

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

RENESAS

ユーザーズ・マニュアル

**保守/廃止**

## 16ビットVシリーズ™

16/8, 16ビット・マイクロプロセッサ

命令編

---

V20™, V30™

V20HL™, V30HL™

V40™, V50™

V40HL™, V50HL™

V33A™

V53A™

資料番号 U11301JU5V1UMJ1 (第5版)  
発行年月 September 2000 N CP(K)

© NEC Corporation 1992

---

(※ 注)

---

**目次要約**

**第1章 概 説 … 1**

**第2章 命 令 … 7**

**付録A レジスタ構成 … 205**

**付録B アドレッシング・モード … 213**

**付録C 命令マップ … 221**

**付録D  $\mu$ PD8086, 8088とのニモニック対応表 … 225**

**付録E 命令索引 (ニモニック：機能別) … 227**

**付録F 命令索引 (ニモニック：アルファベット順) … 231**



## CMOSデバイスの一般的注意事項

## ①静電気対策 (MOS全般)

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレーやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

## ②未使用入力の処理 (CMOS特有)

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性 (タイミングは規定しません) を考慮すると、個別に抵抗を介してV<sub>DD</sub>またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

## ③初期化以前の状態 (MOS全般)

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

- 本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。
  - 文書による当社の承諾なしに本資料の転載複製を禁じます。
  - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
  - 本資料に記載された図路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら図路・ソフトウェア・情報をお客様の機密に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。
  - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
  - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
    - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
    - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災/防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
    - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート/データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

**本版で改訂された主な箇所**

箇 所	内 容
全 般	以下の製品を削除 ● $\mu$ PD70208 (A) (V40) ● $\mu$ PD70216 (A) (V50) ● $\mu$ PD70270 (V41 <sup>TM</sup> ) ● $\mu$ PD70280 (V51 <sup>TM</sup> )

本文欄外の★印は、本版で改訂された箇所を示しています。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽に寄せてください。



## はじめに

**対象者** このマニュアルは、次の16ビットVシリーズ製品の機能を理解し、それを用いたアプリケーション・システムを設計するユーザを対象とします。

製品名	別名称
μFD70108	V20
μFD70116	V30
μFD70108H	V20HL
μFD70116H	V30HL
μPD70208	V40
μPD70216	V50
μPD70208H	V40HL
μPD70216H	V50HL
μPD70136A	V33A
μPD70236A	V33A

**目的** このマニュアルは、上記16ビットVシリーズ製品の持つ各種命令機能をユーザに理解していただくことを目的とします。

**構成** 上記16ビットVシリーズ製品のユーザズ・マニュアルは、ハードウェア編と命令編(このマニュアル)の2冊に分かれています。

### ハードウェア編

- 概説
- 端子機能
- CPU機能
- 内部ブロック機能
- バス制御機能
- 割り込み機能
- スタンバイ機能
- リセット機能
- その他

### 命令編

- 概説
- 命令の説明
- 命令マップ
- μPD8086, 8088とのニモニック対応表

読み方 このマニュアルの読者には、電気、論理回路、マイクロコンピュータの一般知識を必要とします。特に断りのないかぎり、各製品に共通の説明です。  
なお、このマニュアルでは「μPD70…」という製品名を、「V…」の名称に統一して説明してあります。

二モニックが分かっていて、命令機能の詳細を確認するとき

→ 第2章 命令を参照してください（二モニックのアルファベット順に記載してあります）。

一通り命令機能の詳細を理解しようとするとき

→ 目次に従って読んでください。

各製品のハードウェア機能を理解しようとするとき

→ 各製品のユーザーズ・マニュアル ハードウェア編（別冊）を参照してください。

電気的特性を知りたいとき

→ 各デバイスのデータ・シートを参照してください。

- 凡 例
- データ表記の重み : 左側が上位桁、右側が下位桁
  - アクティブ・ロウの表記 :  $\overline{XXXX}$  (端子、信号名称に上線)
  - メモリ・マップのアドレス : 上側-上位、下側-下位
  - アドレスの表記 : 次の場合、 $x$ がセグメント値、 $y$ がオフセット値を表します。  
 $x:yH$
  - 注 : 本文中につけた注の説明
  - 注 意 : 気をつけて読んでいただきたい内容
  - 備 考 : 本文の補足説明
  - 数の表記 : 2進数… $XXXX$ または $XXXXB$   
10進数… $XXXX$   
16進数… $XXXXH$
-

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

資料名 品名	データシート	ユーザーズ・マニュアル		アプリケーション・ ノート	レジスタ 活用表	Q&A集
		ハードウェア編	命令編			
V20	IC-6659	IEM-871	この マニュアル	-	-	IEI-642
V30	IC-6660			-	-	-
V20HL	IC-8154	U12858J		U12281J	-	-
V30HL				ソフトウェア編	-	-
V40	U10184J	U10666J		U10911J	-	U10584J
V50				ソフトウェア編	-	-
V40HL	IC-8221	U11610J		U10037J	IEM-8875	U11123J
V50HL				ハードウェア設計編		
V33A	U10136J	U10032J		-	-	-
V53A	U10120J	U10108J		IEA-752 ハードウェア設計編 U10188J アドレス拡張 ソフトウェア編	IEM-8604	U10875J

( 号 )

---

## 目 次

- 第1章 概 説 … 1
  - 1.1 機能別命令一覧 … 2
  - 1.2 命令書の形式 … 3
  - 1.3 命令の概要 … 3
    - 1.3.1 データ転送 … 3
    - 1.3.2 ブロック操作 … 3
    - 1.3.3 ビット・フィールド操作 … 3
    - 1.3.4 入出力 … 4
    - 1.3.5 演 算 … 4
    - 1.3.6 BCD演算 … 4
    - 1.3.7 BCD補正 … 5
    - 1.3.8 データ変換 … 5
    - 1.3.9 ビット操作 … 5
    - 1.3.10 シフト、ローテート … 5
    - 1.3.11 スタック操作 … 6
    - 1.3.12 プログラム分岐 … 6
    - 1.3.13 CPU制御 … 6
    - 1.3.14 モード切り替え … 6
  
- 第2章 命 令 … 7
  - 2.1 命令の説明（ニモニックのアルファベット順） … 7
  - 2.2 命令実行クロック数 … 188
  
- 付録A レジスタ構成 … 205
  - A.1 汎用レジスタ (AW, BW, CW, DW) … 205
  - A.2 セグメント・レジスタ (PS, SS, DSO, DS1) … 205
  - A.3 ポインタ (SP, BP) … 206
  - A.4 プログラム・カウンタ (PC) … 206
  - A.5 プログラム・ステータス・ワード (PSW) … 206
  - A.6 インデクス・レジスタ (IX, IY) … 211
  
- 付録B アドレッシング・モード … 213
  - B.1 命令アドレス … 213
  - B.2 メモリ・オペランド・アドレス … 215
  
- 付録C 命令マップ … 221
  
- 付録D  $\mu$ PDS086, 8088とのニモニック対応表 … 225

付録E 命令索引 (ニモニック：機能別) … 227

付録F 命令索引 (ニモニック：アルファベット順) … 231

## 図 の 目 次

図番号	タイトル, ページ
1-1	共通命令と各製品の専用命令の関係 … 1
1-2	命令語形式 … 3
1-3	演算命令でのALUの働き … 4
2-1	記述例 … 13
A-1	PSWの構成 … 207

## 表 の 目 次

表番号	タイトル, ページ
1-1	機能別命令一覧 … 2
2-1	フラグ動作の凡例 … 7
2-2	オペランド・タイプの凡例 … 8
2-3	命令語の凡例 … 9
2-4	命令形式またはオペレーション説明上の凡例 … 10
2-5	メモリ・アドレッシング … 11
2-6	8/16ビット汎用レジスタの選択 … 12
2-7	セグメント・レジスタの選択 … 12
2-8	命令実行クロック数一覧 … 190
C-1	命令マップ … 222
C-2	Group1, Group2, Imm, Shiftコード表 … 224
C-3	Group0コード表 … 224
C-4	Group3コード表 … 224
D-1	μPD8086, 8088とのレジスタ対応表 … 225
D-2	μPD8086, 8088とのモニタック対応表 … 226

( 4 号 )

---



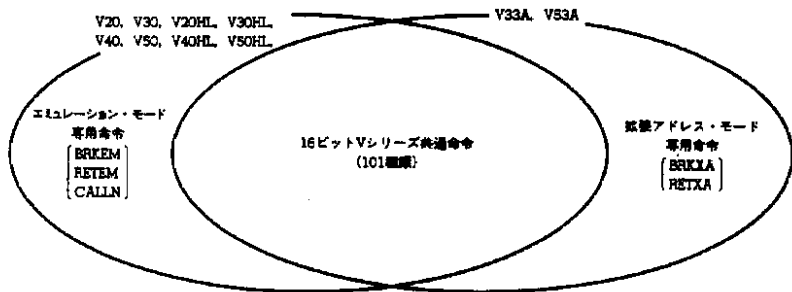
# 第1章 概 説

16ビットVシリーズには、完全なソフトウェア互換性を持つ101種類の共通命令があり、ソフトウェア資産の有効活用ができます。

★ この共通命令に対し、V20, V30, V20HL, V30HL, V40, V50, V40HL, V50HLにはエミュレーション・モード (μPD9090AFの命令を実行) をサポートするための3種類の専用命令 (BRKEM, RETEM, CALLN) が追加されています。

また、V33A, V53Aには拡張アドレス・モードをサポートするための2種類の専用命令 (BRKXA, RETXA) が追加されています。

★ 図 1-1 共通命令と各製品の専用命令の関係



備考 エミュレーション・モードと拡張アドレス・モードについては各製品のユーザーズ・マニュアルハードウェア編を参照してください。

### 1.1 機能別命令一覧

16ビットVシリーズの各命令を機能別に大別すると次の27種類になります。

表 1-1 機能別命令一覧

命 令 群	ニモニック (アルファベット順)
データ転送命令	LDEA, MOV, TRANS, TRANSB, XCH
リピート・プリフィクス	REP, REPC, REPE, REPNC, REPNE, REPNZ, REpz
プリミティブ・ブロック転送命令	CMPBK, CMPBKb, CMPBKw, CMPM, CMPMb, CMPMw, LDM, LDmB, LDmW, MOVbK, MOVbKb, MOVbKw, STM, STmB, STmW
ビット・フィールド操作命令	EXT, INS
入出力命令	IN, OUT
プリミティブ入出力命令	INM, OUTM
加減算命令	ADD, ADDC, SUB, SUBC
BCD演算命令	ADD4S, CMP4S, ROL4, ROR4, SUB4S
増減命令	DEC, INC
乗除算命令	DIV, DIVU, MUL, MULU
BCD補正命令	ADJ4A, ADJ4S, ADJbA, ADJbS
データ変換命令	CVTbD, CVTbW, CVTDb, CVTWL
比較命令	CMP
補数演算命令	NEG, NOT
論理演算命令	AND, OR, TEST, XOR
ビット操作命令	CLR1, NOT1, SET1, TEST1
シフト命令	SHL, SHR, SHRA
ローテート命令	ROL, ROLC, ROR, RORC
サブルーチン制御命令	CALL, RET
スタック操作命令	DISPOSE, POP, PREPARE, PUSH
ブランチ命令	BR
条件付きブランチ命令	BC, BCWZ, BE, BGE, BGT, BH, BL, BLE, BLT, BN, BNC, BNE, BNH, BNL, BNv, BNZ, BP, BPE, BPO, BZ, BV, DBNZ, DBNZE, DBNZNE
割り込み命令	BRK, BRKV, CHKIND, RETI
CPU制御命令	BUSLOCK, DI, EI, FPO1, FPO2, HALT, NOP, POLL
セグメント・オーバライド・プリフィクス	DS0:, DS1:, PS:, SS:
エミュレーション・モード専用命令 <sup>注1</sup>	BRKEM, CALLN, RETEM
拡張アドレス・モード専用命令 <sup>注2</sup>	BRKXA, RETXA

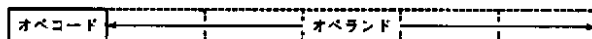
注1. V33A, V53Aを除く。

注2. V33A, V53Aだけ。

## 1.2 命令語の形式

命令語（オブジェクト・コード）は基本的に次の形式で表されます。

図 1-2 命令語形式



備考 オペコード：命令の種類を表す8ビットのコードです。

オペランド：命令の処理の対象となるレジスタ、メモリ・アドレスを示すフィールドです。

0-5バイトのフィールドで表されます。

## 1.3 命令の概要

### 1.3.1 データ転送

データ転送命令により、データの操作なしで、レジスタとレジスタ間またはレジスタとメモリ間のデータ転送ができます。次の4種の命令に分類できます。

汎用データの転送 (MOV) : 第2オペランドから第1オペランドへのバイト/ワード転送をします。また、レジスタやメモリに直接数値を転送することもできます。

実効アドレスの転送 (LDEA) : 第2オペランドのオフセット・アドレス（実効アドレス）を第1オペランドへ転送します。

変換テーブルの転送 (TRANS) : 変換テーブルの1バイトを転送します。

汎用データの交換 (XCH) : 第1オペランドと第2オペランドの内容を交換します。

### 1.3.2 ブロック操作

リピート・プリフィクスとプリミティブ・ブロック転送命令によりバイトまたはワードのブロック（連続したデータ）に対する転送や比較ができます。

プリミティブ・ブロック転送命令には、アキュムレータとの間で行うブロック単位の転送に対する命令と同様、転送、ある値との比較、定数に対しての命令があります。また、1バイトのリピート・プリフィクスを前置するとハードウェアによる反復処理が可能で、連続したデータ操作ができます。

### 1.3.3 ビット・フィールド操作

ビット・フィールド操作命令により連続するメモリをビット・フィールドとみなして、指定した長さのデータを（指定した）ビット・フィールド領域とAWレジスタの間で転送することができます。

この命令は、命令実行終了後にワード・オフセット (IXレジスタまたはIYレジスタ) およびビット・オフセット (8ビット汎用レジスタ) を更新して、連続したビット・フィールド・データを自動的に推定します。コンピュータ・グラフィクスや高級言語に有効で、たとえば、Pascalのバケット・アレイやレコード・タイプのデータ構造に対応できます。

### 1.3.4 入出力

入出力命令、プリミティブ入出力命令によりI/Oデバイスに対してリード/ライトができます。I/Oデバイスは、この命令によりデータ・バスを通じて、CPUとのデータ転送をします。

### 1.3.5 演算

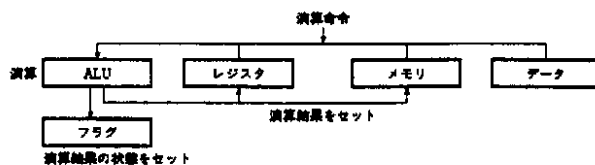
次の命令により8/16ビット・データの演算ができます。

加減算命令、増減命令、乗算命令、除算命令、比較命令、補数演算命令、論理演算命令

また、増減命令により汎用レジスタまたはメモリの8/16ビット・データをインクリメント (+1)/デクリメント (-1) することもできます。

各演算命令は対象となるレジスタやメモリの中で実行されるのではなく、実際は内部のALUで実行され、プログラム・ステータス・ワード (PSW) のフラグに、演算の結果がセット (1)、リセット (0) されます。

図 1-3 演算命令でのALUの働き



### 1.3.6 BCD演算

BCD演算命令により16進数を用いて10進数を表現し、計算することができます。

この命令により、メモリ上のBCDストリングに対して算術演算、比較もできます。

また、BCDストリングに対してのローテートをサポートする命令も含まれています。

演算、比較命令は、使用するレジスタが決まっているため、バケットBCDストリングを指定するオペランドはありません。

ソース・ストリングの先頭アドレス (LSDを含むバイト・データのアドレス) は、データ・セグメント 0 (DS0) 内でIXレジスタの内容で指定します。

デスティネーション・ストリングの先頭アドレス (LSDを含むバイト・データのアドレス) は、データ・セグメント 1 (DS1) 内でIYレジスタの内容で指定します。

桁数は、CLレジスタの内容で指定します。

デスティネーション・ストリングとソース・ストリングは、同一の長さ（桁数）でなければならないので、長さが異なる場合は長い方に合わせて0を拡張します。

### 1.3.7 BCD補正

算術演算の実行前あるいは実行後にBCD補正命令を実行することで、BCD演算をサポートします。

BCD補正命令はALレジスタに対して行うので、オペランドはありません。加算、減算の場合、パックトBCDおよびアンパックトBCDのどちらでも補正できますが、乗除算の場合、アンパックトBCD表現の演算でしか補正できません。

### 1.3.8 データ変換

データ変換命令により2進数と10進数の型変換、長さの変換ができます。

CVTBD命令とCVTDB命令は、2進数と2桁のアンパックトBCDの型変換をする命令です。

CVTBW命令とCVTWL命令は、レジスタ内の符号拡張をする命令です。

### 1.3.9 ビット操作

ビット操作命令により汎用レジスタあるいはメモリのビット・データに対して、論理演算ができます。

命令形式のオペランドは、“reg, bit”または、“mem, bit”となります。

第1オペランドのregまたはmemは、ビット・データを含んでいる8/16ビット・データを指定するもので、汎用レジスタまたは実効アドレスを記述します。

第2オペランドのbitは、ビット・データのバイト/ワード内アドレスを示すもので、CLの内容または8ビット・イミューディエト・データを使用します。

ただし、regまたはmemが8ビット・データの場合は、下位3ビットのみが有効なビット・アドレスとなり、16ビット・データの場合は、下位4ビットのみが有効なビット・アドレスで、上位ビットは無視されます。

### 1.3.10 シフト、ローテート

シフト命令、ローテート命令により汎用レジスタあるいはメモリの8/16ビット・データを1ビットまたは複数ビット（0-255）、シフトまたはローテートできます。

シフトには、算術シフトと論理シフトがあります。シフトする桁数は通常では1ですが、命令のカウント・オペランドで指定すれば、CLレジスタの値によって実行するたびに変更できます（最大255）。算術シフトは、左に1ビットずらすときにLSBに0が入り、右に1ビットずらすとMSBに0が入ります。論理シフトの場合は、1ビットずらしてもLSBビットまたはMSBの値は変化しません。

ローテートもシフトと同様、ローテートする桁数はカウント・オペランドでCLレジスタに入れた値を指定します。この命令では、CYフラグとVフラグだけが変化します。CYフラグには、常にローテートして最後に押し出されたビットが入ります。Vフラグは、2桁以上のローテートの場合は必ず不定となり、1桁の場合はデスティネーションのMSB（延長）が変化するとセット(1)、変化しなければリセット(0)

されます。CYフラグは、ROLC命令とRORC命令でデスティネーションの延長として使用できます。

### 1.3.11 スタック操作

スタック操作命令によりメモリ上のスタック操作ができます。

スタック操作命令には次の4種の命令があります。

**PUSH命令** : スタックヘデータを透過させる命令です。

**POP命令** : スタックからデータを復帰させる命令です。

**PREPARE命令**: スタック・フレームを生成する命令で、ローカル変数の領域確保と、グローバル変数への参照を可能とするためにフレーム・ポインタをコピーします。

**DISPOSE命令**: スタック・ポインタ (SP) およびベース・ポインタ (BP) をPREPARE命令を実行する直前の状態に戻す命令です。

### 1.3.12 プログラム分岐

指定されたアドレスに分岐できます。

次の4種に大別できます。

**サブルーチン制御命令**: プログラム・カウンタ (PC) の内容をスタックに透過させる命令 (CALL) と、スタックからPCの内容をリストアする命令 (RET) があります。

**ジャンプ命令** : 命令の流れを他のアドレスに移します。

**条件付きジャンプ命令**: フラグの値によって命令の実行の流れを他のアドレスに移します。

**割り込み命令** : 外部デバイスからの割り込み要求や演算エラーが発生した際に、プログラムの実行を一時中止し、ソフトウェア割り込みによりプログラム実行の流れを制御します。

### 1.3.13 CPU制御

CPU制御命令により、フラグの操作や、外部デバイスとの同期、データ転送ができます。また、CPUに何もさせない命令 (NOP) もあります。

### 1.3.14 モード切り替え

#### (1) エミュレーション・モード (V33A, V53Aを除く)

エミュレーション・モード専用命令によりネイティブ・モードとエミュレーション・モードの切り替えができます。

#### (2) 拡張アドレス・モード (V33A, V53Aのみ)

拡張アドレス・モード専用命令により通常アドレス・モードと拡張アドレス・モードの切り替えができます。

## 第2章 命 令

### 2.1 命令の説明（ニモニックのアルファベット順）

この節では各命令を次の項目に分けて説明します。

- 【命令形式】
- 【オペレーション】
- 【オペランド】
- 【フラグ】
- 【説明】
- 【記述例】
- 【バイト数】
- 【命令書形式】

【命令形式】、【オペレーション】、【オペランド】の各項目では説明上いくつかの識別子を用いています。表2-2から表2-4に識別子とその意味を、表2-5から表2-7にメモリ・アドレッシング、汎用レジスタ/セグメント・レジスタの選択について示します。

【フラグ】の項目は命令実行により変化するフラグの動作を識別子により示します。各フラグの動作の凡例を表2-1に示します。

表2-1 フラグ動作の凡例

識別子	説 明
空白	変化なし
0	リセット(0)される
1	セット(1)される
×	結果に依ってセット(1)またはリセット(0)される
U	不定
R	以前に通過した値がリストアされる

表 2-2 オペランド・タイプの凡例 (1/2)

識 別 子	説 明
reg	8/16ビット汎用レジスタ (8/16ビット汎用レジスタを2つ用いる命令における、デスティネーション側レジスタ)
reg'	8/16ビット汎用レジスタを2つ用いる命令における、ソース側レジスタ
reg8	8ビット汎用レジスタ (8ビット汎用レジスタを2つ用いる命令における、デスティネーション側レジスタ)
reg8'	8ビット汎用レジスタを2つ用いる命令における、ソース側レジスタ
reg16	16ビット汎用レジスタ (16ビット汎用レジスタを2つ用いる命令における、デスティネーション側レジスタ)
reg16'	16ビット汎用レジスタを2つ用いる命令における、ソース側レジスタ
mem	8/16ビット・メモリ・アドレス
mem8	8ビット・メモリ・アドレス
mem16	16ビット・メモリ・アドレス
mem32	32ビット・メモリ・アドレス
dmem	16ビット・ダイレクト・メモリ・アドレス
imm	8/16ビット・イミューディエト・データ
imm3	3ビット・イミューディエト・データ
imm4	4ビット・イミューディエト・データ
imm8	8ビット・イミューディエト・データ
imm16	16ビット・イミューディエト・データ
acc	アキュムレータ (AWまたはAL)
areg	セグメント・レジスタ
src-table	256バイト変換テーブルの名称
src-block	IXレジスタでアドレスされるソース・ブロックの名称
dst-block	IXレジスタでアドレスされるデスティネーション・ブロックの名称
near-proc	現在のプログラム・セグメント内のプロシージャ
far-proc	別のプログラム・セグメント内のプロシージャ
near-label	現在のプログラム・セグメント内のラベル
abort-label	命令の終わりから-128〜+127バイトの範囲のラベル
far-label	別のプログラム・セグメント内のラベル
regptr16	現在のプログラム・セグメント内のコール・アドレスのオフセットを持つ16ビット汎用レジスタ
memptr16	現在のプログラム・セグメント内のコール・アドレスのオフセットを持つ16ビット・メモリ・アドレス
memptr32	別のプログラム・セグメント内のコール・アドレスのオフセットとセグメント・データを持つ32ビット・メモリ・アドレス
pop-value	スタックから捨てるバイト数 (0-64 K, 通常は偶数)
fpcop	浮動小数点演算用プロセッサの命令コードを判別するイミューディエト値



表 2-2 オペランド・タイプの凡例 (2/2)

識別子	説 明
R	レジスタ・セット (AW, BW, CW, DW, SP, BP, IX, IY)
DS1-spec	DS1, または DS1 に ASSUME してあるセグメント名 / グループ名
Seg-spec	任意のセグメント・レジスタ名, またはセグメント・レジスタに ASSUME してあるセグメント名 / グループ名
( )	省略可能

表 2-3 命令語の凡例

識別子	説 明
W	バイト / ワード・フィールド (0, 1)
reg	レジスタ・フィールド (000-111)
reg'	レジスタ・フィールド (000-111) (レジスタを 2 つ用いる命令における, ソース側レジスタ)
mod, mem	メモリ・アドレッシング指定ビット (mod: 00-10, mem: 000-111)
(disp-low)	オプション 16 ビット・ディスプレイースメント下位バイト
(disp-high)	// 上位バイト
disp-low	PC レラティブ加算用 16 ビット・ディスプレイースメント下位バイト
disp-high	// 上位バイト
imm3	3 ビット・イミューディエイト・データ
imm4	4 ビット //
imm8	8 ビット //
imm16-low	16 ビット・イミューディエイト・データの下位バイト
imm16-high	// 上位バイト
addr-low	16 ビット・ダイレクト・アドレスの下位バイト
addr-high	// 上位バイト
aseg	セグメント・レジスタ指定ビット (00-11)
s	サイン拡張指定ビット (1: サイン拡張あり, 0: サイン拡張なし)
offset-low	PC にロードされる 16 ビット・オフセット・データの下位バイト
offset-high	// 上位バイト
seg-low	PS にロードされる 16 ビット・セグメント・データの下位バイト
seg-high	// 上位バイト
pop-value-low	スタックの捨てるバイト数を指定する 16 ビット・データの下位バイト
pop-value-high	// 上位バイト
disp8	PC にレラティブ加算される 8 ビット・ディスプレイースメント
X	} 浮動小数点演算用コプロセッサのオペレーション・コード
XXX	
YYY	
ZZZ	

表 2-4 命令形式またはオペレーション説明上の凡例 (1/2)

識別子	説明
dst	デスティネーション・オペランド
dst1	#
dst2	#
src	ソース・オペランド
src1	#
src2	#
target	ターゲット・オペランド
AW	アキュムレータ (16ビット)
AH	# (上位バイト)
AL	# (下位バイト)
BW	BWレジスタ (16ビット)
CW	CWレジスタ ( # )
CL	# (下位バイト)
DW	DWレジスタ (16ビット)
BP	ベース・ポインタ (16ビット)
SP	スタック・ポインタ (16ビット)
PC	プログラム・カウンタ (16ビット)
PSW	プログラム・ステータス・ワード (16ビット)
IX	インデックス・レジスタ (ソース) (16ビット)
IY	# (デスティネーション) (16ビット)
FS	プログラム・セグメント・レジスタ (16ビット)
SS	スタック・セグメント・レジスタ (16ビット)
DS0	データ・セグメント0レジスタ (16ビット)
DS1	データ・セグメント1レジスタ (16ビット)
AC	補助キャリー・フラグ
CY	キャリー・フラグ
P	パリティ・フラグ
S	サイン・フラグ
Z	ゼロ・フラグ
DIR	方向フラグ
IE	割り込み許可フラグ
V	オーバフロー・フラグ
BRK	ブレーク・フラグ
MD	モード・フラグ (V33A, V53Aにはありません)
(...)	( ) 内で示されるメモリの内容
disp	ディスプレイメント (8/16ビット)
temp	テンポラリ・レジスタ (8/16/32ビット)

表 2-4 命令形式またはオペレーション説明上の凡例 (2/2)

識別子	説 明
temp1	テンポラリ・レジスタ (16ビット)
temp2	#
TA	テンポラリ・レジスタA (16ビット)
TB	テンポラリ・レジスタB (16ビット)
TC	テンポラリ・レジスタC (16ビット)
ext-disp8	8ビット・ディスプレイメントをサイン拡張した16ビット
seg	イミディエイト・セグメント・データ (16ビット)
offset	イミディエイト・オフセット・データ (16ビット)
←	転送方向
+	加 算
-	減 算
×	乗 算
÷	除 算
%	モジュロ
∧	論理積 (AND)
∨	論理和 (OR)
⊕	排他的論理和 (XOR)
××H	16進数 2 けたの数値
××××H	16進数 4 けたの数値

表 2-5 メモリ・アドレッシング

mem \ mod	00	01	10
000	EW + IX	BW + IX + disp8	BW + IX + disp16
001	EW + IY	BW + IY + disp8	BW + IY + disp16
010	BP + IX	BP + IX + disp8	BP + IX + disp16
011	BP + IY	BP + IY + disp8	BP + IY + disp16
100	IX	IX + disp8	IX + disp16
101	IY	IY + disp8	IY + disp16
110	ダイレクト・アドレス	BP + disp8	BP + disp16
111	EW	BW + disp8	BW + disp16

表 2-6 8/16ビット汎用レジスタの選択

reg, req'	W=0	W=1
000	AL	AW
001	CL	CW
010	DL	DW
011	BL	BW
100	AH	SP
101	CH	BP
110	DH	IX
111	BH	IY

表 2-7 セグメント・レジスタの選択

seg	
00	DS1
01	FS
10	SS
11	DS0

図2-1 記述例

ニモニック	<b>ADD</b>	機能	加算 Add
-------	------------	----	-----------

フルネーム

命令の基本記述形式を略号を用いて示します。 → **【命令形式】** ADD dst, src

命令のオペレーションを略号を用いて示します。 → **【オペレーション】** dst←dst+src

この命令で指定できるオペランドを示します。各オペランドの略号の説明は表2-2から表2-4を参照してください。 → **【オペランド】**

ニモニック	オペランド (dst, src)
ADD	reg, reg'
	mem, reg

命令実行により変化するフラグの動作を示します。各フラグの記号については表2-4を、各動作記号については表2-1を参照してください。 → **【フラグ】**

AC	CY	V	F	S	Z
x	x	x	x	x	x

命令のオペレーションの詳細を解説します。 → **【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) の内容と…

RA70116-i(インターツール™)の記述フォーマットに基づいて記述例を示します。 → **【記述例】** MOV AW, 0

命令長を示します。 → **【バイト数】**

ニモニック	オペランド	バイト数
ADD	reg, reg'	2
	mem, reg	2-4

命令フォーマットを示します。各フィールドの略号については表2-3を参照してください。オペレーション・コード欄は、次のようなバイト順で示してあります(最大6バイト)。 → **【命令形式】**

ニモニック	オペランド	オペレーション・コード									
		7	6	5	4	3	2	1	0		
ADD	reg, reg'	0	0	0	0	0	1	W	1	reg	reg'
	mem, reg'	0	0	0	0	0	0	W	mod	reg	mem

オペレーション・コード

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1バイト目						2バイト目									
3バイト目				4バイト目				5バイト目				6バイト目			

<b>ADD</b>	加算 Add
------------	-----------

**【命令形式】**    `ADD dst, src`

**【オペランド、オペレーション】**

ニモニック	オペランド (dst, src)	オペレーション
<b>ADD</b>	reg, reg'	$dst \leftarrow dst + src$
	mem, reg	
	reg, mem	
	reg, imm	
	mem, imm	
	acc, imm	$[W=0 \text{ のとき}] AL \leftarrow AL + imm8$ $[W=1 \text{ のとき}] AW \leftarrow AW + imm16$

**【フ    ラ    グ】**

AC	CY	V	P	S	Z
x	x	x	x	x	x

**【説    明】**    第1オペランドで指定されるデスティネーション・オペランド (dst) の内容と第2オペランドで指定されるソース・オペランド (src) の内容を加算し、結果をデスティネーション・オペランド (dst) に格納します。

**【配    送    例】**    メモリ0:50Hの内容 (ワード・データ) とDWレジスタの内容を加算して0:50Hへストアする。

```

MOV AW, 0
MOV DS1, AW
MOV IY, 50H
ADD DS1:WORD PTR [IY], DW
    
```

【バイト数】

二モニック	オペランド	バイト数
ADD	reg, reg'	2
	mem, reg	2-4
	reg, mem	2-4
	reg, imm	3, 4
	mem, imm	3-6
	acc, imm	2, 3

【命令書形式】

二モニック	オペランド	オペレーション・コード																	
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
ADD	reg, reg'	0	0	0	0	0	0	1	W	1	1							reg	reg'
	mem, reg	0	0	0	0	0	0	0	W	mod								reg	mem
		(disp-low)								(disp-high)									
	reg, mem	0	0	0	0	0	0	1	W	mod								reg	mem
		(disp-low)								(disp-high)									
	reg, imm	1	0	0	0	0	0	0	s	W	1	1	0	0	0	0			reg
		imm8 or imm16-low								imm16-high									
	mem, imm	1	0	0	0	0	0	0	s	W	mod	0	0	0	0				mem
		(disp-low)								(disp-high)									
		imm8 or imm16-low								imm16-high									
	acc, imm	0	0	0	0	0	1	0	W									imm8 or imm16-low	
		imm16-high								-									

<b>ADD4S</b>	10進加算 Add Nibble String
--------------	----------------------------

**【命令形式】** ADD4S [DS1-spec:] dst-string, [Seg-spec:] src-string  
ADD4S

**【オペレーション】** BCDストリング (IX, CL) ← BCDストリング (IX, CL) + BCDストリング (IX, CL)

**【オペランド】**

メモニック	オペランド (dst, src)
<b>ADD4S</b>	[DS1-spec:] dst-string, [Seg-spec:] src-string
	なし

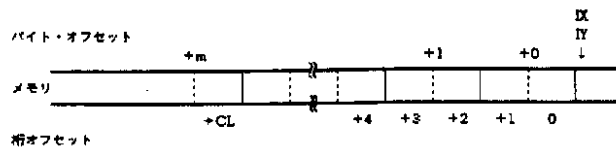
**【フ ラ グ】**

AC	CY	V	P	S	Z
U	×	U	U	U	×

**【説 明】** IXレジスタでアドレスされるバケットBCDストリングとIXレジスタでアドレスされるバケットBCDストリングを加算し、結果をIXレジスタでアドレスされるストリングにストアします。ストリング長 (BCD桁数) は、CLレジスタ (CLの内容がdならばd桁) によって決定され1-254桁まで可能です。

デスティネーション・ストリングは、常にDS1レジスタで指されるセグメント内にロケートされる必要があります。セグメント・オーバーライドはできません。一方、ソース・ストリングは、デフォルト・セグメント・レジスタはDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。

バケットBCDストリングのフォーマットを次に示します。



**注意** BCDストリング命令は常に偶数桁単位で動作します。このため桁数として偶数を指定したときは演算結果、および各フラグは正しく動きますが、桁数として奇数を指定した場合は奇数+1の偶数桁演算を実行し、演算結果、および



各フラグは偶数桁の演算結果を示しますので、桁数として変数を指定する場合は次のようにしてください。

桁数が奇数のときのBCD加算命令は、最上位バイトの上位4ビットを“0”にしてから実行してください。この結果、キャリーは最上位バイトのビット4に示され、フラグには反映されません。

**【記 述 例】**

```

MOV  IX, OFFSET VAR_1
MOV  IY, OFFSET VAR_2
MOV  CL, 4
ADD4S
    
```

**【バ イ ト 数】** 2

**【命 令 書 形 式】**

二モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>ADD4S</b>	[DS1-spec] dst-string, [Seg-spec] src-string	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0
	なし																

**ADDC** キャリーを含む加算  
Add with Carry

**【命令形式】**    `ADDC dst, src`

**【オペランド、オペレーション】**

ニモニック	オペランド (dst, src)	オペレーション
<b>ADDC</b>	reg, reg'	$dst = dst + src + CY$
	mem, reg	
	reg, mem	
	reg, imm	
	mem, imm	
	acc, imm	$[W=0 \text{ のとき}] AL = AL + imm8 + CY$ $[W=1 \text{ のとき}] AW = AW + imm16 + CY$

**【フラグ】**

AC	CY	V	P	S	Z
x	x	x	x	x	x

**【説明】**    第1オペランドで指定されるデスティネーション・オペランド (dst) の内容と第2オペランドで指定されるソース・オペランド (src) の内容をCYフラグの内容も含めて加算し、結果をデスティネーション・オペランド (dst) に格納します。

**【記述例】**    `SET1 CY`        ; CYフラグをセット (1) する。  
                  `XOR AW, AW`     ; AW=0  
                  `MOV BW, OFFH`; BW=OFFH  
                  `ADDC AW, BW` ; AWレジスタの内容=100Hとなる。

**【バイト数】**

ニモニック	オペランド	バイト数
<b>ADDC</b>	reg, reg'	2
	mem, reg	2-4
	reg, mem	2-4
	reg, imm	3, 4
	mem, imm	3-6
	acc, imm	2, 3

【命令形式】

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
ADDC	reg. reg'	0	0	0	1	0	0	1	W	1	1			reg		reg'		
	mem. reg	0	0	0	1	0	0	0	W	mod				reg		mem		
			(disp-low)							(disp-high)								
	reg. mem	0	0	0	1	0	0	1	W	mod				reg		mem		
			(disp-low)							(disp-high)								
	reg. imm	1	0	0	0	0	0	0	W	1	1	0	1	0		reg		
			imm8 or imm16-low							imm16-high								
	mem. imm	1	0	0	0	0	0	0	W	mod	0	1	0			mem		
			(disp-low)							(disp-high)								
	acc. imm		imm8 or imm16-low							imm16-high								
0		0	0	1	0	1	0	W	imm8 or imm16-low							—		
		imm16-high							—									

<b>ADJ4A</b>	加算結果のバケット10進補正 Adjust Nibble Add
--------------	-------------------------------------

**【命令形式】** ADJ4A

**【オペレーション】** AL<math>\wedge</math>0FH>9またはAC=1のとき  
                           AL←AL+6  
                           AC←1  
                           AL>9FHまたはCY=1のとき  
                           AL←AL+60H  
                           CY←1

**【オペランド】**

ニモニック	オペランド
ADJ4A	なし

**【フ ラ グ】**

AC	CY	V	P	S	Z
x	x	U	x	x	x

**【説 明】** 2つのバケット10進数の加算のALレジスタ内の結果を1つのバケット10進数に補正します。

**【配 述 例】** ADJ4A

**【バ イ ト 数】** 1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
ADJ4A	なし	0	0	1	0	0	1	1	1

<b>ADJ4S</b>	減算結果のバケット10進補正 Adjust Nibble Subtract
--------------	--

【命令形式】 ADJ4S

【オペレーション】 AL<0FH>9またはAC=1のとき  
                   AL←AL-6  
                   AC←1  
                   AL>9FHまたはCY=1のとき  
                   AL←AL-60H  
                   CY←1

【オペランド】

ニモニック	オペランド
<b>ADJ4S</b>	なし

【フ ラ グ】

AC	CY	V	P	S	Z
×	×	U	×	×	×

【説 明】 2つのバケット10進数の減算のALレジスタ内の結果を1つのバケット10進数に補正します。

【配 送 例】 SUB AW, BW  
               ADJ4S

【バ イ ト 数】 1

【命令書形式】

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
<b>ADJ4S</b>	なし	0	0	1	0	1	1	1	1

<b>ADJBA</b>	加算結果のアンパクト10進補正 Adjust Byte Add
--------------	------------------------------------

**【命令形式】** ADJBA

**【オペレーション】** AL<math>\wedge</math>OFH>9またはAC=1のとき

AL←AL+6  
 AH←AH+1  
 AC←1  
 CY←AC  
 AL←AL<math>\wedge</math>OFH

**【オペランド】**

ニモニック	オペランド
ADJBA	なし

**【フラグ】**

AC	CY	V	P	S	Z
x	x	U	U	U	U

**【説明】** 2つのアンパクト10進数の加算のALレジスタ内の結果を1つのアンパクト10進数に補正します。上位4ビットは0になります。

**【配列例】** ADJBA

**【バイト数】** 1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
ADJBA	なし	0	0	1	1	0	1	1	1

**ADJBS**

減算結果のアンバクット10進補正  
Adjust Byte Subtract

**【命令形式】** ADJBS

**【オペレーション】** AL $\wedge$ OFH>9またはAC=1のとき  
 AL←AL-6  
 AH←AH-1  
 AC←1  
 CY←AC  
 AL←AL $\wedge$ OFH

**【オペランド】**

モニック	オペランド
ADJBS	なし

**【フラグ】**

AC	CY	V	P	S	Z
x	x	U	U	U	U

**【説明】** 2つのアンバクット10進数の減算のALレジスタ内の結果を1つのアンバクット10進数に補正します。上位4ビットは0になります。

**【記述例】** SUB AW, BW  
 ADJBS

**【バイト数】** 1

**【命令書形式】**

モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
ADJBS	なし	0	0	1	1	1	1	1	1

<b>AND</b>	論理積 And
------------	------------

**【命令形式】**    AND dst, src

**【オペランド, オペレーション】**

ニモニック	オペランド (dst, src)	オペレーション
<b>AND</b>	reg, reg'	dst←dst∧src
	mem, reg	
	reg, mem	
	reg, imm	
	mem, imm	
	acc, imm	[W=0のとき] AL←AL∧imm8 [W=1のとき] AW←AW∧imm16

**【フラグ】**

AC	CY	V	P	S	Z
U	0	0	x	x	x

**【説明】**        第1オペランドで指定されるデスティネーション・オペランド (dst) と第2オペランドで指定されるソース・オペランド (src) の論理積 (AND) をとり, 結果をデスティネーション・オペランド (dst) に格納します。

**【記述例】**     MOV DW, IX  
                   AND DW, 7FFFH

**【バイト数】**

ニモニック	オペランド	バイト数
<b>AND</b>	reg, reg'	2
	mem, reg	2-4
	reg, mem	2-4
	reg, imm	3, 4
	mem, imm	3-6
	acc, imm	2, 3



**[命令形式]**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
AND	reg, reg'	0	0	1	0	0	0	1	W	1	1	reg	reg'				
	mem, reg	0	0	1	0	0	0	0	W	mod	reg	mem.					
		(disp-low)					(disp-high)										
	reg, mem	0	0	1	0	0	0	1	W	mod	reg	mem.					
		(disp-low)					(disp-high)										
	reg, imm. <sup>注</sup>	1	0	0	0	0	0	0	W	1	1	1	0	0	reg		
		imm8 or imm16-low					imm16-high										
	mem, imm	1	0	0	0	0	0	0	W	mod	1	0	0	mem.			
		(disp-low)					(disp-high)										
		imm8 or imm16-low					imm16-high										
acc, imm	0	0	1	0	0	1	0	W	imm8 or imm16-low								
	imm16-high					—											

注 アセンブラ、コンパイラによっては、次に示すようなコードが生成されることがあります。

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
1 0 0 0 0 0 1 W	1 1 1 0 0 reg
imm8	—

このような場合でも正常に命令を実行します。ただし、エミュレータによっては、この命令に対する逆アセンブラ機能、アセンブラ機能がサポートされていない場合がありますのでご注意ください。

**BC  
BL**

CY=1による条件分岐  
Branch if Carry  
Branch if Lower

**【命令形式】** BC short-label  
BL short-label

**【オペレーション】** CY=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BC	short-label
BL	

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** CYフラグが1のとき、現在のPCの値に8ビット・ディスプレースメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にブランチできます。

**【記述例】**

```

TEST AL, BL
BC SHORT LP4 ; LP4=レーベル
:
TEST AL, BL
BL SHORT LP5 ; LP5=レーベル
:
LP4:
    
```

**【バイト数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BC	short-label	0	1	1	1	0	0	1	0	disp8							
BL																	

<b>BCWZ</b>	CW=0による条件分岐 Branch if CW equals Zero
-------------	---

**【命令形式】** BCWZ short-label

**【オペレーション】** CW=0のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BCWZ	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** CWレジスタの値が0ならば、現在のPCの値に8ビット・ディスプレースメント(実際にはサイン拡張された16ビット)を加えた値をロードします。  
 この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。  
 条件不成立のときは次の命令に進みます。

**【記述例】**

```

LP22:
    !
    ADD AL, BL
    BCWZ SHORT LP22 ;LP22=レーベル
    
```

**【バイト数】** 2

**【命令語形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BCWZ	short-label	1	1	1	0	0	0	1	1	disp8							

<b>BE</b> <b>BZ</b>	Z=1による条件分岐 Branch if Equal Branch if Zero
------------------------	---

**【命令形式】** BE short-label  
 BZ short-label

**【オペレーション】** Z=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BE	short-label
BZ	

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** Zフラグが1のとき、現在のPCの値に8ビット・ディスプレースメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
 この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。

**【記法例】**

```

    AND AL, 2
    BE SHORT LOOP ; LOOP=ラベル
    :
    OR AH, BH
    BZ SHORT LOOP1 ; LOOP1=ラベル
    :
    LOOP:
    
```

**【バイト数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BE	short-label	0	1	1	1	0	1	0	0	disp8							
BZ																	

<b>BGE</b>	S≠V=0による条件分岐 Branch if Greater Than or Equal
------------	---

【命令形式】 BGE short-label

【オペレーション】 S≠V=0のとき: PC←PC+ext-disp8

【オペランド】

ニモニック	オペランド
BGE	short-label

【フ ラ グ】

A	C	O	V	P	S	Z
0	0	0	0	0	0	0

【説 明】 SフラグとVフラグの排他的論理和(XOR)が0のとき、現在のPCの値に8ビット・ディスプレイメント(実際にはサイン拡張された16ビット)を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。  
条件不成立のときは次の命令に進みます。

【記 述 例】

```

SHL AL, 1
BGE SHORT LP16 ; LP16=ラベル
:
LP16:
    
```

【バ イ ト 数】 2

【命令書形式】

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BGE	short-label	0	1	1	1	1	1	0	1	disp8							

<b>BGT</b>	(S $\vee$ V) $\vee$ Z=0による条件分岐 Branch if Greater Than
------------	--

**【命令形式】** BGT short-label

**【オペレーション】** (S $\vee$ V) $\vee$ Z=0のとき : PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BGT	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** SフラグとVフラグの排他的論理和 (XOR) をとった結果とZフラグの論理和 (OR) が0のとき、現在のPCの値に8ビット・ディスプレースメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。  
条件不成立のときは次の命令に進みます。

**【記述例】** LP18:  
          :  
          SHL AL, 1  
          BGT LP18

**【バイト数】** 2

**【命令語形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BGT	short-label	0	1	1	1	1	1	1	1	disp8							

<b>BH</b>	CY/VZ=0による条件分枝 Branch if Higher
-----------	------------------------------------

【命令形式】 BH short-label

【オペレーション】 CYVZ=0のとき : PC←PC+ext-disp8

【オペランド】

ニモニック	オペランド
<b>BH</b>	short-label

【フラグ】

AC	CY	V	P	S	Z

【説明】 CYフラグとZフラグの論理和 (OR) が0のとき、現在のPCの値に8ビット・ディスプレイメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。

【記述例】

```

ROL AL, 1
BH SHORT LP10 ; LP10=レーベル
:
LP10:
    
```

【バイト数】 2

【命令形式】

ニモニック	オペランド	オペレーション・コード														
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
<b>BH</b>	short-label	0	1	1	1	0	1	1	1	1	disp8					

<b>BLE</b>	(S≠V)∨Z=1による条件分岐 Branch if Less than or Equal
------------	--

**【命令形式】** BLE short-label

**【オペレーション】** (S≠V)∨Z=1のとき: PC←PC+ext-disp8

**【オペランド】**

二モニック	オペランド
BLE	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** SフラグとVフラグの排他的論理和 (XOR) をとった結果とZフラグの論理和 (OR) が1のとき、現在のPCの値に8ビット・ディスプレイメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。  
条件不成立のときは次の命令に進みます。

**【記述例】** LP17:  
:  
SHR AL, 1  
BLE SHORT LP17

**【バイト数】** 2

**【命令形式】**

二モニック	オペランド	オペレーション・コード														
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
BLE	short-label	0	1	1	1	1	1	1	0	disp8						



<b>BLT</b>	S $\neq$ V=1による条件分岐 Branch if Less Than
------------	--

**【命令形式】**     BLT short-label

**【オペレーション】**   S $\neq$ V=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BLT	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】**     SフラグとVフラグの排他的論理和(XOR)が1のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
 この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にブランチできます。  
 条件不成立のときは次の命令に進みます。

**【記述例】**

```

ADD AL, BL
BLT SHORT LP15 ; LP15=ラベル
    ;
LP15:
```

**【バイト数】**     2

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BLT	short-label	0	1	1	1	1	1	0	0	disp8							

<b>BN</b>	S=1による条件分岐 Branch if Negative
-----------	----------------------------------

**【命令形式】** BN short-label

**【オペレーション】** S=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BN	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** Sフラグが1のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。

**【記述例】**

```

ADD AL, BL
BN LP11 ; LP11=ラベル
:
LP11:
    
```

**【バイト数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BN	short-label	0	1	1	1	1	0	0	0	disp8							

<p><b>BNC</b> <b>BNL</b></p>	<p>CY=0による条件分岐 Branch if Not Carry Branch if Not Lower</p>
----------------------------------	--

**[命令形式]**    **BNC short-label**  
                  **BNL short-label**

**[オペレーション]**    CY=0のとき: PC←PC+ext-disp8

**[オペランド]**

ニモニック	オペランド
<b>BNC</b>	short-label
<b>BNL</b>	

**[フ ラ グ]**

AC	CY	V	P	S	Z

**[説 明]**    CYフラグが0のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。

**[記 述 例]**

```

ROR AL, 1
BNC SHORT LP6 ; LP6=レーベル
:
ROR AL, 1
BNL SHORT LP7 ; LP7=レーベル
:
LP6:
    
```

**[バ イ ト 数]**    2

**[命令形式]**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>BNC</b>	short-label	0	1	1	1	0	0	1	1	disp8							
<b>BNL</b>																	

<p><b>BNE</b> <b>BNZ</b></p>	<p>Z=0による条件分岐 Branch if Not Equal Branch if Not Zero</p>
----------------------------------	--

**【命令形式】**    **BNE** short-label  
                  **BNZ** short-label

**【オペレーション】**    Z=0のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
<b>BNE</b>	short-label
<b>BNZ</b>	

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】**    Zフラグが0のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にブランチできます。

**【記述例】**

```

OR AL, BL
BNE SHORT LP8 ; LP8=レーベル
:
AND SH, BH
BNZ SHORT LP9 ; LP9=レーベル
:
LP8:

```

**【バイト数】**    2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>BNE</b>	short-label	0	1	1	1	0	1	0	1	disp8							
<b>BNZ</b>																	

**BNH**CY/VZ=1による条件分岐  
Branch if Not Higher

**【命令形式】** BNH short-label

**【オペレーション】** CY/VZ=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BNH	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** CYフラグとZフラグの論理和 (OR) が1のとき、現在のPCの値に8ビット・ディスプレイスメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。

**【記述例】**

```
ROR AL, 1
BNH SHORT LP9 ; LP9=ラベル
:
LP9:
```

**【バイト数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BNH	short-label	0	1	1	1	0	1	1	0	disp8							

<b>BNV</b>	V=0による条件分岐 Branch if Not Overflow
------------	--------------------------------------

【命令形式】 **BNV short-label**

【オペレーション】 V=0のとき: PC←PC+ext-disp8

【オペランド】

ニモニック	オペランド
BNV	short-label

【フラグ】

AC	CF	V	P	S	Z
0	0	0	0	0	0

【説明】 Vフラグが0のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にブランチできます。

【記述例】

```
ROR AL, 1
BNV LP3
:
LP3:
```

【バイト数】 2

【命令形式】

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BNV	short-label	0	1	1	1	0	0	0	1	disp8							

<b>BP</b>	S=0による条件分岐 Branch if Positive
-----------	----------------------------------

**【命令形式】** BP short-label

**【オペレーション】** S=0のとき: PC←PC+ext-disp8

**【オペランド】**

二モニック	オペランド
BP	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** Sフラグが0のとき、現在のPCの値に8ビット・ディスプレースメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。

**【記述例】**

```
SHR AL, 1
BP SHORT LP12 ; LP12=レーベル
:
LP12:
```

**【バイト数】** 2

**【命令書形式】**

二モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BP	short-label	0	1	1	1	1	0	0	1	disp8							

**BPE** P=1による条件分岐  
Branch if Parity Even

**【命令形式】** BPE short-label

**【オペレーション】** P=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BPE	short-label

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【説 明】** Pフラグが1のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128〜+127バイトのアドレス内にジャンプできます。

**【配 述 例】**

```

ADD AL, BL
BPE SHORT LP13 ; LP13=レーベル
:
LP13:
    
```

**【バ イ ト 数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BPE	short-label	0	1	1	1	1	0	1	0	disp8							



BPO

P=0による条件分岐  
Branch if Parity Odd

**【命令形式】** BPO short-label

**【オペレーション】** P=0のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BPO	short-label

**【フ ラ グ】**

AC	CY	V	P	S	Z
0	0	0	0	0	0

**【説 明】** Pフラグが0のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。

**【記 述 例】**

```

ADD AL, BL
BPO SHORT LP14 ; LP14=レーベル
:
LP14:
    
```

**【バ イ ト 数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BPO	short-label	0	1	1	1	1	0	1	1	disp8							

<b>BR</b>	無条件分岐 Branch
-----------	-----------------

**【命令形式】** BR target

**【オペランド、オペレーション】**

ニモニック	オペランド (target)	オペレーション
<b>BR</b>	near-label	PC ← PC + disp
	short-label	PC ← PC + ext-disp8
	regptr16	PC ← target
	memptr16	
	far-label	PS ← seg PC ← offset
	memptr32	PS ← (memptr32+3, memptr32+2) PC ← (memptr32+1, memptr32)

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】**

- target = near-label のとき  
現在のPCの値に16ビット・ディスプレースメント (disp) を加えた値をPCに転送します。  
ブランチ・アドレスがこの命令の置かれているセグメント内であれば、アセンブラが自動的にこの命令を実行します。
  
- target = short-label のとき  
現在のPCの値に8ビット・ディスプレースメント (実際にはサイン拡張された16ビット (ext-disp8)) を加えた値をPCに転送します。  
ブランチ・アドレスがこの命令の置かれているセグメント内であつ±127バイト内であれば、アセンブラが自動的にこの命令を実行します。
  
- target = regptr16 または target = memptr16 のとき  
ターゲット・オペランド (target) の内容をPCに転送します。  
この命令の置かれているセグメント内の任意のアドレスにブランチできます。

○target=far-labelのとき

PCに命令の2, 3バイト目の16ビット・オフセット・データを, PSに命令の4, 5バイト目の16ビット・セグメント・データを転送します。  
任意のセグメントの任意のアドレスにジャンプできます。

○target=memptr32のとき

32ビット・メモリの上位2バイトをPSに, 下位2バイトをPCにロードします。  
任意のセグメントの任意のアドレスにジャンプできます。

**【記述例】** BR \$-8

**【バイト数】**

ニモニック	オペランド	バイト数
BR	near-label	3
	short-label	2
	regptr16	2
	memptr16	2-4
	far-label	5
	memptr32	2-4

**【命令形式】**

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
BR	near-label	1	1	1	0	1	0	0	1	disp-low								
		disp-high								—								
	short-label	1	1	1	0	1	0	1	1	disp8								
	regptr16	1	1	1	1	1	1	1	1	1	1	1	0	0	reg			
	memptr16	1	1	1	1	1	1	1	1	mod	1	0	0	mem				
		(disp-low)								(disp-high)								
	far-label	1	1	1	0	1	0	1	0	offset-low								
offset-high								seg-low										
seg-high								—										
memptr32	1	1	1	1	1	1	1	1	mod	1	0	1	mem					
	(disp-low)								(disp-high)									

<b>BRK</b>	ソフトウェア・トラップ <b>Break</b>
------------	-----------------------------

**【命令形式】** BRK target

**【オペランド、オペレーション】**

ニモニック	オペランド (target)	オペレーション
<b>BRK</b>	3	TA← (OODH, OCOH) TC← (OOFH, COEH) SP←SP-2, (SP+1, SP) ←PSW IE←0, BRK←0 SP←SP-2, (SP+1, SP) ←PS PS←TC SP←SP-2, (SP+1, SP) ←PC PC←TA
	imm8 (*3)	TA← (imm8×4+1, imm8×4) TC← (imm8×4+3, imm8×4+2) SP←SP-2, (SP+1, SP) ←PSW IE←0, BRK←0 SP←SP-2, (SP+1, SP) ←PS PS←TC SP←SP-2, (SP+1, SP) ←PC PC←TA

**【フ ラ グ】**

AC	CY	V	P	S	Z	IE	BRK
						0	0

**【説 明】** PSW, PS, PCの値をスタックに退避し, IEフラグとBRKフラグをリセット(0)します。

続いて, target=3の場合は割り込みベクタ・テーブルのベクタ3の下位2バイトをPCに, 上位2バイトをPSにロードします。

target=imm8の場合は8ビット・イミューディエイト・データで指定される割り込みベクタ・テーブル(4ビット)の下位2バイトをPCに, 上位2バイトをPSにロードします。

- 【記述例】 ●BRK 3  
●BRK 5

【バイト数】

モニック	オペランド	バイト数
BRK	3	1
	imm8	2

【命令書形式】

モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BRK	3	1	1	0	0	1	1	0	0	—							
	imm8	1	1	0	0	1	1	0	1	imm8							

**BRKEM** [V33A, V53Aを除く] エミュレーション・モードの起動  
Break for Emulation

**【命令形式】** BRKEM imm8

**【オペレーション】** TA ← (imm8×4+1, imm8×4)  
 TC ← (imm8×4+3, imm8×4+2)  
 SP ← SP-2 (SP+1, SP) ← PSW  
 MD ← 0 : 書き込み許可状態  
 SP ← SP-2 (SP+1, SP) ← PS  
 PS ← TC  
 SP ← SP-2 (SP+1, SP) ← PC  
 PC ← TA

**【オペランド】**

ニモニック	オペランド
BRKEM	imm8

**【フ ラ グ】**

AC	CY	V	P	S	Z	MD
						0

**【説 明】** エミュレーション・モードを起動するための命令です。PSW, PS, PCの値をスタックに過渡し、MDフラグをリセット(0)するとともに書き込み許可状態にしたあと、オペランドに記述された8ビット・イミディエト・データで指定される割り込みベクタがアドレスするエミュレーション番地にジャンプします。  
 こうしてジャンプした割り込み処理ルーチン(エミュレーション用)の命令コードをフェッチすると、CPUはそのコードをμPD8080AFの命令として解釈実行します。エミュレーション・モードからネイティブ・モードに戻るには、RETEM命令またはCALLN命令を使用します。

**【記 述 例】** BRKEM 40H

**【バ イ ト 数】** 3

第 2 章 命 令

**[命令書形式]**

二モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>BRKEM</b>	imm8	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
		imm8								—							

**BRKV** オーバーフロー例外  
Break if Overflow

**【命令形式】** BRKV

**【オペレーション】** V=1のとき TA← (011H, 010H)  
 TC← (013H, 012H)  
 SP←SP-2, (SP+1, SP) ←PSW  
 IE←0, BRK←0  
 SP←SP-2, (SP+1, SP) ←PS  
 PS←TC  
 SP←SP-2, (SP+1, SP) ←PC  
 PC←TA

**【オペランド】**

ニモニック	オペランド
BRKV	なし

**【フラグ】**

AC	CY	V	P	S	Z	IE	BRK
						0	0

**【説明】** Vフラグがセット(1)されていれば、PSW, PS, PCの値をスタックに過渡し、IEフラグとBRKフラグをリセット(0)します。  
 続いて、target=3の場合は割り込みベクタ・テーブルのベクタ4の下位2バイトをPCに、上位2バイトをPSにロードします。  
 Vフラグがリセット(0)されていれば次の命令に進みます。

**【記述例】** BRKV

**【バイト数】** 1

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
BRKV	なし	1	1	0	0	1	1	1	0



**BRKXA** [V33A, V53Aだけ] 拡張アドレス・モードの起動  
Break Extended Address Mode

**【命令形式】** BRKXA imm8

**【オペレーション】**  
 temp1 ← (imm8×4+1, imm8×4)  
 temp2 ← (imm8×4+3, imm8×4+2)  
 XA ← 1  
 PC ← temp1  
 PS ← temp2

**【オペランド】**

二モニック	オペランド
BRKXA	imm8

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【説 明】** 拡張アドレス・モードを起動します。  
 オペランドにより指定された割り込みベクタ・テーブルのエントリに格納されているアドレスに制御を移し、XAMレジスタ (内部I/Oアドレス: FF80H) のXAフラグをセット(1)します。

通常アドレス・モードで実行したときは、通常アドレス・モードのアドレス上のベクタ・テーブルを読み込んだあと、拡張アドレス・モードに移り、先に読み込んだベクタ・テーブルのアドレスにジャンプします。

拡張アドレス・モードで実行したときは、拡張アドレス・モードのアドレス上のベクタ・テーブルを読み込んだあと、このベクタ・テーブルのアドレスにジャンプします。

スタックへのPC, PS, PSWの値の過渡は行いません。

拡張アドレス・モードからの復帰の際にはRETXA命令を使用します。RETI命令では復帰しません。

**【記 述 例】** BRKXA 0AH

【バイト数】 3

【命令書形式】

二モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BRCKA	imm8	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
		imm8							---								

# BUSLOCK

バス・ロック・プリフィクス

Bus Lock Prefix

[命令形式] BUSLOCK

[オペレーション] Bus Lock Prefix

[オペランド]

モニタ	オペランド
BUSLOCK	なし

[フラグ]

AC	CY	V	P	S	Z

[説明] ○V20, V30, V20HL, V30HL

大規模モード時：この命令に続く1命令を実行している間、バス・ロック信号 (BUS LOCK) を出力します。

リピート・プリフィクスの付いたブロック処理命令に対してこの命令が用いられれば、ブロック処理が終了するまでBUSLOCK信号が出力され続けます。

小規模モード時：BUSLOCK信号は出力されませんが、大規模モードでBUSLOCK信号が出力される期間、バス・ホールド要求が禁止されます。したがって、ブロック処理の途中でバス・ホールド要求を受け付けたくない場合、この命令を用いると有効です。

注意1. この命令はPOLL命令の直前に置かないでください。

2. この命令と次の命令の間では、ハードウェア割り込み要求(NMI, INT)およびシングルステップ・ブレークは受け付けられません。

○V20, V30, V20HL, V30HL以外

この命令に続く1命令を実行している間、バス・ロック信号 (BUSLOCK) を出力します。

リピート・プリフィクスの付いたブロック処理命令に対してこの命令が用いられれば、ブロック処理が終了するまでBUSLOCK信号が出力され続けます。

第2章 命 令

- 注意1. この命令はPOLL命令の直前に置かないでください。
- 2. この命令と次の命令の間では、ハードウェア割り込み（マスクブル割り込み、ノンマスクブル割り込み）要求およびシングルステップ・ブレークは受け付けられません。

【記 述 例】 BUSLOCK REP MOVEKB

【バ イ ト 数】 1

【命 令 形 式】

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
BUSLOCK	なし	1	1	1	1	0	0	0	0

<b>BV</b>	V=1による条件分岐 Branch if Overflow
-----------	----------------------------------

**【命令形式】** BV short-label

**【オペレーション】** V=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
BV	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** Vフラグが1のとき、現在のPCの値に8ビット・ディスプレイメント（実際にはサイン拡張された16ビット）を加えた値をロードします。  
この命令の置かれているセグメント内であつ、-128～+127バイトのアドレス内にジャンプできます。

**【記述例】** LP2:  
:  
SHL AL, 1  
BV SHORT LP2

**【バイト数】** 2

**【命令語形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
BV	short-label	0	1	1	1	0	0	0	0	disp8							

<b>CALL</b>	サブルーチン・コール Call
-------------	--------------------

[命令形式] **CALL target**

[オペランド、オペレーション]

モニック	オペランド (target)	オペレーション
<b>CALL</b>	near-proc	SP←SP-2 (SP+1, SP) ←PC PC←PC+disp
	regptr16	SP←SP-2 (SP+1, SP) ←PC PC←regptr16
	memptr16	TA← (memptr16+1, memptr16) SP←SP-2 (SP+1, SP) ←PC PC←TA
	far-proc	SP←SP-2 (SP+1, SP) ←PS PS←seg SP←SP-2 (SP+1, SP) ←PC PS←offset
	memptr32	TA← (memptr32+1, memptr32) TB← (memptr32+3, memptr32+2) SP←SP-2 (SP+1, SP) ←PS PS←TB SP←SP-2 (SP+1, SP) ←PC PC←TA

[フ ラ グ]

AC	CY	V	P	S	Z

- 【説明】**
- target=near-procまたはtarget=regptr16のとき  
PCの値がスタックに退避されたあと、ターゲット・オペランド (target) の次の内容がPCに転送されます。  
target=near-procのとき : 16ビット相対アドレス  
target=regptr16のとき : 16ビット・レジスタの値 (オフセット)
  - target=memptr16のとき  
PCの値がスタックに退避されたあと、ターゲット・オペランド (target) でアドレスされる16ビット・メモリの内容 (オフセット) がPCに転送されます。  
この命令の置かれているセグメント内の任意のアドレスを呼ぶことが可能です。
  - target=far-procのとき  
PCとPSの値がスタックに退避されたあと、PCには命令の2, 3バイト目が、PSには命令の4, 5バイト目が転送されます。  
この命令によって任意のセグメントの任意のアドレスを呼ぶことが可能です。
  - target=memptr32のとき  
PCとPSの値がスタックに退避されたあと、ターゲット・オペランド (target) でアドレスされる32ビット・メモリの上位2バイトがPSに、下位2バイトがPCに転送されます。  
この命令によって任意のセグメントの任意のアドレスを呼ぶことが可能です。

- 【記述例】**
- CALL \$+10
  - CALL SUB1 ;SUB1はラベル

**【バイト数】**

ニモニック	オペランド	バイト数
CALL	near-proc	3
	regptr16	2
	memptr16	2-4
	far-proc	5
	memptr32	2-4

**【命令形式】**

二モニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
CALL	near-proc	1	1	1	0	1	0	0	0	disp-low								
		disp-high								—								
	regptr16	1	1	1	1	1	1	1	1	1	1	0	1	0	reg			
	memptr16	1	1	1	1	1	1	1	1	mod	0	1	0	mem				
		(disp-low)								(disp-high)								
	far-proc	1	0	0	1	1	0	1	0	offset-low								
		offset-high								seg-low								
		seg-high								—								
	memptr32	1	1	1	1	1	1	1	1	mod	0	1	1	mem				
		(disp-low)								(disp-high)								



**CALLN** [V33A, V53Aを除く] ネイティブ・モード・コール  
Call Native

**【命令形式】** CALLN imm8

**【オペレーション】**  
 TA ← (imm8×4+1, imm8×4)  
 TC ← (imm8×4+3, imm8×4+2)  
 SP ← SP-2, (SP+1, SP) ← PSW  
 MD ← 1  
 SP ← SP-2, (SP+1, SP) ← PS  
 PS ← TC  
 SP ← SP-2, (SP+1, SP) ← PC  
 PC ← TA

**【オペランド】**

二モニック	オペランド
CALLN	imm8

**【フラグ】**

AC	CY	V	P	S	Z	MD
						1

**【説明】** エミュレーション・モードにおいてこの命令が実行されると(この命令は $\mu$ PD8080AFの命令として解釈されます), CPUはPS, PC, PSWの値をスタックに逃避し(ここでMD=0が逃避されます), さらにMDフラグをセット(1)して, オペランドに記述された8ビット・イミディエイト・データで指定される割り込みベクタをPSとPCにロードします。  
 こうして, エミュレーション・モードから, ネイティブ・モードにおける割り込みルーチンをコールすることができます。  
 なお, この割り込みルーチンからエミュレーション・モードに戻るには, RETI命令を使用します。

**【記述例】** CALLN 40H

**【バイト数】** 3

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>CALLN</b>	imm8	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1
		imm8								—							

<b>CHKIND</b>	インデクス値チェック Check Index
---------------	---------------------------

**【命令形式】**   CHKIND reg16, mem32

**【オペレーション】**   (mem32) > reg16 または (mem32+2) < reg16 のとき  
 TA ← (015H, 014H)  
 TC ← (017H, 016H)  
 SP ← SP - 2, (SP+1, SP) ← PSW  
 IE ← 0, BRK ← 0  
 SP ← SP - 2, (SP+1, SP) ← PS  
 PS ← TC  
 SP ← SP - 2, (SP+1, SP) ← PC  
 PC ← TA

**【オペランド】**

二モニック	オペランド
CHKIND	reg16, mem32

**【フラグ】**   割り込み条件成立のとき

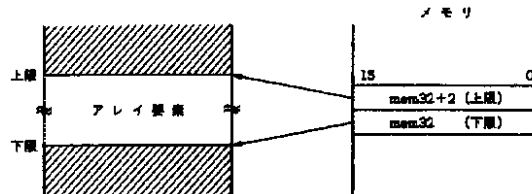
AC	CY	V	P	S	Z	IE	BRK
						0	0

割り込み条件不成立のとき

AC	CY	V	P	S	Z	IE	BRK

第2章 命 令

**【説 明】** アレイ・タイプのデータ構造において、要素を指定するインデクス値が定義域内にあるか否かをチェックするための命令です。インデクスが定義域を超える場合には、BRK 5命令を起動します。定義域値は、あらかじめメモリ中の2ワード（1ワード目は下限値、2ワード目は上限値）にセットしておきます。インデクス値は、アレイ操作プログラムが使用しているレジスタ（任意の16ビット・レジスタ）を対象とします。



**【記 述 例】** CHKIND AW, DWORD\_VAR

**【バ イ ト 数】** 2~4

**【命 令 形 式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>CHKIND</b>	reg16, mem32	0	1	1	0	0	0	1	0	mod	reg	mem					
		(disp-low)								(disp-high)							

<b>CLR1</b>	ビットのリセット Clear Bit
-------------	-----------------------

**【命令形式】** ① CLR1 dst, src  
② CLR1 dst

**【オペレーション】** 命令形式①: dstのビットn (nはsrcで指定) ←0  
命令形式②: dst←0

**【オペランド】** 命令形式① 命令形式②

ニモニック	オペランド (dst, src)
CLR1	reg8, CL
	mem8, CL
	reg16, CL
	mem16, CL
	reg8, imm3
	mem8, imm3
	reg16, imm4
	mem16, imm4

ニモニック	オペランド (dst)
CLR1	CY
	DIR

**【フラグ】** 命令形式①

AC	CY	V	P	S	Z

命令形式② (dst=CYの場合)

AC	CY	V	P	S	Z
	0				

命令形式② (dst=DIRの場合)

AC	CY	V	P	S	Z	DIR
						0

**【説 明】** 命令形式①: 第1オペランドで指定されるデスティネーション・オペランド (dst) のビットn (nは第2オペランドで指定されるソース・オペランド (src) の内容) をリセット (0) し、結果をデスティネーション・オペランド (dst) に格納します。

オペランドがreg8, CLまたはmem8, CLのとき、CLの値は下位3ビット (0-7) のみ有効です。

オペランドがreg16, CLまたはmem16, CLのとき、CLの値は下位4ビット (0-15) のみ有効です。

オペランドがreg8, imm3のとき、命令の4バイト目のイミディエート・データは下位3ビットのみ有効です。

オペランドがmem8, imm3のとき、命令の最終バイトのイミディエート・データは下位3ビットのみ有効です。

オペランドがreg16, imm4のとき、命令の4バイト目のイミディエート・データは下位4ビットのみ有効です。

オペランドがmem16, imm4のとき、命令の最終バイトのイミディエート・データは下位4ビットのみ有効です。

命令形式②: dst=CYの場合、CYフラグをリセット (0) します。

dst=DIRの場合、DIRフラグをリセット (0) します。また、MOVSK, CMPBK, CMPM, LDM, STM, INM, OUTMの各命令の実行時にインデクス・レジスタ (IX, IY) をオートインクリメントするように設定します。

**【配 送 例】**

```
CLR1 CY
SHL AL, 1
BC $+6
```

**【バ イ ト 数】**

ニモニック	オペランド	バイト数
CLR1	reg8, CL	3
	mem8, CL	3-5
	reg16, CL	3
	mem16, CL	3-5
	reg8, imm3	4
	mem8, imm3	4-6
	reg16, imm4	4
	mem16, imm4	4-6
	CY	1
	DIR	1

[命令形式]

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
CLR1	reg8, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	0
		1	1	0	0	0			reg								—
	mem8, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	0
		mod	0	0	0			mem								(disp-low)	
									(disp-high)								—
	reg16, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1
		1	1	0	0	0			reg								—
	mem16, CL	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1
		mod	0	0	0			mem								(disp-low)	
									(disp-high)								—
	reg8, imm3	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0
		1	1	0	0	0			reg								imm3
	mem8, imm3	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0
		mod	0	0	0			mem								(disp-low)	
									(disp-high)								imm3
	reg16, imm4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
		1	1	0	0	0			reg								imm4
	mem16, imm4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
mod		0	0	0			mem								(disp-low)		
								(disp-high)								imm4	
CY		1	1	1	1	1	0	0	0							—	
DIR		1	1	1	1	1	1	0	0							—	

<b>CMP</b>	比較 Compare
------------	---------------

**【命令形式】**    **CMP** *dst, src*

**【オペランド、オペレーション】**

ニモニック	オペランド ( <i>dst, src</i> )	オペレーション
<b>CMP</b>	<i>reg, reg'</i>	<i>dst-src</i>
	<i>mem, reg</i>	
	<i>reg, mem</i>	
	<i>reg, imm</i>	
	<i>mem, imm</i>	
	<i>acc, imm</i>	[W=0のとき] AL- <i>imm8</i> [W=1のとき] AW- <i>imm16</i>

**【フ ラ グ】**

AC	CY	V	P	S	Z
x	x	x	x	x	x

**【説 明】**    第1オペランドで指定されるデスティネーション・オペランド (*dst*) から第2オペランドで指定されるソース・オペランド (*src*) を減算します。  
減算の結果はどこへも格納せず、フラグを変化させます。

- 【配 送 例】**
- **CMP** BL, BYTE PTR [IX]
  - **CMP** CW, [BP+4]

**【バ イ ト 数】**

ニモニック	オペランド	バイト数
<b>CMP</b>	<i>reg, reg'</i>	2
	<i>mem, reg</i>	2-4
	<i>reg, mem</i>	2-4
	<i>reg, imm</i>	3, 4
	<i>mem, imm</i>	3-6
	<i>acc, imm</i>	2, 3



【命令形式】

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
CMP	reg, reg'	0	0	1	1	1	0	1	W	1	1	reg					reg'	
	mem, reg	0	0	1	1	1	0	0	W	mod		reg					mem	
			(disp-low)								(disp-high)							
	reg, mem	0	0	1	1	1	0	1	W	mod		reg					mem	
			(disp-low)								(disp-high)							
	reg, imm	1	0	0	0	0	0	s	W	1	1	1	1	1			reg	
			imm8 or imm16-low								imm16-high							
	mem, imm	1	0	0	0	0	0	s	W	mod	1	1	1				mem	
			(disp-low)								(disp-high)							
			imm8 or imm16-low								imm16-high							
	ecc, imm	0	0	1	1	1	1	0	W	imm8 or imm16-low								
			imm16-high								—							

<b>CMP4S</b>	10進比較 Compare Nibble String
--------------	--------------------------------

**【命令形式】** CMP4S [DS1-spec:] dst-string, [Seq-spec:] src-string  
CMP4S

**【オペレーション】** BCDストリング (IY, CL) - BCDストリング (IX, CL)

**【オペランド】**

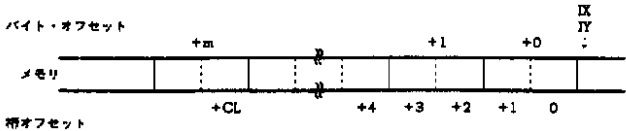
ニモニック	オペランド (dst, src)
<b>CMP4S</b>	[DS1-spec:] dst-string, [Seq-spec:] src-string
	なし

**【フラグ】**

AC	CY	V	P	S	Z
U	x	U	U	U	x

**【説明】** IYレジスタでアドレスされるバケットBCDストリングからIXレジスタでアドレスされるバケットBCDストリングを減算し、結果はストアされずフラグのみ影響を受けます。ストリング長 (BCD桁数) は、CLレジスタ (CLの内容がdならばd桁) によって決定され、1-254桁まで可能です。

デスティネーション・ストリングは、常にDS1レジスタで指されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。一方、ソース・ストリングは、デフォルト・セグメント・レジスタはDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。バケットBCDストリングのフォーマットを次に示します。



**注意** BCDストリング命令は常に偶数桁単位で動作します。このため桁数として偶数を指定したときは演算結果、および各フラグは正しく動きますが、桁数として奇数を指定した場合は奇数+1の偶数桁演算を実行し、演算結果、および

各フラグは偶数桁の演算結果を示しますので、桁数として奇数を指定する場合は次のようにしてください。  
 桁数が奇数のときのBCD比較命令は、最上位バイトの上位4ビットを"0"にしてから実行してください。

**【記 述 例】**   MOV IX, OFFSET VAR\_1  
                   MOV IY, OFFSET VAR\_2  
                   MOV CL, 4  
                   CMP4S

**【バ イ ト 数】**   2

**【命 令 語 形 式】**

二モニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>CMP4S</b>	[DS1-spec:]dst-string, [Seg-spec:]src-string	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0
	なし																

<b>CMPBK</b> <b>CMPBKB</b> <b>CMPBKW</b>	ブロック比較 Compare Block Compare Block Byte Compare Block Word
--	---

**【命令形式】** (repeat) **CMPBK** [*Seg-spec* : ] *src-block*, [*DS1-spec* : ] *dst-block*  
 (repeat) **CMPBKB**  
 (repeat) **CMPBKW**

**【オペレーション】** [*W*=0のとき] (*IX*) - (*IY*)  
                           *DIR*=0 : *IX*←*IX*+1, *IY*←*IY*+1  
                           *DIR*=1 : *IX*←*IX*-1, *IY*←*IY*-1  
 [*W*=1のとき] (*IX*+1, *IX*) - (*IY*+1, *IY*)  
                           *DIR*=0 : *IX*←*IX*+2, *IY*←*IY*+2  
                           *DIR*=1 : *IX*←*IX*-2, *IY*←*IY*-2

**【オペランド】**

ニモニック	オペランド
<b>CMPBK</b>	[ <i>Seg-spec</i> : ] <i>src-block</i> , [ <i>DS1-spec</i> : ] <i>dst-block</i>
<b>CMPBKB</b>	なし
<b>CMPBKW</b>	

**【フ ラ グ】**

AC	CY	V	P	S	Z
x	x	x	x	x	x

**【説 明】** *IX*レジスタでアドレスされるブロックから*IY*レジスタでアドレスされるブロックをバイトまたはワード・データ単位に繰り返し演算を行い、結果をフラグに反映させます。

*IX*レジスタ、*IY*レジスタは、次のバイト/ワード処理のために1バイト/ワード・データが処理されることに自動的にインクリメント (+1/+2) またはデクリメント (-1/-2) されます。ブロックの方向は、*DIR*フラグの状態によって決定されます。

バイトかワードかの指定は、**CMPBK**命令を用いる場合はオペランドの属性によって行われ、**CMPBKB**命令または**CMPBKW**命令を用いる場合はそれぞれバイト、ワード・タイプに直接指定されます。

デスティネーション・ブロックは常にDS1レジスタで指定されるセグメント内にロケートされる必要があります。セグメント・オーバーライドはできません。一方、ソース・ブロックは、デフォルト・セグメント・レジスタはDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。

**【記述例】** CMPBK BYTE\_VAR1, BYTE\_VAR2

**【バイト数】** 1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
<b>CMPBK</b>	[Seg-spec:] src-block, [DS1-spec:] dst-block	1	0	1	0	0	1	1	W
<b>CMPBKB</b>	なし								
<b>CMPBKW</b>									

<p><b>CMPM</b> <b>CMPMB</b> <b>CMPMW</b></p>	<p>アキュムレータとのブロック比較 Compare Multiple Compare Multiple Byte Compare Multiple Word</p>
--	---

**【命令形式】** (repeat) **CMPM** [DS1-spec:] dst-block  
 (repeat) **CMPMB**  
 (repeat) **CMPMW**

**【オペレーション】** [W=0のとき] AL ← (IY)  
 DIR=0: IY ← IY+1  
 DIR=1: IY ← IY-1  
 [W=1のとき] AW ← (IY+1, IY)  
 DIR=0: IY ← IY+2  
 DIR=1: IY ← IY-2

**【オペランド】**

ニモニック	オペランド
<b>CMPM</b>	[DS1-spec:]dst-block
<b>CMPMB</b>	なし
<b>CMPMW</b>	

**【フラグ】**

AC	CY	V	P	S	Z
x	x	x	x	x	x

**【説明】** アキュムレータ (AL/AW) の値から IYレジスタでアドレスされるブロックをバイトまたはワード・データ単位に繰り返し演算を行い、結果をフラグに反映させます。IYレジスタは、次のバイト/ワード処理のために1バイト/ワード・データが処理されるごとに自動的にインクリメント (+1/+2) またはデクリメント (-1/-2) されます。ブロックの方向は、DIRフラグの状態によって決定されます。バイトかワードかの指定は、CMPM命令を用いる場合はオペランドの属性によって行われ、CMPMB命令またはCMPMW命令を用いる場合はそれぞれバイト、ワード・タイプに直接指定されます。デスティネーション・ブロックは、常にDS1レジスタで指定されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。

**【記述例】**

- MOV AW, 5555H
- MOV BW, 1000H
- MOV Y, BW
- REPC CMPM WORD PTR [Y]
- REPNC CMPMW
- REPZ CMPMB

**【バイト数】** 1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
<b>CMPM</b>	[DSI-spec:]dst-block	1	0	1	0	1	1	1	W
<b>CMPMB</b>	なし								
<b>CMPMW</b>									

<b>CVTBD</b>	2 進→アンパクト10進変換 Convert Binary to Decimal
--------------	---

【命令形式】 CVTBD

【オペレーション】 AH←AL+0AH  
AL←AL%0AH

【オペランド】

ニモニック	オペランド
CVTBD	なし

【フ ラ グ】

AC	CY	V	P	S	Z
U	U	U	×	×	×

【説 明】 ALレジスタの2進数8ビットを2桁のアンパクト10進数に変換します。  
AHレジスタはALレジスタの値を10で割った商で置き換えられ、次にALレジスタがその乗算の剰余で置き換えられます。

【記 述 例】 MOV AL, 30H  
CVTBD

【バ イ ト 数】 2

【命令書形式】

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
CVTBD	なし	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0



<b>CVTBW</b>	ワード符号拡張 Convert Byte to Word
--------------	---------------------------------

**【命令形式】** CVTBW

**【オペレーション】** AL<80Hのとき：AH←0  
AL≥80Hのとき：AH←FFH

**【オペランド】**

ニモニック	オペランド
CVTBW	なし

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** ALレジスタ内のバイトの符号をAHレジスタに拡張します。バイト除算を実行する前に、あるバイトから倍長（ワード）の被除数を得るのに有効です。

**【記述例】** MOV AL, BUF1; BUF1はバイト変数  
CVTBW  
MOV DL, 60  
DIV DL

**【バイト数】** 1

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
CVTBW	なし	1	0	0	1	1	0	0	0

<b>CVTDB</b>	アンパクト10進→2進変換 Convert Decimal to Binary
--------------	--

**【命令形式】** CVTDB

**【オペレーション】** AL←AH×0AH+AL  
AH←0

**【オペランド】**

ニモニック	オペランド
CVTDB	なし

**【フ ラ グ】**

AC	CY	V	P	S	Z
U	U	U	×	×	×

**【取 扱 方 法】** AHレジスタおよびALレジスタの2桁のアンパクト10進数を16ビットの2進数に変換します。  
ALレジスタはAHレジスタの値に10を掛けた結果にALレジスタの値を加えたもので置き換えられ、AHレジスタは0で置き換えられます。

**【記 述 例】** MOV AW, [BW]  
CVTDB

**【バ イ ト 数】** 2

**【命令形式】**

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
CVTDB	なし	1	1	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0

**CVTWL**

ダブル・ワード符号拡張  
Convert Word to Long Word

【命令形式】 CVTWL

【オペレーション】 AW<8000Hのとき：DW←0  
AW≥8000Hのとき：DW←FFFFH

【オペランド】

ニモニック	オペランド
CVTWL	なし

【フラグ】

AC	CY	V	P	S	Z

【説明】 AWレジスタのワードの符号をDWレジスタに拡張します。ワード除算を実行する前に、あるワードから倍長（ダブル・ワード）の被除数を得るのに有効です。

【記述例】 MOV AW, BUFFER  
CVTWL  
DIV CW

【バイト数】 1

【命令形式】

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
CVTWL	なし	1	0	0	1	1	0	0	1

**DBNZ** CW≠0による条件ループ  
Decrement and Branch if Not Zero

【命令形式】 DBNZ short-label

【オペレーション】 CW←CW-1  
CW≠0のとき：PC←PC+ext-disp8

【オペランド】

ニモニック	オペランド
DBNZ	short-label

【フ ラ グ】

AC	CY	V	P	S	Z

【説 明】 CWレジスタの値をデクリメント (-1) し、その結果CWレジスタの値が0でなければ、現在のPCの値に8ビット・ディスプレイメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令に置かれているセグメント内であつ、-128~+127バイトのアドレス内にジャンプできます。  
条件不成立のときは次の命令に進みます。

【配 送 例】 LP21 :  
          :  
          SHL AL, 1  
          DBNZ LP21 ; LP21=レーベル

【バ イ ト 数】 2

【命令形式】

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
DBNZ	short-label	1	1	1	0	0	0	1	0	disp8							

<b>DBNZE</b>	CW≠0かつZ=1による条件ループ Decrement and Branch if Not Zero and Equal
--------------	---

**【命令形式】** DBNZE short-label

**【オペレーション】** CW←CW-1  
CW≠0でZ=1のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
<b>DBNZE</b>	short-label

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【説 明】** CWレジスタの値をデクリメント (-1) し、その結果CWレジスタの値が0でなく、かつZフラグがセット (1) されていれば、現在のPCの値に8ビット・ディスプレイメント (実際にはサイン拡張された16ビット) を加えた値をロードします。この命令の置かれているセグメント内であつ、-128~+127バイトのアドレス内にブランチできます。  
条件不成立のときは次の命令に進みます。

**【配 送 例】** LP20 :  
                  :  
                  AND AL, BL  
                  DBNZE LP20 ; LP20=レーベル

**【バ イ ト 数】** 2

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>DBNZE</b>	short-label	1	1	1	0	0	0	0	1	disp8							

<b>DBNZNE</b>	CW≠0かつZ=0による条件ループ Decrement and Branch if Not Zero and Not Equal
---------------	---

**【命令形式】** DBNZNE short-label

**【オペレーション】** CW←CW-1  
CW≠0でZ=0のとき: PC←PC+ext-disp8

**【オペランド】**

ニモニック	オペランド
DBNZNE	short-label

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** CWレジスタの値をデクリメント (-1) したあと、CWレジスタの値が0でなくZフラグがクリアされていれば、現在のPCの値に8ビット・ディスプレイメント (実際にはサイン拡張された16ビット) を加えた値をロードします。  
この命令の置かれているセグメント内かつ、-128~+127バイトのアドレス内にジャンプできます。  
条件不成立のときは次の命令に進みます。

**【記述例】** LP19:  
:  
AND AL, 0FFH  
DBNZNE SHORT LP19 ; LP19=レーベル

**【バイト数】** 2

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
DBNZNE	short-label	1	1	1	0	0	0	0	0								

<b>DEC</b>	デクリメント Decrement
------------	---------------------

**【命令形式】** DEC dst

**【オペレーション】** dst ← dst - 1

**【オペランド】**

ニモニック	オペランド
DEC	reg8
	mem
	reg16

**【フラグ】**

AC	CF	V	P	S	Z
x		x	x	x	x

**【説明】** デスティネーション・オペランド (dst) の内容をデクリメント (-1) します。

- 【記述例】**
- DEC BW
  - DEC BP
  - DEC IX
  - DEC IY

**【バイト数】**

ニモニック	オペランド	バイト数
DEC	reg8	2
	mem	2-4
	reg16	1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7 6 5 4 3 2 1 0								7 6 5 4 3 2 1 0							
DEC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	1	reg		
	mem	1	1	1	1	1	1	1	W	mod	0	0	1	mem			
	(disp-low)								(disp-high)								
reg16		0	1	0	0	1	reg	—									

**DI** マスクابل割り込みの禁止  
Disable Interrupt

【命令形式】 DI

【オペレーション】 IE←0

【オペランド】

モニック	オペランド
DI	なし

【フラグ】

AC	CY	V	P	S	Z	IE
						0

【説明】 IEフラグをリセット (0) し、マスクابل割り込み要求を禁止します。  
この命令によってノンマスクابل割り込み要求およびソフトウェア割り込み要求は禁止されません。

【記述例】 DI  
PUSH R

【バイト数】 1

【命令書形式】

モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
DI	なし	1	1	1	1	1	0	1	0



**DISPOSE** スタック・フレームの削除  
Dispose a Stack Frame

【命令形式】 DISPOSE

【オペレーション】 SP←BP  
BP←(SP+1, SP)  
SP←SP+2

【オペランド】

モニック	オペランド
DISPOSE	なし

【フラグ】

AC	CY	V	P	S	Z

【説明】 この命令は、PREPARE命令で生成されたスタック・フレームを1フレーム分リリースします。  
BPには、1つ前のフレームを指すポインタ値をロードし、SPには、フレームの最下位を示すポインタ値をロードします。

【記述例】 DISPOSE

【バイト数】 1

【命令書形式】

モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
DISPOSE	なし	1	1	0	0	1	0	0	1

<b>DIV</b>	符号付き除算 Divide Signed
------------	-------------------------

【命令形式】 DIV dat

【オペランド、オペレーション】

ニモニック	オペランド (dat)	オペレーション
<b>DIV</b>	reg8	temp ← AW temp + dat > 0 ° temp + dat ≤ 7FH または temp + dat < 0 ° temp + dat > 0 - 7FH - 1 のとき AH ← temp % dat AL ← temp + dat temp + dat > 0 ° temp + dat > 7FH または temp + dat < 0 ° temp + dat ≤ 0 - 7FH - 1 のとき 商と剰余は不定
	mem8	TA ← (001H, 000H) TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS PS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA
	reg16	temp ← DW, AW temp + dat > 0 ° temp + dat ≤ 7FFFH または temp + dat < 0 ° temp + dat > 0 - 7FFFH - 1 のとき DW ← temp % dat AW ← temp + dat temp + dat > 0 ° temp + dat > 7FFFH または temp + dat < 0 ° temp + dat ≤ 0 - 7FFFH - 1 のとき 商と剰余は不定
	mem16	TA ← (001H, 000H) TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS PS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA

**【フ ラ グ】**

AC	CY	V	P	S	Z
U	U	U	U	U	U

**【説 明】**

○src=reg8またはsrc=mem8のとき

AWレジスタの値をデスティネーション・オペランド (dat) の内容で符号付き除算します。商はALレジスタに格納し、余りはAHレジスタに格納します。

正の商の最大値は+127 (7FH) で、負の商の最小値は-127 (81H) です。商が正で、最大値を超えたとき、または商が負で最小値より小さくなったときは、ベクタ0割り込みが発生し、商と余りは不定となります (特にsrc=00Hのときに発生します)。また、整数でない商は整数に切り縮められ、余りは被除数と同じ符号を持ちます。

○src=reg16またはsrc=mem16のとき

AWレジスタの値とDWレジスタの値をデスティネーション・オペランド (dat) の内容で符号付き除算します。商はAWレジスタに格納し、余りはDWレジスタに格納します。

正の商の最大値は+32767 (7FFFH) で、負の商の最小値は-32767 (8001H) です。商が正で、最大値を超えたとき、または商が負で最小値より小さくなったときは、ベクタ0割り込みが発生し、商と余りは不定となります (特にsrc=0000Hのときに発生します)。また、整数でない商は整数に切り縮められ、余りは被除数と同じ符号を持ちます。

**【記 述 例】**

32ビット・データDW: AWをメモリ0: 50の内容で割る。

```
MOV  BW, 0
MOV  DSO, BW
MOV  IX, 50H
DIV  DSO:WORD PTR [IX]
```

**【バ イ ト 数】**

ニモニック	オペランド	バイト数
<b>DIV</b>	reg8	2
	mem8	2-4
	reg16	2
	mem16	2-4

【命令形式】

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
DIV	reg8	1	1	1	1	0	1	1	0	1	1	1	1	1			reg
	mem8	1	1	1	1	0	1	1	0	mod	1	1	1			mem	
			(disp-low)							(disp-high)							
	reg16	1	1	1	1	0	1	1	1	1	1	1	1	1		reg	
mem16	1	1	1	1	0	1	1	1	mod	1	1	1			mem		
		(disp-low)							(disp-high)								

<b>DIVU</b>	符号なし除算 Divide Unsigned
-------------	---------------------------

【命令形式】 DIVU dst

【オペランド、オペレーション】

ニモニック	オペランド (dst)	オペレーション
DIVU	reg8	temp ← AW temp + dst ≤ FFH のとき： AH ← temp % dst AL ← temp + dst temp + dst > FFH のとき： TA ← (001H, 000H) TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS RS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA
	mem8	TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS RS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA
	reg16	temp ← DW, AW temp + dst ≤ FFFFH のとき： DW ← temp % dst AW ← temp + dst temp + dst > FFFFH のとき： TA ← (001H, 000H) TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS RS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA
	mem16	TC ← (003H, 002H) SP ← SP - 2, (SP + 1, SP) ← PSW IE ← 0, BRK ← 0 SP ← SP - 2, (SP + 1, SP) ← PS RS ← TC SP ← SP - 2, (SP + 1, SP) ← PC PC ← TA

【フラグ】

AC	CY	V	P	S	Z
U	U	U	U	U	U

**【説明】**      ○*src=reg8*または*src=mem8*のとき  
 AWレジスタの値をデスティネーション・オペランド (*dst*) の内容で符号なし除算します。商はALレジスタに格納し、余りはAHレジスタに格納します。  
 商がALレジスタの容量 (FFH) を越えた場合は、ベクタ0割り込みが発生し、商と余りは不定となります(特に*src=00H*のときに発生します)。また、整数でない商は整数に切り詰められます。

○*src=reg16*または*src=mem16*のとき  
 AWレジスタの値とDWレジスタの値をデスティネーション・オペランド (*dst*) の内容で符号なし除算します。商はAWレジスタに格納し、余りはDWレジスタに格納します。  
 商がAWレジスタの容量 (FFFFH) を越えた場合、ベクタ0割り込みが発生し、商と余りは不定となります(特に*src=0000H*のときに発生します)。また、整数でない商は整数に切り詰められます。

**【記述例】**      5+3の除算をする。  
                   MOV    AW, 5  
                   MOV    DL, 3  
                   DIVU   DL  
                   ; AH=2 AL=1

**【バイト数】**

ニモニック	オペランド	バイト数
<b>DIVU</b>	<i>reg8</i>	2
	<i>mem8</i>	2-4
	<i>reg16</i>	2
	<i>mem16</i>	2-4

**【命令形式】**

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
<b>DIVU</b>	<i>reg8</i>	1	1	1	1	0	1	1	0	1	1	1	1	0				reg
	<i>mem8</i>	1	1	1	1	0	1	1	0	mod	1	1	0				mem	
		(disp-low)								(disp-high)								
	<i>reg16</i>	1	1	1	1	0	1	1	1	1	1	1	1	0				reg
<i>mem16</i>		1	1	1	1	0	1	1	1	mod	1	1	0				mem	
	(disp-low)								(disp-high)									

<b>DS0 :</b>	セグメント・オーバーライド・プリフィクス	
<b>DS1 :</b>		Data Segment 0
<b>PS :</b>		Data Segment 1
<b>SS :</b>		Program Segment
		Stack Segment

**【命令形式】** DS0 :  
 DS1 :  
 PS :  
 SS :

**【オペレーション】** セグメント・オーバーライド・プリフィクス

**【オペランド】**

ニモニック	オペランド
DS0 :	なし
DS1 :	
PS :	
SS :	

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【説 明】** セグメント・オーバーライド可能なメモリ・オペランドをアクセスする際に、オペラ  
 ンドに付記してそのとき使用されるセグメント・レジスタを指定します。この命令  
 を直接記述しなくとも、ASSUME（アセンブラ擬似命令）を使用することによっ  
 て、セグメント・オーバーライド指定をアセンブラに行わせることができます。

**注意** この命令と次の命令との間では、ハードウェア割り込み（マスカブル割り込  
 み、ノンマスカブル割り込み）要求およびシングルステップ・ブレークは受  
 け付けられません。

**【記 述 例】** MOV DW, DS1 : [BW] ; デフォルトのセグメント・レジスタはDS0

**【バ イ ト 数】** 1

【命令形式】

二モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
D80:	なし	0	0	1	reg	1	1	0	
D81:									
FB:									
SB:									



**EI** マスクابل割り込みの許可  
Enable Interrupt

【命令形式】 EI

【オペレーション】 IE←1

二モニック	オペランド
EI	なし

AC	CY	V	P	S	Z	IE
						1

【説明】 IEフラグをセット (1) し、マスクابل割り込み要求を許可します。  
ただし、実際に割り込み許可状態になるのはEI命令に続く1命令を実行したあとです。

【記述例】 POP R  
EI

【バイト数】 1

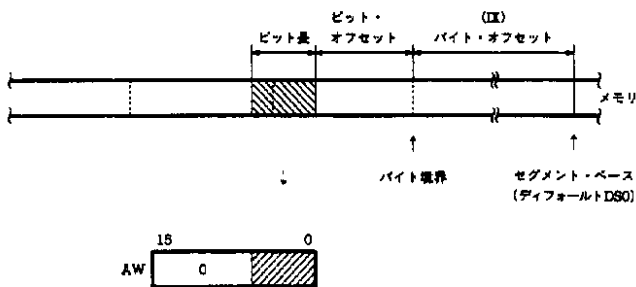
二モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
EI	なし	1	1	1	1	1	0	1	1

**EXT**

ビット・フィールドの抽出  
Extract Bit Field

**【命令形式】** EXT dst, src

**【オペレーション】** AW←16ビット・フィールド



**【オペランド】**

二モニック	オペランド (dst, src)
EXT	reg8, reg8'
	reg8, imm4

**【フラグ】**

AC	CY	V	P	S	Z
U	U	U	U	U	U

**【説明】** IXレジスタでアドレスされるバイト・オフセットおよび第1オペランドの8ビット・レジスタで指定されるビット・オフセットによって決定されるメモリ領域から、ソース・オペランド (src) によって指定されるビット長のビット・フィールド・データをAWレジスタへロードします。このときAWレジスタの余りの上位ビットには0がロードされます。  
転送終了後、IXレジスタおよび第1オペランドで指定される8ビット・レジスタは次のビット・フィールドを示すために自動的に次のように更新されます。

```

reg8←reg8+src+1
if reg8>15 then
(
  reg8←reg8-16
  IX←IX+2
)
  
```

ビット・オフセット (最長15ビット) を指定する第1オペランドの8ビット・レジスタの値は0-15のみ有効です。また、ビット長(最長16ビット)を指定するソース・オペランド (src) の値は0-15のみが有効で、0が1ビット長を、15が16ビット長を指定します。

ビット・フィールド・データはメモリのバイト境界にまたがることができます。ソースのビット・フィールドはデフォルト・セグメント・レジスタがDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。

注意 reg8またはreg8'の上位4ビットは0にしてください。

- [記 述 例]
- EXT CL, DL
  - EXT CL, 8

[バ イ ト 数]

ニモニック	オペランド	バイト数
EXT	reg8, reg8'	3
	reg8, imm4	4

[命 令 形 式]

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
EXT	reg8, reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1
		1	1		reg'		reg		—								
	reg8, imm4	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1
		1	i	0	0	0		reg	imm4								

<b>FPO1</b>	浮動小数点演算用コプロセッサ制御
	Floating Point Operation 1

- 【命令形式】**
- ① FPO1 fp-op
  - ② FPO1 fp-op, mem

**【オペランド、オペレーション】**

命令形式①

モニタック	オペランド	オペレーション
FPO1	fp-op	ノ・オペレーション

命令形式②

モニタック	オペランド	オペレーション
FPO1	fp-op, mem	データ・バスー (mem)

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【説 明】** 命令形式①：外部に接続される浮動小数点演算用コプロセッサ用の命令です。CPUがこの命令をフェッチしたとき、自身では何もせず、演算処理はコプロセッサにまかせます。

命令形式②：この命令は外部に接続される浮動小数点演算用コプロセッサ用の命令です。CPUがこの命令をフェッチしたとき演算処理はコプロセッサにまかせ、必要に応じてコプロセッサに対する補助的な処理（実効アドレス計算、物理アドレス生成、そしてメモリ・リード・サイクルの起動）のみを行います。  
CPUは、CPUが起動するメモリ・リード・サイクルにおいて、データ・バス上のリード・データは一切取り込みません。

- 【記 述 例】**
- FPO1 010101010E
  - FPO1 OFFH
  - FPO1 6. BYTE PTR [IX]
  - FPO1 4. WORD\_VAR

**【バ イ ト 数】**

ニモニック	オペランド	バイト数
FPO1	fp-op	2
	fp-op, mem	2-4

**【命 令 形 式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
FPO1	fp-op	1	1	0	1	1	X	X	X	1	1	Y	Y	Y	Z	Z	Z
	fp-op, mem	1	1	0	1	1	X	X	X	mod	Y	Y	Y	mem			
		(disp-low)								(disp-high)							

**FPO2** 浮動小数点演算用コプロセッサ制御  
Floating Point Operation 2

- 【命令形式】** ① FPO2 fp-op  
② FPO2 fp-op, mem

**【オペランド、オペレーション】**

命令形式①

ニモニック	オペランド	オペレーション
FPO2	fp-op	ノー・オペレーション

命令形式②

ニモニック	オペランド	オペレーション
FPO2	fp-op, mem	データ・バス (mem)

**【フ ラ グ】**

AC	CY	V	P	S	Z

**【解 明】** 命令形式①：外部に接続される浮動小数点演算用コプロセッサ用の命令です。CPUがこの命令をフェッチしたとき、自身では何もせず、演算処理はコプロセッサにまかせます。

命令形式②：この命令は外部に接続される浮動小数点演算用コプロセッサ用の命令です。CPUがこの命令をフェッチしたとき演算処理はコプロセッサにまかせ、必要に応じてコプロセッサに対する補助的な処理（実効アドレス計算、物理アドレス生成、そしてメモリ・リード・サイクルの起動）のみを行います。  
CPUは、CPUが起動するメモリ・リード・サイクルにおいて、データ・バス上のリード・データは一切取り込みません。

- 【記述例】**
- FPO2 010101010B
  - FPO2 0FFH
  - FPO2 0101B, BYTE PTR [IX]
  - FPO2 1010B, WORD\_VAR

**【バイト数】**

ニモニック	オペランド	バイト数
FPO2	fp-op	2
	fp-op, mem	2-4

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
FPO2	fp-op	0	1	1	0	0	1	1	X	1	1	Y	Y	Y	Z	Z	Z
	fp-op, mem	0	1	1	0	0	1	1	X	mod	Y	Y	Y	mem			
		(disp-low)								(disp-high)							

<b>HALT</b>	ホールド Halt
-------------	--------------

**【命令形式】** HALT

**【オペレーション】** CPU Halt

**【オペランド】**

二モニック	オペランド
HALT	なし

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** CPUへのクロック供給を停止し、スタンバイ状態にします。  
スタンバイ状態は次の要因によって解除されます。

- リセット入力
- マスカブル割り込み要求入力
- ノンマスカブル割り込み要求入力

**【記述例】** HALT

**【バイト数】** 1

**【命令形式】**

二モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
HALT	なし	1	1	1	1	0	1	0	0



<b>IN</b>	I/Oデバイスからのデータ入力 Input
-----------	--------------------------

**【命令形式】** IN dst, src

**【オペランド、オペレーション】**

ニモニック	オペランド (dst, src)	オペレーション
<b>IN</b>	acc, imm8	[W=0のとき]AL←(imm8) [W=1のとき]AH←(imm8+1), AL←(imm8)
	acc, DW	[W=0のとき]AL←(DW) [W=1のとき]AH←(DW+1), AL←(DW)

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** デスティネーション・オペランド (dst) で指定されるアキュムレータ (ALレジスタまたはAHレジスタ) にソース・オペランド (src) で指定されるI/Oデバイスのレジスタ内容を転送します。

**【記述例】** ポート・アドレスODAHの内容をALレジスタに転送する。

```
MOV DW, ODAH
IN AL, DW
```

**【バイト数】**

ニモニック	オペランド	バイト数
<b>IN</b>	acc, imm8	2
	acc, DW	1

**【命令形式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
<b>IN</b>	acc, imm8	1	1	1	0	0	1	0	W	imm8							
	acc, DW	1	1	1	0	1	1	0	W	—							

<b>INC</b>	インクリメント Increment
------------	----------------------

【命令形式】 INC *dst*

【オペレーション】  $dst \leftarrow dst + 1$

【オペランド】

ニモニック	オペランド ( <i>dst</i> )
INC	reg8
	mem
	reg16

【フラグ】

AC	CF	V	P	S	Z
x		x	x	x	x

【説明】 デスティネーション・オペランド (*dst*) の内容をインクリメント (+1) します。

- 【記述例】
- INC DW
  - INC BP
  - INC SP

【バイト数】

ニモニック	オペランド	バイト数
INC	reg8	2
	mem	2-4
	reg16	1

【命令書形式】

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
INC	reg8	1	1	1	1	1	1	1	0	1	1	0	0	0			reg
	mem	1	1	1	1	1	1	1	W	mod	0	0	0			mem	
		(disp-low)							(disp-high)								
	reg16	0	1	0	0	0		reg	—								

<b>INM</b>	I/O-メモリのブロック転送 Input Multiple
------------	----------------------------------

**【命令形式】** (repeat) INM [DS1-spec:] dst-block, DW

**【オペレーション】**

[W=0のとき] (IY) ← (DW)  
 DIR=0: IY←IY+1  
 DIR=1: IY←IY-1

[W=1のとき] (IY+1, IY) ← (DW+1, DW)  
 DIR=0: IY←IY+2  
 DIR=1: IY←IY-2

**【オペランド】**

ニモニック	オペランド
INM	[DS1-spec:] dst-block, DW

**【フラグ】**

AC	CY	V	F	S	Z

**【説明】** DWレジスタでアドレスされるI/Oデバイスのレジスタ内容を、IYレジスタでアドレスされるメモリに転送します。繰り返し転送回数は、ペアで使用されるリピート・プリフィックスのREP命令によって制御されます。繰り返し転送の際、DWレジスタの内容(I/Oデバイスのアドレス)は固定ですが、IYレジスタは次のバイト/ワード転送のために1バイト/ワード・データが転送されることに自動的にインクリメント(+1/+2)またはデクリメント(-1/-2)されます。ブロックの方向は、DIRフラグの状態によって決定されます。

バイトかワードかの指定は、オペランドの属性によって行われます。

INM命令は、リピート・プリフィックスのREP命令とともに使用されます。

デスティネーション・ブロックは、常にDS1レジスタで指定されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。

**【記述例】** ●メモリワークエリアにポート・アドレスODAHの内容（バイト・データ）をロードする。

```
MOV AW, 0
MOV DS1, AW
MOV IY, SCH
MOV DW, ODAH
INM DS1:BYTE PTR [IY], DW
```

●メモリ0:0H~0:FFHにポート・アドレスODAHの内容（バイト・データ）をロードする。

```
MOV AW, 0
MOV DS1, AW
MOV IY, 0
MOV DW, ODAH
MOV CW, OFFH
REP INM DS1:BYTE PTR [IY], DW
```

**【バイト数】** 1

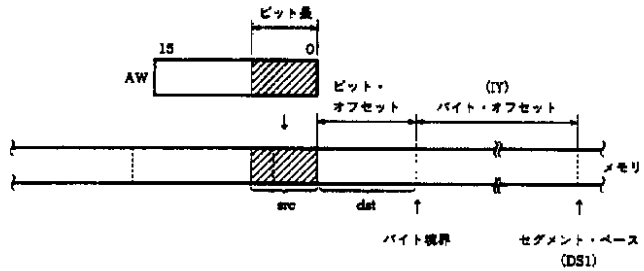
**【命令書形式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
INM	[DS1-spec:] dst-block, DW	0	1	1	0	1	1	0	W

<b>INS</b>	ビット・フィールドの挿入 Insert Bit Field
------------	----------------------------------

**【命令形式】**   INS dst, src

**【オペレーション】**   16ビット・フィールド←AW



**【オペランド】**

ニモニック	オペランド (dst, src)
<b>INS</b>	reg8, reg8'
	reg8, imm4

**【フラグ】**

AC	CY	V	P	S	Z
U	U	U	U	U	U

**【説明】**   AWレジスタの16ビット・データのうち、ソース・オペランド (src) で指定される長さの下位ビット・データをDS1レジスタとIYレジスタでアドレスされるバイト・オフセットおよび第1オペランドの8ビット・レジスタで指定されるビット・オフセットによって決定されるメモリ領域へ転送します。  
転送終了後、IYレジスタおよび第1オペランドで指定される8ビット・レジスタは、次のビット・フィールドを示すために自動的に次のように更新されます。

```

reg8 ← reg8 + src + 1
if reg8 > 15 then
{
    reg8 ← reg8 - 16
    PC ← PC + 2
}
    
```

ビット・オフセット（最長15ビット）を指定する第1オペランドの8ビット・レジスタの値は0-15だけが有効です。また、ビット長（最長16ビット）を指定するソース・オペランド（src）の値は0-15だけが有効で、0が1ビット長を、15が16ビット長を指定します。

ビット・フィールド・データはメモリのバイト境界にまたがることができます。デスティネーションのビット・フィールドは常にDS1レジスタで指定されるセグメントにロケートされる必要があり、セグメント・オーバーライドはできません。

注意 reg8またはreg8'の上位4ビットは0にしてください。

- 【配 送 例】**
- INS DL, CL
  - INS DL, 12

**【バ イ ト 数】**

ニモニック	オペランド	バイト数
INS	reg8, reg8'	3
	reg8, imm4	4

**【命 令 形 式】**

ニモニック	オペランド	オペレーション・コード															
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
INS	reg8, reg8'	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1
		1	1	reg'				reg				—					
	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	
	reg8, imm4	1	1	0	0	reg				imm4							

<b>LDEA</b>	実効アドレスのロード Load Effective Address
-------------	--------------------------------------

**【命令形式】** LDEA reg16, mem16

**【オペレーション】** reg16 ← mem16

**【オペランド】**

ニモニック	オペランド
LDEA	reg16, mem16

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** 第1オペランドで指定される16ビット汎用レジスタに、第2オペランドで生成される実効アドレス（オフセット）をロードします。  
 TRANS命令やプリミティブ・ブロック転送命令でオペランドを指定するために自動的に用いられるレジスタ等にオペランド・アドレスの先頭値をセット（1）するときなどに用いられます。

**【記述例】** AWレジスタにINT\_PROCという手続きの実行アドレスのオフセットをロードします。

```
LDEA AW, INT_PROC
LDEA AW, [BP], VAR01+2
```

**【バイト数】** 2-4

**【命令書形式】**

ニモニック	オペランド	オペレーション・コード																
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
LDEA	reg16, mem16	1	0	0	0	1	1	0	1	mod	reg	mem						
		(disp-low)						(disp-high)										

<b>LDM</b> <b>LDMB</b> <b>LDMW</b>	ブロック・ロード Load Multiple Load Multiple Byte Load Multiple Word
--	---

【命令形式】 (repeat) LDM [Seg-spec:] arc-block  
 (repeat) LDMB  
 (repeat) LDMW

【オペレーション】 [W=0のとき] AL←(IX)  
 DIR=0: IX←IX+1  
 DIR=1: IX←IX-1  
 [W=1のとき] AW←(IX+1, IX)  
 DIR=0: IX←IX+2  
 DIR=1: IX←IX-2

【オペランド】

ニモニック	オペランド
LDM	[Seg-spec:]arc-block
LDMB	なし
LDMW	

【フラグ】

AC	CY	V	P	S	Z

【説明】 IXレジスタでアドレスされるブロックをバイトまたはワード単位にアキュムレータ (AL/AW) に繰り返し転送します。  
 IXレジスタは次のバイト/ワード処理のために1バイト/ワード・データが処理されることに自動的にインクリメント (+1/+2) またはデクリメント (-1/-2) されます。ブロックの方向は、DIRフラグの状態によって決定されます。  
 バイトかワードかの指定は、LDM命令を用いる場合はオペランドの属性によって行われ、LDMB命令またはLDMW命令を用いる場合はそれぞれバイト、ワード・タイプに直接指定されます。  
 ソース・ブロックは、デフォルト・セグメント・レジスタはDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。



**【記 述 例】** ●REP LDM DS1:BYTE\_VAR; DS1セグメント  
 ●REP LDME ; DS0セグメント

**【バ イ ト 数】** 1

**【命 令 書 形 式】**

ニモニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
LDM	[Seg-spec:]src-block	1	0	1	0	1	1	0	W
LDME	な し								
LDMW									

<b>MOV</b>	データ転送 Move
------------	---------------

- 【命令形式】**
- ① MOV dst, src
  - ② MOV dst1, dst2, src

**【オペランド, オペレーション】**

命令形式①

二モニック	オペランド (dst, src)	オペレーション
<b>MOV</b>	reg, reg'	dst←src
	mem, reg	
	reg, mem	
	mem, imm	
	reg, imm	
	acc, dmem	[W=0のとき] AL←(dmem) [W=1のとき] AH←(dmem+1), AL←(dmem)
	dmem, acc	[W=0のとき] (dmem)←AL [W=1のとき] (dmem+1)←AH, (dmem)←AL
	sreg, reg16	dst←src
	sreg, mem16	
	reg16, sreg	
	mem16, sreg	
	AH, PSW	AH←S, Z, ×, AC, ×, P, ×, CY
PSW, AH	S, Z, ×, AC, ×, P, ×, CY←AH	

命令形式②

二モニック	オペランド (dst1, dst2, src)	オペレーション
<b>MOV</b>	DS0, reg16, mem32	reg16←(mem32) DS0←(mem32+2)
	DS1, reg16, mem32	reg16←(mem32) DS1←(mem32+2)

**【フ ラ グ】** オペランドがPSW, AHの場合 左記以外

AC	CY	V	P	S	Z
x	x		x	x	x

AC	CY	V	P	S	Z

**【説明】** 命令形式①：第1オペランドで指定されるデスティネーション・オペランド (dst) に、第2オペランドで指定されるソース・オペランド (src) の内容を転送します。

オペランドがAH, PSWの場合は、S, Z, AC, P, CYの各フラグをAHレジスタに転送します。AHレジスタのビット1, ビット3, ビット5は不定となります。

オペランドがPSW, AHの場合は、AHレジスタのビット2, ビット4, ビット6, ビット7をそれぞれPSWのS, Z, AC, P, CYの各フラグに転送します。

**注意** dst=eregまたはsrc=eregの場合、次の命令との間にハードウェア割り込み(マスクابل割り込み, ノンマスクابل割り込み)要求およびシングルステップ・ブレークは受け付けられません。

命令形式②：ソース・オペランド (src) でアドレスされる32ビット・メモリの下位16ビット (32ビット・ポインタ変数のオフセット・ワード) を、デスティネーション・オペランド2 (dst2) の16ビット・レジスタに、上位16ビット (セグメント・ワード) をデスティネーション・オペランド1 (dst1) のセグメント・レジスタ (DS0レジスタまたはDS1レジスタ) に転送します。

**【記述例】** メモリ0:50Hに55Hを書く動作

```
MOV AW, 0
MOV DS1, AW
MOV IX, 50H
MOV DL, 55H
MOV DS1: [IX], DL
```

【バイト数】

モニック	オペランド	バイト数
MOV	reg, reg'	2
	mem, reg	2-4
	reg, mem	
	mem, imm	3-6
	reg, imm	2, 3
	acc, dmem	3
	dmem, acc	
	sreg, reg16	2
	sreg, mem16	2-4
	reg16, sreg	2
	mem16, sreg	2-4
	DS0, reg16, mem32	
	DS1, reg16, mem32	
	AH, PSW	1
	PSW, AH	

【命令書形式】

ニモニック	オペランド	オペレーション・コード														
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
MOV	reg, reg'	1	0	0	0	1	0	1	W	1	1			reg		reg'
	mem, reg	1	0	0	0	1	0	0	W	mod			reg		mem	
		(disp-low)							(disp-high)							
	reg, mem	1	0	0	0	1	0	1	W	mod			reg		mem	
		(disp-low)							(disp-high)							
	mem, imm	1	1	0	0	0	1	1	W	mod	0	0	0		mem	
		(disp-low)							(disp-high)							
		imm8 or imm16-low							imm16-high							
	reg, imm	1	0	1	1	1	W		reg	imm8 or imm16-low						
		imm16-high							—							
	acc, dmem	1	0	1	0	0	0	0	W	addr-low						
		addr-high							—							
	dmem, acc	1	0	1	0	0	0	1	W	addr-low						
		addr-high							—							
	sreg, reg16	1	0	0	0	1	1	1	0	1	1	0		sreg		reg
	sreg, mem16	1	0	0	0	1	1	1	0	mod	0		sreg		mem	
		(disp-low)							(disp-high)							
	reg16, sreg	1	0	0	0	1	1	0	0	1	1	0		sreg		reg
	mem16, sreg	1	0	0	0	1	1	0	0	mod	0		sreg		mem	
		(disp-low)							(disp-high)							
DS0, reg16, mem32	1	1	0	0	0	1	0	1	mod			reg		mem		
	(disp-low)							(disp-high)								
DS1, reg16, mem32	1	1	0	0	0	1	0	0	mod			reg		mem		
	(disp-low)							(disp-high)								
AH, PSW	1	0	0	1	1	1	1	1	—							
PSW, AH	1	0	0	1	1	1	1	0	—							

<p><b>MOVBK</b> <b>MOVBKB</b> <b>MOVBKW</b></p>	<p>ブロック転送 Move Block Move Block Byte Move Block Word</p>
---	--

**【命令形式】** (repeat) **MOVBK** [DS1-spec:] dst-block, [Seg-spec:] src-block  
 (repeat) **MOVBKB**  
 (repeat) **MOVBKW**

**【オペレーション】** [W=0のとき] (IY) ← (IX)  
 DIR=0: IX←IX+1, IY←IY+1  
 DIR=1: IX←IX-1, IY←IY-1  
 [W=1のとき] (IY+1, IY) ← (IX+1, IX)  
 DIR=0: IX←IX+2, IY←IY+2  
 DIR=1: IX←IX-2, IY←IY-2

**【オペランド】**

二モニック	オペランド
<b>MOVBK</b>	[DS1-spec:] dst-block, [Seg-spec:] src-block
<b>MOVBKB</b>	なし
<b>MOVBKW</b>	

**【フラグ】**

AC	CY	V	P	S	Z

**【説明】** IXレジスタでアドレスされるブロックを、IYレジスタでアドレスされるブロックに、バイトまたはワード・データの繰り返しで転送します。  
 IXレジスタ、IYレジスタは、次のバイト/ワード転送のために1バイト/ワード転送ごとに自動的にインクリメント(+1/+2)、またはデクリメント(-1/-2)されます。ブロックの方向は、DIRフラグの状態によって決定されます。  
 バイトかワードかの指定は、MOVBK命令を用いる場合はオペランドの属性によって行われ、MOVBKB命令またはMOVBKW命令を用いる場合は、それぞれバイト、ワード・タイプに直接指定されます。  
 デスティネーション・ブロックは常にDS1レジスタで指定されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。

一方、ソース・ブロックは、デフォルト・セグメント・レジスタはDS0レジスタ  
 となっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジ  
 スタで指定されるセグメント内にロケートできます。

**【記 述 例】**    `MOVBK BYTE_VAR1, BYTE_VAR2`  
                   `MOVBK WORD_VAR1, MORD_VAR2`

**【バ イ ト 数】**    1

**【命 令 書 形 式】**

二モニック	オペランド	オペレーション・コード							
		7	6	5	4	3	2	1	0
<b>MOVBK</b>	<code>[DS1-spec:]dst-block, [Seg-spec:]src-block</code>	1	0	1	0	0	1	0	W
<b>MOVBE</b>	なし								
<b>MOVBEW</b>									

# MUL

符号付き乗算  
Multiply Signed

- [命令形式]    ① MUL src  
                   ② MUL dst, src  
                   ③ MUL dst, src1, src2

[オペランド, オペレーション]

命令形式①

ニモニック	オペランド	オペレーション
MUL	reg8	AW ← AL × src
	mem8	AH = ALのサイン拡張: CY ← 0, V ← 0 AH = ALのサイン拡張: CY ← 1, V ← 1
	reg16	DW, AW ← AW × src
	mem16	DW = AWのサイン拡張: CY ← 0, V ← 0 DW = AWのサイン拡張: CY ← 1, V ← 1

命令形式②

ニモニック	オペランド	オペレーション
MUL	reg16, imm8	dst ← dst × src
	reg16, imm16	積 ≤ 16ビット: CY ← 0, V ← 0 積 > 16ビット: CY ← 1, V ← 1

命令形式③

ニモニック	オペランド	オペレーション
MUL	reg16, reg16', imm8	dst ← src1 × src2
	reg16, mem16, imm8	積 ≤ 16ビット: CY ← 0, V ← 0
	reg16, reg16', imm16	積 > 16ビット: CY ← 1, V ← 1
	reg16, mem16, imm16	

[フ ラ グ]

AC	CY	V	P	S	Z
U	X	X	U	U	U



- 【説 明】** 命令形式①：○arc=reg8またはarc=mem8のとき
- ALレジスタの値とソース・オペランド (src) の符号付き乗算を行い、倍長結果をAWレジスタに格納します。このとき結果の上位半分 (AHレジスタ) が下位半分 (ALレジスタ) の符号拡張でなければCYフラグとVフラグがセット (1) されます。AHレジスタは拡張レジスタです。
- arc=reg16またはarc=mem16のとき
- AWレジスタの値とソース・オペランド (src) の符号付き乗算を行い、倍長結果をAWレジスタとDWレジスタに格納します。このとき結果の上位半分 (DWレジスタ) が下位半分 (AWレジスタ) の符号拡張でなければCYフラグとVフラグがセット (1) されます。DWレジスタは拡張レジスタです。
- 命令形式②：デスティネーション・オペランド (dat) とソース・オペランド (src) の符号付き乗算を行い、結果をデスティネーション・オペランド (dat) に格納します。
- 命令形式③：第1ソース・オペランド (src1) と第2ソース・オペランド (src2) の符号付き乗算を行い、結果をデスティネーション・オペランド (dat) に格納します。
- 【記 述 例】** AWレジスタの値とメモリ 0:50Hの内容 (ワード・データ) を乗算する。
- ```
MOV BW, 0
MOV DSI, BW
MOV IX, 50H
MUL WORD PTR [IX]
```

【バ イ ト 数】

| ニモニック | オペランド               | バイト数 |
|-------|---------------------|------|
| MUL   | reg8                | 2    |
|       | mem8                | 2-4  |
|       | reg16               | 2    |
|       | mem16               | 2-4  |
|       | reg16, imm8         | 3    |
|       | reg16, imm16        | 4    |
|       | reg16, reg16, imm8  | 3    |
|       | reg16, mem16, imm8  | 3-5  |
|       | reg16, reg16, imm16 | 4    |
|       | reg16, mem16, imm16 | 4-6  |

【命 令 書 形 式】

| ニモニック               | オペランド               | オペレーション・コード |   |   |   |   |   |            |            |     |     |     |   |             |             |             |     |
|---------------------|---------------------|-------------|---|---|---|---|---|------------|------------|-----|-----|-----|---|-------------|-------------|-------------|-----|
|                     |                     | 7           | 6 | 5 | 4 | 3 | 2 | 1          | 0          | 7   | 6   | 5   | 4 | 3           | 2           | 1           | 0   |
| MUL                 | reg8                | 1           | 1 | 1 | 1 | 0 | 1 | 1          | 0          | 1   | 1   | 1   | 0 | 1           |             |             | reg |
|                     | mem8                | 1           | 1 | 1 | 1 | 0 | 1 | 1          | 0          | mod | 1   | 0   | 1 |             |             | mem         |     |
|                     |                     |             |   |   |   |   |   |            | (disp-low) |     |     |     |   |             |             | (disp-high) |     |
|                     | reg16               | 1           | 1 | 1 | 1 | 0 | 1 | 1          | 1          | 1   | 1   | 1   | 0 | 1           |             | reg         |     |
|                     | mem16               | 1           | 1 | 1 | 1 | 0 | 1 | 1          | 1          | mod | 1   | 0   | 1 |             |             | mem         |     |
|                     |                     |             |   |   |   |   |   |            | (disp-low) |     |     |     |   |             |             | (disp-high) |     |
|                     | reg16, imm8         | 0           | 1 | 1 | 0 | 1 | 0 | 1          | 1          | 1   | 1   |     |   |             | reg         | reg'        |     |
|                     |                     |             |   |   |   |   |   |            | imm8       |     |     |     |   |             |             | —           |     |
|                     | reg16, imm16        | 0           | 1 | 1 | 0 | 1 | 0 | 0          | 1          | 1   | 1   |     |   |             | reg         | reg'        |     |
|                     |                     |             |   |   |   |   |   |            | imm16-low  |     |     |     |   |             | imm16-high  |             |     |
|                     | reg16, reg16, imm8  | 0           | 1 | 1 | 0 | 1 | 0 | 1          | 1          | 1   | 1   |     |   |             | reg         | reg'        |     |
|                     |                     |             |   |   |   |   |   |            | imm8       |     |     |     |   |             |             | —           |     |
|                     | reg16, mem16, imm8  | 0           | 1 | 1 | 0 | 1 | 0 | 1          | 1          | mod |     | reg |   |             | mem         |             |     |
|                     |                     |             |   |   |   |   |   |            | (disp-low) |     |     |     |   |             | (disp-high) |             |     |
|                     |                     |             |   |   |   |   |   |            | imm8       |     |     |     |   |             |             | —           |     |
|                     | reg16, reg16, imm16 | 0           | 1 | 1 | 0 | 1 | 0 | 0          | 1          | 1   | 1   |     |   |             | reg         | reg'        |     |
|                     |                     |             |   |   |   |   |   | imm16-low  |            |     |     |     |   | imm16-high  |             |             |     |
| reg16, mem16, imm16 | 0                   | 1           | 1 | 0 | 1 | 0 | 0 | 1          | mod        |     | reg |     |   | mem         |             |             |     |
|                     |                     |             |   |   |   |   |   | (disp-low) |            |     |     |     |   | (disp-high) |             |             |     |
|                     |                     |             |   |   |   |   |   | imm16-low  |            |     |     |     |   | imm16-high  |             |             |     |

|             |                             |
|-------------|-----------------------------|
| <b>MULU</b> | 符号なし乗算<br>Multiply Unsigned |
|-------------|-----------------------------|

【命令形式】 MULU src

【オペランド、オペレーション】

| ニモニック | オペランド (src) | オペレーション                                             |
|-------|-------------|-----------------------------------------------------|
| MULU  | reg8        | AW←AL×src                                           |
|       | mem8        |                                                     |
|       | reg16       | DW, AW←AW×src<br>DW=0: CY←0, V←0<br>DW≠0: CY←1, V←1 |
|       | mem16       |                                                     |

【フラグ】

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| U  | x  | x | U | U | U |

【説明】

○src=reg8またはsrc=mem8のとき  
ALレジスタの値とソース・オペランド (src) の符号なし乗算を行い、倍長結果をAWレジスタに格納します。このとき結果の上位半分 (AHレジスタ) がゼロでなければCYフラグとVフラグがセット (1) されます。AHレジスタは拡張レジスタです。

○src=reg16またはsrc=mem16のとき  
AWレジスタの値とソース・オペランド (src) の符号なし乗算を行い、倍長結果をAWレジスタとDWレジスタに格納します。このとき結果の上位半分 (DWレジスタ) がゼロでなければCYフラグとVフラグがセット (1) されます。DWレジスタは拡張レジスタです。

【記述例】 ALレジスタの内容とCLレジスタの内容を乗算する。

MULU CL

【バイト数】

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| MULU  | reg8  | 2    |
|       | mem8  | 2-4  |
|       | reg16 | 2    |
|       | mem16 | 2-4  |

【命令書形式】

| ニモニック | オペランド | オペレーション・コード |            |   |   |   |   |   |             |             |   |   |   |   |     |     |     |
|-------|-------|-------------|------------|---|---|---|---|---|-------------|-------------|---|---|---|---|-----|-----|-----|
|       |       | 7           | 6          | 5 | 4 | 3 | 2 | 1 | 0           | 7           | 6 | 5 | 4 | 3 | 2   | 1   | 0   |
| MULU  | reg8  | 1           | 1          | 1 | 1 | 0 | 1 | 1 | 0           | 1           | 1 | 1 | 0 | 0 |     |     | reg |
|       | mem8  | 1           | 1          | 1 | 1 | 0 | 1 | 1 | 0           | mod         | 1 | 0 | 0 |   |     | mem |     |
|       |       |             | (disp-low) |   |   |   |   |   |             | (disp-high) |   |   |   |   |     |     |     |
|       | reg16 | 1           | 1          | 1 | 1 | 0 | 1 | 1 | 1           | 1           | 1 | 1 | 0 | 0 |     | reg |     |
| mem16 | 1     | 1           | 1          | 1 | 0 | 1 | 1 | 1 | mod         | 1           | 0 | 0 |   |   | mem |     |     |
|       |       | (disp-low)  |            |   |   |   |   |   | (disp-high) |             |   |   |   |   |     |     |     |

|            |                  |
|------------|------------------|
| <b>NEG</b> | 2の補数演算<br>Negate |
|------------|------------------|

**【命令形式】** NEG *dst*

**【オペレーション】**  $dst \leftarrow \overline{dst} + 1$

**【オペランド】**

| ニモニック | オペランド ( <i>dst</i> ) |
|-------|----------------------|
| NEG   | reg                  |
|       | mem                  |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
| x  | 注  | x | x | x | x |

注 CY=1。ただし実行前の*dst*が0のときはCY=0。

**【説明】** デスティネーション・オペランド (*dst*) の内容の2の補数をとります。

- 【記述例】**
- NEG DL
  - NEG CW
  - NEG IX
  - NEG BP

**【バイト数】**

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| NEG   | reg   | 2    |
|       | mem   | 2-4  |

**【命令書形式】**

| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |             |     |   |   |   |   |     |     |
|-------|-------|-------------|---|---|---|---|---|---|-------------|-----|---|---|---|---|-----|-----|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0           | 7   | 6 | 5 | 4 | 3 | 2   | 1   |
| NEG   | reg   | 1           | 1 | 1 | 1 | 0 | 1 | 1 | W           | 1   | 1 | 0 | 1 | 1 |     | reg |
|       | mem   | 1           | 1 | 1 | 1 | 0 | 1 | 1 | W           | mod | 0 | 1 | 1 |   | mem |     |
|       |       | (disp-low)  |   |   |   |   |   |   | (disp-high) |     |   |   |   |   |     |     |

**NOP** ノー・オペレーション  
No Operation

【命令形式】 NOP

【オペレーション】 ノー・オペレーション

【オペランド】

| モニック | オペランド |
|------|-------|
| NOP  | なし    |

【フラグ】

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

【説明】 何も処理せず、3クロック費やします。

【記述例】 NOP

【バイト数】 1

【命令器形式】

| モニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|------|-------|-------------|---|---|---|---|---|---|---|
|      |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NOP  | なし    | 1           | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

|            |             |
|------------|-------------|
| <b>NOT</b> | 論理否定<br>Not |
|------------|-------------|

**【命令形式】** NOT dat

**【オペレーション】** dat ←  $\overline{\text{dat}}$

**【オペランド】**

| ニモニック | オペランド (dat) |
|-------|-------------|
| NOT   | reg         |
|       | mem         |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** デスティネーション・オペランド (dat) で指定されるビットを反転し (論理否定), 結果をデスティネーション・オペランド (dat) へ格納します。

- 【記述例】**
- NOT AL
  - NOT CW
  - NOT IX

**【バイト数】**

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| NOT   | reg   | 2    |
|       | mem   | 2-4  |

**【命令書形式】**

| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |             |     |   |   |   |   |   |   |     |
|-------|-------|-------------|---|---|---|---|---|---|-------------|-----|---|---|---|---|---|---|-----|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0           | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0   |
| NOT   | reg   | 1           | 1 | 1 | 1 | 0 | 1 | 1 | W           | 1   | 1 | 0 | 1 | 0 |   |   | reg |
|       | mem   | 1           | 1 | 1 | 1 | 0 | 1 | 1 | W           | mod | 0 | 1 | 0 |   |   |   | mem |
|       |       | (disp-low)  |   |   |   |   |   |   | (disp-high) |     |   |   |   |   |   |   |     |

|             |                   |
|-------------|-------------------|
| <b>NOT1</b> | ビットの反転<br>Not Bit |
|-------------|-------------------|

**【命令形式】** ① NOT1 dst, src  
 ② NOT1 dst

**【オペレーション】** 命令形式① : dstのビットn (nはsrcで指定) ←dstのビットn (nはsrcで指定)  
 命令形式② : dst←dst

**【オペランド】** 命令形式① 命令形式②

| 二モニック       | オペランド (dst, src) |
|-------------|------------------|
| <b>NOT1</b> | reg8, CL         |
|             | mem8, CL         |
|             | reg16, CL        |
|             | mem16, CL        |
|             | reg8, imm3       |
|             | mem8, imm3       |
|             | reg16, imm4      |
|             | mem16, imm4      |

| 二モニック       | オペランド (dst) |
|-------------|-------------|
| <b>NOT1</b> | CY          |

**【フ ラ グ】** 命令形式① 命令形式②

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    | x  |   |   |   |   |

**【説 明】** 命令形式① : 第 1 オペランドで指定されるデスティネーション・オペランド (dst) のビットn (nは第 2 オペランドで指定されるソース・オペランド (src) の内容) の論理否定をとり、結果をデスティネーション・オペランド (dst) に格納します。  
 オペランドがreg8, CLまたはmem8, CLのとき、CLの値は下位 3 ビット (0-7) のみ有効です。  
 オペランドがreg16, CLまたはmem16, CLのとき、CLの値は下位 4 ビット (0-15) のみ有効です。  
 オペランドがreg8, imm3のとき、命令の 4 バイト目のイミディエート・データは下位 3 ビットのみ有効です。



オペランドがmem8, imm3のとき、命令の最終バイトのイミディエ  
ト・データは下位3ビットのみ有効です。

オペランドがreg16, imm4のとき、命令の4バイト目のイミディエ  
ト・データは下位4ビットのみ有効です。

オペランドがmem16, imm4のとき、命令の最終バイトのイミディエ  
ト・データは下位4ビットのみ有効です。

命令形式②：CYフラグの内容の論理否定をとり、CYフラグに格納します。

【記述例】 IN AL, 0  
NOT1 AL, 7

【バイト数】

| モニック | オペランド       | バイト数 |
|------|-------------|------|
| NOT1 | reg8, CL    | 3    |
|      | mem8, CL    | 3-5  |
|      | reg16, CL   | 3    |
|      | mem16, CL   | 3-5  |
|      | reg8, imm3  | 4    |
|      | mem8, imm3  | 4-6  |
|      | reg16, imm4 | 4    |
|      | mem16, imm4 | 4-6  |
|      | CY          | 1    |

【命令書形式】

| ニモニック       | オペランド    | オペレーション・コード |   |      |   |   |   |   |   |     |   |   |   |   |   |   |   |
|-------------|----------|-------------|---|------|---|---|---|---|---|-----|---|---|---|---|---|---|---|
|             |          | 7           | 6 | 5    | 4 | 3 | 2 | 1 | 0 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NOT1        | reg8, CL | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
|             |          | 1           | 1 | 0    | 0 | 0 |   |   |   | reg |   |   |   |   |   |   |   |
| mem8, CL    |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
|             |          | mod         | 0 | 0    | 0 |   |   |   |   | mem |   |   |   |   |   |   |   |
|             |          | (disp-high) |   | —    |   |   |   |   |   |     |   |   |   |   |   |   |   |
| reg16, CL   |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
|             |          | 1           | 1 | 0    | 0 | 0 |   |   |   | reg |   |   |   |   |   |   |   |
| mem16, CL   |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
|             |          | mod         | 0 | 0    | 0 |   |   |   |   | mem |   |   |   |   |   |   |   |
|             |          | (disp-high) |   | —    |   |   |   |   |   |     |   |   |   |   |   |   |   |
| reg8, imm3  |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
|             |          | 1           | 1 | 0    | 0 | 0 |   |   |   | reg |   |   |   |   |   |   |   |
| mem8, imm3  |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
|             |          | mod         | 0 | 0    | 0 |   |   |   |   | mem |   |   |   |   |   |   |   |
|             |          | (disp-high) |   | imm3 |   |   |   |   |   |     |   |   |   |   |   |   |   |
| reg16, imm4 |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|             |          | 1           | 1 | 0    | 0 | 0 |   |   |   | reg |   |   |   |   |   |   |   |
| mem16, imm4 |          | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0   | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|             |          | mod         | 0 | 0    | 0 |   |   |   |   | mem |   |   |   |   |   |   |   |
|             |          | (disp-high) |   | imm4 |   |   |   |   |   |     |   |   |   |   |   |   |   |
| CY          |          | 1           | 1 | 1    | 1 | 0 | 1 | 0 | 1 | —   |   |   |   |   |   |   |   |

|           |           |
|-----------|-----------|
| <b>OR</b> | 論理和<br>Or |
|-----------|-----------|

**【命令形式】** OR dst, src

**【オペランド, オペレーション】**

| ニモニック     | オペランド (dst, src) | オペレーション                                     |
|-----------|------------------|---------------------------------------------|
| <b>OR</b> | reg, reg'        | dst←dst∨src                                 |
|           | mem, reg         |                                             |
|           | reg, mem         |                                             |
|           | reg, imm         |                                             |
|           | mem, imm         |                                             |
|           | acc, imm         | [W=0のとき] AL←AL∨imm8<br>[W=1のとき] AW←AW∨imm16 |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| U  | 0  | 0 | x | x | x |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) と第2オペランドで指定されるソース・オペランド (src) の論理和 (OR) をとり、結果をデスティネーション・オペランド (dst) に格納します。

**【記述例】** OR AW, WORD PTR [IX]

**【バイト数】**

| ニモニック     | オペランド     | バイト数 |
|-----------|-----------|------|
| <b>OR</b> | reg, reg' | 2    |
|           | mem, reg  | 2-4  |
|           | reg, mem  | 2-4  |
|           | reg, imm  | 3, 4 |
|           | mem, imm  | 3-6  |
|           | acc, imm  | 2, 3 |

【命令形式】

| ニモニック    | オペランド                 | オペレーション・コード       |   |   |   |   |   |   |                   |             |   |   |   |     |   |      |   |
|----------|-----------------------|-------------------|---|---|---|---|---|---|-------------------|-------------|---|---|---|-----|---|------|---|
|          |                       | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0                 | 7           | 6 | 5 | 4 | 3   | 2 | 1    | 0 |
| OR       | reg, reg'             | 0                 | 0 | 0 | 0 | 1 | 0 | 1 | W                 | 1           | 1 |   |   | reg |   | reg' |   |
|          | mem, reg              | 0                 | 0 | 0 | 0 | 1 | 0 | 0 | W                 | mod         |   |   |   | reg |   | mem  |   |
|          |                       | (disp-low)        |   |   |   |   |   |   |                   | (disp-high) |   |   |   |     |   |      |   |
|          | reg, mem              | 0                 | 0 | 0 | 0 | 1 | 0 | 1 | W                 | mod         |   |   |   | reg |   | mem  |   |
|          |                       | (disp-low)        |   |   |   |   |   |   |                   | (disp-high) |   |   |   |     |   |      |   |
|          | reg, imm <sup>注</sup> | 1                 | 0 | 0 | 0 | 0 | 0 | 0 | W                 | 1           | 1 | 0 | 0 | 1   |   | reg  |   |
|          |                       | imm8 or imm16-low |   |   |   |   |   |   |                   | imm16-high  |   |   |   |     |   |      |   |
| mem, imm | 1                     | 0                 | 0 | 0 | 0 | 0 | 0 | W | mod               | 0           | 0 | 1 |   | mem |   |      |   |
|          | (disp-low)            |                   |   |   |   |   |   |   | (disp-high)       |             |   |   |   |     |   |      |   |
|          | imm8 or imm16-low     |                   |   |   |   |   |   |   | imm16-high        |             |   |   |   |     |   |      |   |
| acc, imm | 0                     | 0                 | 0 | 0 | 1 | 1 | 0 | W | imm8 or imm16-low |             |   |   |   |     |   |      |   |
|          | imm16-high            |                   |   |   |   |   |   |   | —                 |             |   |   |   |     |   |      |   |

注 アセンブラ、コンパイラによっては、次に示すようなコードが生成されることがあります。

| 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1   | 0 |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|
| 1    | 0 | 0 | 0 | 0 | 0 | 1 | W | 1 | 1 | 0 | 0 | 1 |   | reg |   |
| imm8 |   |   |   |   |   |   |   | — |   |   |   |   |   |     |   |

このような場合でも正常に命令を実行します。ただし、エミュレータによっては、この命令に対する逆アセンブラ機能、アセンブラ機能がサポートされていない場合がありますのでご注意ください。

|            |                          |
|------------|--------------------------|
| <b>OUT</b> | I/Oデバイスへのデータ出力<br>Output |
|------------|--------------------------|

【命令形式】 `OUT dst, src`

【オペランド、オペレーション】

| ニモニック      | オペランド (dst, src)       | オペレーション                                                  |
|------------|------------------------|----------------------------------------------------------|
| <b>OUT</b> | <code>immB, src</code> | [W=0のとき] (immB) ←AL<br>[W=1のとき] (immB+1) ←AH, (immB) ←AL |
|            | <code>DW, src</code>   | [W=0のとき] (DW) ←AL<br>[W=1のとき] (DW+1) ←AH, (DW) ←AL       |

【フ ラ グ】

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

【説 明】 デスティネーション・オペランド (dst) で指定されるI/Oデバイスのレジスタにアキュムレータ (ALレジスタまたはAWレジスタ) の内容を転送します。

【記 述 例】 ポート・アドレス0D8HにALレジスタの内容を転送する。  
`MOV DW, 0D8H`  
`OUT DW, AL`

【バ イ ト 数】

| ニモニック      | オペランド                  | バイト数 |
|------------|------------------------|------|
| <b>OUT</b> | <code>immB, src</code> | 2    |
|            | <code>DW, src</code>   | 1    |

【命令語形式】

| ニモニック      | オペランド                  | オペレーション・コード |   |   |   |   |   |   |   |                   |   |   |   |   |   |   |   |
|------------|------------------------|-------------|---|---|---|---|---|---|---|-------------------|---|---|---|---|---|---|---|
|            |                        | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| <b>OUT</b> | <code>immB, src</code> | 1           | 1 | 1 | 0 | 0 | 1 | 1 | W | <code>immB</code> |   |   |   |   |   |   |   |
|            | <code>DW, src</code>   | 1           | 1 | 1 | 0 | 1 | 1 | 1 | W | —                 |   |   |   |   |   |   |   |

|             |                                   |
|-------------|-----------------------------------|
| <b>OUTM</b> | メモリ-I/Oのブロック転送<br>Output Multiple |
|-------------|-----------------------------------|

**【命令形式】** (repeat) OUTM DW, [Seg-spec:] src-block

**【オペレーション】**

[W=0のとき] (DW) ← (IX)  
 DIR=0: IX←IX+1  
 DIR=1: IX←IX-1

[W=1のとき] (DW+1, DW) ← (IX+1, IX)  
 DIR=0: IX←IX+2  
 DIR=1: IX←IX-2

**【オペランド】**

| 二モニック       | オペランド                     |
|-------------|---------------------------|
| <b>OUTM</b> | DW, [Seg-spec:] src-block |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

**【説明】** IXレジスタでアドレスされるメモリの内容を、DWレジスタでアドレスされるI/Oデバイスのレジスタに転送します。繰り返し転送回数は、ペアで使用されるリピート・プリフィクスのREP命令によって制御されます。繰り返し転送の際、DWレジスタの内容(I/Oデバイスのアドレス)は固定ですが、IXレジスタは次のバイト/ワード転送のために1バイト/ワード・データが転送されることに自動的にインクリメント(+1/+2)またはデクリメント(-1/-2)されます。ブロックの方向は、DIRフラグの内容によって決定されます。

バイトかワードかの指定は、オペランド属性によって行われます。

OUTM命令は、リピート・プリフィクスのREP命令とともに使用されます。

ソース・ブロックは、デフォルト・セグメント・レジスタはDSレジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にローケートできます。

**【記 述 例】** ●メモリ0:50Hの内容をポート・アドレス0D8H (バイト・データ) に転送する。

```
MOV AW, 0
MOV DSO, AW
MOV IX, 50H
MOV DW, 0D8H
OUTM DW, DSO: WORD PTR [IX]
```

●メモリ0:0H~0FFHの内容をポート・アドレス0D8H (バイト・データ) に転送する。

```
MOV AW, 0
MOV DSO, AW
MOV IX, 0H
MOV DW, 0D8H
MOV CW, 0FFH
REP OUTM DW, DSO: PTR [IX]
```

**【バ イ ト 数】** 1

**【命 令 語 形 式】**

| ニモニック | オペランド                     | オペレーション・コード |   |   |   |   |   |   |   |
|-------|---------------------------|-------------|---|---|---|---|---|---|---|
|       |                           | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTM  | DW, [Seg-spec:] src-block | 0           | 1 | 1 | 0 | 1 | 1 | 1 | W |

|             |                                   |
|-------------|-----------------------------------|
| <b>POLL</b> | 浮動小数点演算用コプロセッサ待ち<br>Poll and wait |
|-------------|-----------------------------------|

【命令形式】 POLL

【オペレーション】 POLL and wait

【オペランド】

|       |       |
|-------|-------|
| ニモニック | オペランド |
| POLL  | なし    |

【フラグ】

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

【説明】

- V33A, V53A以外  
 POLL端子がアクティブ(ロウ・レベル)になるまでCPUをウエイト状態にします。  
 注意 この命令の直前にBUSLOCK命令を置くことはできません。
- V33A, V53A  
 コプロセッサ接続時: CPBUSY端子がインアクティブ (ハイ・レベル) になるまでCPUをウエイト状態にします。  
 コプロセッサ非接続時: コプロセッサ不在割り込み (ベクタ7) を発生します。  
 このときスタックに退避するアドレスは、この命令の先頭バイトです。  
 注意 この命令の直前にBUSLOCK命令を置くことはできません。

【記述例】 POLL

【バイト数】 1

【命令形式】

|       |       |             |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| POLL  | なし    | 1           | 0 | 0 | 1 | 1 | 0 | 1 | 1 |



|            |                         |
|------------|-------------------------|
| <b>POP</b> | スタックからの復帰<br><b>Pop</b> |
|------------|-------------------------|

**【命令形式】** POP dst

**【オペランド、オペレーション】**

| モニタック      | オペランド (dst)                                                                                                                                              | オペレーション                           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| <b>POP</b> | mem16                                                                                                                                                    | SP←SP+2<br>(mem16) ← (SP-1, SP-2) |
|            | reg16                                                                                                                                                    | SP←SP+2<br>dst← (SP-1, SP-2)      |
|            | sreg                                                                                                                                                     |                                   |
|            | PSW                                                                                                                                                      |                                   |
| R          | TY← (SP+1, SP)<br>IX← (SP+3, SP+2)<br>BP← (SP+5, SP+4)<br>BW← (SP+9, SP+8)<br>DW← (SP+11, SP+10)<br>CW← (SP+13, SP+12)<br>AW← (SP+15, SP+14)<br>SP←SP+16 |                                   |

**【フ ラ グ】** ○dst=PSWの場合

|    |    |   |   |   |   |    |     |    |     |
|----|----|---|---|---|---|----|-----|----|-----|
| AC | CY | V | P | S | Z | MD | DIR | IE | BRK |
| R  | R  | R | R | R | R | R  | R   | R  | R   |

備考 V33A, V53AにはMDフラグがありません。

○dst=PSW以外の場合

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

**【説 明】** デスティネーション・オペランド (dst) の内容にスタックの内容を転送します (ただし、dst=sregの場合、PSへは転送されません)。

- 注意1. `dat=ereg`の場合、次の命令との間にハードウェア割り込み(マスカブル割り込み、ノンマスカブル割り込み)要求およびシングルステップ・ブレークは受け付けられません。
2. `dat=PSW`の場合、MDフラグは書き込み許可状態でのみリストアされ、書き込み禁止状態では変化しません(V33A, V33Aを除く)。
  3. SPレジスタに対してPUSH命令とPOP命令の組み合わせを実行すると、SPレジスタには命令実行前の値から(-2)された値が格納されます。

【記述例】

- POP AW
- POP BW
- POP FY
- POP SP

MOV BP, SP

【バイト数】

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| POP   | mem16 | 2-4  |
|       | reg16 | 1    |
|       | ereg  |      |
|       | PSW   |      |
|       | R     | 1    |

【命令形式】

| ニモニック | オペランド | オペレーション・コード |   |   |   |      |   |   |             |     |   |   |   |     |   |   |   |
|-------|-------|-------------|---|---|---|------|---|---|-------------|-----|---|---|---|-----|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3    | 2 | 1 | 0           | 7   | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| POP   | mem16 | 1           | 0 | 0 | 0 | 1    | 1 | 1 | 1           | mod | 0 | 0 | 0 | mem |   |   |   |
|       |       | (disp-low)  |   |   |   |      |   |   | (disp-high) |     |   |   |   |     |   |   |   |
|       | reg16 | 0           | 1 | 0 | 1 | 1    |   |   | reg         |     |   |   |   |     |   |   |   |
|       | ereg  | 0           | 0 | 0 |   | ereg | 1 | 1 | 1           |     |   |   |   |     |   |   |   |
|       | PSW   | 1           | 0 | 0 | 1 | 1    | 1 | 0 | 1           |     |   |   |   |     |   |   |   |
| R     | 0     | 1           | 1 | 0 | 0 | 0    | 0 | 1 |             |     |   |   |   |     |   |   |   |

**PREPARE**

スタック・フレームの生成  
Prepare New Stack Frame

**【命令形式】** PREPARE imm16, imm8

**【オペレーション】** (SP-1, SP-2) ← BP  
 SP ← SP-2  
 temp ← SPを行ったあと  
 imm8 > 0のとき次の動作を、

|                             |   |    |
|-----------------------------|---|----|
| (SP-1, SP-2) ← (BP-1, BP-2) | } | *1 |
| SP ← SP-2                   |   |    |
| BP ← BP-2                   |   |    |

“imm8-1” 回繰り返したあとに、

|                     |   |    |
|---------------------|---|----|
| (SP-1, SP-2) ← temp | } | *2 |
| SP ← SP-2           |   |    |

を実行します。  
 このあと次の処理を行います。  
 EP ← temp  
 SP ← SP - imm16  
 imm8 = 1のとき、\*1の繰り返し動作を行いません。  
 imm8 = 0のとき、\*1および\*2の動作を行いません。

**【オペランド】**

| ニモニック   | オペランド       |
|---------|-------------|
| PREPARE | imm16, imm8 |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** この命令は、ブロック構造の高級言語(たとえばPascal, Adaなど)において必要な“スタック・フレーム”を生成するために用いられます。スタック・フレームには、そのプロシージャから参照してもよい変数を含むフレームを指すポインタとローカル変数の領域が含まれます。  
 この命令は、ローカル変数の領域確保と、グローバル変数への参照を可能とするため、フレーム・ポインタのコピーを行います。第1オペランドに記載された16ビット

ト・イミューディエト・データでローカル変数用として確保する領域の大きさ（バイト単位）が指定され、第2オペランドに記述された8ビット・イミューディエト・データでプロシージャ・ブロックの深さ（この深さをレキシカル・レベルといいます）が示されます。

この命令によって生成されるフレームのベース・アドレスは、BPにセットされます。

まずBPをスタックへ退避します。これは、プロシージャの終了時点で、コールした側のプロシージャのBPを復帰するためです。次に、コールされたプロシージャから参照可能な範囲のフレーム・ポインタ（退避されたBP）をスタックに預めます。参照可能な範囲は、そのプロシージャのレキシカル・レベルより1減じた値となります。

レキシカル・レベルが1以上の場合には、自分自身のフレーム・ポインタもスタックへ預めます。これは、このプロシージャからコールされたプロシージャにおいてフレーム・ポインタのコピーを行う場合に、コールしたプロシージャのフレーム・ポインタもコピーするためです。

次にBPに新しいフレーム・ポインタの値をセットし、そのプロシージャで使用されるローカル変数の領域をスタックに確保します。つまり、SPをローカル変数分だけ減らします。

```

【記述例】 MOV  SP, 60H
            MOV  BP, SP
            CALL  CHK
            PREPARE 0006, 04
            MOV  AW, [BP+0FAH]
            ADD  AW, [BP+0F8A]
            MOV  [BP+0FCH], AW
    
```

【バイト数】 4

【命令書形式】

| 二モニック   | オペランド       | オペレーション・コード |   |   |   |   |   |   |   |           |   |   |   |   |   |   |   |
|---------|-------------|-------------|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|
|         |             | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PREPARE | imm16, imm8 | 1           | 1 | 0 | 0 | 1 | 0 | 0 | 0 | imm16-low |   |   |   |   |   |   |   |
|         |             | imm16-high  |   |   |   |   |   |   |   | imm8      |   |   |   |   |   |   |   |

|             |                  |
|-------------|------------------|
| <b>PUSH</b> | スタックへの透過<br>Push |
|-------------|------------------|

【命令形式】 **PUSH src**

【オペランド、オペレーション】

| ニモニック       | オペランド (src)                    | オペレーション                                                                                                                                                                                      |
|-------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PUSH</b> | mem16                          | SP←SP-2<br>(SP+1, SP) ← (mem16+1, mem16)                                                                                                                                                     |
|             | reg16                          | SP←SP-2                                                                                                                                                                                      |
|             | areg                           | (SP+1, SP) ← src                                                                                                                                                                             |
|             | PSW                            |                                                                                                                                                                                              |
|             | R                              | temp←SP<br>(SP-1, SP-2) ←AW<br>(SP-3, SP-4) ←CW<br>(SP-5, SP-6) ←DW<br>(SP-7, SP-8) ←BW<br>(SP-9, SP-10) ←temp<br>(SP-11, SP-12) ←BP<br>(SP-13, SP-14) ←IX<br>(SP-15, SP-16) ←IY<br>SP←SP-16 |
|             | imm8                           | (SP-1, SP-2) ←imm8のサイン拡張<br>SP←SP-2                                                                                                                                                          |
| imm16       | (SP-1, SP-2) ←imm16<br>SP←SP-2 |                                                                                                                                                                                              |

【フ ラ グ】

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

【説 明】 ソース・オペランド (src) の内容をスタックに透過します。  
 オペランドに 8 ビット・イミディエイト・データ (imm8) を記述した場合は、imm8 はサイン拡張され、16 ビット・データとして SP でアドレスされるスタックに透過されます。

- 【記 述 例】**
- PUSH DS0
  - PUSH SS
  - PUSH DS1

**【バ イ ト 数】**

| ニモニック       | オペランド | バイト数 |
|-------------|-------|------|
| <b>PUSH</b> | mem16 | 2-4  |
|             | reg16 | 1    |
|             | areg  |      |
|             | PSW   |      |
|             | R     | 1    |
|             | imm8  | 2    |
|             | imm16 | 3    |

**【命 令 書 形 式】**

| ニモニック       | オペランド | オペレーション・コード |   |   |      |   |     |   |             |           |   |   |   |     |   |   |   |
|-------------|-------|-------------|---|---|------|---|-----|---|-------------|-----------|---|---|---|-----|---|---|---|
|             |       | 7           | 6 | 5 | 4    | 3 | 2   | 1 | 0           | 7         | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| <b>PUSH</b> | mem16 | 1           | 1 | 1 | 1    | 1 | 1   | 1 | 1           | mod       | 1 | 1 | 0 | mem |   |   |   |
|             |       | (disp-low)  |   |   |      |   |     |   | (disp-high) |           |   |   |   |     |   |   |   |
|             | reg16 | 0           | 1 | 0 | 1    | 0 | reg |   |             | —         |   |   |   |     |   |   |   |
|             | areg  | 0           | 0 | 0 | areg |   |     | 1 | 1           | 0         | — |   |   |     |   |   |   |
|             | PSW   | 1           | 0 | 0 | 1    | 1 | 1   | 0 | 0           | —         |   |   |   |     |   |   |   |
|             | R     | 0           | 1 | 1 | 0    | 0 | 0   | 0 | 0           | —         |   |   |   |     |   |   |   |
|             | imm8  | 0           | 1 | 1 | 0    | 1 | 0   | 1 | 0           | imm8      |   |   |   |     |   |   |   |
| imm16       |       | 0           | 1 | 1 | 0    | 1 | 0   | 0 | 0           | imm16-low |   |   |   |     |   |   |   |
|             |       | imm16-high  |   |   |      |   |     |   | —           |           |   |   |   |     |   |   |   |

|                                          |                    |
|------------------------------------------|--------------------|
| <b>REP</b><br><b>REPE</b><br><b>REPZ</b> | Z=1によるリピート・プリフィクス  |
|                                          | Repeat             |
|                                          | Repeat while Equal |
|                                          | Repeat while Zero  |

**【命令形式】** REP  
REPE  
REPZ

**【オペレーション】** [CW≠0のとき] PS: PC+1のバイト命令実行  
CW←CW-1  
Z≠1のとき: PC←PC+2  
Z=1のとき: 再実行  
[CW=0のとき] PC←PC+2

**【オペランド】**

| ニモニック | オペランド |
|-------|-------|
| REP   | なし    |
| REPE  |       |
| REPZ  |       |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** CW≠0の間、続くバイトのブロック転送/比較/入出力命令を実行し、CWレジスタの値をデクリメント (-1) します。  
REPはMOVBL, LDM, STM, OUTM, INMの各命令とともに用いられ、Zフラグの値に関係なくCW≠0の間繰り返し処理を行います。  
REPZ命令およびREPE命令はCMPBL, CMPMの各命令とともに用いられ、各ブロック命令による比較の結果Z≠1になるか、またはCW=0になるとループを抜けます。CWレジスタのチェックは、ブロック命令の実行前、すなわちREP/REPE/REPZ命令実行直前の状態に対して行われます。したがって、最初CW=0でREP/REPE/REPZ命令実行に入ると、続くブロック命令は1回も実行されずにその次の命令に移ります。Zフラグのチェックは、あくまで続くブロック命令の結果に対して行われ、最初REPE/REPZ命令に入る直前の内容は関係ありません。

注意 この命令と次の命令との間では、ハードウェア割り込み（マスクابل割り込み、ノンマスクابل割り込み）要求およびシングルステップ・ブレイクは受け付けられません。

- 【記 述 例】 ●REP MOVKBW  
●REPZ CMPKBW

【バ イ ト 数】 1

【命 令 形 式】

| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REP   | なし    | 1           | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| REPZ  |       |             |   |   |   |   |   |   |   |
|       |       |             |   |   |   |   |   |   |   |



**REPC**CY=1によるリピート・プリフィクス  
Repeat while Carry

**【命令形式】** REPC

**【オペレーション】** [CW≠0のとき] FS:PC+1のバイト命令実行  
                           CW←CW-1  
                           CY≠1のとき:PC←PC+2  
                           CY=1のとき:再実行  
                           [CW=0のとき] PC←PC+2

**【オペランド】**

| ニモニック | オペランド |
|-------|-------|
| REPC  | なし    |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** CW≠0の間、続くバイトのブロック比較命令 (CMPBK, CMPM) を実行し、CWレジスタの値をデクリメント (-1) します。  
 ブロック比較命令の結果、CY≠1になるとループを抜けます。  
 CWレジスタのチェックは、ブロック比較命令の実行前、すなわちREPC命令実行の直前の状態に対して行われます。したがって、最初CW=0でREPC命令実行に入ると、続くブロック比較命令は1回も実行されずにその次の命令に移ります。  
 CYフラグのチェックは、あくまで続くブロック比較命令の結果に対して行われ、最初REPC命令に入る直前の内容は関係ありません。

**注意** この命令と次の命令との間では、ハードウェア割り込み (マスカブル割り込み、ノンマスカブル割り込み) 要求およびシングルステップ・ブレークは受け付けられません。

**【記述例】** REPC CMPBKW

**【バイト数】** 1

【命令形式】

| 二モニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEPC  | なし    | 0           | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

## REPNC

CY=0によるリピート・プリフィクス

Repeat while Not Carry

**【命令形式】** REPNC

**【オペレーション】** [CW≠0のとき] PS: PC+1のバイト命令実行

CW←CW-1

CY≠1のとき: 再実行

CY=1のとき: PC←PC+2

[CW=0のとき] PC←PC+2

**【オペランド】**

| ニモニック | オペランド |
|-------|-------|
| REPNC | なし    |

**【フ ラ グ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説 明】** CW≠0の間、続くバイトのブロック比較命令 (CMPBK, CMPM) を実行し、CWレジスタの値をデクリメント (-1) します。

ブロック比較命令の結果、CY=1になるとループを抜けます。

CWレジスタのチェックは、ブロック比較命令の実行前、すなわちREPNC命令実行の直前の状態に対して行われます。したがって、最初CW=0でREPNC命令実行に入ると、続くブロック比較命令は1回も実行されずにその次の命令に移ります。

CYフラグのチェックは、あくまで続くブロック比較命令の結果に対して行われ、最初REPNC命令に入る直前の内容は関係ありません。

**注意** この命令と次の命令との間では、ハードウェア割り込み (マスカブル割り込み、ノンマスカブル割り込み) 要求およびシングルステップ・ブレークは受け付けられません。

**【配 送 例】** REPNC CMPMB

**【バ イ ト 数】** 1

【命令書形式】

| ニモニク  | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REPNC | なし    | 0           | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**REPNE**  
**REPNZ**

Z=0によるリピート・プリフィクス  
Repeat while Not Equal  
Repeat while Not Zero

**【命令形式】** REPNE  
REPZ

**【オペレーション】** [CW≠0のとき] PS:PC+1のバイト命令実行  
CW←CW-1  
Z≠1のとき:再実行  
Z=1のとき:PC←PC+2  
[CW=0のとき] PC←PC+2

**【オペランド】**

| ニモニック | オペランド |
|-------|-------|
| REPNE | なし    |
| REPZ  |       |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** CW≠0の間、続くバイトのブロック比較命令 (CMPBK, CMPBM) を実行し、CWレジスタの値をデクリメント (-1) します。  
ブロック比較命令の結果、Z≠0になるか、またはCW=0になるとループを抜けます。  
CWレジスタのチェックは、ブロック比較命令の実行前、すなわちREPNE/REPZ命令実行直前に対して行われます。したがって、最初CW=0でREPNE/REPZ命令実行に入ると、続くブロック比較命令は1回も実行されずにその次の命令に移ります。  
Zフラグのチェックは、あくまで続くブロック比較命令の結果に対して行われ、最初REPNE/REPZ命令に入る直前の内容は関係ありません。

**注意** この命令と次の命令との間では、ハードウェア割り込み (マスカブル割り込み、ノンマスカブル割り込み) 要求およびシングルステップ・ブレイクは受け付けられません。

【記 述 例】 ●REPNE CMPMB  
●REPZ CMPBKW

【バ イ ト 数】 1

【命 令 書 形 式】

| 二モニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REPNE | なし    | 1           | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| REPZ  |       |             |   |   |   |   |   |   |   |

|            |                                      |
|------------|--------------------------------------|
| <b>RET</b> | サブルーチンからの復帰<br>Return from Procedure |
|------------|--------------------------------------|

- 【命令形式】**
- ① RET
  - ② RET pop-value

**【オペランド、オペレーション】**

○セグメント内コールからのリターンするとき

| ニモニック | オペランド     | オペレーション                                      |
|-------|-----------|----------------------------------------------|
| RET   | なし        | PC← (SP+1, SP)<br>SP←SP+2                    |
|       | pop-value | PC← (SP+1, SP)<br>SP←SP+2<br>SP←SP+pop-value |

○セグメント外コールからのリターンするとき

| ニモニック | オペランド     | オペレーション                                                          |
|-------|-----------|------------------------------------------------------------------|
| RET   | なし        | PC← (SP+1, SP)<br>PS← (SP+3, SP+2)<br>SP←SP+4                    |
|       | pop-value | PC← (SP+1, SP)<br>PS← (SP+3, SP+2)<br>SP←SP+4<br>SP←SP+pop-value |

**【フ ラ グ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

- 【説 明】**
- セグメント内コールからのリターンするとき  
PCをスタックからリストアします。オペランドにpop-valueを記述したときは16ビットのpop-valueからSPに加算されます(PCに続いて過避してあったパラメータが不要になったとき、不要なパラメータの分だけSPの値を飛ばすのに有効です)。  
セグメント外コールからのRET命令との区別はアセンブラが自動的にを行います。
  
  - セグメント外コールからのリターンするとき  
PCとPSをスタックからリストアします。オペランドにpop-valueを記述したときは16ビットのpop-valueがSPに加算されます(PCとPSに続いて過避してあったパラメータが不要になったとき、不要なパラメータの分だけSPの値を飛ばすのに有効です)。  
セグメント内コールからのRET命令との区別はアセンブラが自動的にを行います。

**【記 述 例】** POP R  
RET

**【バ イ ト 数】**

| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| RET   | なし        | 1    |
|       | pop-value | 3    |

**【命 令 形 式】** ○セグメント内コールからのリターンするとき

| ニモニック | オペランド     | オペレーション・コード    |   |   |   |   |   |   |   |               |   |   |   |   |   |   |   |
|-------|-----------|----------------|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|---|
|       |           | 7              | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RET   | なし        | 1              | 1 | 0 | 0 | 0 | 0 | 1 | 1 | —             |   |   |   |   |   |   |   |
|       | pop-value | 1              | 1 | 0 | 0 | 0 | 0 | 1 | 0 | pop-value-low |   |   |   |   |   |   |   |
|       |           | pop-value-high |   |   |   |   |   |   |   | —             |   |   |   |   |   |   |   |

○セグメント外コールからのリターンするとき

| ニモニック | オペランド     | オペレーション・コード    |   |   |   |   |   |   |   |               |   |   |   |   |   |   |   |
|-------|-----------|----------------|---|---|---|---|---|---|---|---------------|---|---|---|---|---|---|---|
|       |           | 7              | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RET   | なし        | 1              | 1 | 0 | 0 | 1 | 0 | 1 | 1 | —             |   |   |   |   |   |   |   |
|       | pop-value | 1              | 1 | 0 | 0 | 1 | 0 | 1 | 0 | pop-value-low |   |   |   |   |   |   |   |
|       |           | pop-value-high |   |   |   |   |   |   |   | —             |   |   |   |   |   |   |   |



**RETEM** [V33A, V83Aを除く] エミュレーション・モードからの復帰  
Return from Emulation

【命令形式】 RETEM

【オペレーション】 PC←(SP+1, SP)  
PS←(SP+3, SP+2)  
PSW←(SP+5, SP+4)  
SP←SP+6  
MDを書き込み禁止にします。

【オペランド】

| ニモニック | オペランド |
|-------|-------|
| RETEM | なし    |

【フラグ】

| AC | CY | V | P | S | Z | MD | DIR | IE | BRK |
|----|----|---|---|---|---|----|-----|----|-----|
| R  | R  | R | R | R | R | R  | R   | R  | R   |

【説明】 エミュレーション・モードでRETEM命令が実行されると(この命令はμPD8080AFの命令として解釈されます), CPUは割り込み処理から戻るように, すなわちBRKEM命令で回避されていたPS, PC, PSWをリストアし, ネイティブ・モードに戻ります。このMDフラグには, BRKEM命令によって回避されていたネイティブ・モード時の内容(つまり"1")がリストアされ, これによってCPUはネイティブ・モードになります。また, このRETEM命令実行後MDフラグは書き込み禁止状態となり, RETIまたはPOP PSW命令を実行してもMDフラグはリストアされなくなります。

【記述例】 RETEM

【バイト数】 2

【命令器形式】

| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RETEM | なし    | 1           | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

|             |                                    |
|-------------|------------------------------------|
| <b>RETI</b> | 割り込みからの復帰<br>Return from Interrupt |
|-------------|------------------------------------|

**【命令形式】** RETI

**【オペレーション】** PC← (SP+1, SP)  
 PS← (SP+3, SP+2)  
 PSW← (SP+5, SP+4)  
 SP←SP+6

**【オペランド】**

|       |       |
|-------|-------|
| ニモニック | オペランド |
| RETI  | なし    |

**【フラグ】**

|    |    |   |   |   |   |    |     |    |     |
|----|----|---|---|---|---|----|-----|----|-----|
| AC | CY | V | P | S | Z | MD | DIR | IE | BRK |
| R  | R  | R | R | R | R | R  | R   | R  | R   |

備考 V33A, V53AにはMDフラグがありません。

**【説明】** PC, PS, PSWにスタックの内容をリストアします。割り込み処理から戻るときに用いられます。

注意 MDフラグは書き込み許可状態でのみリストアされ、書き込み禁止状態では変化しません (V33A, V53Aを除く)。

**【記述例】** POP R  
 RETI

**【バイト数】** 1

**【命令書形式】**

|       |       |             |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|
| ニモニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RETI  | なし    | 1           | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

|                              |                                                      |
|------------------------------|------------------------------------------------------|
| <b>RETXA</b> [V33A, V83A だけ] | 拡張アドレス・モードからの復帰<br>Return from Extended Address Mode |
|------------------------------|------------------------------------------------------|

【命令形式】 RETXA imm8

【オペレーション】

```
temp1 ← (imm8 × 4 + 1, imm8 × 4)
temp2 ← (imm8 × 4 + 3, imm8 × 4 + 2)
XA ← 0
PC ← temp1
PS ← temp2
```

【オペランド】

| 二モニック | オペランド |
|-------|-------|
| RETXA | imm8  |

【フラグ】

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

【説明】 拡張アドレス・モードを解除します。  
命令により指定された割り込みベクタ・テーブルのエントリに格納されているアドレスに制御を移し、XAMレジスタ (内部I/Oアドレス: FF80H) のビット0 (XAフラグ) をリセット (0) します。

通常アドレス・モードで実行したときは、通常アドレス・モードのアドレス上のベクタ・テーブルを読み込んだあと、このベクタ・テーブルのアドレスにジャンプします。

拡張アドレス・モードで実行したときは、拡張アドレス・モードのアドレス上のベクタ・テーブルを読み込んだあと、通常アドレス・モードに移り、先に読み込んだアドレスにジャンプします。

スタックからのPC、PS、PSWの値の復帰は行いません。

【記述例】 RETXA 0AH

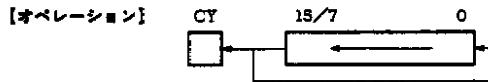
【バイト数】 3

【命令形式】

| 二モニック | オペランド | オペレーション・コード |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RETKA | imm8  | 0           | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|       |       | imm8        |   |   |   |   |   |   | — |   |   |   |   |   |   |   |   |

|            |                          |
|------------|--------------------------|
| <b>ROL</b> | 左方向のローテート<br>Rotate Left |
|------------|--------------------------|

【命令形式】 ROL dst, src



【オペランド】

| ニモニック      | オペランド (dst, src) |
|------------|------------------|
| <b>ROL</b> | reg. 1           |
|            | mem. 1           |
|            | reg. CL          |
|            | mem. CL          |
|            | reg. imm8        |
|            | mem. imm8        |

【フ ラ グ】 src=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
|    | x  | x |   |   |   |    | x  | U |   |   |   |

【説 明】 ○src=1のとき  
 第1オペランドで指定されるアスティネーション・オペランド (dst) の内容を1ビットだけ左シフトします。dstの内容のMSB(ビット7またはビット15)のデータはLSB(ビット0)へシフトされると同時に、CYフラグへも転送されます。また、MSBが変化したときはVフラグがセット(1)され、元のままのときはリセット(0)されます。

○src=CLまたはsrc=imm8のとき  
 第1オペランドで指定されるアスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけ左シフトします。dstの内容のMSB(ビット7またはビット15)のデータはLSB(ビット0)へシフトされると同時に、CYフラグへも転送されます。

**【記述例】** MOV [IX], BL  
 ROL BYTE PTR [IX], 1

**【バイト数】**

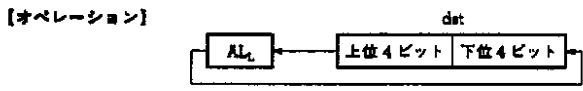
| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| ROL   | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命令形式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |             |   |   |     |   |   |   |   |     |     |     |
|-------|-----------|-------------|---|---|---|---|-------------|---|---|-----|---|---|---|---|-----|-----|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2           | 1 | 0 | 7   | 6 | 5 | 4 | 3 | 2   | 1   | 0   |
| ROL   | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | 1   | 1 | 0 | 0 | 0 | 0   | 0   | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | mod | 0 | 0 | 0 | 0 | 0   | mem |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |     |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | 1   | 1 | 0 | 0 | 0 | 0   | reg |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | mod | 0 | 0 | 0 | 0 | mem |     |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |     |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0           | 0 | W | 1   | 1 | 0 | 0 | 0 | 0   | reg |     |
|       | mem, imm8 | imm8        |   |   |   |   | -           |   |   |     |   |   |   |   |     |     |     |
|       |           | 1           | 1 | 0 | 0 | 0 | 0           | 0 | W | mod | 0 | 0 | 0 | 0 | mem |     |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |     |     |
|       |           | imm8        |   |   |   |   | -           |   |   |     |   |   |   |   |     |     |     |

**ROL4** 左方向のニブル・ローテート  
Rotate Left Nibble

**【命令形式】** ROL4 dst



**【オペランド】**

| ニモニック | オペランド (dst) |
|-------|-------------|
| ROL4  | reg8        |
|       | mem8        |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** アスティネーション・オペランド (dst) の内容を2桁のバックスBCDとして扱い、ALレジスタの下位4ビット (AL<sub>4</sub>) を介して、1桁分左に回転します。この命令の結果、ALレジスタの上位4ビットは保証されません。

- 【記述例】**
- MOV AL, 24H  
  ROL4 AL
  - MOV AL, BYTE PTR [IX]  
  ROL4 AL

**【バイト数】**

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| ROL4  | reg8  | 3    |
|       | mem8  | 3-5  |

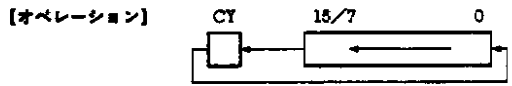
【命令形式】

| ニモニック | オペランド | オペレーション・コード |   |   |   |     |     |   |            |   |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|-----|-----|---|------------|---|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3   | 2   | 1 | 0          | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ROL4  | reg8  | 0           | 0 | 0 | 0 | 1   | 1   | 1 | 1          | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
|       |       | 1           | 1 | 0 | 0 | 0   | reg |   |            | — |   |   |   |   |   |   |   |
|       | mem8  | 0           | 0 | 0 | 0 | 1   | 1   | 1 | 1          | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
|       |       | mod         | 0 | 0 | 0 | mem |     |   | (disp-low) |   |   |   |   |   |   |   |   |
|       |       | (disp-high) |   |   |   |     |     |   | —          |   |   |   |   |   |   |   |   |



|             |                                            |
|-------------|--------------------------------------------|
| <b>ROLC</b> | キャリーを含む左方向のローテート<br>Rotate Left with Carry |
|-------------|--------------------------------------------|

**【命令形式】** ROLC dst, src



**【オペランド】**

| ニモニック       | オペランド (dst, src) |
|-------------|------------------|
| <b>ROLC</b> | reg, 1           |
|             | mem, 1           |
|             | reg, CL          |
|             | mem, CL          |
|             | reg, imm8        |
|             | mem, imm8        |

**【フラグ】** src=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
|    | x  | x |   |   |   |    | x  | U |   |   |   |

**【説明】** ○src=1のとき  
 第1オペランドで指定されるデスティネーション・オペランド (dst) の内容をCYフラグを通して1ビットだけ左シフトします。dstの内容のMSB(ビット7またはビット15)のデータはCYフラグへ転送されCYフラグのデータはLSB(ビット0)へ転送されます。

また、MSBが変化したときはVフラグがセット(1)され、元のままのときはリセット(0)されます。

○src=CLまたはsrc=imm8のとき  
 第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけCYフラグを通して左シフトします。dstの内容のMSB(ビット7またはビット15)のデータはCYフラグへ転送されCYフラグのデータはLSB(ビット0)へ転送されます。

- 【配 述 例】**
- ROLC AL, 1
  - ROLC CL, 1
  - ROLC DW, 1
  - ROLC AW, 1

**【バ イ ト 数】**

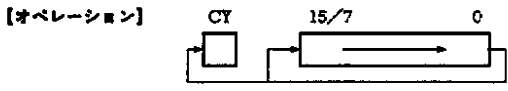
| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| ROLC  | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命 令 書 形 式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |   |   |   |             |     |   |   |   |     |     |     |
|-------|-----------|-------------|---|---|---|---|---|---|---|-------------|-----|---|---|---|-----|-----|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7           | 6   | 5 | 4 | 3 | 2   | 1   | 0   |
| ROLC  | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | 0 | W           | 1   | 1 | 0 | 1 | 0   | 0   | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | 0 | W           | mod | 0 | 1 | 0 | 0   | mem |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W | 1           | 1   | 0 | 1 | 0 | 0   | reg |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W | mod         | 0   | 1 | 0 | 0 | mem |     |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0 | 0 | W | 1           | 1   | 0 | 1 | 0 | 0   | reg |     |
|       | mem, imm8 | imm8        | — |   |   |   |   |   |   |             |     |   |   |   |     |     |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |     |
|       |           | imm8        | — |   |   |   |   |   |   |             |     |   |   |   |     |     |     |

|            |                           |
|------------|---------------------------|
| <b>ROR</b> | 右方向のローテート<br>Rotate Right |
|------------|---------------------------|

**【命令形式】** ROR dst, src



**【オペランド】**

| ニモニック      | オペランド (dst, src) |
|------------|------------------|
| <b>ROR</b> | reg, 1           |
|            | mem, 1           |
|            | reg, CL          |
|            | mem, CL          |
|            | reg, imm8        |
|            | mem, imm8        |

**【フラグ】** src=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
|    | x  | x |   |   |   | x  | U  |   |   |   |   |

**【説明】**

- src=1のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を1ビットだけ右シフトします。dstの内容のLSB (ビット0) のデータはMSB (ビット7またはビット15) ヘシフトされると同時に、CYフラグへも転送されます。また、MSBが変化したときはVフラグがセット(1)され、元のままのときはリセット(0)されます。
- src=CLまたはsrc=imm8のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけ右シフトします。dstの内容のLSB(ビット0)のデータはMSB(ビット7またはビット15) ヘシフトされると同時に、CYフラグへも転送されます。

- 【記述例】**
- ROR AL, 3
  - ROR CW, 6
  - ROR IX, 2

**【バイト数】**

| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| ROR   | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

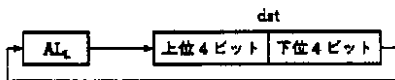
**【命令書形式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |   |   |   |             |     |   |   |      |      |     |
|-------|-----------|-------------|---|---|---|---|---|---|---|-------------|-----|---|---|------|------|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7           | 6   | 5 | 4 | 3    | 2    | 1   |
| ROR   | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | 0 | W           | 1   | 1 | 0 | 0    | 1    | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | 0 | W           | mod | 0 | 0 | 1    | mem. |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |      |      |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W | 1           | 1   | 0 | 0 | 1    | reg  |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W | mod         | 0   | 0 | 1 | mem. |      |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |      |      |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0 | 0 | 0 | W           | 1   | 1 | 0 | 0    | 1    | reg |
|       | mem, imm8 | imm8        |   |   |   |   |   |   |   | -           |     |   |   |      |      |     |
|       |           | (disp-low)  |   |   |   |   |   |   |   | (disp-high) |     |   |   |      |      |     |
|       |           | imm8        |   |   |   |   |   |   |   | -           |     |   |   |      |      |     |

**ROR4** 右方向のニブル・ローテート  
Rotate Right Nibble

**【命令形式】** ROR4 dst

**【オペレーション】**



**【オペランド】**

| ニモニック | オペランド (dst) |
|-------|-------------|
| ROR4  | reg8        |
|       | mem8        |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】**

デスティネーション・オペランド (dst) の内容を2桁のバケットBCDとして扱い、ALレジスタの下位4ビット (AL<sub>L</sub>) を介して、1桁分右に回転します。この命令の結果、ALレジスタの上位4ビットは保証されません。

**【記述例】**

- MOV AL, 24H  
ROR4 AL
- MOV AL, BYTE PTR [IX]  
ROR4 AL

**【バイト数】**

| ニモニック | オペランド | バイト数 |
|-------|-------|------|
| ROR4  | reg8  | 3    |
|       | mem8  | 3-5  |

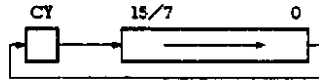
【命令形式】

| ニモニック | オペランド | オペレーション・コード |   |   |   |     |     |   |            |   |   |   |   |   |   |   |   |
|-------|-------|-------------|---|---|---|-----|-----|---|------------|---|---|---|---|---|---|---|---|
|       |       | 7           | 6 | 5 | 4 | 3   | 2   | 1 | 0          | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ROR4  | reg8  | 0           | 0 | 0 | 0 | 1   | 1   | 1 | 1          | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
|       |       | 1           | 1 | 0 | 0 | 0   | reg |   |            | — |   |   |   |   |   |   |   |
|       | mem8  | 0           | 0 | 0 | 0 | 1   | 1   | 1 | 1          | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
|       |       | mod         | 0 | 0 | 0 | mem |     |   | (disp-low) |   |   |   |   |   |   |   |   |
|       |       | (disp-high) |   |   |   |     |     | — |            |   |   |   |   |   |   |   |   |

|             |                                             |
|-------------|---------------------------------------------|
| <b>RORC</b> | キャリーを含む右方向のローテート<br>Rotate Right with Carry |
|-------------|---------------------------------------------|

**【命令形式】** RORC dst, src

**【オペレーション】**



**【オペランド】**

| ニモニック       | オペランド (dst, src) |
|-------------|------------------|
| <b>RORC</b> | reg, 1           |
|             | mem, 1           |
|             | reg, CL          |
|             | mem, CL          |
|             | reg, imm8        |
|             | mem, imm8        |

**【フラグ】**

src=1の場合

左記以外

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    | x  | x |   |   |   |

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    | x  | U |   |   |   |

**【説明】**

○src=1のとき

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容をCYフラグを通して1ビットだけ右シフトします。dstの内容のLSB(ビット0)のデータはCYフラグへ転送されCYフラグのデータはMSB(ビット7またはビット15)へ転送されます。  
 また、MSBが変化したときはVフラグがセット(1)され、元のままのときはリセット(0)されます。

○src=CLまたはsrc=imm8のとき

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけCYフラグを通して右シフトします。dstの内容のLSB(ビット0)のデータはCYフラグへ転送されCYフラグのデータはMSB(ビット7またはビット15)へ転送されます。

- 【配 送 例】**
- RORC AL, 1
  - RORC BL, 1
  - RORC CW, 1
  - RORC IX, 1

**【バ イ ト 数】**

| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| RORC  | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命 令 形 式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |             |   |   |     |   |   |   |   |   |     |     |
|-------|-----------|-------------|---|---|---|---|-------------|---|---|-----|---|---|---|---|---|-----|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2           | 1 | 0 | 7   | 6 | 5 | 4 | 3 | 2 | 1   | 0   |
| RORC  | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | 1   | 1 | 0 | 1 | 1 |   |     | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | mod | 0 | 1 | 1 |   |   | mem |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |   |     |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | 1   | 1 | 0 | 1 | 1 |   | reg |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | mod | 0 | 1 | 1 |   |   | mem |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |   |     |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0           | 0 | W | 1   | 1 | 0 | 1 | 1 |   | reg |     |
|       | mem, imm8 | imm8        |   |   |   |   | —           |   |   |     |   |   |   |   |   |     |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |   |     |     |
|       |           | imm8        |   |   |   |   | —           |   |   |     |   |   |   |   |   |     |     |



|             |                    |
|-------------|--------------------|
| <b>SET1</b> | ビットのセット<br>Set Bit |
|-------------|--------------------|

**【命令形式】** ① SET1 dst, src  
② SET1 dst

**【オペレーション】** 命令形式①: dstのビットn (nはsrcで指定) ←1  
命令形式②: dst←1

**【オペランド】** 命令形式①

| ニモニック       | オペランド (dst, src) |
|-------------|------------------|
| <b>SET1</b> | reg8, CL         |
|             | mem8, CL         |
|             | reg16, CL        |
|             | mem16, CL        |
|             | reg8, imm3       |
|             | mem8, imm3       |
|             | reg16, imm4      |
|             | mem16, imm4      |

命令形式②

| ニモニック       | オペランド (dst) |
|-------------|-------------|
| <b>SET1</b> | CY          |
|             | DIR         |

**【フ ラ グ】** 命令形式①

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    |    |   |   |   |   |

命令形式② (dst=CYの場合)

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
|    | 1  |   |   |   |   |

命令形式② (dst=DIRの場合)

|    |    |   |   |   |   |     |
|----|----|---|---|---|---|-----|
| AC | CY | V | P | S | Z | DIR |
|    |    |   |   |   |   | 1   |

**【説明】** 命令形式①：第1オペランドで指定されるデスティネーション・オペランド (dst) のビットn (nは第2オペランドで指定されるソース・オペランド (src) の内容) をセット(1)し、結果をデスティネーション・オペランド (dst) に格納します。

オペランドがreg8, CLまたはmem8, CLのとき、CLの値は下位3ビット (0-7) のみ有効です。

オペランドがreg16, CLまたはmem16, CLのとき、CLの値は下位4ビット (0-15) のみ有効です。

オペランドがreg8, imm3のとき、命令の4バイト目のイミディエト・データは下位3ビットのみ有効です。

オペランドがmem8, imm3のとき、命令の最終バイトのイミディエト・データは下位3ビットのみ有効です。

オペランドがreg16, imm4のとき、命令の4バイト目のイミディエト・データは下位4ビットのみ有効です。

オペランドがmem16, imm4のとき、命令の最終バイトのイミディエト・データは下位4ビットのみ有効です。

命令形式②：dst=CYの場合、CYフラグをセット(1)します。

dst=DIRの場合、DIRフラグをセット(1)します。また、MOVKB, CMPBK, CMPM, LDM, STM, INM, OUTMの各命令の実行時にインデクス・レジスタ (IX, IY) をオートデクリメントするように設定します。

**【送例】**

```
MOV CL, 4
SETI AL, CL
OUT ODAH, AL
```

【バイト数】

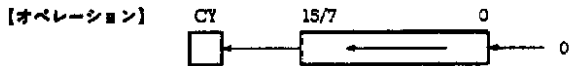
| ニモニック | オペランド       | バイト数 |
|-------|-------------|------|
| SET1  | reg8, CL    | 3    |
|       | mem8, CL    | 3-5  |
|       | reg16, CL   | 3    |
|       | mem16, CL   | 3-5  |
|       | reg8, imm3  | 4    |
|       | mem8, imm3  | 4-6  |
|       | reg16, imm4 | 4    |
|       | mem16, imm4 | 4-6  |
|       | CY          | 1    |
| DIR   | 1           |      |

【命令形式】

| ニモニック       | オペランド         | オペレーション・コード |   |      |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
|-------------|---------------|-------------|---|------|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|---|------|---|---|-----|---|--|--|--|
|             |               | 7           | 6 | 5    | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |             |   |      |   |   |     |   |  |  |  |
| SET1        | reg8, CL      | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1           | 1 | 0    | 0 | 0 | reg | — |  |  |  |
|             |               | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | (disp-high) |   | —    |   |   |     |   |  |  |  |
| mem8, CL    | mod 0 0 0 mem | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | (disp-high) |   | —    |   |   |     |   |  |  |  |
|             |               | (disp-high) |   | —    |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| reg16, CL   | 1 1 0 0 0 reg | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | —           |   |      |   |   |     |   |  |  |  |
|             |               | (disp-high) |   | —    |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| mem16, CL   | mod 0 0 0 mem | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | (disp-high) |   | —    |   |   |     |   |  |  |  |
|             |               | (disp-high) |   | —    |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| reg8, imm3  | 1 1 0 0 0 reg | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | imm3        |   |      |   |   |     |   |  |  |  |
|             |               | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | (disp-high) |   | imm3 |   |   |     |   |  |  |  |
| mem8, imm3  | mod 0 0 0 mem | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | (disp-high) |   | imm3 |   |   |     |   |  |  |  |
|             |               | (disp-high) |   | imm3 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| reg16, imm4 | 1 1 0 0 0 reg | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | imm4        |   |      |   |   |     |   |  |  |  |
|             |               | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | (disp-high) |   | imm4 |   |   |     |   |  |  |  |
| mem16, imm4 | mod 0 0 0 mem | 0           | 0 | 0    | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | (disp-high) |   | imm4 |   |   |     |   |  |  |  |
|             |               | (disp-high) |   | imm4 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| CY          |               | 1           | 1 | 1    | 1 | 1 | 0 | 0 | 1 | — |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |
| DIR         |               | 1           | 1 | 1    | 1 | 1 | 0 | 0 | 1 | — |   |   |   |   |   |   |   |             |   |      |   |   |     |   |  |  |  |

|            |                       |
|------------|-----------------------|
| <b>SHL</b> | 左方向のシフト<br>Shift Left |
|------------|-----------------------|

**【命令形式】** SHL dst, src



**【オペランド】**

| ニモニック      | オペランド (dst, src) |
|------------|------------------|
| <b>SHL</b> | reg, 1           |
|            | mem, 1           |
|            | reg, CL          |
|            | mem, CL          |
|            | reg, imm8        |
|            | mem, imm8        |

**【フラグ】** src=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
| U  | x  | x | x | x | x | U  | x  | U | x | x | x |

**【説明】**

- src=1のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を1ビットだけ左シフトします。

dstの内容のLSB (ビット0) には0がシフト・インされ、MSB (ビット7またはビット15) のデータはCYフラグにセットされます。

シフト後、符号ビット (ビット7またはビット15) が元のままであれば、Vフラグがクリアされます。
- src=CLまたはsrc=imm8のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけ左シフトします。1ビット、シフトすることにdstの内容のLSB (ビット0) には0がシフト・インされ、MSB (ビット7またはビット15) のデータはCYフラグにセットされます。

**【記述例】** IN AW, OCSH  
 MOV [IY], AW  
 SHL WORD PTR [IY], 12

**【バイト数】**

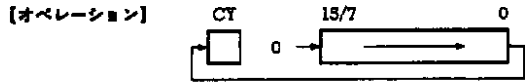
| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| SHL   | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命令形式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |   |   |             |     |   |   |   |     |     |
|-------|-----------|-------------|---|---|---|---|---|---|-------------|-----|---|---|---|-----|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0           | 7   | 6 | 5 | 4 | 3   | 2   |
| SHL   | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | W           | 1   | 1 | 1 | 0 | 0   | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0 | 0 | W           | mod | 1 | 0 | 0 | mem |     |
|       |           | (disp-low)  |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W           | 1   | 1 | 1 | 0 | 0   | reg |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0 | 1 | W           | mod | 1 | 0 | 0 | mem |     |
|       |           | (disp-low)  |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0 | 0 | W           | 1   | 1 | 1 | 0 | 0   | reg |
|       | mem, imm8 | imm8        |   |   |   |   |   |   | —           |     |   |   |   |     |     |
|       |           | (disp-low)  |   |   |   |   |   |   | (disp-high) |     |   |   |   |     |     |
|       |           | imm8        |   |   |   |   |   |   | —           |     |   |   |   |     |     |

|            |                        |
|------------|------------------------|
| <b>SHR</b> | 右方向のシフト<br>Shift Right |
|------------|------------------------|

**【命令形式】** SHR *dst, src*



**【オペランド】**

| ニモニック      | オペランド ( <i>dst, src</i> ) |
|------------|---------------------------|
| <b>SHR</b> | reg. 1                    |
|            | mem. 1                    |
|            | reg. CL                   |
|            | mem. CL                   |
|            | reg. imm8                 |
|            | mem. imm8                 |

**【フラグ】** *src*=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
| U  | X  | X | X | X | X | U  | X  | U | X | X | X |

**【説明】**

- *src*=1のとき
 

第1オペランドで指定されるデスティネーション・オペランド (*dst*) の内容を1ビットだけ右シフトします。

*dst*の内容のMSB(ビット7またはビット15)には0がシフト・インされ、LSB(ビット0)のデータはCYフラグにセットされます。

シフト後、符号ビット(ビット7またはビット15)が元のままであれば、Vフラグがクリアされます。

- *src*=CLまたは*src*=imm8のとき
 

第1オペランドで指定されるデスティネーション・オペランド (*dst*) の内容を第2オペランドで指定されるソース・オペランド (*src*) の内容のビット数だけ右シフトします。1ビット、シフトすることに*dst*の内容のMSB(ビット7またはビット15)には0がシフト・インされ、LSB(ビット0)のデータはCYフラグにセットされます。

**【記 述 例】**

- RCV:IN AL, ODAH
- SHR AL, 3
- BC RCV
- SHR CW, 8

**【バ イ ト 数】**

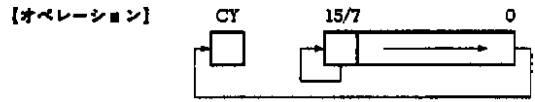
| ニモニック | オペランド     | バイト数 |
|-------|-----------|------|
| SHR   | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命 令 書 形 式】**

| ニモニック | オペランド     | オペレーション・コード |   |   |   |   |             |   |   |     |     |   |   |     |     |     |
|-------|-----------|-------------|---|---|---|---|-------------|---|---|-----|-----|---|---|-----|-----|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2           | 1 | 0 | 7   | 6   | 5 | 4 | 3   | 2   | 1   |
| SHR   | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | 0 | W   | 1   | 1 | 1 | 0   | 1   | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | 0 | W   | mod | 1 | 0 | 1   | mem |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |     |   |   |     |     |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | 1   | 1   | 1 | 0 | 1   | reg |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | mod | 1   | 0 | 1 | mem |     |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |     |   |   |     |     |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0           | 0 | 0 | W   | 1   | 1 | 1 | 0   | 1   | reg |
|       | mem, imm8 | imm8        |   |   |   |   | -           |   |   |     |     |   |   |     |     |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |     |   |   |     |     |     |
|       |           | imm8        |   |   |   |   | -           |   |   |     |     |   |   |     |     |     |

|             |                                     |
|-------------|-------------------------------------|
| <b>SHRA</b> | 右方向の算術シフト<br>Shift Right Arithmetic |
|-------------|-------------------------------------|

**【命令形式】** SHRA dst, src



**【オペランド】**

| 二重ニック       | オペランド (dst, src) |
|-------------|------------------|
| <b>SHRA</b> | reg. 1           |
|             | mem. 1           |
|             | reg. CL          |
|             | mem. CL          |
|             | reg. imm8        |
|             | mem. imm8        |

**【フ ラ グ】**    src=1の場合                      左記以外

|    |    |   |   |   |   |    |    |   |   |   |   |
|----|----|---|---|---|---|----|----|---|---|---|---|
| AC | CY | V | P | S | Z | AC | CY | V | P | S | Z |
| U  | x  | 0 | x | x | x | U  | x  | U | x | x | x |

**【説 明】**

- src=1のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を1ビットだけ算術的に右シフトします。

dstの内容のMSB (ビット7またはビット15) には元と同じ値がシフト・インされ、シフト後も符号は変化しません。LSB (ビット0) のデータはCYフラグにセットされます。
  
- src=CLまたはsrc=imm8のとき
 

第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を第2オペランドで指定されるソース・オペランド (src) の内容のビット数だけ右シフトします。dstの内容のMSB (ビット7またはビット15) には元と同じ値がシフト・インされ、シフト後も符号は変化しません。LSB (ビット0) のデータはCYフラグにセットされます。



- 【記述例】**
- MOV CL, 2
  - SHRA BL, CL
  - MOV CL, 9
  - SHRA DW, CL

**【バイト数】**

| 二モニック | オペランド     | バイト数 |
|-------|-----------|------|
| SHRA  | reg, 1    | 2    |
|       | mem, 1    | 2-4  |
|       | reg, CL   | 2    |
|       | mem, CL   | 2-4  |
|       | reg, imm8 | 3    |
|       | mem, imm8 | 3-5  |

**【命令形式】**

| 二モニック | オペランド     | オペレーション・コード |   |   |   |   |             |   |   |     |   |   |   |   |     |   |     |
|-------|-----------|-------------|---|---|---|---|-------------|---|---|-----|---|---|---|---|-----|---|-----|
|       |           | 7           | 6 | 5 | 4 | 3 | 2           | 1 | 0 | 7   | 6 | 5 | 4 | 3 | 2   | 1 | 0   |
| SHRA  | reg, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | 1   | 1 | 1 | 1 | 1 | 1   | 1 | reg |
|       | mem, 1    | 1           | 1 | 0 | 1 | 0 | 0           | 0 | W | mod | 1 | 1 | 1 | 1 | mem |   |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |   |     |
|       | reg, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | 1   | 1 | 1 | 1 | 1 | reg |   |     |
|       | mem, CL   | 1           | 1 | 0 | 1 | 0 | 0           | 1 | W | mod | 1 | 1 | 1 | 1 | mem |   |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |   |     |
|       | reg, imm8 | 1           | 1 | 0 | 0 | 0 | 0           | 0 | W | 1   | 1 | 1 | 1 | 1 | reg |   |     |
|       | mem, imm8 | imm8        |   |   |   |   | —           |   |   |     |   |   |   |   |     |   |     |
|       |           | 1           | 1 | 0 | 0 | 0 | 0           | 0 | W | mod | 1 | 1 | 1 | 1 | mem |   |     |
|       |           | (disp-low)  |   |   |   |   | (disp-high) |   |   |     |   |   |   |   |     |   |     |
|       |           | imm8        |   |   |   |   | —           |   |   |     |   |   |   |   |     |   |     |

|                                                   |                                                                                    |
|---------------------------------------------------|------------------------------------------------------------------------------------|
| <p><b>STM</b><br/><b>STMB</b><br/><b>STMW</b></p> | <p>ブロック・ストア<br/>Store Multiple<br/>Store Multiple Byte<br/>Store Multiple Word</p> |
|---------------------------------------------------|------------------------------------------------------------------------------------|

**【命令形式】** (repeat) STM [DS1-spec:] dst-block  
 (repeat) STMB  
 (repeat) STMW

**【オペレーション】** [W=0のとき] (IY) ←AL  
 DIR=0: IY←IY+1  
 DIR=1: IY←IY-1  
 [W=1のとき] (IY+1, IY) ←AW  
 DIR=0: IY←IY+2  
 DIR=1: IY←IY-2

**【オペランド】**

| ニモニック | オペランド                 |
|-------|-----------------------|
| STM   | [DS1-spec:] dst-block |
| STMB  | なし                    |
| STMW  | なし                    |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** ALレジスタの値、またはAWレジスタの値をIYレジスタでアドレスされるブロックにバイト/ワード・データ単位に繰り返し転送します。  
 IYレジスタの値は次のバイト/ワード処理のために1バイト/ワード・データが処理されることに自動的にインクリメント(+1/+2)またはデクリメント(-1/-2)されます。ブロックの方向は、DIRフラグの状態によって決定されます。  
 バイトかワードかの指定は、STM命令を用いる場合はオペランドの属性によって、STMB命令またはSTMW命令を用いる場合は、それぞれバイト、ワード・タイプに直接指定されます。  
 デスティネーション・ブロックは、常にDS1レジスタで指定されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。

- 【記 述 例】**
- REP STM DS1 : WORD\_VAR : DS1セグメント
  - REP STMB ; DS1セグメント

**【バ イ ト 数】** 1

**【命 令 書 形 式】**

| ニモニック       | オペランド                 | オペレーション・コード |   |   |   |   |   |   |   |
|-------------|-----------------------|-------------|---|---|---|---|---|---|---|
|             |                       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| <b>STM</b>  | [DS1-spec:] dst-block | 1           | 0 | 1 | 0 | 1 | 0 | 1 | W |
| <b>STMB</b> | な し                   |             |   |   |   |   |   |   |   |
| <b>STMW</b> |                       |             |   |   |   |   |   |   |   |

|            |                |
|------------|----------------|
| <b>SUB</b> | 減算<br>Subtract |
|------------|----------------|

**【命令形式】** SUB dst, src

**【オペランド, オペレーション】**

| 二モニック      | オペランド (dst, src) | オペレーション                                             |
|------------|------------------|-----------------------------------------------------|
| <b>SUB</b> | reg, reg'        | dst ← dst - src                                     |
|            | mem, reg         |                                                     |
|            | reg, mem         |                                                     |
|            | reg, imm         |                                                     |
|            | mem, imm         |                                                     |
|            | acc, imm         |                                                     |
|            |                  | [W=0のとき] AL ← AL - imm8<br>[W=1のとき] AW ← AW - imm16 |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| x  | x  | x | x | x | x |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) の内容から第2オペランドで指定されるソース・オペランド (src) の内容を減算し、結果をデスティネーション・オペランド (dst) に格納します。

**【配列例】** DLレジスタの内容からメモリ0:50Hの内容を減算してDLレジスタへストアする。

```

MOV AW, 0
MOV DSO, AW
MOV IX, 50H
SUB DL, DSO:BYTE PTR [IX]
    
```

【バイト数】

| 二モニック | オペランド     | バイト数 |
|-------|-----------|------|
| SUB   | reg, reg' | 2    |
|       | mem, reg  | 2-4  |
|       | reg, mem  | 2-4  |
|       | reg, imm  | 3, 4 |
|       | mem, imm  | 3-6  |
|       | acc, imm  | 2, 3 |

【命令書形式】

| 二モニック    | オペランド             | オペレーション・コード       |   |   |   |   |   |   |                   |             |   |   |     |     |     |      |     |
|----------|-------------------|-------------------|---|---|---|---|---|---|-------------------|-------------|---|---|-----|-----|-----|------|-----|
|          |                   | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0                 | 7           | 6 | 5 | 4   | 3   | 2   | 1    | 0   |
| SUB      | reg, reg'         | 0                 | 0 | 1 | 0 | 1 | 0 | 1 | W                 | 1           | 1 |   |     | reg |     | reg' |     |
|          | mem, reg          | 0                 | 0 | 1 | 0 | 1 | 0 | 0 | W                 | mod         |   |   | reg |     | mem |      |     |
|          |                   | (disp-low)        |   |   |   |   |   |   |                   | (disp-high) |   |   |     |     |     |      |     |
|          | reg, mem          | 0                 | 0 | 1 | 0 | 1 | 0 | 1 | W                 | mod         |   |   | reg |     | mem |      |     |
|          |                   | (disp-low)        |   |   |   |   |   |   |                   | (disp-high) |   |   |     |     |     |      |     |
|          | reg, imm          | 1                 | 0 | 0 | 0 | 0 | 0 | 0 | s                 | W           | 1 | 1 | 1   | 0   | 1   |      | reg |
|          |                   | imm8 or imm16-low |   |   |   |   |   |   |                   | imm16-high  |   |   |     |     |     |      |     |
| mem, imm | 1                 | 0                 | 0 | 0 | 0 | 0 | 0 | s | W                 | mod         | 1 | 0 | 1   |     | mem |      |     |
|          | (disp-low)        |                   |   |   |   |   |   |   | (disp-high)       |             |   |   |     |     |     |      |     |
|          | imm8 or imm16-low |                   |   |   |   |   |   |   | imm16-high        |             |   |   |     |     |     |      |     |
| acc, imm | 0                 | 0                 | 1 | 0 | 1 | 1 | 0 | W | imm8 or imm16-low |             |   |   |     |     |     |      |     |
|          | imm16-high        |                   |   |   |   |   |   |   | —                 |             |   |   |     |     |     |      |     |

SUB4S
10進減算  
Subtract Nibble String

**【命令形式】** SUB4S [DS1-spec:] dst-string, [Seg-spec:] src-string  
SUB4S

**【オペレーション】** BCDストリング (Y, CL) ← BCDストリング (Y, CL) - BCDストリング (X, CL)

**【オペランド】**

| ニモニック | オペランド (dst, src)                               |
|-------|------------------------------------------------|
| SUB4S | [DS1-spec:] dst-string, [Seg-spec:] src-string |
|       | なし                                             |

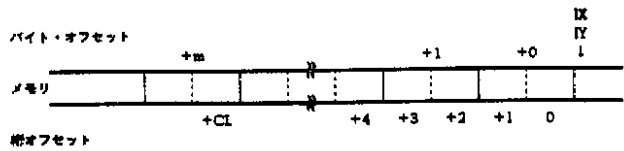
**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| U  | x  | U | U | U | x |

**【説明】** IYレジスタでアドレスされるバケットBCDストリングからIXレジスタでアドレスされるバケットBCDストリングを減算し、結果をIYレジスタでアドレスされるストリングにストアします。ストリング量 (BCD桁数) は、CLレジスタ (CLの内容がdならばd桁) によって決定され、1-254桁まで可能です。

デスティネーション・ストリングは、常にDS1レジスタで指されるセグメント内にロケートされる必要があり、セグメント・オーバーライドはできません。一方、ソース・ストリングは、デフォルト・セグメント・レジスタはDS0レジスタとなっていますが、セグメント・オーバーライド可能で、任意のセグメント・レジスタで指定されるセグメント内にロケートできます。

バケットBCDストリングのフォーマットを次に示します。



**注意** BCDストリング命令は常に偶数桁単位で動作します。このため桁数として偶数を指定したときは演算結果、および各フラグは正しく働きますが、桁数と

第 2 章 命 令

して奇数を指定した場合は奇数+1の偶数桁演算を実行し、演算結果、および各フラグは偶数桁の演算結果を示しますので、桁数として奇数を指定する場合は次のようにしてください。

桁数が奇数のときのBCD減算命令は、最上位バイトの上位4ビットを“0”にしてから実行してください。この結果ローが立った場合は最上位バイトの上位4ビットは“9”になります。

**【配 送 例】**   MOV IX, OFFSET VAR\_1  
                   MOV IY, OFFSET VAR\_2  
                   MOV CL, 4  
                   SUB4S

**【バ イ ト 数】**   2

**【命 令 語 形 式】**

| ニモニック | オペランド                                        | オペレーション・コード |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------|----------------------------------------------|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|       |                                              | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SUB4S | [DS1-spec:]dst-string, [Seg-spec:]src-string | 0           | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|       | なし                                           |             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

|             |                                  |
|-------------|----------------------------------|
| <b>SUBC</b> | キャリーを含む減算<br>Subtract with Carry |
|-------------|----------------------------------|

**【命令形式】** SUBC dst, src

**【オペランド, オペレーション】**

| ニモニック       | オペランド (dst, src) | オペレーション                                                       |
|-------------|------------------|---------------------------------------------------------------|
| <b>SUBC</b> | reg, reg'        | dst ← dst - src - CY                                          |
|             | mem, reg         |                                                               |
|             | reg, mem         |                                                               |
|             | reg, imm         |                                                               |
|             | mem, imm         |                                                               |
|             | acc, imm         | [W=0のとき] AL ← AL - imm8 - CY<br>[W=1のとき] AW ← AW - imm16 - CY |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| x  | x  | x | x | x | x |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) の内容から第2オペランドで指定されるソース・オペランド (src) の内容をCYフラグの内容も含めて減算し、結果をデスティネーション・オペランド (dst) に格納します。

**【記述例】** SUBC DL, BYTE PTR [IX]

**【バイト数】**

| ニモニック       | オペランド     | バイト数 |
|-------------|-----------|------|
| <b>SUBC</b> | reg, reg' | 2    |
|             | mem, reg  | 2-4  |
|             | reg, mem  | 2-4  |
|             | reg, imm  | 3, 4 |
|             | mem, imm  | 3-6  |
|             | acc, imm  | 2, 3 |



[命令書形式]

| エモニック    | オペランド      | オペレーション・コード       |   |   |   |   |             |   |                   |     |     |     |      |     |     |   |   |
|----------|------------|-------------------|---|---|---|---|-------------|---|-------------------|-----|-----|-----|------|-----|-----|---|---|
|          |            | 7                 | 6 | 5 | 4 | 3 | 2           | 1 | 0                 | 7   | 6   | 5   | 4    | 3   | 2   | 1 | 0 |
| SUBC     | reg, reg'  | 0                 | 0 | 0 | 1 | 1 | 0           | 1 | W                 | 1   | 1   | reg | reg' |     |     |   |   |
|          | mem, reg   | 0                 | 0 | 0 | 1 | 1 | 0           | 0 | W                 | mod | reg | mem |      |     |     |   |   |
|          |            | (disp-low)        |   |   |   |   | (disp-high) |   |                   |     |     |     |      |     |     |   |   |
|          | reg, mem   | 0                 | 0 | 0 | 1 | 1 | 0           | 1 | W                 | mod | reg | mem |      |     |     |   |   |
|          |            | (disp-low)        |   |   |   |   | (disp-high) |   |                   |     |     |     |      |     |     |   |   |
|          | reg, imm   | 1                 | 0 | 0 | 0 | 0 | 0           | s | W                 | 1   | 1   | 0   | 1    | 1   | reg |   |   |
|          |            | imm8 or imm16-low |   |   |   |   | imm16-high  |   |                   |     |     |     |      |     |     |   |   |
|          | mem, imm   | 1                 | 0 | 0 | 0 | 0 | 0           | s | W                 | mod | 0   | 1   | 1    | mem |     |   |   |
|          |            | (disp-low)        |   |   |   |   | (disp-high) |   |                   |     |     |     |      |     |     |   |   |
|          |            | imm8 or imm16-low |   |   |   |   | imm16-high  |   |                   |     |     |     |      |     |     |   |   |
| acc, imm | 0          | 0                 | 0 | 1 | 1 | 1 | 0           | W | imm8 or imm16-low |     |     |     |      |     |     |   |   |
|          | imm16-high |                   |   |   |   | - |             |   |                   |     |     |     |      |     |     |   |   |

|             |             |
|-------------|-------------|
| <b>TEST</b> | テスト<br>Test |
|-------------|-------------|

**【命令形式】** TEST *dst, src*

**【オペランド、オペレーション】**

| ニモニック       | オペランド ( <i>dst, src</i> ) | オペレーション                                 |
|-------------|---------------------------|-----------------------------------------|
| <b>TEST</b> | reg, reg'                 | dst/\src                                |
|             | mem, reg                  |                                         |
|             | reg, mem                  |                                         |
|             | reg, imm                  |                                         |
|             | mem, imm                  |                                         |
|             | acc, imm                  | [W=0のとき] AL/\imm8<br>[W=1のとき] AW/\imm16 |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| U  | 0  | 0 | x | x | x |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (*dst*) と第2オペランドで指定されるソース・オペランド (*src*) の論理積 (AND) をとります。結果はどこへも格納せず、フラグを変化させます。

**【記述例】** IN AL, 0D8H  
TEST AL, 'A'

**【バイト数】**

| ニモニック       | オペランド     | バイト数 |
|-------------|-----------|------|
| <b>TEST</b> | reg, reg' | 2    |
|             | mem, reg  | 2-4  |
|             | reg, mem  |      |
|             | reg, imm  | 3, 4 |
|             | mem, imm  | 3-6  |
|             | acc, imm  | 2, 3 |

**【命令形式】**

| ニモニック    | オペランド      | オペレーション・コード       |   |   |   |   |   |             |                   |     |   |   |      |   |     |   |   |
|----------|------------|-------------------|---|---|---|---|---|-------------|-------------------|-----|---|---|------|---|-----|---|---|
|          |            | 7                 | 6 | 5 | 4 | 3 | 2 | 1           | 0                 | 7   | 6 | 5 | 4    | 3 | 2   | 1 | 0 |
| TEST     | reg, reg'  | 1                 | 0 | 0 | 0 | 0 | 1 | 0           | W                 | 1   | 1 |   | reg' |   | reg |   |   |
|          | mem, reg   | 1                 | 0 | 0 | 0 | 0 | 1 | 0           | W                 | mod |   |   | reg  |   | mem |   |   |
|          |            | (disp-low)        |   |   |   |   |   | (disp-high) |                   |     |   |   |      |   |     |   |   |
|          | reg, mem   | 1                 | 0 | 0 | 0 | 0 | 1 | 0           | W                 | mod |   |   | reg  |   | mem |   |   |
|          |            | (disp-low)        |   |   |   |   |   | (disp-high) |                   |     |   |   |      |   |     |   |   |
|          | reg, imm   | 1                 | 1 | 1 | 1 | 0 | 1 | 1           | W                 | 1   | 1 | 0 | 0    |   | reg |   |   |
|          |            | imm8 or imm16-low |   |   |   |   |   | imm16-high  |                   |     |   |   |      |   |     |   |   |
|          | mem, imm   | 1                 | 1 | 1 | 1 | 0 | 1 | 1           | W                 | mod | 0 | 0 | 0    |   | mem |   |   |
|          |            | (disp-low)        |   |   |   |   |   | (disp-high) |                   |     |   |   |      |   |     |   |   |
|          |            | imm8 or imm16-low |   |   |   |   |   | imm16-high  |                   |     |   |   |      |   |     |   |   |
| acc, imm | 1          | 0                 | 1 | 0 | 1 | 0 | 0 | W           | imm8 or imm16-low |     |   |   |      |   |     |   |   |
|          | imm16-high |                   |   |   |   |   | — |             |                   |     |   |   |      |   |     |   |   |

|              |                     |
|--------------|---------------------|
| <b>TEST1</b> | ビットのテスト<br>Test Bit |
|--------------|---------------------|

**【命令形式】** TEST1 dst, src

**【オペレーション】** dstのビットn=0 (nはsrcで指定) のとき: Z←1  
dstのビットn=1 (nはsrcで指定) のとき: Z←0

**【オペランド】**

| ニモニック        | オペランド (dst, src) |
|--------------|------------------|
| <b>TEST1</b> | reg8, CL         |
|              | mem8, CL         |
|              | reg16, CL        |
|              | mem16, CL        |
|              | reg8, imm3       |
|              | mem8, imm3       |
|              | reg16, imm4      |
|              | mem16, imm4      |

**【フラグ】**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CF | V | P | S | Z |
| U  | 0  | 0 | U | U | × |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) のビットn (nは第2オペランドで指定されるソース・オペランド (src) の内容) が0のときはZフラグがセット(1)され、ビットnが1のときはリセット(0)されます。

オペランドがreg8, CLまたはmem8, CLのとき、CLの値は下位3ビット(0-7)のみ有効です。

オペランドがreg16, CLまたはmem16, CLのとき、CLの値は下位4ビット(0-15)のみ有効です。

オペランドがreg8, imm3のとき、命令の4バイト目のイミューディアット・データは下位3ビットのみ有効です。

オペランドがmem8, imm3のとき、命令の最終バイトのイミューディアット・データは下位3ビットのみ有効です。

オペランドがreg16, imm4のとき、命令の4バイト目のイミューディアット・データは下位4ビットのみ有効です。

オペランドがmem16, imm4のとき、命令の最終バイトのイミディエイト・データは下位4ビットのみ有効です。

**【記述例】**  
MOV CL, 01  
IN AL, ODAH  
TEST1 AL, CL; ビット1をテスト

**【バイト数】**

| ニモニック | オペランド       | バイト数 |
|-------|-------------|------|
| TEST1 | reg8, CL    | 3    |
|       | mem8, CL    | 3-5  |
|       | reg16, CL   | 3    |
|       | mem16, CL   | 3-5  |
|       | reg8, imm3  | 4    |
|       | mem8, imm3  | 4-6  |
|       | reg16, imm4 | 4    |
|       | mem16, imm4 | 4-6  |

【命令書形式】

| 二モニック       | オペランド       | オペレーション・コード |   |   |     |     |     |      |            |            |      |   |   |   |   |   |   |
|-------------|-------------|-------------|---|---|-----|-----|-----|------|------------|------------|------|---|---|---|---|---|---|
|             |             | 7           | 6 | 5 | 4   | 3   | 2   | 1    | 0          | 7          | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
| TEST1       | reg8, CL    | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 0 | 0 | 0 | 0 |
|             |             | 1           | 1 | 0 | 0   | 0   | reg |      |            |            | ---  |   |   |   |   |   |   |
|             | mem8, CL    | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 0 | 0 | 0 | 0 |
|             |             | mod         | 0 | 0 | 0   | mem |     |      |            | (disp-low) |      |   |   |   |   |   |   |
|             |             | (disp-high) |   |   |     |     |     |      |            | ---        |      |   |   |   |   |   |   |
|             | reg16, CL   | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 0 | 0 | 0 | 1 |
|             |             | 1           | 1 | 0 | 0   | 0   | reg |      |            |            | ---  |   |   |   |   |   |   |
|             | mem16, CL   | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 0 | 0 | 0 | 1 |
|             |             | mod         | 0 | 0 | 0   | mem |     |      |            | (disp-low) |      |   |   |   |   |   |   |
|             |             | (disp-high) |   |   |     |     |     |      |            | ---        |      |   |   |   |   |   |   |
|             | reg8, imm3  | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 1 | 0 | 0 | 0 |
|             |             | 1           | 1 | 0 | 0   | 0   | reg |      |            |            | imm3 |   |   |   |   |   |   |
|             | mem8, imm3  | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 1 | 0 | 0 | 0 |
|             |             | mod         | 0 | 0 | 0   | mem |     |      |            | (disp-low) |      |   |   |   |   |   |   |
|             |             | (disp-high) |   |   |     |     |     |      |            | imm3       |      |   |   |   |   |   |   |
|             | reg16, imm4 | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 1 | 0 | 0 | 1 |
|             |             | 1           | 1 | 0 | 0   | 0   | reg |      |            |            | imm4 |   |   |   |   |   |   |
|             | mem16, imm4 | 0           | 0 | 0 | 0   | 1   | 1   | 1    | 1          | 0          | 0    | 0 | 1 | 1 | 0 | 0 | 1 |
| mod         |             | 0           | 0 | 0 | mem |     |     |      | (disp-low) |            |      |   |   |   |   |   |   |
| (disp-high) |             |             |   |   |     |     |     | imm4 |            |            |      |   |   |   |   |   |   |

|                                              |                                          |
|----------------------------------------------|------------------------------------------|
| <h2 style="margin: 0;">TRANS<br/>TRANSB</h2> | 変換テーブルの転送<br>Translate<br>Translate Byte |
|----------------------------------------------|------------------------------------------|

**【命令形式】**    **TRANS src-table**  
                   **TRANS**  
                   **TRANSB**

**【オペレーション】**    **AL ← (BW + AL)**

**【オペランド】**

| ニモニック         | オペランド     |
|---------------|-----------|
| <b>TRANS</b>  | src-table |
|               | なし        |
| <b>TRANSB</b> | なし        |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】**            **明**    **BWレジスタとALレジスタによってアドレスされる256バイトの変換テーブルの1バイトをALレジスタに転送します。このときBWレジスタはテーブルの先頭アドレスを指し、先頭アドレスから256バイト内のオフセット値をALレジスタが指定します。**

**【記述例】**        **TRANS SIN\_TBL**

**【バイト数】**     **1**

**【命令形式】**

| ニモニック         | オペランド     | オペレーション・コード |   |   |   |   |   |   |   |
|---------------|-----------|-------------|---|---|---|---|---|---|---|
|               |           | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| <b>TRANS</b>  | src-table | 1           | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
|               | なし        |             |   |   |   |   |   |   |   |
| <b>TRANSB</b> | なし        |             |   |   |   |   |   |   |   |

|            |                   |
|------------|-------------------|
| <b>XCH</b> | データ交換<br>Exchange |
|------------|-------------------|

**【命令形式】** XCH dst, src

**【オペレーション】** dst←src

**【オペランド】**

| ニモニック      | オペランド (dst, src) |
|------------|------------------|
| <b>XCH</b> | reg, reg'        |
|            | mem, reg         |
|            | reg, mem         |
|            | AW, reg16        |
|            | reg16, AW        |

**【フラグ】**

| AC | CY | V | P | S | Z |
|----|----|---|---|---|---|
|    |    |   |   |   |   |

**【説明】** 第1オペランドで指定されるデスティネーション・オペランド (dst) と第2オペランドで指定されるソース・オペランド (src) の内容を交換します。

**【記述例】** MOV AW, 100H  
 MOV BW, 50H  
 XCH AW, BW  
 ; AW=50H, BW=100Hとなる。

**【バイト数】**

| ニモニック      | オペランド     | バイト数 |
|------------|-----------|------|
| <b>XCH</b> | reg, reg' | 2    |
|            | mem, reg  | 2-4  |
|            | reg, mem  | 2-4  |
|            | AW, reg16 | 1    |
|            | reg16, AW | 1    |



【命令形式】

| ニモニック     | オペランド     | オペレーション・コード |   |   |   |   |   |     |             |     |   |   |   |     |   |      |
|-----------|-----------|-------------|---|---|---|---|---|-----|-------------|-----|---|---|---|-----|---|------|
|           |           | 7           | 6 | 5 | 4 | 3 | 2 | 1   | 0           | 7   | 6 | 5 | 4 | 3   | 2 | 1    |
| XCH       | reg, reg' | 1           | 0 | 0 | 0 | 0 | 1 | 1   | W           | 1   | 1 |   |   | reg |   | reg' |
|           | mem, reg  | 1           | 0 | 0 | 0 | 0 | 1 | 1   | W           | mod |   |   |   | reg |   | mem  |
|           |           | (disp-low)  |   |   |   |   |   |     | (disp-high) |     |   |   |   |     |   |      |
|           | reg, mem  | 1           | 0 | 0 | 0 | 0 | 1 | 1   | W           | mod |   |   |   | reg |   | mem  |
|           |           | (disp-low)  |   |   |   |   |   |     | (disp-high) |     |   |   |   |     |   |      |
|           | AW, reg16 | 1           | 0 | 0 | 1 | 0 |   |     | reg         | —   |   |   |   |     |   |      |
| reg16, AW | 1         | 0           | 0 | 1 | 0 |   |   | reg | —           |     |   |   |   |     |   |      |

備考 XCH AW, AWはNOP命令と同じオペレーション・コードです。

|            |                        |
|------------|------------------------|
| <b>XOR</b> | 排他的論理和<br>Exclusive Or |
|------------|------------------------|

**[命令形式]** XOR dst, src

**[オペランド, オペレーション]**

| ニモニック      | オペランド (dst, src) | オペレーション                                             |
|------------|------------------|-----------------------------------------------------|
| <b>XOR</b> | reg, reg'        | dst ← dst ∨ src                                     |
|            | mem, reg         |                                                     |
|            | reg, mem         |                                                     |
|            | reg, imm         |                                                     |
|            | mem, imm         |                                                     |
|            | acc, imm         | [W=0のとき] AL ← AL ∨ imm8<br>[W=1のとき] AW ← AW ∨ imm16 |

**[フラグ]**

|    |    |   |   |   |   |
|----|----|---|---|---|---|
| AC | CY | V | P | S | Z |
| U  | 0  | 0 | x | x | x |

**[説明]** 第1オペランドで指定されるデスティネーション・オペランド (dst) と第2オペランドで指定されるソース・オペランド (src) の排他的論理和 (XOR) をとり、結果をデスティネーション・オペランド (dst) に格納します。

- [記述例]**
- XOR CL, DL
  - XOR CW, CW : CWレジスタのクリア
  - XOR AW, DW

**[バイト数]**

| ニモニック      | オペランド     | バイト数 |
|------------|-----------|------|
| <b>XOR</b> | reg, reg' | 2    |
|            | mem, reg  | 2-4  |
|            | reg, mem  | 2-4  |
|            | reg, imm  | 3, 4 |
|            | mem, imm  | 3-6  |
|            | acc, imm  | 2, 3 |

[命令書形式]

| 二モニック | オペランド                 | オペレーション・コード       |   |   |   |   |   |   |   |                   |     |     |      |     |     |   |   |
|-------|-----------------------|-------------------|---|---|---|---|---|---|---|-------------------|-----|-----|------|-----|-----|---|---|
|       |                       | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7                 | 6   | 5   | 4    | 3   | 2   | 1 | 0 |
| XOR   | reg, reg'             | 0                 | 0 | 1 | 1 | 0 | 0 | 1 | W | 1                 | 1   | reg | reg' |     |     |   |   |
|       | mem, reg              | 0                 | 0 | 1 | 1 | 0 | 0 | 0 | W | mod               | reg | mem |      |     |     |   |   |
|       |                       | (disp-low)        |   |   |   |   |   |   |   | (disp-high)       |     |     |      |     |     |   |   |
|       | reg, mem              | 0                 | 0 | 1 | 1 | 0 | 0 | 1 | W | mod               | reg | mem |      |     |     |   |   |
|       |                       | (disp-low)        |   |   |   |   |   |   |   | (disp-high)       |     |     |      |     |     |   |   |
|       | reg, imm <sup>8</sup> | 1                 | 0 | 0 | 0 | 0 | 0 | 0 | W | 1                 | 1   | 1   | 1    | 0   | reg |   |   |
|       |                       | imm8 or imm16-low |   |   |   |   |   |   |   | imm16-high        |     |     |      |     |     |   |   |
|       | mem, imm              | 1                 | 0 | 0 | 0 | 0 | 0 | 0 | W | mod               | 1   | 1   | 0    | mem |     |   |   |
|       |                       | (disp-low)        |   |   |   |   |   |   |   | (disp-high)       |     |     |      |     |     |   |   |
|       |                       | imm8 or imm16-low |   |   |   |   |   |   |   | imm16-high        |     |     |      |     |     |   |   |
|       | acc, imm              | 0                 | 0 | 1 | 1 | 0 | 1 | 0 | W | imm8 or imm16-low |     |     |      |     |     |   |   |
|       |                       | imm16-high        |   |   |   |   |   |   |   | —                 |     |     |      |     |     |   |   |

注 アセンブラ、コンパイラによっては、次に示すようなコードが生成されることがあります。

| 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2   | 1 | 0 |
|------|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|---|
| 1    | 0 | 0 | 0 | 0 | 0 | 1 | W | 1 | 1 | 1 | 1 | 0 | reg |   |   |
| imm8 |   |   |   |   |   |   |   | — |   |   |   |   |     |   |   |

このような場合でも正常に命令を実行します。ただし、エミュレータによっては、この命令に対する逆アセンブラ機能、アセンブラ機能がサポートされていない場合がありますのでご注意ください。

## 2.2 命令実行クロック数

各命令の実行クロック数とワード転送回数をニモニックのアルファベット順に表2-8に示します。

### (1) クロック数

表の中で示されている数値は、実行ユニットが命令実行に必要とする時間で、次の条件に基づいています。

- (a) プリフェッチ時間、プリデコード時間、バス使用のための待ち時間などは含まれません。
- (b) メモリ・アクセスは、0ウェイトを想定しています。  
つまり、1バス・サイクルのクロック数は次のようになります。
  - V33A, V53A以外：4クロック
  - V33A, V53A：2クロック
- (c) I/Oアクセスは0ウェイトを想定しています。
- (d) プリミティブ・ブロック転送命令、プリミティブ入出力命令はリビート・プリフィクスも含まれています。
- (e) 奇数アドレスへのワード・データの場合、バス・サイクルを2回起動します。  
奇数アドレス、偶数アドレスでそれぞれ場合分けして示しています。
- (f) 外部データ・バス幅は次のとおりです。

- ★ ●V20, V20HL, V40, V40HL：8ビット
- ★ ●V30, V30HL, V50, V50HL, V33A<sup>註</sup>, V53A<sup>註</sup>：16ビット

注 バス・サイジング機能で16ビットを指定した場合、8ビット指定を行う場合、偶数番地へのワード・データに対するバス・サイクルを2倍にしてください。

- (g) V33A, V53Aは通常アドレス・モードについて表記しています。

### (2) ワード転送回数

ワード転送回数とは命令実行によって発生するワード・データ（16ビット）のバス・アクセスを必要とする回数です。

この値によって、ウェイト・スタート挿入時の命令実行クロック数を次の式より求めることができます。

- 偶数アドレス・アクセス時：(0ウェイト時の命令実行クロック数)+(ワード転送回数)  
×(ウェイト・ステート数)
- 奇数アドレス・アクセス時：(0ウェイト時の命令実行クロック数)+(ワード転送回数)  
×(ウェイト・ステート数)×2

表 2-8 命令実行クロック数一覧 (1/15)

| ニモニック             | オペランド                      | ワード<br>毎演算数 | 条 件    |      | ク ロ ッ ク 数  |            |            |            |            |
|-------------------|----------------------------|-------------|--------|------|------------|------------|------------|------------|------------|
|                   |                            |             | W      | アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V33A |
| ADD               | reg, reg'                  | 0           | —      | —    | 2          | 2          | 2          | 2          | 2          |
|                   | mem, reg                   | 2           | 0      | —    | 16         | 16         | 13         | 13         | 7          |
|                   |                            |             | 1      | 寄 数  | 24         | 24         | 21         | 21         | 11         |
|                   |                            |             |        | 偶 数  | 16         | 16         | 13         | 7          |            |
|                   | reg, mem                   | 1           | 0      | —    | 11         | 11         | 10         | 10         | 6          |
|                   |                            |             | 1      | 寄 数  | 15         | 15         | 14         | 14         | 8          |
|                   |                            |             |        | 偶 数  | 11         | 11         | 10         | 6          |            |
|                   | reg, imm                   | 0           | —      | —    | 4          | 4          | 4          | 4          | 2          |
|                   | mem, imm                   | 2           | 0      | —    | 18         | 18         | 15         | 15         | 7          |
|                   |                            |             | 1      | 寄 数  | 26         | 26         | 23         | 23         | 11         |
| 偶 数               |                            |             | 18     | 18   | 15         | 7          |            |            |            |
| acc, imm          | 0                          | —           | —      | 4    | 4          | 4          | 4          | 2          |            |
| ADDS <sup>注</sup> | [DS1-spec:] dst-addr,      | 0           | —      | —    | 19×m+7     | 19×m+7     | 19×m+7     | 19×m+7     | 18×m+2     |
|                   | [Src-spec:] src-addr<br>なし | 0           | —      | —    | 19×m+7     | 19×m+7     | 19×m+7     | 19×m+7     | 18×m+2     |
| ADDC              | reg, reg'                  | 0           | —      | —    | 2          | 2          | 2          | 2          | 2          |
|                   | mem, reg                   | 2           | 0      | —    | 16         | 16         | 13         | 13         | 7          |
|                   |                            |             | 1      | 寄 数  | 24         | 24         | 21         | 21         | 11         |
|                   |                            |             |        | 偶 数  | 16         | 16         | 13         | 7          |            |
|                   | reg, mem                   | 1           | 0      | —    | 11         | 11         | 10         | 10         | 6          |
|                   |                            |             | 1      | 寄 数  | 15         | 15         | 14         | 14         | 8          |
|                   |                            |             |        | 偶 数  | 11         | 11         | 10         | 6          |            |
|                   | reg, imm                   | 0           | —      | —    | 4          | 4          | 4          | 4          | 2          |
|                   | mem, imm                   | 2           | 0      | —    | 18         | 18         | 15         | 15         | 7          |
|                   |                            |             | 1      | 寄 数  | 26         | 26         | 23         | 23         | 11         |
| 偶 数               |                            |             | 18     | 18   | 15         | 7          |            |            |            |
| acc, imm          | 0                          | —           | —      | 4    | 4          | 4          | 4          | 2          |            |
| ADMA              | なし                         | 0           | —      | —    | 3          | 3          | 3          | 3          | 2          |
| ADMS              | なし                         | 0           | —      | —    | 3          | 3          | 3          | 3          | 2          |
| ADMA              | なし                         | 0           | —      | —    | 7          | 7          | 7          | 7          | 4          |
| ADMS              | なし                         | 0           | —      | —    | 7          | 7          | 7          | 7          | 4          |
| AND               | reg, reg'                  | 0           | —      | —    | 2          | 2          | 2          | 2          | 2          |
|                   | mem, reg                   | 2           | 0      | —    | 16         | 16         | 13         | 13         | 7          |
|                   |                            |             | 1      | 寄 数  | 24         | 24         | 21         | 21         | 11         |
|                   |                            |             |        | 偶 数  | 16         | 16         | 13         | 7          |            |
|                   | reg, mem                   | 1           | 0      | —    | 11         | 11         | 10         | 10         | 6          |
|                   |                            |             | 1      | 寄 数  | 15         | 15         | 14         | 14         | 8          |
|                   |                            |             |        | 偶 数  | 11         | 11         | 10         | 6          |            |
|                   | reg, imm                   | 0           | —      | —    | 4          | 4          | 4          | 4          | 2          |
|                   | mem, imm                   | 2           | 0      | —    | 18         | 18         | 15         | 15         | 7          |
|                   |                            |             | 1      | 寄 数  | 26         | 26         | 23         | 23         | 11         |
| 偶 数               |                            |             | 18     | 18   | 15         | 7          |            |            |            |
| acc, imm          | 0                          | —           | —      | 4    | 4          | 4          | 4          | 2          |            |
| BC                | short-label                | 0           | CT=1の時 | —    | 14         | 14         | 14         | 14         | 6          |
|                   |                            |             | CT=0の時 | —    | 4          | 4          | 4          | 4          | 3          |
| BCWZ              | short-label                | 0           | CW=0の時 | —    | 5          | 5          | 5          | 5          | 3          |
|                   |                            |             | CW=0の時 | —    | 13         | 13         | 13         | 13         | 6          |

注 m: BCD桁数×1/2

表 2-5 命令実行クロック数一覧 (2/16)

| ニモニック    | オペランド       | ワード<br>指定回数 | 条件               | クロック数  |            |            |            |            |            |
|----------|-------------|-------------|------------------|--------|------------|------------|------------|------------|------------|
|          |             |             |                  | W アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V33A |
| BE       | short-label | 0           | Z=1のとき           | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | Z=0のとき           | 4      | 4          | 4          | 4          | 3          |            |
| BGE      | short-label | 0           | SVF=1のとき         | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | SVF=0のとき         | 14     | 14         | 14         | 14         | 6          |            |
| BGT      | short-label | 0           | (SVF)VZ=1<br>のとき | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | (SVF)VZ=0<br>のとき | 14     | 14         | 14         | 14         | 6          |            |
| BH       | short-label | 0           | CVZ=1のとき         | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | CVZ=0のとき         | 14     | 14         | 14         | 14         | 6          |            |
| BL       | short-label | 0           | CT=1のとき          | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | CT=0のとき          | 4      | 4          | 4          | 4          | 3          |            |
| BLE      | short-label | 0           | (SVF)VZ=1<br>のとき | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | (SVF)VZ=0<br>のとき | 4      | 4          | 4          | 4          | 3          |            |
| BLT      | short-label | 0           | SVF=1のとき         | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | SVF=0のとき         | 4      | 4          | 4          | 4          | 3          |            |
| BN       | short-label | 0           | S=1のとき           | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | S=0のとき           | 4      | 4          | 4          | 4          | 3          |            |
| BNC      | short-label | 0           | CT=1のとき          | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | CT=0のとき          | 14     | 14         | 14         | 14         | 6          |            |
| BNE      | short-label | 0           | Z=1のとき           | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | Z=0のとき           | 14     | 14         | 14         | 14         | 6          |            |
| BNH      | short-label | 0           | CVZ=1のとき         | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | CVZ=0のとき         | 4      | 4          | 4          | 4          | 3          |            |
| BNL      | short-label | 0           | CT=1のとき          | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | CT=0のとき          | 14     | 14         | 14         | 14         | 6          |            |
| BNV      | short-label | 0           | V=1のとき           | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | V=0のとき           | 14     | 14         | 14         | 14         | 6          |            |
| BNZ      | short-label | 0           | Z=1のとき           | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | Z=0のとき           | 14     | 14         | 14         | 14         | 6          |            |
| BP       | short-label | 0           | S=1のとき           | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | S=0のとき           | 14     | 14         | 14         | 14         | 6          |            |
| BPE      | short-label | 0           | P=1のとき           | 14     | 14         | 14         | 14         | 6          |            |
|          |             |             | P=0のとき           | 4      | 4          | 4          | 4          | 3          |            |
| BPO      | short-label | 0           | P=1のとき           | 4      | 4          | 4          | 4          | 3          |            |
|          |             |             | P=0のとき           | 14     | 14         | 14         | 14         | 6          |            |
| BR       | near-label  | 0           | —                | 13     | 13         | 13         | 13         | 7          |            |
|          | short-label | 0           | —                | 12     | 12         | 12         | 12         | 7          |            |
|          | regptr16    | 0           | —                | 11     | 11         | 11         | 11         | 7          |            |
|          | memptr16    | 1           | —                | 書 数    | 24         | 24         | 23         | 23         | 13         |
|          |             |             |                  | 換 数    | —          | 20         | —          | 19         | 11         |
|          | far-label   | 0           | —                | 15     | 15         | 15         | 15         | 7          |            |
| memptr32 | 2           | —           | 書 数              | 35     | 35         | 34         | 34         | 17         |            |
|          |             |             | 換 数              | —      | 27         | —          | 26         | 13         |            |

表 2-8 命令実行クロック数一覧 (3/15)

| ニモニック    | オペランド                                       | ワード<br>長さ/ビット | 条件 |        | クロック数      |            |            |            |            |
|----------|---------------------------------------------|---------------|----|--------|------------|------------|------------|------------|------------|
|          |                                             |               | W  | アドレス   | V20, V20HL | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V33A |
| BRK      | imm8 (*3)                                   | 5             | -  | 奇数     | 50         | 50         | 50         | 50         | 24         |
|          |                                             |               |    | 偶数     | 38         | 38         | 38         | 18         |            |
| BRKEM    | imm8                                        | 5             | -  | 奇数     | 50         | 50         | 50         | 50         | -          |
|          |                                             |               |    | 偶数     | 38         | 38         | 38         | -          |            |
| BRKV     | なし (V=1のとき)                                 | 5             | -  | 奇数     | 52         | 52         | 52         | 52         | 26         |
|          |                                             |               |    | 偶数     | 40         | 40         | 40         | 20         |            |
| BRKV     | なし (V=0のとき)                                 | 5             | -  | 奇数     | 3          | 3          | 3          | 3          | 3          |
|          |                                             |               |    | 偶数     | 3          | 3          | 3          | 3          |            |
| BRKLA    | imm8                                        | 2             | -  | -      | -          | -          | -          | 12         |            |
| BRKLOCK  | なし                                          | 0             | -  | 2      | 2          | 2          | 2          | 2          |            |
| BV       | short-label                                 | 0             | -  | V=1のとき | 14         | 14         | 14         | 14         | 6          |
|          |                                             |               |    | V=0のとき | 4          | 4          | 4          | 4          | 3          |
| EZ       | short-label                                 | 0             | -  | Z=1のとき | 14         | 14         | 14         | 14         | 6          |
|          |                                             |               |    | Z=0のとき | 4          | 4          | 4          | 4          | 3          |
| CALL     | near-proc                                   | 1             | -  | 奇数     | 20         | 20         | 20         | 20         | 9          |
|          |                                             |               |    | 偶数     | 16         | 16         | 16         | 7          |            |
|          | regptr16                                    | 1             | -  | 奇数     | 18         | 18         | 18         | 18         | 9          |
|          |                                             |               |    | 偶数     | 14         | 14         | 14         | 7          |            |
|          | memptr16                                    | 2             | -  | 奇数     | 31         | 31         | 31         | 31         | 15         |
|          |                                             |               |    | 偶数     | 23         | 23         | 23         | 11         |            |
| far-proc | 2                                           | -             | 奇数 | 29     | 29         | 29         | 29         | 13         |            |
|          |                                             |               | 偶数 | 21     | 21         | 21         | 9          |            |            |
| memptr32 | 4                                           | -             | 奇数 | 47     | 47         | 47         | 47         | 23         |            |
|          |                                             |               | 偶数 | 31     | 31         | 31         | 15         |            |            |
| CALLN    | imm8                                        | 5             | -  | 奇数     | 58         | 58         | 58         | 58         | -          |
|          |                                             |               |    | 偶数     | 36         | 36         | 36         | -          |            |
| CHKIND   | reg16, mem32 <sup>注</sup><br>(割り込み条件成立のとき)  | 7             | -  | 奇数     | 73-76      | 73-76      | 72-75      | 72-75      | 30-32      |
|          |                                             |               |    | 偶数     | 53-56      | 53-56      | 52-55      | 24-26      |            |
| CHKIND   | reg16, mem32 <sup>注</sup><br>(割り込み条件不成立のとき) | 2             | -  | 奇数     | 26         | 26         | 25         | 25         | 14         |
|          |                                             |               |    | 偶数     | 18         | 18         | 17         | 17         |            |
| CLR1     | reg8, CL                                    | 0             | -  | 5      | 5          | 5          | 5          | 4          |            |
|          | mem8, CL                                    | 0             | -  | 14     | 14         | 11         | 11         | 9          |            |
|          | reg16, CL                                   | 0             | -  | 5      | 5          | 5          | 5          | 4          |            |
|          | mem16, CL                                   | 2             | -  | 奇数     | 22         | 22         | 19         | 19         | 13         |
|          |                                             |               |    | 偶数     | 14         | 14         | 11         | 9          |            |
|          | reg8, imm3                                  | 0             | -  | 6      | 6          | 6          | 6          | 4          |            |
|          | mem8, imm3                                  | 0             | -  | 18     | 18         | 12         | 12         | 9          |            |
|          | reg16, imm4                                 | 0             | -  | 6      | 6          | 6          | 6          | 4          |            |
|          | mem16, imm4                                 | 2             | -  | 奇数     | 23         | 23         | 20         | 20         | 13         |
|          |                                             |               |    | 偶数     | 15         | 15         | 12         | 9          |            |
| CY       | 0                                           | -             | 2  | 2      | 2          | 2          | 2          |            |            |
| DIR      | 0                                           | -             | 2  | 2      | 2          | 2          | 2          |            |            |

注: クロック数は割り込み要求を受け付けるタイミングにより異なります。



表 2-8 命令実行クロック数一覧 (4/16)

| * ニモニック  | オペランド                                             | ワード<br>転送回数                  | 条 件            |                            | ク ロ ッ ク 数                                                                                             |                                                                                                       |                                                                                                       |                                                                                                       |                                                                                                       |
|----------|---------------------------------------------------|------------------------------|----------------|----------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|          |                                                   |                              | W              | アドレス                       | V20, V20HL                                                                                            | V30, V30HL                                                                                            | V40, V40HL                                                                                            | V50, V50HL                                                                                            | V33A, V53A                                                                                            |
| CMP      | reg, reg'                                         | 0                            | —              | 2                          | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     |
|          | mem, reg                                          | 1                            | 0              | —                          | 11                                                                                                    | 11                                                                                                    | 10                                                                                                    | 10                                                                                                    | 6                                                                                                     |
|          |                                                   |                              | 1              | 奇数<br>偶数                   | 15<br>11                                                                                              | 15<br>11                                                                                              | 14<br>14                                                                                              | 14<br>10                                                                                              | 8<br>6                                                                                                |
|          | reg, mem                                          | 1                            | 0              | —                          | 11                                                                                                    | 11                                                                                                    | 10                                                                                                    | 10                                                                                                    | 6                                                                                                     |
|          |                                                   |                              | 1              | 奇数<br>偶数                   | 15<br>11                                                                                              | 15<br>11                                                                                              | 14<br>14                                                                                              | 14<br>10                                                                                              | 8<br>6                                                                                                |
|          | reg, imm                                          | 0                            | —              | 4                          | 4                                                                                                     | 4                                                                                                     | 4                                                                                                     | 2                                                                                                     |                                                                                                       |
|          | mem, imm                                          | 1                            | 0              | —                          | 13                                                                                                    | 13                                                                                                    | 12                                                                                                    | 12                                                                                                    | 6                                                                                                     |
|          |                                                   |                              | 1              | 奇数<br>偶数                   | 17<br>13                                                                                              | 17<br>13                                                                                              | 16<br>16                                                                                              | 16<br>12                                                                                              | 8<br>6                                                                                                |
| acc, imm | 0                                                 | —                            | 4              | 4                          | 4                                                                                                     | 4                                                                                                     | 2                                                                                                     |                                                                                                       |                                                                                                       |
| CMP#1    | [DS1-spec:] dst-string,<br>[Seg-spec:] src-rtitag | 0                            | —              | $19 \times m + 7$          | $19 \times m + 7$                                                                                     | $19 \times m + 7$                                                                                     | $19 \times m + 7$                                                                                     | $14 \times m + 2$                                                                                     |                                                                                                       |
|          | なし                                                | 0                            | —              | $19 \times m + 7$          | $19 \times m + 7$                                                                                     | $19 \times m + 7$                                                                                     | $19 \times m + 7$                                                                                     | $14 \times m + 2$                                                                                     |                                                                                                       |
| CMP#2    | [Seg-spec:] src-block,<br>[DS1-spec:] dst-block   | $2 \times \text{rep}$<br>(2) | 0              | —                          | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $12 \times \text{rep} - 1$ (11)                                                                       |
|          |                                                   |                              | 1              | 奇数, 奇数<br>奇数, 偶数<br>偶数, 偶数 | $7 + 22 \times \text{rep}$ (21)<br>$7 + 18 \times \text{rep}$ (17)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 22 \times \text{rep}$ (21)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 22 \times \text{rep}$ (21)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 18 \times \text{rep}$ (17)<br>$7 + 14 \times \text{rep}$ (13) | $16 \times \text{rep} - 1$ (15)<br>$14 \times \text{rep} - 1$ (13)<br>$12 \times \text{rep} - 1$ (11) |
| CMP#3    | なし                                                | $2 \times \text{rep}$ (2)    | 0              | —                          | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $12 \times \text{rep} - 1$ (11)                                                                       |
| CMP#4    | なし                                                | $2 \times \text{rep}$<br>(2) | 1              | 奇数, 奇数<br>奇数, 偶数<br>偶数, 偶数 | $7 + 22 \times \text{rep}$ (21)<br>$7 + 18 \times \text{rep}$ (17)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 22 \times \text{rep}$ (21)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 22 \times \text{rep}$ (21)<br>$7 + 14 \times \text{rep}$ (13) | $7 + 22 \times \text{rep}$ (21)<br>$7 + 18 \times \text{rep}$ (17)<br>$7 + 14 \times \text{rep}$ (13) | $16 \times \text{rep} - 1$ (15)<br>$14 \times \text{rep} - 1$ (13)<br>$12 \times \text{rep} - 1$ (11) |
|          |                                                   |                              | 0              | —                          | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $7 + 14 \times \text{rep}$ (13)                                                                       | $12 \times \text{rep} - 1$ (11)                                                                       |
| CMP#5    | [DS1-spec:] dst-block                             | $1 \times \text{rep}$<br>(1) | 0              | —                          | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $10 \times \text{rep} - 1$ (9)                                                                        |
|          |                                                   |                              | 1              | 奇数<br>偶数                   | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $12 \times \text{rep} - 1$ (11)<br>$10 \times \text{rep} - 1$ (9)                                     |
| CMP#6    | なし                                                | $1 \times \text{rep}$<br>(1) | 0              | —                          | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $7 + 10 \times \text{rep}$ (7)                                                                        | $10 \times \text{rep} - 1$ (9)                                                                        |
| CMP#7    | なし                                                | $1 \times \text{rep}$<br>(1) | 1              | 奇数<br>偶数                   | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $7 + 14 \times \text{rep}$ (11)<br>$7 + 10 \times \text{rep}$ (7)                                     | $12 \times \text{rep} - 1$ (11)<br>$10 \times \text{rep} - 1$ (9)                                     |
| CVTBD    | なし                                                | 0                            | —              | 15                         | 15                                                                                                    | 15                                                                                                    | 15                                                                                                    | 12                                                                                                    |                                                                                                       |
| CVTBW    | なし                                                | 0                            | —              | 2                          | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     |                                                                                                       |
| CVTDB    | なし                                                | 0                            | —              | 7                          | 2                                                                                                     | 2                                                                                                     | 2                                                                                                     | 8                                                                                                     |                                                                                                       |
| CVTBL    | なし                                                | 0                            | —              | 4, 9                       | 4, 5                                                                                                  | 4, 5                                                                                                  | 4, 5                                                                                                  | 2                                                                                                     |                                                                                                       |
| DENZ     | short-label                                       | 0                            | CW≠0の時         | 13                         | 13                                                                                                    | 13                                                                                                    | 13                                                                                                    | 6                                                                                                     |                                                                                                       |
|          |                                                   |                              | CW=0の時         | 5                          | 5                                                                                                     | 5                                                                                                     | 5                                                                                                     | 3                                                                                                     |                                                                                                       |
| DENZE    | short-label                                       | 0                            | CW≠0で<br>Z=1の時 | 14                         | 14                                                                                                    | 14                                                                                                    | 14                                                                                                    | 6                                                                                                     |                                                                                                       |
|          |                                                   |                              | 上記以外の時         | 5                          | 5                                                                                                     | 5                                                                                                     | 5                                                                                                     | 3                                                                                                     |                                                                                                       |
| DENZNE   | short-label                                       | 0                            | CW≠0で<br>Z=0の時 | 14                         | 14                                                                                                    | 14                                                                                                    | 14                                                                                                    | 6                                                                                                     |                                                                                                       |
|          |                                                   |                              | 上記以外の時         | 5                          | 5                                                                                                     | 5                                                                                                     | 5                                                                                                     | 3                                                                                                     |                                                                                                       |

注1. m: BCD桁数×1/2

2. ( ): 一回だけの処理に適用

3. クロック数はデータの値により異なります (V33A, V53Aを除く)。

表 2-8 命令実行クロック数一覧 (8/18)

| エモニック             | オペランド            | ワード<br>伝送回数 | 条件 |       | クロック数            |            |            |            |            |    |
|-------------------|------------------|-------------|----|-------|------------------|------------|------------|------------|------------|----|
|                   |                  |             | W  | アドレス  | V20, V20HL       | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V53A |    |
| DEC               | reg8             | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
|                   | mem              | 2           | 0  | —     | 16               | 16         | 13         | 13         | 7          |    |
|                   |                  |             | 1  | 奇数    | 24               | 24         | 21         | 21         | 11         |    |
|                   |                  |             |    | 偶数    | 16               | 16         | 13         | 13         | 7          |    |
| reg16             | 0                | —           | —  | 2     | 2                | 2          | 2          | 2          |            |    |
| DI                | なし               | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
| DISPOSE           | なし               | 1           | —  | 奇数    | 10               | 10         | 10         | 10         | 8          |    |
|                   |                  |             |    | 偶数    | 6                | 6          | 6          | 6          |            |    |
| DIV <sup>注1</sup> | reg8             | 0           | —  | —     | 28-34            | 28-34      | 28-34      | 28-34      | 17         |    |
|                   | mem8             | 0           | —  | —     | 34-38            | 34-38      | 34-38      | 34-38      | 20         |    |
|                   | reg16            | 0           | —  | —     | 38-43            | 38-43      | 38-43      | 38-43      | 24         |    |
|                   | mem16            | 1           | —  | 奇数    | 47-52            | 47-52      | 47-52      | 47-52      | 30         |    |
| 偶数                |                  |             |    | 43-48 | 43-48            | 43-48      | 28         |            |            |    |
| DIVU              | reg8             | 0           | —  | —     | 19               | 19         | 19         | 19         | 11         |    |
|                   | mem8             | 0           | —  | —     | 25               | 25         | 24         | 24         | 18         |    |
|                   | reg16            | 0           | —  | —     | 25               | 25         | 25         | 25         | 19         |    |
|                   | mem16            | 1           | —  | 奇数    | 34               | 34         | 34         | 34         | 25         |    |
| 偶数                |                  |             |    | 30    | 30               | 30         | 23         |            |            |    |
| DS0:              | なし               | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
| DS1:              | なし               | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
| HI                | なし               | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
| LXT <sup>注2</sup> | reg8, reg8'      | 1または2       | —  | 奇数    | 34-59            | 34-59      | 34-59      | 34-59      | 33-63      |    |
|                   |                  |             |    | 偶数    | 26-58            | 26-58      | 26-58      | 29-61      |            |    |
|                   | reg8, mem4       | 1または2       | —  | 奇数    | 34-59            | 34-59      | 34-59      | 34-59      | 33-63      |    |
|                   |                  |             |    | 偶数    | 26-58            | 26-58      | 26-58      | 29-61      |            |    |
| FPC1              | fp-op            | 0           | —  | —     | 2                | 2          | 2          | 2          | 定数不可       |    |
|                   | fp-op, mem       | 1           | —  | 奇数    | 15               | 15         | 14         | 14         | 定数不可       |    |
| 偶数                |                  |             |    | 11    | 11               | 10         | 定数不可       |            |            |    |
| FPC2              | fp-op            | 0           | —  | —     | 2                | 2          | 2          | 2          | 定数不可       |    |
|                   | fp-op, mem       | 1           | —  | 奇数    | 15               | 15         | 14         | 14         | 定数不可       |    |
| 偶数                |                  |             |    | 11    | 11               | 10         | 定数不可       |            |            |    |
| HALT              | なし               | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
| IN                | acc, imm8        | 1           | 0  | —     | 9                | 9          | 9          | 9          | 5          |    |
|                   |                  |             |    | 1     | 奇数               | 13         | 13         | 13         | 13         | 7  |
|                   |                  |             |    |       | 偶数 <sup>注3</sup> | 9          | 9          | 9          | 5          |    |
|                   | acc, DW          | 1           | 0  | —     | 8                | 8          | 8          | 8          | 5          |    |
| 1                 |                  |             |    | 奇数    | 12               | 12         | 12         | 12         | 7          |    |
|                   | 偶数 <sup>注3</sup> | 8           | 8  | 8     | 5                |            |            |            |            |    |
| INC               | reg8             | 0           | —  | —     | 2                | 2          | 2          | 2          | 2          |    |
|                   | mem              | 2           | 0  | —     | 16               | 16         | 13         | 13         | 7          |    |
|                   |                  |             |    | 1     | 奇数               | 24         | 24         | 21         | 21         | 11 |
|                   |                  |             |    |       | 偶数               | 16         | 16         | 13         | 13         | 7  |
| reg16             | 0                | —           | —  | 2     | 2                | 2          | 2          | 2          |            |    |

- 注1. クロック数はデータの値により異なります (V33A, V53Aを除く)。  
 2. クロック数はデータの値により異なります。  
 3. V50, V50HL, V53Aは、内蔵DMAUへのワード・アクセスの場合バス・サイクルを2回起動するため、奇数アドレスの実行クロック数と同じ値になります。

表 2-8 命令実行クロック数一覧 (6/15)

| ニモニック                | オペランド                    | ワード<br>転送回数  | 条 件 |      | ク ロ ッ ク 数     |               |               |               |               |             |
|----------------------|--------------------------|--------------|-----|------|---------------|---------------|---------------|---------------|---------------|-------------|
|                      |                          |              | W   | アドレス | V20, V20HL    | V30, V30HL    | V40, V40HL    | V50, V50HL    | V33A, V33A    |             |
| INM <sup>2,3,1</sup> | [DSI-spec:] dt-block, DW | 2×rep<br>(2) | 0   | —    | 8+8×rep (10)  | 8+8×rep (10)  | 8+8×rep (10)  | 8+8×rep (10)  | 8+8×rep (10)  | 地 3         |
|                      |                          |              |     | 1    | 奇数, 奇数        | 9+16×rep (18) | 9+16×rep (18) | 9+16×rep (18) | 9+16×rep (18) |             |
|                      |                          |              |     |      | 奇数, 偶数        |               | 9+12×rep (14) |               | 9+12×rep (14) |             |
|                      |                          |              |     |      | 偶数, 偶数        |               | 9+8×rep (10)  |               | 9+8×rep (10)  |             |
| INS <sup>2,3,1</sup> | reg8, reg8'              | 2×tN4        | —   | 奇数   | 35-133        | 35-133        | 35-133        | 35-133        | 35-77         |             |
|                      |                          |              |     | 偶数   |               | 31-117        |               | 31-117        | 37-59         |             |
|                      |                          |              |     | 偶数   |               | 31-117        |               | 31-117        | 37-59         |             |
|                      | reg8, imm4               | 2×tN4        | —   | 奇数   | 35-133        | 35-133        | 35-133        | 35-133        | 35-77         |             |
|                      |                          |              |     | 偶数   |               | 31-117        |               | 31-117        | 37-59         |             |
|                      |                          |              |     | 偶数   |               | 31-117        |               | 31-117        | 37-59         |             |
| LDEA                 | reg16, mem16             | 0            | —   | 4    | 4             | 4             | 4             | 2             |               |             |
| LDM <sup>2,1</sup>   | [Seq-spec:] src-block    | 1×rep<br>(1) | 0   | —    | 7+9×rep (7)   | 7+9×rep (7)   | 7+9×rep (7)   | 7+9×rep (7)   | 2+3×rep (5)   |             |
|                      |                          |              |     | 1    | 奇数            | 7+13×rep (11) | 7+13×rep (11) | 7+13×rep (11) | 7+13×rep (11) | 2+5×rep (7) |
|                      |                          |              |     |      | 偶数            |               | 7+9×rep (7)   |               | 7+9×rep (7)   | 2+3×rep (5) |
| LDM <sup>2,1</sup>   | なし                       | 1×rep(1)     | 0   | —    | 7+9×rep (7)   | 7+9×rep (7)   | 7+9×rep (7)   | 2+3×rep (5)   |               |             |
| LDM <sup>2,1</sup>   | なし                       | 1×rep<br>(1) | 1   | 奇数   | 7+13×rep (11) | 7+13×rep (11) | 7+13×rep (11) | 7+13×rep (11) | 2+5×rep (7)   |             |
|                      |                          |              |     | 偶数   |               | 7+9×rep (7)   |               | 7+9×rep (7)   | 2+3×rep (5)   |             |

- 注1. ( ): 一度だけの処理に適用
2. クロック数はデータの値により異なります。
3. V33A, V53Aのクロック数は次のとおりです。

| ニモニック | オペランド                    | ワード<br>転送回数  | 条 件    |              | ク ロ ッ ク 数    |                                  |                                 |
|-------|--------------------------|--------------|--------|--------------|--------------|----------------------------------|---------------------------------|
|       |                          |              | W      | アドレス         | V33A         | V53A                             |                                 |
| INM   | [DSI-spec:] dt-block, DW | 2×rep<br>(2) | 0      | —            | 4+8×rep (12) | 8×rep (8)                        |                                 |
|       |                          |              |        | 1            | 奇数, 奇数       | 8+14×rep (14)                    | 14×rep (14)                     |
|       |                          |              |        |              | 奇数, 偶数       | I/Oアドレス<br>が奇数:<br>8+8×rep(20)   | I/Oアドレス<br>が奇数:<br>12×rep (12)  |
|       |                          |              |        |              | 偶数, 偶数       | メモリ・アド<br>レスが奇数:<br>4+10×rep(14) | メモリ・アド<br>レスが奇数:<br>10×rep (10) |
|       |                          |              | 偶数, 偶数 | 4+8×rep (12) | 8×rep (8)    |                                  |                                 |

表 2-8 命令実行クロック数一覧 (7/18)

| ★<br>ニモニック         | オペランド                 | ワード<br>毎処理数   | 条件 |             | クロック数          |                |               |               |             |  |
|--------------------|-----------------------|---------------|----|-------------|----------------|----------------|---------------|---------------|-------------|--|
|                    |                       |               | W  | アドレス        | V20, V20HL     | V30, V30HL     | V40, V40HL    | V50, V50HL    | V33A, V33A  |  |
| MOV                | reg, reg              | 0             | —  | —           | 2              | 2              | 2             | 2             | 2           |  |
|                    | mem, reg              | 1             | 0  | —           | 9              | 9              | 7             | 7             | 3           |  |
|                    |                       |               | 1  | 奇数          | 13             | 13             | 11            | 11            | 5           |  |
|                    |                       |               |    | 偶数          | —              | 9              | —             | 7             | 3           |  |
|                    | reg, mem              | 1             | 0  | —           | 11             | 11             | 10            | 10            | 5           |  |
|                    |                       |               | 1  | 奇数          | 15             | 15             | 14            | 14            | 7           |  |
|                    |                       |               |    | 偶数          | —              | 11             | —             | 10            | 5           |  |
|                    | mem, mem              | 1             | 0  | —           | 11             | 11             | 9             | 9             | 3           |  |
|                    |                       |               | 1  | 奇数          | 15             | 15             | 13            | 13            | 5           |  |
|                    |                       |               |    | 偶数          | —              | 11             | —             | 9             | 3           |  |
|                    | reg, mem              | 0             | —  | —           | 4              | 4              | 4             | 4             | 2           |  |
|                    | mem, mem              | 1             | 0  | —           | 10             | 10             | 10            | 10            | 5           |  |
|                    |                       |               | 1  | 奇数          | 14             | 14             | 14            | 14            | 7           |  |
|                    |                       |               |    | 偶数          | —              | 10             | —             | 10            | 5           |  |
|                    | mem, mem              | 1             | 0  | —           | 9              | 9              | 9             | 9             | 3           |  |
|                    | mem, mem              | 1             | 1  | 奇数          | 13             | 13             | 13            | 13            | 5           |  |
|                    |                       |               |    | 偶数          | —              | 9              | —             | 9             | 3           |  |
|                    |                       |               |    | 偶数          | —              | 9              | —             | 9             | 3           |  |
|                    | reg, reg16            | 0             | —  | —           | 2              | 2              | 2             | 2             | 2           |  |
|                    | reg, mem16            | 1             | —  | 奇数          | 15             | 15             | 14            | 14            | 7           |  |
| 偶数                 |                       |               |    | —           | 11             | —              | 10            | 5             |             |  |
| reg16, mem         | 0                     | —             | —  | 2           | 2              | 2              | 2             | 2             |             |  |
| mem16, reg         | 1                     | —             | 奇数 | 14          | 14             | 12             | 12            | 5             |             |  |
|                    |                       |               | 偶数 | —           | 10             | —              | 8             | 3             |             |  |
| DS0, reg16, mem32  | 2                     | —             | 奇数 | 26          | 26             | 26             | 25            | 14            |             |  |
|                    |                       |               | 偶数 | —           | 18             | —              | 17            | 10            |             |  |
| DS1, reg16, mem32  | 2                     | —             | 奇数 | 26          | 26             | 26             | 26            | 14            |             |  |
|                    |                       |               | 偶数 | —           | 18             | —              | 17            | 10            |             |  |
| AH, PSW            | 0                     | —             | —  | 2           | 2              | 2              | 2             | 2             |             |  |
| PSW, AH            | 0                     | —             | —  | 3           | 3              | 3              | 3             | 2             |             |  |
| MOV <sup>REP</sup> | [DS1-spec:] dst-block | 2×rep         | 0  | —           | 11+8×rep (11)  | 11+8×rep (11)  | 9+8×rep (9)   | 9+8×rep (9)   | 6×rep (6)   |  |
|                    | [Seq-spec:] src-block | (2)           | 1  | 奇数, 奇数      | 11+16×rep (19) | 11+16×rep (19) | 9+16×rep (17) | 9+16×rep (17) | 10×rep (10) |  |
|                    |                       |               |    | 奇数, 偶数      | —              | 11+12×rep (15) | —             | 9+12×rep (13) | 8×rep (8)   |  |
| 偶数, 偶数             | —                     | 11+8×rep (11) | —  | 9+8×rep (9) | 8×rep (8)      |                |               |               |             |  |
| MOV <sup>REP</sup> | なし                    | 2×rep (2)     | 0  | —           | 11+8×rep (11)  | 11+8×rep (11)  | 9+8×rep (9)   | 9+8×rep (9)   | 6×rep (6)   |  |
| MOV <sup>REP</sup> | なし                    | 2×rep (2)     | 1  | 奇数, 奇数      | 11+16×rep (19) | 11+16×rep (19) | 9+16×rep (17) | 9+16×rep (17) | 10×rep (10) |  |
|                    |                       |               |    | 奇数, 偶数      | —              | 11+12×rep (15) | —             | 9+12×rep (13) | 8×rep (8)   |  |
|                    |                       |               |    | 偶数, 偶数      | —              | 11+8×rep (11)  | —             | 9+8×rep (9)   | 8×rep (8)   |  |

注 ( ) : 一回だけの処理に適用

表 2-8 命令実行クロック数一覧 (8/16)

| ★<br>二重ニック           | オペランド               | W-F<br>送込回数 | 条 件 | ク ロ ッ ク 数 |            |            |            |            |            |
|----------------------|---------------------|-------------|-----|-----------|------------|------------|------------|------------|------------|
|                      |                     |             |     | W<br>アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V33A |
| MUL <sup>注</sup>     | reg8                | 0           | —   | —         | 33-39      | 33-39      | 33-39      | 33-39      | 8          |
|                      | mem8                | 0           | —   | —         | 39-45      | 39-45      | 39-44      | 39-44      | 12         |
|                      | reg16               | 0           | —   | —         | 41-47      | 41-47      | 41-47      | 41-47      | 12         |
|                      | mem16               | 1           | —   | 奇数        | 51-57      | 51-57      | 50-56      | 50-56      | 18         |
|                      |                     |             |     | 偶数        | —          | 47-53      | —          | 46-52      | 16         |
|                      | reg16, imm8         | 0           | —   | —         | 29-34      | 29-34      | 28-34      | 28-34      | 12         |
|                      | reg16, imm16        | 0           | —   | —         | 36-42      | 36-42      | 36-42      | 36-42      | 12         |
|                      | reg16, reg16', imm8 | 0           | —   | —         | 29-34      | 29-34      | 28-34      | 28-34      | 12         |
|                      | reg16, mem16, imm8  | 1           | —   | 奇数        | 36-44      | 36-44      | 37-43      | 37-43      | 18         |
|                      |                     |             |     | 偶数        | —          | 34-40      | —          | 33-39      | 16         |
| reg16, reg16', imm16 | 0                   | —           | —   | 36-42     | 36-42      | 36-42      | 36-42      | 12         |            |
| reg16, mem16, imm16  | 1                   | —           | 奇数  | 46-52     | 46-52      | 46-51      | 45-51      | 18         |            |
|                      |                     |             | 偶数  | —         | 42-48      | —          | 41-47      | 16         |            |
| MUL <sup>注</sup>     | reg8                | 0           | —   | —         | 21, 22     | 21, 22     | 21, 22     | 21, 22     | 8          |
|                      | mem8                | 1           | —   | —         | 27, 28     | 27, 28     | 26, 27     | 26, 27     | 12         |
|                      | reg16               | 0           | —   | —         | 29, 30     | 29, 30     | 28, 30     | 28, 30     | 12         |
|                      | mem16               | 1           | —   | 奇数        | 39, 40     | 39, 40     | 38, 39     | 38, 39     | 16         |
| 偶数                   |                     |             |     | —         | 35, 36     | —          | 34, 35     | 16         |            |
| NEG                  | reg                 | 0           | —   | —         | 2          | 2          | 2          | 2          | 2          |
|                      | mem                 | 2           | 0   | —         | 16         | 16         | 13         | 13         | 7          |
|                      |                     |             |     | 1         | 奇数         | 24         | 24         | 21         | 21         |
| 偶数                   | —                   | 16          | —   | 13        | 7          |            |            |            |            |
| NOP                  | なし                  | 0           | —   | —         | 3          | 3          | 3          | 3          | 3          |
| NOT                  | reg                 | 0           | —   | —         | 2          | 2          | 2          | 2          | 2          |
|                      | mem                 | 2           | 0   | —         | 16         | 16         | 13         | 13         | 7          |
|                      |                     |             |     | 1         | 奇数         | 24         | 24         | 21         | 21         |
| 偶数                   | —                   | 16          | —   | 13        | 7          |            |            |            |            |
| NOT1                 | reg8, CL            | 0           | —   | —         | 4          | 4          | 4          | 4          | 4          |
|                      | mem8, CL            | 0           | —   | —         | 13         | 13         | 10         | 10         | 9          |
|                      | reg16, CL           | 0           | —   | —         | 4          | 4          | 4          | 4          | 4          |
|                      | mem16, CL           | 2           | —   | 奇数        | 21         | 21         | 19         | 19         | 13         |
|                      |                     |             |     | 偶数        | —          | 13         | —          | 10         | 9          |
|                      | reg8, imm3          | 0           | —   | —         | 5          | 5          | 5          | 5          | 4          |
|                      | mem8, imm3          | 0           | —   | —         | 14         | 14         | 11         | 11         | 9          |
|                      | reg16, imm4         | 0           | —   | —         | 5          | 5          | 5          | 5          | 4          |
|                      | mem16, imm4         | 2           | —   | 奇数        | 22         | 22         | 19         | 19         | 13         |
|                      |                     |             |     | 偶数        | —          | 14         | —          | 11         | 9          |
| CY                   | 0                   | —           | —   | 2         | 2          | 2          | 2          | 2          |            |

注 クロック数はデータの値により異なります (V33A, V53Aを除く)。

表 2-8 命令実行クロック数一覧 (9/18)

| ★<br>ニモニック          | オペランド                     | ワード<br>転送回数  | 条 件 |                            | ク ロ ッ ク 数     |                                                |               |                                                |            |
|---------------------|---------------------------|--------------|-----|----------------------------|---------------|------------------------------------------------|---------------|------------------------------------------------|------------|
|                     |                           |              | W   | アドレス                       | V20, V20HL    | V30, V30HL                                     | V40, V40HL    | V50, V50HL                                     | V33A, V33A |
| OR                  | reg, reg'                 | 0            | —   | —                          | 2             | 2                                              | 2             | 2                                              | 2          |
|                     | mem, reg                  | 2            | 0   | —                          | 16            | 16                                             | 13            | 13                                             | 7          |
|                     |                           |              | 1   | 奇数<br>偶数                   | 24<br>16      | 24<br>16                                       | 21<br>13      | 21<br>13                                       | 11<br>7    |
|                     | reg, mem                  | 1            | 0   | —                          | 11            | 11                                             | 10            | 10                                             | 6          |
|                     |                           |              | 1   | 奇数<br>偶数                   | 15<br>11      | 15<br>11                                       | 14<br>10      | 14<br>10                                       | 8<br>6     |
|                     | reg, imm                  | 0            | —   | —                          | 4             | 4                                              | 4             | 4                                              | 2          |
|                     | mem, imm                  | 2            | 0   | —                          | 18            | 18                                             | 15            | 15                                             | 7          |
|                     |                           |              | 1   | 奇数<br>偶数                   | 25<br>18      | 25<br>18                                       | 23<br>15      | 23<br>15                                       | 11<br>7    |
| acc, imm            | 0                         | —            | —   | 4                          | 4             | 4                                              | 4             | 2                                              |            |
| OUT                 | imm8, acc                 | 1            | 0   | —                          | 8             | 8                                              | 8             | 8                                              | 3          |
|                     |                           |              | 1   | 奇数<br>偶数                   | 12<br>8       | 12<br>8                                        | 12<br>8       | 12<br>8                                        | 5<br>3     |
|                     | DW, acc                   | 1            | 0   | —                          | 8             | 8                                              | 8             | 8                                              | 3          |
|                     |                           |              | 1   | 奇数<br>偶数                   | 12<br>8       | 12<br>8                                        | 12<br>8       | 12<br>8                                        | 5<br>3     |
| OUTM <sup>(1)</sup> | DW, [Seq-spec:] src-block | 2×rep<br>(2) | 0   | —                          | 9+8×rep (10)  | 9+8×rep (10)                                   | 9+8×rep (10)  | 9+8×rep (10)                                   | 2×4        |
|                     |                           |              | 1   | 奇数, 奇数<br>奇数, 偶数<br>偶数, 偶数 | 9+16×rep (18) | 9+16×rep (18)<br>9+12×rep (14)<br>9+8×rep (10) | 9+16×rep (18) | 9+16×rep (18)<br>9+12×rep (14)<br>9+8×rep (10) |            |
| POLL <sup>(2)</sup> | なし                        | 0            | —   | —                          | 2+5×poll      | 2+5×poll                                       | 2+5×poll      | 2+5×poll                                       | 2+2×cpbusy |

注1. ( ) : 一回だけの処理に適用

2. poll : POLL端子サンプリング回数, cpbusy : CPBUSY端子サンプリング回数
3. V50, V50HL, V53Aは、内蔵DMAUへのワード・アクセスの場合バス・サイクルを2回起動するため、奇数アドレスの実行クロック数と同じ値になります。
4. V33A, V53Aのクロック数は次のとおりです。

| ニモニック  | オペランド                     | ワード<br>転送回数  | 条 件 |        | ク ロ ッ ク 数                         |                                   |
|--------|---------------------------|--------------|-----|--------|-----------------------------------|-----------------------------------|
|        |                           |              | W   | アドレス   | V33A                              | V53A                              |
| OUTM   | DW, [Seq-spec:] src-block | 2×rep<br>(2) | 0   | —      | 12×rep-8 (6)                      | 8×rep-2 (6)                       |
|        |                           |              | 1   | 奇数, 奇数 | 22×rep-6 (16)                     | 14×rep-2 (12)                     |
|        |                           |              |     | 奇数, 偶数 | I/Oアドレス<br>が奇数:                   | I/Oアドレス<br>が奇数:                   |
|        |                           |              |     | 偶数, 奇数 | 20×rep-6 (10)<br>メモリ・アド<br>レスが奇数: | 12×rep-2 (10)<br>メモリ・アド<br>レスが奇数: |
|        |                           |              |     | 偶数, 偶数 | 14×rep-6 (8)                      | 10×rep-2 (8)                      |
| 偶数, 偶数 | 12×rep-6 (6)              | 8×rep-2 (6)  |     |        |                                   |                                   |

表 2-8 命令実行クロック数一覧 (10/15)

| ★<br>モニタ         | オペランド                    | ワード<br>数/回数 | 条件<br>W アドレス | クロック数      |              |              |              |              |              |
|------------------|--------------------------|-------------|--------------|------------|--------------|--------------|--------------|--------------|--------------|
|                  |                          |             |              | V23, V23HL | V30, V30HL   | V40, V40HL   | V50, V50HL   | V33A, V33A   |              |
| POP              | mem16                    | 2           | —            | 奇数         | 25           | 25           | 24           | 24           | 9            |
|                  |                          |             |              | 偶数         |              | 17           |              | 16           | 5            |
|                  | reg16                    | 1           | —            | 奇数         | 12           | 12           | 12           | 12           | 7            |
|                  |                          |             |              | 偶数         |              | 8            |              | 8            | 5            |
|                  | wreg                     | 1           | —            | 奇数         | 12           | 12           | 12           | 12           | 7            |
|                  |                          |             |              | 偶数         |              | 8            |              | 8            | 5            |
| PSW              | 1                        | —           | 奇数           | 12         | 12           | 12           | 12           | 7            |              |
|                  |                          |             | 偶数           |            | 8            |              | 8            | 5            |              |
| R                | 7                        | —           | 奇数           | 78         | 75           | 75           | 75           | 38           |              |
|                  |                          |             | 偶数           |            | 49           |              | 49           | 22           |              |
| PREPARE          | mem16, mem8 (mem≠0の時)    | 1           | —            | 奇数         | 16           | 16           | 16           | 16           | 15           |
|                  |                          |             |              | 偶数         |              | 12           |              | 12           |              |
|                  | mem16, mem8 (mem≠1の時)    | 2×mem8      | —            | 奇数         | 23+16(mem-1) | 23+16(mem-1) | 21+16(mem-1) | 21+16(mem-1) | 17+12(mem-1) |
|                  |                          |             |              | 偶数         |              | 19+8(mem-1)  |              | 17+8(mem-1)  | 15+8(mem-1)  |
| PS:              | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| PUSH             | mem16                    | 2           | —            | 奇数         | 26           | 26           | 23           | 23           | 9            |
|                  |                          |             |              | 偶数         |              | 18           |              | 18           | 5            |
|                  | reg16                    | 1           | —            | 奇数         | 12           | 12           | 10           | 10           | 5            |
|                  |                          |             |              | 偶数         |              | 8            |              | 6            | 3            |
|                  | wreg                     | 1           | —            | 奇数         | 12           | 12           | 10           | 10           | 5            |
|                  |                          |             |              | 偶数         |              | 8            |              | 6            | 3            |
|                  | PSW                      | 1           | —            | 奇数         | 12           | 12           | 10           | 10           | 5            |
|                  |                          |             |              | 偶数         |              | 8            |              | 6            | 3            |
|                  | R                        | 8           | —            | 奇数         | 67           | 67           | 66           | 65           | 36           |
|                  |                          |             |              | 偶数         |              | 36           |              | 33           | 20           |
| mem8             | 1                        | —           | 奇数           | 11         | 11           | 9            | 9            | 5            |              |
|                  |                          |             | 偶数           |            | 7            |              | 5            | 3            |              |
| mem16            | 1                        | —           | 奇数           | 12         | 12           | 10           | 10           | 5            |              |
|                  |                          |             | 偶数           |            | 8            |              | 6            | 3            |              |
| REP              | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| REPC             | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| REPE             | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| REPNC            | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| REPNE            | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| REPZ             | なし                       | 0           | —            |            | 2            | 2            | 2            | 2            | 2            |
| RET              | なし<br>(セグメント内コール)        | 1           | —            | 奇数         | 19           | 19           | 19           | 19           | 12           |
|                  |                          |             |              | 偶数         |              | 15           |              | 15           | 10           |
|                  | #<br>(セグメント外コール)         | 2           | —            | 奇数         | 29           | 29           | 29           | 29           | 16           |
|                  |                          |             |              | 偶数         |              | 21           |              | 21           | 12           |
|                  | pop+value<br>(セグメント内コール) | 1           | —            | 奇数         | 24           | 24           | 24           | 24           | 12           |
|                  |                          |             |              | 偶数         |              | 20           |              | 20           | 10           |
| #<br>(セグメント外コール) | 2                        | —           | 奇数           | 32         | 32           | 32           | 32           | 16           |              |
|                  |                          |             | 偶数           |            | 24           |              | 24           | 12           |              |
| RETM             | なし                       | 3           | —            | 奇数         | 39           | 39           | 39           | 39           | —            |
|                  |                          |             |              | 偶数         |              | 27           |              | 27           | —            |

表 2-8 命令実行クロック数一覧 (11/15)

| ★<br>ニモニック        | オペランド     | ワード<br>抽出回数 | 条 件 |      | ク ロ ッ ク 数  |            |            |            |            |      |
|-------------------|-----------|-------------|-----|------|------------|------------|------------|------------|------------|------|
|                   |           |             | W   | アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V80, V80HL | V32A, V32A |      |
| RETI              | なし        | 3           | —   | 奇数   | 30         | 30         | 30         | 30         | 19         |      |
|                   |           |             |     | 偶数   | —          | 27         | —          | —          | 13         |      |
| RETLA             | imm8      | 2           | —   | —    | —          | —          | —          | —          | 12         |      |
| ROCL <sup>2</sup> | reg, 1    | 0           | —   | —    | 6          | 6          | 6          | 6          | 2          |      |
|                   | mem, 1    | 2           | 0   | —    | 16         | 16         | 13         | 13         | 7          |      |
|                   |           |             |     | 1    | 奇数         | 24         | 24         | 21         | 21         | 11   |
|                   |           |             |     |      | 偶数         | 16         | —          | —          | 13         | 7    |
|                   | reg, CL   | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, CL   | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
|                   |           |             |     | 1    | 奇数         | 27+n       | 27+n       | 24+n       | 24+n       | 10+n |
|                   |           |             |     |      | 偶数         | 19+n       | —          | —          | 16+n       | 6+n  |
|                   | reg, imm8 | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, imm8 | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
| 1                 |           |             |     | 奇数   | 27+n       | 27+n       | 24+n       | 24+n       | 10+n       |      |
|                   |           |             |     | 偶数   | 19+n       | —          | —          | 16+n       | 6+n        |      |
| ROLA              | reg8      | 0           | —   | —    | 13         | 13         | 13         | 13         | 9          |      |
|                   | mem8      | 0           | —   | —    | 28         | 28         | 25         | 25         | 15         |      |
| ROLC <sup>2</sup> | reg, 1    | 0           | —   | —    | 6          | 6          | 6          | 6          | 2          |      |
|                   | mem, 1    | 2           | 0   | —    | 16         | 16         | 13         | 13         | 7          |      |
|                   |           |             |     | 1    | 奇数         | 24         | 24         | 21         | 21         | 11   |
|                   |           |             |     |      | 偶数         | 16         | —          | —          | 13         | 7    |
|                   | reg, CL   | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, CL   | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
|                   |           |             |     | 1    | 奇数         | 27+n       | 27+n       | 24+n       | 24+n       | 10+n |
|                   |           |             |     |      | 偶数         | 19+n       | —          | —          | 16+n       | 6+n  |
|                   | reg, imm8 | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, imm8 | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
| 1                 |           |             |     | 奇数   | 27+n       | 27+n       | 24+n       | 24+n       | 10+n       |      |
|                   |           |             |     | 偶数   | 19+n       | —          | —          | 16+n       | 6+n        |      |
| ROR <sup>2</sup>  | reg, 1    | 0           | —   | —    | 6          | 6          | 6          | 6          | 2          |      |
|                   | mem, 1    | 2           | 0   | —    | 16         | 16         | 13         | 13         | 7          |      |
|                   |           |             |     | 1    | 奇数         | 24         | 24         | 21         | 21         | 11   |
|                   |           |             |     |      | 偶数         | 16         | —          | —          | 13         | 7    |
|                   | reg, CL   | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, CL   | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
|                   |           |             |     | 1    | 奇数         | 27+n       | 27+n       | 24+n       | 24+n       | 10+n |
|                   |           |             |     |      | 偶数         | 19+n       | —          | —          | 16+n       | 6+n  |
|                   | reg, imm8 | 0           | —   | —    | 7+n        | 7+n        | 7+n        | 7+n        | 2+n        |      |
|                   | mem, imm8 | 2           | 0   | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |      |
| 1                 |           |             |     | 奇数   | 27+n       | 27+n       | 24+n       | 24+n       | 10+n       |      |
|                   |           |             |     | 偶数   | 19+n       | —          | —          | 16+n       | 6+n        |      |
| RORA              | reg8      | 0           | —   | —    | 17         | 17         | 17         | 17         | 13         |      |
|                   | mem8      | 0           | —   | —    | 32         | 32         | 29         | 29         | 19         |      |

注 n: シフト数



表 2-8 命令実行クロック数一覧 (12/15)

| ★ ニック             | オペランド       | ワード<br>無符号数 | 条件 |      | クロック数      |            |            |            |            |
|-------------------|-------------|-------------|----|------|------------|------------|------------|------------|------------|
|                   |             |             | W  | アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V50, V50HL | V33A, V33A |
| RORC <sup>※</sup> | reg. 1      | 0           | —  | 6    | 6          | 6          | 6          | 2          |            |
|                   | mem. 1      | 2           | 0  | —    | 16         | 16         | 13         | 13         | 7          |
|                   |             |             | 1  | 奇数   | 24         | 24         | 21         | 21         | 11         |
|                   |             |             |    | 偶数   | 16         | 16         | 13         | 7          |            |
|                   | reg. CL     | 0           | —  | 7+n  | 7+n        | 7+n        | 7+n        | 2+n        |            |
|                   | mem. CL     | 2           | 0  | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |
|                   |             |             | 1  | 奇数   | 27+n       | 27+n       | 24+n       | 24+n       | 10+n       |
|                   |             |             |    | 偶数   | 19+n       | 19+n       | 16+n       | 6+n        |            |
|                   | reg. imm8   | 0           | —  | 7+n  | 7+n        | 7+n        | 7+n        | 2+n        |            |
|                   | mem. imm8   | 2           | 0  | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |
| 1                 |             |             | 奇数 | 27+n | 27+n       | 24+n       | 24+n       | 10+n       |            |
|                   |             |             | 偶数 | 19+n | 19+n       | 16+n       | 6+n        |            |            |
| SETI              | reg. CL     | 0           | —  | 4    | 4          | 4          | 4          | 4          |            |
|                   | mem. CL     | 0           | —  | 13   | 13         | 10         | 10         | 9          |            |
|                   | reg16. CL   | 0           | —  | 4    | 4          | 4          | 4          | 4          |            |
|                   | mem16. CL   | 2           | —  | 奇数   | 21         | 21         | 18         | 18         | 13         |
|                   |             |             | 偶数 | 13   | 13         | 10         | 9          |            |            |
|                   | reg. imm3   | 0           | —  | 5    | 5          | 5          | 5          | 4          |            |
|                   | mem. imm3   | 0           | —  | 14   | 14         | 11         | 11         | 9          |            |
|                   | reg16. imm4 | 0           | —  | 5    | 5          | 5          | 5          | 4          |            |
|                   | mem16. imm4 | 2           | —  | 奇数   | 22         | 22         | 19         | 19         | 13         |
|                   |             |             | 偶数 | 14   | 14         | 11         | 9          |            |            |
|                   | CY          | 0           | —  | 2    | 2          | 2          | 2          | 2          |            |
|                   | DIR         | 0           | —  | 2    | 2          | 2          | 2          | 2          |            |
| SETI <sup>※</sup> | reg. 1      | 0           | —  | 6    | 6          | 6          | 6          | 2          |            |
|                   | mem. 1      | 2           | 0  | —    | 16         | 16         | 13         | 13         | 7          |
|                   |             |             | 1  | 奇数   | 24         | 24         | 21         | 21         | 11         |
|                   |             |             |    | 偶数   | 16         | 16         | 13         | 7          |            |
|                   | reg. CL     | 0           | —  | 7+n  | 7+n        | 7+n        | 7+n        | 2+n        |            |
|                   | mem. CL     | 2           | 0  | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |
|                   |             |             | 1  | 奇数   | 27+n       | 27+n       | 24+n       | 24+n       | 10+n       |
|                   |             |             |    | 偶数   | 19+n       | 19+n       | 16+n       | 6+n        |            |
|                   | reg. imm8   | 0           | —  | 7+n  | 7+n        | 7+n        | 7+n        | 2+n        |            |
|                   | mem. imm8   | 2           | 0  | —    | 19+n       | 19+n       | 16+n       | 16+n       | 6+n        |
| 1                 |             |             | 奇数 | 27+n | 27+n       | 24+n       | 24+n       | 10+n       |            |
|                   |             |             | 偶数 | 19+n | 19+n       | 16+n       | 6+n        |            |            |

注 n: シフト数

表 2-8 命令実行クロック数一覧 (13/15)

| ニモニック              | オペランド                  | ワード<br>抽出回数  | 条 件 |      | ク ロ ッ ク 数    |              |             |             |            |  |
|--------------------|------------------------|--------------|-----|------|--------------|--------------|-------------|-------------|------------|--|
|                    |                        |              | W   | アドレス | Y20, Y20HL   | Y30, Y30HL   | Y40, Y40HL  | Y50, Y50HL  | Y33A, Y33A |  |
| SFR <sup>2)</sup>  | reg. l                 | 0            | —   | —    | 6            | 6            | 6           | 6           | 2          |  |
|                    | mem. l                 | 2            | 0   | —    | 16           | 16           | 13          | 13          | 7          |  |
|                    |                        |              | 1   | 奇数   | 24           | 24           | 21          | 21          | 11         |  |
|                    |                        |              |     | 偶数   | 16           | 16           | 13          | 13          | 7          |  |
|                    | reg. CL                | 0            | —   | 7+n  | 7+n          | 7+n          | 7+n         | 2+n         |            |  |
|                    | mem. CL                | 2            | 0   | —    | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
|                    |                        |              | 1   | 奇数   | 27+n         | 27+n         | 24+n        | 24+n        | 10+n       |  |
|                    |                        |              |     | 偶数   | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
|                    | reg. imm8              | 0            | —   | 7+n  | 7+n          | 7+n          | 7+n         | 2+n         |            |  |
|                    | mem. imm8              | 2            | 0   | —    | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
| 1                  |                        |              | 奇数  | 27+n | 27+n         | 24+n         | 24+n        | 10+n        |            |  |
|                    |                        |              | 偶数  | 19+n | 19+n         | 16+n         | 16+n        | 6+n         |            |  |
| SR2A <sup>2)</sup> | reg. l                 | 0            | —   | —    | 6            | 6            | 6           | 2           |            |  |
|                    | mem. l                 | 2            | 0   | —    | 16           | 16           | 13          | 13          | 7          |  |
|                    |                        |              | 1   | 奇数   | 24           | 24           | 21          | 21          | 11         |  |
|                    |                        |              |     | 偶数   | 16           | 16           | 13          | 13          | 7          |  |
|                    | reg. CL                | 0            | —   | 7+n  | 7+n          | 7+n          | 7+n         | 2+n         |            |  |
|                    | mem. CL                | 2            | 0   | —    | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
|                    |                        |              | 1   | 奇数   | 27+n         | 27+n         | 24+n        | 24+n        | 10+n       |  |
|                    |                        |              |     | 偶数   | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
|                    | reg. imm8              | 0            | —   | 7+n  | 7+n          | 7+n          | 7+n         | 2+n         |            |  |
|                    | mem. imm8              | 2            | 0   | —    | 19+n         | 19+n         | 16+n        | 16+n        | 6+n        |  |
| 1                  |                        |              | 奇数  | 27+n | 27+n         | 24+n         | 24+n        | 10+n        |            |  |
|                    |                        |              | 偶数  | 19+n | 19+n         | 16+n         | 16+n        | 6+n         |            |  |
| SS :               | なし                     | 0            | —   | 2    | 2            | 2            | 2           | 2           |            |  |
| STM <sup>2)</sup>  | [DS1-upao : ]dst-block | 1×rep<br>(1) | 0   | —    | 7+4×rep (7)  | 7+4×rep (7)  | 5+4×rep (5) | 5+4×rep (5) | 3×rep (3)  |  |
|                    |                        |              | 1   | 奇数   | 7+8×rep (11) | 7+8×rep (11) | 5+8×rep (9) | 5+8×rep (9) | 3×rep (3)  |  |
|                    |                        |              |     | 偶数   | 7+4×rep (7)  | 7+4×rep (7)  | 5+4×rep (5) | 5+4×rep (5) | 3×rep (3)  |  |
| STMB <sup>2)</sup> | なし                     | 1×rep(2)     | 0   | —    | 7+4×rep (7)  | 7+4×rep (7)  | 5+4×rep (5) | 5+4×rep (5) | 3×rep (3)  |  |
| STMW <sup>2)</sup> | なし                     | 1×rep<br>(1) | 1   | 奇数   | 7+8×rep (11) | 7+8×rep (11) | 5+8×rep (9) | 5+8×rep (9) | 3×rep (3)  |  |
|                    |                        |              |     | 偶数   | 7+4×rep (7)  | 7+4×rep (7)  | 5+4×rep (5) | 5+4×rep (5) | 3×rep (3)  |  |
| SUB                | reg. reg'              | 0            | —   | —    | 2            | 2            | 2           | 2           | 2          |  |
|                    | mem. reg               | 2            | 0   | —    | 16           | 16           | 13          | 13          | 7          |  |
|                    |                        |              | 1   | 奇数   | 24           | 24           | 21          | 21          | 11         |  |
|                    |                        |              |     | 偶数   | 16           | 16           | 13          | 13          | 7          |  |
|                    | reg. mem               | 1            | 0   | —    | 11           | 11           | 10          | 10          | 6          |  |
|                    |                        |              | 1   | 奇数   | 15           | 15           | 14          | 14          | 8          |  |
|                    |                        |              |     | 偶数   | 11           | 11           | 10          | 10          | 6          |  |
|                    | reg. imm               | 0            | —   | —    | 4            | 4            | 4           | 4           | 2          |  |
|                    | mem. imm               | 2            | 0   | —    | 18           | 18           | 15          | 15          | 7          |  |
|                    |                        |              | 1   | 奇数   | 26           | 26           | 23          | 23          | 11         |  |
|                    |                        |              | 偶数  | 18   | 18           | 15           | 15          | 7           |            |  |
| occ. imm           | 0                      | —            | —   | 4    | 4            | 4            | 4           | 2           |            |  |

注1. n: シフト数

2. ( ): 一回だけの処理に適用

表 2-8 命令実行クロック数一覧 (14/15)

| ★<br>ニモニック        | オペランド                                             | ワード<br>発生回数 | 条件<br>W アドレス | クロック数      |            |            |            |            |    |
|-------------------|---------------------------------------------------|-------------|--------------|------------|------------|------------|------------|------------|----|
|                   |                                                   |             |              | V20, V20RL | V30, V30RL | V40, V40RL | V50, V50RL | V33A, V33A |    |
| SUBC <sup>2</sup> | [DS1-spec:] dst-string,<br>[Src-spec:] src-string | 0           | —            | 19×m+7     | 19×m+7     | 18×m+7     | 19×m+7     | 18×m+2     |    |
|                   | なし                                                | 0           | —            | 19×m+7     | 19×m+7     | 18×m+7     | 19×m+7     | 18×m+2     |    |
| SUBC              | req, req'                                         | 0           | —            | 2          | 2          | 2          | 2          | 2          |    |
|                   | mem, req                                          | 2           | 0            | —          | 16         | 16         | 13         | 13         | 7  |
|                   |                                                   |             | 1            | 奇数         | 24         | 24         | 21         | 21         | 11 |
|                   |                                                   |             |              | 偶数         | 16         |            | 13         | 7          |    |
|                   | req, mem                                          | 1           | 0            | —          | 11         | 11         | 10         | 10         | 6  |
|                   |                                                   |             | 1            | 奇数         | 15         | 15         | 14         | 14         | 8  |
|                   |                                                   |             |              | 偶数         | 11         |            | 10         | 6          |    |
|                   | req, imm                                          | 0           | —            | 4          | 4          | 4          | 4          | 2          |    |
|                   | mem, imm                                          | 2           | 0            | —          | 18         | 18         | 15         | 15         | 7  |
|                   |                                                   |             | 1            | 奇数         | 26         | 26         | 23         | 23         | 11 |
|                   |                                                   |             | 偶数           | 18         |            | 15         | 7          |            |    |
| acc, imm          | 0                                                 | —           | 4            | 4          | 4          | 4          | 2          |            |    |
| TEST              | req, req'                                         | 0           | —            | 2          | 2          | 2          | 2          | 2          |    |
|                   | mem, req                                          | 1           | 0            | —          | 10         | 10         | 9          | 9          | 6  |
|                   |                                                   |             | 1            | 奇数         | 14         | 14         | 13         | 13         | 8  |
|                   |                                                   |             |              | 偶数         | 10         |            | 9          | 6          |    |
|                   | req, mem                                          | 1           | 0            | —          | 10         | 10         | 9          | 9          | 6  |
|                   |                                                   |             | 1            | 奇数         | 14         | 14         | 13         | 13         | 8  |
|                   |                                                   |             |              | 偶数         | 10         |            | 9          | 6          |    |
|                   | req, imm                                          | 0           | —            | 4          | 4          | 4          | 4          | 2          |    |
|                   | mem, imm                                          | 1           | 0            | —          | 11         | 11         | 10         | 10         | 6  |
|                   |                                                   |             | 1            | 奇数         | 15         | 15         | 14         | 14         | 8  |
|                   |                                                   |             | 偶数           | 11         |            | 10         | 6          |            |    |
| acc, imm          | 0                                                 | —           | 4            | 4          | 4          | 4          | 2          |            |    |
| TEST1             | req8, CL                                          | 0           | —            | 3          | 3          | 3          | 3          | 4          |    |
|                   | mem8, CL                                          | 0           | —            | 8          | 8          | 7          | 7          | 8          |    |
|                   | req16, CL                                         | 0           | —            | 3          | 3          | 3          | 3          | 4          |    |
|                   | mem16, CL                                         | 1           | —            | 奇数         | 12         | 12         | 11         | 11         | 10 |
|                   |                                                   |             | —            | 偶数         | 8          |            | 7          | 8          |    |
|                   | req8, imm3                                        | 0           | —            | 4          | 4          | 4          | 4          | 4          |    |
|                   | mem8, imm3                                        | 0           | —            | 9          | 9          | 8          | 8          | 8          |    |
|                   | req16, imm4                                       | 0           | —            | 4          | 4          | 4          | 4          | 4          |    |
|                   | mem16, imm4                                       | 1           | —            | 奇数         | 13         | 13         | 12         | 12         | 10 |
|                   |                                                   |             | —            | 偶数         | 9          |            | 8          | 8          |    |
| TRANS             | src-table                                         | 1           | —            | 9          | 9          | 9          | 9          | 5          |    |
|                   | なし                                                | 1           | —            | 9          | 9          | 9          | 9          | 5          |    |
| TRANSB            | なし                                                | 1           | —            | 9          | 9          | 9          | 9          | 5          |    |

注 m: BCD桁数×1/2

表 2-8 命令実行クロック数一覧 (15/15)

| ★        | メモニック     | オペランド     | ワード<br>伝送回数 | 条 件 |      | ク ロ ッ ク 数  |            |            |            |            |
|----------|-----------|-----------|-------------|-----|------|------------|------------|------------|------------|------------|
|          |           |           |             | W   | アドレス | V20, V20HL | V30, V30HL | V40, V40HL | V80, V80HL | V33A, V33A |
| XCH      | reg, reg' | 0         | —           | —   | 3    | 3          | 3          | 3          | 3          |            |
|          | mem, reg  | 2         | 0           | —   | 16   | 16         | 13         | 13         | 8          |            |
|          |           |           | 1           | 奇数  | 24   | 24         | 21         | 21         | 12         |            |
|          |           |           |             | 偶数  | —    | 16         | —          | 13         | 8          |            |
|          | reg, mem  | 2         | 0           | —   | 16   | 16         | 13         | 13         | 8          |            |
|          |           |           | 1           | 奇数  | 24   | 24         | 21         | 21         | 12         |            |
|          |           |           |             | 偶数  | —    | 16         | —          | 13         | 8          |            |
|          | AW, reg16 | 0         | —           | —   | 3    | 3          | 3          | 3          | 3          |            |
|          | reg16, AW | 0         | —           | —   | 3    | 3          | 3          | 3          | 3          |            |
|          | XOR       | reg, reg' | 0           | —   | —    | 2          | 2          | 2          | 2          | 2          |
| mem, reg |           | 2         | 0           | —   | 16   | 16         | 13         | 13         | 7          |            |
|          |           |           | 1           | 奇数  | 24   | 24         | 21         | 21         | 11         |            |
|          |           |           |             | 偶数  | —    | 16         | —          | 13         | 7          |            |
| reg, mem |           | 1         | 0           | —   | 11   | 11         | 10         | 10         | 6          |            |
|          |           |           | 1           | 奇数  | 15   | 15         | 14         | 14         | 8          |            |
|          |           |           |             | 偶数  | —    | 11         | —          | 10         | 6          |            |
| reg, imm |           | 0         | —           | —   | 4    | 4          | 4          | 4          | 2          |            |
| mem, imm |           | 2         | 0           | —   | 18   | 18         | 15         | 15         | 7          |            |
|          |           |           | 1           | 奇数  | 26   | 26         | 23         | 23         | 11         |            |
|          | 偶数        |           |             | —   | 18   | —          | 15         | 7          |            |            |
| acc, imm | 0         | —         | —           | 4   | 4    | 4          | 4          | 2          |            |            |

## 付録A レジスタ構成

### A.1 汎用レジスタ (AW, BW, CW, DW)

汎用レジスタには4個の16ビット・レジスタがあり、16ビット・レジスタとしてはもちろん、各レジスタを上位、下位の8ビットに分けて、8ビット・レジスタ (AH, AL, BH, BL, CH, CL, DH, DL) としてもアクセス可能です。

したがって、これらのレジスタは転送命令、算術演算命令、論理演算命令など広範の命令に対して8または16ビット・レジスタとして利用されます。

また、次に示すように各レジスタは、特定の命令処理にデフォルト・レジスタとして使用されます。

- AW : ワード乗除算, ワード入出力, データ変換
- AL : バイト乗除算, バイト入出力, BCDローテート, データ変換
- AH : バイト乗除算
- BW : データ変換 (テーブル参照)
- CW : ループ制御ブランチ, リピート・プリフィクス
- CL : シフト命令, ローテーション命令, BCD演算
- DW : ワード乗除算, 間接アドレッシング入出力

### A.2 セグメント・レジスタ (PS, SS, DSO, DS1)

16ビットVシリーズは、メモリ空間を64Kバイト単位の論理セグメントに分割し、同時に4つのセグメントまで管理できます (セグメント方式)。各セグメントは、それぞれ次の4つのセグメント・レジスタで先頭アドレスが指定されます。

- プログラム・セグメント・レジスタ (PS) : 命令を格納しているセグメントのベース・アドレスを指定
- スタック・セグメント・レジスタ (SS) : スタック動作を行うセグメントのベース・アドレスを指定
- データ・セグメント0レジスタ (DS0) : データを格納しているセグメントのベース・アドレスを指定
- データ・セグメント1レジスタ (DS1) : データ転送命令で、データの転送先などに使用するセグメントのベース・アドレスを指定

### A.3 ポインタ (SP, BP)

ポインタは2本の16ビット・レジスタ (スタック・ポインタ (SP), ベース・ポインタ (BP)) から構成されます。各レジスタは、メモリのアドレスを指定するためのポインタとして使用され、命令中で参照することもできますが、メモリ・データ参照時にはインデクス・レジスタとして用いられます。

SPはスタック・セグメント内の最新のデータが格納されているアドレスを示し、スタック操作時にはデフォルト・レジスタとして使用されます。

BPはスタック上に格納されたデータの取り出しに使用されます。

### A.4 プログラム・カウンタ (PC)

PCは、現在エグゼキューション・ユニット (EXU) が実行しようとしているプログラム・メモリ・アドレスのオフセット情報を保持する16ビット・バイナリ・カウンタです。

PCの値は、マイクロプログラムが命令キューから命令バイトをフェッチすることに自動的にインクリメント (+1) されます。

また、ジャンプ命令、コール命令、リターン命令およびブレイク命令の実行時には、新たなロケーションがロードされ、このときPCの値はプリフェッチ・ポインタ (PFP) の値と同じになります。

### A.5 プログラム・ステータス・ワード (PSW)

PSWは、6種のステータス・フラグと4種のコントロール・フラグで構成されます。

#### ステータス・フラグ

- オーバフロー・フラグ (V)
- サイン・フラグ (S)
- ゼロ・フラグ (Z)
- 補助キャリー・フラグ (AC)
- パリティ・フラグ (P)
- キャリー・フラグ (CY)

#### コントロール・フラグ

- モード・フラグ (MD) 注
- 方向フラグ (DIR)
- 割り込み許可フラグ (IE)
- ブレイク・フラグ (BRK)

注 V33A, V53Aを除く

ステータス・フラグは、各種命令実行の結果（データ値）に従って自動的にセット(1)、リセット(0)されます。

CYフラグは、命令によって直接にセット、リセット、反転が可能です。

コントロール・フラグは、命令によってセット、リセットされ、CPUの動作を制御します。

また、IEとBRKフラグは、割り込み処理が起動されるとリセットされます。

RESET入力によって、MDフラグのみがセット(1)され、その他の全フラグはリセット(0)されます。

PSWは、以下に示すような処理によりバイトまたはワード単位で操作されます。バイト単位の処理は下位8ビット（Vフラグを除くステータス・フラグが含まれる）に対してのみ行われます。

図Ⅱ-1 PSWの構成

|        |    |    |    |    |             |        |             |   |   |   |        |   |   |   |        |   |
|--------|----|----|----|----|-------------|--------|-------------|---|---|---|--------|---|---|---|--------|---|
|        | 15 | 14 | 13 | 12 | 11          | 10     | 9           | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1      | 0 |
| M<br>D | 1  | 1  | 1  | V  | D<br>I<br>R | I<br>E | B<br>R<br>K | S | Z | O | A<br>C | O | P | 1 | C<br>Y |   |

注 V33A, V53AにはMDフラグはありません。PSWのビット15は1に固定です。

ビット0-ビット7は、MOV命令によってAHにストアまたはリストアできます。

PSWの全ビットは、割り込み発生時およびコール命令実行時にスタックに退避され、リターン命令(RETI, RETEM)によって復帰されます。

また、PUSH PSW命令、POP PSW命令によって、単独にPSWをスタックに退避または復帰できます。

注 MDフラグには書き込み許可状態と書き込み禁止状態があり、書き込み禁止状態ではRETIまたはPOP PSW命令を実行してもMDフラグは復帰されず前の値を保持します。なお、MDフラグはリセット動作とRETEM命令によって書き込み禁止状態になり、BRKEM命令で書き込み許可状態になります。

フラグは各命令の実行後、次に示すような状態になります。

(1) キャリー・フラグ (CY)

(a) 2進加減算

バイト演算の場合には、演算の結果ビット7からのキャリーまたはボローがあったときセットされ、なければリセットされます。

ワード演算の場合には、演算の結果ビット15からのキャリーまたはボローがあったときセッ

トされ、なければリセットされます。

インクリメント、デクリメント命令では変化しません。

(b) 論理演算

演算結果にかかわらずリセットされます。

(c) 2進乗算

符号なしバイト演算の結果、AHが0ならばリセットされ、0以外ならばセットされます。

符号付きバイト演算の結果、AHがALのサイン拡張になっていればリセットされ、それ以外ではセットされます。

符号なしワード演算の結果、DWが0ならばリセットされ、0以外ならばセットされます。

符号付きワード演算の結果、DWがAWのサイン拡張になっていればリセットされ、それ以外ではセットされます。

8ビット・イミディエイト演算の場合は、積が16ビット以内のときにはリセットされ、16ビットを超える場合にはセットされます。

(d) 2進除算

不 定

(e) シフト/ローテート

CYフラグを含むシフトおよびローテートの場合は、CYフラグへシフトされるビットが1の場合はセットされ、0の場合はリセットされます。

(2) パリティ・フラグ (P)

(a) 2進加減算、論理演算、シフト

演算の結果の低位8ビットのうち、1であるビットの数が偶数値のときセットされ、奇数値のときリセットされます。

結果がオール0のときにはセットされます。

(b) 2進乗除算

不 定

(3) 補助キャリー・フラグ (AC)

(a) 2進加減算

バイト演算の場合、低位4ビットから上位4ビットへのキャリーまたは上位4ビットから下



位4ビットへのポローがあるときセットされ、それ以外のときリセットされます。

ワード演算では、下位バイトについてバイト演算の場合と同様の動作を行います。

(b) 論理演算、2進乗除算、シフト/ローテート

不 定

(4) ゼロ・フラグ (Z)

(a) 2進加減算、論理演算、シフト/ローテート

バイト演算の場合には結果の8ビットが、ワード演算の場合には結果の16ビットがすべて0の場合にセットされ、それ以外のときにはリセットされます。

(b) 2進乗除算

不 定

(5) サイン・フラグ (S)

(a) 2進加減算、論理演算、シフト/ローテート

バイト演算の場合、結果のビット7が1のときセットされ、0のときリセットされます。

ワード演算の場合、結果のビット15が1のときセットされ、0のときリセットされます。

(b) 2進乗除算

不 定

(6) オーバフロー・フラグ (V)

(a) 2進加減算

バイト演算の場合、ビット7とビット6からのキャリーが違っていればセットされ、同じであればリセットされます。

ワード演算の場合、ビット15とビット14からのキャリーが違っていればセットされ、同じであればリセットされます。

(b) 2進乗算

符号なしバイト演算の結果、AHが0ならばリセットされ、0以外ならばセットされます。

符号付きバイト演算の結果、AHがALのサイン拡張になっていればリセットされ、それ以外ではセットされます。

符号なしワード演算の結果、DWが0ならばリセットされ、0以外ならばセットされます。

符号付きワード演算の結果、DWがAWのサイン拡張になっていればリセットされ、それ以外ではセットされます。

8ビット・イミディエイト演算の場合は、積が16ビット以内のときにはリセットされ、16ビットを超える場合にはセットされます。

(c) 2進除算

リセットされます。

(d) 論理演算

リセットされます。

(e) シフト/ローテート

左1ビット・シフト/ローテートの場合、演算結果により次のようになります。

CY=最上位ビットのとき：リセット

CY≠最上位ビットのとき：セット

右1ビット・シフト/ローテートの場合、演算結果により次のようになります。

最上位ビット=最上位の次の下位ビットのとき：リセット

最上位ビット≠最上位の次の下位ビットのとき：セット

多ビット・シフト/ローテートの場合は不定となります。

(7) ブレーク・フラグ (BRK)

PSWの一部としてスタックに退避されている状態でのみ、メモリ操作命令によってセットでき、セット後にPSWにリストアされると効果を持ちます。

BRKフラグがセットされていれば、命令を1つ実行するとソフトウェア割り込み（割り込みベクタ1）が自動的に発生し、1命令ずつのトレースが可能となります。

(8) 割り込み許可フラグ (IE)

DI命令でセットされてINT割り込みを許可状態にし、DI命令でリセットされてINT割り込みを禁止状態にします。

(9) 方向フラグ (DIR)

SET1 DIR命令でセットされ、CLR1 DIR命令でリセットされます。

DIRフラグがセットされていると、ブロック転送/入出力系命令において上位アドレスから下位アドレスへ向かって処理を行い、リセットされていると、下位アドレスから上位アドレスへ向かって処理を行います。

## (10) モード・フラグ (MD) (V33A, V53Aを除く)

RESET入力によってセットされてCPUをネイティブ・モードに設定し、BRKEM命令によってリセットされてエミュレーション・モードに設定します。

また、CALLN命令およびRETEM命令によってセットされてネイティブ・モードに設定します。

RESET入力とRETEM命令はMDフラグを書き込み禁止状態に設定し、RETI命令、POP PSW命令を実行してもMDフラグはリストアされなくなります。BRKEM命令はMDフラグを書き込み許可状態にします。

## A.6 インデクス・レジスタ (IX, IY)

2本の16ビット・レジスタ (IX, IY) から構成されます。各レジスタは、命令中で参照することもできますが、メモリ・データ参照時には、実効アドレス生成用のインデクス・レジスタとして用いられます。また、特定の命令処理では次のような特別な役割を持っています。

IX: ブロック・データ操作命令におけるソース・オペランド用アドレス・レジスタ

可変長ビット・フィールド操作命令におけるベース・レジスタ

BCDストリング演算命令におけるソース・オペランド用アドレス・レジスタ

IY: ブロック・データ操作命令におけるデスティネーション・オペランド用アドレス・レジスタ

可変長ビット・フィールド操作命令におけるベース・レジスタ

BCDストリング演算命令におけるデスティネーション・オペランド用アドレス・レジスタ

(ノ 毛)

## 付録B アドレッシング・モード

### B.1 命令アドレス

命令が実行されることに自動的にインクリメントされる以外に、命令の実行順を制御するいくつかの方法があります。それらを以下に示します。

#### (1) ダイレクト・アドレッシング

命令バイト中の2バイトまたは4バイト・イミディエト・データが直接PCまたはPSとPCの両方にロードされ、ブランチ・アドレスとなります。

次の命令を実行する際に用いられます。

```
CALL far-proc
CALL memptr16
CALL memptr32
BR far-label
BR memptr16
BR memptr32
```

#### (2) レラティブ・アドレッシング

命令バイト中の1バイトまたは2バイト・イミディエト・データが符号付きディスプレースメント値となってPCに加算され、ブランチ・アドレスとなります。

8ビット・ディスプレースメントのときは、符号拡張されて16ビット・データとなってPCに加算されます。

加算されるときPCの内容は、次の命令の先頭アドレスを示しています。

次の命令を実行する際に用いられます。

```
CALL near-proc
BR near-label
BR short-label
条件付きブランチ命令 short-label
```

#### (3) レジスタ・アドレッシング

命令バイト中のレジスタ指定フィールド(3ビット)で指定される任意の16ビット・レジスタの内容が、ブランチ・アドレスとしてPCにロードされます。

データの場合と異なり、8個のすべての16ビット・レジスタ (AW, BW, CW, DW, IX, IY, SP, BP) を使用することができます。

次の命令を実行する際に用いられます。

## 記述例

```
CALL regptr16    CALL AW
BR  regptr16    BR  BW
```

## (4) レジスタ・インダイレクト・アドレッシング

命令バイト中のレジスタ指定フィールドで指定される16ビット・レジスタ (IX, IY, BW) でアドレスされるメモリの内容 (ワードまたはダブル・ワード) がブランチ・アドレスとしてPC (またはPCとPSの両方) にロードされます。

## 記述例

```
CALL memptr16    CALL WORD PTR (IX)
CALL memptr32    CALL DWORD PTR (IY)
BR  memptr16    BR  WORD PTR (BW)
BR  memptr32    BR  DWORD PTR (IX)
```

備考 WORD PTRと指定されたものはmemptr16の命令コードが、DWORD PTRと指定されたものはmemptr32の命令コードが、アセンブラによって生成されます。

## (5) インデクスト・アドレッシング

命令バイト中の1バイトまたは2バイト・イミディエイト・データが符号付きディスプレイメント値となって、インデクス・レジスタとして働く16ビット・レジスタ (IXまたはIY) に加算され、その結果がアドレスするメモリの内容 (ワードまたはダブル・ワード) がブランチ・アドレスとしてPCにロードされます。

次の命令を実行する際に用いられます。

## 記述例

```
CALL memptr16    CALL var (IX) (2)
CALL memptr32    CALL var (IY)
BR  memptr16    BR  var (IY)
BR  memptr32    BR  var (IX+4)
```

備考 変数varがワード属性を持つ場合はmemptr16の命令コードが、ダブル・ワード属性を持つ場合はmemptr32の命令コードがアセンブラによって生成されます。

## (6) ベース・アドレッシング

命令バイト中の1バイトまたは2バイト・イミューディエト・データが符号付きディスプレイメント値となり、ベース・レジスタとして働く16ビット・レジスタ (BPまたはBW) に加算され、その結果がアドレスするメモリの内容 (ワードまたはダブル・ワード) がブランチ・アドレスとしてPCにロードされます。

次の命令を実行する際に用いられます。

## 記述例

|               |                 |
|---------------|-----------------|
| CALL memptr16 | CALL var (BP+2) |
| CALL memptr32 | CALL var (BP)   |
| BR memptr16   | BR var (BW) (2) |
| BR memptr32   | BR var (BP)     |

備考 変数varがワード属性を持つ場合はmemptr16の命令コードが、ダブル・ワードの属性を持つ場合はmemptr32の命令コードがアセンブラによって生成されます。

## (7) ベース・インデクスト・アドレッシング

命令バイト中の1バイトまたは2バイト・イミューディエト・データが符号付きディスプレイメント値となり、この値とベース・レジスタとして働く16ビット・レジスタ (BPまたはBW) およびインデクス・レジスタとして働く16ビット・レジスタ (IXまたはIY) の3つが加算され、その結果がアドレスするメモリの内容 (ワードまたはダブル・ワード) が、ブランチ・アドレスとしてPCにロードされます。

次の命令を実行する際に用いられます。

## 記述例

|               |                      |
|---------------|----------------------|
| CALL memptr16 | CALL var (BP) (IX)   |
| CALL memptr32 | CALL var (BW+2) (IY) |
| BR memptr16   | BR var (BW) (2) (IX) |
| BR memptr32   | BR var (BP+4) (IY)   |

備考 変数varがワードの属性を持つ場合はmemptr16の命令コードが、ダブル・ワードの属性を持つ場合はmemptr32の命令コードがアセンブラによって生成されます。

## B.2 メモリ・オペランド・アドレス

命令を実行する際に操作対象となるレジスタやメモリなどをアドレスする方法として、次に示すいくつかの方法があります。

(1) レジスタ・アドレッシング

命令バイト中のレジスタ指定フィールド (reg=3ビット・フィールド, sreg=2ビット・フィールド) の内容が操作対象となるレジスタをアドレスします。

regの場合は、同じく命令バイト中のワードかバイトかを指定する1ビット(W)と組みになって8種のワード・レジスタ (AW, BW, CW, DW, BP, SP, IX, IY) と8種のバイト・レジスタ (AL, AH, BL, BH, CL, CH, DL, DH) を指定します。

sregの場合は、4種のセグメント・レジスタ (PS, SS, DS0, DS1) を指定します。

また、命令のオペレーション・コードが特定のレジスタを指定する場合があります。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式  | 記述方法                                                              |
|-------|-------------------------------------------------------------------|
| reg   | AW, BW, CW, DW, SP, BP, IX, IY,<br>AL, AH, BL, BH, CL, CH, DL, DH |
| reg16 | AW, BW, CW, DW, SP, BP, IX, IY                                    |
| reg8  | AL, AH, BL, BH, CL, CH, DL, DH                                    |
| sreg  | PS, SS, DS0, DS1                                                  |
| acc   | AW, AL                                                            |

記述例

|     |              |
|-----|--------------|
| MOV | reg, reg'の場合 |
| MOV | BP, SP       |
| MOV | AL, CL       |

(2) イミディエイト・アドレッシング

命令バイト中の1バイトまたは2バイト・イミディエイト・データがそのまま操作対象となります。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式      | 記述方法                |
|-----------|---------------------|
| imm       | 8/16ビット・イミディエイト・データ |
| imm16     | 16 ビット //           |
| imm8      | 8 ビット //            |
| pop-value | 16 ビット //           |

immの場合、8ビットか16ビットかの区別は、オペランドに記述されたimmの値または同時に記述される別のオペランドの属性をアセンブラが判断して決定し、ワード/バイト指定ビットWを決定します。



記述例

```
MOV reg, immの場合
MOV AL, 5:バイト
MUL reg16, reg16, imm16の場合
MUL AW, BW, 1000H
```

(3) ダイレクト・アドレッシング

命令バイト中のイミディエイト・データが、操作対象となるメモリをアドレスします。  
次のオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式 | 記述方法                             |
|------|----------------------------------|
| mem  | 8ビットまたは16ビット・メモリ・データを指定する16ビット変数 |
| dmem | //                               |
| imm4 | ビット・フィールド・データのビット長を示す4ビット変数      |

記述例

```
MOV mem, immの場合
MOV WORD_VAR, 2000H
MOV acc, dmemの場合
MOV AL, BYTE_VAR
```

(4) レジスタ・インダイレクト・アドレッシング

命令バイト中のメモリ指定フィールド (mod, mem) によって指定される16ビット・レジスタ (IX, IY, BW) が、操作対象となるメモリをアドレスします。  
次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式 | 記述方法             |
|------|------------------|
| mem  | (IX), (IY), (BW) |

記述例

```
SUB mem, regの場合
SUB (IX), AW
```

(5) オートインクリメント/デクリメント・アドレッシング

レジスタ・インダイレクト・アドレッシングに属するものですが、デフォルト・レジスタの内容で操作対象をアドレスしたあと、そのレジスタの内容を自動的にインクリメント/デクリメント (バイト処理なら+1/-1, ワード処理なら+2/-2) します。

つまり、このアドレッシング機能を用いれば、次のバイト/ワード・オペランド処理のためのアドレス更新が自動的に行われるわけです。

インクリメントとデクリメントの区別は、方向フラグ (DIR) によって行われ、DIR=0ならインクリメント、1ならデクリメントです。

このアドレッシングは、すべてデフォルト・レジスタに対して行われ、次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式      | デフォルト・レジスタ |
|-----------|------------|
| dst-block | IY         |
| src-block | IX         |

このアドレッシングとバイト/ワード・オペランドの繰り返し処理回数をカウントするカウンタ (CW) が組み合わされて、ブロック・データ処理の制御に用いられます。

#### (6) インデクスト・アドレッシング

命令バイト中の1バイトまたは2バイト・イミューディエト・データが符号付きディスプレイメント値となって、インデクス・レジスタとして働く16ビット・レジスタ (IXまたはIY) に加算され、その結果が操作対象となるメモリ・オペランドをアドレスします。

このアドレッシングは、アレイ・タイプのデータをアクセスするのに有効で、ディスプレイメントがアレイの開始アドレスを指し、インデクス・レジスタの内容がそこから何番目のアレイかを決定します。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式  | 記述方法               |
|-------|--------------------|
| mem   | var {IX}, var {IY} |
| mem16 | //                 |
| mem8  | //                 |

#### 記述例

```
TEST mem, immの場合
TEST BYTE_VAR {IX}, 7FH
TEST BYTE_VAR {IX+8}, 7FH
TEST WORD_VAR {IX} (8), 7FFFH
```

**備考** 変数varがバイト属性を持つ場合はバイト・オペランドが指定され、ワード属性を持つ場合はワード・オペランドが指定され、それぞれに該当する命令コードがアセンブラによって生成されます。

(7) ベース・アドレッシング

命令バイト中の1バイトまたは2バイト・イミューディエツ・データが符号付きディスプレイメント値となり、ベース・レジスタとして働く16ビット・レジスタ (BPまたはBW) に加算され、その結果が操作対象となるメモリ・オペランドをアドレスします。

このアドレッシングは、メモリの複数箇所にも置かれる構造タイプのデータをアクセスするのに有効で、ベース・レジスタが各構造の開始アドレスを指し、ディスプレイメントが各構造内の1要素を選択します。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式  | 記述方法               |
|-------|--------------------|
| mem   | var (BP), var (BW) |
| mem16 | //                 |
| mem8  | //                 |

記述例

```
SHL mem, 1の場合
SHL BYTE_VAR (BP), 1
SHL WORD_VAR (BP+2), 1
SHL BYTE_VAR (BP) (4), 1
```

**備考** 変数varがバイト属性を持つ場合はバイト・オペランドが指定され、ワード属性を持つ場合はワード・オペランドが指定され、それぞれに該当する命令コードがアセンブラによって生成されます。

(8) ベース・インデクスト・アドレッシング

命令バイト中の1バイトまたは2バイト・イミューディエツ・データが符号付きディスプレイメント値となり、この値とベース・レジスタとして働く16ビット・レジスタ (BPまたはBW) およびインデクス・レジスタとして働く16ビット・レジスタ (IXまたはIY) が加算され、その結果が操作対象となるメモリ・オペランドをアドレスします。

このアドレッシングは、ベース・レジスタの内容とインデクス・レジスタの内容を両方変化させて1つのデータを指すことができるため、アレイ・タイプを含んだ構造タイプのデータをアクセスするのに非常に有効です。すなわち、ベース・レジスタで各構造の先頭アドレスを指し、ディスプレイメント値がそこからアレイ・データの先頭アドレスまでのオフセット分を示し、アレイ・データの何番目をインデクス・レジスタが指すということができます。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式  | 記述方法                        |
|-------|-----------------------------|
| mem   | var [ベース・レジスタ] [インデクス・レジスタ] |
| mem16 | //                          |
| mem8  | //                          |

記述例

```

PUSH mem16の場合
PUSH WORD_VAR (BP) (IX)
PUSH WORD_VAR (BP+2) (IX+6)
PUSH WORD_VAR (BP) (4) (IX) (8)

```

(B) ビット・アドレッシング

命令バイト中の3/4ビット・イミューディエト・データまたはCLレジスタの下位3/4ビットが操作対象の8/16ビット・レジスタまたはメモリの1つのビットを指定します。

このアドレッシングを用いた命令によれば、レジスタ/メモリの特定の1ビットだけを、ほかのビット内容を意識せずにテスト(0, 1の判定)/セット/クリア/反転することができます。つまり、AND命令やOR命令を使用してセット、リセットするときのように、1ビットを操作するのにバイト/ワード・データを用意しなければならないということは起こりません。

次に示すオペランド記述形式を持つ命令を実行する際に用いられます。

| 記述形式 | 記述方式            |
|------|-----------------|
| imm4 | ワード・オペランドのビット番号 |
| imm3 | バイト・オペランドのビット番号 |
| CL   | CL              |

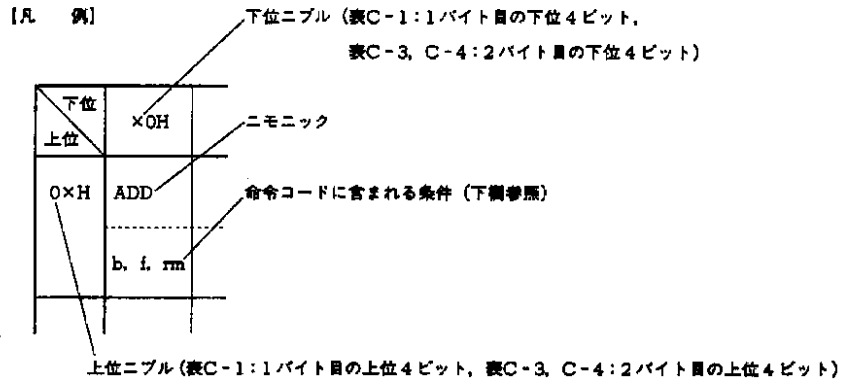
記述例

```

TEST1 reg8, CL
TEST1 AL, CL
NOT1 reg8, imm3
NOT1 CL, 5
CLR1 mem16, CL
CLR1 WORD_VAR (IX), CL
SET1 mem16, imm4
SET1 WORD_VAR (BP), 9

```

## 付録C 命令マップ



**【命令コードに含まれる条件】**

- b : バイト動作を行う
- d : ダイレクト・アドレッシングを用いる
- f : CPU内のレジスタからのリード動作を伴う
- i : イミューディエト・データを用いる
- ia : イミューディエト・データを用い、アキュムレータへの書き戻しがある
- id : インダイレクト・アドレッシングを用いる
- l : セグメント間の制御を伴う
- m : メモリ・データを用いる
- reg8 : 8ビット・レジスタを用いる
- rm : 2バイト目に実効アドレス・フィールドを持つ
- s : 符号拡張した16ビット・イミューディエト・データを用いる
- sr : セグメント・レジスタを使用する
- t : CPU内のレジスタへのライト動作
- v : インダイレクトでポート番号を指定する
- w : ワード動作を行う

上記以外の記号については表 2-4 命令形式またはオペレーション説明上の凡例を参照してください。


付録C 命令マップ

表C-1 命令マップ (1/2)

(a) ネイティブ・モード

| 下位<br>上位 | x0H           | x1H           | x2H            | x3H           | x4H            | x5H            | x6H          | x7H          | x8H           | x9H           | xAH           | xBH           | xCH           | xDH           | xEH           | xFH          |
|----------|---------------|---------------|----------------|---------------|----------------|----------------|--------------|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|
| 0xH      | ADD<br>b.l,m  | ADD<br>w.l,m  | ADD<br>b.l,m   | ADD<br>w.l,m  | ADD<br>b.l,m   | ADD<br>w.l,m   | FUSH<br>DSI  | POP<br>DSI   | OR<br>b.l,m   | OR<br>w.l,m   | OR<br>b.l,m   | OR<br>w.l,m   | OR<br>b.l,m   | OR<br>w.l,m   | PUSH<br>PS    |              |
| 1xH      | ADDC<br>b.l,m | ADDC<br>w.l,m | ADDC<br>b.l,m  | ADDC<br>w.l,m | ADEC<br>b.l,m  | ADEC<br>w.l,m  | FUSH<br>SS   | POP<br>SS    | SUBC<br>b.l,m | SUBC<br>w.l,m | SUBC<br>b.l,m | SUBC<br>w.l,m | SUBC<br>b.l,m | SUBC<br>w.l,m | FUSH<br>DSO   | POP<br>DSO   |
| 2xH      | AND<br>b.l,m  | AND<br>w.l,m  | AND<br>b.l,m   | AND<br>w.l,m  | AND<br>b.l,m   | AND<br>w.l,m   | DSI:<br>SS   | ADJAA<br>SUB | SUB<br>b.l,m  | SUB<br>w.l,m  | SUB<br>b.l,m  | SUB<br>w.l,m  | SUB<br>b.l,m  | SUB<br>w.l,m  | PS:<br>DSO    | ADJAS<br>DSO |
| 3xH      | XOR<br>b.l,m  | XOR<br>w.l,m  | XOR<br>b.l,m   | XOR<br>w.l,m  | XOR<br>b.l,m   | XOR<br>w.l,m   | SS:<br>ADJBA | CMP<br>b.l,m | CMP<br>w.l,m  | CMP<br>b.l,m  | CMP<br>w.l,m  | CMP<br>b.l,m  | CMP<br>w.l,m  | CMP<br>b.l,m  | DSO:<br>DSO   | ADJBS<br>DSO |
| 4xH      | INC<br>AW     | INC<br>CW     | INC<br>DW      | INC<br>BW     | INC<br>SP      | INC<br>BP      | INC<br>IX    | INC<br>IX    | DEC<br>AW     | DEC<br>CW     | DEC<br>DW     | DEC<br>BW     | DEC<br>SP     | DEC<br>BP     | DEC<br>IX     | DEC<br>IX    |
| 5xH      | FUSH<br>AW    | FUSH<br>CW    | FUSH<br>DW     | FUSH<br>BW    | FUSH<br>SP     | FUSH<br>BP     | FUSH<br>IX   | FUSH<br>IX   | POP<br>AW     | POP<br>CW     | POP<br>DW     | POP<br>BW     | POP<br>SP     | POP<br>BP     | POP<br>IX     | POP<br>IX    |
| 6xH      | FUSH<br>R     | POP<br>R      | CMEND<br>b.l,m | 未定義<br>b.l,m  | RSPHC<br>b.l,m | REPC<br>b.l,m  | FPO2<br>0    | FPO2<br>1    | FUSH<br>w.l   | MUL<br>w.l    | FUSH<br>a.l   | MUL<br>a.l    | INM<br>b      | INM<br>w      | OUTM<br>b     | OUTM<br>w    |
| 7xH      | BV            | BVY<br>BL     | BC<br>BNL      | BNC<br>BNL    | BE<br>BZ       | BNE<br>BNZ     | BH<br>BH     | BH<br>BH     | BN<br>BN      | BP<br>BP      | BPE<br>BPE    | BPO<br>BPO    | BCT<br>BCT    | BGE<br>BGE    | BLE<br>BLE    | BGT<br>BGT   |
| 8xH      |               |               |                |               | TEST<br>b.l,m  | TEST<br>w.l,m  | XCH<br>b.l,m | XCH<br>w.l,m | MOV<br>b.l,m  | MOV<br>w.l,m  | MOV<br>b.l,m  | MOV<br>w.l,m  | MOV<br>b.l,m  | MOV<br>w.l,m  | LDEA<br>a.l,m | MOV<br>a.l,m |
| 9xH      | NOP<br>CW     | XCH<br>DW     | XCH<br>BW      | XCH<br>SP     | XCH<br>BP      | XCH<br>IX      | XCH<br>IX    | CVTBY<br>IX  | CVTBY<br>IX   | CALL<br>l,d   | POLL          | FUSH<br>PSW   | POP<br>PSW    | MOV<br>PSW    | MOV<br>PSW    | MOV<br>PSW   |
| AxH      | MOV<br>AL,m   | MOV<br>AW,m   | MOV<br>AL,m    | MOV<br>AW,m   | MOV<br>b       | MOV<br>w       | CMPE<br>b    | CMPE<br>w    | TEST<br>b     | TEST<br>w     | STM<br>b      | STM<br>w      | LDM<br>b      | LDM<br>w      | CMPM<br>b     | CMPM<br>w    |
| BxH      | MOV<br>AL,l   | MOV<br>CL,l   | MOV<br>DL,l    | MOV<br>BL,l   | MOV<br>AH,l    | MOV<br>CH,l    | MOV<br>DH,l  | MOV<br>BH,l  | MOV<br>AW,l   | MOV<br>CW,l   | MOV<br>DW,l   | MOV<br>BW,l   | MOV<br>SP,l   | MOV<br>BP,l   | MOV<br>IX,l   | MOV<br>IX,l  |
| CxH      |               |               | RET<br>(SP)    | RET<br>DSI    | MOV<br>DSO     | MOV<br>b.l,m   | MOV<br>w.l,m | MOV<br>b.l,m | MOV<br>w.l,m  | MOV<br>b.l,m  | MOV<br>w.l,m  | MOV<br>b.l,m  | MOV<br>w.l,m  | MOV<br>b.l,m  | MOV<br>w.l,m  | MOV<br>b.l,m |
| DxH      |               |               |                |               | CVTBY<br>TRANS | CVTBY<br>TRANS | 未定義<br>TRANS | TRANS<br>0   | TRANS<br>1    | FPO1<br>2     | FPO1<br>3     | FPO1<br>4     | FPO1<br>5     | FPO1<br>6     | FPO1<br>7     | FPO1<br>7    |
| ExH      | DENZE         | DENZE         | DENZE          | SCWZ<br>b     | DI<br>w        | DI<br>w        | OUT<br>b     | OUT<br>w     | CALL<br>d     | BR<br>d       | BR<br>l,d     | BR<br>a,d     | BR<br>a,d     | BR<br>b,v     | BR<br>w,v     | OUT<br>b,v   |
| FxH      | RELOC         | 未定義           | REPNE<br>REPNE | REP<br>REP    | HALT           | NOTI           |              |              | CLR1<br>CY    | SET1<br>CY    | DI<br>CY      | DI<br>CY      | CLR1<br>DIR   | SET1<br>DIR   |               |              |

注 XCH AW, AWと同じオペレーション・コードです。

注意  : Group1, Group2, Imm, Shiftは命令コードの2バイト目のビット3-ビット5によって命令が定まります (表C-2参照)。


Group3は命令コードの2バイト目によって命令が定まります (表C-4参照)。

付録C 命令マップ

表C-1 命令マップ (2/2)

(b) エミュレーション・モード※

| 下位<br>上位 | *0H         | *1H          | *2H         | *3H         | *4H         | *5H         | *6H         | *7H         | *8H         | *9H         | *AH         | *BH         | *CH         | *DH         | *EH         | *FH         |
|----------|-------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0xH      | NOP         | LXI<br>B, m  | STAX<br>(m) | INX<br>B    | INR<br>B    | DCR<br>B    | MVI<br>B, n | RCL         | 未定義         | DAD<br>B    | LDAX<br>B   | DCX<br>B    | INR<br>C    | DCR<br>C    | MVI<br>C, n | RRC         |
| 1xH      | 未定義         | LXI<br>D, m  | STAX<br>(m) | INX<br>D    | INR<br>D    | DCR<br>D    | MVI<br>D, n | RAL         | 未定義         | DAD<br>D    | LDAX<br>D   | DCX<br>D    | INR<br>E    | DCR<br>E    | MVI<br>E, n | RAR         |
| 2xH      | 未定義         | LXI<br>H, m  | SHLD<br>(m) | INX<br>H    | INR<br>H    | DCR<br>H    | MVI<br>H, n | DAA         | 未定義         | DAD<br>H    | SHLD<br>(m) | DCX<br>H    | INR<br>L    | DCR<br>L    | MVI<br>L, n | CMA         |
| 3xH      | 未定義         | LXI<br>SP, m | STA<br>(m)  | INX<br>SP   | INR<br>M    | DCR<br>M    | MVI<br>M, n | SCF         | 未定義         | DAD<br>SP   | LDA<br>(m)  | DCX<br>SP   | INR<br>A    | DCR<br>A    | MVI<br>A, n | CMC         |
| 4xH      | MOV<br>B, B | MOV<br>B, C  | MOV<br>B, D | MOV<br>B, E | MOV<br>B, H | MOV<br>B, L | MOV<br>B, M | MOV<br>B, A | MOV<br>C, B | MOV<br>C, C | MOV<br>C, D | MOV<br>C, E | MOV<br>C, H | MOV<br>C, L | MOV<br>C, M | MOV<br>C, A |
| 5xH      | MOV<br>B, B | MOV<br>D, C  | MOV<br>D, D | MOV<br>D, E | MOV<br>D, H | MOV<br>D, L | MOV<br>D, M | MOV<br>D, A | MOV<br>E, B | MOV<br>E, C | MOV<br>E, D | MOV<br>E, E | MOV<br>E, H | MOV<br>E, L | MOV<br>E, M | MOV<br>E, A |
| 6xH      | MOV<br>H, B | MOV<br>H, C  | MOV<br>H, D | MOV<br>H, E | MOV<br>H, H | MOV<br>H, L | MOV<br>H, M | MOV<br>H, A | MOV<br>L, B | MOV<br>L, C | MOV<br>L, D | MOV<br>L, E | MOV<br>L, H | MOV<br>L, L | MOV<br>L, M | MOV<br>L, A |
| 7xH      | MOV<br>M, B | MOV<br>M, C  | MOV<br>M, D | MOV<br>M, E | MOV<br>M, H | MOV<br>M, L | MOV<br>M, M | MOV<br>M, A | MOV<br>A, B | MOV<br>A, C | MOV<br>A, D | MOV<br>A, E | MOV<br>A, H | MOV<br>A, L | MOV<br>A, M | MOV<br>A, A |
| 8xH      | ADD<br>B    | ADD<br>C     | ADD<br>D    | ADD<br>E    | ADD<br>H    | ADD<br>L    | ADD<br>M    | ADD<br>A    | ADC<br>B    | ADC<br>C    | ADC<br>D    | ADC<br>E    | ADC<br>H    | ADC<br>L    | ADC<br>M    | ADC<br>A    |
| 9xH      | SUB<br>B    | SUB<br>C     | SUB<br>D    | SUB<br>E    | SUB<br>H    | SUB<br>L    | SUB<br>M    | SUB<br>A    | SBB<br>B    | SBB<br>C    | SBB<br>D    | SBB<br>E    | SBB<br>H    | SBB<br>L    | SBB<br>M    | SBB<br>A    |
| AxH      | ANA<br>B    | ANA<br>C     | ANA<br>D    | ANA<br>E    | ANA<br>H    | ANA<br>L    | ANA<br>M    | ANA<br>A    | XRA<br>B    | XRA<br>C    | XRA<br>D    | XRA<br>E    | XRA<br>H    | XRA<br>L    | XRA<br>M    | XRA<br>A    |
| BxH      | ORA<br>B    | ORA<br>C     | ORA<br>D    | ORA<br>E    | ORA<br>H    | ORA<br>L    | ORA<br>M    | ORA<br>A    | CMP<br>B    | CMP<br>C    | CMP<br>D    | CMP<br>E    | CMP<br>H    | CMP<br>L    | CMP<br>M    | CMP<br>A    |
| CxH      | RNZ<br>B    | POP<br>m     | INZ<br>m    | JMP<br>m    | CNZ<br>B    | PUSH<br>B   | ADI<br>n    | RST<br>0    | RZ          | RET<br>n    | IF          | 未定義         | CF          | CALL<br>n   | ACI<br>n    | RST<br>1    |
| DxH      | RNC<br>D    | POP<br>m     | INC<br>n    | OUT<br>m    | CNC<br>D    | PUSH<br>D   | ADI<br>n    | RST<br>2    | RC          | 未定義         | IC          | IN<br>n     | CC          | 未定義         | SBI<br>n    | RST<br>3    |
| ExH      | RPO<br>H    | POP<br>m     | IPO<br>n    | XTHL<br>CP  | CPO<br>H    | PUSH<br>H   | ADI<br>n    | RST<br>4    | RPE         | PCRL<br>n   | IFE<br>n    | ICRQ        | CPE         | 未定義         | XRI<br>n    | RST<br>5    |
| FxH      | RP<br>PSW   | POP<br>m     | DI<br>m     | CP<br>m     | PUSH<br>PSW | ADI<br>n    | RST<br>n    | RM          | 未定義         | IM<br>n     | IE          | CM          | 未定義         | CPI<br>n    | RST<br>n    | RST<br>7    |

注意  : Group0は命令コードの2バイト目によって命令が定まります (表C-3参照)。

注 対象: V33A, V53A以外

付録C 命令マップ

表C-2 Group1, Group2, Imm, Shiftコード表

| 法      | 000  | 001 | 010  | 011  | 100  | 101 | 110  | 111  |
|--------|------|-----|------|------|------|-----|------|------|
| Imm    | AED  | OR  | ADDC | SUBC | AND  | SUB | XOR  | CMP  |
| Shift  | RDL  | ROR | ROL  | RORC | SHL  | SHR | 未定義  | SRRA |
| Group1 | TEST | 未定義 | NOT  | NEG  | MULU | MUL | DIVU | DIV  |
|        | rm   |     | rm   | rm   | rm   | rm  | rm   | rm   |
| Group2 | INC  | DEC | CALL | CALL | BR   | BR  | PUSH | 未定義  |
|        | rm   | rm  | ld   | ld   | ld   | ld  | rm   |      |

注 2バイト目のビット5-ビット3

表C-3 Group0コード表<sup>注</sup>

| 下位<br>上位 | ×0H | ×DH  | ×FH |
|----------|-----|------|-----|
| 0×H      |     |      |     |
| E×H      |     | CALL |     |
| F×H      |     | RETM |     |

注 対象：V33A, V53A以外  
備考 空白の欄は未定義コードです。

表C-4 Group3コード表

| 下位<br>上位 | ×0H  | ×1H  | ×2H | ×3H  | ×4H | ×5H | ×6H  | ×7H | ×8H  | ×9H  | ×AH | ×BH | ×CH | ×DH | ×EH | ×FH |
|----------|------|------|-----|------|-----|-----|------|-----|------|------|-----|-----|-----|-----|-----|-----|
| 0×H      |      |      |     |      |     |     |      |     |      |      |     |     |     |     |     |     |
| 1×H      | TEST | TEST | CLR | CLR  | SET | SET | NOT  | NOT | TEST | TEST | CLR | CLR | SET | SET | NOT | NOT |
|          | b    | w    | b   | w    | b   | w   | b    | w   | lb   | lw   | lb  | lw  | lb  | lw  | lb  | lw  |
| 2×H      | ADD  |      | SUB |      |     |     | CMPS |     | ROL  |      | ROR |     |     |     |     |     |
| 3×H      |      | INS  |     | EXT  |     |     |      |     | INS  |      | EXT |     |     |     |     |     |
|          |      | reg8 |     | reg8 |     |     |      |     | l    |      | l   |     |     |     |     |     |
| E×H      | MOV  |      |     |      |     |     |      |     |      |      |     |     |     |     |     |     |
|          | l    |      |     |      |     |     |      |     |      |      |     |     |     |     |     |     |
| F×H      | MOV  |      |     |      |     |     |      |     |      |      |     |     |     |     |     | MOV |
|          | l    |      |     |      |     |     |      |     |      |      |     |     |     |     |     | l   |

- 注1. V33A, V53Aのみ (V33A, V53A以外に対しては未定義コード)  
2. V33A, V53A以外 (V33A, V53Aに対しては未定義コード)

備考 空白の欄は未定義コードです。



### 付録D $\mu$ PD8086, 8088とのニモニク対応表

16ビットVシリーズの命令セットは、オブジェクト・コード・レベルで、 $\mu$ PD8086, 8088に対して上位互換性があります。

表D-1に $\mu$ PD8086, 8088と16ビットVシリーズのレジスタの対応を、表D-2にニモニクの対応を示します。

表D-1  $\mu$ PD8086, 8088とのレジスタ対応表

| $\mu$ PD8086,<br>8088 | 16ビット<br>Vシリーズ | $\mu$ PD8086,<br>8088 | 16ビット<br>Vシリーズ |
|-----------------------|----------------|-----------------------|----------------|
| AL                    | AL             | AX                    | AW             |
| CL                    | CL             | CX                    | CW             |
| DL                    | DL             | DX                    | DW             |
| BL                    | BL             | BX                    | BW             |
| AH                    | AH             | SP                    | SP             |
| CH                    | CH             | BP                    | BP             |
| DH                    | DH             | SI                    | IX             |
| BH                    | BH             | DI                    | IY             |

表D-2 μPD8086, 8088とのニモニック対応表

| μPD8086,<br>8088 | 16ビット<br>Vシリーズ   | μPD8086,<br>8088 | 16ビット<br>Vシリーズ    | μPD8086,<br>8088 | 16ビット<br>Vシリーズ   | μPD8086,<br>8088 | 16ビット<br>Vシリーズ    |
|------------------|------------------|------------------|-------------------|------------------|------------------|------------------|-------------------|
| AAA              | ADJBA            | JB               | BC/BL             | LOOP             | DBNZ             | SHR              | SHR               |
| AAD              | CVTDE            | JBE              | BNH               | LOOPE            | DBNZE            | SS :             | SS :              |
| AAM              | CVTBD            | JC               | BC/BL             | LOOPNE           | DBNZNE           | STC              | SET1 CY           |
| AAS              | ADJBS            | JCXZ             | BCWZ              | LOOPNZ           | DBNZNE           | STD              | SET1 DIR          |
| ADC              | ADDC             | JE               | BE/BZ             | LOOPZ            | DBNZE            | STI              | EI                |
| ADD              | ADD              | JG               | BGT               | MOV              | MOV              | STOS             | STM/STMB/<br>STMW |
| AND              | AND              | JGE              | BGE               | MOVS             | MOVSB            | SUB              | SUB               |
| CALL             | CALL             | JL               | BLT               | MOVSB            | MOVSB            | TEST             | TEST              |
| CBW              | CVTBW            | JLE              | BLE               | MOVSW            | MOVSB            | WAIT             | POLL              |
| CLC              | CLR1 CY          | JMP              | BR                | MUL              | MULU             | XCHG             | XCH               |
| CLD              | CLR1 DIR         | JNA              | BNH               | NEG              | NEG              | XLAT             | TRANS             |
| CLI              | DI               | JNAE             | BC/BL             | NOP              | NOP              | XLATB            | TRANSB            |
| CMC              | NOT1 CY          | JNB              | BNC/BNL           | NOT              | NOT              | XOR              | XOR               |
| CMP              | CMP              | JNBE             | BH                | OR               | OR               | -                | ADDAS             |
| CMPS             | CMPBK/<br>CMPBKW | JNC              | BNC/BNL           | OUT              | OUT              | -                | BRKEM             |
| CS :             | PS :             | JNE              | BNE/BNZ           | POP              | POP              | -                | BEKXA             |
| CWD              | CVTDL            | JNG              | BLE               | POPF             | POP PSW          | -                | CALLN             |
| DAA              | ADJ4A            | JNGE             | BLT               | PUSH             | PUSH             | -                | CHKIND            |
| DAS              | ADJ4S            | JNLE             | BGE               | PUSHF            | PUSH PSW         | -                | CMP4S             |
| DEC              | DEC              | JNLC             | BGT               | RCL              | ROL              | -                | DISPOSE           |
| DIV              | DIVU             | JNLO             | BNV               | RCL              | ROR              | -                | EXT               |
| DS :             | DS0 :            | JNO              | BPO               | REP              | REP              | -                | FPO2              |
| ES :             | DS1 :            | JNS              | BP                | REPE             | REPE             | -                | INM               |
| ESC              | FPC1             | JNZ              | BNE/BNZ           | REPNE            | REPNE            | -                | INS               |
| HLT              | HALT             | JO               | BV                | REPZ             | REPZ             | -                | OUTM              |
| IDIV             | DIV              | JPE              | BPE               | RET              | RET              | -                | PREPARE           |
| IMUL             | MUL              | JPO              | BPO               | ROL              | ROL              | -                | REPC              |
| IN               | IN               | JS               | BN                | ROR              | ROR              | -                | REPNC             |
| INC              | INC              | JZ               | BE/BZ             | SAHF             | MOV PSW, AH      | -                | RETEM             |
| INT              | BRK              | LAHF             | MOV AH, PSW       | SAL              | SHL              | -                | RETXA             |
| INT 3            | BRK 3            | LDS              | MOV DS, A         | SAR              | SHR              | -                | ROL4              |
| INTO             | BRKV             | LEA              | LDEA              | SBB              | SUBC             | -                | ROR4              |
| IRET             | RETI             | LES              | MOV DS, A         | SCAS             | CMPMB/<br>CMPMBW | -                | SUB4S             |
| JA               | BH               | LOCK             | BUSLOCK           |                  | CMPMB/<br>CMPMBW | -                | TEST1             |
| JAE              | BNC/BNL          | LDS              | LDM/LDMB/<br>LDMW | SHL              | SHL              |                  |                   |

備考 - : 該当する命令なし

付録E 命令索引 (ニモニク : 機能別)

|                        |                      |
|------------------------|----------------------|
| <b>【データ転送】</b>         | <b>【ビット・フィールド操作】</b> |
| LDEA ... 103           | EXT ... 90           |
| MOV ... 106            | INS ... 101          |
| TRANS ... 183          |                      |
| TRANSB ... 183         | <b>【入出力】</b>         |
| XCH ... 184            | IN ... 97            |
|                        | OUT ... 125          |
| <b>【リピート・プリフィクス】</b>   | <b>【プリミティブ入出力】</b>   |
| REP ... 135            | INM ... 99           |
| REPC ... 137           | OUTM ... 126         |
| REPE ... 135           |                      |
| REPNC ... 139          | <b>【加減算】</b>         |
| REPNE ... 141          | ADD ... 14           |
| REPNZ ... 141          | ADDC ... 18          |
| REPZ ... 135           | SUB ... 172          |
| <b>【プリミティブ・ブロック転送】</b> | SUBC ... 176         |
| CMPBK ... 68           | <b>【BCD演算】</b>       |
| CMPBKB ... 68          | ADD4S ... 16         |
| CMPBKW ... 68          | CMP4S ... 66         |
| CMPM ... 70            | ROL4 ... 151         |
| CMPMB ... 70           | ROR4 ... 157         |
| CMPMW ... 70           | SUB4S ... 174        |
| LDM ... 104            |                      |
| LDMB ... 104           | <b>【増減】</b>          |
| LDMW ... 104           | DEC ... 79           |
| MOVBK ... 110          | INC ... 98           |
| MOVKB ... 110          |                      |
| MOVBK ... 110          | <b>【乗除算】</b>         |
| STM ... 170            | DIV ... 82           |
| STMB ... 170           | DIVU ... 88          |
| STMW ... 170           | MUL ... 112          |

MULU ... 115

**【BCD補正】**

ADJ4A ... 20

ADJ4S ... 21

ADJBA ... 22

ADJBS ... 23

**【データ変換】**

CVTBD ... 72

CVTBW ... 73

CVTDB ... 74

CVTWL ... 75

**【比較】**

CMP ... 64

**【補数演算】**

NEG ... 117

NOT ... 119

**【論理演算】**

AND ... 24

OR ... 123

TEST ... 178

XOR ... 186

**【ビット操作】**

CLR1 ... 61

NOT1 ... 120

SET1 ... 161

TEST1 ... 180

**【シフト】**

SHL ... 164

SHR ... 166

SHRA ... 168

**【ローテート】**

ROL ... 149

ROLC ... 153

ROR ... 155

RORC ... 159

**【サブルーチン制御】**

CALL ... 54

RET ... 143

**【スタック操作】**

DISPOSE ... 81

POP ... 129

PREPARE ... 131

PUSH ... 133

**【ブランチ】**

BR ... 42

**【条件付きブランチ】**

BC ... 26

BCWZ ... 27

BE ... 28

BGE ... 29

BGT ... 30

BH ... 31

BL ... 26

BLE ... 32

BLT ... 33

BN ... 34

BNC ... 35

BNE ... 36

BNH ... 37

BNL ... 35

BNV ... 38

BNZ ... 36

BP ... 39

BPE ... 40  
BPO ... 41  
BV ... 53  
BZ ... 28  
DBNZ ... 76  
DENZE ... 77  
DBNZNE ... 78

**[割り込み]**

BRK ... 44  
BRKV ... 48  
CHKIND ... 59  
RETI ... 146

**[CPU制御]**

BUSLOCK ... 51  
DI ... 80  
EI ... 89  
FPO1 ... 92  
FPO2 ... 94  
HALT ... 96  
NOP ... 118  
POLL ... 128

**[セグメント・オーバーライド・プリフィクス]**

DS0: ... 87  
DS1: ... 87  
PS: ... 87  
SS: ... 87

**[エミュレーション・モード専用]**

BRKEM ... 46  
CALLN ... 57  
RETEM ... 145

**[拡張アドレス・モード専用]**

BRKXA ... 49

(ア ㉔)

**付録 F 命令索引 (ニモニック : アルファベット順)**

|              |                |
|--------------|----------------|
| <b>[A]</b>   | BRKV ... 48    |
| ADD ... 14   | BRKKA ... 49   |
| ADD4S ... 16 | BUSLOCK ... 51 |
| ADDC ... 18  | BV ... 53      |
| ADJ4A ... 20 | BZ ... 28      |
| ADJ4S ... 21 |                |
| ADJBA ... 22 | <b>[C]</b>     |
| ADJBS ... 23 | CALL ... 54    |
| AND ... 24   | CALLN ... 57   |
|              | CHKIND ... 59  |
|              | CLR1 ... 61    |
| <b>[B]</b>   | CMP ... 64     |
| BC ... 26    | CMP4S ... 66   |
| BCWZ ... 27  | CMPBK ... 68   |
| BE ... 28    | CMPBKB ... 68  |
| BGE ... 29   | CMPBKW ... 68  |
| BGT ... 30   | CMPM ... 70    |
| BH ... 31    | CMPMB ... 70   |
| EL ... 26    | CMPMW ... 70   |
| BLE ... 32   | CVTBD ... 72   |
| BLT ... 33   | CVTBW ... 73   |
| BN ... 34    | CVTDB ... 74   |
| BNC ... 35   | CVTWL ... 75   |
| BNE ... 36   |                |
| BNH ... 37   | <b>[D]</b>     |
| BNL ... 35   | DBNZ ... 76    |
| BNV ... 38   | DBNZE ... 77   |
| BNZ ... 36   | DBNZNE ... 78  |
| BP ... 39    | DEC ... 79     |
| BPE ... 40   | DI ... 80      |
| BPO ... 41   | DISPOSE ... 81 |
| BR ... 42    | DIV ... 82     |
| BRK ... 44   | DIVU ... 85    |
| BRKEM ... 46 |                |

|                |                 |
|----------------|-----------------|
| DS0: ... 87    | NOT ... 119     |
| DS1: ... 87    | NOT1 ... 120    |
| <b>[E]</b>     | <b>[O]</b>      |
| EI ... 89      | OR ... 123      |
| EXT ... 90     | OUT ... 125     |
|                | OUTM ... 126    |
| <b>[F]</b>     | <b>[P]</b>      |
| FPO1 ... 92    | POLL ... 128    |
| FPO2 ... 94    | POP ... 129     |
|                | PREPARE ... 131 |
| <b>[H]</b>     | PS: ... 87      |
| HALT ... 96    | PUSH ... 133    |
|                |                 |
| <b>[I]</b>     | <b>[R]</b>      |
| IN ... 97      | REP ... 135     |
| INC ... 98     | REPC ... 137    |
| INM ... 99     | REPE ... 135    |
| INS ... 101    | REPNC ... 139   |
|                | REPNE ... 141   |
| <b>[L]</b>     | REPNZ ... 141   |
| LDEA ... 103   | REPZ ... 135    |
| LDM ... 104    | RET ... 143     |
| LDMB ... 104   | RETEM ... 145   |
| LDMW ... 104   | RETI ... 146    |
|                | RETXA ... 147   |
| <b>[M]</b>     | ROL ... 149     |
| MOV ... 106    | ROL4 ... 151    |
| MOVBK ... 110  | ROLC ... 153    |
| MOVBKB ... 110 | ROR ... 135     |
| MOVBKW ... 110 | ROR4 ... 157    |
| MUL ... 112    | RORC ... 159    |
| MULU ... 115   |                 |
|                | <b>[S]</b>      |
| <b>[N]</b>     | SET1 ... 161    |
| NEG ... 117    | SHL ... 164     |
| NOP ... 118    |                 |



SHR ... 166  
SHRA ... 168  
SS: ... 87  
STM ... 170  
STMB ... 170  
STMW ... 170  
SUB ... 172  
SUB4S ... 174  
SUBC ... 176

**[T]**

TEST ... 178  
TESTI ... 180  
TRANS ... 183  
TRANSB ... 183

**[X]**

XCH ... 184  
XOR ... 186

— お問い合わせ先 —

**【技術的なお問い合わせ先】**

NEC半導体テクニカルネットライン  
 (電話：午前 9:00～12:00、午後 1:00～5:00)

電話 : 044-435-9494  
 FAX : 044-435-9606  
 E-mail : s-info@aed.tmg.nec.co.jp

**【営業関係お問い合わせ先】**

| 第一販売事業部                            | 第二販売事業部                      | 第三販売事業部                                        |
|------------------------------------|------------------------------|------------------------------------------------|
| 東京 (03)3798-8106, 8107, 8108       | 東京 (03)3798-8110, 8111, 8112 | 東京 (03)3798-8151, 8155, 8586, 1822, 1823, 8166 |
| 名古屋 (042)222-2378                  | 立川 (042)828-8981, 8187       | 水戸 (029)226-1702                               |
| 大阪 (06)8945-3178, 3200, 3208, 3212 | 株本 (0263)35-1682             | 広島 (083)242-6604                               |
| 仙台 (022)287-8740                   | 神岡 (054)254-4794             | 高松 (027)326-1303                               |
| 郡山 (024)823-8891                   | 金沢 (076)232-7303             | 鳥取 (0887)27-5313                               |
| 千葉 (043)238-8116                   | 松山 (089)846-6148             | 大田 (0278)46-4014                               |
|                                    |                              | 名古屋 (063)222-2170, 2190                        |
|                                    |                              | 福岡 (092)281-2808                               |

**【資料の請求先】**

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

**【インターネット電子デバイス・ニュース】**

NECエレクトロニクスデバイスの情報がインターネットでご覧になります。 URL(アドレス) <http://www.lc.nec.co.jp/>

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] 16ビットVシリーズ ユーザーズ・マニュアル 命令編 (U11301JJ5V1UMJ1 (第5版))

[お名前など] (さしつかえのない範囲で)
御社名 (学校名, その他) ( )
ご住所 ( )
お電話番号 ( )
お仕事の内容 ( )
お名前 ( )

1. ご評価 (各欄に○をご記入ください)

Table with 6 columns: Item, Very Good, Good, Average, Bad, Very Bad. Rows include Overall Structure, Description Content, Easy to Understand, Design/Font Size, and Other.

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 [ ]

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 [ ]

4. ご意見, ご要望
[ ]

5. このドキュメントをお届けしたのは
NEC販売員, 特約店販売員, その他 ( )

ご協力ありがとうございました。
下記あてにFAXで送信いただくか、最寄りの販売員にコピーをお渡しく下さい。

日本電気 (株) NECエレクトロニクス
半導体テクニカルホットライン
FAX: (044) 435-9608

2000.6

キリトリ