) と異なります。

CLC の例

COMPARE LOGICAL (CLC) 命令を使用すると、最大 256 バイトの長さの記憶域フィールドをバイト単位で比較することができます。例えば、記憶域内に次の 2 つのデータ・フィールドがあるとします。

フィールド 1
1886 1891

D1	D6	C8	D5	E2	D6	D5	6B	C1	4B	C2	4B
----	----	----	----	----	----	----	----	----	----	----	----

フィールド 2
1900 190B

D1	D6	C8	D5	E2	D6	D5	6B	C1	4B	C3	4B
----	----	----	----	----	----	----	----	----	----	----	----

また、以下のように仮定します。

レジスター 9 の内容は、00 00 18 80。

レジスター 7 の内容は、00 00 19 00。

次の命令を実行します。

マシン・フォーマット
命令コード L B₁ D₁ B₂ D₂

D5	0B	9	006	7	000
----	----	---	-----	---	-----

アセンブラー・フォーマット
命令コード D₁(L,B₁),D₂(B₂)

CLC 6(12,9),0(7)

実行の結果、条件コード 1 が設定されます。これは、フィールド 1 の内容がフィールド 2 の内容より小さい値であることを示します。

EBCDIC コードの照合シーケンスはコード内のビットの論理比較によってのみ決定されるので、CLC 命令を

使用して EBCDIC コードのフィールドを照合することができます。例えば、上記の 2 つのフィールドは EBCDIC では以下ようになります。

フィールド 1: JOHNSON,A.B.
フィールド 2: JOHNSON,A.C.

条件コード 1 は、フィールドをアルファベット順に並べるには、JOHNSON,A.B. が JOHNSON,A.C. より前になければならないことを示します。

CLI の例

COMPARE LOGICAL (CLI) 命令は、命令ストリームからのバイトと記憶域からのバイトとを比較します。例えば次のように仮定します。

レジスター 10 の内容は、00 00 17 00。

記憶位置 1703 の内容は、7E。

次の命令を実行します。

マシン・フォーマット
命令コード I₂ B₁ D₁

95	AF	A	003
----	----	---	-----

アセンブラー・フォーマット
命令コード D₁(B₁),I₂

CLI 3(10),X'AF'

この実行により、条件コード 1 が設定されます。これは、第 1 オペランド (主記憶機構内の数量値) が、第 2 オペランド (即値) より小さいことを示します。

CLR の例

以下のように仮定します。

レジスター 4 の内容は、00 00 00 01 = 1。

レジスター 7 の内容は、FF FF FF FF = 2³² - 1。

次の命令を実行します。

マシン・フォーマット
命令コード R₁ R₂

15	4	7
----	---	---

アセンブラー・フォーマット
命令コード R₁,R₂

CLR 4,7

この実行により、条件コード 1 が設定されます。条件コード 1 は、第 1 オペランドが第 2 オペランドより小さいことを意味します。

この命令の代わりに、符号付き 2 進数比較命令 COMPARE (CR) を実行したとすれば、レジスタ 4 の内容は +1 と解釈され、レジスタ 7 の内容は -1 と解釈されます。したがって、第 1 オペランドの方が大きくなるため、条件コード 2 が設定されていたはずですが。

COMPARE LOGICAL CHARACTERS UNDER MASK (CLM)

COMPARE LOGICAL CHARACTERS UNDER MASK (CLM) 命令は、汎用レジスタから選択したバイトを、記憶機構内の連続したバイトから成るフィールドと比較するために使用できます。CLM 命令の M₃ フィールドは、0 個から 4 個までのバイトを汎用レジスタから選択する 4 ビットのマスクで、各マスク・ビットは、左から右へ、それぞれ 1 つのレジスタ・バイトに対応しています。比較では、マスク内の 1 に対応しているレジスタ・バイトの集合が 1 つの連続フィールドとして扱われます。操作は左から右へ進められます。例えば次のように仮定します。

記憶位置 10200-10202 の内容は、F0 BC 7B。

レジスタ 12 の内容は、00 01 00 00。

レジスタ 6 の内容は、F0 BC 5C 7B。

次の命令を実行します。

マシン・フォーマット

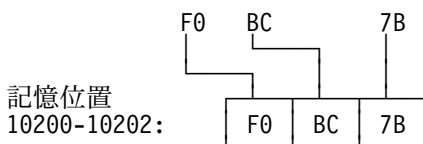
命令コード	R ₁	M ₃	B ₂	D ₂
BD	6	D	C	200

アセンブラー・フォーマット

命令コード	R ₁ , M ₃ , D ₂ (B ₂)
CLM	6, B'1101', X'200' (12)

この実行により、以下の比較が行われます。

レジスタ 6:	F0	BC	5C	7B
マスク M ₃ :	1	1	0	1
	--	--		--

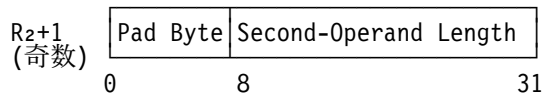
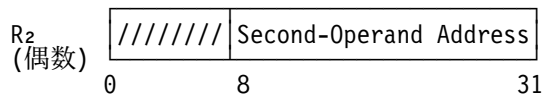
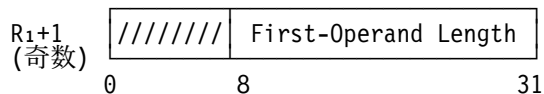
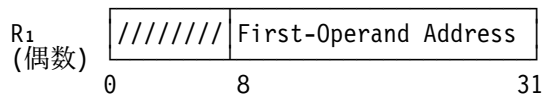


選択したバイトは等しいので、条件コード 0 が設定されます。

COMPARE LOGICAL LONG (CLCL)

COMPARE LOGICAL LONG 命令は、記憶機構内の 2 つのオペランドをバイト単位で比較するために使用します。各オペランドはどのような長さでも構いません。オペランドを指示し、CLCL 命令の実行を制御するために、以下の図に示すように、汎用レジスタの奇偶ペアが 2 つ (合計 4 つのレジスタ) が使用されます。各ペアの最初のレジスタは偶数番号のレジスタでなければならない、このレジスタにはオペランドの記憶域アドレスが含まれています。各ペアの奇数番号のレジスタには、そのレジスタ・ペアが指しているオペランドの長さが含まれています。また、第 2 オペランドの奇数番号レジスタの左端 1 バイトには、埋め込みバイトが含まれています。このバイトは、短い方のオペランドを長い方のオペランドと同じ長さに拡張するために使用されます。

次の図は、24 ビット・アドレッシング・モードでのレジスタの割り当てを示しています。



31 ビット・アドレッシング・モードでは、オペランド・アドレスは、上記に示した偶数番号レジスタのビット位置 1-31 を占めます。

CLCL 命令は、実行中に割り込みを受けることがあるので、割り込み側のプログラムは、この命令の再開時に使用できるように、4 つのレジスタの内容を保存しておくなければなりません。

以下の命令は、テキスト・ストリングの比較を制御するための2つのレジスター・ペアをセットアップします。例えば次のように仮定します。

オペランド 1
アドレス: 20800₁₆
長さ: 100₁₀

オペランド 2
アドレス: 20A00₁₆
長さ: 132₁₀

埋め込みバイト
アドレス: 20003₁₆
長さ: 1
値: 40₁₆

レジスター 12 の内容は、00 02 00 00。

セットアップのための命令は以下のとおりです。

LA	4,X'800'(12)	レジスター 4 で第 1 オペランドの起点を設定します。
LA	5,100	レジスター 5 で第 1 オペランドの長さを設定します。
LA	8,X'A00'(12)	レジスター 8 で第 2 オペランドの起点を設定します。
LA	9,132	レジスター 9 で第 2 オペランドの長さを設定します。
ICM	9,B'1000',3(12)	レジスター 9 の左端バイト位置に埋め込みバイトを挿入します。

レジスター・ペア 4,5 は第 1 オペランドを定義します。レジスター 4 のビット 8-31 には、EBCDIC テキスト・ストリーミングの起点の記憶域アドレスが含まれており、レジスター 5 のビット 8-31 には、ストリングの長さ (この例では 100 バイト) が含まれています。

レジスター・ペア 8,9 は第 2 オペランドを定義します。レジスター 8 のビット 8-31 には、第 2 オペランドの開始位置が含まれており、レジスター 9 のビット 8-31 には、第 2 オペランドの長さ (この例では 132 バイト) が含まれています。レジスター 9 のビット 0-7 には、短い方のオペランドに埋め込むための EBCDIC ブランク文字 (X'40') が含まれています。この例では、第 1 オペランドの 100 番目のバイトの後に埋め込みバイトが使用され、第 2 オペランドの余ったバイトと比較されます。

上記のようにレジスター・ペアをセットアップした場合、CLCL 命令のフォーマットは以下のようになります。

マシン・フォーマット
命令コード R₁ R₂

0F	4	8
----	---	---

アセンブラー・フォーマット
命令コード R₁,R₂

CLCL 4,8

この命令を実行すると、各オペランドの左端から比較が開始され、右方向へ進められます。不一致が検出されるか、または長い方のオペランドの終わりに達した時点で、操作は終了します。

この CLCL 命令で、60 バイトが等しいことが確認された後で割り込みが起きたとすれば、レジスター 5 および 9 の中のオペランド長は、それぞれ 40 および 72 に減少しています。また、レジスター 4 および 8 の中のオペランド・アドレスは、それぞれ、X'2083C' および X'20A3C' に増加しています。レジスター 4 および 8 の左端バイトは、0 に設定されています。レジスター 9 の中の埋め込みバイト X'40' は、そのまま残されています。レジスターの内容がこのように設定されている状態で CLCL 命令が再実行されると、割り込みが起きた位置から比較が再開されます。

次に、110 バイトを過ぎた時点でこの命令が割り込まれたとします。これは、第 2 オペランドの最初の 100 バイトが第 1 オペランドと等しいものとして確認され、第 2 オペランドのその後の 10 バイトが埋め込みバイト (ブランク) と比較され、これも等しいものと確認されたことを意味します。このとき、レジスター 5 および 9 の中の残りのオペランド長は、それぞれ 0 と 22 であり、レジスター 4 および 8 の中のオペランド・アドレスは、それぞれ、X'20864' (第 1 オペランドの長さの終わりに達した時点の値) および X'20A6E' (第 2 オペランドの現在値) です。

比較が終了すると、条件コードが設定されます。条件コードは、第 1 オペランドが第 2 オペランドに等しければ 0、第 1 オペランドの方が小さければ 1、そして第 1 オペランドの方が大きければ 2 です。

2 つのオペランドが等しくないときは、レジスター 4 および 8 の中のアドレスは、どのバイトが一致していないかを示しています。

COMPARE LOGICAL STRING (CLST)

COMPARE LOGICAL STRING 命令は、汎用レジスタ R₁ に指定する第 1 オペランドと、汎用レジスタ R₂ に指定する第 2 オペランドと比較するために使用します。比較は、左から右へ 1 バイト単位で行われます。等しくないバイトが検出されるか、いずれかのオペランドの中で、汎用レジスタ 0 に指定されている終了文字が見つかるか、または CPU で決定されたバイト数の比較が完了した時点で、操作は終了します。条件コードは、両オペランドが等しければ 0、第 1 オペランドの方が小さければ 1、第 2 オペランドの方が小さければ 2、そして、CPU 決定のバイト数の比較が完了した場合は 3 に設定されます。両方のオペランド内で同時に終了文字が検出された場合は、両オペランドは等しいものと見なされます。いずれか一方のオペランド内で終了文字が検出された場合は、そのオペランドの方が小さいものと見なされます。

条件コード 1 または 2 が設定された場合は、第 1 オペランドおよび第 2 オペランド内で処理された最後のバイトのアドレスが、それぞれ、汎用レジスタ R₁ および R₂ に入れます。これらのアドレスは、2 つのオペランド内の等しくないバイトのアドレスか、または、一方のオペランド内の終了文字のアドレスと、もう一方のオペランド内のそれに対応するバイト位置にあるバイトのアドレスです。条件コード 3 が設定された場合は、次に処理されることになっていたバイトのアドレスがレジスタに入れます。条件コード 0 が設定された場合は、レジスタの内容は変更されません。

以下に、第 1 オペランドが 10 進記憶位置 1000 から始まり、第 2 オペランドが 10 進記憶位置 2000 から始まっている場合の例を、幾つか示します。汎用レジスタ R₁ および R₂ に含まれているアドレスは、それぞれ 1000 および 2000 です。汎用レジスタ 0 に含まれている終了文字は、16 進数 00 です (これは C プログラミング言語で使用される値です)。オペランドのバイトの値は 16 進数で示してあります。また、結果の条件コードと、汎用レジスタ R₁ および R₂ の最終的な内容も示します。

例 1
1000 2000
C1 C2 C3 00 C1 C2 C3 00
CC: 0; (R₁): 1000; (R₂): 2000

例 2
1000 2000
40 40 40 C1 40 40 40 C2
CC: 1; (R₁): 1003; (R₂): 2003

例 3
1000 2000
40 40 40 C2 40 40 40 C1
CC: 2; (R₁): 1003; (R₂): 2003

例 4
1000 2000
C1 C2 C3 00 C1 C2 C3 C4
CC: 1; (R₁): 1003; (R₂): 2003

例 5
1000 2000
C1 C2 C3 C4 C1 C2 C3 00
CC: 2; (R₁): 1003; (R₂): 2003

例 6
CPU 決定の比較対象バイト数が 256 であるものとします。

1000 1256 2000 2256
40 .. 40 00 40 .. 40 00
CC: 3; (R₁): 1256; (R₂): 2256

例 7
1000 2000
00 40 40 40 40 40 40 40
CC: 1; (R₁): 1000; (R₂): 2000

例 8
1000 2000
40 40 40 40 00 40 40 40
CC: 2; (R₁): 1000; (R₂): 2000

例 9
1000 2000
00 40 40 40 00 40 40 40
CC: 0; (R₁): 1000; (R₂): 2000

CONVERT TO BINARY (CVB)

CONVERT TO BINARY 命令は、8 バイトのパック 10 進数を符号付き 2 進整数に変換し、その結果を汎用レジスタにロードします。変換操作が完了した後の数値は、符号付き 2 進数演算でオペランドとして使用できる適切な形式になっています。例えば次のように仮定します。

記憶位置 7608-760F の内容は、パック・フォーマットの 10 進数 00 00 00 00 0025 59 4C (+25,594)。

レジスタ 7 の内容は意味を持たない。

レジスタ 13 の内容は、00 00 76 00。

この場合の変換命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
4F	7	0	D	008

アセンブラー・フォーマット

命令コード	R ₁ ,D ₂ (X ₂ ,B ₂)
CVB	7,8(0,13)

この命令の実行後、レジスター 7 には 00 00 63 FA が入ります。

CONVERT TO DECIMAL (CVD)

CONVERT TO DECIMAL 命令は、CONVERT TO BINARY 命令と逆の働きをします。つまり、CVD は、レジスター内の符号付き 2 進整数をパック 10 進数に変換して、8 バイトの結果を格納します。例えば次のように仮定します。

レジスター 1 の内容は、符号付き 2 進整数 00 00 0F 0F。

レジスター 13 の内容は、00 00 76 00。

この場合の命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
4E	1	0	D	008

アセンブラー・フォーマット
命令コード R₁,D₂(X₂,B₂)

CVD	1,8(0,13)
-----	-----------

この命令の実行後、記憶位置 7608-760F には 00 00 00 00 03 85 5C (+3855) が入ります。

生成される正符号は、優先正符号である 1100₂ です。

DIVIDE (D, DR)

DIVIDE 命令は、レジスターの奇偶ペア内の被除数を、レジスターまたは記憶域内にある除数で除算します。この命令では、被除数の長さは 64 ビットであることが前提となっているので、まず、32 ビットの被除数を、符号ビットと同じビットで左側に拡張しておく必要があります。例えば次のように仮定します。

記憶位置 3550-3553 の内容は、00 00 08 DE = 2270₁₀ (被除数)。

記憶位置 3554-3557 の内容は、00 00 00 32 = 50₁₀ (除数)。

レジスター 6 および 7 の初期の内容は意味を持たない。

レジスター 8 の内容は、00 00 35 50。

以下のアセンブラー言語ステートメントは、レジスターを適切にロードし、除算操作を行います。

ステートメント	説明
L 6,0(0,8)	レジスター 6 に 00 00 08 DE を入れる。
SRDA 6,32(0)	00 00 08 DE をレジスター 7 にシフトする。レジスター 6 は 0 (符号ビット) で満たされる。
D 6,4(0,8)	除算を行う。

上記の DIVIDE 命令のマシン・フォーマットは、以下のとおりです。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
5D	6	0	8	004

上記のすべての命令が実行されると、結果は以下のようになります。

レジスター 6 の内容は、00 00 00 14 = 20₁₀ = 剰余。

レジスター 7 の内容は、00 00 00 2D = 45₁₀ = 商。

最初に被除数をレジスター 6 に入れ、レジスター 7 にシフトしておかなかったとすれば、レジスター 7 は被除数の正しい符号ビット (この例では 0) で満たされないため、DIVIDE 命令から意図したとおりの結果が得られないことがあります。

EXCLUSIVE OR (X, XC, XI, XR)

2 個のビットにブール演算子 EXCLUSIVE OR を適用すると、いずれか一方のビット (両方ではなく) が 1 であれば、結果は 1 になります。そうでない場合は、結果は 0 です。2 個のバイトに対して EXCLUSIVE OR 演算を行うと、対応するビットが一対ずつ個別に処理されます。1 つのビット位置と他のビット位置との間には関連性はありません。以下の例は、2 個のバイトに対する EXCLUSIVE OR 演算の結果を示しています。

第 1 オペランドのバイト: 0011 0101₂
 第 2 オペランドのバイト: 0101 1100₂
 結果のバイト: 0110 1001₂

XC の例

EXCLUSIVE OR (XC) 命令を使用すると、記憶機構内の 2 つの区域の内容を、中間記憶域を使用せずに交換することができます。例えば、記憶域内に次のような 2 つの 3 バイト・フィールドがあるとします。

359 35B
 フィールド 1

00	17	90
----	----	----

 360 362
 フィールド 2

00	14	01
----	----	----

以下の命令を実行します (レジスタ 7 の内容は 00 00 03 58 であるとして)。

マシン・フォーマット
 命令コード L B₁ D₁ B₂ D₂

D7	02	7	001	7	008
----	----	---	-----	---	-----

アセンブラー・フォーマット
 命令コード D₁(L,B₁),D₂(B₂)

XC 1(3,7),8(7)

EXCLUSIVE OR により、以下のように、フィールド 1 とフィールド 2 の排他的論理和が求められます。

フィールド 1: 00000000 00010111 10010000₂ = 00 17 90₁₆
 フィールド 2: 00000000 00010100 00000001₂ = 00 14 01₁₆
 結果: 00000000 00000011 10010001₂ = 00 03 91₁₆

結果は、フィールド 1 の前の内容と置き換わります。結果が 0 でないことを示す条件コード 1 が設定されます。

今度は、次の命令を実行します。

マシン・フォーマット
 命令コード L B₁ D₁ B₂ D₂

D7	02	7	008	7	001
----	----	---	-----	---	-----

アセンブラー・フォーマット
 命令コード D₁(L,B₁),D₂(B₂)

XC 8(3,7),1(7)

結果は以下のようになります。

フィールド 1: 00000000 00000011 10010001₂ = 00 03 91₁₆
 フィールド 2: 00000000 00010100 00000001₂ = 00 14 01₁₆
 結果: 00000000 00010111 10010000₂ = 00 17 90₁₆

この演算の結果は、フィールド 2 の前の内容と置き換わります。つまり、フィールド 2 にはフィールド 1 の元の値が入っています。結果が 0 でないことを示す条件コード 1 が設定されます。

最後に、次の命令を実行します。

マシン・フォーマット
 命令コード L B₁ D₁ B₂ D₂

D7	02	7	001	7	008
----	----	---	-----	---	-----

アセンブラー・フォーマット
 命令コード D₁(L,B₁),D₂(B₂)

XC 1(3,7),8(7)

結果は以下のようになります。

フィールド 1: 00000000 00000011 10010001₂ = 00 03 91₁₆
 フィールド 2: 00000000 00010111 10010000₂ = 00 17 90₁₆
 結果: 00000000 00010100 00000001₂ = 00 14 01₁₆

この演算の結果は、フィールド 1 の前の内容と置き換わります。つまり、フィールド 1 にはフィールド 2 の元の値が入っています。結果が 0 でないことを示す条件コード 1 が設定されます。

XI の例

EXCLUSIVE OR (XI) は、ビットを反転させる (0 のビットを 1 に、または 1 のビットを 0 にする) ために、よく使用されます。例えば、記憶位置 8082 に 0110 1001₂ が含まれているとします。この記憶位置の左端と右端のビットのみを反転させ、他のビットは変わらないようにするには、次のような命令を使用できます (レジスタ 9 の内容は 00 00 80 80 であるものとします)。

マシン・フォーマット
 命令コード I₂ B₁ D₁

97	81	9	002
----	----	---	-----

アセンブラー・フォーマット
 命令コード D₁(B₁),I₂

XI 2(9),X'81'

この命令が実行されると、記憶域内のバイトと即値バイト (命令の I₂ フィールド) との EXCLUSIVE OR 演算が行われます。

記憶位置 8082: 0110 1001₂
即値バイト: 1000 0001₂

結果: 1110 1000₂

結果のバイトは、記憶位置 8082 に戻されて格納されま
す。結果が 0 でないことを示す条件コード 1 が設定さ
れます。

注:

1. XC 命令では、最大 256 バイトの長さのフィールドを交換できます。
2. XR 命令では、2 個のレジスタの内容を交換できます。
3. X 命令は、記憶域からレジスタへの操作を行うだけなので、X 命令単独で交換を行うことはできません。
4. あるフィールドとそれ自体との間で EXCLUSIVE OR 演算を行うと、そのフィールドは消去されて 0 になります。
5. EXCLUSIVE OR の他の使用例については、A-42 ページの『16 進浮動小数点数の変換』を参照してください。

EXECUTE (EX)

EXECUTE 命令は、主記憶域内にある 1 つのターゲット命令を、実際にそこへブランチすることなく番外として実行させます。EXECUTE 命令の R₁ フィールドが 0 の場合を除き、ターゲット命令を実行する前に、ターゲット命令のビット 8-15 と R₁ レジスタのビット 24-31 との間で OR 演算が行われます。したがって、EXECUTE 命令を使用することにより、記憶域内の SS 命令を変更せずに、その SS 命令用の長さフィールドを供給することができます。例えば、ターゲット命令が、アドレス 3820 にある以下のフォーマットの MOVE (MVC) 命令であるとしてま

マシン・フォーマット
命令コード L B₁ D₁ B₂ D₂

D2	00	C	003	D	000
----	----	---	-----	---	-----

アセンブラー・フォーマット
命令コード D₁(L,B₁),D₂(B₂)

MVC 3(1,12),0(13)

ここで、レジスタ 12 の内容は 00 00 89 13 で、レジスタ 13 の内容は 00 00 90 A0 であるものとします。

さらに、記憶域アドレス 5000 に次のような EXECUTE 命令があるとします。

マシン・フォーマット
命令コード R₁ X₂ B₂ D₂

44	1	0	A	000
----	---	---	---	-----

アセンブラー・フォーマット
命令コード R₁,D₂(X₂,B₂)

EX 1,0(0,10)

ここで、レジスタ 10 の内容は 00 00 38 20 で、レジスタ 1 の内容は 00 0F F0 03 であるものとします。

アドレス 5000 にある命令が実行されると、レジスタ 1 の右端バイトとターゲット命令の第 2 バイトの間で OR 演算が行われます。

命令のバイト: 0000 0000₂ = 00
レジスタのバイト: 0000 0011₂ = 03

結果: 0000 0011₂ = 03

これにより、アドレス 3820 にある命令が、最初から以下のようなものであったかのように実行されます。

マシン・フォーマット
命令コード L B₁ D₁ B₂ D₂

D2	03	C	003	D	000
----	----	---	-----	---	-----

アセンブラー・フォーマット
命令コード D₁(L,B₁),D₂(B₂)

MVC 3(4,12),0(13)

しかし、実行後の結果は以下のようになります。

レジスタ 1 は変更されません。

アドレス 3820 の命令は変更されません。

記憶位置 90A0 から始まる 4 バイトの内容が、記憶位置 8916 から始まる 4 バイトに移されています。

CPU は、次に、アドレス 5004 にある命令を実行します (PSW のビット 40-63 に 00 50 04 が入っている)。

INSERT CHARACTERS UNDER MASK (ICM)

INSERT CHARACTERS UNDER MASK (ICM) 命令を使用すると、汎用レジスタ内のすべてまたは選択したバイトを記憶域内のバイトで置き換え、挿入されたフィールドの値を示す条件コードを設定することができます。

例えば、FIELDA にある 3 バイト・アドレスをレジスタ 5 に挿入し、このレジスタの左端バイトは無変更のままにしたいとすれば、次のような命令を実行します。

マシン・フォーマット
命令コード R₁ M₃ S₂

BF	5	7	* * * *
----	---	---	---------

アセンブラー・フォーマット
命令コード R₁,M₃,S₂

ICM 5,B'0111',FIELDA

FIELDA:
レジスタ 5 (実行前): FE DC BA
レジスタ 5 (実行後): 12 34 56 78
条件コード (実行後): 1 (挿入されたフィールドの左端ビットが 1)

もう 1 つ例を示します。

マシン・フォーマット
命令コード R₁ M₃ S₂

BF	6	9	* * * *
----	---	---	---------

アセンブラー・フォーマット
命令コード R₁,M₃,S₂

ICM 6,B'1001',FIELDB

FIELDB:
レジスタ 6 (実行前): 12 34
 00 00 00 00
レジスタ 6 (実行後): 12 00 00 34
条件コード (実行後): 2 (挿入されたフィールドが 0 でなく、左端のビットが 0)

マスク・フィールドに 1111 が含まれているときは、ICM 命令の結果は、LOAD (L) の場合と同じになります (RX フォーマットの指標付け機能が不要な場合)。ただし、ICM では条件コードも設定される点が異なります。すべて 0 のフィールド (条件コード) または左端の 1 のビット (条件コード 1) をフラグとして使用するとき、条件コードが設定される方が便利です。

LOAD (L, LR)

LOAD 命令は、記憶域または汎用レジスタから 4 バイトをとり、それを変更せずに汎用レジスタに入れます。例えば、記憶位置 21003 から始まる 4 バイトをレジスタ 10 にロードしたいとします。それぞれの初期値は以下のとおりであるとします。

レジスタ 5 の内容は、00 02 00 00。

レジスタ 6 の内容は、00 00 10 03。

レジスタ 10 の内容は意味を持たない。

記憶位置 21003-21006 の内容は、00 00 AB CD。

レジスタ 10 にロードするには、次のような RX 形式の命令を使用できます。

マシン・フォーマット
命令コード R₁ X₂ B₂ D₂

58	A	5	6	000
----	---	---	---	-----

アセンブラー・フォーマット
命令コード R₁,D₂(X₂,B₂)

L 10,0(5,6)

この命令の実行後、レジスタ 10 には 00 00 AB CD が入ります。

LOAD ADDRESS (LA)

LOAD ADDRESS 命令は、最初に定数を定義してからそれをオペランドとして使用するという手間を省いて、最大 4095₁₀ までの負でない 2 進整数をレジスターに入れるための便利な手段です。例えば、以下の命令は、2048₁₀ という数値をレジスター 1 に入れます。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
41	1	0	0	800

アセンブラー・フォーマット

命令コード R₁,D₂(X₂,B₂)

LA 1,2048(0,0)

また、LOAD ADDRESS 命令を使用して、レジスターの値を、最大 4095₁₀ までの範囲内で D₂ フィールドに指定する値だけ増加させることもできます。ただし、アドレッシング・モードによっては、合計の右端 24 ビットまたは 31 ビットのみが保持されます。32 ビットの結果の左端ビットは 0 に設定されます。例えば、レジスター 5 に 00 12 34 56 が入っているとします。

ここで次の命令を実行したとします。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
41	5	0	5	00A

アセンブラー・フォーマット

命令コード R₁,D₂(X₂,B₂)

LA 5,10(0,5)

この実行結果として、次のように、レジスター 5 の内容に 10 (10 進数) が加算されます。

レジスター 5 (旧): 00 12 34 56
D₂ フィールド: 00 00 00 0A

レジスター 5 (新): 00 12 34 60

レジスターは、B₂ または X₂ のどちらかに指定しても構いません。したがって、命令 LA 5,10(5,0) を使用しても結果は同じです。

最も一般的な例として、命令 LA 6,10(5,4) は、3 つの値 (レジスター 4 の内容、レジスター 5 の内容、および 10 の変位) を合計し、その結果の 24 ビットまたは 31 ビットの合計値の左側に必要数の 0 を付加して、レジスター 6 に入れます。

LOAD HALFWORD (LH)

LOAD HALFWORD は、記憶域から取り出したハーフワードを、変更せずにレジスターの右半分に入れます。レジスターの左半分には、そのハーフワードの符号 (左端ビット) に応じて、0 または 1 がロードされます。

例えば、記憶位置 1803-1804 の 2 バイトをレジスター 6 にロードしたいとします。また、以下のように仮定します。

レジスター 6 の内容は意味を持たない。

レジスター 14 の内容は、00 00 18 03。

記憶位置 1803-1804 の内容は、00 20。

このレジスターにロードするために必要な命令は、以下のとおりです。

マシン・フォーマット

命令コード	R ₁	X ₂	B ₂	D ₂
48	6	0	E	000

アセンブラー・フォーマット

命令コード R₁,D₂(X₂,B₂)

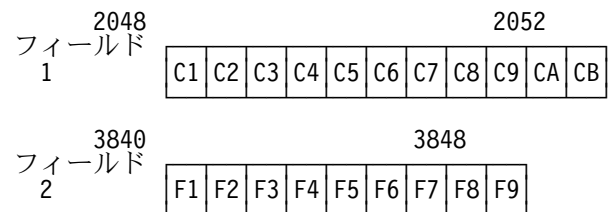
LH 6,0(0,14)

この命令の実行後、レジスター 6 の内容は 00 00 00 20 になります。記憶位置 1803-1804 の内容が負の数値 (例えば A7 B6) であったとすれば、負符号が左側に波及し、レジスター 6 の最終的な結果は FF FF A7 B6 となります。

MOVE (MVC、MVI)

MVC の例

MOVE (MVC) 命令を使用すると、データを 1 つの記憶位置から別の記憶位置へ移動することができます。例えば、記憶域内に次の 2 つのフィールドがあるとします。



また、以下のように仮定します。

レジスター 1 の内容は、00 00 20 48。

レジスター 2 の内容は、00 00 38 40。

次の命令を実行すると、フィールド 1 の最初の 8 バイトがフィールド 2 の最初の 8 バイトで置き換えられます。

マシン・フォーマット

命令コード	L	B ₁	D ₁	B ₂	D ₂
D2	07	1	000	2	000

アセンブラー・フォーマット

命令コード D₁(L,B₁),D₂(B₂)

MVC 0(8,1),0(2)

この命令の実行後、フィールド 1 は以下のようになります。

フィールド 1	2048								2052			
	F1	F2	F3	F4	F5	F6	F7	F8	C9	CA	CB	

フィールド 2 は変わりません。

MVC 命令では、第 1 オペランド・フィールドを第 2 オペランド・フィールドより 1 バイト位置だけ右側の位置から開始することにより、フィールド全体を同じバイトで埋めることもできます。例えば、記憶域内のアドレス 358 から始まる区域に、次のようなデータが含まれているとします。

358								360	
00	F1	F2	F3	F4	F5	F6	F7	F8	

以下の MVC 命令を実行すると、記憶位置 358 に含まれている 0 を、フィールド全体に波及させることができます (レジスター 11 の内容は 00 00 03 58 であるものとします)。

マシン・フォーマット

命令コード	L	B ₁	D ₁	B ₂	D ₂
D2	07	B	001	B	000

アセンブラー・フォーマット

命令コード D₁(L,B₁),D₂(B₂)

MVC 1(8,11),0(11)

MVC は一度に 1 バイトずつ処理するものとして実行されるので、実際には、まずアドレス 358 のバイトが

359 に格納され (その結果 359 には 00 が入ります)、次に 359 のバイトが 35A に格納されるというようにして、フィールド全体が 0 で埋まるまで同じ操作が繰り返されます。なお、MVI 命令を使用すれば、最初に記憶位置 358 に 0 のバイトを入れておくことができます。

注:

1. 記憶位置 358-360 を占めるフィールドには 9 個のバイトが含まれていますが、アセンブラー・フォーマットでコーディングされる長さは、移動の回数と同じ (フィールド長から 1 を引いた値) になります。
2. 実際に関係するフィールドは 1 つだけですが、オペランドの順序は重要です。

MVI の例

MOVE (MVI) 命令は、命令ストリーム内の 1 バイトの情報を記憶域に格納します。この命令は、例えば次のように使われます。

マシン・フォーマット

命令コード	I ₂	B ₁	D ₁
92	5B	1	000

アセンブラー・フォーマット

命令コード D₁(B₁),I₂

MVI 0(1),C'\$'

上記の命令を EDIT AND MARK 命令と一緒に使用することにより、ドル記号を表す EBCDIC コードを、汎用レジスター 1 に含まれている記憶域アドレスに格納することができます (EDIT AND MARK の例も参照)。

MOVE INVERSE (MVCIN)

MOVE INVERSE (MVCIN) を使用すると、1 つの記憶位置にあるデータを、フィールド内でのバイトの順序を逆にして、別の記憶位置に移動することができます。例えば、記憶域内に次の 2 つのフィールドがあるとします。

フィールド 1	2048								2052			
	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	

フィールド 2	3840								3848	
	F1	F2	F3	F4	F5	F6	F7	F8	F9	

また、以下のように仮定します。

レジスタ 1 の内容は、00 00 20 48。

レジスタ 2 の内容は、00 00 38 40。

次の命令を実行すると、フィールド 1 の最初の 8 バイトがフィールド 2 の最初の 8 バイトで置き換えられます。

マシン・フォーマット

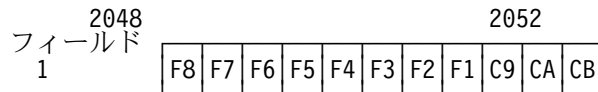
命令コード	L	B ₁	D ₁	B ₂	D ₂
E8	07	1	000	2	007

アセンブラー・フォーマット

命令コード D₁(L,B₁),D₂(B₂)

MVCIN 0(8,1),7(2)

この命令の実行後、フィールド 1 は以下のようになりません。



フィールド 2 は変わりません。

注: この例では、MVC の最初の例で使用したものと同一汎用レジスタ、記憶位置、および元の値を使用しています。MVCIN の場合は、第 2 オペランド・アドレスは、移動するフィールドの右端バイト (この例では記憶位置 3847) を指定していなければなりません。そのために、この命令の D₂ フィールドに 7 が指定されています。

MOVE LONG (MVCL)

MOVE LONG (MVCL) 命令を使用すると、MVC 命令の最初の例の場合と同様に、記憶域内のデータを移動することができます。ただし、2 つのオペランドがオーバーラップしていないことが必要です。MVCL が MVC と違うのは、各オペランドのアドレスと長さを、奇数番号と偶数番号から成る汎用レジスタのペアで指定するという点です。したがって、MVCL では、1 つの命令で 256 バイトを超えるデータを移動することができます。例えば次のように仮定します。

レジスタ 2 の内容は、00 0A 00 00。

レジスタ 3 の内容は、00 00 08 00。

レジスタ 8 の内容は、00 06 00 00。

レジスタ 9 の内容は、00 00 08 00。

次の命令を実行します。

マシン・フォーマット
命令コード R₁ R₂

0E	8	2
----	---	---

アセンブラー・フォーマット

命令コード R₁,R₂

MVCL 8,2

これにより、2,048₁₀ バイトが、記憶位置 A0000-A07FF から記憶位置 60000-607FF へ移動します。CPU が 24 ビット・アドレッシング・モードで稼働しているとすれば、レジスタ 2 および 8 のビット 8-31 が 800₁₆ だけ増加し、レジスタ 2 および 8 のビット 0-7 は 0 に設定されます。レジスタ 3 および 9 のビット 8-31 は 0 に減少します。条件コード 0 が設定されて、オペランド長が等しいことを示します。

レジスタ 3 の内容が F0 00 04 00 であったとすれば、1,024₁₀ バイトのみが記憶位置 A0000-A03FF から記憶位置 60000-603FF へ移動しています。そして、第 1 オペランドの残りの記憶位置 60400-607FF には、レジスタ 3 の左端バイトの指定に従い、埋め込みバイト X'F0' が 1,024 個コピーされます。レジスタ 2 のビット 8-31 は 400₁₆ だけ増加し、レジスタ 8 のビット 8-31 は 800₁₆ だけ増加し、レジスタ 2 および 8 のビット 0-7 は 0 に設定されています。レジスタ 3 および 9 のビット 8-31 は、0 になるまで減少していません。そして、第 1 オペランドの方が第 2 オペランドより長いことを示す条件コード 2 が設定されています。

MVCL では、MVC の 2 番目の例で示した、フィールド全体を 0 に設定する手法は使用できません。MVCL で、そのような操作を行うようにレジスタをセットアップした場合は、データの移動は行われず、破壊的オーバーラップを示す条件コード 3 が設定されます。

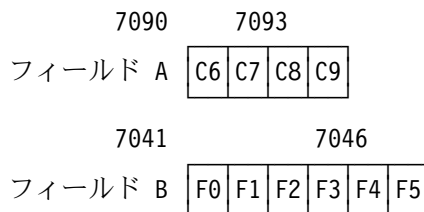
その代わりに、MVCL を以下のように使用することにより、1 つの記憶域を消去して 0 にすることができます。レジスタ 8 および 9 は、前の例と同じにセットアップされているものとします。レジスタ 3 には 0 のみが含まれています。これは、第 2 オペランドの長さが 0 であり、埋め込みバイトが 0 であることを指定しています。レジスタ 2 は記憶域にアクセスするためには使用されず、その内容には意味はありません。ここで MVCL 8,2 という命令を実行すると、記憶位置 60000-607FF が 0 で埋められます。レジスタ 8 のビット 8-31 は 800₁₆ だけ増やされ、レジスタ 2 およ

び 8 のビット 0-7 は 0 に設定されます。レジスター 9 のビット 8-31 は 0 に減少し、第 1 オペランドの方が第 2 オペランドより長いことを示す条件コード 2 が設定されます。

MOVE NUMERICS (MVN)

互いに関連する 2 つの命令 MOVE NUMERICS および MOVE ZONES を、ゾーン・フォーマットの 10 進数データに対して使用することにより、各バイトの右端 4 ビット (数字ビット) と左端 4 ビット (ゾーン・ビット) を別々に操作することができます。どちらの命令も MOVE (MVC) に似ていますが、MOVE NUMERICS は数字ビットのみを移動し、MOVE ZONES はゾーン・ビットのみを移動する点が異なります。

MOVE NUMERICS 命令の働きを説明するために、記憶域内に次の 2 つのフィールドがあると仮定します。



また、以下のように仮定します。

レジスター 14 の内容は、00 00 70 90。

レジスター 15 の内容は、00 00 70 40。

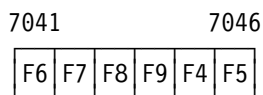
次の命令を実行します。

マシン・フォーマット					
命令コード	L	B ₁	D ₁	B ₂	D ₂
D1	03	F	001	E	000

アセンブラー・フォーマット
命令コード D₁(L,B₁),D₂(B₂)

MVN 1(4,15),0(14)

実行の結果、フィールド B は以下のようになります。



記憶位置 7090-7093 にあるバイトの数字ビットが、記憶位置 7041-7044 にあるバイトの数字ビットに格納されています。記憶位置 7090-7093 および 7045-7046 の内容は、変更されません。

MOVE STRING (MVST)

MOVE STRING 命令は、汎用レジスター R₂ に指定されている第 2 オペランドを、汎用レジスター R₁ に指定されている第 1 オペランド位置に移動するために使用します。移動は左から右へ行われ、汎用レジスター 0 に指定されている終了文字の移動が完了するまで、または CPU で決定されたバイト数の移動が完了するまで続けられます。条件コードは、終了文字が移動された場合は 1 に設定され、CPU 決定のバイト数が移動された場合は 3 に設定されます。

条件コード 1 が設定された場合は、第 1 オペランド内の終了文字が汎用レジスター R₁ に入れられ、汎用レジスター R₂ の内容は変更されません。条件コード 3 が設定された場合は、第 1 オペランドおよび第 2 オペランド内で次に処理されるバイトのアドレスが、それぞれ汎用レジスター R₁ および R₂ に入れられます。

次の例に示すプログラムは、ストリング A を、ストリング B の後にストリング C を連結したストリングに設定します。ここで、ストリング B および C の長さはいずれも不明であり、これらのストリングの終わりは、終了文字である 16 進数 00 (C プログラミング言語で使用される) によって示されます。このプログラムは、アクセス・レジスター・モードで実行するためのものとしては書かれていません。

```

L      4,STRAADR
L      5,STRBADR
SR     0,0
LOOP1  MVST 4,5
        BC  1,LOOP1
        L   5,STRCADR
LOOP2  MVST 4,5
        BC  1,LOOP2
        [任意の命令]

```

MOVE WITH OFFSET (MVO)

MOVE WITH OFFSET は、パック 10 進数を奇数の桁数分だけシフトするため、または符号なしパック 10 進数に符号を連結するために使用できます。

記憶位置 4500-4502 にある 3 バイトの符号なしパック 10 進数を、記憶位置 5600-5603 に移動し、それに記憶位置 5603 で終わっているパック 10 進数の符号を付けるものとしします。また、以下のように仮定します。

レジスター 12 の内容は、00 00 56 00。

レジスター 15 の内容は、00 00 45 00。

記憶位置 5600-5603 の内容は、77 88 99 0C。

記憶位置 4500-4502 の内容は、12 34 56。

次の命令を実行します。

マシン・フォーマット

命令コード L₁ L₂ B₁ D₁ B₂ D₂

F1	3	2	C	000	F	000
----	---	---	---	-----	---	-----

アセンブラー・フォーマット

命令コード D₁(L₁,B₁),D₂(L₂,B₂)

MV0 0(4,12),0(3,15)

実行後、記憶位置 5600-5603 の内容は 01 23 45 6C になります。第 1 オペランド・フィールドを満たすために、第 2 オペランドの左側に 0 が 1 個付加されて拡張されるという点に注意してください。

MOVE ZONES (MVZ)

MOVE ZONES 命令は、MOVE (MVC) および MOVE NUMERICS 命令と同様に、オーバーラップ・フィールドでも非オーバーラップ・フィールドでも取り扱うことができます。非オーバーラップ・フィールドを取り扱うときは、MOVE ZONES は MOVE NUMERICS と同じような働きをします (MOVE NUMERICS の例を参照)。ただし、MOVE ZONES は、各バイトのゾーン・ビットのみを移動します。オーバーラップ・フィールドに対する MOVE ZONES の使用方法を説明するために、記憶域内に以下のデータ・フィールドがあるものと仮定します。

800 805

F1	C2	F3	C4	F5	C6
----	----	----	----	----	----

また、レジスター 15 の内容は 00 00 08 00 であるものとします。ここで次の命令を実行したとします。

マシン・フォーマット

命令コード L B₁ D₁ B₂ D₂

D3	04	F	001	F	000
----	----	---	-----	---	-----

アセンブラー・フォーマット

命令コード D₁(L,B₁),D₂(B₂)

MVZ 1(5,15),0(15)

実行の結果、アドレス 800 にあるバイトのゾーン・ビットがフィールド全体に波及し、フィールドは次のようになります。

800 805

F1	F2	F3	F4	F5	F6
----	----	----	----	----	----

MULTIPLY (M, MR)

レジスター 5 に含まれている数に、アドレス 3750 にある 4 バイト・フィールドの内容を乗算するものとします。それぞれの初期値は以下のとおりとします。

レジスター 4 の内容は意味を持たない。

レジスター 5 の内容は、00 00 00 9A = 154₁₀ = 被乗数。

レジスター 11 の内容は、00 00 06 00。

レジスター 12 の内容は、00 00 30 00。

記憶位置 3750-3753 の内容は、00 00 00 83 = 131₁₀ = 乗数。

この乗算を行うために必要な命令は、以下のとおりです。

マシン・フォーマット

命令コード R₁ X₂ B₂ D₂

5C	4	B	C	150
----	---	---	---	-----

アセンブラー・フォーマット

命令コード R₁,D₂(X₂,B₂)

M 4,X'150'(11,12)

実行後、レジスター 4 と 5 のペアに積が入ります。

レジスター 4 の内容は、00 00 00 00。

レジスター 5 の内容は、00 00 4E CE = 20,174₁₀。

記憶位置 3750-3753 は変わらない。

この命令の RR フォーマットを使用すると、レジスター内の数を 2 乗することができます。レジスター 7 に、00 01 00 05 が含まれているとします。レジスター 6 の内容は意味を持ちません。ここで次の命令を実行したとします。

マシン・フォーマット

命令コード R₁ R₂

1C	6	7
----	---	---

アセンブラー・フォーマット
命令コード R₁,R₂

MR 6,7

実行の結果、レジスター 7 の中の数が 2 乗され、その結果が次のようにレジスター 6 と 7 のペアに入れられます。

レジスター 6 には 00 00 00 01 が入ります。

レジスター 7 には 00 0A 00 19 が入ります。

MULTIPLY HALFWORD (MH)

MULTIPLY HALFWORD は、レジスターの内容に、記憶域内の 2 バイト・フィールドを乗算するために使用します。例えば次のように仮定します。

レジスター 11 の内容は、00 00 00 15 = 21₁₀ = 被乗数。

レジスター 14 の内容は、00 00 01 00。

レジスター 15 の内容は、00 00 20 00。

記憶位置 2102-2103 の内容は、FF D9 = -39₁₀ = 乗数。

ここで次の命令を実行したとします。

マシン・フォーマット
命令コード R₁ X₂ B₂ D₂

4C	B	E	F	002
----	---	---	---	-----

アセンブラー・フォーマット
命令コード R₁,D₂(X₂,B₂)

MH 11,2(14,15)

これにより、2 つの数が乗算されます。そして、積 (FF FC CD = -819₁₀) がレジスター 11 の元の内容と置き換わります。

レジスターに格納されるのは積の右端 32 ビットのみで、それより左にある有効ビットはすべての失われます。オーバーフローが生じて、プログラム割り込みは起きません。

OR (O, OC, OI, OR)

2 個のビットにブール演算子 OR を適用すると、いずれかのビットが 1 であれば結果は 1 となり、それ以外の場合は結果は 0 になります。2 個のバイトに対して OR 演算を行うと、対応するビットが一對ずつ別個に処理されます。1 つのビット位置と他のビット位置との間には関連性はありません。以下の例は、2 個のバイトに対する OR 演算の結果を示しています。

第 1 オペランドのバイト: 0011 0101₂
第 2 オペランドのバイト: 0101 1100₂

結果のバイト: 0111 1101₂

OI の例

OR 命令は、特定のビットを 1 に設定するためによく使用されます。例えば、記憶位置 4891 に 0100 0010₂ が含まれているとします。このバイトの右端のビットのみを 1 に設定し、他のビットは変わらないようにするには、次のような命令を使用できます (レジスター 8 の内容は、00 00 48 90 であるものとします)。

マシン・フォーマット
命令コード I₂ B₁ D₁

96	01	8	001
----	----	---	-----

アセンブラー・フォーマット
命令コード D₁(B₁),I₂

OI 1(8),X'01'

この命令が実行されると、記憶域内のバイトと即値バイト (命令の I₂ フィールド) との OR 演算が行われます。

記憶位置 4891: 0100 0010₂
即値バイト: 0000 0001₂

結果: 0100 0011₂

結果のバイト (ビット 7 が 1 に設定されたもの) が、記憶位置 4891 に戻されて格納されます。そして、条件コード 1 が設定されます。

PACK (PACK)

記憶位置 1000-1003 に以下のゾーン 10 進数が含まれており、その数をそのままの位置でパック 10 進数に変換するものとします。

	1000	1003		
ゾーン数	F1	F2	F3	C4

レジスター 12 に、00 00 10 00 が入っているものとします。次の命令を実行します。

マシン・フォーマット

命令コード	L1	L2	B1	D1	B2	D2
F2	3	3	C	000	C	000

アセンブラー・フォーマット

命令コード D1(L1,B1),D2(L2,B2)

PACK 0(4,12),0(4,12)

実行が完了すると、記憶位置 1000-1003 には以下のパック 10 進数フォーマットの結果が含まれています。

	1000	1003
パック数	00 01 23 4C	

注:

- この例は、第 1 オペランド・フィールドと第 2 オペランド・フィールドが完全にオーバーラップしている場合の PACK の働きを示しています。
- 操作中に、第 2 オペランドは左側に 0 を付加して拡張されます。

SEARCH STRING (SRST)

SEARCH STRING 命令は、汎用レジスター R2 に指定されている第 2 オペランドから、汎用レジスター 0 に指定されている文字を検索するために使用します。第 2 オペランドの長さは確定しています。つまり、汎用レジスター R1 に、第 2 オペランドの後の最初のバイトのアドレスが入っています。

指定した文字が見つかったと、条件コード 1 が設定され、その文字のアドレスが汎用レジスター R1 に入れられ、汎用レジスター R2 の内容は変わりません。第 2 オペランドの中の次に検査する順番に当たっているバイトのアドレスが、汎用レジスター R1 の中のアドレスに一致している場合は、条件コード 2 が設定され、汎用レジスター R1 および R2 の内容は変わりません。CPU が決定した第 2 オペランドのバイト数をすべて検査し終えた場合は、条件コード 3 が設定され、第 2 オペランド内の次に処理すべきバイトのアドレスが汎用レジスター R2 に入れられ、汎用レジスター R1 の内容は変更されません。

SRST の例 1

次の例に示すプログラムは、終了文字として指定されている 16 進数 00 (C プログラミング言語で終了文字として使用されるように) により、string A の終わりを判別し、そのstring内で 16 進数 C1 に相当する最初の文字のアドレスを判別します。このプログラムは、第 2 オペランドが位置 0 から始まっておらず、記憶域内で循環しないことを前提としているので、汎用レジスター 0 の中のアドレスが原因で、最初の SEARCH STRING 命令により条件コード 2 が設定されることはありません。このプログラムは、アクセス・レジスター・モードで実行するためのものとしては書かれていません。

```

L      5,STRAADR
SR     0,0
LOOP1  SRST 0,5
      BC   1,LOOP1
      L    5,STRAADR
      LR   4,0
      LA   0,X'C1'
LOOP2  SRST 4,5
      BC   1,LOOP2
      BC   2,NOTFND
FOUND  [任意の命令]
NOTFND [任意の命令]

```

SRST の例 2

次の例に示すプログラムは、長さが確定しているstring A の中で、16 進数 C1 に相当する最初の文字のアドレスを判別します。このプログラムは、アクセス・レジスター・モードで実行するためのものとしては書かれていません。

```

L      5,STRAADR
L      4,STRALEN
AR     4,5
LA     0,X'C1'
LOOP1  SRST 4,5
      BC   1,LOOP1
      BC   2,NOTFND
FOUND  [任意の命令]
NOTFND [任意の命令]

```

この例では、終了文字が検索対象の文字ではない場合は、STRALEN の中の値は、stringの終わりの終了文字を含む長さのこともあり、それを含まない長さのこともあります。

SHIFT LEFT DOUBLE (SLDA)

SHIFT LEFT DOUBLE 命令は、偶数番号と奇数番号を持つ一対のレジスターの 63 個のビットを左にシフトし、符号ビットは無変更のまま残しておきます。つまり、この命令は、64 ビットの符号付き 2 進整数を代数的に左にシフトします。

例えば、レジスタ 2 と 3 の内容が以下のとおりであるとします。

```
00 7F 0A 72   FE DC BA 98 =
00000000 01111111 00001010 01110010
11111110 11011100 10111010 100110002
```

ここで次の命令を実行したとします。

マシン・フォーマット
命令コード R₁ B₂ D₂

8F	2	////	0	01F
----	---	------	---	-----

アセンブラー・フォーマット
命令コード R₁,D₂(B₂)

```
SLDA      2,31(0)
```

実行の結果、レジスタ 2 と 3 はどちらも左に 31 ビット位置だけシフトされ、これらのレジスタの新しい内容は以下ようになります。

```
7F 6E 5D 4C   00 00 00 00 =
01111111 01101110 01011101 01001100
00000000 00000000 00000000 000000002
```

この例では、レジスタ 2 のビット位置 1 から外へ有効ビットがシフトされるため、オーバーフローを示す条件コード 3 が設定されます。さらに、PSW の固定小数点オーバーフロー・マスク・ビットが 1 であれば、固定小数点オーバーフロー・プログラム割り込みも起きません。

SHIFT LEFT SINGLE (SLA)

SHIFT LEFT SINGLE 命令は、単一のレジスタの 31 個の数字ビットのみをシフトするという点以外は、SHIFT LEFT DOUBLE と同じです。つまり、この命令は、32 ビットの符号付き 2 進整数を代数的に左にシフトします。

例えば、レジスタ 2 の内容が以下のとおりであるとします。

```
00 7F 0A 72 = 00000000 01111111 00001010 011100102
```

ここで次の命令を実行したとします。

マシン・フォーマット
命令コード R₁ B₂ D₂

8B	2	////	0	008
----	---	------	---	-----

アセンブラー・フォーマット
命令コード R₁,D₂(B₂)

```
SLA      2,8(0)
```

この実行の結果、レジスタ 2 は左に 8 ビット位置だけシフトされ、このレジスタの新しい内容は以下のようになります。

```
7F 0A 72 00 = 01111111 00001010 01110010 000000002
```

条件コード 2 が設定されて、結果が 0 より大きいことを示します。

この例で 9 桁の左桁移動が指定していたとすれば、ビット位置 1 から外へ有効ビットがシフトされることとなります。その場合は、そのオーバーフローを示す条件コード 3 が設定されます。さらに、PSW の固定小数点オーバーフロー・マスク・ビットが 1 であれば、固定小数点オーバーフロー割り込みも起きます。

STORE CHARACTERS UNDER MASK (STCM)

STORE CHARACTERS UNDER MASK (STCM)

は、レジスタ内の選択したバイトを記憶域に入れるために使用できます。例えば、汎用レジスタ 8 から 3 バイトのアドレスを記憶位置 FIELD3 に格納したいとすれば、以下のような命令を実行します。

マシン・フォーマット
命令コード R₁ M₃ S₂

BE	8	7	* * * *
----	---	---	---------

レジスタ・フォーマット
命令コード R₁,M₃,S₂

```
STCM      8,B'0111',FIELD3
```

```
レジスタ 8:      12 34 56 78
FIELD3 (実行前): 意味を持たない
FIELD3 (実行後): 34 56 78
```

もう 1 つ例を示します。

マシン・フォーマット
命令コード R₁ M₃ S₂

BE	9	5	* * * *
----	---	---	---------

レジスター・フォーマット
命令コード R₁,M₃,S₂

STCM 9,B'0101',FIELD2

レジスター 9: 01 23 45 67
FIELD2 (実行前): 意味を持たない
FIELD2 (実行後): 23 67

STORE MULTIPLE (STM)

以下の条件下で、汎用レジスター 14、15、0、および 1 の内容を、記憶位置 4050 から始まる連続した 4 バイトのフィールドに格納するものとします。

レジスター 14 の内容は、00 00 25 63。

レジスター 15 の内容は、00 01 27 36。

レジスター 0 の内容は、12 43 00 62。

レジスター 1 の内容は、73 26 12 57。

レジスター 6 の内容は、00 00 40 00。

記憶位置 4050-405F の初期内容は意味を持たない。

STORE MULTIPLE 命令は、1 回使用するだけで、4 個のレジスターの内容を格納することができます。

マシン・フォーマット
命令コード R₁ R₃ B₂ D₂

90	E	1	6	050
----	---	---	---	-----

アセンブラー・フォーマット
命令コード R₁,R₃,D₂(B₂)

STM 14,1,X'50' (6)

命令の実行結果は以下のようになります。

記憶位置 4050-4053 の内容は、00 00 25 63。

記憶位置 4054-4057 の内容は、00 01 27 36。

記憶位置 4058-405B の内容は、12 43 00 62。

記憶位置 405C-405F の内容は、73 26 12 57。

TEST UNDER MASK (TM)

TEST UNDER MASK 命令は、1 つのバイトの中の選択されたビットを検査し、その結果を示す条件コードを設定します。例えば次のように仮定します。

記憶位置 9999 の内容は、FB。

レジスター 7 の内容は、00 00 99 90。

次の命令を実行したとします。

マシン・フォーマット
命令コード I₂ B₁ D₁

91	C3	7	009
----	----	---	-----

アセンブラー・フォーマット
命令コード D₁(B₁),I₂

TM 9(7),B'11000011'

この命令は、記憶域内の指定されたバイトのビットのうち、対応するマスク・ビットが 1 であるビットのみをテストします。

FB = 1111 1011₂

マスク = 1100 0011₂

テスト = 11xx xx11₂

条件コード 3 が設定されて、テスト結果の中の選択されたビットがすべて 1 であることを示します。(「x」のマークが付いたビットは無視されます。)

記憶位置 9999 の内容が B9 であったとすれば、テスト結果は以下のようになります。

B9 = 1011 1001₂

マスク = 1100 0011₂

テスト = 10xx xx01₂

条件コード 1 が設定されて、選択されたビットに 0 と 1 の両方があることを示します。

記憶位置 9999 の内容が 3C であったとすれば、テスト結果は以下のようになります。

3C = 0011 1100₂

マスク = 1100 0011₂

テスト = 00xx xx00₂

条件コード 0 が設定されて、選択されたビットがすべて 0 であることを示します。

注: 記憶位置 9999 の内容は変更されません。

TRANSLATE (TR)

TRANSLATE 命令を使用すると、データを任意の文字コードから他の任意のコードに変換することができます。ただし、各文字コードは 8 ビット以下でなければなりません。また、記憶域内に適切な変換テーブルがあることが必要です。

以下の例では、EBCDIC コードが ASCII コードに変換されます。最初のステップは、記憶位置 1000-10FF に 256 バイトのテーブルを作成することです。このテーブルには、ASCII コードの文字を EBCDIC コードの 2 進数表現の順序で並べたものが入っています。つまり、ある文字の ASCII 表現が入れられる記憶位置は、テーブルの開始アドレスに、その文字の EBCDIC 表現の 2 進数値を加えた位置になります。

以下の例では、説明を簡単にするために、テーブルの中の 10 進数字を含む部分のみを示してあります。

10F0	10F9								
30	31	32	33	34	35	36	37	38	39

記憶位置 2100 にある 4 バイト・フィールドに、数字 1984 を表す EBCDIC コードが含まれているとします。

記憶位置 2100-2103 の内容は、F1 F9 F8 F4。

レジスター 12 の内容は、00 00 21 00。

レジスター 15 の内容は、00 00 10 00。

ここで次の命令を実行したとします。

マシン・フォーマット

命令コード	L	B ₁	D ₁	B ₂	D ₂
DC	03	C	000	F	000

アセンブラー・フォーマット

命令コード	D ₁ (L, B ₁), D ₂ (B ₂)
TR	0(4, 12), 0(15)

実行の結果、各 EBCDIC バイトの 2 進値がテーブルの開始アドレスに加算され、その結果のアドレスに該当する ASCII バイトが取り出されます。

テーブル開始アドレス: 1000
最初の EBCDIC バイト: F1

ASCII バイトのアドレス: 10F1

命令の実行結果は次のとおりです。

記憶位置 2100-2103 の内容は、31 39 38 34。

このように、記憶位置 2100 にある 4 バイト・フィールドの中で、数字 1984 を表す ASCII コードが EBCDIC コードで置き換えられています。

TRANSLATE AND TEST (TRT)

TRANSLATE AND TEST 命令は、データ・フィールドをスキャンして、特殊な意味を持つ文字を見つけるために使用できます。どの文字が特殊な意味を持つのかを示すために、TRANSLATE 命令で使用したものと同様のテーブルがセットアップされます。ただし、このテーブルでは、特殊な意味を持たない文字は 0 で示され、特殊な意味を持つ文字は 0 以外の値で示されます。

図A-4 は、英数字 (A から Z までと 0 から 9 まで) を、ブランク、特定の特殊記号、および、無効と見なされる他のすべての文字と区別するようにセットアップされています。文字コードは EBCDIC であるものとします。また、この 256 バイトのテーブルは記憶位置 2000-20FF に格納されているものとします。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
200	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
201	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
202	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
203	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
204	04	40	40	40	40	40	40	40	40	40	08	40	0C	10	40	40
205	14	40	40	40	40	40	40	40	40	40	18	1C	20	40	40	40
206	24	28	40	40	40	40	40	40	40	40	2C	40	40	40	40	40
207	40	40	40	40	40	40	40	40	40	40	30	34	38	3C	40	40
208	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
209	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
20A	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
20B	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
20C	40	00	00	00	00	00	00	00	00	00	40	40	40	40	40	40
20D	40	00	00	00	00	00	00	00	00	00	40	40	40	40	40	40
20E	40	40	00	00	00	00	00	00	00	00	40	40	40	40	40	40
20F	00	00	00	00	00	00	00	00	00	00	40	40	40	40	40	40

注: 変換するステートメントに含まれる文字コードが 00 から FF₁₆ までより小さい範囲に収まる場合は、256 バイトより小さいテーブルを使用できます。

図 A-4. 変換およびテスト・テーブル

EBCDIC の英数字に対応するテーブル・エントリーは、すべて 00 です。したがって、英字の A (コード

C1) に対応するバイト位置 20C1 には 00 が入っています。

15 個の特殊記号には、増分値を 4 として 04₁₆ から 3C₁₆ までの非ゼロ値が割り当てられています。例えば、ブランク (コード 40) に対応するエントリーは 04₁₆ で、ピリオド (コード 4B) に対応するエントリーは 08₁₆ です。

その他のすべてのテーブル位置には、無効文字を表すエントリー 40₁₆ を入れてあります。

テーブル・エントリーは、16 個のワードの 1 つを選択するために使用できます。この 16 個のワードには、スキャン中に特殊記号または無効文字が検出されたときに、それぞれの記号または文字に応じて実行すべき異なるルーチンのアドレスが含まれています。

例えば、この 16 個のブランチ・アドレスが記憶位置 3004-3043 に格納されているとします。

記憶位置 CA80 から始まって、以下に示す 21₁₆ 個の EBCDIC 文字が含まれています。「b」はブランクを表します。

記憶位置 CA80-CA94:
UNPKbPROUT(9),WORD(5)

また、以下のように仮定します。

レジスター 1 の内容は、00 00 CA 7F。

レジスター 2 の内容は、00 00 30 00。

レジスター 15 の内容は、00 00 20 00。

ここで次の命令を実行したとします。

マシン・フォーマット

命令コード	L	B ₁	D ₁	B ₂	D ₂
	DD	14	1	001	F 000

アセンブラー・フォーマット

命令コード D₁(L,B₁),D₂(B₂)

TRT 1(21,1),0(15)

実行の結果、最初のソース・バイトの値 (英字 U を表す EBCDIC コード) がテーブルの開始アドレスに加算されて、検査するテーブル・エントリーのアドレスが求められます。

テーブル開始アドレス	2000
最初のソース・バイト (U)	E4

テーブル・エントリーのアドレス 20E4

記憶位置 20E4 には 0 が入っているため、特別の処置はとられません。操作は 2 番目以降のソース・バイトに進み、記憶位置 CA84 のブランクに達するまで続けられます。ブランク記号に達すると、通常と同様に、その値がテーブルの開始アドレスに加算されます。

テーブル開始アドレス	2000
ソース・バイト (ブランク)	40

テーブル・エントリーのアドレス 2040

記憶位置 2040 には 0 以外の値が入っているため、以下の処置がとられます。

ソース・バイトのアドレス 00CA84 が、レジスター 1 の右端 24 ビットに入れられます。

テーブル・エントリー 04 がレジスター 2 の右端 8 ビットに入れられ、レジスター 2 の内容は 00 00 30 04 となります。

条件コード 1 (スキャン未完了) が設定されます。

TRANSLATE AND TEST の後に、記憶位置 3004 のアドレスにあるルーチンにブランチするための命令を配置しておくことができます。これは、スキャン中にブランク文字が検出されたときに実行されるルーチンです。このルーチンが完了した時点で、プログラム制御を TRANSLATE AND TEST 命令に戻して、スキャンが続行されるようにすることができます。ただし、すでにスキャン済みの文字を考慮に入れて、まず長さを調整する必要があります。

この目的のためには、EXECUTE 命令を使用して TRANSLATE AND TEST を実行します。これにより、汎用レジスターに指定されている長さが適用されます。この方法で、EXECUTE により単一の TRANSLATE AND TEST 命令を繰り返し実行することによって、記憶域内のどの命令も変更せずに、完全なステートメント・スキャンを行うことができます。上記の例では、TRANSLATE AND TEST の最初の実行後には、レジスター 1 には、最後に変換されたソース・バイトのアドレスが入っています。このアドレスを最後のソース・バイトのアドレス (CA94) から減算することにより、簡単に長さ指定が得られます。この長さから 1 を引いた値が、EXECUTE 命令の R₁ フィールドで指定されているレジスターに入れられます。(マシン・フォーマットでの長さコードは、フィールド内の総バイト数から 1 を引いた値です。) EXECUTE 命令は、TRANSLATE AND TEST 命令を指し示しています。これは上記に示した命令と同じですが、長さ (L) が 0 に設定される点が異なります。

UNPACK (UNPK)

記憶位置 2501-2502 に含まれている符号付きパック 10 進数をアンパックして、記憶位置 1000-1004 に格納するものとします。また、以下のように仮定します。

レジスター 12 の内容は、00 00 10 00。

レジスター 13 の内容は、00 00 25 00。

記憶位置 2501-2502 の内容は、12 3D。

記憶位置 1000-1004 の初期内容は意味を持ちません。

次の命令を実行します。

マシン・フォーマット

命令コード	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
F3	4	1	C	000	D	001

アセンブラー・フォーマット

命令コード D₁(L₁,B₁),D₂(L₂,B₂)

UNPK 0(5,12),1(2,13)

実行後、記憶位置 1000-1004 の内容は F0 F0 F1 F2 になります。

UPDATE TREE (UPT)

A-51 ページの『ソート命令』を参照してください。

10 進数命令

(10 進数命令の詳細な説明については、第8章、『10 進数命令』を参照してください。)

ADD DECIMAL (AP)

記憶位置 500-503 に入っている符号付きパック 10 進数を、記憶位置 2000-2002 に入っている符号付きパック 10 進数に加算するものとします。また、以下のように仮定します。

レジスター 12 の内容は、00 00 20 00。

レジスター 13 の内容は、00 00 05 00。

記憶位置 2000-2002 の内容は、38 46 0D (負の数)。

記憶位置 500-503 の内容は、01 12 34 5C (正の数)。

次の命令を実行します。

マシン・フォーマット

命令コード L₁ L₂ B₁ D₁ B₂ D₂

命令コード	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
FA	2	3	C	000	D	000

アセンブラー・フォーマット

命令コード D₁(L₁,B₁),D₂(L₂,B₂)

AP 0(3,12),0(4,13)

実行の結果、記憶位置 2000-2002 の内容は 73 88 5C となり、結果が 0 より大きいことを示す条件コード 2 が設定されます。以下の点に注意してください。

- 2 つの数の符号が異なっているため、実際にはこれは減算です。
- 第 1 オペランドより第 2 オペランドの方が長いのですが、結果は第 1 オペランドの長さに完全に収まるので、オーバーフロー割り込みは起きません。

COMPARE DECIMAL (CP)

記憶位置 700-703 に入っている符号付きパック 10 進数の内容を、記憶位置 500-502 に入っている符号付きパック 10 進数の内容と代数的に比較するものとします。また、以下のように仮定します。

レジスター 12 の内容は、00 00 06 00。

レジスター 13 の内容は、00 00 03 00。

記憶位置 700-703 の内容は、17 25 35 6D。

記憶位置 500-502 の内容は、72 14 2D。

次の命令を実行します。

マシン・フォーマット

命令コード L₁ L₂ B₁ D₁ B₂ D₂

命令コード	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
F9	3	2	C	100	D	200

アセンブラー・フォーマット

命令コード D₁(L₁,B₁),D₂(L₂,B₂)

CP X'100'(4,12),X'200'(3,13)

実行の結果、条件コード 1 が設定されます。これは、第 1 オペランド (記憶位置 700-703 の内容) が第 2 オペランドより小さいことを意味します。

DIVIDE DECIMAL (DP)

記憶位置 2000-2004 に入っている符号付きパック 10 進数 (被除数) を、記憶位置 3000-3001 に入っている符号付きパック 10 進数 (除数) で除算するものとします。また、以下のように仮定します。

レジスター 12 の内容は、00 00 20 00。

レジスター 13 の内容は、00 00 30 00。

記憶位置 2000-2004 の内容は、01 23 45 67 8C。

記憶位置 3000-3001 の内容は、32 1D。

次の命令を実行します。

マシン・フォーマット

命令コード	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
FD	4	1	C	000	D	000

アセンブラー・フォーマット

命令コード	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)
DP	0(5,12),0(2,13)

実行の結果、以下のように、被除数全体が符号付きの商と剰余で置き換えられます。

記憶位置	2000	2001	2002	2003	2004
2000-2004	38	46	0D	01	8C
	商			剰余	

注:

1. 被除数と除数の符号が異なっているので、商の符号は負になります。
2. 剰余の符号は被除数の符号と同じで、長さは除数と同じです。
3. 被除数を、記憶位置 3001 の 1 バイト・フィールドで除算しようとしたとすれば、商が長くなりすぎて、割り当てられている 4 バイトには収まらなくなります。その場合は、10 進数除算例外が生じ、プログラム割り込みが起きることになります。

EDIT (ED)

パック・フォーマットの 10 進数データを報告書に印刷できるようにするには、数字と符号を印刷可能な文字に変換する必要があります。さらに、コンマや小数点などの句読記号を、適切な位置に挿入することが必要になることもあります。高度な柔軟性を備えた EDIT 命令を使用すれば、1 回の実行でこれらの機能をすべて実施することができます。

この例では、EDIT 命令の使用方法を段階を追って示します。編集するフィールド (ソース) の長さは 4 バイトです。これを、13 バイトの長さのパターンに合わせて編集します。使用する記号は以下のとおりです。

記号	意味
b (16 進数 40)	空白文字 有効数字開始文字 数字選択文字
((16 進数 21)	
d (16 進数 20)	

レジスター 12 の内容は以下のとおりとします。

00 00 10 00

ソース・フィールドとパターン・フィールドは以下のとおりとします。

ソース

1200 1203

02	57	42	6C
----	----	----	----

↑
+

パターン

1000 100C

40	20	20	6B	20	21	20	4B	20	20	40	C3	D9
b	d	d	,	d	(d	.	d	d	b	C	R

次の命令を実行します。

マシン・フォーマット

命令コード	L	B ₁	D ₁	B ₂	D ₂
DE	0C	C	000	C	200

アセンブラー・フォーマット

命令コード	D ₁ (L,B ₁),D ₂ (B ₂)
ED	0(13,12),X'200'(12)

実行の結果、パターン・フィールドは以下のように変更されます。

パターン	数字	有効数字標識 (前/後)	規則	記憶位置 1000-100C
b d d	0 2	オフ/オフ オフ/オフ オフ/オン	残る (1) 充てん 数字	bdd,d(d.ddbCR bbd,d(d.ddbCR bb2,d(d.ddbCR
, d (d	5 7 4	オン/オン オン/オン オン/オン	残る 数字 数字	同上 bb2,5(d.ddbCR bb2,57d.ddbCR bb2,574.ddbCR
. d d	2 6+	オン/オン オン/オフ	残る 数字 数字	同上 bb2,574.2dbCR bb2,574.26bCR
b C R		オフ/オフ オフ/オフ オフ/オフ	充てん 充てん 充てん	同上 bb2,574.26bbR bb2,574.26bbb

注:

- この文字は充てんバイトです。
- ソースの中の 0 でない最初の 10 進数字の位置で、有効数字標識がオンになります。
- このバイトの右端 4 ビットに含まれる正符号により、有効数字標識がオフにされます。

命令実行後、パターン・フィールドには以下の結果が入っています。

パターン 1000 100C

40	40	F2	6B	F5	F7	F4	4B	F2	F6	40	40	40
----	----	----	----	----	----	----	----	----	----	----	----	----

b b 2 , 5 7 4 . 2 6 b b b

このパターン・フィールドを印刷すると次のようになります。

2,574.26

ソース・フィールドは変更されません。数値が 0 より大きいので、条件コード 2 が設定されます。

ソース・フィールドの数値を 00 00 02 6D という負の数に変えて、同じパターンを適用したとすれば、編集結果は以下のようになります。

パターン 1000 100C

40	40	40	40	40	40	F0	4B	F2	F6	40	C3	D9
----	----	----	----	----	----	----	----	----	----	----	----	----

b b b b b b 0 . 2 6 b C R

このパターン・フィールドを印刷すると次のようになります。

0.26 CR

有効数字開始文字が現れると、有効数字標識がオンの状態になり、その結果、先行 0 と小数点が保持されます。負符号コードは有効数字標識には影響を与えないので、負 (credit) の数量を示す CR の文字が印刷されます。

条件コード 1 (数が 0 より小さいことを示す) が設定されます。

EDIT AND MARK (EDMK)

EDIT AND MARK を使用すると、EDIT の機能に加えて、編集結果の中の適切な位置に、通貨記号 (例えばドル記号) を挿入することができます。この例では、上記の EDIT 命令の場合と同じ記憶位置 1200-1203 にあるソースを使用し、記憶位置 1000-100C のパターンも同じで、汎用レジスタ 12 の内容も同じであるものとします。汎用レジスタ 1 (GR1) の以前の内容は意味を持ちません。LOAD ADDRESS 命令を使用して、左側に有効数字が現れない場合に強制印刷される最初の数字位置がセットアップされます。

次の命令を実行します。

LA	1,6(0,12)	強制有効数字のアドレスを GR1 にロードする。
EDMK	0(13,12),X'200'(12)	最初の有効数字のアドレスを GR1 の中に残す。
BCTR	1,0	GR1 の中のアドレスから 1 を引く。
MVI	0(1),C'\$'	GR1 の中のアドレスにドル記号を格納する。

これにより、EDIT の項に示した 2 つの例の結果は以下のようにになります。

パターン 1000 100C

40	5B	F2	6B	F5	F7	F4	4B	F2	F6	40	40	40
----	----	----	----	----	----	----	----	----	----	----	----	----

b \$ 2 , 5 7 4 . 2 6 b b b

このパターン・フィールドを印刷すると次のようになります。

\$2,574.26

条件コード 2 が設定されて、編集した数が 0 より大きいことを示します。

パターン

1000 100C

40	40	40	40	40	5B	F0	4B	F2	F6	40	C3	D9
----	----	----	----	----	----	----	----	----	----	----	----	----

b b b b b \$ 0 . 2 6 b C R

このパターン・フィールドを印刷すると次のようになります。

\$0.26 CR

数が 0 より小さいので、条件コード 1 が設定されます。

MULTIPLY DECIMAL (MP)

記憶位置 1202-1204 に入っている符号付きパック 10 進数 (被乗数) に、記憶位置 500-501 に入っている符号付きパック 10 進数 (乗数) を乗算するものとします。

1202 1204
被乗数

38	46	0D
----	----	----

500 501
乗数

32	1D
----	----

乗算でデータ例外が起きるのを避けるために、まず、被乗数を拡張して、乗数の長さに相当する少なくとも 2 バイト分の 0 を左側に付加する必要があります。被乗数を別のもっと長いフィールドに移すには、ZERO AND ADD を使用します。次のように仮定します。

レジスター 4 の内容は、00 00 12 00。

レジスター 6 の内容は、00 00 05 00。

次の命令を実行します。

ZAP X'100'(5,4),2(3,4)

次のように、記憶位置 1300-1304 に新しい被乗数がセットアップされます。

1300 1304
被乗数 (新)

00	00	38	46	0D
----	----	----	----	----

ここで次の命令を実行したとします。

マシン・フォーマット

命令コード L₁ L₂ B₁ D₁ B₂ D₂

FC	4	1	4	100	6	000
----	---	---	---	-----	---	-----

アセンブラー・フォーマット

命令コード D₁(L₁,B₁),D₂(L₂,B₂)

MP X'100'(5,4),0(2,6)

記憶位置 1300-1304 に、積として 01 23 45 66 0C が入ります。

SHIFT AND ROUND DECIMAL (SRP)

SHIFT AND ROUND DECIMAL (SRP) 命令は、記憶域内の 10 進数を左または右にシフトするために使用できます。数を右にシフトするときは、同時に丸めも行うことができます。

10 進数の左シフト

この例では、記憶位置 FIELD1 の内容を左に 3 桁移動させます。これは、実質的には FIELD1 の内容に 1000 を掛けるのと同じことです。FIELD1 の長さは 6 バイトです。この操作を行うには以下の命令を使用します。

マシン・フォーマット

命令コード L₁ I₃ S₁ B₂ D₂

F0	5	0	****	0	003
----	---	---	------	---	-----

アセンブラー・フォーマット

命令コード S₁(L₁),S₂,I₃

SRP FIELD1(6),3,0

FIELD1 (実行前): 00 01 23 45 67 8C

FIELD1 (実行後): 12 34 56 78 00 0C

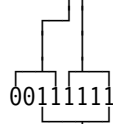
この命令の第 2 オペランドには、シフトの量 (3 桁) を指定します。丸め数字 I₃ は左シフトでは使用されませんが、常に有効な 10 進数字でなければなりません。実行後、結果が 0 より大きいことを示す条件コード 2 が設定されます。

10 進数の右シフト

この例では、記憶位置 FIELD2 の内容を右に 1 桁移動させます。これは、実質的には FIELD2 の内容を 10 で割るのと同じことです。FIELD2 の長さは 5 バイトです。この操作を行うには以下の命令を使用します。

マシン・フォーマット

命令コード	L ₁	I ₃	S ₁	B ₂	D ₂
F0	4	0	****	0	03F



-1 を表す 2 の補数 (6 ビット)

アセンブラー・フォーマット

命令コード	S ₁ (L ₁), S ₂ , I ₃
SRP	FIELD2(5), 64-1, 0

FIELD 2 (実行前): 01 23 45 67 8C

FIELD 2 (実行後): 00 12 34 56 7C

SRP 命令の第 2 オペランド・アドレスは、右へのシフトの数を示す負の値を指定します。この値は、2 の補数形式の 6 ビット値で表されます。

数値 n の 2 の補数 (6 ビット) は、64 - n として指定することができます。この例では、右への 1 桁のシフトが 64 - 1 と表現されます。

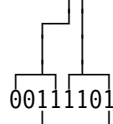
条件コード 2 が設定されます。

10 進数の右シフトと丸め

この例では、記憶位置 FIELD3 の内容を右に 3 桁移動させて丸めます。これは、実質的には 1000 で四捨五入するのと同じことです。FIELD3 の長さは 4 バイトです。

マシン・フォーマット

命令コード	L ₁	I ₃	S ₁	B ₂	D ₂
F0	3	5	****	0	03D



-3 を表す 2 の補数 (6 ビット)

アセンブラー・フォーマット

命令コード	S ₁ (L ₁), S ₂ , I ₃
SRP	FIELD3(4), 64-3, 5

FIELD 3 (実行前): 12 39 60 0D

FIELD 3 (実行後): 00 01 24 0D

シフトの数 (3 桁) は、D₂ フィールドに指定されています。I₃ フィールドには、丸めの数字として 5 が指定されています。シフトにより右へ送り出された最後の数字 (この例では 6) に丸めの数字が加算され、繰り上がりが左へ波及します。この加算では符号は無視されます。

結果が 0 より小さいので、条件コード 1 が設定されます。

10 の可変累乗による乗算

SRP 命令で指定するシフト値は、シフトの方向と量の両方を指定するので、この操作は、第 1 オペランドの 10 進数に、10 をシフト数だけ累乗した値を掛けるのと同じことです。

シフト値を可変数とする場合は、SRP 命令の変位 D₂ の代わりに、B₂ フィールドにシフト値を指定することができます。B₂ が指し示す汎用レジスターには、シフト値 (10 の累乗) を示す符号付き 2 進整数が含まれていることが必要です。

10 の可変累乗数を変更する固定した位取り因数を指定することができます。そのためには、B₂ フィールド (可変部分を指定する汎用レジスター) と、D₂ フィールド (固定部分を指定する変位) の両方を使用します。

SRP 命令は、有効アドレス D₂(B₂) の右端 6 ビットのみを使用し、それを 6 ビットの符号付き 2 進整数として解釈して、前に示した例のように、左または右へのシフトを制御します。

ZERO AND ADD (ZAP)

記憶位置 4500-4502 にある符号付きパック 10 進数を、記憶位置 4000-4004 に移動し、結果のフィールドに 4 個の先行 0 を付加するものとします。また、以下のように仮定します。

レジスター 9 の内容は、00 00 40 00。

記憶位置 4000-4004 の内容は、12 34 56 78 90。

記憶位置 4500-4502 の内容は、38 46 0D。

次の命令を実行します。

マシン・フォーマット						
命令コード	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
F8	4	2	9	000	9	500

アセンブラー・フォーマット	
命令コード	D ₁ (L ₁ ,B ₁),D ₂ (L ₂ ,B ₂)
ZAP	0(5,9),X'500'(3,9)

記憶位置 4000-4004 の内容は、00 00 38 46 0D となります。そして、結果が負であり、オーバーフローがないことを示す条件コード 1 が設定されます。

第 1 オペランドの符号コードおよび数字コードが有効かどうかの検査は行われないので、操作の前に、第 1 オペランドにどのような組み合わせの 16 進数字が入っているも構いません。

16 進浮動小数点命令

(16 進浮動小数点命令の詳しい説明については、第9章、『浮動小数点の概要とサポート命令』を参照してください。)

このセクションで使用している FPR0、FPR2、FPR4、および FPR6 という省略語は、それぞれ、浮動小数点レジスター 0、2、4、および 6 を表しています。

ADD NORMALIZED (AD、ADR、AE、AER、AXR)

ADD NORMALIZED 命令は、2 個の HFP 数を加算して、正規化された結果を浮動小数点レジスターに入れます。加算する 2 つの数は、どちらも、必ずしも加算の前に正規化されている必要はありません。例えば次のように仮定します。

FPR6 の内容は、非正規化数 C3 08 21 00 00 00 00 00 = -82.11₁₆ = 約 -130.06₁₀。

記憶位置 2000-2007 の内容は、正規化数 41 12 34 56 00 00 00 00 = +1.23456₁₆ = 約 +1.14₁₀。

レジスター 13 の内容は、00 00 20 00。

次の命令を実行します。

マシン・フォーマット				
命令コード	R ₁	X ₂	B ₂	D ₂
7A	6	0	D	000

アセンブラー・フォーマット	
命令コード	R ₁ ,D ₂ (X ₂ ,B ₂)
AE	6,0(0,13)

この命令は、2 個のオペランドの短精度加算を以下のように行います。

2 つの数の指数部 (43 と 41) が比較されます。記憶域内の数の指数部の方が 2 だけ小さいので、その数は 16 進数の 2 桁分だけ右にシフトされます。右側には保護桁が 1 個保持されます。そして、2 つの数の小数部が代数的に加算されます。

	小数部	GD ¹
FPR6	-43 08 21 00	
記憶域からのシフト済みの数	+43 00 12 34	5
中間和	-43 08 0E CB	B
左シフト後の和	-42 80 EC BB	

1 保護桁

中間和は正規化されていないので、その数が左にシフトされて、正規化された HFP 数 -80.ECBB₁₆ = 約 -128.92₁₀ となります。符号と指数部を組み合わせると、結果は C2 80 EC BB となり、これが FPR6 の左半分と置き換わります。FPR6 の右半分と記憶位置 2000-2007 の内容は変更されません。結果が 0 より小さいことを示す条件コード 1 が設定されます。

長精度命令 AD を使用したとすれば、FPR6 は C2 80 EC BA A0 00 00 00 となります。この例の場合、長精度命令を使用すれば、精度の低下を避けることができるという点に注意してください。

ADD UNNORMALIZED (AU、AUR、AW、AWR)

ADD UNNORMALIZED 命令は、最終結果が正規化されないという点以外は、ADD NORMALIZED 命令と同じです。例えば、ADD NORMALIZED の例と同じオペランドを使用して、次の短精度命令を実行したとします。

マシン・フォーマット				
命令コード	R ₁	X ₂	B ₂	D ₂
7E	6	0	D	000

アセンブラー・フォーマット	
命令コード	R ₁ ,D ₂ (X ₂ ,B ₂)
AU	6,0(0,13)

次のように、2つの数が加算されます。

	小数部	GD ¹
FPR6	-43 08 21 00	
記憶域からのシフト済みの数	+43 00 12 34	5
<hr/>		
中間和	-43 08 0E CB	B

1 保護桁

保護桁は加算に関与しますが、破棄されます。正規化されていない和が、FPR6の左半分と置き換わります。結果が0より小さいので、条件コード1が設定されます。

FPR6に入っている切り捨てられた結果(C3 08 0E CB 00 00 00 00)を、短精度の正規化された加算の結果と比べると、有効数字が1桁失われていることが分かります。

COMPARE (CD, CDR, CE, CER)

FPR4の内容が、43 00 00 00 00 00 00 (ゼロ)で、FPR6の内容が、35 12 34 56 78 9A BC DE (正の数)であるとします。この2つのレジスタの内容を、長精度 COMPARE 命令を使用して比較します。

マシン・フォーマット
命令コード R₁ R₂

29	4	6
----	---	---

アセンブラー・フォーマット
命令コード R₁,R₂

CDR	4,6
-----	-----

2つの指数部を一致させるために、指数部の小さい方の数(つまりレジスタ FPR6に入っている数)が、16進数 43 - 35 (10進数 67 - 53)、つまり14桁だけ右にシフトされます。シフトの後の数は、43 00 00 00 00 00 00 00で、保護桁は1です。したがって、この2つの数を比較した結果、条件コード1が設定されます。これは、FPR4内のオペランド1がFPR6内のオペランド2より小さいことを示します。

この例で、第2オペランドの指数部が35ではなく34であったとすれば、FPR6の内容は34 12 34 56 78 9A BC DEとなり、このオペランドは15桁右にシフトされ、小数桁および保護桁はすべて0のまま残されます。そして、両者が等しいことを示す条件コード0が設定されます。この例は、数が正規化されないかまたは0である場合に、指数部または小数部の異なる2つのHFP数が、比較の結果等しいと見なされることがあるということを示しています。

正規化されていないHFP数の比較のもう1つの例として、41 00 12 34 56 78 9A BCは、3F 12 34 56 78 9A BC 0Xの形式のすべての数(Xは任意の16進数字)と等しいものと見なされます。COMPARE命令を実行すると、右端の2桁が2桁右にシフトされ、0が保護桁となり、したがってXは比較に関与しないこととなります。

ただし、正規化された2つのHFP数を比較した場合は、2つの数が等しいと見なされる関係は1つだけです。つまり、一方の数のすべての桁の数字が、もう一方の数の対応する桁の数字と同じでなければなりません。

DIVIDE (DD, DDR, DE, DER)

第1オペランド(被除数)がFPR2にあり、第2オペランド(除数)がFPR0にあるとします。両オペランドが短精度フォーマットであれば、次の命令により、結果の商がFPR2に戻されます。

マシン・フォーマット
命令コード R₁ R₂

3D	2	0
----	---	---

アセンブラー・フォーマット
命令コード R₁,R₂

DER	2,0
-----	-----

被除数がFPR2にあり除数がFPR0にある場合の短精度HFP除算の例を、幾つか以下に挙げておきます。ケースAの場合、被除数と置き換わる結果の値は、以下の手順で求められます。

```

              7.2522F
    .123400 | .821000
              7F6C00
              -----
              2A400 0
              24680 0
              -----
              5D80 00
              5B04 00
              -----
              27C 000
              246 800
              -----
              35 8000
              24 6800
              -----
              11 18000
              11 10C00
              -----
              7400
  
```

ケース	実行前の FPR2 (被除数)	FPR0 (除数)	実行後の FPR2 (商)
A	-43 082100	+43 001234	-42 72522F
B	+42 101010	+45 111111	+3D F0F0F0
C	+48 30000F	+41 400000	+47 C0003C
D	+48 30000F	+41 200000	+48 180007
E	+48 180007	+41 200000	+47 C00038

ケース C は、ある数を 4.0 で除算する場合を示しています。ケース D では同じ数を 2.0 で割り、ケース E ではケース D の結果をさらに 2.0 で割っています。ケース C および E の結果は、右端の 16 進数字が異なります。これは、結果の切り捨ての影響の一例を示しています。

HALVE (HDR, HER)

HALVE の結果は、除数を 2.0 とする HFP DIVIDE と同じです。FPR2 に、長精度数 +48 30 00 00 00 00 00 0F が含まれているとします。次の HALVE 命令を実行すると、結果として +48 18 00 00 00 00 00 07 が FPR2 に入れられます。

マシン・フォーマット
命令コード R1 R2

24	2	2
----	---	---

アセンブラー・フォーマット
命令コード R1,R2

HDR 2,2

MULTIPLY (MD, MDR, MDE, MDER, MXD, MXDR, MXR)

この例では、FPR0 および FPR2 に以下の長精度オペランドが入っているものとします。

FPR0: -33 606060 60606060
FPR2: -5A 200000 20000020

次の命令により、長精度の積が求められます。

マシン・フォーマット
命令コード R1 R2

2C	0	2
----	---	---

アセンブラー・フォーマット
命令コード R1,R2

MDR 0,2

オペランドがまだ正規化されていない場合は、この命令はまず正規化を実行します。次に、中間結果として、オペランドの 14 桁の 16 進小数部を乗算することによって求められる全 28 桁の 16 進数から成る積の小数部と、適切な符号、およびオペランドの指数部から 64 (16 進数 40) を引いた値の和である指数部を生成します。

小数部の乗算は以下のように行われます。

```

              .60606060606060
              .20000020000020
              -----
              C0C0C0C0C0C0C0
              C0C0C0C0C0C0C0
              C0C0C0C0C0C0C0
              -----
              .0C0C0C181818241818180C0C0C00
  
```

この小数部に符号と指数部を付加すると、以下のようになります。

+4D 0C0C0C 18181824 1818180C 0C0C00

この中間積には先行 0 が 1 つ付いているので、正規化が行われます。そして、切り捨て後の最終結果が FPR0 に入れられます。

+4C C0C0C1 81818241

16 進浮動小数点数の変換

以下の例は、2 進固定小数点数 (32 ビット符号付き 2 進整数) と正規化された HFP 数との間で変換を行う方法の 1 つを示しています。負の数を表すときに、符号付き 2 進整数には 2 の補数形式、そして HFP 数の小数部には符号付き絶対値形式という異なる表現が使用されるため、変換が必要とされます。

固定小数点から 16 進浮動小数点へ

ここでは、32 ビット符号付き 2 進整数の左端の 1 ビットを逆転させる方法を使用します。これは、その数に 2^{31} を加え、結果が正であると見なすことと同じです。これにより、 $2^{31} - 1$ から -2^{31} までの範囲内の符号付き整数が、 $2^{32} - 1$ から 0 までの範囲内の符号なし整数に変更されます。長精度 HFP フォーマットへの変換後に、再び 2^{31} の値が引かれます。

汎用レジスタ 9 (GR9) に、整数 -59 が 2 の補数形式で入っているものとします。

```
GR9:    FF FF FF C5
```

さらに、記憶域内に 2 つの 8 バイト・フィールドがあるものとします。1 つは、一時記憶域として使用するための TEMP で、もう 1 つは、次のフォーマットの浮動小数点定数 2^{31} が入っている TWO31 です。

```
TWO31:  4E 00 00 00 80 00 00 00
```

これは、正規化されていない長精度 HFP 数で、指数部 4E が付いており、数の右側の小数点 (16 進小数点) に対応しています。

この変換には、以下の命令シーケンスを使用します。

	結果
X 9,TWO31+4	GR9: 7FFF FFC5
ST 9,TEMP+4	TEMP: xxxx xxxx 7FFF FFC5
MVC TEMP(4),TWO31	TEMP: 4E00 0000 7FFF FFC5
LD 2,TEMP	FPR2: 4E00 0000 7FFF FFC5
SD 2,TWO31	FPR2: C23B 0000 0000 0000

EXCLUSIVE OR (X) 命令は、汎用レジスタ 9 の左端のビットを逆転させます。そのために、定数の右半分が、左端の 1 のビットのソースとして使用されます。その次の 2 つの命令は、修正された数を、正規化されていない長精度 HFP フォーマットでアセンブルします。そのために、定数の左半分が、正符号、指数部、および、小数部の先行 0 として使用されます。LOAD (LD) は、その数を変更せずに、浮動小数点レジスタ 2 に入れます。SUBTRACT NORMALIZED (SD) 命令は、最終的な 2 つのステップとして、HFP 形式の 2^{31} を引き、そして結果を正規化します。

16 進浮動小数点から固定小数点へ

ここで述べる手順は、基本的には前記の手順を逆にしたものです。それに加えて、2 つの事項を考慮に入れる必要があります。その 1 つは、HFP 数は正確な整数ではないことがあるという点です。右側の余分な 16 進数字を切り捨てるには、数を、最終結果に必要な桁よりさらに 1 桁右にシフトすることが必要であり、その結果、単位桁が保護桁の位置を占めることになります。第 2 の考慮事項は、HFP 数が、32 ビット符号付き 2 進整数として表現できる数の範囲から外れていないかどうかを検査しなければならないことがあるという点です。

浮動小数点レジスタ 6 に、 $59.25_{10} = 3B.4_{16}$ の数が正規化された形式で入っているとします。

```
FPR6:    42 3B 40 00 00 00 00 00
```

さらに、記憶域内に 3 つの 8 バイト・フィールドがあるものとします。それは、一時記憶域として使用する

TEMP と、次のフォーマットの定数 2^{32} (TWO32) および定数 2^{31} (TWO31R) です。

```
TWO32:   4E 00 00 01 00 00 00 00  
TWO31R:  4F 00 00 00 08 00 00 00
```

定数 TWO31R は、前の例の定数 TWO31 より 1 桁右にシフトされています。これは、単位桁を強制的に保護桁に置くためです。

以下の命令シーケンスは、整数の切り捨て、範囲テスト、および符号付き 2 進整数への変換を行い、結果を汎用レジスタ 8 (GR8) に入れます。

		結果
SD	6,TWO31R	FPR6: C87F FFFF C500 0000
BC	11,OVERFLOW	結果が 0 またはそれより 大きい場合は、オーバーフ ロー・ルーチンへブランチ
AW	6,TWO32	FPR6: 4E00 0000 8000 003B
BC	4,OVERFLOW	結果が 0 より小さい場合 は、オーバーフロー・ルー チンへブランチ
STD	6,TEMP	TEMP: 4E00 0000 8000 003B
XI	TEMP+4,X'80'	TEMP: 4E00 0000 0000 003B
L	8,TEMP+4	GR8: 0000 003B

SUBTRACT NORMALIZED (SD) 命令は、数の小数部を、TWO31R と揃うまで右にシフトします。その結果、小数桁の 4 が保護桁の右に落ちて消滅します。そして、残りの数字から 2^{31} を引いた結果が正規化されます。結果は 0 より小さくなるはずですが、そうならない場合は、元の数が正の方向に大きすぎたこととなります。最初の BRANCH ON CONDITION (BC) により、このテストが行われています。

ADD UNNORMALIZED (AW) 命令は、 2^{32} を加算します。 2^{31} は前の減算の訂正のためであり、もう 1 つの 2^{31} はすべて正の範囲に変更するためです。2 番目の BC は、結果が 0 より小さい (つまり元の数がある負の方向に大きすぎる) かどうかをテストします。そして、STORE (STD) 命令により、正規化されていない結果が一時記憶域に入れます。EXCLUSIVE OR (XI) 命令は、 2^{31} を引くことによりこの 2 進整数の左端ビットを逆転させ、その結果符号なしの数が符号付きのフォーマットに変換されます。最終結果は GR8 にロードされます。

マルチプログラミングとマルチプロセッシングの例

マルチプログラミングまたはマルチプロセッシング環境において、同じ記憶位置を共用する複数のプログラムが実行されているときに、例えば、1 つのプログラムが共通記憶域内にフラグ・ビットを設定し、それを他のプログラムにテストさせることがあります。マルチプログラミング環境では、AND (NI または NC)、EXCLUSIVE OR (XI または XC)、および OR (OI または OC) 命令を使用して、フラグ・ビットを設定できます。しかし、

マルチプロセッシング構成では、複数の CPU が、同じ記憶位置内で同時にデータの取り出し、修正、および格納を行うことができるため、これらの命令を使用するとプログラム・ロジック・エラーが起きることがあります。

即値 OR を用いたプログラム障害の例

2 つの独立したプログラムが、記憶域内の同じバイトの異なるビットを 1 に設定しようとしたとします。次の例は、これらのプログラムが 2 つの異なる CPU で同時に実行された場合に、即値 OR 命令 (OI) がどのように失敗する可能性があるかを示しています。ここでは、起き得るエラー状況のうちの 1 つを取り上げて説明します。

CPU A での OI FLAGS,X'01' 命令の実行	FLAGS	CPU B での OI FLAGS,X'80' 命令の実行
FLAGS X'00' の取り出し	X'00'	FLAGS X'00' の取り出し
X'01' と X'00' の OR 結合	X'00'	X'80' と X'00' の OR 結合
X'01' を FLAGS に格納	X'80'	X'80' を FLAGS に格納
両方の更新後、FLAGS の値は X'81' になります。		

ここに示されている問題点は、CPU A で実行されている OI により格納された値が、CPU B により格納された値をオーバーレイするという点です。X'80' のフラグ・ビットが誤ってオフにされてしまったため、データは無効になっています。

この問題および類似の問題を回避するために、COMPARE AND SWAP 命令が提供されています。

条件付きスワップ命令 (CS、CDS)

マルチプログラミングまたはマルチプロセッシング環境で、COMPARE AND SWAP (CS) および COMPARE DOUBLE AND SWAP (CDS) 命令を使用すると、カウンタ、フラグ、制御ワード、およびその他の共通記憶域へのアクセスを直列化することができます。

以下に示す COMPARE AND SWAP および COMPARE DOUBLE AND SWAP 命令の使用例は、これらの命令がどのような局面に対応することを意図しているかを示しています。ここで注意すべき重要な点

は、これらの例が示しているのは、CPU で割り込みが使用可能にされているときに (つまりマルチプログラミング環境で) プログラムが実行できる機能、またはマルチプロセッシング構成で実行されているプログラムが実行できる機能であるということです。つまり、以下に示すルーチンは、CPU が使用可能にされている間は、プログラムがいつでも記憶域の内容を変更できるようにします。これは、このルーチンが同じ CPU 上の別のプログラムにより割り込まれることがあったとしても、また、他の CPU が同じ記憶位置を同時に更新する可能性があったとしても、プログラムが記憶域の内容を変更できることを意味します。

COMPARE AND SWAP 命令は、まず記憶位置の値を検査し、それがプログラムの予期している値である場合のみ、その値を変更します。通常、これは前に取り出されている値です。記憶位置の値がプログラムが予期している値ではない場合は、その記憶位置は変更されません。代わりに、プログラムがループバックして再試行できるようにするための準備として、その記憶位置の現在値が汎用レジスターにロードされます。COMPARE AND SWAP の実行中は、他のどの CPU も、指定された記憶位置への格納アクセスまたはインターロック更新アクセスを行うことはできません。

2 つ以上の CPU による同一の共通域フィールドの更新が常に正常に実行できるようにするには、すべての更新をインターロック更新参照を使用して行う必要があります。他の CPU により実行されている OR IMMEDIATE が原因で COMPARE AND SWAP が失敗する場合の例については、COMPARE AND SWAP の『プログラミング上の注意』を参照してください。

単一ビットの設定

以下の命令シーケンスは、COMPARE AND SWAP 命令を使用して、記憶域内の 1 つのビットを 1 に設定する方法を示しています。記憶域内に「WORD」というワードがあり、その最初のバイトに 8 個のフラグ・ビットが含まれているものとします。

LA	6,X'80'	OR	結合するビットを GR6 に入れる。
SLL	6,24	OR	24 ビット左にシフトして、結合するバイトを WORD 内のフラグ・ビットの位置にそろえる。
L	7,WORD	現在	現在のフラグ値を取り出す。
RETRY	LR 8,7	OR	フラグを GR8 にロードする。
	OR 8,6	OR	ビットを 1 に設定する。
	CS 7,8,WORD	現在	現在のフラグが変更されていないならば、新しいフラグを格納し、変更されていれば、現在のフラグ値を再取り出す。
BC	4,RETRY	OR	新しいフラグが格納されなかった場合は、再試行する。

COMPARE AND SWAP 命令のフォーマットは以下のとおりです。

マシン・フォーマット

命令コード R₁ R₃ S₂

BA	7	8	****
----	---	---	------

アセンブラー・フォーマット

命令コード R₁,R₃,S₂

CS 7,8,WORD

この COMPARE AND SWAP 命令は、第 1 オペランド (現在のフラグ値を含む汎用レジスター 7) を、記憶域内の第 2 オペランド (WORD) と比較します。その間、この COMPARE AND SWAP 命令を実行している CPU 以外の CPU は、指定した記憶位置への格納アクセスまたはインターロック更新アクセスを行うことはできません。

比較が成功し、フラグ・ビットが取り出し後に変更されていないことが示された場合は、汎用レジスター 8 内の変更済みコピーが WORD に格納されます。フラグが変更されている場合は、比較は失敗し、新しいフラグ値が汎用レジスター 7 にロードされます。

条件付きブランチ (BC) 命令は、条件コードをテストし、COMPARE AND SWAP 命令が比較の失敗 (条件コード 1) を示している場合は、フラグ変更命令を再実行します。COMPARE AND SWAP 命令が成功している場合は (条件コード 0)、フラグには有効なデータが含まれており、プログラムはループから抜け出します。

RETRY へのブランチが行われるのは、他のプログラムが WORD の内容を変更した場合のみです。このタイプのループは、典型的な「ビット・スピン」ループとは異なります。ビット・スピン・ループでは、プログラムはビットが変更されるまでループを続けます。しかし、この例のプログラムは、個々の反復の間に値が変化したときのみループを続けます。複数の CPU が、例に示した命令シーケンスを使用して、1 つの記憶位置を同時に変更しようとした場合、1 つの CPU は 1 回目の試行でフォールスルーし、次の CPU は 1 回ループするという方法で、最終的にはすべての CPU が成功します。

カウンターの更新

以下の例では、32 ビット・カウンターが正しく更新されるようにするために、プログラムで COMPARE AND SWAP 命令を使用して 32 ビット・カウンターを更新します。まず、このカウンターを含むワードを汎用レジスタ 7 にロードすることにより、カウンターのオリジナルの値が求められます。次に、変更可能なコピーを用意するためにその値が汎用レジスタ 8 に移され、その変更可能なコピーに汎用レジスタ 6 (カウンターの増分値を含む) が加算されて、更新済みのカウンター値が求められます。この COMPARE AND SWAP 命令は、有効なカウンターを確実に格納するために使用されています。

カウンターを更新するプログラムは、条件コードを調べて結果を確認します。条件コード 0 は更新が成功したことを示し、プログラムは先へ進むことができます。プログラムが元の値をロードしてから COMPARE AND SWAP 命令を実行するまでの間に、カウンター値が変更されている場合は、命令の実行により、汎用レジスタ 7 に新しいカウンター値がロードされており、更新の失敗を示す条件コード 1 が設定されます。その場合は、プログラムは、COMPARE AND SWAP 命令の実行による更新が成功するまで、更新シーケンスを繰り返す必要があります。

上記の手順を行うには、以下のような命令シーケンスを実行します。

LA	6,1	増分値 (1) を GR6 に入れる。
L	7,CNTR	元のカウンター値を GR7 に入れる。
LOOP LR	8,7	変更用のコピーを GR8 の中にセットアップする。
AR	8,6	コピーに増分値をコピーに加える。
CS	7,8,CNTR	記憶域内でカウンターを更新する。
BC	4,LOOP	元の値が変更されている場合は、新しい値に更新する。

次の例は、2 つの CPU (A および B) がこの命令シーケンスを同時に実行した場合を示しています。どちらの CPU も CNTR に 1 を加算しようとしています。

CPU A		CNTR	CPU B		説明
GR7	GR8		GR7	GR8	
		16			
16	16				CPU A が CNTR から GR7 と GR8 にロードする。
			16	16	CPU B が CNTR から GR7 と GR8 にロードする。
				17	CPU B が GR8 に 1 を加算する。
	17				CPU A が GR8 に 1 を加算する。
		17			CPU A が CS を実行する。一致するので格納する。
			17		CPU B が CS を実行する。一致しないため、GR7 が CNTR の値に変更される。
				18	CPU B が GR7 から GR8 にロードし、GR8 に 1 を加算する。
		18			CPU B が CS を実行する。一致するので格納する。

POST と WAIT のバイパス

POST バイパス・ルーチン

以下のルーチンを使用すると、監視プログラムの WAIT および POST ルーチンが COMPARE AND SWAP を使用してイベント制御ブロック (ECB) を操作している場合に、対応する WAIT がまだ実行されていない限り、MVS/ESA で使用される SVC POST がバイパスされるようにすることができます。

初期条件は以下のとおりです。

GR0 には POST コードが入っている。

GR1 には ECB のアドレスが入っている。

GR5 には、40 00 00 00₁₆ が入っている。

HSPPOST	OR	0,5	GR0 のビット 1 を 1 に設定する。
L		3,0(1)	GR3 = ECB の内容。
LTR		3,3	ECB に「WAITING」のマークが付いているか。
BC		4,PSVC	そうであれば、POST SVC を実行する。
CS		3,0,0(1)	そうでなければ POST コードを格納する。
BC		8,EXITHP	続行する。
PSVC	POST	(1),(0)	ECB アドレスは GR1 があり、POST コードは GR0 にある。

EXITHP [任意の命令]

GR0 の内容のビット 1 がすでに 1 に設定されていて、ECB に「WAITING」のマークが付いていないときに ECB に 0 が入っているとすれば、上記の HSPPOST ルーチンの代わりに以下のルーチンを使用することができます。

```

HSPOST SR 3,3
        CS 3,0,0(1)
        BC 8,EXITHP
        POST (1),(0)
EXITHP [任意の命令]

```

WAIT バイパス・ルーチン

BYPASS POST に対応する BYPASS WAIT 機能は CS 命令を使用しませんが、この後に示す FIFO LOCK/UNLOCK ルーチンでは CS 命令の使用が前提とされます。

```

HSWAIT TM 0(1),X'40'
        BC 1,EXITHW
EXITHW WAIT ECB=(1)
        [任意の命令]

```

ビット 1 が 1 であれば、ECB はすでに通知されている。出口にブランチする。

ロック/アンロック

ダブルワードより大きい共通記憶域を更新する必要があるときは、通常、一度に 1 つのプログラムのみがその共通域を更新するようにするために、特殊なインターロックを用意することが必要です。このような記憶域を、逐次再使用可能リソース (SRR) と呼びます。

一般に、リストを更新するときはもちろんのこと、単にリストをスキャンするときさえ、まずリストを「凍結」しておかない限り、安全確実に操作を行うことはできません。しかし、一定の制限された状況下では、

COMPARE AND SWAP および COMPARE

DOUBLE AND SWAP 命令を使用して、キューイングおよびリスト操作を行うことができます。特に重要なのは、ロック/アンロック機能を実行する能力と、LIFO または FIFO のどちらの方式でも、競合を解決するための十分なキューイングを提供する能力を備えていることです。したがって、どれほど複雑な SRR を更新するときでも、このロック/アンロック機能をインターロック・メカニズムとして使用できます。

ロック/アンロック機能は、SRR に関連した「ヘッダー」に基づいて実行されます。ヘッダーは、SRR が解放状態か使用中状態かを判別するための共通の開始点であると同時に、競合が起きたときにキューイングを要求するためにも使用されます。競合は、WAIT および POST を使用して解決されます。一般的なプログラミング手法では、「ロックされた」SRR に出会ったプログ

ラムは、自身が待機している ECB のアドレスを示す「マークを壁に残す」ことが必要とされます。「アンロックする」プログラムは、そのマークを見て、その ECB に通知して、待機中のプログラムが続行できるようにします。以下に示す 2 つの例では、特定の SRR を使用するすべてのプログラムが LIFO キューイング方式または FIFO キューイング方式のいずれか一方のみを使用しなければならず、両方の混用はできません。もっと複雑なキューイングが必要な場合は、以下の 2 つの方法の 1 つを使用して、SRR 用のキューをロックすることをお勧めします。

競合に対する LIFO キューイングを使用したロック/アンロック

ヘッダーは 1 ワードから成っています。このワードは、ワード境界に位置合わせされた 4 バイトのフィールドです。このワードには、0、正の値、または負の値が含まれています。

- 0 の値は、逐次再使用可能リソース (SRR) が解放状態にあることを示します。
- 負の値は、SRR が使用中であるが、ほかにはその SRR を待っているプログラムはないことを示します。
- 正の値は、SRR が使用中であり、その SRR を待っているプログラムが 1 つ以上あることを示します。待機中の各プログラムは、チェーン・リスト内の 1 つのエレメントで識別されます。ヘッダー内の正の値は、リストに追加された最も新しいエレメントのアドレスです。

各エレメントはそれぞれ 2 ワードから成っています。最初のワードは ECB として使用され、2 番目のワードは、リスト内の次のエレメントを指すポインターとして使用されます。ポインター内に負の値がある場合は、エレメントがリスト内の最後のエレメントであることを示します。このエレメントが必要とされるのは、プログラムが、SRR がロックされていることを検出し、リストに登録されることを必要としている場合のみです。

以下の表は、LIFO LOCK および LIFO UNLOCK ルーチンについてとられるアクションを示しています。表の後には、コードを使用可能にして表に記述されているアクションを実行できるようにするためのルーチンを示してあります。

機能	アクション		
	ヘッダーは 0	ヘッダーは 正の値	ヘッダーは 負の値
LIFO LOCK (入ってくる エレメントは 記憶位置 A に ある)	SRR は解放され ている。ヘッダ ーを負の値に設 定する。この SRR を使用する。	SRR は使用中。ヘッダーの内容を 記憶位置 A+4 に格納する。アド レス A をヘッダーに格納する。 WAIT を出す。ECB は記憶位置 A にある。	
LIFO UNLOCK	Error	この SRR を待つ ているプログラ ムがある。「最 後に入った」エ レメントからポ インターをヘッ ダーに移す。 POST を出す。 ECB は「最後に に入った」エレ メント内にある。	リストは空であ る。ヘッダーに 0 を格納する。 SRR は解放され る。

LIFO LOCK ルーチン:

初期条件は以下のとおりです。

GR1 には、入ってくるエレメントのアドレスが入
っている。

GR2 には、ヘッダーのアドレスが入っている。

LLOCK	SR	3,3	GR3 = 0
	ST	3,0(1)	ECB を初期設定する。
	LNR	0,1	GR0 = 負の値
TRYAGN	CS	3,0,0(2)	ヘッダーが 0 であれば、 ヘッダーを負の値に設定 する。
	BC	8,USE	ヘッダーが 0 か。
	ST	3,4(1)	そうでなければ、ヘッダ ーの値を、入ってくるエ レメント内のポインター に格納する。
	CS	3,1,0(2)	入ってくるエレメントの アドレスをヘッダーに格 納する。
	LA	3,0(0)	GR3 = 0
	BC	7,TRYAGN	ヘッダーは更新されてい るか。
	WAIT	ECB=(1)	そうであれば、リソースが 解放されるのを待つ。ECB は入ってくるエレメン トの中にある。

USE [任意の命令]

LIFO UNLOCK ルーチン:

初期条件は以下のとおりです。

GR2 には、ヘッダーのアドレスが入っている。

LUNLK	L	1,0(2)	GR1 = ヘッダーの内容 ヘッダーに負の値が含 まれているか。
A	LTR	1,1	
	BC	4,B	そうでなければ、「最後 に入った」エレメントから ポインターをロードし、 それをヘッダーに格納する。
	L	0,4(1)	
	CS	1,0,0(2)	
	BC	7,A	ヘッダーは更新されてい るか。
	POST	(1)	そうであれば、「最後に入 った」エレメントを通知す る。
	BC	15,EXIT	続行する。
B	SR	0,0	ヘッダーには負の値が含 まれている。ヘッダーを解放 して続行する。
	CS	1,0,0(2)	
	BC	7,A	
	EXIT	[任意の命令]	

記憶域オペランドの一貫性に関する規則がないとすれ
ば、LUNLK の位置にある LOAD 命令 L 1,0(2) は、
CS 1,1,0(2) でなければならぬという点に注意してく
ださい。この一貫性規則では、LOAD 命令は、ワード
境界に位置合わせされた 4 バイト・オペランドを取り出
す必要があります。これは、別の CPU がその取り出し
中のワードを変更する場合に、少なくともワードの一貫
性を維持する操作に従っていけば、そのワードの完全に
新しい値かまたは完全に古い値が取得され、両者が混在
する値が取得されることのないようにするためです。
(5-86ページの『記憶域オペランドの一貫性』を参照し
てください。)

競合に対する FIFO キューイングを使用した ロック/アンロック

ヘッダーには、常に最も新しく入ってきたエレメントの
アドレスが含まれています。初期設定の時点では、ヘッ
ダーには通知された ECB のアドレスが含まれていま
す。逐次再使用可能リソース (SRR) を使用する各プロ
グラムは、競合が起きるかどうかに関係なく、エレメン
トを 1 つずつ用意する必要があります。そして、各プロ
グラムは、そのエレメントのアドレスをヘッダーに入力し、同時に、それまでヘッダーに含まれていたアドレス
を除去します。したがって、SRR を使用しようとして
いる 1 つのプログラムについて、「入力されるエレメン
ト」と「除去されるエレメント」と呼ばれる 2 つのエレ
メントが関連付けられています。あるプログラムの「入
力されるエレメント」は、その直後に続くプログラムに
とっては「除去されるエレメント」になります。その結
果、各プログラムは除去されるエレメント上で待機し、
SRR を使用し、そして入力されるエレメントを通知し
ます。

競合が起きないとき、つまり、最初のプログラムが
SRR の使用を終えるまで 2 番目のプログラムがその

SRR を使用しようとしないうちは、2 番目のプログラムの WAIT が起こる前に最初のプログラムの POST が発生します。この場合は、前のセクションで述べた POST バイパスおよび WAIT バイパス・ルーチンを適用できます。説明を簡素化するために、この 2 つのルーチンは、個々の命令ではなく名前のみで示すことにします。

この例では、エレメントは、単一のワード、つまり 1 つの ECB のみでよいものとします。しかし、実際の運用では、エレメントをもっと大きくして、プログラム ID に加えて前のエレメントを指すポインタを含めることもできます。この種の情報は、エラー状況が発生したときに、ヘッダーから始めてエレメントのリストをたどり、現在 SRR を保持しているプログラムを見つけるために役立ちます。

プログラムが用意したエレメントは、次のプログラムがロックしようとするまで、ヘッダーにより指し示されたままになっているという点に注意してください。したがって、一般に、入力されるエレメントをプログラムが再使用することはできません。ただし、除去されるエレメントは使用可能なので、各プログラムは、1 つのエレメントを放棄して新しいエレメントを獲得します。特定のプログラムが、SRR の 1 回の使用時に除去したエレメントは、そのプログラムが次回にその SRR を要求するときのエントリー・エレメントとして使用されるものと予期されています。

エレメントは、1 つのプログラムから次のプログラムへと交換されていくので、プログラムの終了時に解放され再利用される記憶域からは、エレメントを割り振ることはできないという点に注意してください。プログラムは、A-49ページの『フリー・プールの操作』で述べるルーチンを使用して、最初のエレメントを取得し、最後のエレメントを解放するものと予期されています。

以下の表は、FIFO LOCK および FIFO UNLOCK についてとられるアクションを示しています。

機能	アクション
FIFO LOCK (入ってくるエレメントは記憶位置 A にある)	アドレス A をヘッダーに格納する。 WAIT を出す。ECB は、ヘッダーの古い内容が指す記憶位置にある。
FIFO UNLOCK	POST を出す。ECB は記憶位置 A にある。

以下のルーチンは、コードを使用可能にして表に記述されているアクションを実行できるようにするためのものです。

FIFO ロック・ルーチン:

初期条件は以下のとおりです。

GR3 には、ヘッダーのアドレスが入っている。

GR4 には、このプログラムが現在所有しているエレメントのアドレス (A) が入っている。このエレメントは、入力されるエレメントになります。

FLOCK	LR	2,4	GR2 には、入力されるエレメントのアドレスが入っている。
	SR	1,1	GR1 = 0
	ST	1,0(2)	ECB を初期設定する。
	L	1,0(3)	GR1 = ヘッダーの内容
TRYAGN	CS	1,2,0(3)	ヘッダーにアドレス A を入力し、ヘッダーの古い内容は GR1 に入れておく。GR1 には除去されたエレメントのアドレスが入っている。
	BC	7,TRYAGN	
	LR	4,1	除去されたエレメントが、新たな現在所有されているエレメントになる。
	HSWAIT		BYPASS WAIT ルーチンを実行する。ECB がすでに通知されている場合は続行する。そうでない場合は WAIT を出す。GR1 には ECB のアドレスが含まれている。

USE [任意の命令]

FIFO アンロック・ルーチン:

初期条件は以下のとおりです。

GR2 には、FLOCK ルーチンで得られた除去されるエレメントのアドレスが含まれている。

GR5 には、40 00 00 00₁₆ が入っている。

FUNLK	LR	1,2	入力されるエレメントのアドレスを GR1 に入れる。GR1 = 通知する ECB のアドレス
	SR	0,0	GR0 = 0。GR0 には 0 の通知コードが入っている。
	OR	0,5	GR0 のビット 1 を 1 に設定する。
	HSPOST		BYPASS POST ルーチンを実行する。ECB が待たれていなければ、通知済みのマークを付け、続行する。待たれている場合は、通知する。

CONTINUE [任意の命令]

フリー・プールの操作

プログラムが、ロック/アンロック・ルーチンを使用せずに、フリー・リストに項目を追加したり、そこから項目を削除したりする必要がでてくることが予想されます。これは、主として、ロック/アンロック・ルーチンで、キューイングのための記憶域エレメントが必要になったり、作業用記憶域が必要になる場合があるからです。前記で説明したロック/アンロック・ルーチンの場合、ロック・ルーチンは同時に複数実行できますが、アンロック・ルーチンは一度に 1 つしか実行できません。このような状況下では、リストに対する複数の追加と 1 つの削除がすべて同時に起こることはありますが、同時に複数の削除が起こることはありません。フリーの記憶域バッファを含ポインターのチェーンの場合は、追加と共に、複数の削除を同時に行うことができます。この場合に、COMPARE AND SWAP 命令を使用して除去を行うと、どうしてもある程度の危険にさらされることになります。

LIFO ロック/アンロックの例で使用したタイプのチェーン・リストを考えてみましょう。最初の 2 つのエレメントは、それぞれ記憶位置 A と B にあるものとします。あるプログラムが最初のエレメントを除去しようとし、LUNLK ルーチンの 4 番目と 5 番目の命令の間で割り込みが発生すると、その割り込まれたプログラムの実行が再開される時には、リストが変更されて最初の 2 つのエレメントがエレメント A および C に変わっている可能性があります。その後で、COMPARE AND SWAP 命令が値 B をヘッダーに格納することに成功すると、リストは破棄されてしまいます。

ヘッダーに、リストにエレメントが追加された回数を示すカウンターを付加すれば、このようなリストの破棄が起こる可能性は、ほとんどゼロにまで減少させることができます。32 ビットのカウンターを使用すれば、以下に示すイベントが以下の順序のとおりが発生しない限り、リストが破棄されないことが保証されます。

1. アンロック・ルーチンが、最初のエレメントからポインターを取り出してから、ヘッダーを更新するまでの間に割り込みを受ける。
2. リストが操作され (上記 1 に示したエレメントの削除も含む)、そして、リストに対して正確に 2^{32} 回 (または 2^{32} の整数倍の回数) の追加が行われる。実際の環境では、これは何日かかけて行われるという点に注意してください。
3. 1 に示したエレメントがリストに追加される。

4. 1 で割り込まれたアンロック・ルーチンの実行が再開される。

以下に示すルーチンは、ポイントのチェーンの先頭で複数の追加と除去を同時に行えるようにするために、この種のカウンターを使用しています。

リストは、ダブルワードのヘッダーと、エレメントのチェーンから成っています。ヘッダーの最初のワードには、リスト内の最初のエレメントを指すポインターが含まれています。ヘッダーの 2 番目のワードには、リストに対して行われた追加の回数を示す 32 ビット・カウンターが含まれています。各エレメントには、リスト内の次のエレメントを指すポインターが含まれています。0 の値はリストの終わりを示します。

次の表は、フリー・プール・リストがどのように操作されるかを示しています。

機能	アクション	
	ヘッダー = 0、 カウンタ	ヘッダー = A、 カウンタ
ADD TO LIST (入ってくるエレメントは記憶位置 A にある)	ヘッダーの最初のワードを記憶位置 A に格納する。アドレス A をヘッダーの最初のワードに格納する。ヘッダーの 2 番目のワードから 1 を引く。	
DELETE FROM LIST。	リストは空である。	ヘッダーの最初のワードを、記憶位置 A の内容の値に設定する。エレメント A を使用する。

下のルーチンは、コードを使用可能にして上記の表に記述されているフリー・プール・リストの操作を実行できるようにするためのものです。

ADD TO FREE LIST ルーチン:

初期条件は以下のとおりです。

GR2 には、追加するエレメントのアドレスが入っている。

GR4 には、ヘッダーのアドレスが入っている。

```

ADDQ  LM  0,1,0(4) GR0,GR1 = ヘッダーの内容
TRYAGN ST  0,0(2)   新しいエレメントがリスト
                               の先頭を指す。
                               カウンタを GR3 に移す。
                               BCTR  3,1   カウンタを減らす。
                               CDS   3,0   ヘッダーを更新する。
                               BC    0,2,0(4)
                               BC    7,TRYAGN
    
```

DELETE FROM FREE LIST ルーチン: 初期条件は以下のとおりです。

GR4 には、ヘッダーのアドレスが入っている。

DELETQ	LM	2,3,0(4)	GR2,GR3 = ヘッダーの内容 リストは空か。 そうであれば、ヘルプ を求める。 そうでなければ、GR0 = 最初のエレメントから のポインター。 カウントを GR1 に移 す。 ヘッダーを更新する。
TRYAGN	LTR	2,2	
	BC	8,EMPTY	
	L	0,0(2)	
	LR	1,3	
	CDS	2,0,0(4)	
	BC	7,TRYAGN	
USE		[任意の命令]	除去されたエレメン トのアドレスは GR2 に入っている。

記憶域オペランドの一貫性に関する規則がないとすれば、記憶位置 ADDQ および DELETQ にある LM (LOAD MULTIPLE) 命令は CDS (COMPARE DOUBLE AND SWAP) 命令でなければならないという点に注意してください。この規則では、LOAD MULTIPLE 命令は、ダブルワード境界に位置合わせされた 8 バイト・オペランドを取り出す必要があります。これは、別の CPU がその取り出し中のダブルワードを変更する場合に、少なくともダブルワードの一貫性を維持するような操作に従っていれば、そのダブルワードの完全に新しい値かまたは完全に古い値が取得され、両者が混在する値が取得されることのないようにするためです。(5-86ページの『記憶域オペランドの一貫性』を参照してください。)

PERFORM LOCKED OPERATION (PLO)

PERFORM LOCKED OPERATION 命令は、マルチプログラミング環境またはマルチプロセッシング環境において、2 つ以上の連続していない記憶位置 (ワードまたはダブルワード) に対して、比較、ロード、比較およびスワップ、および格納操作を行うために使用できます。これらの操作は、複数の命令の実行にわたってではなく PERFORM LOCKED OPERATION 命令の実行中のみ保持されるロックの制御下で、操作のアトミック・セットとして実行されます。ロック競合は CPU により解決され、非常に短時間なので、使用すべきロックを他の CPU で実行されているプログラムが保持している場合に対処するためのメソッドを、プログラムに組み込んでおく必要はありません。また、PERFORM LOCKED OPERATION は、割り込みが起こる前に、操作を完了しロックを解放するので、プログラムがロックを保持している間に割り込まれる可能性も、考慮に入れる必要はありません。

PERFORM LOCKED OPERATION は、複数のオペランドのインターロックされた更新を同時に行うものと考えられます。しかし、この命令は実際にインタ

ーロックされた更新を行うものではなく、PERFORM LOCKED OPERATION および条件付きスワップ命令 (CS および CDS) の両方を使用して、結果を事前に予測できる形で逐次再使用可能リソースを更新することはできません。

以下の例は、PERFORM LOCKED OPERATION を使用して、キューの始めにエレメントを追加する方法を示しています。

キューには、次に示す変数が関連付けられているものとします。それは、キューが変更されるたびに増加するシーケンス番号 S、キューの最初のエレメントのアドレス H (head の頭文字)、そして、キュー内のエレメント数のカウントを示す C です。個々のキュー・エレメントには、キュー内の次のエレメントのアドレスを示す変数 F (forward の頭文字) が含まれているとします。新しいエレメント N をキューの先頭にエンキューするには、N の中で F を H に設定し、以下に示す操作のアトミック・セットを実行します。

$$\begin{array}{l} S+1 \rightarrow S \\ A(N) \rightarrow H \\ C+1 \rightarrow C \end{array}$$

ここで、A(N) は N のアドレスです。

N は、以下の手順でエンキューすることができます。

1. S、H、および C の一貫性のある値を取得します。つまり、S を取得し、S の値と一貫性のある H と C を取得します。
2. H を N.F に格納します。
3. PLO.csdst (比較およびスワップと 2 回の格納を行う PERFORM LOCKED OPERATION) で、スワップ変数として S、そして格納変数として H と C を使用して、S に 1 を加算し、H を A(N) に設定し、C に 1 を加算します。ただし、S がまだステップ 1 で取得された値のままであるとして、すでに S が変更されている場合は、ステップ 1 に戻ります。

他の CPU で実行されている PERFORM LOCKED OPERATION 命令が、S の更新は完了しているが、まだ H または C は更新していないということが考えられるので、単に 3 つの LOAD 命令を使用しただけでは、必ずしも、S、H、および C の一貫性のある値が取得されるとは限りません。この場合、3 つの LOAD 命令では、S については新しい値が取得されますが、H または C については古い値が取得されます。ただし、以下に述べるように、3 つの LOAD 命令を使用することもできます。

ロックの保持中に S が取得された場合、つまり **PERFORM LOCKED OPERATION** を使用して S が取得された場合、以後は、**LOAD** 命令を使用して H および C を取得することができます。他の CPU は S を変更せずに H または C を変更することはできないからです。

PLO.csdst で使用されるパラメーター・リストは、以下のとおりです。ただし、アクセス・レジスター・モードを使用していないものとしてします。

0	
8	
48	
56	A(N)
64	
72	A(H)
80	
88	C+1
96	
104	A(C)

プログラムは以下のとおりです。

LA	RT,H	PL 内のアドレスを初期設定する (T = temp)
ST	RT,PL+76	Op4 アドレス (H のアドレス)
LA	RT,C	
ST	RT,PL+108	Op6 アドレス (C のアドレス)
LA	RN,N	N のアドレス
ST	RN,PL+60	PL 内の op3 を初期設定する (N のアドレス)
LA	R1,S	PLT アドレス = S のアドレス

SR	RS,RS	ダミーの S。おそらく CC1 が 設定される。
SR	R0,R0	機能コード 0 (比較および ロード)
PLO	RS,S,RS,S	ロックの保持中に S を取得する。

LOOP	LA R0,16	機能コード 16 (csdst)
	L RT,H	一貫性のある H
	ST RT,OFSTF(,RN)	OFSTF = N 中の F の オフセット
	L RT,C	一貫性のある C
	LA RT,1(,RT)	C+1
	ST RT,PL+92	PL 内の op5 を初期設定する (C+1)
	LA RSP,1(,RS)	RS/RSP = 偶数/奇数ペア RSP 内の S+1
	PLO RS,S,0,PL	
	BNZ LOOP	S が変更されている (CC が 0 でない) 場合はブランチ

最初の **PERFORM LOCKED OPERATION** 命令 (PLO.cl) については、以下の点に注意してください。S が 0 でない場合 (これはおそらく真です)、S (第 2 オペ

ランド op2) が、RS (第 1オペランド比較値 op1c) にロードされます。S が 0 である場合は、S (第 4 オペランド op4) が、RS (第 3 オペランド op3) にロードされます。これらのロードは、いずれもロックが保持されているときに起きます。どのロードが行われたかを判別するために、条件コードをテストする必要はありません。

上記のプログラムは単純化された例です。キューに、キュー内の最後のエレメントのアドレスを示す変数 T (tail = テール) が関連付けられていて、キューが現在空である場合は、N をキューに追加するときに T も設定する必要があります。そのためには、比較およびスワップおよび三重格納操作を使用する別のプログラムが必要になります。

第 2 オペランドが常にシーケンス番号 S である **PERFORM LOCKED OPERATION** 命令を使用して、キューへの追加、キューからの削除、またはキューの再配置を行う場合、**PERFORM LOCKED OPERATION** の定義により、第 2 オペランドは常に最後に格納されることが指定されているため、上記のプログラムの中の最初の **PERFORM LOCKED OPERATION** 命令は、**LOAD** 命令で置き換えることができます。その場合、破線で囲まれている 3 つの命令は、L RS,S で置き換えられます。

ソート命令

ツリー・フォーマット

COMPARE AND FORM CODEWORD と **UPDATE TREE** の 2 つの命令は、ツリー (特定のフォーマットを持つデータ構造) を参照します。ツリーは、連続番号が付けられた幾つかのノード (常に奇数) から成っています。ノード 1 はツリーのルートです。ルート以外の各ノードは、どれも同じツリー内に 1 つずつ親ノードを持っています。どの親ノードも、子ノードを 2 つずつ持っています。偶数番号のノードはそれぞれの親ノードの左の子であり、奇数番号のノード (ノード 1 を除く) はそれぞれの親ノードの右の子です。ノード番号を 2 で割った商 (剰余は無視) が、親ノード番号です。子を持つノードは内部ノードとも呼ばれ、子を持たないノードは終端ノードとも呼ばれます。A-53ページの図A-5 は、21 個のノードからなるツリーを体系的に表したもので、矢印は各親ノードとそれぞれの子ノードを結び付けています。

ツリーは、幾つかのソートされたレコード・シーケンスをマージして、単一のレコード・シーケンスにするために使用されます。マージ・プロセスの各ステップにおい

て、マージ済みシーケンスの初期部分と、個々のソート済みシーケンスのまだマージされていない部分があります。各ステップでは、ソート済みシーケンスのまだマージされていないすべての部分から、最下位のレコード (昇順でソートする場合に最小のキーを持つレコード) が選択され、それがマージ済みシーケンスに追加されます。ツリー内の各終端ノードは、それぞれソート済みシーケンスの 1 つを表します。ツリー内の内部ノードの数は、ソート済みシーケンスの数から 1 を引いた値です。各内部ノードには、概念上、1 つを除くすべてのソート済みシーケンスから 1 つずつレコードが含まれます。これらのレコードは、1 つを除くすべてのソート済みシーケンス内のまだマージ済みシーケンスに追加されていないレコードのうちの、最下位のレコードです。そのほかに、残っている 1 つのソート済みシーケンス内の最小番号のレコードがあります。この追加のレコードは、次にマージ済みシーケンスに追加するレコードを選択するために、ツリーのノードと比較され交換されます。この処理は、残りの 1 つのソート済みシーケンスを表す終端ノードの親から始められ、そのノードからツリーのルートへのパスに沿って続けられます。選択されたレコードは、ツリーのルートから抜け出します。

ツリーは、各ノードが、最下位のレコードを見つけるための「消去法トーナメント」における個々の比較操作を表しているものと考えれば、よく分かります。トーナメントが完了した後は、各ノードについて、そのノードが表す比較において大きい方のキーを持っていた「敗者」レコードが関連付けられています。各ノードの敗者レコードのほかに、比較の結果大きいと見なされなかったためにどのノードにも関連付けられていないレコード (「勝者」) が 1 つずつあります。その次のステップでは、勝者レコードが発生した同じソート済みシーケンスから新しいレコードが取り出され、そのレコードを前の勝者の位置に置いて再度トーナメントが行われます。ツリー内のすべてのノードが表すすべての比較を行う必要はないものと考えられます。そのほとんどは、前の勝者と置き換わる新しいレコードの影響を受けません。実際には、前の勝者レコードが参加したノード比較を再実行するだけで十分です。個々の新しいレコードは、ツリー内において、そのレコードを含むソート済みシーケンスを表す終端ノードに挿入されます。ツリーを使用するには、プログラミングにより、個々の勝者レコードが発生した終端ノードを記憶しておくためのメソッドを用意することが必要です。UPDATE TREE 命令を使用することにより、終端ノードに新しいレコードが挿入され、そして、新しい勝者レコードが汎用レジスター内に残されるようにツリーが更新されます。

マージ・ロジックの多くは、レコードの実際のキーを比較する代わりに、「コードワード」(実際のキーを参照するのではなくレコード・キーを表す) を使用して実行

することができます。ツリー内のノードにおけるコードワードの値は、単にレコードのキーだけでなく、そのノードでの最後の比較における勝者レコードのキーによっても変化します。コードワードは次の 2 つの部分から成っています。

1. ビット 16-31 には、このレコード・キーとノードの勝者レコードのレコード・キーとの間で異なっている最初のハーフワードの 1 の補数が含まれています。
2. ビット 0-15 は、このレコードのキーの中での、ビット位置 16-31 にあるハーフワード値 (補数) の直後のハーフワードのバイト・オフセットを示します。

最後の勝者レコードのパス内でレコードを比較したときに、新しいレコードも、最後の勝者との比較結果のコードワードで表されている場合は、更新パス内のすべてのコードワードが同じ勝者に関連しています。このようなコードワードを比較した場合、大きいコードワードは小さいキーを表します (逆も言えます)。したがって、コードワードが等しくないときは、大きいコードワード (実際のキーは小さいことを表す) を持つノード・エントリーは、ツリー上で上に移動させる必要があります。

コードワードの値がタイ (等位) である場合は、実際のキーを参照する必要があります。そのためには、**COMPARE AND FORM CODEWORD** 命令を使用します。この命令は、あいまいさを解決して、大きいキーのレコード (敗者) の新しいコードワードを計算します。

ツリーの各ノードの 8 バイトは、(1) このレコードと比較して低位と見なされた最後のレコードとの関係で計算されたこのレコードのコードワード、および、(2) このレコードを見つけるために使用できるパラメーター (例えば、直接アドレスまたは間接アドレス) から成っています。

UPDATE TREE 命令は、等価のコードワードが検出された後でツリー更新を停止するように定義されています。また、タイ・ブレイク命令 **COMPARE AND FORM CODEWORD** の使用後に、**UPDATE TREE** により、前に等位のコードワードが検出された位置からツリー更新を再開することができます。

COMPARE AND FORM CODEWORD は、降順にマージを行うためにも使用できます。その場合は、ノードのコードワードのビット 16-31 には、そのノードの勝者レコードのレコード・キーが異なっている最初のハーフワードの真の値が含まれています。**COMPARE AND FORM CODEWORD** の降順オプションを使用する場合

は、2つのコードワードのうち大きい方が、大きいキーを表します。

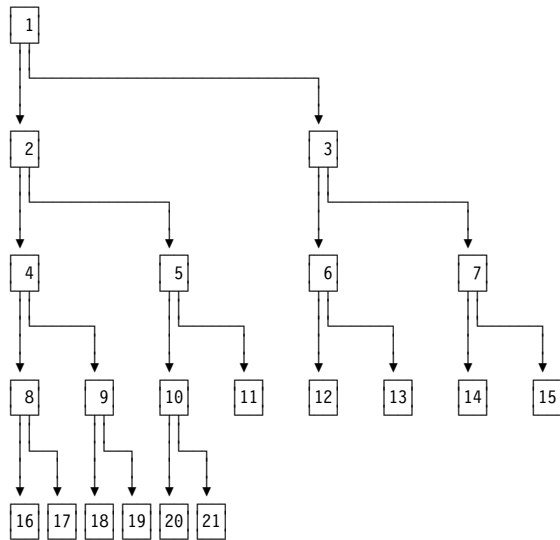


図 A-5. 21 個のノードを持つマージ制御ツリーの概略図

ソート命令の使用例

ソート・プログラムの中で、マージ操作に UPDATE TREE および COMPARE AND FORM CODEWORD 命令を使用する方法の例を示します。5 ウェイ・マージには、4 個の内部ノードと 5 個の終端ノード位置を持つツリー・データ構造が必要です。この後で示すツリーの概略図には、4 個の内部ノード (ダミー・ノードは含めない) と、マージする 5 個の入力シーケンスがあり、それらのシーケンスは各終端ノードに 1 つずつ入れられています。この図では、入力シーケンス内の各レコードはそれぞれのアドレスで示されています。実際のレコード内容は、A-57 ページの図 A-7 に示してあります。各レコードには、以下のフィールドから成る 16 バイトが含まれています。

バイト・オフセット

(16 進数)

フィールド

バイト・オフセット (16 進数)	フィールド
0-5	6 バイトのレコード・キー。
6-7	この入力シーケンスの次のレコードの入力シーケンスを指定するハーフワードのノード指標。
8-B	同じ入力シーケンス内の次のレコードのアドレス。
C-F	このチェーニング・フィールドは、最初は 0 です。マージが完了すると、マージ・シーケンス内の次のレコードのアドレスがこのフィールドに入ります。

マージ・プロセスでは、それぞれソート済みの順序になっている 5 個の入力シーケンスから、単一のソート済みシーケンスが形成されます。このプロセスは、次の 3 つのステップに分かれています。

1. プライミング (準備) ステップでは、5 個の入力シーケンスのそれぞれから最初のレコードを取り出して、ツリー・データ構造に入れます。ツリーに取り込む各レコードについて、最初に、可能と認められる最下位のキー値 (すべて 0) との関係に基づいて、それぞれのコードワード値が計算されます。このコードワードと、実際のレコードのアドレスを含む第 2 のワードが、該当ノードに入力できるダブルワードのノード値を形成します。プライミングが終わると、ノード値 (5 個の入力シーケンスからそれぞれ 1 つずつ) が、ツリーに取り込まれています。その結果、4 個の内部ノードのそれぞれにノード値が 1 つずつ含まれ、1 つの勝者レコードがツリーのルートから抜け出しています。
2. 勝者レコードがそれぞれツリーから抜け出した後で、メイン・マージ・プロセスが反復して行われます。反復のたびに、1 つの新しいレコードのノード値がツリーに取り込まれ、新しい勝者レコードのノード値が作成されます。ツリーと勝者を合わせたものには、どの時点においても、マージしている各入力シーケンスからそれぞれ 1 つずつのノード値が含まれていなければなりません。したがって、各反復でツリーに取り込まれる新しい値は、その前の反復における勝者ノード値が発生したものと同一入力シーケンスからの値でなければなりません。
3. 1 つの入力シーケンスの最後のレコードのノード値が勝者として抜け出した後は、次の反復の時点では、その入力シーケンスにはツリーに取り込む後続レコードは残っていません。したがって、この状況が生じるたびに、マージの回数を 1 つずつ減らす必要

があります。このランナウト・プロセスは、4 ウェイ、3 ウェイ、2 ウェイ、および 1 ウェイのマーゼのそれぞれについて、1 回以上の反復の結果発生します。例では、1 つのシーケンスからの次の入力レコードが先行するレコードより低位であることが判明した時点 (シーケンス・ブレイク) から、ランナウトが始まります。

以下、プライミング・プロセスについて説明し、プライミング完了後のツリーの状態を示します。その後で、**UPDATE TREE** および **COMPARE AND FORM CODEWORD** を使用してメイン・マーゼを行うための短いプログラムについて説明します。さらに、メイン・マーゼを 16 回反復した場合のツリーおよび特定の汎用レジスタの状況を示す簡略トレースについて説明します。この例では、ランナウト・プロセスについては説明しません。

プライミングは、各入力シーケンスの最初のレコードのノード値を作成することから始まります。ノード値の第 1 ワードは、すべての桁が 2 進数の 0 であるレコード・キーを基準として **COMPARE AND FORM CODEWORD** を実行した結果として形成されるコードワードです。ノード値の第 2 ワードは、そのノード値が表すレコードのアドレスです。各入力シーケンスの最初のレコードのノード値は、以下のとおりです。

シーケンス指標	ノード値
28	0006 FFFC 0000 1030
30	0006 FFFB 0000 1040
38	0006 FFFA 0000 1050
40	0004 FFFE 0000 1080
48	0006 FFF0 0000 1060

この例のツリー・データ構造では、基底アドレスは `X'1000'` で、汎用レジスタ 4 に入っているものとします (意図されている **UPDATE TREE** の使用に合わせるため)。同様に、内部ノード指標値および入力シーケンス指標値は、常に汎用レジスタ 5 から使用されるものとします。

この例にはツリー・プライミング・プログラムは含まれていませんが、このプログラムを作成するには、以下のように **UPDATE TREE** 命令を使用します。最初に、ツリーの各内部ノードのコードワード位置は、すべて 1 (`X'FFFF FFFF'`) に初期設定されます。この手法により、ツリーはダミーの低位レコードで満たされます。次に、表の中の各レコードについて、(1) シーケンス指標が汎用レジスタ 5 にロードされ、(2) ノード値が汎用レジスタ 0 および 1 にロードされ、(3) **UPDATE TREE** が実行されます。このプライミング・プロセスが完了すると、この例のツリー・ノードの内容は、A-59ページの図A-9 の行 0 に示すようになっていきます。汎用

レジスタの内容は、A-58ページの図A-8 の最初の行に示すようになっていきます。

メイン・マーゼのためのプログラムを示す図は、3 グループの列に分かれており、各グループには、絶対プログラム、汎用レジスタ・トレース、およびシンボリック・プログラムが含まれています。プログラムの最初の部分は、記号位置 L1 から L2 にわたっています。この部分は、新しいレコードをツリーに取り込み、**UPDATE TREE** 命令を実行します。**UPDATE TREE** でタイのコードワードが検出されなかった場合は、**UPDATE TREE** の後の **BRANCH ON CONDITION** 命令が L1 にループバックして、次のレコードをツリーに取り込みます。この **BRANCH ON CONDITION** 命令は、**UPDATE TREE** が、メソッド 1 (条件コード 1 の設定) またはメソッド 2 (条件コード 3 の設定) に従って操作を行うときに使用するのに適しています。(上記の文は 370-XA に適用されます。ESA/370 および ESA/390 では、**UPDATE TREE** はメソッド 2 のみに従って操作を行います。ただし、これは、**UPDATE TREE** が条件コード 1 を設定できないことを意味するわけではありません。メソッド 2 では、条件コード 3 を設定するような条件が存在するかどうかテストされますが、メソッド 1 ではこのテストは行われません。)

UPDATE TREE でタイのコードワードが検出されると、**UPDATE TREE** 命令は完了し、それに続く **BRANCH ON CONDITION** 命令はブランチを行わず、プログラムの第 2 部分 (タイのコードワードを持つエントリーを操作する) に制御が移ります。この部分では、**UPDATE TREE** の L2 にブランチし、そこでツリー更新が再開されます。タイのコードワードは、ツリー内のどのレベルでも (あるいはすべてのレベルで) 検出される可能性があるため、取り込まれる各レコードについて最大 3 回まで、プログラムのタイ・コードワード部分に入る可能性があります。

メイン・マーゼの第 1 部分に関する汎用レジスタ・トレースは、最初の反復で各命令が実行された後の、最初の 7 つの汎用レジスタの内容を示します。マーゼ・チェーン・フィールド (1140 の位置にある) は、レコード間のマーゼ・チェーン・アドレス・チェーンのアンカーの役割を果たすという点に注意してください。ある種のレジスタについては、このトレースで示されるのは低位の半分のみで、高位の半分は常に 0 になります。

A-59ページの図A-9 は、16 個のレコードのメイン・マーゼ全体に対する簡略トレースを示しています。この図には、ツリーに取り込まれる各レコードについて、それぞれ 1 つ以上 (常に奇数) の行があり、最終的に勝者が GR0 および GR1 に入れられることになるツリー更新を

示しています。各レコードについての最初の行は、最初または唯一の UPDATE TREE の実行前の、GR5、GR2、および GR3 の値を示します。偶数番号の行には、ツリー・ノードの UPDATE TREE による記憶域更新が示されています (左から右へ読むとスワッピングの順序をたどることができます)。例えば、行 10 およびそれに対応する UPDATE TREE について言えば、GR5 に 28 が含まれているので、最初に調べられる記憶域ノードは 1010 です (概略図を参照)。GR0 中のコードワードは 0004 FFFE (GR2 と同じ) であり、これは 1010 にあるワード (0006 FFF0) より小さいので、1010 にあるダブルワードは、GR0 および GR1 中のダブルワードと交換されます。UPDATE TREE の同じ実行の中で、1008 において第 2 の比較が行われた結果、再び別のレジスター/記憶域ダブルワード・スワップが行われ、UPDATE TREE の完了時には、勝者 (レコード 1040) が GR1 に残されています (A-59ページの図A-9の右端の列を参照)。

コードワードの比較の結果が、タイまたはスワップとならない場合 (つまり記憶域コードワード値の方が小さい場合) は、トレースの中で、その記憶域エントリーについてはアスタリスクが示されます。

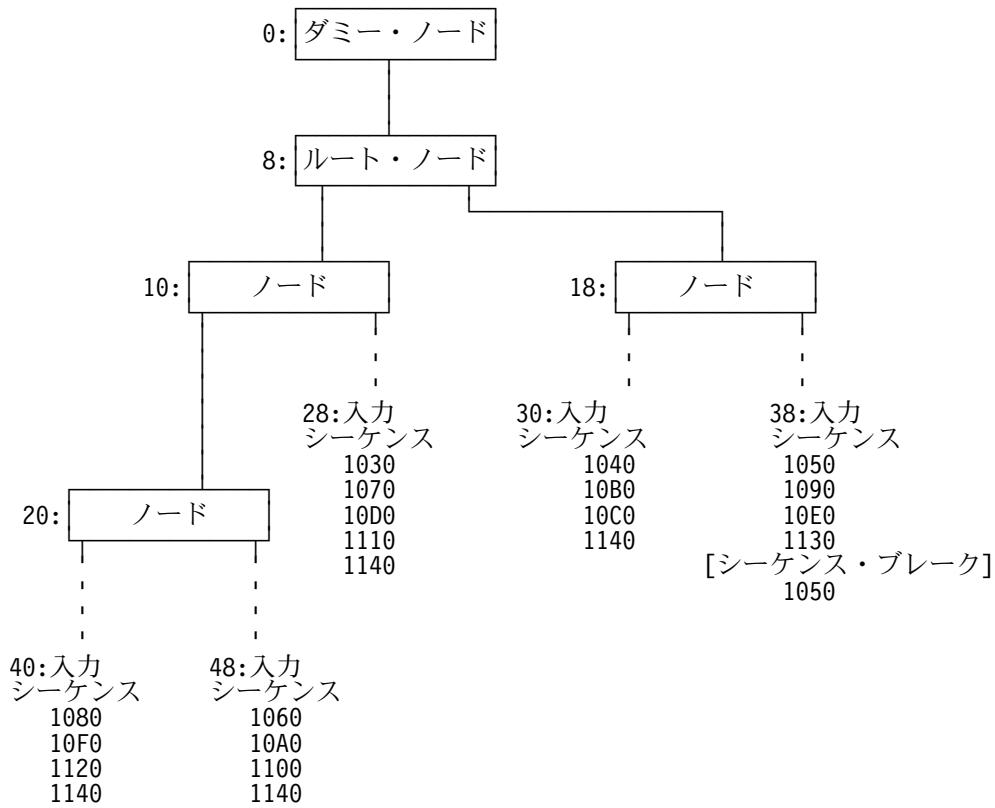
等しいコードワードが見つかると、UPDATE TREE の実行は完了します。このような場合は、後続の行に、タイ・コードワードの結果が示されます。このルーチンは、常に新しいコードワードを格納し、さらに、L2 にブランチして再度 UPDATE TREE を実行する前に、新

しいレコード・アドレスを格納することもあります。この行の中の「loses」または「wins」という表記は、それぞれ、ノードが勝者か敗者かを示します。

A-58ページの図A-8 のタイ・ブレイク・トレース部分は、第 3 のレコード (つまり、UPDATE TREE がタイ・コードワードを検出した最初のレコード) の取り扱いを示しています。これは、A-59ページの図A-9 の行 31 に対応しています。

以下に、確認の目的で、この例を使用するために必要なステップを要約して示します。

1. 記憶域を以下のように初期設定します。
 - a. 1008 - 102F は、A-59ページの図A-9 の行 0 から
 - b. 1030 - 114F は、A-57ページの図A-7 から
 - c. 1150 - 1189 は、A-58ページの図A-8 から
2. 図A-8 の最初の行に従って GR を初期設定し、図A-8 に従って最初のレコードをトレースします。
3. 各 UPTまたは BC 15,L2 の完了までトレースします (図A-9 の各行について 1 回ずつ)。図A-9 の行 31 のタイ・コードワード部分に関する GR の詳細トレースは、図A-8 の下部に示されています。
4. 103C から始まり 114C で終わるチェーン内のアドレスが、図A-7 の右側の列に示されているものと同じであることを確認します。



注: 各ノードおよび入力シーケンスは、16進数のノード指標で表した番号で示されています。各入力シーケンスは、レコード・アドレスのリスト (同じく 16進数) の形で示されています。

図 A-6. 実行するマージの例の概略図

位置	以下の 16 進バイト・ オフセットにある レコード・キー					後続レコード						マージ・ チェーン・ アドレス														
						指標		位置																		
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F										
1030	0	0	0	0	0	0	0	0	0	0	0	1	0	7	0	0	0	0	0	0	0	1	0	4	0	
1040	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1060	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1080	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1090	0	0	0	0	0	F	F	F	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10A0	0	0	0	0	F	F	F	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10B0	0	0	0	0	F	F	F	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10C0	0	0	0	0	F	F	F	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10D0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10E0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10F0	0	0	8	0	0	0	0	0	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1100	0	0	8	0	0	0	0	0	2	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1110	0	0	8	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1120	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1130	F	F	F	F	F	F	F	F	F	F	F	E	0	0	3	8	0	0	0	0	0	0	0	0	0	0
1140	F	F	F	F	F	F	F	F	F	F	F	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図 A-7. マージするレコードの内容

絶対		汎用レジスタ・トレース						シンボリック・プログラム		
位置	命令	GR0	GR1	GR2	GR3	GR4	GR5	GR6	位置	命令
		0006FFFC	1030		0000	1000	0000	1140		X'1000'4,を使用
1150	5010600C	↓	↓		↓		↓	↓	L1	ST 1,12(,6) マージ・チェーン・アドレスを格納する。
1154	48501006	↓	↓		↓		0028	↓	LH 5,6(,1) 勝者の入力シーケンスのノード指標をロードする。	
1158	58301008	↓	↓		1070		↓	↓	L 3,8(,1) 後続レコード・アドレスをロードする。	
115C	1861	↓	↓		↓		↓	1030	LR 6,1 次のマージ・チェーン格納のために古い勝者アドレスを保管する。	
1153	1B22	↓	↓	00000000	↓		↓	↓	SR 2,2 GR2を初期オフセットとして0にする。	
1160	B21A0004	↓	↓	0004FFFE	↓		↓	↓	CFC 4 最後の勝者に基づいてレコードのコードワードを計算する。	
1164	4720418A	↓	↓	↓	↓		↓	↓	BC 2,L3 CC=2(シーケンス・ブレーク)で終了する	
1168	1813	↓	1070	↓	↓		↓	↓	LR 1,3	
116A	1802	0004FFFE	↓	↓	↓		↓	↓	LR 0,2	
116C	0102	0006FFFB	1040	↓	↓		0000	↓	L2 UPT 新しいレコード・エントリーをGR 0-1に移す。	
116E	47504150	↓	↓	↓	↓		↓	↓	BC 5,L1 ツリー・データ構造を更新する。コードワード・タイが見つからない場合は、次の反復へジャンプする。	
		00040000	1090	00040000	10B0		0018	1050	*	タイ・コードワードでフォールスルーする。
		↓	↓	↓	↓		↓	↓	*	← タイ・ブレーク・トレースのGR値
1172	88200010	↓	↓	00000004	↓		↓	↓	SRL 2,16	コードワード・オフセットをCFCの初期オフセット位置にシフトする。
1176	B21A0004	↓	↓	0006FFFD	↓		↓	↓	CFC 4	敗者のコードワードを計算する。
117A	50254000	↓	↓	[CC=1]	↓		↓	↓	ST 2,0(5,4)	敗者のコードワードを現記憶域ノードに格納する。
117E	47C0416C	↓	↓	ブランチする	↓		↓	↓	BC 12,L2	古い記憶域ノード・エントリーが敗者である場合はツリー更新を再開する。
1182	50354004	↓	↓	↓	↓		↓	↓	ST 3,4(5,4)	敗者のレコード・アドレスを格納する。
1186	47F0416C	↓	↓	↓	↓		↓	↓	BC 15,L2	ツリー更新を再開する。
118A	...	↓	↓	↓	↓		↓	↓	L3 ...	終了時には制御はここに到達する。

図 A-8. メイン・マージ用のプログラム

行#	記憶位置 1160 での CFC 後の 汎用レジスタ				ノード・エントリーの記憶域トレース								UPT または BC15、L2 の 後の汎用レジ スタ	
	GR 5	GR2	GR3	注釈	1020		1018		1010		1008		GR0	GR1
0 ¹					0004FFFE	1080	0006FFFA	1050	0006FFF0	1060	0006FFFB	1040	0006FFFC	1030
10	28	0004FFFE	1070	No tie					0004FFFE	1070	0006FFF0	1060	0006FFFB	1040
20	30	00040000	10B0	No tie			00040000	10B0			*		0006FFFA	1050
30 31 32	38	00040000	1090	CC = 0 Loses No tie			Tie 0006FFFD				00040000	1090	00040000 00040000 0006FFF0	1090 1090 1060
40 41 42	48	00040000	10A0	CC = 0 Equal No tie	00040000	10A0			Tie 80001070		*		0004FFFE 0004FFFE 0004FFFE	1080 1080 1080
50	40	0002FF7F	10F0	No tie	0002FF7F	10F0			00040000	10A0	**		80001070	1070
60 61 62	28	0002FFFE	10D0	CC = 0 Wins No comp					0002FFFE	10D0	Tie 0006FFFE	10A0	00040000 00040000 00040000	10A0 1090 1090
70	38	0002FF7F	10E0	No tie			0002FF7F	10E0			0006FFFD	10B0	0006FFFE	10A0
80 81 82	48	0002FF7F	1100	CC = 0 Wins No tie	Tie 0006FFAF	1100			0002FF7F	10F0	0002FFFE	10D0	0002FF7F 0002FF7F 0006FFFD	1100 10F0 10B0
90	30	800010C0	10C0	No tie			*				*		800010C0	10C0
100	30	00020000	1140	No tie			00020000	1140			0002FF7F	10E0	0002FFFE	10D0
110 111 112 113 114	28	0002FF7F	1110	CC = 0 Wins CC = 0 Wins No comp					Tie 0004FFFC	1110	Tie 0004FFFD	10F0	0002FF7F 0002FF7F 0002FF7F 0002FF7F 0002FF7F	1110 10F0 10F0 10E0 10E0
120 121 122	38	00020000	1130	CC = 0 Loses No tie			Tie 00060000				00020000	1130	00020000 00020000 0004FFFD	1130 1130 10F0
130	40	0002FF6F	1120	No tie	0002FF6F	1120			*		*		0006FFAF	1100
140	48	00020000	1140	No tie	00020000	1140			0002FF6F	1120	*		0004FFFC	1110
150	28	00020000	1140	No tie					00020000	1140	*		0002FF6F	1120
160 161 162 163 164 165 166	40	00020000	1140	CC = 0 Equal CC = 0 Equal CC = 0 Wins No comp	Tie 80001140				Tie 80001140		Tie 00060000	1140	00020000 00020000 00020000 00020000 00020000 00020000 00020000	1140 1140 1140 1140 1140 1130 1130
170	38	00020000	1050	Branch										

説明:

1 行 0 は、プライミング後のツリー内の値を示しています。
* スワップなしを意味します。
** UPDATE TREE メソッド 1 が使用される場合はスワップなしを意味し、UPDATE TREE メソッド 2 が使用される場合は検査が行われないことを意味します。ESA/370 および ESA/390 にはメソッド 2 のみが組み込まれています。
CC = 0 UPDATE TREE は、タイを検出し、条件コード 0 を設定します。
Loses タイ・コードワード・ルーチンは、このノードが敗者であると判断します。
Wins タイ・コードワード・ルーチンは、このノードが勝者であると判断します。
Equal タイ・コードワード・ルーチンは、キーが等しいと判断します。
Branch シーケンス・ブレイクでブランチし 118A で終了します。
No comp 比較なし。

図 A-9. メイン・マージ処理の簡略トレース

付録B. 命令一覧表

以下の表には、命令の名称、ニーモニック、および命令コードをリストしてあります。一部のモデルでは、例えば特殊機構や特注機構の補助のためまたはそれらの一部として提供されている命令などのように、表には含まれていない命令が用意されている場合があります。

解釈実行機能のための命令コードは、この付録には収めてありません。この機能に関連する命令コードについては、「*IBM System/370 Extended Architecture Interpretive Execution*」(SA22-7095)を参照してください。

2 バイト命令フォーマットを持つ 16 進数 00 の命令コードは、無効な命令の標識が必要となるときにプログラムで使用できるようにするために割り振られています。この命令コードが、CPU に組み込まれる命令に割り当てられる可能性はないものと考えられます。

「特性」および「ページ」の欄の記号の説明:

¢	逐次化とチェックポイント同期化を起こす。
¢ ¹	M ₁ フィールドがすべて 1 で R ₂ フィールドがすべて 0 のときに、逐次化とチェックポイント同期化を起こす。
¢ ²	アンスタックすべき状態エントリーがプログラム呼び出し状態エントリーであるときに、逐次化とチェックポイント同期化を起こす。
\$	逐次化を起こす。
A	論理アドレスについてのアクセス例外。
A ¹	アクセス例外。すべてのアクセス例外が起きるとは限らない。詳細については、命令の説明を参照。
AI	命令アドレスについてのアクセス例外。
AS	ASN 変換指定例外および特殊操作例外。
AT	ASN 変換指定例外。
B	PER プランチ・イベント。
B ₁	B ₁ フィールドは、アクセス・レジスター・モードではアクセス・レジスターを指定する。
B ₂	B ₂ フィールドは、アクセス・レジスター・モードではアクセス・レジスターを指定する。
BP	PSW ビット 16 および 17 の値が 01 のときは、B ₂ フィールドはアクセス・レジスターを指定する。
C	条件コードが設定される。
Da	AFP レジスター・データ例外。
Db	BFP 命令データ例外。
Dd	10 進オペランド・データ例外。
DF	10 進オーバーフロー例外。
DK	10 進除算例外。

DM	モデルによっては、DIAGNOSE は各種のプログラム例外を生成し、条件コードを変更することがある。
E	E 命令フォーマット。
E2	拡張変換機能 2。
EO	HFP 指数オーバーフロー例外。
ES	拡張記憶機構。
EU	HFP 指数アンダーフロー例外。
EX	実行例外。
FC	アクセス・レジスターの指定は、命令の機能コードに応じて決まる。
FK	HFP 浮動小数点除算例外。
G0	命令の実行時に、汎用レジスター 0 が暗黙的に使用される。
G1	命令の実行時に、汎用レジスター 1 が暗黙的に使用される。
G2	命令の実行時に、汎用レジスター 2 が暗黙的に使用される。
G4	命令の実行時に、汎用レジスター 4 が暗黙的に使用される。
GM	命令の実行時に、複数の汎用レジスターが暗黙的に使用される。
GS	命令の実行時に、汎用レジスター 1 が暗黙的にサブシステム識別ワードとして使用される。
IF	固定小数点オーバーフロー例外。
II	割り込み可能命令。
IK	固定小数点除算例外。
IS	割り込み可能命令および特殊操作例外。
I1	アクセス・レジスター・モードでは、アクセス・レジスター 1 が暗黙的に指定される。
I4	アクセス・レジスター・モードでは、アクセス・レジスター 4 が暗黙的に指定される。
L	新しい条件コードがロードされる。
LS	HFP 有効数字例外。
MD	アクセス・レジスター・モードのときにアクセス・レジスターを指すかどうかは、モデルによって決まる。
MO	モニター・イベント。
N	z/Architecture の新規の命令で、ESA/390 にはない。
N3	z/Architecture の新規の命令で、ESA/390 にも追加されている。
OP	オペランド例外。
P	特権命令例外。
Q	準特権命令に関する特権命令例外。

R1	R1 フィールドは、アクセス・レジスター・モードではアクセス・レジスターを指定する。	U2	R2 フィールドは、無条件にアクセス・レジスターを指定する。
R2	R2 フィールドは、アクセス・レジスター・モードではアクセス・レジスターを指定する。	UB	R1 および R3 フィールドは無条件にアクセス・レジスターを指定し、B2 フィールドはアクセス・レジスター・モードの場合にアクセス・レジスターを指定する。
RI	RI 命令フォーマット。	WE	スペース切り替えイベント。
RIE	RIE 命令フォーマット。	Xi	IEEE 無効操作条件。
RIL	RIL 命令フォーマット。	Xo	IEEE オーバーフロー条件。
RR	RR 命令フォーマット。	Xu	IEEE アンダーフロー条件。
RRE	RRE 命令フォーマット。	Xx	IEEE 不正確条件。
RRF	RRF 命令フォーマット。	Xz	IEEE ゼロ除算条件。
RS	RS 命令フォーマット。	Z1	PROGRAM CALL に関するその他の例外およびイベント (これには、ASX 変換例外、EX 変換例外、LX 変換例外、PC 変換指定例外、特殊操作例外、スタック一杯例外、スタック指定例外、およびスペース切り替えイベントが含まれます)。
RSE	RSE 命令フォーマット。	Z2	PROGRAM TRANSFER に関するその他の例外およびイベント (これには、AFX 変換例外、ASX 変換例外、1 次権限例外、特殊操作例外、およびスペース切り替えイベントが含まれます)。
RX	RX 命令フォーマット。	Z3	SET SECONDARY ASN に関するその他の例外 (これには、AFX 変換例外、ASX 変換例外、2 次権限例外、および特殊操作例外が含まれます)。
RXE	RXE 命令フォーマット。	Z4	PROGRAM RETURN に関するその他の例外およびイベント (これには、AFX 変換例外、ASX 変換例外、2 次権限例外、特殊操作例外、スタック空例外、スタック操作例外、スタック指定例外、スタック・タイプ例外、およびスペース切り替えイベントが含まれます)。
RXF	RXF 命令フォーマット。		
S	S 命令フォーマット。		
SE	特殊操作例外、スタック空例外、スタック指定例外、およびスタック・タイプ例外。		
SF	特殊操作例外、スタック一杯例外、およびスタック指定例外。		
SI	SI 命令フォーマット。		
SO	特殊操作例外。		
SP	指定例外。		
SQ	HFP 平方根例外。		
SS	SS 命令フォーマット。		
SSE	SSE 命令フォーマット。		
ST	PER 記憶域変更イベント。		
SU	PER 実アドレスを使用した格納イベント。		
SW	特殊操作例外およびスペース切り替えイベント。		
T	トレース例外 (トレース・テーブル、アドレスリング、および低アドレス保護を含む)。		
U	条件コードは予測不能。		
U1	R1 フィールドは、無条件にアクセス・レジスターを指定する。		

名前	二 モ ニ ック	特性						命 令 コ ド	ペ ー ジ
ADD (拡張精度 BFP)	AXBR	RRE C		SP	Db Xi	Xo Xu Xx		B34A	19-19
ADD (長精度 BFP)	ADBR	RRE C			Db Xi	Xo Xu Xx		B31A	19-19
ADD (長精度 BFP)	ADB	RXE C	A		Db Xi	Xo Xu Xx	B ₂	ED1A	19-19
ADD (短精度 BFP)	AEBR	RRE C			Db Xi	Xo Xu Xx		B30A	19-19
ADD (短精度 BFP)	AEB	RXE C	A		Db Xi	Xo Xu Xx	B ₂	ED0A	19-19
ADD (32)	AR	RR C			IF			1A	7-16
ADD (32)	A	RX C	A		IF		B ₂	5A	7-16
ADD (64<32)	AGFR	RRE C N			IF			B918	7-16
ADD (64<32)	AGF	RXE C N	A		IF		B ₂	E318	7-16
ADD (64)	AGR	RRE C N			IF			B908	7-16
ADD (64)	AG	RXE C N	A		IF		B ₂	E308	7-16
ADD DECIMAL	AP	SS C	A		Dd DF		ST B ₁	FA	8-5
ADD HALFWORD	AH	RX C	A		IF		B ₂	4A	7-16
ADD HALFWORD IMMEDIATE (32)	AHI	RI C			IF			A7A	7-16
ADD HALFWORD IMMEDIATE (64)	AGHI	RI C N			IF			A7B	7-16
ADD LOGICAL (32)	ALR	RR C						1E	7-17
ADD LOGICAL (32)	AL	RX C	A				B ₂	5E	7-17
ADD LOGICAL (64<32)	ALGFR	RRE C N						B91A	7-17
ADD LOGICAL (64<32)	ALGF	RXE C N	A				B ₂	E31A	7-17
ADD LOGICAL (64)	ALGR	RRE C N						B90A	7-17
ADD LOGICAL (64)	ALG	RXE C N	A				B ₂	E30A	7-17
ADD LOGICAL WITH CARRY (32)	ALCR	RRE C N3						B998	7-17
ADD LOGICAL WITH CARRY (32)	ALC	RXE C N3	A				B ₂	E398	7-18
ADD LOGICAL WITH CARRY (64)	ALCGR	RRE C N						B988	7-17
ADD LOGICAL WITH CARRY (64)	ALCG	RXE C N	A				B ₂	E388	7-18
ADD NORMALIZED (拡張精度 HFP)	AXR	RR C		SP	Da EU EO LS			36	18-7
ADD NORMALIZED (長精度 HFP)	ADR	RR C			Da EU EO LS			2A	18-7
ADD NORMALIZED (長精度 HFP)	AD	RX C	A		Da EU EO LS		B ₂	6A	18-7
ADD NORMALIZED (短精度 HFP)	AER	RR C			Da EU EO LS			3A	18-7
ADD NORMALIZED (短精度 HFP)	AE	RX C	A		Da EU EO LS		B ₂	7A	18-7
ADD UNNORMALIZED (長精度 HFP)	AWR	RR C			Da EO LS			2E	18-8
ADD UNNORMALIZED (長精度 HFP)	AW	RX C	A		Da EO LS		B ₂	6E	18-8
ADD UNNORMALIZED (短精度 HFP)	AUR	RR C			Da EO LS			3E	18-8
ADD UNNORMALIZED (短精度 HFP)	AU	RX C	A		Da EO LS		B ₂	7E	18-8
AND (文字)	NC	SS C	A				ST B ₁ B ₂	D4	7-18
AND (即値)	NI	SI C	A				ST B ₁	94	7-18
AND (32)	NR	RR C						14	7-18
AND (32)	N	RX C	A				B ₂	54	7-18
AND (64)	NGR	RRE C N						B980	7-18
AND (64)	NG	RXE C N	A				B ₂	E380	7-18
AND IMMEDIATE (高 高)	NIHH	RI C N						A54	7-19
AND IMMEDIATE (高 低)	NIHL	RI C N						A55	7-19
AND IMMEDIATE (低 高)	NILH	RI C N						A56	7-19
AND IMMEDIATE (低 低)	NILL	RI C N						A57	7-19
BRANCH AND LINK	BALR	RR			T		B	05	7-20

図 B-1 (1/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性					命 令 コ ド	ペ ー ジ
BRANCH AND LINK BRANCH AND SAVE BRANCH AND SAVE BRANCH AND SAVE AND SET MODE BRANCH AND SET AUTHORITY	BAL BASR BAS BASSM BSA	RX RR RX RR RRE			T T SO T	B B B B B	45 0D 4D 0C B25A	7-20 7-20 7-21 7-21 10-5
BRANCH AND SET MODE BRANCH AND STACK BRANCH IN SUBSPACE GROUP BRANCH ON CONDITION BRANCH ON CONDITION	BSM BAKR BSG BCR BC	RR RRE RRE RR RX	A ¹ A ¹	Z ⁵ SO T	T T T φ ¹	B B B B B	0B B240 B258 07 47	7-22 10-9 10-12 7-23 7-23
BRANCH ON COUNT (32) BRANCH ON COUNT (32) BRANCH ON COUNT (64) BRANCH ON COUNT (64) BRANCH ON INDEX HIGH (32)	BCTR BCT BCTGR BCTG BXH	RR RX RRE RXE RS				B B B B B	06 46 B946 E346 86	7-24 7-24 7-24 7-24 7-25
BRANCH ON INDEX HIGH (64) BRANCH ON INDEX LOW OR EQUAL (32) BRANCH ON INDEX LOW OR EQUAL (64) BRANCH RELATIVE AND SAVE BRANCH RELATIVE AND SAVE LONG	BXHG BXLE BXLEG BRAS BRASL	RSE RS RSE RI RIL	N N N N3			B B B B B	EB44 87 EB45 A75 C05	7-25 7-25 7-25 7-26 7-26
BRANCH RELATIVE ON CONDITION BRANCH RELATIVE ON CONDITION LONG BRANCH RELATIVE ON COUNT (32) BRANCH RELATIVE ON COUNT (64) BRANCH RELATIVE ON INDEX HIGH (32)	BRC BRCL BRCT BRCTG BRXH	RI RIL RI RI RSI	N3 N			B B B B B	A74 C04 A76 A77 84	7-26 7-26 7-27 7-27 7-28
BRANCH RELATIVE ON INDEX HIGH (64) BRANCH RELATIVE ON INDEX L OR E (32) BRANCH RELATIVE ON INDEX L OR E (64) CANCEL SUBCHANNEL CHECKSUM	BRXHG BRXLE BRXLG XSCH CKSM	RIE RSI RIE S RRE	N N N C C	P A SP	OP φ GS	B B B	EC44 85 EC45 B276 B241	7-28 7-28 7-28 14-4 7-29
CLEAR SUBCHANNEL COMPARE (拡張精度 BFP) COMPARE (拡張精度 HFP) COMPARE (長精度 BFP) COMPARE (長精度 HFP)	CSCH CXBR CXR CDBR CDB	S RRE RRE RRE RXE	C C C C C	P SP SP A	OP φ GS Da Da Db Xi Db Xi		B230 B349 B369 B319 ED19	14-4 19-24 18-9 19-24 19-24
COMPARE (長精度 HFP) COMPARE (長精度 HFP) COMPARE (短精度 BFP) COMPARE (短精度 BFP) COMPARE (短精度 HFP)	CDR CD CEBR CEB CER	RR RX RRE RXE RR	C C C C C	A A	Da Da Db Xi Db Xi Da		29 B ₂ 69 B309 B ₂ ED09 39	18-9 18-9 19-24 19-24 18-9
COMPARE (短精度 HFP) COMPARE (32) COMPARE (32) COMPARE (64<32) COMPARE (64<32)	CE CR C CGFR CGF	RX RR RX RRE RXE	C C C C C N N	A A	Da		B ₂ 79 19 B ₂ 59 B930 B ₂ E330	18-9 7-33 7-33 7-33 7-33

図 B-1 (2/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ード	ペ ー ジ	
COMPARE (64) COMPARE (64) COMPARE AND FORM CODEWORD COMPARE AND SIGNAL (拡張精度 BFP) COMPARE AND SIGNAL (長精度 BFP)	CGR CG CFC KXBR KDBR	RRE C N RXE C N S C RRE C RRE C	A A SP A SP	II Db Xi Db Xi	GM		I1 B2	B920 E320 B21A B348 B318	7-33 7-33 7-33 19-25 19-25	
COMPARE AND SIGNAL (長精度 BFP) COMPARE AND SIGNAL (短精度 BFP) COMPARE AND SIGNAL (短精度 BFP) COMPARE AND SWAP (32) COMPARE AND SWAP (64)	KDB KEBR KEB CS CSG	RXE C RRE C RXE C RS C RSE C N	A A A SP A SP	Db Xi Db Xi Db Xi \$ \$			B2 B2 B2 ST ST	ED18 B308 ED08 BA EB30	19-25 19-25 19-25 7-40 7-40	
COMPARE AND SWAP AND PURGE COMPARE DECIMAL COMPARE DOUBLE AND SWAP (32) COMPARE DOUBLE AND SWAP (64) COMPARE HALFWORD	CSP CP CDS CDSG CH	RRE C SS C RS C RSE C N RX C	P A ¹ SP A A SP A SP A	\$ Dd \$ \$			ST B1 ST ST	R2 B2 B2 B2 B2	B250 F9 BB EB3E 49	10-17 8-6 7-40 7-40 7-42
COMPARE HALFWORD IMMEDIATE (32) COMPARE HALFWORD IMMEDIATE (64) COMPARE LOGICAL (文字) COMPARE LOGICAL (即値) COMPARE LOGICAL (32)	CHI CGHI CLC CLI CLR	RI C RI C N SS C SI C RR C	A A				B1 B2 B1	A7E A7F D5 95 15	7-42 7-42 7-43 7-43 7-42	
COMPARE LOGICAL (32) COMPARE LOGICAL (64<32) COMPARE LOGICAL (64<32) COMPARE LOGICAL (64) COMPARE LOGICAL (64)	CL CLGFR CLGF CLGR CLG	RX C RRE C N RXE C N RRE C N RXE C N	A A				B2 B2 B2	55 B931 E331 B921 E321	7-43 7-43 7-43 7-43 7-43	
COMPARE LOGICAL C. UNDER MASK (高) COMPARE LOGICAL C. UNDER MASK (低) COMPARE LOGICAL LONG COMPARE LOGICAL LONG EXTENDED COMPARE LOGICAL LONG UNICODE	CLMH CLM CLCL CLCLE CLCLU	RSE C N RS C RR C RS C RSE C E2	A A A SP A SP A SP	II			B2 B2 R1 R2 R1 R3 R1 R2	EB20 BD 0F A9 EB8F	7-43 7-43 7-44 7-46 7-50	
COMPARE LOGICAL STRING COMPARE UNTIL SUBSTRING EQUAL COMPRESSION CALL CONVERT BFP TO HFP (長精度) CONVERT BFP TO HFP (短精度から長精度)	CLST CUSE CMPSC THDR THDER	RRE C RRE C RRE C RRE C RRE C	A SP A SP A SP	II II Dd Da Da	G0 GM GM		ST	R1 R2 R1 R2 R1 R2	B25D B257 B263 B359 B358	7-53 7-54 7-57 9-8 9-8
CONVERT FROM FIXED (32 から拡張精度 BFP) CONVERT FROM FIXED (32 から拡張精度 HFP) CONVERT FROM FIXED (32 から長精度 BFP) CONVERT FROM FIXED (32 から長精度 HFP) CONVERT FROM FIXED (32 から短精度 BFP)	CXFBR CXFR CDFBR CDFR CEFBR	RRE RRE RRE RRE RRE	SP SP	Db Da Db Da Db				B396 B3B6 B395 B3B5 B394	19-27 18-10 19-27 18-10 19-27	
CONVERT FROM FIXED (32 から短精度 HFP) CONVERT FROM FIXED (64 から拡張精度 BFP) CONVERT FROM FIXED (64 から拡張精度 HFP) CONVERT FROM FIXED (64 から長精度 BFP) CONVERT FROM FIXED (64 から長精度 HFP)	CEFR CXGBR CXGR CDGBR CDGR	RRE RRE N RRE N RRE N RRE N	SP SP	Da Db Da Db Da				B3B4 B3A6 B3C6 B3A5 B3C5	18-10 19-27 18-10 19-27 18-10	

図 B-1 (3/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ード	ペ ー ジ
CONVERT FROM FIXED (64 から短精度 BFP)	CEGBR	RRE	N		Db	Xx		B3A4	19-27
CONVERT FROM FIXED (64 から短精度 HFP)	CEGR	RRE	N		Da			B3C4	18-10
CONVERT HFP TO BFP (長精度から短精度)	TBEDR	RRF	C		SP Da			B350	9-9
CONVERT HFP TO BFP (長精度)	TBDR	RRF	C		SP Da			B351	9-9
CONVERT TO BINARY (32)	CVB	RX		A	Dd IK		B ₂	4F	7-68
CONVERT TO BINARY (64)	CVBG	RXE	N	A	Dd IK		B ₂	E30E	7-68
CONVERT TO DECIMAL (32)	CVD	RX		A			ST	B ₂ 4E	7-68
CONVERT TO DECIMAL (64)	CVDG	RXE	N	A			ST	B ₂ E32E	7-68
CONVERT TO FIXED (拡張精度 BFP から 32)	CFXBR	RRF	C		SP Db Xi	Xx		B39A	19-28
CONVERT TO FIXED (拡張精度 BFP から 64)	CGXBR	RRF	C N		SP Db Xi	Xx		B3AA	19-28
CONVERT TO FIXED (拡張精度 HFP から 32)	CFXR	RRF	C		SP Da			B3BA	18-10
CONVERT TO FIXED (拡張精度 HFP から 64)	CGXR	RRF	C N		SP Da			B3CA	18-10
CONVERT TO FIXED (長精度 BFP から 32)	CFDBR	RRF	C		SP Db Xi	Xx		B399	19-28
CONVERT TO FIXED (長精度 BFP から 64)	CGDBR	RRF	C N		SP Db Xi	Xx		B3A9	19-28
CONVERT TO FIXED (長精度 HFP から 32)	CFDR	RRF	C		SP Da			B3B9	18-10
CONVERT TO FIXED (長精度 HFP から 64)	CGDR	RRF	C N		SP Da			B3C9	18-10
CONVERT TO FIXED (短精度 BFP から 32)	CFEBR	RRF	C		SP Db Xi	Xx		B398	19-28
CONVERT TO FIXED (短精度 BFP から 64)	CGEBR	RRF	C N		SP Db Xi	Xx		B3A8	19-28
CONVERT TO FIXED (短精度 HFP から 32)	CFER	RRF	C		SP Da			B3B8	18-10
CONVERT TO FIXED (短精度 HFP から 64)	CGER	RRF	C N		SP Da			B3C8	18-10
CONVERT UNICODE TO UTF-8	CUUTF	RRE	C	A	SP		ST	R ₁ R ₂ B2A6	7-69
CONVERT UTF-8 TO UNICODE	CUTFU	RRE	C	A	SP		ST	R ₁ R ₂ B2A7	7-72
COPY ACCESS	CPYA	RRE					U ₁ U ₂ MD	B24D	7-75
DIAGNOSE			DM	P DM				83	10-18
DIVIDE (拡張精度 BFP)	DXBR	RRE			SP Db Xi Xz Xo Xu Xx			B34D	19-31
DIVIDE (拡張精度 HFP)	DXR	RRE			SP Da EU EO FK			B22D	18-11
DIVIDE (長精度 BFP)	DDBR	RRE			Db Xi Xz Xo Xu Xx			B31D	19-31
DIVIDE (長精度 BFP)	DDB	RXE		A	Db Xi Xz Xo Xu Xx		B ₂	ED1D	19-31
DIVIDE (長精度 HFP)	DDR	RR			Da EU EO FK			2D	18-11
DIVIDE (長精度 HFP)	DD	RX		A	Da EU EO FK		B ₂	6D	18-11
DIVIDE (短精度 BFP)	DEBR	RRE			Db Xi Xz Xo Xu Xx			B30D	19-31
DIVIDE (短精度 BFP)	DEB	RXE		A	Db Xi Xz Xo Xu Xx		B ₂	ED0D	19-31
DIVIDE (短精度 HFP)	DER	RR			Da EU EO FK			3D	18-11
DIVIDE (短精度 HFP)	DE	RX		A	Da EU EO FK		B ₂	7D	18-11
DIVIDE (32<64)	DR	RR			IK			1D	7-75
DIVIDE (32<64)	D	RX		A	SP IK			B ₂ 5D	7-75
DIVIDE DECIMAL	DP	SS		A	SP Dd DK		ST	B ₁ B ₂ FD	8-6
DIVIDE LOGICAL (32<64)	DLR	RRE	N3		SP IK			B997	7-75
DIVIDE LOGICAL (32<64)	DL	RXE	N3	A	SP IK		B ₂	E397	7-75
DIVIDE LOGICAL (64<128)	DLGR	RRE	N		SP IK			B987	7-75
DIVIDE LOGICAL (64<128)	DLG	RXE	N	A	SP IK		B ₂	E387	7-75
DIVIDE SINGLE (64<32)	DSGFR	RRE	N		SP IK			B91D	7-76
DIVIDE SINGLE (64<32)	DSGF	RXE	N	A	SP IK		B ₂	E31D	7-76
DIVIDE SINGLE (64)	DSGR	RRE	N		SP IK			B90D	7-76
DIVIDE SINGLE (64)	DSG	RXE	N	A	SP IK		B ₂	E30D	7-76

図 B-1 (4/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ド	ペ ー ジ	
DIVIDE TO INTEGER (長精度 BFP) DIVIDE TO INTEGER (短精度 BFP) EDIT EDIT AND MARK EXCLUSIVE OR (文字)	DIDBR DIEBR ED EDMK XC	RRF C RRF C SS C SS C SS C		SP SP A A A	Db Xi Db Xi Dd Dd	Xu Xx Xu Xx G1	ST ST ST	B ₁ B ₂ B ₁ B ₂ B ₁ B ₂	B35B B353 DE DF D7	19-32 19-32 8-7 8-9 7-77
EXCLUSIVE OR (即値) EXCLUSIVE OR (32) EXCLUSIVE OR (32) EXCLUSIVE OR (64) EXCLUSIVE OR (64)	XI XR X XGR XG	SI C RR C RX C RRE C N RXE C N		A A A A			ST	B ₁ B ₂ B ₂	97 17 57 B982 E382	7-77 7-77 7-77 7-77 7-77
EXECUTE EXTRACT ACCESS EXTRACT AND SET EXTENDED AUTHORITY EXTRACT FPC EXTRACT PRIMARY ASN	EX EAR ESEA EFPC EPAR	RX RRE RRE N RRE RRE		AI SP P Q	EX Db SO			U ₂	44 B24F B99D B38C B226	7-78 7-79 10-19 19-37 10-19
EXTRACT PSW EXTRACT SECONDARY ASN EXTRACT STACKED REGISTERS (32) EXTRACT STACKED REGISTERS (64) EXTRACT STACKED STATE	EPSW ESAR EREG EREGG ESTA	RRE N3 RRE RRE RRE N RRE C		Q A ¹ A ¹ A ¹ SP	SO SE SE SE			U ₁ U ₂ U ₁ U ₂	B98D B227 B249 B90E B24A	7-79 10-20 10-20 10-20 10-22
HALT SUBCHANNEL HALVE (長精度 HFP) HALVE (短精度 HFP) INSERT ADDRESS SPACE CONTROL INSERT CHARACTER	HSCH HDR HER IAC IC	S C RR RR RRE C RX		P A	OP φ GS Da EU Da EU SO			B ₂	B231 24 34 B224 43	14-6 18-12 18-12 10-24 7-79
INSERT CHARACTERS UNDER MASK (高) INSERT CHARACTERS UNDER MASK (低) INSERT IMMEDIATE (高 高) INSERT IMMEDIATE (高 低) INSERT IMMEDIATE (低 高)	ICMH ICM IIHH IIHL IILH	RSE C N RS C RI N RI N RI N		A A				B ₂ B ₂	EB80 BF A50 A51 A52	7-79 7-79 7-80 7-80 7-80
INSERT IMMEDIATE (低 低) INSERT PROGRAM MASK INSERT PSW KEY INSERT STORAGE KEY EXTENDED INSERT VIRTUAL STORAGE KEY	IILL IPM IPK ISKE IVSK	RI N RRE S RRE RRE		Q P A ¹ Q A ¹	G2 SO			R ₂	A53 B222 B20B B229 B223	7-80 7-81 10-25 10-25 10-26
INVALIDATE PAGE TABLE ENTRY LOAD (拡張精度) LOAD (長精度) LOAD (長精度) LOAD (短精度)	IPTE LXR LDR LD LER	RRE RRE RR RX RR		P A ¹ SP A	\$ Da Da Da Da			B ₂	B221 B365 28 68 38	10-27 9-11 9-11 9-11 9-11
LOAD (短精度) LOAD (32) LOAD (32) LOAD (64<32) LOAD (64<32)	LE LR L LGFR LGF	RX RR RX RRE N RXE N		A A A A	Da			B ₂ B ₂ B ₂	78 18 58 B914 E314	9-11 7-81 7-81 7-81 7-81

図 B-1 (5/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ード	ペ ー ジ	
LOAD (64) LOAD (64) LOAD ACCESS MULTIPLE LOAD ADDRESS LOAD ADDRESS EXTENDED	LGR LG LAM LA LAE	RRE RXE RS RX RX	N N	A A	SP			B ₂ UB 41 51 U ₁ BP	B904 E304 9A 7-81 7-82 7-82	
LOAD ADDRESS RELATIVE LONG LOAD ADDRESS SPACE PARAMETERS LOAD AND TEST (拡張精度 BFP) LOAD AND TEST (拡張精度 HFP) LOAD AND TEST (長精度 BFP)	LARL LASP LTXBR LTXR LTDBR	RIL SSE C RRE C RRE C RRE C	N ³	P A ¹	SP SP SP	SO Db Xi Da Db Xi		B ₁	C00 E500 B342 B362 B312	7-83 10-28 19-37 18-12 19-37
LOAD AND TEST (長精度 HFP) LOAD AND TEST (短精度 BFP) LOAD AND TEST (短精度 HFP) LOAD AND TEST (32) LOAD AND TEST (64<32)	LTDR LTEBR LTER LTR LTGFR	RR C RRE C RR C RR C RRE C				Da Db Xi Da			22 B302 32 12 B912	18-12 19-37 18-12 7-83 7-84
LOAD AND TEST (64) LOAD COMPLEMENT (拡張精度 BFP) LOAD COMPLEMENT (拡張精度 HFP) LOAD COMPLEMENT (長精度 BFP) LOAD COMPLEMENT (長精度 HFP)	LTGR LCXBR LCXR LCDBR LCDR	RRE C N RRE C RRE C RRE C RR C			SP SP	Db Da Db Da			B902 B343 B363 B313 23	7-83 19-37 18-13 19-37 18-13
LOAD COMPLEMENT (短精度 BFP) LOAD COMPLEMENT (短精度 HFP) LOAD COMPLEMENT (32) LOAD COMPLEMENT (64<32) LOAD COMPLEMENT (64)	LCEBR LCER LCR LCGFR LCGR	RRE C RR C RR C RRE C N RRE C N				Db Da IF IF IF			B303 33 13 B913 B903	19-37 18-13 7-84 7-84 7-84
LOAD CONTROL (32) LOAD CONTROL (64) LOAD FP INTEGER (拡張精度 BFP) LOAD FP INTEGER (拡張精度 HFP) LOAD FP INTEGER (長精度 BFP)	LCTL LCTLG FIXBR FIXR FIDBR	RS RSE RRF RRF RRF	N	P A P A	SP SP SP SP SP			B ₂ B ₂	B7 EB2F B347 B367 B35F	10-37 10-37 19-38 18-13 19-38
LOAD FP INTEGER (長精度 HFP) LOAD FP INTEGER (短精度 BFP) LOAD FP INTEGER (短精度 HFP) LOAD FPC LOAD HALFWORD (32)	FIDR FIEBR FIER LFPC LH	RRE RRF RRE S RX			SP SP A SP A	Da Db Xi Da Da Db			B37F B357 B377 B ₂ B ₂	18-13 19-38 18-13 19-39 7-84
LOAD HALFWORD (64) LOAD HALFWORD IMMEDIATE (32) LOAD HALFWORD IMMEDIATE (64) LOAD LENGTHENED (長精度から拡張精度 BFP) LOAD LENGTHENED (長精度から拡張精度 HFP)	LGH LHI LGHI LXDBR LXDB	RXE RI RI RRE RXE	N N	A	SP	Db Xi		B ₂	E315 A78 A79 B305	7-84 7-85 7-85 19-40
LOAD LENGTHENED (長精度から拡張精度 HFP) LOAD LENGTHENED (長精度から拡張精度 HFP) LOAD LENGTHENED (短精度から拡張精度 BFP) LOAD LENGTHENED (短精度から拡張精度 HFP) LOAD LENGTHENED (短精度から拡張精度 HFP)	LXDR LXD LXEBr LXEB LXER	RRE RXE RRE RXE RRE			SP A SP SP A SP SP	Da Da Db Xi Db Xi Da		B ₂	B325 ED25 B306 ED06 B326	18-14 18-14 19-40 19-40 18-14

図 B-1 (6/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性					命 令 コ ード	ペ ー ジ
LOAD LENGTHENED (短精度から拡張精度 HFP)	LXE	RXE		A	SP	Da	B ₂ ED26	18-14
LOAD LENGTHENED (短精度から長精度 BFP)	LDEBR	RRE				Db Xi	B304	19-40
LOAD LENGTHENED (短精度から長精度 BFP)	LDEB	RXE		A		Db Xi	B ₂ ED04	19-40
LOAD LENGTHENED (短精度から長精度 HFP)	LDER	RRE				Da	B324	18-14
LOAD LENGTHENED (短精度から長精度 HFP)	LDE	RXE		A		Da	B ₂ ED24	18-14
LOAD LOGICAL (64<32)	LLGFR	RRE	N				B ₂ B916	7-85
LOAD LOGICAL (64<32)	LLGF	RXE	N	A			E316	7-85
LOAD LOGICAL CHARACTER	LLGC	RXE	N	A			B ₂ E390	7-85
LOAD LOGICAL HALFWORD	LLGH	RXE	N	A			B ₂ E391	7-85
LOAD LOGICAL IMMEDIATE (高 高)	LLIHH	RI	N				A5C	7-86
LOAD LOGICAL IMMEDIATE (高 低)	LLIHL	RI	N				A5D	7-86
LOAD LOGICAL IMMEDIATE (高 高)	LLILH	RI	N				A5E	7-86
LOAD LOGICAL IMMEDIATE (低 低)	LLILL	RI	N				A5F	7-86
LOAD LOGICAL THIRTY ONE BITS	LLGTR	RRE	N				B917	7-86
LOAD LOGICAL THIRTY ONE BITS	LLGT	RXE	N	A			B ₂ E317	7-86
LOAD MULTIPLE (32)	LM	RS		A			B ₂ 98	7-86
LOAD MULTIPLE (64)	LMG	RSE	N	A			B ₂ EB04	7-86
LOAD MULTIPLE DISJOINT	LMD	SS	N	A			B ₂ B4 EF	7-87
LOAD MULTIPLE HIGH	LMH	RSE	N	A			B ₂ EB96	7-87
LOAD NEGATIVE (拡張精度 BFP)	LNXR	RRE	C		SP	Db	B ₂ B341	19-40
LOAD NEGATIVE (拡張精度 HFP)	LNXR	RRE	C		SP	Da	B361	18-15
LOAD NEGATIVE (長精度 BFP)	LNDBR	RRE	C			Db	B311	19-40
LOAD NEGATIVE (長精度 HFP)	LNDR	RR	C			Da	21	18-15
LOAD NEGATIVE (短精度 BFP)	LNEBR	RRE	C			Db	B301	19-40
LOAD NEGATIVE (短精度 HFP)	LNDR	RR	C			Da	31	18-15
LOAD NEGATIVE (32)	LNR	RR	C				11	7-88
LOAD NEGATIVE (64<32)	LNGFR	RRE	C	N			B911	7-88
LOAD NEGATIVE (64)	LNGR	RRE	C	N			B901	7-88
LOAD PAIR FROM QUADWORD	LPQ	RXE	N	A	SP		B ₂ E38F	7-88
LOAD POSITIVE (拡張精度 BFP)	LPXBR	RRE	C		SP	Db	B340	19-41
LOAD POSITIVE (拡張精度 HFP)	LPXR	RRE	C		SP	Da	B360	18-15
LOAD POSITIVE (長精度 BFP)	LPDBR	RRE	C			Db	B310	19-41
LOAD POSITIVE (長精度 HFP)	LPDR	RR	C			Da	20	18-15
LOAD POSITIVE (短精度 BFP)	LPEBR	RRE	C			Db	B300	19-41
LOAD POSITIVE (短精度 HFP)	LPER	RR	C			Da	30	18-15
LOAD POSITIVE (32)	LPR	RR	C			IF	10	7-88
LOAD POSITIVE (64<32)	LPGFR	RRE	C	N		IF	B910	7-88
LOAD POSITIVE (64)	LPGR	RRE	C	N		IF	B900	7-88
LOAD PSW	LPSW	S	L	P	A SP	¢	B ₂ 82	10-37
LOAD PSW EXTENDED	LPSWE	S	L	N	P A SP	¢	B ₂ B2B2	10-38
LOAD REAL ADDRESS (32)	LRA	RX	C	P	A ¹	SO	BP B1	10-39
LOAD REAL ADDRESS (64)	LRAG	RXE	C	N	P A ¹		BP E303	10-39
LOAD REVERSED (16)	LRVH	RXE	N3	A			B ₂ E31F	7-89
LOAD REVERSED (32)	LRVR	RRE	N3				B91F	7-89
LOAD REVERSED (32)	LRV	RXE	N3	A			B ₂ E31E	7-89

図 B-1 (7/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ド	ペ ー ジ	
LOAD REVERSED (64) LOAD REVERSED (64) LOAD ROUNDED (拡張精度から長精度 BFP)	LRVGR LRVG LDXBR	RRE RXE RRE	N N	A	SP	Db Xi Xo Xu Xx	B ₂	B90F E30F B345	7-89 7-89 19-41	
LOAD ROUNDED (拡張精度から長精度 HFP)	LDXR	RR			SP	Da E0		25	18-16	
LOAD ROUNDED (拡張精度から長精度 HFP)	LRDR	RR			SP	Da E0		25	18-16	
LOAD ROUNDED (拡張精度から短精度 BFP)	LEXBR	RRE			SP	Db Xi Xo Xu Xx		B346	19-41	
LOAD ROUNDED (拡張精度から短精度 HFP)	LEXR	RRE			SP	Da E0		B366	18-16	
LOAD ROUNDED (長精度から短精度 BFP)	LEDBR	RRE				Db Xi Xo Xu Xx		B344	19-41	
LOAD ROUNDED (長精度から短精度 HFP)	LEDR	RR				Da E0		35	18-16	
LOAD ROUNDED (長精度から短精度 HFP)	LRER	RR				Da E0		35	18-16	
LOAD USING REAL ADDRESS (32) LOAD USING REAL ADDRESS (64) LOAD ZERO (拡張精度) LOAD ZERO (長精度) LOAD ZERO (短精度)	LURA LURAG LZXR LZDR LZER	RRE RRE RRE RRE RRE	N	P A ¹	SP SP SP	Da Da Da		B24B B905 B376 B375 B374	10-44 10-44 9-11 9-11 9-11	
MODIFY STACKED STATE MODIFY SUBCHANNEL MONITOR CALL MOVE (文字) MOVE (即値)	MSTA MSCH MC MVC MVI	RRE S SI SS SI	C	A ¹ P A	SP SP SP	SE OP φ GS MO	ST	B ₂ B232 AF D2 92	10-44 14-7 7-90 7-91 7-91	
MOVE INVERSE MOVE LONG MOVE LONG EXTENDED MOVE LONG UNICODE MOVE NUMERICS	MVCIN MVCL MVCLE MVCLU MVN	SS RR RS RSE SS	C	A A A A	SP SP SP	II	ST ST ST ST ST	B ₁ B ₂ R ₁ R ₂ R ₁ R ₃ R ₁ R ₂ B ₁ B ₂	E8 0E A8 EB8E D1	7-91 7-92 7-95 7-99 7-102
MOVE PAGE MOVE STRING MOVE TO PRIMARY MOVE TO SECONDARY MOVE WITH DESTINATION KEY	MVPG MVST MVCP MVCS MVCDK	RRE RRE SS SS SSE	C C C C	Q A A Q A Q A Q A	SP SP	G0 G0 SO φ SO φ GM	ST ST ST ST ST	R ₁ R ₂ R ₁ R ₂ DA DB E50F	B254 B255 10-48 10-48 10-50	
MOVE WITH KEY MOVE WITH OFFSET MOVE WITH SOURCE KEY MOVE ZONES MULTIPLY (拡張精度 BFP)	MVCK MVO MVCSK MVZ MXBR	SS SS SSE SS RRE	C	Q A A Q A A		GM	ST ST ST ST ST	B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₂	D9 F1 E50E D3 B34C	10-50 7-104 10-52 7-104 19-42
MULTIPLY (拡張精度 HFP) MULTIPLY (長精度から拡張精度 BFP) MULTIPLY (長精度から拡張精度 BFP) MULTIPLY (長精度から拡張精度 HFP) MULTIPLY (長精度から拡張精度 HFP)	MXR MXDBR MXDB MXDR MXD	RR RRE RXE RR RX			SP SP A SP SP A SP	Da EU E0 Db Xi Db Xi Da EU E0 Da EU E0 Da Xi	B ₂	26 B307 ED07 27 67	18-16 19-42 19-42 18-16 18-16	
MULTIPLY (長精度 BFP) MULTIPLY (長精度 BFP) MULTIPLY (長精度 HFP) MULTIPLY (長精度 HFP) MULTIPLY (短精度から長精度 BFP)	MDBR MDB MDR MD MDEBR	RRE RXE RR RX RRE		A A		Db Xi Xo Xu Xx Db Xi Xo Xu Xx Da EU E0 Da EU E0 Db Xi	B ₂ B ₂	B31C ED1C 2C 6C B30C	19-42 19-42 18-16 18-16 19-42	

図 B-1 (8/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性				命 令 コ ード	ペ ー ジ	
MULTIPLY (短精度から長精度 BFP)	MDEB	RXE		A	Db Xi	B ₂	E00C	19-42
MULTIPLY (短精度から長精度 HFP)	MDER	RR			Da EU EO		3C	18-16
MULTIPLY (短精度から長精度 HFP)	MER	RR			Da EU EO		3C	18-16
MULTIPLY (短精度から長精度 HFP)	MDE	RX		A	Da EU EO	B ₂	7C	18-16
MULTIPLY (短精度から長精度 HFP)	ME	RX		A	Da EU EO	B ₂	7C	18-16
MULTIPLY (短精度 BFP)	MEEBR	RRE			Db Xi Xo Xu Xx		B317	19-42
MULTIPLY (短精度 BFP)	MEEB	RXE		A	Db Xi Xo Xu Xx	B ₂	ED17	19-42
MULTIPLY (短精度 HFP)	MEER	RRE			Da EU EO		B337	18-16
MULTIPLY (短精度 HFP)	MEE	RXE		A	Da EU EO	B ₂	ED37	18-16
MULTIPLY (64<32)	MR	RR		SP			1C	7-105
MULTIPLY (64<32)	M	RX		A SP		B ₂	5C	7-105
MULTIPLY AND ADD (長精度 BFP)	MADBR	RRF			Db Xi Xo Xu Xx		B31E	19-44
MULTIPLY AND ADD (長精度 BFP)	MADB	RXF		A	Db Xi Xo Xu Xx	B ₂	ED1E	19-44
MULTIPLY AND ADD (短精度 BFP)	MAEBR	RRF			Db Xi Xo Xu Xx		B30E	19-44
MULTIPLY AND ADD (短精度 BFP)	MAEB	RXF		A	Db Xi Xo Xu Xx	B ₂	ED0E	19-44
MULTIPLY AND SUBTRACT (長精度 BFP)	MSDBR	RRF			Db Xi Xo Xu Xx		B31F	19-44
MULTIPLY AND SUBTRACT (長精度 BFP)	MSDB	RXF		A	Db Xi Xo Xu Xx	B ₂	ED1F	19-44
MULTIPLY AND SUBTRACT (短精度 BFP)	MSEBR	RRF			Db Xi Xo Xu Xx		B30F	19-44
MULTIPLY AND SUBTRACT (短精度 BFP)	MSEB	RXF		A	Db Xi Xo Xu Xx	B ₂	ED0F	19-44
MULTIPLY DECIMAL	MP	SS		A SP	Dd	ST B ₁ B ₂	FC	8-11
MULTIPLY HALFWORD (32)	MH	RX		A		B ₂	4C	7-106
MULTIPLY HALFWORD IMMEDIATE (32)	MHI	RI					A7C	7-106
MULTIPLY HALFWORD IMMEDIATE (64)	MGHI	RI	N				A7D	7-106
MULTIPLY LOGICAL (128<64)	MLGR	RRE	N	SP			B986	7-106
MULTIPLY LOGICAL (128<64)	MLG	RXE	N	A SP		B ₂	E386	7-106
MULTIPLY LOGICAL (64<32)	MLR	RRE	N3	SP			B996	7-106
MULTIPLY LOGICAL (64<32)	ML	RXE	N3	A SP		B ₂	E396	7-106
MULTIPLY SINGLE (32)	MSR	RRE					B252	7-107
MULTIPLY SINGLE (32)	MS	RX		A		B ₂	71	7-107
MULTIPLY SINGLE (64<32)	MSGFR	RRE	N				B91C	7-107
MULTIPLY SINGLE (64<32)	MSGF	RXE	N	A		B ₂	E31C	7-107
MULTIPLY SINGLE (64)	MSGR	RRE	N				B90C	7-107
MULTIPLY SINGLE (64)	MSG	RXE	N	A		B ₂	E30C	7-107
OR (文字)	OC	SS	C	A		ST B ₁ B ₂	D6	7-108
OR (即値)	OI	SI	C	A		ST B ₁	96	7-108
OR (32)	OR	RR	C				16	7-108
OR (32)	O	RX	C	A		B ₂	56	7-108
OR (64)	OGR	RRE	C N				B981	7-108
OR (64)	OG	RXE	C N	A		B ₂	E381	7-108
OR IMMEDIATE (高 高)	OIHH	RI	C N				A58	7-109
OR IMMEDIATE (高 低)	OIHL	RI	C N				A59	7-109
OR IMMEDIATE (低 高)	OILH	RI	C N				A5A	7-109
OR IMMEDIATE (低 低)	OILL	RI	C N				A5B	7-109
PACK	PACK	SS		A		ST B ₁ B ₂	F2	7-109
PACK ASCII	PKA	SS	E2	A SP		ST B ₁ B ₂	E9	7-110

図 B-1 (9/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ード	ページ
PACK UNICODE PAGE IN PAGE OUT PERFORM LOCKED OPERATION PROGRAM CALL	PKU PGIN PGOUT PLO PC	SS RRE C RRE C SS C S	E2 ES ES	A SP P A ¹ P A ¹ A SP Q A ¹		¢ ¢ ¢ GM ¢ GM	ST ST ST B ST	B ₁ B ₂ FC	E1 7-111 B22E 10-53 B22F 10-54 EE 7-112 B218 10-55
PROGRAM RETURN PROGRAM TRANSFER PURGE ALB PURGE TLB RESET CHANNEL PATH	PR PT PALB PTLB RCHP	E L RRE RRE S S C		Q A ¹ SP Q A ¹ SP P P P	Z ⁴ T ¢ ² Z ² T ¢ \$ \$ OP ¢ G1	B B		0101 10-68 B228 10-72 B248 10-78 B20D 10-78 B23B 14-8	
RESET REFERENCE BIT EXTENDED RESUME PROGRAM RESUME SUBCHANNEL ROTATE LEFT SINGLE LOGICAL (32) ROTATE LEFT SINGLE LOGICAL (64)	RRBE RP RSCH RLL RLLG	RRE C S L S C RSE RSE		P A ¹ Q A SP P P N3 N	WE T OP ¢ GS	B	B ₂	B22A 10-78 B277 10-79 B238 14-10 EB1D 7-127 EB1C 7-127	
SEARCH STRING SET ACCESS SET ADDRESS LIMIT SET ADDRESS SPACE CONTROL SET ADDRESS SPACE CONTROL FAST	SRST SAR SAL SAC SACF	RRE C RRE S S S		A SP P Q SP Q SP	G0 OP ¢ G1 SW ¢ SW		U ₁ R ₂	B25E 7-128 B24E 7-129 B237 14-11 B219 10-81 B279 10-81	
SET ADDRESSING MODE (24) SET ADDRESSING MODE (31) SET ADDRESSING MODE (64) SET CHANNEL MONITOR SET CLOCK	SAM24 SAM31 SAM64 SCHM SCK	E N3 E N3 E N S S C		SP T SP T T P A SP P A SP	T T T OP ¢ GM		B ₂	010C 7-129 010D 7-129 010E 7-129 B23C 14-12 B204 10-83	
SET CLOCK COMPARATOR SET CLOCK PROGRAMMABLE FIELD SET CPU TIMER SET FPC SET PREFIX	SCKC SCKPF SPT SFPC SPX	S E S RRE S		P A SP P SP P A SP P SP P A SP	G0 Db \$		B ₂ B ₂ B ₂	B206 10-84 0107 10-84 B208 10-84 B384 19-46 B210 10-85	
SET PROGRAM MASK SET PSW KEY FROM ADDRESS SET ROUNDING MODE SET SECONDARY ASN SET STORAGE KEY EXTENDED	SPM SPKA SRNM SSAR SSKE	RR L S S RRE RRE		Q A ¹ P A ¹	Db Z ³ T ¢ ¢			04 7-130 B20A 10-85 B299 19-47 B225 10-86 B22B 10-89	
SET SYSTEM MASK SHIFT AND ROUND DECIMAL SHIFT LEFT DOUBLE SHIFT LEFT DOUBLE LOGICAL SHIFT LEFT SINGLE (32)	SSM SRP SLDA SLDL SLA	S SS C RS C RS RS C		P A SP A SP SP	S0 Dd DF IF IF	ST	B ₁ B ₂	80 10-89 F0 8-11 8F 7-130 8D 7-131 8B 7-132	
SHIFT LEFT SINGLE (64) SHIFT LEFT SINGLE LOGICAL (32) SHIFT LEFT SINGLE LOGICAL (64) SHIFT RIGHT DOUBLE SHIFT RIGHT DOUBLE LOGICAL	SLAG SLL SLLG SRDA SRDL	RSE C N RS RSE N RS C RS			IF			EB0B 7-132 89 7-132 EB0D 7-133 8E 7-133 8C 7-133	

図 B-1 (10/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ド	ペ ー ジ	
SHIFT RIGHT SINGLE (32) SHIFT RIGHT SINGLE (64) SHIFT RIGHT SINGLE LOGICAL (32) SHIFT RIGHT SINGLE LOGICAL (64) SIGNAL PROCESSOR	SRA SRAG SRL SRLG SIGP	RS C RSE C N RS RSE N RS C						8A EB0A 88 EB0C AE	7-134 7-134 7-135 7-135 10-89	
SQUARE ROOT (拡張精度 BFP) SQUARE ROOT (拡張精度 HFP) SQUARE ROOT (長精度 BFP) SQUARE ROOT (長精度 HFP) SQUARE ROOT (長精度 HFP)	SQXBR SQXR SQDBR SQDB SQDR	RRE RRE RRE RRE RRE		SP SP	Db Xi Da	Xx SQ		B2 B2	B316 B336 B315 ED15 B244	19-47 18-18 19-47 19-47 18-18
SQUARE ROOT (長精度 HFP) SQUARE ROOT (短精度 BFP) SQUARE ROOT (短精度 BFP) SQUARE ROOT (短精度 HFP) SQUARE ROOT (短精度 HFP)	SQD SQEBR SQEB SQER SQE	RXE RRE RRE RRE RXE	A A A A		Da Db Xi Da Da	SQ Xx SQ Xx SQ		B2 B2 B2 B2	ED35 B314 ED14 B245 ED34	18-18 19-47 19-47 18-18 18-18
START SUBCHANNEL STORE (長精度) STORE (短精度) STORE (32) STORE (64)	SSCH STD STE ST STG	S C RX RX RX RXE N	P A A A A A	SP	OP Da Da	¢ GS		B2 B2 B2 B2 B2	B233 60 70 50 E324	14-14 9-11 9-11 7-135 7-135
STORE ACCESS MULTIPLE STORE CHANNEL PATH STATUS STORE CHANNEL REPORT WORD STORE CHARACTER STORE CHARACTERS UNDER MASK (高)	STAM STCPS STCRW STC STCMH	RS S S C RX RSE N	P A P A P A A A	SP SP SP		¢ ¢		UB B2 B2 B2 B2	9B B23A B239 42 EB2C	7-135 14-15 14-16 7-136 7-136
STORE CHARACTERS UNDER MASK (低) STORE CLOCK STORE CLOCK COMPARATOR STORE CLOCK EXTENDED STORE CONTROL (32)	STCM STCK STCKC STCKE STCTL	RS S C S S C RS	A A P A A P A	SP		\$ \$		B2 B2 B2 B2 B2	BE B205 B207 B278 B6	7-136 7-137 10-91 7-137 10-91
STORE CONTROL (64) STORE CPU ADDRESS STORE CPU ID STORE CPU TIMER STORE FACILITY LIST	STCTG STAP STIDP STPT STFL	RSE N S S S S N3	P A P A P A P A P	SP SP SP SP				B2 B2 B2 B2 B2	EB25 B212 B202 B209 B2B1	10-91 10-91 10-92 10-93 10-93
STORE FPC STORE HALFWORD STORE MULTIPLE (32) STORE MULTIPLE (64) STORE MULTIPLE HIGH	STFPC STH STM STMG STMH	S RX RS RSE N RSE N	A A A A A		Db			B2 B2 B2 B2 B2	B29C 40 90 EB24 EB26	19-47 7-139 7-139 7-139 7-140
STORE PAIR TO QUADWORD STORE PREFIX STORE REAL ADDRESS STORE REVERSED (16) STORE REVERSED (32)	STPQ STPX STRAG STRVH STRV	RXE N S SSE N RXE N3 RXE N3	A P A N A A	SP SP A ¹ A A				B2 B2 B1 B2 B2	E38E B211 E502 E33F E33E	7-140 10-93 10-94 7-140 7-140

図 B-1 (11/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ード	ペ ー ジ		
STORE REVERSED (64) STORE SUBCHANNEL STORE SYSTEM INFORMATION STORE THEN AND SYSTEM MASK STORE THEN OR SYSTEM MASK	STRVG STSCH STSI STNSM STOSM	RXE S C S C SI SI	N	A P A SP P A SP P A P A SP	OP	φ	GS GM	ST ST ST ST ST ST B ₁	B ₂ B ₂ B ₂ AC AD	E32F B234 B27D AC AD	7-140 14-17 10-95 10-105 10-105
STORE USING REAL ADDRESS (32) STORE USING REAL ADDRESS (64) SUBTRACT (拡張精度 BFP) SUBTRACT (長精度 BFP) SUBTRACT (長精度 BFP)	STURA STURG SXBR SDBR SDB	RRE RRE RRE C RRE C RXE C	N	P A ¹ SP P A ¹ SP SP A			Db Xi Xo Xu Xx Db Xi Xo Xu Xx Db Xi Xo Xu Xx	SU SU	B ₂ B ₂ B ₂	B246 B925 B34B B31B ED1B	10-105 10-105 19-48 19-48 19-48
SUBTRACT (短精度 BFP) SUBTRACT (短精度 BFP) SUBTRACT (32) SUBTRACT (32) SUBTRACT (64<32)	SEBR SEB SR S SGFR	RRE C RXE C RR C RX C RRE C N		A A A A			Db Xi Xo Xu Xx Db Xi Xo Xu Xx IF IF IF		B ₂ B ₂	B30B ED0B 1B 5B B919	19-48 19-48 7-141 7-141 7-141
SUBTRACT (64<32) SUBTRACT (64) SUBTRACT (64) SUBTRACT DECIMAL SUBTRACT HALFWORD	SGF SGR SG SP SH	RXE C N RRE C N RXE C N SS C RX C		A A A A			IF IF IF Dd DF IF	ST	B ₂ B ₂ B ₂ B ₁ B ₂	E319 B909 E309 FB 4B	7-141 7-141 7-141 8-12 7-142
SUBTRACT LOGICAL (32) SUBTRACT LOGICAL (32) SUBTRACT LOGICAL (64<32) SUBTRACT LOGICAL (64<32) SUBTRACT LOGICAL (64)	SLR SL SLGFR SLGF SLGR	RR C RX C RRE C N RXE C N RRE C N		A A A					B ₂ B ₂	1F 5F B91B E31B B90B	7-142 7-142 7-142 7-142 7-142
SUBTRACT LOGICAL (64) SUBTRACT LOGICAL WITH BORROW (32) SUBTRACT LOGICAL WITH BORROW (32) SUBTRACT LOGICAL WITH BORROW (64) SUBTRACT LOGICAL WITH BORROW (64)	SLG SLBR SLB SLBGR SLBG	RXE C N RRE C N3 RXE C N3 RRE C N RXE C N		A A A A					B ₂ B ₂ B ₂ B ₂	E30B B999 E399 B989 E389	7-142 7-143 7-143 7-143 7-143
SUBTRACT NORMALIZED (拡張精度 HFP) SUBTRACT NORMALIZED (長精度 HFP) SUBTRACT NORMALIZED (長精度 HFP) SUBTRACT NORMALIZED (短精度 HFP) SUBTRACT NORMALIZED (短精度 HFP)	SXR SDR SD SER SE	RR C RR C RX C RR C RX C		SP A A			Da EU EO LS Da EU EO LS Da EU EO LS Da EU EO LS Da EU EO LS		B ₂ B ₂	37 2B 6B 3B 7B	18-19 18-19 18-19 18-19 18-19
SUBTRACT UNNORMALIZED (長精度 HFP) SUBTRACT UNNORMALIZED (長精度 HFP) SUBTRACT UNNORMALIZED (短精度 HFP) SUBTRACT UNNORMALIZED (短精度 HFP) SUPERVISOR CALL	SWR SW SUR SU SVC	RR C RX C RR C RX C RR		A A A A			Da EO LS Da EO LS Da EO LS Da EO LS φ		B ₂ B ₂	2F 6F 3F 7F 0A	18-20 18-20 18-20 18-20 7-144
TEST ACCESS TEST ADDRESSING MODE TEST AND SET TEST BLOCK TEST DATA CLASS (拡張精度 BFP)	TAR TAM TS TB TCXB	RRE C E C N3 S C RRE C RXE C		A ¹ A P A ¹ SP			II \$ G0 Db	ST	U ₁ B ₂	B24C 010B 93 B22C ED12	10-106 7-144 7-144 10-108 19-48

図 B-1 (12/13). 名称順の命令一覧表

名前	二 モ ニ ック	特性						命 令 コ ド	ペ ー ジ
TEST DATA CLASS (長精度 BFP) TEST DATA CLASS (短精度 BFP) TEST DECIMAL TEST PENDING INTERRUPTION TEST PROTECTION	TCDB TCEB TP TPI TPROT	RXE C RXE C RSL C E2 S C SSE C		A A ¹ A ¹	SP	Db Db ¢	ST	B ₁ B ₂ B ₁	ED11 19-48 ED10 19-48 EBC0 8-12 B236 14-17 E501 10-110
TEST SUBCHANNEL TEST UNDER MASK TEST UNDER MASK (高 高) TEST UNDER MASK (高 低) TEST UNDER MASK (低 高)	TSCH TM TMHH TMHL TMLH	S C SI C RI C N RI C N RI C N		A A	SP	OP ¢ GS	ST	B ₂ B ₁	B235 14-19 91 7-145 A72 7-145 A73 7-145 A70 7-145
TEST UNDER MASK (低 低) TEST UNDER MASK HIGH TEST UNDER MASK LOW TRACE (32) TRACE (64)	TMLL TMH TML TRACE TRACG	RI C N RI C RI C RS RSE N			SP	T ¢ T ¢		B ₂ B ₂	A71 7-145 A70 7-145 A71 7-145 99 10-113 EB0F 10-113
TRANSLATE TRANSLATE AND TEST TRANSLATE EXTENDED TRANSLATE ONE TO ONE TRANSLATE ONE TO TWO	TR TRT TRE TROO TROT	SS C SS C RRE C RRE C E2 RRE C E2		A A A SP A SP A SP		GM GM GM	ST ST ST ST ST	B ₁ B ₂ B ₁ B ₂ R ₁ R ₂ R ₁ R ₂ R ₁ R ₂	DC 7-146 DD 7-147 B2A5 7-148 B993 7-150 B992 7-150
TRANSLATE TWO TO ONE TRANSLATE TWO TO TWO TRAP TRAP UNPACK	TRTO TRTT TRAP2 TRAP4 UNPK	RRE C E2 RRE C E2 E S SS		A SP A SP A A A		GM GM SO T SO T	ST ST ST ST ST	RM R ₂ RM R ₂ B B B ₁ B ₂	B991 7-150 B990 7-150 01FF 10-114 B2FF 10-114 F3 7-155
UNPACK ASCII UNPACK UNICODE UPDATE TREE ZERO AND ADD	UNPKA UNPKU UPT ZAP	SS C E2 SS C E2 E C SS C		A SP A SP A SP A		GM II Dd DF	ST ST ST ST	B ₁ B ₂ B ₁ B ₂ I4 B ₁ B ₂	EA 7-155 E2 7-156 0102 7-157 F8 8-13

図 B-1 (13/13). 名称順の命令一覧表

ニーモニック	名前	特性					命令コード	ページ
A	DIAGNOSE		DM	P DM		MD	B3	10-18
AD	ADD (32)	RX	C	A	IF	B ₂	5A	7-16
ADB	ADD NORMALIZED (長精度 HFP)	RX	C	A	Da EU EO LS	B ₂	6A	18-7
ADBR	ADD (長精度 BFP)	RXE	C	A	Db Xi Xo Xu Xx	B ₂	ED1A	19-19
	ADD (長精度 HFP)	RRE	C		Db Xi Xo Xu Xx	B ₂	B31A	19-19
ADR	ADD NORMALIZED (長精度 HFP)	RR	C		Da EU EO LS		2A	18-7
AE	ADD NORMALIZED (短精度 HFP)	RX	C	A	Da EU EO LS	B ₂	7A	18-7
AEB	ADD (短精度 BFP)	RXE	C	A	Db Xi Xo Xu Xx	B ₂	ED0A	19-19
AEBR	ADD (短精度 BFP)	RRE	C		Db Xi Xo Xu Xx		B30A	19-19
AER	ADD NORMALIZED (短精度 HFP)	RR	C		Da EU EO LS		3A	18-7
AG	ADD (64)	RXE	C N	A	IF	B ₂	E308	7-16
AGF	ADD (64<32)	RXE	C N	A	IF	B ₂	E318	7-16
AGFR	ADD (64<32)	RRE	C N		IF		B918	7-16
AGHI	ADD HALFWORD IMMEDIATE (64)	RI	C N		IF		A7B	7-16
AGR	ADD (64)	RRE	C N		IF		B908	7-16
AH	ADD HALFWORD	RX	C	A	IF	B ₂	4A	7-16
AHI	ADD HALFWORD IMMEDIATE (32)	RI	C		IF		A7A	7-16
AL	ADD LOGICAL (32)	RX	C	A		B ₂	5E	7-17
ALC	ADD LOGICAL WITH CARRY (32)	RXE	C N3	A		B ₂	E398	7-18
ALCG	ADD LOGICAL WITH CARRY (64)	RXE	C N	A		B ₂	E388	7-18
ALCGR	ADD LOGICAL WITH CARRY (64)	RRE	C N				B988	7-17
ALCR	ADD LOGICAL WITH CARRY (32)	RRE	C N3				B998	7-17
ALG	ADD LOGICAL (64)	RXE	C N	A		B ₂	E30A	7-17
ALGF	ADD LOGICAL (64<32)	RXE	C N	A		B ₂	E31A	7-17
ALGFR	ADD LOGICAL (64<32)	RRE	C N				B91A	7-17
ALGR	ADD LOGICAL (64)	RRE	C N				B90A	7-17
ALR	ADD LOGICAL (32)	RR	C			ST B ₁ B ₂	1E	7-17
AP	ADD DECIMAL	SS	C	A	Dd DF		FA	8-5
AR	ADD (32)	RR	C		IF		1A	7-16
AU	ADD UNNORMALIZED (短精度 HFP)	RX	C	A	Da EO LS	B ₂	7E	18-8
AUR	ADD UNNORMALIZED (短精度 HFP)	RR	C		Da EO LS		3E	18-8
AW	ADD UNNORMALIZED (長精度 HFP)	RX	C	A	Da EO LS	B ₂	6E	18-8
AWR	ADD UNNORMALIZED (長精度 HFP)	RR	C		Da EO LS		2E	18-8
AXBR	ADD (拡張精度 BFP)	RRE	C	SP	Db Xi Xo Xu Xx		B34A	19-19
AXR	ADD NORMALIZED (拡張精度 HFP)	RR	C	SP	Da EU EO LS		36	18-7
BAKR	BRANCH AND STACK	RRE		A ¹	Z ⁵ T	B ST	B240	10-9
BAL	BRANCH AND LINK	RX				B	45	7-20
BALR	BRANCH AND LINK	RR			T	B	05	7-20
BAS	BRANCH AND SAVE	RX				B	4D	7-21
BASR	BRANCH AND SAVE	RR			T	B	0D	7-20
BASSM	BRANCH AND SAVE AND SET MODE	RR			T	B	0C	7-21
BC	BRANCH ON CONDITION	RX				B	47	7-23
BCR	BRANCH ON CONDITION	RR			φ ¹	B	07	7-23
BCT	BRANCH ON COUNT (32)	RX				B	46	7-24
BCTG	BRANCH ON COUNT (64)	RXE	N			B	E346	7-24

図 B-2 (1/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ
BCTGR BCTR BRAS BRASL BRC	BRANCH ON COUNT (64) BRANCH ON COUNT (32) BRANCH RELATIVE AND SAVE BRANCH RELATIVE AND SAVE LONG BRANCH RELATIVE ON CONDITION	RRE RR RI RIL RI	N N3 			B B B B B		B946 06 A75 C05 A74	7-24 7-24 7-26 7-26 7-26
BRCL BRCT BRCTG BRXH BRXHG	BRANCH RELATIVE ON CONDITION LONG BRANCH RELATIVE ON COUNT (32) BRANCH RELATIVE ON COUNT (64) BRANCH RELATIVE ON INDEX HIGH (32) BRANCH RELATIVE ON INDEX HIGH (64)	RIL RI RI RSI RIE	N3 N N			B B B B B		C04 A76 A77 84 EC44	7-26 7-27 7-27 7-28 7-28
BRXLE BRXLG BSA BSG BSM	BRANCH RELATIVE ON INDEX L OR E (32) BRANCH RELATIVE ON INDEX L OR E (64) BRANCH AND SET AUTHORITY BRANCH IN SUBSPACE GROUP BRANCH AND SET MODE	RSI RIE RRE RRE RR	 N 	Q A ¹ A ¹	SO T SO T T	B B B B B	R ₂	85 EC45 B25A B258 0B	7-28 7-28 10-5 10-12 7-22
BXH BXHG BXLE BXLEG C	BRANCH ON INDEX HIGH (32) BRANCH ON INDEX HIGH (64) BRANCH ON INDEX LOW OR EQUAL (32) BRANCH ON INDEX LOW OR EQUAL (64) COMPARE (32)	RS RSE RS RSE RX	 N N C	 A		B B B B B	B ₂	86 EB44 87 EB45 59	7-25 7-25 7-25 7-25 7-33
CD CDB CDBR CDFBR CDFR	COMPARE (長精度 HFP) COMPARE (長精度 BFP) COMPARE (長精度 BFP) CONVERT FROM FIXED (32 から長精度 BFP) CONVERT FROM FIXED (32 から長精度 HFP)	RX RXC RXC RRE RRE	C C C 	A A	Da Db Xi Db Xi Db		B ₂ B ₂	69 ED19 B319 B395	18-9 19-24 19-24 19-27
CDGBR CDGR CDR CDS CDSG	CONVERT FROM FIXED (64 から長精度 BFP) CONVERT FROM FIXED (64 から長精度 HFP) COMPARE (長精度 HFP) COMPARE DOUBLE AND SWAP (32) COMPARE DOUBLE AND SWAP (64)	RRE RRE RR RS RSE	N N C C N	 A SP A SP	Da Da \$ \$	 ST ST	B ₂ B ₂	B3A5 B3C5 29 BB EB3E	19-27 18-10 18-9 7-40 7-40
CE CEB CEBR CEFBR CEFR	COMPARE (短精度 HFP) COMPARE (短精度 BFP) COMPARE (短精度 BFP) CONVERT FROM FIXED (32 から短精度 BFP) CONVERT FROM FIXED (32 から短精度 HFP)	RX RXE RXC RRE RRE	C C C 	A A	Da Db Xi Db Xi Db	 Xx	B ₂ B ₂	79 ED09 B309 B394	18-9 19-24 19-24 19-27
CEGBR CEGR CER CFC CFDBR	CONVERT FROM FIXED (64 から短精度 BFP) CONVERT FROM FIXED (64 から短精度 HFP) COMPARE (短精度 HFP) COMPARE AND FORM CODEWORD CONVERT TO FIXED (長精度 BFP から 32)	RRE RRE RR S RRF	N N C C C	 A SP A SP	Da Da II Db Xi	 GM Xx	 II	B3A4 B3C4 39 B21A B399	19-27 18-10 18-9 7-33 19-28
CFDR CFEBR CFER CFXBR CFXR	CONVERT TO FIXED (長精度 HFP から 32) CONVERT TO FIXED (短精度 BFP から 32) CONVERT TO FIXED (短精度 HFP から 32) CONVERT TO FIXED (拡張精度 BFP から 32) CONVERT TO FIXED (拡張精度 HFP から 32)	RRF RRF RRF RRF RRF	C C C C C	SP SP SP SP SP	Da Db Xi Da Db Xi Da	 Xx Xx		B3B9 B398 B3B8 B39A B3BA	18-10 19-28 18-10 19-28 18-10

図 B-2 (2/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ
CG CGDBR	COMPARE (64) CONVERT TO FIXED (長精度 BFP から 64)	RXE C N RRF C N	A	SP	Db Xi	Xx	B ₂	E320 B3A9	7-33 19-28
CGDR	CONVERT TO FIXED (長精度 HFP から 64)	RRF C N		SP	Da			B3C9	18-10
CGEBR	CONVERT TO FIXED (短精度 BFP から 64)	RRF C N		SP	Db Xi	Xx		B3A8	19-28
CGER	CONVERT TO FIXED (短精度 HFP から 64)	RRF C N		SP	Da			B3C8	18-10
CGF CGFR CGHI CGR CGXBR	COMPARE (64<32) COMPARE (64<32) COMPARE HALFWORD IMMEDIATE (64) COMPARE (64) CONVERT TO FIXED (拡張精度 BFP から 64)	RXE C N RRE C N RI C N RRE C N RRF C N	A				B ₂	E330 B930 A7F B920 B3AA	7-33 7-33 7-42 7-33 19-28
CGXR	CONVERT TO FIXED (拡張精度 HFP から 64)	RRF C N		SP	Da			B3CA	18-10
CH CHI CKSM CL	COMPARE HALFWORD COMPARE HALFWORD IMMEDIATE (32) CHECKSUM COMPARE LOGICAL (32)	RX C RI C RRE C RX C	A	SP			B ₂ R ₂ B ₂	49 A7E B241 55	7-42 7-42 7-29 7-43
CLC CLCL CLCLE CLCLU CLG	COMPARE LOGICAL (文字) COMPARE LOGICAL LONG COMPARE LOGICAL LONG EXTENDED COMPARE LOGICAL LONG UNICODE COMPARE LOGICAL (64)	SS C RR C RS C RSE C E2 RXE C N	A	SP	II		B ₁ B ₂ R ₁ R ₂ R ₁ R ₃ R ₁ R ₂ B ₂	D5 0F A9 EB8F E321	7-43 7-44 7-46 7-50 7-43
CLGF CLGFR CLGR CLI CLM	COMPARE LOGICAL (64<32) COMPARE LOGICAL (64<32) COMPARE LOGICAL (64) COMPARE LOGICAL (即値) COMPARE LOGICAL C. UNDER MASK (低)	RXE C N RRE C N RRE C N SI C RS C	A				B ₂ B ₁ B ₂	E331 B931 B921 95 BD	7-43 7-43 7-43 7-43 7-43
CLMH CLR CLST CMPSC CP	COMPARE LOGICAL C. UNDER MASK (高) COMPARE LOGICAL (32) COMPARE LOGICAL STRING COMPRESSION CALL COMPARE DECIMAL	RSE C N RR C RRE C RRE C SS C	A	SP		G0 GM	B ₂ R ₁ R ₂ R ₁ R ₂ B ₁ B ₂	EB20 15 B25D B263 F9	7-43 7-42 7-53 7-57 8-6
CPYA CR CS CSCH CSG	COPY ACCESS COMPARE (32) COMPARE AND SWAP (32) CLEAR SUBCHANNEL COMPARE AND SWAP (64)	RRE RR C RS C S C RSE C N	P	A SP	OP	\$ ¢ \$	U ₁ U ₂ B ₂	B24D 19 BA B230 EB30	7-75 7-33 7-40 14-4 7-40
CSP CUSE CUTFU CUUTF CVB	COMPARE AND SWAP AND PURGE COMPARE UNTIL SUBSTRING EQUAL CONVERT UTF-8 TO UNICODE CONVERT UNICODE TO UTF-8 CONVERT TO BINARY (32)	RRE C RRE C RRE C RRE C RX	P A ¹	A SP	II	\$ GM	ST R ₁ R ₂ R ₁ R ₂ R ₁ R ₂ B ₂	B250 B257 B2A7 B2A6 4F	10-17 7-54 7-72 7-69 7-68
CVBG CVD CVDG CXBR CXFBR	CONVERT TO BINARY (64) CONVERT TO DECIMAL (32) CONVERT TO DECIMAL (64) COMPARE (拡張精度 BFP) CONVERT FROM FIXED (32 から拡張精度 BFP)	RXE N RX RXE N RRE C RRE	A A A	Dd	IK		ST ST B ₂ B ₂	E30E 4E E32E B349 B396	7-68 7-68 7-68 19-24 19-27

図 B-2 (3/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ
CXFR	CONVERT FROM FIXED (32 から拡張精度 HFP)	RRE		SP	Da			B3B6	18-10
CXGBR	CONVERT FROM FIXED (64 から拡張精度 BFP)	RRE	N	SP	Db			B3A6	19-27
CXGR	CONVERT FROM FIXED (64 から拡張精度 HFP)	RRE	N	SP	Da			B3C6	18-10
CXR D	COMPARE (拡張精度 HFP) DIVIDE (32<64)	RRE C RX		SP A SP	Da IK			B369 B2 5D	18-9 7-75
DD	DIVIDE (長精度 HFP)	RX		A	Da EU EO FK			B2 6D	18-11
DDB	DIVIDE (長精度 BFP)	RXE		A	Db Xi Xz Xo Xu Xx			B2 ED1D	19-31
DDBR	DIVIDE (長精度 BFP)	RRE			Db Xi Xz Xo Xu Xx			B31D	19-31
DDR	DIVIDE (長精度 HFP)	RR			Da EU EO FK			2D	18-11
DE	DIVIDE (短精度 HFP)	RX		A	Da EU EO FK			B2 7D	18-11
DEB	DIVIDE (短精度 BFP)	RXE		A	Db Xi Xz Xo Xu Xx			B2 E00D	19-31
DEBR	DIVIDE (短精度 BFP)	RRE			Db Xi Xz Xo Xu Xx			B30D	19-31
DER	DIVIDE (短精度 HFP)	RR			Da EU EO FK			3D	18-11
DIDBR	DIVIDE TO INTEGER (長精度 BFP)	RRF C		SP	Db Xi Xu Xx			B35B	19-32
DIEBR	DIVIDE TO INTEGER (短精度 BFP)	RRF C		SP	Db Xi Xu Xx			B353	19-32
DL	DIVIDE LOGICAL (32<64)	RXE	N3	A SP	IK			B2 E397	7-75
DLG	DIVIDE LOGICAL (64<128)	RXE	N	A SP	IK			B2 E387	7-75
DLGR	DIVIDE LOGICAL (64<128)	RRE	N	SP	IK			B987	7-75
DLR	DIVIDE LOGICAL (32<64)	RRE	N3	SP	IK			B997	7-75
DP	DIVIDE DECIMAL	SS		A SP	Dd DK		ST	B1 B2 FD	8-6
DR	DIVIDE (32<64)	RR		SP	IK			B2 1D	7-75
DSG	DIVIDE SINGLE (64)	RXE	N	A SP	IK			B2 E30D	7-76
DSGF	DIVIDE SINGLE (64<32)	RXE	N	A SP	IK			B2 E31D	7-76
DSGFR	DIVIDE SINGLE (64<32)	RRE	N	SP	IK			B91D	7-76
DSGR	DIVIDE SINGLE (64)	RRE	N	SP	IK			B90D	7-76
DXBR	DIVIDE (拡張精度 BFP)	RRE		SP	Db Xi Xz Xo Xu Xx			B34D	19-31
DXR	DIVIDE (拡張精度 HFP)	RRE		SP	Da EU EO FK			B22D	18-11
EAR	EXTRACT ACCESS	RRE						U2 B24F	7-79
ED	EDIT	SS C		A	Dd		ST	B1 B2 DE	8-7
EDMK	EDIT AND MARK	SS C		A	Dd G1		ST	B1 B2 DF	8-9
EFPC	EXTRACT FPC	RRE			Db			B38C	19-37
EPAR	EXTRACT PRIMARY ASN	RRE		Q	S0			B226	10-19
EPSW	EXTRACT PSW	RRE	N3					B98D	7-79
EREG	EXTRACT STACKED REGISTERS (32)	RRE		A ¹	SE		U1 U2	B249	10-20
EREGG	EXTRACT STACKED REGISTERS (64)	RRE	N	A ¹	SE		U1 U2	B90E	10-20
ESAR	EXTRACT SECONDARY ASN	RRE		Q	S0			B227	10-20
ESEA	EXTRACT AND SET EXTENDED AUTHORITY	RRE	N	P				B99D	10-19
ESTA	EXTRACT STACKED STATE	RRE C		A ¹ SP	SE			B24A	10-22
EX	EXECUTE	RX		AI SP	EX			44	7-78
FIDBR	LOAD FP INTEGER (長精度 BFP)	RRF		SP	Db Xi Xx			B35F	19-38
FIDR	LOAD FP INTEGER (長精度 HFP)	RRE		SP	Da			B37F	18-13
FIEBR	LOAD FP INTEGER (短精度 BFP)	RRF		SP	Db Xi Xx			B357	19-38
FIER	LOAD FP INTEGER (短精度 HFP)	RRE			Da			B377	18-13
FIXBR	LOAD FP INTEGER (拡張精度 BFP)	RRF		SP	Db Xi Xx			B347	19-38
FIXR	LOAD FP INTEGER (拡張精度 HFP)	RRE		SP	Da			B367	18-13

図 B-2 (4/13). ニーモニック順の命令一覧表

ニーモニツク	名前	特性					命令コード	ページ
HDR HER HSC IAC IC	HALVE (長精度 HFP) HALVE (短精度 HFP) HALT SUBCHANNEL INSERT ADDRESS SPACE CONTROL INSERT CHARACTER	RR RR S C RRE C RX	P Q A	Da EU Da EU OP ¢ GS SO			24 34 B231 B224 43	18-12 18-12 14-6 10-24 7-79
ICM ICMH IIHH IIHL IILH	INSERT CHARACTERS UNDER MASK (低) INSERT CHARACTERS UNDER MASK (高) INSERT IMMEDIATE (高 高) INSERT IMMEDIATE (高 低) INSERT IMMEDIATE (低 高)	RS C RSE C N RI N RI N RI N	A A			B ₂ B ₂	BF EB80 A50 A51 A52	7-79 7-79 7-80 7-80 7-80
IILL IPK IPM IPT ISKE	INSERT IMMEDIATE (低 低) INSERT PSW KEY INSERT PROGRAM MASK INVALIDATE PAGE TABLE ENTRY INSERT STORAGE KEY EXTENDED	RI N S RRE RRE RRE	Q P A ¹ P A ¹	G2 \$			A53 B20B B222 B221 B229	7-80 10-25 7-81 10-27 10-25
IVSK KDB KDBR KEB KEBR	INSERT VIRTUAL STORAGE KEY COMPARE AND SIGNAL (長精度 BFP) COMPARE AND SIGNAL (長精度 BFP) COMPARE AND SIGNAL (短精度 BFP) COMPARE AND SIGNAL (短精度 BFP)	RRE RXE C RRE C RXE C RRE C	Q A ¹ A	SO Db Xi Db Xi Db Xi Db Xi		R ₂ B ₂ B ₂	B223 ED18 B318 ED08 B308	10-26 19-25 19-25 19-25 19-25
KXBR L LA LAE LAM	COMPARE AND SIGNAL (拡張精度 BFP) LOAD (32) LOAD ADDRESS LOAD ADDRESS EXTENDED LOAD ACCESS MULTIPLE	RRE C RX RX RX RS	A A A A SP	SP Db Xi		B ₂ U ₁ BP UB	B348 58 41 51 9A	19-25 7-81 7-82 7-82 7-81
LARL LASP LCDR LCDBR LCEBR	LOAD ADDRESS RELATIVE LONG LOAD ADDRESS SPACE PARAMETERS LOAD COMPLEMENT (長精度 BFP) LOAD COMPLEMENT (長精度 HFP) LOAD COMPLEMENT (短精度 BFP)	RIL N ³ SSE C RRE C RR C RRE C	P A ¹ SP	SO Db Da Db		B ₁	C00 E500 B313 23 B303	7-83 10-28 19-37 18-13 19-37
LCER LCGFR LCGR LCR LCTL	LOAD COMPLEMENT (短精度 HFP) LOAD COMPLEMENT (64<32) LOAD COMPLEMENT (64) LOAD COMPLEMENT (32) LOAD CONTROL (32)	RR C RRE C N RRE C N RR C RS	P A SP	Da IF IF IF		B ₂	33 B913 B903 13 B7	18-13 7-84 7-84 7-84 10-37
LCTLG LCXBR LCXR LD LDE	LOAD CONTROL (64) LOAD COMPLEMENT (拡張精度 BFP) LOAD COMPLEMENT (拡張精度 HFP) LOAD (長精度) LOAD LENGTHENED (短精度から長精度 HFP)	RSE N RRE C RRE C RX RXE	P A SP A A	SP Db Da Da Da		B ₂ B ₂ B ₂ B ₂	EB2F B343 B363 68 ED24	10-37 19-37 18-13 9-11 18-14
LDEB LDEBR LDER LDR LDXBR	LOAD LENGTHENED (短精度から長精度 BFP) LOAD LENGTHENED (短精度から長精度 BFP) LOAD LENGTHENED (短精度から長精度 HFP) LOAD (長精度) LOAD ROUNDED (拡張精度から長精度 BFP)	RXE RRE RRE RR RRE	A A A SP	Db Xi Db Xi Da Da Da Xi Xo Xu Xx		B ₂	ED04 B304 B324 28 B345	19-40 19-40 18-14 9-11 19-41

図 B-2 (5/13). ニーモニツク順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ
LDXR	LOAD ROUNDED (拡張精度から長精度 HFP)	RR		SP	Da	E0		25	18-16
LE	LOAD (短精度)	RX		A	Da		B ₂	78	9-11
LEDBR	LOAD ROUNDED (長精度から短精度 BFP)	RRE			Db	Xi Xo Xu Xx		B344	19-41
LEDR	LOAD ROUNDED (長精度から短精度 HFP)	RR			Da	E0		35	18-16
LER	LOAD (短精度)	RR			Da			38	9-11
LXBR	LOAD ROUNDED (拡張精度から短精度 BFP)	RRE		SP	Db	Xi Xo Xu Xx		B346	19-41
LXBR	LOAD ROUNDED (拡張精度から短精度 HFP)	RRE		SP	Da	E0		B366	18-16
LFPC	LOAD FPC	S		A	SP	Db	B ₂	B29D	19-39
LG	LOAD (64)	RXE	N	A			B ₂	E304	7-81
LGF	LOAD (64<32)	RXE	N	A			B ₂	E314	7-81
LGFR	LOAD (64<32)	RRE	N					B914	7-81
LGH	LOAD HALFWORD (64)	RXE	N	A			B ₂	E315	7-84
LGHI	LOAD HALFWORD IMMEDIATE (64)	RI	N					A79	7-85
LGR	LOAD (64)	RRE	N					B904	7-81
LH	LOAD HALFWORD (32)	RX		A			B ₂	48	7-84
LHI	LOAD HALFWORD IMMEDIATE (32)	RI						A78	7-85
LLGC	LOAD LOGICAL CHARACTER	RXE	N	A			B ₂	E390	7-85
LLGF	LOAD LOGICAL (64<32)	RXE	N	A			B ₂	E316	7-85
LLGFR	LOAD LOGICAL (64<32)	RRE	N					B916	7-85
LLGH	LOAD LOGICAL HALFWORD	RXE	N	A			B ₂	E391	7-85
LLGT	LOAD LOGICAL THIRTY ONE BITS	RXE	N	A			B ₂	E317	7-86
LLGTR	LOAD LOGICAL THIRTY ONE BITS	RRE	N					B917	7-86
LLIHH	LOAD LOGICAL IMMEDIATE (高 高)	RI	N					A5C	7-86
LLIHL	LOAD LOGICAL IMMEDIATE (高 低)	RI	N					A5D	7-86
LLILH	LOAD LOGICAL IMMEDIATE (低 高)	RI	N					A5E	7-86
LLILL	LOAD LOGICAL IMMEDIATE (低 低)	RI	N					A5F	7-86
LM	LOAD MULTIPLE (32)	RS		A			B ₂	98	7-86
LMD	LOAD MULTIPLE DISJOINT	SS	N	A			B ₂ B ₄	EF	7-87
LMG	LOAD MULTIPLE (64)	RSE	N	A			B ₂	EB04	7-86
LMH	LOAD MULTIPLE HIGH	RSE	N	A			B ₂	EB96	7-87
LNDBR	LOAD NEGATIVE (長精度 BFP)	RRE	C		Db			B311	19-40
LNDR	LOAD NEGATIVE (長精度 HFP)	RR	C		Da			21	18-15
LNEBR	LOAD NEGATIVE (短精度 BFP)	RRE	C		Db			B301	19-40
LNER	LOAD NEGATIVE (短精度 HFP)	RR	C		Da			31	18-15
LNGFR	LOAD NEGATIVE (64<32)	RRE	C N					B911	7-88
LNGR	LOAD NEGATIVE (64)	RRE	C N					B901	7-88
LNR	LOAD NEGATIVE (32)	RR	C					11	7-88
LNDBR	LOAD NEGATIVE (拡張精度 BFP)	RRE	C	SP	Db			B341	19-40
LNDR	LOAD NEGATIVE (拡張精度 HFP)	RRE	C	SP	Da			B361	18-15
LPDBR	LOAD POSITIVE (長精度 BFP)	RRE	C		Db			B310	19-41
LPDR	LOAD POSITIVE (長精度 HFP)	RR	C		Da			20	18-15
LPEBR	LOAD POSITIVE (短精度 BFP)	RRE	C		Db			B300	19-41
LPER	LOAD POSITIVE (短精度 HFP)	RR	C		Da			30	18-15
LPGFR	LOAD POSITIVE (64<32)	RRE	C N			IF		B910	7-88
LPGR	LOAD POSITIVE (64)	RRE	C N			IF		B900	7-88

図 B-2 (6/13). ニーモニック順の命令一覧表

ニーモニク	名前	特性				命令コード	ページ
LPQ LPR LPSW LPSWE LPXBR	LOAD PAIR FROM QUADWORD LOAD POSITIVE (32) LOAD PSW LOAD PSW EXTENDED LOAD POSITIVE (拡張精度 BFP)	RXE RR S S RRE	N C L L C	A P P	SP SP SP SP SP	B ₂ E38F 10 B ₂ 82 B ₂ B2B2 B340	7-88 7-88 10-37 10-38 19-41
LPXR LR LRA LRAG LRDR	LOAD POSITIVE (拡張精度 HFP) LOAD (32) LOAD REAL ADDRESS (32) LOAD REAL ADDRESS (64) LOAD ROUNDED (拡張精度から長精度 HFP)	RRE RR RX RXE RR	C C N	 P P	SP SO A ¹ SP	Da Da E0	B360 18 7-81 BP B1 E303 25 18-15 10-39 10-39 18-16
LRER LRV LRVG LRVGR LRVH	LOAD ROUNDED (長精度から短精度 HFP) LOAD REVERSED (32) LOAD REVERSED (64) LOAD REVERSED (64) LOAD REVERSED (16)	RR RXE RXE RRE RXE	 N3 N N N3	A A A	Da E0	35 B ₂ E31E B ₂ E30F B90F B ₂ E31F	18-16 7-89 7-89 7-89 7-89
LRVR LTDBR LTDR LTEBR LTER	LOAD REVERSED (32) LOAD AND TEST (長精度 BFP) LOAD AND TEST (長精度 HFP) LOAD AND TEST (短精度 BFP) LOAD AND TEST (短精度 HFP)	RRE RRE RR RRE RR	N3 C C C C	 Db Da Db Da	Xi Xi	B91F B312 22 B302 32	7-89 19-37 18-12 19-37 18-12
LTGFR LTGR LTR LTXBR LTXR	LOAD AND TEST (64<32) LOAD AND TEST (64) LOAD AND TEST (32) LOAD AND TEST (拡張精度 BFP) LOAD AND TEST (拡張精度 HFP)	RRE RRE RR RRE RRE	C C C C C	N N SP SP	Da Xi Da	B912 B902 12 B342 B362	7-84 7-83 7-83 19-37 18-12
LURA LURAG LXD LXDB LXDBR	LOAD USING REAL ADDRESS (32) LOAD USING REAL ADDRESS (64) LOAD LENGTHENED (長精度から拡張精度 HFP) LOAD LENGTHENED (長精度から拡張精度 BFP) LOAD LENGTHENED (長精度から拡張精度 BFP)	RRE RRE RXE RXE RRE	 N A	P P A A SP SP	Da Da Xi Xi	B24B B905 E025 B ₂ ED05 B305	10-44 10-44 18-14 19-40 19-40
LXDR LXE LXEB LXEBR LXER	LOAD LENGTHENED (長精度から拡張精度 HFP) LOAD LENGTHENED (短精度から拡張精度 HFP) LOAD LENGTHENED (短精度から拡張精度 BFP) LOAD LENGTHENED (短精度から拡張精度 BFP) LOAD LENGTHENED (短精度から拡張精度 HFP)	RRE RXE RXE RRE RRE	 A A SP SP	Da Da Xi Xi Da	Da Xi Xi	B325 B ₂ ED26 B ₂ ED06 B306 B326	18-14 18-14 19-40 19-40 18-14
LXR LZDR LZER LZXR M	LOAD (拡張精度) LOAD ZERO (長精度) LOAD ZERO (短精度) LOAD ZERO (拡張精度) MULTIPLY (64<32)	RRE RRE RRE RRE RX	 A	SP SP SP SP	Da Da Da Da	B365 B375 B374 B376 B ₂ 5C	9-11 9-11 9-11 9-11 7-105
MADB MADBR MAEB MAEBR MC	MULTIPLY AND ADD (長精度 BFP) MULTIPLY AND ADD (長精度 BFP) MULTIPLY AND ADD (短精度 BFP) MULTIPLY AND ADD (短精度 BFP) MONITOR CALL	RXF RRF RXF RRF SI	 A A SP	Da Xi Xi Xi MO	Xo Xu Xx Xx Xx	B ₂ ED1E B31E B ₂ ED0E B30E AF	19-44 19-44 19-44 19-44 7-90

図 B-2 (7/13). ニーモニク順の命令一覧表

ニーモニック	名前	特性				命令コード	ページ
MD	MULTIPLY (長精度 HFP)	RX		A	Da EU EO	B ₂	6C 18-16
MDB	MULTIPLY (長精度 BFP)	RXE		A	Db Xi Xo Xu Xx	B ₂	ED1C 19-42
MDBR	MULTIPLY (長精度 BFP)	RRE			Db Xi Xo Xu Xx		B31C 19-42
MDE	MULTIPLY (短精度から長精度 HFP)	RX		A	Da EU EO	B ₂	7C 18-16
MDEB	MULTIPLY (短精度から長精度 BFP)	RXE		A	Db Xi	B ₂	ED0C 19-42
MDEBR	MULTIPLY (短精度から長精度 BFP)	RRE			Db Xi		B30C 19-42
MDER	MULTIPLY (短精度から長精度 HFP)	RR			Da EU EO		3C 18-16
MDR	MULTIPLY (長精度 HFP)	RR			Da EU EO		2C 18-16
ME	MULTIPLY (短精度から長精度 HFP)	RX		A	Da EU EO	B ₂	7C 18-16
MEE	MULTIPLY (短精度 HFP)	RXE		A	Da EU EO	B ₂	ED37 18-16
MEEB	MULTIPLY (短精度 BFP)	RXE		A	Db Xi Xo Xu Xx	B ₂	ED17 19-42
MEEBR	MULTIPLY (短精度 BFP)	RRE			Db Xi Xo Xu Xx		B317 19-42
MEER	MULTIPLY (短精度 HFP)	RRE			Da EU EO		B337 18-16
MER	MULTIPLY (短精度から長精度 HFP)	RR			Da EU EO		3C 18-16
MGHI	MULTIPLY HALWORD IMMEDIATE (64)	RI	N				A7D 7-106
MH	MULTIPLY HALWORD (32)	RX		A		B ₂	4C 7-106
MHI	MULTIPLY HALWORD IMMEDIATE (32)	RI					A7C 7-106
ML	MULTIPLY LOGICAL (64<32)	RXE	N3	A SP		B ₂	E396 7-106
MLG	MULTIPLY LOGICAL (128<64)	RXE	N	A SP		B ₂	E386 7-106
MLGR	MULTIPLY LOGICAL (128<64)	RRE	N	SP			B986 7-106
MLR	MULTIPLY LOGICAL (64<32)	RRE	N3	A SP			B996 7-106
MP	MULTIPLY DECIMAL	SS		A SP	Dd	ST B ₁ B ₂	FC 8-11
MR	MULTIPLY (64<32)	RR		SP			1C 7-105
MS	MULTIPLY SINGLE (32)	RX		A		B ₂	71 7-107
MSCH	MODIFY SUBCHANNEL	S	C	P A SP	OP ¢ GS	B ₂	B232 14-7
MSDB	MULTIPLY AND SUBTRACT (長精度 BFP)	RXF		A	Db Xi Xo Xu Xx	B ₂	ED1F 19-44
MSDBR	MULTIPLY AND SUBTRACT (長精度 BFP)	RRF			Db Xi Xo Xu Xx		B31F 19-44
MSEB	MULTIPLY AND SUBTRACT (短精度 BFP)	RXF		A	Db Xi Xo Xu Xx	B ₂	ED0F 19-44
MSEBR	MULTIPLY AND SUBTRACT (短精度 BFP)	RRF			Db Xi Xo Xu Xx		B30F 19-44
MSG	MULTIPLY SINGLE (64)	RXE	N	A		B ₂	E30C 7-107
MSGF	MULTIPLY SINGLE (64<32)	RXE	N	A		B ₂	E31C 7-107
MSGFR	MULTIPLY SINGLE (64<32)	RRE	N				B91C 7-107
MSGR	MULTIPLY SINGLE (64)	RRE	N				B90C 7-107
MSR	MULTIPLY SINGLE (32)	RRE					B252 7-107
MSTA	MODIFY STACKED STATE	RRE		A ¹ SP	SE	ST	B247 10-44
MVC	MOVE (文字)	SS		A		ST B ₁ B ₂	D2 7-91
MVCBK	MOVE WITH DESTINATION KEY	SSE		Q A	GM	ST B ₁ B ₂	E50F 10-50
MVCIN	MOVE INVERSE	SS		A		ST B ₁ B ₂	E8 7-91
MVCK	MOVE WITH KEY	SS	C	Q A		ST B ₁ B ₂	D9 10-50
MVCL	MOVE LONG	RR	C	A SP	II	ST R ₁ R ₂	0E 7-92
MVCLC	MOVE LONG EXTENDED	RS	C	A SP		ST R ₁ R ₂ R ₃	A8 7-95
MVCLU	MOVE LONG UNICODE	RSE	C	A SP		ST R ₁ R ₂	EB8E 7-99
MVCP	MOVE TO PRIMARY	SS	C	Q A	SO ¢	ST	DA 10-48
MVCS	MOVE TO SECONDARY	SS	C	Q A	SO ¢	ST	DB 10-48
MVCSK	MOVE WITH SOURCE KEY	SSE		Q A	GM	ST B ₁ B ₂	E50E 10-52

図 B-2 (8/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ	
MVI	MOVE (即値)	SI	A				ST	B ₁	92	7-91
MVN	MOVE NUMERICS	SS	A				ST	B ₁ B ₂	D1	7-102
MVO	MOVE WITH OFFSET	SS	A				ST	B ₁ B ₂	F1	7-104
MVPG	MOVE PAGE	RRE C	Q A	SP	G0		ST	R ₁ R ₂	B254	10-46
MVST	MOVE STRING	RRE C	A	SP	G0		ST	R ₁ R ₂	B255	7-103
MVZ	MOVE ZONES	SS	A				ST	B ₁ B ₂	D3	7-104
MXBR	MULTIPLY (拡張精度 BFP)	RRE	A	SP	Db Xi Xo Xu Xx			B ₂	B34C	19-42
MXD	MULTIPLY (長精度から拡張精度 HFP)	RX	A	SP	Da EU EO			B ₂	67	18-16
MXDB	MULTIPLY (長精度から拡張精度 BFP)	RXE	A	SP	Db Xi			B ₂	E007	19-42
MXDBR	MULTIPLY (長精度から拡張精度 BFP)	RRE	A	SP	Db Xi				B307	19-42
MXDR	MULTIPLY (長精度から拡張精度 HFP)	RR		SP	Da EU EO				27	18-16
MXR	MULTIPLY (拡張精度 HFP)	RR		SP	Da EU EO				26	18-16
N	AND (32)	RX C	A					B ₂	54	7-18
NC	AND (文字)	SS C	A				ST	B ₁ B ₂	D4	7-18
NG	AND (64)	RXE C N	A					B ₂	E380	7-18
NGR	AND (64)	RRE C N							B980	7-18
NI	AND (即値)	SI C	A				ST	B ₁	94	7-18
NIHH	AND IMMEDIATE (高 高)	RI C N							A54	7-19
NIHL	AND IMMEDIATE (高 低)	RI C N							A55	7-19
NILH	AND IMMEDIATE (低 高)	RI C N							A56	7-19
NILL	AND IMMEDIATE (低 低)	RI C N							A57	7-19
NR	AND (32)	RR C							14	7-18
O	OR (32)	RX C	A					B ₂	56	7-108
OC	OR (文字)	SS C	A				ST	B ₁ B ₂	D6	7-108
OG	OR (64)	RXE C N	A					B ₂	E381	7-108
OGR	OR (64)	RRE C N							B981	7-108
OI	OR (即値)	SI C	A				ST	B ₁	96	7-108
OIHH	OR IMMEDIATE (高 高)	RI C N							A58	7-109
OIHL	OR IMMEDIATE (高 低)	RI C N							A59	7-109
OILH	OR IMMEDIATE (低 高)	RI C N							A5A	7-109
OILL	OR IMMEDIATE (低 低)	RI C N							A5B	7-109
OR	OR (32)	RR C							16	7-108
PACK	PACK	SS	A				ST	B ₁ B ₂	F2	7-109
PALB	PURGE ALB	RRE	P		\$				B248	10-78
PC	PROGRAM CALL	S	Q A ¹		Z ¹ T ¢ GM		B	ST	B218	10-55
PGIN	PAGE IN	RRE C ES	P A ¹		¢				B22E	10-53
PGOUT	PAGE OUT	RRE C ES	P A ¹		¢				B22F	10-54
PKA	PACK ASCII	SS E2	A SP				ST	B ₁ B ₂	E9	7-110
PKU	PACK UNICODE	SS E2	A SP				ST	B ₁ B ₂	E1	7-111
PLO	PERFORM LOCKED OPERATION	SS C	A SP		\$ GM		ST	FC	EE	7-112
PR	PROGRAM RETURN	E L	Q A ¹ SP		Z ⁴ T ¢ ²		B	ST	0101	10-68
PT	PROGRAM TRANSFER	RRE	Q A ¹ SP		Z ² T ¢		B		B228	10-72
PTLB	PURGE TLB	S	P		\$				B20D	10-78
RCHP	RESET CHANNEL PATH	S C	P		OP ¢ G1				B23B	14-8
RLL	ROTATE LEFT SINGLE LOGICAL (32)	RSE	N3						EB1D	7-127

図 B-2 (9/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性					命令コード	ページ
RLLG RP RRBE RSCH S	ROTATE LEFT SINGLE LOGICAL (64) RESUME PROGRAM RESET REFERENCE BIT EXTENDED RESUME SUBCHANNEL SUBTRACT (32)	RSE S L RRE C S C RX C	N	Q A SP P A ¹ P A	WE T OP ¢ GS IF	B	B ₂ B ₂	EB1C 7-127 B277 10-79 B22A 10-78 B238 14-10 5B 7-141
SAC SACF SAL SAM24 SAM31	SET ADDRESS SPACE CONTROL SET ADDRESS SPACE CONTROL FAST SET ADDRESS LIMIT SET ADDRESSING MODE (24) SET ADDRESSING MODE (31)	S S S E E	N3 N3	Q SP Q SP P SP SP	SW ¢ SW OP ¢ G1 T T		B ₂ B ₂ B ₂ 010C 010D	B219 10-81 B279 10-81 B237 14-11 7-129 7-129
SAM64 SAR SCHM SCK SCKC	SET ADDRESSING MODE (64) SET ACCESS SET CHANNEL MONITOR SET CLOCK SET CLOCK COMPARATOR	E RRE S S C S	N	P A SP P A SP	T OP ¢ GM		U ₁ B ₂ B ₂	010E 7-129 B24E 7-129 B23C 14-12 B204 10-83 B206 10-84
SCKPF SD SDB SDBR SDR	SET CLOCK PROGRAMMABLE FIELD SUBTRACT NORMALIZED (長精度 HFP) SUBTRACT (長精度 BFP) SUBTRACT (長精度 BFP) SUBTRACT NORMALIZED (長精度 HFP)	E RX C RXE C RRE C RR C		P SP A A	G0 Da EU EO LS Db Xi Xo Xu Xx Db Xi Xo Xu Xx Da EU EO LS		B ₂ B ₂	0107 10-84 6B 18-19 ED1B 19-48 B31B 19-48 2B 18-19
SE SEB SEBR SER SFPC	SUBTRACT NORMALIZED (短精度 HFP) SUBTRACT (短精度 BFP) SUBTRACT (短精度 BFP) SUBTRACT NORMALIZED (短精度 HFP) SET FPC	RX C RXE C RRE C RR C RRE		A A SP	Da EU EO LS Db Xi Xo Xu Xx Db Xi Xo Xu Xx Da EU EO LS Db		B ₂ B ₂	7B 18-19 ED0B 19-48 B30B 19-48 3B 18-19 B384 19-46
SG SGF SGFR SGR SH	SUBTRACT (64) SUBTRACT (64<32) SUBTRACT (64<32) SUBTRACT (64) SUBTRACT HALFWORD	RXE C N RXE C N RRE C N RRE C N RX C	N	A A	IF IF IF IF IF		B ₂ B ₂ B ₂	E309 7-141 E319 7-141 B919 7-141 B909 7-141 4B 7-142
SIGP SL SLA SLAG SLB	SIGNAL PROCESSOR SUBTRACT LOGICAL (32) SHIFT LEFT SINGLE (32) SHIFT LEFT SINGLE (64) SUBTRACT LOGICAL WITH BORROW (32)	RS C RX C RS C RSE C N RXE C N3		P A A	\$ IF IF		B ₂ B ₂	AE 10-89 5F 7-142 8B 7-132 EB0B 7-132 E399 7-143
SLBG SLBGR SLBR SLDA SLDL	SUBTRACT LOGICAL WITH BORROW (64) SUBTRACT LOGICAL WITH BORROW (64) SUBTRACT LOGICAL WITH BORROW (32) SHIFT LEFT DOUBLE SHIFT LEFT DOUBLE LOGICAL	RXE C N RRE C N RRE C N3 RS C RS	N	A SP SP	IF		B ₂	E389 7-143 B989 7-143 B999 7-143 8F 7-130 8D 7-131
SLG SLGF SLGFR SLGR SLL	SUBTRACT LOGICAL (64) SUBTRACT LOGICAL (64<32) SUBTRACT LOGICAL (64<32) SUBTRACT LOGICAL (64) SHIFT LEFT SINGLE LOGICAL (32)	RXE C N RXE C N RRE C N RRE C N RS		A A			B ₂ B ₂	E30B 7-142 E31B 7-142 B91B 7-142 B90B 7-142 89 7-132

図 B-2 (10/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性				命令コード	ページ
SLLG SLR SP SPKA SPM	SHIFT LEFT SINGLE LOGICAL (64) SUBTRACT LOGICAL (32) SUBTRACT DECIMAL SET PSW KEY FROM ADDRESS SET PROGRAM MASK	RSE RR SS S RR	C C C L	N A Q	Dd DF	ST B ₁ B ₂	EB0D 7-133 1F 7-142 FB 8-12 B20A 10-85 04 7-130
SPT SPX SQD SQDB SQDBR	SET CPU TIMER SET PREFIX SQUARE ROOT (長精度 HFP) SQUARE ROOT (長精度 BFP) SQUARE ROOT (長精度 BFP)	S S RXE RXE RRE	C C C C C	P A SP P A SP A A	\$ Da SQ Db Xi Xx Db Xi Xx	B ₂ B ₂ B ₂ B ₂	B208 10-84 B210 10-85 ED35 18-18 ED15 19-47 B315 19-47
SQDR SQE SQEB SQEBR SQER	SQUARE ROOT (長精度 HFP) SQUARE ROOT (短精度 HFP) SQUARE ROOT (短精度 BFP) SQUARE ROOT (短精度 BFP) SQUARE ROOT (短精度 HFP)	RRE RXE RXE RRE RRE	C C C C C	A A A	Da SQ Da SQ Db Xi Xx Db Xi Xx Da SQ	B ₂ B ₂	B244 18-18 ED34 18-18 ED14 19-47 B314 19-47 B245 18-18
SQXBR SQXR SR SRA SRAG	SQUARE ROOT (拡張精度 BFP) SQUARE ROOT (拡張精度 HFP) SUBTRACT (32) SHIFT RIGHT SINGLE (32) SHIFT RIGHT SINGLE (64)	RRE RRE RR RS RSE	C C C C C N	SP SP C C N	Db Xi SQ Xx Da SQ IF	B316 19-47 B336 18-18 1B 7-141 8A 7-134 EB0A 7-134	
SRDA SRDL SRL SRLG SRNM	SHIFT RIGHT DOUBLE SHIFT RIGHT DOUBLE LOGICAL SHIFT RIGHT SINGLE LOGICAL (32) SHIFT RIGHT SINGLE LOGICAL (64) SET ROUNDING MODE	RS RS RS RSE S	C C C C C	SP SP N N	Db	8E 7-133 8C 7-133 88 7-135 EB0C 7-135 B299 19-47	
SRP SRST SSAR SSCH SSKE	SHIFT AND ROUND DECIMAL SEARCH STRING SET SECONDARY ASN START SUBCHANNEL SET STORAGE KEY EXTENDED	SS RRE RRE S RRE	C C C C C	A A SP A ¹ P A SP P A ¹	Dd DF G0 Z ³ T ¢ OP ¢ GS ¢	ST B ₁ R ₂ B ₂	F0 8-11 B25E 7-128 B225 10-86 B233 14-14 B22B 10-89
SSM ST STAM STAP STC	SET SYSTEM MASK STORE (32) STORE ACCESS MULTIPLE STORE CPU ADDRESS STORE CHARACTER	S RX RS S RX	C C C C C	P A SP A A SP P A SP A	SO	ST ST ST ST ST	B ₂ 80 10-89 B ₂ 50 7-135 UB 9B 7-135 B ₂ B212 10-91 B ₂ 42 7-136
STCK STCKC STCKE STCM STCMH	STORE CLOCK STORE CLOCK COMPARATOR STORE CLOCK EXTENDED STORE CHARACTERS UNDER MASK (低) STORE CHARACTERS UNDER MASK (高)	S S S RS RSE	C C C C N	A P A SP A A A	\$ \$	ST ST ST ST ST	B ₂ B205 7-137 B ₂ B207 10-91 B ₂ B278 7-137 B ₂ BE 7-136 B ₂ EB2C 7-136
STCPS STCRW STCTG STCTL STD	STORE CHANNEL PATH STATUS STORE CHANNEL REPORT WORD STORE CONTROL (64) STORE CONTROL (32) STORE (長精度)	S S RSE RS RX	C C C C C	P A SP P A SP P A SP P A SP A	¢ ¢ Da	ST ST ST ST ST	B ₂ B23A 14-15 B ₂ B239 14-16 B ₂ EB25 10-91 B ₂ B6 10-91 B ₂ 60 9-11

図 B-2 (11/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ			
STE	STORE (短精度)	RX		A			ST	B ₂	70	9-11		
STFL	STORE FACILITY LIST	S	N3	P					B2B1	10-93		
STFPC	STORE FPC	S		A			ST	B ₂	B29C	19-47		
STG	STORE (64)	RXE	N	A			ST	B ₂	E324	7-135		
STH	STORE HALFWORD	RX		A			ST	B ₂	40	7-139		
STIDP	STORE CPU ID	S		P	A	SP						
STM	STORE MULTIPLE (32)	RS		A			ST	B ₂	B202	10-92		
STMG	STORE MULTIPLE (64)	RSE	N	A			ST	B ₂	90	7-139		
STMH	STORE MULTIPLE HIGH	RSE	N	A			ST	B ₂	EB24	7-139		
STNSM	STORE THEN AND SYSTEM MASK	SI		P	A		ST	B ₁	EB26	7-140		
									AC	10-105		
STOSM	STORE THEN OR SYSTEM MASK	SI		P	A	SP						
STPQ	STORE PAIR TO QUADWORD	RXE	N	A	A	SP	ST	B ₁	AD	10-105		
STPT	STORE CPU TIMER	S		P	A	SP	ST	B ₂	E38E	7-140		
STPX	STORE PREFIX	S		P	A	SP	ST	B ₂	B209	10-93		
STRAG	STORE REAL ADDRESS	SSE	N	P	A	A ¹	ST	B ₁	B211	10-93		
									BP	10-94		
STRV	STORE REVERSED (32)	RXE	N3	A			ST	B ₂	E33E	7-140		
STRVG	STORE REVERSED (64)	RXE	N	A			ST	B ₂	E32F	7-140		
STRVH	STORE REVERSED (16)	RXE	N3	A			ST	B ₂	E33F	7-140		
STSCH	STORE SUBCHANNEL	S	C	P	A	SP	OP	¢	GS	B234	14-17	
STSI	STORE SYSTEM INFORMATION	S	C	P	A	SP			GM	B27D	10-95	
STURA	STORE USING REAL ADDRESS (32)	RRE		P	A	A ¹ SP						
STURG	STORE USING REAL ADDRESS (64)	RRE	N	P	A	A ¹ SP		SU				
SU	SUBTRACT UNNORMALIZED (短精度 HFP)	RX	C	A			Da	EO	LS	B246	10-105	
SUR	SUBTRACT UNNORMALIZED (短精度 HFP)	RR	C				Da	EO	LS	B925	10-105	
SVC	SUPERVISOR CALL	RR							¢	7F	18-20	
										3F	18-20	
										0A	7-144	
SW	SUBTRACT UNNORMALIZED (長精度 HFP)	RX	C	A			Da	EO	LS	B246	18-20	
SWR	SUBTRACT UNNORMALIZED (長精度 HFP)	RR	C				Da	EO	LS	2F	18-20	
SXBR	SUBTRACT (拡張精度 BFP)	RRE	C		SP		Db	Xi	Xo Xu Xx	B34B	19-48	
SXR	SUBTRACT NORMALIZED (拡張精度 HFP)	RR	C		SP		Da	EU	EO	37	18-19	
TAM	TEST ADDRESSING MODE	E	C	N3						010B	7-144	
TAR	TEST ACCESS	RRE	C	A ¹								
TB	TEST BLOCK	RRE	C	P	A ¹		II	\$	G0	U ₁	B24C	10-106
TBDR	CONVERT HFP TO BFP (長精度)	RRE	C		A ¹		Da			B22C	10-108	
TBEDR	CONVERT HFP TO BFP (長精度から短精度)	RRF	C		SP		Da			B351	9-9	
		RRF	C		SP		Da			B350	9-9	
TCDB	TEST DATA CLASS (長精度 BFP)	RXE	C				Db			ED11	19-48	
TCEB	TEST DATA CLASS (短精度 BFP)	RXE	C				Db			ED10	19-48	
TCXB	TEST DATA CLASS (拡張精度 BFP)	RXE	C		SP		Db			ED12	19-48	
THDER	CONVERT BFP TO HFP (短精度から長精度)	RRE	C				Da			B358	9-8	
THDR	CONVERT BFP TO HFP (長精度)	RRE	C				Da			B359	9-8	
TM	TEST UNDER MASK	SI	C	A						B ₁	91	7-145
TMH	TEST UNDER MASK HIGH	RI	C							A70	7-145	
TMHH	TEST UNDER MASK (高高)	RI	C	N						A72	7-145	
TMHL	TEST UNDER MASK (高低)	RI	C	N						A73	7-145	
TML	TEST UNDER MASK LOW	RI	C							A71	7-145	
TMLH	TEST UNDER MASK (低高)	RI	C	N						A70	7-145	

図 B-2 (12/13). ニーモニック順の命令一覧表

ニーモニック	名前	特性						命令コード	ページ	
TMLL TP TPI TPROT TR	TEST UNDER MASK (低 低) TEST DECIMAL TEST PENDING INTERRUPTION TEST PROTECTION TRANSLATE	RI C N RSL C E2 S C SSE C SS		A P A ¹ SP P A ¹ A		¢	ST ST	B ₁ B ₂ B ₁ B ₂	A71 EBC0 B236 E501 DC	7-145 8-12 14-17 10-110 7-146
TRACE TRACG TRAP2 TRAP4 TRE	TRACE (32) TRACE (64) TRAP TRAP TRANSLATE EXTENDED	RS RSE N E S RRE C		P A SP P A SP A A SP	T ¢ T ¢ SO T SO T		B B ST ST	B ₂ B ₂	99 EB0F 01FF B2FF B2A5	10-113 10-113 10-114 10-114 7-148
TROO TROT TRT TRTO TRTT	TRANSLATE ONE TO ONE TRANSLATE ONE TO TWO TRANSLATE AND TEST TRANSLATE TWO TO ONE TRANSLATE TWO TO TWO	RRE C E2 RRE C E2 SS C RRE C E2 RRE C E2		A SP A SP A A SP A SP		GM GM GM GM GM	ST ST ST ST ST	RM R ₂ RM R ₂ B ₁ B ₂ RM R ₂ RM R ₂	B993 B992 DD B991 B990	7-150 7-150 7-147 7-150 7-150
TS TSCH UNPK UNPKA UNPKU	TEST AND SET TEST SUBCHANNEL UNPACK UNPACK ASCII UNPACK UNICODE	S C S C SS SS C E2 SS C E2		A SP P A SP A A SP A SP		\$ ¢ GS	ST ST ST ST ST	B ₂ B ₂ B ₁ B ₂ B ₁ B ₂ B ₁ B ₂	93 B235 F3 EA E2	7-144 14-19 7-155 7-155 7-156
UPT X XC XG XGR	UPDATE TREE EXCLUSIVE OR (32) EXCLUSIVE OR (文字) EXCLUSIVE OR (64) EXCLUSIVE OR (64)	E C RX C SS C RXE C N RRE C N		A SP A A A A	II	GM	ST ST	I4 B ₂ B ₁ B ₂ B ₂	0102 57 D7 E382 B982	7-157 7-77 7-77 7-77 7-77
XI XR XSCH ZAP	EXCLUSIVE OR (即値) EXCLUSIVE OR (32) CANCEL SUBCHANNEL ZERO AND ADD	SI C RR C S C SS C		A A P A		OP ¢ GS Dd DF	ST ST	B ₁ B ₁ B ₂	97 17 B276 F8	7-77 7-77 14-4 8-13

図 B-2 (13/13). ニーモニック順の命令一覧表

命令コード	名前	二 モ ニ ツ ク	特性							ページ
0101 0102 0107 010B 010C	PROGRAM RETURN UPDATE TREE SET CLOCK PROGRAMMABLE FIELD TEST ADDRESSING MODE SET ADDRESSING MODE (24)	PR UPT SCKPF TAM SAM24	E E E E	L C C N3	Q A P N3	A ¹ A SP SP	Z ⁴ T II	φ ² GM G0	B ST I4	10-68 7-157 10-84 7-144 7-129
010D 010E 01FF 04 05	SET ADDRESSING MODE (31) SET ADDRESSING MODE (64) TRAP SET PROGRAM MASK BRANCH AND LINK	SAM31 SAM64 TRAP2 SPM BALR	E E E RR RR	N3 N L		SP A	T T SO T		B ST B	7-129 7-129 10-114 7-130 7-20
06 07 0A 0B 0C	BRANCH ON COUNT (32) BRANCH ON CONDITION SUPERVISOR CALL BRANCH AND SET MODE BRANCH AND SAVE AND SET MODE	BCTR BCR SVC BSM BASSM	RR RR RR RR RR				φ ¹ φ T T	B B B B		7-24 7-23 7-144 7-22 7-21
0D 0E 0F 10 11	BRANCH AND SAVE MOVE LONG COMPARE LOGICAL LONG LOAD POSITIVE (32) LOAD NEGATIVE (32)	BASR MVCL CLCL LPR LNR	RR RR RR RR RR	C C C C C		A A SP SP	T II II IF	B ST R ₁ R ₂ R ₁ R ₂		7-20 7-92 7-44 7-88 7-88
12 13 14 15 16	LOAD AND TEST (32) LOAD COMPLEMENT (32) AND (32) COMPARE LOGICAL (32) OR (32)	LTR LCR NR CLR OR	RR RR RR RR RR	C C C C C			IF			7-83 7-84 7-18 7-42 7-108
17 18 19 1A 1B	EXCLUSIVE OR (32) LOAD (32) COMPARE (32) ADD (32) SUBTRACT (32)	XR LR CR AR SR	RR RR RR RR RR	C C C C C			IF IF			7-77 7-81 7-33 7-16 7-141
1C 1D 1E 1F 20	MULTIPLY (64<32) DIVIDE (32<64) ADD LOGICAL (32) SUBTRACT LOGICAL (32) LOAD POSITIVE (長精度 HFP)	MR DR ALR SLR LPDR	RR RR RR RR RR	C C C C C		SP SP	IK Da			7-105 7-75 7-17 7-142 18-15
21 22 23 24 25	LOAD NEGATIVE (長精度 HFP) LOAD AND TEST (長精度 HFP) LOAD COMPLEMENT (長精度 HFP) HALVE (長精度 HFP) LOAD ROUNDED (拡張精度から長精度 HFP)	LNDR LTDR LCDR HDR LDXR	RR RR RR RR RR	C C C C C			Da Da Da Da EU Da E0			18-15 18-12 18-13 18-12 18-16
25 26 27 28 29	LOAD ROUNDED (拡張精度から長精度 HFP) MULTIPLY (拡張精度 HFP) MULTIPLY (長精度から拡張精度 HFP) LOAD (長精度) COMPARE (長精度 HFP)	LRDR MXR MXDR LDR CDR	RR RR RR RR RR	C C C C C		SP SP SP	Da E0 Da EU E0 Da EU E0 Da Da			18-16 18-16 18-16 9-11 18-9

図 B-3 (1/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニツク	特性						ページ	
2A	ADD NORMALIZED (長精度 HFP)	ADR	RR	C					18-7	
2B	SUBTRACT NORMALIZED (長精度 HFP)	SDR	RR	C					18-19	
2C	MULTIPLY (長精度 HFP)	MDR	RR						18-16	
2D	DIVIDE (長精度 HFP)	DDR	RR				FK		18-11	
2E	ADD UNNORMALIZED (長精度 HFP)	AWR	RR	C					18-8	
2F	SUBTRACT UNNORMALIZED (長精度 HFP)	SWR	RR	C					18-20	
30	LOAD POSITIVE (短精度 HFP)	LPER	RR	C					18-15	
31	LOAD NEGATIVE (短精度 HFP)	LNER	RR	C					18-15	
32	LOAD AND TEST (短精度 HFP)	LTER	RR	C					18-12	
33	LOAD COMPLEMENT (短精度 HFP)	LCER	RR	C					18-13	
34	HALVE (短精度 HFP)	HER	RR						18-12	
35	LOAD ROUNDED (長精度から短精度 HFP)	LEDR	RR						18-16	
35	LOAD ROUNDED (長精度から短精度 HFP)	LRER	RR						18-16	
36	ADD NORMALIZED (拡張精度 HFP)	AXR	RR	C		SP			18-7	
37	SUBTRACT NORMALIZED (拡張精度 HFP)	SXR	RR	C		SP			18-19	
38	LOAD (短精度)	LER	RR						9-11	
39	COMPARE (短精度 HFP)	CER	RR	C					18-9	
3A	ADD NORMALIZED (短精度 HFP)	AER	RR	C					18-7	
3B	SUBTRACT NORMALIZED (短精度 HFP)	SER	RR	C					18-19	
3C	MULTIPLY (短精度から長精度 HFP)	MDER	RR						18-16	
3C	MULTIPLY (短精度から長精度 HFP)	MER	RR						18-16	
3D	DIVIDE (短精度 HFP)	DER	RR						18-11	
3E	ADD UNNORMALIZED (短精度 HFP)	AUR	RR	C					18-8	
3F	SUBTRACT UNNORMALIZED (短精度 HFP)	SUR	RR	C					18-20	
40	STORE HALFWORD	STH	RX		A			ST	B ₂	7-139
41	LOAD ADDRESS	LA	RX							7-82
42	STORE CHARACTER	STC	RX		A			ST	B ₂	7-136
43	INSERT CHARACTER	IC	RX		A				B ₂	7-79
44	EXECUTE	EX	RX		AI	SP				7-78
45	BRANCH AND LINK	BAL	RX				EX	B		7-20
46	BRANCH ON COUNT (32)	BCT	RX					B		7-24
47	BRANCH ON CONDITION	BC	RX					B		7-23
48	LOAD HALFWORD (32)	LH	RX		A				B ₂	7-84
49	COMPARE HALFWORD	CH	RX	C	A				B ₂	7-42
4A	ADD HALFWORD	AH	RX	C	A		IF		B ₂	7-16
4B	SUBTRACT HALFWORD	SH	RX	C	A		IF		B ₂	7-142
4C	MULTIPLY HALFWORD (32)	MH	RX		A				B ₂	7-106
4D	BRANCH AND SAVE	BAS	RX					B		7-21
4E	CONVERT TO DECIMAL (32)	CVD	RX		A			ST	B ₂	7-68
4F	CONVERT TO BINARY (32)	CVB	RX		A		Dd IK		B ₂	7-68
50	STORE (32)	ST	RX		A			ST	B ₂	7-135
51	LOAD ADDRESS EXTENDED	LAE	RX						U ₁ BP	7-82
54	AND (32)	N	RX	C	A				B ₂	7-18
55	COMPARE LOGICAL (32)	CL	RX	C	A				B ₂	7-43
56	OR (32)	O	RX	C	A				B ₂	7-108

図 B-3 (2/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニツク	特性					ページ			
57	EXCLUSIVE OR (32)	X	RX	C	A			B ₂	7-77		
58	LOAD (32)	L	RX		A			B ₂	7-81		
59	COMPARE (32)	C	RX	C	A			B ₂	7-33		
5A	ADD (32)	A	RX	C	A	IF		B ₂	7-16		
5B	SUBTRACT (32)	S	RX	C	A	IF		B ₂	7-141		
5C	MULTIPLY (64<32)	M	RX		A	SP		B ₂	7-105		
5D	DIVIDE (32<64)	D	RX		A	SP	IK	B ₂	7-75		
5E	ADD LOGICAL (32)	AL	RX	C	A			B ₂	7-17		
5F	SUBTRACT LOGICAL (32)	SL	RX	C	A			B ₂	7-142		
60	STORE (長精度)	STD	RX		A		Da	ST	B ₂	9-11	
67	MULTIPLY (長精度から拡張精度 HFP)	MXD	RX		A	SP	Da EU EO		B ₂	18-16	
68	LOAD (長精度)	LD	RX		A		Da		B ₂	9-11	
69	COMPARE (長精度 HFP)	CD	RX	C	A		Da		B ₂	18-9	
6A	ADD NORMALIZED (長精度 HFP)	AD	RX	C	A		Da EU EO LS		B ₂	18-7	
6B	SUBTRACT NORMALIZED (長精度 HFP)	SD	RX	C	A		Da EU EO LS		B ₂	18-19	
6C	MULTIPLY (長精度 HFP)	MD	RX		A		Da EU EO		B ₂	18-16	
6D	DIVIDE (長精度 HFP)	DD	RX		A		Da EU EO FK		B ₂	18-11	
6E	ADD UNNORMALIZED (長精度 HFP)	AW	RX	C	A		Da EO LS		B ₂	18-8	
6F	SUBTRACT UNNORMALIZED (長精度 HFP)	SW	RX	C	A		Da EO LS		B ₂	18-20	
70	STORE (短精度)	STE	RX		A		Da	ST	B ₂	9-11	
71	MULTIPLY SINGLE (32)	MS	RX		A				B ₂	7-107	
78	LOAD (短精度)	LE	RX		A		Da		B ₂	9-11	
79	COMPARE (短精度 HFP)	CE	RX	C	A		Da		B ₂	18-9	
7A	ADD NORMALIZED (短精度 HFP)	AE	RX	C	A		Da EU EO LS		B ₂	18-7	
7B	SUBTRACT NORMALIZED (短精度 HFP)	SE	RX	C	A		Da EU EO LS		B ₂	18-19	
7C	MULTIPLY (短精度から長精度 HFP)	MDE	RX		A		Da EU EO		B ₂	18-16	
7C	MULTIPLY (短精度から長精度 HFP)	ME	RX		A		Da EU EO		B ₂	18-16	
7D	DIVIDE (短精度 HFP)	DE	RX		A		Da EU EO FK		B ₂	18-11	
7E	ADD UNNORMALIZED (短精度 HFP)	AU	RX	C	A		Da EO LS		B ₂	18-8	
7F	SUBTRACT UNNORMALIZED (短精度 HFP)	SU	RX	C	A		Da EO LS		B ₂	18-20	
80	SET SYSTEM MASK	SSM	S		P	A	SP	S0		B ₂	10-89
82	LOAD PSW	LPSW	S	L	P	A	SP	¢		B ₂	10-37
83	DIAGNOSE			DM	P	DM				MD	10-18
84	BRANCH RELATIVE ON INDEX HIGH (32)	BRXH	RSI						B		7-28
85	BRANCH RELATIVE ON INDEX LOW OR EQUAL (32)	BRXLE	RSI						B		7-28
86	BRANCH ON INDEX HIGH (32)	BXH	RS						B		7-25
87	BRANCH ON INDEX LOW OR EQUAL (32)	BXLE	RS						B		7-25
88	SHIFT RIGHT SINGLE LOGICAL (32)	SRL	RS								7-135
89	SHIFT LEFT SINGLE LOGICAL (32)	SLL	RS								7-132
8A	SHIFT RIGHT SINGLE (32)	SRA	RS	C							7-134
8B	SHIFT LEFT SINGLE (32)	SLA	RS	C				IF			7-132
8C	SHIFT RIGHT DOUBLE LOGICAL	SRDL	RS			SP					7-133
8D	SHIFT LEFT DOUBLE LOGICAL	SLDL	RS			SP					7-131
8E	SHIFT RIGHT DOUBLE	SRDA	RS	C		SP					7-133
8F	SHIFT LEFT DOUBLE	SLDA	RS	C		SP	IF				7-130

図 B-3 (3/13). 命令コードの順の命令一覧表

命令コード	名前	二モニック	特性					ページ	
90	STORE MULTIPLE (32)	STM	RS		A		ST	B ₂	7-139
91	TEST UNDER MASK	TM	SI	C	A			B ₁	7-145
92	MOVE (即値)	MVI	SI		A		ST	B ₁	7-91
93	TEST AND SET	TS	S	C	A	\$	ST	B ₂	7-144
94	AND (即値)	NI	SI	C	A		ST	B ₁	7-18
95	COMPARE LOGICAL (即値)	CLI	SI	C	A			B ₁	7-43
96	OR (即値)	OI	SI	C	A		ST	B ₁	7-108
97	EXCLUSIVE OR (即値)	XI	SI	C	A		ST	B ₁	7-77
98	LOAD MULTIPLE (32)	LM	RS		A			B ₂	7-86
99	TRACE (32)	TRACE	RS		P A SP	T ¢		B ₂	10-113
9A	LOAD ACCESS MULTIPLE	LAM	RS		A SP			UB	7-81
9B	STORE ACCESS MULTIPLE	STAM	RS		A SP		ST	UB	7-135
A50	INSERT IMMEDIATE (高 高)	IIHH	RI	N					7-80
A51	INSERT IMMEDIATE (高 低)	IIHL	RI	N					7-80
A52	INSERT IMMEDIATE (低 高)	IILH	RI	N					7-80
A53	INSERT IMMEDIATE (低 低)	IILL	RI	N					7-80
A54	AND IMMEDIATE (高 高)	NIHH	RI	C N					7-19
A55	AND IMMEDIATE (高 低)	NIHL	RI	C N					7-19
A56	AND IMMEDIATE (低 高)	NILH	RI	C N					7-19
A57	AND IMMEDIATE (低 低)	NILL	RI	C N					7-19
A58	OR IMMEDIATE (高 高)	OIHH	RI	C N					7-109
A59	OR IMMEDIATE (高 低)	OIHL	RI	C N					7-109
A5A	OR IMMEDIATE (低 高)	OILH	RI	C N					7-109
A5B	OR IMMEDIATE (低 低)	OILL	RI	C N					7-109
A5C	LOAD LOGICAL IMMEDIATE (高 高)	LLIHH	RI	N					7-86
A5D	LOAD LOGICAL IMMEDIATE (高 低)	LLIHL	RI	N					7-86
A5E	LOAD LOGICAL IMMEDIATE (低 高)	LLILH	RI	N					7-86
A5F	LOAD LOGICAL IMMEDIATE (低 低)	LLILL	RI	N					7-86
A70	TEST UNDER MASK (低 高)	TMLH	RI	C N					7-145
A70	TEST UNDER MASK HIGH	TMH	RI	C					7-145
A71	TEST UNDER MASK (低 低)	TMLL	RI	C N					7-145
A71	TEST UNDER MASK LOW	TML	RI	C					7-145
A72	TEST UNDER MASK (高 高)	TMHH	RI	C N					7-145
A73	TEST UNDER MASK (高 低)	TMHL	RI	C N					7-145
A74	BRANCH RELATIVE ON CONDITION	BRC	RI				B		7-26
A75	BRANCH RELATIVE AND SAVE	BRAS	RI				B		7-26
A76	BRANCH RELATIVE ON COUNT (32)	BRCT	RI				B		7-27
A77	BRANCH RELATIVE ON COUNT (64)	BRCTG	RI	N			B		7-27
A78	LOAD HALFWORD IMMEDIATE (32)	LHI	RI						7-85
A79	LOAD HALFWORD IMMEDIATE (64)	LGHI	RI	N					7-85
A7A	ADD HALFWORD IMMEDIATE (32)	AHI	RI	C		IF			7-16
A7B	ADD HALFWORD IMMEDIATE (64)	AGHI	RI	C N		IF			7-16
A7C	MULTIPLY HALFWORD IMMEDIATE (32)	MHI	RI						7-106
A7D	MULTIPLY HALFWORD IMMEDIATE (64)	MGHI	RI	N					7-106
A7E	COMPARE HALFWORD IMMEDIATE (32)	CHI	RI	C					7-42

図 B-3 (4/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニック	特性						ページ			
A7F A8 A9 AC AD	COMPARE HALFWORD IMMEDIATE (64) MOVE LONG EXTENDED COMPARE LOGICAL LONG EXTENDED STORE THEN AND SYSTEM MASK STORE THEN OR SYSTEM MASK	CGHI MVCLE CLCLE STNSM STOSM	RI RS RS SI SI	C C C C C	N 	 A A P P	SP SP SP SP SP	 ST ST ST	 R ₁ R ₁ R ₁ B ₁ B ₁	 R ₃ R ₃ R ₃ B ₁	7-42 7-95 7-46 10-105 10-105	
AE AF B1 B202 B204	SIGNAL PROCESSOR MONITOR CALL LOAD REAL ADDRESS (32) STORE CPU ID SET CLOCK	SIGP MC LRA STIDP SCK	RS SI RX S S	C C C	 	P A ¹ P P	SP SP SP SP SP	 ST 	 	 BP B ₂ B ₂	10-89 7-90 10-39 10-92 10-83	
B205 B206 B207 B208 B209	STORE CLOCK SET CLOCK COMPARATOR STORE CLOCK COMPARATOR SET CPU TIMER STORE CPU TIMER	STCK SCKC SCKC SPT STPT	S S S S S	C C C	 	A P P P P	SP SP SP SP SP	ST ST ST	 	B ₂ B ₂ B ₂ B ₂ B ₂	7-137 10-84 10-91 10-84 10-93	
B20A B20B B20D B210 B211	SET PSW KEY FROM ADDRESS INSERT PSW KEY PURGE TLB SET PREFIX STORE PREFIX	SPKA IPK PTLB SPX STPX	S S S S S	 	 	Q Q P P P	 A A	SP SP SP SP SP	 ST	 B ₂ B ₂	10-85 10-25 10-78 10-85 10-93	
B212 B218 B219 B21A B221	STORE CPU ADDRESS PROGRAM CALL SET ADDRESS SPACE CONTROL COMPARE AND FORM CODEWORD INVALIDATE PAGE TABLE ENTRY	STAP PC SAC CFC IPTE	S S S S RRE	 C 	 	P Q Q A P	SP A ¹ SP SP A ¹	 Z ¹ T SW II	 B I1	 ST ST 	B ₂ 	10-91 10-55 10-81 7-33 10-27
B222 B223 B224 B225 B226	INSERT PROGRAM MASK INSERT VIRTUAL STORAGE KEY INSERT ADDRESS SPACE CONTROL SET SECONDARY ASN EXTRACT PRIMARY ASN	IPM IVSK IAC SSAR EPAR	RRE RRE RRE RRE RRE	 C 	 	Q Q Q A ¹ Q	A ¹ A ¹ A ¹ A ¹ A ¹	S0 S0 Z ³ T S0	 	 	R ₂ 	7-81 10-26 10-24 10-86 10-19
B227 B228 B229 B22A B22B	EXTRACT SECONDARY ASN PROGRAM TRANSFER INSERT STORAGE KEY EXTENDED RESET REFERENCE BIT EXTENDED SET STORAGE KEY EXTENDED	ESAR PT ISKE RRBE SSKE	RRE RRE RRE RRE RRE	 C 	 	Q Q P P P	A ¹ A ¹ A ¹ A ¹ A ¹	S0 Z ² T ¢	 B 	 	 	10-20 10-72 10-25 10-78 10-89
B22C B22D B22E B22F B230	TEST BLOCK DIVIDE (拡張精度 HFP) PAGE IN PAGE OUT CLEAR SUBCHANNEL	TB DXR PGIN PGOUT CSCH	RRE RRE RRE RRE S	C C C	 ES ES 	P P P P P	A ¹ SP A ¹ A ¹ A ¹	II \$ G0 Da EU E0 FK ¢ ¢ OP ¢ GS	 	 	 	10-108 18-11 10-53 10-54 14-4
B231 B232 B233 B234 B235	HALT SUBCHANNEL MODIFY SUBCHANNEL START SUBCHANNEL STORE SUBCHANNEL TEST SUBCHANNEL	HSCH MSCH SSCH STSCH TSCH	S S S S S	C C C C C	 	P P P P P	OP A A A A	¢ GS ¢ GS ¢ GS ¢ GS ¢ GS	 ST ST	 B ₂ B ₂ B ₂ B ₂	14-6 14-7 14-14 14-17 14-19	

図 B-3 (5/13). 命令コードの順の命令一覧表

命令コード	名前	二 モ ニ ック	特性						ページ		
B236	TEST PENDING INTERRUPTION	TPI	S	C	P A ¹ SP	OP	¢	G1	ST	B ₂	14-17
B237	SET ADDRESS LIMIT	SAL	S		P		¢				14-11
B238	RESUME SUBCHANNEL	RSCH	S	C	P	OP	¢	GS			14-10
B239	STORE CHANNEL REPORT WORD	STCRW	S	C	P A SP		¢		ST	B ₂	14-16
B23A	STORE CHANNEL PATH STATUS	STCPS	S		P A SP		¢		ST	B ₂	14-15
B23B	RESET CHANNEL PATH	RCHP	S	C	P	OP	¢	G1			14-8
B23C	SET CHANNEL MONITOR	SCHM	S		P	OP	¢	GM			14-12
B240	BRANCH AND STACK	BAKR	RRE		A ¹	Z ⁵ T			B	ST	10-9
B241	CHECKSUM	CKSM	RRE	C	A SP					R ₂	7-29
B244	SQUARE ROOT (長精度 HFP)	SQDR	RRE			Da		SQ			18-18
B245	SQUARE ROOT (短精度 HFP)	SQER	RRE			Da		SQ			18-18
B246	STORE USING REAL ADDRESS (32)	STURA	RRE		P A ¹ SP				SU		10-105
B247	MODIFY STACKED STATE	MSTA	RRE		A ¹ SP	SE			ST		10-44
B248	PURGE ALB	PALB	RRE		P			\$			10-78
B249	EXTRACT STACKED REGISTERS (32)	EREG	RRE		A ¹	SE				U ₁ U ₂	10-20
B24A	EXTRACT STACKED STATE	ESTA	RRE	C	A ¹ SP	SE					10-22
B24B	LOAD USING REAL ADDRESS (32)	LURA	RRE		P A ¹ SP						10-44
B24C	TEST ACCESS	TAR	RRE	C	A ¹					U ₁	10-106
B24D	COPY ACCESS	CPYA	RRE							U ₁ U ₂	7-75
B24E	SET ACCESS	SAR	RRE							U ₁	7-129
B24F	EXTRACT ACCESS	EAR	RRE							U ₂	7-79
B250	COMPARE AND SWAP AND PURGE	CSP	RRE	C	P A ¹ SP			\$	ST	R ₂	10-17
B252	MULTIPLY SINGLE (32)	MSR	RRE								7-107
B254	MOVE PAGE	MVPG	RRE	C	Q A SP			G0	ST	R ₁ R ₂	10-46
B255	MOVE STRING	MVST	RRE	C	A SP			G0	ST	R ₁ R ₂	7-103
B257	COMPARE UNTIL SUBSTRING EQUAL	CUSE	RRE	C	A SP	II		GM		R ₁ R ₂	7-54
B258	BRANCH IN SUBSPACE GROUP	BSG	RRE		A ¹	SO T			B		10-12
B25A	BRANCH AND SET AUTHORITY	BSA	RRE		Q A ¹	SO T			B		10-5
B25D	COMPARE LOGICAL STRING	CLST	RRE	C	A SP			G0		R ₁ R ₂	7-53
B25E	SEARCH STRING	SRST	RRE	C	A SP			G0		R ₂	7-128
B263	COMPRESSION CALL	CMpsc	RRE	C	A SP	II Dd		GM	ST	R ₁ R ₂	7-57
B276	CANCEL SUBCHANNEL	XSCH	S	C	P	OP	¢	GS			14-4
B277	RESUME PROGRAM	RP	S	L	Q A SP	WE T			B	B ₂	10-79
B278	STORE CLOCK EXTENDED	STCKE	S	C	A			\$	ST	B ₂	7-137
B279	SET ADDRESS SPACE CONTROL FAST	SACF	S		Q SP	SW				B ₂	10-81
B27D	STORE SYSTEM INFORMATION	STSI	S	C	P A SP			GM	ST	B ₂	10-95
B299	SET ROUNDING MODE	SRNM	S			Db					19-47
B29C	STORE FPC	STFPC	S		A	Db			ST	B ₂	19-47
B29D	LOAD FPC	LFPC	S		A SP	Db				B ₂	19-39
B2A5	TRANSLATE EXTENDED	TRE	RRE	C	A SP				ST	R ₁ R ₂	7-148
B2A6	CONVERT UNICODE TO UTF-8	CUUTF	RRE	C	A SP				ST	R ₁ R ₂	7-69
B2A7	CONVERT UTF-8 TO UNICODE	CUTFU	RRE	C	A SP				ST	R ₁ R ₂	7-72
B2B1	STORE FACILITY LIST	STFL	S		N ³ P						10-93
B2B2	LOAD PSW EXTENDED	LPSWE	S	L	N P A SP			¢		B ₂	10-38
B2FF	TRAP	TRAP4	S		A	SO T			B	ST	10-114

図 B-3 (6/13). 命令コードの順の命令一覧表

命令コード	名前	二一モック	特性				ページ
B300	LOAD POSITIVE (短精度 BFP)	LPEBR	RRE	C	Db		19-41
B301	LOAD NEGATIVE (短精度 BFP)	LNEBR	RRE	C	Db		19-40
B302	LOAD AND TEST (短精度 BFP)	LTEBR	RRE	C	Db Xi		19-37
B303	LOAD COMPLEMENT (短精度 BFP)	LCEBR	RRE	C	Db		19-37
B304	LOAD LENGTHENED (短精度から長精度 BFP)	LDEBR	RRE		Db Xi		19-40
B305	LOAD LENGTHENED (長精度から拡張精度 BFP)	LXDBR	RRE		SP Db Xi		19-40
B306	LOAD LENGTHENED (短精度から拡張精度 BFP)	LXEBr	RRE		SP Db Xi		19-40
B307	MULTIPLY (長精度から拡張精度 BFP)	MXDBR	RRE		SP Db Xi		19-42
B308	COMPARE AND SIGNAL (短精度 BFP)	KEBR	RRE	C	Db Xi		19-25
B309	COMPARE (短精度 BFP)	CEBR	RRE	C	Db Xi		19-24
B30A	ADD (短精度 BFP)	AEBR	RRE	C	Db Xi	Xo Xu Xx	19-19
B30B	SUBTRACT (短精度 BFP)	SEBR	RRE	C	Db Xi	Xo Xu Xx	19-48
B30C	MULTIPLY (短精度から長精度 BFP)	MDEBR	RRE		Db Xi		19-42
B30D	DIVIDE (短精度 BFP)	DEBR	RRE		Db Xi Xz	Xo Xu Xx	19-31
B30E	MULTIPLY AND ADD (短精度 BFP)	MAEBr	RRF		Db Xi	Xo Xu Xx	19-44
B30F	MULTIPLY AND SUBTRACT (短精度 BFP)	MSEBR	RRF		Db Xi	Xo Xu Xx	19-44
B310	LOAD POSITIVE (長精度 BFP)	LPDBR	RRE	C	Db		19-41
B311	LOAD NEGATIVE (長精度 BFP)	LNDBR	RRE	C	Db		19-40
B312	LOAD AND TEST (長精度 BFP)	LTDBR	RRE	C	Db Xi		19-37
B313	LOAD COMPLEMENT (長精度 BFP)	LCDBR	RRE	C	Db		19-37
B314	SQUARE ROOT (短精度 BFP)	SQEBR	RRE		Db Xi	Xx	19-47
B315	SQUARE ROOT (長精度 BFP)	SQDBR	RRE		Db Xi	Xx	19-47
B316	SQUARE ROOT (拡張精度 BFP)	SQXBR	RRE		SP Db Xi	Xx	19-47
B317	MULTIPLY (短精度 BFP)	MEEBR	RRE		Db Xi	Xo Xu Xx	19-42
B318	COMPARE AND SIGNAL (長精度 BFP)	KDBR	RRE	C	Db Xi		19-25
B319	COMPARE (長精度 BFP)	CDBR	RRE	C	Db Xi		19-24
B31A	ADD (長精度 BFP)	ADBR	RRE	C	Db Xi	Xo Xu Xx	19-19
B31B	SUBTRACT (長精度 BFP)	SDBR	RRE	C	Db Xi	Xo Xu Xx	19-48
B31C	MULTIPLY (長精度 BFP)	MDBR	RRE		Db Xi	Xo Xu Xx	19-42
B31D	DIVIDE (長精度 BFP)	DDBR	RRE		Db Xi Xz	Xo Xu Xx	19-31
B31E	MULTIPLY AND ADD (長精度 BFP)	MADBR	RRF		Db Xi	Xo Xu Xx	19-44
B31F	MULTIPLY AND SUBTRACT (長精度 BFP)	MSDBR	RRF		Db Xi	Xo Xu Xx	19-44
B324	LOAD LENGTHENED (短精度から長精度 HFP)	LDER	RRE		Da		18-14
B325	LOAD LENGTHENED (長精度から拡張精度 HFP)	LXDR	RRE		SP Da		18-14
B326	LOAD LENGTHENED (短精度から拡張精度 HFP)	LXER	RRE		SP Da		18-14
B336	SQUARE ROOT (拡張精度 HFP)	SQXR	RRE		SP Da	SQ	18-18
B337	MULTIPLY (短精度 HFP)	MEER	RRE		Da EU E0		18-16
B340	LOAD POSITIVE (拡張精度 BFP)	LPXBR	RRE	C	SP Db		19-41
B341	LOAD NEGATIVE (拡張精度 BFP)	LNxBR	RRE	C	SP Db		19-40
B342	LOAD AND TEST (拡張精度 BFP)	LTXBR	RRE	C	SP Db Xi		19-37
B343	LOAD COMPLEMENT (拡張精度 BFP)	LCXBR	RRE	C	SP Db		19-37
B344	LOAD ROUNDED (長精度から短精度 BFP)	LEDBR	RRE		SP Db Xi	Xo Xu Xx	19-41
B345	LOAD ROUNDED (拡張精度から長精度 BFP)	LDXBR	RRE		SP Db Xi	Xo Xu Xx	19-41
B346	LOAD ROUNDED (拡張精度から短精度 BFP)	LEXBR	RRE		SP Db Xi	Xo Xu Xx	19-41
B347	LOAD FP INTEGER (拡張精度 BFP)	FIXBR	RRF		SP Db Xi	Xx	19-38

図 B-3 (7/13). 命令コードの順の命令一覧表

命令コード	名前	二 モ ニ ック	特性					ページ
B348	COMPARE AND SIGNAL (拡張精度 BFP)	KXBR	RRE C	SP	Db Xi			19-25
B349	COMPARE (拡張精度 BFP)	CXBR	RRE C	SP	Db Xi			19-24
B34A	ADD (拡張精度 BFP)	AXBR	RRE C	SP	Db Xi	Xo Xu Xx		19-19
B34B	SUBTRACT (拡張精度 BFP)	SXBR	RRE C	SP	Db Xi	Xo Xu Xx		19-48
B34C	MULTIPLY (拡張精度 BFP)	MXBR	RRE	SP	Db Xi	Xo Xu Xx		19-42
B34D	DIVIDE (拡張精度 BFP)	DXBR	RRE	SP	Db Xi Xz Xo Xu Xx			19-31
B350	CONVERT HFP TO BFP (長精度から短精度)	TBEDR	RRF C	SP	Da			9-9
B351	CONVERT HFP TO BFP (長精度)	TBDR	RRF C	SP	Da			9-9
B353	DIVIDE TO INTEGER (短精度 BFP)	DIEBR	RRF C	SP	Db Xi	Xu Xx		19-32
B357	LOAD FP INTEGER (短精度 BFP)	FIEBR	RRF	SP	Db Xi	Xx		19-38
B358	CONVERT BFP TO HFP (短精度から長精度)	THDER	RRE C		Da			9-8
B359	CONVERT BFP TO HFP (長精度)	THDR	RRE C		Da			9-8
B35B	DIVIDE TO INTEGER (長精度 BFP)	DIDBR	RRF C	SP	Db Xi	Xu Xx		19-32
B35F	LOAD FP INTEGER (長精度 BFP)	FIDBR	RRF	SP	Db Xi	Xx		19-38
B360	LOAD POSITIVE (拡張精度 HFP)	LPXR	RRE C	SP	Da			18-15
B361	LOAD NEGATIVE (拡張精度 HFP)	LNXR	RRE C	SP	Da			18-15
B362	LOAD AND TEST (拡張精度 HFP)	LTXR	RRE C	SP	Da			18-12
B363	LOAD COMPLEMENT (拡張精度 HFP)	LCXR	RRE C	SP	Da			18-13
B365	LOAD (拡張精度)	LXR	RRE	SP	Da			9-11
B366	LOAD ROUNDED (拡張精度から短精度 HFP)	LEXR	RRE	SP	Da	E0		18-16
B367	LOAD FP INTEGER (拡張精度 HFP)	FIXR	RRE	SP	Da			18-13
B369	COMPARE (拡張精度 HFP)	CXR	RRE C	SP	Da			18-9
B374	LOAD ZERO (短精度)	LZER	RRE		Da			9-11
B375	LOAD ZERO (長精度)	LZDR	RRE		Da			9-11
B376	LOAD ZERO (拡張精度)	LZXR	RRE	SP	Da			9-11
B377	LOAD FP INTEGER (短精度 HFP)	FIER	RRE		Da			18-13
B37F	LOAD FP INTEGER (長精度 HFP)	FIDR	RRE		Da			18-13
B384	SET FPC	SFPC	RRE	SP	Db			19-46
B38C	EXTRACT FPC	EFPC	RRE		Db			19-37
B394	CONVERT FROM FIXED (32 から短精度 BFP)	CEFBR	RRE		Db	Xx		19-27
B395	CONVERT FROM FIXED (32 から長精度 BFP)	CDFBR	RRE		Db			19-27
B396	CONVERT FROM FIXED (32 から拡張精度 BFP)	CFXBR	RRE	SP	Db			19-27
B398	CONVERT TO FIXED (短精度 BFP から 32)	CFEBR	RRF C	SP	Db Xi	Xx		19-28
B399	CONVERT TO FIXED (長精度 BFP から 32)	CFDBR	RRF C	SP	Db Xi	Xx		19-28
B39A	CONVERT TO FIXED (拡張精度 BFP から 32)	CFXBR	RRF C	SP	Db Xi	Xx		19-28
B3A4	CONVERT FROM FIXED (64 から短精度 BFP)	CEGBR	RRE N		Db	Xx		19-27
B3A5	CONVERT FROM FIXED (64 から長精度 BFP)	CDGBR	RRE N		Db	Xx		19-27
B3A6	CONVERT FROM FIXED (64 から拡張精度 BFP)	CXGBR	RRE N	SP	Db			19-27
B3A8	CONVERT TO FIXED (短精度 BFP から 64)	CEGBR	RRF C N	SP	Db Xi	Xx		19-28
B3A9	CONVERT TO FIXED (長精度 BFP から 64)	CGDBR	RRF C N	SP	Db Xi	Xx		19-28
B3AA	CONVERT TO FIXED (拡張精度 BFP から 64)	CGXBR	RRF C N	SP	Db Xi	Xx		19-28
B3B4	CONVERT FROM FIXED (32 から短精度 HFP)	CEFR	RRE		Da			18-10
B3B5	CONVERT FROM FIXED (32 から長精度 HFP)	CDFR	RRE		Da			18-10
B3B6	CONVERT FROM FIXED (32 から拡張精度 HFP)	CXFR	RRE	SP	Da			18-10
B3B8	CONVERT TO FIXED (短精度 HFP から 32)	CFER	RRF C	SP	Da			18-10

図 B-3 (8/13). 命令コードの順の命令一覧表

命令コード	名前	二 モ ニ ツ ク	特性					ページ					
B3B9	CONVERT TO FIXED (長精度 HFP から 32)	CFDR	R	R	C	SP	Da			18-10			
B3BA	CONVERT TO FIXED (拡張精度 HFP から 32)	CFXR	R	R	C	SP	Da			18-10			
B3C4	CONVERT FROM FIXED (64 から短精度 HFP)	CEGR	R	R	E	N		Da		18-10			
B3C5	CONVERT FROM FIXED (64 から長精度 HFP)	CDGR	R	R	E	N		Da		18-10			
B3C6	CONVERT FROM FIXED (64 から拡張精度 HFP)	CXGR	R	R	E	N	SP	Da		18-10			
B3C8	CONVERT TO FIXED (短精度 HFP から 64)	CGER	R	R	F	C	N	SP	Da		18-10		
B3C9	CONVERT TO FIXED (長精度 HFP から 64)	CGDR	R	R	F	C	N	SP	Da		18-10		
B3CA	CONVERT TO FIXED (拡張精度 HFP から 64)	CGXR	R	R	F	C	N	SP	Da		18-10		
B6	STORE CONTROL (32)	STCTL	R	S			P	A	SP		ST	B ₂	10-91
B7	LOAD CONTROL (32)	LCTL	R	S			P	A	SP			B ₂	10-37
B900	LOAD POSITIVE (64)	LPGR	R	R	E	C	N			IF			7-88
B901	LOAD NEGATIVE (64)	LNGR	R	R	E	C	N						7-88
B902	LOAD AND TEST (64)	LTGR	R	R	E	C	N						7-83
B903	LOAD COMPLEMENT (64)	LCGR	R	R	E	C	N			IF			7-84
B904	LOAD (64)	LGR	R	R	E	C	N						7-81
B905	LOAD USING REAL ADDRESS (64)	LURAG	R	R	E	C	N	P	A ¹	SP			10-44
B908	ADD (64)	AGR	R	R	E	C	N			IF			7-16
B909	SUBTRACT (64)	SGR	R	R	E	C	N			IF			7-141
B90A	ADD LOGICAL (64)	ALGR	R	R	E	C	N						7-17
B90B	SUBTRACT LOGICAL (64)	SLGR	R	R	E	C	N						7-142
B90C	MULTIPLY SINGLE (64)	MSGR	R	R	E	C	N						7-107
B90D	DIVIDE SINGLE (64)	DSGR	R	R	E	C	N						7-76
B90E	EXTRACT STACKED REGISTERS (64)	EREGR	R	R	E	C	N	A ¹	SP	SE	IK	U ₁ U ₂	10-20
B90F	LOAD REVERSED (64)	LRVGR	R	R	E	C	N						7-89
B910	LOAD POSITIVE (64<32)	LPGFR	R	R	E	C	N			IF			7-88
B911	LOAD NEGATIVE (64<32)	LNGFR	R	R	E	C	N						7-88
B912	LOAD AND TEST (64<32)	LTGFR	R	R	E	C	N						7-84
B913	LOAD COMPLEMENT (64<32)	LCGFR	R	R	E	C	N			IF			7-84
B914	LOAD (64<32)	LGFR	R	R	E	C	N						7-81
B916	LOAD LOGICAL (64<32)	LLGFR	R	R	E	C	N						7-85
B917	LOAD LOGICAL THIRTY ONE BITS	LLGTR	R	R	E	C	N						7-86
B918	ADD (64<32)	AGFR	R	R	E	C	N			IF			7-16
B919	SUBTRACT (64<32)	SGFR	R	R	E	C	N			IF			7-141
B91A	ADD LOGICAL (64<32)	ALGFR	R	R	E	C	N						7-17
B91B	SUBTRACT LOGICAL (64<32)	SLGFR	R	R	E	C	N						7-142
B91C	MULTIPLY SINGLE (64<32)	MSGFR	R	R	E	C	N						7-107
B91D	DIVIDE SINGLE (64<32)	DSGFR	R	R	E	C	N			SP	IK		7-76
B91F	LOAD REVERSED (32)	LRVR	R	R	E	C	N ³						7-89
B920	COMPARE (64)	CGR	R	R	E	C	N						7-33
B921	COMPARE LOGICAL (64)	CLGR	R	R	E	C	N						7-43
B925	STORE USING REAL ADDRESS (64)	STURG	R	R	E	C	N	P	A ¹	SP		SU	10-105
B930	COMPARE (64<32)	CGFR	R	R	E	C	N						7-33
B931	COMPARE LOGICAL (64<32)	CLGFR	R	R	E	C	N						7-43
B946	BRANCH ON COUNT (64)	BCTGR	R	R	E	C	N					B	7-24
B980	AND (64)	NGR	R	R	E	C	N						7-18

図 B-3 (9/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニック	特性						ページ		
B981	OR (64)	OGR	RRE	C	N					7-108	
B982	EXCLUSIVE OR (64)	XGR	RRE	C	N					7-77	
B986	MULTIPLY LOGICAL (128<64)	MLGR	RRE		N		SP			7-106	
B987	DIVIDE LOGICAL (64<128)	DLGR	RRE		N		SP		IK	7-75	
B988	ADD LOGICAL WITH CARRY (64)	ALCGR	RRE	C	N					7-17	
B989	SUBTRACT LOGICAL WITH BORROW (64)	SLBGR	RRE	C	N					7-143	
B98D	EXTRACT PSW	EPSW	RRE		N3					7-79	
B990	TRANSLATE TWO TO TWO	TRTT	RRE	C	E2	A	SP	GM	ST	RM R2	7-150
B991	TRANSLATE TWO TO ONE	TRTO	RRE	C	E2	A	SP	GM	ST	RM R2	7-150
B992	TRANSLATE ONE TO TWO	TROT	RRE	C	E2	A	SP	GM	ST	RM R2	7-150
B993	TRANSLATE ONE TO ONE	TROO	RRE	C	E2	A	SP	GM	ST	RM R2	7-150
B996	MULTIPLY LOGICAL (64<32)	MLR	RRE		N3		SP				7-106
B997	DIVIDE LOGICAL (32<64)	DLR	RRE		N3		SP		IK		7-75
B998	ADD LOGICAL WITH CARRY (32)	ALCR	RRE	C	N3						7-17
B999	SUBTRACT LOGICAL WITH BORROW (32)	SLBR	RRE	C	N3						7-143
B99D	EXTRACT AND SET EXTENDED AUTHORITY	ESEA	RRE		N	P					10-19
BA	COMPARE AND SWAP (32)	CS	RS	C		A	SP	\$	ST	B2	7-40
BB	COMPARE DOUBLE AND SWAP (32)	CDS	RS	C		A	SP	\$	ST	B2	7-40
BD	COMPARE LOGICAL C. UNDER MASK (低)	CLM	RS	C		A			ST	B2	7-43
BE	STORE CHARACTERS UNDER MASK (低)	STCM	RS			A			ST	B2	7-136
BF	INSERT CHARACTERS UNDER MASK (低)	ICM	RS	C		A				B2	7-79
C00	LOAD ADDRESS RELATIVE LONG	LARL	RIL		N3						7-83
C04	BRANCH RELATIVE ON CONDITION LONG	BRCL	RIL		N3				B		7-26
C05	BRANCH RELATIVE AND SAVE LONG	BRASL	RIL		N3				B		7-26
D1	MOVE NUMERICS	MVN	SS			A			ST	B1 B2	7-102
D2	MOVE (文字)	MVC	SS			A			ST	B1 B2	7-91
D3	MOVE ZONES	MVZ	SS			A			ST	B1 B2	7-104
D4	AND (文字)	NC	SS	C		A			ST	B1 B2	7-18
D5	COMPARE LOGICAL (文字)	CLC	SS	C		A			ST	B1 B2	7-43
D6	OR (文字)	OC	SS	C		A			ST	B1 B2	7-108
D7	EXCLUSIVE OR (文字)	XC	SS	C		A			ST	B1 B2	7-77
D9	MOVE WITH KEY	MVCK	SS	C		Q	A		ST	B1 B2	10-50
DA	MOVE TO PRIMARY	MVCP	SS	C		Q	A	SO ¢	ST		10-48
DB	MOVE TO SECONDARY	MVCS	SS	C		Q	A	SO ¢	ST		10-48
DC	TRANSLATE	TR	SS			A			ST	B1 B2	7-146
DD	TRANSLATE AND TEST	TRT	SS	C		A		GM		B1 B2	7-147
DE	EDIT	ED	SS	C		A		Dd	ST	B1 B2	8-7
DF	EDIT AND MARK	EDMK	SS	C		A		Dd	ST	B1 B2	8-9
E1	PACK UNICODE	PKU	SS		E2	A	SP		ST	B1 B2	7-111
E2	UNPACK UNICODE	UNPKU	SS	C	E2	A	SP		ST	B1 B2	7-156
E303	LOAD REAL ADDRESS (64)	LRAG	RXE	C	N	P	A ¹			BP	10-39
E304	LOAD (64)	LG	RXE		N	A				B2	7-81
E308	ADD (64)	AG	RXE	C	N	A		IF		B2	7-16
E309	SUBTRACT (64)	SG	RXE	C	N	A		IF		B2	7-141
E30A	ADD LOGICAL (64)	ALG	RXE	C	N	A				B2	7-17

図 B-3 (10/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニツク	特性						ページ			
E30B	SUBTRACT LOGICAL (64)	SLG	RXE	C	N	A			B ₂	7-142		
E30C	MULTIPLY SINGLE (64)	MSG	RXE	N	A				B ₂	7-107		
E30D	DIVIDE SINGLE (64)	DSG	RXE	N	A	SP		IK	B ₂	7-76		
E30E	CONVERT TO BINARY (64)	CVBG	RXE	N	A		Dd	IK	B ₂	7-68		
E30F	LOAD REVERSED (64)	LRVG	RXE	N	A				B ₂	7-89		
E314	LOAD (64<32)	LGF	RXE	N	A				B ₂	7-81		
E315	LOAD HALFWORD (64)	LGH	RXE	N	A				B ₂	7-84		
E316	LOAD LOGICAL (64<32)	LLGF	RXE	N	A				B ₂	7-85		
E317	LOAD LOGICAL THIRTY ONE BITS	LLGT	RXE	N	A				B ₂	7-86		
E318	ADD (64<32)	AGF	RXE	C	N	A		IF	B ₂	7-16		
E319	SUBTRACT (64<32)	SGF	RXE	C	N	A		IF	B ₂	7-141		
E31A	ADD LOGICAL (64<32)	ALGF	RXE	C	N	A			B ₂	7-17		
E31B	SUBTRACT LOGICAL (64<32)	SLGF	RXE	C	N	A			B ₂	7-142		
E31C	MULTIPLY SINGLE (64<32)	MSGF	RXE	N	A				B ₂	7-107		
E31D	DIVIDE SINGLE (64<32)	DSGF	RXE	N	A	SP		IK	B ₂	7-76		
E31E	LOAD REVERSED (32)	LRV	RXE	N3	A				B ₂	7-89		
E31F	LOAD REVERSED (16)	LRVH	RXE	N3	A				B ₂	7-89		
E320	COMPARE (64)	CG	RXE	C	N	A			B ₂	7-33		
E321	COMPARE LOGICAL (64)	CLG	RXE	C	N	A			B ₂	7-43		
E324	STORE (64)	STG	RXE	N	A				ST	B ₂	7-135	
E32E	CONVERT TO DECIMAL (64)	CVDG	RXE	N	A				ST	B ₂	7-68	
E32F	STORE REVERSED (64)	STRVG	RXE	N	A				ST	B ₂	7-140	
E330	COMPARE (64<32)	CGF	RXE	C	N	A			B ₂	7-33		
E331	COMPARE LOGICAL (64<32)	CLGF	RXE	C	N	A			B ₂	7-43		
E33E	STORE REVERSED (32)	STRV	RXE	N3	A				ST	B ₂	7-140	
E33F	STORE REVERSED (16)	STRVH	RXE	N3	A				B	ST	B ₂	7-140
E346	BRANCH ON COUNT (64)	BCTG	RXE	N	A						7-24	
E380	AND (64)	NG	RXE	C	N	A			B ₂	7-18		
E381	OR (64)	OG	RXE	C	N	A			B ₂	7-108		
E382	EXCLUSIVE OR (64)	XG	RXE	C	N	A			B ₂	7-77		
E386	MULTIPLY LOGICAL (128<64)	MLG	RXE	N	A	SP		IK	B ₂	7-106		
E387	DIVIDE LOGICAL (64<128)	DLG	RXE	N	A	SP			B ₂	7-75		
E388	ADD LOGICAL WITH CARRY (64)	ALCG	RXE	C	N	A			B ₂	7-18		
E389	SUBTRACT LOGICAL WITH BORROW (64)	SLBG	RXE	C	N	A			B ₂	7-143		
E38E	STORE PAIR TO QUADWORD	STPQ	RXE	N	A	SP			ST	B ₂	7-140	
E38F	LOAD PAIR FROM QUADWORD	LPQ	RXE	N	A	SP			B ₂	7-88		
E390	LOAD LOGICAL CHARACTER	LLGC	RXE	N	A				B ₂	7-85		
E391	LOAD LOGICAL HALFWORD	LLGH	RXE	N	A				B ₂	7-85		
E396	MULTIPLY LOGICAL (64<32)	ML	RXE	N3	A	SP			B ₂	7-106		
E397	DIVIDE LOGICAL (32<64)	DL	RXE	N3	A	SP		IK	B ₂	7-75		
E398	ADD LOGICAL WITH CARRY (32)	ALC	RXE	C	N3	A			B ₂	7-18		
E399	SUBTRACT LOGICAL WITH BORROW (32)	SLB	RXE	C	N3	A			B ₂	7-143		
E500	LOAD ADDRESS SPACE PARAMETERS	LASP	SSE	C		P A ¹	SP	SO		B ₁	10-28	
E501	TEST PROTECTION	TPROT	SSE	C		P A ¹				B ₁	10-110	
E502	STORE REAL ADDRESS	STRAG	SSE	N		P A ¹			ST	B ₁	BP	10-94

図 B-3 (11/13). 命令コードの順の命令一覧表

命令コード	名前	二一モニック	特性						ページ
			SSE	Q A	GM	ST	B ₁ B ₂		
E50E	MOVE WITH SOURCE KEY	MVCSK	SSE	Q A	GM	ST	B ₁ B ₂	10-52	
E50F	MOVE WITH DESTINATION KEY	MVCDK	SSE	Q A	GM	ST	B ₁ B ₂	10-50	
E8	MOVE INVERSE	MVCIN	SS	A		ST	B ₁ B ₂	7-91	
E9	PACK ASCII	PKA	SS	A		ST	B ₁ B ₂	7-110	
EA	UNPACK ASCII	UNPKA	SS C E2	A SP		ST	B ₁ B ₂	7-155	
EB04	LOAD MULTIPLE (64)	LMG	RSE N	A			B ₂	7-86	
EB0A	SHIFT RIGHT SINGLE (64)	SRAG	RSE C N					7-134	
EB0B	SHIFT LEFT SINGLE (64)	SLAG	RSE C N		IF			7-132	
EB0C	SHIFT RIGHT SINGLE LOGICAL (64)	SRLG	RSE N					7-135	
EB0D	SHIFT LEFT SINGLE LOGICAL (64)	SLLG	RSE N					7-133	
EB0F	TRACE (64)	TRACG	RSE N	P A SP	T ¢		B ₂	10-113	
EB1C	ROTATE LEFT SINGLE LOGICAL (64)	RLLG	RSE N					7-127	
EB1D	ROTATE LEFT SINGLE LOGICAL (32)	RLL	RSE N ³					7-127	
EB20	COMPARE LOGICAL C. UNDER MASK (高)	CLMH	RSE C N	A			B ₂	7-43	
EB24	STORE MULTIPLE (64)	STMG	RSE N	A		ST	B ₂	7-139	
EB25	STORE CONTROL (64)	STCTG	RSE N	P A SP		ST	B ₂	10-91	
EB26	STORE MULTIPLE HIGH	STMH	RSE N	A		ST	B ₂	7-140	
EB2C	STORE CHARACTERS UNDER MASK (高)	STCMH	RSE N	A		ST	B ₂	7-136	
EB2F	LOAD CONTROL (64)	LCTLG	RSE N	P A SP			B ₂	10-37	
EB30	COMPARE AND SWAP (64)	CSG	RSE C N	A SP	\$	ST	B ₂	7-40	
EB3E	COMPARE DOUBLE AND SWAP (64)	CDSG	RSE C N	A SP	\$	ST	B ₂	7-40	
EB44	BRANCH ON INDEX HIGH (64)	BXHG	RSE N			B		7-25	
EB45	BRANCH ON INDEX LOW OR EQUAL (64)	BXLEG	RSE N			B		7-25	
EB80	INSERT CHARACTERS UNDER MASK (高)	ICMH	RSE C N	A			B ₂	7-79	
EB8E	MOVE LONG UNICODE	MVCLU	RSE C E2	A SP		ST	R ₁ R ₂	7-99	
EB8F	COMPARE LOGICAL LONG UNICODE	CLCLU	RSE C E2	A SP			R ₁ R ₂	7-50	
EB96	LOAD MULTIPLE HIGH	LMH	RSE N	A			B ₂	7-87	
EBC0	TEST DECIMAL	TP	RSL C E2	A			B ₁	8-12	
EC44	BRANCH RELATIVE ON INDEX HIGH (64)	BRXHG	RIE N			B		7-28	
EC45	BRANCH RELATIVE ON INDEX L OR E (64)	BRXLG	RIE N			B		7-28	
ED04	LOAD LENGTHENED (短精度から長精度 BFP)	LDEB	RXE	A	Db Xi		B ₂	19-40	
ED05	LOAD LENGTHENED (長精度から拡張精度 BFP)	LXDB	RXE	A SP	Db Xi		B ₂	19-40	
ED06	LOAD LENGTHENED (短精度から拡張精度 BFP)	LXEB	RXE	A SP	Db Xi		B ₂	19-40	
ED07	MULTIPLY (長精度から拡張精度 BFP)	MXDB	RXE	A SP	Db Xi		B ₂	19-42	
ED08	COMPARE AND SIGNAL (短精度 BFP)	KEB	RXE C	A	Db Xi		B ₂	19-25	
ED09	COMPARE (短精度 BFP)	CEB	RXE C	A	Db Xi		B ₂	19-24	
ED0A	ADD (短精度 BFP)	AEB	RXE C	A	Db Xi Xo Xu Xx		B ₂	19-19	
ED0B	SUBTRACT (短精度 BFP)	SEB	RXE C	A	Db Xi Xo Xu Xx		B ₂	19-48	
ED0C	MULTIPLY (短精度から長精度 BFP)	MDEB	RXE	A	Db Xi		B ₂	19-42	
ED0D	DIVIDE (短精度 BFP)	DEB	RXE	A	Db Xi Xz Xo Xu Xx		B ₂	19-31	
ED0E	MULTIPLY AND ADD (短精度 BFP)	MAEB	RXF	A	Db Xi Xo Xu Xx		B ₂	19-44	
ED0F	MULTIPLY AND SUBTRACT (短精度 BFP)	MSEB	RXF	A	Db Xi Xo Xu Xx		B ₂	19-44	
ED10	TEST DATA CLASS (短精度 BFP)	TCEB	RXE C		Db			19-48	
ED11	TEST DATA CLASS (長精度 BFP)	TCDB	RXE C		Db			19-48	
ED12	TEST DATA CLASS (拡張精度 BFP)	TCXB	RXE C	SP	Db			19-48	

図 B-3 (12/13). 命令コードの順の命令一覧表

命令コード	名前	二 モ ニ ック	特性					ページ	
ED14	SQUARE ROOT (短精度 BFP)	SQEB	RXE	A	Db Xi	Xx	B ₂	19-47	
ED15	SQUARE ROOT (長精度 BFP)	SQDB	RXE	A	Db Xi	Xx	B ₂	19-47	
ED17	MULTIPLY (短精度 BFP)	MEEB	RXE	A	Db Xi	Xo Xu Xx	B ₂	19-42	
ED18	COMPARE AND SIGNAL (長精度 BFP)	KDB	RXE C	A	Db Xi		B ₂	19-25	
ED19	COMPARE (長精度 BFP)	CDB	RXE C	A	Db Xi		B ₂	19-24	
ED1A	ADD (長精度 BFP)	ADB	RXE C	A	Db Xi	Xo Xu Xx	B ₂	19-19	
ED1B	SUBTRACT (長精度 BFP)	SDB	RXE C	A	Db Xi	Xo Xu Xx	B ₂	19-48	
ED1C	MULTIPLY (長精度 BFP)	MDB	RXE	A	Db Xi	Xo Xu Xx	B ₂	19-42	
ED1D	DIVIDE (長精度 BFP)	DDB	RXE	A	Db Xi Xz	Xo Xu Xx	B ₂	19-31	
ED1E	MULTIPLY AND ADD (長精度 BFP)	MADB	RXF	A	Db Xi	Xo Xu Xx	B ₂	19-44	
ED1F	MULTIPLY AND SUBTRACT (長精度 BFP)	MSDB	RXF	A	Db Xi	Xo Xu Xx	B ₂	19-44	
ED24	LOAD LENGTHENED (短精度から長精度 HFP)	LDE	RXE	A	Da		B ₂	18-14	
ED25	LOAD LENGTHENED (長精度から拡張精度 HFP)	LXD	RXE	A SP	Da		B ₂	18-14	
ED26	LOAD LENGTHENED (短精度から拡張精度 HFP)	LXE	RXE	A SP	Da		B ₂	18-14	
ED34	SQUARE ROOT (短精度 HFP)	SQE	RXE	A	Da	SQ	B ₂	18-18	
ED35	SQUARE ROOT (長精度 HFP)	SQD	RXE	A	Da	SQ	B ₂	18-18	
ED37	MULTIPLY (短精度 HFP)	MEE	RXE	A	Da EU E0		B ₂	18-16	
EE	PERFORM LOCKED OPERATION	PLO	SS C	A SP		\$ GM	ST	FC	7-112
EF	LOAD MULTIPLE DISJOINT	LMD	SS	A			B ₂ B ₄		7-87
F0	SHIFT AND ROUND DECIMAL	SRP	SS C	A	Dd DF		ST	B ₁	8-11
F1	MOVE WITH OFFSET	MVO	SS	A			ST	B ₁ B ₂	7-104
F2	PACK	PACK	SS	A			ST	B ₁ B ₂	7-109
F3	UNPACK	UNPK	SS	A			ST	B ₁ B ₂	7-155
F8	ZERO AND ADD	ZAP	SS C	A	Dd DF		ST	B ₁ B ₂	8-13
F9	COMPARE DECIMAL	CP	SS C	A	Dd		ST	B ₁ B ₂	8-6
FA	ADD DECIMAL	AP	SS C	A	Dd DF		ST	B ₁ B ₂	8-5
FB	SUBTRACT DECIMAL	SP	SS C	A	Dd DF		ST	B ₁ B ₂	8-12
FC	MULTIPLY DECIMAL	MP	SS	A SP	Dd		ST	B ₁ B ₂	8-11
FD	DIVIDE DECIMAL	DP	SS	A SP	Dd	DK	ST	B ₁ B ₂	8-6

図 B-3 (13/13). 命令コードの順の命令一覧表

付録C. 条件コードの設定

この付録には、ESAME の命令のうち、条件コードを設定する命令について、条件コードの設定値をリストしてあります。条件コードは、ここにリストする命令により設定されるほかに、DIAGNOSE や、EXECUTE のターゲットにより変更されることもあります。条件コードは、LOAD PSW、LOAD PSW EXTENDED、PROGRAM RETURN、RESUME PROGRAM、SET PROGRAM MASK、および割り込みによりロードされ

ます。条件コードは、初期 CPU リセットにより 0 に設定され、初期プログラム・ローディング・シーケンスが正常に完了した時点でロードされます。

一部のモデルでは、例えば特殊機構や特注機構の補助のためまたはそれらの一部として提供されている命令などのように、本書では説明していない条件コードを設定する命令が用意されている場合があります。

命令	条件コード			
	0	1	2	3
ADD (gen) ADD (BFP) ADD DECIMAL ADD HALFWORD ADD HALFWORD IMMEDIATE	0 0 0 0 0	< 0 < 0 < 0 < 0 < 0	> 0 > 0 > 0 > 0 > 0	オーバーフロー NaN オーバーフロー オーバーフロー オーバーフロー
ADD LOGICAL ADD LOGICAL WITH CARRY ADD NORMALIZED ADD UNNORMALIZED AND	0、繰り上がりなし 0、繰り上がりなし 0 0 0	0 以外、繰り上がりなし 0 以外、繰り上がりなし < 0 < 0 0 以外	0、繰り上がりあり 0、繰り上がりあり > 0 > 0 --	0 以外、繰り上がりあり 0 以外、繰り上がりあり -- -- --
CANCEL SUBCHANNEL CHECKSUM CLEAR SUBCHANNEL COMPARE (gen, HFP) COMPARE (BFP)	機能が開始された チェックサム完了 機能が開始された 等しい 等しい	-- -- -- 小さい 小さい	-- -- -- 大きい 大きい	非稼働 CPU 決定の完了 非稼働 -- 順序付け不能
COMPARE AND FORM CODEWORD COMPARE AND SIGNAL COMPARE AND SWAP COMPARE AND SWAP AND PURGE COMPARE DECIMAL	等しい 等しい 等しい 等しい 等しい	OCB=0: 小さい OCB=1: 大きい 小さい 等しくない 等しくない 小さい	OCB=0: 大きい OCB=1: 小さい 大きい -- -- 大きい	-- 順序付け不能 -- -- --

図 C-1 (1/5). 条件コード設定の要約

命令	条件コード			
	0	1	2	3
COMPARE DOUBLE AND SWAP COMPARE HALFWORD COMPARE HALFWORD IMMEDIATE COMPARE LOGICAL COMPARE LOGICAL CHARACTERS UNDER MASK	等しい 等しい 等しい 等しい 等しい	等しくない 小さい 小さい 小さい 小さい	-- 大きい 大きい 大きい 大きい	-- -- -- -- --
COMPARE LOGICAL LONG COMPARE LOGICAL LONG EXTENDED COMPARE LOGICAL LONG UNICODE	等しい 等しい 等しい	小さい 小さい 小さい	大きい 大きい 大きい	-- CPU 決定の完了 CPU 決定の完了
COMPARE LOGICAL STRING	等しい	小さい	大きい	CPU 決定の完了
COMPARE UNTIL SUBSTRING EQUAL	等しいサブスト リング	最後のバイトが 等しい	最後のバイトが 等しくない	CPU 決定の完了
COMPRESSION CALL	Op2 が処理 された	Op1 は一杯、 op2 は処理 されない	--	CPU 決定の完了
CONVERT BFP TO HFP CONVERT HFP TO BFP CONVERT TO FIXED CONVERT UNICODE TO UTF-8	0 0 0 データが処理 された	< 0 < 0 < 0 Op1 が一杯	> 0 > 0 > 0 --	特殊ケース 特殊ケース 特殊ケース CPU 決定の完了
CONVERT UTF-8 TO UNICODE DIVIDE TO INTEGER	データが処理 された 完全な剰余、 正常な商	Op1 が一杯 完全な剰余、 商のオーバー フローまたは NaN	-- 不完全な剰余、 正常な商	CPU 決定の完了 不完全な剰余、 商のオーバー フローまたは NaN
EDIT EDIT AND MARK EXCLUSIVE OR	0 0 0	< 0 < 0 0 以外	> 0 > 0 --	-- -- --
EXTRACT STACKED STATE HALT SUBCHANNEL INSERT ADDRESS SPACE CONTROL INSERT CHARACTERS UNDER MASK LOAD ADDRESS SPACE PARAMETERS	ブランチ状態 エントリー 機能が開始 された 1 次スペース・ モード すべて 0 パラメーターが ロードされた	プログラム 呼び出し状態 エントリー 中間状況以外の 状況を伴う 状況保留 2 次スペース・ モード 最初のビットが1 1 次 ASN が使用 可能でない	-- 使用中 アクセス・ レジスター・ モード 最初のビットが0 2 次 ASN が使用 可能でないか または許可さ れていない	-- 非稼働 ホーム・スペー ス・モード -- スペース切り替 えイベント

図 C-1 (2/5). 条件コード設定の要約

命令	条件コード			
	0	1	2	3
LOAD AND TEST (gen, HFP) LOAD AND TEST (BFP) LOAD COMPLEMENT (gen) LOAD COMPLEMENT (BFP) LOAD COMPLEMENT (HFP)	0 0 0 0 0	< 0 < 0 < 0 < 0 < 0	> 0 > 0 > 0 > 0 > 0	-- NaN オーバーフロー NaN --
LOAD NEGATIVE (gen, HFP) LOAD NEGATIVE (BFP) LOAD POSITIVE (gen) LOAD POSITIVE (BFP) LOAD POSITIVE (HFP)	0 0 0 0 0	< 0 < 0 -- -- --	-- -- > 0 無効 > 0 > 0	-- NaN オーバーフロー 利用不可 または 長さ違反 NaN --
LOAD REAL ADDRESS MODIFY SUBCHANNEL MOVE LONG MOVE LONG EXTENDED MOVE LONG UNICODE	変換 SCHIB 情報が サブチャネル に入れられた 長さが等しい 長さが等しい 長さが等しい	ST エントリーが 無効 状況保留 長さが小さい 長さが小さい 長さが小さい	PT エントリーが 無効 使用中 長さが大きい 長さが大きい 長さが大きい	ASCE または エントリーが 利用不可 非稼働 破壊的オーバー ラップ CPU 決定の完了 CPU 決定の完了
MOVE PAGE MOVE STRING MOVE TO PRIMARY MOVE TO SECONDARY MOVE WITH KEY	データが移動 された -- 長さ > 256 長さ > 256 長さ > 256	オペランド 1 無効、ES 内で 両方有効、 ロック、また は ES エラー データが移動 された -- -- --	オペランド 2 無効 -- -- -- --	-- CPU 決定の完了 長さ > 256 長さ > 256 長さ > 256

図 C-1 (3/5). 条件コード設定の要約

命令	条件コード			
	0	1	2	3
OR PAGE IN	0 ページイン 操作が完了 した	0 以外 拡張記憶域 データ・ エラー	-- --	-- 拡張記憶域 ブロックが使 用可能でない
PAGE OUT	ページアウト 操作が完了 した	拡張記憶域 データ・ エラー	--	拡張記憶域 ブロックが使 用可能でない
PERFORM LOCKED OPERATION (テスト・ビットが 0 の 場合)	等しい	Op1 が等しく ない	Op1 は等しく、 op3 は等しく ない (dcs のみ)	--
PERFORM LOCKED OPERATION (テスト・ビットが 1 の 場合)	機能コードは 有効	--	--	機能コードが 無効
RESET CHANNEL PATH	機能が開始 された	--	使用中	非稼働
RESET REFERENCE BIT EXTENDED	R ビットは 0、 C ビットは 0	R ビットは 0、 C ビットは 1	R ビットは 1、 C ビットは 0	R ビットは 1、 C ビットは 1
RESUME SUBCHANNEL	機能が開始 された	状況保留中	機能は適用不能	非稼働
SEARCH STRING SET CLOCK	-- 設定	検出された セキュア	検出されない --	CPU 決定の完了 非稼働
SHIFT AND ROUND DECIMAL SHIFT LEFT (DOUBLE/SINGLE)	0	< 0	> 0	オーバーフロー
SHIFT RIGHT (DOUBLE/SINGLE)	0	< 0	> 0	オーバーフロー
SIGNAL PROCESSOR	0 指令が受け入れ られた	< 0 状況が格納 された	> 0 使用中	-- 非稼働
START SUBCHANNEL	機能が開始 された	状況保留中	使用中	非稼働
STORE CHANNEL REPORT WORD	CRW が格納 された	0 が格納された	--	--
STORE CLOCK	設定	設定されない	エラー	停止または 非稼働
STORE CLOCK EXTENDED	設定	設定されない	エラー	停止または 非稼働
STORE SUBCHANNEL	SCHIB が格納 された	--	--	非稼働
STORE SYSTEM INFORMATION	情報が提供 された	--	--	情報は利用不可
SUBTRACT (gen)	0	< 0	> 0	オーバーフロー
SUBTRACT (BFP)	0	< 0	> 0	NaN
SUBTRACT DECIMAL	0	< 0	> 0	オーバーフロー
SUBTRACT HALFWORD	0	< 0	> 0	オーバーフロー
SUBTRACT LOGICAL	--	0 以外、繰り 下げ	0、繰り下げ なし	0 以外、繰り 下げなし

図 C-1 (4/5). 条件コード設定の要約

命令	条件コード			
	0	1	2	3
SUBTRACT LOGICAL WITH BORROW SUBTRACT NORMALIZED (HFP) SUBTRACT UNNORMALIZED (HFP) TEST ACCESS	0、繰り下げあり 0 0 ALET 0	0 以外、繰り下げあり < 0 < 0 DU アクセス・リスト、例外なし	0、繰り下げなし > 0 > 0 PS アクセス・リスト、例外なし	0 以外、繰り下げなし -- -- ALET 1 または例外
TEST ADDRESSING MODE	24 ビット・モード	31 ビット・モード	--	64 ビット・モード
TEST AND SET TEST BLOCK TEST DATA CLASS TEST DECIMAL TEST PENDING INTERRUPTION	左ビットが 0 使用可能 0 (不一致) 数字および符号が有効 割り込みコードは格納されていない	左ビットが 1 使用可能でない 1 (不一致) 符号が無効 割り込みコードが格納された	-- -- -- 数字が無効	-- -- -- 符号および数字が無効
TEST PROTECTION TEST SUBCHANNEL TEST UNDER MASK TEST UNDER MASK (HIGH/LOW) TRANSLATE AND TEST	取り出し可能、格納可能 IRB が格納された。サブチャネル状況保留中 すべて 0 すべて 0 すべて 0	取り出し可能、格納不可 IRB が格納された。サブチャネルは状況保留中でない 混合 混合、左ビットは 0 未完了	取り出し不可、格納不可 -- -- 混合、左ビットは 1 完了	変換は使用可能でない 非稼働 すべて 1 すべて 1 --
TRANSLATE EXTENDED TRANSLATE ONE TO ONE, ONE TO TWO, TWO TO ONE, TWO TO TWO UNPACK ASCII UNPACK UNICODE UPDATE TREE	データが処理された 文字は等しい。テスト文字は検出されない 符号は正 符号は正 等しい	0p1 バイトがテスト・バイトに等しい テスト文字に等しい文字が検出された 符号は負 符号は負 等しくない、または比較なし	-- -- -- -- --	CPU 決定の完了 CPU 決定の完了 符号が無効 符号が無効 GR5 は 0 以外、GR0 は負
ZERO AND ADD	0	< 0	> 0	オーバフロー
説明: > 0 結果は 0 より大きい。 < 0 結果は 0 より小さい。 =< 256 256 またはそれより小さい。 > 256 256 より大きい。 gen 汎用命令。 BFP 2 進浮動小数点命令。 大きい 第 1 オペランドの方が大きい。 HFP 16 進浮動小数点命令。 小さい 第 1 オペランドの方が小さい。 長さ 第 1 オペランドの長さ。 NaN 非数値。 OCB オペランド制御ビット。				

図 C-1 (5/5). 条件コード設定の要約

付録D. 2 の累乗数の表

正	負
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1,024	10
2,048	11
4,096	12
8,192	13
16,384	14
32,768	15
65,536	16
131,072	17
262,144	18
524,288	19
1,048,576	20
2,097,152	21
4,194,304	22
8,388,608	23
16,777,216	24
33,554,432	25
67,108,864	26
134,217,728	27
268,435,456	28
536,870,912	29
1,073,741,824	30
2,147,483,648	31
4,294,967,296	32
8,589,934,592	33
17,179,869,184	34
34,359,738,368	35
68,719,476,736	36
137,438,953,472	37
274,877,906,944	38
549,755,813,888	39
1,099,511,627,776	40
2,199,023,255,552	41
4,398,046,511,104	42
8,796,093,022,208	43
17,592,186,044,416	44
35,184,372,088,832	45
70,368,744,177,664	46
140,737,488,355,328	47
281,474,976,710,656	48
562,949,953,421,312	49
1,125,899,906,842,624	50
2,251,799,813,685,248	51
4,503,599,627,370,496	52
9,007,199,254,740,992	53
18,014,398,509,481,984	54
36,028,797,018,963,968	55
72,057,594,037,927,936	56
144,115,188,075,855,872	57
288,230,376,151,711,744	58
576,460,752,303,423,488	59
1,152,921,504,606,846,976	60
2,305,843,009,213,693,952	61
4,611,686,018,427,387,904	62
9,223,372,036,854,775,808	63
18,446,744,073,709,551,616	64

図 D-1 (1/2). 2 の累乗数

18,446,744,073,709,551,616	64
36,893,488,147,419,103,232	65
73,786,976,294,838,206,464	66
147,573,952,589,676,412,928	67
295,147,905,179,352,825,856	68
590,295,810,358,705,651,712	69
1,180,591,620,717,411,303,424	70
2,361,183,241,434,822,606,848	71
4,722,366,482,869,645,213,696	72
9,444,732,965,739,290,427,392	73
18,889,465,931,478,580,854,784	74
37,778,931,862,957,161,709,568	75
75,557,863,725,914,323,419,136	76
151,115,727,451,828,646,838,272	77
302,231,454,903,657,293,676,544	78
604,462,909,807,314,587,353,088	79
1,208,925,819,614,629,174,706,176	80
2,417,851,639,229,258,349,412,352	81
4,835,703,278,458,516,698,824,704	82
9,671,406,556,917,033,397,649,408	83
19,342,813,113,834,066,795,298,816	84
38,685,626,227,668,133,590,597,632	85
77,371,252,455,336,267,181,195,264	86
154,742,504,910,672,534,362,390,528	87
309,485,009,821,345,068,724,781,056	88
618,970,019,642,690,137,449,562,112	89
1,237,940,039,285,380,274,899,124,224	90
2,475,880,078,570,760,549,798,248,448	91
4,951,760,157,141,521,099,596,496,896	92
9,903,520,314,283,042,199,192,993,792	93
19,807,040,628,566,084,398,385,987,584	94
39,614,081,257,132,168,796,771,975,168	95
79,228,162,514,264,337,593,543,950,336	96
158,456,325,028,528,675,187,087,900,672	97
316,912,650,057,057,350,374,175,801,344	98
633,825,300,114,114,700,748,351,602,688	99
1,267,650,600,228,229,401,496,703,205,376	100
2,535,301,200,456,458,802,993,406,410,752	101
5,070,602,400,912,917,605,986,812,821,504	102
10,141,204,801,825,835,211,973,625,643,008	103
20,282,409,603,651,670,423,947,251,286,016	104
40,564,819,207,303,340,847,894,502,572,032	105
81,129,638,414,606,681,695,789,005,144,064	106
162,259,276,829,213,363,391,578,010,288,128	107
324,518,553,658,426,726,783,156,020,576,256	108
649,037,107,316,853,453,566,312,041,152,512	109
1,298,074,214,633,706,907,132,624,082,305,024	110
2,596,148,429,267,413,814,265,248,164,610,048	111
5,192,296,858,534,827,628,530,496,329,220,096	112
10,384,593,717,069,655,257,060,992,658,440,192	113
20,769,187,434,139,310,514,121,985,316,880,384	114
41,538,374,868,278,621,028,243,970,633,760,768	115
83,076,749,736,557,242,056,487,941,267,521,536	116
166,153,499,473,114,484,112,975,882,535,043,072	117
332,306,998,946,228,968,225,951,765,070,086,144	118
664,613,997,892,457,936,451,903,530,140,172,288	119
1,329,227,995,784,915,872,903,807,060,280,344,576	120
2,658,455,991,569,831,745,807,614,120,560,689,152	121
5,316,911,983,139,663,491,615,228,241,121,378,304	122
10,633,823,966,279,326,983,230,456,482,242,756,608	123
21,267,647,932,558,653,966,460,912,964,485,513,216	124
42,535,295,865,117,307,932,921,825,928,971,026,432	125
85,070,591,730,234,615,865,843,651,857,942,052,864	126
170,141,183,460,469,231,731,687,303,715,884,105,728	127
340,282,366,920,938,463,463,374,607,431,768,211,456	128

図 D-1 (2/2). 2 の累乗数

付録E. 16 進数表

この付録の内容は、英語版のハードコピー・バージョンのみに収めてあります。

付録F. EBCDIC およびその他のコード

この付録の内容は、英語版のハードコピー・バージョンのみに収めてあります。

索引

日本語、英字、数字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

〔ア行〕

- アーキテクチャー
 - 互換性 1-14
- アーキテクチャー設定
 - プロセッサ信号指令 4-50
- アーキテクチャー・モード
 - 識別 3-50
 - 指示 12-2
 - 手動制御による選択 12-2
 - プロセッサ信号指令による選択 4-50
 - IML 制御による選択 12-3
- アクセス例外 6-35, 6-40
 - 認識 6-35
 - 優先順位 6-40
- アクセス・キー 3-8
 - チャンネル・サブシステム・モニター用 3-8
 - チャンネル・プログラム実行用 3-8, 15-20
 - CPU の 3-8
- アクセス・リスト 5-45
 - アクセス能力の取り消し 5-40
 - エントリーの使用許可 5-38
 - エントリーの割り振りとは無効化 5-37
 - 概念 5-36
 - 起点 (ALO) 5-45
 - 指定 (ALD) 5-45
 - 長さ (ALL) 5-45
 - 参照：アクセス・リスト・エントリー
- アクセス・リスト指定 5-45
- アクセス・リスト制御保護 3-10
 - 例外 6-27
- アクセス・リスト・エントリー (ALE) 5-45
 - 許可指標 (ALEAX) 5-46
 - シーケンス番号 (ALESN)
 - ALE 内の 5-46
 - ALET の中の 5-43
 - シーケンス例外 6-18
 - アクセス例外としての 6-35
 - トークン
 - 参照：ALET
 - 番号
 - 参照：ALEN
- アクセス・レジスター 2-4
 - 機能 5-35
 - 指示 5-35
- アクセス・レジスター (続き)
 - 使用するための命令 5-41
 - 妥当性ビット 11-21
 - 保管域 3-53
- アクセス・レジスター変換 (ART) 5-42
 - 概要 5-36
 - 索引バッファ
 - 参照：ALB
 - テーブル取り出しの順序 5-82
 - LOAD REAL ADDRESS、STORE REAL ADDRESS、TEST ACCESS、および TEST PROTECTION の一部と
- アクセス・レジスター変換 (ART) テーブル 5-44
- アクセス・レジスター・モード 3-27
- アクティブ
 - サブチャンネル 16-15
 - 装置 16-15
- アクティブ従属関係 15-11
- アクティブ通信 15-11
- アセンブラ言語 A-7
 - 命令フォーマット
 - 参照：付録 B の命令リストおよびページ番号
- 圧縮機能
 - 参照される資料 xxi
- 圧縮辞書
 - 記号変換の場合のサイズ 7-58
- 圧縮データ記号サイズ (CDSS) 7-58
- 圧縮データの左にあるビット 7-61
- 圧縮データの右にあるビット 7-61
- 圧縮データ・ビット番号 (CBN) 7-58
 - 例 7-63
- 圧縮プロセス
 - フローチャート 7-65
- アドレス 3-2
 - 演算 5-8
 - 符号なし 2 進数 7-4
 - 仮想 3-4
 - 記憶域 3-2
 - 基底
 - 参照：基底アドレス
 - 逆方向スタック・エントリー 5-69
 - 計算 3-5
 - 故障記憶機構
 - 参照：故障記憶機構アドレス
 - サイズ 3-5
 - アドレッシング・モードにより制御される 5-7
 - 循環
 - 参照：循環
 - 生成 5-7
 - 記憶域アドレッシング用の 3-6

アドレス (続き)

- 正方向セクション・ヘッダー 5-69
- 絶対 3-3
- タイプ 3-3
- 入出力 13-4
- バイト位置の番号付け 3-2
- 比較 12-1
 - 制御 12-1
 - CPU 状態への影響 4-2
- フォーマット 3-2
- ブランチ
 - 参照: ブランチ・アドレス
- プレフィックシング
 - 参照: プレフィックス
- 変換
 - 参照: 動的アドレス変換、プレフィックス
- 実 3-4
- 無効 6-15
- 命令
 - 参照: 命令アドレス
- 有効
 - 参照: 有効アドレス
- 要約情報 3-46
- 論理
 - 参照: 論理アドレス
- CPU
 - 参照: CPU アドレス
- PER
 - 参照: PER アドレス
- 1 次仮想
 - 参照: 1 次仮想アドレス
- 2 次仮想
 - 参照: 2 次仮想アドレス
- 24 ビット、31 ビット、および 64 ビット 3-5
 - オペランド・アドレス生成における 5-8
 - ブランチ・アドレス生成における 5-9
- アドレス限界検査 (入出力) 17-16
 - 限界モード (PMCW のビット) 15-2
 - 入出力システム・リセットの影響 17-12
- アドレス限界検査制御 (入出力) 15-22, 16-11
 - IPL に使用される 17-14
- アドレス指定状況記憶 (プロセッサ信号指令) 4-49
- アドレスのサイズ 3-5
 - アドレッシング・モードにより制御される 5-7
- CCW 内の 15-26
- アドレス・スペース 3-15
 - 制御ビット
 - アドレス変換での使用 3-27
 - 制御ビット 5-64
 - PSW 内の 4-6
 - 番号
 - 参照: ASN
 - 変更 3-16

アドレス・スペース (続き)

- AR 指定 5-35
 - DAT により作成される 3-25
- アドレス・スペース制御エレメント (ASCE) 3-27
 - タイプ例外 6-18
 - ホーム 3-29
 - AST エントリー内の 3-19
 - アクセス・レジスター変換に使用される 5-47
 - サブスペース・グループ用に使用される 5-57
 - 1 次 3-27
 - 2 次 3-29
- アドレッシングおよび変換モード識別 (ATMID) 4-26
- アドレッシング例外 6-15
 - アクセス例外としての 6-35, 6-40
- アドレッシング・モード 5-7
 - アドレス・サイズへの影響 3-6
 - オペランド・アドレス生成における 5-8
 - オペランド・アドレス生成への影響 5-8
 - 循環への影響 3-6
 - 順次命令アドレス生成への影響 5-7
 - 使用 5-17
 - ブランチ・アドレス生成における 5-9
 - リンケージ・スタック状態エントリー内のビット 5-71
 - リンケージ・スタック状態エントリー内のビット例の中の A-7
 - BRANCH AND SAVE AND SET MODE 命令により設定される 7-21
 - BRANCH AND SET MODE 命令により設定される 7-22
 - PSW のビット 4-6
 - SET ADDRESSING MODE 命令により設定される 7-129
- 誤り状態 (プロセッサ信号状況) 4-53
- 暗号化機構 1-11, 2-6
- 暗号操作例外 6-20
- アンスタッキング・プロセス 5-75
- アンスタック抑止ビット 5-67
- アンダーフロー
 - 参照: 指数アンダーフロー
- 暗黙的アドレス変換 3-33
- アンロック A-46
 - FIFO キューイングを使用した例 A-48
 - LIFO キューイングを使用した例 A-47
- 位置
 - 参照: アドレス
- 位置合わせ 3-3
 - 一貫性 (記憶域オペランド) 5-86
 - 例 A-47, A-50
- 一般命令 7-2
 - 例 A-7

イベント 6-13
 スペース切り替え 6-29
 モニター 7-90
 PER 4-24
 インストール 2-1
 インターフェース
 直列入出力
 参照される資料 xxi
 並列入出力
 参照されている OEMI 資料 xxi
 ESCON 入出力
 参照される資料 xxi
 インターフェース制御チェック (サブチャネル状況) 16-27
 インターロック更新記憶域参照 5-85
 埋め込みバイト
 COMPARE LOGICAL LONG 命令の場合 7-44
 COMPARE LOGICAL LONG EXTENDED 命令の場合
 7-47
 MOVE LONG 命令の場合 7-92
 MOVE LONG EXTENDED 命令の場合 7-96
 エポック (TOD クロック) 4-36
 エラー
 間接記憶機構 11-20
 記憶キー 11-19
 記憶機構 11-19
 検査/訂正 11-2
 固定 11-5
 断続的 11-5
 入出力エラー警報 16-34
 DIAGNOSE 命令からの 10-19
 PSW フォーマット 6-9
 TOD クロックの状態 4-35
 2 次 (入出力) 16-34
 エラー・リカバリー・コード (ERC) 17-20
 延期機能 13-8
 延期制御 15-20
 延期制御ビット 16-8
 IPL に使用される 17-14
 延期ビット (SCSW の) 16-15
 延期割り込み抑止制御 (入出力) 15-23, 16-11
 IPL に使用される 17-14
 演算
 アドレス
 参照: アドレス演算
 浮動小数点 9-1
 例 A-5, A-39
 論理 (符号なし 2 進数) 7-4
 例 A-3
 10 進
 例 A-4, A-34
 10 進数 8-2
 2 進数 7-4
 例 A-2
 エンタープライズ・システム接続アーキテクチャー (ESCON)
 入出力インターフェース
 参照される資料 xxi
 エンディアン 7-89
 エントリー
 拡張許可指標 5-65
 キー 5-65
 エントリー (トレース用) 4-13
 エントリー記述子 5-67
 エントリー指標 (EX) 5-28
 エントリー・キー・マスク (EKM) 5-30
 エントリー・タイプ・コード 5-67
 エントリー・テーブル (ET)
 起点 (ETO) 5-29
 指定 5-28
 長さ (ETL) 5-29
 エントリー・テーブル・エントリー 5-29
 エントリー・テーブル・エントリー (ETE) 5-64
 オーバーフロー
 固定小数点 6-22, 7-4
 指数
 参照: 指数オーバーフロー
 CRW 内の 17-19
 10 進 6-21
 2 進数 7-4
 例 A-2
 オーバーラップ
 オペラント 5-80
 一般命令の 7-2
 10 進数命令の 8-3
 操作 5-78
 破壊的 7-92, 7-96, 7-99
 オペラント 5-3
 アクセス 5-84
 入出力命令の場合 14-1
 アクセス識別 3-50
 アドレス生成 5-8
 オーバーラップ
 一般命令の 7-2
 10 進数命令の 8-3
 記憶域 5-6
 結果に使用される 5-3
 参照の順序 5-84
 即値 5-6
 タイプ (取り出し、格納、更新) 5-84
 長さ 5-3
 例外 6-24
 レジスター 5-6
 オペレーター介入 (プロセッサ信号状況) 4-54
 オペレーター機能 2-7, 12-1
 基本 12-1

〔力行〕

- 開始 (CPU)
 - キー 12-4
 - 機能 4-2
 - プロセッサ信号指令 4-48
- 開始機能 (入出力) 13-6, 15-17
 - バス管理 15-17
 - 保留 16-14
 - SCSW 内のビット 16-12
 - START SUBCHANNEL により開始される 14-14
- 解釈実行
 - 参照される資料 xxi
- 解釈実行機能 1-13
- 概念上の順序 5-78
 - 記憶域オペランド・アクセスとの関係 5-88
- 外部開始機能 4-40
 - 入出力 17-13
- 外部時刻参照 (ETR) 2-6
- 外部時刻参照 (ETR) 機能 1-11
- 外部損傷 11-16
 - サブクラス・マスク・ビット 11-24
- 外部損傷コード 11-22
 - 妥当性ビット 11-21
 - 割り当て済み記憶位置 3-52
- 外部呼び出し
 - 外部割り込み 6-11
 - プロセッサ信号指令 4-48
 - 保留 (プロセッサ信号状況) 4-53
- 外部割り込み 6-10
 - 外部呼び出し 6-11
 - 緊急信号 6-11
 - クロック・コンパレータ 4-38, 6-10
 - 誤動作警報 6-12
 - サービス信号 6-12
 - 条件の優先順位 6-10
 - 直接条件 6-10
 - パラメータ 6-10
 - 割り当て済み記憶位置 3-48
 - 保留条件 6-10
 - 割り込みキー 6-12
 - CPU タイマー 4-39, 6-11
 - ETR 6-11
 - PSW 内のマスク 4-5
- カウンターの更新 (例) A-45
- カウント操作 7-24
- カウント・フィールド
 - CCW 内の 15-26
 - 無効 16-24
 - SCSW 内の 16-28
- 拡張アドレッシング・モード
 - エンタリー・テーブル・エンタリー内のビット 5-30
- 拡張記憶機構 2-2
 - ブロック番号 2-2
- 拡張許可 5-52
- 拡張許可指標 (EAX) 5-43
 - エンタリー・テーブル・エンタリー内の 5-65
 - 制御ビット 5-64
 - リンケージ・スタック状態エンタリー内の 5-71
- 拡張権限例外 6-21
 - アクセス例外としての 6-35
- 拡張状況
 - サブチャネル・ログアウト内のフラグ 16-31
 - フォーマット 0 16-31
 - フォーマット 1 16-37
 - フォーマット 2 16-37
 - フォーマット 3 16-38
 - 2 次 CCW アドレス 16-36
- 拡張状況ワード 16-31
- 拡張状況ワード・フォーマット・ビット 16-8
- 拡張制御 (SCSW のビット) 16-11
- 拡張制御ワード 16-39
- 拡張精度 16 進浮動小数点数 18-3
- 拡張精度 2 進浮動小数点数 19-4
- 拡張ソート機能 1-11
- 格納参照 5-84
 - アクセス例外 6-37
- 格納時の遅延 5-84
- 数
 - 10 進 8-1
 - 例 A-4
 - 16 進 5-5
 - 16 進浮動小数点 18-3
 - 例 A-5
 - 2 進数 7-3
 - 例 A-2
 - 2 進浮動小数点 19-4
- 仮数 19-5
- 仮想アドレス 3-4
- 仮想記憶域参照のためのインターロック 5-79
- 仮想記憶機構 3-25
- 仮想計算機
 - 拡張機能 1-12
- 活動制御フィールド (SCSW) 16-13
 - TEST SUBCHANNEL の後の 14-19
- 可能性リスト 5-40
- 可変長フィールド 3-3
- 可用性 (システムの特性) 1-15
- 監視プログラム状態 4-6
- 監視プログラム呼び出し割り込み 6-44

- 間接記憶機構エラー 11-20
- 隔離された状態 16-35
- キー
 - アクセス
 - 参照：アクセス・キー
 - 記憶
 - 参照：記憶キー
 - 手動
 - 参照：手動操作
 - PSW
 - 参照：PSW キー
- キー制御保護 3-8
 - 例外 6-27
- キー・チェック (サブチャネル・ログアウト内の) 16-31
- キー・マスク
 - エントリ 5-30
 - 許可 5-30
 - PSW (PKM) 5-24
- 記憶域
 - アドレス循環
 - 参照：循環
 - アドレッシング
 - 参照：アドレス
 - インターロック更新 5-85
 - オペランド 5-6
 - 更新参照 5-85
 - 参照 (取り出し、格納、更新) 5-84
 - オペランドの一貫性 5-86
 - 例 A-47, A-50
 - 仮想記憶域参照のためのインターロック 5-79
 - 故障記憶機構
 - 参照：故障記憶機構アドレス
 - サイズ
 - 表記法 xx
 - 参照の順序
 - 浮動小数点データの 9-2
 - 参照のためのアクセスの同時性 5-87
 - 消去
 - 参照：消去操作
 - 非インターロック更新 5-85
 - ブロック 3-3
 - 使用可能性のテスト 10-108
 - 変更
 - スペース制御ビット 4-24
 - 変更 PER イベント 4-31
 - マスク 4-24
- 記憶域アクセス・コード (サブチャネル・ログアウト内の) 16-33
- 記憶域参照のためのアクセスの同時性 5-87
- 記憶域指示
 - 入出力操作での 15-27
 - PER イベントの 4-29
- 記憶域のブロック 3-3
 - 使用可能性のテスト 10-108
 - 参照：ページ
- 記憶域ブロックと記憶キーの使用可能性のテスト 10-108
- 記憶域へのアクセス 5-77
 - 参照：参照
- 記憶域変更イベント・ビット 4-25
- 記憶位置 3-2
 - 構成内で使用可能でない 6-15
- 記憶キー 3-8
 - エラー 11-19
 - 参照の順序 5-83
 - 使用可能性のテスト 10-108
 - 有効化 11-7
- 記憶キー機能 1-12
- 記憶キー内のアクセス制御ビット 3-8
- 記憶キー内の変更ビット 3-8
- 記憶機構 3-1, 3-28
 - アドレッシング 3-2
 - エラー 11-19
 - 間接 11-20
 - 拡張 2-2
 - 仮想 3-25
 - 揮発性 3-2
 - パワーオン・リセットの影響 4-46
 - 構成 3-3
 - 参照の順序 5-77
 - 主 3-2
 - 絶対 3-3
 - 直接アクセス 3-2
 - バッファ (キャッシュ) 3-2
 - 表示 12-2
 - 不揮発性 3-2
 - プレフィックス 3-14
 - 変更 PER イベント 3-28
 - ビット 3-28
 - 変更手動制御 12-2
 - 補助 3-2, 3-25
 - 実 3-4
 - 有効化 11-6
 - 割り当て済みの記憶位置 3-49
- 記憶機構共用
 - アドレス・スペースによる 3-25
 - マルチプロセッシングでの 4-47
 - 例 A-43
 - CPU およびチャネル・サブシステムによる 3-3
- 記憶機構性能低下 (マシン・チェック条件) 11-19
- 記憶機構変更イベント制御ビット 3-28
 - ホーム 3-30
 - 1 次 3-28
 - 2 次 3-29

- 記憶機構保護 3-8
 - トレース中の 4-23
- 記憶機構論理妥当性ビット 11-21
- 記憶装置再構成 1-12
- 記憶保護オーバーライド制御ビット 3-9
- 記憶保護チェック
 - サブチャネル状況としての 16-26
 - 測定ブロック 16-32
- 記憶保護例外 6-27
 - アクセス例外としての 6-35, 6-40
- 記号変換
 - VTAM による使用 7-64
- 記号変換オプション・ビット 7-58
- 記号変換テーブルのオフセット 7-58
- 疑似 AST エントリー 3-17
- 基数
 - 16 進 9-1
 - 2 進 9-1
- 基底アドレス 5-8
 - レジスター 2-3
- 規定境界 3-3
- 機能制御 (入出力) 16-12
- 機能保留時間 17-2
 - 測定ブロック内の 17-4
- 揮発性記憶機構 3-2
 - パワーオン・リセットの影響 4-46
- 基本 AST エントリー起点 (BASTEO) 5-56
- 基本 PROGRAM CALL 5-60, 10-57
- 基本アドレッシング・モード
 - エントリー・テーブル・エントリー内のビット 5-29
- 基本オペレーター機能 12-1
- 基本権限状態 10-6
- 基本スペース 5-14
- 基本スペース・ビット 5-57
- 基本入出力機能 15-1
- 逆移動
 - 参照: MOVE INVERSE 命令、逆移動機能
- 逆移動機能 7-91
- 逆方向スタック・エントリー妥当性ビット 5-69
- 逆方向スタック・エントリー・アドレス 5-69
- 逆方向スタック・エントリー・アドレスの妥当性ビット 5-69
- キャッシュ 3-2
- キューイング
 - FIFO
 - ロックおよびアンロックの例 A-47
 - LIFO
 - ロックおよびアンロックの例 A-46
- 旧 PSW 6-2
 - 割り当て済み記憶位置 3-52
- 境界合わせ 3-3
 - 命令の 5-3
- 共通セグメント・ビット 3-32
- 共通入出力装置コマンド
 - 参照される資料 xxi
- 共用 TOD クロック 4-33
- 共用記憶域
 - 参照: 記憶域共用
- 許可
 - キー・マスク (AKM) 5-30
 - 指標 (AX) 3-22, 5-25
 - テスト 5-63
 - メカニズム 5-23
 - 要約 5-26
 - ASN 3-22
- 許可検査 16-35
- 許可された割り込み 6-6
- 緊急信号 (外部割り込み) 6-11
 - プロセッサ信号指令 4-48
- 緊急マシン・チェック条件 11-11
- 禁止された割り込み 6-6
- 近似的有効 CBC 11-2
 - 記憶域内の 11-5
- クワイエット NaN (QNaN) 19-7
- クラス
 - BFP データの 19-6
 - テスト 19-48
- クリア操作
 - クリア・リセット機能による 4-45
 - システム・リセット消去キーによる 12-5
- クリア・リセット 4-45
- 繰り上がり 7-3
- 繰り下がり 7-143
- グリニッジ標準時 (GMT): 協定世界時と同義の廃語 4-36
- グルー・モジュール 5-18
- クロック
 - 参照: TOD クロック
- クロック単位 4-37, 7-139
- クロック・コンパレーター 4-38
 - 外部割り込み 6-10
 - 妥当性ビット 11-21
 - 保管域 3-53
- クワッドワード単位の記憶域参照 5-87
- 警告 (マシン・チェック条件) 11-17
 - サブクラス・マスク・ビット 11-24
- 警報 (マシン・チェック条件のクラス) 11-11
- 警報状況ビット (入出力) 16-16
- 警報割り込み条件 (入出力) 16-4
- 結果オペランド 5-3
- 結合機能 1-12
- 現 CCW 15-25
- 現 PSW 4-3, 5-10
 - 割り込み時の格納 6-2
 - 参照: PSW

限界モード (入出力) 15-2
権限テーブル (AT) 5-25
 起点 3-19, 5-47
 指定 3-19, 5-47
 長さ 3-19, 5-47
検査ビット 3-2, 11-2
検査ブロック 11-2
検査ブロック・コード
 参照: CBC
コード
 エラー・リカバリー (入出力) 17-20
 外部損傷 11-22
 妥当性ビット 11-21
 記憶域アクセス (サブチャンネル・ログアウト内の) 16-33
 検査ブロック
 参照: CBC
 条件
 参照: 条件コード
 据え置き条件 (入出力) 16-8
 データ例外 (DXC) 6-14
 入出力割り込みサブクラス 15-2
 バージョン 10-92
 報告ソース (入出力) 17-19
 命令 5-2
 命令長
 参照: ILC
 モニター
 参照: モニター・コード
 リンケージ・スタック・エントリー・タイプ 5-67
 割り込み
 参照: 割り込みコード
ASCII
 アーキテクチャーにより処理される xxi
EBCDIC
 アーキテクチャーにより処理される xxi
PER
 参照: PER コード
10 進数字および符号 8-2
8 ビット
 アーキテクチャーにより処理される xxi
コードワード (ソート操作) 7-33
 例 A-52
更新参照 5-85
構成 2-1
 記憶機構 3-3
構成警報機能 (入出力) 17-17
高速データ転送 (入出力) 13-3
互換性 1-14
 異なるアーキテクチャーをインプリメントするシステム間の
 1-15
 制御プログラム 1-15
 同一アーキテクチャーをインプリメントするシステム間の
 1-14

互換性 (続き)
 問題プログラム状態 1-15
故障記憶機構アドレス 11-23
 妥当性ビット 11-21
 妥当性フラグ (ERW 内の) 16-35
 割り当て済み記憶位置 3-52
 ESW 内の 16-31, 16-36
 チャンネル制御チェックの結果としての 16-27
 チャンネル・データ・チェックの結果としての 16-26
故障記憶機構アドレス・フォーマット
 標識 (ERW 内の) 16-36
 誤長 (サブチャンネル状況) 16-23
 即時操作での 15-29
 誤長表示モード 15-23
 誤長表示抑止機能 17-17
 即時操作への影響 15-29
 誤長抑止モード 15-23
 固定エラー 11-5
 固定小数点
 オーバフロー例外 6-22
 PSW 内のマスク 4-6
 除算例外 6-22
 固定長フィールド 3-2
 固定ログアウト
 マシン・チェック 11-24
 割り当て済み記憶位置 3-53
誤動作 11-2
 指示 11-4
 手動操作への影響 12-1
 チャンネル・サブシステムでの 16-27
 訂正 11-2
 入出力装置での 16-27
 入出力の一部として誤動作が検出された場合のマシン・チェ
 ックの取り扱い 11-5
 DIAGNOSE 命令からの 10-19
 誤動作警報 (外部割り込み) 6-12
 チェック停止状態に入るときの 11-11
 誤動作ユニットの削除 11-4
 コマンド (入出力) 15-27
 チャンネル内転送 15-39
 コマンド再試行 15-39
 PCI への影響 15-34
 コンソール装置 12-1
 コンソール統合 1-12

〔サ行〕

サービス信号外部割り込み 6-12
 サブクラス・マスク・ビット 6-12
サービス呼び出し論理プロセッサ (SCLP) 機能 1-13

サービス・プロセッサ操作不能 (プロセッサ信号状況) 4-54

サービス・プロセッサ損傷 11-17

再開機能 13-8, 15-17

- バス管理 15-17
- 保留 16-13
- RESUME SUBCHANNEL により開始される 14-10

再試行

- CPU 11-2

再始動

- キー 12-4
- プロセッサ信号指令 4-48
- 割り込み 6-44

サイズの表記法 xx

最大負数 7-3

作業従属関係 (入出力) 15-11

作動状態 4-1, 4-2

サブクラス・マスク・ビット

- 外部割り込み 6-10
- マシン・チェック 11-23

サブシステム識別ワード (SID)

- 割り当て済み記憶位置 3-51

サブシステム識別割り込み (SID) 14-1

サブシステム・リセット 4-44

サブシステム・リンケージ制御ビット 5-25, 5-28

サブスペース AST エントリ起点 (SSASTE0) 5-56

サブスペース AST エントリ・シーケンス番号 (SSASTESN) 5-57

サブスペース置き換え

- 操作 5-58
- 例外 6-43

サブスペース・アクティブ・ビット 5-56

サブスペース・グループ 5-55

- 概要 5-14

サブスペース・グループ制御 3-28

サブスペース・グループ制御ビット

- 1次 3-28
- 2次 3-29

サブチャンネル 13-2

- アイドル 16-13
- アクティブ従属関係 15-11
- 作業従属関係 15-11
- 専用従属関係 15-11
- 入出力システム・リセットの影響 17-11

サブチャンネル状況 16-23

サブチャンネルのアイドル状態 16-13

サブチャンネルの使用可能化 15-2, 16-5

サブチャンネル番号 13-5

サブチャンネル・アクティブ・ビット 16-15

サブチャンネル・アドレッシング 13-5

サブチャンネル・キー 15-20, 16-7

- アクセス・キーとして使用される 3-8
- IPL に使用される 17-14

サブチャンネル・キー・チェック (サブチャンネル・ログアウト内の) 16-31

サブチャンネル・ログアウト 16-31

サブルーチン用のリンケージ 5-10

サブルーチン・リンケージ 5-10

サポート機能 (入出力) 17-1

参照

- 格納順序
 - 参照: 順序
- 記憶キー内のビット 3-8
- 記憶機構参照の順序 5-77
- 記録 3-13
- 単一アクセス 5-86
- 複数アクセス 5-86

サンプル・カウント (ESW 内の) 17-3

残余フリー・スペース (リンケージ・スタック内の) 5-68

シーケンス・コード (サブチャンネル・ログアウト内の) 16-34

- フィールド妥当性フラグ 16-33

試行実行

- 編集命令および TRANSLATE 命令 5-23
- PER 4-26

時刻機構

- 参照: TOD クロック

事後正規化 18-3

辞書

- 適応 7-64

指数 18-1

- アンダーフロー
 - HFP 18-1
 - PSW 内のマスク 4-6
- オーバーフロー
 - HFP 18-1
- 参照: 指数部、浮動小数点
- 参照: 浮動小数点

指数のバイアス 19-5

指数バイアス 19-5

指数部 (HFP 数の) 18-1

- 参照: 指数

システム

- 手動制御 12-1
- 編成 2-1

システム損傷 11-15

システム・チェック停止 11-11

システム・マスク (PSW 内の) 4-3

- 妥当性ビット 11-20

システム・リカバリー 11-16

システム・リセット

- 参照: リセット

- システム・リセット消去キー 12-5
- システム・リセット通常キー 12-5
- 実モード 3-27
- 事前正規化 18-3
- 事前取り出し
 - オペランドの 5-84
 - 記憶キー内の無効 CBC の取り扱い 11-8
 - チャンネル制御チェック 16-27
 - チャンネル・データ・チェック 16-26
 - 入出力用のデータ 15-28
 - 認識されないアクセス例外 6-37
 - 命令の 5-81
 - ART テーブル・エントリーおよび DAT テーブル・エントリーの 5-82
- 事前取り出し制御 15-22
- 実記憶機構 3-4
- 実行例外 6-21
- 指定
 - エントリー・テーブル 5-28
 - 権限テーブル 3-19
 - セグメント・テーブル 3-27
 - アクセス・レジスタ変換に使用される 5-47
 - サブスペース・グループ用に使用される 5-57
 - AST エントリー内の 3-19
 - データの記憶域 (入出力) 15-27
 - ページ・テーブル 3-31
 - ホーム実スペース
 - ホーム領域テーブル 3-29
 - ホーム・実スペース 3-29
 - ホーム・セグメント・テーブル 3-29
 - 実スペース 3-27
 - アクセス・レジスタ変換に使用される 5-47
 - サブスペース・グループ用に使用される 5-57
 - AST エントリー内の 3-19
 - 領域テーブル 3-27
 - アクセス・レジスタ変換に使用される 5-47
 - サブスペース・グループ用に使用される 5-57
 - AST エントリー内の 3-19
 - リンケージ・テーブル 5-28
 - 1 次実スペース 3-27
 - 1 次セグメント・テーブル 3-27
 - 1 次領域テーブル 3-27
 - 2 次実スペース 3-29
 - 2 次セグメント・テーブル 3-29
 - 2 次領域テーブル 3-29
- 指定 (起点と長さ)
 - アクセス・リスト 5-44
- 指定タイプ制御
 - ホーム 3-30
 - 1 次 3-28
 - 2 次 3-29
- 指定例外 6-31
- 自動再構成 1-12
- 指標
 - アクセス・リストへの 5-43
 - アドレス生成のための 5-8
 - エントリー・テーブルおよびリンケージ・テーブルへの 5-28
 - 権限テーブル内の 5-25
 - ブランチ用の命令 7-25
 - レジスター 2-3
 - ASN 第 1 および第 2 テーブルへの 3-17
- シフト
 - 浮動小数点
 - 参照：正規化
- 従属関係
 - アクティブ 15-11
 - 作業 15-11
 - 専用 15-11
 - チャンネル・パス 15-10
 - CLEAR SUBCHANNEL への影響 15-10
- 充てんバイト (EDIT での) 8-7
- 終了コード (サブチャンネル・ログアウト内の) 16-33
 - フィールド妥当性フラグ 16-33
- 主記憶機構 3-2
 - 共用 (マルチプロセッシングでの) 4-47
 - パワーオン・リセットの影響 4-46
 - 参照：記憶機構
- 縮小権限状態 10-6
- 受信側チェック (プロセッサ信号状況) 4-54
- 手動操作 12-1
 - キー
 - 開始 12-4
 - 再始動 12-4
 - システム・リセット消去 12-5
 - システム・リセット通常 12-5
 - 状況記憶 12-5
 - 停止 12-4
 - ロード消去 12-3
 - ロード通常 12-3
 - 割り込み 12-3
- 制御
 - アドレス比較 12-1
 - 速度 12-4
 - 電源 12-4
 - 変更/表示 12-2
 - ロード装置アドレス 12-3
 - IML 12-3
 - TOD クロック 12-5
 - CPU の信号発信への影響 4-51
- 手動標識
 - 参照：停止状態

手動標識ライト 12-3
 循環
 記憶域アドレスの 3-6
 アドレッシング・モードにより制御される 3-6
 MOVE INVERSE 命令の 7-91
 MOVE LONG 命令の場合 7-92
 MOVE LONG EXTENDED 命令の場合 7-96
 MOVE LONG UNICODE 命令の場合 7-100
 命令アドレスの 5-7
 レジスター番号の
 LOAD MULTIPLE 命令の場合 7-87
 STORE MULTIPLE 命令の場合 7-139
 PER アドレスの 4-30
 TOD クロックの 4-34
 順序
 概念上の 5-78
 記憶域参照の
 オペランド 5-84
 記憶キー 5-83
 浮動小数点データの 9-2
 命令 5-81
 ART テーブル・エントリーおよび DAT テーブル・エントリー 5-82
 記憶機構参照の 5-77
 無効な CCW の 16-25
 命令実行 5-2
 順序付け不能 (NaN との比較) 19-9
 順序付け不能比較 19-24
 準特権
 プログラム 4-6, 5-23
 プログラム許可 5-23
 要約 5-26
 命令 4-6
 説明 10-1
 使用可能 (PMCW のビット) 15-2
 使用可能 (TRAP のビット) 10-114
 使用可能な ALB エントリー 5-54
 使用可能な TLB エントリー 3-43
 状況
 記憶 4-46
 手動キー 12-5
 警報 16-16
 サブチャンネル 16-23
 装置 16-22
 消去機能の影響 15-14
 フィールド妥当性フラグ (サブチャンネル・ログアウト内の) 16-33
 不適切なビットの組み合わせを持つ 16-34
 装置状況チェック 16-34
 中間 16-16
 プログラム
 参照 : PSW
 状況 (続き)
 プロセッサ信号指令の結果としての 4-52
 SIGNAL PROCESSOR の 4-48, 10-90
 1 次 16-17
 2 次 16-17
 状況記憶 4-46
 キー 12-5
 プロセッサ信号指令 4-48
 状況記憶アーキテクチャー・モード識別 3-50
 状況検査機能 17-16
 状況修飾ビット (装置状況)
 コマンド・チェーニングでの影響 15-32
 状況制御フィールド (SCSW 内の) 16-16
 状況保留 16-17
 消去機能 15-13
 完了 15-14
 サブチャンネル変更 15-13
 信号発信 15-14
 パス管理 15-13
 保留 16-14
 CLEAR SUBCHANNEL により開始される 14-4
 SCSW 内のビット 16-13
 消去信号 17-8
 消去機能の一環として発信される 15-14
 消去操作
 ロード消去キーによる 12-3
 TEST BLOCK 命令による 10-108
 条件コード 4-6
 据え置き 16-8
 妥当性ビット 11-21
 判断処理に使用される 5-10
 要約 C-1
 BFP 命令の 19-9
 BRANCH ON CONDITION 命令でテストされる 7-23
 PSW 内の 4-6
 条件付きスワッピング命令
 参照 : COMPARE AND SWAP 命令、COMPARE
 DOUBLE AND SWAP 命令
 小数部 18-1
 16 進数と 10 進数の間の変換 E-1
 状態
 CPU
 参照 : CPU 状態
 TOD クロック 4-34
 状態エントリー 5-69
 使用中
 入出力操作での 13-7
 SIGNAL PROCESSOR での 4-51
 冗長性 11-2
 情報のフォーマット 3-2

剰余 19-9
 DIVIDE TO INTEGER の結果 19-32
 初期 CPU リセット 4-44
 プロセッサ信号指令 4-49
 初期状況割り込み制御 15-22, 16-11
 IPL に使用される 17-14
 Z ビットとの関係 16-11
 初期プログラム・ローディング
 参照: IPL
 初期マシン・ローディング (IML) 制御 12-3
 除算例外
 固定小数点 6-22
 HFP 6-22
 10 進 6-21
 処理損傷 (同期マシン・チェック条件) 11-19
 処理バックアップ (同期マシン・チェック条件) 11-18
 資料
 他の関連資料 xxi
 指令 (入出力) 13-6, 15-27
 指令 (プロセッサ信号) 4-48
 アーキテクチャ設定 4-50
 アドレス指定状況記憶 4-49
 応答を妨げる条件 4-51
 開始 4-48
 外部呼び出し 4-48
 緊急信号 4-48
 再始動 4-48
 初期 CPU リセット 4-49
 センス 4-48
 停止 4-48
 停止/状況記憶 4-48
 プレフィックス設定 4-49
 CPU リセット 4-49
 新 PSW 4-3
 割り当て済み記憶位置 3-52
 割り込み時の取り出し 6-2
 シングル・バス・モード 15-3, 15-19
 信号 (入出力) 17-8
 信号付き NaN (SNaN) 19-7
 真の 0 (HFP 数) 18-1
 数字コード (10 進数) 8-2
 数字選択文字 (EDIT での) 8-7
 数字ビット 8-1
 移動 7-102
 据え置き条件コード 16-8
 スタッキング PROGRAM CALL 5-60
 スタッキング・プロセス 5-72
 スタック一杯例外 6-33
 スタック空例外 6-33
 スタック指定例外 6-33
 スタック操作例外 6-33
 スタック・タイプ例外 6-33
 スタンドアロン・ダンプ 12-5
 ストリーム・モード制御 15-21
 スペース切り替えイベント 6-29
 制御ビット
 ASTE 内の 3-19
 スペース切り替えイベント制御ビット
 ホーム 3-30
 1 次 3-28
 スワッピング
 COMPARE (DOUBLE) AND SWAP 命令による 7-40
 EXCLUSIVE OR 命令による 7-77
 正規化
 BFP 数の 19-8
 HFP 数 18-3
 請求割り込み条件 (入出力) 16-3
 制御 4-1
 手動
 参照: 手動操作
 命令 10-1
 制御装置 2-7, 13-3
 共用 13-4
 タイプ 15-12
 入出力システム・リセットの影響 17-10
 制御装置キューイング時間間隔 (測定ブロック内の) 17-4
 制御装置キューイング測定 (入出力) 17-6
 制御装置据え置き時間 (入出力) 17-5, 17-6
 制御プログラムの互換性 1-15
 制御レジスター 2-4, 4-7
 妥当性ビット 11-21
 保管域 3-54
 制御レジスター割り当て 4-8
 CR0.33:
 SSM 抑止制御ビット 6-30, 10-89
 CR0.34:
 TOD クロック同期制御ビット 4-34, 4-38
 CR0.35:
 低アドレス保護制御ビット 3-11
 CR0.36:
 抽出権限制御ビット 5-24
 CR0.37:
 2 次スペース制御ビット 3-27, 5-24
 CR0.38:
 取り出し保護オーバーライド制御ビット 3-10
 CR0.39:
 記憶保護オーバーライド制御ビット 3-9
 CR0.45:
 AFP レジスター制御ビット 9-2
 CR0.48:
 誤動作警報サブクラス・マスク・ビット 6-12
 CR0.49:
 緊急信号サブクラス・マスク・ビット 6-11

制御レジスター割り当て (続き)

- CR0.50:
外部呼び出しサブクラス・マスク・ビット 6-11
- CR0.52:
クロック・コンパレーター・サブクラス・マスク・ビット 6-11
- CR0.53:
CPU タイマー・サブクラス・マスク・ビット 6-11
- CR0.54:
サービス信号サブクラス・マスク・ビット 6-12
- CR0.57:
割り込みキー・サブクラス・マスク・ビット 6-12
- CR0.59:
ETR サブクラス・マスク・ビット 6-11
- CR0.61:
暗号制御ビット 6-20
- CR10.0-63:
PER 開始アドレス 4-24
- CR11.0-63:
PER 終了アドレス 4-24
- CR12.0:
ブランチ・トレース制御ビット 4-13
- CR12.1-61:
トレース・エントリー・アドレス 4-13
- CR12.1:
モード・トレース制御ビット 4-13
- CR12.62:
ASN トレース制御ビット 4-13
- CR12.63:
明示トレース制御ビット 4-13
- CR13.0-51:
ホーム実スペース・トークン起点 (HRSTKO) 3-30
ホーム領域テーブル起点 (HRTO) 3-30
ホーム・セグメント・テーブル起点 (HSTO) 3-30
- CR13.55:
ホーム専用スペース制御ビット 3-30
- CR13.56:
ホーム記憶機構変更イベント制御ビット 3-30
- CR13.57:
ホーム・スペース切り替えイベント制御ビット 3-30, 6-29
- CR13.58:
ホーム実スペース制御ビット
ホーム実スペース制御ビット 3-30
- CR13.60-61:
ホーム指定タイプ制御ビット 3-30
- CR13.62-63:
ホーム領域テーブル長さ (HRTL) 3-30
ホーム・セグメント・テーブル長さ (HSTL) 3-30
- CR13:
ホーム・アドレス・スペース制御エレメント (HASCE) 3-29

制御レジスター割り当て (続き)

- CR14.35:
チャンネル報告保留サブクラス・マスク・ビット 11-24
- CR14.36:
性能低下サブクラス・マスク・ビット 11-24
リカバリー・サブクラス・マスク・ビット 11-24
- CR14.38:
外部損傷サブクラス・マスク・ビット 11-24
- CR14.39:
警告サブクラス・マスク・ビット
警告サブクラス・マスク・ビット 11-24
- CR14.42:
TOD クロック制御オーバーライド制御 4-34
- CR14.44:
ASN 変換制御ビット 3-17, 5-25
- CR14.45-63:
ASN 第 1 テーブル起点 (AFTO) 3-18
- CR15.0-60:
リンケージ・スタック・エントリー・アドレス 5-67
- CR1.0-51:
CR1.0-51:
1 次実スペース・トークン起点 (PRSTKO) 3-29
1 次セグメント・テーブル起点 (PSTO) 3-28
1 次領域テーブル起点 (PRTO) 3-28
- CR1.54:
1 次サブスペース・グループ制御ビット 3-28
- CR1.55:
1 次専用スペース制御ビット 3-28
- CR1.56:
1 次記憶機構変更イベント制御ビット 3-28
- CR1.57:
1 次スペース切り替えイベント制御ビット 3-28, 6-29
- CR1.58:
1 次実スペース制御ビット 3-28
- CR1.60-61:
1 次指定タイプ制御ビット 3-28
- CR1.62-63:
1 次セグメント・テーブル長さ (PSTL) 3-29
1 次領域テーブル長さ (PRTL) 3-29
- CR1:
1 次アドレス・スペース制御エレメント (PASCE) 3-27
- CR2.33-57:
ディスパッチ可能単位制御テーブル起点 (DUCTO) 5-42
- CR3.32-47:
PSW キー・マスク (PKM) 5-24
- CR3.48-63:
2 次 ASN (SASN) 3-16
- CR4.32-47:
許可指標 (AX) 3-22, 5-25

制御レジスター割り当て (続き)

- CR4.48-63:
 - 1 次 ASN (PASN) 3-16
 - CR5.33-57:
 - 1 次 AST エントリー起点 (PASTE0) 5-28, 5-42
 - CR6.32-39:
 - 入出力割り込みサブクラス・マスク 6-13
 - CR7.0-51:
 - SSTO (2 次セグメント・テーブル起点) 3-29
 - 2 次実スペース・トークン起点 (SRSTKO) 3-29
 - 2 次領域テーブル起点 (SSTO) 3-29
 - 2 セグメント・テーブル起点 (SSTO)
 - CR7.54:
 - 2 次サブスペース・グループ制御ビット 3-29
 - CR7.55:
 - 2 次専用スペース制御ビット 3-29
 - CR7.56:
 - 2 次記憶機構変更イベント制御ビット 3-29
 - CR7.58:
 - 2 次実スペース制御ビット 3-29
 - CR7.60-61:
 - 2 次指定タイプ制御ビット 3-29
 - CR7.62-63:
 - 2 次セグメント・テーブル長さ (SSTL) 3-29
 - 2 次領域テーブル長さ (SRTL) 3-29
 - CR7:
 - 2 次アドレス・スペース制御エレメント (SASCE) 3-29
 - CR8.32-47:
 - 拡張許可指標 (EAX) 5-43
 - CR8.48-63:
 - モニター・マスク・ビット 6-23
 - CR9.32:
 - PER のブランチ成功イベント・マスク・ビット 4-24
 - CR9.33:
 - PER の命令取り出しイベント・マスク・ビット 4-24
 - CR9.34:
 - PER の記憶域変更イベント・マスク・ビット 4-24
 - CR9.36:
 - PER の実アドレスを使用した格納イベント・マスク・ビット 4-24
 - CR9.40:
 - PER のブランチ・アドレス制御ビット 4-24
 - CR9.42:
 - PER の記憶域変更スペース制御ビット 4-24
- (CRx.y は、制御レジスター x、ビット位置 y を示す)

整数

- 10 進数 8-2
- 2 進数 7-3
 - アドレス 5-8
 - 浮動小数点からの変換 18-10, 19-28
 - 浮動小数点への変換 18-10, 19-27

整数 (続き)

- 2 進数 (続き)
 - 例 A-2
- 整数商 19-32
- 精度 (浮動小数点) 9-1
- 性能低下 (マシン・チェック条件) 11-17
 - サブクラス・マスク・ビット 11-24
- 性能低下、記憶機構 (マシン・チェック条件) 11-19
- 正方向セクション妥当性ビット 5-69
- 正方向セクション・ヘッダー・アドレス 5-69
- 正方向セクション・ヘッダー・アドレスの妥当性ビット 5-69
- セグメント 3-26
- セグメント指標 (SX) 3-26
- セグメント変換例外 6-29
 - アクセス例外としての 6-35, 6-40
- セグメント無効ビット (セグメント・テーブル・エントリー内の) 3-32
- セグメント・テーブル 3-31
 - 起点 (STO) 3-27
 - 長さ (STL) 3-27
- セグメント・テーブル指定 (STD)
 - アクセス・レジスター変換での取得 5-36
 - ホーム 3-29
 - 1 次 3-27
 - 2 次 3-29
- 絶対アドレス 3-3
- 絶対記憶機構 3-3
- 設定状態 (TOD クロックの) 4-34
- ゼロ
 - バイト値としての通常の意味 xxi
- センス
 - プロセッサ信号指令としての 4-48
- 選択可能なバス 15-12
- 専用従属関係 15-11
- 専用スペース制御影響
 - 共通セグメントの使用 3-32
 - 低アドレス保護 3-11
 - 取り出し保護オーバーライド 3-10
 - 取り出し保護オーバーライド
- 専用スペース制御ビット 3-28
 - ホーム 3-30
 - 1 次 3-28
 - 2 次 3-29
- 専用ビット 5-46
- ソート
 - 拡張 1-11
 - ソートのためのツリー構造 7-157
 - 例 A-51
 - ソート命令
 - 例 A-51

- ゾーン 10 進数 8-1
 - 例 A-4
- ゾーン・ビット 8-1
 - 移動 7-104
- 早期例外認識 6-9
- 走行中 (TOD クロックの状態) 4-34
- 操作
 - 単位 5-20
 - 操作単位 5-20
 - 操作単位の完了 5-21
 - 操作不能 (プロセッサ信号状況) 4-54
 - 操作例外 6-24
 - 相対ブランチ 5-9
 - 装置 2-7, 13-4
 - コンソール 12-1
 - 入出力システム・リセットの影響 17-10
 - 装置 ID 13-5
 - 装置アクティブ時間間隔 (測定ブロック内の) 17-5
 - 装置アクティブ測定 (入出力) 17-6
 - 装置アクティブ・ビット 16-15
 - 装置アドレス 13-5
 - 装置状況 16-22
 - フィールド妥当性フラグ (サブチャンネル・ログアウトでの) 16-27, 16-33
 - 不適切なビットの組み合わせを持つ 16-34
 - 装置状況チェック 16-34
 - 装置接続時間測定 17-7
 - 延期の影響 15-38
 - 使用可能 15-3
 - 装置切断時間間隔 (測定ブロック内の) 17-4
 - 装置チェック
 - プロセッサ信号状況の中の 4-53
 - 装置チェック (装置状況)
 - 専用従属関係を確立するときの 15-11
 - 装置番号 13-5
 - 割り当て 13-5
 - PMCW 内の 15-4
 - 装置番号有効 (PMCW のビット) 15-4
 - 即値オペランド 5-6
 - 即時操作
 - CCW 内の SLI フラグ 15-29
 - 即時操作 (入出力) 15-40
 - 測定
 - 装置接続時間 17-7
 - 測定ブロック更新 (入出力) 17-2
 - ブロック (入出力)
 - 起点 17-5
 - 測定データ (入出力)
 - 累積 17-3
 - CSCH の影響 14-5
 - HSCH の影響 14-6

- 測定ブロック (入出力) 17-3
 - キー (MBK)
 - アクセス・キーとして使用される 3-8
 - 記憶保護チェック 16-32
 - 更新使用可能 15-3
 - 指標 15-6
 - データ・チェック 16-32
 - 複数使用 15-10
 - プログラム・チェック 16-32
- 測定モード制御 (入出力) 15-3
- 速度制御 12-4
- 損傷
 - 外部 11-16
 - サブクラス・マスク・ビット 11-24
 - コード (外部) 11-22
 - 妥当性ビット 11-21
 - サービス・プロセッサ 11-17
 - システム 11-15
 - 処理 11-19
 - タイミング機構 11-16
 - チャンネル・サブシステム 11-17
 - 命令処理 11-15
- 損傷点 11-14

〔夕行〕

- ターゲット命令 7-78
- タイマー
 - 参照 : CPU タイマー
- タイミング
 - チャンネル・サブシステム 17-2
 - タイミング機構 4-33
 - タイミング機構の損傷 11-16
 - TOD クロック 4-35
 - タイミング機構ビット (PMCW の) 15-3
 - 妥当性ビット
 - サブチャンネル・ログアウト内の 16-33
 - マシン・チェック割り込みコード内の 11-20
- ダブルワード 3-3
- ダブルワード単位の記憶域参照 5-87
- 単一アクセス参照 5-86
- 短精度 16 進浮動小数点数 18-3
- 短精度 2 進浮動小数点数 19-4
- 断続的エラー 11-5
- ダンプ (スタンドアロン) 12-5
- チェーニング・チェック (サブチャンネル状況) 16-28
- チェック停止 4-2, 11-10
 - 誤動作警報 6-12
 - システム 11-11
 - 手動操作時の 12-1
 - 入る 11-13
 - 標識 12-3

チェック停止 (続き)
 プロセッサ信号状況としての 4-54
 CPU タイマーへの影響 4-39
 チェックポイント 11-2
 チェックポイント同期化 11-3
 アクション 11-4
 操作 11-3
 遅延アクセス例外 (マシン・チェック条件) 11-18
 遅延例外認識 6-9
 逐次化 5-89
 格納操作の完了時の 5-84
 チャンネル・プログラム 5-90
 入出力命令により生じる 14-1
 CPU 5-89
 チャンネル制御チェック (サブチャンネル状況) 16-27
 チャンネル報告 17-18
 RCHP の結果として生成される 14-9
 チャンネル報告保留 11-17, 17-18
 サブクラス・マスク・ビット 11-24
 入出力システム・リセットの影響 17-12
 チャンネル・サブシステム 2-6, 13-1
 隔離された状態 16-35
 使用されるアドレッシング 13-4
 損傷 11-17
 入出力システム・リセットの影響 17-10
 パワーオン・リセットの影響 4-46
 チャンネル・サブシステム呼び出し機能 1-11
 チャンネル・サブシステム・タイマー 17-2
 入出力システム・リセットの影響 17-12
 チャンネル・サブシステム・タイミング 17-2
 チャンネル・サブシステム・タイミング機構ビット (PMCW
 の) 15-3
 チャンネル・サブシステム・モニター 17-1
 入出力システム・リセットの影響 17-12
 チャンネル・サブシステム・リカバリー 11-4, 17-17
 チャンネル・データ・チェック (サブチャンネル状況) 16-26
 チャンネル・パス 13-2
 アクティブ従属関係 15-11
 作業従属関係 15-11
 状況の格納 14-15
 選択可能な 15-12
 専用従属関係 15-11
 タイプ 13-2, 13-5
 直列入出力インターフェース・タイプ 13-2
 入出力システム・リセットの影響 17-11
 非稼働 16-12
 並列入出力インターフェース・タイプ 13-3
 マルチバス・モード 15-3, 15-19
 チャンネル・パス状況ワード 14-15

 チャンネル・パス・タイムアウト
 標識 (ERW 内の) 16-35
 チャンネル・パス・リセット 17-9
 入出力システム・リセットの影響 17-12
 チャンネル・パス・リセット機能 15-42
 完了 15-43
 信号発信 15-42
 その一部として発信されるリセット信号 17-9
 RESET CHANNEL PATH の開始 14-8
 チャンネル・プログラム 15-25
 延期制御 15-20
 実行 13-5, 15-19
 延期 13-8, 15-36
 再開 14-10
 ストリーム・モード制御 15-21
 逐次化 5-90
 同期制御 15-22
 フォーマット 2 IDAW 制御 15-23
 変更制御 15-21
 2K IDAW 制御 15-23
 チャンネル・プログラム実行の延期 15-36
 中間割り込み条件 16-17
 DCTI への影響 15-38
 チャンネル・プログラム・アドレス 16-18
 IPL に使用される 17-14
 IRB 内のフィールド妥当性フラグ 16-33
 中央演算処理装置
 参照: CPU
 中間状況ビット (入出力) 16-16
 中間割り込み条件 (入出力) 16-4
 中止
 操作単位の 5-21
 緊急マシン・チェック条件の場合 11-11
 命令実行の 5-20
 緊急マシン・チェック条件の場合 11-11
 抽出権限制御ビット 5-24
 長精度 16 進浮動小数点数 18-3
 長精度 2 進浮動小数点数 19-4
 直接アクセス記憶機構 3-2
 直列入出力インターフェース 13-2
 参照される資料 xxi
 直列入出力チャンネル間アダプター
 参照される資料 xxi
 追加浮動小数点 (AFP) レジスター 9-2
 次エントリー・サイズ (リンケージ・スタック内の) 5-68
 データ
 間接アドレッシング (入出力) 13-7, 15-34
 入出力操作での事前取り出し 15-28
 フォーマット
 一般命令 7-2
 10 進数命令 8-1
 16 進浮動小数点命令 18-3
 2 進浮動小数点命令 19-4

- データ (続き)
 - ブロック化 (入出力) 15-20
- データ例外 6-20
 - プログラム割り込みの優先順位 6-14
 - AFP レジスター 6-20
 - BFP 命令 6-20
 - IEEE 例外条件 6-20, 19-11
 - 10 進オペランド 6-20, 8-4
- データ例外コード
 - 参照: DXC
- データ例外コード (DXC) 6-14
- データ・アドレス (入出力) 15-27
 - 無効 16-24
 - 無効な指定 16-24
- データ・ストリーミング (入出力) 13-3
 - CCW カウントの影響 15-32
- データ・チェック
 - 測定ブロック 16-32
- テーブル
 - エントリー
 - 参照: エントリー・テーブル
 - 権限
 - 参照: 権限テーブル
 - セグメント
 - 参照: セグメント・テーブル
 - トレース 4-10
 - ページ
 - 参照: ページ・テーブル
 - 変換 3-30
 - 領域
 - 参照: 領域テーブル
 - リンケージ
 - 参照: リンケージ・テーブル
- ASN
 - 参照: ASN 第 1 テーブル、ASN 第 2 テーブル
- DAT
 - 参照: ページ・テーブル、セグメント・テーブル、領域
テーブル
- テーブル・タイプ・ビット 3-31, 3-32
 - セグメント・テーブル・エントリー内の 3-32
 - 領域テーブル・エントリー内の 3-31
- 低アドレス保護 3-11
 - 制御ビット 3-11
 - 例外 6-27
- 停止
 - キー 12-4
 - 機能 4-2
 - プロセッサ信号指令 4-48
- 停止 (プロセッサ信号状況) 4-53
- 停止機能 15-14
 - 完了 15-15
 - 信号発信 15-15
 - バス管理 15-14
- 停止機能 (続き)
 - 保留 16-14
 - HALT SUBCHANNEL により開始される 14-6
 - SCSW 内のビット 16-12
- 停止状態
 - CPU の 4-1
 - 格納操作の完了への影響 5-84
 - TOD クロックの 4-34
- 停止信号 17-8
 - 停止機能の一環として発信される 15-15
- 停止/状況記憶 (プロセッサ信号指令) 4-48
- ディスパッチ可能単位 (DU) 5-37
 - アクセス・リスト指定 (DUALD) 5-44
 - 制御テーブル (DUCT) 5-44
 - 起点 (DUCTO) 5-42
 - サブスペース・グループ機能がインストールされている
とき 5-55
 - トラップ機能がインストールされている場合 10-114
- 適応辞書 7-64
- テスト標識ライト 12-5
- デフォルトの QNaN 19-7
- 展開辞書
 - 圧縮時の位置 7-59
- 展開操作ビット 7-58
- 展開プロセス
 - フローチャート 7-67
- 電源制御 12-4
- 同期
 - CPU タイマーと TOD クロック 4-39
 - TOD クロックの 4-34, 4-38
- 同期化
 - チェックポイント 11-3
- 同期制御 15-22
- 同期マシン・チェック割り込み条件 11-18
- 凍結 7-64
- 動的アドレス変換 (DAT) 3-25
 - 制御 3-27
 - テーブル取り出しの順序 5-82
 - 明示および暗黙 3-33
 - LOAD REAL ADDRESS 命令による 10-39
 - PSW のモード・ビット 4-5
 - アドレス変換での使用 3-27
 - STORE REAL ADDRESS 命令による 10-94
- 動的再接続機能 13-2
- 特殊 QNaN 19-7
- 特殊操作例外 6-30
- 特権命令 4-6
 - 制御 10-1
 - 入出力 14-1
- 特権命令例外 6-26

トラップ制御ブロック 10-114
トラップ保管域 10-115
取り出し
 記憶域オペランドの 5-84
 記憶キー内の無効 CBC の取り扱い 11-8
 命令の 5-81
 割り込み次の PSW の 5-89
 ART テーブル・エントリーおよび DAT テーブル・エントリーの 5-82
取り出し参照 5-84
 アクセス例外 6-37
取り出し専用ビット 5-46
取り出し保護 3-8
 オーバーライド制御ビット 3-10
 記憶キー内のビット 3-8
トレース 4-10
 エントリー 4-13
 エントリー・アドレス 4-13
 トレース例外 6-34
 例外 6-44
トレーラー・エントリー 5-69

〔ナ行〕

長い入出力ブロック 16-23
長さ
 可変 (記憶域オペランド) 5-7
 第 1 オペランドと同じ第 2 オペランド 5-6
 フィールド 3-2
 命令 5-5
 レジスター・オペランド 5-6
BFP データの
 減少 19-41
 増加 19-40
HFP データの
 減少 18-16
 増加 18-14
入出力アドレッシング 13-4
入出力インターフェース
 参照されている ESCON 資料 xxi
 参照されている OEMI 資料 xxi
入出力インターフェースに関する OEMI (相手先商標製造会社情報) xxi
 参照される資料 xxi
入出力エラー警報 (サブチャネル・ログアウト内の) 16-34
入出力機能の完了
 開始時 15-40
 開始と再開 15-40
 消去機能による 15-14
 即時コマンドの場合 15-40
 チャネル・パス・リセット機能による 15-43
 データ転送時の 15-41

入出力機能の完了 (続き)
 停止機能による 15-15
入出力コマンド
 参照される資料 xxi
入出力システムの再構成 17-16
入出力システム・リセット 17-10
 サブシステム・リセットの一環としての 4-44
入出力操作 13-5
 実行 15-19
 初期表示 16-11
 即時 15-40
入出力操作の終結 13-7, 16-1
 開始時 15-40
 開始前の 15-40
 即時コマンドの場合 15-40
 データ転送時の 15-41
入出力装置の接続 13-2
入出力データの自己規定ブロック 15-32
入出力データのブロック 15-20
入出力取り消し機能 1-8
入出力命令 14-1, 14-2
 オペランド・アクセス 14-1
 据え置き条件コード 16-8
 入出力操作における役割 13-5
入出力割り込み 6-12, 16-1
 アクション 16-5
 マスキング 13-8
 優先順位 16-4
入出力割り込みコード 6-13, 14-18
 入出力割り込み識別ワード 14-18
 TPI により格納される 14-18
入出力割り込みサブクラス 13-8
入出力割り込みサブクラス・マスク 6-13, 16-5
 優先順位との関係 16-4
入出力割り込み識別ワード 14-18
 割り当て済み記憶位置 3-51
入出力割り込み条件 13-8, 16-2
 警報 16-4
 請求 16-3
 中間 16-4
 非請求 16-2
 1 次 13-7, 16-4
 2 次 13-7, 16-4
入出力割り込みパラメーター
 入出力割り込みコード内の 14-18
 割り当て済み記憶位置 3-51
 IPL に使用される 17-14
 ORB 内の 15-20
 PMCW 内の 15-2
入出力割り込み要求
 サブチャネルからの 16-5

入出力割り込み要求 (続き)
消去 13-9
ノーオペレーション
命令 (BRANCH ON CONDITION) 7-23
命令 (BRANCH RELATIVE ON CONDITION) 7-27
ノード (ツリー構造の) 7-157

〔八行〕

バージョン・コード 10-92
バースト・モード (チャンネル・バス操作) 13-3
ハーフワード 3-3
ハーフワード単位の記憶域参照 5-87
バイト 3-2
記憶機構内での番号付け 3-2
バイト指標 (BX) 3-26
バイト多重モード (チャンネル・バス操作) 13-3
破壊的オーバーラップ 5-88, 7-92, 7-96, 7-99
アクセス・レジスタ・モードでの 5-80
バス管理 13-6
開始機能と再開機能での 15-17
消去機能での 15-13
停止機能での 15-14
バス非稼働条件 15-4
パターン (EDIT での) 8-7
バック 10 進数 8-1
ゾーン・フォーマットとの間の変換 7-109
ゾーン・フォーマットの変換 7-155
例 A-4
バックアップ
処理 (同期マシン・チェック条件) 11-18
バックアップ・ビット (マシン・チェック条件) 11-18
バッファ記憶機構 (キャッシュ) 3-2
パラメーター
外部割り込み 6-10
割り当て済み記憶位置 3-48
変換 3-27
SIGNAL PROCESSOR 用のレジスタ 4-49, 10-90
パリティ・ビット 11-2
パワーオン・リセット 4-45
番号
装置 13-5
マシン・タイプ 10-92
CPU モデル 10-92
番号付け
アドレス (バイト位置) 3-2
ビットの 3-2
汎用レジスタ 2-3
妥当性ビット 11-21
保管域 3-53

非インターロック更新記憶域参照 5-85
比較
アドレス
参照: アドレス比較
符号付き 2 進数 7-5
論理 7-5
例 A-15
BFP データの 19-9
TOD クロック 4-38
10 進数 8-6
例 A-34
16 進浮動小数点
例 A-40
非稼働
チャンネル・バス状態としての 16-12
CPU の状態 4-51
TOD クロックの状態 4-35
非数値 (NaN) 19-7
非数値エンティティ
2 進 19-7
非正規化数 19-8
非正規化浮動小数点数 18-3
HFP データのみ 9-1
非請求割り込み条件 (入出力) 16-2
非設定 (TOD クロックの状態) 4-34
左から右へのアドレッシング 3-2
ビッグ・エンディアン 7-89
ビット 3-2
バイトのグループ内での番号付け 3-2
非特権命令 4-6, 7-2
表示 (手動制御) 12-2
標識
手動 12-3
チェック停止 12-3
テスト 12-5
待ち 12-6
モード 12-2
ロード 12-3
標準エポック (TOD クロック) 4-36
フィールド 3-2
フィールド妥当性フラグ (サブチャンネル・ログアウトでの)
チャンネル制御チェックとの関係 16-27
フィールド妥当性フラグ (サブチャンネル・ログアウト内
の) 16-33
フィールド分離文字 (EDIT での) 8-7
フォーマット
アドレス 3-2
一般データ 7-2
エラー
DAT テーブル・エントリ内の 6-34
PSW 内の 6-9
情報 3-2

フォーマット (続き)
 命令 5-3
 PSW 4-5
 10 進数データ 8-1
 16 進浮動小数点データ 18-3
 2 進浮動小数点データ 19-4
 フォーマット 0 とフォーマット 1 の CCW 15-25
 フォーマット 1 兄弟記述子ビット 7-59
 フォーマット 2 IDAW 制御 15-23
 フォーマット制御 15-22
 不揮発性記憶機構 3-2
 複数アクセス記憶域参照 5-86
 符号コード (10 進数) 8-2
 符号付き 2 進数
 演算 7-4
 整数 7-3
 例 A-2
 比較 7-5
 符号なし 2 進数
 演算 7-4
 整数 7-3
 アドレス生成における 5-8
 例 A-3
 符号ビット
 浮動小数点 18-1
 2 進数 7-3
 浮動小数点
 数 19-4
 シフト
 参照: 正規化
 データ
 フォーマットの延長 18-14, 19-40
 フォーマットの短縮 18-16, 19-41
 データ・クラス 19-6
 変換
 フォーマット間の 19-7
 命令 9-1
 レジスター 2-3, 9-2
 消去 9-11
 妥当性ビット 11-21
 保管域 3-53
 16 進数 (HFP) 9-1
 16 進データのフォーマット 18-3
 2 進 (BFP) 9-1
 2 進整数からの変換 18-10, 19-27
 2 進整数への変換 18-10, 19-28
 2 進データのフォーマット 19-4
 参照: 指数
 参照: 2 進数
 浮動小数点制御 (FPC) レジスター 19-2
 妥当性ビット 11-21
 浮動小数点制御レジスター
 保管域 3-53
 浮動割り込み条件 6-6, 11-23
 消去 4-44
 負の 0
 10 進
 例 A-5
 10 進数 8-2
 2 進数 7-3
 フラグ
 浮動小数点演算例外の 19-3
 BFP 演算例外の 19-3
 IEEE 例外条件の 19-3
 ブランチ
 相対 5-9
 チャンネル・プログラム内の 15-38
 判断処理、ループ制御、およびサブルーチン・リンケージを
 行うための 5-10
 ブランチ・アドレス生成 5-9
 リンケージ・スタックの使用 5-62
 ブランチ状態エンタリー 5-69, 10-9
 ブランチ成功 PER イベント 4-30
 マスク 4-24
 ブランチ・アドレス 5-9
 制御ビット 4-24
 トレース・エンタリー内の 4-20
 リンケージ・スタック状態エンタリー内の 5-71
 ブランチ・トレース制御ビット 4-13
 フリー・プールの操作
 プログラミング例 A-49
 フルワード
 参照: ワード
 フレーム多重モード (チャンネル・パス操作) 13-3
 プレフィックス 3-14
 状況記憶保管域 3-53
 プロセッサ信号指令により設定される 4-49
 プレフィックス域 3-14
 プレフィックス設定 (プロセッサ信号指令) 4-49
 ブロードキャスト・ページ機能 1-8
 プログラム 5-37
 実行 5-2
 初期ローディング 4-46, 17-13
 変更可能な SCHIB のフィールド 15-7
 マスク (PSW 内の) 4-6
 例外 6-13
 割り込み 6-13
 優先順位 6-14, 6-37
 プログラム状況ワード
 参照: PSW
 プログラム制御の転送 5-60

- プログラム呼び出し状態エントリー 5-69, 10-58
- プログラム・イベント
 - 参照：PER イベント
- プログラム・イベント記録
 - 参照：PER
- プログラム・チェック
 - サブチャンネル状況としての 16-23
- 測定ブロック 16-32
- プログラム・マスク
 - 妥当性ビット 11-21
- プロセッサ
 - 参照：CPU
- プロセッサ可用性機能 1-12
- ブロック単位の記憶域参照 5-87
- ブロック番号
 - 拡張記憶機構 2-2
- 分解能
 - クロック・コンパレーターの 4-38
 - CPU タイマーの 4-39
 - TOD クロックの 4-34
- 分割可能な命令実行 5-78
- 分類命令
 - 参照：COMPARE AND FORM CODEWORD 命令、UPDATE TREE 命令
- ページ 3-26
- ページ指標 (PX) 3-26
- ページ変換例外 6-25
 - アクセス例外としての 6-35, 6-40
- ページ保護 3-11
 - セグメント・テーブル・エントリー内のビット 3-31
 - ページ・テーブル・エントリー内のビット 3-32
 - 例外 6-27
- ページ無効ビット (ページ・テーブル・エントリー内の) 3-32
- ページ・スワッピング 3-25
- ページ・テーブル 3-32
 - 起点 (PTO) 3-31
 - 検索 3-40
 - 指定 3-31
- ページ・フレーム実アドレス (PFRA) 3-32
- 並行センス
 - 標識 (ERW 内の) 16-35
 - ECW 内の 16-39
- 並行センス機能 17-17
- 並行センス・カウント (ERW 内の) 16-36
- 並列入出力インターフェース 13-3
 - 参照されている OEMI 資料 xxi
- 並列入出力チャンネル間アダプター
 - 参照される資料 xxi
- ヘッダー・エントリー 5-68
- 変位 (相対アドレッシングにおける) 5-8
- 変換
 - アドレス 3-25
 - 参照：動的アドレス変換
 - 索引バッファ
 - 参照：TLB
 - 指定例外 6-34
 - テーブル 3-30
 - 浮動小数点フォーマット
 - 浮動小数点フォーマットの 19-7
 - 例外識別 3-50
 - HFP データと BFP データの間の 9-8
 - PC 番号 5-28
 - 10 進数から 2 進数への 7-68
 - 例 A-18
 - 16 進小数部と 10 進小数部の E-1
 - 16 進浮動小数点数
 - 基本的な例 A-6
 - 命令による変換の例 A-42
 - 2 進数から 10 進数への 7-68
 - 例 A-19
 - 変換バス 3-42
 - 変換パラメーター 3-27
 - 変換モード 3-27
 - 変更
 - 記憶域 (PER イベント) 4-31
 - 変更可能域 (リンケージ・スタック状態エントリー内の) 5-72
 - 変更記録 3-13
 - 変更制御 15-21
 - 変更/表示制御 12-2
 - 編集命令 8-3
 - 参照：ED 命令、EDMK 命令
 - ホーム仮想アドレス 3-5
 - 有効アドレス・スペース制御エレメント 3-33
 - ホーム記憶機構変更イベント制御ビット 3-30
 - ホーム実スペース制御ビット 3-30
 - ホーム実スペース・トークン起点 (HRSTKO) 3-30
 - ホーム指定タイプ制御ビット 3-30
 - ホーム専用スペース制御ビット 3-30
 - ホーム領域テーブル
 - 起点 (HRTO) 3-30
 - 指定 (HRTD) 3-29
 - 長さ (HRTL) 3-30
 - ホーム・アドレス・スペース 3-15, 5-33
 - 機能 5-33
 - ホーム・スペース切り替えイベント制御ビット 3-30
 - ホーム・スペース・モード 3-27
 - ホーム・セグメント・テーブル
 - 起点 (HSTO) 3-30
 - 指定 (HSTD) 3-29
 - 長さ (HSTL) 3-30

報告ソース ID (RSID) 17-21
 報告ソース・コード (RSC) 17-19
 保護 (記憶域)
 アクセス・リスト制御
 参照: アクセス・リスト制御保護
 キー制御
 参照: キー制御保護
 低アドレス
 参照: 低アドレス保護
 取り出し
 参照: 取り出し保護
 トレース中の 4-23
 ページ
 参照: ページ保護
 保護 (記憶機構) 3-8
 保護桁 18-4
 保護時の抑止 3-12
 補助記憶機構 3-2, 3-25
 補助報告ビット
 サブチャンネル・ログアウト内の 16-33
 チャンネル報告ワード内の 17-20
 マシン・チェック割り込みコード内の 11-18
 保留チャンネル報告 (入出力システム・リセットの影響) 17-12

〔マ行〕

マシン・タイプ番号 (CPU ID の中の) 10-92
 マシン・チェック 11-2
 入出力の一部として検出された誤動作の取り扱い 11-5
 マスク
 PSW 内の 4-6
 ログアウト 11-24
 割り込み 6-13, 11-11
 アクション 11-12
 コード (MCIC) 3-52, 11-14
 制御レジスター内のサブクラス・マスク 11-23
 浮動条件 11-23
 マシン・チェック割り込み 11-23
 PSW 内のマスク 4-6
 参照: 誤動作
 マシン・チェック・アーキテクチャー・モード識別 3-50
 マスク 6-6
 サブクラス
 参照: サブクラス・マスク・ビット
 バス管理 15-2, 15-23
 プログラム割り込み 6-13
 モニター 6-23
 BFP 演算例外の 19-3
 BRANCH ON CONDITION 命令の中での 7-23
 BRANCH RELATIVE ON CONDITION 命令の中での
 7-27
 COMPARE LOGICAL CHARACTERS UNDER MASK
 命令の中での 7-43

マスク (続き)
 IEEE 例外条件の 19-3
 INSERT CHARACTERS UNDER MASK 命令の中での
 7-79
 PER イベント 4-24
 PSW 内の 4-5
 STORE CHARACTERS UNDER MASK 命令の中での
 7-136
 TEST UNDER MASK 命令の中での 7-145
 参照: 入出力割り込み、割り込み
 待ち状態ビット
 PSW 内の 4-6
 待ち状態ライト 12-6
 マルチバス・モード 15-3
 入る 15-19
 マルチプログラミングの例 A-43
 マルチプロセッシング 4-47
 手動操作 12-6
 タイミング機構割り込み 4-37
 プログラミング上の考慮事項 8-3, A-43
 プログラミング例 A-43
 TOD クロック 4-33
 丸め (浮動小数点)
 BFP の結果の 19-8
 HFP の結果 18-16
 丸め (10 進数) 8-11
 例 A-38
 丸めアクション
 要約 9-3
 実アドレス 3-4
 実アドレスを使用した格納 (PER イベント) 4-31
 実アドレスを使用した格納イベント・マスク・ビット 4-24
 短い入出力ブロック 16-23
 実スペース
 トークン起点 (RSTKO) 3-27
 実スペース指定 (RSD)
 ホーム 3-29
 1 次 3-27
 2 次 3-29
 実スペース制御ビット 3-30
 ホーム 3-30
 1 次 3-28
 2 次 3-29
 無限大 19-7
 無効
 アクセス・リスト・エントリー 5-46
 アドレス 6-15
 指令 (プロセッサ信号状況) 4-54
 セグメント・テーブル・エントリー内のビット 3-32
 パラメーター (プロセッサ信号状況) 4-53
 ページ・テーブル・エントリー内のビット 3-32
 変換アドレス 3-41

無効 (続き)

変換フォーマット

例外認識 3-41

命令コード 6-24

領域テーブル・エントリー内のビット 3-31

リンケージ・テーブル・エントリー内のビット 5-28

ASN 第 1 テーブル・エントリー内のビット 3-18

ASN 第 2 テーブル・エントリー内のビット 3-19

CBC 11-2

記憶キー内の 11-7

記憶機構内の 11-6

レジスター内の 11-9

無効化

緊急マシン・チェック条件の場合 11-11

操作単位の 5-21

命令実行の 5-20

例外的状況 5-22

無効な CCW フィールド

延期フラグ 16-25

カウント 16-24

コマンド・コード 16-24

データ・アドレス 16-24

無効なアドレス指定

チャンネル・プログラム・アドレス内の 16-23

CCW 内でのデータの 16-24

IDAW 内の 16-25

IDAW の 16-24

TIC CCW の 16-23

無効フォーマット

CCW の 16-25

ORB の 16-25

明示的地址変換 3-33

明示トレース制御ビット 4-13

命令

一覧表 B-1

一般 7-2

例 A-7

クラス 2-2

事前取り出し 5-81

実行 5-10

実行の順序 5-2

終了 5-19

準特権 4-6, 10-1

使用例 A-6

ステップ (速度制御) 12-4

CPU 状態への影響 4-2

CPU タイマーへの影響 4-39

制御 10-1

損傷 11-15, 11-19

特権 4-6

制御用 10-1

取り出し 5-81

アクセス例外 6-37

命令 (続き)

取り出し (続き)

PER イベント 4-31

PER イベント・マスク 4-24

長さ 5-5

入出力

参照: 入出力命令

バックアップ 11-18

非特権 4-6, 7-2

フォーマット 5-3

浮動小数点 9-1

分割可能な実行 5-78

割り込み可能

参照: 割り込み可能命令

EXECUTE 命令による変更 7-78

10 進数 8-1

例 A-34

16 進浮動小数点

例 A-39

参照: 付録 B の命令リストおよびページ番号

命令アドレス

アドレスのタイプ 3-5

エントリー・テーブル・エントリー内の 5-30

妥当性ビット 11-21

DAT による取り扱い 3-27

PSW 内の 4-6

命令コード

参照: 命令コード

命令コード (op コード) 5-2

無効 6-24

命令実行の完了 5-19

命令実行の終了 5-19

命令実行の断片的ステップ 5-78

命令実行の部分完了 5-20

命令処理損傷 11-15

処理損傷をもたらす 11-19

処理バックアップをもたらす 11-18

命令長コード

参照: ILC

命令の B フィールド 5-8

命令の D フィールド 5-8

命令の I フィールド 5-6

命令の L フィールド 5-7

命令の R フィールド 5-6

命令の X フィールド 5-8

メッセージ・バイト (EDIT での) 8-7

モード

アーキテクチャー

参照: アーキテクチャー・モード

アクセス・レジスター 3-27

アドレッシング

参照: アドレッシング・モード

誤長表示 15-23

モード (続き)

- 誤長抑止 15-23
- 準特権命令の場合の要件 5-24
- シングル・パス 15-3, 15-19
- バースト (チャンネル・パス操作) 13-3
- バイト多重 (チャンネル・パス操作) 13-3
- 標識
 - アーキテクチャー 12-2
- フレーム多重 (チャンネル・パス操作) 13-3
- 変換 3-27
- ホーム・スペース 3-27
- 丸め 19-8
- 実 3-27
- 1 次スペース 3-27
- 2 次スペース 3-27
- モード・トレース制御ビット 4-13
- 文字
 - 8 ビット・コードで表される xxi
- モデル番号 (CPU ID の中の) 10-92
- モニター
 - チャンネル・サブシステム 17-1
 - MONITOR CALL による 6-23, 7-90
 - PER イベント用
 - 参照: PER
- モニター・イベント 6-23
- モニター・クラス番号 6-23
 - 割り当て済み記憶位置 3-49
- モニター・コード 6-23
 - 割り当て済み記憶位置 3-51
- モニター・マスク 6-23
- 問題プログラム状態 4-6
 - エントリー・テーブル・エントリー内のビット 5-30
 - 互換性 1-15
 - PSW のビット 4-6

〔ヤ行〕

- 有効 CBC 11-2
- 有効アクセス・リスト指定 5-44
- 有効アドレス 3-5
 - アドレッシング・モードにより制御される 5-7
 - 記憶域インターロックに使用される 5-79
 - 生成 5-7
- 有効アドレス・スペース制御エレメント 3-33
- 有効アドレス・スペース制御エレメント (ASCE)
 - 有効 3-33
- 有効化 11-5
 - 記憶キーの 11-7
 - 記憶機構 11-6
 - レジスターの 11-9
 - TOD クロックの 11-9

有効数字

- 開始文字 (EDIT での) 8-7
- 消失 18-1
 - HFP 加算での 18-8
 - マスク (PSW 内の) 4-6
- 有効な ART テーブル・エントリー 5-53
- 有効な、領域テーブル、セグメント・テーブル、またはページ・テーブル・エントリー 3-42
- 有効な浮動小数点レジスター番号 9-2
- 優先順位
 - アクセス例外の 6-40
 - 外部割り込み条件の
 - 外部割り込み条件の 6-10
 - サブスペース置き換え例外の 6-43
 - データ例外の 6-14
 - トレース例外の 6-44
 - 入出力割り込みの 16-4
 - プログラム割り込み条件の 6-37
 - 演算例外の 6-14
 - 割り込みの (CPU) 6-44
 - ASN 変換例外の 6-43
 - PER イベントの 4-28
- 優先符号コード 8-2
- 要パス検査
 - 標識 (ERW 内の) 16-35
- 抑止
 - 操作単位の 5-21
 - 命令実行の 5-20
 - 例外的状況 5-22
- 抑制可能マシン・チェック条件
- 抑制マシン・チェック条件 11-11
- 呼び出し先スペース ID 5-72

〔ラ行〕

- ライト
 - 参照: 表示ライト
- リカバリー
 - システム 11-16
 - サブクラス・マスク・ビット 11-24
 - チャンネル・サブシステム 17-17
 - マシン・チェック条件のクラス 11-11
- リセット 4-40, 17-9
 - 機能の要約 4-42
 - クリア 4-45
 - サブシステム 4-44
 - システム・リセット消去キー 12-5
 - システム・リセット通常キー 12-5
 - 手動開始により実行される機能の要約 4-41
 - 初期 CPU 4-44
 - チャンネル・パス 17-9
 - 入出力システム 17-10
 - サブシステム・リセットの一環としての 4-44

- リセット (続き)
 - パワーオン 4-45
 - CPU 4-43
 - CPU 状態への影響 4-2
 - TOD クロックへの影響 4-34
- リセット信号 (入出力) 17-9
 - チャンネル・パスリセットでの 17-9
 - 入出力システム・リセットでの 17-10, 17-11
 - RCHP の一環として発信される 15-42
- リトル・エンディアン 7-89
- リモート操作ステーション 12-1
- 領域 3-26
- 領域指標 (RX)
 - 仮想アドレス内の 3-26
- 領域第 1 指標 (RFX) 3-26
- 領域第 1 変換例外 6-27
 - アクセス例外としての 6-40
- 領域第 2 指標 (RSX) 3-26
- 領域第 2 変換例外 6-28
 - アクセス例外としての 6-40
- 領域第 3 指標 (RTX) 3-26
- 領域第 3 変換例外 6-28
 - アクセス例外としての 6-40
- 領域テーブル
 - 起点 (RTO) 3-27
- 領域テーブル指定 (RTD) 3-27
 - ホーム 3-29
 - 1 次 3-27
 - 2 次 3-29
- 領域テーブル・エントリー (RTE) 3-30
- 領域変換例外 6-27
- 領域無効ビット (領域テーブル・エントリー内の) 3-31
- リンク情報
 - BRANCH AND LINK 命令の場合 7-20
 - BRANCH AND SAVE 命令の場合 7-20
 - BRANCH AND SAVE AND SET MODE 命令の場合 7-21
- リンケージ指標 (LX) 5-28
- リンケージ・スタック 5-59, 5-67
 - エントリー記述子 5-67
 - エントリー・アドレス 5-67
 - エントリー・タイプ・コード 5-67
 - 概要 5-65
 - 関連の PER イベント 5-64
 - 関連のトレース・エントリー 5-63
 - 残余フリー・スペース 5-68
 - 状態エントリー 5-69
 - 情報の取り扱い 5-63
 - セクション 5-65
 - ID 5-68
 - 操作 5-65
 - 制御 5-67
- リンケージ・スタック (続き)
 - 次エントリー・サイズ 5-68
 - トレーラー・エントリー 5-69
 - ブランチ状態エントリー 10-9
 - プログラム呼び出し状態エントリー 10-58
 - ヘッダー・エントリー 5-68
 - 命令 5-60
- リンケージ・スタックの機能 5-60
- リンケージ・テーブル (LT) 5-28
 - 起点 (LTO) 5-28
 - 指定 (LTD) 5-28
 - 長さ (LTL) 5-28
- ループ制御 5-10
- 例外 6-13
 - アクセス (集合プログラム割り込み名) 6-35, 6-40
 - アドレッシング 6-15
 - 暗号操作 6-20
 - オペラント (入出力命令の) 6-24
 - 拡張権限 6-21
 - 関連の
 - アンスタッキング・プロセス 5-77
 - スタッキング・プロセス 5-75
 - ART 5-53
 - 記憶保護 6-27
 - 固定小数点オーバーフロー 6-22
 - 固定小数点除算 6-22
 - サブスペース置き換え (集合プログラム割り込み名) 6-43
 - 実行 6-21
 - 指定 6-31
 - スタック一杯 6-33
 - スタック空 6-33
 - スタック指定 6-33
 - スタック操作 6-33
 - スタック・タイプ 6-33
 - セグメント変換 6-29
 - 操作 6-24
 - 遅延アクセス (マシン・チェック条件) 11-18
 - データ 6-20
 - 特殊操作 6-30
 - 特権命令 6-26
 - トレース (集合プログラム割り込み名) 6-44
 - トレース・テーブル 6-34
 - 認識
 - 早期および遅延 6-9
 - ページ変換 6-25
 - 変換指定 6-34
 - 変換時の 3-41
 - 領域第 1 変換 6-27
 - 領域第 2 変換 6-28
 - 領域第 3 変換 6-28
 - 領域変換 6-27

例外 (続き)

AFX 変換 6-18
ALE シーケンス 6-18
ALEN 変換 6-18
ALET 指定 6-18
ASCE タイプ 6-18
ASN 変換 (集合プログラム割り込み名) 6-43
ASTE シーケンス 6-19
ASTE 妥当性 6-19
ASX 変換 6-20
EX 変換 6-21
HFP 指数アンダーフロー 6-22
HFP 指数オーバーフロー 6-22
HFP 除算 6-22
HFP 平方根 6-23
HFP 有効数字 6-23
IEEE 19-11
LX 変換 6-23
PC 変換指定 6-25
PSW 関連 6-9
1 次権限 6-26
1 進オーバーフロー 6-21
10 進除算 6-21
2 次権限 6-29
例外アクセス識別 3-49
レジスター
アクセス 2-4
基底アドレス 2-3
指示 5-6
指標 2-3
制御 2-4
妥当性ビット 11-21
汎用 2-3
浮動小数点 2-3, 9-2
浮動小数点制御 19-2
プレフィックス 3-14
保管域 3-53, 11-22
有効化 11-9
レジスターの保管域 3-53, 11-22
連結子
参照: 論理連結子
連結された ART テーブル・エントリー 5-53
連結されている領域テーブル、セグメント・テーブル、または
ページ・テーブル・エントリー 3-
ローディング、初期
参照: IML、IPL
ロード、初期
参照: IML、IPL
ロード消去キー 12-3
ロード状態 4-1, 4-2
IPL 時の 4-46

ロード装置アドレス制御 12-3
ロード通常キー 12-3
ロード標識ライト 12-3
ログアウト
固定
マシン・チェック 11-24
割り当て済み記憶位置 3-53
サブチャンネル (入出力) 16-31
ロック A-46
FIFO キューイングを使用した例 A-48
LIFO キューイングを使用した例 A-47
論理
演算 (符号なし 2 進数) 7-4
データ 7-2
比較 7-5
連結子
AND 7-18, 7-19
EXCLUSIVE OR 7-77
OR 7-108, 7-109
論理アドレス 3-5
DAT による取り扱い 3-27
論理ストリング補助 1-8
論理分割 (LPAR) モード 1-12, 1-14

〔ワ行〕

ワード 3-3
ワード単位の記憶域参照 5-87
割り当て済みの記憶位置 3-49
割り込み 6-2
アクション 6-2
入出力 16-5
マシン・チェック 11-12
外部
参照: 外部割り込み
監視プログラム呼び出し 6-44
クラス 6-5
再始動 6-44
ストリング
参照: 割り込みのストリング
プログラム
参照: プログラム割り込み
保留 6-6
保留中
外部 6-10
マシン・チェック 11-13
CPU 状態との関係 4-2
マシン・チェック
参照: マシン・チェック割り込み
マスキング 6-6
命令シーケンスへの影響 5-19
優先順位
参照: 優先順位

割り込み (続き)
 参照: マスク
 割り込み可能命令 5-20
 終了に影響を与える PER イベント 4-29
 停止 4-2
 COMPARE AND FORM CODEWORD 7-34
 COMPARE LOGICAL LONG 7-45
 COMPARE UNTIL SUBSTRING EQUAL 7-56
 MOVE LONG 7-93
 TEST BLOCK 10-109
 UPDATE TREE 7-158
 割り込みキー 12-3
 外部割り込み 6-12
 割り込みコード 6-5
 外部 6-10
 監視プログラム呼び出し 6-44
 プログラム 6-13
 マシン・チェック (MCIC) 3-52, 11-14
 要約 6-2
 割り込み条件 6-2
 消去 4-43
 浮動 6-6, 11-23
 割り込み点 5-20
 マシン・チェックの 11-13
 割り込みに対する使用可能 6-6
 割り込みに対する使用可能化
 サブチャンネル 16-5
 割り込みの条件
 参照: 割り込み条件
 割り込みのストリング 4-3, 6-45
 クロック・コンパレーターにより生じる 4-39
 CPU タイマーにより生じる 4-40
 割り込みのソース
 割り込みコードにより識別される 6-5
 割り込みのループ
 参照: 割り込みのストリング
 割り込みパラメーター
 外部 (割り当て済み記憶位置) 3-48
 割り込み保留
 参照: 割り込み保留

A

A (ADD) 2 進数命令 7-16
 AD (ADD NORMALIZED) HFP 命令 18-7
 例 A-39
 ADB (ADD) BFP 命令 19-19
 ADBR (ADD) BFP 命令 19-19
 ADD BFP 命令 19-19
 ADD DECIMAL 命令 8-5
 例 A-34

ADD HALFWORD 命令 7-16
 例 A-7
 ADD HALFWORD IMMEDIATE 命令 7-16
 ADD LOGICAL 命令 7-17
 ADD LOGICAL WITH CARRY 命令 7-17
 ADD NORMALIZED HFP 命令 18-7
 例 A-39
 ADD UNNORMALIZED HFP 命令 18-8
 例 A-39
 ADD 2 進数命令 7-16
 ADR (ADD NORMALIZED) HFP 命令 18-7
 AE (ADD NORMALIZED) HFP 命令 18-7
 例 A-39
 AEB (ADD) BFP 命令 19-19
 AEBR (ADD) BFP 命令 19-19
 AER (ADD NORMALIZED) HFP 命令 18-7
 AFP レジスター・データ例外 6-20
 AFP (追加浮動小数点) レジスター 9-2
 AFT (ASN 第 1 テーブル) 3-18
 AFTE (ASN 第 1 テーブル・エントリー) 3-18
 AFTO (ASN 第 1 テーブル起点) 3-18
 AFX (ASN 第 1 テーブル指標) 3-17
 変換例外 6-18
 無効ビット 3-18
 AG (ADD) 2 進数命令 7-16
 AGF (ADD) 2 進数命令 7-16
 AGFR (ADD) 2 進数命令 7-16
 AGHI (ADD HALFWORD IMMEDIATE) 命令 7-16
 AGR (ADD) 2 進数命令 7-16
 AH (ADD HALFWORD) 命令 7-16
 例 A-7
 AHI (ADD HALFWORD IMMEDIATE) 命令 7-16
 AKM (許可キー・マスク) 5-30
 AL (ADD LOGICAL) 命令 7-17
 ALB (ART 索引バッファ) 5-53
 エントリー
 使用可能状態 5-54
 消去 5-55
 変換変更の影響 5-54
 ALC (ADD LOGICAL WITH CARRY) 命令 7-18
 ALCG (ADD LOGICAL WITH CARRY) 命令 7-18
 ALCGR (ADD LOGICAL WITH CARRY) 命令 7-17
 ALCR (ADD LOGICAL WITH CARRY) 命令 7-17
 ALD (アクセス・リスト指定) 5-45
 ALE
 参照: アクセス・リスト・エントリー
 ALEAX (アクセス・リスト・エントリー許可指標) 5-46
 ALEN (アクセス・リスト・エントリー番号) 5-43
 変換例外 6-18
 アクセス例外としての 6-35
 無効ビット 5-46

ALESN (アクセス・リスト・エントリー・シーケンス番号)
ALE 内の 5-46
ALET の中の 5-43
ALET (アクセス・リスト・エントリー・トークン) 5-37,
5-43
指定例外 6-18
アクセス例外としての 6-35
ALG (ADD LOGICAL) 命令 7-17
ALGF (ADD LOGICAL) 命令 7-17
ALGFR (ADD LOGICAL) 命令 7-17
ALGR (ADD LOGICAL) 命令 7-17
ALL (アクセス・リスト長) 5-45
ALO (アクセス・リスト起点) 5-45
ALR (ADD LOGICAL) 命令 7-17
AND 命令 7-18
例 A-7
AND IMMEDIATE 命令 7-19
AP (ADD DECIMAL) 命令 8-5
例 A-34
AR 指定 (アクセス・レジスター指定) 3-5, 3-15
アドレス・スペース 3-15, 5-35
仮想アドレス 3-5
AR 指定 (アクセス・レジスター指定) 仮想アドレス
有効アドレス・スペース制御エレメント 3-33
AR (ADD) 2 進数命令 7-16
ART
参照: アクセス・レジスター変換
ART 索引バッファ
参照: ALB
ASCE
参照: アドレス・スペース制御エレメント
ASCE (アドレス・スペース制御エレメント)
タイプ例外 6-18
ASCII 文字コード
アーキテクチャーにより処理される xxi
ASF 制御ビット
参照: アドレス・スペース機能制御ビット
ASN (アドレス・スペース番号) 3-16
エントリー・テーブル・エントリー内の 5-30
許可 3-22
第 1 テーブル (AFT) 3-18
第 1 テーブル (AFT) 起点 (AFTO) 3-18
第 1 テーブル起点 (AFTO) 3-18
第 1 テーブル指標
参照: AFX
第 1 テーブル・エントリー (ASTE)
サブスペース・グループ用 5-57
第 2 テーブル (AST) 3-18
第 2 テーブル (AST) 起点 (ASTO) 3-18
第 2 テーブル指標
参照: ASX

ASN (アドレス・スペース番号) (続き)
第 2 テーブル・エントリー (ASTE)
アクセス例外としてのシーケンス例外 6-35
アクセス例外としての妥当性例外 6-35
疑似 3-17
起点、ALE 内の 5-46
シーケンス番号 (ASTESN)、ALE の中の 5-46
シーケンス番号 (ASTESN)、ASTE の中の 5-47
シーケンス例外 6-19
妥当性例外 6-19
1 次 (PASTE) 5-28
トレース制御ビット 4-13
変換 3-17
例外 6-43
変換制御ビット 3-17, 5-25
ETE 内の第 2 テーブル・アドレス 5-30
AST エントリー 5-46
参照: ASN 第 2 テーブル・エントリー
AST (ASN 第 2 テーブル) 3-18
ASTE
参照: ASN 第 2 テーブル・エントリー
ASTESN (AST エントリー・シーケンス番号)
ALE 内の 5-46
ASTE 内の 5-47
ASTO (ASN 第 2 テーブル起点) 3-18
ASX (ASN 第 2 テーブル指標)
ASX (ASN 第 2 テーブル指標) 3-17
変換例外 6-20
無効ビット 3-19
ART での使用 5-47
AT
参照: 権限テーブル
ATL (権限テーブル長)
ART での使用 5-47
ATL (権限テーブルの長さ) 3-19
ATMID (アドレッシングおよび変換モード識別) 4-26
ATO (権限テーブル起点) 3-19
ART での使用 5-47
AU (ADD UNNORMALIZED) HFP 命令 18-8
例 A-39
AUR (ADD UNNORMALIZED) HFP 命令 18-8
AW (ADD UNNORMALIZED) HFP 命令 18-8
AWR (ADD UNNORMALIZED) HFP 命令 18-8
AX (許可指標) 3-22, 5-25
AXBR (ADD) BFP 命令 19-19
AXR (ADD NORMALIZED) HFP 命令 18-7
B
BAKR (BRANCH AND STACK) 命令 10-9
例 A-10

BAL (BRANCH AND LINK) 命令 7-20
例 A-8
BALR (BRANCH AND LINK) 命令 7-20
例 A-8
BAS (BRANCH AND SAVE) 命令 7-21
例 A-8
BASR (BRANCH AND SAVE) 命令 7-20
例 A-8
BASSM (BRANCH AND SAVE AND SET MODE) 命令
7-21
例 A-8
BASTEO (基本 AST エントリー起点) 5-56
BC (BRANCH ON CONDITION) 命令 7-23
例 A-11
BCR (BRANCH ON CONDITION) 命令 7-23
BCT (BRANCH ON COUNT) 命令 7-24
例 A-12
BCTG (BRANCH ON COUNT) 命令 7-24
BCTGR (BRANCH ON COUNT) 命令 7-24
BCTR (BRANCH ON COUNT) 命令 7-24
例 A-12
BFP
データ・クラス
テスト 19-48
BFP 機能 19-1
BFP データ 19-4
変換 9-8
BFP 命令データ例外 6-20
BFP (2 進浮動小数点) 9-1
BRANCH AND LINK 命令 7-20
例 A-8
BRANCH AND SAVE 命令 7-20
例 A-8
BRANCH AND SAVE AND SET MODE 命令 7-21
例 A-8
BRANCH AND SET AUTHORITY 命令 10-5
BRANCH AND SET MODE 命令 7-22
例 A-8
BRANCH AND STACK 命令 10-9
例 A-10
BRANCH IN SUBSPACE GROUP 命令 10-12
BRANCH ON CONDITION 命令 7-23
例 A-11
BRANCH ON COUNT 命令 7-24
例 A-12
BRANCH ON INDEX HIGH 命令 7-25
例 A-12
BRANCH ON INDEX LOW OR EQUAL 命令 7-25
例 A-13

BRANCH RELATIVE AND SAVE 命令 7-26
BRANCH RELATIVE AND SAVE LONG 命令 7-26
BRANCH RELATIVE ON CONDITION 命令 7-26
BRANCH RELATIVE ON CONDITION LONG 命令 7-26
BRANCH RELATIVE ON COUNT 命令 7-27
BRANCH RELATIVE ON INDEX HIGH 命令 7-28
BRANCH RELATIVE ON INDEX LOW OR EQUAL 命令
7-28
BRAS (BRANCH RELATIVE AND SAVE) 命令 7-26
BRASL (BRANCH RELATIVE AND SAVE LONG) 命令
7-26
BRC (BRANCH RELATIVE ON CONDITION) 命令 7-26
BRCL (BRANCH RELATIVE ON CONDITION LONG) 命
令 7-26
BRCT (BRANCH RELATIVE ON COUNT) 命令 7-27
BRCTG (BRANCH RELATIVE ON COUNT) 命令 7-27
BRXH (BRANCH RELATIVE ON INDEX HIGH) 命令
7-28
BRXHG (BRANCH RELATIVE ON INDEX HIGH) 命令
7-28
BRXLE (BRANCH RELATIVE ON INDEX LOW OR
EQUAL) 命令 7-28
BRXLG (BRANCH RELATIVE ON INDEX LOW OR
EQUAL) 命令 7-28
BSA (BRANCH AND SET AUTHORITY) 命令 10-5
BSG (BRANCH IN SUBSPACE GROUP) 命令 10-12
BSM (BRANCH AND SET MODE) 命令 7-22
例 A-8
BXH (BRANCH ON INDEX HIGH) 命令 7-25
例 A-12
BXHG (BRANCH ON INDEX HIGH) 命令 7-25
BXLE (BRANCH ON INDEX LOW OR EQUAL) 命令
7-25
例 A-13
BXLEG (BRANCH ON INDEX LOW OR EQUAL) 命令
7-25

C

C (COMPARE) 2 進数命令 7-33
CANCEL SUBCHANNEL 命令 14-4
CBC (検査ブロック・コード) 11-2
近似的有効 11-2
無効 11-2
記憶キー内の 11-7
記憶機構内の 11-6
レジスター内の 11-9
有効 11-2
CBN (圧縮データ・ビット番号) 7-58
例 7-63

- CCC (チャンネル制御チェック) 16-27
- CCW 内の SLI (誤長表示抑止) フラグ 15-26
 - 即時操作での 15-29
- CCW 内の延期フラグ 15-26
 - 無効 16-25
- CCW 内のコマンド・コード 15-27
 - 適用されるフラグ 15-38
 - 無効 16-24
- CCW 内のスキップ・フラグ 15-26
 - データ転送への影響 15-33
- CCW のコマンド・チェーニング 15-32
 - 概要 13-8
 - 状況修飾ビットの影響 15-32
 - CCW 内のフラグ 15-26
- CCW のチェーニング 15-29
- CCW のデータ・チェーニング 15-31
 - 概要 13-8
 - CCW 内のフラグ 15-26
- CCW の無効シーケンス 16-25
- CCW (チャンネル・コマンド・ワード) 15-25
 - アドレス 16-18
 - 延期フラグ 15-26
 - 記憶域の指定 15-26, 15-27
 - 現 15-25
 - 事前取り出し 15-31
 - 事前取り出し制御 16-11
 - IPL に使用される 17-14
 - 使用される間接データ・アドレッシング 13-7, 15-34
 - スキップ・フラグ 15-26
 - チェーニング 15-29
 - チェック (サブチャンネル・ログアウト内の) 16-32
 - 内容 15-25
 - 入出力操作における役割 13-5
 - バイト・カウント 15-26
 - フォーマット 0 とフォーマット 1 15-25
 - フォーマット制御 16-10
 - IPL に使用される 17-14
 - 無効な指定 16-23
 - 無効なフォーマット 16-25
 - IDA フラグ 15-26
 - PCI フラグ 15-26
- CD (COMPARE) HFP 命令 18-9
- CDB (COMPARE) BFP 命令 19-24
- CDBR (COMPARE) BFP 命令 19-24
- CDFBR (CONVERT FROM FIXED) BFP 命令 19-27
- CDFR (CONVERT FROM FIXED) HFP 命令 18-10
- CDGBR (CONVERT FROM FIXED) BFP 命令 19-27
- CDGR (CONVERT FROM FIXED) HFP 命令 18-10
- CDR (COMPARE) HFP 命令 18-9
 - 例 A-40
- CDS (COMPARE DOUBLE AND SWAP) 命令 7-40
 - 例 A-43
- CDSG (COMPARE DOUBLE AND SWAP) 命令 7-40
- CDSS (圧縮データ記号サイズ) 7-58
- CE (COMPARE) HFP 命令 18-9
- CEB (COMPARE) BFP 命令 19-24
- CEBR (COMPARE) BFP 命令 19-24
- CEFBR (CONVERT FROM FIXED) BFP 命令 19-27
- CEFR (CONVERT FROM FIXED) HFP 命令 18-10
- CEGBR (CONVERT FROM FIXED) BFP 命令 19-27
- CEGR (CONVERT FROM FIXED) HFP 命令 18-10
- CER (COMPARE) HFP 命令 18-9
- CFC (COMPARE AND FORM CODEWORD) 命令 7-33
 - 例 A-51
- CFDBR (CONVERT TO FIXED) BFP 命令 19-28
- CFDR (CONVERT TO FIXED) HFP 命令 18-10
- CFEBR (CONVERT TO FIXED) BFP 命令 19-28
- CFER (CONVERT TO FIXED) HFP 命令 18-10
- CFXBR (CONVERT TO FIXED) BFP 命令 19-28
- CFXR (CONVERT TO FIXED) HFP 命令 18-10
- CG (COMPARE) 2 進数命令 7-33
- CGDBR (CONVERT TO FIXED) BFP 命令 19-28
- CGDR (CONVERT TO FIXED) HFP 命令 18-10
- CGEBR (CONVERT TO FIXED) BFP 命令 19-28
- CGER (CONVERT TO FIXED) HFP 命令 18-10
- CGF (COMPARE) 2 進数命令 7-33
- CGFR (COMPARE) 2 進数命令 7-33
- CGHI (COMPARE HALFWORD IMMEDIATE) 命令 7-42
- CGR (COMPARE) 2 進数命令 7-33
- CGXBR (CONVERT TO FIXED) BFP 命令 19-28
- CGXR (CONVERT TO FIXED) HFP 命令 18-10
- CH (COMPARE HALFWORD) 命令 7-42
 - 例 A-14
- CHECKSUM 命令 7-29
- CHI (COMPARE HALFWORD IMMEDIATE) 命令 7-42
- CHPID (チャンネル・パス ID) 13-5
 - PMCW 内の 15-7
 - RESET CHANNEL PATH で使用される 14-9
- CKSM (CHECKSUM) 命令 7-29
- CL (COMPARE LOGICAL) 命令 7-43
- CLC (COMPARE LOGICAL) 命令 7-43
 - 例 A-15
- CLCL (COMPARE LOGICAL LONG) 命令 7-44
 - 例 A-16
- CLCLE (COMPARE LOGICAL LONG EXTENDED) 命令 7-46
- CLCLU (COMPARE LOGICAL LONG UNICODE) 命令 7-50
- CLEAR SUBCHANNEL 命令 14-4
 - 開始される機能 15-13
 - 装置状況への影響 15-14

CLEAR SUBCHANNEL 命令 (続き)

- RESET CHANNEL PATH の後での使用 14-9
- CLG (COMPARE LOGICAL) 2 進数命令 7-43
- CLGF (COMPARE LOGICAL) 2 進数命令 7-43
- CLGFR (COMPARE LOGICAL) 2 進数命令 7-43
- CLGR (COMPARE LOGICAL) 2 進数命令 7-43
- CLI (COMPARE LOGICAL) 命令
 - 例 A-15
- CLI (COMPARE LOGICAL) 2 進数 7-43
- CLM (COMPARE LOGICAL CHARACTERS UNDER MASK) 命令 7-43
 - 例 A-16
- CLMH (COMPARE LOGICAL CHARACTERS UNDER MASK) 命令 7-43
- CLR (COMPARE LOGICAL) 命令 7-42
 - 例 A-15
- CLST (COMPARE LOGICAL STRING) 命令 7-53
 - 例 A-18
- CMPSC (COMPRESSION CALL) 命令 7-57
- COMPARE AND FORM CODEWORD 命令 7-33
 - 例 A-51
- COMPARE AND SIGNAL BFP 命令 19-25
- COMPARE AND SWAP 命令 7-40
 - 例 A-43
- COMPARE AND SWAP AND PURGE 命令 10-17
- COMPARE BFP 命令 19-24
- COMPARE DECIMAL 命令 8-6
 - 例 A-34
- COMPARE DOUBLE AND SWAP 命令 7-40
 - 例 A-43
- COMPARE HALFWORD 命令 7-42
 - 例 A-14
- COMPARE HALFWORD IMMEDIATE 命令 7-42
- COMPARE HFP 命令 18-9
 - 例 A-40
- COMPARE LOGICAL 命令 7-42
 - 例 A-15
- COMPARE LOGICAL CHARACTERS UNDER MASK 命令 7-43
 - 例 A-16
- COMPARE LOGICAL LONG 命令 7-44
 - 例 A-16
- COMPARE LOGICAL LONG EXTENDED 命令 7-46
- COMPARE LOGICAL LONG UNICODE 命令 7-50
- COMPARE LOGICAL STRING 命令 7-53
 - 例 A-18
- COMPARE UNTIL SUBSTRING EQUAL 命令 7-54
- COMPARE 2 進数命令 7-33
- COMPRESSION CALL 命令 7-57
 - 関連した制限
 - 圧縮辞書エントリーのチェーンの長さ 7-62
 - 圧縮時の文字記号の長さ 7-62

COMPRESSION CALL 命令 (続き)

- 関連した制限 (続き)
 - 記号変換テーブルへのオフセット 7-58
 - 記号変換テーブル・エントリーの左端ビット 7-61
 - 子の数 7-62, 7-64
 - 展開辞書エントリーのチェーンの長さ 7-62
 - 展開時の文字記号の長さ 7-62
 - フォーマット 1 兄弟記述子を使用する記号変換 7-59
- CONVERT BFP TO HFP 浮動小数点命令 9-8
- CONVERT FROM FIXED BFP 命令 19-27
- CONVERT FROM FIXED HFP 命令 18-10
- CONVERT HFP TO BFP 浮動小数点命令 9-9
- CONVERT TO BINARY 命令 7-68
 - 例 A-18
- CONVERT TO DECIMAL 命令 7-68
 - 例 A-19
- CONVERT TO FIXED BFP 命令 19-28
- CONVERT TO FIXED HFP 命令 18-10
- CONVERT UNICODE TO UTF-8 命令 7-69
- CONVERT UTF-8 TO UNICODE 命令 7-72
- COPY ACCESS 命令 7-75
- CP (COMPARE DECIMAL) 命令 8-6
 - 例 A-34
- CPU タイマー 4-39
 - 外部割り込み 6-11
 - 妥当性ビット 11-21
 - 保管域 3-53
- CPU (中央演算処理装置) 2-2
 - アドレス 4-47
 - 外部割り込み中に格納されたとき 6-10
 - 割り当て済み記憶位置 3-49
 - 再試行 11-2
 - 識別 (ID) 10-92
 - 状態 4-1
 - 作動 4-2
 - チェック停止 4-2
 - 停止 4-2
 - ロード 4-2
 - TOD クロックへの影響なし 4-34
 - 信号発信 4-48
 - チェックポイント 11-2
 - 逐次化 5-89
 - バージョン・コード 10-92
 - パワーオン・リセットの影響 4-45
 - マシン・タイプ番号 10-92
 - モデル番号 10-92
 - リセット 4-43
 - プロセッサ信号指令 4-49
 - レジスター 2-3
 - 保管域 3-53
 - 割り込みのストリングを原因とするハングアップ 4-3

CPYA (COPY ACCESS) 命令 7-75
CR
参照：制御レジスター
CR (COMPARE) 2 進数命令 7-33
CRW のチェーニング 17-18, 17-19
CRW (チャンネル報告ワード) 17-19
エラー・リカバリー・コード (ERC) 17-20
オーバーフロー 17-19
記憶 14-16
請求 17-19
チェーニング 17-18, 17-19
報告ソース ID (RSID) 17-21
報告ソース・コード (RSC) 17-19
CS (COMPARE AND SWAP) 命令 7-40
例 A-43
CSCH (CLEAR SUBCHANNEL) 命令 14-4
CSG (COMPARE AND SWAP) 命令 7-40
CSP (COMPARE AND SWAP AND PURGE) 命令 10-17
CUSE (COMPARE UNTIL SUBSTRING EQUAL) 命令
7-54
CUTFU (CONVERT UTF-8 TO UNICODE) 命令 7-72
CUUTF (CONVERT UNICODE TO UTF-8) 命令 7-69
CVB (CONVERT TO BINARY) 命令 7-68
例 A-18
CVBG (CONVERT TO BINARY) 命令 7-68
CVD (CONVERT TO DECIMAL) 命令 7-68
例 A-19
CVDG (CONVERT TO DECIMAL) 命令 7-68
CXBR (COMPARE) BFP 命令 19-24
CXFBR (CONVERT FROM FIXED) BFP 命令 19-27
CXFR (CONVERT FROM FIXED) HFP 命令 18-10
CXGBR (CONVERT FROM FIXED) BFP 命令 19-27
CXGR (CONVERT FROM FIXED) HFP 命令 18-10
CXR (COMPARE) HFP 命令 18-9

D

D (DIVIDE) 2 進数命令 7-75
例 A-19
DAT
参照：動的アドレス変換
DAT テーブル・フォーマット・エラー 6-34
DAT モード (PSW のビット) 4-5
アドレス変換での使用 3-27
DCTI (装置接続時間間隔)
ESW 内の 16-38
DCTI (装置接続時間)
測定ブロック内の 17-4
DD (DIVIDE) HFP 命令 18-11
DDB (DIVIDE) BFP 命令 19-31

DDBR (DIVIDE) BFP 命令 19-31
DDR (DIVIDE) HFP 命令 18-11
DE (DIVIDE) HFP 命令 18-11
DEB (DIVIDE) BFP 命令 19-31
DEBR (DIVIDE) BFP 命令 19-31
DER (DIVIDE) HFP 命令 18-11
例 A-40
DIAGNOSE 命令 10-18
DIDBR (DIVIDE TO INTEGER) BFP 命令 19-32
DIEBR (DIVIDE TO INTEGER) BFP 命令 19-32
DIVIDE BFP 命令 19-31
DIVIDE DECIMAL 命令 8-6
例 A-35
DIVIDE HFP 命令 18-11
例 A-40
DIVIDE LOGICAL 命令 7-75
DIVIDE SINGLE 2 進数命令 7-76
DIVIDE TO INTEGER BFP 命令 19-32
剰余結果 19-9
DIVIDE 2 進数命令 7-75
例 A-19
DL (DIVIDE LOGICAL) 命令 7-75
DLG (DIVIDE LOGICAL) 2 進数命令 7-75
DLGR (DIVIDE LOGICAL) 2 進数命令 7-75
DLR (DIVIDE LOGICAL) 命令 7-75
DP (DIVIDE DECIMAL) 命令 8-6
例 A-35
DR (DIVIDE) 2 進数命令 7-75
DSG (DIVIDE SINGLE) 2 進数命令 7-76
DSGF (DIVIDE SINGLE) 2 進数命令 7-76
DSGFR (DIVIDE SINGLE) 2 進数命令 7-76
DSGR (DIVIDE SINGLE) 2 進数命令 7-76
DU (ディスパッチ可能単位) 5-37
DUALD (ディスパッチ可能単位アクセス・リスト指定) 5-44
DUCT (ディスパッチ可能単位制御テーブル) 5-44, 5-55
DUCTO (ディスパッチ可能単位制御テーブル起点) 5-42
DXBR (DIVIDE) BFP 命令 19-31
DXC (データ例外コード) 6-14, 19-15
要約図 19-16
IEEE の要約図 19-15
DXR (DIVIDE) HFP 命令 18-11

E

E 命令フォーマット 5-5
E (エクサ) xx
EAR (EXTRACT ACCESS) 命令 7-79
EAX
参照：拡張許可指標
EBCDIC (Extended Binary-Coded-Decimal Interchange Code)

EBCDIC (Extended Binary-Coded-Decimal Interchange Code) (続き)
EBCDIC 用に設計されたアーキテクチャー xxi
ECC (エラー検査/訂正) 11-2
ECW (拡張制御ワード) 16-39
SCSW 内の表示 16-11
ED (EDIT) 命令 8-7
例 A-35
EDIT 命令 8-7
例 A-35
EDIT AND MARK 命令 8-9
例 A-36
EDMK (EDIT AND MARK) 命令 8-9
例 A-36
EFPC (EXTRACT FPC) 命令 19-37
EKM (エントリー・キー・マスク) 5-30
EMIF (ESCON-multiple-image facility) 1-12
EPAR (EXTRACT PRIMARY ASN) 命令 10-19
EPSW (EXTRACT PSW) 命令 7-79
ERC (エラー・リカバリー・コード) 17-20
EREG (EXTRACT STACKED REGISTERS) 命令 10-20
EREGG (EXTRACT STACKED REGISTERS) 命令 10-20
ERW (拡張報告ワード) 16-31, 16-35
チャンネル制御チェックの結果としての 16-27
チャンネル・データ・チェックの結果としての 16-26
ESAR (EXTRACT SECONDARY ASN) 命令 10-20
ESA/370 アーキテクチャー 1-13
ESA/390
z/Architecture との互換性 1-15
ESA/390 アーキテクチャー
アーキテクチャー・モード制御 12-2
ESCON チャンネル間アダプター
参照される資料 xxi
ESCON (エンタープライズ・システム接続アーキテクチャー) 13-2
ESCON (エンタープライズ・システム接続アーキテクチャー)
入出力インターフェース
参照される資料 xxi
ESCON-multiple-image facility (EMIF) 1-12
ESEA (EXTRACT AND SET EXTENDED AUTHORITY)
命令 10-19
ESTA (EXTRACT STACKED STATE) 命令 10-22
ESW フォーマット・ビット (SCSW の) 16-8
ESW (拡張状況ワード) 16-31
ET
参照: エントリー・テーブル
ETE
参照: エントリー・テーブル・エントリー
ETL (エントリー・テーブル長) 5-29
ETO (エントリー・テーブル起点) 5-29

ETR
外部割り込み 6-11
ETR 同期検査 (マシン・チェック条件) 11-22
ETR (外部時刻参照) 2-6
ETR (external time reference: 外部時刻参照) 機能 1-11
EX (エントリー指標) 5-28
変換例外 6-21
EX (EXECUTE)
参照: EXECUTE 命令
EXCLUSIVE OR 命令 7-77
例 A-19
EXECUTE 命令 7-78
アドレス比較の影響 12-1
ターゲットに関する PER イベント 4-31
ターゲットの取り出し中の例外 6-8
例 A-21
EXTRACT ACCESS 命令 7-79
EXTRACT AND SET EXTENDED AUTHORITY 命令
10-19
EXTRACT FPC 命令 19-37
EXTRACT PRIMARY ASN 命令 10-19
EXTRACT PSW 命令 7-79
EXTRACT SECONDARY ASN 命令 10-20
EXTRACT STACKED REGISTERS 命令 10-20
EXTRACT STACKED STATE 命令 10-22

F

FICON 入出力インターフェース
参照される資料 xxi
FIDBR (LOAD FP INTEGER) BFP 命令 19-38
FIDR (LOAD FP INTEGER) HFP 命令 18-13
FIEBR (LOAD FP INTEGER) BFP 命令 19-38
FIER (LOAD FP INTEGER) HFP 命令 18-13
FIFO (先入れ先出し) キューイング
ロックおよびアンロックの例 A-47
FIXBR (LOAD FP INTEGER) BFP 命令 19-38
FIXR (LOAD FP INTEGER) HFP 命令 18-13
FPC (浮動小数点制御) レジスター 19-2

G

G (ギガ) xx
GMT (グリニッジ標準時): UTC と同義の廃語 4-36

H

HALT SUBCHANNEL 命令 14-6
開始される機能 15-14
RESET CHANNEL PATH の後での使用 14-9
SCSW のカウント・フィールドへの影響 15-17

HALVE HFP 命令 18-12
例 A-41
HASCE (ホーム・アドレス・スペース制御エレメント) 3-29
HDR (HALVE) HFP 命令 18-12
例 A-41
HER (HALVE) HFP 命令 18-12
HFP 指数
アンダーフロー
例外 6-22
オーバーフロー
例外 6-22
HFP データ 18-3
変換 9-8
HFP 平方根
例外 6-23
HFP 有効数字
例外 6-23
HFP (16 進浮動小数点) 9-1
HRSD (ホーム実スペース指定) 3-29
HRTD (ホーム領域テーブル指定) 3-29
HRTL (ホーム領域テーブル長さ) 3-30
HRTO (ホーム領域テーブル起点) 3-30
HSCH (HALT SUBCHANNEL) 命令 14-6
HSTD (ホーム・セグメント・テーブル指定) 3-29
HSTL (ホーム・セグメント・テーブル長さ) 3-30
HSTO (ホーム・セグメント・テーブル起点) 3-30

I

IAC (INSERT ADDRESS SPACE CONTROL) 命令 10-24
IC (命令カウンター)
参照: 命令アドレス
IC (INSERT CHARACTER) 命令 7-79
ICM (INSERT CHARACTERS UNDER MASK) 命令 7-79
例 A-22
ICMH (INSERT CHARACTERS UNDER MASK) 命令
7-79
ID
参照: CPU 識別、センス ID
IDA (間接データ・アドレス) 15-34
CCW 内のフラグ 15-26
IDAW (間接データ・アドレス・ワード) 15-34
チェック (サブチャネル・ログアウト内の) 16-32
無効なアドレス 16-24
無効なアドレス指定 16-24, 16-25
IEEE 標準 1-10
IEEE 例外条件 19-15
要約図 19-15
IEEE 例外条件データ例外 6-20, 19-11
IFCC (インターフェース制御チェック) 16-27

IIHH (INSERT IMMEDIATE) 命令 7-80
IIHL (INSERT IMMEDIATE) 命令 7-80
IILH (INSERT IMMEDIATE) 命令 7-80
IILL (INSERT IMMEDIATE) 命令 7-80
ILC (命令長コード) 6-7
監視プログラム呼び出し割り込みの 6-44
プログラム割り込みの 6-13
割り当て済み記憶位置 3-49
IML (初期マシン・ローディング) 制御 12-3
INSERT ADDRESS SPACE CONTROL 命令 10-24
INSERT CHARACTER 命令 7-79
INSERT CHARACTERS UNDER MASK 命令 7-79
例 A-22
INSERT IMMEDIATE 命令 7-80
INSERT PROGRAM MASK 命令 7-81
INSERT PSW KEY 命令 10-25
INSERT STORAGE KEY EXTENDED 命令 10-25
INSERT VIRTUAL STORAGE KEY 命令 10-26
INVALIDATE PAGE TABLE ENTRY 命令 10-27
CPU が停止しているときの影響 4-2
IPK (INSERT PSW KEY) 命令 10-25
IPL (初期プログラム・ローディング) 4-46, 17-13
CPU 状態への影響 4-2
IPM (INSERT PROGRAM MASK) 命令 7-81
IPTE (INVALIDATE PAGE TABLE ENTRY) 命令 10-27
IRB (割り込み応答ブロック) 16-6
記憶域要件 16-11
ISC (入出力割り込みサブクラス・コード) 15-2
拡張 14-18
ISKE (INSERT STORAGE KEY EXTENDED) 命令 10-25
IVSK (INSERT VIRTUAL STORAGE KEY) 命令 10-26
I/O (入出力) 2-6
基本機能 15-1
サポート機能 17-1
データのブロック化 15-20
CPU タイマーへの影響 4-39

K

K (キロ) xx
KDB (COMPARE AND SIGNAL) BFP 命令 19-25
KDBR (COMPARE AND SIGNAL) BFP 命令 19-25
KEB (COMPARE AND SIGNAL) BFP 命令 19-25
KEBR (COMPARE AND SIGNAL) BFP 命令 19-25
KXBR (COMPARE AND SIGNAL) BFP 命令 19-25

L

L (LOAD) 2 進数命令 7-81
例 A-22

LA (LOAD ADDRESS) 命令 7-82
 例 A-23
 LAE (LOAD ADDRESS EXTENDED) 命令 7-82
 LAM (LOAD ACCESS MULTIPLE) 命令 7-81
 LARL (LOAD ADDRESS RELATIVE LONG) 命令 7-83
 LASP (LOAD ADDRESS SPACE PARAMETERS) 命令
 10-28
 LCDBR (LOAD COMPLEMENT) BFP 命令 19-37
 LCDR (LOAD COMPLEMENT) HFP 命令 18-13
 LCEBR (LOAD COMPLEMENT) BFP 命令 19-37
 LCER (LOAD COMPLEMENT) HFP 命令 18-13
 LCGFR (LOAD COMPLEMENT) 2 進数命令 7-84
 LCGR (LOAD COMPLEMENT) 2 進数命令 7-84
 LCR (LOAD COMPLEMENT) 2 進数命令 7-84
 LCTL (LOAD CONTROL) 命令 10-37
 LCTLG (LOAD CONTROL) 命令 10-37
 LCXBR (LOAD COMPLEMENT) BFP 命令 19-37
 LCXR (LOAD COMPLEMENT) HFP 命令 18-13
 LD (LOAD) 浮動小数点命令 9-11
 LDE (LOAD LENGTHENED) HFP 命令 18-14
 LDEB (LOAD LENGTHENED) BFP 命令 19-40
 LDEBR (LOAD LENGTHENED) BFP 命令 19-40
 LDER (LOAD LENGTHENED) HFP 命令 18-14
 LDR (LOAD) 浮動小数点命令 9-11
 LDXBR (LOAD ROUNDED) BFP 命令 19-41
 LDXR (LOAD ROUNDED) HFP 命令 18-16
 LE (LOAD) 浮動小数点命令 9-11
 LEDBR (LOAD ROUNDED) BFP 命令 19-41
 LEDR (LOAD ROUNDED) HFP 命令 18-16
 LER (LOAD) 浮動小数点命令 9-11
 LEXBR (LOAD ROUNDED) BFP 命令 19-41
 LEXR (LOAD ROUNDED) HFP 命令 18-16
 LFPC (LOAD FPC) 命令 19-39
 LG (LOAD) 2 進数命令 7-81
 LGF (LOAD) 2 進数命令 7-81
 LGFR (LOAD) 2 進数命令 7-81
 LGH (LOAD HALFWORD) 命令 7-84
 LGHI (LOAD HALFWORD IMMEDIATE) 命令 7-85
 LGR (LOAD) 2 進数命令 7-81
 LH (LOAD HALFWORD) 命令 7-84
 例 A-23
 LHI (LOAD HALFWORD IMMEDIATE) 命令 7-85
 LIFO (後入れ先出し) キューイング
 ロックおよびアンロックの例 A-46
 LLGC (LOAD LOGICAL CHARACTER) 命令 7-85
 LLGF (LOAD LOGICAL) 命令 7-85
 LLGFR (LOAD LOGICAL) 命令 7-85
 LLGH (LOAD LOGICAL HALFWORD) 命令 7-85
 LLGT (LOAD LOGICAL THIRTY ONE BITS) 命令 7-86

LLGTR (LOAD LOGICAL THIRTY ONE BITS) 命令
 7-86
 LLIHH (LOAD LOGICAL IMMEDIATE) 命令 7-86
 LLIHL (LOAD LOGICAL IMMEDIATE) 命令 7-86
 LLILH (LOAD LOGICAL IMMEDIATE) 命令 7-86
 LLILL (LOAD LOGICAL IMMEDIATE) 命令 7-86
 LM (LOAD MULTIPLE) 命令 7-86
 LMD (LOAD MULTIPLE DISJOINT) 命令 7-87
 LMG (LOAD MULTIPLE) 命令 7-86
 LMH (LOAD MULTIPLE HIGH) 命令 7-87
 LNDBR (LOAD NEGATIVE) BFP 命令 19-40
 LNDR (LOAD NEGATIVE) HFP 命令 18-15
 LNEBR (LOAD NEGATIVE) BFP 命令 19-40
 LNER (LOAD NEGATIVE) HFP 命令 18-15
 LNGFR (LOAD NEGATIVE) 2 進数命令 7-88
 LNGR (LOAD NEGATIVE) 2 進数命令 7-88
 LNR (LOAD NEGATIVE) 2 進数命令 7-88
 LNXBR (LOAD NEGATIVE) BFP 命令 19-40
 LNXR (LOAD NEGATIVE) HFP 命令 18-15
 LOAD 浮動小数点命令 9-11
 LOAD ACCESS MULTIPLE 命令 7-81
 LOAD ADDRESS 命令 7-82
 例 A-23
 LOAD ADDRESS EXTENDED 命令 7-82
 LOAD ADDRESS RELATIVE LONG 命令 7-83
 LOAD ADDRESS SPACE PARAMETERS 命令 10-28
 LOAD AND TEST BFP 命令 19-37
 LOAD AND TEST HFP 命令 18-12
 LOAD AND TEST 2 進数命令 7-83
 LOAD COMPLEMENT BFP 命令 19-37
 LOAD COMPLEMENT HFP 命令 18-13
 LOAD COMPLEMENT 2 進数命令 7-84
 LOAD CONTROL 命令 10-37
 LOAD FP INTEGER BFP 命令 19-38
 LOAD FP INTEGER HFP 命令 18-13
 LOAD FPC 命令 19-39
 LOAD HALFWORD 命令 7-84
 例 A-23
 LOAD HALFWORD IMMEDIATE 命令 7-85
 LOAD LENGTHENED BFP 命令 19-40
 LOAD LENGTHENED HFP 命令 18-14
 LOAD LOGICAL 命令 7-85
 LOAD LOGICAL CHARACTER 命令 7-85
 LOAD LOGICAL HALFWORD 命令 7-85
 LOAD LOGICAL IMMEDIATE 命令 7-86
 LOAD LOGICAL THIRTY ONE BITS 命令 7-86
 LOAD MULTIPLE 命令 7-86
 LOAD MULTIPLE DISJOINT 命令 7-87
 LOAD MULTIPLE HIGH 命令 7-87

LOAD NEGATIVE BFP 命令 19-40
 LOAD NEGATIVE HFP 命令 18-15
 LOAD NEGATIVE 2 進数命令 7-88
 LOAD PAIR FROM QUADWORD 命令 7-88
 LOAD POSITIVE BFP 命令 19-41
 LOAD POSITIVE HFP 命令 18-15
 LOAD POSITIVE 2 進数命令 7-88
 LOAD PSW 命令 10-37
 LOAD PSW EXTENDED 命令 10-38
 LOAD REAL ADDRESS 命令 10-39
 LOAD REVERSED 命令 7-89
 LOAD ROUNDED BFP 命令 19-41
 LOAD ROUNDED HFP 命令 18-16
 LOAD USING REAL ADDRESS 命令 10-44
 LOAD ZERO 浮動小数点命令 9-11
 LOAD 2 進数命令 7-81
 例 A-22
 LPAR (論理分割) モード 1-12, 1-14
 LPDBR (LOAD POSITIVE) BFP 命令 19-41
 LPDR (LOAD POSITIVE) HFP 命令 18-15
 LPEBR (LOAD POSITIVE) BFP 命令 19-41
 LPER (LOAD POSITIVE) HFP 命令 18-15
 LPGFR (LOAD POSITIVE) 2 進数命令 7-88
 LPGR (LOAD POSITIVE) 2 進数命令 7-88
 LPM (論理バス・マスク) 15-4, 15-23
 システム・パフォーマンスへの影響 15-10
 IPL に使用される 17-14
 LPQ (LOAD PAIR FROM QUADWORD) 命令 7-88
 LPR (LOAD POSITIVE) 2 進数命令 7-88
 LPSW (LOAD PSW) 命令 10-37
 LPSWE (LOAD PSW EXTENDED) 命令 10-38
 LPUM (最終使用バス・マスク) 15-5
 フィールド妥当性フラグ (サブチャネル・ログアウトでの) 16-33
 ESW 内の 16-33
 LPXBR (LOAD POSITIVE) BFP 命令 19-41
 LPXR (LOAD POSITIVE) HFP 命令 18-15
 LR (LOAD) 2 進数命令 7-81
 LRA (LOAD REAL ADDRESS) 命令 10-39
 LRAG (LOAD REAL ADDRESS) 命令 10-39
 LRDR (LOAD ROUNDED) HFP 命令 18-16
 LRER (LOAD ROUNDED) HFP 命令 18-16
 LRV (LOAD REVERSED) 命令 7-89
 LRVG (LOAD REVERSED) 命令 7-89
 LRVGR (LOAD REVERSED) 命令 7-89
 LRVH (LOAD REVERSED) 命令 7-89
 LRVR (LOAD REVERSED) 命令 7-89
 LT (リンケージ・テーブル) 5-28
 LTD (リンケージ・テーブル指定) 5-28

LTDBR (LOAD AND TEST) BFP 命令 19-37
 LTDR (LOAD AND TEST) HFP 命令 18-12
 LTEBR (LOAD AND TEST) BFP 命令 19-37
 LTER (LOAD AND TEST) HFP 命令 18-12
 LTGFR (LOAD AND TEST) 2 進数命令 7-84
 LTGR (LOAD AND TEST) 2 進数命令 7-83
 LTL (リンケージ・テーブル長) 5-28
 LTO (リンケージ・テーブル起点) 5-28
 LTR (LOAD AND TEST) 2 進数命令 7-83
 LTXBR (LOAD AND TEST) BFP 命令 19-37
 LTXR (LOAD AND TEST) HFP 命令 18-12
 LURA (LOAD USING REAL ADDRESS) 命令 10-44
 LURAG (LOAD USING REAL ADDRESS) 命令 10-44
 LX (リンケージ指標) 5-28
 変換例外 6-23
 無効ビット 5-28
 LXD (LOAD LENGTHENED) HFP 命令 18-14
 LXDB (LOAD LENGTHENED) BFP 命令 19-40
 LXDBR (LOAD LENGTHENED) BFP 命令 19-40
 LXDR (LOAD LENGTHENED) HFP 命令 18-14
 LXE (LOAD LENGTHENED) HFP 命令 18-14
 LXEB (LOAD LENGTHENED) BFP 命令 19-40
 LXEBR (LOAD LENGTHENED) BFP 命令 19-40
 LXER (LOAD LENGTHENED) HFP 命令 18-14
 LXR (LOAD) 浮動小数点命令 9-11
 LZDR (LOAD ZERO) 浮動小数点命令 9-11
 LZER (LOAD ZERO) 浮動小数点命令 9-11
 LZXR (LOAD ZERO) 浮動小数点命令 9-11

M

M (メガ) xx
 M (MULTIPLY) 2 進数命令 7-105
 例 A-27
 MADB (MULTIPLY AND ADD) BFP 命令 19-44
 MADBR (MULTIPLY AND ADD) BFP 命令 19-44
 MAEB (MULTIPLY AND ADD) BFP 命令 19-44
 MAEBR (MULTIPLY AND ADD) BFP 命令 19-44
 MC (MONITOR CALL) 命令 7-90
 MCIC (マシン・チェック割り込みコード) 3-52, 11-14
 MD (MULTIPLY) HFP 命令 18-16
 MDB (MULTIPLY) BFP 命令 19-42
 MDBR (MULTIPLY) BFP 命令 19-42
 MDE (MULTIPLY) HFP 命令 18-16
 MDEB (MULTIPLY) BFP 命令 19-42
 MDEBR (MULTIPLY) BFP 命令 19-42
 MDER (MULTIPLY) HFP 命令 18-16
 MDR (MULTIPLY) HFP 命令 18-16
 例 A-41
 ME (MULTIPLY) HFP 命令 18-16

MEE (MULTIPLY) HFP 命令 18-16
MEEB (MULTIPLY) BFP 命令 19-42
MEEBR (MULTIPLY) BFP 命令 19-42
MEER (MULTIPLY) HFP 命令 18-16
MER (MULTIPLY) HFP 命令 18-16
MGHI (MULTIPLY HALFWORD IMMEDIATE) 命令
7-106
MH (MULTIPLY HALFWORD) 命令 7-106
例 A-28
MHI (MULTIPLY HALFWORD IMMEDIATE) 命令
7-106
ML (MULTIPLY LOGICAL) 命令 7-106
MLG (MULTIPLY LOGICAL) 命令 7-106
MLGR (MULTIPLY LOGICAL) 命令 7-106
MLR (MULTIPLY LOGICAL) 命令 7-106
MODIFY STACKED STATE 命令 10-44
MODIFY SUBCHANNEL 命令 14-7
MONITOR CALL 命令 7-90
MOVE 命令 7-91
例 A-21, A-23
MOVE INVERSE 命令 7-91
例 A-24
MOVE LONG 命令 7-92
例 A-25
MOVE LONG EXTENDED 命令 7-95
MOVE LONG UNICODE 命令 7-99
MOVE NUMERICS 命令 7-102
例 A-26
MOVE PAGE 命令 10-46
MOVE STRING 命令 7-103
例 A-26
MOVE TO PRIMARY 命令 10-48
MOVE TO SECONDARY 命令 10-48
MOVE WITH DESTINATION KEY 命令 10-50
MOVE WITH KEY 命令 10-50
MOVE WITH OFFSET 命令 7-104
例 A-26
MOVE WITH SOURCE KEY 命令 10-52
MOVE ZONES 命令 7-104
例 A-27
MP (MULTIPLY DECIMAL) 命令 8-11
例 A-37
MR (MULTIPLY) 2 進数命令 7-105
例 A-27
MS (MULTIPLY SINGLE) 命令 7-107
MSCH (MODIFY SUBCHANNEL) 命令 14-7
MSDB (MULTIPLY AND SUBTRACT) BFP 命令 19-44
MSDBR (MULTIPLY AND SUBTRACT) BFP 命令 19-44
MSEB (MULTIPLY AND SUBTRACT) BFP 命令 19-44
MSEBR (MULTIPLY AND SUBTRACT) BFP 命令 19-44
MSG (MULTIPLY SINGLE) 命令 7-107
MSGF (MULTIPLY SINGLE) 命令 7-107
MSGFR (MULTIPLY SINGLE) 命令 7-107
MSGR (MULTIPLY SINGLE) 命令 7-107
MSR (MULTIPLY SINGLE) 命令 7-107
MSTA (MODIFY STACKED STATE) 命令 10-44
MULTIPLY AND ADD BFP 命令 19-44
MULTIPLY AND SUBTRACT BFP 命令
MULTIPLY AND SUBTRACT BFP 命令 19-44
MULTIPLY BFP 命令 19-42
MULTIPLY DECIMAL 命令 8-11
例 A-37
MULTIPLY HALFWORD 命令 7-106
例 A-28
MULTIPLY HALFWORD IMMEDIATE 命令 7-106
MULTIPLY HFP 命令 18-16
例 A-41
MULTIPLY LOGICAL 命令 7-106
MULTIPLY SINGLE 命令 7-107
MULTIPLY 2 進数命令 7-105
例 A-27
MVC (MOVE) 命令 7-91
例 A-21, A-23
MVCDK (MOVE WITH DESTINATION KEY) 命令
10-50
MVCIN (MOVE INVERSE) 命令 7-91
例 A-24
MVCK (MOVE WITH KEY) 命令 10-50
MVCL (MOVE LONG) 命令 7-92
例 A-25
MVCLE (MOVE LONG EXTENDED) 命令 7-95
MVCLU (MOVE LONG UNICODE) 命令 7-99
MVCP (MOVE TO PRIMARY) 命令 10-48
MVCS (MOVE TO SECONDARY) 命令 10-48
MVCSK (MOVE WITH SOURCE KEY) 命令 10-52
MVI (MOVE) 命令 7-91
例 A-24
MVN (MOVE NUMERICS) 命令 7-102
例 A-26
MVO (MOVE WITH OFFSET) 命令 7-104
例 A-26
MVPG (MOVE PAGE) 命令 10-46
MVST (MOVE STRING) 命令 7-103
例 A-26
MVZ (MOVE ZONES) 命令 7-104
例 A-27
MXBR (MULTIPLY) BFP 命令 19-42
MXD (MULTIPLY) HFP 命令 18-16

MXDB (MULTIPLY) BFP 命令 19-42
MXDBR (MULTIPLY) BFP 命令 19-42
MXDR (MULTIPLY) HFP 命令 18-16
MXR (MULTIPLY) HFP 命令 18-16

N

N 条件 (入出力) 16-12
N (AND) 命令 7-18
NaN (非数値) 19-7
NC (AND) 命令 7-18
NG (AND) 命令 7-18
NGR (AND) 命令 7-18
NI (AND) 命令 7-18
例 A-8
NIHH (AND IMMEDIATE) 命令 7-19
NIHL (AND IMMEDIATE) 命令 7-19
NILH (AND IMMEDIATE) 命令 7-19
NILL (AND IMMEDIATE) 命令 7-19
NR (AND) 命令 7-18

O

O (OR) 命令 7-108
OC (OR) 命令 7-108
OG (OR) 命令 7-108
OGR (OR) 命令 7-108
OI (OR) 命令 7-108
問題の例 A-43
例 A-28
OIHH (OR IMMEDIATE) 命令 7-109
OIHL (OR IMMEDIATE) 命令 7-109
OILH (OR IMMEDIATE) 命令 7-109
OILL (OR IMMEDIATE) 命令 7-109
OR 命令 7-108
即値 OR での問題の例 A-43
例 A-28
OR IMMEDIATE 命令 7-109
ORB (操作要求ブロック) 15-20
拡張制御 15-23
無効 16-25
論理バス・マスク (LPM) 15-23
割り込みパラメーター 15-20
CSS 優先順位 15-24

P

P (ベタ) xx
PACK 命令 7-109
例 A-28

PACK ASCII 命令 7-110
PACK UNICODE 命令 7-111
PAGE IN 命令 10-53
PAGE OUT 命令 10-54
PALB (PURGE ALB) 命令 10-78
PAM (バス使用可能マスク) 15-7
再構成の影響 15-10
従属関係への影響 15-10
リセットの影響 15-10
PASCE (1 次アドレス・スペース制御エレメント) 3-27
PASN (1 次アドレス・スペース番号) 3-16
トレース・エントリー内の 4-22
PASTE (1 次 AST エントリー) 5-28
PASTE0 (1 次 AST エントリー起点) 5-28, 5-42
PC タイプ・ビット 5-30
PC 番号 10-56
トレース・エントリー内の 4-22
変換 5-28
リンケージ・スタック状態エントリー内の 5-72
PC 変換指定例外 6-25
PC ss (スペース切り替えを伴う PROGRAM CALL) 10-59
PC (PROGRAM CALL) 命令 10-55
PCI (プログラム制御割り込み) 15-33
サブチャンネル状況 16-23
中間割り込み条件 16-17
CCW 内のフラグ 15-26
PC-cp (現 1 次への PROGRAM CALL 命令) 10-58
PC-ss (スペース切り替えを伴う PROGRAM CALL 命令)
PER (プログラム・イベント記録) 4-24
アクセス識別 3-50, 4-27
アドレス 4-27
割り当て済み記憶位置 3-49
アドレス・スペース制御エレメント (ASCE) 識別 4-27
イベント 4-24
拡張機能 1-11
記憶域指示 4-29
開始アドレス 4-25
終了アドレス 4-25
循環 4-30
記憶域変更イベント 4-31
コード 4-26
割り当て済み記憶位置 3-49
標識の優先順位 4-28
ブランチ成功イベント 4-30
プログラム割り込み条件 6-25
マスク
PER イベント 4-24
PSW のビット 4-5
実アドレスを使用した格納イベント 4-31
命令取り出しイベント 4-31
ASCE (アドレス・スペース制御エレメント) 識別 4-27

PER (プログラム・イベント記録) (続き)
 ATMID (アドレッシングおよび変換モード識別) 4-26
PERFORM LOCKED OPERATION 命令 7-112
 例 A-50
PERFORM LOCKED OPERATION 命令で使用されるロック
 7-123
PFRA (ページ・フレーム実アドレス) 3-32
PGIN (PAGE IN) 命令 10-53
PGOUT (PAGE OUT) 命令 10-54
PIM (インストール済みパス・マスク) 15-6
PKA (PACK ASCII) 命令 7-110
PKM (PSW キー・マスク) 5-24
PKU (PACK UNICODE) 命令 7-111
PLO (PERFORM LOCKED OPERATION) 命令 7-112
 例 A-50
PMCW のサブチャンネル使用可能ビット 15-2
PMCW (パス管理制御ワード) 15-2
 チャンネル・パス ID (CHPID) 15-7
PNOM (パス非稼働マスク) 15-4
 POM への影響 15-10
 SCSW で示される 16-12
POM (パス稼働マスク) 15-6
 PNOM への影響 15-10
POST と WAIT のバイパス A-45
POST (SVC)
 バイパスするルーチンの例 A-45
PR (PROGRAM RETURN) 命令 10-68
Processor Resource/Systems Manager (PR/SM) 1-12, 1-14
PROGRAM CALL のタイプ 5-30
PROGRAM CALL 命令 10-55
 タイプ 5-30
 トレース・エントリー 4-22
PROGRAM RETURN 命令 10-68
PROGRAM TRANSFER 命令 10-72
 トレース・エントリー 4-22
PRSD (1 次実スペース指定) 3-27
PRSTKO (1 次実スペース・トークン起点) 3-29
PRTD (1 次領域テーブル指定) 3-27
PRTL (1 次領域テーブル長さ) 3-29
PRTO (1 次領域テーブル起点) 3-28
PR-cp (現 1 次への PROGRAM RETURN 命令) 10-68
PR-ss (スペース切り替えを伴う PROGRAM RETURN 命
令) 10-68
PR/SM (Processor Resource/Systems Manager) 1-12, 1-14
PSALD (1 次スペース・アクセス・リスト指定) 5-45
PSTD (1 次セグメント・テーブル指定)
PSTD (1 次セグメント・テーブル指定) 3-27
PSTL (1 次セグメント・テーブル長さ) 3-29
PSTO (1 次セグメント・テーブル起点) 3-28

PSW キー 4-5
 アクセス・キーとして使用される 3-8
 エントリー・テーブル・エントリー内の 5-65
 制御ビット 5-64
 妥当性ビット 11-20
 トレース・エントリー内の 4-22
PSW キー・マスク (PKM) 5-24
 制御ビット 5-64
 リンケージ・スタック状態エントリー内の 5-70
PSW 内の入出力マスク 4-5
PSW (プログラム状況ワード) 2-3, 4-3
 関連の例外 6-9
 現 4-3, 5-10
 割り込み時の格納 6-2
 状況記憶保管域 3-53
 妥当性ビット 11-20
 フォーマット・エラー 6-9
 プログラム実行における 5-10
 リンケージ・スタック状態エントリー内の 5-71
PT ss (スペース切り替えを伴う PROGRAM TRANSFER 命
令) 10-73
PT (PROGRAM TRANSFER) 命令 10-72
PTLB (PURGE TLB) 命令 10-78
PTO (ページ・テーブル起点) 3-31
PT-cp (現 1 次への PROGRAM TRANSFER 命令) 10-73
PT-ss (スペース切り替えを伴う PROGRAM TRANSFER 命
令)
PURGE ALB 命令 10-78
PURGE TLB 命令 10-78
PX (ページ指標) 3-26

Q

QNaN (クァイエット NaN) 19-7

R

RCHP (RESET CHANNEL PATH) 命令 14-8
RESET CHANNEL PATH 命令 14-8
 開始される機能 15-42
RESET REFERENCE BIT EXTENDED 命令 10-78
RESUME PROGRAM 命令 10-79
RESUME SUBCHANNEL 命令 14-10
 開始される機能 15-17
 測定ブロックのカウント 17-3
 チャンネル・プログラム要件 14-10
RFX (領域第 1 指標) 3-26
RI 命令フォーマット 5-5
RIE 命令フォーマット 5-5
RIL 命令フォーマット 5-5

RLL (ROTATE LEFT SINGLE LOGICAL) 命令 7-127
 RLLG (ROTATE LEFT SINGLE LOGICAL) 命令 7-127
 ROTATE LEFT SINGLE LOGICAL 命令 7-127
 RP (RESUME PROGRAM) 命令 10-79
 RR 命令フォーマット 5-5
 RRBE (RESET REFERENCE BIT EXTENDED) 命令
 10-78
 RRE 命令フォーマット 5-5
 RRF 命令フォーマット 5-5
 RS 命令フォーマット 5-5
 RSC (報告ソース・コード) 17-19
 RSCH (RESUME SUBCHANNEL) 命令 14-10
 RSE 命令フォーマット 5-5
 RSI 命令フォーマット 5-5
 RSID (報告ソース ID) 17-21
 RSTKO (実スペース・トークン起点) 3-27
 RSX (領域第 2 指標) 3-26
 RTE (領域テーブル・エントリー) 3-30
 RTO (領域テーブル起点) 3-27
 RTX (領域第 3 指標) 3-26
 RX 命令フォーマット 5-5
 RX (領域指標) 3-26
 RXE 命令フォーマット 5-5
 RXF 命令フォーマット 5-5

S

S 命令フォーマット 5-5
 S (SUBTRACT) 2 進数命令 7-141
 SAC (SET ADDRESS SPACE CONTROL) 命令 10-81
 SACF (SET ADDRESS SPACE CONTROL FAST) 命令
 10-81
 SAL (SET ADDRESS LIMIT) 命令 14-11
 SAM24 (SET ADDRESSING MODE) 命令 7-129
 SAM31 (SET ADDRESSING MODE) 命令 7-129
 SAM64 (SET ADDRESSING MODE) 命令 7-129
 SAR (SET ACCESS) 命令 7-129
 SASCE (2 次アドレス・スペース制御エレメント) 3-29
 SASN (2 次アドレス・スペース番号) 3-16
 トレース・エントリー内の 4-22
 SCHIB (サブチャネル情報ブロック) 15-1
 オペランドとしての
 MODIFY SUBCHANNEL 14-7
 STORE SUBCHANNEL 14-17
 サブチャネル状況ワード (SCSW) 15-7
 パス管理制御ワード (PMCW) 15-2
 変更可能フィールドの要約 15-7
 モデル依存区域 15-7
 SCHM (SET CHANNEL MONITOR) 命令 14-12
 SCK (SET CLOCK) 命令 10-83

SCKC (SET CLOCK COMPARATOR) 命令 10-84
 SCKPF (SET CLOCK PROGRAMMABLE FIELD) 命令
 10-84
 SCP 開始リセット 1-12
 SCSW 内のサブチャネル制御情報 16-11
 SCSW 内のバス非稼働ビット (N) 16-12
 SCSW (サブチャネル状況ワード) 16-6
 カウント 16-28
 活動制御フィールド 16-13
 機能制御フィールド 16-12
 サブチャネル状況フィールド 16-23
 サブチャネル制御フィールド 16-11
 状況制御フィールド 16-16
 装置状況フィールド 16-22
 CCW アドレス 16-18
 IRB の中の 16-6
 SCHIB 15-7
 SD (SUBTRACT NORMALIZED) HFP 命令 18-19
 SDB (SUBTRACT) BFP 命令 19-48
 SDBR (SUBTRACT) BFP 命令 19-48
 SDR (SUBTRACT NORMALIZED) HFP 命令 18-19
 SE (SUBTRACT NORMALIZED) HFP 命令 18-19
 SEARCH STRING 命令 7-128
 例 A-29
 SEB (SUBTRACT) BFP 命令 19-48
 SEBR (SUBTRACT) BFP 命令 19-48
 SER (SUBTRACT NORMALIZED) HFP 命令 18-19
 SET ACCESS 命令 7-129
 SET ADDRESS LIMIT 命令 14-11
 SET ADDRESS SPACE CONTROL 命令 10-81
 SET ADDRESS SPACE CONTROL FAST 命令 10-81
 SET ADDRESSING MODE 命令 7-129
 SET CHANNEL MONITOR 命令 14-12
 測定モードへの影響 17-1
 SET CLOCK 命令 10-83
 SET CLOCK COMPARATOR 命令 10-84
 SET CLOCK PROGRAMMABLE FIELD 命令 10-84
 SET CPU TIMER 命令 10-84
 SET FPC 命令 19-46
 SET PREFIX 命令 10-85
 SET PROGRAM MASK 命令 7-130
 SET PSW KEY FROM ADDRESS 命令 10-85
 SET ROUNDING MODE (SRNM) 19-47
 SET SECONDARY ASN 命令 10-86
 アクセス・レジスター 5-40
 SET STORAGE KEY EXTENDED 命令 10-89
 SET SYSTEM MASK 命令 10-89
 SFPC (SET FPC) 命令 19-46
 SG (SUBTRACT) 2 進数命令 7-141

SGF (SUBTRACT) 2 進数命令 7-141
 SGFR (SUBTRACT) 2 進数命令 7-141
 SGR (SUBTRACT) 2 進数命令 7-141
 SH (SUBTRACT HALFWORD) 命令 7-142
 SHIFT AND ROUND DECIMAL 命令 8-11
 例 A-37
 SHIFT LEFT DOUBLE 命令 7-130
 例 A-29
 SHIFT LEFT DOUBLE LOGICAL 命令 7-131
 SHIFT LEFT SINGLE 命令 7-132
 例 A-30
 SHIFT LEFT SINGLE LOGICAL 命令 7-132
 SHIFT RIGHT DOUBLE 命令 7-133
 SHIFT RIGHT DOUBLE LOGICAL 命令 7-133
 SHIFT RIGHT SINGLE 命令 7-134
 SHIFT RIGHT SINGLE LOGICAL 命令 7-135
 SI 命令フォーマット 5-5
 SIGNAL PROCESSOR 命令 10-89
 状況 4-52
 指令 4-48
 SIGP
 参照 : SIGNAL PROCESSOR 命令
 SIGP (SIGNAL PROCESSOR) 命令 10-89
 SL (SUBTRACT LOGICAL) 命令 7-142
 SLA (SHIFT LEFT SINGLE) 命令 7-132
 例 A-30
 SLAG (SHIFT LEFT SINGLE) 命令 7-132
 SLB (SUBTRACT LOGICAL WITH BORROW) 命令
 7-143
 SLBG (SUBTRACT LOGICAL WITH BORROW) 命令
 7-143
 SLBGR (SUBTRACT LOGICAL WITH BORROW) 命令
 7-143
 SLBR (SUBTRACT LOGICAL WITH BORROW) 命令
 7-143
 SLDA (SHIFT LEFT DOUBLE) 命令 7-130
 例 A-29
 SLDL (SHIFT LEFT DOUBLE LOGICAL) 命令 7-131
 SLGR (SUBTRACT LOGICAL) 命令 7-142
 SLL (SHIFT LEFT SINGLE LOGICAL) 命令 7-132
 SLLG (SHIFT LEFT SINGLE LOGICAL) 命令 7-133
 SLR (SUBTRACT LOGICAL) 命令 7-142
 SNaN (信号付き NaN) 19-7
 SP (SUBTRACT DECIMAL) 命令 8-12
 SPKA (SET PSW KEY FROM ADDRESS) 命令 10-85
 SPM (SET PROGRAM MASK) 命令 7-130
 SPT (SET CPU TIMER) 命令 10-84
 SPX (SET PREFIX) 命令 10-85
 SQD (SQUARE ROOT) HFP 命令 18-18
 SQDB (SQUARE ROOT) BFP 命令 19-47
 SQDBR (SQUARE ROOT) BFP 命令 19-47
 SQDR (SQUARE ROOT) HFP 命令 18-18
 SQE (SQUARE ROOT) HFP 命令 18-18
 SQEB (SQUARE ROOT) BFP 命令 19-47
 SQEBR (SQUARE ROOT) BFP 命令 19-47
 SQER (SQUARE ROOT) HFP 命令 18-18
 SQUARE ROOT BFP 命令 19-47
 SQUARE ROOT HFP 命令 18-18
 SQXBR (SQUARE ROOT) BFP 命令 19-47
 SQXR (SQUARE ROOT) HFP 命令 18-18
 SR (SUBTRACT) 2 進数命令 7-141
 SRA (SHIFT RIGHT SINGLE) 命令 7-134
 SRAG (SHIFT RIGHT SINGLE) 命令 7-134
 SRDA (SHIFT RIGHT DOUBLE) 命令 7-133
 SRDL (SHIFT RIGHT DOUBLE LOGICAL) 命令 7-133
 SRL (SHIFT RIGHT SINGLE LOGICAL) 命令 7-135
 SRLG (SHIFT RIGHT SINGLE LOGICAL) 命令 7-135
 SRNM (SET ROUNDING MODE) 19-47
 SRP (SHIFT AND ROUND DECIMAL) 命令 8-11
 例 A-37
 SRSD (2 次実スペース指定) 3-29
 SRST (SEARCH STRING) 命令 7-128
 例 A-29
 SRTD (2 次領域テーブル指定) 3-29
 SRTL (2 次領域テーブル長さ) 3-29
 SS 命令フォーマット 5-5
 SSAR (SET SECONDARY ASN) 命令 10-86
 アクセス・レジスター 5-40
 SSAR-cp (SET SECONDARY ASN 命令、現 1 次へ
 の) 10-86
 SSAR-ss (SET SECONDARY ASN 命令、スペース切り替え
 を伴う) 10-86
 SSASTEO (サブスペース AST エントリー起点) 5-56
 SSASTESN (サブスペース AST エントリー・シーケンス番
 号) 5-57
 SSCH (START SUBCHANNEL) 命令 14-14
 SSE 命令フォーマット 5-5
 SSKE (SET STORAGE KEY EXTENDED) 命令 10-89
 SSM 抑止制御ビット 6-30, 10-89
 SSM (SET SYSTEM MASK) 命令 10-89
 SSTD (2 次セグメント・テーブル指定) 3-29
 SSSL (2 次セグメント・テーブル長さ) 3-29
 SSTO (2 次セグメント・テーブル起点) 3-29
 SSTO (2 次領域テーブル起点) 3-29
 ST (STORE) 2 進数命令 7-135
 STAM (STORE ACCESS MULTIPLE) 命令 7-135
 STAP (STORE CPU ADDRESS) 命令 10-91
 START SUBCHANNEL 命令 14-14
 開始される機能 15-17
 使用される操作要求ブロック (ORB) 15-20

START SUBCHANNEL 命令 (続き)
 据え置き条件コード (SCSW の) 16-8
 測定ブロックのカウント 17-3
 STC (STORE CHARACTER) 命令 7-136
 STCK (STORE CLOCK) 命令 7-137
 STCKC (STORE CLOCK COMPARATOR) 命令 10-91
 STCKE (STORE CLOCK EXTENDED) 命令 7-137
 STCM (STORE CHARACTERS UNDER MASK) 命令
 7-136
 例 A-30
 STCMH (STORE CHARACTERS UNDER MASK) 命令
 7-136
 STCPS (STORE CHANNEL PATH STATUS) 命令 14-15
 STCRW (STORE CHANNEL REPORT WORD) 命令
 14-16
 STCTG (STORE CONTROL) 命令 10-91
 STCTL (STORE CONTROL) 命令 10-91
 STD
 参照：セグメント・テーブル指定
 STD (STORE) 浮動小数点命令 9-11
 STE (STORE) 浮動小数点命令 9-11
 STFL 機能リスト 3-52
 STFL (STORE FACILITY LIST) 命令 10-93
 STFPC (STORE FPC) 命令 19-47
 STG (STORE) 2 進数命令 7-135
 STH (STORE HALFWORD) 命令 7-139
 STIDP (STORE CPU ID) 命令 10-92
 STL (セグメント・テーブル長さ) 3-27
 STM (STORE MULTIPLE) 命令 7-139
 例 A-31
 STMG (STORE MULTIPLE) 命令 7-139
 STMH (STORE MULTIPLE HIGH) 命令 7-140
 STNSM (STORE THEN AND SYSTEM MASK) 命令
 10-105
 STO (セグメント・テーブル起点) 3-27
 STORE 浮動小数点命令 9-11
 STORE ACCESS MULTIPLE 命令 7-135
 STORE CHANNEL PATH STATUS 命令 14-15
 STORE CHANNEL REPORT WORD 命令 14-16
 格納されるチャンネル報告ワード (CRW) 17-19
 STORE CHARACTER 命令 7-136
 STORE CHARACTERS UNDER MASK 命令 7-136
 例 A-30
 STORE CLOCK 命令 7-137
 STORE CLOCK COMPARATOR 命令 10-91
 STORE CLOCK EXTENDED 命令 7-137
 STORE CONTROL 命令 10-91
 STORE CPU ADDRESS 命令 10-91
 STORE CPU ID 命令 10-92
 STORE CPU TIMER 命令 10-93
 STORE FACILITY LIST 命令 10-93
 STORE FPC 命令 19-47
 STORE HALFWORD 命令 7-139
 STORE MULTIPLE 命令 7-139
 例 A-31
 STORE MULTIPLE HIGH 命令 7-140
 STORE PAIR TO QUADWORD 命令 7-140
 STORE PREFIX 命令 10-93
 STORE REAL ADDRESS 命令 10-94
 STORE REVERSED 命令 7-140
 STORE SUBCHANNEL 命令 14-17
 STORE SYSTEM INFORMATION 命令 10-95
 STORE THEN AND SYSTEM MASK 命令 10-105
 STORE THEN OR SYSTEM MASK 命令 10-105
 STORE USING REAL ADDRESS 命令 10-105
 STORE 2 進数命令 7-135
 STOSM (STORE THEN OR SYSTEM MASK) 命令
 10-105
 STPQ (STORE PAIR TO QUADWORD) 命令 7-140
 STPT (STORE CPU TIMER) 命令 10-93
 STPX (STORE PREFIX) 命令 10-93
 STRAG (STORE REAL ADDRESS) 命令 10-94
 STRV (STORE REVERSED) 命令 7-140
 STRVG (STORE REVERSED) 命令 7-140
 STRVH (STORE REVERSED) 命令 7-140
 STSCH (STORE SUBCHANNEL) 命令 14-17
 STSI (STORE SYSTEM INFORMATION) 命令 10-95
 STURA (STORE USING REAL ADDRESS) 命令 10-105
 STURG (STORE USING REAL ADDRESS) 命令 10-105
 SU (SUBTRACT UNNORMALIZED) HFP 命令 18-20
 SUBTRACT BFP 命令 19-48
 SUBTRACT DECIMAL 命令 8-12
 SUBTRACT HALFWORD 命令 7-142
 SUBTRACT LOGICAL 命令 7-142
 SUBTRACT LOGICAL WITH BORROW 命令 7-143
 SUBTRACT NORMALIZED
 参照：SUBTRACT BFP 命令
 SUBTRACT NORMALIZED HFP 命令 18-19
 SUBTRACT UNNORMALIZED HFP 命令 18-20
 SUBTRACT 2 進数命令 7-141
 SUPERVISOR CALL 命令 7-144
 SUR (SUBTRACT UNNORMALIZED) HFP 命令 18-20
 SVC (SUPERVISOR CALL) 命令 7-144
 SW (SUBTRACT UNNORMALIZED) HFP 命令 18-20
 SWR (SUBTRACT UNNORMALIZED) HFP 命令 18-20
 SX (セグメント指標) 3-26
 SXBR (SUBTRACT) BFP 命令 19-48
 SXR (SUBTRACT NORMALIZED) HFP 命令 18-19

T

T (テラ) xx

TAM (TEST ADDRESSING MODE) 命令 7-144
TAR (TEST ACCESS) 命令 10-106
TB (TEST BLOCK) 命令 10-108
TBDER (CONVERT HFP TO BFP) 浮動小数点命令
TBDR (CONVERT HFP TO BFP) 浮動小数点命令 9-9
TBEDR (CONVERT HFP TO BFP) 浮動小数点命令 9-9
TCDB (TEST DATA CLASS) BFP 命令 19-48
TCEB (TEST DATA CLASS) BFP 命令 19-48
TCXB (TEST DATA CLASS) BFP 命令 19-48
TEST ACCESS 命令 10-106
TEST ADDRESSING MODE 命令 7-144
TEST AND SET 命令 7-144
TEST BLOCK 命令 10-108
TEST DATA CLASS BFP 命令 19-48
TEST DECIMAL 命令 8-12
TEST PENDING INTERRUPTION 命令 14-17, 14-18
TEST PROTECTION 命令 10-110
TEST SUBCHANNEL 命令 14-19
 使用される割り込み応答ブロック (IRB) 16-6
TEST UNDER MASK 命令 7-145
 例 A-31
TEST UNDER MASK HIGH 命令 7-145
TEST UNDER MASK LOW 命令 7-145
THDER (CONVERT BFP TO HFP) 浮動小数点命令 9-8
THDR (CONVERT BFP TO HFP) 浮動小数点命令 9-8
TIC (チャンネル内転送) 15-39
 無効なシーケンス 16-25
TLB (変換索引バッファ) 3-41
 エントリー 3-42
 使用可能状態 3-43
 消去 3-44
 変換変更の影響 3-44
 連結 3-42
TM (TEST UNDER MASK) 命令 7-145
 例 A-31
TMH (TEST UNDER MASK HIGH) 命令 7-145
TMHH (TEST UNDER MASK) 命令 7-145
TMHL (TEST UNDER MASK) 命令 7-145
TML (TEST UNDER MASK LOW) 命令 7-145
TMLH (TEST UNDER MASK) 命令 7-145
TMLL (TEST UNDER MASK) 命令 7-145
TOD エポックで使用される UTC (協定世界時) 4-36
TOD エポックで使用される協定世界時 (UTC) 4-36
TOD クロック 4-34
 クロック・コンパレーター割り込みへの影響 6-11
 固有値 4-35
 手動制御 4-34, 12-5
 トレース・エントリー内の値 4-23

TOD クロック (続き)

 パワーオン・リセットの影響 4-45
 有効化 11-9
 CPU タイマーの減少への影響 4-39
 CPU タイマー割り込みへの影響 6-11
TOD クロック制御オーバーライド制御 4-34
TOD クロック同期化機能 4-38
TOD クロック同期制御ビット 4-34, 4-38
TOD クロックのプログラマブル・フィールド 4-36
TOD クロック・プログラマブル・フィールド 4-36
TOD クロック・プログラマブル・レジスター 4-36
 妥当性ビット 11-21
 保管域 3-53
TP (TEST DECIMAL) 命令 8-12
TPI (TEST PENDING INTERRUPTION) 命令 14-17,
14-18
 格納される割り込みコード 14-18
TPROT (TEST PROTECTION) 命令 10-110
TR (TRANSLATE) 命令 7-146
 例 A-32
TRACE 命令 10-113
 トレース・エントリー 4-22
TRACG (TRACE) 命令 10-113
TRANSLATE 命令 7-146
 例 A-32
TRANSLATE AND TEST 命令 7-147
 例 A-32
TRANSLATE EXTENDED 命令 7-148
TRANSLATE ONE TO ONE 命令 7-150
TRANSLATE ONE TO TWO 命令 7-150
TRANSLATE TWO TO ONE 命令 7-150
TRANSLATE TWO TO TWO 命令 7-150
TRAP 命令 10-114
TRAP2 (TRAP) 命令 10-114
TRAP4 (TRAP) 命令 10-114
TRE (TRANSLATE EXTENDED) 命令 7-148
TROO (TRANSLATE ONE TO ONE) 命令 7-150
TROT (TRANSLATE ONE TO TWO) 命令 7-150
TRT (TRANSLATE AND TEST) 命令 7-147
 例 A-32
TRTO (TRANSLATE TWO TO ONE) 命令 7-150
TRTT (TRANSLATE TWO TO TWO) 命令 7-150
TS (TEST AND SET) 命令 7-144
TSCH (TEST SUBCHANNEL) 命令 14-19

U

ulp (最終位置単位) 19-5
UNPACK 命令 7-155
 例 A-34

UNPACK ASCII 命令 7-155
UNPACK UNICODE 命令 7-156
UNPK (UNPACK) 命令 7-155
例 A-34
UNPKA (PACK ASCII) 命令 7-155
UNPKU (PACK UNICODE) 命令 7-156
UPDATE TREE 命令 7-157
例 A-51
UPT (UPDATE TREE) 命令 7-157
例 A-51

V

VM データ・スペース機能 1-12
VTAM による記号変換の使用 7-64

W

WAIT (SVC)
バイパスするルーチンの例 A-46

X

X (EXCLUSIVE OR) 命令 7-77
XA (拡張アーキテクチャー)
参照: 370-XA アーキテクチャー
XC (EXCLUSIVE OR) 命令 7-77
例 A-20
XG (EXCLUSIVE OR) 命令 7-77
XGR (EXCLUSIVE OR) 命令 7-77
XI (EXCLUSIVE OR) 命令 7-77
例 A-20
XR (EXCLUSIVE OR) 命令 7-77
XSCH (CANCEL SUBCHANNEL) 命令 14-4

Z

Z ビット (0 条件コード・ビット) 16-11
中間割り込み条件の原因としての 16-16
ZAP (ZERO AND ADD) 命令 8-13
例 A-38
ZERO AND ADD 命令 8-13
例 A-38
z/Architecture アーキテクチャー
特長 1-1

数字

0
真の (HFP 数) 18-1
負
参照: 負の 0

0 (続き)
浮動小数点レジスターの設定 9-11
命令長コード 6-7
0 条件コード (SCSW の Z ビット) 16-11
0 (ゼロ)
負
参照: 負の 0
1 次 ASN (PASN) 3-16
リンケージ・スタック状態エントリー内の 5-71
1 次 AST エントリー (PASTE)
起点 (PASTEO) 5-28, 5-42
1 次アドレス・スペース 3-15
1 次仮想アドレス 3-4
有効アドレス・スペース制御エレメント 3-33
1 次記憶機構変更イベント制御ビット 3-28
1 次権限 3-23
例外 6-26
1 次サブスペース・グループ制御ビット 3-28
1 次サブスペース・グループ制御ビット
1 次実スペース制御ビット 3-28
1 次実スペース・トークン起点 (PRSTKO) 3-29
1 次指定タイプ制御ビット 3-28
1 次状況ビット (入出力) 16-17
1 次スペース切り替えイベント制御ビット 3-28
1 次スペース・アクセス・リスト指定 (PSALD) 5-45
1 次スペース・モード 3-27
1 次セグメント・テーブル
起点 (PSTO) 3-28
指定 (PSTD) 3-27
長さ (PSTL) 3-29
1 次専用スペース制御ビット 3-28
1 次リスト・ビット 5-43
1 次領域テーブル
起点 (PRTO) 3-28
指定 (PRTD) 3-27
長さ (PRTL) 3-29
1 次割り込み条件 (入出力) 16-4
1 進浮動小数点 (HFP)
変換
基本的な例 A-6
1 の補数の 2 進表記 7-3
SUBTRACT LOGICAL 命令に使用される 7-142
SUBTRACT LOGICAL WITH BORROW 命令に使用さ
れる 7-143
10 進オペランド・データ例外 6-20, 8-4
10 進数
演算 8-2
オーバーフロー
例外 6-21
PSW 内のマスク 4-6
オペランド・オーバーラップ 8-3

- 10 進数 (続き)
 - 数の表現
 - 例 A-4
 - 除算例外 6-21
 - 数字コード 8-2
 - 数表現 8-1
 - 比較 8-6
 - 符号コード 8-2
 - 命令 8-1
 - 例 A-34
 - 10 進数から 2 進数への変換 7-68
 - 例 A-18
 - 16 進数
 - 参照: 16 進数
 - 16 進数 (hex) 表現 5-5
 - 16 進浮動小数点
 - 変換
 - 命令による変換の例 A-42
 - 命令
 - 例 A-39
 - 16 進浮動小数点 (HFP) 9-1
 - 16 進浮動小数点数
 - 例 A-5
 - 2次 ASN (SASN)
 - 制御ビット 5-65
 - リンケージ・スタック状態エントリ内の 5-71
 - 2 次 ASN (SASN) 3-16
 - 2 次 CCW アドレス妥当性 (ERW 内の) 16-36
 - 2 次アドレス・スペース 3-15
 - 2 次エラー (サブチャネル・ログアウト内の) 16-34
 - 2 次仮想アドレス 3-4
 - 有効アドレス・スペース制御エレメント 3-33
 - 2 次記憶機構変更イベント制御ビット 3-29
 - 2 次権限 3-23
 - 例外 6-29
 - 2 次サブスペース・グループ制御ビット 3-29
 - 2 次実スペース制御ビット 3-29
 - 2 次実スペース・トークン起点 (SRSTKO) 3-29
 - 2 次指定タイプ制御ビット 3-29
 - 2 次状況ビット (入出力) 16-17
 - 2 次スペース制御ビット 3-27, 5-24
 - 2 次スペース・モード 3-27
 - 2 次セグメント・テーブル
 - 起点 (SSTO) 3-29
 - 指定 (SSTD) 3-29
 - 長さ (SSTL) 3-29
 - 2 次専用スペース制御ビット 3-29
 - 2 次領域テーブル
 - 起点 (SSTO) 3-29
 - 指定 (SRTD) 3-29
 - 長さ (SRTL) 3-29
- 2 次割り込み条件 (入出力) 16-4
- 2 進数
 - 演算 7-4
 - 例 A-2
 - オーバーフロー 7-4
 - 例 A-2
 - 数の表現
 - 例 A-2
 - 数表現 7-3
 - 符号ビット 7-3
 - 負の 0 7-3
 - 参照: 固定小数点
- 2 進数から 10 進数への変換 7-68
 - 例 A-19
- 2 進整数
 - 浮動小数点からの変換 18-10, 19-28
 - 浮動小数点への変換 18-10, 19-27
- 2 進浮動小数点 (BFP) 9-1
- 2 の補数の 2 進表記 7-3
 - 例 A-2
- 2 の累乗数
 - 表 D-1
- 2 の累乗数の表 D-1
- 2 モード・アドレッシング
 - 参照: アドレッシング・モード
- 2K IDAW 制御 15-23
- 3 モード・アドレッシング 5-7
- 370-XA アーキテクチャー 1-13



Printed in Japan

日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12



SA88-8773-00