

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、誤りがな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。



ユーザーズ・マニュアル

78K/0 シリーズ

命令編

78K/0 シリーズ共通

資料番号 U12326JJ4V0UM00 (第4版)

発行年月 August 2001 N CP(K)

© NEC Corporation 1993

[メモ]

目次要約

第1章	メモリ空間	...	12
第2章	レジスタ	...	15
第3章	アドレッシング	...	21
第4章	命令セット	...	34
第5章	命令の説明	...	49
付 録		...	126

CMOSデバイスの一般的注意事項

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。

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_{DD}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。

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。

IEBusは、日本電気株式会社の商標です。

注意：I²Cバス・インタフェースを使用される場合には、カスタム・コードをご発注いただく時に、事前にその旨ご申告下さい。申告に基づき、以下の特典が受けられます。

日本電気株式会社のI²Cバス対応部品をご購入いただくことにより、これらの部品をI²Cシステムに使用する実施権がフィリップス社I²C特許に基づき許諾されることとなります。ただし、これらのI²Cシステムはフィリップス社によって設定されたI²C標準規格に合致しているものとします。

Purchase of NEC I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

本製品のうち、外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当するものについては、日本国外に輸出する際に、同法に基づき日本国政府の輸出許可が必要です。

- 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回路、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、これら回路・ソフトウェア・情報をお客様の機器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	78K/0シリーズ共通の情報以外は、削除（製品個別の情報については、各製品のユーザーズ・マニュアルを参照）

本文欄外の★印は、本版で改訂された主な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お寄せください。

はじめに

- 対象者** このマニュアルは、78K/0シリーズ製品の機能を理解し、その応用システムを設計するユーザのエンジニアを対象としています。
- 目的** このマニュアルは、78K/0シリーズ製品の持つ各種命令機能を理解していただくことを目的とします。
- 構成** このマニュアルは、大きく分けて次の内容で構成しています。
CPU機能
命令セット
命令の説明
- 読み方** このマニュアルを読むにあたっては、電気、論理回路およびマイクロコンピュータの一通りの知識を必要とします。

二モニックが分かっている、命令機能の詳細を確認するとき
付録B, C 命令索引を利用してください。

二モニックは分からないが、大体の機能が分かっている命令を確認するとき
第4章 命令セットでその二モニックを調べ、そのあと**第5章 命令の説明**で機能を調べてください。

一通り78K/0シリーズ製品の各種命令を理解しようとするとき
目次に従って読んでください。

78K/0シリーズ製品のハードウェア機能について知りたいとき
別冊のユーザーズ・マニュアル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
- 注 :本文中につけた注の説明
- 注意 :特に気をつけていただきたい内容
- 備考 :本文の補足説明
- 数の表記 : 2進数...××××Bまたは××××
10進数...××××
16進数...××××H

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。

★ **78K/0シリーズ共通資料**

資料名		和文資料番号	英文資料番号
ユーザーズ・マニュアル 命令編		このマニュアル	U12326E
アプリケーション・ノート ^注	基礎編	U12704J	U12704E
	基礎編	U10121J	U10121E
	基礎編	U10182J	U10182E

注 サブシリーズによっては，対象製品に含まれていない場合があります。

注意 上記関連資料は予告なしに内容を変更することがあります。設計などには必ず最新の資料をご使用ください。

目 次

第1章	メモリ空間	...	12
1.1	メモリ空間	...	12
1.2	内部プログラム・メモリ（内部ROM）空間	...	12
1.3	ベクタ・テーブル領域	...	12
1.4	CALLT命令テーブル領域	...	12
1.5	CALLF命令エントリ領域	...	12
1.6	内部データ・メモリ（内部RAM）空間	...	13
1.7	特殊機能レジスタ（SFR：Special Function Register）領域	...	14
1.8	外部メモリ空間	...	14
1.9	IEBus™レジスタ領域	...	14
第2章	レジスタ	...	15
2.1	制御レジスタ	...	15
2.1.1	プログラム・カウンタ（PC）	...	15
2.1.2	プログラム・ステータス・ワード（PSW）	...	15
2.1.3	スタック・ポインタ（SP）	...	17
2.2	汎用レジスタ	...	18
2.3	特殊機能レジスタ（SFR）	...	20
第3章	アドレッシング	...	21
3.1	命令アドレスのアドレッシング	...	21
3.1.1	レラティブ・アドレッシング	...	21
3.1.2	イミーディエト・アドレッシング	...	22
3.1.3	テーブル・インダイレクト・アドレッシング	...	23
3.1.4	レジスタ・アドレッシング	...	24
3.2	オペランド・アドレスのアドレッシング	...	25
3.2.1	インプライド・アドレッシング	...	25
3.2.2	レジスタ・アドレッシング	...	26
3.2.3	ダイレクト・アドレッシング	...	27
3.2.4	ショート・ダイレクト・アドレッシング	...	28
3.2.5	特殊機能レジスタ（SFR）アドレッシング	...	30
3.2.6	レジスタ・インダイレクト・アドレッシング	...	31
3.2.7	ベースト・アドレッシング	...	32
3.2.8	ベースト・インデクスト・アドレッシング	...	33
3.2.9	スタック・アドレッシング	...	33

第4章 命令セット ... 34

- 4.1 オペレーション ... 35
 - 4.1.1 オペランドの表現形式と記述方法 ... 35
 - 4.1.2 オペレーション欄の説明 ... 36
 - 4.1.3 フラグ動作欄の説明 ... 36
 - 4.1.4 クロック数の説明 ... 37
 - 4.1.5 アドレッシング別命令一覧 ... 37
- 4.2 命令コード ... 41
 - 4.2.1 命令コード表の説明 ... 41
 - 4.2.2 命令コード一覧 ... 42

第5章 命令の説明 ... 49

- 5.1 8ビット・データ転送命令 ... 51
- 5.2 16ビット・データ転送命令 ... 54
- 5.3 8ビット演算命令 ... 57
- 5.4 16ビット演算命令 ... 66
- 5.5 乗除算命令 ... 70
- 5.6 増減命令 ... 73
- 5.7 ローテート命令 ... 78
- 5.8 BCD補正命令 ... 85
- 5.9 ビット操作命令 ... 88
- 5.10 コール・リターン命令 ... 96
- 5.11 スタック操作命令 ... 104
- 5.12 無条件分岐命令 ... 108
- 5.13 条件付き分岐命令 ... 110
- 5.14 CPU制御命令 ... 119

付録A 改版履歴 ... 126

付録B 命令索引（ニモニック：機能別） ... 127

付録C 命令索引（ニモニック：アルファベット順） ... 129

図 の 目 次

図番号	タイトル, ページ
2 - 1	プログラム・カウンタの構成 ... 15
2 - 2	プログラム・ステータス・ワードの構成 ... 15
2 - 3	スタック・ポインタの構成 ... 17
2 - 4	スタック・メモリへ退避されるデータ ... 17
2 - 5	スタック・メモリから復帰されるデータ ... 17
2 - 6	汎用レジスタの構成 ... 19

表 の 目 次

表番号	タイトル, ページ
2 - 1	汎用レジスタの絶対アドレス対照表 ... 18
4 - 1	オペランドの表現形式と記述方法 ... 35

第 1 章 メモリ空間

1.1 メモリ空間

78K/0シリーズの製品は、内蔵するメモリの容量などによってプログラム・メモリのマッピングが異なります。メモリ・マップのアドレス領域の詳細については、**各製品のユーザーズ・マニュアル**を参照してください。

1.2 内部プログラム・メモリ（内部ROM）空間

★ 78K/0シリーズの製品は、アドレス空間にそれぞれROMを内蔵しており、プログラムやテーブル・データなどを格納します。通常、プログラム・カウンタ（PC）でアドレスされます。内部ROM空間については、**各製品のユーザーズ・マニュアル**を参照してください。

1.3 ベクタ・テーブル領域

★ 0000H-003FHの64バイトの領域はベクタ・テーブル領域として予約されています。ベクタ・テーブル領域には、 $\overline{\text{RESET}}$ 入力、各割り込み要求発生により分岐するときのプログラム・スタート・アドレスを格納しておきます。16ビット・アドレスのうち下位8ビットが偶数アドレスに、上位8ビットが奇数アドレスに格納されます。ベクタ・テーブル領域については、**各製品のユーザーズ・マニュアル**を参照してください。

1.4 CALLT命令テーブル領域

0040H-007FHの64バイトの領域には、1バイト・コール命令（CALLT）のサブルーチン・エントリ・アドレスを格納できます。

1.5 CALLF命令エントリ領域

0800H-0FFFHの領域は、2バイト・コール命令（CALLF）で直接サブルーチン・コールできます。

1.6 内部データ・メモリ（内部RAM）空間

- 78K/0シリーズの製品は、次に示すRAMを内蔵しています。内蔵しているRAMについては、**各製品のユーザーズ・マニュアル**を参照してください。

（1）内部高速RAM

78K/0シリーズの製品は、それぞれ内部高速RAMを内蔵しています。

この領域のうち、FEE0H-FEFFHの32バイトの領域には、8ビット・レジスタ8個を1バンクとする汎用レジスタが4バンク割り付けられています。

また、内部高速RAMはスタック・メモリとしても使用できます。

（2）バッファRAM

- ★ 78K/0シリーズには、バッファRAMが割り付けられている製品があります。バッファRAMは、シリアル・インタフェース・チャンネル1（自動送受信機能付き3線式シリアルI/Oモード）の送信/受信データを格納するために使用します。自動送受信機能付き3線式シリアルI/Oモードで使用しない場合は、バッファRAMは通常のRAMとしても使用できます。

（3）VFD表示用RAM

- ★ 78K/0シリーズには、VFD表示用RAMが割り付けられている製品があります。VFD表示用RAMは通常のRAMとしても使用できます。

（4）内部拡張RAM

- ★ 78K/0シリーズには、内部拡張RAMが割り付けられている製品があります。

（5）LCD表示用RAM

- ★ 78K/0シリーズには、LCD表示用RAMが割り付けられている製品があります。LCD表示用RAMは通常のRAMとしても使用できます。

1.7 特殊機能レジスタ(SFR : Special Function Register)領域

FF00H-FFFFHの領域には、オンチップ周辺ハードウェアの特殊機能レジスタ(SFR)が割り付けられています。

- ★ 特殊機能レジスタについては、**各製品のユーザーズ・マニュアル**を参照してください。

注意 この領域内で、SFRが割り付けられていないアドレスをアクセスしないでください。誤ってアクセスすると、CPUがデッド・ロック状態になることがあります。

1.8 外部メモリ空間

メモリ拡張モード・レジスタの設定によりアクセスが可能な外部メモリ空間です。プログラム、テーブル・データなどの格納、および周辺デバイスを割り付けることができます。

- ★ 外部メモリ空間を使用できる製品については、**各製品のユーザーズ・マニュアル**を参照してください。

1.9 IEBus™ レジスタ領域

- ★ IEBusレジスタ領域には、IEBusコントローラの制御に使用するIEBusレジスタが割り付けられています。IEBusコントローラを内蔵する製品については、**各製品のユーザーズ・マニュアル**を参照してください。

第2章 レジスタ

2.1 制御レジスタ

プログラム・シーケンス、ステータス、スタック・メモリの制御など専用の機能を持ったレジスタです。制御レジスタには、プログラム・カウンタ、プログラム・ステータス・ワード、スタック・ポインタがあります。

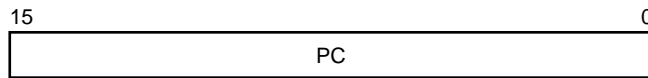
2.1.1 プログラム・カウンタ (PC)

プログラム・カウンタは、次に実行するプログラムのアドレス情報を保持する16ビット・レジスタです。

通常動作時には、フェッチする命令のバイト数に応じて、自動的にインクリメントされます。分岐命令実行時には、イミディエイト・データやレジスタの内容がセットされます。

$\overline{\text{RESET}}$ 入力により、0000Hと0001H番地のリセット・ベクタ・テーブルの値がプログラム・カウンタにセットされます。

図2 - 1 プログラム・カウンタの構成



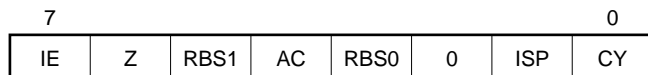
2.1.2 プログラム・ステータス・ワード (PSW)

プログラム・ステータス・ワードは、命令の実行によってセット、リセットされる各種フラグで構成される8ビット・レジスタです。

プログラム・ステータス・ワードの内容は、割り込み要求発生時およびPUSH PSW命令の実行時に自動的にスタックされ、RETB, RETI命令およびPOP PSW命令の実行時に自動的に復帰されます。

$\overline{\text{RESET}}$ 入力により、02Hになります。

図2 - 2 プログラム・ステータス・ワードの構成



(1) 割り込み許可フラグ (IE)

CPUの割り込み要求受け付け動作を制御するフラグです。

IE = 0のときは割り込み禁止 (DI) 状態となり、ノンマスクابل割り込み以外の割り込みはすべて禁止されます。

IE = 1のときは割り込み許可 (EI) 状態となります。このとき割り込み要求の受け付けは、インサースビス・プライオリティ・フラグ (ISP)、各割り込み要因に対する割り込みマスク・フラグおよび優先順位指定フラグにより制御されます。

このフラグは、DI命令の実行または割り込み要求の受け付けでリセット (0) され、EI命令の実行によりセット (1) されます。

(2) ゼロ・フラグ (Z)

演算結果がゼロのときセット (1) され、それ以外のときにリセット (0) されるフラグです。

(3) レジスタ・バンク選択フラグ (RBS0, RBS1)

4個のレジスタ・バンクのうちの1つを選択する2ビットのフラグです。

SBL RBN命令の実行によって選択されたレジスタ・バンクを示す2ビットの情報が格納されています。

(4) 補助キャリー・フラグ (AC)

演算結果が、ビット3からキャリーがあったとき、またはビット3へのポローがあったときセット (1) され、それ以外のときリセット (0) されるフラグです。

(5) インサースビス・プライオリティ・フラグ (ISP)

受け付け可能なマスクابل・ベクタ割り込みの優先順位を管理するフラグです。ISP = 0のときは優先順位指定フラグ・レジスタ (PR) で低位に指定されたベクタ割り込み要求は受け付け禁止となります。なお、実際に割り込み要求が受け付けられるかどうかは、割り込み許可フラグ (IE) の状態により制御されます。

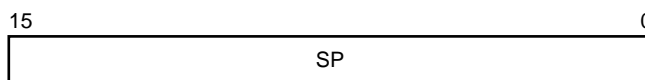
(6) キャリー・フラグ (CY)

加減算命令実行時のオーバフロー、アンダフローを記憶するフラグです。また、ローテート命令実行時はシフト・アウトされた値を記憶し、ビット演算命令実行時には、ビット・アキュムレータとして機能します。

2.1.3 スタック・ポインタ (SP)

メモリのスタック領域の先頭アドレスを保持する16ビットのレジスタです。スタック領域としては内部高速RAM領域のみ設定可能です。

図2 - 3 スタック・ポインタの構成



スタック・メモリへの書き込み（退避）動作に先立ってデクリメントされ、スタック・メモリからの読み取り（復帰）動作のあとインクリメントされます。

各スタック動作によって退避／復帰されるデータは図2 - 4，2 - 5のようになります。

注意 SPの内容はRESET入力により、不定になりますので、必ず命令実行前にイニシャライズしてください。

図2 - 4 スタック・メモリへ退避されるデータ

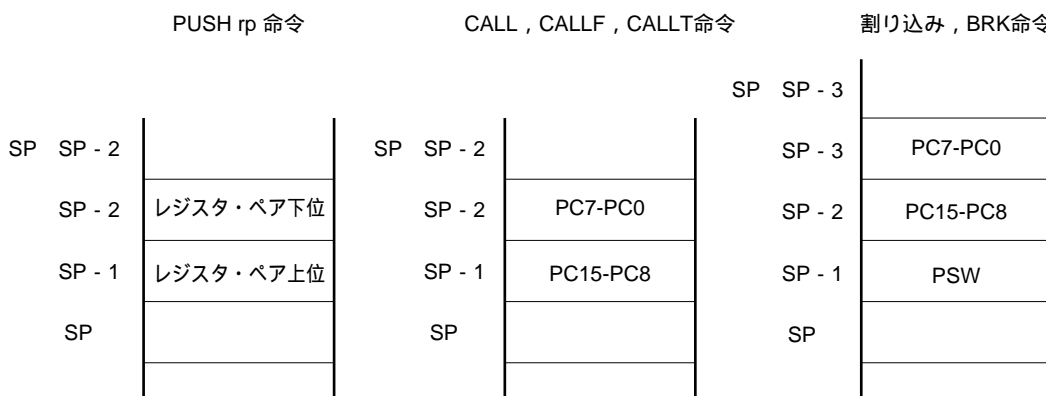
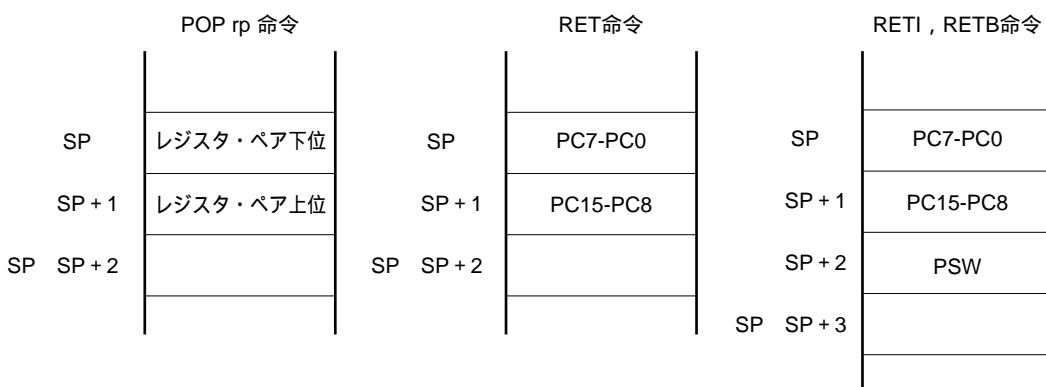


図2 - 5 スタック・メモリから復帰されるデータ



2.2 汎用レジスタ

汎用レジスタは、データ・メモリの特定番地（FEE0H-FEFFH）にマッピングされており、8ビット・レジスタ8個（X, A, C, B, E, D, L, H）を1バンクとして4バンクのレジスタで構成されています。

各レジスタは、それぞれ8ビット・レジスタとして使用できるほか、2個の8ビット・レジスタをペアとして16ビット・レジスタとしても使用できます（AX, BC, DE, HL）。

また、機能名称（X, A, C, B, E, D, L, H, AX, BC, DE, HL）のほか、絶対名称（R0-R7, RP0-RP3）でも記述できます。

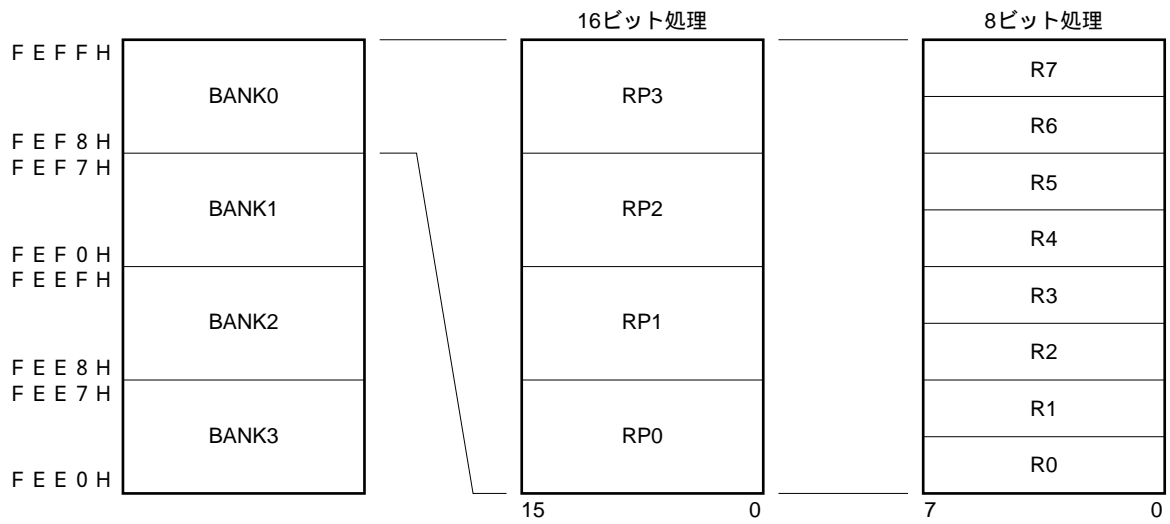
命令実行時に使用するレジスタ・バンクは、CPU制御命令（SEL R_n）によって設定します。4レジスタ・バンク構成になっていますので、通常処理で使用するレジスタと割り込み時で使用するレジスタをバンクごとに切り替えることにより、効率のよいプログラムを作成できます。

表2 - 1 汎用レジスタの絶対アドレス対照表

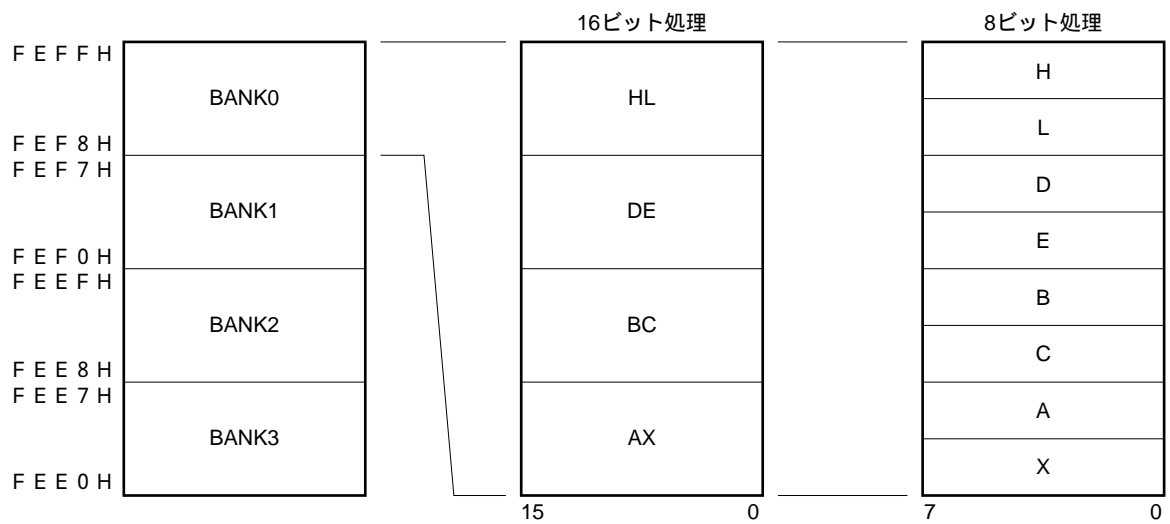
バンク名	レジスタ		絶対アドレス	バンク名	レジスタ		絶対アドレス
	機能名称	絶対名称			機能名称	絶対名称	
BANK0	H	R7	FEFFH	BANK2	H	R7	FEEFH
	L	R6	FEFEH		L	R6	FEEEH
	D	R5	FEFDH		D	R5	FEEDH
	E	R4	FEFCH		E	R4	FEECH
	B	R3	FEFBH		B	R3	FEEBH
	C	R2	FEFAH		C	R2	FEEAH
	A	R1	FEF9H		A	R1	FEE9H
	X	R0	FEF8H		X	R0	FEE8H
BANK1	H	R7	FEF7H	BANK3	H	R7	FEE7H
	L	R6	FEF6H		L	R6	FEE6H
	D	R5	FEF5H		D	R5	FEE5H
	E	R4	FEF4H		E	R4	FEE4H
	B	R3	FEF3H		B	R3	FEE3H
	C	R2	FEF2H		C	R2	FEE2H
	A	R1	FEF1H		A	R1	FEE1H
	X	R0	FEF0H		X	R0	FEE0H

図2 - 6 汎用レジスタの構成

(a) 絶対名称



(b) 機能名称



2.3 特殊機能レジスタ (SFR)

特殊機能レジスタは、汎用レジスタとは異なり、それぞれ特別な機能を持つレジスタです。

FF00H-FFFFHの256バイトの空間に割り付けられています。

特殊機能レジスタは、演算命令、転送命令、ビット操作命令などにより、汎用レジスタと同じように操作できます。操作可能なビット単位 (1 , 8 , 16) は、各特殊機能レジスタで異なります。

各操作ビット単位ごとの指定方法を次に示します。

- ・ 1 ビット操作

1 ビット操作命令のオペランド (sfr.bit) にアセンブラで予約されている略号を記述します。アドレスでも指定できます。

- ・ 8 ビット操作

8 ビット操作命令のオペランド (sfr) にアセンブラで予約されている略号を記述します。アドレスでも指定できます。

- ・ 16ビット操作

16ビット操作命令のオペランド (sfrp) にアセンブラで予約されている略号を記述します。アドレスを指定するときは偶数アドレスを記述してください。

特殊機能レジスタについては、**各製品のユーザーズ・マニュアル**を参照してください。

注意 この領域でSFRの割り付けられていないアドレスをアクセスしないでください。誤ってアクセスすると、CPUがデッド・ロック状態になることがあります。

第3章 アドレッシング

3.1 命令アドレスのアドレッシング

命令アドレスは、プログラム・カウンタ（PC）の内容によって決定されます。PCの内容は、通常、命令を1つ実行するごとにフェッチする命令のバイト数に応じて自動的にインクリメント（1バイトに対して+1）されます。しかし、分岐を伴う命令を実行する際には、次に示すようなアドレッシングにより分岐先アドレス情報がPCにセットされて分岐します（各命令についての詳細は第5章 命令の説明を参照してください）。

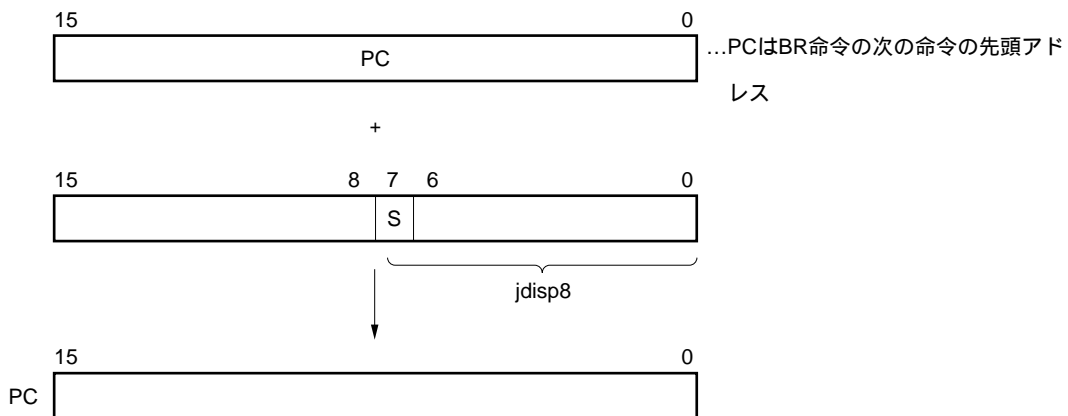
3.1.1 レラティブ・アドレッシング

【機能】

次に続く命令の先頭アドレスに命令コードの8ビット・イミューディエト・データ（ディスプレイメント値：jdisp8）を加算した値が、プログラム・カウンタ（PC）に転送されて分岐します。ディスプレイメント値は、符号付きの2の補数データ（-128～+127）として扱われ、ビット7が符号ビットとなります。つまり、レラティブ・アドレッシングでは、次に続く命令の先頭アドレスから相対的に-128～+127の範囲に分岐するということです。

BR \$addr16命令および条件付き分岐命令を実行する際に行われます。

【図解】



S=0のとき、 は全ビット0

S=1のとき、 は全ビット1

3.1.2 イミディエト・アドレッシング

【機能】

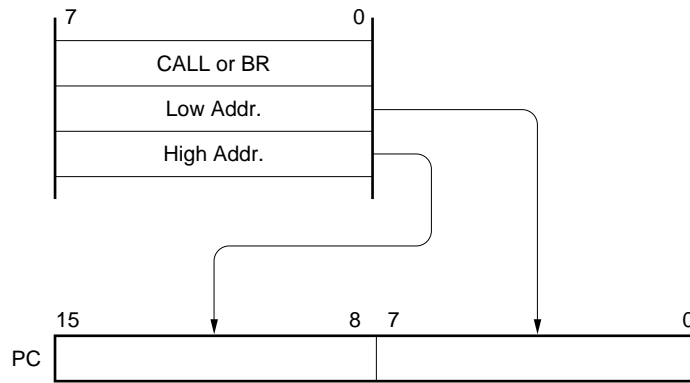
命令語中のイミディエト・データがプログラム・カウンタ（PC）に転送され、分岐します。

CALL !addr16, BR !addr16, CALLF !addr11命令を実行する際に行われます。

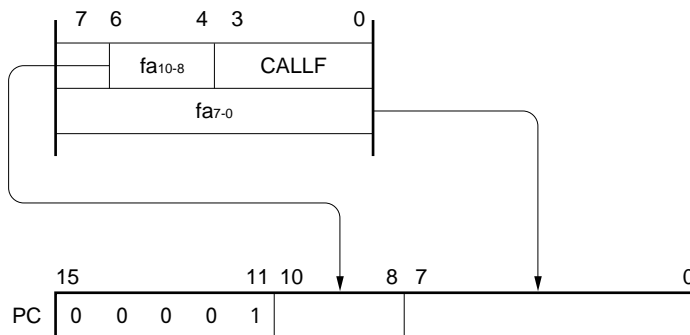
CALL !addr16, BR !addr16命令は、全メモリ空間に分岐できます。CALLF !addr11命令は、0800H-0FFFHの領域に分岐します。

【図解】

CALL !addr16, BR !addr16命令の場合



CALLF !addr11命令の場合



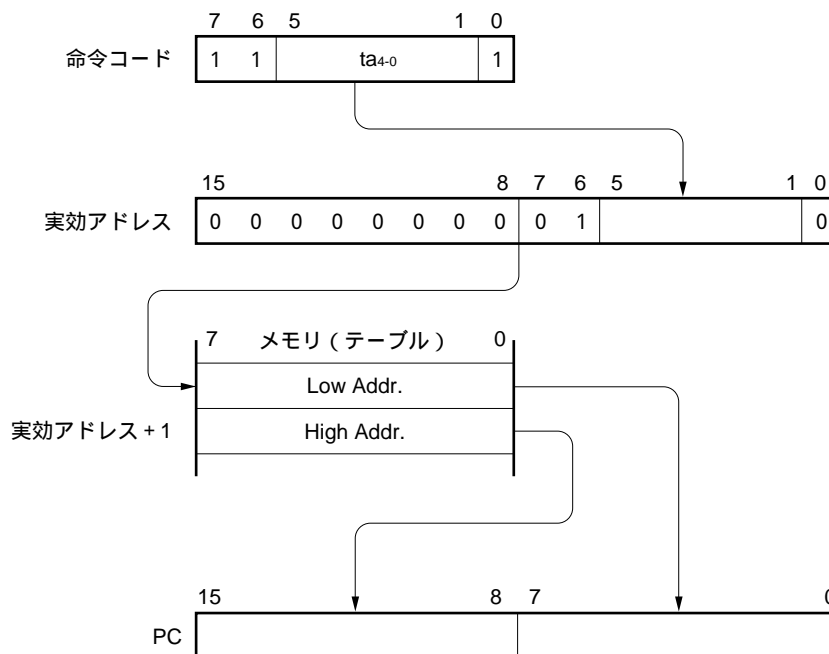
3.1.3 テーブル・インダイレクト・アドレッシング

【機能】

命令コードのビット1からビット5のイミディエイト・データによりアドレスされる特定ロケーションのテーブルの内容（分岐先アドレス）がプログラム・カウンタ（PC）に転送され、分岐します。

CALLT [addr5] 命令を実行する際にテーブル・インダイレクト・アドレッシングが行われます。この命令では、40H-7FHのメモリ・テーブルに格納されたアドレスを参照し、全メモリ空間に分岐できます。

【図解】

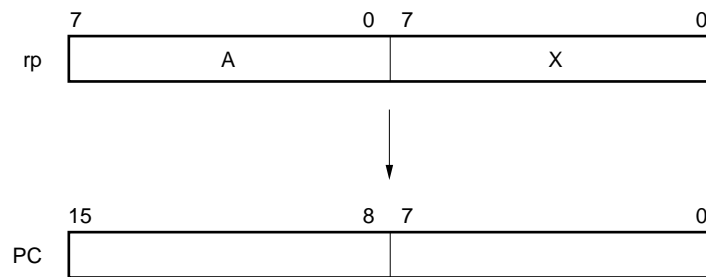


3.1.4 レジスタ・アドレッシング

【機能】

命令語によって指定されるレジスタ・ペア (AX) の内容がプログラム・カウンタ (PC) に転送され、分岐します。

BR AX命令を実行する際に行われます。

【図解】

3.2 オペランド・アドレスのアドレッシング

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指定する方法（アドレッシング）として次に示すいくつかの方法があります。

3.2.1 インプライド・アドレッシング

【機能】

汎用レジスタの領域にあるアキュムレータ（A, AX）として機能するレジスタを自動的にアドレス指定するアドレッシングです。

78K/0シリーズの命令語中でインプライド・アドレッシングを使用する命令は次のとおりです。

命 令	インプライド・アドレッシングで指定されるレジスタ
MULU	被乗数としてAレジスタ，積が格納されるレジスタとしてAXレジスタ
DIVUW	被除数および商を格納するレジスタとしてAXレジスタ
ADJBA/ADJBS	10進補正の対象となる数値を格納するレジスタとしてAレジスタ
ROR4/ROL4	ディジット・ローテートの対象となるディジット・データを格納するレジスタとしてAレジスタ

【オペランド形式】

命令によって自動的に使用できるため、特定のオペランド形式を持ちません。

【記述例】

MULU Xの場合

8ビット×8ビットの乗算命令において、AレジスタとXレジスタの積をAXに格納する。ここで、A, AXレジスタがインプライド・アドレッシングで指定されている。

3.2.2 レジスタ・アドレッシング

【機能】

オペランドとして汎用レジスタをアクセスするアドレッシングです。アクセスされる汎用レジスタは、レジスタ・バンク選択フラグ (RBS0, RBS1) および、命令コード中のレジスタ指定コード (Rn, RPn) により指定されます。

レジスタ・アドレッシングは、次に示すオペランド形式を持つ命令を実行する際に行われ、8ビット・レジスタを指定する場合は命令コード中の3ビットにより8本中の1本を指定します。

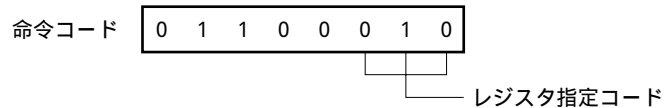
【オペランド形式】

表現形式	記述方法
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

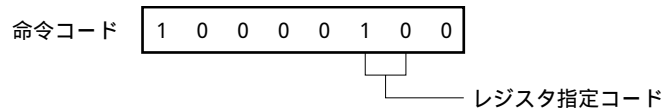
r, rpは、機能名称 (X, A, C, B, E, D, L, H, AX, BC, DE, HL) のほかに絶対名称 (R0-R7, RP0-RP3) で記述できます。

【記述例】

MOV A, C ; rにCレジスタを選択する場合



INCW DE ; rpにDEレジスタ・ペアを選択する場合



3.2.3 ダイレクト・アドレッシング

【機能】

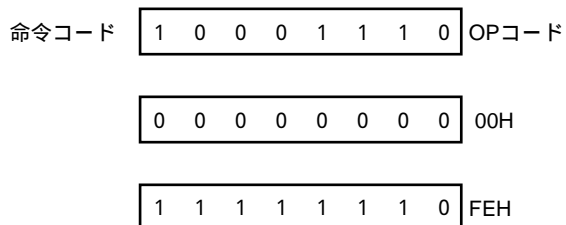
命令語中のイミディエト・データが示すメモリを直接アドレスするアドレッシングです。

【オペランド形式】

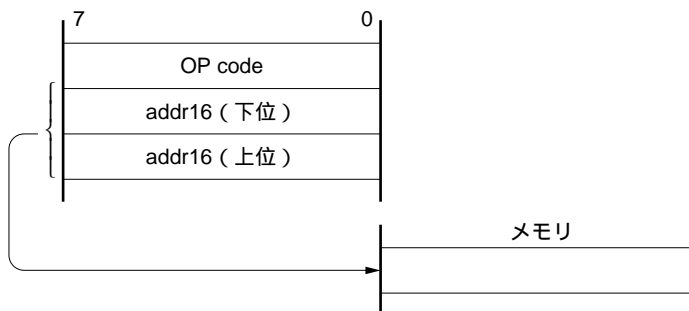
表現形式	記述方法
addr16	ラベルまたは16ビット・イミディエト・データ

【記述例】

MOV A, !FE00H ; !addr16をFE00Hとする場合



【図解】



3.2.4 ショート・ダイレクト・アドレッシング

【機能】

命令語中の8ビット・データで、固定空間の操作対象メモリを直接アドレスするアドレッシングです。

このアドレッシングが適用される固定空間とは、FE20H-FF1FHの256バイト空間です。FE20H-FEFFFHには内部高速RAMが、FF00H-FF1FHには特殊機能レジスタ（SFR）がマッピングされています。

ショート・ダイレクト・アドレッシングが適用されるSFR領域（FF00H-FF1FH）は、全SFR領域の一部です。この領域には、プログラム上でひんばんにアクセスされるポートや、タイマ/イベント・カウンタのコンペア・レジスタ、キャプチャ・レジスタがマッピングされており、短いバイト数、短いクロック数でこれらのSFRを操作できます。

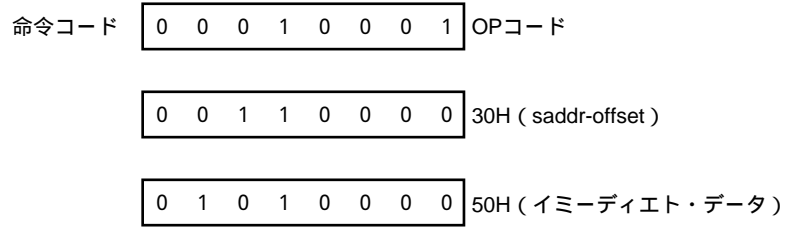
実効アドレスのビット8には、8ビット・イミディエト・データが20H-FFHの場合は0になり、00H-1FHの場合は1になります。次ページの【図解】を参照してください。

【オペランド形式】

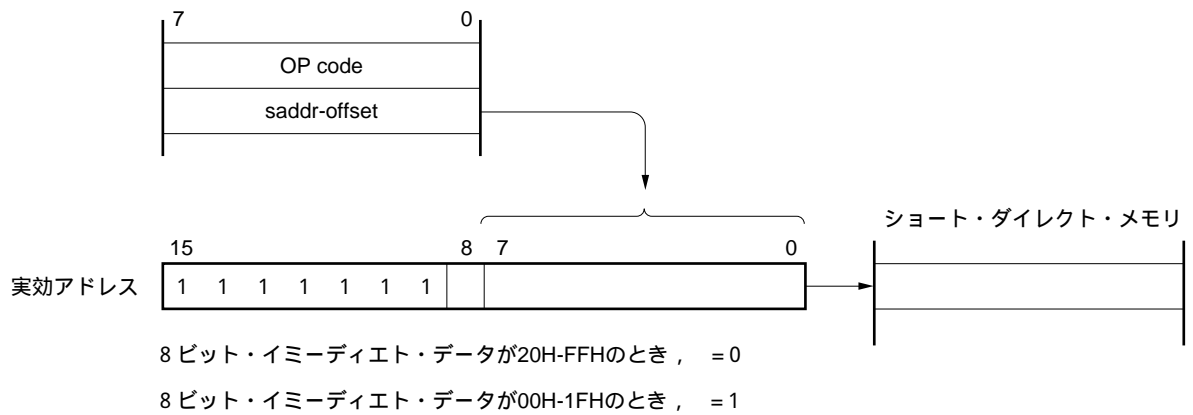
表現形式	記述方法
saddr	レーベルまたはFE20H-FF1FHのイミディエト・データ
saddrp	レーベルまたはFE20H-FF1FHのイミディエト・データ（偶数アドレスのみ）

【記述例】

MOV FE30H, #50H ; saddrをFE30H, イミディエト・データを50Hとする場合



【図解】



3.2.5 特殊機能レジスタ (SFR) アドレッシング

【機能】

命令語中の 8 ビット・イミディエト・データでメモリ・マッピングされている特殊機能レジスタ (SFR) をアドレスするアドレッシングです。

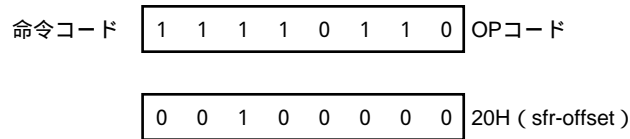
このアドレッシングが適用されるのはFF00H-FFCFH, FFE0H-FFFFHの240バイト空間です。ただし, FF00H-FF1FHにマッピングされているSFRは, ショート・ダイレクト・アドレッシングでもアクセスできます。

【オペランド形式】

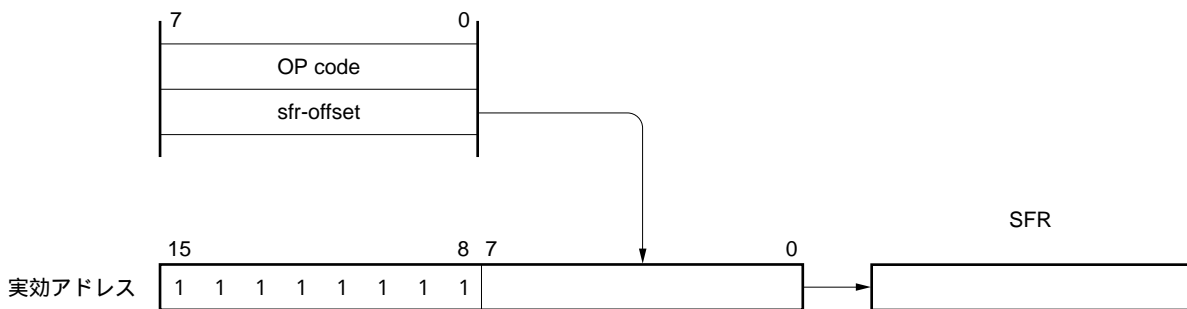
表現形式	記述方法
sfr	特殊機能レジスタ名
sfrp	16ビット操作可能な特殊機能レジスタ名 (偶数アドレスのみ)

【記述例】

MOV PM0, A ; sfrにPM0を選択する場合



【図解】



3.2.6 レジスタ・インダイレクト・アドレッシング

【機能】

オペランドとして指定されるレジスタ・ペアの内容でメモリをアドレスするアドレッシングです。アクセスされるレジスタ・ペアは、レジスタ・バンク選択フラグ (RBS0, RBS1) および、命令コード中のレジスタ・ペア指定コードにより指定されます。

【オペランド形式】

表現形式	記述方法
-	[DE], [HL]

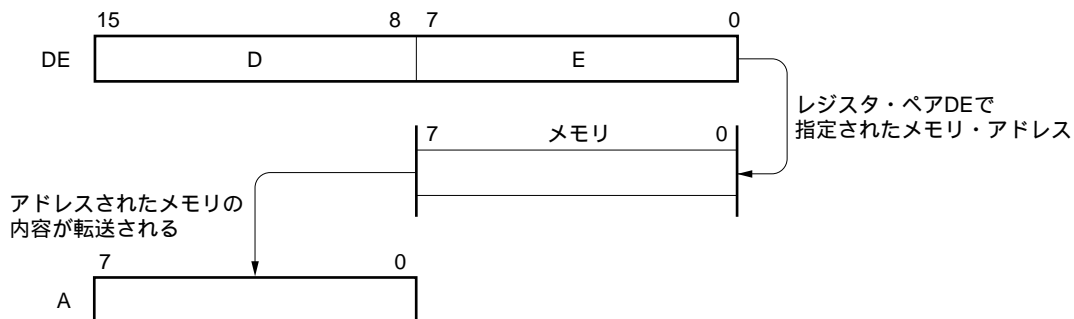
【記述例】

MOV A, [DE] ; レジスタ・ペア [DE] を選択する場合

命令コード

1	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

【図解】



3.2.7 ベース・アドレッシング

【機能】

HLレジスタ・ペアをベース・レジスタとし、この内容に8ビットのイミディエト・データを加算した結果でメモリをアドレスするアドレッシングです。アクセスされるHLレジスタ・ペアは、レジスタ・バンク選択フラグ (RBS0, RBS1) で指定されるレジスタ・バンク中のものです。加算は、オフセット・データを正の数として16ビットに拡張して行います。16ビット目からの桁上がりは無視します。すべてのメモリ空間に対してアドレッシングできます。

【オペランド形式】

表現形式	記述方法
-	[HL + byte]

【記述例】

MOV A, [HL + 10H] ; byteを10Hとする場合

命令コード

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

3.2.8 ベース・インデクスト・アドレッシング

【機能】

HLレジスタ・ペアをベース・レジスタとし、この内容に命令語中で指定されるBレジスタまたはCレジスタの内容を加算した結果でメモリをアドレスするアドレッシングです。アクセスされるHL, B, Cレジスタは、レジスタ・バンク選択フラグ (RBS0, RBS1) で指定されるレジスタ・バンク中のレジスタです。加算は、BレジスタまたはCレジスタの内容を正の数として16ビットに拡張して行います。16ビット目からの桁上りは無視します。すべてのメモリ空間に対してアドレッシングできます。

【オペランド形式】

表現形式	記述方法
-	[HL + B], [HL + C]

【記述例】

MOV A, [HL + B] の場合

命令コード

1 0 1 0 1 0 1 1

3.2.9 スタック・アドレッシング

【機能】

スタック・ポインタ (SP) の内容により、スタック領域を間接的にアドレスするアドレッシングです。

PUSH, POP, サブルーチン・コール, リターン命令の実行時および割り込み要求発生によるレジスタの退避 / 復帰時に自動的に用いられます。

スタック・アドレッシングは、内部高速RAM領域のみアクセスできます。

【記述例】

PUSH DEの場合

命令コード

1 0 1 1 0 1 0 1

第4章 命令セット

この章では、78K/0シリーズの命令セットを一覧表にして示します。

78K/0シリーズ製品の命令は、すべて共通です。

4.1 オペレーション

- ★ オペレーション一覧については、各製品のユーザーズ・マニュアルを参照してください。

4.1.1 オペランドの表現形式と記述方法

各命令のオペランド欄には、その命令のオペランド表現形式に対する記述方法に従ってオペランドを記述しています（詳細は、アセンブラ仕様による）。記述方法の中で複数個あるものは、それらの要素の1つを選択します。大文字で書かれた英字および#、!、\$、[]の記号はキー・ワードであり、そのまま記述します。記号の説明は、次のとおりです。

- ・ # : イミューディアット・データ指定
- ・ ! : 絶対アドレス指定
- ・ \$: 相対アドレス指定
- ・ [] : 間接アドレス指定

イミューディアット・データのときは、適当な数値またはレーベルを記述します。レーベルで記述する際も#、!、\$、[]記号は必ず記述してください。

また、オペランドのレジスタの記述形式、r、rpには、機能名称（X、A、Cなど）、絶対名称（下表の中のカッコ内の名称、R0、R1、R2など）のいずれの形式でも記述可能です。

表4-1 オペランドの表現形式と記述方法

表現形式	記述方法
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	特殊機能レジスタ略号 ^注
sfrp	特殊機能レジスタ略号 (16ビット操作可能なレジスタの偶数アドレスのみ) ^注
saddr	FE20H-FF1FH イミューディアット・データまたはレーベル
saddrp	FE20H-FF1FH イミューディアット・データまたはレーベル (偶数アドレスのみ)
addr16	0000H-FFFFH イミューディアット・データまたはレーベル (16ビット・データ転送命令時は偶数アドレスのみ)
addr11	0800H-0FFFH イミューディアット・データまたはレーベル
addr5	0040H-007FH イミューディアット・データまたはレーベル (偶数アドレスのみ)
word	16ビット・イミューディアット・データまたはレーベル
byte	8ビット・イミューディアット・データまたはレーベル
bit	3ビット・イミューディアット・データまたはレーベル
RBn	RB0-RB3

注 FFD0H-FFDFHは、アドレスできません。

備考 特殊機能レジスタの略号は各製品のユーザーズ・マニュアルを参照してください。

4.1.2 オペレーション欄の説明

A	: Aレジスタ; 8ビット・アキュムレータ
X	: Xレジスタ
B	: Bレジスタ
C	: Cレジスタ
D	: Dレジスタ
E	: Eレジスタ
H	: Hレジスタ
L	: Lレジスタ
AX	: AXレジスタ・ペア; 16ビット・アキュムレータ
BC	: BCレジスタ・ペア
DE	: DEレジスタ・ペア
HL	: HLレジスタ・ペア
PC	: プログラム・カウンタ
SP	: スタック・ポインタ
PSW	: プログラム・ステータス・ワード
CY	: キャリー・フラグ
AC	: 補助キャリー・フラグ
Z	: ゼロ・フラグ
RBS	: レジスタ・バンク選択フラグ
IE	: 割り込み要求許可フラグ
NMIS	: ノンマスクブル割り込み処理中フラグ
()	: ()内のアドレスまたはレジスタの内容で示されるメモリの内容
\times_H, \times_L	: 16ビット・レジスタの上位8ビット, 下位8ビット
	: 論理積 (AND)
	: 論理和 (OR)
∇	: 排他的論理和 (exclusive OR)
	: 反転データ
addr16	: 16ビット・イミディエト・データまたはレーベル
jdisp8	: 符号付き8ビット・データ (ディスプレイメント値)

4.1.3 フラグ動作欄の説明

(ブランク)	: 変化なし
0	: 0にクリアされる
1	: 1にセットされる
\times	: 結果に従ってセット/クリアされる
R	: 以前に退避した値がストアされる

4.1.4 クロック数の説明

命令の1クロックはプロセッサ・クロック・コントロール・レジスタ (PCC) で選択したCPUクロック (fCPU) の1クロック分です。

4.1.5 アドレッシング別命令一覧

(1) 8ビット命令

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

第2オペランド 第1オペランド	#byte	A	r ^注	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte] [HL+B] [HL+C]	\$saddr16	1	なし	
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP			ROR ROL RORC ROLC		
r	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC	
B, C											DBNZ			
sfr	MOV	MOV												
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC	
!addr16		MOV												
PSW	MOV	MOV											PUSH POP	

注 r=Aは除く。

第2オペランド 第1オペランド	#byte	A	r ^注	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$addr16	1	なし
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

注 r=Aは除く。

(2) 16ビット命令

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

第2オペランド 第1オペランド	#word	AX	rp ^注	sfrp	saddrp	!addr16	SP	なし
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW ^注						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

注 rp=BC, DE, HLのときのみ。

(3) ビット操作命令

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

第2オペランド 第1オペランド	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	なし
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) コール命令/分岐命令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

第2オペランド 第1オペランド	AX	!addr16	!addr11	[addr5]	\$addr16
基本命令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
複合命令					BT BF BTCLR DBNZ

(5) その他の命令

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

4.2 命令コード

4.2.1 命令コード表の説明

R ₂	R ₁	R ₀	reg	
0	0	0	R0	X
0	0	1	R1	A
0	1	0	R2	C
0	1	1	R3	B
1	0	0	R4	E
1	0	1	R5	D
1	1	0	R6	L
1	1	1	R7	H

P ₁	P ₀	reg-pair	
0	0	RP0	AX
0	1	RP1	BC
1	0	RP2	DE
1	1	RP3	HL

RB ₁	RB ₀	reg-bank
0	0	RB0
0	1	RB1
1	0	RB2
1	1	RB3

Bn	: bitに対応するイミディエト・データ
Data	: byteに対応する8ビット・イミディエト・データ
Low/High byte	: wordに対応する16ビット・イミディエト・データ
Saddr-offset	: saddrに対応する16ビット・アドレスの下位8ビット・オフセット・データ
Sfr-offset	: sfrの16ビット・アドレスの下位8ビット・オフセット・データ
Low/High addr	: addr16に対応する16ビット・イミディエト・データ
jdisp	: 次の命令の先頭アドレスと分岐アドレスとの相対アドレス距離の符号付き2の補数データ(8ビット)
fa ₁₀₋₀	: addr11に対応するイミディエト・データの11ビット
ta ₄₋₀	: addr5に対応するイミディエト・データの5ビット

4.2.2 命令コード一覧

命令群	ニモニック	オペランド	命令コード			
			B1	B2	B3	B4
8ビット・データ転送	MOV	r, #byte	1 0 1 0 0 R ₂ R ₁ R ₀	Data		
		saddr, #byte	0 0 0 1 0 0 0 1	Saddr-offset	Data	
		sfr, #byte	0 0 0 1 0 0 1 1	Sfr-offset	Data	
		A, r 注	0 1 1 0 0 R ₂ R ₁ R ₀			
		r, A 注	0 1 1 1 0 R ₂ R ₁ R ₀			
		A, saddr	1 1 1 1 0 0 0 0	Saddr-offset		
		saddr, A	1 1 1 1 0 0 1 0	Saddr-offset		
		A, sfr	1 1 1 1 0 1 0 0	Sfr-offset		
		sfr, A	1 1 1 1 0 1 1 0	Sfr-offset		
		A, !addr16	1 0 0 0 1 1 1 0	Low addr	High addr	
		!addr16, A	1 0 0 1 1 1 1 0	Low addr	High addr	
		PSW, #byte	0 0 0 1 0 0 0 1	0 0 0 1 1 1 1 0	Data	
		A, PSW	1 1 1 1 0 0 0 0	0 0 0 1 1 1 1 0		
		PSW, A	1 1 1 1 0 0 1 0	0 0 0 1 1 1 1 0		
		A, [DE]	1 0 0 0 0 1 0 1			
		[DE], A	1 0 0 1 0 1 0 1			
		A, [HL]	1 0 0 0 0 1 1 1			
		[HL], A	1 0 0 1 0 1 1 1			
		A, [HL + byte]	1 0 1 0 1 1 1 0	Data		
		[HL + byte], A	1 0 1 1 1 1 1 0	Data		
		A, [HL + B]	1 0 1 0 1 0 1 1			
		[HL + B], A	1 0 1 1 1 0 1 1			
		A, [HL + C]	1 0 1 0 1 0 1 0			
		[HL + C], A	1 0 1 1 1 0 1 0			
	XCH	A, r 注	0 0 1 1 0 R ₂ R ₁ R ₀			
		A, saddr	1 0 0 0 0 0 1 1	Saddr-offset		
		A, sfr	1 0 0 1 0 0 1 1	Sfr-offset		
		A, !addr16	1 1 0 0 1 1 1 0	Low addr	High addr	
		A, [DE]	0 0 0 0 0 1 0 1			
		A, [HL]	0 0 0 0 0 1 1 1			
		A, [HL + byte]	1 1 0 1 1 1 1 0	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	1 0 0 0 1 0 1 1		
A, [HL + C]	0 0 1 1 0 0 0 1	1 0 0 0 1 0 1 0				

注 r=Aを除く。

命令群	二モニック	オペランド	命令コード			
			B1	B2	B3	B4
16ビット・データ転送	MOVW	rp, #word	0 0 0 1 0 P ₁ P ₀ 0	Low byte	High byte	
		saddrp, #word	1 1 1 0 1 1 1 0	Saddr-offset	Low byte	High byte
		sfrp, #word	1 1 1 1 1 1 1 0	Sfr-offset	Low byte	High byte
		AX, saddrp	1 0 0 0 1 0 0 1	Saddr-offset		
		saddrp, AX	1 0 0 1 1 0 0 1	Saddr-offset		
		AX, sfrp	1 0 1 0 1 0 0 1	Sfr-offset		
		sfrp, AX	1 0 1 1 1 0 0 1	Sfr-offset		
		AX, rp 注1	1 1 0 0 0 P ₁ P ₀ 0			
		rp, AX 注1	1 1 0 1 0 P ₁ P ₀ 0			
		AX, !addr16	0 0 0 0 0 0 1 0	Low addr	High addr	
	!addr16, AX	0 0 0 0 0 0 1 1	Low addr	High addr		
XCHW	AX, rp 注1	1 1 1 0 0 P ₁ P ₀ 0				
8ビット演算	ADD	A, #byte	0 0 0 0 1 1 0 1	Data		
		saddr, #byte	1 0 0 0 1 0 0 0	Saddr-offset	Data	
		A, r 注2	0 1 1 0 0 0 0 1	0 0 0 0 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 0 0 0 0 R ₂ R ₁ R ₀		
		A, saddr	0 0 0 0 1 1 1 0	Saddr-offset		
		A, !addr16	0 0 0 0 1 0 0 0	Low addr	High addr	
		A, [HL]	0 0 0 0 1 1 1 1			
		A, [HL + byte]	0 0 0 0 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 0 0 0 1 0 1 1		
	A, [HL + C]	0 0 1 1 0 0 0 1	0 0 0 0 1 0 1 0			
	ADDC	A, #byte	0 0 1 0 1 1 0 1	Data		
		saddr, #byte	1 0 1 0 1 0 0 0	Saddr-offset	Data	
		A, r 注2	0 1 1 0 0 0 0 1	0 0 1 0 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 0 1 0 0 R ₂ R ₁ R ₀		
		A, saddr	0 0 1 0 1 1 1 0	Saddr-offset		
		A, !addr16	0 0 1 0 1 0 0 0	Low addr	High addr	
		A, [HL]	0 0 1 0 1 1 1 1			
		A, [HL + byte]	0 0 1 0 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 0 1 0 1 0 1 1		
A, [HL + C]		0 0 1 1 0 0 0 1	0 0 1 0 1 0 1 0			

注1 . rp=BC, DE, HLのときのみ。

2 . r=Aを除く。

命令群	二モニック	オペランド	命令コード			
			B1	B2	B3	B4
8ビット演算	SUB	A, #byte	0 0 0 1 1 1 0 1	Data		
		saddr, #byte	1 0 0 1 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 0 0 1 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 0 0 1 0 R ₂ R ₁ R ₀		
		A, saddr	0 0 0 1 1 1 1 0	Saddr-offset		
		A, laddr16	0 0 0 1 1 0 0 0	Low addr	High addr	
		A, [HL]	0 0 0 1 1 1 1 1			
		A, [HL + byte]	0 0 0 1 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 0 0 1 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 0 0 1 1 0 1 0		
	SUBC	A, #byte	0 0 1 1 1 1 0 1	Data		
		saddr, #byte	1 0 1 1 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 0 1 1 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 0 1 1 0 R ₂ R ₁ R ₀		
		A, saddr	0 0 1 1 1 1 1 0	Saddr-offset		
		A, laddr16	0 0 1 1 1 0 0 0	Low addr	High addr	
		A, [HL]	0 0 1 1 1 1 1 1			
		A, [HL + byte]	0 0 1 1 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 0 1 1 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 0 1 1 1 0 1 0		
	AND	A, #byte	0 1 0 1 1 1 0 1	Data		
		saddr, #byte	1 1 0 1 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 1 0 1 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 1 0 1 0 R ₂ R ₁ R ₀		
		A, saddr	0 1 0 1 1 1 1 0	Saddr-offset		
		A, laddr16	0 1 0 1 1 0 0 0	Low addr	High addr	
		A, [HL]	0 1 0 1 1 1 1 1			
		A, [HL + byte]	0 1 0 1 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 1 0 1 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 1 0 1 1 0 1 0		

注 r=Aを除く。

命令群	二モニック	オペランド	命令コード			
			B1	B2	B3	B4
8ビット演算	OR	A, #byte	0 1 1 0 1 1 0 1	Data		
		saddr, #byte	1 1 1 0 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 1 1 0 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 1 1 0 0 R ₂ R ₁ R ₀		
		A, saddr	0 1 1 0 1 1 1 0	Saddr-offset		
		A, !addr16	0 1 1 0 1 0 0 0	Low addr	High addr	
		A, [HL]	0 1 1 0 1 1 1 1			
		A, [HL + byte]	0 1 1 0 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 1 1 0 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 1 1 0 1 0 1 0		
	XOR	A, #byte	0 1 1 1 1 1 0 1	Data		
		saddr, #byte	1 1 1 1 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 1 1 1 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 1 1 1 1 0 R ₂ R ₁ R ₀		
		A, saddr	0 1 1 1 1 1 1 0	Saddr-offset		
		A, !addr16	0 1 1 1 1 0 0 0	Low addr	High addr	
		A, [HL]	0 1 1 1 1 1 1 1			
		A, [HL + byte]	0 1 1 1 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 1 1 1 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 1 1 1 1 0 1 0		
	CMP	A, #byte	0 1 0 0 1 1 0 1	Data		
		saddr, #byte	1 1 0 0 1 0 0 0	Saddr-offset	Data	
		A, r 注	0 1 1 0 0 0 0 1	0 1 0 0 1 R ₂ R ₁ R ₀		
		r, A	0 1 1 0 0 0 0 1	0 1 0 0 0 R ₂ R ₁ R ₀		
		A, saddr	0 1 0 0 1 1 1 0	Saddr-offset		
		A, !addr16	0 1 0 0 1 0 0 0	Low addr	High addr	
		A, [HL]	0 1 0 0 1 1 1 1			
		A, [HL + byte]	0 1 0 0 1 0 0 1	Data		
		A, [HL + B]	0 0 1 1 0 0 0 1	0 1 0 0 1 0 1 1		
		A, [HL + C]	0 0 1 1 0 0 0 1	0 1 0 0 1 0 1 0		

注 r=Aを除く。

命令群	二モニック	オペランド	命令コード			
			B1	B2	B3	B4
16 演 ビット 算	ADDW	AX, #word	1 1 0 0 1 0 1 0	Low byte	High byte	
	SUBW	AX, #word	1 1 0 1 1 0 1 0	Low byte	High byte	
	CMPW	AX, #word	1 1 1 0 1 0 1 0	Low byte	High byte	
乗 除 算	MULU	X	0 0 1 1 0 0 0 1	1 0 0 0 1 0 0 0		
	DIVUW	C	0 0 1 1 0 0 0 1	1 0 0 0 0 0 1 0		
増 減	INC	r	0 1 0 0 0 R ₂ R ₁ R ₀			
		saddr	1 0 0 0 0 0 0 1	Saddr-offset		
	DEC	r	0 1 0 1 0 R ₂ R ₁ R ₀			
		saddr	1 0 0 1 0 0 0 1	Saddr-offset		
	INCW	rp	1 0 0 0 0 P ₁ P ₀ 0			
	DECW	rp	1 0 0 1 0 P ₁ P ₀ 0			
ロ ー テ ー ト	ROR	A, 1	0 0 1 0 0 1 0 0			
	ROL	A, 1	0 0 1 0 0 1 1 0			
	RORC	A, 1	0 0 1 0 0 1 0 1			
	ROLC	A, 1	0 0 1 0 0 1 1 1			
	ROR4	[HL]	0 0 1 1 0 0 0 1	1 0 0 1 0 0 0 0		
	ROL4	[HL]	0 0 1 1 0 0 0 1	1 0 0 0 0 0 0 0		
B 補 C 正 D 正	ADJBA		0 1 1 0 0 0 0 1	1 0 0 0 0 0 0 0		
	ADJBS		0 1 1 0 0 0 0 1	1 0 0 1 0 0 0 0		
ビ ッ ト 操 作	MOV1	CY, saddr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 0 0	Saddr-offset	
		CY, sfr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 0 0	Sfr-offset	
		CY, A.bit	0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 1 0 0		
		CY, PSW.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 0 0	0 0 0 1 1 1 1 0	
		CY, [HL] .bit	0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 0 0		
		saddr.bit, CY	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 0 1	Saddr-offset	
		sfr.bit, CY	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 0 0 1	Sfr-offset	
		A.bit, CY	0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 0 0 1		
		PSW.bit, CY	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 0 1	0 0 0 1 1 1 1 0	
		[HL] .bit, CY	0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 0 0 1		
	AND1	CY, saddr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 0 1	Saddr-offset	
		CY, sfr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 0 1	Sfr-offset	
		CY, A.bit	0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 1 0 1		
		CY, PSW.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 0 1	0 0 0 1 1 1 1 0	
CY, [HL] .bit		0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 0 1			

命令群	二モニック	オペランド	命令コード				
			B1	B2	B3	B4	
ビット操作	OR1	CY, saddr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 0	Saddr-offset		
		CY, sfr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 1 0	Sfr-offset		
		CY, A.bit	0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 1 1 0			
		CY, PSW.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 0	0 0 0 1 1 1 1 0		
		CY, [HL] .bit	0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 1 0			
	XOR1	CY, saddr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 1	Saddr-offset		
		CY, sfr.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 1 1	Sfr-offset		
		CY, A.bit	0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 1 1 1			
		CY, PSW.bit	0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 1	0 0 0 1 1 1 1 0		
		CY, [HL] .bit	0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 1 1			
	SET1	saddr.bit		0 B ₂ B ₁ B ₀ 1 0 1 0	Saddr-offset		
		sfr.bit		0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 0 1 0	Sfr-offset	
		A.bit		0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 0 1 0		
		PSW.bit		0 B ₂ B ₁ B ₀ 1 0 1 0	0 0 0 1 1 1 1 0		
		[HL] .bit		0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 0 1 0		
	CLR1	saddr.bit		0 B ₂ B ₁ B ₀ 1 0 1 1	Saddr-offset		
		sfr.bit		0 1 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 0 1 1	Sfr-offset	
		A.bit		0 1 1 0 0 0 0 1	1 B ₂ B ₁ B ₀ 1 0 1 1		
		PSW.bit		0 B ₂ B ₁ B ₀ 1 0 1 1	0 0 0 1 1 1 1 0		
		[HL] .bit		0 1 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 0 1 1		
SET1	CY	0 0 1 0 0 0 0 0					
CLR1	CY	0 0 1 0 0 0 0 1					
NOT1	CY	0 0 0 0 0 0 0 1					
コール・リターン	CALL	laddr16	1 0 0 1 1 0 1 0	Low addr	High addr		
	CALLF	laddr11	0 fa ₁₀₋₈ 1 1 1 0 0	fa ₇₋₀			
	CALLT	[addr5]	1 1 ta ₄₋₀ 1				
	BRK		1 0 1 1 1 1 1 1				
	RET		1 0 1 0 1 1 1 1				
	RETB		1 0 0 1 1 1 1 1				
	RETI		1 0 0 0 1 1 1 1				
スタック操作	PUSH	PSW	0 0 1 0 0 0 1 0				
		rp	1 0 1 1 0 P ₁ P ₀ 1				
	POP	PSW	0 0 1 0 0 0 1 1				
		rp	1 0 1 1 0 P ₁ P ₀ 0				
	MOVW	SP, #word	1 1 1 0 1 1 1 0	0 0 0 1 1 1 1 0	Low byte	High byte	
		SP, AX	1 0 0 1 1 0 0 1	0 0 0 1 1 1 1 0			
AX, SP		1 0 0 0 1 0 0 1	0 0 0 1 1 1 1 0				

命令群	二モニック	オペランド	命令コード				
			B1	B2	B3	B4	
無条件分岐	BR	!addr16	1 0 0 1 1 0 1 1	Low addr	High addr		
		\$addr16	1 1 1 1 1 0 1 0	jdisp			
		AX	0 0 1 1 0 0 0 1	1 0 0 1 1 0 0 0			
条件付き分岐	BC	\$addr16	1 0 0 0 1 1 0 1	jdisp			
	BNC	\$addr16	1 0 0 1 1 1 0 1	jdisp			
	BZ	\$addr16	1 0 1 0 1 1 0 1	jdisp			
	BNZ	\$addr16	1 0 1 1 1 1 0 1	jdisp			
	BT	saddr.bit, \$addr16	1 B ₂ B ₁ B ₀ 1 1 1 0 0	Saddr-offset	jdisp		
		sfr.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 0	Sfr-offset	jdisp	
		A.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 1 0	jdisp		
		PSW.bit, \$addr16	1 B ₂ B ₁ B ₀ 1 1 1 0 0	0 0 0 1 1 1 1 0	jdisp		
		[HL] .bit, \$addr16	0 0 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 1 0	jdisp		
	BF	saddr.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 1 1	Saddr-offset	jdisp	
		sfr.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 1 1	Sfr-offset	jdisp	
		A.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 1 1	jdisp		
		PSW.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 1 1	0 0 0 1 1 1 1 0	jdisp	
		[HL] .bit, \$addr16	0 0 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 1 1	jdisp		
	BTCLR	saddr.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 0 1	Saddr-offset	jdisp	
		sfr.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 1 0 1	Sfr-offset	jdisp	
		A.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 1 1 0 1	jdisp		
		PSW.bit, \$addr16	0 0 1 1 0 0 0 1	0 B ₂ B ₁ B ₀ 0 0 0 1	0 0 0 1 1 1 1 0	jdisp	
		[HL] .bit, \$addr16	0 0 1 1 0 0 0 1	1 B ₂ B ₁ B ₀ 0 1 0 1	jdisp		
	DBNZ	B, \$addr16	1 0 0 0 1 0 1 1	jdisp			
		C, \$addr16	1 0 0 0 1 0 1 0	jdisp			
		saddr, \$addr16	0 0 0 0 0 1 0 0	Saddr-offset	jdisp		
	CPU制御	SEL	RBn	0 1 1 0 0 0 0 1	1 1 RB ₁ RB ₀ 0 0 0		
		NOP		0 0 0 0 0 0 0 0			
		EI		0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0		
		DI		0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0		
		HALT		0 1 1 1 0 0 0 1	0 0 0 1 0 0 0 0		
STOP			0 1 1 1 0 0 0 1	0 0 0 0 0 0 0 0			

第5章 命令の説明

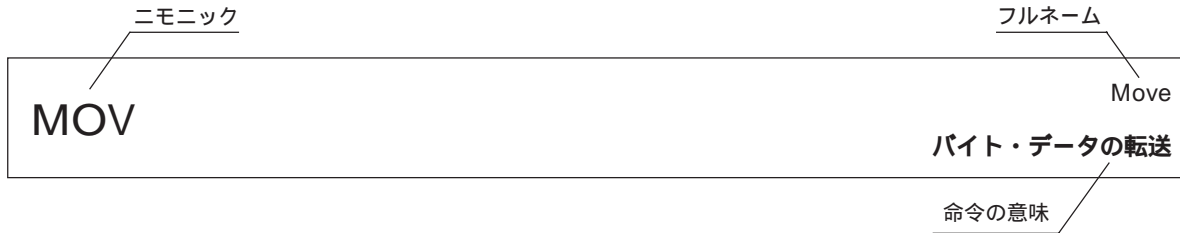
この章では、78K/0シリーズ製品の命令の説明をします。各命令は、二モニック単位で、複数のオペランドをまとめて説明します。

命令の説明の基本構成を次ページに示します。

- ★ なお、命令のバイト数は各製品のユーザーズ・マニュアルを、命令コードは第4章 命令セットを参照してください。

78K/0シリーズ製品の命令は、すべて共通です。

記述例



【命令形式】 MOV dst, src : 命令の基本記述形式を示します。

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

【オペランド】 : 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、4.1 オペレーションを参照してください。

ニモニック	オペランド (dst, src)	ニモニック	オペランド (dst, src)
MOV	r, #byte	MOV	A, PSW
	A, saddr		[HL], A
	saddr, A		A, [HL + byte]
	PSW, #byte		[HL + C], A

【フラグ】 : 命令実行により変化するフラグの動作を示します。

各フラグの動作記号を凡例に示します。

Z	AC	CY

凡例

記号	解説
ブランク	変化なし
0	0にクリアされる
1	1にセットされる
X	結果に従ってセットまたはクリアされる
R	以前に退避した値がリストアされる

【説明】 : 命令のオペレーションの詳細を解説します。

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。

【記述例】

MOV A, #4DH ; Aレジスタに4DHを転送

5.1 8ビット・データ転送命令

8ビット・データ転送命令には、次の命令があります。

MOV ... 52

XCH ... 53

MOV

Move
バイト・データの転送

【命令形式】 MOV dst, src

【オペレーション】 dst src

【オペランド】

二モニック	オペランド (dst, src)
MOV	r, #byte
	saddr, #byte
	sfr, #byte
	A, r 注
	r, A 注
	A, saddr
	saddr, A
	A, sfr
	sfr, A
	A, laddr16
	laddr16, A
	PSW, #byte

二モニック	オペランド (dst, src)
MOV	A, PSW
	PSW, A
	A, [DE]
	[DE], A
	A, [HL]
	[HL], A
	A, [HL + byte]
	[HL + byte], A
	A, [HL + B]
	[HL + B], A
	A, [HL + C]
	[HL + C], A

注 r=Aを除く。

【フラグ】

PSW, #byteとPSW, A

のオペランドの場合

Z	AC	CY
x	x	x

左記以外

Z	AC	CY

【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) 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。

MOV PSW, #byte命令、MOV PSW, A命令と次に続く命令の間では、すべての割り込みを受け付けません。

【記述例】

MOV A, #4DH ; Aレジスタに4DHを転送

XCH

Exchange
バイト・データの交換

【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)
XCH	A, r 注
	A, saddr
	A, sfr
	A, !addr16
	A, [DE]

ニモニック	オペランド (dst, src)
XCH	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY

【説明】

第1オペランドと第2オペランドの内容を交換します。

【記述例】

XCH A, FEBCH ; Aレジスタの内容とFEBCH番地の内容を交換

5.2 16ビット・データ転送命令

16ビット・データ転送命令には、次の命令があります。

MOVW ... 55

XCHW ... 56

MOVW

Move Word
ワード・データの転送

【命令形式】 MOVW dst, src

【オペレーション】 dst src

【オペランド】

二モニック	オペランド (dst, src)
MOVW	rp, #word
	saddrp, #word
	sfrp, #word
	AX, saddrp
	saddrp, AX
	AX, sfrp

二モニック	オペランド (dst, src)	
MOVW	sfrp, AX	
	AX, rp	注
	rp, AX	注
	AX, !addr16	
	!addr16, AX	

注 rp=BC, DE, HLのときのみ。

【フラグ】

Z	AC	CY

【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) 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。

【記述例】

MOVW AX, HL ; HLレジスタの内容をAXレジスタへ転送

【注意】

偶数アドレスのみ指定できます。奇数アドレスは指定できません。

XCHW

Exchange Word
ワード・データの交換

【命令形式】 XCHW 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)
XCHW	AX, rp 注

注 rp=BC, DE, HLのときのみ。

【フラグ】

Z	AC	CY

【説明】

第1オペランドと第2オペランドの内容を交換します。

【記述例】

XCHW AX, BC ; AXレジスタとBCレジスタの内容を交換

5.3 8ビット演算命令

8ビット演算命令には、次の命令があります。

ADD ...	58
ADDC ...	59
SUB ...	60
SUBC ...	61
AND ...	62
OR ...	63
XOR ...	64
CMP ...	65

ADD

Add

バイト・データの加算

【命令形式】 ADD dst, src

【オペレーション】 dst, CY dst + src

【オペランド】

二モニック	オペランド (dst, src)
ADD	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
ADD	A, !addr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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) を加算し、その結果をCYフラグとデスティネーション・オペランド (dst) へ格納します。

加算の結果、dstが0になった場合、Zフラグがセット(1)、その他の場合はZフラグはクリア(0)されます。

加算の結果、ビット7からのキャリーが発生した場合は、CYフラグはセット(1)、その他の場合はCYフラグはクリア(0)されます。

加算の結果、ビット3からビット4へのキャリーが発生した場合は、ACフラグはセット(1)、その他の場合はACフラグはクリア(0)されます。

【記述例】

ADD CR10, #56H ; CR10レジスタに56Hを加算し、結果をCR10レジスタへ格納

ADDC

Add with Carry

キャリーを含むバイト・データの加算

【命令形式】 ADDC dst, src

【オペレーション】 dst, CY dst + src + CY

【オペランド】

二モニック	オペランド (dst, src)
ADDC	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
ADDC	A, laddr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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) とCYフラグを加算して、結果をデスティネーション・オペランド (dst) とCYフラグに格納します。

CYフラグは最下位ビットへ加算されます。

この命令は、主として複数バイトの加算を行うときに使用します。

加算の結果、dstが0になった場合、Zフラグがセット(1)、その他の場合はZフラグはクリア(0)されます。

加算の結果、ビット7からのキャリーが発生した場合は、CYフラグはセット(1)、その他の場合はCYフラグはクリア(0)されます。

加算の結果、ビット3からビット4へのキャリーが発生した場合は、ACフラグはセット(1)、その他の場合はACフラグがクリア(0)されます。

【記述例】

ADDC A, [HL + B]; Aレジスタと(HLレジスタ + (Bレジスタ))番地の内容とCYフラグを加算し、結果をAレジスタに格納

SUB

Subtract
バイト・データの減算

【命令形式】 SUB dst, src

【オペレーション】 dst, CY dst - src

【オペランド】

二モニック	オペランド (dst, src)
SUB	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
SUB	A, !addr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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、結果をデスティネーション・オペランド (dst) とCYフラグへ格納します。

ソース・オペランド (src) とデスティネーション・オペランド (dst) を同一のものとするにより、デスティネーション・オペランドの0クリアが可能です。

減算の結果、dstが0なら、Zフラグはセット(1)、その他の場合はZフラグはクリア(0)されます。

減算の結果、ビット7でポローが発生した場合、CYフラグはセット(1)、その他の場合はクリア(0)されます。

減算の結果、ビット4からビット3へのポローが発生した場合、ACフラグはセット(1)、その他の場合はクリア(0)されます。

【記述例】

SUB D, A ; DレジスタからAレジスタを減算し、結果をDレジスタへ格納

SUBC

Subtract with Carry

キャリーを含むバイト・データの減算

【命令形式】 SUBC dst, src

【オペレーション】 dst, CY dst - src - CY

【オペランド】

二モニック	オペランド (dst, src)
SUBC	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
SUBC	A, laddr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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) とCYフラグを減算し、結果をデスティネーション・オペランド (dst) へ格納します。

CYフラグは最下位ビットから減算します。

この命令は、主として複数バイトの減算を行うときに使用します。

減算の結果、dstが0ならZフラグはセット(1)、その他の場合はZフラグはクリア(0)されます。

減算の結果、ビットで7でボローが発生した場合、CYフラグはセット(1)、その他の場合はクリア(0)されます。

減算の結果、ビット4からビット3へのボローが発生した場合は、ACフラグはセット(1)、その他の場合はクリア(0)されます。

【記述例】

SUBC A, [HL]; Aレジスタから(HLレジスタ)番地の内容とCYフラグを減算し、結果をAレジスタへ格納

AND

And

バイト・データの論理積

【命令形式】 AND dst, src

【オペレーション】 dst dst src

【オペランド】

二モニック	オペランド (dst, src)
AND	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
AND	A, !addr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
×		

【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) のビットごとの論理積をとり、結果をデスティネーション・オペランド (dst) へ格納します。

論理積をとった結果、全ビットが0であればZフラグはセット (1)、その他の場合は、Zフラグはクリア (0) されます。

【記述例】

AND FEBAH, #11011100B ; FEBAHの内容と11011100Bのビットごとの論理積をとり、結果をFEBAHへ格納

OR

Or

バイト・データの論理和

【命令形式】 OR dst, src

【オペレーション】 dst dst src

【オペランド】

二モニック	オペランド (dst, src)
OR	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
OR	A, laddr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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) のビットごとの論理和をとり、結果をデスティネーション・オペランド (dst) へ格納します。

論理和をとった結果、全ビットが0であればZフラグはセット(1)、その他の場合はクリア(0)されます。

【記述例】

OR A, FE98H ; AレジスタとFE98Hのビットごとの論理和をとり、結果をAレジスタへ格納

XOR

Exclusive Or
バイト・データの排他的論理和

【命令形式】 XOR dst, src

【オペレーション】 dst dst ∨ src

【オペランド】

二モニック	オペランド (dst, src)
XOR	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
XOR	A, !addr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
×		

【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) のビットごとの排他的論理和をとり、結果をデスティネーション・オペランド (dst) へ格納します。

この命令でソース・オペランド (src) に #0FFH を選択することにより、デスティネーション・オペランド (dst) の全ビットの論理否定がとれます。

排他的論理和の結果、全ビットが0であればZフラグはセット (1)、その他の場合はクリア (0) されます。

【記述例】

XOR A, L ; AレジスタとLレジスタのビットごとの排他的論理和をとり、結果をAレジスタへ格納

CMP

Compare

バイト・データの比較

【命令形式】 CMP dst, src

【オペレーション】 dst - src

【オペランド】

二モニック	オペランド (dst, src)
CMP	A, #byte
	saddr, #byte
	A, r 注
	r, A
	A, saddr

二モニック	オペランド (dst, src)
CMP	A, laddr16
	A, [HL]
	A, [HL + byte]
	A, [HL + B]
	A, [HL + C]

注 r=Aを除く。

【フラグ】

Z	AC	CY
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にZ, AC, CYの各フラグだけを変化させます。

減算の結果、0ならZフラグはセット(1)、その他の場合はZフラグはクリア(0)されます。

減算の結果、ビット7でボローが発生した場合、CYフラグはセット(1)、その他の場合はクリア(0)されます。

減算の結果、ビット4からビット3へのボローが発生した場合、ACフラグはセット(1)、その他の場合はクリア(0)されます。

【記述例】

CMP FE38H, #38H ; FE38H番地の内容から38Hを減算し、フラグだけを変化させる (FE38H番地の内容とイミディエト・データの比較)

5.4 16ビット演算命令

16ビット演算命令には、次の命令があります。

ADDW ... 67

SUBW ... 68

CMPW ... 69

ADDW

Add Word
ワード・データの加算

【命令形式】 ADDW dst, src

【オペレーション】 dst, CY dst + src

【オペランド】

二モニック	オペランド (dst, src)
ADDW	AX, #word

【フラグ】

Z	AC	CY
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) の加算を行い、結果をデスティネーション・オペランド (dst) へ格納します。

加算の結果、dstが0になった場合、Zフラグがセット (1)、その他の場合はZフラグはクリア (0) されます。

加算の結果、ビット15からのキャリーが発生した場合は、CYフラグはセット (1)、その他の場合はCYフラグはクリア (0) されます。

加算の結果、ACフラグは不定となります。

【記述例】

ADDW AX, #ABCDH ; AXレジスタとABCDHを加算し、結果をAXレジスタへ格納

SUBW

Subtract Word
ワード・データの減算

【命令形式】 SUBW dst, src

【オペレーション】 dst, CY dst - src

【オペランド】

モニタック	オペランド (dst, src)
SUBW	AX, #word

【フラグ】

Z	AC	CY
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、結果をデスティネーション・オペランド (dst) とCYフラグへ格納します。

ソース・オペランド (src) とデスティネーション・オペランド (dst) を同一のものとするにより、デスティネーション・オペランドの0クリアが可能です。

減算の結果、dstが0ならZフラグはセット (1)、その他の場合はZフラグはクリア (0) されます。

減算の結果、ビット15でボローが発生した場合、CYフラグはセット (1)、その他の場合はクリア (0) されます。

減算の結果、ACフラグは不定となります。

【記述例】

SUBW AX, #ABCDH ; AXレジスタの内容からABCDHを減算し、結果をAXレジスタへ格納

CMPW

Compare Word
ワード・データの比較

【命令形式】 CMPW dst, src

【オペレーション】 dst - src

【オペランド】

二モニック	オペランド (dst, src)
CMPW	AX, #word

【フラグ】

Z	AC	CY
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にZ, AC, CYの各フラグだけを変化させます。

減算の結果, 0ならZフラグはセット(1), その他の場合はZフラグはクリア(0)されます。

減算の結果, ビット15でボローが発生した場合, CYフラグはセット(1), その他の場合はクリア(0)されます。

減算の結果, ACフラグは不定となります。

【記述例】

CMPW AX, #ABCDH ; AXレジスタからABCDHを減算し, フラグだけを変化させる (AXレジスタとイミディエト・データとの比較)

5.5 乗除算命令

乗除算命令には、次の命令があります。

MULU ... 71

DIVUW ... 72

MULUMultiply Unsigned
データの符号なし乗算

【命令形式】 MULU src

【オペレーション】 AX $A \times \text{src}$

【オペランド】

二モニック	オペランド (src)
MULU	X

【フラグ】

Z	AC	CY

【説明】

Aレジスタの内容とソース・オペランド (src) のデータを符号なしのデータとして乗算し、結果をAXレジスタへ格納する。

【記述例】

MULU X ; Aレジスタの内容とXレジスタの内容を乗算し、結果をAXレジスタへ格納

DIVUW

Divide Unsigned Word
ワード・データの符号なし除算

【命令形式】 DIVUW dst

【オペレーション】 AX (商), dst (余り) $AX \div dst$

【オペランド】

ニモニック	オペランド (dst)
DIVUW	C

【フラグ】

Z	AC	CY

【説明】

AXレジスタの内容をデスティネーション・オペランド (dst) の内容で除算し、商をAXレジスタに、余りをデスティネーション・オペランド (dst) へ格納します。

除算はAXレジスタおよびデスティネーション・オペランド (dst) の内容を符号なしのデータとして行われます。

ただし、デスティネーション・オペランド (dst) が0のときは、CレジスタにはXレジスタの内容が格納され、AX=0FFFFHとなります。

【記述例】

DIVUW C ; AXレジスタの内容をCレジスタの内容で除算し、商をAXレジスタへ、余りをCレジスタへ格納

5.6 増減命令

増減命令には、次の命令があります。

INC ... 74

DEC ... 75

INCW ... 76

DECW ... 77

INC

Increment

バイト・データのインクリメント

【命令形式】 INC dst

【オペレーション】 dst dst + 1

【オペランド】

モニタ	オペランド (dst)
INC	r
	saddr

【フラグ】

Z	AC	CY
x	x	

【説明】

デスティネーション・オペランド (dst) の内容を 1 だけインクリメントします。

インクリメントした結果が 0 になれば Z フラグはセット (1) , その他の場合はクリア (0) されません。

インクリメントした結果, ビット 3 からビット 4 へのキャリーがあれば, AC フラグはセット (1) , その他の場合はクリア (0) されます。

繰り返し処理のカウンタやインデクス・アドレッシングのオフセット・レジスタのインクリメントに使用することが多いため, CY フラグの内容は変化させません (複数バイトの演算時に, CY フラグの内容を保持させるため)。

【記述例】

INC B ; Bレジスタをインクリメント

DEC

Decrement

バイト・データのデクリメント

【命令形式】 DEC dst

【オペレーション】 dst dst - 1

【オペランド】

二モニック	オペランド (dst)
DEC	r
	saddr

【フラグ】

Z	AC	CY
x	x	

【説明】

デスティネーション・オペランド (dst) の内容を 1 だけデクリメントします。

デクリメントした結果が 0 であれば、Z フラグはセット (1) , その他の場合はクリア (0) されません。

デクリメントした結果がビット 4 からビット 3 へのキャリーがあれば、AC フラグはセット (1) , その他の場合はクリア (0) されます。

繰り返し処理のカウンタやインデクスト・アドレッシング時のオフセット用レジスタのデクリメントに使用することが多いため、CY フラグの内容は変化させません (複数バイトの演算時に CY フラグを保持させるため) 。

dst が B レジスタ、C レジスタ、または saddr の場合で AC、CY の各フラグを変化させたくない場合、DBNZ 命令を使用できます。

【記述例】

DEC FE92H ; FE92H 番地の内容をデクリメント

INCWIncrement Word
ワード・データのインクリメント**【命令形式】** INCW 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 (dst)
INCW	rp

【フラグ】

Z	AC	CY

【説明】

デスティネーション・オペランド (dst) の内容を 1 だけインクリメントします。

レジスタを使用するアドレッシングで、使用するレジスタ (ポインタ) のインクリメントに使用することが多いため、Z, AC, CY の各フラグを変化させません。

【記述例】

INCW HL ; HLレジスタをインクリメント

DECW

Decrement Word
ワード・データのデクリメント

【命令形式】 DECW 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 (dst)
DECW	rp

【フラグ】

Z	AC	CY

【説明】

デスティネーション・オペランド (dst) の内容を 1 だけデクリメントします。

レジスタを使用するアドレッシングで、使用するレジスタ (ポインタ) のデクリメントに使用することが多いため、Z, AC, CYの各フラグを変化させません。

【記述例】

DECW DE ; DEレジスタをデクリメント

5.7 ローテート命令

ローテート命令には、次の命令があります。

ROR ... 79

ROL ... 80

RORC ... 81

ROLC ... 82

ROR4 ... 83

ROL4 ... 84

ROR

Rotate Right

バイト・データの右方向のローテート

【命令形式】 ROR dst, cnt

【オペレーション】 (CY, dst7 dst0, dstm - 1 dstm) × 1回

【オペランド】

ニモニック	オペランド (dst, cnt)
ROR	A, 1

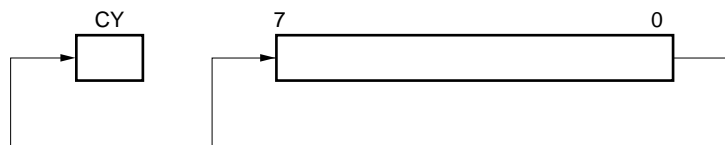
【フラグ】

Z	AC	CY
		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) の内容を, 1回だけ右方向へ回転させます。

LSB (ビット0) の内容はMSB (ビット7) へ回転されると同時にCYフラグへも転送されます。



【記述例】

ROR A, 1 ; Aレジスタの内容を右へ1ビット回転

ROL

Rotate Left

バイト・データの左方向のローテート

【命令形式】 ROL dst, cnt

【オペレーション】 (CY, dst₇, dst_{m+1} ... dst_m) × 1回

【オペランド】

ニモニック	オペランド (dst, cnt)
ROL	A, 1

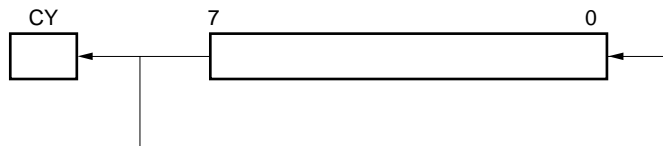
【フラグ】

Z	AC	CY
		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) の内容を1回だけ左方向へ回転させます。

MSB (ビット7) の内容は, LSB (ビット0) へ回転されると同時にCYフラグへも転送されます。



【記述例】

ROL A, 1 ; Aレジスタの内容を左へ1ビット回転

RORC

Rotate Right with Carry

キャリーを含むバイト・データの右方向のローテート

【命令形式】 RORC dst, cnt

【オペレーション】 (CY dst0, dst7 CY, dstm-1 dstm) × 1回

【オペランド】

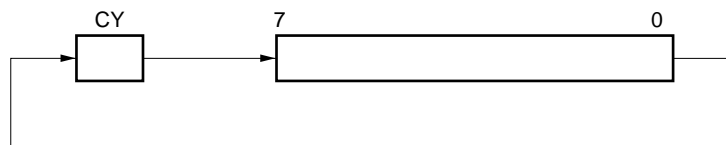
ニモニック	オペランド (dst, cnt)
RORC	A, 1

【フラグ】

Z	AC	CY
		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) の内容を、CYフラグを含め、1回だけ右方向へ回転します。



【記述例】

RORC A, 1 ; Aレジスタの内容を、CYフラグを含めて1ビット右方向へ回転

ROLC

Rotate Left with Carry

キャリーを含むバイト・データの左方向のローテート

【命令形式】 ROLC dst, cnt

【オペレーション】 (CY dst7, dst0 CY, dstm+1 dstm) × 1回

【オペランド】

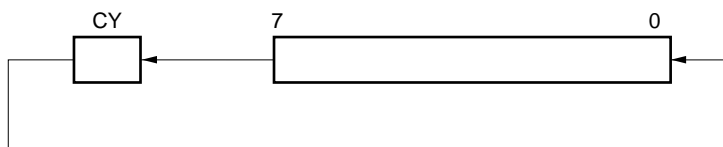
ニモニック	オペランド (dst, cnt)
ROLC	A, 1

【フラグ】

Z	AC	CY
		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) の内容を, CYフラグを含め, 1回だけ左方向へ回転させます。



【記述例】

ROLC A, 1 ; Aレジスタの内容を, CYフラグを含めて1ビット左へ回転

ROR4

Rotate Right Digit

右方向のディジット・ローテート

【命令形式】 ROR4 dst

【オペレーション】 A_{3-0} (dst) $_{3-0}$, (dst) $_{7-4}$ A_{3-0} , (dst) $_{3-0}$ (dst) $_{7-4}$

【オペランド】

二モニック	オペランド (dst)
ROR4	[HL] 注

注 オペランド [HL] は、SFR領域以外を対象としています。

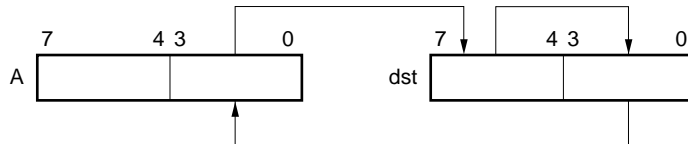
【フラグ】

Z	AC	CY

【説明】

Aレジスタの下位4ビットとデスティネーション・オペランド (dst) の2桁のディジット・データ (4ビット・データ) を右方向へ回転させます。

Aレジスタの上位4ビットは変化しません。



【記述例】

ROR4 [HL] ; AレジスタとHLレジスタで指定されるメモリの内容で右方向へディジット・ローテートする

	A				(HL)			
	7	4	3	0	7	4	3	0
実行前	1	0	1	0	1	1	0	0
実行後	1	0	1	0	0	0	1	1

ROL4

Rotate Left Digit

左方向のディジット・ローテート

【命令形式】 ROL4 dst

【オペレーション】 $A_{3-0} \leftarrow (dst)_{7-4}, (dst)_{3-0} \leftarrow A_{3-0}, (dst)_{7-4} \leftarrow (dst)_{3-0}$

【オペランド】

ニモニック	オペランド (dst)
ROL4	[HL] 注

注 オペランド [HL] は、SFR領域以外を対象としています。

【フラグ】

Z	AC	CY

【説明】

Aレジスタの下位4ビットとデスティネーション・オペランド (dst) の2桁のディジット・データ (4ビット・データ) を左方向へ回転させます。

Aレジスタの上位4ビットは変化しません。



【記述例】

ROL4 [HL] ; AレジスタとHLレジスタで指定されるメモリの内容で左方向へディジット・ローテートする

	A				(HL)			
	7	4	3	0	7	4	3	0
実行前	0001		0010		0100		1000	
実行後	0001		0100		1000		0010	

5.8 BCD補正命令

BCD補正命令には、次の命令があります。

ADJBA ... 86

ADJBS ... 87

ADJBA

Decimal Adjust Register for Addition

加算結果の10進補正

【命令形式】 ADJBA

【オペレーション】 Decimal Adjust Accumulator for Addition

【オペランド】

なし

【フラグ】

Z	AC	CY
×	×	×

【説明】

AレジスタとCYフラグ、ACフラグの内容から、AレジスタとCYフラグ、ACフラグを10進補正します。この命令は、BCD（2進化10進数）形式のデータを加算したのちに、加算結果がAレジスタに格納されている場合のみ、意味のある動作をします（その他の場合は、無意味な動作をします）。補正の方法は下表のとおりです。

補正の結果、Aレジスタの内容が0になるとZフラグがセット（1）、その他の場合は、クリア（0）されます。

条 件		オペレーション
A ₃₋₀ 9 AC = 0	A ₇₋₄ 9 and CY = 0	A A, CY 0, AC 0
	A ₇₋₄ 10 or CY = 1	A A + 01100000B, CY 1, AC 0
A ₃₋₀ 10 AC = 0	A ₇₋₄ < 9 and CY = 0	A A + 00000110B, CY 0, AC 1
	A ₇₋₄ 9 or CY = 1	A A + 01100110B, CY 1, AC 1
AC = 1	A ₇₋₄ 9 and CY = 0	A A + 00000110B, CY 0, AC 0
	A ₇₋₄ 10 or CY = 1	A A + 01100110B, CY 1, AC 0

ADJBS

Decimal Adjust Register for Subtraction

減算結果の10進補正

【命令形式】 ADJBS

【オペレーション】 Decimal Adjust Accumulator for Subtraction

【オペランド】

なし

【フラグ】

Z	AC	CY
×	×	×

【説明】

AレジスタとCYフラグ、ACフラグの内容から、AレジスタとCYフラグ、ACフラグを10進補正します。この命令は、BCD（2進化10進数）形式のデータを減算したのちに、減算結果がAレジスタに格納されている場合のみ、意味のある動作をします（その他の場合は、無意味な動作をします）。補正の方法は下表のとおりです。

補正の結果、Aレジスタの内容が0になるとZフラグがセット（1）、その他の場合はクリア（0）されます。

条 件		オペレーション
AC = 0	CY = 0	A A, CY 0, AC 0
	CY = 1	A A - 01100000B, CY 1, AC 0
AC = 1	CY = 0	A A - 00000110B, CY 0, AC 0
	CY = 1	A A - 01100110B, CY 1, AC 0

5.9 ビット操作命令

ビット操作命令には、次の命令があります。

MOV1 ... 89

AND1 ... 90

OR1 ... 91

XOR1 ... 92

SET1 ... 93

CLR1 ... 94

NOT1 ... 95

MOV1

Move Single Bit
1ビット・データの転送

【命令形式】 MOV1 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)
MOV1	CY, saddr.bit
	CY, sfr.bit
	CY, A.bit
	CY, PSW.bit
	CY, [HL].bit

ニモニック	オペランド (dst, src)
MOV1	saddr.bit, CY
	sfr.bit, CY
	A.bit, CY
	PSW.bit, CY
	[HL].bit, CY

【フラグ】

dstがCY

Z	AC	CY
		x

PSW.bit

Z	AC	CY
x	x	

左記以外

Z	AC	CY

【説明】

第1オペランドで指定されたデスティネーション・オペランド (dst) へ第2オペランドで指定されたソース・オペランド (src) のビット・データを転送します。

デスティネーション・オペランド (dst) がCY, またはPSW.bitの場合, 該当するフラグのみが変化します。

【記述例】

MOV1 P3.4, CY ; CYフラグの内容をポート3のビット4へ転送

AND1

And Single Bit

1ビット・データの論理積

【命令形式】 AND1 dst, src

【オペレーション】 dst dst src

【オペランド】

モニタック	オペランド (dst, src)
AND1	CY, saddr.bit
	CY, sfr.bit
	CY, A.bit
	CY, PSW.bit
	CY, [HL].bit

【フラグ】

Z	AC	CY
		×

【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) のビット・データとの論理積をとり、結果をデスティネーション・オペランド (dst) へ格納します。

CYフラグは、演算結果が格納されます (デスティネーション・オペランド (dst) であるため)。

【記述例】

AND1 CY, FE7FH.3 ; FE7FHのビット3とCYフラグの論理積をとり、結果をCYフラグに格納

OR1

Or Single Bit
1ビット・データの論理和

【命令形式】 OR1 dst, src

【オペレーション】 dst dst src

【オペランド】

ニモニック	オペランド (dst, src)
OR1	CY, saddr.bit
	CY, sfr.bit
	CY, A.bit
	CY, PSW.bit
	CY, [HL].bit

【フラグ】

Z	AC	CY
		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) のビット・データとの論理和をとり、結果をデスティネーション・オペランド (dst) へ格納します。

CYフラグは、演算結果が格納されます (デスティネーション・オペランド (dst) であるため)。

【記述例】

OR1 CY, P2.5 ; ポート2のビット5とCYフラグの論理和をとり、結果をCYフラグへ格納

XOR1

Exclusive Or Single Bit
1ビット・データの排他的論理和

【命令形式】 XOR1 dst, src

【オペレーション】 dst dst ∇ src

【オペランド】

モニタック	オペランド (dst, src)
XOR1	CY, saddr.bit
	CY, sfr.bit
	CY, A.bit
	CY, PSW.bit
	CY, [HL].bit

【フラグ】

Z	AC	CY
		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) のビット・データとの排他的論理和をとり、結果をデスティネーション・オペランド (dst) へ格納します。

CYフラグは、演算結果が格納されます (デスティネーション・オペランド (dst) であるため)。

【記述例】

XOR1 CY, A.7 ; Aレジスタのビット7とCYフラグの排他的論理和をとり、結果をCYフラグに格納

SET1

Set Single Bit (Carry Flag)

1ビット・データのセット

【命令形式】 SET1 dst

【オペレーション】 dst 1

【オペランド】

二モニック	オペランド (dst)
SET1	saddr.bit
	sfr.bit
	A.bit
	PSW.bit
	[HL] .bit
	CY

【フラグ】

dstがPSW.bit

Z	AC	CY
x	x	x

dstがCY

Z	AC	CY
		1

左記以外

Z	AC	CY

【説明】

デスティネーション・オペランド (dst) をセット (1) します。

デスティネーション・オペランド (dst) がCY, またはPSW.bitの場合, 該当するフラグのみがセット (1) されます。

【記述例】

SET1 FE55H.1 ; FE55Hのビット1をセット (1) する。

CLR1

Clear Single Bit (Carry Flag)

1ビット・データのクリア

【命令形式】 CLR1 dst

【オペレーション】 dst 0

【オペランド】

ニモニック	オペランド (dst)
CLR1	saddr.bit
	sfr.bit
	A.bit
	PSW.bit
	[HL] .bit
	CY

【フラグ】

dstがPSW.bit

Z	AC	CY
x	x	x

dstがCY

Z	AC	CY
		0

左記以外

Z	AC	CY

【説明】

デスティネーション・オペランド (dst) をクリア (0) します。

デスティネーション・オペランド (dst) がCY, またはPSW.bitの場合, 該当するフラグのみがクリア (0) されます。

【記述例】

CLR1 P3.7 ; ポート3のビット7をクリア (0) する。

NOT1

Not Single Bit (Carry Flag)

1ビット・データの論理否定

【命令形式】 NOT1 dst**【オペレーション】** dst $\overline{\text{dst}}$ **【オペランド】**

二モニック	オペランド (dst)
NOT1	CY

【フラグ】

Z	AC	CY
		x

【説明】

CYフラグを反転します。

【記述例】

NOT1 CY ; CYフラグを反転します。

5.10 コール・リターン命令

コール・リターン命令には、次の命令があります。

CALL	...	97
CALLF	...	98
CALLT	...	99
BRK	...	100
RET	...	101
RETI	...	102
RETB	...	103

CALL

Call

サブルーチン・コール(16ビット直接)

【命令形式】 CALL target

【オペレーション】 (SP - 1) (PC + 3) H ,
 (SP - 2) (PC + 3) L ,
 SP SP - 2 ,
 PC target

【オペランド】

ニモニック	オペランド (target)
CALL	laddr16

【フラグ】

Z	AC	CY

【説明】

16ビットの絶対アドレスまたは、レジスタ間接アドレスによるサブルーチン・コールです。

次の命令の先頭アドレス (PC + 3) をスタックに退避し、ターゲット・オペランド (target) で指定されるアドレスへ分岐します。

【記述例】

CALL !3059H ; 3059Hへサブルーチン・コールする

CALLF

Call Flag

サブルーチン・コール(11ビット直接指定)

【命令形式】 CALLF Target

【オペレーション】 (SP - 1) (PC + 2)_H ,
 (SP - 2) (PC + 2)_L ,
 SP SP - 2 ,
 PC target

【オペランド】

二モニック	オペランド (target)
CALLF	!addr11

【フラグ】

Z	AC	CY

【説明】

0800H-0FFFFH番地へのみ分岐可能なサブルーチン・コールです。

次の命令の先頭アドレス (PC + 2) をスタックに退避し、0800H-0FFFFH番地の範囲内へ分岐します。

アドレスは、下位の11ビットのみ指定します (上位5ビットは00001Bに固定)。

サブルーチンを0800H-0FFFFHへ配置し、この命令を使用することでプログラム・サイズを圧縮することが可能です。また、外部メモリにプログラムがある場合は、実行時間も高速になります。

【記述例】

CALLF !0C2AH ; 0C2AHのサブルーチン・コール

CALLT

Call Table

サブルーチン・コール(コール・テーブル参照)

【命令形式】 CALLT [addr5]

【オペレーション】 (SP - 1) (PC + 1)_H,
 (SP - 2) (PC + 1)_L,
 SP SP - 2,
 PC_H (00000000,addr5 + 1)
 PC_L (00000000,addr5)

【オペランド】

ニモニック	オペランド ([addr5])
CALLT	[addr5]

【フラグ】

Z	AC	CY

【説明】

コール・テーブル参照のサブルーチン・コールです。

次の命令の先頭アドレス(PC + 1)をスタックに退避し、コール・テーブル(アドレスの上位8ビットは00000000Bに固定で、次の5ビットをaddr5で指定します)のワード・データで示されるアドレスへ分岐します。

【記述例】

CALLT [40H]; 0040H, 0041H番地にあるワード・データをアドレスとして、そのアドレスへサブルーチン・コール

BRK

Break

ソフトウェア・ベクタ割り込み

【命令形式】 BRK

【オペレーション】 (SP - 1) PSW ,
 (SP - 2) (PC + 1)_H ,
 (SP - 3) (PC + 1)_L ,
 IE 0 ,
 SP SP - 3 ,
 PC_H (3FH) ,
 PC_L (3EH)

【オペランド】

なし

【フラグ】

Z	AC	CY

【説明】

ソフトウェア割り込み命令です。

PSWと次の命令のアドレス(PC + 1)をスタックに退避し、次にIEフラグをクリア(0)して、ベクタ・アドレス(003EH)のワード・データで指示されるアドレスへ分岐します。

IEフラグがクリア(0)されるため、以後のマスカブル・ベクタ割り込みは禁止されます。

この命令で発生したソフトウェア・ベクタ割り込みからの復帰には、RETB命令を使用します。

RETReturn
サブルーチンからの復帰**【命令形式】** RET**【オペレーション】** PCL (SP) ,
PCH (SP + 1) ,
SP SP + 2**【オペランド】**

なし

【フラグ】

Z	AC	CY

【説明】

CALL, CALLF, CALLT命令でコールされたサブルーチン・コールからのリターン命令です。
スタックに退避されているワード・データをPCに復帰し、サブルーチンからリターンします。

RETI

Return from Interrupt

ハードウェア・ベクタ割り込みからの復帰

【命令形式】 RETI**【オペレーション】** PCL (SP) ,
PCH (SP + 1) ,
PSW (SP + 2) ,
SP SP + 3 ,
NMIS 0**【オペランド】**

なし

【フラグ】

Z	AC	CY
R	R	R

【説明】

ベクタ割り込みからの復帰命令です。

スタックに退避されているデータをPCとPSWに復帰し、割り込み処理ルーチンからリターンします。

BRK命令によるソフトウェア割り込みからの復帰には使用できません。

この命令と次に実行する命令の間では、すべての割り込みを受け付けません。

NMISフラグはノンマスクابل割り込み受け付けにより1にセットされ、RETI命令により0にクリアされます。

【注意】

ノンマスクابل割り込み処理からの復帰を、RETI命令以外の命令で行うと、NMISフラグが0にクリアされないためソフトウェア割り込み以外のすべての割り込み（ノンマスクابل割り込みを含む）を受け付けなくなります。

RETB

Return from Break

ソフトウェア・ベクタ割り込みからの復帰

【命令形式】 RETB**【オペレーション】** PCL (SP) ,
PCH (SP + 1) ,
PSW (SP + 2) ,
SP SP + 3 ,**【オペランド】**

なし

【フラグ】

Z	AC	CY
R	R	R

【説明】

BRK命令で発生したソフトウェア割り込みからの復帰命令です。

スタックに退避されているPCとPSWを復帰し、割り込み処理ルーチンからリターンします。

この命令と次に実行する命令の間では、すべての割り込みを受け付けません。

5.11 スタック操作命令

スタック操作命令には、次の命令があります。

PUSH ... 105

POP ... 106

MOVW SP, src ... 107

MOVW AX, SP ... 107

PUSH

Push
プッシュ

【命令形式】 PUSH src

【オペレーション】 srcがrpの場合

(SP - 1) srcH ,
(SP - 2) srcL ,
SP SP - 2

srcがPSWの場合

(SP - 1) src
SP SP - 1

【オペランド】

二モニック	オペランド (src)
PUSH	PSW
	rp

【フラグ】

Z	AC	CY

【説明】

ソース・オペランド (src) で指定されたレジスタのデータをスタックに退避します。

【記述例】

PUSH AX ; AXレジスタの内容をスタックへ退避

POP

Pop
ポップ

【命令形式】 POP dst

【オペレーション】 dstがrpの場合 dstがPSWの場合

dstL	(SP) ,	dst	(SP)
dstH	(SP + 1) ,	SP	SP + 1
SP	SP + 2		

【オペランド】

二モニック	オペランド (dst)
POP	PSW
	rp

【フラグ】

dstがrp

Z	AC	CY

PSW

Z	AC	CY
R	R	R

【説明】

デスティネーション・オペランド (dst) で指定されたレジスタへ、データをスタックから復帰します。
 オペランドがPSWの場合、各フラグはスタックのデータで置き換わります。
 POP PSW命令と次に続く命令の間では、すべての割り込みを受け付けません。

【記述例】

POP AX ; AXレジスタへスタックのデータを復帰

MOVW SP,src

Move Word

MOVW AX,SP

スタック・ポインタとのワード・データの転送

【命令形式】 MOVW dst, src

【オペレーション】 dst src

【オペランド】

二モニック	オペランド (dst, src)
MOVW	SP, #word
	SP, AX
	AX, SP

【フラグ】

Z	AC	CY

【説明】

スタック・ポインタの内容を操作するための命令です。

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) へ第2オペランドで指定されるソース・オペランド (src) を格納します。

【記述例】

MOVW SP, #FE1FH ; スタック・ポインタへFE1FHを格納

5.12 無条件分岐命令

無条件分岐命令には、次の命令があります。

BR ... 109

BRBranch
無条件分岐**【命令形式】** BR target**【オペレーション】** PC target**【オペランド】**

ニモニック	オペランド (target)
BR	laddr16
	AX
	\$addr16

【フラグ】

Z	AC	CY

【説明】

無条件に分岐を行う命令です。

ターゲット・アドレス・オペランド (target) のワード・データをPCに転送し、分岐します。

【記述例】

BR AX ; AXレジスタの内容をアドレスとして分岐

5.13 条件付き分岐命令

条件付き分岐命令には、次の命令があります。

BC ... 111
BNC ... 112
BZ ... 113
BNZ ... 114
BT ... 115
BF ... 116
BTCLR ... 117
DBNZ ... 118

BC

Branch if Carry

キャリー・フラグによる条件分岐 (CY=1)

【命令形式】 BC \$addr16

【オペレーション】 PC PC + 2 + jdisp8 if CY = 1

【オペランド】

ニモニック	オペランド (\$addr16)
BC	\$addr16

【フラグ】

Z	AC	CY

【説明】

CY = 1 の場合に、オペランドで指定されたアドレスへ分岐します。

CY = 0 の場合、何も処理を行わず、次に続く命令を実行します。

【記述例】

BC \$300H ; CY=1 なら0300Hへ分岐 (ただし、この命令の先頭は027FH-037EH番地内にある)。

BNC

Branch if Not Carry

キャリー・フラグによる条件分岐 (CY = 0)

【命令形式】 BNC \$addr16

【オペレーション】 PC PC + 2 + jdisp8 if CY = 0

【オペランド】

モニック	オペランド (\$addr16)
BNC	\$addr16

【フラグ】

Z	AC	CY

【説明】

CY = 0 の場合に、オペランドで指定されたアドレスへ分岐します。

CY = 1 の場合、何も処理を行わず、次に続く命令を実行します。

【記述例】

BNC \$300H ; CY = 0 なら 0300H へ分岐 (ただし、この命令の先頭は 027FH-037EH 番地内にある)。

BZ

Branch if Zero

ゼロ・フラグによる条件分岐 (Z = 1)

【命令形式】 BZ \$addr16

【オペレーション】 PC PC + 2 + jdisp8 if Z = 1

【オペランド】

ニモニック	オペランド (\$addr16)
BZ	\$addr16

【フラグ】

Z	AC	CY

【説明】

Z = 1 の場合に、オペランドで指定されたアドレスへ分岐します。

Z = 0 の場合は、何も処理を行わず、次に続く命令を実行します。

【記述例】

DEC B

BZ \$3C5H ; Bレジスタが0なら03C5Hに分岐 (ただし、この命令の先頭は、0344H-0443H番地内にある)

BNZ

Branch if Not Zero

ゼロ・フラグによる条件分岐 (Z = 0)

【命令形式】 BNZ \$addr16

【オペレーション】 PC PC + 2 + jdisp8 if Z = 0

【オペランド】

ニモニック	オペランド (\$addr16)
BNZ	\$addr16

【フラグ】

Z	AC	CY

【説明】

Z = 0 の場合に、オペランドで指定されたアドレスへ分岐します。

Z = 1 の場合は、何も処理を行わず、次に続く命令を実行します。

【記述例】

CMP A, #55H

BNZ \$0A39H ; Aレジスタが0055Hでないとき、0A39Hへ分岐 (ただし、この命令の先頭は09B8H-0AB7H番地内にある)。

BT

Branch if True

ビット・テストによる条件分岐 (バイト・データのビット=1)

【命令形式】 BT bit, \$addr16

【オペレーション】 PC PC + b + jdisp8 if bit = 1

【オペランド】

ニモニック	オペランド (bit, \$addr16)	b (バイト数)
BT	saddr.bit, \$addr16	3
	sfr.bit, \$addr16	4
	A.bit, \$addr16	3
	PSW.bit, \$addr16	3
	[HL] .bit, \$addr16	3

【フラグ】

Z	AC	CY

【説明】

第1オペランド (bit) の内容がセット (1) されているとき、第2オペランド (\$addr16) で指定されるアドレスへ分岐します。

第1オペランド (bit) の内容がセット (1) されていないときは、何も処理を行わず、次に続く命令を実行します。

【記述例】

BT FE47H.3, \$55CH ; FE47H番地のビット3が1のとき、055CHへ分岐 (ただし、この命令の先頭は、04DAH-05D9H番地内にある)。

BF

Branch if False

ビット・テストによる条件分岐 (バイト・データのビット=0)

【命令形式】 BF bit, \$addr16

【オペレーション】 PC PC + b + jdisp8 if bit = 0

【オペランド】

ニモニック	オペランド (bit, \$addr16)	b (バイト数)
BF	saddr.bit, \$addr16	4
	sfr.bit, \$addr16	4
	A.bit, \$addr16	3
	PSW.bit, \$addr16	4
	[HL] .bit, \$addr16	3

【フラグ】

Z	AC	CY

【説明】

第1オペランド (bit) の内容がクリア (0) されているとき、第2オペランド (\$addr16) で指定されるアドレスへ分岐します。

第1オペランド (bit) の内容がクリア (0) されていないときは、何も処理を行わず、次に続く命令を実行します。

【記述例】

BF P2.2, \$1549H ; ポート2のビット2が0のとき、1549H番地へ分岐 (ただし、この命令の先頭は、14C6H-15C5H番地内にある)。

BTCLR

Branch if True and Clear

ビット・テストによる条件分岐とクリア (バイト・データのビット=1)

【命令形式】 BTCLR bit, \$addr16

【オペレーション】 PC ← PC + b + jdisp8 if bit = 1, then bit ← 0

【オペランド】

ニモニック	オペランド (bit, \$addr16)	b (バイト数)
BTCLR	saddr.bit, \$addr16	4
	sfr.bit, \$addr16	4
	A.bit, \$addr16	3
	PSW.bit, \$addr16	4
	[HL] .bit, \$addr16	3

【フラグ】

bitがPSW.bit

Z	AC	CY
x	x	x

左記以外

Z	AC	CY

【説明】

第1オペランド (bit) の内容がセット (1) されているとき、第1オペランド (bit) の内容をクリア (0) し、第2オペランドで指定されたアドレスへ分岐します。

第1オペランド (bit) の内容がセット (1) されていないときは、何も処理を行わず、次に続く命令を実行します。

第1オペランド (bit) がPSW.bitの場合、該当するフラグの内容がクリア (0) されます。

【記述例】

BTCLR PSW.0, \$356H ; PSWのビット0 (CYフラグ) が1の場合、CYフラグをクリアして、0356H番地へ分岐 (ただし、この命令の先頭は、02D4H-03D3H番地内にある) 。

DBNZ

Decrement and Branch if Not Zero

条件ループ (R1 0)

【命令形式】 DBNZ dst, \$addr16

【オペレーション】 dst dst - 1 ,
then PC PC + b + jdisp16 if dst R1 0

【オペランド】

ニモニック	オペランド (dst, \$addr16)	b (バイト数)
DBNZ	B, \$addr16	2
	C, \$addr16	2
	saddr, \$addr16	3

【フラグ】

Z	AC	CY

【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) の内容を - 1 して、デスティネーション・オペランド (dst) へ格納します。

デスティネーション・オペランド (dst) を - 1 した結果が 0 でなかった場合、第2オペランド (\$addr16) で示されるアドレスへ分岐します。

デスティネーション・オペランド (dst) を - 1 した結果が 0 のときは、何も処理を行わず、次に続く命令を実行します。

フラグは変化しません。

【記述例】

DBNZ B, \$1215H ; Bレジスタの内容をデクリメントし、0 にならなければ1215Hへ分岐 (ただし、この命令の先頭は、1194H-1293H番地内にある)。

5.14 CPU制御命令

CPU制御命令には、次の命令があります。

SEL RBn ... 120

NOP ... 121

EI ... 122

DI ... 123

HALT ... 124

STOP ... 125

SEL RBn

Select Register Bank
レジスタ・バンクの選択

【命令形式】 SEL RBn

【オペレーション】 RBS0, RBS1 n ; (n=0-3)

【オペランド】

ニモニック	オペランド (RBn)
SEL	RBn

【フラグ】

Z	AC	CY

【説明】

オペランド (RBn) で指定されたレジスタ・バンクを次命令以降で使用するレジスタ・バンクとします。

RBnには、RB0-RB3まであります。

【記述例】

SEL RB2 ; 次命令以降で使用するレジスタ・バンクとして、レジスタ・バンク 2 を選択

NOP

No Operation

ノー・オペレーション

【命令形式】 NOP**【オペレーション】** no operation**【オペランド】**

なし

【フラグ】

Z	AC	CY

【説明】

何も処理をせずに時間だけを消費します。

EI

Enable Interrupt

割り込みの許可

【命令形式】 EI

【オペレーション】 IE 1

【オペランド】

なし

【フラグ】

Z	AC	CY

【説明】

マスク可能割り込みの受け付け可能な状態にします（割り込み許可フラグ（IE）をセット（1）します）。

この命令と次に続く1命令の間では、すべての割り込みを受け付けません。

この命令を実行しても、他の要因によりベクタ割り込みの受け付けを行わないようにすることができます。詳細については、**各製品のユーザーズ・マニュアルの割り込み機能**を参照してください。

DI

Disable Interrupt

割り込みの禁止

【命令形式】 DI

【オペレーション】 IE 0

【オペランド】

なし

【フラグ】

Z	AC	CY

【説明】

マスクブル割り込みのベクタ割り込みによる受け付けを禁止します（割り込み許可フラグ（IE）をクリア（0）します）。

この命令と次に続く1命令の間では、すべての割り込みを受け付けません。

割り込み処理の詳細については、**各製品のユーザーズ・マニュアルの割り込み機能**を参照してください。

HALT

Halt

ホルト・モードの設定

【命令形式】 HALT**【オペレーション】** Set HALT Mode**【オペランド】**

なし

【フラグ】

Z	AC	CY

【説明】

HALTモードになります。CPUの動作クロックを停止させるモードです。通常動作モードとの組み合わせによる間欠動作により、システムのトータル消費電力を低下させることができます。

STOP

Stop

ストップ・モードの設定

【命令形式】 STOP**【オペレーション】** Set STOP Mode**【オペランド】**

なし

【フラグ】

Z	AC	CY

【説明】

STOPモードになります。メイン・システム・クロック発振回路を停止させ、システム全体が停止するモードです。リーク電流だけの超低消費電力にすることができます。

付録A 改版履歴

これまでの改版履歴を次に示します。なお、適用箇所は各版での章を示します。

版数	前版からの改版内容	適用箇所
第2版	以下の新製品を追加 μ PD78055, 78P058およびμ PD78018F, 78044A, 78054Y, 78078, 78083, 78098, 780208サブシリーズ	全 般
	関連資料に英文資料番号を追加	はじめに
	IEBusレジスタ領域(μ PD78098サブシリーズのみ)を追加	第1章 メモリ空間
	クロック欄の説明に、外部ROMにプログラムがある場合のクロック数についての説明を追加	第4章 命令セット
	ローテート命令のROR4, ROL4命令の説明に注を追加	第5章 命令の説明
	BCD補正命令のADJBA, ADJBS命令のオペレーションを修正	
第3版	以下の新製品を追加 μ PD78014H, 78018FY, 78044F, 78044H, 78058F, 78058FY, 78064Y, 78064B, 78075B, 78075BY, 78078Y, 78098B, 780018Y, 780024, 780024Y, 780034, 780034Y, 780058, 780058Y, 780228, 780308, 780308Y, 780924, 780964サブシリーズ, およびμ PD78011F, 78012F, 78070A, 78070AY, 780001, 78P0914, 780206, 780208	全 般
	以下の製品を削除 μ PD78024, 78044, 78044Aサブシリーズ	
	すべての内部RAM空間を製品別に一覧した表を追加	第1章 メモリ空間
	外部メモリ空間一覧表のフォーマットを変更	
第4版	78K/0シリーズ共通の情報以外は、削除(製品個別の情報については、 各製品のユーザーズ・マニュアル を参照)	全 般

付録 B 命令索引 (二モニク : 機能別)

【8ビット・データ転送命令】

MOV ... 52

XCH ... 53

【16ビット・データ転送命令】

MOVW ... 55

XCHW ... 56

【8ビット演算命令】

ADD ... 58

ADDC ... 59

SUB ... 60

SUBC ... 61

AND ... 62

OR ... 63

XOR ... 64

CMP ... 65

【16ビット演算命令】

ADDW ... 67

SUBW ... 68

CMPW ... 69

【乗除算命令】

MULU ... 71

DIVUW ... 72

【増減命令】

INC ... 74

DEC ... 75

INCW ... 76

DECW ... 77

【ローテート命令】

ROR ... 79

ROL ... 80

RORC ... 81

ROLC ... 82

ROR4 ... 83

ROL4 ... 84

【BCD補正命令】

ADJBA ... 86

ADJBS ... 87

【ビット操作命令】

MOV1 ... 89

AND1 ... 90

OR1 ... 91

XOR1 ... 92

SET1 ... 93

CLR1 ... 94

NOT1 ... 95

【コール・リターン命令】

CALL ... 97
 CALLF ... 98
 CALLT ... 99
 BRK ... 100
 RET ... 101
 RETI ... 102
 RETB ... 103

【CPU制御命令】

SEL RBn ... 120
 NOP ... 121
 EI ... 122
 DI ... 123
 HALT ... 124
 STOP ... 125

【スタック操作命令】

PUSH ... 105
 POP ... 106
 MOVW SP, src ... 107
 MOVW AX, SP ... 107

【無条件分岐命令】

BR ... 109

【条件付き分岐命令】

BC ... 111
 BNC ... 112
 BZ ... 113
 BNZ ... 114
 BT ... 115
 BF ... 116
 BTCLR ... 117
 DBNZ ... 118

付録C 命令索引 (ニモニック : アルファベット順)

【A】

ADD ... 58
ADDC ... 59
ADDW ... 67
ADJBA ... 86
ADJBS ... 87
AND ... 62
AND1 ... 90

【B】

BC ... 111
BF ... 116
BNC ... 112
BNZ ... 114
BR ... 109
BRK ... 100
BT ... 115
BTCLR ... 117
BZ ... 113

【C】

CALL ... 97
CALLF ... 98
CALLT ... 99
CLR1 ... 94
CMP ... 65
CMPW ... 69

【D】

DBNZ ... 118
DEC ... 75
DECW ... 77
DI ... 123
DIVUW ... 72

【E】

EI ... 122

【H】

HALT ... 124

【I】

INC ... 74
INCW ... 76

【M】

MOV ... 52
MOVW ... 55
MOVW AX, SP ... 107
MOVW SP, src ... 107
MOV1 ... 89
MULU ... 71

【N】

NOP ... 121
 NOT1 ... 95

【O】

OR ... 63
 OR1 ... 91

【P】

POP ... 106
 PUSH ... 105

【R】

RET ... 101
 RETB ... 103
 RETI ... 102
 ROL ... 80
 ROLC ... 82
 ROL4 ... 84
 ROR ... 79
 RORC ... 81
 ROR4 ... 83

【S】

SEL RBn ... 120
 SET1 ... 93
 STOP ... 125
 SUB ... 60
 SUBC ... 61
 SUBW ... 68

【X】

XCH ... 53
 XCHW ... 56
 XOR ... 64
 XOR1 ... 92

(メモ)

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494
FAX : 044-435-9608
E-mail : info@lsi.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東京 (03)3798-6106, 6107,
6108
大阪 (06)6945-3178, 3200,
3208, 3212
広島 (082)242-5504
仙台 (022)267-8740
郡山 (024)923-5591
千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,
6112
立川 (042)526-5981, 6167
松本 (0263)35-1662
静岡 (054)254-4794
金沢 (076)232-7303
松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156
水戸 (029)226-1702
前橋 (027)243-6060
鳥取 (0857)27-5313
太田 (0276)46-4014
名古屋 (052)222-2170, 2190
福岡 (092)261-2806

【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。

【NECエレクトロニクス ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.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] 78K/0シリーズ ユーザーズ・マニュアル 命令編
(U12326JJ4V0UM00 (第4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)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他()					
()					

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してください。

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