

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。



ユーザース・マニュアル

# 78K/ シリーズ

16ビット・シングルチップ・マイクロコンピュータ

命令編

---

78K/ シリーズ共通

資料番号 U10905JJ8V1UM00 (第8版)

発行年月 January 2001 N CP(K)

© NEC Corporation 1993

〔メモ〕

## 目次要約

第1章	78K/ シリーズ製品の特徴	...	21
第2章	メモリ空間	...	111
第3章	レジスタ	...	123
第4章	割り込み機能	...	139
第5章	アドレッシング	...	143
第6章	命令セット	...	177
第7章	命令の説明	...	287
第8章	開発ツール	...	473
第9章	組み込み用ソフトウェア	...	479
付録		...	481

## 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<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。

### 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, QTOPは、日本電気株式会社の商標です。

MS-DOSおよびWindowsは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

PC/ATは、米国IBM社の商標です。

イーサネットは、米国ゼロックス社の商標です。

TRONは、The Realtime Operating system Nucleusの略称です。

ITRONは、Industrial TRONの略称です。

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。

注意：μ PD784038Y, 784216AY, 784218AY, 784225Y, 784928YサブシリーズはI<sup>2</sup>Cバス・インタフェース回路を内蔵しています。

I<sup>2</sup>C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I<sup>2</sup>C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I<sup>2</sup>Cシステムに使用する実施権がフィリップス社I<sup>2</sup>C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I<sup>2</sup>Cシステムはフィリップス社によって設定されたI<sup>2</sup>C標準規格に合致しているものとします。

Purchase of NEC I<sup>2</sup>C components conveys a license under the Philips I<sup>2</sup>C Patent Rights to use these components in an I<sup>2</sup>C system, provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.

• 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	<p>μ PD784216A, 784216AY, 784218A, 784218AY, 784938A, 784956A, 784976Aサブシリーズを追加, μ PD784216, 784216Y, 784218, 784218Y, 784937, 784955サブシリーズを削除, μ PD784928, 784928Yを追加, μ PD784915, 784915A, 784916Aを削除</p> <p>次の製品が開発中 開発済み</p> <p>μ PD784224, 784225, 78F4225, 784224Y, 784225Y, 78F4225Y</p> <p>μ PD784907, 784908, 78P4908</p>
p.211	第6章 命令セット
p.212	<p>6. 4. 1 命令コードの記号 ( 8 ) post byteの条件を修正</p> <p>6. 4. 2 命令コード一覧 ( 1 ) 8ビット・データ転送命令 : MOV オペランド「MOV saddr2, A」の命令コードを修正</p>

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/ 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ます。

## 78K/ シリーズ製品

- ・  $\mu$  PD784026サブシリーズ :  $\mu$  PD784020, 784021, 784025, 784026, 78P4026
- ・  $\mu$  PD784038サブシリーズ :  $\mu$  PD784031, 784035, 784036, 784037, 784038, 78P4038, 784031(A), 784035(A), 784036(A)
- ・  $\mu$  PD784038Yサブシリーズ :  $\mu$  PD784031Y, 784035Y, 784036Y, 784037Y, 784038Y, 78P4038Y
- ・  $\mu$  PD784046サブシリーズ :  $\mu$  PD784044, 784046, 784054, 78F4046, 784044(A), (A1), (A2),  $\mu$  PD784046(A), (A1), (A2), 784054(A), (A1), (A2)
- ★  $\mu$  PD784216Aサブシリーズ :  $\mu$  PD784214A, 784215A, 784216A, 78F4216A
- ★  $\mu$  PD784216AYサブシリーズ :  $\mu$  PD784214AY, 784215AY, 784216AY, 78F4216AY
- ★  $\mu$  PD784218Aサブシリーズ<sup>注</sup> :  $\mu$  PD784217A, 784218A, 78F4218A
- ★  $\mu$  PD784218AYサブシリーズ<sup>注</sup> :  $\mu$  PD784217AY, 784218AY, 78F4218AY
- ・  $\mu$  PD784225サブシリーズ :  $\mu$  PD784224, 784225, 78F4225
- ・  $\mu$  PD784225Yサブシリーズ :  $\mu$  PD784224Y, 784225Y, 78F4225Y
- ・  $\mu$  PD784908サブシリーズ :  $\mu$  PD784907, 784908, 78P4908
- ★  $\mu$  PD784915サブシリーズ :  $\mu$  PD784915B, 784916B, 78P4916
- ★  $\mu$  PD784928サブシリーズ :  $\mu$  PD784927, 784928, 78F4928<sup>注</sup>
- ★  $\mu$  PD784928Yサブシリーズ :  $\mu$  PD784927Y, 784928Y, 78F4928Y<sup>注</sup>
- ★  $\mu$  PD784938Aサブシリーズ :  $\mu$  PD784935A, 784936A, 784937A, 784938A, 78F4938A<sup>注</sup>
- ★  $\mu$  PD784956Aサブシリーズ<sup>注</sup> :  $\mu$  PD784953A, 784956A, 78F4956A<sup>注</sup>
- ★  $\mu$  PD784976Aサブシリーズ<sup>注</sup> :  $\mu$  PD784975A<sup>注</sup>, 78F4976A<sup>注</sup>

注 開発中

**目的** このマニュアルは、78K/ 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。

## 78K/ シリーズ製品の特徴

CPU機能

命令セット

命令の説明

開発ツール

**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が、命令機能の詳細を確認するとき

付録A、付録B 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6章 命令セットでその二モニックを調べ、そのあと第7章 命令の説明で機能を調べてください。

一通り78K/ シリーズ製品の各種命令機能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/ 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。

μPD784026サブシリーズ ユーザーズ・マニュアル ハードウェア編 (U10898J)

μPD784038, 784038Yサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U11316J)

μPD784046サブシリーズ ユーザーズ・マニュアル ハードウェア編 (U11515J)

μPD784054 ユーザーズ・マニュアル ハードウェア編 (U11719J)

μPD784216A, 784216AY, 784218A, 784218AYサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U12015J)

μPD784225, 784225Yサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U12679J)

μPD784908サブシリーズ ユーザーズ・マニュアル ハードウェア編 (U11787J)

μPD784915サブシリーズ ユーザーズ・マニュアル ハードウェア編 (U10444J)

μPD784928, 784928Yサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U12648J)

★ μPD784938Aサブシリーズ ユーザーズ・マニュアル ハードウェア編 (作成予定)

★ μPD784956Aサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U14395J)

★ μPD784976Aサブシリーズ ユーザーズ・マニュアル ハードウェア編 (U15017J)

<b>凡 例</b>	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	: $\overline{\text{xxx}}$ (端子, 信号名称に上線)
	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... $\text{xxx}$ Bまたは $\text{xxx}$ 10進数... $\text{xxx}$ 16進数... $\text{xxx}$ 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/ シリーズ共通資料**

資料名	資料番号	
	和文	英文
パンフレット	U10752J	-
ユーザーズ・マニュアル 命令編	このマニュアル	U10905E
インストラクション・セット	U10595J	-
インストラクション活用表	U10594J	-
アプリケーション・ノート ソフトウエア基礎編	U10095J	U10095E

**個別資料**

**μ PD784026サブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784020, 784021 データ・シート	U11514J	U11514E
μ PD784025, 784026 データ・シート	U11605J	U11605E
μ PD78P4026 データ・シート	U11609J	U11609E
μ PD784026サブシリーズ ユーザーズ・マニュアル ハードウエア編	U10898J	U10898E
μ PD784026サブシリーズ 特殊機能レジスタ活用表	U10593J	-
μ PD784026サブシリーズ アプリケーション・ノート ハードウエア基礎編	U10573J	U10573E

**μ PD784038, 784038Yサブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784031 データ・シート	U11507J	U11507E
μ PD784035, 784036, 784037, 784038 データ・シート	U10847J	U10847E
μ PD784031(A) データ・シート	U13009J	U13009E
μ PD784035(A), 784036(A) データ・シート	U13010J	U13010E
μ PD78P4038 データ・シート	U10848J	U10848E
μ PD784038サブシリーズ 特殊機能レジスタ活用表	U11090J	-
μ PD784031Y データ・シート	U11504J	U11504E
μ PD784035Y, 784036Y, 784037Y, 784038Y データ・シート	U10741J	U10741E
μ PD78P4038Y データ・シート	U10742J	U10742E
μ PD784038Yサブシリーズ 特殊機能レジスタ活用表	U11091J	-
μ PD784038, 784038Yサブシリーズ ユーザーズ・マニュアル ハードウエア編	U11316J	U11316E
μ PD784038, 784038Yサブシリーズ アプリケーション・ノート ハードウエア基礎編	U13285J	-

**μ PD784046サブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784044, 784046 データ・シート	U10951J	U10951E
μ PD784044(A), 784046(A) データ・シート	U13121J	U13121E
μ PD784054 データ・シート	U11154J	U11154E
μ PD784054(A) データ・シート	U13122J	U13122E
μ PD78F4046 ペーパ・マシン	U11447J	U11447E
μ PD784046サブシリーズ 特殊機能レジスタ活用表	U10986J	-
μ PD784054 特殊機能レジスタ活用表	U11113J	-
μ PD784046サブシリーズ ユーザーズ・マニュアル ハードウェア編	U11515J	U11515E
μ PD784054 ユーザーズ・マニュアル ハードウェア編	U11719J	U11719E

**μ PD784216A, 784216AY, 784218A, 784218AYサブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784214A, 784215A, 784216A, 784217A, 784218A, 784214AY, 784215AY, 784216AY, 784217AY, 784218AY データ・シート	U14121J	U14121E
μ PD78F4216A, 78F4216AY, 78F4218A, 78F4218AY データ・シート	U14125J	作成中
μ PD784216A, 784216AYサブシリーズ ユーザーズ・マニュアル ハードウェア編	U12015J	U12015E

**μ PD784225, 784225Yサブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784224, 784225, 784224Y, 784225Y データ・シート	U12376J	U12376E
μ PD78F4225 ペーパ・マシン	U12499J	U12499E
μ PD784225サブシリーズ 特殊機能レジスタ活用表	U12689J	-
μ PD78F4225Y ペーパ・マシン	U12377J	U12377E
μ PD784225Yサブシリーズ 特殊機能レジスタ活用表	U12699J	-
μ PD784225, 784225Yサブシリーズ ユーザーズ・マニュアル ハードウェア編	U12679J	U12679E

**μ PD784908サブシリーズ**

資料名	資料番号	
	和文	英文
μ PD784907, 784908 データ・シート	U11680J	U11680E
μ PD78P4908 データ・シート	U11681J	U11681E
μ PD784908サブシリーズ 特殊機能レジスタ活用表	U11589J	-
μ PD784908サブシリーズ ユーザーズ・マニュアル ハードウェア編	U11787J	U11787E

μ PD784915サブシリーズ

資料名	資料番号	
	和文	英文
μ PD784915B, 784916B データ・シート	U13118J	U13118E
μ PD78P4916 データ・シート	U11045J	U11045E
μ PD784915サブシリーズ 特殊機能レジスタ活用表	U10976J	-
μ PD784915サブシリーズ ユーザーズ・マニュアル ハードウェア編	U10444J	U10444E
μ PD784915, 784928, 784928Yサブシリーズ アプリケーション・ノート VTRサーボ基礎編	U11361J	U11361E

★ μ PD784928, 784928Yサブシリーズ

資料名	資料番号	
	和文	英文
μ PD784927, 784928, 784927Y, 784928Y データ・シート	U12255J	U12255E
μ PD78F4928 ペーパー・マシン	U12188J	U12188E
μ PD784928サブシリーズ 特殊機能レジスタ活用表	U11045J	-
μ PD78F4928Y ペーパー・マシン	U12271J	U12271E
μ PD784928Yサブシリーズ 特殊機能レジスタ活用表	U12719J	-
μ PD784928, 784928Yサブシリーズ ユーザーズ・マニュアル ハードウェア編	U12648J	U12648E

★ μ PD784938Aサブシリーズ

資料名	資料番号	
	和文	英文
μ PD784935A, 784936A, 784937A, 784938A データ・シート	U13572J	作成中
μ PD78F4938A データ・シート	U14118J	作成予定
μ PD784938Aサブシリーズ ユーザーズ・マニュアル ハードウェア編	作成予定	作成予定

★ μ PD784956Aサブシリーズ

資料名	資料番号	
	和文	英文
μ PD784953A, 784956A ペーパー・マシン	作成予定	作成予定
μ PD78F4956A ペーパー・マシン	作成予定	作成予定
μ PD784955サブシリーズ 特殊機能レジスタ活用表	U12832J	-
μ PD784956Aサブシリーズ ユーザーズ・マニュアル ハードウェア編	U14395J	作成予定

★ μ PD784976Aサブシリーズ

資料名	資料番号	
	和文	英文
μ PD784975A データ・シート	作成中	作成予定
μ PD78F4976A データ・シート	作成予定	作成予定
μ PD784976Aサブシリーズ ユーザーズ・マニュアル ハードウェア編	U15017J	作成中

〔メモ〕

# 目 次

<b>第1章</b>	<b>78K/ シリーズ製品の特徴</b>	<b>... 21</b>
1.1	78K/ シリーズ製品展開図	... 23
1.2	$\mu$ PD784026サブシリーズ製品概要	... 24
1.2.1	特 徴	... 24
1.2.2	応用分野	... 24
1.2.3	オーダー情報と品質水準	... 25
1.2.4	機能概要	... 27
1.2.5	ブロック図	... 29
1.3	$\mu$ PD784038サブシリーズ製品概要	... 30
1.3.1	特 徴	... 30
1.3.2	応用分野	... 30
1.3.3	オーダー情報と品質水準	... 31
1.3.4	機能概要	... 33
1.3.5	ブロック図	... 35
1.4	$\mu$ PD784038Yサブシリーズ製品概要	... 36
1.4.1	特 徴	... 36
1.4.2	応用分野	... 36
1.4.3	オーダー情報と品質水準	... 37
1.4.4	機能概要	... 39
1.4.5	ブロック図	... 41
1.5	$\mu$ PD784046サブシリーズ製品概要	... 42
1.5.1	特 徴	... 42
1.5.2	応用分野	... 42
1.5.3	オーダー情報と品質水準	... 43
1.5.4	機能概要	... 44
1.5.5	ブロック図	... 47
★	1.6 $\mu$ PD784216Aサブシリーズ製品概要	... 49
1.6.1	特 徴	... 49
1.6.2	応用分野	... 49
1.6.3	オーダー情報と品質水準	... 50
1.6.4	機能概要	... 51
1.6.5	ブロック図	... 53
★	1.7 $\mu$ PD784216AYサブシリーズ製品概要	... 54
1.7.1	特 徴	... 54
1.7.2	応用分野	... 54
1.7.3	オーダー情報と品質水準	... 55
1.7.4	機能概要	... 56
1.7.5	ブロック図	... 58
1.8	$\mu$ PD784218Aサブシリーズ製品概要	... 59
1.8.1	特 徴	... 59
1.8.2	応用分野	... 60
1.8.3	オーダー情報と品質水準	... 60
1.8.4	機能概要	... 61

	1.8.5	ブロック図	...	63
1.9		<b>μPD784218AYサブシリーズ製品概要</b>	...	64
	1.9.1	特徴	...	64
	1.9.2	応用分野	...	65
	1.9.3	オーダ情報と品質水準	...	65
	1.9.4	機能概要	...	66
	1.9.5	ブロック図	...	68
1.10		<b>μPD784225サブシリーズ製品概要</b>	...	69
	1.10.1	特徴	...	69
	1.10.2	応用分野	...	69
	1.10.3	オーダ情報と品質水準	...	70
	1.10.4	機能概要	...	71
	1.10.5	ブロック図	...	73
1.11		<b>μPD784225Yサブシリーズ製品概要</b>	...	74
	1.11.1	特徴	...	74
	1.11.2	応用分野	...	74
	1.11.3	オーダ情報と品質水準	...	75
	1.11.4	機能概要	...	76
	1.11.5	ブロック図	...	78
1.12		<b>μPD784908サブシリーズ製品概要</b>	...	79
	1.12.1	特徴	...	79
	1.12.2	応用分野	...	79
	1.12.3	オーダ情報と品質水準	...	80
	1.12.4	機能概要	...	81
	1.12.5	ブロック図	...	83
1.13		<b>μPD784915サブシリーズ製品概要</b>	...	84
	1.13.1	特徴	...	84
	1.13.2	応用分野	...	84
	1.13.3	オーダ情報と品質水準	...	85
	1.13.4	機能概要	...	86
	1.13.5	ブロック図	...	87
1.14		<b>μPD784928サブシリーズ製品概要</b>	...	88
	1.14.1	特徴	...	88
	1.14.2	応用分野	...	88
	1.14.3	オーダ情報と品質水準	...	89
	1.14.4	機能概要	...	90
	1.14.5	ブロック図	...	91
1.15		<b>μPD784928Yサブシリーズ製品概要</b>	...	92
	1.15.1	特徴	...	92
	1.15.2	応用分野	...	92
	1.15.3	オーダ情報と品質水準	...	93
	1.15.4	機能概要	...	94
	1.15.5	ブロック図	...	95
★	1.16	<b>μPD784938Aサブシリーズ製品概要</b>	...	96
	1.16.1	特徴	...	96
	1.16.2	応用分野	...	96
	1.16.3	オーダ情報と品質水準	...	97
	1.16.4	機能概要	...	98
	1.16.5	ブロック図	...	100

★	1. 17	<b>μPD784956Aサブシリーズ製品概要</b>	...	101
	1. 17. 1	特 徴	...	101
	1. 17. 2	応用分野	...	101
	1. 17. 3	オーダ情報と品質水準	...	102
	1. 17. 4	機能概要	...	103
	1. 17. 5	ブロック図	...	105
★	1. 18	<b>μPD784976Aサブシリーズ製品概要</b>	...	106
	1. 18. 1	特 徴	...	106
	1. 18. 2	応用分野	...	106
	1. 18. 3	オーダ情報と品質水準	...	107
	1. 18. 4	機能概要	...	108
	1. 18. 5	ブロック図	...	110

## 第2章 メモリ空間 ... 111

2. 1	<b>メモリ空間</b>	...	111	
2. 2	<b>内部ROM領域</b>	...	113	
2. 3	<b>ベース領域</b>	...	115	
	2. 3. 1	ベクタ・テーブル領域	...	115
	2. 3. 2	CALLT命令テーブル領域	...	116
	2. 3. 3	CALLF命令エントリ領域	...	116
2. 4	<b>内部データ領域</b>	...	117	
	2. 4. 1	内部RAM領域	...	117
	2. 4. 2	特殊機能レジスタ (SFR) 領域	...	122
	2. 4. 3	外部SFR領域	...	122
2. 5	<b>外部メモリ空間</b>	...	122	

## 第3章 レジスタ ... 123

3. 1	<b>制御レジスタ</b>	...	123	
	3. 1. 1	プログラム・カウンタ (PC)	...	123
	3. 1. 2	プログラム・ステータス・ワード (PSW)	...	124
	3. 1. 3	RSSビットの使用方法	...	127
	3. 1. 4	スタック・ポインタ (SP)	...	130
3. 2	<b>汎用レジスタ</b>	...	133	
	3. 2. 1	構 成	...	133
	3. 2. 2	機 能	...	135
3. 3	<b>特殊機能レジスタ (SFR)</b>	...	138	

## 第4章 割り込み機能 ... 139

4. 1	<b>割り込み要求の種類</b>	...	140	
	4. 1. 1	ソフトウェア割り込み要求	...	140
	4. 1. 2	ノンマスカブル割り込み要求	...	140
	4. 1. 3	マスカブル割り込み要求	...	140
4. 2	<b>割り込み処理モード</b>	...	141	
	4. 2. 1	ベクタ割り込み	...	141
	4. 2. 2	コンテキスト・スイッチング	...	141

4.2.3	マクロ・サービス機能	...	142
-------	------------	-----	-----

## 第5章 アドレッシング ... 143

5.1	命令アドレスのアドレッシング	...	143
5.1.1	レラティブ・アドレッシング	...	144
5.1.2	イミーディエト・アドレッシング	...	145
5.1.3	テーブル・インダイレクト・アドレッシング	...	147
5.1.4	16ビット・レジスタ・アドレッシング	...	148
5.1.5	20ビット・レジスタ・アドレッシング	...	149
5.1.6	16ビット・レジスタ・インダイレクト・アドレッシング	...	150
5.1.7	20ビット・レジスタ・インダイレクト・アドレッシング	...	151
5.2	オペランド・アドレスのアドレッシング	...	152
5.2.1	インプライド・アドレッシング	...	153
5.2.2	レジスタ・アドレッシング	...	154
5.2.3	イミーディエト・アドレッシング	...	156
5.2.4	8ビット・ダイレクト・アドレッシング	...	157
5.2.5	16ビット・ダイレクト・アドレッシング	...	158
5.2.6	24ビット・ダイレクト・アドレッシング	...	159
5.2.7	ショート・ダイレクト・アドレッシング	...	160
5.2.8	特殊機能レジスタ (SFR) アドレッシング	...	162
5.2.9	ショート・ダイレクト16ビット・メモリ・インダイレクト・アドレッシング	...	163
5.2.10	ショート・ダイレクト24ビット・メモリ・インダイレクト・アドレッシング	...	165
5.2.11	スタック・アドレッシング	...	166
5.2.12	24ビット・レジスタ・インダイレクト・アドレッシング	...	167
5.2.13	16ビット・レジスタ・インダイレクト・アドレッシング	...	169
5.2.14	ベースト・アドレッシング	...	170
5.2.15	インデクスト・アドレッシング	...	172
5.2.16	ベースト・インデクスト・アドレッシング	...	174

## 第6章 命令セット ... 177

6.1	凡例	...	177
6.2	命令のオペレーション一覧	...	180
6.3	アドレッシング別命令一覧	...	204
6.4	命令コード	...	209
6.4.1	命令コードの記号	...	209
6.4.2	命令コード一覧	...	212
6.5	命令のクロック数	...	262
6.5.1	命令の実行時間	...	262
6.5.2	クロック数欄の凡例	...	262
6.5.3	クロック数欄の説明	...	263
6.5.4	クロック数一覧	...	264

## 第7章 命令の説明 ... 287

7.1	8ビット・データ転送命令	...	289
7.2	16ビット・データ転送命令	...	292

7.3	24ビット・データ転送命令	...	295
7.4	8ビット・データ交換命令	...	297
7.5	16ビット・データ交換命令	...	299
7.6	8ビット演算命令	...	301
7.7	16ビット演算命令	...	315
7.8	24ビット演算命令	...	322
7.9	乗除算命令	...	325
7.10	特殊演算命令	...	331
7.11	増減命令	...	341
7.12	補正命令	...	348
7.13	シフト・ローテート命令	...	352
7.14	ビット操作命令	...	363
7.15	スタック操作命令	...	374
7.16	コール・リターン命令	...	386
7.17	無条件分岐命令	...	401
7.18	条件付き分岐命令	...	403
7.19	CPU制御命令	...	423
7.20	特殊命令	...	433
7.21	ストリング命令	...	438

## 第8章 開発ツール ... 473

8.1	開発ツール	...	474
8.2	PROM書き込み用ツール	...	477
8.3	フラッシュ・メモリ書き込み用ツール	...	477

## 第9章 組み込み用ソフトウェア ... 479

9.1	リアルタイムOS	...	479
-----	----------	-----	-----

## 付録A 命令索引（ニモニック：機能別） ... 481

## 付録B 命令索引（ニモニック：アルファベット順） ... 485

## 付録C 改版履歴 ... 487

# 図の目次

図番号	タイトル, ページ
1 - 1	78Kシリーズおよび78K $\mu$ シリーズの構成 ... 22
2 - 1	メモリ・マップ ... 112
2 - 2	内部RAMのメモリ・マップ ... 120
3 - 1	プログラム・カウンタ (PC) の構成 ... 123
3 - 2	プログラム・ステータス・ワード (PSW) の構成 ... 124
3 - 3	スタック・ポインタ (SP) の構成 ... 130
3 - 4	スタック領域へ退避されるデータ ... 131
3 - 5	スタック領域から復帰されるデータ ... 132
3 - 6	汎用レジスタ構成 ... 133
3 - 7	汎用レジスタのアドレス ... 134
4 - 1	割り込み要求の発生によるコンテキスト・スイッチング動作 ... 141
8 - 1	開発ツールの構成 ... 476

# 表の目次

表番号	タイトル, ページ
2 - 1	78K/ シリーズ製品別内部ROM領域一覧 ... 113
2 - 2	ベクタ・テーブル ... 115
2 - 3	78K/ シリーズ製品別内部RAM領域一覧 ... 118
3 - 1	レジスタ・バンクの選択 ... 126
3 - 2	機能名称-絶対名称の対応 ... 137
4 - 1	割り込み要求の処理 ... 139
6 - 1	8ビット・アドレッシング別命令一覧表 ... 204
6 - 2	16ビット・アドレッシング別命令一覧表 ... 205
6 - 3	24ビット・アドレッシング別命令一覧表 ... 206
6 - 4	ビット操作命令アドレッシング別命令一覧表 ... 207
6 - 5	コール・リターン命令 / 分岐命令アドレッシング別命令一覧表 ... 208
8 - 1	開発ツールの種類と機能 ... 474

〔メモ〕

## 第1章 78K/ シリーズ製品の特徴

78Kシリーズは、図1 - 1に示すように6つのシリーズで構成されています。

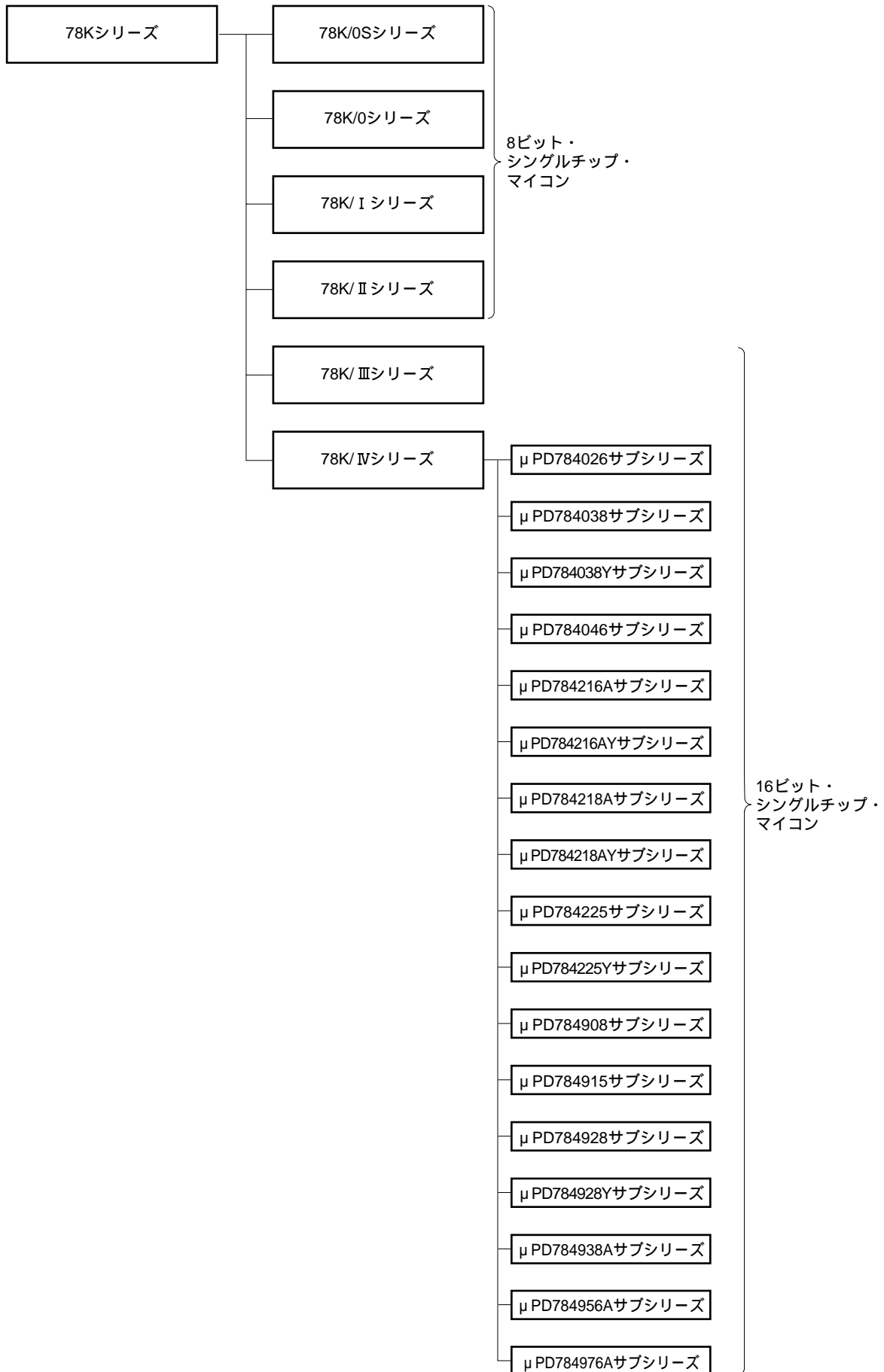
78K/ シリーズは、その6つのシリーズのうちの1つで、16ビットCPUを内蔵した製品です。

制御用途に適した命令体系、高性能割り込みコントローラを持ち、さらに最大1 Mバイトのプログラム・メモリ空間および最大16 Mバイトのデータ・メモリ空間を備えた高性能CPUを搭載しています。

78K/ シリーズは、さまざまなサブシリーズを用意しており、用途に応じて最適なサブシリーズを選択することができます。

各サブシリーズは、周辺ハードウェアが異なるだけでCPUは同じです。したがって、命令セットはすべて共通です。また、サブシリーズ内の各製品については、内蔵するメモリのサイズだけが異なります。

図1 - 1 78Kシリーズおよび78K/ シリーズの構成

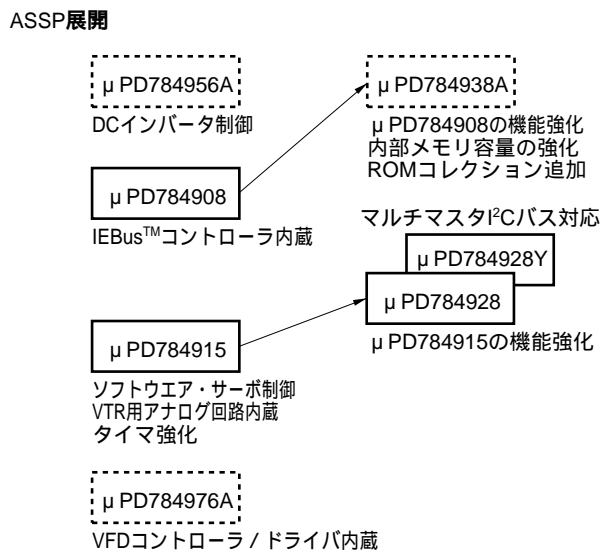
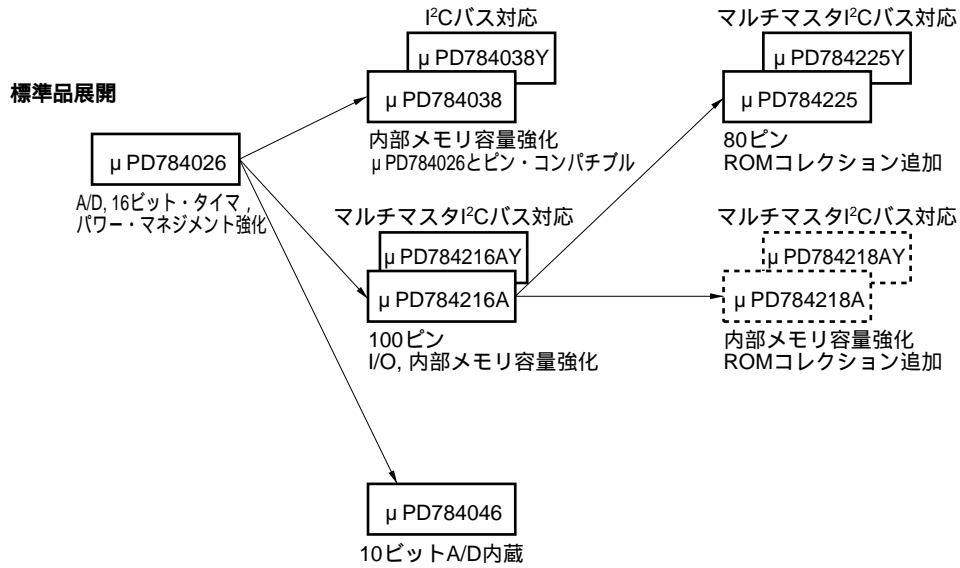


★  
★  
★

★ 1.1 78K/ シリーズ製品展開図

□ : 量産中

□ (点線) : 開発中



## 1.2 $\mu$ PD784026サブシリーズ製品概要 ( $\mu$ PD784020, 784021, 784025, 784026, 78P4026)

### 1.2.1 特 徴

$\mu$ PD78234サブシリーズとピン・コンパチブル

最小命令実行時間：160 ns/320 ns/640 ns/1280 ns/ (25 MHz動作時)

内部メモリ

ROM

マスクROM：48 Kバイト ( $\mu$ PD784025)

64 Kバイト ( $\mu$ PD784026)

内蔵せず ( $\mu$ PD784020, 784021)

PROM：64 Kバイト ( $\mu$ PD78P4026)

RAM：2048バイト ( $\mu$ PD784021, 784025, 784026)

512バイト ( $\mu$ PD784020)

I/O端子：64本

46本 ( $\mu$ PD784020, 784021のみ)

タイマ/カウンタ：16ビット・タイマ/カウンタ×3ユニット

16ビット・タイマ×1ユニット

ウォッチドッグ・タイマ：1チャンネル

A/Dコンバータ：8ビット分解能×8チャンネル

D/Aコンバータ：8ビット分解能×2チャンネル

シリアル・インタフェース：3チャンネル

UART/IOE (3線式シリアルI/O)：2チャンネル (ボー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O, SBI)：1チャンネル

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

スタンバイ機能：HALT/STOP/IDLEモード

クロック出力機能

fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/16から選択 ( $\mu$ PD784020, 784021以外)

電源電圧：V<sub>DD</sub> = 2.7 ~ 5.5 V

### 1.2.2 応用分野

LBP, オートフォーカス・カメラ, PPC, プリンタ, 電子タイプライタ, エアコン, 電子楽器, 携帯電話など

## 1.2.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784020GC-3B9	80ピン・プラスチックQFP (14 × 14 mm)	なし
μ PD784021GC-3B9	"	"
μ PD784021GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 × 12 mm)	"
μ PD784025GC- x x x -3B9	80ピン・プラスチックQFP (14 × 14 mm)	マスクROM
μ PD784026GC- x x x -3B9	"	"
μ PD78P4026GC-3B9	"	ワン・タイムPROM
μ PD78P4026GC- x x x -3B9 <sup>注</sup>	"	書き込み済み ワン・タイムPROM
μ PD78P4026KK-T	80ピン・セラミックWQFN (14 × 14 mm)	EPROM

**注** QTOP™マイコンです。QTOPマイコンとは、NECの書き込みサービス（書き込みから捺印、スクリーニング、検査）によりトータル・サポートされた、ワン・タイムPROM内蔵シングルチップ・マイコンの総称です。

**備考** x x x は、ROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784020GC-3B9	80ピン・プラスチックQFP (14 × 14 mm)	標準 (一般電子機器用)
μ PD784021GC-3B9	"	"
μ PD784021GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 × 12 mm)	"
μ PD784025GC- x x x -3B9	80ピン・プラスチックQFP (14 × 14 mm)	"
μ PD784026GC- x x x -3B9	"	"
μ PD78P4026GC-3B9	"	"
μ PD78P4026GC- x x x -3B9 <sup>注</sup>	"	"
μ PD78P4026KK-T	80ピン・セラミックWQFN (14 × 14 mm)	適用外 (機能評価用)

**注** QTOPマイコンです。QTOPマイコンとは、NECの書き込みサービス (書き込みから捺印、スクリーニング、検査) によりトータル・サポートされた、ワン・タイムPROM内蔵シングルチップ・マイコンの総称です。

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J) をご覧ください。

**注意** μ PD78P4026のうち、EPROM製品は、お客様の装置の量産製品に使用されることを意図した信頼性を保持しておりません。実験または試作時の機能評価用에만使用してください。

**備考** x x x は、ROMコード番号です。

## 1.2.4 機能概要

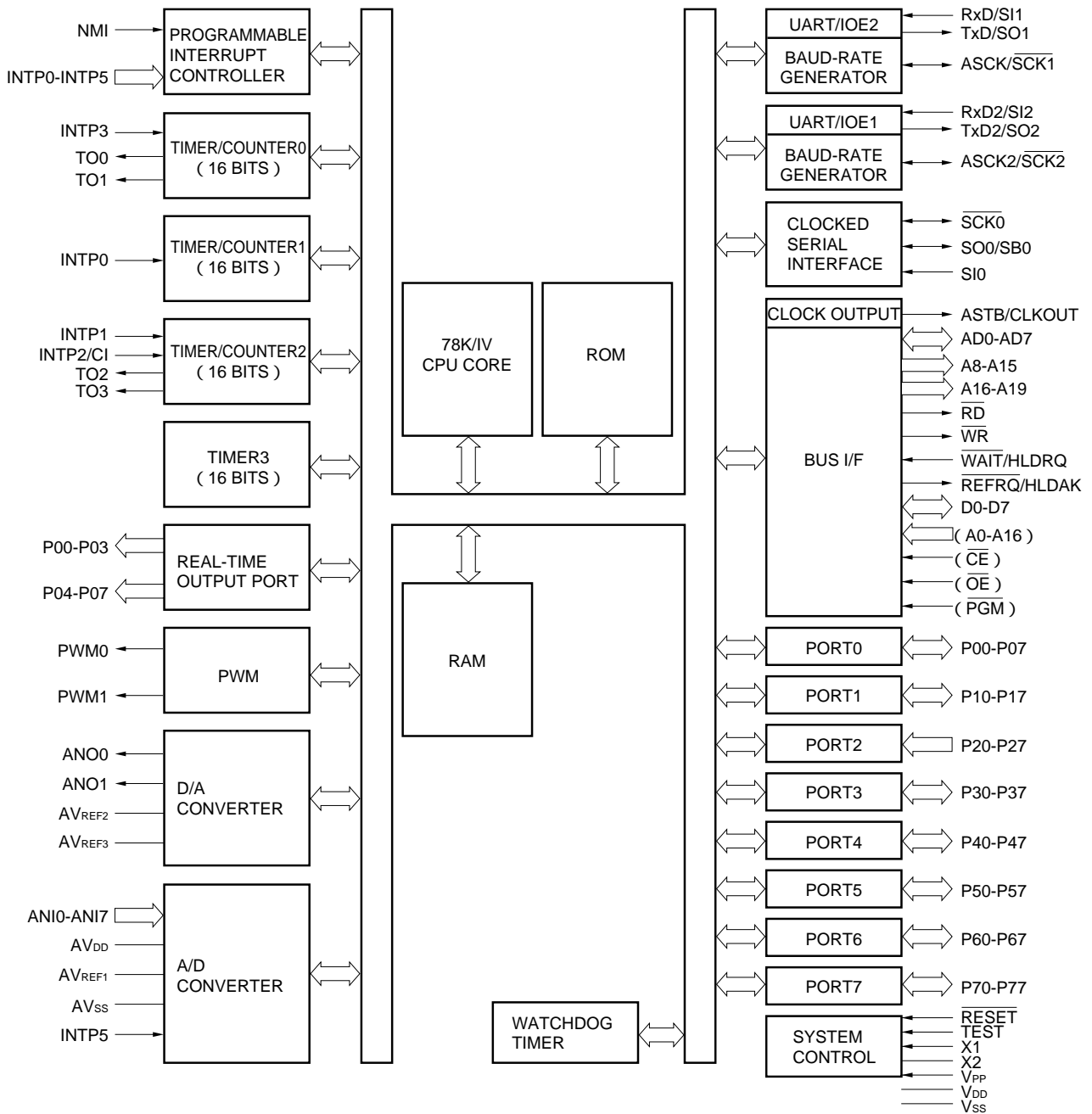
(1/2)

品名		μ PD784020	μ PD784021	μ PD784025	μ PD784026	μ PD78P4026
項目						
基本命令（二モニック）数		113				
汎用レジスタ		8ビット×16レジスタ×8バンク，または16ビット×8レジスタ×8バンク（メモリ・マッピング）				
最小命令実行時間		160 ns/320 ns/640 ns/1280 ns（25 MHz動作時）				
内部メモリ	ROM	なし		48 Kバイト （マスクROM）	64 Kバイト （マスクROM）	64 Kバイト （PROM）
	RAM	512バイト	2048バイト			
メモリ空間		プログラム/データあわせて1 Mバイト				
I/Oポート	合計	46本			64本	
	入力	8本			8本	
	入出力	34本			56本	
	出力	4本			0本	
注 付加機能付き端子	プルアップ抵抗付き端子	32本			54本	
	LEDダイレクト・ドライブ出力	8本			24本	
	トランジスタ・ダイレクト・ドライブ	8本				
リアルタイム出力ポート		4ビット×2，または8ビット×1				
タイマ/カウンタ		タイマ/カウンタ0：タイマ・レジスタ×1 (16ビット)      コンペア・レジスタ×2 キャプチャ・レジスタ×1			パルス出力可 ・トグル出力 ・PWM/PPG出力 ・ワンショット・パルス出力	
		タイマ/カウンタ1：タイマ・レジスタ×1 (8/16ビット)      コンペア・レジスタ×1 キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1			パルス出力可 ・リアルタイム出力：4ビット×2	
		タイマ/カウンタ2：タイマ・レジスタ×1 (8/16ビット)      コンペア・レジスタ×1 キャプチャ/コンペア・レジスタ×1 キャプチャ・レジスタ×1			パルス出力可 ・トグル出力 ・PWM/PPG出力	
		タイマ3 (8/16ビット)      タイマ・レジスタ×1 コンペア・レジスタ×1				
ウォッチドッグ・タイマ		1チャンネル				
PWM出力		12ビット分解能×2チャンネル				
シリアル・インタフェース		UART×IOE（3線式シリアルI/O）：2チャンネル（ポーレート・ジェネレータ内蔵） CSI（3線式シリアルI/O, SBI）：1チャンネル				
A/Dコンバータ		8ビット分解能×8チャンネル				
D/Aコンバータ		8ビット分解能×2チャンネル				
スタンバイ		HALT/STOP/IDLEモード				

注 付加機能付き端子は，I/O端子の中に含まれています。

品名		μ PD784020	μ PD784021	μ PD784025	μ PD784026	μ PD78P4026
割り込み	ハードウェア要因	23 (内部16, 外部7 (サンプリング・クロック可変入力: 1))				
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー				
	ノンマスクابل	内部: 1本, 外部: 1本				
	マスクابل	内部: 15本, 外部: 6本				
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態 (ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング)				
クロック出力機能		-		f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択 (1ビット出力ポートとしても使用可能)		
電源電圧		V <sub>DD</sub> = 2.7 ~ 5.5 V				
パッケージ		・80ピン・プラスチックQFP (14 × 14 mm) ・80ピン・プラスチックTQFP (ファインピッチ, 12 × 12 mm : μ PD784021のみ) ・80ピン・セラミックWQFN (14 × 14 mm : μ PD78P4026のみ)				

### 1.2.5 ブロック図



備考1. 内部ROM, RAM容量は、製品により異なります。

2. V<sub>PP</sub>はμ PD78P4026のみ

3. ( )内はPROMプログラミング・モード時

### 1.3 $\mu$ PD784038サブシリーズ製品概要

( $\mu$ PD784031, 784035, 784036, 784037, 784038, 78P4038,  
784031(A), 784035(A), 784036(A) )

#### 1.3.1 特 徴

$\mu$  PD78234サブシリーズ,  $\mu$  PD784026サブシリーズおよび $\mu$  PD784038Yサブシリーズとピン・コンパチブル

$\mu$  PD78234サブシリーズ,  $\mu$  PD784026サブシリーズの内部メモリ容量を拡張

最小命令実行時間: 125 ns/250 ns/500 ns/1000 ns (32 MHz動作時)

内部メモリ

ROM

マスクROM: なし ( $\mu$ PD784031, 784031(A) )

48 Kバイト ( $\mu$ PD784035, 784035(A) )

64 Kバイト ( $\mu$ PD784036, 784036(A) )

96 Kバイト ( $\mu$ PD784037) )

128 Kバイト ( $\mu$ PD784038) )

PROM : 128 Kバイト ( $\mu$ PD78P4038) )

RAM : 2048バイト ( $\mu$ PD784031, 784035, 784036, 784031(A), 784035(A), 784036(A) )

3584バイト ( $\mu$ PD784037) )

4352バイト ( $\mu$ PD784038) )

I/Oポート: 64本

タイマ/カウンタ: 16ビット・タイマ/カウンタ×3ユニット

16ビット・タイマ×1ユニット

ウォッチドッグ・タイマ: 1チャンネル

A/Dコンバータ: 8ビット分解能×8チャンネル

D/Aコンバータ: 8ビット分解能×2チャンネル

12ビットPWM出力: 2チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル

CSI (3線式シリアルI/O, 2線式シリアルI/O) : 1チャンネル

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

クロック出力機能

fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/16から選択 ( $\mu$ PD784031以外)

電源電圧:  $V_{DD} = 2.7 \sim 5.5$  V

#### 1.3.2 応用分野

標準品: LBP, オートフォーカス・カメラ, PPC, プリンタ, 電子タイプライタ, エアコン, 電子楽器, 携帯電話など

特別品: 自動車電装の制御装置, ガス検知遮断機, 各種安全装置など

## 1.3.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784031GC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	なし
μ PD784031GC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784031GC(A)-x x x-3B9	" (14×14 mm, 2.7 mm厚)	"
μ PD784031GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784035GC-x x x-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	マスクROM
μ PD784035GC-x x x-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784035GC(A)-x x x-3B9	" (14×14 mm, 2.7 mm厚)	"
μ PD784035GK-x x x-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784036GC-x x x-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784036GC-x x x-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784036GC(A)-x x x-3B9	" (14×14 mm, 2.7 mm厚)	"
μ PD784036GK-x x x-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784037GC-x x x-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784037GC-x x x-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784037GK-x x x-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784038GC-x x x-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784038GC-x x x-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784038GK-x x x-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD78P4038GC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	ワン・タイムPROM
μ PD78P4038GC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038GC-x x x-3B9 <sup>注</sup>	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	書き込み済み ワン・タイムPROM
μ PD78P4038GC-x x x-8BT <sup>注</sup>	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	ワン・タイムPROM
μ PD78P4038GK-x x x-BE9 <sup>注</sup>	"	書き込み済み ワン・タイムPROM
μ PD78P4038KK-T	80ピン・セラミックWQFN (14×14 mm)	EPROM

注 QTOPマイコンです。QTOPマイコンとは、NECの書き込みサービス（書き込みから捺印、スクリーニング、検査）によりトータル・サポートされた、ワン・タイムPROM内蔵マイコンの総称です。

備考 x x xはROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784031GC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	標準 (一般電子機器用)
μ PD784031GC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784031GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784035GC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784035GC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784035GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784036GC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784036GC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784036GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784037GC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784037GC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784037GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784038GC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784038GC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784038GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD78P4038GC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD78P4038GC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038GC- x x x -3B9 <sup>注</sup>	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD78P4038GC- x x x -8BT <sup>注</sup>	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD78P4038GK- x x x -BE9 <sup>注</sup>	"	"
μ PD784031GC(A)- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	特別(高信頼度電子機器用)
μ PD784035GC(A)- x x x -3B9	"	"
μ PD784036GC(A)- x x x -3B9	"	"
μ PD78P4038KK-T	80ピン・セラミックWQFN (14×14 mm)	適用外 (機能評価用)

注 QTOPマイコンです。QTOPマイコンとは、NECの書き込みサービス (書き込みから捺印、スクリーニング、検査) によりトータル・サポートされた、ワン・タイムPROM内蔵マイコンの総称です。

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J) をご覧ください。

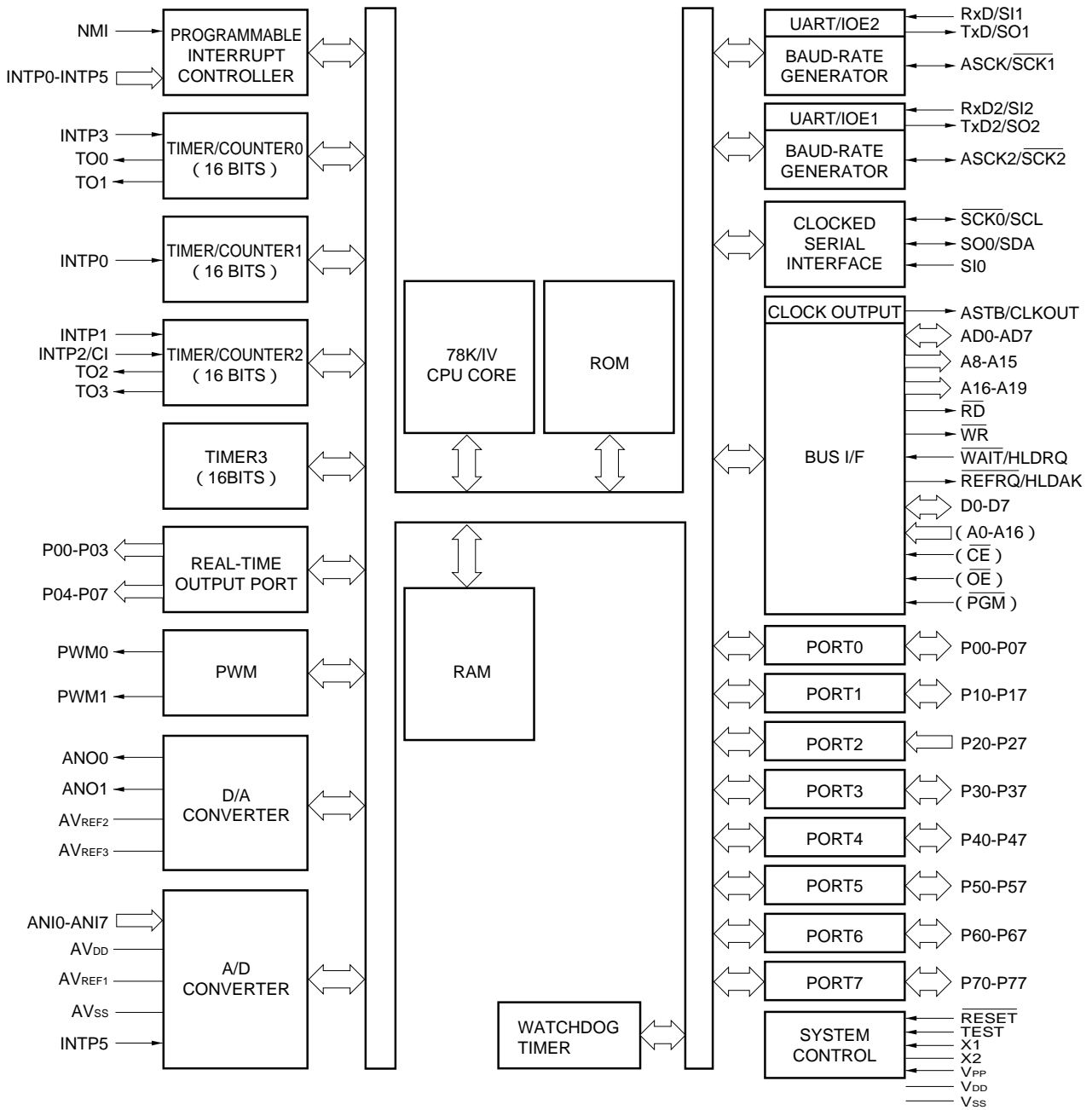
備考 x x x はROMコード番号です。

注意 μ PD78P4038のうち、EPROM製品は、お客様の装置の量産製品に使用されることを意図した信頼性を保持しておりません。実験または試作時の機能評価用のみ使用してください。



品 名		$\mu$ PD784031, 784031(A)	$\mu$ PD784035, 784035(A)	$\mu$ PD784036, 784036(A)	$\mu$ PD784037	$\mu$ PD784038	$\mu$ PD78P4038
項 目		-	f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択 (1ビット出力ポートとしても使用可能)				
クロック出力		-	f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択 (1ビット出力ポートとしても使用可能)				
ウォッチドッグ・タイマ		1チャンネル					
スタンバイ		HALT/STOP/IDLEモード					
割り込み	ハードウェア要因	23 ( 内部16, 外部7 ( サンプリング・クロック可変入力 : 1 ) )					
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー					
	ノンマスカブル	内部 : 1本, 外部 : 1本					
	マスカブル	内部 : 15本, 外部 : 6本					
		・ 4レベルのプログラマブル・プライオリティ ・ 3種類の処理形態 ( ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング )					
電源電圧		V <sub>DD</sub> = 2.7 ~ 5.5 V					
パッケージ		・ 80ピン・プラスチックQFP ( 14 × 14 mm, 1.4 mm厚 ) ・ 80ピン・プラスチックQFP ( 14 × 14 mm, 2.7 mm厚 ) ・ 80ピン・プラスチックTQFP ( ファインピッチ ) ( 12 × 12 mm ) ・ 80ピン・セラミックWQFN ( 14 × 14 mm ) : $\mu$ PD78P4038のみ					

### 1.3.5 ブロック図



備考1. 内部ROM, RAM容量は、製品により異なります。

2. V<sub>PP</sub>はμ PD78P4038のみ

3. ( )内はPROMプログラミング・モード時

## 1.4 $\mu$ PD784038Yサブシリーズ製品概要

( $\mu$ PD784031Y, 784035Y, 784036Y, 784037Y, 784038Y, 78P4038Y)

### 1.4.1 特徴

$\mu$ PD784038サブシリーズに $I^2C$ バス制御機能を追加

$\mu$ PD78234サブシリーズ,  $\mu$ PD784026サブシリーズおよび $\mu$ PD784038サブシリーズとピン・コンパチブル

$\mu$ PD78234サブシリーズ,  $\mu$ PD784026サブシリーズの内部メモリ容量を拡張

最小命令実行時間: 125 ns/250 ns/500 ns/1000 ns (32 MHz動作時)

内部メモリ

ROM

マスクROM: なし ( $\mu$ PD784031Y)

48 Kバイト ( $\mu$ PD784035Y)

64 Kバイト ( $\mu$ PD784036Y)

96 Kバイト ( $\mu$ PD784037Y)

128 Kバイト ( $\mu$ PD784038Y)

PROM : 128 Kバイト ( $\mu$ PD78P4038Y)

RAM : 2048バイト ( $\mu$ PD784031Y, 784035Y, 784036Y)

3584バイト ( $\mu$ PD784037Y)

4352バイト ( $\mu$ PD784038Y)

I/Oポート: 64本

タイマ/カウンタ: 16ビット・タイマ/カウンタ×3ユニット

16ビット・タイマ×1ユニット

ウォッチドッグ・タイマ: 1チャンネル

A/Dコンバータ: 8ビット分解能×8チャンネル

D/Aコンバータ: 8ビット分解能×2チャンネル

12ビットPWM出力: 2チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O): 2チャンネル

CSI (3線式シリアルI/O, 2線式シリアルI/O,  $I^2C$ バス): 1チャンネル

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

クロック出力機能

fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/16から選択 ( $\mu$ PD784031Y以外)

電源電圧:  $V_{DD} = 2.7 \sim 5.5$  V

### 1.4.2 応用分野

携帯電話, コードレス電話, AV機器など

## 1.4.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784031YGC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	なし
μ PD784031YGC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784031YGK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784035YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	マスクROM
μ PD784035YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784035YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784036YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784036YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784036YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784037YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784037YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784037YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD784038YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
μ PD784038YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD784038YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
μ PD78P4038YGC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	ワン・タイムPROM
μ PD78P4038YGC-8BT	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038YGC- x x x -3B9 <sup>注</sup>	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	書き込み済み ワン・タイムPROM
μ PD78P4038YGC- x x x -8BT <sup>注</sup>	" (14×14 mm, 1.4 mm厚)	"
μ PD78P4038YGK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	ワン・タイムPROM
μ PD78P4038YGK- x x x -BE9 <sup>注</sup>	"	書き込み済み ワン・タイムPROM
μ PD78P4038YKK-T	80ピン・セラミックWQFN (14×14 mm)	EPROM

注 QTOPマイコンです。QTOPマイコンとは、NECの書き込みサービス（書き込みから捺印、スクリーニング、検査）によりトータル・サポートされた、ワン・タイムPROM内蔵マイコンの総称です。

備考 x x x はROMコード番号です。

注意 μ PD784035YGK- x x x -BE9, 784036YGK- x x x -BE9は開発中です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784031YGC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	標準 (一般電子機器用)
$\mu$ PD784031YGC-8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD784031YGK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD784035YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD784035YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD784035YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD784036YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD784036YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD784036YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD784037YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD784037YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD784037YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD784038YGC- x x x -3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD784038YGC- x x x -8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD784038YGK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD78P4038YGC-3B9	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD78P4038YGC-8BT	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD78P4038YGC- x x x -3B9 <sup>注</sup>	80ピン・プラスチックQFP (14×14 mm, 2.7 mm厚)	"
$\mu$ PD78P4038YGC- x x x -8BT <sup>注</sup>	" (14×14 mm, 1.4 mm厚)	"
$\mu$ PD78P4038YGK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12×12 mm)	"
$\mu$ PD78P4038YGK- x x x -BE9 <sup>注</sup>	"	"
$\mu$ PD78P4038YKK-T	80ピン・セラミックWQFN (14×14 mm)	適用外 (機能評価用)

注 QTOPマイコンです。QTOPマイコンとは、NECの書き込みサービス (書き込みから捺印、スクリーニング、検査) によりトータル・サポートされた、ワン・タイムPROM内蔵マイコンの総称です。

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J) をご覧ください。

備考 x x x はROMコード番号です。

注意1 .  $\mu$  PD78P4038Yのうち、EPROM製品は、お客様の装置の量産製品に使用されることを意図した信頼性を保持しておりません。実験または試作時の機能評価用のみ使用してください。

2 .  $\mu$  PD784035YGK- x x x -BE9, 784036YGK- x x x -BE9は開発中です。

## 1.4.4 機能概要

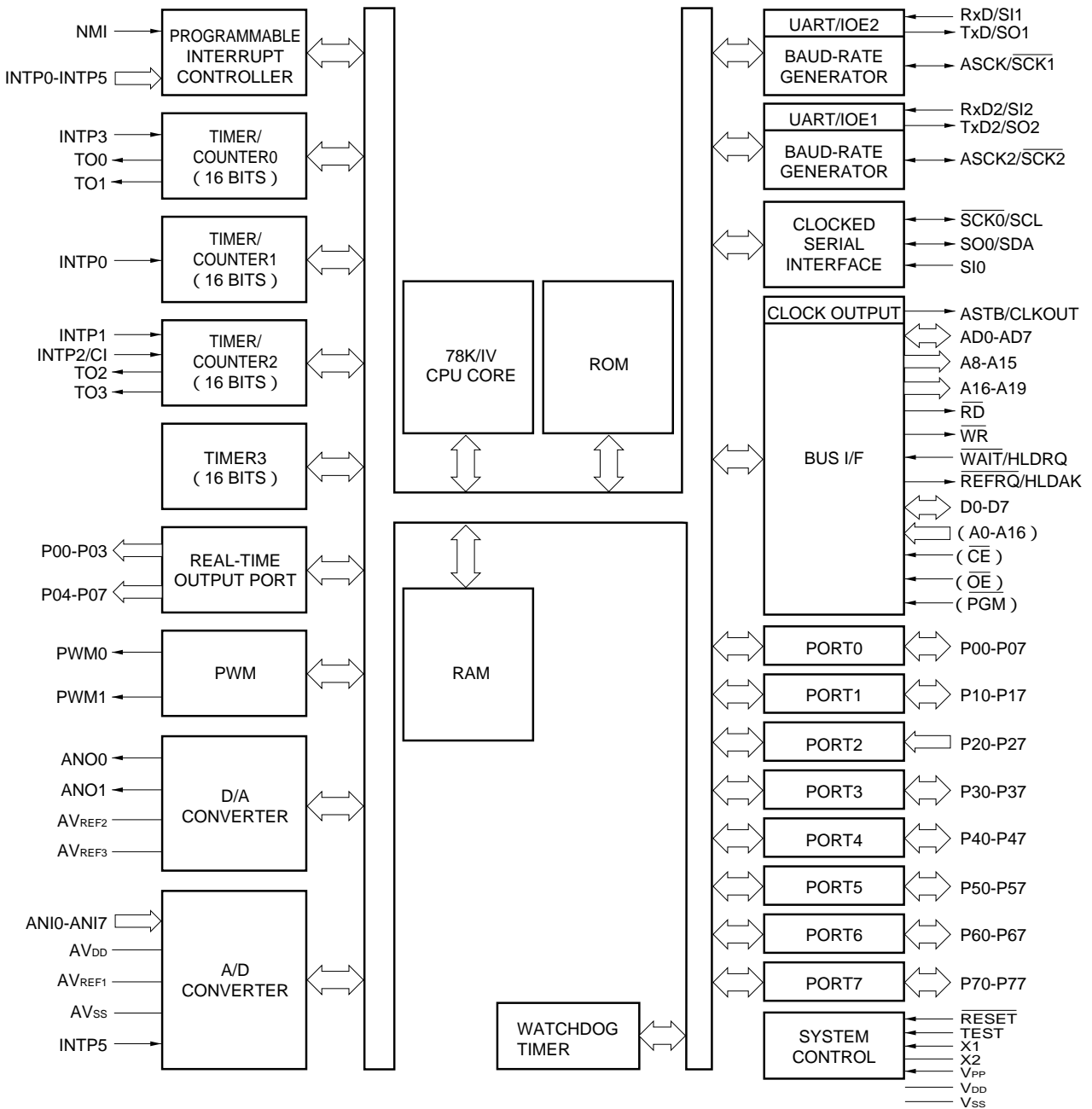
(1/2)

品名		μ PD784031Y	μ PD784035Y	μ PD784036Y	μ PD784037Y	μ PD784038Y	μ PD78P4038Y
項目							
基本命令(ニモニック)数		113					
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)					
最小命令実行時間		125 ns/250 ns/500 ns/1000 ns (32 MHz動作時)					
内部メモリ	ROM	なし	48 Kバイト (マスクROM)	64 Kバイト (マスクROM)	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (ワン・タイム PROMまたは EPROM)
	RAM	2048バイト			3584バイト	4352バイト	
メモリ空間		プログラム/データあわせて1 Mバイト					
I/Oポート	合計	64本					
	入力	8本					
	入出力	56本					
注 付加機能付き端子	ブルアップ抵抗付き端子	54本					
	LEDダイレクト・ドライブ出力	24本					
	トランジスタ・ダイレクト・ドライブ	8本					
リアルタイム出力ポート		4ビット×2, または8ビット×1					
タイマ/カウンタ		タイマ/カウンタ0 : タイマ・レジスタ×1 キャプチャ・レジスタ×1 コンペア・レジスタ×2			パルス出力可 ・トグル出力 ・PWM/PPG出力 ・ワンショット・パルス出力		
		タイマ/カウンタ1 : タイマ・レジスタ×1 キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1			パルス出力可 ・リアルタイム出力(4ビット×2)		
		タイマ/カウンタ2 : タイマ・レジスタ×1 キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1			パルス出力可 ・トグル出力 ・PWM/PPG出力		
		タイマ3 : タイマ・レジスタ×1 コンペア・レジスタ×1					
PWM出力		12ビット分解能×2チャンネル					
シリアル・インタフェース		UART×IOE(3線式シリアルI/O) : 2チャンネル(ボー・レート・ジェネレータ内蔵) CSI(3線式シリアルI/O, 2線式シリアルI/O, I <sup>2</sup> Cバス) : 1チャンネル					
A/Dコンバータ		8ビット分解能×8チャンネル					
D/Aコンバータ		8ビット分解能×2チャンネル					

注 付加機能付き端子は、I/O端子の中に含まれています。

品 名		μ PD784031Y	μ PD784035Y	μ PD784036Y	μ PD784037Y	μ PD784038Y	μ PD78P4038Y
項 目							
クロック出力		-	f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択 (1ビット出力ポートとしても使用可能)				
ウォッチドッグ・タイマ		1チャンネル					
スタンバイ		HALT/STOP/IDLEモード					
割り込み	ハードウェア要因	24 (内部17, 外部7 (サンプリング・クロック可変入力: 1))					
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー					
	ノンマスカブル	内部: 1本, 外部: 1本					
	マスカブル	内部: 16本, 外部: 6本					
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態 (ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング)					
電源電圧		V <sub>DD</sub> = 2.7 ~ 5.5 V					
パッケージ		・80ピン・プラスチックQFP (14 × 14 mm, 1.4 mm厚) ・80ピン・プラスチックQFP (14 × 14 mm, 2.7 mm厚) ・80ピン・プラスチックTQFP (ファインピッチ) (12 × 12 mm) ・80ピン・セラミックWQFN (14 × 14 mm) : μ PD78P4038Yのみ					

### 1.4.5 ブロック図



備考1. 内部ROM, RAM容量は、製品により異なります。

2. VPPはμ PD78P4038Yのみ

3. ( )内はPROMプログラミング・モード時

## 1.5 $\mu$ PD784046サブシリーズ製品概要

( $\mu$ PD784044, 784054, 784046, 78F4046, 784044(A), 784044(A1), 784044(A2), 784046(A), 784046(A1), 784046(A2), 784054(A), 784054(A1), 784054(A2))

### 1.5.1 特徴

最小命令実行時間：125 ns (内部16 MHz動作時) ... $\mu$ PD784044, 784046, 784054, 78F4046  
 160 ns (内部12.5 MHz動作時) ... $\mu$ PD784044(A), 784046(A), 784054(A)  
 200 ns (内部10 MHz動作時) ... $\mu$ PD784044(A1), (A2), 784046(A1), (A2),  
 784054(A1), (A2)

内部メモリ

ROM

マスクROM : 64 Kバイト ( $\mu$ PD784046, 784046(A), (A1), (A2))  
 32 Kバイト ( $\mu$ PD784044, 784044(A), (A1), (A2), 784054, 784054(A), (A1),  
 (A2))

フラッシュ・メモリ : 64 Kバイト ( $\mu$ PD78F4046)

RAM : 2048バイト ( $\mu$ PD784046, 784046(A), (A1), (A2), 78F4046)  
 1024バイト ( $\mu$ PD784044, 784044(A), (A1), (A2), 784054, 784054(A), (A1),  
 (A2))

I/Oポート : 65本 ( $\mu$ PD784054, 784054(A), (A1), (A2)のみ64本)

タイマ/カウンタ : 16ビット・タイマ/カウンタ×2ユニット  
 16ビット・タイマ×3ユニット  
 ( $\mu$ PD784054, 784054(A), (A1), (A2)は, 16ビット・タイマ×3ユニットのみ)

ウォッチドッグ・タイマ : 1チャンネル

A/Dコンバータ : 10ビット分解能×16チャンネル ( $AV_{DD} = 4.5 \sim 5.5$  V)

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル (ポーレート・ジェネレータ内蔵)

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード (ノスタンバイ機能無効モード... $\mu$ PD784054, 784054(A), (A1), (A2)のみ)

電源電圧 :  $V_{DD} = 4.0 \sim 5.5$  V

### 1.5.2 応用分野

標準品 : 給湯器, 自動販売機など, PPC, プリンタなどのOA分野, ロボット, 自動工作機械などのFA分野

特別品 : 自動車電装など

## 1.5.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784044GC- x x x -3B9	80ピン・プラスチックQFP (14 x 14 mm)	マスクROM
μ PD784044GC(A)- x x x -3B9	"	"
μ PD784044GC(A1)- x x x -3B9	"	"
μ PD784044GC(A2)- x x x -3B9	"	"
μ PD784046GC- x x x -3B9 <sup>注</sup>	"	"
μ PD784046GC(A)- x x x -3B9 <sup>注</sup>	"	"
μ PD784046GC(A1)- x x x -3B9 <sup>注</sup>	"	"
μ PD784046GC(A2)- x x x -3B9 <sup>注</sup>	"	"
μ PD784054GC- x x x -3B9	"	"
μ PD784054GC(A)- x x x -3B9	"	"
μ PD784054GC(A1)- x x x -3B9	"	"
μ PD784054GC(A2)- x x x -3B9	"	"
μ PD78F4046GC-3B9 <sup>注</sup>	"	フラッシュ・メモリ

備考 x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784044GC- x x x -3B9	80ピン・プラスチックQFP (14 x 14 mm)	標準 (一般電子機器用)
μ PD784046GC- x x x -3B9 <sup>注</sup>	"	"
μ PD784054GC- x x x -3B9	"	"
μ PD78F4046GC-3B9 <sup>注</sup>	"	"
μ PD784044GC(A)- x x x -3B9	"	特別 (高信頼度電子機器用)
μ PD784044GC(A1)- x x x -3B9	"	"
μ PD784044GC(A2)- x x x -3B9	"	"
μ PD784046GC(A)- x x x -3B9 <sup>注</sup>	"	"
μ PD784046GC(A1)- x x x -3B9 <sup>注</sup>	"	"
μ PD784046GC(A2)- x x x -3B9 <sup>注</sup>	"	"
μ PD784054GC(A)- x x x -3B9	"	"
μ PD784054GC(A1)- x x x -3B9	"	"
μ PD784054GC(A2)- x x x -3B9	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

注 開発中

備考 x x x はROMコード番号です。

### 1.5.4 機能概要

(1)  $\mu$  PD784044, 784044(A), (A1), (A2), 784046, 784046(A), (A1), (A2), 78F4046

(1/2)

品名		$\mu$ PD784044, 784044(A), (A1), (A2)	$\mu$ PD784046, 784046(A), (A1), (A2)	$\mu$ PD78F4046
項目				
基本命令(ニモニック)数		113		
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)		
最小命令実行時間		125 ns(内部クロック: 16 MHz動作時) ... $\mu$ PD784044, 78F4046 160 ns(内部クロック: 12.5 MHz動作時) ... $\mu$ PD784044(A), 784046(A) 200 ns(内部クロック: 10 MHz動作時) ... $\mu$ PD784044(A1), (A2), 784046(A1), (A2)		
内部メモリ	ROM	32 Kバイト (マスクROM)	64 Kバイト (マスクROM)	64 Kバイト (フラッシュ・メモリ)
	RAM	1024バイト	2048バイト	
メモリ空間		プログラム/データあわせて1 Mバイト		
I/Oポート	合計	65本		
	入力	17本		
	入出力	48本		
注 付加機能 付き端子	プルアップ抵抗付き 端子	29本		
リアルタイム出力ポート		4ビット×1		
タイマ/カウンタ		タイマ0	: タイマ・レジスタ×1 キャプチャ/コンペア・レジスタ×4	パルス出力可 ・トグル出力 ・セット/リセット出力
		タイマ1	: タイマ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・トグル出力 ・セット/リセット出力
		タイマ/カウンタ2	: タイマ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・トグル出力 ・PWM/PPG出力
		タイマ/カウンタ3	: タイマ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・トグル出力 ・PWM/PPG出力
		タイマ4	: タイマ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・リアルタイム出力(4ビット×1)
A/Dコンバータ		10ビット分解能×16チャンネル ( $AV_{DD} = 4.5 \sim 5.5 V$ )		
シリアル・インタフェース		UART/IOE (3線式シリアルI/O) : 2チャンネル (ポーレート・ジェネレータ内蔵)		
ウォッチドッグ・タイマ		1チャンネル		

注 付加機能付き端子は, I/O端子の中に含まれています。

品名		$\mu$ PD784044, 784044(A), (A1), (A2)	$\mu$ PD784046, 784046(A), (A1), (A2)	$\mu$ PD78F4046
割り込み	ハードウェア要因	27 (内部: 23, 外部: 8 (内部兼用: 4))		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスクابل	内部: 1本, 外部: 1本		
	マスクابل	内部: 22本, 外部: 7本 (内部兼用: 4本)		
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態: ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング		
バス・サイジング機能		8ビット / 16ビット外部データ・バス幅選択可能		
スタンバイ		HALT/STOP/IDLEモード		
電源電圧		$V_{DD} = 4.0 \sim 5.5 V$		
パッケージ		80ピン・プラスチックQFP (14 x 14 mm)		

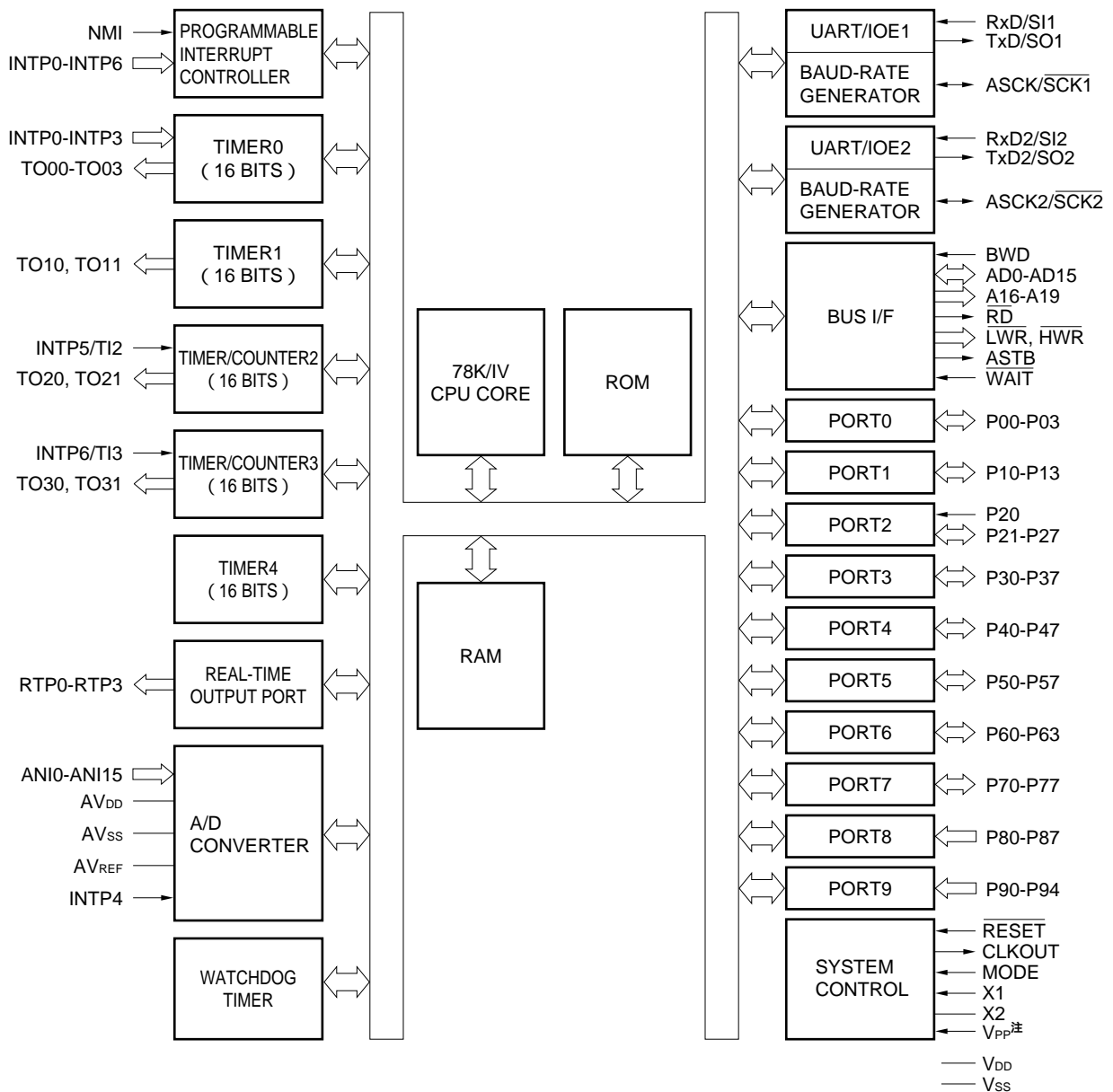
(2)  $\mu$ PD784054, 784054(A), (A1), (A2)

品名		$\mu$ PD784054, 784054(A), (A1), (A2)	
項目			
基本命令(ニモニック)数		113	
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)	
最小命令実行時間		125 ns (内部クロック: 16 MHz動作時) ... $\mu$ PD784054 160 ns (内部クロック: 12.5 MHz動作時) ... $\mu$ PD784054(A) 200 ns (内部クロック: 10 MHz動作時) ... $\mu$ PD784054(A1), (A2)	
内部メモリ	ROM	32 Kバイト(マスクROM)	
	RAM	1024バイト	
メモリ空間		プログラム/データあわせて1 Mバイト	
I/Oポート	合計	64本	
	入力	17本	
	入出力	47本	
付加機能付き端子	ブルアップ抵抗付き端子	29本	
タイマ	タイマ0 (16ビット)	タイマ・レジスタ×1 キャプチャ/コンペア・レジスタ×4	パルス出力可 ・トグル出力 ・セット/リセット出力
	タイマ1 (16ビット)	タイマ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・トグル出力 ・セット/リセット出力
	タイマ4 (16ビット)	タイマ・レジスタ×1 コンペア・レジスタ×2	
A/Dコンバータ		10ビット分解能×16チャンネル ( $AV_{DD} = 4.5 \sim 5.5 V$ )	
シリアル・インタフェース		UART/IOE (3線式シリアルI/O) : 2チャンネル (ポーレート・ジェネレータ内蔵)	
ウォッチドッグ・タイマ		1チャンネル	
割り込み	ハードウェア要因	23 (内部: 19, 外部: 8 (内部兼用: 4))	
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー	
	ノンマスカブル	内部: 1本, 外部: 1本	
	マスカブル	内部: 18本, 外部: 7本 (内部兼用: 4本) ・4レベルのプログラマブル・プライオリティ ・3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング	
バス・サイジング		8ビット/16ビット外部データ・バス幅選択可能	
スタンバイ		HALT/STOP/IDLE/スタンバイ機能無効モード	
電源電圧		$V_{DD} = 4.0 \sim 5.5 V$	
パッケージ		80ピン・プラスチックQFP (14×14 mm)	

注 付加機能付き端子は, I/O端子の中に含まれています。

### 1.5.5 ブロック図

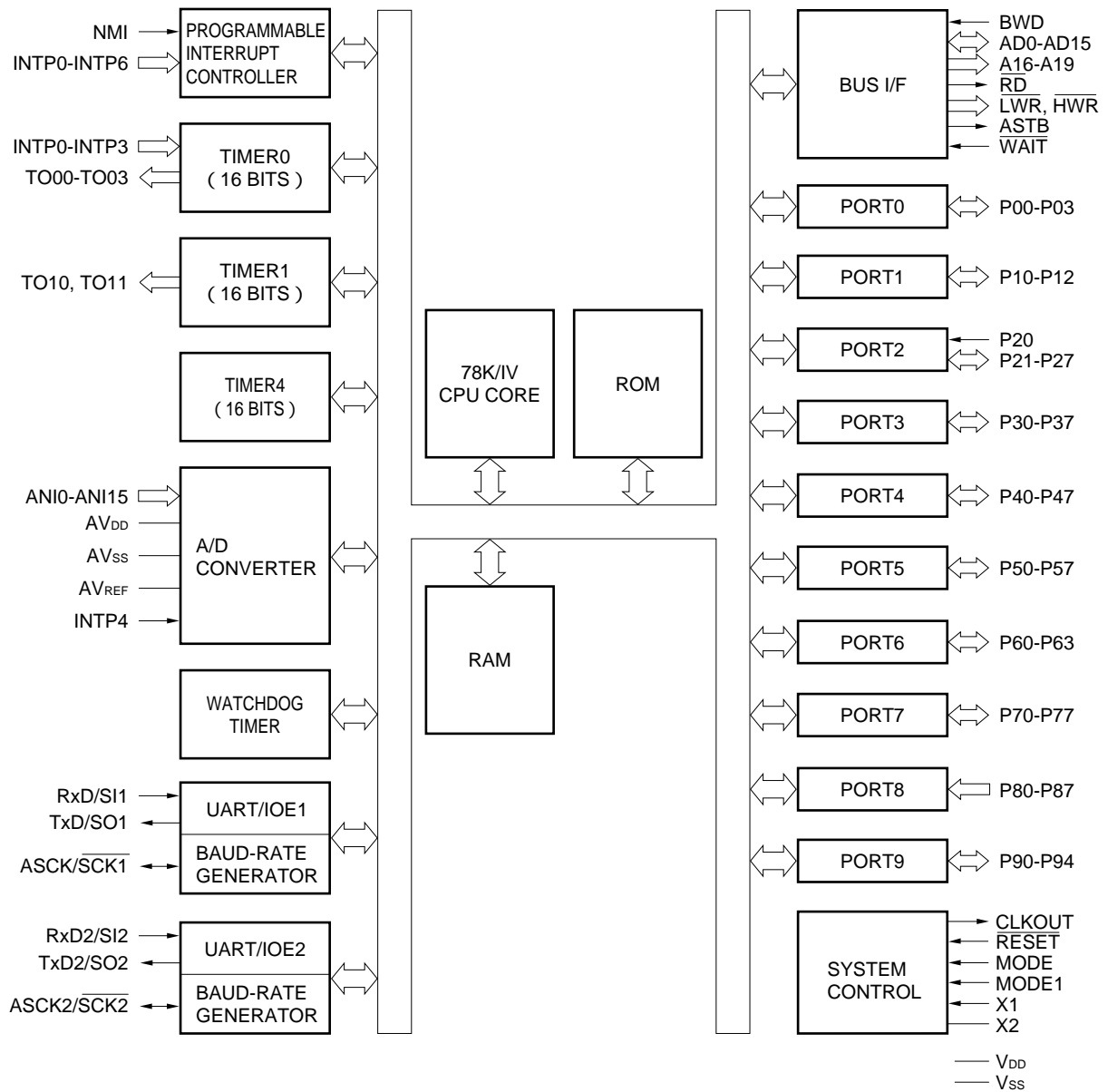
(1)  $\mu$  PD784044, 784044(A), (A1), (A2), 784046, 784046(A), (A1), (A2), 78F4046



注 V<sub>PP</sub>端子は $\mu$  PD78F4046のみ。

備考 内部ROM, RAM容量は製品によって異なります。

(2)  $\mu$  PD784054, 784054(A), (A1), (A2)



## ★ 1.6 $\mu$ PD784216Aサブシリーズ製品概要 ( $\mu$ PD784214A, 784215A, 784216A, 78F4216A)

### 1.6.1 特 徴

$\mu$ PD78078サブシリーズの周辺機能を継承

最小命令実行時間：160 ns (メイン・システム・クロック12.5 MHz動作時)

61  $\mu$ s (サブシステム・クロック32.768 kHz動作時)

内部メモリ

ROM

マスクROM : 96 Kバイト ( $\mu$ PD784214A)  
128 Kバイト ( $\mu$ PD784215A, 784216A)

フラッシュ・メモリ : 128 Kバイト ( $\mu$ PD78F4216A)

RAM : 3584バイト ( $\mu$ PD784214A)  
5120バイト ( $\mu$ PD784215A)  
8192バイト ( $\mu$ PD784216A, 78F4216A)

I/Oポート：86本

タイマ/カウンタ：16ビット・タイマ/カウンタ×1ユニット

8ビット・タイマ/カウンタ×6ユニット

時計用タイマ：1チャンネル

ウォッチドッグ・タイマ：1チャンネル

A/Dコンバータ：8ビット分解能×8チャンネル

D/Aコンバータ：8ビット分解能×2チャンネル

シリアル・インタフェース：3チャンネル

UART/IOE (3線式シリアルI/O)：2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O)：1チャンネル

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

クロック出力機能

$f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/2^2$ ,  $f_{xx}/2^3$ ,  $f_{xx}/2^4$ ,  $f_{xx}/2^5$ ,  $f_{xx}/2^6$ ,  $f_{xx}/2^7$ ,  $f_{XT}$ から選択

ブザー出力機能

$f_{xx}/2^{10}$ ,  $f_{xx}/2^{11}$ ,  $f_{xx}/2^{12}$ ,  $f_{xx}/2^{13}$ から選択

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時：HALT/IDLEモード (サブシステム・クロック動作時)

電源電圧： $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$ PD784214A, 784215A, 784216A)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$ PD78F4216A)

### 1.6.2 応用分野

携帯電話, PHS, コードレス電話, CD-ROM, AV機器など

## 1.6.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784214AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	マスクROM
μ PD784214AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD784215AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	"
μ PD784215AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD784216AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	"
μ PD784216AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD78F4216AGC-8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	フラッシュ・メモリ
μ PD78F4216AGF-3BA	100ピン・プラスチックQFP (14 × 20 mm)	"

備考 x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784214AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	標準 (一般電子機器用)
μ PD784214AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD784215AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	"
μ PD784215AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD784216AGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	"
μ PD784216AGF- x x x -3BA	100ピン・プラスチックQFP (14 × 20 mm)	"
μ PD78F4216AGC-8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 × 14 mm)	"
μ PD78F4216AGF-3BA	100ピン・プラスチックQFP (14 × 20 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

備考 x x x はROMコード番号です。

1.6.4 機能概要

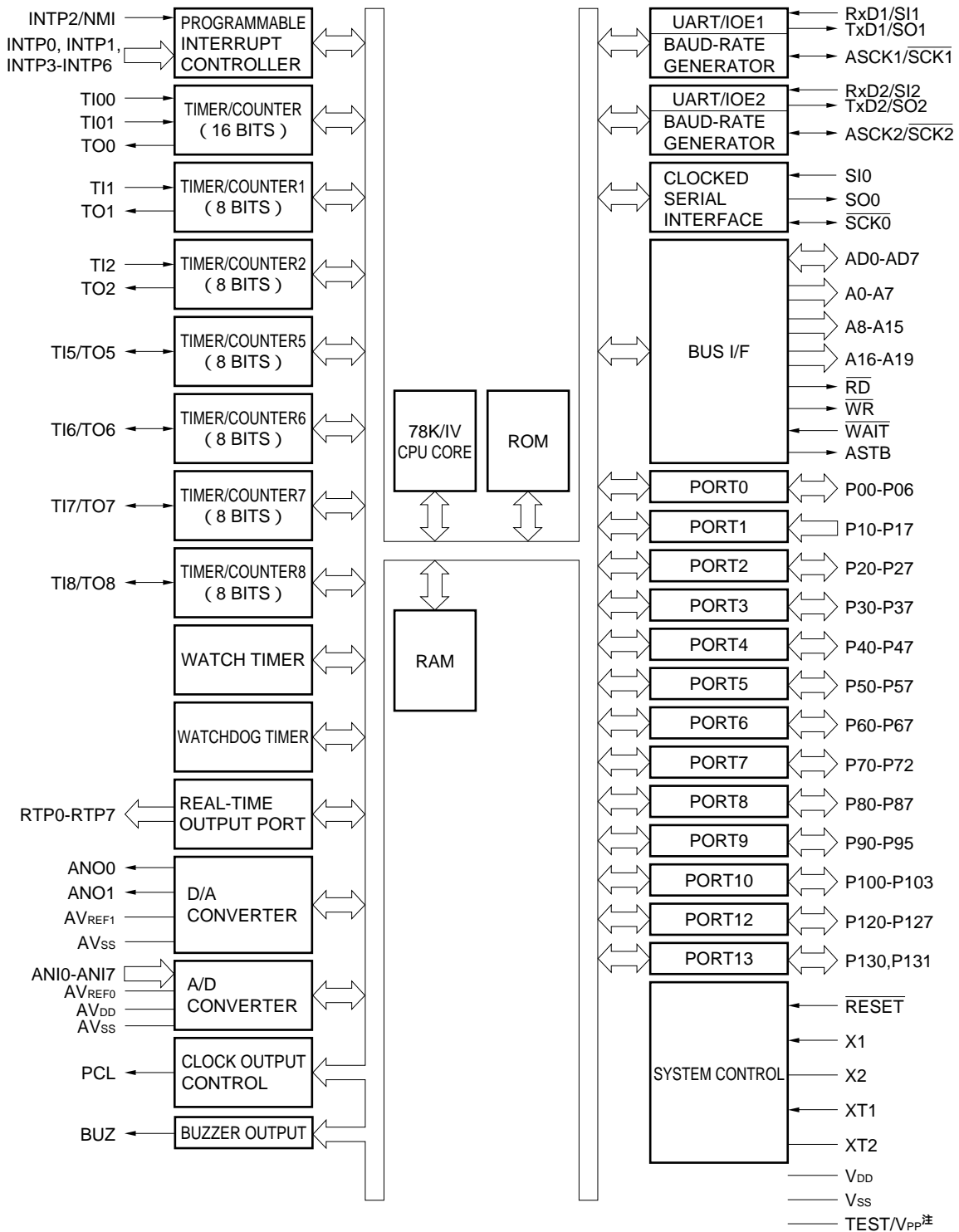
(1/2)

品名		μ PD784214A	μ PD784215A	μ PD784216A	μ PD78F4216A
項目					
基本命令(ニモニック)数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)			
最小命令実行時間	メイン・システム・クロック選択時	160 ns/320 ns/640 ns/1280 ns/2560 ns (12.5 MHz動作時)			
	サブシステム・クロック選択時	61 μs (32.768 kHz動作時)			
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)	
	RAM	3584バイト	5120バイト	8192バイト	
メモリ空間		プログラム/データあわせて1 Mバイト			
I/Oポート	合計	86本			
	CMOS入力	2本			
	CMOS入出力	72本			
	N-chオープン・ドレイン入出力	6本			
	付加機能付き端子 <sup>注</sup>	プルアップ抵抗付き端子	70本		
LEDダイレクト・ドライブ出力		22本			
中耐圧端子		6本			
リアルタイム出力ポート		4ビット×2, または8ビット×1			

注 付加機能付き端子は、I/O端子の中に含まれています。

品名		μ PD784214A	μ PD784215A	μ PD784216A	μ PD78F4216A
タイマ/カウンタ		タイマ/カウンタ (16ビット)	: タイマ・レジスタ×1 キャプチャ/コンペア・レジスタ×2		パルス出力可 ・PWM/PPG出力 ・方形波出力 ・ワンショット・パルス出力
		タイマ/カウンタ1 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
		タイマ/カウンタ2 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
		タイマ/カウンタ5 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
		タイマ/カウンタ6 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
		タイマ/カウンタ7 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
		タイマ/カウンタ8 (8ビット)	: タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力
A/Dコンバータ		8ビット分解能×8チャンネル			
D/Aコンバータ		8ビット分解能×2チャンネル			
シリアル・インタフェース		UART×IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵) CSI (3線式シリアルI/O) : 1チャンネル			
クロック出力		$f_{xx}$ , $f_{xx}/2$ , $f_{xx}/2^2$ , $f_{xx}/2^3$ , $f_{xx}/2^4$ , $f_{xx}/2^5$ , $f_{xx}/2^6$ , $f_{xx}/2^7$ , $f_{XT}$ から選択			
ブザー出力		$f_{xx}/2^{10}$ , $f_{xx}/2^{11}$ , $f_{xx}/2^{12}$ , $f_{xx}/2^{13}$ から選択			
時計用タイマ		1チャンネル			
ウォッチドッグ・タイマ		1チャンネル			
割り込み	ハードウェア要因	29 (内部: 20, 外部: 9)			
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー			
	ノンマスカブル	内部: 1本, 外部: 1本			
	マスカブル	内部: 19本, 外部: 8本			
		<ul style="list-style-type: none"> <li>・4レベルのプログラマブル・プライオリティ</li> <li>・3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング</li> </ul>			
スタンバイ		<ul style="list-style-type: none"> <li>・HALT/STOP/IDLEモード</li> <li>・低消費電力モード (サブシステム・クロックによるCPU動作) 時: HALT/IDLEモード</li> </ul>			
電源電圧		$V_{DD} = 1.8 \sim 5.5 V$		$V_{DD} = 1.9 \sim 5.5 V$	
パッケージ		<ul style="list-style-type: none"> <li>・100ピン・プラスチックLQFP (ファインピッチ) (14×14 mm)</li> <li>・100ピン・プラスチックQFP (14×20 mm)</li> </ul>			

1.6.5 ブロック図



注 V<sub>PP</sub>は、μ PD78F4216Aのみ。

備考 内部ROM, RAM容量は製品によって異なります。

## ★ 1.7 $\mu$ PD784216AYサブシリーズ製品概要 ( $\mu$ PD784214AY, 784215AY, 784216AY, 78F4216AY)

### 1.7.1 特 徴

$\mu$ PD784216AサブシリーズにI<sup>2</sup>Cバス・インタフェースを追加

最小命令実行時間：160 ns (メイン・システム・クロック：12.5 MHz動作時)

61  $\mu$ s (サブシステム・クロック：32.768 kHz動作時)

内部メモリ

ROM

マスクROM : 96 Kバイト ( $\mu$ PD784214AY)  
128 Kバイト ( $\mu$ PD784215AY, 784216AY)

フラッシュ・メモリ : 128 Kバイト ( $\mu$ PD78F4216AY)

RAM : 3584バイト ( $\mu$ PD784214AY)  
5120バイト ( $\mu$ PD784215AY)  
8192バイト ( $\mu$ PD784216AY, 78F4216AY)

I/Oポート：86本

タイマ/カウンタ：16ビット・タイマ/カウンタ×1ユニット

8ビット・タイマ/カウンタ×6ユニット

時計用タイマ：1チャンネル

ウォッチドッグ・タイマ：1チャンネル

A/Dコンバータ：8ビット分解能×8チャンネル

D/Aコンバータ：8ビット分解能×2チャンネル

シリアル・インタフェース：3チャンネル

UART/IOE (3線式シリアルI/O)：2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O, マルチマスタ対応I<sup>2</sup>Cバス)：1チャンネル

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング

クロック出力機能

$f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/2^2$ ,  $f_{xx}/2^3$ ,  $f_{xx}/2^4$ ,  $f_{xx}/2^5$ ,  $f_{xx}/2^6$ ,  $f_{xx}/2^7$ ,  $f_{XT}$ から選択

ブザー出力機能

$f_{xx}/2^{10}$ ,  $f_{xx}/2^{11}$ ,  $f_{xx}/2^{12}$ ,  $f_{xx}/2^{13}$ から選択

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時：HALT/IDLEモード (サブシステム・クロック動作時)

電源電圧： $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$ PD784214AY, 784215AY, 784216AY)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$ PD78F4216AY)

### 1.7.2 応用分野

携帯電話, PHS, コードレス電話, CD-ROM, AV機器など

## 1.7.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784214AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	マスクROM
$\mu$ PD784214AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784215AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784215AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784216AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784216AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD78F4216AYGC-8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	フラッシュ・メモリ
$\mu$ PD78F4216AYGF-3BA	100ピン・プラスチックQFP (14 x 20 mm)	"

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784214AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	標準 (一般電子機器用)
$\mu$ PD784214AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784215AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784215AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784216AYGC- x x x -8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784216AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD78F4216AYGC-8EU	100ピン・プラスチックLQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD78F4216AYGF-3BA	100ピン・プラスチックQFP (14 x 20 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

1.7.4 機能概要

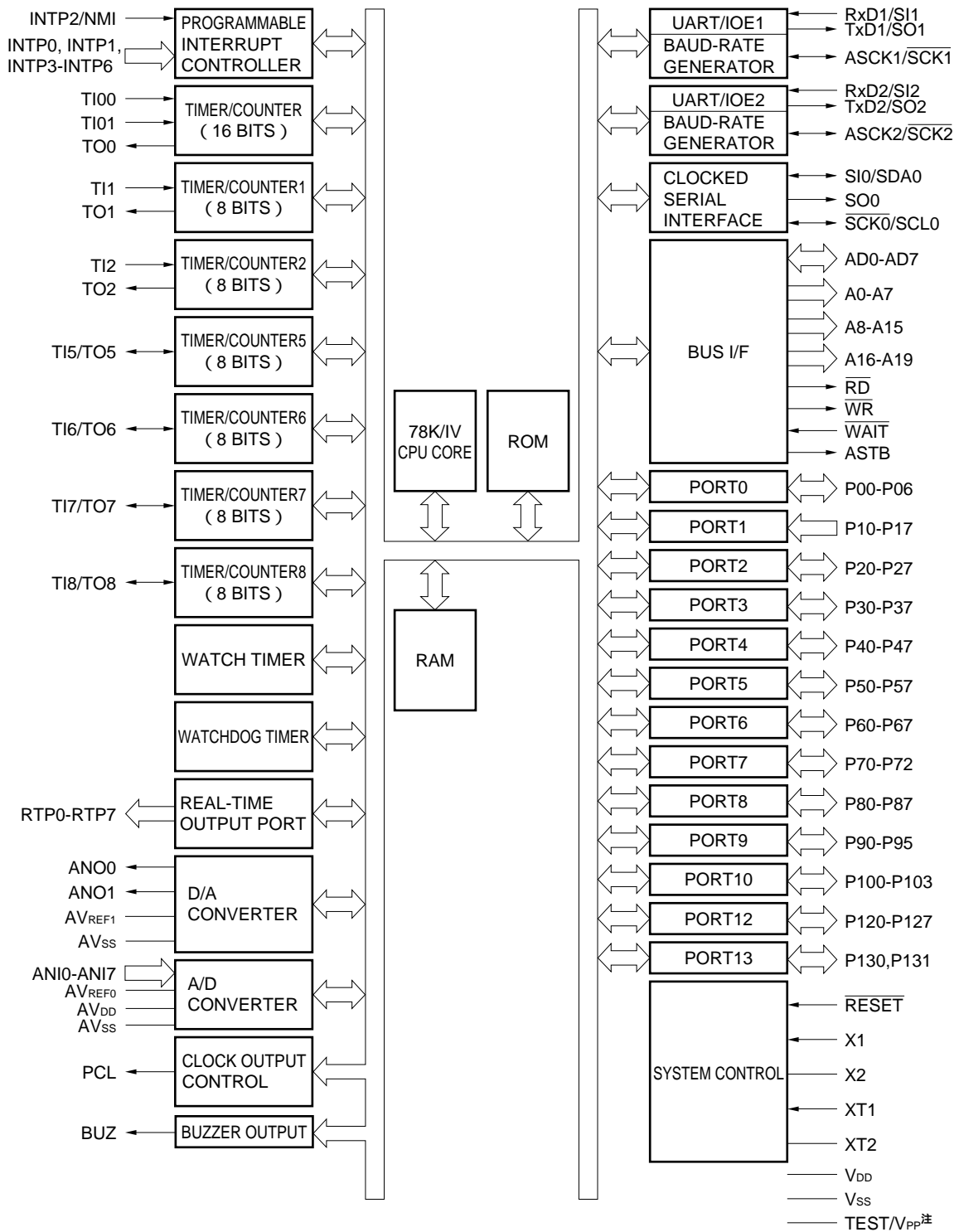
(1/2)

品名		μ PD784214AY	μ PD784215AY	μ PD784216AY	μ PD78F4216AY
項目					
基本命令(ニモニック)数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)			
最小命令実行時間	メイン・システム・クロック選択時	160 ns/320 ns/640 ns/1280 ns/2560 ns (12.5 MHz動作時)			
	サブシステム・クロック選択時	61 μs (32.768 kHz動作時)			
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)	
	RAM	3584バイト	5120バイト	8192バイト	
メモリ空間		プログラム/データあわせて1 Mバイト			
I/Oポート	合計	86本			
	CMOS入力	2本			
	CMOS入出力	72本			
	N-chオープン・ドレイン入出力	6本			
	注 付加機能付き端子	プルアップ抵抗付き端子	70本		
LEDダイレクト・ドライブ出力		22本			
中耐圧端子		6本			
リアルタイム出力ポート		4ビット×2, または8ビット×1			

注 付加機能付き端子は, I/O端子の中に含まれています。

品 名		μ PD784214AY	μ PD784215AY	μ PD784216AY	μ PD78F4216AY
タイマ/カウンタ	タイマ/カウンタ (16ビット)	タイマ・レジスタ×1 キャプチャ/コンペア・レジスタ×2		パルス出力可 ・PWM/PPG出力 ・方形波出力 ・ワンショット・パルス出力	
	タイマ/カウンタ1 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	タイマ/カウンタ2 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	タイマ/カウンタ5 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	タイマ/カウンタ6 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	タイマ/カウンタ7 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	タイマ/カウンタ8 (8ビット)	タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・PWM出力 ・方形波出力	
	A/Dコンバータ	8ビット分解能×8チャンネル			
D/Aコンバータ	8ビット分解能×2チャンネル				
シリアル・インタフェース	UART×IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵) CSI (3線式シリアルI/O, マルチマスタ対応I <sup>2</sup> Cバス) : 1チャンネル				
クロック出力	f <sub>xx</sub> , f <sub>xx</sub> /2, f <sub>xx</sub> /2 <sup>2</sup> , f <sub>xx</sub> /2 <sup>3</sup> , f <sub>xx</sub> /2 <sup>4</sup> , f <sub>xx</sub> /2 <sup>5</sup> , f <sub>xx</sub> /2 <sup>6</sup> , f <sub>xx</sub> /2 <sup>7</sup> , f <sub>XT</sub> から選択				
ブザー出力	f <sub>xx</sub> /2 <sup>10</sup> , f <sub>xx</sub> /2 <sup>11</sup> , f <sub>xx</sub> /2 <sup>12</sup> , f <sub>xx</sub> /2 <sup>13</sup> から選択				
時計用タイマ	1チャンネル				
ウォッチドッグ・タイマ	1チャンネル				
割り込み	ハードウェア要因	29 (内部 : 20, 外部 : 9)			
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー			
	ノンマスクابل	内部 : 1本, 外部 : 1本			
	マスクابل	内部 : 19本, 外部 : 8本			
		<ul style="list-style-type: none"> <li>・4レベルのプログラマブル・プライオリティ</li> <li>・3種類の処理形態 : ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング</li> </ul>			
スタンバイ	<ul style="list-style-type: none"> <li>・HALT/STOP/IDLEモード</li> <li>・低消費電力モード (サブシステム・クロックによるCPU動作) 時 : HALT/IDLEモード</li> </ul>				
電源電圧	V <sub>DD</sub> = 1.8 ~ 5.5 V			V <sub>DD</sub> = 1.9 ~ 5.5 V	
パッケージ	<ul style="list-style-type: none"> <li>・100ピン・プラスチックLQFP (ファインピッチ) (14×14 mm)</li> <li>・100ピン・プラスチックQFP (14×20 mm)</li> </ul>				

### 1.7.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4216AYのみ。

備考 内部ROM, RAM容量は製品によって異なります。

## ★ 1.8 $\mu$ PD784218Aサブシリーズ製品概要 ( $\mu$ PD784217A, 784218A, 78F4218A)

### 1.8.1 特 徴

ROMコレクション内蔵

$\mu$ PD78078サブシリーズの周辺機能を継承

最小命令実行時間

160 ns (メイン・システム・クロック :  $f_{XX} = 12.5$  MHz動作時)

61  $\mu$ s (サブシステム・クロック :  $f_{XT} = 32.768$  kHz動作時)

制御用途に適する命令セット

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み処理 / マクロ・サービス / コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時 : HALT/IDLEモード (サブシステム・クロック動作時)

内部メモリ : マスクROM 256 Kバイト ( $\mu$ PD784218A)

192 Kバイト ( $\mu$ PD784217A)

フラッシュ・メモリ 256 Kバイト ( $\mu$ PD78F4218A)

RAM 12800バイト

I/O端子 : 86本

ソフトウエア・プログラマブル・プルアップ : 70入力

LEDダイレクト・ドライブ可能 : 22出力

トランジスタ・ダイレクト・ドライブ可能 : 6出力

タイマ / カウンタ : 16ビット・タイマ / カウンタ  $\times$  1ユニット

8ビット・タイマ / カウンタ  $\times$  6ユニット

時計用タイマ : 1チャンネル

ウォッチドッグ・タイマ : 1チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O) : 1チャンネル

A/Dコンバータ : 8ビット分解能  $\times$  8チャンネル

D/Aコンバータ : 8ビット分解能  $\times$  2チャンネル

リアルタイム出力ポート (タイマ / カウンタと組み合わせることにより, 2系統のステッピング・モータの独立制御が可能)

クロック分周機能

クロック出力機能 :  $f_{XX}$ ,  $f_{XX}/2$ ,  $f_{XX}/2^2$ ,  $f_{XX}/2^3$ ,  $f_{XX}/2^4$ ,  $f_{XX}/2^5$ ,  $f_{XX}/2^6$ ,  $f_{XX}/2^7$ ,  $f_{XT}$ から選択

ブザー出力機能 :  $f_{XX}/2^{10}$ ,  $f_{XX}/2^{11}$ ,  $f_{XX}/2^{12}$ ,  $f_{XX}/2^{13}$ から選択

外部アクセス・ステータス機能

電源電圧 :  $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$ PD784217A, 784218A)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$ PD78F4218A)

## 1.8.2 応用分野

携帯電話，PHS，コードレス電話，CD-ROM，AV機器など

## 1.8.3 オーダ情報と品質水準

### (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784217AGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	マスクROM
μ PD784217AGF- x x x -3BA	100ピン・プラスチックQFP (14×20 mm)	"
μ PD784218AGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	"
μ PD784218AGF- x x x -3BA	100ピン・プラスチックQFP (14×20 mm)	"
μ PD78F4218AGC-7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	フラッシュ・メモリ
μ PD78F4218AGF-3BA	100ピン・プラスチックQFP (14×20 mm)	"

**備考** x x x はROMコード番号です。

### (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784217AGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	標準 (一般電子機器用)
μ PD784217AGF- x x x -3BA	100ピン・プラスチックQFP (14×20 mm)	"
μ PD784218AGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	"
μ PD784218AGF- x x x -3BA	100ピン・プラスチックQFP (14×20 mm)	"
μ PD78F4218AGC-7EA	100ピン・プラスチックQFP (ファインピッチ) (14×14 mm)	"
μ PD78F4218AGF-3BA	100ピン・プラスチックQFP (14×20 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

## 1.8.4 機能概要

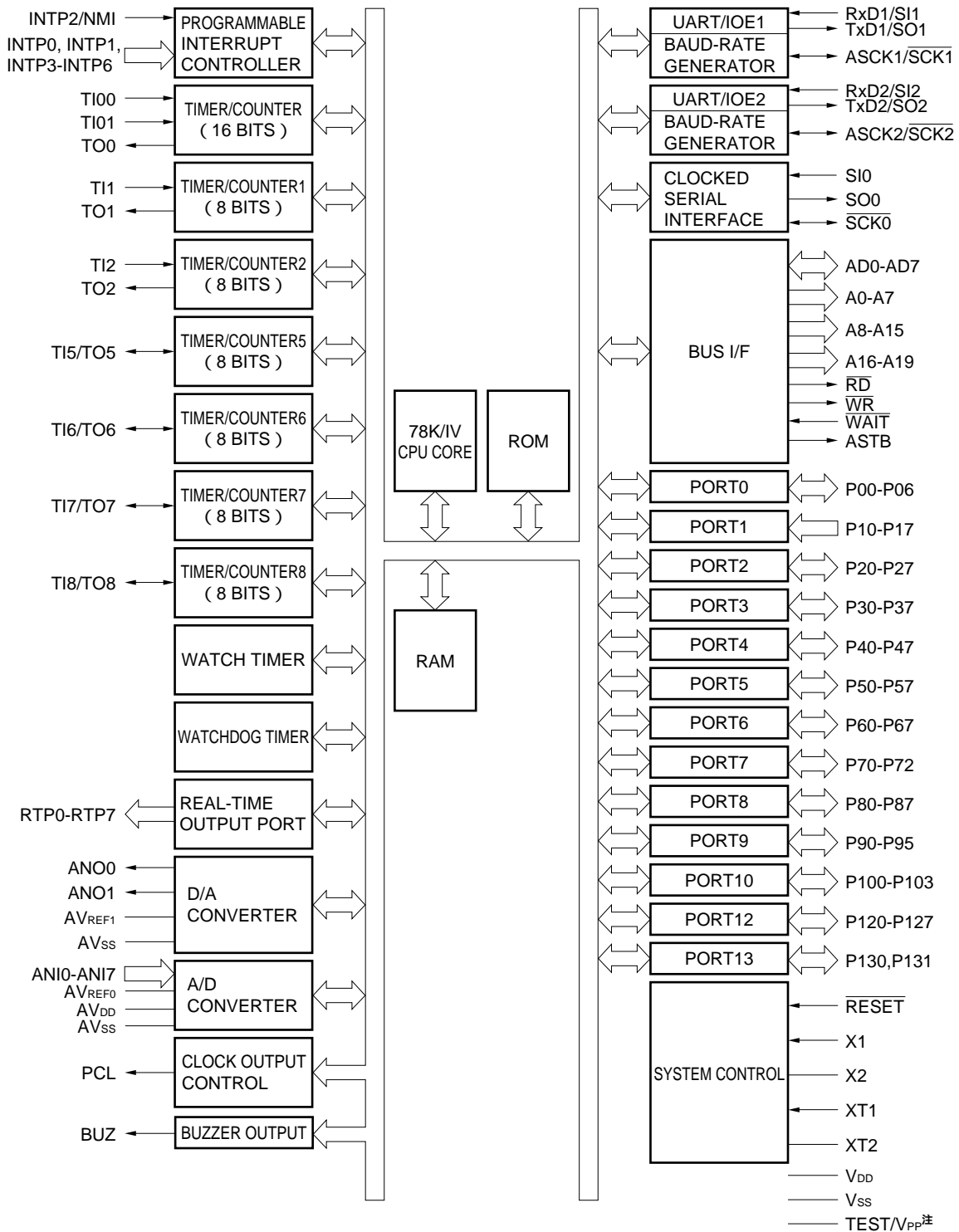
(1/2)

品名		$\mu$ PD784217A	$\mu$ PD784218A	$\mu$ PD78F4218A	
項目					
基本命令(ニモニック)数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)			
最小命令実行時間		・160 ns/320 ns/640 ns/1280 ns/2560 ns(メイン・システム・クロック: 12.5 MHz動作時) ・61 $\mu$ s(サブシステム・クロック: 32.768 kHz動作時)			
内部メモリ	ROM	192 Kバイト (マスクROM)	256 Kバイト (マスクROM)	256 Kバイト (フラッシュ・メモリ)	
	RAM	12800バイト			
メモリ空間		プログラム/データあわせて1 Mバイト			
I/Oポート	合計	86本			
	CMOS入力	8本			
	CMOS入出力	72本			
	N-chオープン・ドレ ーン入出力	6本			
	注 付 加 機 能 付 き 端 子	プルアップ抵抗付き 端子	70本		
		LEDダイレクト・ド ライブ出力	22本		
		中耐圧端子	6本		
リアルタイム出力ポート		4ビット×2, または8ビット×1			
タイマ/カウンタ		タイマ/カウンタ : タイマ・レジスタ×1 (16ビット)      キャプチャ/コンペア・レジスタ×2	パルス出力可 ・PWM/PPG出力 ・方形波出力 ・ワンショット・パルス出力		
		タイマ/カウンタ1 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		
		タイマ/カウンタ2 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		
		タイマ/カウンタ5 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		
		タイマ/カウンタ6 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		
		タイマ/カウンタ7 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		
		タイマ/カウンタ8 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力		

注 付加機能付き端子は, I/O端子の中に含まれています。

品名		$\mu$ PD784217A	$\mu$ PD784218A	$\mu$ PD78F4218A
項目				
シリアル・インタフェース		<ul style="list-style-type: none"> <li>・ UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)</li> <li>・ CSI (3線式シリアルI/O) : 1チャンネル</li> </ul>		
A/Dコンバータ		8ビット分解能 × 8チャンネル		
D/Aコンバータ		8ビット分解能 × 2チャンネル		
クロック出力		$f_{xx}$ , $f_{xx}/2$ , $f_{xx}/2^2$ , $f_{xx}/2^3$ , $f_{xx}/2^4$ , $f_{xx}/2^5$ , $f_{xx}/2^6$ , $f_{xx}/2^7$ , $f_{XT}$ から選択		
プザー出力		$f_{xx}/2^{10}$ , $f_{xx}/2^{11}$ , $f_{xx}/2^{12}$ , $f_{xx}/2^{13}$ から選択		
時計用タイマ		1チャンネル		
ウォッチドッグ・タイマ		1チャンネル		
スタンバイ		<ul style="list-style-type: none"> <li>・ HALT/STOP/IDLEモード</li> <li>・ 低消費電力モード (サブシステム・クロックによるCPU動作) 時: HALT/IDLEモード</li> </ul>		
割り込み	ハードウェア要因	29 (内部: 20, 外部: 9)		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスカブル	内部: 1本, 外部: 1本		
	マスカブル	内部: 19本, 外部: 8本		
		<ul style="list-style-type: none"> <li>・ 4レベルのプログラマブル・プライオリティ</li> <li>・ 3種類の処理形態: ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング</li> </ul>		
電源電圧		$V_{DD} = 1.8 \sim 5.5 V$		$V_{DD} = 1.9 \sim 5.5 V$
パッケージ		<ul style="list-style-type: none"> <li>・ 100ピン・プラスチックQFP (ファインピッチ) (14 × 14 mm)</li> <li>・ 100ピン・プラスチックQFP (14 × 20 mm)</li> </ul>		

1.8.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4218Aのみ

備考 内部ROM容量は製品によって異なります。

## ★ 1.9 $\mu$ PD784218AYサブシリーズ製品概要 ( $\mu$ PD784217AY, 784218AY, 78F4218AY)

### 1.9.1 特 徴

$\mu$ PD784218AサブシリーズにI<sup>2</sup>Cバス・インタフェースを追加

ROMコレクション内蔵

$\mu$ PD78078Yサブシリーズの周辺機能を継承

最小命令実行時間

160 ns (メイン・システム・クロック :  $f_{XX} = 12.5$  MHz動作時)

61  $\mu$ s (サブシステム・クロック :  $f_{XT} = 32.768$  kHz動作時)

制御用途に適する命令セット

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み処理 / マクロ・サービス / コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時 : HALT/IDLEモード (サブシステム・クロック動作時)

内部メモリ : マスクROM                    256 Kバイト ( $\mu$ PD784218AY)

192 Kバイト ( $\mu$ PD784217AY)

フラッシュ・メモリ    256 Kバイト ( $\mu$ PD78F4218AY)

RAM    12800バイト

I/O端子 : 86本

ソフトウエア・プログラマブル・プルアップ : 70入力

LEDダイレクト・ドライブ可能    : 22出力

トランジスタ・ダイレクト・ドライブ可能    : 6出力

タイマ / カウンタ : 16ビット・タイマ / カウンタ  $\times$  1ユニット

8ビット・タイマ / カウンタ  $\times$  6ユニット

時計用タイマ : 1チャンネル

ウォッチドッグ・タイマ : 1チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O, マルチマスタ対応I<sup>2</sup>Cバス) : 1チャンネル

A/Dコンバータ : 8ビット分解能  $\times$  8チャンネル

D/Aコンバータ : 8ビット分解能  $\times$  2チャンネル

リアルタイム出力ポート (タイマ / カウンタと組み合わせることにより, 2系統のステッピング・モータの独立制御が可能)

クロック分周機能

クロック出力機能 :  $f_{XX}$ ,  $f_{XX}/2$ ,  $f_{XX}/2^2$ ,  $f_{XX}/2^3$ ,  $f_{XX}/2^4$ ,  $f_{XX}/2^5$ ,  $f_{XX}/2^6$ ,  $f_{XX}/2^7$ ,  $f_{XT}$ から選択

ブザー出力機能 :  $f_{XX}/2^{10}$ ,  $f_{XX}/2^{11}$ ,  $f_{XX}/2^{12}$ ,  $f_{XX}/2^{13}$ から選択

外部アクセス・ステータス機能

★ 電源電圧 :  $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$ PD784217AY, 784218AY)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$ PD78F4218AY)

## 1.9.2 応用分野

携帯電話，PHS，コードレス電話，CD-ROM，AV機器など

## 1.9.3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784217AYGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	マスクROM
$\mu$ PD784217AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784218AYGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784218AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD78F4218AYGC-7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	フラッシュ・メモリ
$\mu$ PD78F4218AYGF-3BA	100ピン・プラスチックQFP (14 x 20 mm)	"

備考 x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784217AYGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	標準 (一般電子機器用)
$\mu$ PD784217AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD784218AYGC- x x x -7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD784218AYGF- x x x -3BA	100ピン・プラスチックQFP (14 x 20 mm)	"
$\mu$ PD78F4218AYGC-7EA	100ピン・プラスチックQFP (ファインピッチ) (14 x 14 mm)	"
$\mu$ PD78F4218AYGF-3BA	100ピン・プラスチックQFP (14 x 20 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

備考 x x x はROMコード番号です。

1.9.4 機能概要

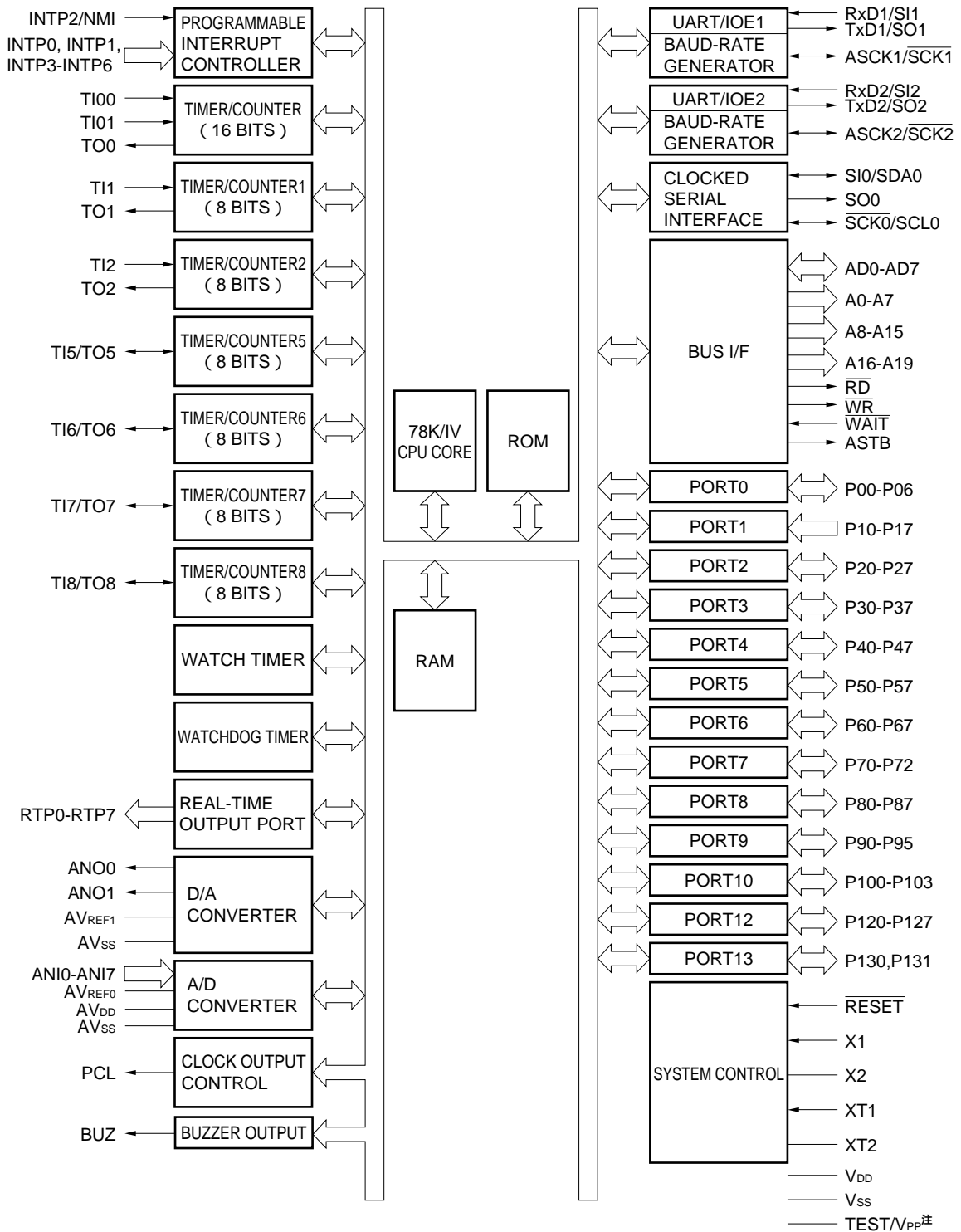
(1/2)

品名		μ PD784217AY	μ PD784218AY	μ PD78F4218AY	
項目					
基本命令（二モニック）数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク，または16ビット×8レジスタ×8バンク（メモリ・マッピング）			
最小命令実行時間		・ 160 ns/320 ns/640 ns/1280 ns/2560 ns（メイン・システム・クロック：12.5 MHz動作時） ・ 61 μs（サブシステム・クロック：32.768 kHz動作時）			
内部メモリ	ROM	192 Kバイト （マスクROM）	256 Kバイト （マスクROM）	256 Kバイト （フラッシュ・メモリ）	
	RAM	12800バイト			
メモリ空間		プログラム/データあわせて1 Mバイト			
I/Oポート	合計	86本			
	CMOS入力	8本			
	CMOS入出力	72本			
	N-chオープン・ドレイン入出力	6本			
	注 付加機能付き端子	プルアップ抵抗付き端子	70本		
		LEDダイレクト・ドライブ出力	22本		
		中耐圧端子	6本		
リアルタイム出力ポート		4ビット×2，または8ビット×1			
タイマ/カウンタ		タイマ/カウンタ：タイマ・レジスタ×1 (16ビット)      キャプチャ/コンペア・レジスタ×2	パルス出力可 ・ PWM/PPG出力 ・ 方形波出力 ・ ワンショット・パルス出力		
		タイマ/カウンタ1：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ/カウンタ2：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ/カウンタ5：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ/カウンタ6：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ/カウンタ7：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ/カウンタ8：タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		

注 付加機能付き端子は，I/O端子の中に含まれています。

品名		μ PD784217AY	μ PD784218AY	μ PD78F4218AY
項目				
シリアル・インタフェース		<ul style="list-style-type: none"> <li>・ UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)</li> <li>・ CSI (3線式シリアルI/O, マルチマスタ対応I<sup>2</sup>Cバス) : 1チャンネル</li> </ul>		
A/Dコンバータ		8ビット分解能 × 8チャンネル		
D/Aコンバータ		8ビット分解能 × 2チャンネル		
クロック出力		f <sub>xx</sub> , f <sub>xx</sub> /2, f <sub>xx</sub> /2 <sup>2</sup> , f <sub>xx</sub> /2 <sup>3</sup> , f <sub>xx</sub> /2 <sup>4</sup> , f <sub>xx</sub> /2 <sup>5</sup> , f <sub>xx</sub> /2 <sup>6</sup> , f <sub>xx</sub> /2 <sup>7</sup> , f <sub>XT</sub> から選択		
プザー出力		f <sub>xx</sub> /2 <sup>10</sup> , f <sub>xx</sub> /2 <sup>11</sup> , f <sub>xx</sub> /2 <sup>12</sup> , f <sub>xx</sub> /2 <sup>13</sup> から選択		
時計用タイマ		1チャンネル		
ウォッチドッグ・タイマ		1チャンネル		
スタンバイ		<ul style="list-style-type: none"> <li>・ HALT/STOP/IDLEモード</li> <li>・ 低消費電力モード (サブシステム・クロックによるCPU動作) 時 : HALT/IDLEモード</li> </ul>		
★ 割り込み	ハードウェア要因	29 (内部 : 20, 外部 : 9)		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスカブル	内部 : 1本, 外部 : 1本		
	マスカブル	内部 : 19本, 外部 : 8本		
		<ul style="list-style-type: none"> <li>・ 4レベルのプログラマブル・プライオリティ</li> <li>・ 3種類の処理形態 : ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング</li> </ul>		
★ 電源電圧		V <sub>DD</sub> = 1.8 ~ 5.5 V		V <sub>DD</sub> = 1.9 ~ 5.5 V
パッケージ		<ul style="list-style-type: none"> <li>・ 100ピン・プラスチックQFP (ファインピッチ) (14 × 14 mm)</li> <li>・ 100ピン・プラスチックQFP (14 × 20 mm)</li> </ul>		

1.9.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4218AYのみ

備考 内部ROM容量は製品によって異なります。

## 1. 10 $\mu$ PD784225サブシリーズ製品概要 ( $\mu$ PD784224, 784225, 78F4225)

### 1. 10. 1 特 徴

$\mu$  PD780058サブシリーズの周辺機能を継承

最小命令実行時間

160 ns (メイン・システム・クロック :  $f_{XX} = 12.5$  MHz動作時)

61  $\mu$ s (サブシステム・クロック :  $f_{XT} = 32.768$  kHz動作時)

制御用途に適する命令セット

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み処理 / マクロ・サービス / コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時 : HALT/IDLEモード (サブシステム・クロック動作時)

内部メモリ : マスクROM 128 Kバイト ( $\mu$  PD784225)

96 Kバイト ( $\mu$  PD784224)

フラッシュ・メモリ 128 Kバイト ( $\mu$  PD78F4225)

RAM 4352バイト ( $\mu$  PD784225, 78F4225)

3584バイト ( $\mu$  PD784224)

I/O端子 : 67本

ソフトウエア・プログラマブル・プルアップ : 50入力

LEDダイレクト・ドライブ可能 : 16出力

タイマ / カウンタ : 16ビット・タイマ / カウンタ  $\times$  1ユニット

8ビット・タイマ / カウンタ  $\times$  4ユニット

時計用タイマ : 1チャンネル

ウォッチドッグ・タイマ : 1チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O) : 1チャンネル

A/Dコンバータ : 8ビット分解能  $\times$  8チャンネル

D/Aコンバータ : 8ビット分解能  $\times$  2チャンネル

リアルタイム出力ポート (タイマ / カウンタと組み合わせることにより, 2系統のステッピング・モータの独立制御が可能)

クロック分周機能

クロック出力機能 :  $f_{XX}$ ,  $f_{XX}/2$ ,  $f_{XX}/2^2$ ,  $f_{XX}/2^3$ ,  $f_{XX}/2^4$ ,  $f_{XX}/2^5$ ,  $f_{XX}/2^6$ ,  $f_{XX}/2^7$ ,  $f_{XT}$ から選択

ブザー出力機能 :  $f_{XX}/2^{10}$ ,  $f_{XX}/2^{11}$ ,  $f_{XX}/2^{12}$ ,  $f_{XX}/2^{13}$ から選択

★ 電源電圧 :  $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$  PD784224, 784225)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$  PD78F4225)

### 1. 10. 2 応用分野

カー・オーディオ, ポータブル・オーディオ, エアコン, 電話機など

## 1. 10. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784224GC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	マスクROM
μ PD784224GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
μ PD784225GC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
μ PD784225GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
μ PD78F4225GC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	フラッシュ・メモリ
μ PD78F4225GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784224GC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	標準 (一般電子機器用)
μ PD784224GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
μ PD784225GC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
μ PD784225GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
μ PD78F4225GC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
μ PD78F4225GK-BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

## 1. 10. 4 機能概要

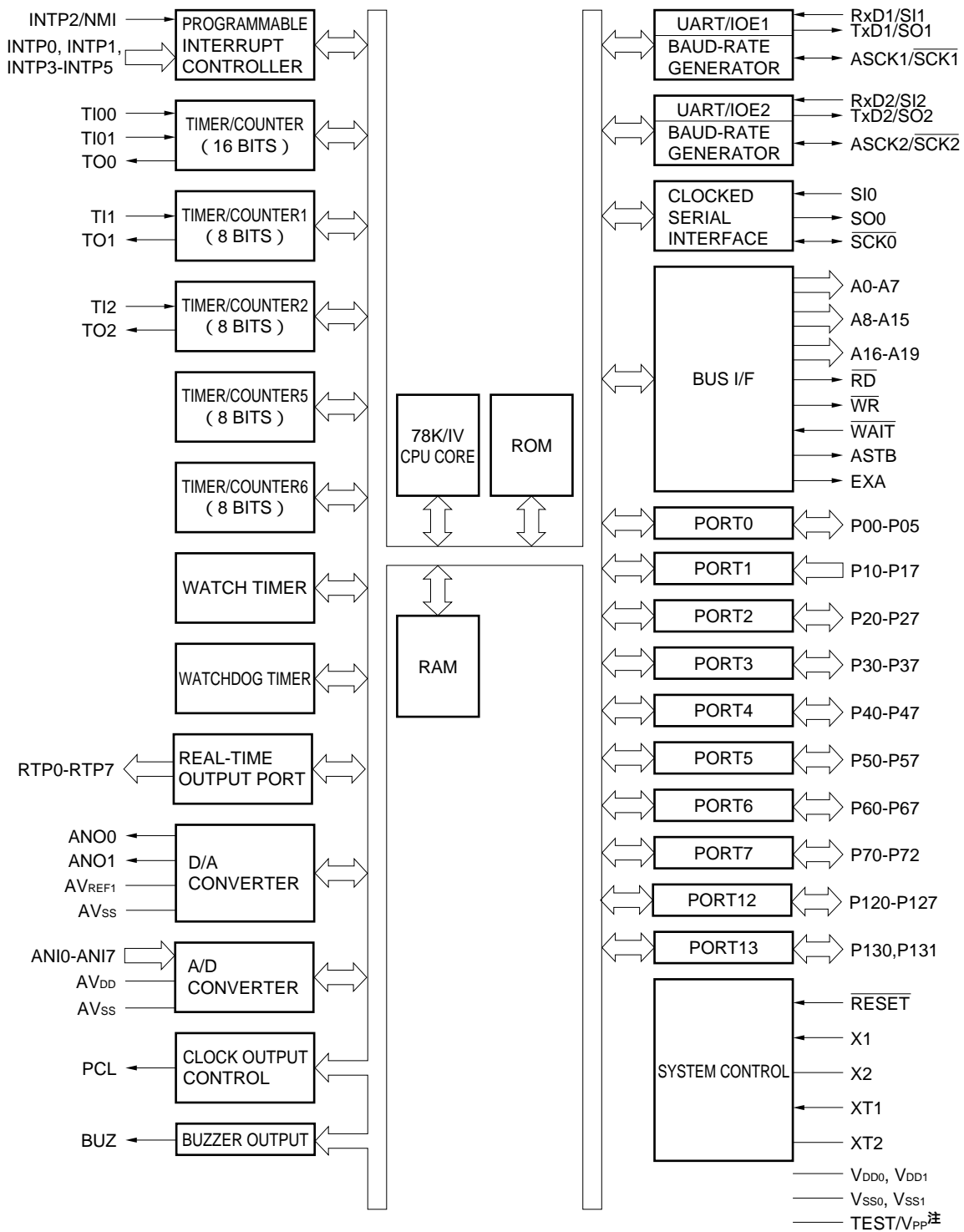
( 1/2 )

品 名		$\mu$ PD784224	$\mu$ PD784225	$\mu$ PD78F4225	
項 目					
基本命令 (ニモニック) 数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク (メモリ・マッピング)			
最小命令実行時間		・ 160 ns/320 ns/640 ns/1280 ns/2560 ns (メイン・システム・クロック : 12.5 MHz動作時) ・ 61 $\mu$ s (サブシステム・クロック : 32.768 kHz動作時)			
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)	
	RAM	3584バイト	4352バイト		
メモリ空間		プログラム / データあわせて1 Mバイト			
I/Oポート	合計	67本			
	CMOS入力	8本			
	CMOS入出力	59本			
	付加機能付き端子	プルアップ抵抗付き端子	57本		
		LEDダイレクト・ドライブ出力	16本		
リアルタイム出力ポート		4ビット×2, または8ビット×1			
タイマ / カウンタ		タイマ / カウンタ : タイマ・レジスタ×1 (16ビット)      キャプチャ / コンペア・レジスタ×2	パルス出力可 ・ PWM/PPG出力 ・ 方形波出力 ・ ワンショット・パルス出力		
		タイマ / カウンタ1 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ / カウンタ2 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・ PWM出力 ・ 方形波出力		
		タイマ / カウンタ5 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1			
		タイマ / カウンタ6 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1			
		タイマ / カウンタ7 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1			
シリアル・インタフェース		・ UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵) ・ CSI (3線式シリアルI/O) : 1チャンネル			
A/Dコンバータ		8ビット分解能×8チャンネル			
D/Aコンバータ		8ビット分解能×2チャンネル			
クロック出力		$f_{xx}$ , $f_{xx}/2$ , $f_{xx}/2^2$ , $f_{xx}/2^3$ , $f_{xx}/2^4$ , $f_{xx}/2^5$ , $f_{xx}/2^6$ , $f_{xx}/2^7$ , $f_{XT}$ から選択			
ブザー出力		$f_{xx}/2^{10}$ , $f_{xx}/2^{11}$ , $f_{xx}/2^{12}$ , $f_{xx}/2^{13}$ から選択			
時計用タイマ		1チャンネル			
ウォッチドッグ・タイマ		1チャンネル			

注 付加機能付き端子は, I/O端子の中に含まれています。

品名		μ PD784224	μ PD784225	μ PD78F4225
項目				
スタンバイ		・ HALT/STOP/IDLEモード ・ 低消費電力モード (サブシステム・クロックによるCPU動作) 時: HALT/IDLEモード		
割り込み	ハードウェア要因	25 (内部: 18, 外部: 7)		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスカブル	内部: 1, 外部: 1		
	マスカブル	内部: 17, 外部: 6		
		・ 4レベルのプログラマブル・プライオリティ ・ 3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング		
★ 電源電圧		$V_{DD} = 1.8 \sim 5.5 V$	$V_{DD} = 1.9 \sim 5.5 V$	
パッケージ		・ 80ピン・プラスチックTQFP (ファインピッチ) (12 × 12 mm) ・ 80ピン・プラスチックQFP (14 × 14 mm)		

1.10.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4225のみ

備考 内部ROM, RAM容量は製品によって異なります。

## 1. 11 $\mu$ PD784225Yサブシリーズ製品概要 ( $\mu$ PD784224Y, 784225Y, 78F4225Y)

### 1. 11. 1 特 徴

$\mu$  PD784225サブシリーズにI<sup>2</sup>Cバス・インタフェースを追加

$\mu$  PD780058Yサブシリーズの周辺機能を継承

最小命令実行時間

160 ns (メイン・システム・クロック :  $f_{XX} = 12.5$  MHz動作時)

61  $\mu$ s (サブシステム・クロック :  $f_{XT} = 32.768$  kHz動作時)

制御用途に適する命令セット

割り込みコントローラ (4レベル・プライオリティ)

ベクタ割り込み処理 / マクロ・サービス / コンテキスト・スイッチング

スタンバイ機能

HALT/STOP/IDLEモード

低消費電力モード時 : HALT/IDLEモード (サブシステム・クロック動作時)

内部メモリ : マスクROM 128 Kバイト ( $\mu$  PD784225Y)

96 Kバイト ( $\mu$  PD784224Y)

フラッシュ・メモリ 128 Kバイト ( $\mu$  PD78F4225Y)

RAM 4352バイト ( $\mu$  PD784225Y, 78F4225Y)

3584バイト ( $\mu$  PD784224Y)

I/O端子 : 67本

ソフトウエア・プログラマブル・プルアップ : 50入力

LEDダイレクト・ドライブ可能 : 16出力

タイマ / カウンタ : 16ビット・タイマ / カウンタ  $\times$  1ユニット

8ビット・タイマ / カウンタ  $\times$  4ユニット

時計用タイマ : 1チャンネル

ウォッチドッグ・タイマ : 1チャンネル

シリアル・インタフェース

UART/IOE (3線式シリアルI/O) : 2チャンネル (ポー・レート・ジェネレータ内蔵)

CSI (3線式シリアルI/O, マルチマスタ対応I<sup>2</sup>Cバス) : 1チャンネル

A/Dコンバータ : 8ビット分解能  $\times$  8チャンネル

D/Aコンバータ : 8ビット分解能  $\times$  2チャンネル

リアルタイム出力ポート (タイマ / カウンタと組み合わせることにより, 2系統のステッピング・モータの独立制御が可能)

クロック分周機能

クロック出力機能 :  $f_{XX}, f_{XX}/2, f_{XX}/2^2, f_{XX}/2^3, f_{XX}/2^4, f_{XX}/2^5, f_{XX}/2^6, f_{XX}/2^7, f_{XT}$  から選択

ブザー出力機能 :  $f_{XX}/2^{10}, f_{XX}/2^{11}, f_{XX}/2^{12}, f_{XX}/2^{13}$  から選択

★ 電源電圧 :  $V_{DD} = 1.8 \sim 5.5$  V ( $\mu$  PD784224Y, 784225Y)

$V_{DD} = 1.9 \sim 5.5$  V ( $\mu$  PD78F4225Y)

### 1. 11. 2 応用分野

カー・オーディオ, ポータブル・オーディオ, エアコン, 電話機など

## 1. 11. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784224YGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	マスクROM
$\mu$ PD784224Y GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
$\mu$ PD784225YGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
$\mu$ PD784225Y GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
$\mu$ PD78F4225YGC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	フラッシュ・メモリ
$\mu$ PD78F4225Y GK- BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784224YGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	標準 (一般電子機器用)
$\mu$ PD784224Y GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
$\mu$ PD784225YGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
$\mu$ PD784225Y GK- x x x -BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"
$\mu$ PD78F4225YGC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
$\mu$ PD78F4225Y GK- BE9	80ピン・プラスチックTQFP (ファインピッチ) (12 x 12 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

**注意**  $\mu$  PD784225Yサブシリーズは開発中です。

## 1.11.4 機能概要

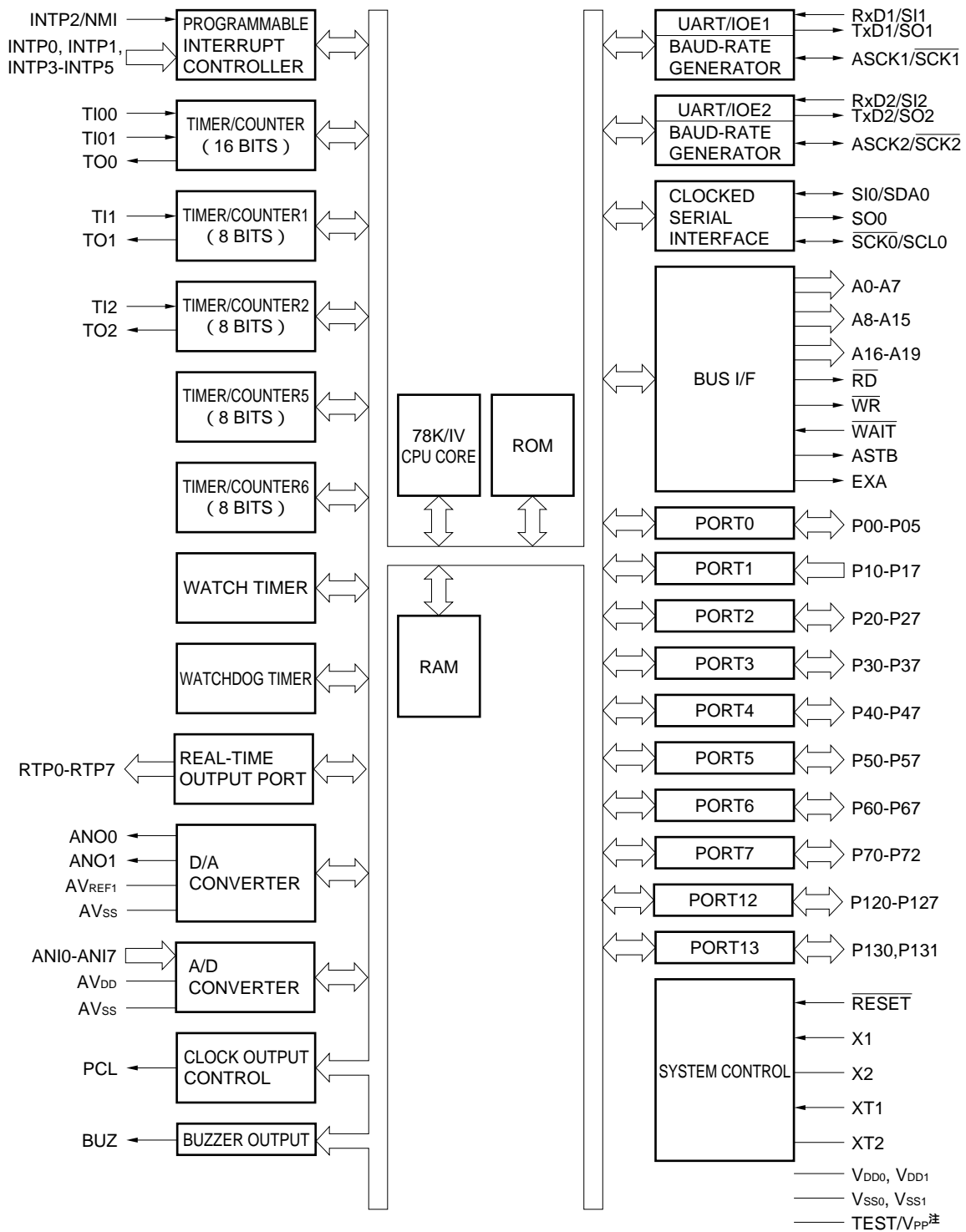
(1/2)

品名		$\mu$ PD784224Y	$\mu$ PD784225Y	$\mu$ PD78F4225Y
項目				
基本命令(ニモニック)数		113		
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)		
最小命令実行時間		・160 ns/320 ns/640 ns/1280 ns/2560 ns(メイン・システム・クロック: 12.5 MHz動作時) ・61 $\mu$ s(サブシステム・クロック: 32.768 kHz動作時)		
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)
	RAM	3584バイト	4352バイト	
メモリ空間		プログラム/データあわせて1 Mバイト		
I/Oポート	合計	67本		
	CMOS入力	8本		
	CMOS入出力	59本		
	付加機能付き端子			
	プルアップ抵抗付き端子	57本		
	LEDダイレクト・ドライブ出力	16本		
リアルタイム出力ポート		4ビット×2, または8ビット×1		
タイマ/カウンタ		タイマ/カウンタ : タイマ・レジスタ×1 (16ビット)      キャプチャ/コンペア・レジスタ×2	パルス出力可 ・PWM/PPG出力 ・方形波出力 ・ワンショット・パルス出力	
		タイマ/カウンタ1 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力	
		タイマ/カウンタ2 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1	パルス出力可 ・PWM出力 ・方形波出力	
		タイマ/カウンタ5 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1		
		タイマ/カウンタ6 : タイマ・レジスタ×1 (8ビット)      コンペア・レジスタ×1		
シリアル・インタフェース		・UART/IOE(3線式シリアルI/O): 2チャンネル(ポーレート・ジェネレータ内蔵) ・CSI(3線式シリアルI/O, マルチマスタ対応I <sup>2</sup> Cバス): 1チャンネル		
A/Dコンバータ		8ビット分解能×8チャンネル		
D/Aコンバータ		8ビット分解能×2チャンネル		
クロック出力		$f_{xx}$ , $f_{xx}/2$ , $f_{xx}/2^2$ , $f_{xx}/2^3$ , $f_{xx}/2^4$ , $f_{xx}/2^5$ , $f_{xx}/2^6$ , $f_{xx}/2^7$ , $f_{XT}$ から選択		
ブザー出力		$f_{xx}/2^{10}$ , $f_{xx}/2^{11}$ , $f_{xx}/2^{12}$ , $f_{xx}/2^{13}$ から選択		
時計用タイマ		1チャンネル		
ウォッチドッグ・タイマ		1チャンネル		

注 付加機能付き端子は, I/O端子の中に含まれています。

品名		μ PD784224Y	μ PD784225Y	μ PD78F4225Y
項目				
スタンバイ		<ul style="list-style-type: none"> <li>・ HALT/STOP/IDLEモード</li> <li>・ 低消費電力モード (サブシステム・クロックによるCPU動作) 時: HALT/IDLEモード</li> </ul>		
割り込み	ハードウェア要因	25 (内部: 18, 外部: 7)		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスカブル	内部: 1, 外部: 1		
	マスカブル	内部: 17, 外部: 6		
		<ul style="list-style-type: none"> <li>・ 4レベルのプログラマブル・プライオリティ</li> <li>・ 3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング</li> </ul>		
★ 電源電圧		V <sub>DD</sub> = 1.8 ~ 5.5 V		V <sub>DD</sub> = 1.9 ~ 5.5 V
パッケージ		<ul style="list-style-type: none"> <li>・ 80ピン・プラスチックTQFP (ファインピッチ) (12 × 12 mm)</li> <li>・ 80ピン・プラスチックQFP (14 × 14 mm)</li> </ul>		

1.11.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4225Yのみ

備考 内部ROM, RAM容量は製品によって異なります。

## 1. 12 $\mu$ PD784908サブシリーズ製品概要 ( $\mu$ PD784907, 784908, 78P4908)

### 1. 12. 1 特 徴

最小命令実行時間：160 ns (12.58 MHz動作時)

内部メモリ

ROM

マスクROM：96 Kバイト ( $\mu$  PD784907)

128 Kバイト ( $\mu$  PD784908)

PROM：128 Kバイト ( $\mu$  PD78P4908)

RAM：3584バイト ( $\mu$  PD784907)

4352バイト ( $\mu$  PD784908, 78P4908)

I/Oポート：80本

タイマ/カウンタ：16ビット・タイマ/カウンタ×3ユニット

16ビット・タイマ×1ユニット

時計用タイマ：1チャンネル

ウォッチドッグ・タイマ：1チャンネル

シリアル・インタフェース：4チャンネル

UART/IOE (3線式シリアルI/O)：2チャンネル

CSI (3線式シリアルI/O)：2チャンネル

スタンバイ機能

HALT/STOP/IDLEモード

クロック分周機能

クロック出力機能

fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/16から選択

A/Dコンバータ：8ビット分解能×8チャンネル

IEBusコントローラ内蔵

低消費電力

電源電圧： $V_{DD} = 3.5 \sim 5.5$  V (マスクROM品)

$V_{DD} = 4.0 \sim 5.5$  V (PROM品)

### 1. 12. 2 応用分野

カー・オーディオなど

## 1. 12. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784907GF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
μ PD784908GF- x x x -3BA	"	"
μ PD78P4908GF-3BA	"	ワン・タイムPROM

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784907GF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
μ PD784908GF- x x x -3BA	"	"
μ PD78P4908GF-3BA	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

1.12.4 機能概要

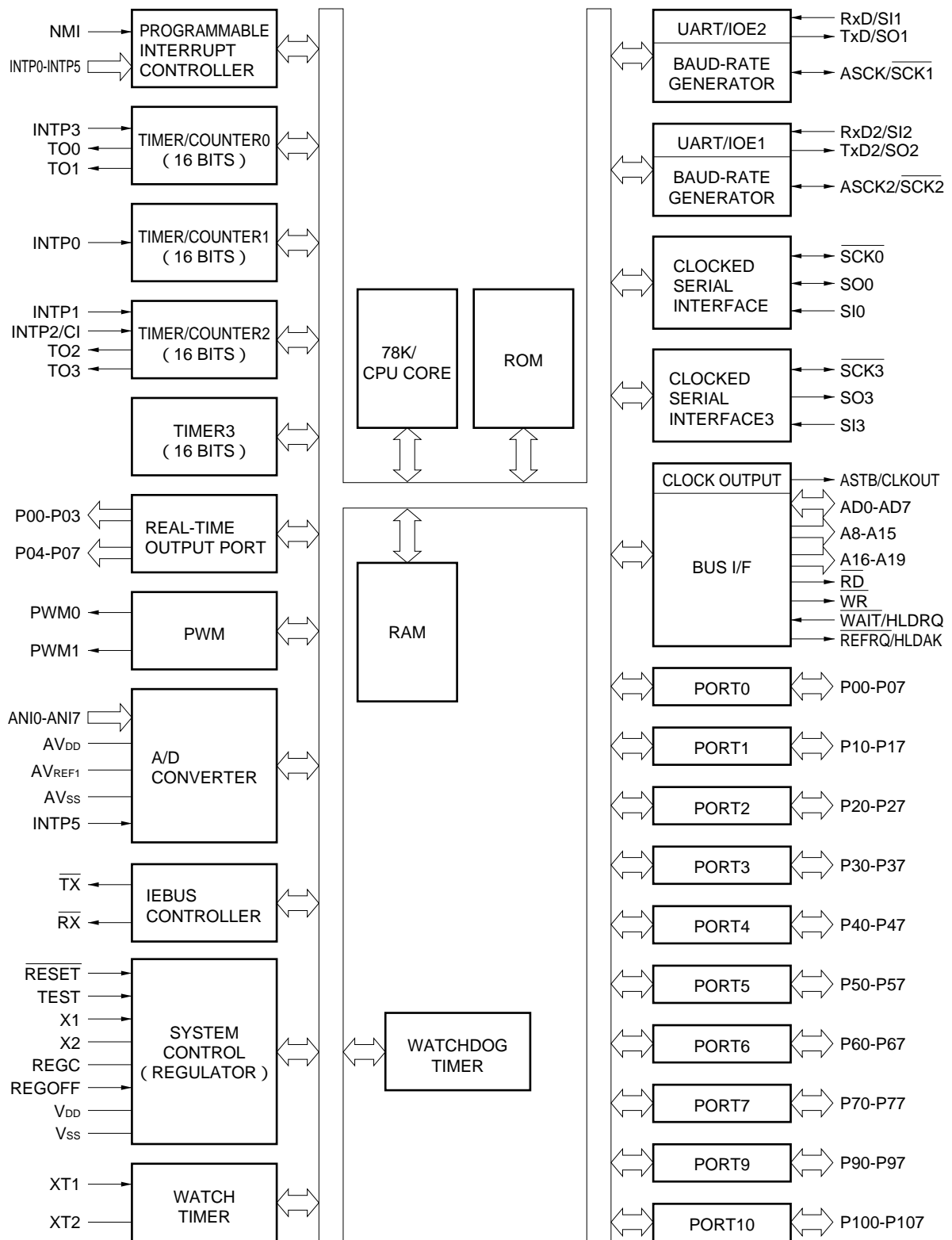
(1/2)

品名		μ PD784907	μ PD784908	μ PD78P4908	
項目					
基本命令(二モニック)数		113			
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク(メモリ・マッピング)			
最小命令実行時間		160 ns/320 ns/636 ns/1.27 μs (12.58 MHz動作時)			
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (PROM)	
	RAM	3584バイト	4352バイト		
メモリ空間		プログラム/データあわせて1 Mバイト			
I/Oポート	合計	80本			
	入力	8本			
	入出力	72本			
	注 付加機能付き端子	LEDダイレクト・ドライブ出力	24本		
		トランジスタ・ダイレクト・ドライブ	8本		
		N-chオープン・ドレイン	4本		
リアルタイム出力ポート		4ビット×2, または8ビット×1			
IEBusコントローラ		内蔵(簡易版)			
タイマ/カウンタ	タイマ/カウンタ0 : タイマ・レジスタ×1 (16ビット)	キャプチャ・レジスタ×1 コンペア・レジスタ×2	パルス出力可 ・トグル出力 ・PWM/PPG出力 ・ワンショット・パルス出力		
	タイマ/カウンタ1 : タイマ・レジスタ×1 (16ビット)	キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1	リアルタイム出力ポート		
	タイマ/カウンタ2 : タイマ・レジスタ×1	キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1	パルス出力可 ・トグル出力 ・PWM/PPG出力		
	タイマ3	タイマ・レジスタ×1 コンペア・レジスタ×1			
時計用タイマ		0.5秒間隔で割り込み要求発生(時計クロック発振器を内蔵) 入カクロックは, メイン・クロック(12.58 MHz)と時計クロック(32.7 kHz)の選択可能			
クロック出力		f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択(1ビット出力ポートとしても使用可能)			
PWM出力		12ビット分解能×2チャンネル			
シリアル・インタフェース		UART/IOE(3線式シリアルI/O) : 2チャンネル(ポーレート・ジェネレータ内蔵) CSI(3線式シリアルI/O) : 2チャンネル			

注 付加機能付き端子は, I/O端子の中に含まれています。

品名		$\mu$ PD784907	$\mu$ PD784908	$\mu$ PD78P4908
項目				
A/Dコンバータ		8ビット分解能×8チャンネル		
ウォッチドッグ・タイマ		1チャンネル		
スタンバイ		HALT/STOP/IDLEモード		
割り込み	ハードウェア要因	27 (内部: 20, 外部: 7 (サンプリング・クロック可変入力: 1))		
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスクابل	内部: 1, 外部: 1		
	マスクابل	内部: 19, 外部: 6		
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング		
電源電圧		$V_{DD} = 3.5 \sim 5.5 V$		$V_{DD} = 4.0 \sim 5.5 V$
パッケージ		100ピン・プラスチックQFP (14×20 mm)		

### 1. 12. 5 ブロック図



**備考** 内部ROM, RAM容量は製品によって異なります。

## 1. 13 $\mu$ PD784915サブシリーズ製品概要 ( $\mu$ PD784915B, 784916B, 78P4916)

### 1. 13. 1 特 徴

78K/ シリーズ (16ビットCPUコア採用) : 最小命令実行時間250 ns (内部クロック : 8 MHz動作時)

VTRサーボ制御用タイマ・ユニット (スーパー・タイマ・ユニット) 内蔵

VHS規格VTR用アナログ回路内蔵

CTLアンプ

RECCTLドライバ (再書き込み対応)

DPFG分離回路 (3値分離回路)

DFGアンプ, DPGコンパレータ, CFGコンパレータ

リールFGコンパレータ (2チャンネル), CSYNCコンパレータ

I/Oポート : 54本

シリアル・インタフェース : 2チャンネル (3線式シリアルI/O)

A/Dコンバータ : 12チャンネル (変換時間10  $\mu$ s)

PWM出力 : 16ビット分解能  $\times$  3チャンネル, 8ビット分解能  $\times$  3チャンネル

割り込み機能

ベクタ割り込み機能

マクロ・サービス機能

コンテキスト・スイッチング機能

低周波発振モード対応 : メイン・システム・クロック周波数 = 内部クロック周波数

低消費電力モード対応 : サブシステム・クロックによるCPU動作可能

ハードウェア時計機能内蔵 : 低電圧 ( $V_{DD} = 2.7$  V (MIN.)), 低消費電流の時計動作が可能

高密度実装対応パッケージ : 100ピン・プラスチックQFP (0.65 mmピッチ, 14  $\times$  20 mm)

### 1. 13. 2 応用分野

VTRのシステム / サーボ / タイマ制御用 (据え置き型, カメラ一体型)

## 1. 13. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784915BGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
μ PD784916BGF- x x x -3BA	"	"
μ PD78P4916GF-3BA	"	ワン・タイムPROM

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784915BGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
μ PD784916BGF- x x x -3BA	"	"
μ PD78P4916GF-3BA	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

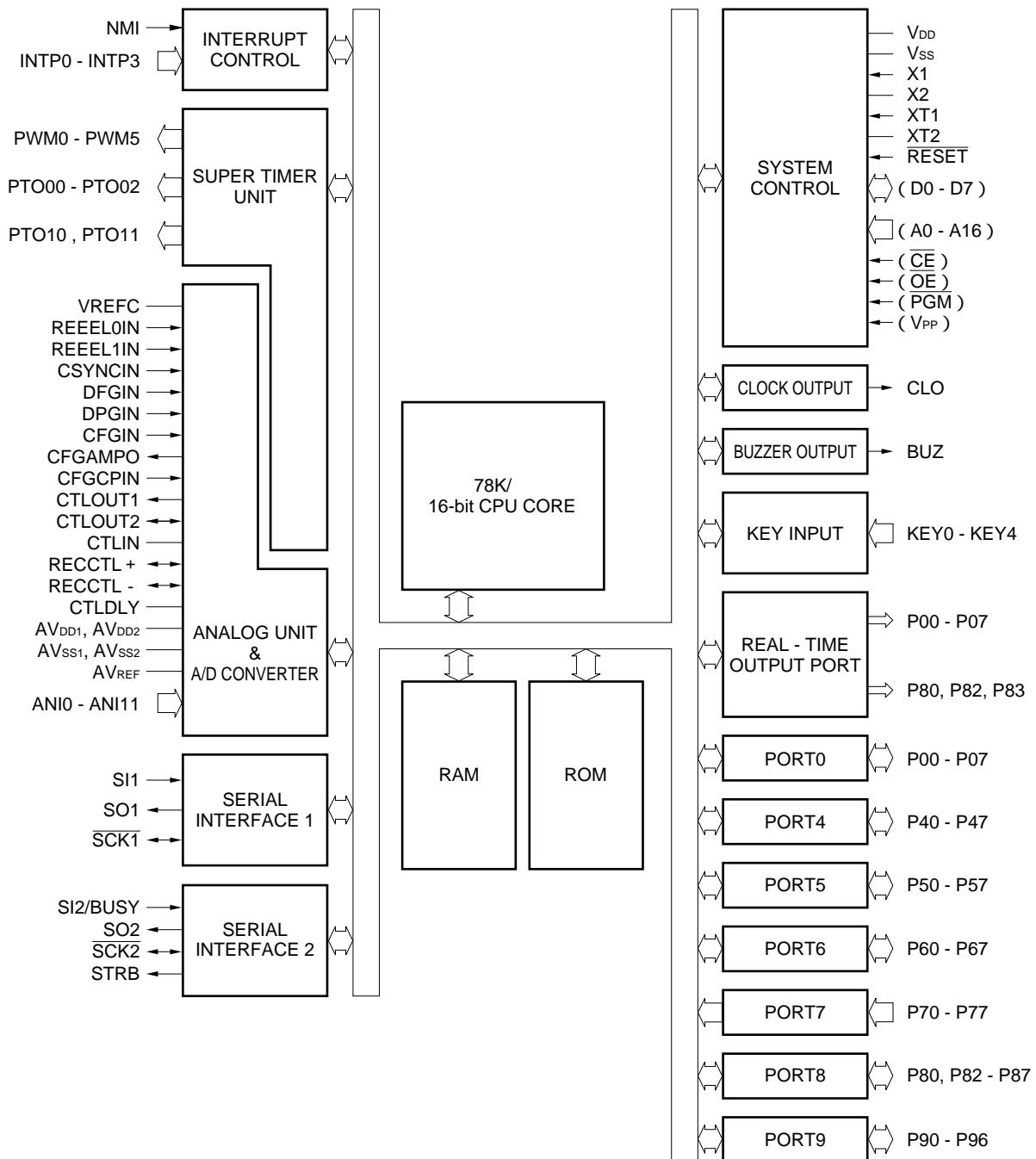
**備考** x x x はROMコード番号です。

## 1.13.4 機能概要

★

品名		$\mu$ PD784915B	$\mu$ PD784916B	$\mu$ PD78P4916																																
項目																																				
基本命令(二モニック)数		113																																		
最小命令実行時間		250 ns (内部クロック: 8 MHz動作時)																																		
内部メモリ	ROM	48 Kバイト (マスクROM)	62 Kバイト (マスクROM)	62 Kバイト (ワン・タイムPROM)																																
	RAM	1280バイト		2048バイト																																
割り込み		4レベル(プログラマブル), ベクタ割り込み, マクロ・サービス, コンテキスト・スイッチング																																		
外部要因		9本(NMI含む)																																		
内部要因		19本																																		
マクロ・サービス使用可能割り込み数		25本																																		
マクロ・サービスの種類		4タイプ10種類																																		
I/Oポート		入力: 8本, 入出力: 46本																																		
タイム・ベース・カウンタ		・ 22ビットFRC ・ 分解能: 125 ns, 最大カウント時間: 524 ms																																		
キャプチャ・レジスタ		<table border="1"> <thead> <tr> <th>入力信号</th> <th>ビット数</th> <th>計測周期</th> <th>動作エッジ</th> </tr> </thead> <tbody> <tr> <td>CFG</td> <td>22</td> <td>125 ns ~ 524 ms</td> <td></td> </tr> <tr> <td>DFG</td> <td>22</td> <td>125 ns ~ 524 ms</td> <td></td> </tr> <tr> <td>HSW</td> <td>16</td> <td>1 <math>\mu</math>s ~ 65.5 ms</td> <td></td> </tr> <tr> <td>V<sub>SYNC</sub></td> <td>22</td> <td>125 ns ~ 524 ms</td> <td></td> </tr> <tr> <td>CTL</td> <td>16</td> <td>1 <math>\mu</math>s ~ 65.5 ms</td> <td></td> </tr> <tr> <td>TREEL</td> <td>22</td> <td>125 ns ~ 524 ms</td> <td></td> </tr> <tr> <td>SREEL</td> <td>22</td> <td>125 ns ~ 524 ms</td> <td></td> </tr> </tbody> </table>			入力信号	ビット数	計測周期	動作エッジ	CFG	22	125 ns ~ 524 ms		DFG	22	125 ns ~ 524 ms		HSW	16	1 $\mu$ s ~ 65.5 ms		V <sub>SYNC</sub>	22	125 ns ~ 524 ms		CTL	16	1 $\mu$ s ~ 65.5 ms		TREEL	22	125 ns ~ 524 ms		SREEL	22	125 ns ~ 524 ms	
入力信号	ビット数	計測周期	動作エッジ																																	
CFG	22	125 ns ~ 524 ms																																		
DFG	22	125 ns ~ 524 ms																																		
HSW	16	1 $\mu$ s ~ 65.5 ms																																		
V <sub>SYNC</sub>	22	125 ns ~ 524 ms																																		
CTL	16	1 $\mu$ s ~ 65.5 ms																																		
TREEL	22	125 ns ~ 524 ms																																		
SREEL	22	125 ns ~ 524 ms																																		
汎用タイマ		16ビット・タイマ×3																																		
PBCTLデューティ判別		・ 再生コントロール信号のデューティ ・ VISS検出, ワイド・アスペクト検出																																		
リニア・タイム・カウンタ		5ビットUDCによるCTL信号のカウント																																		
リアルタイム出力ポート		11本																																		
シリアル・インタフェース		クロック周期式(3線式): 2チャンネル																																		
A/Dコンバータ		8ビット分解能×12チャンネル, 変換時間: 10 $\mu$ s																																		
PWM出力		・ 16ビット分解能×3チャンネル, 8ビット分解能×3チャンネル ・ キャリア周波数: 62.5 kHz																																		
時計機能		0.5秒計測, 低電圧動作可能																																		
スタンバイ機能		HALTモード/STOPモード																																		
アナログ回路		・ CTLアンプ ・ RECCTLドライバ(再書き込み対応) ・ DPGF分離回路(3値分離回路) ・ DFGアンプ, DPGコンパレータ, CFGアンプ ・ リールFGコンパレータ ・ CSYNCコンパレータ																																		
電源電圧		V <sub>DD</sub> = 2.7 ~ 5.5 V																																		
パッケージ		100ピン・プラスチックQFP(14×20 mm)																																		

### 1. 13. 5 ブロック図



備考1. 内部ROM, RAM容量は、製品により異なります。

2.  $V_{PP}$ は $\mu$  PD78P4916のみ

3. ( )内はPROMプログラミング・モード時

## 1. 14 $\mu$ PD784928サブシリーズ製品概要 ( $\mu$ PD784927, 784928, 78F4928)

### 1. 14. 1 特 徴

16ビットCPUコア採用：最小命令実行時間250 ns（内部クロック：8 MHz動作時）

VTRサーボ制御用タイマ・ユニット（スーパー・タイマ・ユニット）内蔵

I/Oポート：74本

VHS規格VTR用アナログ回路内蔵

CTLアンプ

DPGアンプ

RECCTLドライバ（再書き込み対応）

DPFG分離回路（3値分離回路）

CFGアンプ

リールFGコンパレータ（2チャンネル）

DFGアンプ

CSYNCコンパレータ

シリアル・インタフェース：2チャンネル

3線式シリアルI/O：2チャンネル

A/Dコンバータ：12チャンネル（変換時間：10  $\mu$ s）

PWM出力：16ビット分解能×3チャンネル，8ビット分解能×3チャンネル

割り込み機能

ベクタ割り込み機能

マクロ・サービス機能

コンテキスト・スイッチング機能

低周波発振モード対応：メイン・システム・クロック周波数 = 内部クロック周波数

低消費電力モード対応：サブシステム・クロックによるCPU動作可能

電源電圧： $V_{DD} = 2.7 \sim 5.5$  V

ハードウェア時計機能内蔵：低電圧（ $V_{DD} = 2.7$  V（MIN.）），低消費電流の時計動作が可能

### 1. 14. 2 応用分野

据え置き型VTR，カメラ一体型VTRなど

## 1. 14. 3 オーダ情報と品質水準

## (1) オーダ情報

	オーダ名称	パッケージ	内部ROM
	$\mu$ PD784927GF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
★	$\mu$ PD784928GF- x x x -3BA	"	"
	$\mu$ PD78F4928GF-3BA	"	フラッシュ・メモリ

**備考** x x x はROMコード番号です。

## (2) 品質水準

	オーダ名称	パッケージ	品質水準
	$\mu$ PD784927GF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
★	$\mu$ PD784928GF- x x x -3BA	"	"
	$\mu$ PD78F4928GF-3BA	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

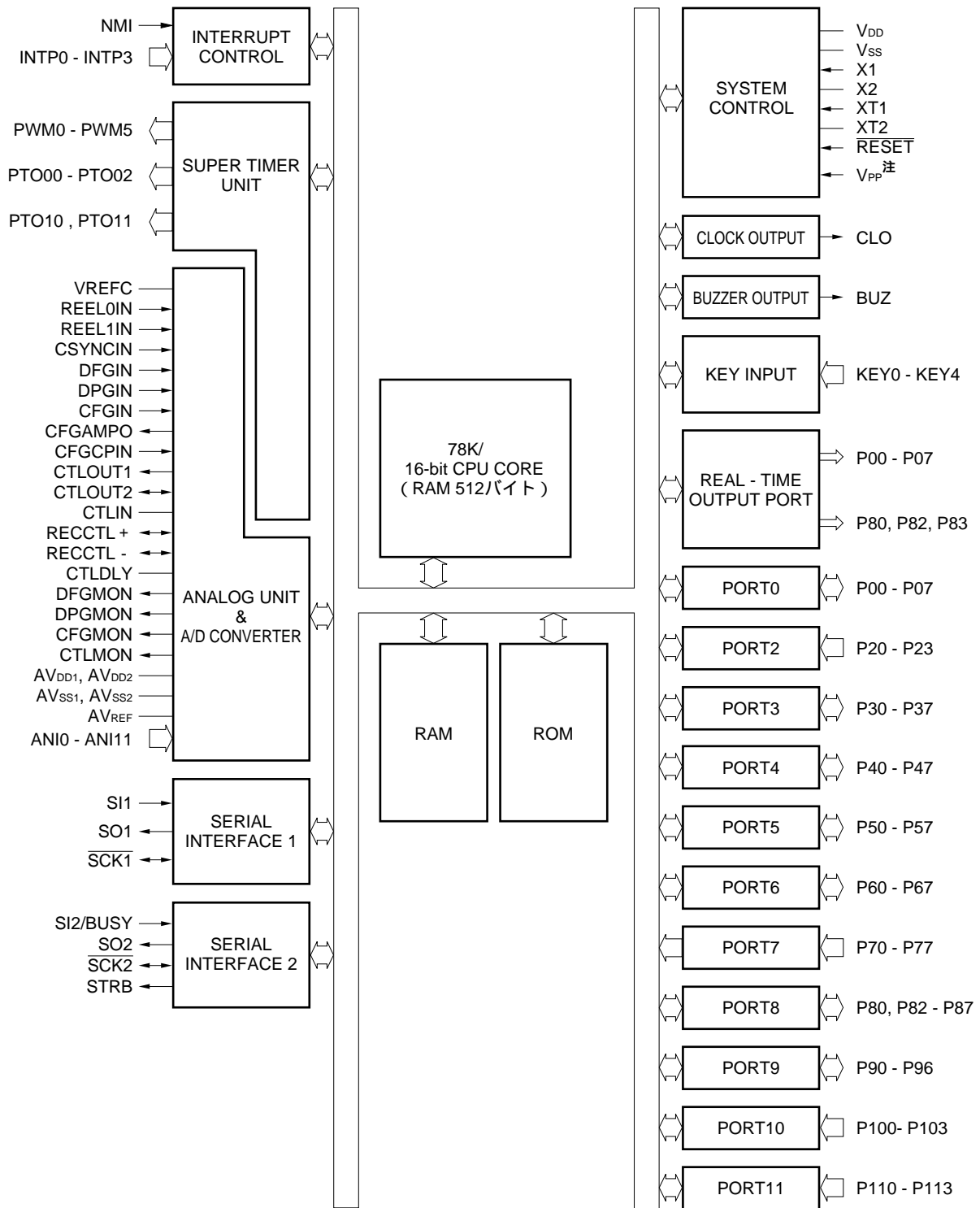
**備考** x x x はROMコード番号です。

## 1.14.4 機能概要

★

品名		$\mu$ PD784927	$\mu$ PD784928	$\mu$ PD78F4928
項目				
基本命令(二モニック)数		113		
最小命令実行時間		250 ns (内部クロック: 8 MHz動作時)		
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)
	RAM	2048バイト	3584バイト	3584バイト
割り込み要因	外部	9本 (NMI含む)		
	内部	22本 (ソフトウェア割り込み含む)		
		<ul style="list-style-type: none"> <li>・4レベルのプログラマブル・プライオリティ</li> <li>・3種類の処理形態</li> </ul> ベクタ割り込み, マクロ・サービス, コンテキスト・スイッチング		
I/Oポート	入力	20本		
	入出力	54本 (うちLEDダイレクト・ドライブ用ポート: 8本)		
タイム・ベース・カウンタ		<ul style="list-style-type: none"> <li>・22ビットFRC</li> <li>・分解能: 125 ns, 最大カウント時間: 524 ms</li> </ul>		
キャプチャ・レジスタ		入力信号	ビット数	計測周期
		CFG	22	125 ns ~ 524 ms
		DFG	22	125 ns ~ 524 ms
		HSW	16	1 $\mu$ s ~ 65.5 ms
		VSYNC	22	125 ns ~ 524 ms
		CTL	16	1 $\mu$ s ~ 65.5 ms
		TREEL	22	125 ns ~ 524 ms
		SREEL	22	125 ns ~ 524 ms
汎用タイマ		16ビット・タイマ×3		
PBCTLデューティ判別		<ul style="list-style-type: none"> <li>・再生コントロール信号のデューティ判別</li> <li>・VISS検出, ワイド・アスペクト検出</li> </ul>		
リニア・タイム・カウンタ		5ビットUDCによるCTL信号のカウント		
リアルタイム出力ポート		11本		
シリアル・インタフェース		3線式シリアルI/O: 2チャンネル (うちBUSY/STRB制御可能: 1チャンネル)		
ブザー出力機能		1.95 kHz, 3.91 kHz, 7.81 kHz, 15.6 kHz (内部クロック: 8 MHz動作時) 2.048 kHz, 4.096 kHz, 32.768 kHz (サブシステム・クロック: 32.768 kHz動作時)		
A/Dコンバータ		8ビット分解能×12チャンネル, 変換時間: 10 $\mu$ s		
PWM出力		<ul style="list-style-type: none"> <li>・16ビット分解能×3チャンネル, 8ビット分解能×3チャンネル</li> <li>・キャリア周波数: 62.5 kHz</li> </ul>		
時計機能		0.5秒計測, 低電圧動作 ( $V_{DD} = 2.7$ V) 可能		
スタンバイ機能		HALTモード / STOPモード / 低消費電力モード / 低消費電力HALTモード		
アナログ回路		<ul style="list-style-type: none"> <li>・CTLアンプ</li> <li>・RECCTLドライバ (再書き込み対応)</li> <li>・CFGアンプ</li> <li>・DFGアンプ</li> <li>・DPGアンプ</li> <li>・DPFG分離回路 (3値分離回路)</li> <li>・リールFGコンパレータ</li> <li>・CSYNCコンパレータ</li> </ul>		
電源電圧		$V_{DD} = +2.7 \sim 5.5$ V		
パッケージ		100ピン・プラスチックQFP (14×20 mm)		

### 1.14.5 ブロック図



注 V<sub>PP</sub>端子はμ PD78F4928のみ

備考 内部ROM, RAM容量は、製品によって異なります。

## 1. 15 $\mu$ PD784928Yサブシリーズ製品概要 ( $\mu$ PD784927Y, 784928Y, 78F4928Y)

### 1. 15. 1 特 徴

$\mu$  PD784928サブシリーズに $I^2C$ バス・インタフェースを追加

16ビットCPUコア採用：最小命令実行時間250 ns (内部クロック：8 MHz動作時)

VTRサーボ制御用タイマ・ユニット (スーパー・タイマ・ユニット) 内蔵

I/Oポート：74本

VHS規格VTR用アナログ回路内蔵

CTLアンプ

DPGアンプ

RECCTLドライバ (再書き込み対応)

DPFG分離回路 (3値分離回路)

CFGアンプ

リールFGコンパレータ (2チャンネル)

DFGアンプ

CSYNCコンパレータ

シリアル・インタフェース：2チャンネル

3線式シリアルI/O：2チャンネル

$I^2C$ バス・インタフェース：1チャンネル

A/Dコンバータ：12チャンネル (変換時間：10  $\mu$ s)

PWM出力：16ビット分解能 $\times$ 3チャンネル, 8ビット分解能 $\times$ 3チャンネル

割り込み機能

ベクタ割り込み機能

マクロ・サービス機能

コンテキスト・スイッチング機能

低周波発振モード対応：メイン・システム・クロック周波数 = 内部クロック周波数

低消費電力モード対応：サブシステム・クロックによるCPU動作可能

電源電圧： $V_{DD} = 2.7 \sim 5.5$  V

ハードウェア時計機能内蔵：低電圧 ( $V_{DD} = 2.7$  V (MIN.)) , 低消費電流の時計動作が可能

### 1. 15. 2 応用分野

据え置き型VTR, カメラ一体型VTRなど

## 1. 15. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784927YGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
★ $\mu$ PD784928YGF- x x x -3BA	"	"
$\mu$ PD78F4928YGF-3BA	"	フラッシュ・メモリ

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784927YGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
★ $\mu$ PD784928YGF- x x x -3BA	"	"
$\mu$ PD78F4928YGF-3BA	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

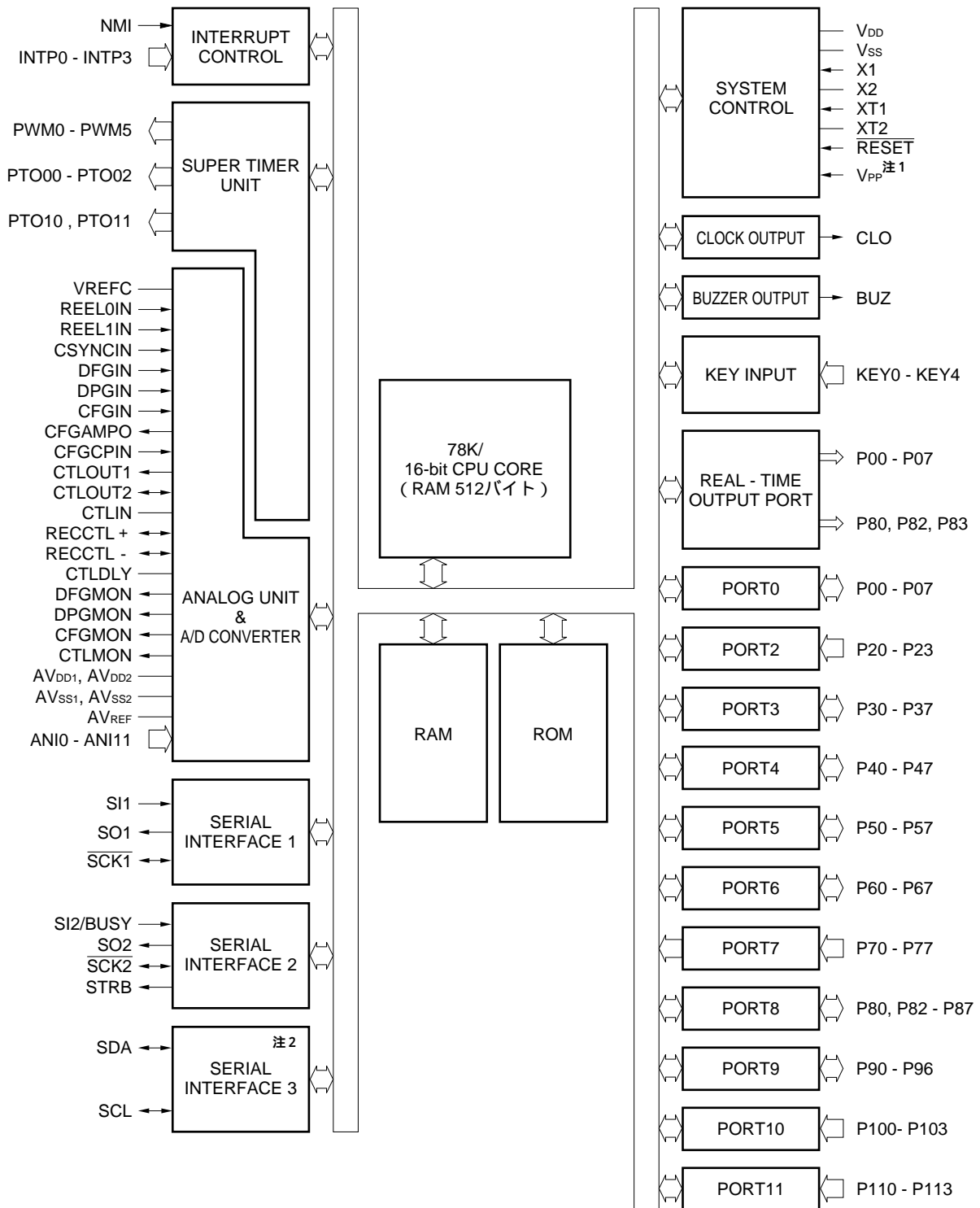
**備考** x x x はROMコード番号です。

1.15.4 機能概要

★

品名		μ PD784927Y	μ PD784928Y	μ PD78F4928Y
項目				
基本命令（二モニック）数		113		
最小命令実行時間		250 ns（内部クロック：8 MHz動作時）		
内部メモリ	ROM	96 Kバイト （マスクROM）	128 Kバイト （マスクROM）	128 Kバイト （フラッシュ・メモリ）
	RAM	2048バイト	3584バイト	
割り込み要因	外部	9本（NMI含む）		
	内部	23本（ソフトウェア割り込み含む）		
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態 ベクタ割り込み，マクロ・サービス，コンテキスト・スイッチング		
I/Oポート	入力	20本		
	入出力	54本（うちLEDダイレクト・ドライブ用ポート：8本）		
タイム・ベース・カウンタ		・22ビットFRC ・分解能：125 ns，最大カウント時間：524 ms		
キャプチャ・レジスタ		入力信号	ビット数	計測周期
		CFG	22	125 ns ~ 524 ms
		DFG	22	125 ns ~ 524 ms
		HSW	16	1 μs ~ 65.5 ms
		VSYNC	22	125 ns ~ 524 ms
		CTL	16	1 μs ~ 65.5 ms
		TREEL	22	125 ns ~ 524 ms
		SREEL	22	125 ns ~ 524 ms
汎用タイマ		16ビット・タイマ×3		
PBCTLデューティ判別		・再生コントロール信号のデューティ判別 ・VISS検出，ワイド・アスペクト検出		
リニア・タイム・カウンタ		5ビットUDCによるCTL信号のカウント		
リアルタイム出力ポート		11本		
シリアル・インタフェース		・3線式シリアルI/O：2チャンネル（うちBUSY/STRB制御可能：1チャンネル） ・I <sup>2</sup> Cバス・インタフェース（マルチマスタ対応）：1チャンネル		
ブザー出力機能		1.95 kHz, 3.91 kHz, 7.81 kHz, 15.6 kHz（内部クロック：8 MHz動作時） 2.048 kHz, 4.096 kHz, 32.768 kHz（サブシステム・クロック：32.768 kHz動作時）		
A/Dコンバータ		8ビット分解能×12チャンネル，変換時間：10 μs		
PWM出力		・16ビット分解能×3チャンネル，8ビット分解能×3チャンネル ・キャリア周波数：62.5 kHz		
時計機能		0.5秒計測，低電圧動作（V <sub>DD</sub> = 2.7 V）可能		
スタンバイ機能		HALTモード / STOPモード / 低消費電力モード / 低消費電力HALTモード		
アナログ回路		・CTLアンプ ・RECCTLドライバ（再書き込み対応） ・CFGアンプ ・DFGアンプ ・DPGアンプ ・DPFG分離回路（3値分離回路） ・リールFGコンパレータ ・CSYNCコンパレータ		
電源電圧		V <sub>DD</sub> = +2.7 ~ 5.5 V		
パッケージ		100ピン・プラスチックQFP（14×20 mm）		

### 1. 15. 5 ブロック図



注1. VPP端子はμ PD78F4928Yのみ

2. I<sup>2</sup>Cバス・インタフェースに対応しています。

備考 内部ROM, RAM容量は、製品によって異なります。

## ★ 1. 16 $\mu$ PD784938Aサブシリーズ製品概要 ( $\mu$ PD784935A, 784936A, 784937A, 784938A, 78F4938A)

### 1. 16. 1 特 徴

$\mu$  PD784908サブシリーズの周辺機能を継承

最小命令実行時間：320 ns (  $f_{xx} = 6.29$  MHz動作時 )

160 ns (  $f_{xx} = 12.5$  MHz動作時 )

内部メモリ

- ・ マスクROM : 96 Kバイト (  $\mu$  PD784935A )  
128 Kバイト (  $\mu$  PD784936A )  
192 Kバイト (  $\mu$  PD784937A )  
256 Kバイト (  $\mu$  PD784938A )
- ・ フラッシュ・メモリ : 192 Kバイト (  $\mu$  PD78F4938A )
- ・ RAM : 5120バイト (  $\mu$  PD784935A )  
6656バイト (  $\mu$  PD784936A )  
8192バイト (  $\mu$  PD784937A )  
10496バイト (  $\mu$  PD784938A, 78F4938A )

I/Oポート：80本

タイマ/カウンタ：16ビット・タイマ/カウンタ×1

16ビット・タイマ/カウンタ×2

16ビット・タイマ×1

シリアル・インタフェース：4チャンネル

- ・ UART/IOE ( 3線式シリアルI/O ) : 2チャンネル ( ボー・レート・ジェネレータ内蔵 )
- ・ CSI ( 3線式シリアルI/O ) : 2チャンネル

PWM出力：2出力

スタンバイ機能

HALT/STOP/IDLEモード

クロック分周機能

クロック出力機能： $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/2^2$ ,  $f_{xx}/2^3$ ,  $f_{xx}/2^4$ ,  $f_{xx}/2^5$ から選択

外部拡張機能

ROMコレクション機能内蔵

A/Dコンバータ：8ビット分解能×8チャンネル

IEBusコントローラ内蔵

ウォッチドッグ・タイマ：1チャンネル

低消費電力

電源電圧： $V_{DD} = 4.0 \sim 5.5$  V ( 12.58 MHz動作時 )

$V_{DD} = 3.0 \sim 5.5$  V ( 6.29 MHz動作時 )

### 1. 16. 2 応用分野

カー・オーディオなど

## 1. 16. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
μ PD784935AGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
μ PD784936AGF- x x x -3BA	"	"
μ PD784937AGF- x x x -3BA	"	"
μ PD784938AGF- x x x -3BA	"	"
μ PD78F4938AGF-3BA <sup>注</sup>	"	フラッシュ・メモリ

注 開発中

備考 x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
μ PD784935AGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
μ PD784936AGF- x x x -3BA	"	"
μ PD784937AGF- x x x -3BA	"	"
μ PD784938AGF- x x x -3BA	"	"
μ PD78F4938AGF-3BA <sup>注</sup>	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

注 開発中

備考 x x x はROMコード番号です。

1.16.4 機能概要

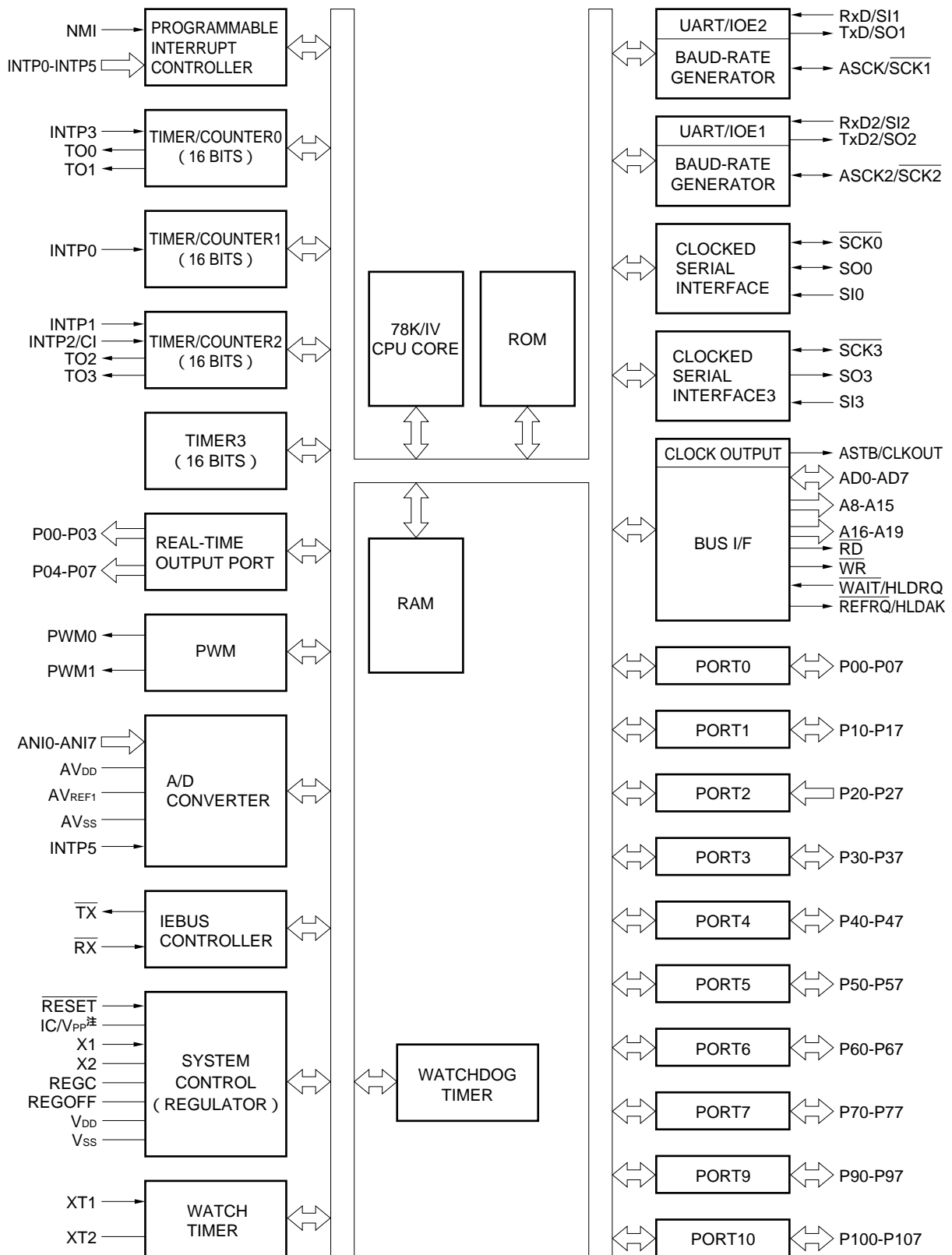
(1/2)

品名		μ PD784935A	μ PD784936A	μ PD784937A	μ PD784938A	μ PD78F4938A	
項目							
基本命令（二モニック）数		113					
汎用レジスタ		8ビット×16レジスタ×8バンク，または16ビット×8レジスタ×8バンク（メモリ・マッピング）					
最小命令実行時間		320 ns/636 ns/1.27 μs/2.54 μs（6.29 MHz動作時） 160 ns/320 ns/636 ns/1.27 μs（12.58 MHz動作時）					
内部メモリ	ROM	96 Kバイト （マスクROM）	128 Kバイト （マスクROM）	192 Kバイト （マスクROM）	256 Kバイト （マスクROM）	256 Kバイト （フラッシュ・メモリ）	
	RAM	5120バイト	6656バイト	8192バイト	10496バイト		
メモリ空間		プログラム/データあわせて1 Mバイト					
I/Oポート	合計	80本					
	入力	8本					
	入出力	72本					
	付加機能付き端子	LEDダイレクト・ドライブ出力	24本				
		トランジスタ・ダイレクト・ドライブ出力	8本				
N-chオープン・ドレイン入出力		4本					
リアルタイム出力ポート		4ビット×2，または8ビット×1					
IEBusコントローラ		内蔵（簡易版）					
タイマ/カウンタ	タイマ/カウンタ0：タイマ・レジスタ×1 （16ビット）		キャプチャ・レジスタ×1 コンペア・レジスタ×2		パルス出力可 ・トグル出力 ・PWM/PPG出力 ・ワンショット・パルス出力		
	タイマ/カウンタ1：タイマ・レジスタ×1 （16ビット）		キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1		リアルタイム出力ポート		
	タイマ/カウンタ2：タイマ・レジスタ×1 （16ビット）		キャプチャ・レジスタ×1 キャプチャ/コンペア・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・トグル出力 ・PWM/PPG出力		
	タイマ3 （16ビット）		：タイマ・レジスタ×1 コンペア・レジスタ×1		パルス出力可 ・トグル出力 ・PWM/PPG出力		
時計用タイマ		0.5秒間隔で割り込み発生（時計クロック発振回路を内蔵） 入力クロックは，メイン・クロック（12.58 MHz）と時計クロック（32.7 kHz）の選択可能					
PWM出力		12ビット分解能×2チャンネル					
シリアル・インタフェース		・UART/IOE（3線式シリアルI/O）：2チャンネル（ポーレート・ジェネレータ内蔵） ・CSI（3線式シリアルI/O）：2チャンネル					

注 付加機能付き端子は，I/O端子の中に含まれています。

品 名		$\mu$ PD784935A	$\mu$ PD784936A	$\mu$ PD784937A	$\mu$ PD784938A	$\mu$ PD78F4938A
項 目						
A/Dコンバータ		8ビット分解能×8チャンネル				
クロック出力機能		f <sub>CLK</sub> , f <sub>CLK</sub> /2, f <sub>CLK</sub> /4, f <sub>CLK</sub> /8, f <sub>CLK</sub> /16から選択 (1ビット出力ポートとしても使用可能)				
ウォッチドッグ・タイマ		1チャンネル				
ROMコレクション機能		内蔵 (修正アドレス4ポイント設定可)				
外部拡張機能		あり (1 M/バイトまで設定可)				
スタンバイ機能		HALT/STOP/IDLEモード				
割り込み	ハードウェア要因	27 (内部: 20, 外部: 7 (サンプリング・クロック可変入力: 1))				
	ソフトウェア要因	BRK命令, BRKCS命令, オペランド・エラー				
	ノンマスクابل	内部: 1, 外部: 1				
	マスクابل	内部: 19, 外部: 6				
		4レベルのプログラマブル・プライオリティ 3種類の処理形態 (マクロ・サービス/ベクタ割り込み/コンテキスト・スイッチング)				
電源電圧		V <sub>DD</sub> = 4.0 ~ 5.5 V (12.58 MHz動作時) V <sub>DD</sub> = 3.0 ~ 5.5 V (6.29 MHz動作時)				
パッケージ		・ 100ピン・プラスチックQFP (14 × 20 mm)				

### 1. 16. 5 ブロック図



注  $\mu$  PD78F4938Aのフラッシュ・メモリ・プログラミング・モード時

備考 内部ROM, RAM容量は製品によって異なります。

## ★ 1. 17 $\mu$ PD784956Aサブシリーズ製品概要 ( $\mu$ PD784953A, 784956A, 78F4956A)

### 1. 17. 1 特 徴

最小命令実行時間：160 ns ( $f_{CLK} = 12.5$  MHz動作時)

内部メモリ

- ・ マスクROM : 24 Kバイト ( $\mu$  PD784953A)  
48 Kバイト ( $\mu$  PD784956A)
- ・ フラッシュ・メモリ : 64 Kバイト ( $\mu$  PD78F4956A)
- ・ RAM : 768バイト ( $\mu$  PD784953A)  
2048バイト ( $\mu$  PD784956A, 78F4956A)

I/Oポート：67本

タイマ/カウンタ：16ビット・タイマ/カウンタ×6  
8ビット・タイマ/カウンタ×2

シリアル・インタフェース：2チャンネル

- ・ UART：1チャンネル (ポー・レート・ジェネレータ内蔵)
- ・ CSI (3線式シリアルI/O)：1チャンネル

A/Dコンバータ：8ビット分解能×8チャンネル

リアルタイム出力機能：6ビット分解能×2チャンネル

ウォッチドッグ・タイマ：1チャンネル

スタンバイ機能：HALT/STOP/IDLEモード

低消費電力モード時：HALT/IDLEモード (サブシステム・クロック動作時)

割り込みコントローラ (4レベル・プライオリティ)

- ・ ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング

電源電圧： $V_{DD} = 4.5 \sim 5.5$  V

### 1. 17. 2 応用分野

インバータ・エアコンなどのモータ制御用

## 1. 17. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784953AGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	マスクROM
$\mu$ PD784956AGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
$\mu$ PD78F4956AGC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	フラッシュ・メモリ

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784953AGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	標準 (一般電子機器用)
$\mu$ PD784956AGC- x x x -8BT	80ピン・プラスチックQFP (14 x 14 mm)	"
$\mu$ PD78F4956AGC-8BT	80ピン・プラスチックQFP (14 x 14 mm)	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

**注意**  $\mu$  PD784956Aサブシリーズは開発中です。

1. 17. 4 機能概要

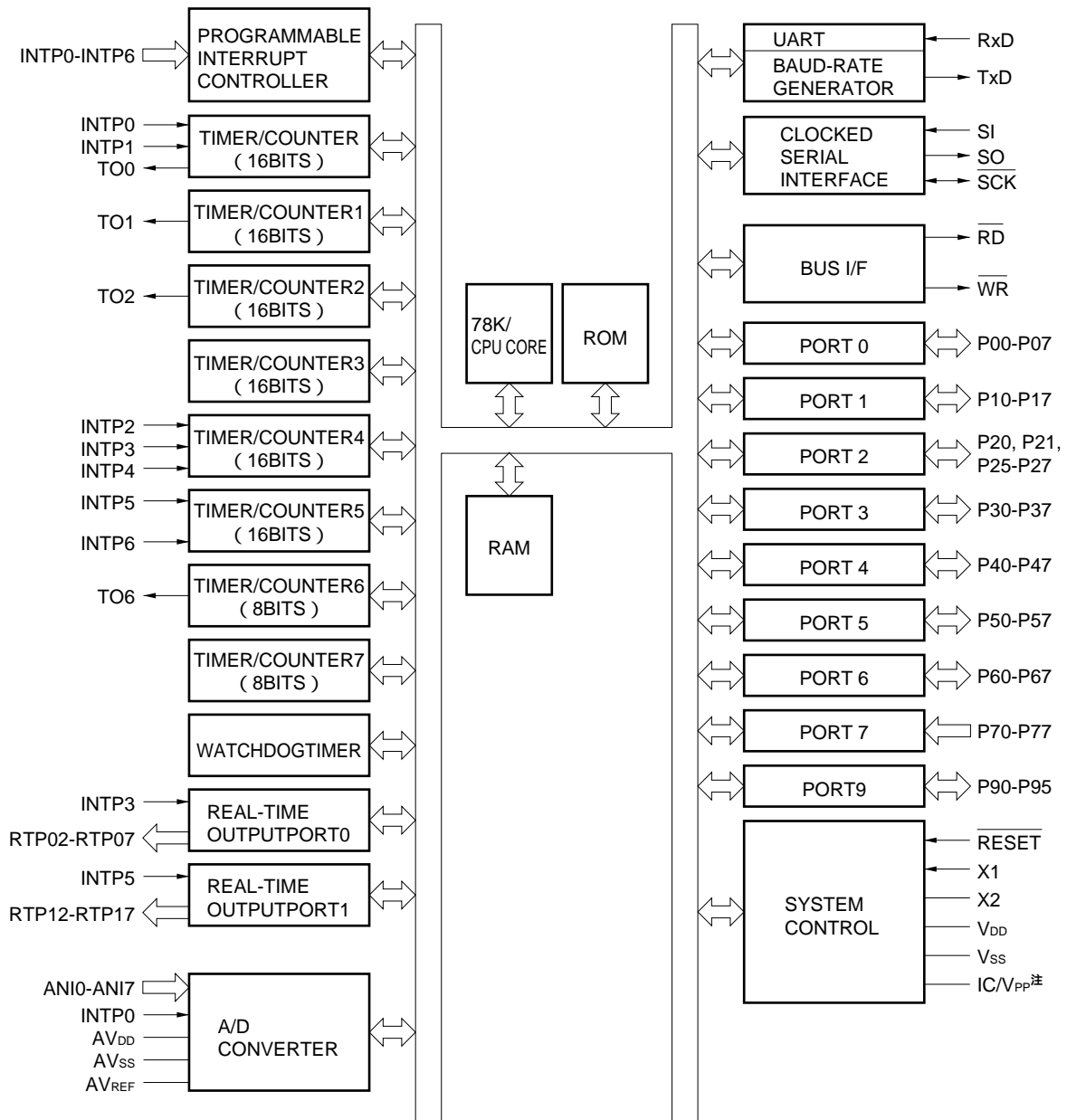
( 1/2 )

品 名		$\mu$ PD784953A	$\mu$ PD784956A	$\mu$ PD78F4956A
項 目				
基本命令 (ニモニック) 数		113		
汎用レジスタ		8ビット×16×8バンク, または16ビット×8×8バンク (メモリ・マッピング)		
最小命令実行時間		160 ns ( $f_{CLK} = 12.5$ MHz動作時)		
内部メモリ	ROM	24 Kバイト (マスクROM)	64 Kバイト (マスクROM)	64 Kバイト (フラッシュ・メモリ)
	RAM	768バイト	2048バイト	
I/Oポート	合計	67本		
	CMOS入力	8本		
	CMOS入出力	59本		
	注 付加機能付き端子	ブルアップ抵抗付き端子	59本	
LEDダイレクト・ドライブ出力		32本		
リアルタイム出力ポート		6ビット×2		
タイマ/カウンタ	16ビット・タイマ/カウンタ0 : タイマ・レジスタ×1		パルス出力可	
	キャプチャ/コンペア・レジスタ×2		・PWM出力	
	16ビット・タイマ/カウンタ1 : タイマ・レジスタ×1		パルス出力可	
	コンペア・レジスタ×2		・PWM出力	
	16ビット・タイマ/カウンタ2 : タイマ・レジスタ×1		パルス出力可	
	コンペア・レジスタ×2		・PWM出力	
	16ビット・タイマ/カウンタ3 : タイマ・レジスタ×1		コンペア・レジスタ×2	
	16ビット・タイマ/カウンタ4 : タイマ・レジスタ×1		キャプチャ/コンペア・レジスタ×3	
16ビット・タイマ/カウンタ5 : タイマ・レジスタ×1		コンペア・レジスタ×1		
キャプチャ/コンペア・レジスタ×2		8ビット・タイマ/カウンタ6 : タイマ・レジスタ×1		
コンペア・レジスタ×1		パルス出力可		
8ビット・タイマ/カウンタ7 : タイマ・レジスタ×1		・PWM出力		
コンペア・レジスタ×1				
シリアル・インタフェース		・ UART : 1チャンネル (ポー・レート・ジェネレータ内蔵) ・ CSI (3線式シリアルI/O) : 1チャンネル		
A/Dコンバータ		8ビット分解能×8チャンネル		
ウォッチドッグ・タイマ		1チャンネル		
スタンバイ機能		HALT/STOP/IDLEモード		

注 付加機能付き端子は, I/O端子の中に含まれています。

品 名		$\mu$ PD784953A	$\mu$ PD784956A	$\mu$ PD78F4956A
項目	割り込み	ハードウェア要因		
		28要因 (外部8 (内部兼用2), 内部22)		
		ソフトウェア要因		
		BRK命令, BRKCS命令, オペランド・エラー		
	ノンマスクابل	内部: 1, 外部: 1		
	マスクابل	内部: 20, 外部: 7		
		・4レベルのプログラマブル・プライオリティ ・3種類の処理形態: ベクタ割り込み/マクロ・サービス/コンテキスト・スイッチング		
	電源電圧	$V_{DD} = 4.5 \sim 5.5 V$		
	パッケージ	80ピン・プラスチックQFP (14 x 14 mm)		

1.17.5 ブロック図



注 μ PD78F4956Aのフラッシュ・メモリ・プログラミング・モード時

備考 内部ROM, RAM容量は製品によって異なります。

★ 1. 18  $\mu$  PD784976Aサブシリーズ製品概要 ( $\mu$  PD784975A, 78F4976A)

## 1. 18. 1 特 徴

最小命令実行時間：160 ns ( $f_{xx} = 12.58$  MHz動作時)

内部メモリ

- ・マスクROM : 96 Kバイト ( $\mu$  PD784975A)
- ・フラッシュ・メモリ : 128 Kバイト ( $\mu$  PD78F4976A)
- ・RAM : 3072バイト ( $\mu$  PD784975A)  
4608バイト ( $\mu$  PD78F4976A)

I/Oポート：72本

VFDコントローラ/ドライバ：表示出力48本 (ユニバーサル・グリッド対応)

- ・表示電流 10mA : 16本
- ・表示電流 3 mA : 32本

タイマ/カウンタ：16ビット・タイマ/イベント・カウンタ×1

8ビットPWMタイマ×1

シリアル・インタフェース：3チャンネル

- ・3線式シリアルI/Oモード：2チャンネル
- ・UART/IOE (3線式シリアルI/O)：1チャンネル

A/Dコンバータ：8ビット分解能×12チャンネル

ウォッチドッグ・タイマ：1チャンネル

スタンバイ機能：HALT/STOP/IDLEモード

電源電圧： $V_{DD} = 4.5 \sim 5.5$  V

## 1. 18. 2 応用分野

一体型ミニコンポ，セパレート型ミニコンポ，チューナ，カセット・デッキ，CDプレーヤ，オーディオ・アンプなど

## 1. 18. 3 オーダ情報と品質水準

## (1) オーダ情報

オーダ名称	パッケージ	内部ROM
$\mu$ PD784975AGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	マスクROM
$\mu$ PD78F4976AGF-3BA	"	ワン・タイムPROM

**備考** x x x はROMコード番号です。

## (2) 品質水準

オーダ名称	パッケージ	品質水準
$\mu$ PD784975AGF- x x x -3BA	100ピン・プラスチックQFP ( 14 x 20 mm )	標準 ( 一般電子機器用 )
$\mu$ PD78F4976AGF-3BA	"	"

品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

**備考** x x x はROMコード番号です。

**注意**  $\mu$  PD784976Aサブシリーズは開発中です。

1. 18. 4 機能概要

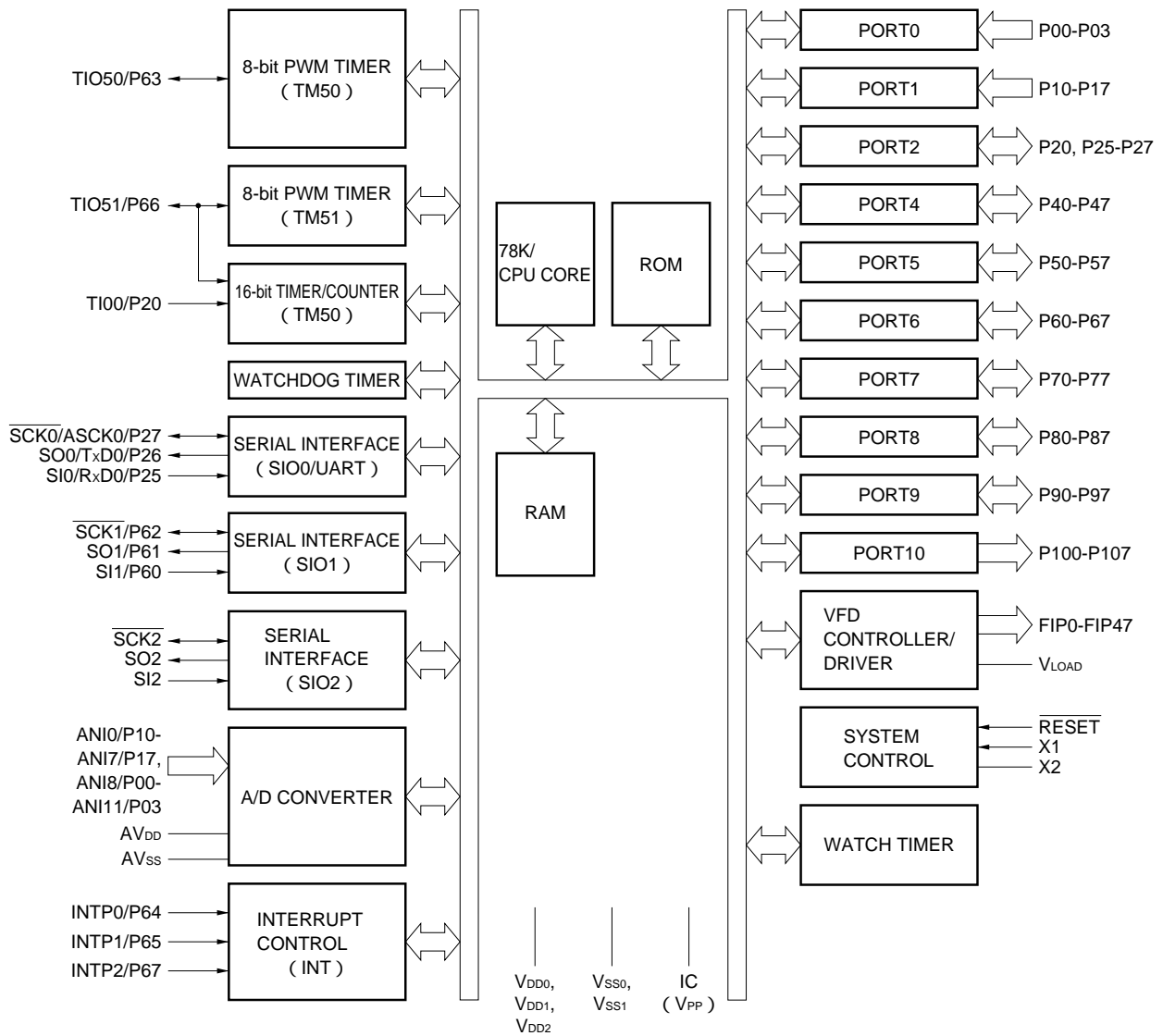
( 1/2 )

品 名		$\mu$ PD784975A	$\mu$ PD78F4976A	
項 目				
基本命令 (二モニック数)		113		
汎用レジスタ		8ビット×16レジスタ×8バンク, または16ビット×8レジスタ×8バンク (メモリ・マッピング)		
最小命令実行時間		160 ns/320 ns/640 ns/1280 ns/2560 ns ( f <sub>xx</sub> = 12.5 MHz動作時 )		
内部メモリ	ROM	96 Kバイト (マスクROM)	128 Kバイト (フラッシュ・メモリ)	
	RAM	3072バイト	4608バイト	
	VFD表示用RAM	96バイト		
メモリ空間		プログラム / データあわせて1 Mバイト		
I/Oポート ( VFD 兼用 端 子 を 含 む )	合計	72本		
	CMOS入力	12本		
	CMOS入出力	20本		
	N-chオープン・ドレ ーン入出力	8本		
	P-chオープン・ドレ ーン入出力	24本		
	P-chオープン・ドレ ーン入力	8本		
	注 付 加 機 能 付 き 端 子	ブルアップ抵抗付き 端子	20本	
		LEDダイレクト・ド ライブ出力	16本	
		高耐圧端子	32本	
		中耐圧端子	8本	
VFDコントローラ・ドライバ		・表示出力合計 : 48本 ・表示電力 10 mA : 16本 ・表示電力 3 mA : 32本		
タイマ / カウンタ		タイマ / カウンタ : タイマ・レジスタ×1 (16ビット)                      キャプチャ / コンペア・レジスタ×2		
		PWMタイマ50 : タイマ・レジスタ×1 (8ビット)                      コンペア・レジスタ×1	パルス出力可 ・PWM出力	
		PWMタイマ51 : タイマ・レジスタ×1 (8ビット)                      コンペア・レジスタ×1	パルス出力可 ・PWM出力	
シリアル・インタフェース		・UART/IOE (3線式シリアルI/O) : 1チャンネル (ポー・レート・ジェネレータ内蔵) ・CSI (3線式シリアルI/O) : 2チャンネル		
A/Dコンバータ		8ビット分解能×12チャンネル		
時計用タイマ		1チャンネル		
ウォッチドッグ・タイマ		1チャンネル		
スタンバイ機能		HALT/STOP/IDLEモード		

注 付加機能付き端子は, I/O端子の中に含まれています。

品 名		$\mu$ PD784975A	$\mu$ PD78F4976A
項 目			
割り込み	ハードウェア	12要因 ( 外部7, 外部3, 内外部兼用2 )	
	ソフトウェア	BRK命令, BRKCS命令, オペランド・エラー	
	ノンマスクابل	内部 : 1, 外部 : 1	
	マスクابل	内部 : 14, 外部 : 3, 内外部兼用 : 2	
		・ 4レベルのプログラマブル・プライオリティ ・ 3種類の処理形態 : ベクタ割り込み / マクロ・サービス / コンテキスト・スイッチング	
電源電圧		$V_{DD} = 4.5 \sim 5.5 V$	
パッケージ		100ピン・プラスチックQFP ( 14 x 20 mm )	

1. 18. 5 ブロック図



- 備考1. 内部ROM容量は製品によって異なります。  
 2.  $V_{PP}$ 端子は、 $\mu$  PD78F4976Aのときのみ。

## 第2章 メモリ空間

### 2.1 メモリ空間

78K/ シリーズは、最大16 Mバイトのメモリ空間をアクセスできます。しかし、内蔵するメモリの容量や端子の状態によって、メモリのマッピングが製品ごとに異なります。したがって、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**ユーザズ・マニュアル ハードウェア編**を参照してください。

78K/ シリーズは、16 Mバイトの空間をアクセスできます。LOCATION命令によって、内部データ領域（特殊機能レジスタおよび内部RAM）のマッピングが異なります。LOCATION命令は、リセット解除後に必ず実行する必要があり、2回以上使用することはできません。

リセット解除後のプログラムは、次のようになっている必要があります。

```
RSTVCT  CSEG  AT  0
        DW   RSTSTRT
        }
INITSEG  CSEG  BASE
RSTSTRT : LOCATION 0H ; or LOCATION 0FH
        MOVG SP, #STKBGN
```

#### (1) LOCATION 0H命令を実行した場合

内部データ領域は、その最上位アドレスがFFFFHになるようにマッピングされます。

内部ROMのうち、内部データ領域と重なる領域は、LOCATION 0H命令を実行した場合には内部ROMとして使用できません。

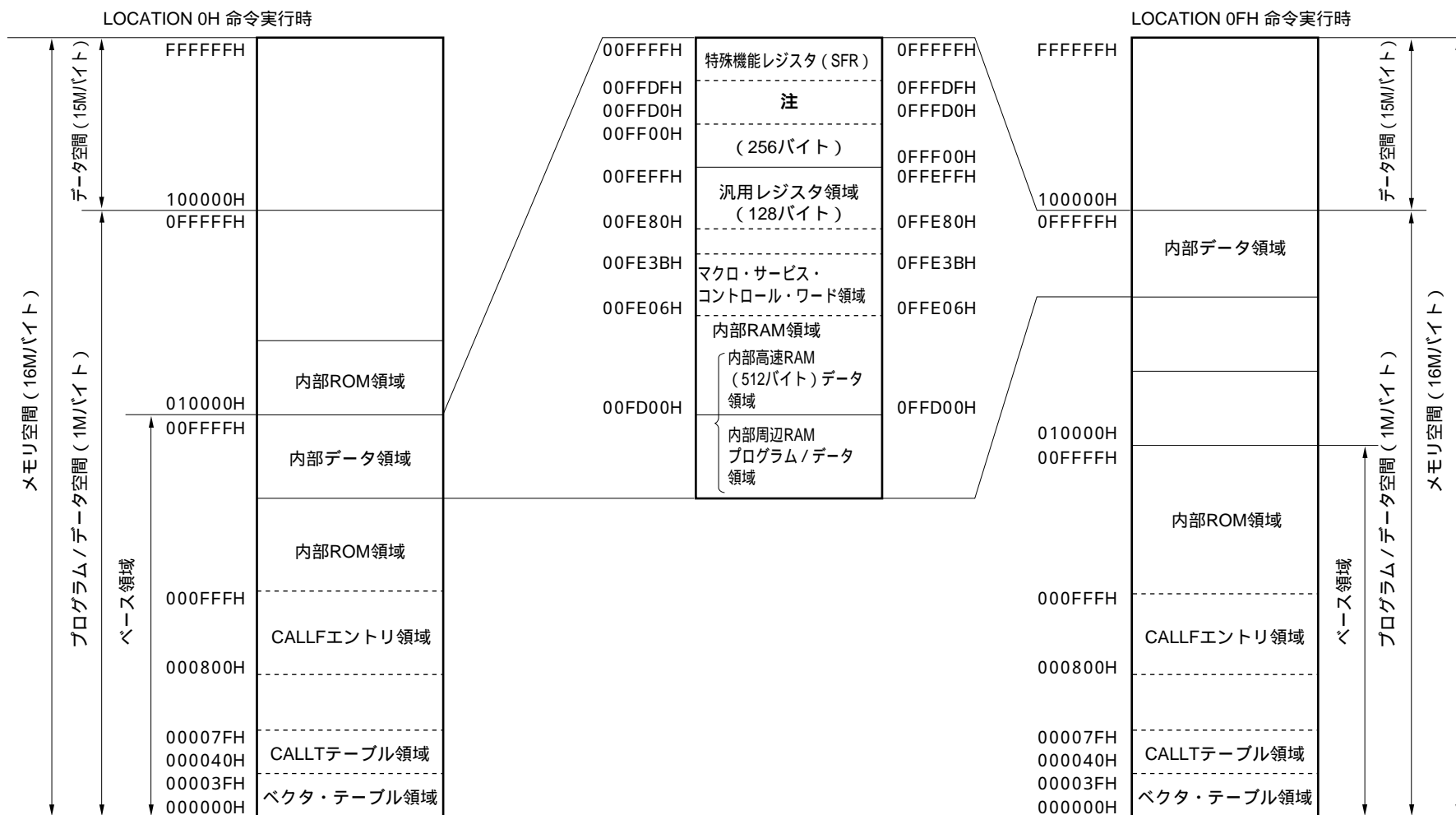
外部メモリは、外部メモリ拡張モードでアクセスします。

#### (2) LOCATION 0FH命令を実行した場合

内部データ領域は、その最上位アドレスがFFFFFFHになるようにマッピングされます。

なお、 $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。

図2-1 メモリ・マップ



注 外部SFR領域

注意 μPD784915サブシリーズは、LOCATION 0H命令固定です。

## 2.2 内部ROM領域

78K/ シリーズの次に示す製品は、ROMを内蔵しており、プログラムやテーブル・データなどを格納できます。

また、LOCATION 0H命令実行時に内部ROM領域と内部データ領域が重なった場合は、内部データ領域がアクセスの対象となり、重なった部分の内部ROM領域はアクセスできません。

なお、 $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。

表2 - 1 78K/ シリーズ製品別内部ROM領域一覧 (1/2)

サブシリーズ名	品 名	アドレス空間	内部ROM
$\mu$ PD784026サブシリーズ	$\mu$ PD784020	なし	
	$\mu$ PD784021		
	$\mu$ PD784025	00000H-0BFFFH	48 K×8ビット
	$\mu$ PD784026 $\mu$ PD78P4026	00000H-0FFFFH	64 K×8ビット
$\mu$ PD784038サブシリーズ $\mu$ PD784038Yサブシリーズ	$\mu$ PD784031	なし	
	$\mu$ PD784031Y		
	$\mu$ PD784035	00000H-0BFFFH	48 K×8ビット
	$\mu$ PD784035Y		
	$\mu$ PD784036	00000H-0FFFFH	64 K×8ビット
	$\mu$ PD784036Y		
	$\mu$ PD784037	00000H-17FFFH	96 K×8ビット
	$\mu$ PD784037Y		
$\mu$ PD784046サブシリーズ	$\mu$ PD784044	00000H-07FFFH	32 K×8ビット
	$\mu$ PD784054		
	$\mu$ PD784046	00000H-0FFFFH	64 K×8ビット
	$\mu$ PD78F4046		
$\mu$ PD784216Aサブシリーズ $\mu$ PD784216Aサブシリーズ	$\mu$ PD784214A	00000H-17FFFH	96 K×8ビット
	$\mu$ PD784214AY		
	$\mu$ PD784215A	00000H-1FFFFH	128 K×8ビット
	$\mu$ PD784215AY		
	$\mu$ PD784216A		
	$\mu$ PD78F4216A		
$\mu$ PD784218Aサブシリーズ $\mu$ PD784218AYサブシリーズ	$\mu$ PD784217A	00000H-2FFFFH	192 K×8ビット
	$\mu$ PD784217AY		
	$\mu$ PD784218A	00000H-3FFFFH	256 K×8ビット
	$\mu$ PD784218AY		
	$\mu$ PD78F4218A		
	$\mu$ PD78F4218AY		

備考 ROMレス製品の場合は、このアドレス空間は外部メモリとなります。

表2 - 1 78K/ シリーズ製品別内部ROM領域一覧 (2/2)

サブシリーズ名	品 名	アドレス空間	内部ROM
μ PD784225サブシリーズ μ PD784225Yサブシリーズ	μ PD784224	00000H-17FFFH	96 K×8ビット
	μ PD784224Y		
	μ PD784225	00000H-1FFFFH	128 K×8ビット
	μ PD784225Y μ PD78F4225 μ PD78F4225Y		
μ PD784908サブシリーズ	μ PD784907	00000H-17FFFH	96 K×8ビット
	μ PD784908	00000H-1FFFFH	128 K×8ビット
	μ PD78P4908		
μ PD784915サブシリーズ	μ PD784915B	00000H-0BFFFH	48 K×8ビット
	μ PD784916B	00000H-0F6FFFH	62 K×8ビット
	μ PD78P4916		
★ μ PD784928サブシリーズ ★ μ PD784928Yサブシリーズ	μ PD784927	00000H-17FFFH	96 K×8ビット
	μ PD784927Y		
	μ PD784928	00000H-1FFFFH	128 K×8ビット
	μ PD784928Y μ PD78F4928 μ PD78F4928Y		
★ μ PD784938Aサブシリーズ	μ PD784935A	00000H-17FFFH	96 K×8ビット
	μ PD784936A	00000H-1FFFFH	128 K×8ビット
	μ PD784937A	00000H-2FFFFH	192 K×8ビット
	μ PD784938A	00000H-3FFFFH	256 K×8ビット
	μ PD78F4938A		
★ μ PD784956Aサブシリーズ	μ PD784953A	00000H-05FFFH	24 K×8ビット
	μ PD784956A	00000H-0F6FFFH	64 K×8ビット
	μ PD78F4956A	00000H-0F6FFFH	64 K×8ビット
★ μ PD784976Aサブシリーズ	μ PD784975A	00000H-17FFFH	96 K×8ビット
	μ PD78F4976A	00000H-1FFFFH	128 K×8ビット

## 2.3 ベース領域

00000H-FFFFFFHの空間は、ベース領域となっています。次の用途については、ベース領域が対象となります。

- リセットのエントリ・アドレス
- 割り込みのエントリ・アドレス
- CALLT命令のエントリ・アドレス
- 16ビット・イミューディエト・アドレッシング・モード（命令アドレスのアドレッシング時）
- 16ビット・ダイレクト・アドレッシング・モード
- 16ビット・レジスタ・アドレッシング・モード（命令アドレスのアドレッシング時）
- 16ビット・レジスタ・インダイレクト・アドレッシング・モード
- ショート・ダイレクト16ビット・メモリ・インダイレクト・アドレッシング・モード

また、ベース領域には、ベクタ・テーブル領域、CALLT命令テーブル領域、CALLF命令エントリ領域が割り付けられています。

なお、LOCATION 0H命令実行時には、内部データ領域がベース領域内に配置されます。内部データ領域のうち、内部高速RAM領域および特殊機能レジスタ（SFR）領域からは、プログラムのフェッチは行えませんので注意が必要です。また、内部RAM領域のデータは、初期化を行ってから使用してください。

### 2.3.1 ベクタ・テーブル領域

00000H-0003FHの64バイト領域はベクタ・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RESET入力、各割り込み要求発生により分岐する場合のプログラム・スタート・アドレスを格納しておきます。また、各割り込みでコンテキスト・スイッチングを使用する場合に、切り替え先のレジスタ・バンクの番号を格納します。

なお、ベクタ・テーブルとして使用していない部分は、プログラム・メモリまたはデータ・メモリとして使用できます。

ベクタ・テーブルに書ける値は、16ビットの値です。したがって、分岐できるのはベース領域のみです。

表2-2 ベクタ・テーブル

ベクタ・テーブル・アドレス	割り込み
00000H	リセット（RESET入力）
00002H	NMI <sup>注</sup>
00004H	WDT <sup>注</sup>
00006H	} 各製品で異なります
}	
0003AH	
0003CH	オペランド・エラー割り込み
0003EH	BRK

注 使用しない製品もあり

### 2.3.2 CALLT命令テーブル領域

00040H-0007FH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트리・アドレスを格納することができます。

CALLT命令では、このテーブルを参照し、テーブル中に書かれているベース領域のアドレスへサブルーチンとして分岐します。CALLT命令は1バイトであるため、プログラム中で記述回数の多いサブルーチン・コールをCALLT命令とすることで、プログラムのオブジェクト・サイズを圧縮することが可能です。なお、テーブルには最大32個のサブルーチン・エン트리・アドレスを記述できますので、記述頻度の多い順に登録することをお勧めします。

また、CALLT命令のテーブルとして使用しない場合は、通常のプログラム・メモリまたはデータ・メモリとして使用可能です。

CALLT命令のテーブルに書ける値は、16ビットの値です。したがって、分岐できるのは、ベース領域のみです。

### 2.3.3 CALLF命令エン트리領域

00800H-00FFFHの領域は、2バイト・コール命令（CALLF）で直接サブルーチン・コールすることができます。

CALLF命令は、2バイトのコール命令であるため、直接サブルーチン・コールのCALL命令（3バイト）を使用した場合に比べ、オブジェクト・サイズを圧縮することが可能です。

高速性を生かしたいときには、この領域に直接サブルーチンを記述することが有効です。

オブジェクト・サイズを小さくしたい場合には、この領域に無条件分岐（BR）命令を記述し、サブルーチン本体はこの領域外に置くことで、5カ所以上からコールされているサブルーチンについて、オブジェクト・サイズの圧縮が図れます。この場合、CALLFエン트리領域中ではBR命令の4バイトしか場所をとらないため、多くのサブルーチンでオブジェクト・サイズの圧縮が可能です。

## 2.4 内部データ領域

内部データ領域は、内部RAM領域、特殊機能レジスタ領域で構成される領域です。また、製品によっては、その他のハードウェアに依存したメモリなどがこの領域に割り付けられます(各製品の**ユーザーズ・マニュアル ハードウェア編**参照)。

内部データ領域は、LOCATION命令によって、内部データ領域の最終アドレスをFFFFH (LOCATION 0H命令実行時)にするか、FFFFFFH (LOCATION 0FH命令実行時)にするかを選択できます。このLOCATION命令による内部データ領域のアドレスの選択は、リセット解除直後に必ず1回実行する必要があるため、一度選択したあとは、変更することはできません。リセット解除後のプログラムは、例のようにしている必要があります。内部データ領域と他の領域が同一アドレスに割り当てられた場合は、内部データ領域がアクセスの対象となり、他の領域にはアクセスできません。

```
例 RSTVCT   CSEG  AT 0
      DW     RSTSTRT
      {
INITSEG   CSEG  BASE
RSTSTRT : LOCATION 0H ; or LOCATION 0FH
      MOVG SP, #STKBGN
```

**注意1** . LOCATION 0H命令実行時には、リセット解除後のプログラムが内部データ領域に重ならないようにする必要があります。また、NMIなどのマスク不可能な割り込み処理ルーチンのエントリ・アドレスも、内部データ領域と重ならないようにしてください。なお、マスク可能割り込みのエントリ領域などについては、内部データ領域を参照する前に初期化を行う必要があります。

2 .  $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。

### 2.4.1 内部RAM領域

78K/ シリーズ製品は、汎用スタティックRAMを内蔵しています。

この領域は、次のように構成されています。

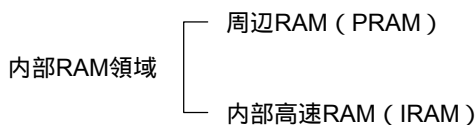


表2 - 3 78K/ シリーズ製品別内部RAM領域一覧 (1/2)

サブシリーズ名	品 名	内部RAM領域		
			周辺RAM : PRAM	内部高速RAM : IRAM
μ PD784026サブシリーズ	μ PD784020	512バイト (0FD00H-0FEFFH)	0バイト	512バイト (0FD00H-0FEFFH)
	μ PD784021	2048バイト (0F700H-0FEFFH)	1536バイト	
	μ PD784025		(0F700H-0FCFFH)	
	μ PD784026 μ PD78P4026			
μ PD784038サブシリーズ μ PD784038Yサブシリーズ	μ PD784031	2048バイト (0F700H-0FEFFH)	1536バイト	
	μ PD784031Y		(0F700H-0FCFFH)	
	μ PD784035 μ PD784036			
	μ PD784035Y μ PD784036Y			
	μ PD784037	3584バイト	3072バイト	
	μ PD784037Y	(0F100H-0FEFFH)	(0F100H-0FCFFH)	
	μ PD784038 μ PD78P4038 μ PD784038Y μ PD78P4038Y	4352バイト (0EE00H-0FEFFH)	3840バイト (0FE00H-0FCFFH)	
μ PD784046サブシリーズ	μ PD784044	1024バイト	512バイト	
	μ PD784054	(0FB00H-0FEFFH)	(0FB00H-0FCFFH)	
	μ PD784046 μ PD78F4046	2048バイト (0F700H-0FEFFH)	1536バイト (0F700H-0FCFFH)	
★ μ PD784216Aサブシリーズ ★ μ PD784216AYサブシリーズ	μ PD784214A	3584バイト	3072バイト	
	μ PD784214AY	(0F100H-0FEFFH)	(0F100H-0FCFFH)	
	μ PD784215A	5120バイト	4608バイト	
	μ PD784215AY	(0EB00H-0FEFFH)	(0EB00H-0FCFFH)	
	μ PD784216A μ PD784216AY μ PD78F4216A μ PD78F4216AY	8192バイト (0DF00H-0FEFFH)	7680バイト (0DF00H-0FCFFH)	
★ μ PD784218Aサブシリーズ ★ μ PD784218AYサブシリーズ	μ PD784217A	12800バイト	12288バイト	
	μ PD784217AY	(0CD00H-0FEFFH)	(0CD00H-0FCFFH)	
	μ PD784218A μ PD784218AY μ PD78F4218A μ PD78F4218AY			

備考 表中のアドレスは、LOCATION 0H命令実行時の値です。LOCATION 0FH命令実行時には、上記の値に0F000Hを加えてください。

表2 - 3 78K/ シリーズ製品別内部RAM領域一覧 (2/2)

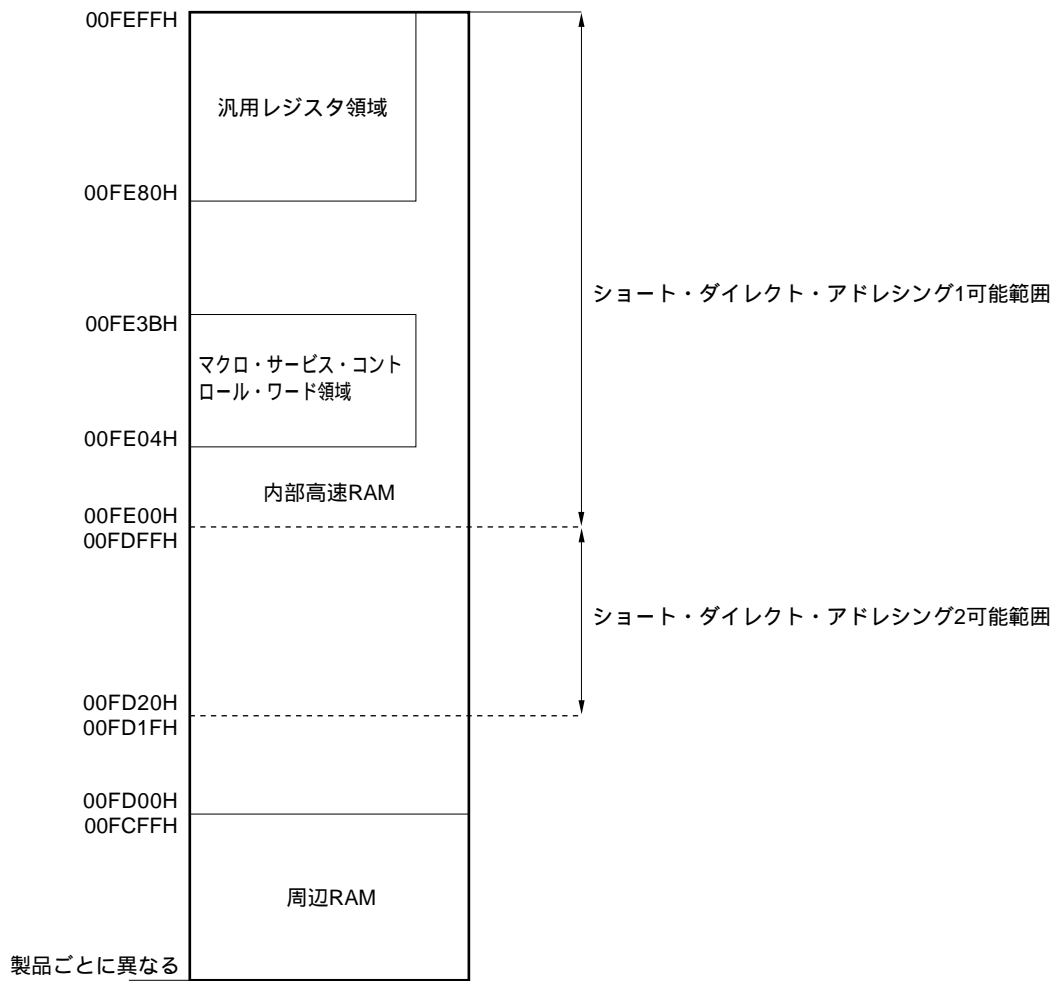
サブシリーズ名	品 名	内部RAM領域			
			周辺RAM : PRAM	内部高速RAM : IRAM	
μ PD784225サブシリーズ μ PD784225Yサブシリーズ	μ PD784224	3584バイト	3072バイト	512バイト (0FD00H-0FEFFH)	
	μ PD784224Y	(0F100H-0FEFFH)	(0F100H-0FCFFH)		
	μ PD784225	4352バイト	3840バイト		
	μ PD784225Y μ PD78F4225 μ PD78F4225Y	(0EE00H-0FEFFH)	(0EE00H-0FCFFH)		
μ PD784908サブシリーズ	μ PD784907	3584バイト	3072バイト		
	μ PD784908 μ PD78P4908	(0F100H-0FEFFH)	(0F100H-0FCFFH)		
μ PD784915サブシリーズ	μ PD784915B	1280バイト	768バイト		
	μ PD784916B	(0FA00H-0FEFFH)	(0FA00H-0FCFFH)		
	μ PD78P4916	2048バイト	1536バイト		
μ PD784928サブシリーズ μ PD784928Yサブシリーズ	μ PD784927	2048バイト	1536バイト		
	μ PD784927Y	(0F700H-0FEFFH)	(0F700H-0FCFFH)		
★	μ PD784928	3584バイト			
	μ PD784928Y	(0F100H-0FEFFH)			
	μ PD78F4928				
	μ PD78F4928Y				
★	μ PD784938Aサブシリーズ	μ PD784935A	5120バイト	4608バイト	
		μ PD784936A	6656バイト	6144バイト	
		μ PD784937A	8192バイト	7680バイト	
		μ PD784938A μ PD78F4938A	10496バイト	9984バイト	
★	μ PD784956Aサブシリーズ	μ PD784953A	768バイト	256バイト	
		μ PD784956A	2048バイト	1536バイト	
		μ PD78F4956A	(0F700H-0FEFFH)	(0F700H-0FCFFH)	
★	μ PD784976Aサブシリーズ	μ PD784975A	3584バイト	3072バイト	
		μ PD78F4976A	5120バイト	4608バイト	
		(0EB00H-0FEFFH)	(0EB00H-0FCFFH)		

備考 表中のアドレスは、LOCATION 0H命令実行時の値です。LOCATION 0FH命令実行時には、上記の値に0F000Hを加えてください。

なお、μ PD784915サブシリーズは、LOCATION 0H命令固定です。

内部RAMのメモリ・マップを図2 - 2に示します。

図2 - 2 内部RAMのメモリ・マップ



**備考** 図中のアドレスは、LOCATION 0H命令実行時の値です。LOCATION 0FH命令実行時には、上記の値に0F0000Hを加えてください。

なお、μPD784915サブシリーズは、LOCATION 0H命令固定です。

### (1) 内部高速RAM (IRAM)

内部高速RAM (IRAM) は、高速アクセスが可能です。このうち、0FD20H-0FEFFHは、高速アクセス用のショート・ダイレクト・アドレッシング・モードが使用できます。ショート・ダイレクト・アドレッシング・モードには、その対象となるアドレスにより、ショート・ダイレクト・アドレッシング1とショート・ダイレクト・アドレッシング2の2種類があります。いずれのアドレッシング・モードも、その機能は同一です。一部の命令では、ショート・ダイレクト・アドレッシング2は、ショート・ダイレクト・アドレッシング1より語長が短くなっています。詳細は、第6章 命令セットを参照してください。

IRAMから、プログラムのフェッチを行うことはできません。IRAMが、マッピングされているアドレスからプログラムのフェッチを行うと、CPUは暴走します。

また、IRAMには、次の領域が予約されています。

汎用レジスタ領域

0FE80H-0FEFFH

マクロ・サービス・コントロール・ワード領域

0FE06H-0FE3BH (実際に予約されるアドレスについては、製品ごとに異なります)

マクロ・サービス・チャンネル領域

0FE00H-0FEFFH (アドレスは、マクロ・サービス・コントロール・ワードで指定します)

これらの領域で、予約されている機能を使用していない場合は、通常のデータ・メモリとして使用できます。

**備考** 本文中のアドレスは、LOCATION 0H命令実行時のアドレスです。LOCATION 0FH命令実行時には、本文中の値に0F0000Hを加えてください。

なお、 $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。

### (2) 周辺RAM (PRAM)

周辺RAM (PRAM) は、普通のプログラム・メモリまたはデータ・メモリとして使用します。プログラム・メモリとして使用する場合は、事前に、プログラムによって周辺RAMにプログラムを書き込んでおく必要があります。

## 2.4.2 特殊機能レジスタ (SFR) 領域

0FF00H-0FFFFH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) がマッピングされています (各製品のユーザーズ・マニュアル ハードウェア編参照)。

また、一部の製品では、0FFD0H-0FFDFHの領域は、外部SFR領域としてマッピングされており、ROMレス製品やROM内蔵の製品で外部メモリ拡張モード (メモリ拡張モード・レジスタ (MM) で設定) 時に外付けされた周辺I/Oなどをアクセスすることができます。

**注意** この領域内で、SFRのマッピングさ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となることがあります。デッドロック状態は、リセット入力によってのみ解除されます。

**備考** 本文中のアドレスは、LOCATION 0H命令実行時のアドレスです。LOCATION 0FH命令実行時には、本文中の値に0F0000Hを加えてください。

なお、 $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。

## 2.4.3 外部SFR領域

78K $\mu$  シリーズの一部の製品では、SFR領域のうち、0FFD0H-0FFDFH (LOCATION 0H命令実行時、LOCATION 0FH命令実行時は0FFFD0H-0FFDFH) の16バイトの領域は、外部SFR領域としてマッピングされています。ROMレス品やROM内蔵の製品で外部メモリ拡張モードの場合に、アドレス・バスおよびアドレス/データ・バスなどを使用して、外付けされた周辺I/Oなどをアクセスすることができます。

外部SFR領域は、SFRアドレッシングでアクセスすることができるので、周辺I/Oなどの操作が簡単にできる、オブジェクト・サイズを圧縮できる、マクロ・サービスを使用できるなどの特徴があります。

なお、外部SFR領域に対するアクセス時のバスの動作は、通常のメモリ・アクセスと同様になります。

## 2.5 外部メモリ空間

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

なお、外部メモリ空間のうち、100000H-0FFFFFFHの領域にはプログラムを配置することはできません。

また、一部の製品では外部メモリ空間を持っていません。

## 第3章 レジスタ

### 3.1 制御レジスタ

制御レジスタには、プログラム・カウンタ（PC）、プログラム・ステータス・ワード（PSW）、スタック・ポインタ（SP）があります。

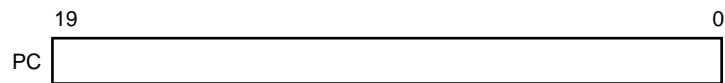
#### 3.1.1 プログラム・カウンタ（PC）

次に実行するプログラムのアドレス情報を保持する20ビット・バイナリ・カウンタです（図3-1参照）。

通常、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。分岐を伴う命令を実行した場合、イミディエト・データやレジスタの内容などがセットされます。

$\overline{\text{RESET}}$ 入力により、0番地と1番地の16ビット・データがPCの下位16ビットに、0000がPCの上位4ビットにセットされます。

図3-1 プログラム・カウンタ（PC）の構成



### 3.1.2 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 (PSW) は、命令の実行の結果によってセット、リセットされる各種フラグで構成される16ビット・レジスタです。

上位8ビット (PSWH)、下位8ビット (PSWL) 単位でリード・アクセス/ライト・アクセスします。また、ビット操作命令により、各フラグの操作ができます。

PSWの内容は、ベクタ割り込み要求の受け付け時、およびBRK命令の実行時に自動的にスタックに退避し、RETI命令またはRETB命令の実行時に自動的に復帰します。また、コンテキスト・スイッチング使用時には、RP3に自動的に退避し、RETCS命令またはRETCSB命令の実行時に自動的に復帰します。

RESET入力により、全ビットがリセット (0) されます。

図3-2で“0”と書かれているビットには、必ず“0”を書き込んでください。また、“-”と書かれているビットの内容は、読み出し時には不定となります。

図3-2 プログラム・ステータス・ワード (PSW) の構成

	7	6	5	4	3	2	1	0
PSWH	UF	RBS2	RBS1	RBS0	-	-	-	-
PSWL	S	Z	RSS	AC	IE	P/V	0	CY

各フラグについて、次に示します。

#### (1) キャリー・フラグ (CY)

演算結果のキャリー、ボローを記憶するフラグです。

また、シフト・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、ビット操作命令実行時は、ビット・アキュムレータとして機能します。

CYフラグの状態は、条件付き分岐命令でテストできます。

#### (2) パリティ/オーバーフロー・フラグ (P/V)

P/Vフラグは、演算命令の実行に伴い、次の2種類の動作をします。

P/Vフラグの状態は、条件付き分岐命令でテストできます。

##### パリティ・フラグ動作

論理演算命令、シフト・ローテート命令、CHKL、CHKLA命令の実行の結果、セット (1) されたビット数が偶数のときにはセット (1) されます。奇数のときにはリセット (0) されます。ただし、16ビットのシフト命令の場合、演算結果の下位8ビットのみパリティ・フラグに有効です。

##### オーバーフロー・フラグ動作

算術演算命令の実行の結果、2の補数で表現される数値範囲を越えたときのみセット (1) されます。それ以外のときにはリセット (0) されます。具体的には、MSBからのキャリーとMSBへのキャリーの排他的論理和の結果が、本フラグの内容になります。たとえば、8ビットの算術演算では、2の補数の範囲は80H (-128) ~ 7FH (+127) であり、演算結果がこの範囲以外になったときセット (1) され、範囲内のときはリセット (0) されます。

**例** 8ビット加算命令実行時のオーバフロー・フラグの動作を次に示します。

78H (+120) と69H (+105) の加算を行うと、演算結果がE1H (+225) となり、2の補数の上限を越えるため、P/Vフラグがセット(1)されます。また、2の補数表現で、E1Hは-31になります。

$$\begin{array}{r}
 78H (+120) = \quad 0111 \ 1000 \\
 + ) 69H (+105) = + ) 0110 \ 1001 \\
 \hline
 \quad \quad \quad 0 \ 1110 \ 0001 = -31 \ P/V = 1
 \end{array}$$

CY

また、次のような2つの負数の加算は、演算結果が2の補数の範囲内にあるため、P/Vフラグはリセット(0)されます。

$$\begin{array}{r}
 FBH (-5) = \quad 1111 \ 1011 \\
 + ) F0H (-16) = + ) 1111 \ 0000 \\
 \hline
 \quad \quad \quad 1 \ 1110 \ 1011 = -21 \ P/V = 0
 \end{array}$$

CY

### (3) 割り込み要求許可フラグ (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。

“0”のときは割り込み禁止となり、ノンマスクブル割り込み、およびマスク解除したマクロ・サービスの受け付けのみが可能となります。それ以外は、すべて禁止されます。

“1”のときは割り込み許可状態となり、割り込み要求受け付けの許可は、各割り込み要求に対応する割り込みマスク・フラグおよび各割り込みの優先順位により制御されます。

EI命令実行によりセットされ(1)され、DI命令実行または割り込みの受け付けでリセット(0)されます。

### (4) 補助キャリー・フラグ (AC)

演算の結果、ビット3からのキャリーがあったとき、またはビット3へのボローがあったときにセット(1)されます。それ以外の際にはリセット(0)されます。

ADJBA, ADJBS命令実行時に使用されます。

### (5) レジスタ・セット選択フラグ (RSS)

X, A, C, Bとして機能する汎用レジスタ、およびAX, BCとして機能する汎用レジスタ・ペア(16ビット)を指定するフラグです。

このフラグは、78K/ シリーズとの互換性を保つために用意されているフラグです。78K/ シリーズ用のプログラムを流用する場合を除いて、必ず0にしてください。

**(6) ゼロ・フラグ (Z)**

演算の結果が“0”であることを記憶するフラグです。

演算の結果が“0”のときにセット(1)されます。それ以外のときにはリセット(0)されます。Zフラグの状態は、条件付き分岐命令でテストできます。

**(7) サイン・フラグ (S)**

演算の結果、MSBが“1”であることを記憶するフラグです。

演算の結果、MSBが“1”のときセット(1)されます。“0”のときリセット(0)されます。Sフラグの状態は、条件付き分岐命令でテストできます。

**(8) レジスタ・バンク選択フラグ (RBS0-RBS2)**

8個のレジスタ・バンク(レジスタ・バンク0-レジスタ・バンク7)のうち、1つを選択する3ビットのフラグです(表3-1参照)。

SEL R<sub>n</sub>命令の実行などにより選択されたレジスタ・バンクを示す3ビットの情報が格納されています。

表3-1 レジスタ・バンクの選択

RBS2	RBS1	RBS0	指定レジスタ・バンク
0	0	0	レジスタ・バンク0
0	0	1	レジスタ・バンク1
0	1	0	レジスタ・バンク2
0	1	1	レジスタ・バンク3
1	0	0	レジスタ・バンク4
1	0	1	レジスタ・バンク5
1	1	0	レジスタ・バンク6
1	1	1	レジスタ・バンク7

**(9) ユーザ・フラグ (UF)**

ユーザ・プログラム上でセットおよびリセットし、プログラムの制御に利用できるフラグです。

### 3.1.3 RSSビットの使用方法

基本的にRSSビットは、常時、0に固定して使用してください。

次の説明は、78K<sub>L</sub> シリーズ用のプログラムを流用する場合で、流用するプログラムがRSSビットを1にしている場合のための説明です。RSSビットを0に固定して使用するには読む必要はありません。

RSSビットは、A(R1)、X(R0)、B(R3)、C(R2)、AX(RP0)、BC(RP1)の持っている機能をR4-R7(RP2、RP3)レジスタでも使用できるようにするためのビットです。このビットを有効に使用することで、プログラム・サイズやプログラムの実行に関して効率の良いプログラムを作成することができます。

しかし、不用意に使用すると思わぬ不具合が発生することがあります。したがって、通常は、RSSビットは0にして使用してください。RSSビットを1にして使用するのは、78K<sub>L</sub> シリーズ用のプログラムを流用する場合だけにしてください。

すべてのプログラム中で、RSSビットを0にして使用することにより、プログラム作成およびデバッグ作業の効率が向上します。

RSSビットを1にして使用しているプログラムを流用する場合でも、可能であれば、RSSビットを1にしないプログラムへ修正してから流用することを推奨します。

#### (1) RSSビットの機能

命令のオペレーション一覧(6.2参照)で、オペランド欄にA、X、B、C、AXレジスタが直接記載されている命令で使用するこれらのレジスタ

インプライド・アドレッシングにより、A、AX、B、Cレジスタを使用する命令で、インプライドで指定されるレジスタ

インデクスト・アドレッシング、ベースト・インデクスト・アドレッシングでA、B、Cレジスタを使用する命令で、アドレッシング時に使用するレジスタ

これらの場合に使用するレジスタをRSSビットによって、次のように切り替えます。

RSS = 0の場合

A R1, X R0, B R3, C R2, AX RP0, BC RP1

RSS = 1の場合

A R5, X R4, B R7, C R6, AX RP2, BC RP3

前記以外で使用するレジスタは、RSSビットの内容に関わらず常に同一のレジスタになります。NEC製のアセンブラ(RA78K4)では、A、X、B、C、AX、BCレジスタについて、この名前で記述されている場合にどちらのレジスタ用の命令コードを生成するかは、アセンブラのRSS疑似命令で決定されます。

RSSビットをセット、リセットする場合には、その命令の直前(または直後)に、必ずRSS疑似命令を記述してください(下記例参照)。

#### プログラム例

RSS = 0にする場合

RSS 0 ; RSS疑似命令

CLR1 PSWL.5

MOV B, A ; この記述は, “ MOV R3, R1 ” に該当します。

RSS = 1にする場合

RSS 1 ; RSS疑似命令

SET1 PSWL.5

MOV B, A ; この記述は, “ MOV R7, R5 ” に該当します。

### (2) RA78K4での命令コードの生成方法

RA78K4では, 命令のオペレーション一覧のオペランド欄にAまたはAXが直接記載されている命令と同機能の命令がある場合, オペランド欄にAまたはAXが直接記載されている命令コードを優先して生成します。

**例** MOV A, r命令でrをBとした場合と, MOV r, r 命令でrをA, r をBとした場合では同一の機能です。また, アセンブラのソース・プログラム上では同一の記述 (MOV A, B) となります。この場合, RA78K4は, MOV A, r命令に相当するコードを生成します。

**備考** この命令で実際に使用されるレジスタは, PSW中のRSSビットの内容によりプログラムの実行時に決まります。RSS = 0のときはR1またはRP0が, RSS = 1のときにはR5またはRP2が使用されます。

オペランド欄にr, r , rp, rp が指定されている命令で, A, X, B, C, AX, BCが記述された場合, RA78K4のRSS疑似命令のオペランドによって, A, X, B, C, AX, BC命令はそれぞれ次のレジスタを指定する命令コードを生成します。

レジスタ	RSS 0	RSS 1
A	R1	R5
X	R0	R4
B	R3	R7
C	R2	R6
AX	RP0	RP2
BC	RP1	RP3

オペランド欄のr, r , rp, rp にR0-R7, RP0-RP4を記述した場合, その指定どおりの命令コードを出力します (オペランド欄にA, AXが直接記載されている命令コードは出力しません)。

インデクスト・アドレッシング, ベースト・インデクスト・アドレッシングで使用するA, B, Cレジスタは, R1, R3, R2やR5, R7, R6といった記述はできません。

### (3) 使用上の注意

RSSビットを切り替えることで, 2組のレジスタ・セットを持つのと同様な効果が得られます。しかし, 次のような点に注意する必要があります。RSS = 1として使用しなければならない場合は, これらの欠点

を十分に考慮のうえ、プログラムを作成してください。

- (a) プログラムの静的な記述と、プログラム実行時の動的なRSSビットの変化を常に一致するように、十分に注意をしてプログラムを記述しなければなりません。

たとえば、RA78K4でMOV A, B命令をアセンブルした場合、MOV A, rのコードが生成されます。このとき、実際に使用されるレジスタは、ソース・プログラム中でMOV A, B命令の直前に記述されたRSS疑似命令と、プログラム実行時のPSW中のRSSビットによって次のようになります。

		RSS疑似命令のオペランド	
		0	1
PSW中のRSSビット	0	MOV R1, R3	MOV R1, R7
	1	MOV R5, R3	MOV R5, R7

- (b) RSS = 1にしたプログラムは、コンテキスト・スイッチング機能を使用しているプログラムでは使用できないので、プログラムの流用性が悪くなります。
- (c) PSW中のRSSビットを“1”にしている部分が一箇所以上あるプログラムで、割り込みを使用する場合には、割り込み処理プログラムの先頭で、PSW中のRSSビットを“0”または“1”に設定し、ソース・プログラム上でそれに対応したRSS疑似命令を記述しておく必要があります。これをしないと、「ときどき実行結果がおかしくなる」という現象が発生する場合があります。たとえば、次のような割り込み処理プログラムがあるとします。

```
INT :
    PUSH AX
    MOV A, #byte
    ADD !!addr24, A
    POP AX
    RETI
```

このプログラムでは、PUSH AX, MOV A, #byte, POP AX命令では、直前に記述したRSS疑似命令によりアセンブル時に決定されるレジスタがAXおよびAレジスタとして使用されます。しかし、ADD !!addr24, A命令では、割り込まれたプログラムがPSW中のRSSビットを何にしていたかによって、Aレジスタとして使用するレジスタが決定されます。したがって、!!addr24で指定されるメモリには、期待どおりの値が格納される場合と、予期しない値が格納される場合が発生します。

この例では、割り込み処理プログラムの実行結果が異常になるだけですが、たとえば、ADD命令のオペランドが逆の場合（ADD A, !!addr24）には、割り込まれたプログラムで使用していたレジスタが破壊される場合があります。

このような不具合は非定常的に現象が発生するため、デバッグ時に原因を探するのは非常に難しいものになります。

- (d) 同一の名称で異なるレジスタを使用するため、プログラムの可読性が悪くなり、デバッグ作業が難しくなります。

### 3.1.4 スタック・ポインタ (SP)

スタック領域 (LIFO形式: 000000H-FFFFFFH) の先頭アドレスを保持する24ビット・レジスタです (図3-3参照)。サブルーチン処理や割り込み処理時にスタック領域をアドレスするために使用されます。

SPの内容は、スタック領域への書き込み前にデクリメントされ、スタック領域からの読み出し後にインクリメントされます (図3-4, 図3-5参照)。

SPは、専用命令によりアクセスします。

SPの内容は、 $\overline{\text{RESET}}$ 入力により不定になりますので、リセット解除直後に (サブルーチン・コールや割り込みを受け付ける前に) 必ず初期化プログラムによりSPをイニシャライズしてください。

なお、SPの上位側の数ビットを0に固定しなければならない製品があります。詳細については、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。

例 SPのイニシャライズ

```
MOVG SP, #0FEE0H; SP      0FEE0H ( FEDFHから使用の場合 )
```

図3-3 スタック・ポインタ (SP) の構成

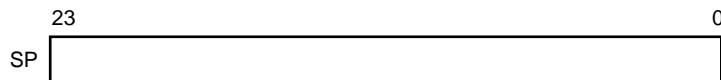


図3 - 4 スタック領域へ退避されるデータ

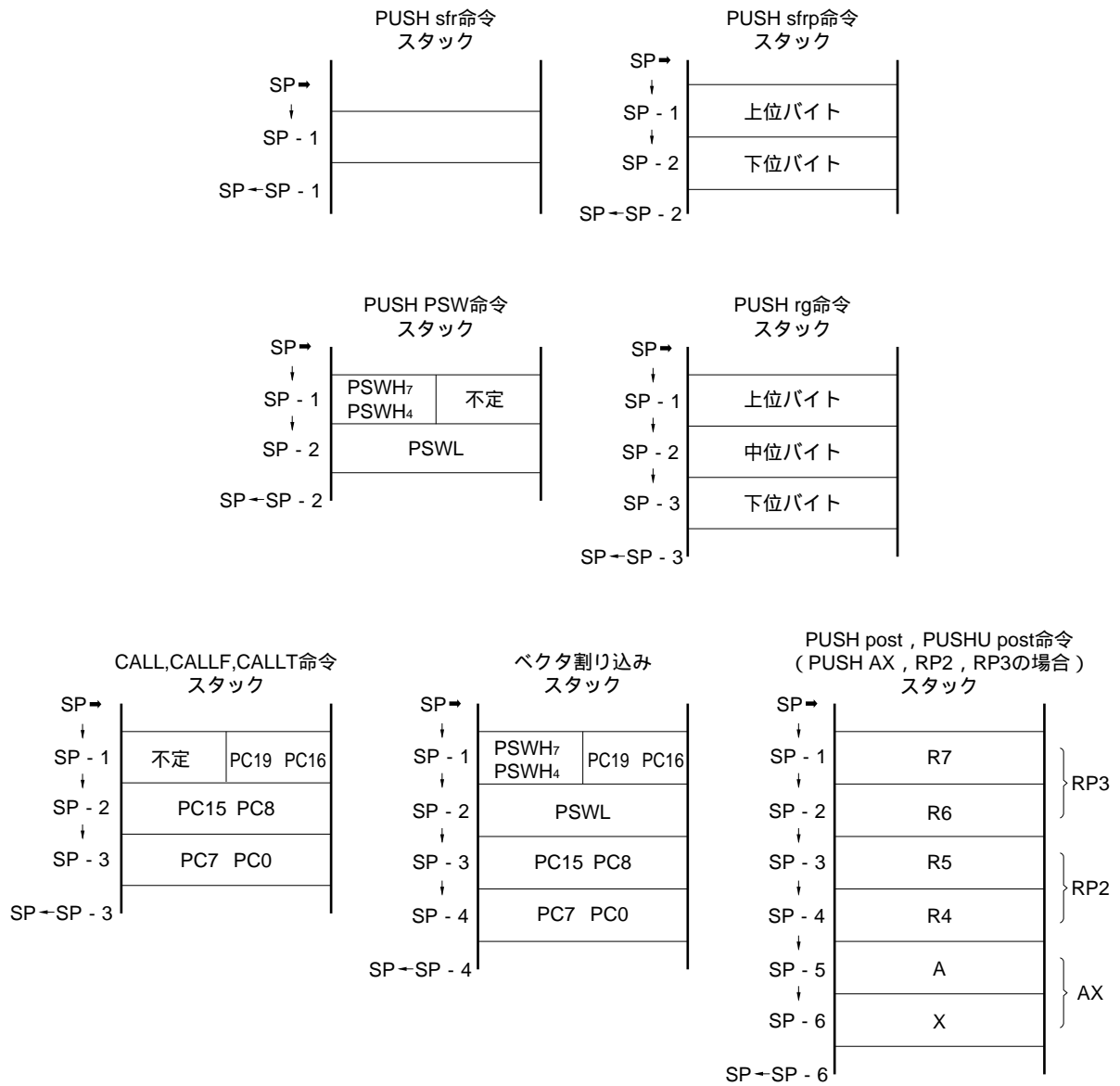
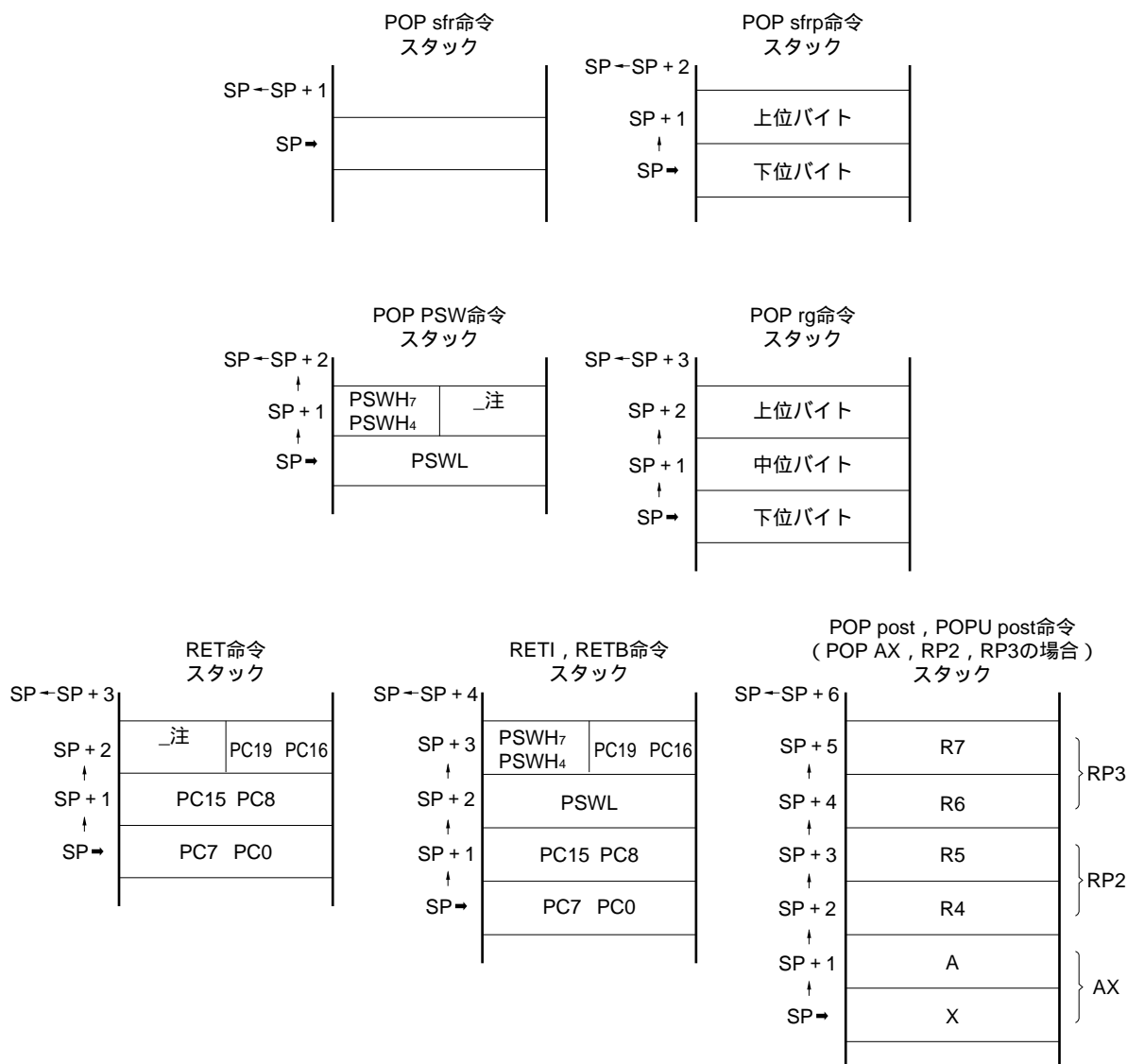


図3 - 5 スタック領域から復帰されるデータ



注 この4ビットのデータは無視されます。

注意1. スタック・アドレッシングでは、16 Mバイトすべての空間でアクセス可能ですが、SFR領域および内部ROM領域にスタック・エリアを確保することはできません。

2. SPは、RESET入力により不定になります。また、SPが不定の状態でもノンマスカブル割り込みは受け付け可能です。したがって、リセット解除直後のSPが不定な状態でノンマスカブル割り込みの要求が発生すると、予期しない動作を行う場合があります。この危険を回避するために、リセット解除後のプログラムは、必ず次のようにしてください。

なお、μ PD784915サブシリーズは、LOCATION 0H命令固定です。

```

RSTVCT  CSEG  AT 0
        DW   RSTSTRT
        }
INITSEG  CSEG  BASE
RSTSTRT : LOCATION 0H ; or LOCATION 0FH
MOVG    SP, #STKBGN
    
```

## 3.2 汎用レジスタ

### 3.2.1 構成

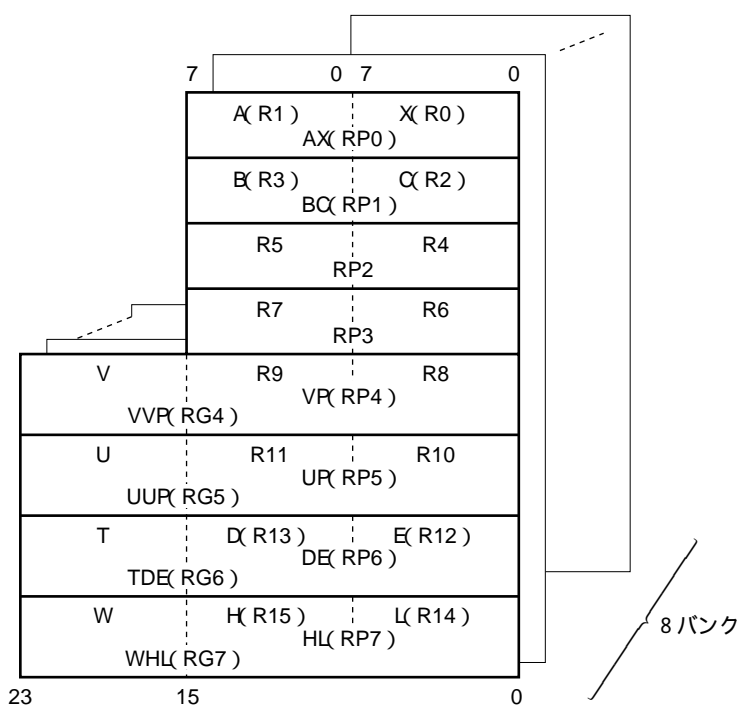
8ビットの汎用レジスタ16本で構成されています。また、8ビット汎用レジスタ2本を組み合わせ、16ビット汎用レジスタとして使用することもできます。さらに、16ビット汎用レジスタのうち4本は、アドレス拡張用の8ビット・レジスタと組み合わせ、24ビット・アドレス指定用レジスタとして使用することができます。

アドレス拡張用のV, U, T, Wレジスタを除いた汎用レジスタは、内部RAMにマッピングされています。

これらのレジスタ・セットは、8バンク用意されており、ソフトウェアまたはコンテキスト・スイッチング機能により切り替えて使用することができます。

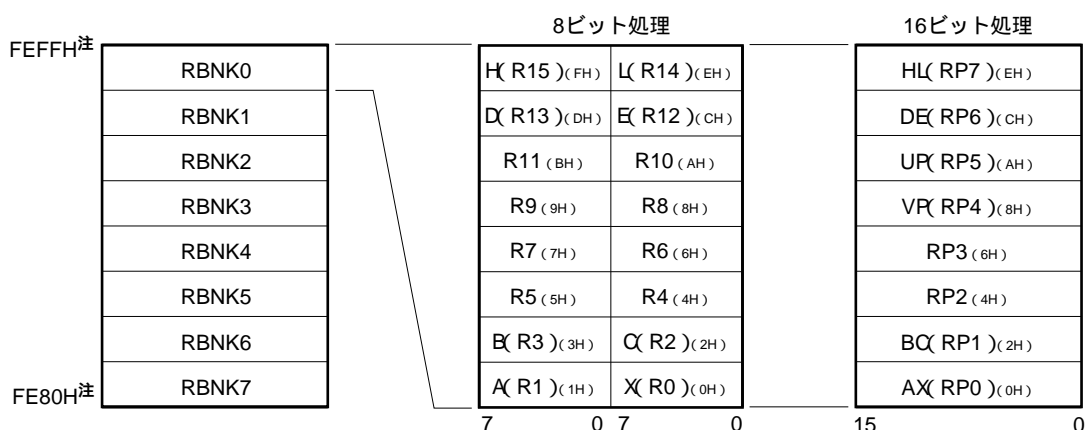
RESET入力により、レジスタ・バンク0が選択されます。また、実行中のプログラムで使用しているレジスタ・バンクは、PSW内のレジスタ・バンク選択フラグ (RBS0, RBS1, RBS2) を読み出すことによって確認できます。

図3 - 6 汎用レジスタ構成



備考 ( )内は絶対名称です。

図3 - 7 汎用レジスタのアドレス



**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時のアドレスは、本値に0F0000Hを加えた値です。

なお、μPD784915サブシリーズは、LOCATION 0H命令固定です。

**注意** R4, R5, R6, R7, RP2, RP3は、PSWのRSSビットを1にすることで、それぞれ、X, A, C, B, AX, BCレジスタとして使用することができますが、この機能を使用するのは78K<sup>II</sup>シリーズ用のプログラムを流用する場合だけにしてください。

**備考** レジスタ・バンクを変更する場合で、元のレジスタ・バンクに戻す必要がある場合は、PUSH PSW命令でPSWをスタックへ退避してからSEL RBN命令を実行してください。元に戻すときは、スタックの位置に変化がなければPOP PSW命令で戻せます。

なお、ベクタ割り込み処理プログラムなどでレジスタ・バンクを変更する場合には、PSWは割り込みの受け付け時に自動的にスタックへ退避され、RETI, RETB命令で復帰されますので、割り込み処理プログラムで使用するレジスタ・バンクが1つの場合には、単にSEL RBN命令を実行するだけで、PUSH PSWやPOP PSW命令を実行する必要はありません。

**例1** . レジスタ・バンク2を指定する場合

```

:
PUSH PSW
SEL RB2 } レジスタ・バンク2で動作
:
POP PSW }
:
元レジスタ・バンクで動作
    
```

**例2** . ベクタ割り込み処理プログラムでレジスタ・バンクを指定する場合

```

INT :
SEL RB5 } レジスタ・バンク5で動作
:
RETI } 元レジスタ・バンクに自動復帰
    
```

### 3.2.2 機能

汎用レジスタは、それぞれ8ビット単位で操作できるほか、2個の8ビット・レジスタがペアとなって16ビット単位で操作することもできます。また、16ビット・レジスタのうち4本は、アドレス拡張用の8ビット・レジスタと組み合わせて24ビット単位で操作できます。

また、各レジスタは、演算結果の一時保管や、レジスタ間演算命令のオペランドとして汎用的に使用することが可能です。

0FE80H-0FEFFH (LOCATION 0H命令実行時、LOCATION 0FH命令実行時は0FFE80H-0FEFFH) の領域は、汎用レジスタ・エリアとして使用する、しないにかかわらず、通常のデータ・メモリとしてアドレス指定し、アクセスすることができます。

78K<sup>+</sup> シリーズでは、8つのレジスタ・バンクを有しているため、通常の処理と割り込み時の処理でレジスタ・バンクを使い分けることにより、効率のよいプログラムを作成できます。

各レジスタは、それぞれ次に示す固有機能を持っています。

#### A (R1) :

8ビット・データの転送や演算処理の中心となるレジスタです。8ビット・データに対するすべてのアドレッシング・モードと組み合わせて使用できます。

ビット・データの格納にも使用できます。

インデクスト・アドレッシング、ベースト・インデクスト・アドレッシング時のオフセット値を格納するレジスタとして使用できます。

#### X (R0) :

ビット・データの格納に使用できます。

#### AX (RP0) :

16ビット・データの転送や演算処理の中心となるレジスタです。16ビット・データに対するすべてのアドレッシング・モードと組み合わせて使用できます。

#### AXDE :

DIVUX, MACW, MACSW命令実行時に、32ビット・データの格納用として使用されます。

#### B (R3) :

ループ・カウンタとしての機能を持っており、DBNZ命令で使用できます。

インデクスト・アドレッシング、ベースト・インデクスト・アドレッシング時のオフセット値を格納するレジスタとして使用できます。

MACW, MACSW命令のデータ・ポインタとして使用します。

#### C (R2) :

ループ・カウンタとしての機能を持っており、DBNZ命令で使用できます。

ベースト・インデクスト・アドレッシング時のオフセット値を格納するレジスタとして使用できます。

ストリング命令、SACW命令のカウンタとして使用します。

MACW, MACSW命令のデータ・ポインタとして使用します。

RP2 :

コンテキスト・スイッチング使用時に、プログラム・カウンタ (PC) の下位16ビットを退避するために使用します。

RP3 :

コンテキスト・スイッチング使用時に、プログラム・カウンタ (PC) の上位4ビットおよびプログラム・ステータス・ワード (PSW) (PSWHのビット0-3を除く) を退避するために使用します。

VVP (RG4) :

ポインタとしての機能を持っており、レジスタ・インダイレクト・アドレッシング、ベースト・アドレッシング、ベースト・インデクスト・アドレッシング時に、ベース・アドレスを指定するレジスタとして動作します。

UUP (RG5) :

ユーザ・スタック・ポインタとしての機能を持っており、PUSHU, POPU命令により、システム・スタックとは別のスタックを実現することができます。

ポインタとしての機能を持っており、レジスタ・インダイレクト・アドレッシング、ベースト・アドレッシング時に、ベース・アドレスを指定するレジスタとして動作します。

DE (RP6) , HL (RP7) :

インデクスト・アドレッシング、ベースト・インデクスト・アドレッシング時には、オフセット値を格納するレジスタとして動作します。

TDE (RG6) :

ポインタとしての機能を持っており、レジスタ・インダイレクト・アドレッシング、ベースト・アドレッシング時に、ベース・アドレスを指定するレジスタとして動作します。

ストリング命令, SACW命令のポインタとして動作します。

WHL (RG7) :

24ビット・データの転送や演算処理の中心となるレジスタです。

ポインタとしての機能を持っており、レジスタ・インダイレクト・アドレッシング、ベースト・アドレッシング時に、ベース・アドレスを指定するレジスタとして動作します。

ストリング命令, SACW命令のポインタとして動作します。

各レジスタは、それぞれの固有機能を重視した機能名称 (X, A, C, B, E, D, L, H, AX, BC, VP, UP, DE, HL, VVP, UUP, TDE, WHL) のほか、絶対名称 (R0-R15, RP0-RP7, RG4-RG7) でも記述することができます。対応は、表3 - 2を参照してください。

表3 - 2 機能名称-絶対名称の対応

## (a) 8ビット・レジスタ

絶対名称	機能名称	
	RSS = 0	RSS = 1 <sup>注</sup>
R0	X	
R1	A	
R2	C	
R3	B	
R4		X
R5		A
R6		C
R7		B
R8		
R9		
R10		
R11		
R12	E	E
R13	D	D
R14	L	L
R15	H	H

## (b) 16ビット・レジスタ

絶対名称	機能名称	
	RSS = 0	RSS = 1 <sup>注</sup>
RP0	AX	
RP1	BC	
RP2		AX
RP3		BC
RP4	VP	VP
RP5	UP	UP
RP6	DE	DE
RP7	HL	HL

## (c) 24ビット・レジスタ

絶対名称	機能名称
RG4	VVP
RG5	UUP
RG6	TDE
RG7	WHL

注 RSS = 1として使用するのは、78K/ シリーズ用のプログラムを流用する場合だけにしてください。

備考 R8-R11には機能名称はありません。

### 3.3 特殊機能レジスタ (SFR)

内蔵周辺ハードウェアのモード・レジスタ、コントロール・レジスタなどの特別な機能が割り付けられたレジスタで、0FF00H-0FFFFH<sup>※</sup>の256バイトの空間にマッピングされています。特殊機能レジスタについては、各製品の個別資料を参照してください。

**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時は0FFF00H-0FFFFFFH  
なお、 $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。

**注意** この領域内で、SFRのマッピングさ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となることがあります。デッドロック状態は、リセット入力によってのみ解除されます。

## 第4章 割り込み機能

割り込み要求に対する処理として、表4 - 1のような3つの処理をプログラムできます。

また、マスク可能なベクタ割り込みについては、4レベルの優先順位を持った多重処理制御を容易に行うことができます。

表4 - 1 割り込み要求の処理

処理モード	処理の主体	処 理	PC, PSWの内容
ベクタ割り込み	ソフトウェア	処理ルーチンへ分岐して実行（処理内容は任意）	スタックへの退避， 復帰を行う
コンテキスト・スイッチング		レジスタ・バンクで自動的に切り替えて，処理ルーチンへ分岐して実行（処理内容は任意）	レジスタ・バンク中の 固定エリアへの退 避，復帰を行う
マクロ・サービス	ファームウェア	メモリ-I/O間のデータ転送などを実行（処理内容は固定）	保持

**備考** 詳細については、各製品のユーザーズ・マニュアル ハードウェア編で確認してください。

## 4.1 割り込み要求の種類

割り込み要求の種類には、次の3種類があります。

- ソフトウェア割り込み要求
- ノンマスクブル割り込み要求
- マスクブル割り込み要求

### 4.1.1 ソフトウェア割り込み要求

ソフトウェアによる割り込み要求は、BRK命令、BRKCS Rn命令の実行およびMOV WDM, #byte命令、MOV STBC, #byte命令、LOCATION命令のオペランドが異常な場合（オペランド・エラー割り込み）に発生します。ソフトウェアによる割り込み要求は、割り込み禁止（DI）状態でも受け付けられます。また、割り込み優先順位制御の対象になりません。したがって、ソフトウェアによる割り込みの要求が発生すると、無条件に割り込み処理プログラムに分岐します。

BRK命令から復帰する際は、RETB命令を実行します。

BRKCS Rn命令のサービス・ルーチンから復帰する際は、RETCB !addr16命令を実行します。

オペランド・エラー割り込みは、オペランドに異常があった場合に発生する割り込みですので、リセット解除による初期化プログラムなどへ必要な処理を行ったあとに分岐するなどの処理が必要になります。

### 4.1.2 ノンマスクブル割り込み要求

ノンマスクブル割り込み要求は、NMI端子に有効エッジが入力されたとき、またはウォッチドッグ・タイマがオーバーフローしたときに発生します。なお、NMI端子、ウォッチドッグ・タイマについては、製品ごとにその機能の有無が異なります。詳細は、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。

ノンマスクブル割り込み要求は、割り込み禁止（DI）状態であっても無条件に受け付けられます。また、割り込み優先順位制御の対象にならず、すべての割り込みに対して最優先の割り込み要求です。

### 4.1.3 マスクブル割り込み要求

マスクブル割り込み要求は、割り込み制御レジスタの設定によってマスク制御を受ける割り込み要求です。また、PSWのIEフラグによって、マスクブル割り込み全体に対して受け付けの許可/禁止の指定ができます。

マスクブル割り込み要求は、同一優先順位を持つ複数の割り込み要求が同時に発生している場合の優先順位が決められています（デフォルト優先順位）。また、割り込み制御レジスタの指定によって、割り込み優先順位を4レベルに分けて多重処理を行うことができます。ただし、マクロ・サービスは、優先順位制御やIEフラグとは無関係に受け付けられます。

## 4.2 割り込み処理モード

### 4.2.1 ベクタ割り込み

割り込み要因に対応するベクタ・テーブル・アドレスのメモリ内容を、分岐先のアドレスとして処理ルーチンへ分岐します。

CPUが割り込み処理を行うため、次のような動作が起こります。

分岐時：CPUの状態（PC, PSWの内容）をスタックへ退避  
 復帰時： " " をスタックから復帰

なお、処理ルーチンからメイン・ルーチンへの復帰は、RETI命令（BRK命令およびオペランド・エラー割り込みの場合はRETB命令）で行います。

また、分岐先のアドレスは、0000H-FFFFHのベース領域に限定されます。

ベクタ・テーブルの詳細については、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。

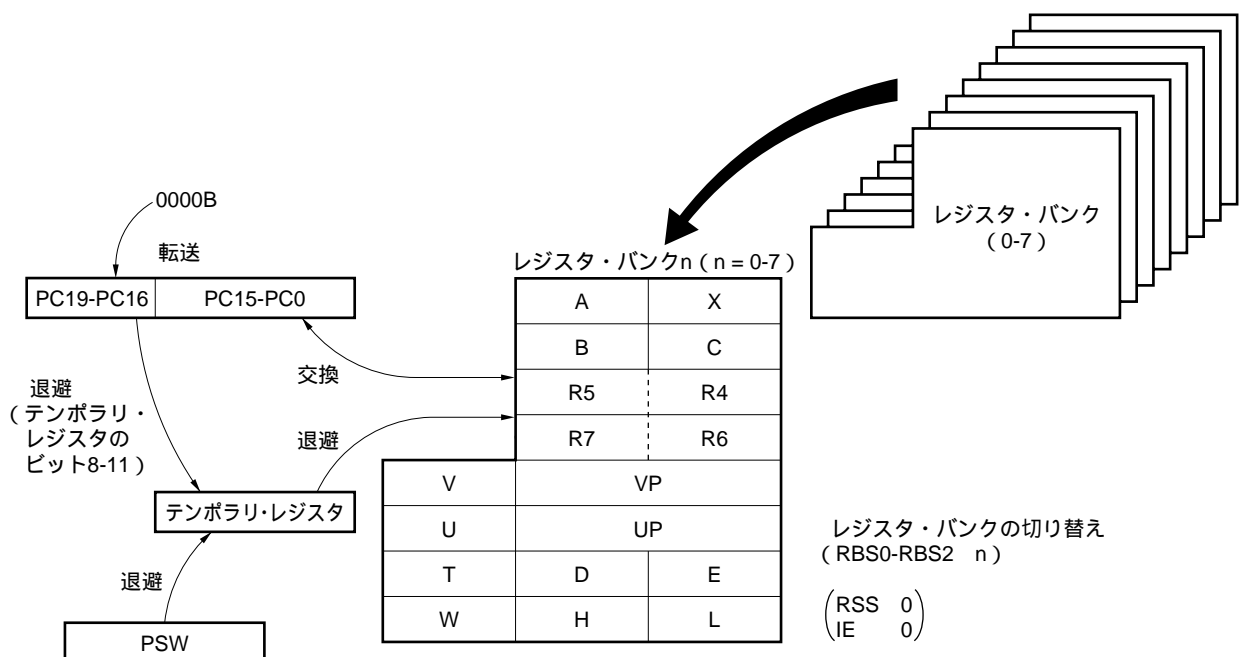
### 4.2.2 コンテキスト・スイッチング

割り込み要求の発生またはBRKCS RBn命令の実行により、ハードウェア的に所定のレジスタ・バンクを選択します。レジスタ・バンク内にあらかじめストアしておいたベクタ・アドレスへ分岐すると同時に、現在のプログラム・カウンタ（PC）、プログラム・ステータス・ワード（PSW）の内容を、レジスタ・バンク内にスタックする機能です。

なお、処理ルーチンからの復帰は、RETCS !addr16命令（BRKCS RBn命令の場合はRETCSB !addr16命令）で行います。

また、分岐先のアドレスは、0000H-FFFFHのベース領域に限定されます。

図4-1 割り込み要求の発生によるコンテキスト・スイッチング動作



### 4.2.3 マクロ・サービス機能

マクロ・サービスは、割り込みが受け付けられるとCPUの実行を一時中断し、ファームウェアで設定されているサービスを実行します。また、マクロ・サービスは、CPUを介さずに行われるため、PC、PSWなどのCPUステータスを退避/復帰する必要がありません。したがって、CPUのサービス時間を向上させる大きな効果があります。

マクロ・サービスの詳細については、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。

## 第5章 アドレッシング

### 5.1 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、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）されますが、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。

命令アドレスのアドレッシングには、次のものがあります。

- （8ビット/16ビット）レラティブ・アドレッシング
- （11ビット/16ビット/20ビット）イミディエト・アドレッシング
- テーブル・インダイレクト・アドレッシング
- 16ビット・レジスタ・アドレッシング
- 20ビット・レジスタ・アドレッシング
- 16ビット・レジスタ・インダイレクト・アドレッシング
- 20ビット・レジスタ・インダイレクト・アドレッシング

各アドレッシングの詳細については、次節以降を参照してください。

### 5.1.1 レラティブ・アドレッシング

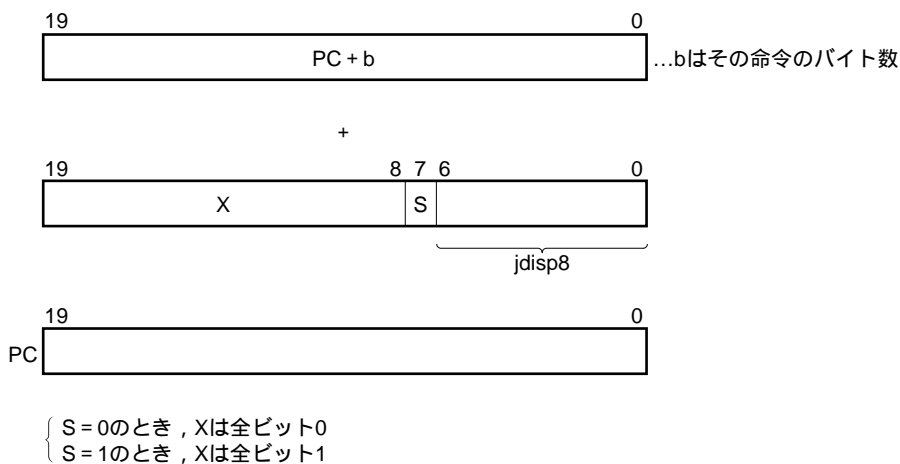
**【機能】**

次に続く命令の先頭アドレスに、命令コードの8ビットまたは16ビットのイミディエト・データ（ディスプレイメント値：jdisp8, jdisp16）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, -32768～+32767）として扱われ、最上位ビットが符号ビットとなります。

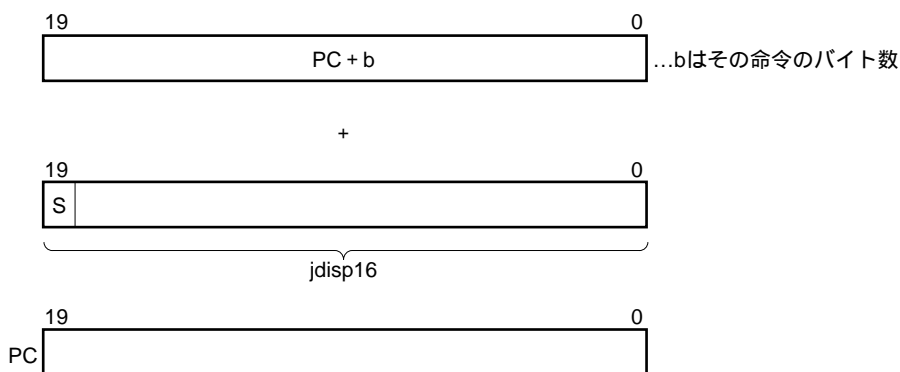
CALL \$!addr20, BR \$!addr20, BR \$addr20命令、および条件付き分岐命令を実行する際に行われます（条件付き分岐命令は、8ビット・イミディエト・データだけが使用可能です）。

**【図解】**

**8ビット・レラティブ・アドレッシング**



**16ビット・レラティブ・アドレッシング**



### 5.1.2 イミディエト・アドレッシング

**【機能】**

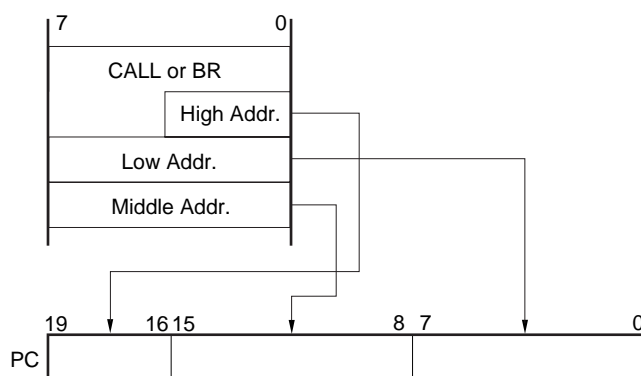
命令語中のイミディエト・データがプログラム・カウンタ（PC）に転送され、分岐します。

CALL !addr20, BR !addr20, CALL !addr16, BR !addr16, CALLF !addr11命令を実行する際に行われます。

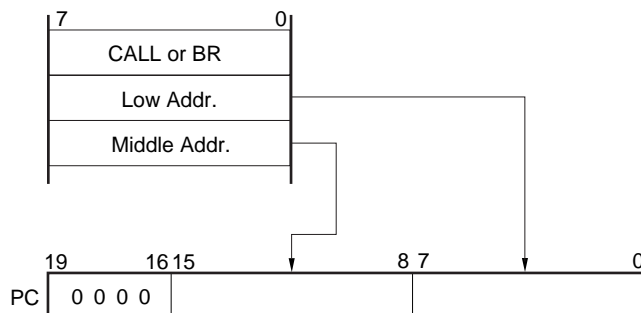
なお、CALL !addr16, BR !addr16命令（16ビット・イミディエト・アドレッシング）の場合は、上位の4ビット・アドレスは、0000に固定で、ベース領域への分岐になります。また、CALLF !addr11命令の場合は、上位の9ビット・アドレスは000000001に固定です。

**【図解】**

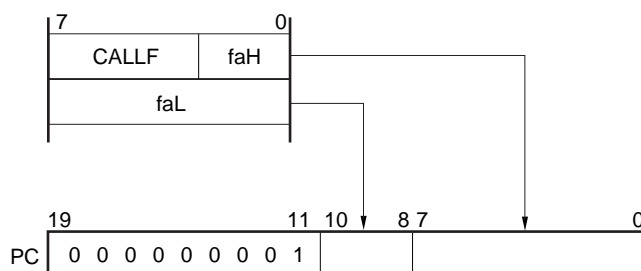
**20ビット・イミディエト・アドレッシング**



**16ビット・イミディエト・アドレッシング**



**11ビット・イミディエト・アドレッシング**



**【注 意】**

BR !addr16命令は、分岐先が限定されるため、78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合にだけ使用するようになっています。

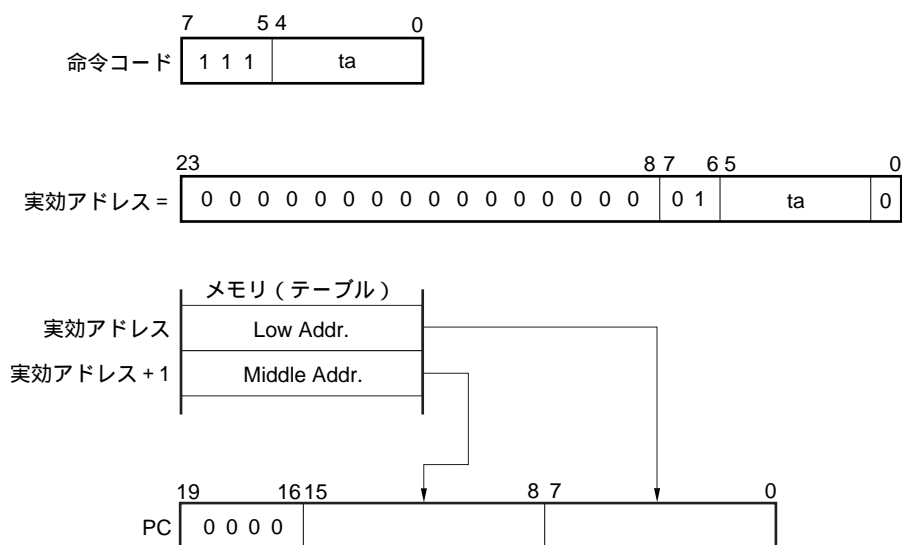
### 5.1.3 テーブル・インダイレクト・アドレッシング

**【機能】**

命令コードの下位5ビットのイミディエト・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）の下位16ビットに，上位4ビットには0000が転送されて分岐します（分岐先のアドレスは，ベース領域に限定されます）。

CALLT [ addr5 ] 命令を実行する際に行われます。

**【図解】**



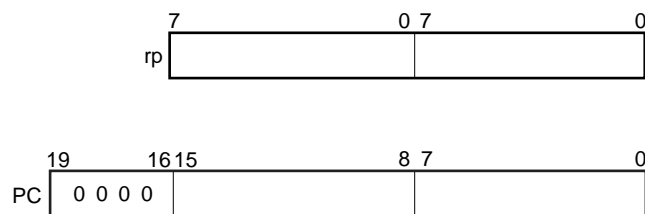
## 5.1.4 16ビット・レジスタ・アドレッシング

## 【機能】

命令語によって指定されるレジスタrp (RP0-RP7) の内容が、プログラム・カウンタ (PC) の下位16ビットに、上位4ビットには0000が転送されて分岐します (分岐先のアドレスは、ベース領域に限定されます)。

BR rp, CALL rpを実行する際に行われます。

## 【図解】



## 【注意】

BR rp命令は、分岐先が限定されるため、78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合にだけ使用するようにしてください。

また、rpとしてAX, BCを記述した場合、NEC製アセンブラRA78K4では直前に記述したRSS疑似命令により生成されるオブジェクト・コードが変わります。RSS疑似命令で1を指定するのは、78K/ シリーズ用のプログラムを流用する場合だけにしてください。(3.1.3 RSSビットの使用法参照)。

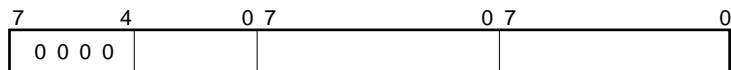
## 5.1.5 20ビット・レジスタ・アドレッシング

## 【機能】

命令語によって指定されるレジスタrg (RG4-RG7) の内容が、プログラム・カウンタ (PC) に転送されて分岐します。なお、rgの上位4ビットは0000にしてください。

BR rg, CALL rg命令を実行する際に行われます。

## 【図解】



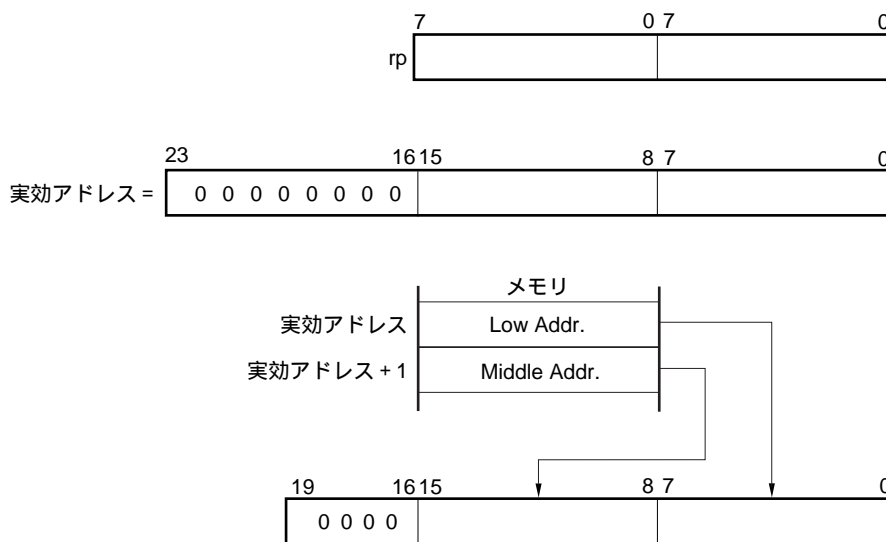
### 5.1.6 16ビット・レジスタ・インダイレクト・アドレッシング

**【機能】**

命令語によって指定されるレジスタrp ( RP0-RP7 ) の内容によりアドレスされるメモリの連続する2バイト・データが、プログラム・カウンタ ( PC ) の下位16ビットに、上位4ビットには0000が転送されて分岐します ( 分岐先のアドレスは、ベース領域に限定されます ) 。

BR [ rp ] , CALL [ rp ] 命令を実行する際に行われます。

**【図解】**



**【注意】**

BR [ rp ] 命令は、分岐アドレスを格納するアドレスおよび分岐先が限定されるため、78K<sub>1</sub> シリーズ用のプログラムを流用する場合にだけ使用するようになっています。

また、rpとしてAX, BCを記述した場合、NEC製アセンブラRA78K4では直前に記述したRSS疑似命令により生成されるオブジェクト・コードが変わります。RSS疑似命令で1を指定するのは、78K<sub>1</sub> シリーズ用のプログラムを流用する場合だけにしてください ( 3.1.3 RSSビットの使用法参照 ) 。

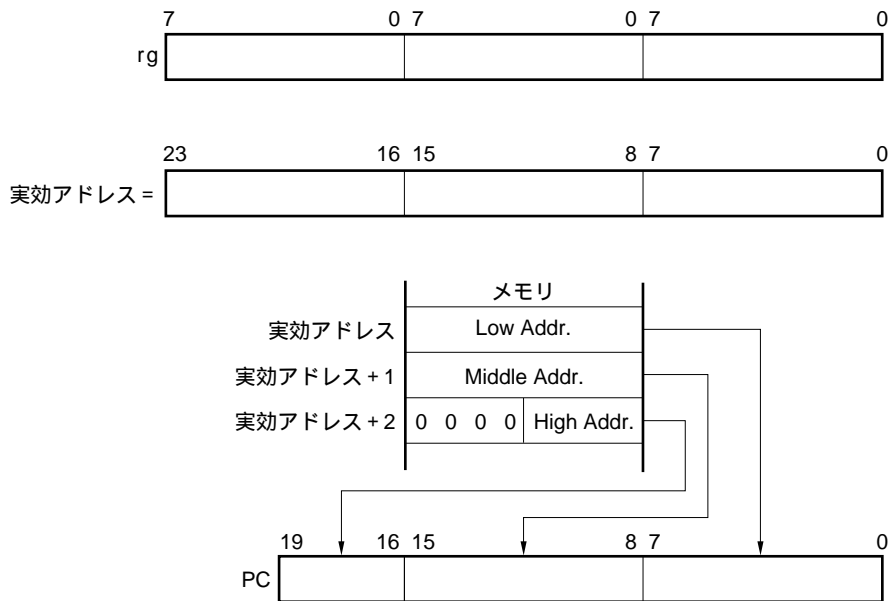
### 5.1.7 20ビット・レジスタ・インダイレクト・アドレッシング

**【機能】**

命令語によって指定されるレジスタrg ( RP0-RP7 ) の内容によりアドレスされるメモリの連続する3バイト・データが、プログラム・カウンタ ( PC ) に転送されて分岐します。なお、メモリに格納している3バイト・データの上位4ビットは、0000にしてください。

BR [ rg ] , CALL [ rg ] 命令を実行する際に行われます。

**【図解】**



## 5.2 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、次に示す方法があります。

インプライド・アドレッシング

レジスタ・アドレッシング

イミーディエト・アドレッシング

8ビット・ダイレクト・アドレッシング

16ビット・ダイレクト・アドレッシング

24ビット・ダイレクト・アドレッシング

ショート・ダイレクト・アドレッシング

特殊機能レジスタ（SFR）アドレッシング

ショート・ダイレクト16ビット・メモリ・インダイレクト・アドレッシング

ショート・ダイレクト24ビット・メモリ・インダイレクト・アドレッシング

スタック・アドレッシング

24ビット・レジスタ・インダイレクト・アドレッシング

（オート・インクリメント/デクリメント付き24ビット・レジスタ・インダイレクト・アドレッシングを含む）

16ビット・レジスタ・インダイレクト・アドレッシング

ベースト・アドレッシング

インデクスト・アドレッシング

ベースト・インデクスト・アドレッシング

各アドレッシングの詳細については、次節以降を参照してください。

## 5.2.1 インプライド・アドレッシング

## 【機能】

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中のレジスタを、自動的にアドレスするアドレッシングです。

78K/ 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している命令は次のとおりです。

これらの命令で使用されるA, AX, C, Bレジスタは、PSW中のRSSビットの影響を受けます。RSS = 0の場合、A, AX, C, Bレジスタは、それぞれR1, RP0, R2, R3がアクセスの対象となり、RSS = 1の場合は、R5, RP2, R6, R7になります。なお、RSS = 1として使用するものは、78K/ シリーズ用のプログラムを流用する場合だけにしてください (3.1.3 RSSビットの使用方法参照)。

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
MULUW, MULW	被乗数および、積の上位16ビット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
DIVUX	被除数および商を格納するレジスタとしてAXDEレジスタ
MACW, MACSW	積和演算結果を格納するレジスタとしてAXDEレジスタ、データを指定するポインタ用のレジスタとしてB, C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
CVTBW	符号拡張を行う前のデータを格納するレジスタとしてAレジスタ、符号拡張の結果を格納するレジスタとしてAXレジスタ
CHKLA	端子レベルとポートの出力ラッチの比較結果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 (下位4ビットのみ使用)
SACW, スtring命令	データ数のカウンタとしてCレジスタ

## 【オペランド形式】

命令によって自動的に使用されるため、特定のオペランド形式を持ちません。

## 【記述例】

**MULU r**; 8ビット×8ビット乗算命令において、Aレジスタとrレジスタの積をAXレジスタに格納する。

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。

## 5.2.2 レジスタ・アドレッシング

## 【機能】

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の、命令語中のレジスタ指定コードにより指定される汎用レジスタを、オペランドとしてアクセスするアドレッシングです。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ます。

## 【オペランド形式】

次に示すオペランドを持つ命令を実行する際に行われます。

表現形式	記述方法
A	A
C	C
X	X
B	B
r	X ( R0 ) , A ( R1 ) , C ( R2 ) , B ( R3 ) , R4, R5, R6, R7, R8, R9, R10, R11, E ( R12 ) , D ( R13 ) , L ( R14 ) , H ( R15 )
r1	X ( R0 ) , A ( R1 ) , C ( R2 ) , B ( R3 ) , R4, R5, R6, R7
r2	R8, R9, R10, R11, E ( R12 ) , D ( R13 ) , L ( R14 ) , H ( R15 )
r3	V, U, T, W
AX	AX
rp	AX ( RP0 ) , BC ( RP1 ) , RP2, RP3, VP ( RP4 ) , UP ( RP5 ) , DE ( RP6 ) , HL ( RP7 )
rp1	AX ( RP0 ) , BC ( RP1 ) , RP2, RP3
rp2	VP ( RP4 ) , UP ( RP5 ) , DE ( RP6 ) , HL ( RP7 )
WHL	WHL
rg	VVP ( RG4 ) , UUP ( RG5 ) , TDE ( RG6 ) , WHL ( RP7 )

備考1. ( )内は、絶対名称です。

- レジスタ・アドレッシングのオペランドとして直接A, X, AX, B, Cが指定されている命令 (例 ADDW AX, #wordなど) では、A, X, AX, B, Cレジスタとして使用するレジスタは、実行時のPSW中にあるRSSビットによって決定されます。PSW中のRSSビットを“1”として使用するのは、78K/ シリーズ用のプログラムを流用する場合だけにしてください (3.1.3 RSSビットの使用方法参照)。
- レジスタ・アドレッシングのオペランドとしてr, r1, rp, rp1が指定されている命令で、オペランドとしてA, X, B, C, AX, BCを記述した場合に、NEC製アセンブラ RA78K4では直前に記述したRSS疑似命令によって生成されるオブジェクト・コードが変わります。RSS疑似命令のオペランドに“1”を指定するのは、78K/ シリーズ用のプログラムを流用する場合だけにしてください (3.1.3 RSSビットの使用方法参照)。

**【記述例1】**

一般例

**MOV A, r**

具体例

**MOV A, C ; rとしてCレジスタを選択する場合**

**【記述例2】**

一般例

**INCW rp**

具体例

**INCW DE ; rpとしてDEレジスタ・ペアを選択する場合**

### 5.2.3 イミディエト・アドレッシング

**【機能】**

命令コード中に、操作対象となる8ビット・データ、16ビット・データおよび24ビット・データを持つアドレッシングです。

**【オペランド形式】**

次に示すオペランドを持つ命令を実行する際に行われます。

表現形式	記述方法
byte	レーベル, または8ビット・イミディエト・データ
word	レーベル, または16ビット・イミディエト・データ
imm24	レーベル, または24ビット・イミディエト・データ

**【記述例】**

一般例

**ADD A, #byte**

具体例

**ADD A, #77H** ; byteとして77Hをとる場合

### 5.2.4 8ビット・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、オペランド・アドレスとなって操作対象となるメモリをアドレスするアドレッシングです。MOVTBLW命令で使用します。LOCATION 0H命令実行時には0FE00H-0FEFFHのメモリを、LOCATION 0FH命令実行時には0FFE00H-0FFEFFFHのメモリをアドレスします。

**【オペランド形式】**

次に示すオペランドを持つ命令を実行する際に行われます。

表現形式	記述方法
!addr8	ラベル, または0FE00H-0FEFFH <sup>注</sup> イミディエト・データ

**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時は0FFE00H-0FFEFFFH

なお,  $\mu$  PD784915サブシリーズは, LOCATION 0H命令固定です。

**【記述例】**

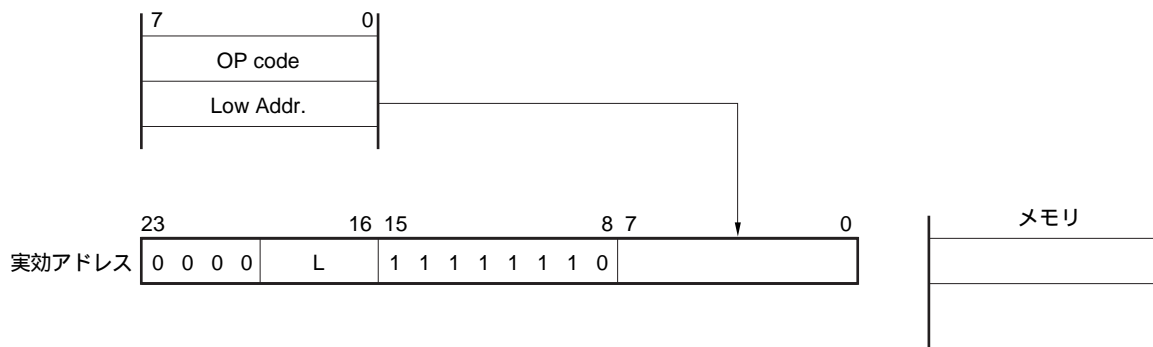
一般例

**MOVTBLW !addr8, n**

具体例

**MOVTBLW !0FE24H, n ; addr8としてFE24Hをとる場合**

**【図 解】**



**備考** Lは, LOCATION命令によって異なります。

- ・ LOCATION 0H命令実行時 : 0000
- ・ LOCATION 0FH命令実行時 : 1111

### 5.2.5 16ビット・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、オペランド・アドレスとなって操作対象となるメモリをアドレスするアドレッシングです。ベース領域に対して、アドレッシングができます。

**【オペランド形式】**

次に示すオペランドを持つ命令を実行する際に行われます。

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】**

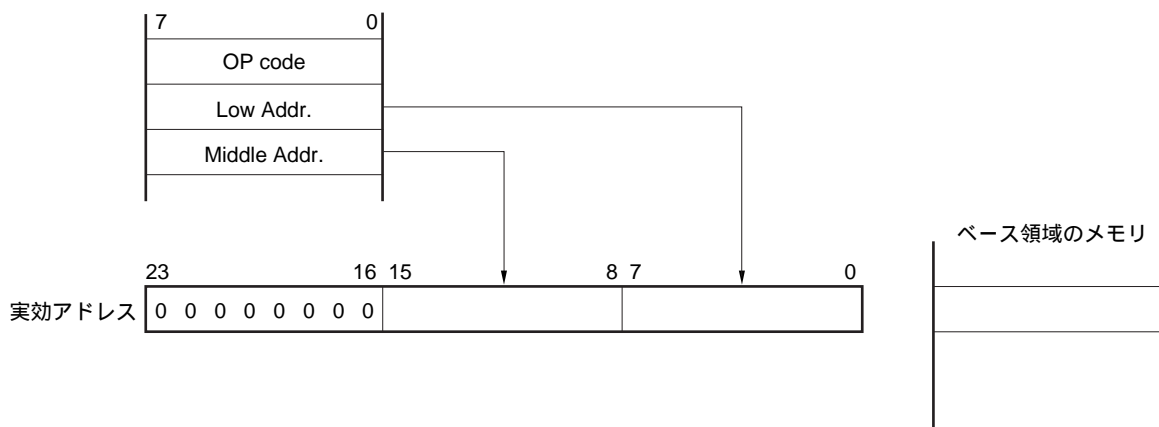
一般例

**MOV A, !addr16**

具体例

**MOV A, !0FE00H ; addr16としてFE00Hをとる場合**

**【図解】**



**【備考】**

このアドレッシングは、実行時間の短縮やオブジェクト・サイズの圧縮などがどうしても必要な場合と、78K/0, 78K/ , 78K/ , 78K/ シリーズ用のソフトウェアを流用する場合で、プログラムの修正が困難な場合にのみ使用してください。

このアドレッシングを使用したプログラムは、そのプログラムを流用しようとしたときに修正が必要になる可能性があります。

### 5.2.6 24ビット・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・データがオペランド・アドレスとなって操作対象となるメモリをアドレスするアドレッシングです。全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】**

次に示すオペランドを持つ命令を実行する際に行われます。

表現形式	記述方法
addr24	ラベル, または24ビット・イミディエト・データ

**【記述例】**

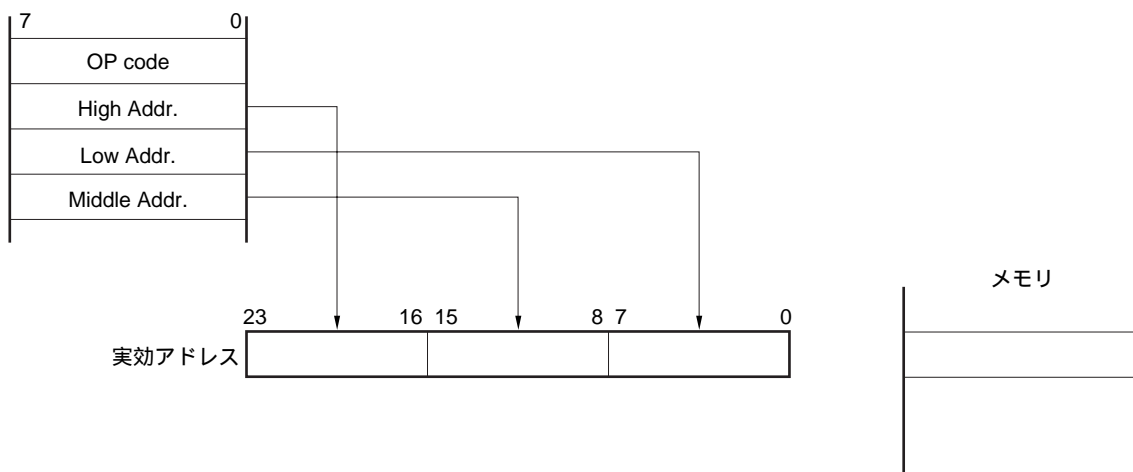
一般例

**MOV A, !!addr24**

具体例

**MOV A, !!54FE00H ; addr24として54FE00Hをとる場合**

**【図解】**



## 5.2.7 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・イミディエト・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は、ほとんどの命令で使用することができ、短いバイト数、短いクロック数で各種のデータを操作することができます。

ショート・ダイレクト・アドレッシングは、LOCATION命令によって適用されるアドレス範囲が、内部データ領域の配置アドレスと同じように変わります。LOCATION 0H命令実行時には、0FD20H-0FEFFHの内部RAMと0FF00H-0FF1FHの特殊機能レジスタ（SFR）がアクセスの対象になります。LOCATION 0FH命令実行時には、0FFD20H-0FFEFFHの内部RAMと0FFF00H-0FFF1FHのSFRがアクセスの対象になります。

ショート・ダイレクト・アドレッシングが適用されるSFR領域には、プログラム上で頻繁にアクセスされるポートや、タイマ/カウンタ・ユニットのコンペア・レジスタ、キャプチャ・レジスタがマッピングされています。短いバイト数、短いクロック数でこれらの特殊機能レジスタを操作することができます。

## 【オペランド形式】

次に示すオペランドを持つ命令を実行する際に行われます。

表現形式	記述方法
saddr	レーベルまたは、0FD20H-0FF1FHのイミディエト・データ
saddr1	レーベルまたは、0FE00H-0FEFFHのイミディエト・データ
saddr2	レーベルまたは、0FD20H-0FDFFHおよび0FF00H-0FF1FHのイミディエト・データ
saddrp	レーベルまたは、0FD20H-0FF1EHのイミディエト・データ
saddrp1	レーベルまたは、0FE00H-0FEFEHのイミディエト・データ
saddrp2	レーベルまたは、0FD20H-0FDFFHおよび0FF00H-0FF1EHのイミディエト・データ (0FDFFHが指定された場合、上位バイトは0FE00Hになります)
saddrg	レーベルまたは、0FD20H-0FEFDHのイミディエト・データ
saddrg1	0FE00H-0FEFDH イミディエト・データまたはレーベル (24ビット操作時)
saddrg2	0FD20H-0FDFFH イミディエト・データまたはレーベル (24ビット操作時)

**備考** 本表に記載されているアドレスは、LOCATION 0H命令実行時の値です。

LOCATION 0FH命令実行時には、上記の値にF0000Hを加えてください。

なお、 $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。

## 【記述例】

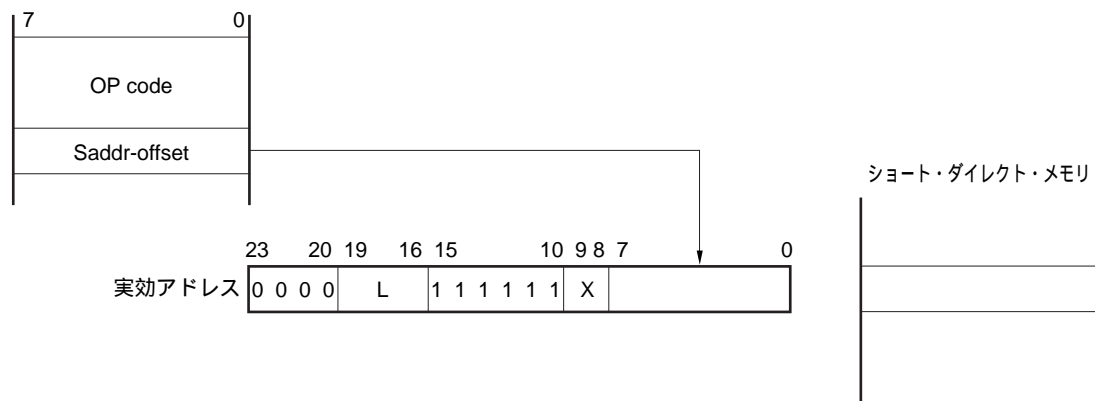
一般例

**MOV saddr, saddr**

具体例

**MOV 0FE30H, 0FE50H**

【図 解】



**備考** Lは、LOCATION命令によって異なります。

- ・ LOCATION 0H命令実行時 : 0000
- ・ LOCATION 0FH命令実行時 : 1111

Xは、オペコード (OP code) 側の情報とSaddr-offsetの値で決まります。

- ・ OP codeでsaddr1が指定されているとき : 10
- ・ OP codeでsaddr2が指定され、かつSaddr-offsetが20H-FFHのとき : 01
- ・ OP codeでsaddr2が指定され、かつSaddr-offsetが00H-1FHのとき : 11

### 5.2.8 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・イミューディエト・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空間は、LOCATION命令によって、適用されるアドレスが内部データ領域の配置アドレスと同じように変わります。LOCATION 0H命令実行時は、0FF00H-0FFFFHの256バイト空間です。LOCATION 0FH命令実行時は、0FFF00H-0FFFFFHの256バイト空間です。ただし、0FF00H-0FF1FH (LOCATION 0H命令実行時、LOCATION 0FH命令実行時は0FFF00H-0FFF1FH) 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。

**備考1.** NEC製のアセンブラ・パッケージ (RA78K4) では、ショート・ダイレクト・アドレッシングでアクセス可能なアドレスにあるSFRに対する命令は、自動的 (強制的) にショート・ダイレクト・アドレッシングを使用します。

**2.** μPD784915サブシリーズは、LOCATION 0H命令固定です。

**【オペランド形式】**

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ます。

表現形式	記述方法
sfr	特殊機能レジスタ名
sfrp	16ビット操作可能特殊機能レジスタ名

**【記述例】**

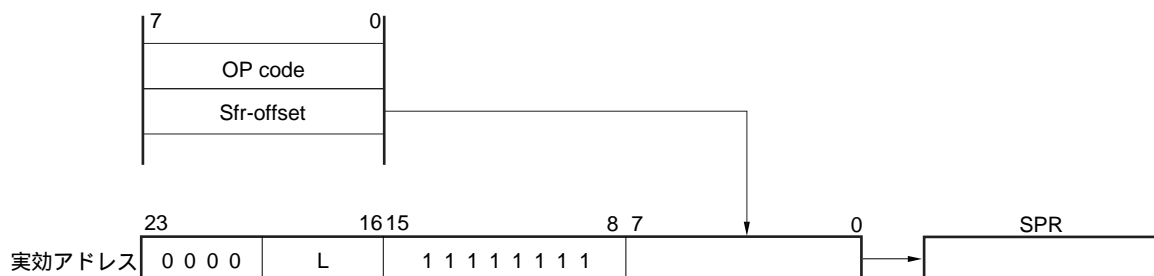
一般例

**MOV sfr, A**

具体例

**MOV PM0, A ; sfrとしてPM0を指定する場合**

**【図解】**



**備考** Lは、LOCATION命令によって異なります。

- ・ LOCATION 0H命令実行時 : 0000
- ・ LOCATION 0FH命令実行時 : 1111

### 5.2.9 ショート・ダイレクト16ビット・メモリ・インダイレクト・アドレッシング

**【機能】**

命令語中の、8ビット・イミディエト・データでアドレスされる2バイトの連続したショート・ダイレクト・メモリの内容がオペランド・アドレスの下位16ビットに、オペランド・アドレスの上位8ビットは00000000となって操作対象となるベース領域のメモリ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は、オペランドに [ saddrp ] を持つ命令を実行する際に行われます。

**【オペランド形式】**

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ます。

表現形式	記述方法
[ saddrp ]	[ レーベル, FD20H-FEFEH <sup>注</sup> のイミディエト・データ ]

注 LOCATION 0命令実行時。LOCATION 0FH命令実行時はFFD20H-FFEFEH  
 なお、μPD784915サブシリーズは、LOCATION 0H命令固定です。

**【記述例】**

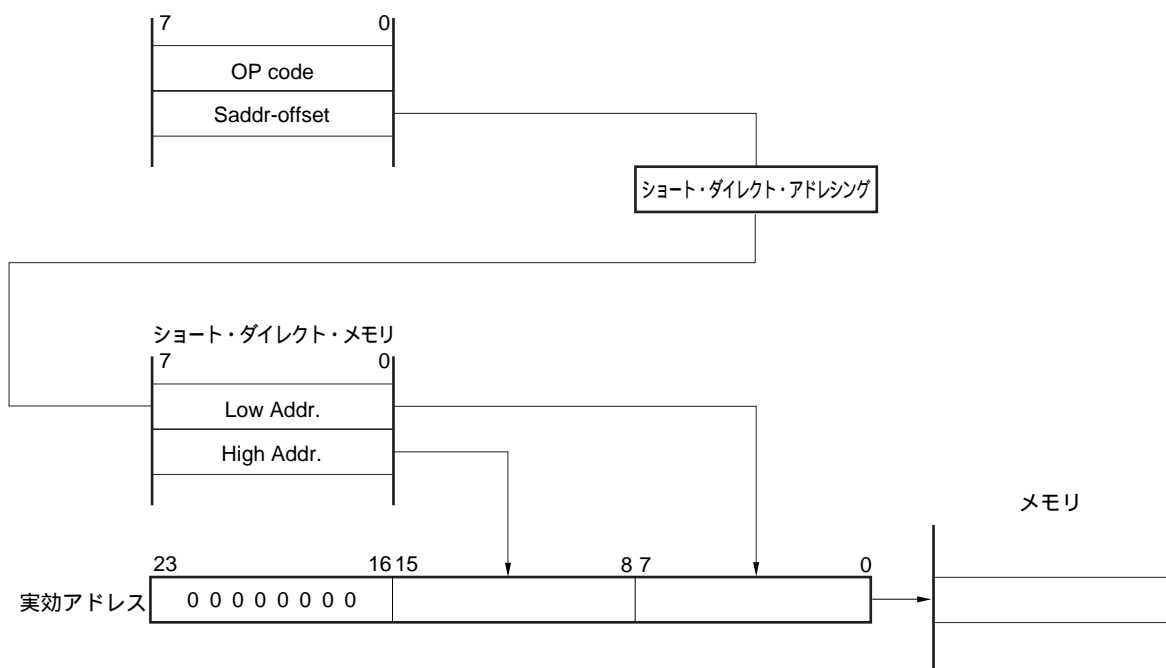
一般例

**XCH A, [ saddrp ]**

具体例

**XCH A, [ 0FEA0H ] ; 0FEA0H, 0FEA1H番地の2バイトのデータで示されるメモリを指定する場合**

**【図 解】**



**【備 考】**

このアドレッシングは、実行時間の短縮やオブジェクト・サイズの圧縮などがどうしても必要な場合と、78K/0, 78K/ , 78K/ , 78K/ シリーズ用のソフトウェアを流用する場合で、プログラムの修正が困難な場合にのみ使用してください。

このアドレッシングを使用したプログラムは、そのプログラムを流用しようとしたときに修正が必要になる可能性があります。

### 5.2.10 ショート・ダイレクト24ビット・メモリ・インダイレクト・アドレッシング

**【機能】**

命令語中の、8ビット・イミディエト・データでアドレスされる3バイトの連続したショート・ダイレクト・メモリの内容がオペランド・アドレスとなって操作対象となるメモリ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は、オペランドに [ %saddrg ] を持つ命令を実行する際に行われます。

**【オペランド形式】**

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ます。

表現形式	記述方法
[ %saddrg ]	[ %レーベル, FD20H-FEFDH <sup>注</sup> のイミディエト・データ ]

**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時は0FFD20H-0FFEFDH  
 なお、μPD784915サブシリーズは、LOCATION 0H命令固定です。

**【記述例】**

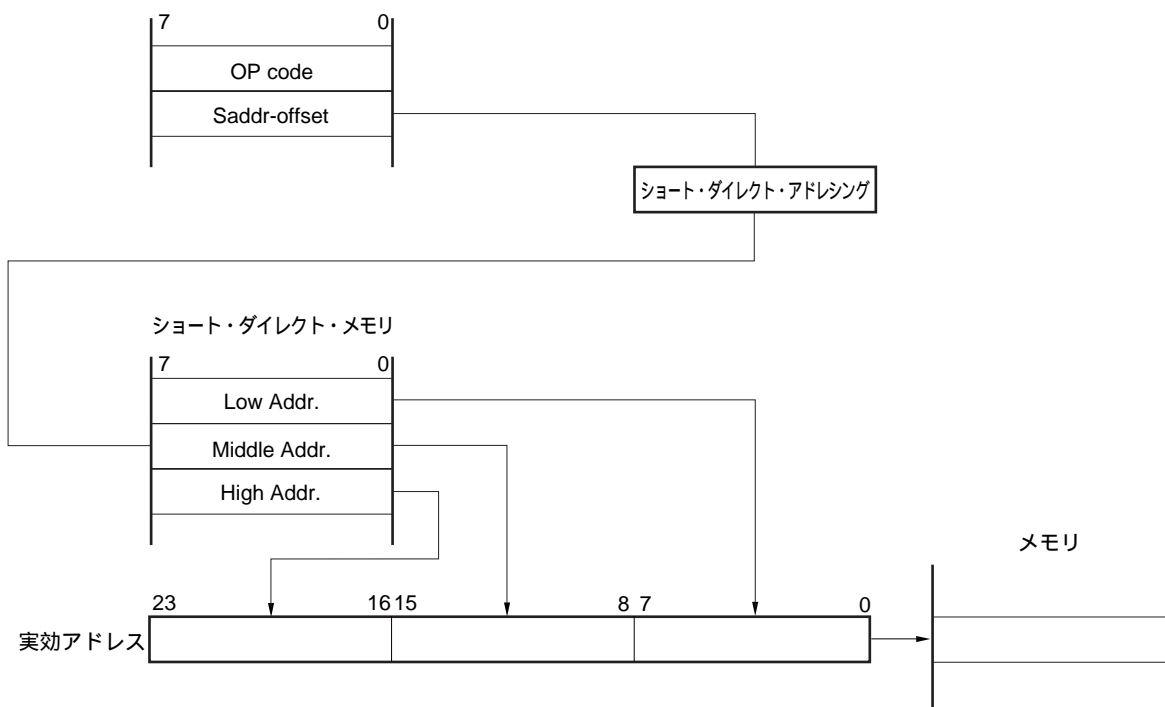
一般例

**XCH A, [ %saddrg ]**

具体例

**XCH A, [ %0FEA0H ] ; 0FEA0H, 0FEA1H, 0FEA2H番地の3バイトのデータで示されるメモリを指定する場合**

**【図解】**



## 5.2.11 スタック・アドレッシング

## 【機能】

スタック・ポインタ (SP) およびユーザ・スタック・ポインタ (UUP) 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。

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

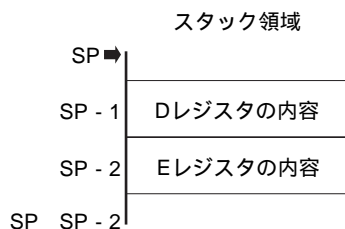
UUPIは、PUSHU, POPU命令の実行時に自動的に使用されます。

## 【記述例】

**PUSH DE** ; PUSH命令を使用して、DEレジスタの内容をスタック領域に退避する場合

この命令の実行により、SPが自動的にデクリメント (-2) されて、DEレジスタの内容がスタック領域に退避される。

## 【図解】



**注意** スタック・アドレッシングでは、すべての空間でアクセス可能ですが、SFR領域および内部ROM領域にスタック・エリアを確保することはできません。

## 5.2.12 24ビット・レジスタ・インダイレクト・アドレッシング

## 【機能】

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の、命令語中のレジスタ・ペア指定コードで指定されるレジスタrg (RG4-RG7) の内容が、オペランド・アドレスとなって操作対象となるメモリをアドレスするアドレッシングです。全メモリ空間に対してアドレッシングできます。

さらに、アドレス指定したレジスタを命令実行後にインクリメント (+1/+2/+3) するオートインクリメント付きレジスタ・インダイレクト・アドレッシングおよび命令実行後にデクリメント (-1/-2/-3) されるオートデクリメント付きレジスタ・インダイレクト・アドレッシングを備えています。インクリメントおよびデクリメントする値は、操作対象のデータのサイズに従います。

このアドレッシングは、連続して複数のデータを処理する場合に最適です。

## 【オペランド形式】

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ます。

表現形式	記述方法
mem	[ TDE ], [ WHL ], [ TDE+ ], [ WHL+ ], [ TDE- ], [ WHL- ], [ VVP ], [ UUP ]
mem1	[ TDE ], [ WHL ], [ TDE+ ], [ TDE- ]
mem2	[ TDE ], [ WHL ]
mem3	[ TDE ], [ WHL ], [ VVP ], [ UUP ]

備考 レジスタ名の後ろに “ + ” : オートインクリメント付き

“ - ” : オートデクリメント付き

## 【記述例】

一般例

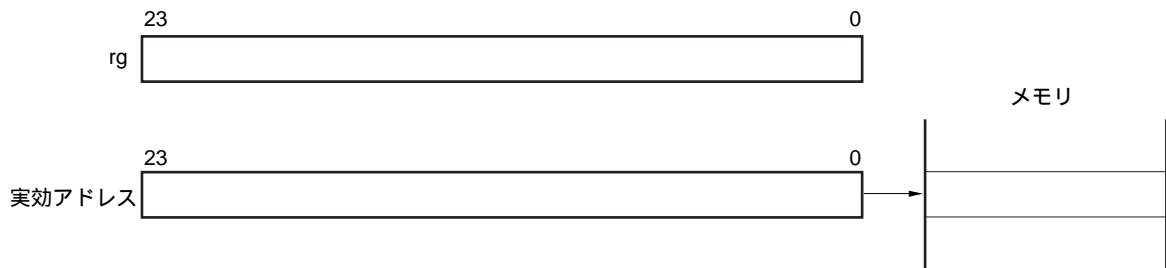
**MOV A, mem**

具体例

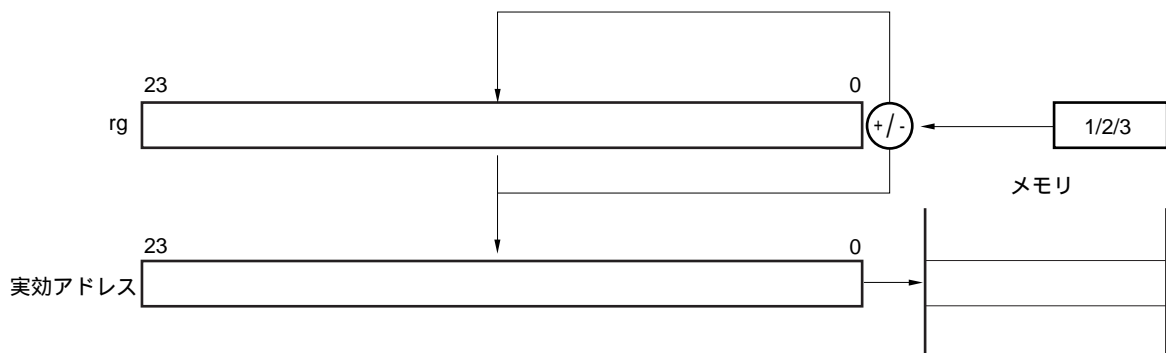
**ADD A, [ TDE ] ; memとして [ TDE ] を指定する場合**

【図 解】

24ビット・レジスタ・インダイレクト・アドレッシング



オートインクリメント/オートデクリメント付きレジスタ・インダイレクト・アドレッシング



備考 + / -

- + : オートインクリメント付き
- : オートデクリメント付き

1/2/3

- 1 : データのサイズが1バイトのとき
- 2 : " 2バイト (1ワード) のとき
- 3 : " 3バイトのとき

### 5.2.13 16ビット・レジスタ・インダイレクト・アドレッシング

**【機能】**

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の、命令語中のレジスタ指定コードで指定されるレジスタrp (RP0-RP3) の内容が、オペランド・アドレスとなって操作対象となるメモリをアドレスするアドレッシングです。ベース領域のメモリ空間に対してアドレッシングできません。

このアドレッシングは、ROR4, ROL4命令でのみ使用され、連続した複数バイトのBCDデータを処理する場合に使用します。

このアドレッシングは、78K $\mu$  シリーズとの互換性を保つために用意されたアドレッシング・モードです。78K $\mu$  シリーズ用のプログラムを流用する場合以外には使用しないでください。

**【オペランド形式】**

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ます。

表現形式	記述方法
mem3	[AX], [BC], [RP2], [RP3]

**【記述例】**

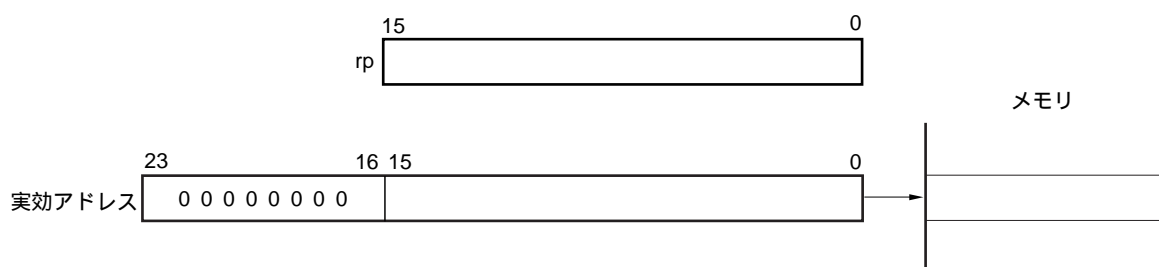
一般例

**ROR4 mem3**

具体例

**ROR4 [BC]** ; mem3として [BC] を記述した場合

**【図解】**



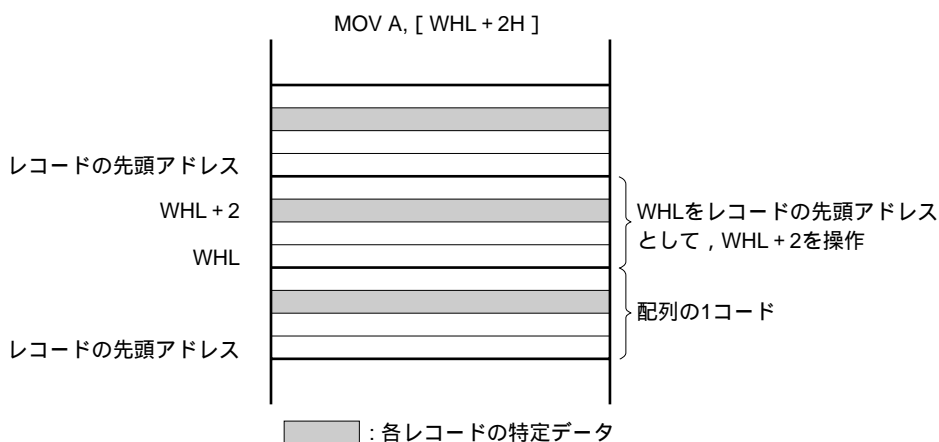
### 5.2.14 ベースト・アドレッシング

**【機能】**

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の、命令語中のレジスタ指定コードで指定されるレジスタrg (RG4-RG7) またはスタック・ポインタ (SP) をベース・レジスタとして、この内容に8ビットのイミディエト・データをオフセット・データとして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。加算は、オフセット・データを正の数として24ビットに拡張して行います。24ビット目からの桁上りは無視します。

全メモリ空間をアドレスすることができます。

このアドレッシングは、複数バイトのデータを1レコードとする配列中の特定データを指定する場合などに使用します。



**【オペランド形式】**

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ます。

表現形式	記述方法
mem	[ TDE + byte ], [ WHL + byte ], [ SP + byte ], [ VVP + byte ], [ UUP + byte ]
mem1	[ TDE + byte ], [ WHL + byte ], [ SP + byte ], [ VVP + byte ], [ UUP + byte ]

**【記述例】**

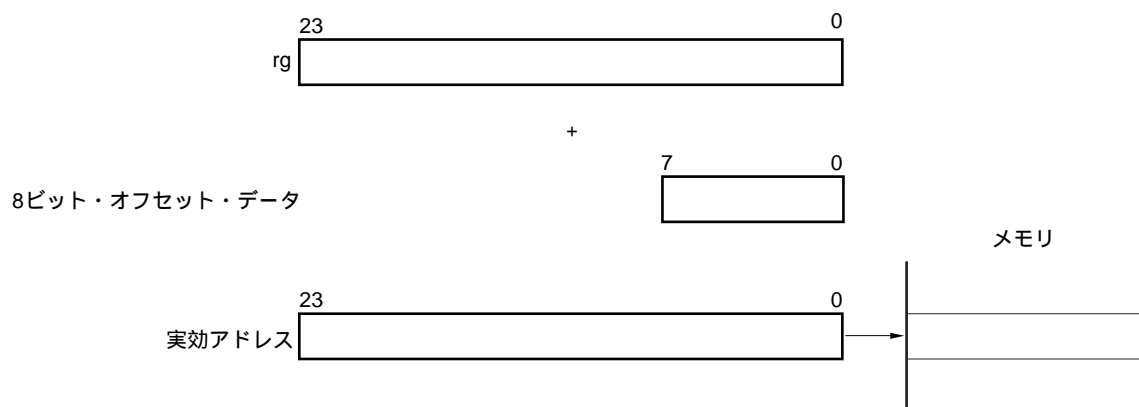
一般例

**AND A, mem**

具体例

**AND A, [ TDE + 10H ] ; memとしてレジスタTDEと10Hとの和のベースト・アドレッシングを選択する場合**

【図 解】



### 5.2.15 インデクスト・アドレッシング

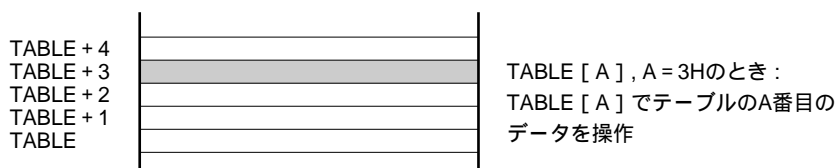
**【機能】**

命令語中のオペランドに記述した24ビット・アドレス・データをインデクスとして、この値にレジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の命令語中で指定する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。加算は、レジスタの内容を正の数として、24ビットに拡張して行います。24ビット目からの桁上がりは無視します。

全メモリ空間をアドレスすることができます。

このアドレッシングは、テーブル・データの読み出しなどに使用します。

なお、このアドレッシングで使用されるA, Bレジスタは、PSWのRSSビットの内容によって変わります。RSS = 0のときは、それぞれ、R1, R3となります。RSS = 1のときは、それぞれ、R5, R7となります。RSS = 1として使用するの、78K/ シリーズ用のプログラムを流用する場合だけにしてください。



**【オペランド形式】**

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ます。

表現形式	記述方法
mem	imm24 [ A ] , imm24 [ B ] , imm24 [ DE ] , imm24 [ HL ]
mem1	imm24 [ A ] , imm24 [ B ] , imm24 [ DE ] , imm24 [ HL ]

**【記述例】**

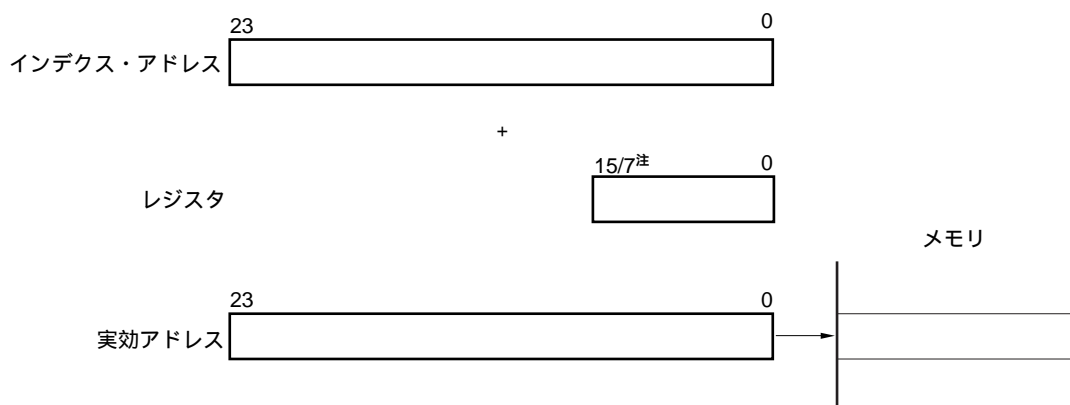
一般例

**ADDC A, mem**

具体例

**ADDC A, 4010H [ DE ] ; memとしてレジスタDEと04010Hとの和のインデクスト・アドレッシングを選択する場合**

【図 解】



注 15 : レジスタがDE, HLの場合

7 : レジスタがA, Bの場合

### 5.2.16 ベースト・インデクスト・アドレッシング

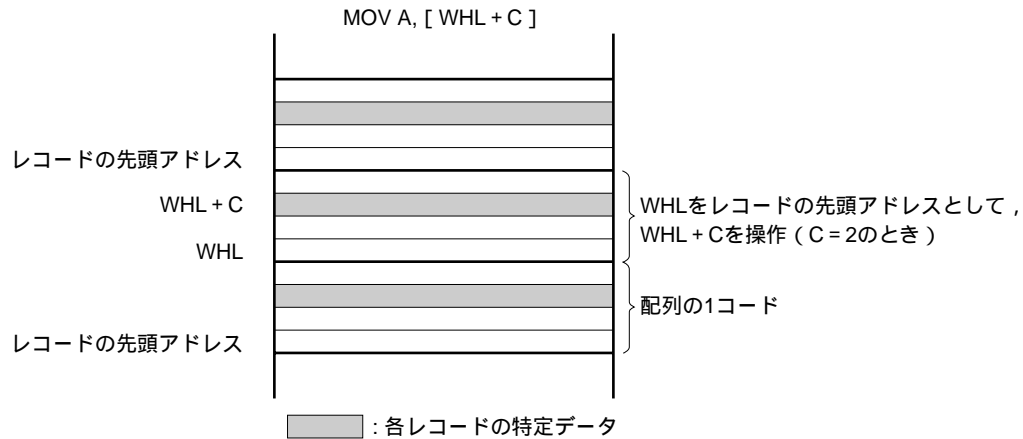
**【機能】**

レジスタ・バンク選択フラグ (RBS2, RBS1, RBS0) で指定されるレジスタ・バンク中の、命令語中のレジスタ指定コードで指定されるレジスタをベース・レジスタとして、この内容に、同様に指定されるレジスタの値をオフセット・データとして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。加算は、オフセット・データを正の数として24ビットに拡張して行います。24ビット目からの桁上りは無視します。

全メモリ空間をアドレスすることができます。

このアドレッシングは、複数バイトのデータを1レコードとする配列中のデータを順番に指定する場合などに使用します。

なお、このアドレッシングで使用されるA, B, Cレジスタは、PSWのRSSビットの内容によって変わります。RSS = 0のときは、それぞれ、R1, R3, R2となります。RSS = 1のときは、それぞれ、R5, R7, R6となります。RSS = 1として使用するのは、78K/ シリーズ用のプログラムを流用する場合だけにしてください。



**【オペランド形式】**

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ます。

表現形式	記述方法
mem	[ TDE + A ], [ TDE + B ], [ TDE + C ], [ WHL + A ], [ WHL + B ], [ WHL + C ], [ VVP + DE ], [ VVP + HL ]
mem1	[ TDE + A ], [ TDE + B ], [ TDE + C ], [ WHL + A ], [ WHL + B ], [ WHL + C ], [ VVP + DE ], [ VVP + HL ]

**【記述例】**

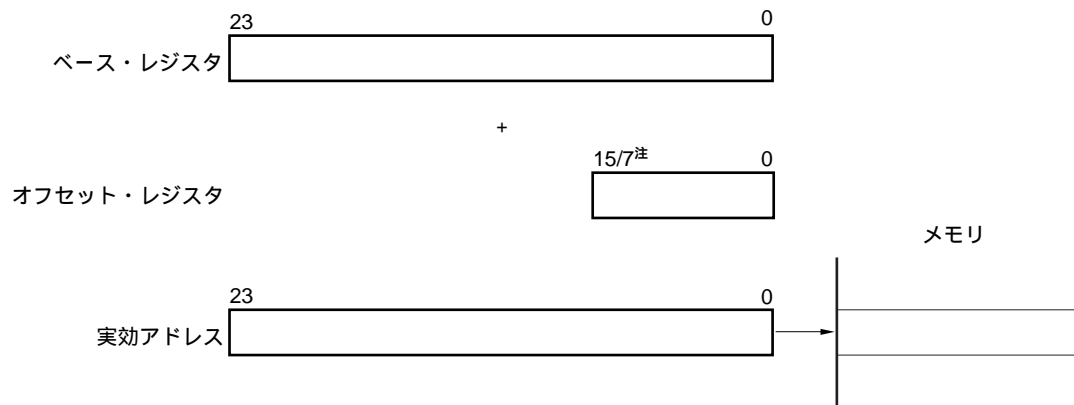
一般例

**AND A, mem**

具体例

**AND A, [ TDE + B ]** ; memとしてレジスタTDEとレジスタBとの和のベースト・インデクスト・アドレッシングを選択する場合

【図 解】



注 15 : レジスタがDE, HLの場合  
 7 : レジスタがA, B, Cの場合

〔メモ〕

## 第6章 命令セット

この章では、78K<sup>1</sup> シリーズの命令セットを一覧表にして示します。

### 6.1 凡 例

#### (1) オペランドの表現形式と記述方法 (1/2)

表現形式	記述方法
r, r <sup>注1</sup>	X ( R0 ), A ( R1 ), C ( R2 ), B ( R3 ), R4, R5, R6, R7, R8, R9, R10, R11, E ( R12 ), D ( R13 ), L ( R14 ), H ( R15 )
r1 <sup>注1</sup>	X ( R0 ), A ( R1 ), C ( R2 ), B ( R3 ), R4, R5, R6, R7
r2	R8, R9, R10, R11, E ( R12 ), D ( R13 ), L ( R14 ), H ( R15 )
r3	V, U, T, W
rp, rp <sup>注2</sup>	AX ( RP0 ), BC ( RP1 ), RP2, RP3, VP ( RP4 ), UP ( RP5 ), DE ( RP6 ), HL ( RP7 )
rp1 <sup>注2</sup>	AX ( RP0 ), BC ( RP1 ), RP2, RP3
rp2	VP ( RP4 ), UP ( RP5 ), DE ( RP6 ), HL ( RP7 )
rg, rg'	VVP ( RG4 ), UUP ( RG5 ), TDE ( RG6 ), WHL ( RG7 )
sfr	特殊機能レジスタ略号 ( <b>特殊機能レジスタ活用表</b> 参照 )
sfrp	特殊機能レジスタ略号 ( 16ビット操作可能レジスタ : <b>特殊機能レジスタ活用表</b> 参照 )
post <sup>注2</sup>	AX ( RP0 ), BC ( RP1 ), RP2, RP3, VP ( RP4 ), UP ( RP5 ) /PSW, DE ( RP6 ), HL ( RP7 ) 複数記述可能。ただし、UPIはPUSH/POP命令、PSWはPUSHU/POPU命令にかぎる
mem	[ TDE ], [ WHL ], [ TDE + ], [ WHL + ], [ TDE - ], [ WHL - ], [ VVP ], [ UUP ] : レジスタ・インダイレクト・アドレッシング [ TDE + byte ], [ WHL + byte ], [ SP + byte ], [ UUP + byte ], [ VVP + byte ] : ベースト・アドレッシング imm24 [ A ], imm24 [ B ], imm24 [ DE ], imm24 [ HL ] : インデクスト・アドレッシング [ TDE + A ], [ TDE + B ], [ TDE + C ], [ WHL + A ], [ WHL + B ], [ WHL + C ], [ VVP + DE ], [ VVP + HL ] : ベースト・インデクスト・アドレッシング
mem1	memから [ WHL + ], [ WHL - ]を除いたすべて
mem2	[ TDE ], [ WHL ]
mem3	[ AX ], [ BC ], [ RP2 ], [ RP3 ], [ VVP ], [ UUP ], [ TDE ], [ WHL ]

注1 . RSSビットを1とすることで、R4-R7をX, A, C, Bとして使用することができますが、この機能を使用するのは、78K<sup>1</sup> シリーズ用のプログラムを流用する場合だけにしてください。

2 . RSSビットを1とすることで、RP2, RP3をAX, BCとして使用することができますが、この機能を使用するのは、78K<sup>1</sup> シリーズ用のプログラムを流用する場合だけにしてください。

## (1) オペランドの表現形式と記述方法 (2/2)

表現形式	記述方法
<b>注</b>	
saddr, saddr'	FD20H-FF1FH イミーディエト・データまたはレーベル
saddr1, saddr1'	FE00H-FEFFFH イミーディエト・データまたはレーベル
saddr2, saddr2'	FD20H-FDFFFH, FF00H-FF1FH イミーディエト・データまたはレーベル
saddrp	FD20H-FF1EH イミーディエト・データまたはレーベル (16ビット操作時)
saddrp1	FE00H-FEFEH イミーディエト・データまたはレーベル (16ビット操作時)
saddrp2	FD20H-FDFFFH, FF00H-FF1EH イミーディエト・データまたはレーベル (16ビット操作時)
saddrg	FD20H-FEFDH イミーディエト・データまたはレーベル (24ビット操作時)
saddrg1	FE00H-FEFDH イミーディエト・データまたはレーベル (24ビット操作時)
saddrg2	FD20H-FDFFFH イミーディエト・データまたはレーベル (24ビット操作時)
addr24	0H-FFFFFFH イミーディエト・データまたはレーベル
addr20	0H-FFFFFFH イミーディエト・データまたはレーベル
addr16	0H-FFFFFH イミーディエト・データまたはレーベル
addr11	800H-FFFFH イミーディエト・データまたはレーベル
addr8	0FE00H-0FEFFFH <sup>注</sup> イミーディエト・データまたはレーベル
addr5	40H-7EH イミーディエト・データまたはレーベル
imm24	24ビット・イミーディエト・データまたはレーベル
word	16ビット・イミーディエト・データまたはレーベル
byte	8ビット・イミーディエト・データまたはレーベル
bit	3ビット・イミーディエト・データまたはレーベル
n	3ビット・イミーディエト・データ
locaddr	00Hまたは0FH

**注** LOCATION命令で00Hを指定した場合は、ここに示したアドレスになります。

LOCATION命令で0FHを指定した場合は、ここに示したアドレスにF0000Hを加えた値がアドレスになります。

なお、 $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。

## (2) オペランド欄の記号

記号	説明
+	オートインクリメント
-	オートデクリメント
#	イミーディエト・データ
!	16ビット絶対アドレス
!!	24ビット/20ビット絶対アドレス
\$	8ビット相対アドレス
#!	16ビット相対アドレス
/	ビット反転
[ ]	インダイレクト・アドレッシング
[ % ]	24ビット・インダイレクト・アドレッシング

## (3) フラグ欄の記号

記号	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
x	結果に従ってセット/クリアされる
P	PNフラグがパリティ・フラグとして動作する
V	PNフラグがオーバフロー・フラグとして動作する
R	以前に退避した値がリストアされる

## (4) オペレーション欄の記号

記号	説明
jdisp8	次の命令の先頭アドレスと分岐先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)
jdisp16	次の命令の先頭アドレスと分岐先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 (16ビット)
PC <sub>HW</sub>	PCのビット16-19
PC <sub>LW</sub>	PCのビット0-15

## (5) オペランドにmemを含む命令のバイト数

memのモード	レジスタ・インダイレクト・ アドレッシング	ベースト・ アドレッシング	インデクスト・ アドレッシング	ベースト・インデク スト・アドレッシング
バイト数	1	2 <sup>注</sup>	3	2

注 MOV命令でmemに [ TDE ], [ WHL ], [ TDE + ], [ TDE - ], [ WHL + ], [ WHL - ] を記述した場合のみ, 1バイト命令になります。

## (6) オペランドにsaddr, saddrp, r, rpを含む命令のバイト数

オペランドにsaddr, saddrp, r, rpを含む命令の中には, バイト数をスラッシュ “ / ” で2つに分けて記述しているものがあります。どちらのバイト数になるかは下表によります。

表現形式	バイト数の左側	バイト数の右側
saddr	saddr2	saddr1
saddrp	saddrp2	saddrp1
r	r1	r2
rp	rp1	rp2

## (7) オペランドにmemを含む命令, スtring命令の記述

オペランドのTDE, WHL, VVP, UUP (24ビット・レジスタ) は, それぞれDE, HL, VP, UPと記述することもできます。ただし, DE, HL, VP, UPと記述した場合でもTDE, WHL, VVP, UUP (24ビット・レジスタ) として扱われます。

## 6.2 命令のオペレーション一覧

## (1) 8ビット・データ転送命令：MOV

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	PV	CY
MOV	r, #byte	2/3	r byte					
	saddr, #byte	3/4	(saddr) byte					
	sfr, #byte	3	sfr byte					
	!addr16, #byte	5	(addr16) byte					
	!!addr24, #byte	6	(addr24) byte					
	r, r'	2/3	r r'					
	A, r	1/2	A r					
	A, saddr2	2	A (saddr2)					
	r, saddr	3	r (saddr)					
	saddr2, A	2	(saddr2) A					
	saddr, r	3	(saddr) r					
	A, sfr	2	A sfr					
	r, sfr	3	r sfr					
	sfr, A	2	sfr A					
	sfr, r	3	sfr r					
	saddr, saddr'	4	(saddr) (saddr')					
	r, !addr16	4	r (addr16)					
	!addr16, r	4	(addr16) r					
	r, !!addr24	5	r (addr24)					
	!!addr24, r	5	(addr24) r					
	A, [saddrp]	2/3	A ((saddrp))					
	A, [%saddrg]	3/4	A ((saddrg))					
	A, mem	1-5	A (mem)					
	[saddrp], A	2/3	((saddrp)) A					
	[%saddrg], A	3/4	((saddrg)) A					
	mem, A	1-5	(mem) A					
	PSWL, #byte	3	PSW <sub>L</sub> byte			x	x	x
	PSWH, #byte	3	PSW <sub>H</sub> byte					
	PSWL, A	2	PSW <sub>L</sub> A			x	x	x
	PSWH, A	2	PSW <sub>H</sub> A					
	A, PSWL	2	A PSW <sub>L</sub>					
	A, PSWH	2	A PSW <sub>H</sub>					
r3, #byte	3	r3 byte						
A, r3	2	A r3						
r3, A	2	r3 A						

(2) 16ビット・データ転送命令 : MOVW

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	PV	CY
MOVW	rp, #word	3	rp word					
	saddrp, #word	4/5	( saddrp ) word					
	sfrp, #word	4	sfrp word					
	!addr16, #word	6	( addr16 ) word					
	!!addr24, #word	7	( addr24 ) word					
	rp, rp'	2	rp rp'					
	AX, saddrp2	2	AX ( saddrp2 )					
	rp, saddrp	3	rp ( saddrp )					
	saddrp2, AX	2	( saddrp2 ) AX					
	saddrp, rp	3	( saddrp ) rp					
	AX, sfrp	2	AX sfrp					
	rp, sfrp	3	rp sfrp					
	sfrp, AX	2	sfrp AX					
	sfrp, rp	3	sfrp rp					
	saddrp, saddrp'	4	( saddrp ) ( saddrp' )					
	rp, !addr16	4	rp ( addr16 )					
	!addr16, rp	4	( addr16 ) rp					
	rp, !!addr24	5	rp ( addr24 )					
	!!addr24, rp	5	( addr24 ) rp					
	AX, [ saddrp ]	3/4	AX ( ( saddrp ) )					
	AX, [ %saddrg ]	3/4	AX ( ( saddrg ) )					
	AX, mem	2-5	AX ( mem )					
	[ saddrp ], AX	3/4	( ( saddrp ) ) AX					
	[ %saddrg ], AX	3/4	( ( saddrg ) ) AX					
mem, AX	2-5	( mem ) AX						

## (3) 24ビット・データ転送命令：MOVG

モニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
MOVG	rg, #imm24	5	rg imm24						
	rg, rg'	2	rg rg'						
	rg, !addr24	5	rg ( addr24 )						
	!addr24, rg	5	( addr24 ) rg						
	rg, saddrg	3	rg ( saddrg )						
	saddrg, rg	3	( saddrg ) rg						
	WHL, [ %saddrg ]	3/4	WHL ( ( saddrg ) )						
	[ %saddrg ] ,WHL	3/4	( ( saddrg ) ) WHL						
	WHL, mem1	2-5	WHL ( mem1 )						
	mem1, WHL	2-5	( mem1 ) WHL						

## (4) 8ビット・データ交換命令：XCH

モニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
XCH	r, r'	2/3	r r'						
	A, r	1/2	A r						
	A, saddr2	2	A ( saddr2 )						
	r, saddr	3	r ( saddr )						
	r, sfr	3	r sfr						
	saddr, saddr'	4	( saddr ) ( saddr' )						
	r, !addr16	4	r ( addr16 )						
	r, !addr24	5	r ( addr24 )						
	A, [ saddrp ]	2/3	A ( ( saddrp ) )						
	A, [ %saddrg ]	3/4	A ( ( saddrg ) )						
	A, mem	2-5	A ( mem )						

## (5) 16ビット・データ交換命令：XCHW

モニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
XCHW	rp, rp'	2	rp rp'						
	AX, saddrp2	2	AX ( saddrp2 )						
	rp, saddrp	3	rp ( saddrp )						
	rp, sfrp	3	rp sfrp						
	AX, [ saddrp ]	3/4	AX ( ( saddrp ) )						
	AX, [ %saddrg ]	3/4	AX ( ( saddrg ) )						
	AX, !addr16	4	AX ( addr16 )						
	AX, !addr24	5	AX ( addr24 )						
	saddrp, saddrp'	4	( saddrp ) ( saddrp' )						
	AX, mem	2-5	AX ( mem )						

## (6) 8ビット演算命令 : ADD, ADDC, SUB, SUBC, CMP, AND, OR, XOR

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ADD	A, #byte	2	A, CY A + byte	x	x	x	V	x
	r, #byte	3	r, CY r + byte	x	x	x	V	x
	saddr, #byte	3/4	(saddr), CY (saddr) + byte	x	x	x	V	x
	sfr, #byte	4	sfr, CY sfr + byte	x	x	x	V	x
	r, r'	2/3	r, CY r + r'	x	x	x	V	x
	A, saddr2	2	A, CY A + (saddr2)	x	x	x	V	x
	r, saddr	3	r, CY r + (saddr)	x	x	x	V	x
	saddr, r	3	(saddr), CY (saddr) + r	x	x	x	V	x
	r, sfr	3	r, CY r + sfr	x	x	x	V	x
	sfr, r	3	sfr, CY sfr + r	x	x	x	V	x
	saddr, saddr'	4	(saddr), CY (saddr) + (saddr')	x	x	x	V	x
	A, [saddrp]	3/4	A, CY A + ((saddrp))	x	x	x	V	x
	A, [%saddrg]	3/4	A, CY A + ((saddrg))	x	x	x	V	x
	[saddrp], A	3/4	((saddrp)), CY ((saddrp)) + A	x	x	x	V	x
	[%saddrg], A	3/4	((saddrg)), CY ((saddrg)) + A	x	x	x	V	x
	A, !addr16	4	A, CY A + (addr16)	x	x	x	V	x
	A, !!addr24	5	A, CY A + (addr24)	x	x	x	V	x
	!addr16, A	4	(addr16), CY (addr16) + A	x	x	x	V	x
	!!addr24, A	5	(addr24), CY (addr24) + A	x	x	x	V	x
	A, mem	2-5	A, CY A + (mem)	x	x	x	V	x
mem, A	2-5	(mem), CY (mem) + A	x	x	x	V	x	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>ADDC</b>	A, #byte	2	A, CY A + byte + CY	x	x	x	V	x
	r, #byte	3	r, CY r + byte + CY	x	x	x	V	x
	saddr, #byte	3/4	(saddr), CY (saddr) + byte + CY	x	x	x	V	x
	sfr, #byte	4	sfr, CY sfr + byte + CY	x	x	x	V	x
	r, r'	2/3	r, CY r + r' + CY	x	x	x	V	x
	A, saddr2	2	A, CY A + (saddr2) + CY	x	x	x	V	x
	r, saddr	3	r, CY r + (saddr) + CY	x	x	x	V	x
	saddr, r	3	(saddr), CY (saddr) + r + CY	x	x	x	V	x
	r, sfr	3	r, CY r + sfr + CY	x	x	x	V	x
	sfr, r	3	sfr, CY sfr + r + CY	x	x	x	V	x
	saddr, saddr'	4	(saddr), CY (saddr) + (saddr') + CY	x	x	x	V	x
	A, [saddrp]	3/4	A, CY A + ((saddrp)) + CY	x	x	x	V	x
	A, [%saddrg]	3/4	A, CY A + ((saddrg)) + CY	x	x	x	V	x
	[saddrp], A	3/4	((saddrp)), CY ((saddrp)) + A + CY	x	x	x	V	x
	[%saddrg], A	3/4	((saddrg)), CY ((saddrg)) + A + CY	x	x	x	V	x
	A, !addr16	4	A, CY A + (addr16) + CY	x	x	x	V	x
	A, !!addr24	5	A, CY A + (addr24) + CY	x	x	x	V	x
	!addr16, A	4	(addr16), CY (addr16) + A + CY	x	x	x	V	x
	!!addr24, A	5	(addr24), CY (addr24) + A + CY	x	x	x	V	x
	A, mem	2-5	A, CY A + (mem) + CY	x	x	x	V	x
mem, A	2-5	(mem), CY (mem) + A + CY	x	x	x	V	x	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>SUB</b>	A, #byte	2	A, CY A - byte	x	x	x	V	x
	r, #byte	3	r, CY r - byte	x	x	x	V	x
	saddr, #byte	3/4	(saddr), CY (saddr) - byte	x	x	x	V	x
	sfr, #byte	4	sfr, CY sfr - byte	x	x	x	V	x
	r, r'	2/3	r, CY r - r'	x	x	x	V	x
	A, saddr2	2	A, CY A - (saddr2)	x	x	x	V	x
	r, saddr	3	r, CY r - (saddr)	x	x	x	V	x
	saddr, r	3	(saddr), CY (saddr) - r	x	x	x	V	x
	r, sfr	3	r, CY r - sfr	x	x	x	V	x
	sfr, r	3	sfr, CY sfr - r	x	x	x	V	x
	saddr, saddr'	4	(saddr), CY (saddr) - (saddr')	x	x	x	V	x
	A, [saddrp]	3/4	A, CY A - ((saddrp))	x	x	x	V	x
	A, [%saddrg]	3/4	A, CY A - ((saddrg))	x	x	x	V	x
	[saddrp], A	3/4	((saddrp)), CY ((saddrp)) - A	x	x	x	V	x
	[%saddrg], A	3/4	((saddrg)), CY ((saddrg)) - A	x	x	x	V	x
	A, !addr16	4	A, CY A - (addr16)	x	x	x	V	x
	A, !!addr24	5	A, CY A - (addr24)	x	x	x	V	x
	!addr16, A	4	(addr16), CY (addr16) - A	x	x	x	V	x
	!!addr24, A	5	(addr24), CY (addr24) - A	x	x	x	V	x
	A, mem	2-5	A, CY A - (mem)	x	x	x	V	x
mem, A	2-5	(mem), CY (mem) - A	x	x	x	V	x	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>SUBC</b>	A, #byte	2	A, CY A - byte - CY	x	x	x	V	x
	r, #byte	3	r, CY r - byte - CY	x	x	x	V	x
	saddr, #byte	3/4	(saddr), CY (saddr) - byte - CY	x	x	x	V	x
	sfr, #byte	4	sfr, CY sfr - byte - CY	x	x	x	V	x
	r, r'	2/3	r, CY r + r' - CY	x	x	x	V	x
	A, saddr2	2	A, CY A - (saddr2) - CY	x	x	x	V	x
	r, saddr	3	r, CY r - (saddr) - CY	x	x	x	V	x
	saddr, r	3	(saddr), CY (saddr) - r - CY	x	x	x	V	x
	r, sfr	3	r, CY r - sfr - CY	x	x	x	V	x
	sfr, r	3	sfr, CY sfr - r - CY	x	x	x	V	x
	saddr, saddr'	4	(saddr), CY (saddr) - (saddr') - CY	x	x	x	V	x
	A, [saddrp]	3/4	A, CY A - ((saddrp)) - CY	x	x	x	V	x
	A, [%saddrg]	3/4	A, CY A - ((saddrg)) - CY	x	x	x	V	x
	[saddrp], A	3/4	((saddrp)), CY ((saddrp)) - A - CY	x	x	x	V	x
	[%saddrg], A	3/4	((saddrg)), CY ((saddrg)) - A - CY	x	x	x	V	x
	A, !addr16	4	A, CY A - (addr16) - CY	x	x	x	V	x
	A, !!addr24	5	A, CY A - (addr24) - CY	x	x	x	V	x
	!addr16, A	4	(addr16), CY (addr16) - A - CY	x	x	x	V	x
	!!addr24, A	5	(addr24), CY (addr24) - A - CY	x	x	x	V	x
	A, mem	2-5	A, CY A - (mem) - CY	x	x	x	V	x
mem, A	2-5	(mem), CY (mem) - A - CY	x	x	x	V	x	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>CMP</b>	A, #byte	2	A - byte	x	x	x	V	x
	r, #byte	3	r - byte	x	x	x	V	x
	saddr, #byte	3/4	(saddr) - byte	x	x	x	V	x
	sfr, #byte	4	sfr - byte	x	x	x	V	x
	r, r'	2/3	r - r'	x	x	x	V	x
	A, saddr2	2	A - (saddr2)	x	x	x	V	x
	r, saddr	3	r - (saddr)	x	x	x	V	x
	saddr, r	3	(saddr) - r	x	x	x	V	x
	r, sfr	3	r - sfr	x	x	x	V	x
	sfr, r	3	sfr - r	x	x	x	V	x
	saddr, saddr'	4	(saddr) - (saddr')	x	x	x	V	x
	A, [saddrp]	3/4	A - ((saddrp))	x	x	x	V	x
	A, [%saddrg]	3/4	A - ((saddrg))	x	x	x	V	x
	[saddrp], A	3/4	((saddrp)) - A	x	x	x	V	x
	[%saddrg], A	3/4	((saddrg)) - A	x	x	x	V	x
	A, !addr16	4	A - (addr16)	x	x	x	V	x
	A, !!addr24	5	A - (addr24)	x	x	x	V	x
	!addr16, A	4	(addr16) - A	x	x	x	V	x
	!!addr24, A	5	(addr24) - A	x	x	x	V	x
	A, mem	2-5	A - (mem)	x	x	x	V	x
mem, A	2-5	(mem) - A	x	x	x	V	x	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
AND	A, #byte	2	A A byte	x	x		P	
	r, #byte	3	r r byte	x	x		P	
	saddr, #byte	3/4	(saddr) (saddr) byte	x	x		P	
	sfr, #byte	4	sfr sfr byte	x	x		P	
	r, r'	2/3	r r r'	x	x		P	
	A, saddr2	2	A A (saddr2)	x	x		P	
	r, saddr	3	r r (saddr)	x	x		P	
	saddr, r	3	(saddr) (saddr) r	x	x		P	
	r, sfr	3	r r sfr	x	x		P	
	sfr, r	3	sfr sfr r	x	x		P	
	saddr, saddr'	4	(saddr) (saddr) (saddr')	x	x		P	
	A, [saddrp]	3/4	A A ((saddrp))	x	x		P	
	A, [%saddrg]	3/4	A A ((saddrg))	x	x		P	
	[saddrp], A	3/4	((saddrp)) ((saddrp)) A	x	x		P	
	[%saddrg], A	3/4	((saddrg)) ((saddrg)) A	x	x		P	
	A, !addr16	4	A A (addr16)	x	x		P	
	A, !!addr24	5	A A (addr24)	x	x		P	
	!addr16, A	4	(addr16) (addr16) A	x	x		P	
	!!addr24, A	5	(addr24) (addr24) A	x	x		P	
	A, mem	2-5	A A (mem)	x	x		P	
mem, A	2-5	(mem) (mem) A	x	x		P		

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
OR	A, #byte	2	A A byte	x	x			P
	r, #byte	3	r r byte	x	x			P
	saddr, #byte	3/4	(saddr) (saddr) byte	x	x			P
	sfr, #byte	4	sfr sfr byte	x	x			P
	r, r'	2/3	r r r'	x	x			P
	A, saddr2	2	A A (saddr2)	x	x			P
	r, saddr	3	r r (saddr)	x	x			P
	saddr, r	3	(saddr) (saddr) r	x	x			P
	r, sfr	3	r r sfr	x	x			P
	sfr, r	3	sfr sfr r	x	x			P
	saddr, saddr'	4	(saddr) (saddr) (saddr')	x	x			P
	A, [saddrp]	3/4	A A ((saddrp))	x	x			P
	A, [%saddrg]	3/4	A A ((saddrg))	x	x			P
	[saddrp], A	3/4	((saddrp)) ((saddrp)) A	x	x			P
	[%saddrg], A	3/4	((saddrg)) ((saddrg)) A	x	x			P
	A, !addr16	4	A A (addr16)	x	x			P
	A, !!addr24	5	A A (addr24)	x	x			P
	!addr16, A	4	(addr16) (addr16) A	x	x			P
	!!addr24, A	5	(addr24) (addr24) A	x	x			P
	A, mem	2-5	A A (mem)	x	x			P
mem, A	2-5	(mem) (mem) A	x	x			P	

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>XOR</b>	A, #byte	2	A A - byte	x	x			P
	r, #byte	3	r r - byte	x	x			P
	saddr, #byte	3/4	(saddr) (saddr) - byte	x	x			P
	sfr, #byte	4	sfr sfr - byte	x	x			P
	r, r'	2/3	r r - r'	x	x			P
	A, saddr2	2	A A - (saddr2)	x	x			P
	r, saddr	3	r r - (saddr)	x	x			P
	saddr, r	3	(saddr) (saddr) - r	x	x			P
	r, sfr	3	r r - sfr	x	x			P
	sfr, r	3	sfr sfr - r	x	x			P
	saddr, saddr'	4	(saddr) (saddr) - (saddr')	x	x			P
	A, [saddrp]	3/4	A A - ((saddrp))	x	x			P
	A, [%saddrg]	3/4	A A - ((saddrg))	x	x			P
	[saddrp], A	3/4	((saddrp)) ((saddrp)) - A	x	x			P
	[%saddrg], A	3/4	((saddrg)) ((saddrg)) - A	x	x			P
	A, !addr16	4	A A - (addr16)	x	x			P
	A, !!addr24	5	A A - (addr24)	x	x			P
	!addr16, A	4	(addr16) (addr16) - A	x	x			P
	!!addr24, A	5	(addr24) (addr24) - A	x	x			P
	A, mem	2-5	A A - (mem)	x	x			P
mem, A	2-5	(mem) (mem) - A	x	x			P	

## (7) 16ビット演算命令 : ADDW, SUBW, CMPW

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ADDW	AX, #word	3	AX, CY AX + word	x	x	x	V	x
	rp, #word	4	rp, CY rp + word	x	x	x	V	x
	rp, rp'	2	rp, CY rp + rp'	x	x	x	V	x
	AX, saddrp2	2	AX, CY AX + (saddrp2)	x	x	x	V	x
	rp, saddrp	3	rp, CY rp + (saddrp)	x	x	x	V	x
	saddrp, rp	3	(saddrp), CY (saddrp) + rp	x	x	x	V	x
	rp, sfrp	3	rp, CY rp + sfrp	x	x	x	V	x
	sfrp, rp	3	sfrp, CY sfrp + rp	x	x	x	V	x
	saddrp, #word	4/5	(saddrp), CY (saddrp) + word	x	x	x	V	x
	sfrp, #word	5	sfrp, CY sfrp + word	x	x	x	V	x
	saddrp, saddrp'	4	(saddrp), CY (saddrp) + (saddrp')	x	x	x	V	x
SUBW	AX, #word	3	AX, CY AX - word	x	x	x	V	x
	rp, #word	4	rp, CY rp - word	x	x	x	V	x
	rp, rp'	2	rp, CY rp - rp'	x	x	x	V	x
	AX, saddrp2	2	AX, CY AX - (saddrp2)	x	x	x	V	x
	rp, saddrp	3	rp, CY rp - (saddrp)	x	x	x	V	x
	saddrp, rp	3	(saddrp), CY (saddrp) - rp	x	x	x	V	x
	rp, sfrp	3	rp, CY rp - sfrp	x	x	x	V	x
	sfrp, rp	3	sfrp, CY sfrp - rp	x	x	x	V	x
	saddrp, #word	4/5	(saddrp), CY (saddrp) - word	x	x	x	V	x
	sfrp, #word	5	sfrp, CY sfrp - word	x	x	x	V	x
	saddrp, saddrp'	4	(saddrp), CY (saddrp) - (saddrp')	x	x	x	V	x
CMPW	AX, #word	3	AX - word	x	x	x	V	x
	rp, #word	4	rp - word	x	x	x	V	x
	rp, rp'	2	rp - rp'	x	x	x	V	x
	AX, saddrp2	2	AX - (saddrp2)	x	x	x	V	x
	rp, saddrp	3	rp - (saddrp)	x	x	x	V	x
	saddrp, rp	3	(saddrp) - rp	x	x	x	V	x
	rp, sfrp	3	rp - sfrp	x	x	x	V	x
	sfrp, rp	3	sfrp - rp	x	x	x	V	x
	saddrp, #word	4/5	(saddrp) - word	x	x	x	V	x
	sfrp, #word	5	sfrp - word	x	x	x	V	x
	saddrp, saddrp'	4	(saddrp) - (saddrp')	x	x	x	V	x

(8) 24ビット演算命令 : ADDG, SUBG

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ADDG	rg, rg'	2	rg, CY rg + rg'	x	x	x	V	x
	rg, #imm24	5	rg, CY rg + imm24	x	x	x	V	x
	WHL, saddrg	3	WHL, CY WHL + (saddrg)	x	x	x	V	x
SUBG	rg, rg'	2	rg, CY rg - rg'	x	x	x	V	x
	rg, #imm24	5	rg, CY rg - imm24	x	x	x	V	x
	WHL, saddrg	3	WHL, CY WHL - (saddrg)	x	x	x	V	x

(9) 乗除算命令 : MULU, MULUW, MULW, DIVUW, DIVUX

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
MULU	r	2/3	AX A × r					
MULUW	rp	2	AX (上位), rp (下位) AX × rp					
MULW	rp	2	AX (上位), rp (下位) AX × rp					
DIVUW	r	2/3	AX (商), r (余り) AX ÷ r <sup>注1</sup>					
DIVUX	rp	2	AXDE (商), rp (余り) AXDE ÷ rp <sup>注2</sup>					

注1 . r = 0の場合, r X, AX FFFFH

2 . rp = 0の場合, rp DE, AXDE FFFFFFFFH

(10) 特殊演算命令 : MACW, MACSW, SACW

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
MACW	byte	3	AXDE (B) × (C) + AXDE, B B + 2, C C + 2, byte byte - 1 End if ( byte = 0 or P/V = 1 )	x	x	x	V	x
MACSW	byte	3	AXDE (B) × (C) + AXDE, B B + 2, C C + 2, byte byte - 1 if byte = 0 then End if P/V = 1 then if overflow AXDE 7FFFFFFFH, End if underflow AXDE 80000000H, End	x	x	x	V	x
SACW	[ TDE + ], [ WHL + ]	4	AX   (TDE) - (WHL)   + AX, TDE TDE + 2, WHL WHL + 2 C C - 1 End if ( C = 0 or CY = 1 )	x	x	x	V	x

(11) 増減命令 : INC, DEC, INCW, DECW, INCG, DECG

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
INC	r	1/2	r r + 1	x	x	x	V	
	saddr	2/3	(saddr) (saddr) + 1	x	x	x	V	
DEC	r	1/2	r r - 1	x	x	x	V	
	saddr	2/3	(saddr) (saddr) - 1	x	x	x	V	
INCW	rp	2/1	rp rp + 1					
	saddrp	3/4	(saddrp) (saddrp) + 1					
DECW	rp	2/1	rp rp - 1					
	saddrp	3/4	(saddrp) (saddrp) - 1					
INCG	rg	2	rg rg + 1					
DECG	rg	2	rg rg - 1					

(12) 補正命令 : ADJBA, ADJBS, CVTBW

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ADJBA		2	Decimal Adjust Accumulator after Addition	x	x	x	P	x
ADJBS		2	Decimal Adjust Accumulator after Subtract	x	x	x	P	x
CVTBW		1	X A, A 00H if A <sub>7</sub> = 0 X A, A FFH if A <sub>7</sub> = 1					

(13) シフト・ローテート命令 : ROR, ROL, RORC, ROLC, SHR, SHL, SHRW, SHLW, ROR4, ROL4

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ROR	r, n	2/3	(CY, r <sub>7</sub> r <sub>0</sub> , r <sub>m-1</sub> r <sub>m</sub> ) × n回 n = 0 - 7				P	x
ROL	r, n	2/3	(CY, r <sub>0</sub> r <sub>7</sub> , r <sub>m+1</sub> r <sub>m</sub> ) × n回 n = 0 - 7				P	x
RORC	r, n	2/3	(CY r <sub>0</sub> , r <sub>7</sub> CY, r <sub>m-1</sub> r <sub>m</sub> ) × n回 n = 0 - 7				P	x
ROLC	r, n	2/3	(CY r <sub>7</sub> , r <sub>0</sub> CY, r <sub>m+1</sub> r <sub>m</sub> ) × n回 n = 0 - 7				P	x
SHR	r, n	2/3	(CY r <sub>0</sub> , r <sub>7</sub> 0, r <sub>m-1</sub> r <sub>m</sub> ) × n回 n = 0 - 7	x	x	0	P	x
SHL	r, n	2/3	(CY r <sub>7</sub> , r <sub>0</sub> 0, r <sub>m+1</sub> r <sub>m</sub> ) × n回 n = 0 - 7	x	x	0	P	x
SHRW	rp, n	2	(CY rp <sub>0</sub> , rp <sub>15</sub> 0, rp <sub>m-1</sub> rp <sub>m</sub> ) × n回 n = 0 - 7	x	x	0	P	x
SHLW	rp, n	2	(CY rp <sub>15</sub> , rp <sub>0</sub> 0, rp <sub>m+1</sub> rp <sub>m</sub> ) × n回 n = 0 - 7	x	x	0	P	x
ROR4	mem3	2	A <sub>3-0</sub> (mem3) <sub>3-0</sub> , (mem3) <sub>7-4</sub> A <sub>3-0</sub> , (mem3) <sub>3-0</sub> (mem3) <sub>7-4</sub>					
ROL4	mem3	2	A <sub>3-0</sub> (mem3) <sub>7-4</sub> , (mem3) <sub>3-0</sub> A <sub>3-0</sub> , (mem3) <sub>7-4</sub> (mem3) <sub>3-0</sub>					

(14) ビット操作命令 : MOV1, AND1, OR1, XOR1, NOT1, SET1, CLR1

二モニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	PV	CY	
MOV1	CY, saddr.bit	3/4	CY (saddr.bit)						x
	CY, sfr.bit	3	CY sfr.bit						x
	CY, X.bit	2	CY X.bit						x
	CY, A.bit	2	CY A.bit						x
	CY, PSWL.bit	2	CY PSWL.bit						x
	CY, PSWH.bit	2	CY PSWH.bit						x
	CY, !addr16.bit	5	CY !addr16.bit						x
	CY, !!addr24.bit	6	CY !!addr24.bit						x
	CY, mem2.bit	2	CY mem2.bit						x
	saddr.bit, CY	3/4	(saddr.bit) CY						
	sfr.bit, CY	3	sfr.bit CY						
	X.bit, CY	2	X.bit CY						
	A.bit, CY	2	A.bit CY						
	PSWL.bit, CY	2	PSWL.bit CY	x	x	x	x	x	
	PSWH.bit, CY	2	PSWH.bit CY						
	!addr16.bit, CY	5	!addr16.bit CY						
	!!addr24.bit, CY	6	!!addr24.bit CY						
	mem2.bit, CY	2	mem2.bit CY						
AND1	CY, saddr.bit	3/4	CY CY (saddr.bit)						x
	CY, /saddr.bit	3/4	CY CY ( $\overline{\text{saddr.bit}}$ )						x
	CY, sfr.bit	3	CY CY sfr.bit						x
	CY, /sfr.bit	3	CY CY $\overline{\text{sfr.bit}}$						x
	CY, X.bit	2	CY CY X.bit						x
	CY, /X.bit	2	CY CY $\overline{\text{X.bit}}$						x
	CY, A.bit	2	CY CY A.bit						x
	CY, /A.bit	2	CY CY $\overline{\text{A.bit}}$						x
	CY, PSWL.bit	2	CY CY PSWL.bit						x
	CY, /PSWL.bit	2	CY CY $\overline{\text{PSWL.bit}}$						x
	CY, PSWH.bit	2	CY CY PSWH.bit						x
	CY, /PSWH.bit	2	CY CY $\overline{\text{PSWH.bit}}$						x
	CY, !addr16.bit	5	CY CY !addr16.bit						x
	CY, /!addr16.bit	5	CY CY $\overline{\text{!addr16.bit}}$						x
	CY, !!addr24.bit	6	CY CY !!addr24.bit						x
	CY, /!!addr24.bit	6	CY CY $\overline{\text{!!addr24.bit}}$						x
	CY, mem2.bit	2	CY CY mem2.bit						x
	CY, /mem2.bit	2	CY CY $\overline{\text{mem2.bit}}$						x

ニモニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
<b>OR1</b>	CY, saddr.bit	3/4	CY CY ( saddr.bit )						x
	CY, /saddr.bit	3/4	CY CY ( $\overline{\text{saddr.bit}}$ )						x
	CY, sfr.bit	3	CY CY sfr.bit						x
	CY, /sfr.bit	3	CY CY $\overline{\text{sfr.bit}}$						x
	CY, X.bit	2	CY CY X.bit						x
	CY, /X.bit	2	CY CY $\overline{\text{X.bit}}$						x
	CY, A.bit	2	CY CY A.bit						x
	CY, /A.bit	2	CY CY $\overline{\text{A.bit}}$						x
	CY, PSWL.bit	2	CY CY PSWL.bit						x
	CY, /PSWL.bit	2	CY CY $\overline{\text{PSWL.bit}}$						x
	CY, PSWH.bit	2	CY CY PSWH.bit						x
	CY, /PSWH.bit	2	CY CY $\overline{\text{PSWH.bit}}$						x
	CY, !addr16.bit	5	CY CY !addr16.bit						x
	CY, /!addr16.bit	5	CY CY $\overline{\text{!addr16.bit}}$						x
	CY, !!addr24.bit	6	CY CY !!addr24.bit						x
	CY, /!!addr24.bit	6	CY CY $\overline{\text{!!addr24.bit}}$						x
CY, mem2.bit	2	CY CY mem2.bit						x	
CY, /mem2.bit	2	CY CY $\overline{\text{mem2.bit}}$						x	
<b>XOR1</b>	CY, saddr.bit	3/4	CY CY - ( saddr.bit )						x
	CY, sfr.bit	3	CY CY - sfr.bit						x
	CY, X.bit	2	CY CY - X.bit						x
	CY, A.bit	2	CY CY - A.bit						x
	CY, PSWL.bit	2	CY CY - PSWL.bit						x
	CY, PSWH.bit	2	CY CY - PSWH.bit						x
	CY, !addr16.bit	5	CY CY - !addr16.bit						x
	CY, !!addr24.bit	6	CY CY - !!addr24.bit						x
	CY, mem2.bit	2	CY CY - mem2.bit						x
<b>NOT1</b>	saddr.bit	3/4	( saddr.bit ) ( $\overline{\text{saddr.bit}}$ )						
	sfr.bit	3	sfr.bit $\overline{\text{sfr.bit}}$						
	X.bit	2	X.bit $\overline{\text{X.bit}}$						
	A.bit	2	A.bit $\overline{\text{A.bit}}$						
	PSWL.bit	2	PSWL.bit $\overline{\text{PSWL.bit}}$	x	x	x	x	x	
	PSWH.bit	2	PSWH.bit $\overline{\text{PSWH.bit}}$						
	!addr16.bit	5	!addr16.bit $\overline{\text{!addr16.bit}}$						
	!!addr24.bit	6	!!addr24.bit $\overline{\text{!!addr24.bit}}$						
	mem2.bit	2	mem2.bit $\overline{\text{mem2.bit}}$						
	CY	1	CY $\overline{\text{CY}}$						x

ニモニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
<b>SET1</b>	saddr, bit	2/3	(saddr.bit) 1						
	sfr.bit	3	sfr.bit 1						
	X.bit	2	X.bit 1						
	A.bit	2	A.bit 1						
	PSWL.bit	2	PSWL.bit 1	x	x	x	x	x	
	PSWH.bit	2	PSWH.bit 1						
	!addr16.bit	5	!addr16.bit 1						
	!!addr24.bit	6	!!addr24.bit 1						
	mem2.bit	2	mem2.bit 1						
CY	1	CY 1						1	
<b>CLR1</b>	saddr, bit	2/3	(saddr.bit) 0						
	sfr.bit	3	sfr.bit 0						
	X.bit	2	X.bit 0						
	A.bit	2	A.bit 0						
	PSWL.bit	2	PSWL.bit 0	x	x	x	x	x	
	PSWH.bit	2	PSWH.bit 0						
	!addr16.bit	5	!addr16.bit 0						
	!!addr24.bit	6	!!addr24.bit 0						
	mem2.bit	2	mem2.bit 0						
CY	1	CY 0						0	

(15) スタック操作命令 : PUSH, PUSHU, POP, POPU, MOVG, ADDWG, SUBWG, INCG, DECG

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>PUSH</b> <sup>注1</sup>	PSW	1	(SP - 2) PSW, SP SP - 2					
	sfrp	3	(SP - 2) sfrp, SP SP - 2					
	sfr	3	(SP - 1) sfr, SP SP - 1					
	post	2	{ (SP - 2) post, SP SP - 2 } × m回 <sup>注2</sup>					
	rg	2	(SP - 3) rg, SP SP - 3					
<b>PUSHU</b> <sup>注1</sup>	post	2	{ (UUP - 2) post, UUP UUP - 2 } × m回 <sup>注2</sup>					
<b>POP</b> <sup>注1</sup>	PSW	1	PSW (SP) , SP SP + 2	R	R	R	R	R
	sfrp	3	sfrp (SP) , SP SP + 2					
	sfr	3	sfr (SP) , SP SP + 1					
	post	2	{ post (SP) , SP SP + 2 } × m回 <sup>注2</sup>					
	rg	2	rg (SP) , SP SP + 3					
<b>POPU</b> <sup>注1</sup>	post	2	{ post (UUP) , UUP UUP + 2 } × m回 <sup>注2</sup>					
<b>MOVG</b>	SP, #imm24	5	SP imm24					
	SP, WHL	2	SP WHL					
	WHL, SP	2	WHL SP					
<b>ADDWG</b>	SP, #word	4	SP SP + word					
<b>SUBWG</b>	SP, #word	4	SP SP - word					
<b>INCG</b>	SP	2	SP SP + 1					
<b>DECG</b>	SP	2	SP SP - 1					

注1 . オペレーションの詳細については , 第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

2 . mは , postで指定されたレジスタ数

(16) コール・リターン命令 : CALL, CALLF, CALLT, BRK, BRKCS, RET, RETI, RETB, RETCS, RETCSB

ニモニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
CALL <sup>注</sup>	!addr16	3	(SP - 3) (PC + 3), SP SP - 3, PC <sub>HW</sub> 0, PC <sub>LW</sub> addr16					
	!!addr20	4	(SP - 3) (PC + 4), SP SP - 3, PC addr20					
	rp	2	(SP - 3) (PC + 2), SP SP - 3, PC <sub>HW</sub> 0, PC <sub>LW</sub> rp					
	rg	2	(SP - 3) (PC + 2), SP SP - 3, PC rg					
	[rp]	2	(SP - 3) (PC + 2), SP SP - 3, PC <sub>HW</sub> 0, PC <sub>LW</sub> (rp)					
	[rg]	2	(SP - 3) (PC + 2), SP SP - 3, PC (rg)					
	\$!addr20	3	(SP - 3) (PC + 3), SP SP - 3, PC PC + 3 + jdisp16					
CALLF <sup>注</sup>	!addr11	2	(SP - 3) (PC + 2), SP SP - 3, PC <sub>19-12</sub> 0, PC <sub>11</sub> 1, PC <sub>10-0</sub> addr11					
CALLT <sup>注</sup>	[addr5]	1	(SP - 3) (PC + 1), SP SP - 3, PC <sub>HW</sub> 0, PC <sub>LW</sub> (addr5)					
BRK		1	(SP - 2) PSW, (SP - 1) <sub>0-3</sub> (PC + 1) <sub>HW</sub> , (SP - 4) PC + 1, SP SP - 4 PC <sub>HW</sub> 0, PC <sub>LW</sub> (003EH)					
BRKCS	RBn	2	PC <sub>LW</sub> RP2, RP3 PSW, RBS2 - 0 n, RSS 0, IE 0, RP <sub>38-11</sub> PC <sub>HW</sub> , PC <sub>HW</sub> 0					
RET <sup>注</sup>		1	PC (SP), SP SP + 3					
RETI <sup>注</sup>		1	PC (SP), PSW (SP + 2), SP SP + 4	R	R	R	R	R
RETB <sup>注</sup>		1	PC (SP), PSW (SP + 2), SP SP + 4	R	R	R	R	R
RETCS	!addr16	3	PSW RP3, PC <sub>LW</sub> RP2, RP2 addr16, PC <sub>HW</sub> RP <sub>38-11</sub>	R	R	R	R	R
RETCSB	!addr16	4	PSW RP3, PC <sub>LW</sub> RP2, RP2 addr16, PC <sub>HW</sub> RP <sub>38-11</sub>	R	R	R	R	R

注 オペレーションの詳細については、第3章 レジスタの図3-4 スタック領域へ退避されるデータおよび図3-5 スタック領域から復帰されるデータを参照してください。

## (17) 無条件分岐命令 : BR

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
BR	!addr16	3	PC <sub>HW</sub> 0, PC <sub>LW</sub> addr16					
	!!addr20	4	PC addr20					
	rp	2	PC <sub>HW</sub> 0, PC <sub>LW</sub> rp					
	rg	2	PC rg					
	[ rp ]	2	PC <sub>HW</sub> 0, PC <sub>LW</sub> ( rp )					
	[ rg ]	2	PC ( rg )					
	\$addr20	2	PC PC + 2 + jdisp8					
	!\$addr20	3	PC PC + 3 + jdisp16					

(18) 条件付き分岐命令 : BNZ, BNE, BZ, BE, BNC, BNL, BC, BL, BNV, BPO, BV, BPE, BP, BN, BLT, BGE, BLE, BGT, BNH, BH, BF, BT, BTCLR, BFSET, DBNZ

二モニク	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
<b>BNZ</b>	\$addr20	2	PC PC + 2 + jdisp8 if Z = 0					
<b>BNE</b>								
<b>BZ</b>	\$addr20	2	PC PC + 2 + jdisp8 if Z = 1					
<b>BE</b>								
<b>BNC</b>	\$addr20	2	PC PC + 2 + jdisp8 if CY = 0					
<b>BNL</b>								
<b>BC</b>	\$addr20	2	PC PC + 2 + jdisp8 if CY = 1					
<b>BL</b>								
<b>BNV</b>	\$addr20	2	PC PC + 2 + jdisp8 if P/V = 0					
<b>BPO</b>								
<b>BV</b>	\$addr20	2	PC PC + 2 + jdisp8 if P/V = 1					
<b>BPE</b>								
<b>BP</b>	\$addr20	2	PC PC + 2 + jdisp8 if S = 0					
<b>BN</b>	\$addr20	2	PC PC + 2 + jdisp8 if S = 1					
<b>BLT</b>	\$addr20	3	PC PC + 3 + jdisp8 if P/V - S = 1					
<b>BGE</b>	\$addr20	3	PC PC + 3 + jdisp8 if P/V - S = 0					
<b>BLE</b>	\$addr20	3	PC PC + 3 + jdisp8 if ( P/V - S ) Z = 1					
<b>BGT</b>	\$addr20	3	PC PC + 3 + jdisp8 if ( P/V - S ) Z = 0					
<b>BNH</b>	\$addr20	3	PC PC + 3 + jdisp8 if Z CY = 1					
<b>BH</b>	\$addr20	3	PC PC + 3 + jdisp8 if Z CY = 0					
<b>BF</b>	saddr.bit, \$addr20	4/5	PC PC + 4 <sup>注1</sup> + jdisp8 if ( saddr.bit ) = 0					
	sfr.bit, \$addr20	4	PC PC + 4 + jdisp8 if sfr.bit = 0					
	X.bit, \$addr20	3	PC PC + 3 + jdisp8 if X.bit = 0					
	A.bit, \$addr20	3	PC PC + 3 + jdisp8 if A.bit = 0					
	PSWL.bit, \$addr20	3	PC PC + 3 + jdisp8 if PSWL.bit = 0					
	PSWH.bit, \$addr20	3	PC PC + 3 + jdisp8 if PSWH.bit = 0					
	!addr16.bit, \$addr20	6	PC PC + 3 + jdisp8 if !addr16.bit = 0					
	!!addr24.bit, \$addr20	7	PC PC + 3 + jdisp8 if !!addr24.bit = 0					
mem2.bit, \$addr20	3	PC PC + 3 + jdisp8 if mem2.bit = 0						
<b>BT</b>	saddr.bit, \$addr20	3/4	PC PC + 3 <sup>注2</sup> + jdisp8 if ( saddr.bit ) = 1					
	sfr.bit, \$addr20	4	PC PC + 4 + jdisp8 if sfr.bit = 1					
	X.bit, \$addr20	3	PC PC + 3 + jdisp8 if X.bit = 1					
	A.bit, \$addr20	3	PC PC + 3 + jdisp8 if A.bit = 1					
	PSWL.bit, \$addr20	3	PC PC + 3 + jdisp8 if PSWL.bit = 1					
	PSWH.bit, \$addr20	3	PC PC + 3 + jdisp8 if PSWH.bit = 1					
	!addr16.bit, \$addr20	6	PC PC + 3 + jdisp8 if !addr16.bit = 1					
	!!addr24.bit, \$addr20	7	PC PC + 3 + jdisp8 if !!addr24.bit = 1					
mem2.bit, \$addr20	3	PC PC + 3 + jdisp8 if mem2.bit = 1						

注1 . バイト数が4のとき。5のときはPC PC + 5 + jdisp8になります。

2 . バイト数が3のとき。4のときはPC PC + 4 + jdisp8になります。

ニモニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
<b>BTCLR</b>	saddr.bit, \$addr20	4/5	{ PC PC + 4 <sup>注1</sup> + jdisp8, ( addr.bit ) 0 } if ( saddr.bit ) = 1						
	sfr.bit, \$addr20	4	{ PC PC + 4 + jdisp8, sfr.bit 0 } if sfr.bit = 1						
	X.bit, \$addr20	3	{ PC PC + 3 + jdisp8, X.bit 0 } if X.bit = 1						
	A.bit, \$addr20	3	{ PC PC + 3 + jdisp8, A.bit 0 } if A.bit = 1						
	PSWL.bit, \$addr20	3	{ PC PC + 3 + jdisp8, PSWL.bit 0 } if PSWL.bit = 1	x	x	x	x	x	
	PSWH.bit, \$addr20	3	{ PC PC + 3 + jdisp8, PSWH.bit 0 } if PSWH.bit = 1						
	!addr16.bit, \$addr20	6	{ PC PC + 3 + jdisp8, !addr16.bit 0 } if !addr16 = 1						
	!!addr24.bit, \$addr20	7	{ PC PC + 3 + jdisp8, !!addr24.bit 0 } if !!addr24.bit = 1						
	mem2.bit, \$addr20	3	{ PC PC + 3 + jdisp8, mem2.bit 0 } if mem2.bit = 1						
<b>BFSET</b>	saddr.bit, \$saddr20	4/5	{ PC PC + 4 <sup>注1</sup> + jdisp8, ( saddr.bit ) 1 } if ( saddr.bit ) = 0						
	sfr.bit, \$addr20	4	{ PC PC + 4 + jdisp8, sfr.bit 1 } if sfr.bit = 0						
	X.bit, \$addr20	3	{ PC PC + 3 + jdisp8, X.bit 1 } if X.bit = 0						
	A.bit, \$addr20	3	{ PC PC + 3 + jdisp8, A.bit 1 } if A.bit = 0						
	PSWL.bit, \$addr20	3	{ PC PC + 3 + jdisp8, PSWL.bit 1 } if PSWL.bit = 0	x	x	x	x	x	
	PSWH.bit, \$addr20	3	{ PC PC + 3 + jdisp8, PSWH.bit 1 } if PSWH.bit = 0						
	!addr16.bit, \$addr20	6	{ PC PC + 3 + jdisp8, !addr16.bit 1 } if !addr16 = 0						
	!!addr24.bit, \$addr20	7	{ PC PC + 3 + jdisp8, !!addr24.bit 1 } if !!addr24 = 0						
	mem2.bit, \$addr20	3	{ PC PC + 3 + jdisp8, mem2.bit 1 } if mem2.bit = 0						
<b>DBNZ</b>	B, \$addr20	2	B B - 1, PC PC + 2 + jdisp8 if B 0						
	C, \$addr20	2	C C - 1, PC PC + 2 + jdisp8 if C 0						
	saddr, \$addr20	3/4	( saddr ) ( saddr ) - 1, PC PC + 3 <sup>注2</sup> + jdisp8 if ( saddr ) 0						

注1 . バイト数が4のとき。5のときはPC PC + 5 + jdisp8になります。

2 . バイト数が3のとき。4のときはPC PC + 4 + jdisp8になります。

## (19) CPU制御命令：MOV, LOCATION, SEL, SWRS, NOP, EI, DI

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
MOV	STBC, #byte	4	STBC byte					
	WDM, #byte	4	WDM byte					
LOCATION	locaddr	4	SFR, 内部データ領域の配置アドレスの上位ワード指定					
SEL	RBn	2	RSS 0, RBS2 - 0 n					
	RBn, ALT	2	RSS 1, RBS2 - 0 n					
SWRS		2	RSS $\overline{\text{RSS}}$					
NOP		1	No Operation					
EI		1	IE 1 ( Enable interrupt )					
DI		1	IE 0 ( Disable interrupt )					

## (20) 特殊命令：CHKL, CHKLA

二モニック	オペランド	バイト	オペレーション	フラグ				
				S	Z	AC	P/V	CY
CHKL	sfr	3	( 端子レベル ) - ( 出力ラッチ )	x	x		P	
CHKLA	sfr	3	A ( 端子レベル ) - ( 出力ラッチ )	x	x		P	

**注意**  $\mu$  PD784216, 784216Y, 784218, 784218Y, 784225, 784225Y, 784937サブシリーズは、CHKL, CHKLAをサポートしていません。これらの命令は、使用しないでください。

これらの命令を実行した場合、次のような動作をします。

- CHKL命令・・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Zフラグがセット(1)されます。
- CHKLA命令・・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Aレジスタにその結果を格納するとともに、Zフラグをセット(1)します。

(21) スtring命令 : MOVTLBW, MOVML, XCHM, MOVBLK, XCHBK, CMPME, CMPMNE, CMPML, CMPMNC, CMPBKE, CMPBKNE, CMPBKC, CMPBKNC

二モニック	オペランド	バイト	オペレーション	フラグ					
				S	Z	AC	PV	CY	
<b>MOVTLBW</b>	!addr8, byte	4	(addr8+2) (addr8), byte byte - 1, addr8 addr8 - 2 End if byte = 0						
<b>MOVML</b>	[TDE+], A	2	(TDE) A, TDE TDE + 1, C C - 1 End if C = 0						
	[TDE-], A	2	(TDE) A, TDE TDE - 1, C C - 1 End if C = 0						
<b>XCHM</b>	[TDE+], A	2	(TDE) A, TDE TDE + 1, C C - 1 End if C = 0						
	[TDE-], A	2	(TDE) A, TDE TDE - 1, C C - 1 End if C = 0						
<b>MOVBLK</b>	[TDE+], [WHL+]	2	(TDE) (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0						
	[TDE-], [WHL-]	2	(TDE) (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0						
<b>XCHBK</b>	[TDE+], [WHL+]	2	(TDE) (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0						
	[TDE-], [WHL-]	2	(TDE) (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0						
<b>CMPME</b>	[TDE+], A	2	(TDE) -A, TDE TDE + 1, C C - 1 End if C = 0 or Z = 0	x	x	x	V	x	
	[TDE-], A	2	(TDE) -A, TDE TDE - 1, C C - 1 End if C = 0 or Z = 0	x	x	x	V	x	
<b>CMPMNE</b>	[TDE+], A	2	(TDE) -A, TDE TDE + 1, C C - 1 End if C = 0 or Z = 1	x	x	x	V	x	
	[TDE-], A	2	(TDE) -A, TDE TDE - 1, C C - 1 End if C = 0 or Z = 1	x	x	x	V	x	
<b>CMPML</b>	[TDE+], A	2	(TDE) -A, TDE TDE + 1, C C - 1 End if C = 0 or CY = 0	x	x	x	V	x	
	[TDE-], A	2	(TDE) -A, TDE TDE - 1, C C - 1 End if C = 0 or CY = 0	x	x	x	V	x	
<b>CMPMNC</b>	[TDE+], A	2	(TDE) -A, TDE TDE + 1, C C - 1 End if C = 0 or CY = 1	x	x	x	V	x	
	[TDE-], A	2	(TDE) -A, TDE TDE - 1, C C - 1 End if C = 0 or CY = 1	x	x	x	V	x	
<b>CMPBKE</b>	[TDE+], [WHL+]	2	(TDE) - (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0 or Z = 0	x	x	x	V	x	
	[TDE-], [WHL-]	2	(TDE) - (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0 or Z = 0	x	x	x	V	x	
<b>CMPBKNE</b>	[TDE+], [WHL+]	2	(TDE) - (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0 or Z = 1	x	x	x	V	x	
	[TDE-], [WHL-]	2	(TDE) - (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0 or Z = 1	x	x	x	V	x	
<b>CMPBKC</b>	[TDE+], [WHL+]	2	(TDE) - (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0 or CY = 0	x	x	x	V	x	
	[TDE-], [WHL-]	2	(TDE) - (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0 or CY = 0	x	x	x	V	x	
<b>CMPBKNC</b>	[TDE+], [WHL+]	2	(TDE) - (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1 End if C = 0 or CY = 1	x	x	x	V	x	
	[TDE-], [WHL-]	2	(TDE) - (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1 End if C = 0 or CY = 1	x	x	x	V	x	

## 6.3 アドレッシング別命令一覧

(1) 8ビット命令 ( ) 内は, rとしてAを記述することで実現している組み合わせです

MOV, XCH, ADD, ADDC, SUB, SUBC, AND OR XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, SHR, SHL, ROR4, ROL4, DBNZ, PUSH, POP, MOV, XCHM, CMPME, CMPMNE, CMPMNC, CMPMC, MOV, XCHBK, CMPBKE, CMPBKNE, CMPBKNC, CMPBKC, CHKL, CHKLA

表6-1 8ビット・アドレッシング別命令一覧表

第2オペランド 第1オペランド	#byte	A	r r'	saddr saddr'	sfr	!addr16 !!addr24	mem [ saddrp ] [ %saddrg ]	r3 PSWL PSWH	[ WHL + ] [ WHL - ]	n	なし <sup>注2</sup>
A	( MOV ) ADD <sup>注1</sup>	( MOV ) ( XCH ) ( ADD ) <sup>注1</sup>	MOV XCH ( ADD ) <sup>注1</sup>	( MOV ) <sup>注6</sup> ( XCH ) <sup>注6</sup> ( ADD ) <sup>注1, 6</sup>	MOV ( XCH ) ( ADD ) <sup>注1</sup>	( MOV ) ( XCH ) ADD <sup>注1</sup>	MOV XCH ADD <sup>注1</sup>	MOV	( MOV ) ( XCH ) ( ADD ) <sup>注1</sup>		
r	MOV ADD <sup>注1</sup>	( MOV ) ( XCH ) ( ADD ) <sup>注1</sup>	MOV XCH ADD <sup>注1</sup>	MOV XCH ADD <sup>注1</sup>	MOV XCH ADD <sup>注1</sup>	MOV XCH				ROR <sup>注3</sup>	MULU DIVUW INC DEC
saddr	MOV ADD <sup>注1</sup>	( MOV ) <sup>注6</sup> ( XCH ) ( ADD ) <sup>注1</sup>	MOV ADD <sup>注1</sup> XCH ADD <sup>注1</sup>	MOV XCH ADD <sup>注1</sup>							INC DEC DBNZ
sfr	MOV ADD <sup>注1</sup>	MOV ( ADD ) <sup>注1</sup>	MOV ADD <sup>注1</sup>								PUSH POP CHKL CHKLA
!addr16 !!addr24	MOV	( MOV ) ADD <sup>注1</sup>	MOV								
mem [ saddrp ] [ %saddrg ]		MOV ADD <sup>注1</sup>									
mem3											ROR4 ROL4
r3 PSWL PSWH	MOV	MOV									
B, C											DBNZ
STBC, WDM	MOV										
[ TDE + ] [ TDE - ]		( MOV ) ( ADD ) <sup>注1</sup> MOV <sup>注4</sup>							MOV <sup>注5</sup>		

注1 . ADDC, SUB, SUBC, AND, OR, XOR, CMPIはADDと同じ

2 . 第2オペランドがないか, 第2オペランドがオペランド・アドレスでない

3 . ROL, RORC, ROLC, SHR, SHLはRORと同じ

4 . XCHM, CMPME, CMPMNE, CMPMNC, CMPMCはMOVと同一

5 . XCHBK, CMPBKE, CMPBKNE, CMPBKNC, CMPBKCはMOVと同一

6 . この組み合わせでsaddrがsaddr2の場合, 短いコード長の命令がある

(2) 16ビット命令 ( ) 内は, rpとしてAXを記述することで実現している組み合わせです)

MOVW, XCHW, ADDW, SUBW, CMPW, MULUW, MULW, DIVUX, INCW, DECW, SHRW, SHLW, PUSH,  
POP, ADDWG, SUBWG, PUSHU, POPU, MOVTLW, MACW, MACSW, SACW

表6 - 2 16ビット・アドレッシング別命令一覧表

第2オペランド 第1オペランド	#word	AX	rp rp'	saddrp saddrp'	sfrp	!addr16 !!addr24	mem [ saddrp ] [ %saddrg ]	[ WHL + ]	byte	n	なし <sup>注2</sup>
AX	(MOVW) ADDW <sup>注1</sup>	(MOVW) (XCHW) (ADD) <sup>注1</sup>	(MOVW) (XCHW) (ADDW) <sup>注1</sup>	(MOVW) <sup>注3</sup> (XCHW) <sup>注3</sup> (ADD) <sup>注1, 3</sup>	MOVW (XCHW) (ADDW) <sup>注1</sup>	(MOVW) XCHW	MOVW XCHW	(MOVW) (XCHW)			
rp	MOVW ADDW <sup>注1</sup>	(MOVW) (XCHW) (ADDW) <sup>注1</sup>	MOVW XCHW ADDW <sup>注1</sup>	MOVW XCHW ADDW <sup>注1</sup>	MOVW XCHW ADDW <sup>注1</sup>	MOVW				SHRW SHLW	MULW <sup>注4</sup> INCW DECW
saddrp	MOVW ADDW <sup>注1</sup>	(MOVW) <sup>注3</sup> (ADDW) <sup>注1</sup>	MOVW ADDW <sup>注1</sup>	MOVW XCHW ADDW <sup>注1</sup>							INCW DECW
sfrp	MOVW ADDW <sup>注1</sup>	MOVW (ADDW) <sup>注1</sup>	MOVW ADDW <sup>注1</sup>								PUSH POP
!addr16 !!addr24	MOVW	(MOVW)	MOVW						MOVW		
mem [ saddrp ] [ %saddrg ]		MOVW									
PSW											PUSH POP
SP	ADDWG SUBWG										
post											PUSH POP PUSHU POPU
[ TDE + ]		(MOVW)						SACW			
byte											MACW MACSW

注1 . SUBW, CMPWはADDWと同じ

2 . 第2オペランドがないか, 第2オペランドがオペランド・アドレスでない

3 . この組み合わせでsaddrpがsaddrp2の場合, 短いコード長の命令がある

4 . MULUW, DIVUXはMULWと同じ

(3) 24ビット命令 ( ( ) 内は, rgとしてWHLを記述することで実現している組み合わせです)

MOVG, ADDG, SUBG, INCG, DECG, PUSH, POP

表6 - 3 24ビット・アドレッシング別命令一覧表

第2オペランド 第1オペランド	#imm24	WHL	rg rg'	saddrg	!!addr24	mem1	[ %saddrg ]	SP	なし <sup>注</sup>
WHL	(MOVG) (ADDG) (SUBG)	(MOVG) (ADDG) (SUBG)	(MOVG) (ADDG) (SUBG)	(MOVG) ADDG SUBG	(MOVG)	MOVG	MOVG	MOVG	
rg	MOVG ADDG SUBG	(MOVG) (ADDG) (SUBG)	MOVG ADDG SUBG	MOVG	MOVG				INCG DECG PUSH POP
saddrg		(MOVG)	MOVG						
!!addr24		(MOVG)	MOVG						
mem1		MOVG							
[ %saddrg ]		MOVG							
SP	MOVG	MOVG							INCG DECG

注 第2オペランドがないか, 第2オペランドがオペランド・アドレスでない

(4) ビット操作命令

NOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR, BFSET

表6 - 4 ビット操作命令アドレッシング別命令一覧表

第2オペランド 第1オペランド	CY	saddr.bit sfr.bit A.bit X.bit PSWL.bit PSWH.bit mem2.bit !addr16.bit !!addr24.bit	/saddr.bit /sfr.bit /A.bit /X.bit /PSWL.bit /PSWH.bit /mem2.bit /!addr16.bit /!!addr24.bit	なし <sup>注</sup>
CY		MOV1 AND1 OR1 XOR1	AND1 OR1	NOT1 SET1 CLR1
saddr.bit sfr.bit A.bit X.bit PSWL.bit PSWH.bit mem2.bit !addr16.bit !!addr24.bit	MOV1			NOT1 SET1 CLR1 BF BT BTCLR BFSET

注 第2オペランドがないか、第2オペランドがオペランド・アドレスでない

(5) コール・リターン命令 / 分岐命令

CALL, CALLF, CALLT, BRK, RET, RETI, RETB, RETCS, RETCSB, BRKCS, BR, BNZ, BNE, BZ, BE, BNC, BNL, BC, BL, BNV, BPO, BV, BPE, BP, BN, BLT, BGE, BLE, BGT, BNH, BH, BF, BT, BTCLR, BFSET, DBNZ

表6 - 5 コール・リターン命令 / 分岐命令アドレッシング別命令一覧表

命令アドレス のオペランド	\$addr20	\$addr20	!addr16	!!addr20	rp	rg	[ rp ]	[ rg ]	!addr11	[ addr5 ]	RBn	なし
基本命令	BC <sup>注</sup> BR	CALL BR	CALL BR RETCS RETCSB	CALL BR	CALL BR	CALL BR	CALL BR	CALL BR	CALLF	CALLT	BRKCS	BRK RET RETI RETB
複合命令	BF BT BTCLR BFSET DBNZ											

注 BNZ, BNE, BZ, BE, BNC, BNL, BL, BNV, BPO, BV, BPE, BP, BN, BLT, BGE, BLE, BGT, BNH, BHはBCと同じ

(6) その他の命令

ADJBA, ADJBS, CVTBW, LOCATION, SEL, NOT, EI, DI, SWRS

## 6.4 命令コード

## 6.4.1 命令コードの記号

(1) r1

R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	r1
0	0	0	R0
0	0	1	R1
0	1	0	R2
0	1	1	R3
1	0	0	R4
1	0	1	R5
1	1	0	R6
1	1	1	R7

(2) r2

R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	r2
0	0	0	R8
0	0	1	R9
0	1	0	R10
0	1	1	R11
1	0	0	R12
1	0	1	R13
1	1	0	R14
1	1	1	R15

(3) r, r'

R <sub>3</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	r
R <sub>7</sub>	R <sub>6</sub>	R <sub>5</sub>	R <sub>4</sub>	r'
0	0	0	0	R0
0	0	0	1	R1
0	0	1	0	R2
0	0	1	1	R3
0	1	0	0	R4
0	1	0	1	R5
0	1	1	0	R6
0	1	1	1	R7
1	0	0	0	R8
1	0	0	1	R9
1	0	1	0	R10
1	0	1	1	R11
1	1	0	0	R12
1	1	0	1	R13
1	1	1	0	R14
1	1	1	1	R15

(4) rp

P <sub>7</sub>	P <sub>6</sub>	P <sub>5</sub>	rp
0	0	0	RP0
0	0	1	RP1
0	1	0	RP2
0	1	1	RP3
1	0	0	RP4
1	0	1	RP5
1	1	0	RP6
1	1	1	RP7

(5) rp, rp'

P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>	rp
			rp'
0	0	0	RP0
0	0	1	RP4
0	1	0	RP1
0	1	1	RP5
1	0	0	RP2
1	0	1	RP6
1	1	0	RP3
1	1	1	RP7

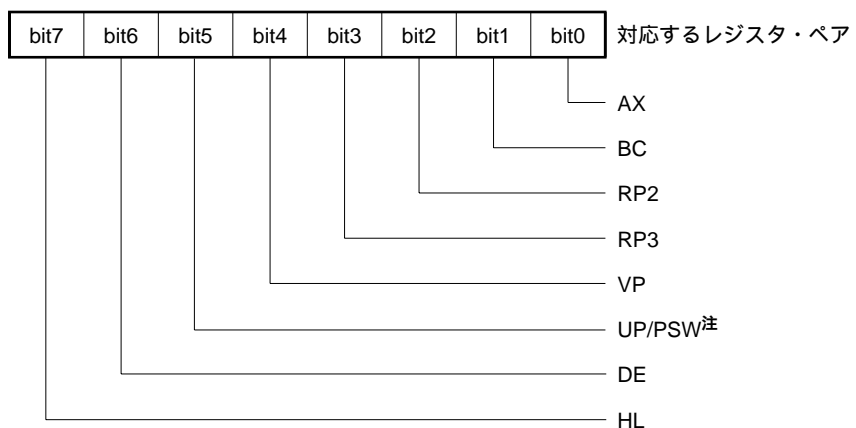
(6) rg, rg'

G <sub>6</sub>	G <sub>5</sub>	rg
G <sub>2</sub>	G <sub>1</sub>	rg'
0	0	RG4
0	1	RG5
1	0	RG6
1	1	RG7

(7) mem3

P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>	mem3
0	0	0	[ RP0 ]
0	0	1	[ RG4 ]
0	1	0	[ RP1 ]
0	1	1	[ RG5 ]
1	0	0	[ RP2 ]
1	0	1	[ RG6 ]
1	1	0	[ RP3 ]
1	1	1	[ RG7 ]

(8) post byte



0	スタック・メモリと退避/復帰動作をしない
1	スタック・メモリと退避/復帰動作をする

★

注 PUSH/POP命令の場合はUP, PUSHU/POPU命令の場合はPSWになります。

(9) locaddr

locaddr	locaddrl	locaddrh
0	FEH	01H
0FH	FFH	00H

6.4.2 命令コード一覧

(1) 8ビット・データ転送命令 : MOV

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOV	r1, #byte	1011 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	#byte	
	r2, #byte	0011 1100	1011 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	#byte
	saddr2, #byte	0011 1010	Saddr2-offset	#byte
	saddr1, #byte	0011 1100	0011 1010	Saddr1-offset
		#byte		
	sfr, #byte	0010 1011	Sfr-offset	#byte
	!addr16, #byte	0000 1001	0100 0000	Low Address
		High Address	#byte	
	!!addr24, #byet	0000 1001	0101 0000	High-w Address
		Low Address	High Address	#byte
	r, r1	0010 0100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r, r2	0011 1100	0010 0100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	A, r1	1101 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
	A, r2	0011 1100	1101 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	A, saddr2	0010 0000	Saddr2-offset	
	r, saddr2	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0000	Saddr2-offset
	r, saddr1	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0001	Saddr1-offset
	saddr2, A	0010 0010	Saddr2-offset	
	saddr2, r	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0100	Saddr2-offset
	saddr1, r	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0101	Saddr1-offset
	A, sfr	0001 0000	Sfr-offset	
	r, sfr	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0010	Sfr-offset
	sfr, A	0001 0010	Sfr-offset	
	sfr, r	0011 1000	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0110	Sfr-offset
	saddr2, saddr2'	0010 1010	0000 0000	Saddr2'-offset
		Saddr2-offset		
	saddr2, saddr1	0010 1010	0001 0000	Saddr1-offset
		Saddr2-offset		
	saddr1, saddr2	0010 1010	0010 0000	Saddr2-offset
		Saddr1-offset		
	saddr1, saddr1'	0010 1010	0011 0000	Saddr1'-offset
		Saddr1-offset		
r, !addr16	0011 1110	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0000	Low Address	
	High Address			
!addr16, r	0011 1110	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0001	Low Address	
	High Address			

(次ページに続く)

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOV	r, !!addr24	0011 1110 ----- Low Address	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0010 ----- High Address	High-w Address
	!!addr24, r	0011 1110 ----- Low Address	R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub> 0011 ----- High Address	High-w Address
	A, [ saddrp2 ]	0001 1000	Saddr2-offset	
	A, [ saddrp1 ]	0011 1100	0001 1000	Saddr1-offset
	A, [ %saddrg2 ]	0000 0111	0011 0000	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 0000
	A, [ TDE + ]	0101 1000		
	A, [ WHL + ]	0101 1001		
	A, [ TDE - ]	0101 1010		
	A, [ WHL - ]	0101 1011		
	A, [ TDE ]	0101 1100		
	A, [ WHL ]	0101 1101		
	A, [ VVP ]	0001 0110	0110 0000	
	A, [ UUP ]	0001 0110	0111 0000	
	A, [ TDE + byte ]	0000 0110	0000 0000	Low Offset
	A, [ SP + byte ]	0000 0110	0001 0000	Low Offset
	A, [ WHL + byte ]	0000 0110	0010 0000	Low Offset
	A, [ UUP + byte ]	0000 0110	0011 0000	Low Offset
	A, [ VVP + byte ]	0000 0110	0100 0000	Low Offset
	A, imm24 [ DE ]	0000 1010 ----- High Offset	0000 0000 ----- High-w Offset	Low Offset
	A, imm24 [ A ]	0000 1010 ----- High Offset	0001 0000 ----- High-w Offset	Low Offset
	A, imm24 [ HL ]	0000 1010 ----- High Offset	0010 0000 ----- High-w Offset	Low Offset
	A, imm24 [ B ]	0000 1010 ----- High Offset	0011 0000 ----- High-w Offset	Low Offset
	A, [ TDE + A ]	0001 0111	0000 0000	
	A, [ WHL + A ]	0001 0111	0001 0000	
	A, [ TDE + B ]	0001 0111	0010 0000	
	A, [ WHL + B ]	0001 0111	0011 0000	
	A, [ VVP + DE ]	0001 0111	0100 0000	
	A, [ VVP + HL ]	0001 0111	0101 0000	
	A, [ TDE + C ]	0001 0111	0110 0000	
	A, [ WHL + C ]	0001 0111	0111 0000	
	[ saddrp2 ], A	0001 1001	Saddr2-offset	

(次ページに続く)

二モニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOV	[ saddrp1 ], A	0011 1100	0001 1001	Saddr1-offset
	[ %saddrg2 ], A	0000 0111	1011 0000	Saddr2-offset
	[ %saddrg1 ], A	0011 1100	0000 0111	1011 0000
		Saddr1-offset		
	[ TDE + ], A	0101 0000		
	[ WHL + ], A	0101 0001		
	[ TDE - ], A	0101 0010		
	[ WHL - ], A	0101 0011		
	[ TDE ], A	0101 0100		
	[ WHL ], A	0101 0101		
	[ VVP ], A	0001 0110	1110 0000	
	[ UUP ], A	0001 0110	1111 0000	
	[ TDE + byte ], A	0000 0110	1000 0000	Low Offset
	[ SP + byte ], A	0000 0110	1001 0000	Low Offset
	[ WHL + byet ], A	0000 0110	1010 0000	Low Offset
	[ UUP + byte ], A	0000 0110	1011 0000	Low Offset
	[ VVP + byte ], A	0000 0110	1100 0000	Low Offset
	imm24 [ DE ], A	0000 1010	1000 0000	Low Offset
		High Offset	High-w Offset	
	imm24 [ A ], A	0000 1010	1001 0000	Low Offset
		High Offset	High-w Offset	
	imm24 [ HL ], A	0000 1010	1010 0000	Low Offset
		High Offset	High-w Offset	
	imm24 [ B ], A	0000 1010	1011 0000	Low Offset
		High Offset	High-w Offset	
	[ TDE + A ], A	0001 0111	1000 0000	
	[ WHL + A ], A	0001 0111	1001 0000	
	[ TDE + B ], A	0001 0111	1010 0000	
	[ WHL + B ], A	0001 0111	1011 0000	
	[ VVP + DE ], A	0001 0111	1100 0000	
	[ VVP + HL ], A	0001 0111	1101 0000	
	[ TDE + C ], A	0001 0111	1110 0000	
	[ WHL + C ], A	0001 0111	1111 0000	
	PSWL, #byte	0010 1011	1111 1110	#byte
PSWH, #byte	0010 1011	1111 1111	#byte	
PSWL, A	0001 0010	1111 1110		
PSWH, A	0001 0010	1111 1111		
A, PSWL	0001 0000	1111 1110		
A, PSWH	0001 0000	1111 1111		

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MOV</b>	V, #byte	0000 0111	0110 0001	#byte
	U, #byte	0000 0111	0110 0011	#byte
	T, #byte	0000 0111	0110 0101	#byte
	W, #byte	0000 0111	0110 0111	#byte
	A, V	0000 0101	1100 0001	
	A, U	0000 0101	1100 0011	
	A, T	0000 0101	1100 0101	
	A, W	0000 0101	1100 0111	
	V, A	0000 0101	1100 1001	
	U, A	0000 0101	1100 1011	
	T, A	0000 0101	1100 1101	
	W, A	0000 0101	1100 1111	

(2) 16ビット・データ転送命令：MOVW

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOVW	rp, #word	0110 0P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	Low Byte	High Byte
	saddrp2, #word	0000 1100 ----- High Byte	Saddr2-offset	Low Byte
	saddrp1, #word	0011 1100 ----- Low Byte	0000 1100 ----- High Byte	Saddr1-offset
	sfrp, #word	0000 1011 ----- High Byte	Sfr-offset	Low Byte
	!addr16, #word	0000 1001 ----- High Address	0100 0001 ----- Low Byte	Low Address High Byte
	!!addr24, #word	0000 1001 ----- Low Address ----- High Byte	0101 0001 ----- High Address	High-w Address Low Byte
	rp, rp'	0010 0100	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AX, saddrp2	0001 1100	Saddr2-offset	
	rp, saddrp2	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Saddr2-offset
	rp, saddrp1	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Saddr1-offset
	saddrp2, AX	0001 1010	Saddr2-offset	
	saddrp2, rp	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1100	Saddr2-offset
	saddrp1, rp	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1101	Saddr1-offset
	AX, sfrp	0001 0001	Sfr-offset	
	rp, sfrp	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010	Sfr-offset
	sfrp, AX	0001 0011	Sfr-offset	
	sfrp, rp	0011 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1110	Sfr-offset
	saddrp2, saddrp2'	0010 1010 ----- Saddr2-offset	1000 0000	Saddr2'-offset
	saddrp2, saddrp1	1001 01010 ----- Saddr2-offset	1001 0000	Saddr1-offset
	saddrp1, saddrp2	0010 1010 ----- Saddr1-offset	1010 0000	Saddr2-offset
	saddrp1, saddrp1'	0010 1010 ----- Saddr1-offset	1011 0000	Saddr1'-offset
	rp, !addr16	0011 1110 ----- High Address	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Low Address
	!addr16, rp	0011 1110 ----- High Address	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Low Address
	rp, !!addr24	0011 1110 ----- Low Address	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010 ----- High Address	High-w Address
	!!addr24, rp	0011 1110 ----- Low Address	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1011 ----- High Address	High-w Address

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOVW	AX, [ saddrp2 ]	0000 0111	0010 0001	Saddr2-offset
	AX, [ saddrp1 ]	0011 1100	0000 0111	0010 0001
		Saddr1-offset		
	AX, [ %saddrg2 ]	0000 0111	0011 0001	Saddr2-offset
	AX, [ %saddrg1 ]	0011 1100	0000 0111	0011 0001
		Saddr1-offset		
	AX, [ TDE + ]	0001 0110	0000 0001	
	AX, [ WHL + ]	0001 0110	0001 0001	
	AX, [ TDE - ]	0001 0110	0010 0001	
	AX, [ WHL - ]	0001 0110	0011 0001	
	AX, [ TDE ]	0001 0110	0100 0001	
	AX, [ WHL ]	0001 0110	0101 0001	
	AX, [ VVP ]	0001 0110	0110 0001	
	AX, [ UUP ]	0001 0110	0111 0001	
	AX, [ TDE + byte ]	0000 0110	0000 0001	Low Offset
	AX, [ SP + byte ]	0000 0110	0001 0001	Low Offset
	AX, [ WHL + byte ]	0000 0110	0010 0001	Low Offset
	AX, [ UUP + byte ]	0000 0110	0011 0001	Low Offset
	AX, [ VVP + byte ]	0000 0110	0100 0001	Low Offset
	AX, imm24 [ DE ]	0000 1010	0000 0001	Low Offset
		High Offset	High-w Offset	
	AX, imm24 [ A ]	0000 1010	0001 0001	Low Offset
		High Offset	High-w Offset	
	AX, imm24 [ HL ]	0000 1010	0010 0001	Low Offset
		Hihg Offset	High-w Offset	
	AX, imm24 [ B ]	0000 1010	0011 0001	Low Offset
		High Offset	High-w Offset	
	AX, [ TDE + A ]	0001 0111	0000 0001	
	AX, [ WHL + A]	0001 0111	0001 0001	
	AX, [ TDE + B]	0001 0111	0010 0001	
	AX, [ WHL + B]	0001 0111	0011 0001	
	AX, [ VVP + DE]	0001 0111	0100 0001	
	AX, [ VVP + HL]	0001 0111	0101 0001	
AX, [ TDE + C]	0001 0111	0110 0001		
AX, [ WHL + C]	0001 0111	0111 0001		
[ saddrp2 ], AX	0000 0111	1010 0001	Saddr2-offset	
[ sadrp1 ], AX	0011 1100	0000 0111	1010 0001	
	Saddr1-offset			
[ %saddrg2 ], AX	0000 0111	1011 0001	Saddr2-offset	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MOVW</b>	[ %saddr1 ], AX	0011 1100	0000 0111	1011 0001
		----- Saddr1-offset		
	[ TDE + ], AX	0001 0110	1000 0001	
	[ WHL + ], AX	0001 0110	1001 0001	
	[ TDE - ], AX	0001 0110	1010 0001	
	[ WHL - ], AX	0001 0110	1011 0001	
	[ TDE ], AX	0001 0110	1100 0001	
	[ WHL ], AX	0001 0110	1101 0001	
	[ VVP ], AX	0001 0110	1110 0001	
	[ UUP ], AX	0001 0110	1111 0001	
	[ TDE + byte ], AX	0000 0110	1000 0001	Low Offset
	[ SP + byte ], AX	0000 0110	1001 0001	Low Offset
	[ WHL + byte ], AX	0000 0110	1010 0001	Low Offset
	[ UUP + byte ], AX	0000 0110	1011 0001	Low Offset
	[ VVP + byte ], AX	0000 0110	1100 0001	Low Offset
	imm24 [ DE ], AX	0000 1010	1000 0001	Low Offset
		----- High Offset		
	imm24 [ A ], AX	0000 1010	1001 0001	Low Offsdt
		----- High Offset		
			High-w Offset	
	imm24 [ HL ], AX	0000 1010	1010 0001	Low Offset
		----- High Offset		
			High-w Offset	
	imm24 [ B ], AX	0000 1010	1011 0001	Low Offset
		----- High Offset		
			High-w Offset	
	[ TDE + A ], AX	0001 0111	1000 0001	
	[ WHL + A ], AX	0001 0111	1001 0001	
[ TDE + B ], AX	0001 0111	1010 0001		
[ WHL + B ], AX	0001 0111	1011 0001		
[ VVP + DE ], AX	0001 0111	1100 0001		
[ VVP + HL ], AX	0001 0111	1101 0001		
[ TDE + C ], AX	0001 0111	1110 0001		
[ WHL + C ], AX	0001 0111	1111 0001		

(3) 24ビット・データ転送命令：MOVG

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOVG	rg, #imm24	0011 1000	1G <sub>6</sub> G <sub>5</sub> 1 1011	Low Byte
		----- High Byte	----- High-w Byte	-----
	rg, rg'	0010 0100	1G <sub>6</sub> G <sub>5</sub> 1 1G <sub>2</sub> G <sub>1</sub> 1	
	rg, !!addr24	0011 1110	1G <sub>6</sub> G <sub>5</sub> 1 1010	High-w Address
		----- Low Address	----- High Address	-----
	!!addr24, rg	0011 1110	1G <sub>6</sub> G <sub>5</sub> 1 1011	High-w Address
		----- Low Address	----- High Address	-----
	rg, saddrg2	0011 1000	1G <sub>6</sub> G <sub>5</sub> 1 1000	Saddr2-offset
	rg, saddrg1	0011 1000	1G <sub>6</sub> G <sub>5</sub> 1 1001	Saddr1-offset
	saddrg2, rg	0011 1000	1G <sub>6</sub> G <sub>5</sub> 1 1100	Saddr2-offset
	saddrg1, rg	0011 1000	1G <sub>6</sub> G <sub>5</sub> 1 1101	saddr1-offset
	WHL, [ %saddrg2 ]	0000 0111	0011 0010	Saddr2-offset
	WHL, [ %saddrg1 ]	0011 1100	0000 0111	0011 0010
		----- Saddr1-offset	-----	-----
	[ %saddrg2 ], WHL	0000 0111	1011 0010	Ssddr2-offset
	[ %saddrg1 ], WHL	0011 1100	0000 0111	1011 0010
		----- Saddr1-offset	-----	-----
	WHL, [ TDE + ]	0001 0110	0000 0010	
	WHL, [ TDE - ]	0001 0110	0010 0010	
	WHL, [ TDE ]	0001 0110	0100 0010	
	WHL, [ WHL ]	0001 0110	0101 0010	
	WHL, [ VVP ]	0001 0110	0110 0010	
	WHL, [ UUP ]	0001 0110	0111 0010	
	WHL, [ TDE + byte ]	0000 0110	0000 0010	Low Offset
	WHL, [ SP + byte ]	0000 0110	0001 0010	Low Offset
	WHL, [ WHL + byte ]	0000 0110	0010 0010	Low Offset
	WHL, [ UUP + byte ]	0000 0110	0011 0010	Low Offset
	WHL, [ VVP + byte ]	0000 0110	0100 0010	Low Offset
	WHL, imm24 [ DE ]	0000 1010	0000 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
	WHL, imm24 [ A ]	0000 1010	0001 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
WHL, imm24 [ HL ]	0000 1010	0010 0010	Low Offset	
	----- High Offset	----- High-w Offset	-----	
WHL, imm24 [ B ]	0000 1010	0011 0010	Low Offset	
	----- High Offset	----- High-w Offset	-----	
WHL, [ TDE + A ]	0001 0111	0000 0010		
WHL, [ WHL + A ]	0001 0111	0001 0010		
WHL, [ TDE + B ]	0001 0111	0010 0010		

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MOVG</b>	WHL, [ WHL + B ]	0001 0111	0011 0010	
	WHL, [ VVP + DE ]	0001 0111	0100 0010	
	WHL, [ VVP + HL ]	0001 0111	0101 0010	
	WHL, [ TDE + C ]	0001 0111	0110 0010	
	WHL, [ WHL + C ]	0001 0111	0111 0010	
	[ TDE + ], WHL	0001 0110	1000 0010	
	[ TDE - ], WHL	0001 0110	1010 0010	
	[ TDE ], WHL	0001 0110	1100 0010	
	[ WHL ], WHL	0001 0110	1101 0010	
	[ VVP ], WHL	0001 0110	1110 0010	
	[ UUP ], WHL	0001 0110	1111 0010	
	[ TDE + byte ], WHL	0000 0110	1000 0010	Low Offset
	[ SP + byte ], WHL	0000 0110	1001 0010	Low Offset
	[ WHL + byte ], WHL	0000 0110	1010 0010	Low Offset
	[ UUP + byte ], WHL	0000 0110	1011 0010	Low Offset
	[ VVP + byte ], WHL	0000 0110	1100 0010	Low Offset
	imm24 [ DE ], WHL	0000 1010	1000 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
	imm24 [ A ], WHL	0000 1010	1001 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
	imm24 [ HL ], WHL	0000 1010	1010 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
	imm24 [ B ], WHL	0000 1010	1011 0010	Low Offset
		----- High Offset	----- High-w Offset	-----
	[ TDE + A ], WHL	0001 0111	1000 0010	
	[ WHL + A ], WHL	0001 0111	1001 0010	
	[ TDE + B ], WHL	0001 0111	1010 0010	
	[ WHL + B ], WHL	0001 0111	1011 0010	
	[ VVP + DE ], WHL	0001 0111	1100 0010	
	[ VVP + HL ], WHL	0001 0111	1101 0010	
[ TDE + C ], WHL	0001 0111	1110 0010		
[ WHL + C ], WHL	0001 0111	1111 0010		

(4) 8ビット・データ交換命令 : XCH

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
XCH	r, r1	0010 0101	R7R6R5R4 0R2R1R0	
	r, r2	0011 1100	0010 0101	R7R6R5R4 0R2R1R0
	A, r1	1101 1R2R1R0		
	A, r2	0011 1100	1101 1R2R1R0	
	A, saddr2	0010 0001	Saddr2-offset	
	r, saddr2	0011 1001	R7R6R5R4 0000	Saddr2-offset
	r, saddr1	0011 1001	R7R6R5R4 0001	Saddr1-offset
	r, sfr	0011 1001	R7R6R5R4 0010	Sfr-offset
	saddr2, sassr2'	0010 1010	0000 0100	Saddr2'-offset
		Saddr2-offset		
	saddr2, saddr1	0010 1010	0001 0100	Saddr1-offset
		Saddr2-offset		
	saddr1, saddr2	0010 1010	0010 0100	Saddr2-offset
		Saddr1-offset		
	saddr1, saddr1'	0010 1010	0011 0100	Saddr1'-offset
		Saddr1-offset		
	r, !addr16	0011 1110	R7R6R5R4 0100	Low Address
		High Address		
	r, !!addr24	0011 1110	R7R6R5R4 0110	High-w Address
		Low Address	High Address	
	A, [ saddrp2]	0010 0011	Saddr2-offset	
	A, [ saddrp1]	0011 1100	0010 0011	Saddr1-offset
	A, [ %saddrg2]	0000 0111	0011 0100	Saddr2-offset
	A, [ %saddrg1]	0011 1100	0000 0111	0011 0100
		Saddr1-offset		
	A, [ TDE + ]	0001 0110	0000 0100	
	A, [ WHL + ]	0001 0110	0001 0100	
	A, [ TDE - ]	0001 0110	0010 0100	
	A, [ WHL - ]	0001 0110	0011 0100	
	A, [ TDE]	0001 0110	0100 0100	
	A, [ WHL]	0001 0110	0101 0100	
	A, [ VVP]	0001 0110	0110 0100	
A, [ UUP]	0001 0110	0111 0100		
A, [ TDE + byte]	0000 0110	0000 0100	Low Offset	
A, [ SP + byte]	0000 0110	0001 0100	Low Offset	
A, [ WHL + byte]	0000 0110	0010 0100	Low Offset	
A, [ UUP + byte]	0000 0110	0011 0100	Low Offset	
A, [ VVP + byte]	0000 0110	0100 0100	Low Offset	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
XCH	A, imm24 [ DE]	0000 1010 ----- High Offset	0000 0100 ----- High-w Offset	Low Offset -----
	A, imm24 [ A]	0000 1010 ----- High Offset	0001 0100 ----- High-w Offset	Low Offset -----
	A, imm24 [ HL]	0000 1010 ----- High Offset	0010 0100 ----- High-w Offset	Low Offset -----
	A, imm24 [ B]	0000 1010 ----- High Offset	0011 0100 ----- High-w Offset	Low Offset -----
	A, [ TDE + A]	0001 0111	0000 0100	
	A, [ WHL + A]	0001 0111	0001 0100	
	A, [ TDE + B]	0001 0111	0010 0100	
	A, [ WHL + B]	0001 0111	0011 0100	
	A, [ VVP + DE]	0001 0111	0100 0100	
	A, [ VVP + HL]	0001 0111	0101 0100	
	A, [ TDE + C]	0001 0111	0110 0100	
	A, [ WHL + C]	0001 0111	0111 0100	

(5) 16ビット・データ交換命令：XCHW

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
XCHW	rp, rp'	0010 0101	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AX, saddrp2	0001 1011	Saddr2-offset	
	rp, saddrp2	0011 1001	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Saddr2-offset
	rp, saddrp1	0011 1001	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Saddr1-offset
	rp, sfrp	0011 1001	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010	Sfr-offset
	AX, [ saddrp2 ]	0000 0111	0010 0101	Saddr2-offset
	AX, [ Saddrp1 ]	0011 1100	0000 0111	0010 0101
		Saddr1-offset		
	AX, [%saddrg2]	0000 0111	0011 0101	Saddr2-offset
	AX, [%saddrg1]	0011 1100	0000 0111	0011 0101
		Saddr1-offset		
	AX, !addr16	0000 1010	0100 0101	Low Address
		High Address		
	AX, !!addr24	0000 1010	0101 0101	High-w Address
		Low Address	High Address	
	saddrp2, saddrp2'	0010 1010	1000 0100	Saddr2'-offset
		Saddr2-offset		
	saddrp2, saddrp1	0010 1010	1001 0100	Saddr1-offset
		Saddr2-offset		
	saddrp1, saddrp2	0010 1010	1010 0100	Saddr2-offset
		Saddr1-offset		
	Saddr1, Saddr1'	0010 1010	1011 0100	Saddr1'- offset
		Saddr1-offset		
	AX, [ TDE + byte]	0000 0110	0000 0101	Low Offset
	AX, [ SP + byte]	0000 0110	0001 0101	Low Offset
	AX, [ WHL + byte]	0000 0110	0010 0101	Low Offset
	AX, [ UUP + byte]	0000 0110	0011 0101	Low Offset
	AX, [ VVP + byte]	0000 0110	0100 0101	Low Offset
	AX, imm24 [ DE]	0000 1010	0000 0101	Low Offset
		High Offset	High-w Offset	
AX, imm24 [ A]	0000 1010	0001 0101	Low Offset	
	High Offset	High-w Offset		
AX, imm24 [ HL]	0000 1010	0010 0101	Low Offset	
	High Offset	High-w Offset		
AX, imm24 [ B]	0000 1010	0011 0101	Low Offset	
	High Offset	High-w Offset		
AX, [ TDE + ]	0001 0110	0000 0101		
AX, [ WHL + ]	0001 0110	0001 0101		
AX, [ TDE - ]	0001 0110	0010 0101		

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>XCHW</b>	AX, [ WHL - ]	0001 0110	0011 0101	
	AX, [ TDE]	0001 0110	0100 0101	
	AX, [ WHL]	0001 0110	0101 0101	
	AX, [ VVP]	0001 0110	0110 0101	
	AX, [ UUP]	0001 0110	0111 0101	
	AX, [ TDE + A]	0001 0111	0000 0101	
	AX, [ WHL + A]	0001 0111	0001 0101	
	AX, [ TDE + B]	0001 0111	0010 0101	
	AX, [ WHL + B]	0001 0111	0011 0101	
	AX, [ VVP + DE]	0001 0111	0100 0101	
	AX, [ VVP + HL]	0001 0111	0101 0101	
	AX, [ TDE + C]	0001 0111	0110 0101	
	AX, [ WHL + C]	0001 0111	0111 0101	

(6) 8ビット演算命令：ADD, ADDC, SUB, SUBC, CMP, AND, OR, XOR

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
ADD	A, #byte	1010 1000	#byte	
	r, #byte	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0011	#byte
	saddr2, #byte	0110 1000	Saddr2-offset	#byte
	saddr1, #byte	0011 1100	0110 1000	saddr1-offset
		#byte		
	sfr, #byte	0000 0001	0110 1000	Sfr-offset
		#byte		
	r, r1	1000 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r, r2	0011 1100	1000 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	A, saddr2	1001 1000	Saddr2-offset	
	r, saddr2	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0000	Saddr2-offset
	r, saddr1	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0001	Saddr1-offset
	saddr2, r	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0100	Saddr2-offset
	saddr1, r	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0101	Saddr1-offset
	r, sfr	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0010	Sfr-offset
	sfr, r	0111 1000	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0110	Sfr-offset
	saddr2, saddr2'	0010 1010	0000 1000	Saddr2'-offset
		Saddr2-offset		
	saddr2, saddr1	0010 1010	0001 1000	Saddr1-offset
		Saddr2-offset		
	saddr1, saddr2	0010 1010	0010 1000	Saddr2-offset
		Saddr1-offset		
	saddr1, saddr1'	0010 1010	0011 1000	Saddr1'-offset
		Saddr1-offset		
	A, [ saddrp2 ]	0000 0111	0010 1000	Saddr2-offset
	A, [ saddrp1 ]	0011 1100	0000 0111	0010 1000
		Saddr1-offset		
	A, [ %saddrg2 ]	0000 0111	0011 1000	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100	0000 0111	0011 1000
		Saddr1-offset		
	[ saddrp2 ], A	0000 0111	1010 1000	Saddr2-offset
	[ saddrp1 ], A	0011 1100	0000 0111	1010 1000
	Saddr1-offset			
[ %saddrg2 ], A	0000 0111	1011 1000	Saddr2-offset	
[ %saddrg1 ], A	0011 1100	0000 0111	1011 1000	
	Saddr1 Offset			
A, !addr16	0000 1010	0100 1000	Low Address	
	High Address			

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
ADD	A, !addr24	0000 1010 ----- Low Address	0101 1000 ----- High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1000	Low Address
	!!addr24, A	0000 1010 ----- Low Address	1101 1000 ----- High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1000	
	A, [ WHL + ]	0001 0110	0001 1000	
	A, [ TDE - ]	0001 0110	0010 1000	
	A, [ WHL - ]	0001 0110	0011 1000	
	A, [ TDE]	0001 0110	0100 1000	
	A, [ WHL]	0001 0110	0101 1000	
	A, [ VVP]	0001 0110	0110 1000	
	A, [ UUP]	0001 0110	0111 1000	
	A, [ TDE + byte]	0000 0110	0000 1000	Low Offset
	A, [ SP + byte]	0000 0110	0001 1000	Low Offset
	A, [ WHL + byte]	0000 0110	0010 1000	Low Offset
	A, [ UUP + byte]	0000 0110	0011 1000	Low Offset
	A, [ VVP + byte]	0000 0110	0100 1000	Low Offset
	A, imm24 [ DE]	0000 1010 ----- High Offset	0000 1000 ----- High-w Offset	Low Offset
	A, imm24 [ A]	0000 1010 ----- High Offset	0001 1000 ----- High-w Offset	Low Offset
	A, imm24 [ HL]	0000 1010 ----- High Offset	0010 1000 ----- High-w Offset	Low Offset
	A, imm24 [ B]	0000 1010 ----- High Offset	0011 1000 ----- High-w Offset	Low Offset
	A, [ TDE + A]	0001 0111	0000 1000	
	A, [ WHL + A]	0001 0111	0001 1000	
	A, [ TDE + B]	0001 0111	0010 1000	
	A, [ WHL + B]	0001 0111	0011 1000	
	A, [ VVP + DE]	0001 0111	0100 1000	
	A, [ VVP + HL]	0001 0111	0101 1000	
	A, [ TDE + C]	0001 0111	0110 1000	
	A, [ WHL + C]	0001 0111	0111 1000	
	[ TDE + ], A	0001 0110	1000 1000	
	[ WHL + ], A	0001 0110	1001 1000	
	[ TDE - ], A	0001 0110	1010 1000	
	[ WHL - ], A	0001 0110	1011 1000	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADD</b>	[ TDE ], A	0001 0110	1100 1000	
	[ WHL ], A	0001 0110	1101 1000	
	[ VVP ], A	0001 0110	1110 1000	
	[ UUP ], A	0001 0110	1111 1000	
	[ TDE + byte ], A	0000 0110	1000 1000	Low Offset
	[ SP + byte ], A	0000 0110	1001 1000	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1000	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1000	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1000	Low Offset
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1000 ----- High-w Offset	Low Offset ----- -----
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1000 ----- High-w Offset	Low Offset ----- -----
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1000 ----- High-w Offset	Low Offset ----- -----
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1000 ----- High-w Offset	Low Offset ----- -----
	[ TDE + A ], A	0001 0111	1000 1000	
	[ WHL + A ], A	0001 0111	1001 1000	
	[ TDE + B ], A	0001 0111	1010 1000	
	[ WHL + B ], A	0001 0111	1011 1000	
	[ VVP + DE ], A	0001 0111	1100 1000	
	[ VVP + HL ], A	0001 0111	1101 1000	
	[ TDE + C ], A	0001 0111	1110 1000	
	[ WHL + C ], A	0001 0111	1111 1000	
	<b>ADDC</b>	A, #byte	1010 1001	#byte
r, #byte		0111 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0011	#byte
saddr2, #byte		0110 1001	Saddr2-offset	#byte
saddr1, #byte		0011 1100 ----- #byte	0110 1001	Saddr1-offset ----- -----
sfr, #byte		0000 0001 ----- #byte	0110 1001	Sfr-offset ----- -----
r, r1		1000 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
r, r2		0011 1100	1000 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
A, saddr2		1001 1001	Saddr2-offset	
r, saddr2		0111 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0000	Saddr2-offset
r, saddr1		0111 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0001	Saddr1-offset
saddr2, r		0111 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0100	Saddr2-offset
saddr1, r		0111 1001	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0101	Saddr1-offset

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADDC</b>	r, sfr	0111 1001	R7R6R5R4 0010	Sfr-offset
	sfr, r	0111 1001	R7R6R5R4 0110	Sfr-offset
	saddr2, saddr2'	0010 1010 ----- Saddr2-offset	0000 1001	Saddr2'-offset
	saddr2, saddr1	0010 1010 ----- Saddr2-offset	0001 1001	Saddr1-offset
	saddr1, saddr2	0010 1010 ----- Saddr1-offset	0010 1001	Saddr2-offset
	saddr1, saddr1'	0010 1010 ----- Saddr1-offset	0011 1001	Saddr1'-offset
	A, [ saddrp2 ]	0000 0111	0010 1001	Saddr2-offset
	A, [ saddrp1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0010 1001
	A, [ %saddrg2 ]	0000 0111	0011 1001	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1001
	[ saddrp2 ], A	0000 0111	1010 1001	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1001
	[ %saddrg2 ], A	0000 0111	1011 1001	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1001
	A, !addr16	0000 1010 ----- High Address	0100 1001	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1001 High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1001	Low Address
	!!addr24, A	0000 1010 ----- Low Address	1101 1001 High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1001	
	A, [ WHL + ]	0001 0110	0001 1001	
	A, [ TDE - ]	0001 0110	0010 1001	
	A, [ WHL - ]	0001 0110	0011 1001	
	A, [ TDE ]	0001 0110	0100 1001	
	A, [ WHL ]	0001 0110	0101 1001	
	A, [ VVP ]	0001 0110	0110 1001	
	A, [ UUP ]	0001 0110	0111 1001	
	A, [ TDE + byte ]	0000 0110	0000 1001	Low Offset

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
ADDC	A, [ SP + byte]	0000 0110	0001 1001	Low Offset
	A, [ WHL + byte]	0000 0110	0010 1001	Low Offset
	A, [ UUP + byte]	0000 0110	0011 1001	Low Offset
	A, [ VVP + byte]	0000 0110	0100 1001	Low Offset
	A, imm24 [ DE]	0000 1010	0000 1001	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ A]	0000 1010	0001 1001	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ HL]	0000 1010	0010 1001	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ B]	0000 1010	0011 1001	Low Offset
		High Offset	High-w Offset	
	A, [ TDE + A]	0001 0111	0000 1001	
	A, [ WHL + A]	0001 0111	0001 1001	
	A, [ TDE + B]	0001 0111	0010 1001	
	A, [ WHL + B]	0001 0111	0011 1001	
	A, [ VVP + DE]	0001 0111	0100 1001	
	A, [ VVP + HL]	0001 0111	0101 1001	
	A, [ TDE + C]	0001 0111	0110 1001	
	A, [ WHL + C]	0001 0111	0111 1001	
	[ TDE + ], A	0001 0110	1000 1001	
	[ WHL + ], A	0001 0110	1001 1001	
	[ TDE - ], A	0001 0110	1010 1001	
	[ WHL - ], A	0001 0110	1011 1001	
	[ TDE ], A	0001 0110	1100 1001	
	[ WHL ], A	0001 0110	1101 1001	
	[ VVP ], A	0001 0110	1110 1001	
	[ UUP ], A	0001 0110	1111 1001	
	[ TDE + byte ], A	0000 0110	1000 1001	Low Offset
	[ SP + byte ], A	0000 0110	1001 1001	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1001	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1001	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1001	Low Offset
	imm24 [ DE ], A	0000 1010	1000 1001	Low Offset
		High Offset	High-w Offset	
	imm24 [ A ], A	0000 1010	1001 1001	Low Offset
		High Offset	High-w Offset	
	imm24 [ HL ], A	0000 1010	1010 1001	Low Offset
		High Offset	High-w Offset	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADDC</b>	imm24 [ B ], A	0000 1010	1011 1001	Low Offset
		----- High Offset	----- High-w Offset	
	[ TDE + A ], A	0001 0111	1000 1001	
	[ WHL + A ], A	0001 0111	1001 1001	
	[ TDE + B ], A	0001 0111	1010 1001	
	[ WHL + B ], A	0001 0111	1011 1001	
	[ VVP + DE ], A	0001 0111	1100 1001	
	[ VVP + HL ], A	0001 0111	1101 1001	
	[ TDE + C ], A	0001 0111	1110 1001	
[ WHL + C ], A	0001 0111	1111 1001		
<b>SUB</b>	A, #byte	1010 1010	#byte	
	r, #byte	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0011	#byte
	saddr2, #byte	0110 1010	Saddr2-Offset	#byte
	saddr1, #byte	0011 1100	0110 1010	Saddr1-Offset
		----- #byte		
	sfr, #byte	0000 0001	0110 1010	Sfr-offset
		----- #byte		
	r, r1	1000 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r, r2	0011 1100	1000 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	A, saddr2	1001 1010	Saddr2-offset	
	r, saddr2	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0000	Saddr2-offset
	r, saddr1	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0001	Saddr1-offset
	saddr2, r	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0100	Saddr2-offset
	saddr1, r	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0101	Saddr1-offset
	r, sfr	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0010	Sfr-offset
	sfr, r	0111 1010	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0110	Sfr-offset
	saddr2, saddr2'	0010 1010	0000 1010	Saddr2'-offset
		----- Saddr2'-offset		
	saddr2, saddr1	0010 1010	0001 1010	Saddr1'-offset
		----- Saddr2'-offset		
	saddr1, saddr2	0010 1010	0010 1010	Saddr2'-offset
		----- Saddr1'-offset		
	saddr1, saddr1'	0010 1010	0011 1010	Saddr1'-offset
		----- Saddr1'-offset		
A, [ saddrp2 ]	0000 0111	0010 1010	Saddr2'-offset	
A, [ saddrp1 ]	0011 1100	0000 0111	0010 1010	
	----- Saddr1'-offset			
A, [ %saddrg2 ]	0000 0111	0011 1010	Saddr2'-offset	

( 次ページに続く )

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
SUB	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1010
	[ saddrp2 ], A	0000 0111	1010 1010	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1010
	[ %saddrg2 ], A	0000 0111	1011 1010	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1010
	A, !addr16	0000 1010 ----- High Address	0100 1010	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1010 ----- High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1010	Low Address
	!!addr24, A	0000 1010 ----- Low Address	1101 1010 ----- High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1010	
	A, [ WHL + ]	0001 0110	0001 1010	
	A, [ TDE - ]	0001 0110	0010 1010	
	A, [ WHL - ]	0001 0110	0011 1010	
	A, [ TDE]	0001 0110	0100 1010	
	A, [ WHL]	0001 0110	0101 1010	
	A, [ VVP]	0001 0110	0110 1010	
	A, [ UUP]	0001 0110	0111 1010	
	A, [ TDE + byte]	0000 0110	0000 1010	Low Offset
	A, [ SP + byte]	0000 0110	0001 1010	Low Offset
	A, [ WHL + byte]	0000 0110	0010 1010	Low Offset
	A, [ UUP + byte]	0000 0110	0011 1010	Low Offset
	A, [ VVP + byte]	0000 0110	0100 1010	Low Offset
	A, imm24 [ DE]	0000 1010 ----- High Offset	0000 1010 ----- High-w Offset	Low Offset
	A, imm24 [ A]	0000 1010 ----- High Offset	0001 1010 ----- High-w Offset	Low Offset
	A, imm24 [ HL]	0000 1010 ----- High Offset	0010 1010 ----- High-w Offset	Low Offset
	A, imm24 [ B]	0000 1010 ----- High Offset	0011 1010 ----- High-w Offset	Low Offset
	A, [ TDE + A]	0001 0111	0000 1010	
	A, [ WHL + A]	0001 0111	0001 1010	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>SUB</b>	A, [ TDE + B ]	0001 0111	0010 1010	
	A, [ WHL + B ]	0001 0111	0011 1010	
	A, [ VVP + DE ]	0001 0111	0100 1010	
	A, [ VVP + HL ]	0001 0111	0101 1010	
	A, [ TDE + C ]	0001 0111	0110 1010	
	A, [ WHL + C ]	0001 0111	0111 1010	
	[ TDE + ], A	0001 0110	1000 1010	
	[ WHL + ], A	0001 0110	1001 1010	
	[ TDE - ], A	0001 0110	1010 1010	
	[ WHL - ], A	0001 0110	1011 1010	
	[ TDE ], A	0001 0110	1100 1010	
	[ WHL ], A	0001 0110	1101 1010	
	[ VVP ], A	0001 0110	1110 1010	
	[ UUP ], A	0001 0110	1111 1010	
	[ TDE + byte ], A	0000 0110	1000 1010	Low Offset
	[ SP + byte ], A	0000 0110	1001 1010	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1010	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1010	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1010	Low Offset
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1010 ----- High-w Offset	Low Offset
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1010 ----- High-w Offset	Low Offset
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1010 ----- High-w Offset	Low Offset
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1010 ----- High-w Offset	Low Offset
	[ TDE + A ], A	0001 0111	1000 1010	
	[ WHL + A ], A	0001 0111	1001 1010	
	[ TDE + B ], A	0001 0111	1010 1010	
	[ WHL + B ], A	0001 0111	1011 1010	
	[ VVP + DE ], A	0001 0111	1100 1010	
	[ VVP + HL ], A	0001 0111	1101 1010	
	[ TDE + C ], A	0001 0111	1110 1010	
[ WHL + C ], A	0001 0111	1111 1010		
<b>SUBC</b>	A, #byte	1010 1011	#byte	
	r, #byte	0111 1011	R7R6R5R4 0011	#byte
	saddr2, #byte	0110 1011	Saddr2-offset	#byte

( 次ページに続く )

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
SUBC	saddr1, #byte	0011 1100 ----- #byte	0110 1011	Saddr1-offset
	sfr, #byte	0000 0001 ----- #byte	0110 1011	Sfr-offset
	r, r1	1000 1011	R7R6R5R4 0R2R1R0	
	r, r2	0011 1100	1000 1011	R7R6R5R4 0R2R1R0
	A, saddr2	1001 1011	Saddr2-offset	
	r, saddr2	0111 1011	R7R6R5R4 0000	Saddr2-offset
	r, saddr1	0111 1011	R7R6R5R4 0001	Saddr1-offset
	saddr2, r	0111 1011	R7R6R5R4 0100	Saddr2-offset
	saddr1, r	0111 1011	R7R6R5R4 0101	Saddr1-offset
	r, sfr	0111 1011	R7R6R5R4 0010	Sfr-offset
	sfr, r	0111 1011	R7R6R5R4 0110	Sfr-offset
	saddr2, saddr2'	0010 1010 ----- Saddr2-offset	0000 1011	Saddr2'-offset
	saddr2, saddr1	0010 1010 ----- Saddr2-offset	0001 1011	Saddr1-offset
	saddr1, saddr2	0010 1010 ----- Saddr1-offset	0010 1011	Saddr2-offset
	saddr1, saddr1'	0010 1010 ----- Saddr1-offset	0011 1011	Saddr1'-offset
	A, [ saddrp2 ]	0000 0111	0010 1011	Saddr2-offset
	A, [ saddrp1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0010 1011
	A, [ %saddrg2 ]	0000 0111	0011 1011	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1011
	[ saddrp2 ], A	0000 0111	1010 1011	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1011
	[ %saddrg2 ], A	0000 0111	1011 1011	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1011
	A, !addr16	0000 1010 ----- High Address	0100 1011	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1011 ----- High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1011	Low Address

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
SUBC	!!addr24, A	0000 1010	1101 1011	High-w Address
		----- Low Address	----- High Address	-----
	A, [ TDE + ]	0001 0110	0000 1011	
	A, [ WHL + ]	0001 0110	0001 1011	
	A, [ TDE - ]	0001 0110	0010 1011	
	A, [ WHL - ]	0001 0110	0011 1011	
	A, [ TDE ]	0001 0110	0100 1011	
	A, [ WHL ]	0001 0110	0101 1011	
	A, [ VVP ]	0001 0110	0110 1011	
	A, [ UUP ]	0001 0110	0111 1011	
	A, [ TDE + byte ]	0000 0100	0000 1011	Low Offset
	A, [ SP + byte ]	0000 0110	0001 1011	Low Offset
	A, [ WHL + byte ]	0000 0110	0010 1011	Low Offset
	A, [ UUP + byte ]	0000 0110	0011 1011	Low Offset
	A, [ VVP + byte ]	0000 0110	0100 1011	Low Offset
	A, imm24 [ DE ]	0000 1010	0000 1011	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ A ]	0000 1010	0001 1011	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ HL ]	0000 1010	0010 1011	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ B ]	0000 1010	0011 1011	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, [ TDE + A ]	0001 0111	0000 1011	
	A, [ WHL + A ]	0001 0111	0001 1011	
	A, [ TDE + B ]	0001 0111	0010 1011	
	A, [ WHL + B ]	0001 0111	0011 1011	
	A, [ VVP + DE ]	0001 0111	0100 1011	
	A, [ VVP + HL ]	0001 0111	0101 1011	
	A, [ TDE + C ]	0001 0111	0110 1011	
	A, [ WHL + C ]	0001 0111	0111 1011	
	[ TDE + ], A	0001 0110	1000 1011	
	[ WHL + ], A	0001 0110	1001 1011	
	[ TDE - ], A	0001 0110	1010 1011	
[ WHL - ], A	0001 0110	1011 1011		
[ TDE ], A	0001 0110	1100 1011		
[ WHL ], A	0001 0110	1101 1011		
[ VVP ], A	0001 0110	1110 1011		
[ UUP ], A	0001 0110	1111 1011		

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>SUBC</b>	[ TDE + byte ], A	0000 0110	1000 1011	Low Offset
	[ SP + byte ], A	0000 0110	1001 1011	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1011	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1011	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1011	Low Offset
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1011 ----- High-w Offset	----- Low Offset
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1011 ----- High-w Offset	----- Low Offset
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1011 ----- High-w Offset	----- Low Offset
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1011 ----- High-w Offset	----- Low Offset
	[ TDE + A ], A	0001 0111	1000 1011	
	[ WHL + A ], A	0001 0111	1001 1011	
	[ TDE + B ], A	0001 0111	1010 1011	
	[ WHL + B ], A	0001 0111	1011 1011	
	[ VVP + DE ], A	0001 0111	1100 1011	
	[ VVP + HL ], A	0001 0111	1101 1011	
	[ TDE + C ], A	0001 0111	1110 1011	
	[ WHL + C ], A	0001 0111	1111 1011	
	<b>CMP</b>	A, #byte	1010 1111	#byte
r, #byte		0111 1111	R7R6R5R4 0011	#byte
saddr2, #byte		0110 1111	Saddr2-offset	#byte
saddr1, #byte		0011 1100 ----- #byte	0110 1111 ----- #byte	Saddr1-offset ----- Saddr1-offset
sfr, #byte		0000 0001 ----- #byte	0110 1111 ----- #byte	Sfr-offset ----- Sfr-offset
r, r1		1000 1111	R7R6R5R4 0R2R1R0	
r, r2		0011 1100	1000 1111	R7R6R5R4 0R2R1R0
A, saddr2		1001 1111	Saddr2-offset	
r, saddr2		0111 1111	R7R6R5R4 0000	Saddr2-offset
r, saddr1		0111 1111	R7R6R5R4 0001	Saddr1-offset
saddr2, r		0111 1111	R7R6R5R4 0100	Saddr2-offset
saddr1, r		0111 1111	R7R6R5R4 0101	Saddr1-offset
r, sfr		0111 1111	R7R6R5R4 0010	Sfr-offset
sfr, r		0111 1111	R7R6R5R4 0110	Sfr-offset
saddr2, saddr2'		0010 1010 ----- Saddr2-offset	0000 1111 ----- Saddr2-offset	Saddr2'-offset ----- Saddr2'-offset

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>CMP</b>	saddr2, saddr1	0010 1010 ----- Saddr2-offset	0001 1111	Saddr1-offset
	saddr1, saddr2	0010 1010 ----- Saddr1-offset	0010 1111	Saddr2-offset
	saddr1, saddr1'	0010 1010 ----- Saddr1-offset	0011 1111	Saddr1'-offset
	A, [ saddrp2 ]	0000 0111	0010 1111	Saddr2-offset
	A, [ saddrp1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0010 1111
	A, [ %saddrg2 ]	0000 0111	0011 1111	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1111
	[ saddrp2 ], A	0000 0111	1010 1111	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1111
	[ %saddrg2 ], A	0000 0111	1011 1111	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1111
	A, !addr16	0000 1010 ----- High Address	0100 1111	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1111 High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1111	Low Address
	!!addr24, A	0000 0101 ----- Low Address	0110 11111 High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1111	
	A, [ WHL + ]	0001 0110	0001 1111	
	A, [ TDE - ]	0001 0110	0010 1111	
	A, [ WHL - ]	0001 0110	0011 1111	
	A, [ TDE ]	0001 0110	0100 1111	
	A, [ WHL ]	0001 0110	0101 1111	
	A, [ VVP ]	0001 0110	0110 1111	
	A, [ UUP ]	0001 0110	0111 1111	
	A, [ TDE + byte ]	0000 0110	0000 1111	Low Offset
	A, [ SP + byte ]	0000 0110	0001 1111	Low Offset
	A, [ WHL + byte ]	0000 0110	0010 1111	Low Offset
	A, [ UUP + byte ]	0000 0110	0011 1111	Low Offset
	A, [ VVP + byte ]	0000 0110	0100 1111	Low Offset

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
CMP	A, imm24 [ DE]	0000 1010 ----- High Offset	0000 1111 ----- High-w Offset	Low Offset
	A, imm24 [ A]	0000 1010 ----- High Offset	0001 1111 ----- High-w Offset	Low Offset
	A, imm24 [ HL]	0000 1010 ----- High Offset	0010 1111 ----- High-w Offset	Low Offset
	A, imm24 [ B]	0000 1010 ----- High Offset	0011 1111 ----- High-w Offset	Low Offset
	A, [ TDE + A]	0001 0111	0000 1111	
	A, [ WHL + A]	0001 0111	0001 1111	
	A, [ TDE + B]	0001 0111	0010 1111	
	A, [ WHL + B]	0001 0111	0011 1111	
	A, [ VVP + DE]	0001 0111	0100 1111	
	A, [ VVP + HL]	0001 0111	0101 1111	
	A, [ TDE + C]	0001 0111	0110 1111	
	A, [ WHL + C]	0001 0111	0111 1111	
	[ TDE + ], A	0001 0110	1000 1111	
	[ WHL + ], A	0001 0110	1001 1111	
	[ TDE - ], A	0001 0110	1010 1111	
	[ WHL - ], A	0001 0110	1011 1111	
	[ TDE ], A	0001 0110	1100 1111	
	[ WHL ], A	0001 0110	1101 1111	
	[ VVP ], A	0001 0110	1110 1111	
	[ UUP ], A	0001 0110	1111 1111	
	[ TDE + byte ], A	0000 0110	1000 1111	Low Offset
	[ SP + byte ], A	0000 0110	1001 1111	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1111	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1111	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1111	Low Offset
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1111 ----- High-w Offset	Low Offset
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1111 ----- High-w Offset	Low Offset
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1111 ----- High-w Offset	Low Offset
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1111 ----- High-w Offset	Low Offset
	[ TDE + A ], A	0001 0111	1000 1111	
	[ WHL + A ], A	0001 0111	1001 1111	

(次ページに続く)

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>CMP</b>	[ TDE + B ] , A	0001 0111	1010 1111	
	[ WHL + B ] , A	0001 0111	1011 1111	
	[ VVP + DE ] , A	0001 0111	1100 1111	
	[ VVP + HL ] , A	0001 0111	1101 1111	
	[ TDE + C ] , A	0001 0111	1110 1111	
	[ WHL + C ] , A	0001 0111	1111 1111	
<b>AND</b>	A, #byte	1010 1100	#byte	
	r, #byte	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0011	#byte
	saddr2, #byte	0110 1100	Saddr2-offset	#byte
	saddr1, #byte	0011 1100	0110 1100	Saddr1-offset
		#byte		
	sfr, #byte	0000 0001	0110 1100	Sfr-offset
		#byte		
	r, r1	1000 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r, r2	0011 1100	1000 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
	A, saddr2	1001 1100	Saddr2-offset	
	r, saddr2	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0000	Saddr2-offset
	r, saddr1	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0001	Saddr1-offset
	saddr2, r	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0100	Saddr2-offset
	saddr1, r	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0101	Saddr1-offset
	r, sfr	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0010	Sfr-offset
	sfr, r	0111 1100	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0110	Sfr-offset
	saddr2, saddr2'	0010 1010	0000 1100	Saddr2'-offset
		Saddr2-offset		
	saddr2, saddr1	0010 1010	0001 1100	Saddr1-offset
		Saddr2-offset		
	saddr1, saddr2	0010 1010	0010 1100	Saddr2-offset
		Saddr1-offset		
	saddr1, saddr1'	0010 1010	0011 1100	Saddr1'-offset
		Saddr1-offset		
	A, [ saddrp2 ]	0000 0111	0010 1100	Saddr2-offset
	A, [ saddrp1 ]	0011 1100	0000 0111	0010 1100
		Saddr1-offset		
	A, [ %saddrg2 ]	0000 0111	0011 1100	Saddr2-offset
A, [ %saddrg1 ]	0011 1100	0000 0111	0011 1100	
	Saddr1-offset			
[ saddrp2 ] , A	0000 0111	1010 1100	Saddr2-offset	
[ saddrp1 ] , A	0011 1100	0000 0111	1010 1100	
	Saddr1-offset			

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
AND	[ %saddrg2 ], A	0000 0111	1011 1100	Saddr2-offset
	[ %saddrg1 ], A	0011 1100	0000 0111	1011 1100
		Saddr1-offset		
	A, !addr16	0000 1010	0100 1100	Low Address
		High Address		
	A, !!addr24	0000 1010	0101 1100	High-w Address
		Low Address	High Address	
	!addr16, A	0000 1010	1100 1100	Low Address
		High Address		
	!!addr24, A	0000 1010	1101 1100	High-w Address
		Low Address	High Address	
	A, [ TDE + ]	0001 0110	0000 1100	
	A, [ WHL + ]	0001 0110	0001 1100	
	A, [ TDE - ]	0001 0110	0010 1100	
	A, [ WHL - ]	0001 0110	0011 1100	
	A, [ TDE]	0001 0110	0100 1100	
	A, [ WHL]	0001 0110	0101 1100	
	A, [ VVP]	0001 0110	0110 1100	
	A, [ UUP]	0001 0110	0111 1100	
	A, [ TDE + byte]	0000 0110	0000 1100	Low Offset
	A, [ SP + byte]	0000 0110	0001 1100	Low Offset
	A, [ WHL + byte]	0000 0110	0010 1100	Low Offset
	A, [ UUP + byte]	0000 0110	0011 1100	Low Offset
	A, [ VVP + byte]	0000 0110	0100 1100	Low Offset
	A, imm24 [ DE]	0000 1010	0000 1100	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ A]	0000 1010	0001 1100	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ HL]	0000 1010	0010 1100	Low Offset
		High Offset	High-w Offset	
	A, imm24 [ B]	0000 1010	0011 1100	Low Offset
		High Offset	High-w Offset	
	A, [ TDE + A]	0001 0111	0000 1100	
	A, [ WHL + A]	0001 0111	0001 1100	
A, [ TDE + B]	0001 0111	0010 1100		
A, [ WHL + B]	0001 0111	0011 1100		
A, [ VVP + DE]	0001 0111	0100 1100		
A, [ VVP + HL]	0001 0111	0101 1100		
A, [ TDE + C]	0001 0111	0110 1100		

(次ページに続く)

ニモニク	オペランド	命令コード			
		B1	B2	B3	
		B4	B5	B6	
		B7			
<b>AND</b>	A, [ WHL + C ]	0001 0111	0111 1100		
	[ TDE + ], A	0001 0110	1000 1100		
	[ WHL + ], A	0001 0110	1001 1100		
	[ TDE - ], A	0001 0110	1010 1100		
	[ WHL - ], A	0001 0110	1011 1100		
	[ TDE ], A	0001 0110	1100 1100		
	[ WHL ], A	0001 0110	1101 1100		
	[ VVP ], A	0001 0110	1110 1100		
	[ UUP ], A	0001 0110	1111 1100		
	[ TDE + byte ], A	0000 0110	1000 1100	Low Offset	
	[ SP + byte ], A	0000 0110	1001 1100	Low Offset	
	[ WHL + byte ], A	0000 0110	1010 1100	Low Offset	
	[ UUP + byte ], A	0000 0110	1011 1100	Low Offset	
	[ VVP + byte ], A	0000 0110	1100 1100	Low Offset	
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1100 ----- High-w Offset	Low Offset -----	
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1100 ----- High-w Offset	Low Offset -----	
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1100 ----- High-w Offset	Low Offset -----	
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1100 ----- High-w Offset	Low Offset -----	
	[ TDE + A ], A	0001 0111	1000 1100		
	[ WHL + A ], A	0001 0111	1001 1100		
	[ TDE + B ], A	0001 0111	1010 1100		
	[ WHL + B ], A	0001 0111	1011 1100		
	[ VVP + DE ], A	0001 0111	1100 1100		
	[ VVP + HL ], A	0001 0111	1101 1100		
	[ TDE + C ], A	0001 0111	1110 1100		
	[ WHL + C ], A	0001 0111	1111 1100		
	<b>OR</b>	A, #byte	1010 1110	#byte	
		r, #byte	0111 1110	R7R6R5R4 0011	#byte
saddr2, #byte		0110 1110	Saddr2-offset	#byte	
saddr1, #byte		0011 1100 ----- #byte	0110 1110	Saddr1-offset -----	
sfr, #byte		0000 0001 ----- #byte	0110 1110	Sfr-offset -----	
r, r1		1000 1110	R7R6R5R4 0R2R1R0		
r, r2		0011 1100	1000 1110	R7R6R5R4 0R2R1R0	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
OR	A, saddr2	1001 1110	Saddr2-offset	
	r, saddr2	0111 1110	R7R6R5R4 0000	Saddr2-offset
	r, saddr1	0111 1110	R7R6R5R4 0001	Saddr1-offset
	saddr2, r	0111 1110	R7R6R5R4 0100	Saddr2-offset
	saddr1, r	0111 1110	R7R6R5R4 0101	Saddr1-offset
	r, sfr	0111 1110	R7R6R5R4 0010	Sfr-offset
	sfr, r	0111 1110	R7R6R5R4 0110	Sfr-offset
	saddr2, saddr2'	0010 1010 ----- Saddr2-offset	0000 1110	Saddr2'-offset
	saddr2, saddr1	0010 1010 ----- Saddr2-offset	0001 1110	Saddr1-offset
	saddr1, saddr2	0010 1010 ----- Saddr1-offset	0010 1110	Saddr2-offset
	saddr1, saddr1'	0010 1010 ----- Saddr1-offset	0011 1110	Saddr1'-offset
	A, [ saddrp2 ]	0000 0111	0010 1110	Saddr2-offset
	A, [ saddrp1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0010 1110
	A, [ %saddrg2 ]	0000 0111	0011 1110	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1110
	[ saddrp2 ], A	0000 0111	1010 1110	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1110
	[ %saddrg2 ], A	0000 0111	1011 1110	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1110
	A, !addr16	0000 1010 ----- High Address	0100 1110	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1110 ----- High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1110	Low Address
	!!addr24, A	0000 1010 ----- Low Address	1101 1110 ----- High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1110	
	A, [ WHL + ]	0001 0110	0001 1110	
	A, [ TDE - ]	0001 0110	0010 1110	
	A, [ WHL - ]	0001 0110	0011 1110	

(次ページに続く)

ニモニク	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
OR	A, [ TDE]	0001 0110	0100 1110	
	A, [ WHL]	0001 0110	0101 1110	
	A, [ VVP]	0001 0110	0110 1110	
	A, [ UUP]	0001 0110	0111 1110	
	A, [ TDE + byte]	0000 0110	0000 1110	Low Offset
	A, [ SP + byte]	0000 0110	0001 1110	Low Offset
	A, [ WHL + byte]	0000 0110	0010 1110	Low Offset
	A, [ UUP + byte]	0000 0110	0011 1110	Low Offset
	A, [ VVP + byte]	0000 0110	0100 1110	Low Offset
	A, imm24 [ DE]	0000 1010	0000 1110	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ A]	0000 1010	0001 1110	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ HL]	0000 1010	0010 1110	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, imm24 [ B]	0000 1010	0011 1110	Low Offset
		----- High Offset	----- High-w Offset	-----
	A, [ TDE + A]	0001 0111	0000 1110	
	A, [ WHL + A]	0001 0111	0001 1110	
	A, [ TDE + B]	0001 0111	0010 1110	
	A, [ WHL + B]	0001 0111	0011 1110	
	A, [ VVP + DE]	0001 0111	0100 1110	
	A, [ VVP + HL]	0001 0111	0101 1110	
	A, [ TDE + C]	0001 0111	0110 1110	
	A, [ WHL + C]	0001 0111	0111 1110	
	[ TDE + ], A	0001 0110	1000 1110	
	[ WHL + ], A	0001 0110	1001 1110	
	[ TDE - ], A	0001 0110	1010 1110	
	[ WHL - ], A	0001 0110	1011 1110	
	[ TDE ], A	0001 0110	1100 1110	
	[ WHL ], A	0001 0110	1101 1110	
	[ VVP ], A	0001 0110	1110 1110	
	[ UUP ], A	0001 0110	1111 1110	
	[ TDE + byte ], A	0000 0110	1000 1110	Low Offset
	[ SP + byte ], A	0000 0110	1001 1110	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1110	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1110	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1110	Low Offset

(次ページに続く)

ニモニック	オペランド	命令コード			
		B1	B2	B3	
		B4	B5	B6	
		B7			
OR	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1110 ----- High-w Offset	Low Offset	
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1110 ----- High-w Offset	Low Offset	
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1110 ----- High-w Offset	Low Offset	
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1110 ----- High-w Offset	Low Offset	
	[ TDE + A ], A	0001 0111	1000 1110		
	[ WHL + A ], A	0001 0111	1001 1110		
	[ TDE + B ], A	0001 0111	1010 1110		
	[ WHL + B ], A	0001 0111	1011 1110		
	[ VVP + DE ], A	0001 0111	1100 1110		
	[ VVP + HL ], A	0001 0111	1101 1110		
	[ TDE + C ], A	0001 0111	1110 1110		
	[ WHL + C ], A	0001 0111	1111 1110		
	XOR	A, #byte	1010 1101	#byte	
		r, #byte	0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0011	#byte
saddr2, #byte		0110 1101	Saddr2-offset	#byte	
saddr1, #byte		0011 1100 ----- #byte	0110 1101	Saddr1-offset	
sfr, #byte		0000 0001 ----- #byte	0110 1101	Sfr-offset	
r, r1		1000 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
r, r2		0011 1100	1000 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
A, saddr2		1001 1101	Saddr2-offset		
r, saddr2		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0000	Saddr2-offset	
r, saddr1		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0001	Saddr1-offset	
saddr2, r		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0100	Saddr2-offset	
saddr1, r		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0101	Saddr1-offset	
r, sfr		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0010	Sfr-offset	
sfr, r		0111 1101	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> 0110	Sfr-offset	
saddr2, saddr2'		0010 1010 ----- Saddr2-offset	0000 1101	Saddr2'-offset	
saddr2, saddr1		0010 1010 ----- Saddr2-offset	0001 1101	Saddr1-offset	
saddr1, saddr2		0010 1010 ----- Saddr1-offset	0010 1101	Saddr2-offset	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
XOR	saddr1, saddr1'	0010 1010 ----- Saddr1-offset	0011 1101	Saddr1'-offset
	A, [ saddrp2 ]	0000 0111	0010 1101	Saddr2-offset
	A, [ saddrp1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0010 1101
	A, [ %saddrg2 ]	0000 0111	0011 1101	Saddr2-offset
	A, [ %saddrg1 ]	0011 1100 ----- Saddr1-offset	0000 0111	0011 1101
	[ saddrp2 ], A	0000 0111	1010 1101	Saddr2-offset
	[ saddrp1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1010 1101
	[ %saddrg2 ], A	0000 0111	1011 1101	Saddr2-offset
	[ %saddrg1 ], A	0011 1100 ----- Saddr1-offset	0000 0111	1011 1101
	A, !addr16	0000 1010 ----- High Address	0100 1101	Low Address
	A, !!addr24	0000 1010 ----- Low Address	0101 1101 ----- High Address	High-w Address
	!addr16, A	0000 1010 ----- High Address	1100 1101	Low Address
	!!addr24, A	0000 1010 ----- Low Address	1101 1101 ----- High Address	High-w Address
	A, [ TDE + ]	0001 0110	0000 1101	
	A, [ WHL + ]	0001 0110	0001 1101	
	A, [ TDE - ]	0001 0110	0010 1101	
	A, [ WHL - ]	0001 0110	0011 1101	
	A, [ TDE ]	0001 0110	0100 1101	
	A, [ WHL ]	0001 0110	0101 1101	
	A, [ VVP ]	0001 0110	0110 1101	
	A, [ UUP ]	0001 0110	0111 1101	
	A, [ TDE + byte ]	0000 0110	0000 1101	Low Offset
	A, [ SP + byte ]	0000 0110	0001 1101	Low Offset
	A, [ WHL + byte ]	0000 0110	0010 1101	Low Offset
	A, [ UUP + byte ]	0000 0110	0011 1101	Low Offset
	A, [ VVP + byte ]	0000 0110	0100 1101	Low Offset
	A, imm24 [ DE ]	0000 1010 ----- High Offset	0000 1101 ----- High-w Offset	Low Offset
	A, imm24 [ A ]	0000 1010 ----- High Offset	0001 1101 ----- High-w Offset	Low Offset

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
XOR	A, imm24 [ HL ]	0000 1010 ----- High Offset	0010 1101 ----- High-w Offset	Low Offset -----
	A, imm24 [ B ]	0000 1010 ----- High Offset	0011 1101 ----- High-w Offset	Low Offset -----
	A, [ TDE + A ]	0001 0111	0000 1101	
	A, [ WHL + A ]	0001 0111	0001 1101	
	A, [ TDE + B ]	0001 0111	0010 1101	
	A, [ WHL + B ]	0001 0111	0011 1101	
	A, [ VVP + DE ]	0001 0111	0100 1101	
	A, [ VVP + HL ]	0001 0111	0101 1101	
	A, [ TDE + C ]	0001 0111	0110 1101	
	A, [ WHL + C ]	0001 0111	0111 1101	
	[ TDE + ], A	0001 0110	1000 1101	
	[ WHL + ], A	0001 0110	1001 1101	
	[ TDE - ], A	0001 0110	1010 1101	
	[ WHL - ], A	0001 0110	1011 1101	
	[ TDE ], A	0001 0110	1100 1101	
	[ WHL ], A	0001 0110	1101 1101	
	[ VVP ], A	0001 0110	1110 1101	
	[ UUP ], A	0001 0110	1111 1101	
	[ TDE + byte ], A	0000 0110	1000 1101	Low Offset
	[ SP + byte ], A	0000 0110	1001 1101	Low Offset
	[ WHL + byte ], A	0000 0110	1010 1101	Low Offset
	[ UUP + byte ], A	0000 0110	1011 1101	Low Offset
	[ VVP + byte ], A	0000 0110	1100 1101	Low Offset
	imm24 [ DE ], A	0000 1010 ----- High Offset	1000 1101 ----- High-w Offset	Low Offset -----
	imm24 [ A ], A	0000 1010 ----- High Offset	1001 1101 ----- High-w Offset	Low Offset -----
	imm24 [ HL ], A	0000 1010 ----- High Offset	1010 1101 ----- High-w Offset	Low Offset -----
	imm24 [ B ], A	0000 1010 ----- High Offset	1011 1101 ----- High-w Offset	Low Offset -----
	[ TDE + A ], A	0001 0111	1000 1101	
	[ WHL + A ], A	0001 0111	1001 1101	
	[ TDE + B ], A	0001 0111	1010 1101	
	[ WHL + B ], A	0001 0111	1011 1101	
	[ VVP + DE ], A	0001 0111	1100 1101	
	[ VVP + HL ], A	0001 0111	1101 1101	
	[ TDE + C ], A	0001 0111	1110 1101	
	[ WHL + C ], A	0001 0111	1111 1101	

(7) 16ビット演算命令 : ADDW, SUBW, CMPW

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADDW</b>	AX, #word	0010 1101	Low Byte	High Byte
	rp, #word	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1011	Low Byte
		High Byte		
	rp, rp'	1000 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AX, saddrp2	0001 1101	Saddr2-offset	
	rp, saddrp2	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Saddr2-offset
	rp, saddrp1	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Saddr1-offset
	saddrp2, rp	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1100	Saddr2-offset
	saddrp1, rp	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1101	Saddr1-offset
	rp, sfrp	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010	Sfr-offset
	sfrp, rp	0111 1000	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1110	Sfr-offset
	saddrp2, #word	0000 1101	Saddr2-offset	Low Byte
		High Byte		
	saddrp1, #word	0011 1100	0000 1101	Saddr1-offset
		Low Byte	High Byte	
	sfrp, #word	0000 0001	0000 1101	Sfr-offset
		Low Byte	High Byte	
	saddrp2, saddrp2'	0010 1010	1000 1101	Saddr2'-offset
	Saddr2-offset			
saddrp2, saddrp1	0010 1010	1001 1101	Saddr1-offset	
	Saddr2-offset			
saddrp1, saddrp2	0010 1010	1010 1101	Saddr2-offset	
	Saddr1-offset			
saddrp1, saddrp1'	0010 1010	1011 1101	Saddr1'-offset	
	Saddr1-offset			
<b>SUBW</b>	AX, #word	0010 1110	Low Byte	High Byte
	rp, #word	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1011	Low Byte
		High Byte		
	rp, rp'	1000 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AX, saddrp2	0001 1110	Saddr2-offset	
	rp, saddrp2	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Saddr2-offset
	rp, saddrp1	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Saddr1-offset
	saddrp2, rp	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1100	Saddr2-offset
	saddrp1, rp	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1101	Saddr1-offset
	rp, sfrp	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010	Sfr-offset
	sfrp, rp	0111 1010	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1110	Sfr-offset
saddrp2, #word	0000 1110	Saddr2-offset	Low Byte	
	High Byte			

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>SUBW</b>	saddrp1, #word	0011 1100 ----- Low Byte	0000 1110 ----- High Byte	Saddr1-offset
	sfrp, #word	0000 0001 ----- Low Byte	0000 1110 ----- High Byte	Sfr-offset
	saddrp2, saddrp2'	0010 1010 ----- Saddr2-offset	1000 1110	Saddr2'-offset
	saddrp2, saddrp1	0010 1010 ----- Saddr2-offset	1001 1110	Saddr1-offset
	saddrp1, saddrp2	0010 1010 ----- Saddr1-offset	1010 1110	Saddr2-offset
	saddrp1, saddrp1'	0010 1010 ----- Saddr1-offset	1011 1110	Saddr1'-offset
<b>CMPW</b>	AX, #word	0010 1111	Low Byte	High Byte
	rp, #word	0111 1111 ----- High Byte	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1011	Low Byte
	rp, rp'	1000 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AX, saddrp2	0001 1111	Saddr2-offset	
	rp, saddrp2	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1000	Saddr2-offset
	rp, saddrp1	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1001	Saddr1-offset
	saddrp2, rp	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1100	Saddr2-offset
	saddrp1, rp	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1101	Saddr1-offset
	rp, sfrp	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1010	Sfr-offset
	sfrp, rp	0111 1111	P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> 0 1110	Sfr-offset
	saddrp2, #word	0000 1111 ----- High Byte	Saddr2-offset	Low Byte
	saddrp1, #word	0011 1100 ----- Low Byte	0000 1111 ----- High Byte	Saddr1-offset
	sfrp, #word	0000 0001 ----- Low Byte	0000 1111 ----- High Byte	Sfr-offset
	saddrp2, saddrp2'	0010 1010 ----- Saddr2-offset	1000 1111	Saddr2'-offset
	saddrp2, saddrp1	0010 1010 ----- Saddr2-offset	1001 1111	Saddr1-offset
	saddrp1, saddrp2	0010 1010 ----- Saddr1-offset	1010 1111	Saddr2-offset
saddrp1, saddrp1'	0010 1010 ----- Saddr1-offset	1011 1111	Saddr1'-offset	

(8) 24ビット演算命令 : ADDG, SUBG

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADDG</b>	rg, rg'	1000 1000	1G <sub>6</sub> G <sub>5</sub> 1 1G <sub>2</sub> G <sub>1</sub> 1	
	rg, #imm24	0111 1000	1G <sub>6</sub> G <sub>5</sub> 1 1011	Low Byte
		High Byte	High-w Byte	
	WHL, saddrg2	0111 1000	1111 1000	Saddr2-offset
	WHL, saddrg1	0111 1000	1111 1001	Saddr1-offset
<b>SUBG</b>	rg, rg'	1000 1010	1G <sub>6</sub> G <sub>5</sub> 1 1G <sub>2</sub> G <sub>1</sub> 1	
	rg, #imm24	0111 1010	1G <sub>6</sub> G <sub>5</sub> 1 1011	Low Byte
		High Byte	High-w Byte	
	WHL, saddrg2	0111 1010	1111 1000	Saddr2-offset
	WHL, saddrg1	0111 1010	1111 1001	Saddr1-offset

(9) 乗除算命令 : MULU, MULUW, MULW, DIVUW, DIVUX

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MULU</b>	r1	0000 0101	0000 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2	0011 1100	0000 0101	0000 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>MULUW</b>	rp	0000 0101	0010 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
<b>MULW</b>	rp	0000 0101	0011 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
<b>DIVUW</b>	r1	0000 0101	0001 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2	0011 1100	0000 0101	0001 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>DIVUX</b>	rp	0000 0101	1110 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	

(10) 特殊演算命令 : MACW, MACSW, SACW

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MACW</b>	byte	0000 0111	1000 0101	byte
<b>MACSW</b>	byte	0000 0111	1001 0101	byte
<b>SACW</b>	[ TDE + ], [ WHL + ]	0000 1001	0110 0100	0100 0001
		0100 0110		

(11) 増減命令 : INC, DEC, INCW, DECW, INCG, DECG

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>INC</b>	r1	1100 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
	r2	0011 1100	1100 0R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	saddr2	0010 0110	Saddr2-offset	
	saddr1	0011 1100	0010 0110	Saddr1-offset
<b>DEC</b>	r1	1100 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
	r2	0011 1100	1100 1R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	saddr2	0010 0111	Saddr2-offset	
	saddr1	0011 1100	0010 0111	Saddr1-offset
<b>INCW</b>	RP0	0011 1110	0000 1101	
	RP1	0011 1110	0010 1101	
	RP2	0011 1110	0100 1101	
	RP3	0011 1110	0110 1101	
	VP ( RP4 )	0100 0100		
	UP ( RP5 )	0100 0101		
	DE ( RP6 )	0100 0110		
	HL ( RP7 )	0100 0111		
	saddrp2	0000 0111	1110 1000	Saddr2-offset
	saddrp1	0011 1100	0000 0111	1110 1000
	Saddr1-offset			
<b>DECW</b>	RP0	0011 1110	0000 1111	
	RP1	0011 1110	0010 1111	
	RP2	0011 1110	0100 1111	
	RP3	0011 1110	0110 1111	
	VP ( RP4 )	0100 1100		
	UP ( RP5 )	0100 1101		
	DE ( RP6 )	0100 1110		
	HL ( RP7 )	0100 1111		
	saddrp2	0000 0111	1110 1001	Saddr2-offset
	saddrp1	0011 1100	0000 0111	1110 1001
	Saddr1-offset			
<b>INCG</b>	rg	0011 1110	1G <sub>6</sub> G <sub>5</sub> 1 1101	
<b>DECG</b>	rg	0011 1110	1G <sub>6</sub> G <sub>5</sub> 1 1111	

(12) 補正命令 : ADJBA, ADJBS, CVTBW

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ADJBA</b>		0000 0101	1111 1110	
<b>ADJBS</b>		0000 0101	1111 1111	
<b>CVTBW</b>		0000 0100		

(13) シフト・ローテート命令 : ROR, ROL, RORC, ROLC, SHR, SHL, SHRW, SHLW, ROR4, ROL4

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>ROR</b>	r1, n	0011 0000	01N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0000	01N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>ROL</b>	r1, n	0011 0001	01N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0001	01N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>RORC</b>	r1, n	0011 0000	00N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0000	00N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>ROLC</b>	r1, n	0011 0001	00N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0001	00N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>SHR</b>	r1, n	0011 0000	10N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0000	10N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>SHL</b>	r1, n	0011 0001	10N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	r2, n	0011 1100	0011 0001	10N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>
<b>SHRW</b>	rp, n	0011 0000	11N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
<b>SHLW</b>	rp, n	0011 0001	11N <sub>2</sub> N <sub>1</sub> N <sub>0</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
<b>ROR4</b>	mem3	0000 0101	1000 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
<b>ROL4</b>	mem3	0000 0101	1001 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	

(14) ビット操作命令 : MOV1, AND1, OR1, XOR1, NOT1, SET1, CLR1

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MOV1</b>	CY, saddr2. bit	0000 1000	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, saddr1. bit	0011 1100	0000 1000	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		Saddr1-offset		
	CY, sfr. bit	0000 1000	0000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, X. bit	0000 0011	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, A. bit	0000 0011	0000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWL. bit	0000 0010	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWH. bit	0000 0010	0000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ TDE ] . bit	0011 1101	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ WHL ] . bit	0011 1101	0000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, !addr16. bit	0000 1001	1101 0000	0000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		Low Address	High Address	
	CY, !!addr24. bit	0000 1001	1101 0000	0000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		High-w Address	Low Address	High Address
	saddr2. bit, CY	0000 1000	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	saddr1. bit, CY	0011 1100	0000 1000	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		Saddr1-offset		
	sfr. bit, CY	0000 1000	0001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	X. bit, CY	0000 0011	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	A. bit, CY	0000 0011	0001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWL. bit, CY	0000 0010	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
PSWH. bit, CY	0000 0010	0001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>		
[ TDE ] . bit, CY	0011 1101	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>		
[ WHL ] . bit, CY	0011 1101	0001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>		
!addr16. bit, CY	0000 1001	1101 0000	0001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	Low Address	High Address		
!!addr24. bit, CY	0000 1001	1101 0000	0001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	High-w Address	Low Address	High Address	
<b>AND1</b>	CY, saddr2. bit	0000 1000	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, saddr1. bit	0011 1100	0000 1000	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		Saddr1-offset		
	CY, /saddr2. bit	0000 1000	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, /saddr1. bit	0011 1100	0000 1000	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		Saddr1-offset		
	CY, sfr. bit	0000 1000	0010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, /sfr. bit	0000 1000	0011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, X. bit	0000 0011	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
CY, /X. bit	0000 0011	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>		
CY, A. bit	0000 0011	0010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>		

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>AND1</b>	CY, /A. bit	0000 0011	0011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWL. bit	0000 0010	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /PSWL. bit	0000 0010	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWH. bit	0000 0010	0010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /PSWH. bit	0000 0010	0011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ TDE ] . bit	0011 1101	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, / [ TDE ] . bit	0011 1101	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ WHL ] . bit	0011 1101	0010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, / [ WHL ] . bit	0011 1101	0011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, !addr16. bit	0000 1001	1101 0000	0010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Low Address	----- High Address	-----
	CY, /!addr16. bit	0000 1001	1101 0000	0011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Low Address	----- High Address	-----
	CY, !!addr24. bit	0000 1001	1101 0000	0010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	----- High-w Address	----- Low Address	----- High Address	
CY, /!!addr24. bit	0000 1001	1101 0000	0011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	----- High-w Address	----- Low Address	----- High Address	
<b>OR1</b>	CY, saddr2. bit	0000 1000	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, saddr1. bit	0011 1100	0000 1000	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Saddr1-offset	-----	-----
	CY, /saddr2. bit	0000 1000	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, /saddr1. bit	0011 1100	0000 1000	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Saddr1-offset	-----	-----
	CY, sfr. bit	0000 1000	0100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, /sfr. bit	0000 1000	0101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, X. bit	0000 0011	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /X. bit	0000 0011	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, A. bit	0000 0011	0100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /A. bit	0000 0011	0101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWL. bit	0000 0010	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /PSWL. bit	0000 0010	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWH. bit	0000 0010	0100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, /PSWH. bit	0000 0010	0101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ TDE ] . bit	0011 1101	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, / [ TDE ] . bit	0011 1101	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ WHL ] . bit	0011 1101	0100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, / [ WHL ] . bit	0011 1101	0101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
CY, !addr16. bit	0000 1001	1101 0000	0100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	----- Low Address	----- High Address	-----	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>OR1</b>	CY, /!addr16. bit	0000 1001 ----- Low Address	1101 0000 ----- High Address	0101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	CY, !!addr24. bit	0000 1001 ----- High-w Address	1101 0000 ----- Low Address	0100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address
	CY, /!!addr24. bit	0000 1001 ----- High-w Address	1101 0000 ----- Low Address	0101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address
<b>XOR1</b>	CY, saddr2. bit	0000 1000	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	CY, saddr1. bit	0011 1100 ----- Saddr1-offset	0000 1000	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	CY, sfr. bit	0000 1000	0110 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	CY, X. bit	0000 0011	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, A. bit	0000 0011	0110 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWL. bit	0000 0010	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, PSWH. bit	0000 0010	0110 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ TDE ] . bit	0011 1101	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, [ WHL ] . bit	0011 1101	0110 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	CY, !addr16. bit	0000 1001 ----- Low Address	1101 0000 ----- High Address	0110 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
<b>NOT1</b>	saddr2. bit	0000 1000	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
	saddr1. bit	0011 1100 ----- Saddr1-offset	0000 1000	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	sfr. bit	0000 1000	0111 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	X. bit	0000 0011	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	A. bit	0000 0011	0111 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWL. bit	0000 0010	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWH. bit	0000 0010	0111 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ TDE ] . bit	0011 1101	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ WHL ] . bit	0011 1101	0111 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	!addr16. bit	0000 1001 ----- Low Address	1101 0000 ----- High Address	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
<b>SET1</b>	!!addr24. bit	0000 1001 ----- High-w Address	1101 0000 ----- Low Address	0111 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address
	CY	0100 0010		
	saddr2. bit	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset	
	saddr1. bit	0011 1100	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr1-offset
	sfr. bit	0000 1000	1000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	X. bit	0000 0011	1000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>SET1</b>	A. bit	0000 0011	1000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWL. bit	0000 0010	1000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWH. bit	0000 0010	1000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ TDE ] . bit	0011 1101	1000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ WHL ] . bit	0011 1101	1000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	!addr16. bit	0000 1001	1101 0000	1000 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Low Address	----- High Address	-----
	!!addr24. bit	0000 1001	1101 0000	1000 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
	----- High-w Address	----- Low Address	----- High Address	
CY	0100 0001			
<b>CLR1</b>	saddr2. bit	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset	
	saddr1. bit	0011 1100	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr1-offset
	sfr. bit	0000 1000	1001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
	X. bit	0000 0011	1001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	A. bit	0000 0011	1001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWL. bit	0000 0010	1001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	PSWH. bit	0000 0010	1001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ TDE ] . bit	0011 1101	1001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	[ WHL ] . bit	0011 1101	1001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	!addr16. bit	0000 1001	1101 0000	1001 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Low Address	----- High Address	-----
	!!addr24. bit	0000 1001	1101 0000	1001 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- High-w Address	----- Low Address	----- High Address
	CY	0100 0000		

(15) スタック操作命令 : PUSH, PUSHU, POP, POPU, MOVG, ADDWG, SUBWG, INCG, DECG

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>PUSH</b>	PSW	0100 1001		
	sfrp	0000 0111	1101 1001	sfr-offset
	sfr	0000 0111	1101 1011	sfr-offset
	post	0011 0101	post	
	rg	0000 1001	1000 1G <sub>2</sub> G <sub>1</sub> 1	
<b>PUSHU</b>	post	0011 0111	post	
<b>POP</b>	PSW	0100 1000		
	sfrp	0000 0111	1101 1000	Sfr-offset
	sfr	0000 0111	1101 1010	Sfr-offset
	post	0011 0100	post	
	rg	0000 1001	1001 1G <sub>2</sub> G <sub>1</sub> 1	
<b>POPU</b>	post	0011 0110	post	
<b>MOVG</b>	SP, #imm24	0000 1001 ----- High Byte	0010 0000 ----- Hing-w Byte	Low Byte
	SP, WHL	0000 0101	1111 1011	
	WHL, SP	0000 0101	1111 1010	
<b>ADDWG</b>	SP, #word	0000 1001 ----- High Byte	0010 1000	Low Byte
	SP, #word	0000 1001 ----- High Byte	0010 1010	Low Byte
<b>INCG</b>	SP	0000 0101	1111 1000	
<b>DECG</b>	SP	0000 0101	1111 1001	

(16) コール・リターン命令 : CALL, CALLF, CALLT, BRK, BRKCS, RET, RETI, RETB, RETCS, RETCSB

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
CALL	!addr16	0010 1000	Low Address	High Address
	!!addr20	0000 1001	1111 Hi-w Add	Low Address
		-----	-----	-----
		High Address		
	rp	0000 0101	0101 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	rg	0000 0101	0101 0G <sub>2</sub> G <sub>1</sub> 1	
	[ rp ]	0000 0101	0111 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
[ rg ]	0000 0101	0111 0G <sub>2</sub> G <sub>1</sub> 1		
	\$ !addr20	0011 1111	\$ addr Low	\$ addr High
CALLF	!addr1	1100 0	fa	
CALLT	[ addr5 ]	111T <sub>4</sub> T <sub>3</sub> T <sub>2</sub> T <sub>1</sub> T <sub>0</sub>		
BRK		0101 1110		
BRKCS	RBn	0000 0101	1101 1E <sub>2</sub> E <sub>1</sub> E <sub>0</sub>	
RET		0101 0110		
RETI		0101 0111		
RETB		0101 1111		
RETCS	!addr16	0010 1001	Low Address	High Address
RETCSB	!addr16	0000 1001	1011 0000	Low Address
		-----	-----	-----
		High Address		

(17) 無条件分岐命令 : BR

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
BR	!addr16	0010 1100	Low Address	High Address
	!!addr20	0000 1001	1110 Hi-w Add	Low Address
		-----	-----	-----
		High Address		
	rp	0000 0101	0100 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	rg	0000 0101	0100 0G <sub>2</sub> G <sub>1</sub> 1	
	[ rp ]	0000 0101	0110 1P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	[ rg ]	0000 0101	0110 0G <sub>2</sub> G <sub>1</sub> 1	
	\$ addr20	0001 0100	\$ addr20	
	\$ !addr20	0100 0011	\$ addr Low	\$ addr High

(18) 条件付き分岐命令 : BNZ, BNE, BZ, BE, BNC, BNL, BC, BL, BNV, BPO, BV, BPE, BP, BN, BLT, BGE, BLE, BGT, BNH, BH, BF, BT, BTCLR, BFSET, DBNZ

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>BNZ</b>	\$ addr20	1000 0000	\$ addr20	
<b>BNE</b>				
<b>BZ</b>	\$ addr20	1000 0001	\$ addr20	
<b>BE</b>				
<b>BNC</b>	\$ addr20	1000 0010	\$ addr20	
<b>BNL</b>				
<b>BC</b>	\$ addr20	1000 0011	\$ addr20	
<b>BL</b>				
<b>BNV</b>	\$ addr20	1000 0100	\$ addr20	
<b>BPO</b>				
<b>BV</b>	\$ addr20	1000 0101	\$ addr20	
<b>BPE</b>				
<b>BP</b>	\$ addr20	1000 0110	\$ addr20	
<b>BN</b>	\$ addr20	1000 0111	\$ addr20	
<b>BLT</b>	\$ addr20	0000 0111	1111 1000	\$ addr20
<b>BGE</b>	\$ addr20	0000 0111	1111 1001	\$ addr20
<b>BLE</b>	\$ addr20	0000 0111	1111 1010	\$ addr20
<b>BGT</b>	\$ addr20	0000 0111	1111 1011	\$ addr20
<b>BNH</b>	\$ addr20	0000 0111	1111 1100	\$ addr20
<b>BH</b>	\$ addr20	0000 0111	1111 1101	\$ addr20
<b>BF</b>	saddr2. bit, \$ addr20	0000 1000	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset
		----- \$ addr20		
	saddr1. bit, \$ addr20	0011 1100	0000 1000	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Saddr1-offset	\$ addr20	
	sfr. bit, \$ addr20	0000 1000	1010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset
		----- \$ addr20		
	X. bit, \$ addr20	0000 0011	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	A. bit, \$ addr20	0000 0011	1010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWL. bit, \$ addr20	0000 0010	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWH. bit, \$ addr20	0000 0010	1010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ TDE ] . bit, \$ addr20	0011 1101	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ WHL ] . bit, \$ addr20	0011 1101	1010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	!addr16. bit, \$ addr20	0000 1001	1101 0000	1010 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
		----- Low Address	High Address	\$ addr20
!!addr24. bit, \$ addr20	0000 1001	1101 0000	1010 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	
	----- High-w Address	Low Address	High Address	
	----- \$ addr20			

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>BT</b>	saddr2. bit, \$ addr20	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset	\$ addr20
	saddr1. bit, \$ addr20	0011 1100 ----- \$ addr20	0111 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr1-offset ----- \$ addr20
	sfr. bit, \$ addr20	0000 1000 ----- \$ addr20	1011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset ----- \$ addr20
	X. bit, \$ addr20	0000 0011	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	A. bit, \$ addr20	0000 0011	1011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWL. bit, \$ addr20	0000 0010	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWH. bit, \$ addr20	0000 0010	1011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ TDE ] . bit, \$ addr20	0011 1101	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ WHL ] . bit, \$ addr20	0011 1101	1011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	!addr16. bit, \$ addr20	0000 1001 ----- Low Address	1101 0000 ----- High Address	1011 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- \$ addr20
	!!addr24. bit, \$ addr20	0000 1001 ----- High-w Address ----- \$ addr20	1101 0000 ----- Low Address	1011 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address
<b>BTCLR</b>	saddr2. bit, \$ addr20	0000 1000 ----- \$ addr20	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset ----- \$ addr20
	saddr1. bit, \$ addr20	0011 1100 ----- Saddr1-offset	0000 1000 ----- \$ addr20	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- \$ addr20
	sfr. bit, \$ addr20	0000 1000 ----- \$ addr20	1101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset ----- \$ addr20
	X. bit, \$ addr20	0000 0011	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	A. bit, \$ addr20	0000 0011	1101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWL. bit, \$ addr20	0000 0010	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWH. bit, \$ addr20	0000 0010	1101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ TDE ] . bit, \$ addr20	0011 1101	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ WHL ] . bit, \$ addr20	0011 1101	1101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	!addr16. bit, \$ addr20	0000 1001 ----- Low Address	1101 0000 ----- High Address	1101 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- \$ addr20
	!!addr24. bit, \$ addr20	0000 1001 ----- High-w Address ----- \$ addr20	1101 0000 ----- Low Address	1101 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address
<b>BFSET</b>	saddr2. bit, \$ addr20	0000 1000 ----- \$ addr20	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Saddr2-offset ----- \$ addr20
	saddr1. bit, \$ addr20	0011 1100 ----- Saddr1-offset	0000 1000 ----- \$ addr20	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- \$ addr20
	sfr. bit, \$ addr20	0000 1000 ----- \$ addr20	1100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	Sfr-offset ----- \$ addr20

(次ページに続く)

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>BFSET</b>	X. bit, \$ addr20	0000 0011	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	A. bit, \$ addr20	0000 0011	1100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWL. bit, \$ addr20	0000 0010	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	PSWH. bit, \$ addr20	0000 0010	1100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ TDE ] . bit, \$ addr20	0011 1101	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	[ WHL ] . bit, \$ addr20	0011 1101	1100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	\$ addr20
	!addr16. bit, \$ addr20	0000 1001 ----- Low Address	1101 0000 ----- High Address	1100 0B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- \$ addr20
	!!addr24. bit, \$ addr20	0000 1001 ----- High-w Address ----- \$ addr20	1101 0000 ----- Low Address -----	1100 1B <sub>2</sub> B <sub>1</sub> B <sub>0</sub> ----- High Address -----
<b>DBNZ</b>	B, \$ addr20	0011 0011	\$ addr20	
	C, \$ addr20	0011 0010	\$ addr20	
	saddr2, \$ addr20	0011 1011	Saddr2-offset	\$ addr20
	saddr1, \$ addr20	0011 1100 ----- \$ addr20	0011 1011	Saddr1-offset -----

(19) CPU制御命令 : MOV, LOCATION, SEL, SWRS, NOP, EI, DI

ニモニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
<b>MOV</b>	STBC, #byte	0000 1001 ----- #byte	1100 0000	#byte
	WDM, #byte	0000 1001 ----- #byte	1100 0010	#byte
<b>LOCATION</b>	locaddr	0000 1001 ----- locaddrh	1100 0001	locaddrl -----
<b>SEL</b>	RBn	0000 0101	1010 1E <sub>2</sub> E <sub>1</sub> E <sub>0</sub>	
	RBn, ALT	0000 0101	1011 1E <sub>2</sub> E <sub>1</sub> E <sub>0</sub>	
<b>SWRS</b>		0000 0101	1111 1100	
<b>NOP</b>		0000 0000		
<b>EI</b>		0100 1011		
<b>DI</b>		0100 1010		

## (20) 特殊命令 : CHKL, CHKLA

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
CHKL	sfr	0000 0111	1100 1000	Sfr address
CHKLA	sfr	0000 0111	1100 1001	Sfr address

**注意**  $\mu$  PD784216, 784216Y, 784218, 784218Y, 784225, 784225Y, 784937サブシリーズの製品は、CHKL, CHKLAをサポートしていません。これらの命令は、使用しないでください。

これらの命令を実行した場合、次のような動作をします。

- CHKL命令・・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Zフラグがセット(1)されます。
- CHKLA命令・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Aレジスタにその結果を格納するとともに、Zフラグをセット(1)します。

(21) ストリング命令 : MOVTLBW, MOVW, MOVBK, XCHM, XCHBK, CMPME, CMPBKE, CMPMNE, CMPBKNE, CMPMC, CMPBKC, CMPMNC, CMPBKNC

二モニック	オペランド	命令コード		
		B1	B2	B3
		B4	B5	B6
		B7		
MOVTLBW	!addr8, byte	0000 1001	1010 0000	Low Address
		byte		
MOVW	[ TDE + ], A	0001 0101	0000 0000	
	[ TDE - ], A	0001 0101	0001 0000	
MOVBK	[ TDE + ], [ WHL + ]	0001 0101	0010 0000	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0000	
XCHM	[ TDE + ], A	0001 0101	0000 0001	
	[ TDE - ], A	0001 0101	0001 0001	
XCHBK	[ TDE + ], [ WHL + ]	0001 0101	0010 0001	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0001	
CMPME	[ TDE + ], A	0001 0101	0000 0100	
	[ TDE - ], A	0001 0101	0001 0100	
CMPBKE	[ TDE + ], [ WHL + ]	0001 0101	0010 0100	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0100	
CMPMNE	[ TDE + ], A	0001 0101	0000 0101	
	[ TDE - ], A	0001 0101	0001 0101	
CMPBKNE	[ TDE + ], [ WHL + ]	0001 0101	0010 0101	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0101	
CMPMC	[ TDE + ], A	0001 0101	0000 0111	
	[ TDE - ], A	0001 0101	0001 0111	
CMPBKC	[ TDE + ], [ WHL + ]	0001 0101	0010 0111	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0111	
CMPMNC	[ TDE + ], A	0001 0101	0000 0110	
	[ TDE - ], A	0001 0101	0001 0110	
CMPBKNC	[ TDE + ], [ WHL + ]	0001 0101	0010 0110	
	[ TDE - ], [ WHL - ]	0001 0101	0011 0110	

## 6.5 命令のクロック数

### 6.5.1 命令の実行時間

命令の実行時間は、 $f_{CLK}$ のクロック数で記載しています。

78K<sup>i</sup> シリーズのCPUは命令キューを内蔵しており、命令の実行と命令のプリフェッチ動作を並行して行うことができます。このため実質的な命令の実行時間は、その命令の実行前に実行した命令に依存します。

また、命令の実行時間はメモリ・アクセスに入れているウエイト数によっても変化します。したがって、これらの命令の実行クロック数を単純に加算しただけでは、プログラムの実行時間を正確に求めることはできません。

命令の実行クロック数は、BR命令やCALL命令、RET命令などの分岐動作を伴う命令を除いて最短の命令実行クロック数を記載しています。分岐を伴う命令の実行クロック数はやや長めのクロック数を記載しています。

### 6.5.2 クロック数欄の凡例

#### (1) 内部ROM

命令によりアクセスされるデータが内部ROMにあり、メモリ・マッピング・モード・レジスタ (MM) のビット7にあるIFCHビットがセット (1) されている場合。IFCHビットがクリア (0) されている場合は、PRAM/EMEM/SFRの欄を参照してください。

#### (2) IRAM

命令によりアクセスされるデータが内部高速RAM (LOCATION 0H命令実行時にはFD00H-FEFFFH, LOCATION 0FH命令実行時にはFFD00H-FFEFFFHの領域) の場合。

なお、 $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。

#### (3) PRAM/EMEM/SFR

命令によりアクセスされるデータが、内部RAMのうち、IRAMでない領域と、外部メモリ (外部SFRを含む)、SFRの場合。

#### (4) その他

命令によりアクセスされるデータがない場合。

### 6.5.3 クロック数欄の説明

#### (1) ワード・データ・アクセス時のクロック数

PRAM/EMEM/SFR欄にあるクロック数は、バス幅が16ビットで、偶数番地にデータが配置されている場合の値です。バス幅が8ビットの場合と、バス幅が16ビットで、奇数番地にデータが配置されている場合は、表中のクロック数に4を加えてください。なお、内部RAMのバス幅は16ビットになります。また、内部ROMのワード・データについても同様に奇数番地に配置されている場合は、4を加えてください。

スタック操作命令で“n”が記載されている命令で退避/復帰をするワード・データのアドレスが奇数番地になる場合は、“n”の係数に4を加えてください。

#### (2) 3バイト・データ・アクセス時のクロック数

PRAM/EMEM/SER欄にあるクロック数は、バス幅が16ビットの場合の値です。バス幅が8ビットの場合と、バス幅が16ビットで、奇数番地にデータが配置されている場合は表中のクロック数に4を加えてください。なお、内部RAMのバス幅は16ビットになります。

#### (3) “/”でクロック数が区切られている場合

クロック数が“/”で区切られている場合は、その命令のバイト数についても“/”で区切られています。命令のバイト数と同じ側のクロック数がその命令の実行時間になります。

#### (4) クロック数に“n”が記載されている場合

MACW/MACSW/MOVTBLW命令の場合、オペランドのbyteで指定した数を代入します。

SACW/MOVM/XCHM/MOVBK/XCHBK/CMPME/CMPMNE/CMPMC/CMPMNC/CMPBKE/CMPBKNE/

CMPBKC/CMPBKNC命令の場合は、命令の実行開始時にCレジスタに設定した値をnに代入します。

なお、このクロック数は、割り込みまたは、マクロ・サービスにより命令が中断されなかったときの値です。

シフト・ローテート命令の場合、シフトまたはローテートするビット数をnに代入します。

スタック操作命令の場合、スタックへ退避するレジスタまたは、スタックから復帰するレジスタの数をnに代入します。

## 6.5.4 クロック数一覧

## (1) 8ビット・データ転送命令 : MOV

(1/2)

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MOV	r, #byte	2/3		2/3		
	saddr, #byte	3/4		3/4	7	
	sfr, #byte	3			7	
	!addr16, #byte	5		7	9	
	!!addr24, #byte	6		8	10	
	r, r'	2/3		2/3		
	A, r	1/2				
	A, saddr2	2		3	7	
	r, saddr	3		4	8	
	saddr2, A	2		2	6	
	saddr, r	3		4	8	
	A, sfr	2			7	
	r, sfr	3			8	
	sfr, A	2			6	
	sfr, r	3			8	
	saddr, saddr'	4		6	14	
	r, !addr16	4	9	7	9	
	!addr16, r	4		6	8	
	r, !!addr24	5	10	8	10	
	!!addr24, r	5		7	9	
	A, [ saddrp ]	2/3	9/10	7/8	9/10	
	A, [ %saddrg ]	3/4	14/15	12/13	14/15	
	A, [ TDE + ]	1	9	7	9	
	A, [ WHL + ]	1				
	A, [ TDE - ]	1				
	A, [ WHL - ]	1				
	A, [ TDE ]	1	8	6	8	
	A, [ WHL ]	1				
	A, [ VVP ]	2	9	7	9	
	A, [ UUP ]	2				
	A, [ TDE + byte ]	3	10	8	10	
	A, [ SP + byte ]	3	11	9	11	
	A, [ WHL + byte ]	3	10	8	10	
A, [ UUP + byte ]	3					
A, [ VVP + byte ]	3					
A, imm24 [ DE ]	5	12	10	12		
A, imm24 [ A ]	5					
A, imm24 [ HL ]	5					
A, imm24 [ B ]	5					

ニモニック	オペランド	バイト	クロック数						
			内部ROM	IRAM	PRAM/EMEM/SFR	その他			
MOV	A, [ TDE + A ]	2	10	8	10				
	A, [ WHL + A ]	2							
	A, [ TDE + B ]	2							
	A, [ WHL + B ]	2							
	A, [ VVP + DE ]	2							
	A, [ VVP + HL ]	2							
	A, [ TDE + C ]	2							
	A, [ WHL + C ]	2							
	[ saddrp ], A	2/3						6/7	8/9
	[ %saddrg ], A	3/4						12/13	14/15
	[ TDE + ], A	1						8	10
	[ WHL + ], A	1							
	[ TDE - ], A	1							
	[ WHL - ], A	1							
	[ TDE ], A	1						5	7
	[ WHL ], A	1							
	[ VVP ], A	2						7	9
	[ UUP ], A	2							
	[ TDE + byte ], A	3						8	10
	[ SP + byte ], A	3	9	11					
	[ WHL + byte ], A	3	8	10					
	[ UUP + byte ], A	3							
	[ VVP + byte ], A	3							
	imm24 [ DE ], A	5	10	12					
	imm24 [ A ], A	5							
	imm24 [ HL ], A	5							
	imm24 [ B ], A	5							
	[ TDE + A ], A	2	8	10					
	[ WHL + A ], A	2							
	[ TDE + B ], A	2							
	[ WHL + B ], A	2							
	[ VVP + DE ], A	2							
	[ VVP + HL ], A	2							
	[ TDE + C ], A	2							
	[ WHL + C ], A	2							
	PSWL, #byte	3						7	
	PSWH, #byte	3							
	PSWL, A	2	6						
	PSWH, A	2							
	A, PSWL	2	7						
A, PSWH	2								
r3, #byte	3	3							
A, r3	2	4							
r3, A	2	3							

## (2) 16ビット・データ転送命令：MOVW

( 1/2 )

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MOVW	rp, #word	3		3		
	saddrp, #word	4/5		4	8	
	sfrp, #word	4				
	laddr16, #word	6		8	10	
	!!addr24, #word	7		9	11	
	rp, rp'	2		2		
	AX, saddrp2	2		3	7	
	rp, saddrp	3		4	8	
	saddrp2, AX	2		2	6	
	saddrp, rp	3		3	7	
	AX, sfrp	2			7	
	rp, sfrp	3			8	
	sfrp, AX	2			6	
	sfrp, rp	3			7	
	saddrp, saddrp'	4		6	14	
	rp, laddr16	4	4	9	7	9
	laddr16, rp	4			6	8
	rp, !!addr24	5		10	8	10
	!!addr24, rp	5			7	9
	AX, [ saddrp ]	3/4		10/11	8/9	10/11
	AX, [ %saddrg ]	3/4		14/15	12/13	14/15
	AX, [ TDE + ]	2		11	9	11
	AX, [ WHL + ]	2				
	AX, [ TDE - ]	2				
	AX, [ WHL - ]	2				
	AX, [ TDE ]	2		9	7	9
	AX, [ WHL ]	2				
	AX, [ VVP ]	2				
	AX, [ UUP ]	2				
	AX, [ TDE + byte ]	3		10	8	10
	AX, [ SP + byte ]	3		11	9	11
	AX, [ WHL + byte ]	3		10	8	10
AX, [ UUP + byte ]	3					
AX, [ VVP + byte ]	3					
AX,imm24 [ DE ]	5		12	10	12	
AX, imm24 [ A ]	5					
AX, imm24 [ HL ]	5					
AX, imm24 [ B ]	5					

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MOVW	AX, [ TDE + A ]	2	10	8	10	
	AX, [ WHL + A ]	2				
	AX, [ TDE + B ]	2				
	AX, [ WHL + B ]	2				
	AX, [ VVP + DE ]	2				
	AX, [ VVP + HL ]	2				
	AX, [ TDE + C ]	2				
	AX, [ WHL + C ]	2				
	[ saddrg ], AX	3/4		8/9	10/11	
	[ %saddrg ], AX	3/4		12/13	14/15	
	[ TDE + ], AX	2		9	11	
	[ WHL + ], AX	2		7	9	
	[ TDE - ], AX	2				
	[ WHL - ], AX	2				
	[ TDE ], AX	2				
	[ WHL ], AX	2				
	[ VVP ], AX	2				
	[ UUP ], AX	2				
	[ TDE + byte ], AX	3				
	[ SP + byte ], AX	3		9	11	
	[ WHL + byte ], AX	3		8	10	
	[ UUP + byte ], AX	3		10	12	
	[ VVP + byte ], AX	3				
	imm24 [ DE ], AX	5				
	imm24 [ A ], AX	5				
	imm24 [ HL ], AX	5		8	10	
	imm24 [ B ], AX	5				
	[ TDE + A ], AX	2				
	[ WHL + A ], AX	2				
	[ TDE + B ], AX	2				
	[ WHL + B ], AX	2				
	[ VVP + DE ], AX	2				
	[ VVP + HL ], AX	2				
[ TDE + C ], AX	2					
[ WHL + C ], AX	2					

## (3) 24ビット・データ転送命令 : MOVG

( 1/2 )

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MOVG	rg, #imm24	5		5		
	rg, rg'	2		4		
	rg, !!addr24	5	17	13	17	
	!!addr24, rg	5		12	16	
	rg, saddrg	3		9	17	
	saddrg, rg	3		7	15	
	WHL, [ %saddrg ]	3/4	21/22	17/18	21/22	
	[ %saddrg ], WHL	3/4				
	WHL, [ TDE + ]	2	19	15	19	
	WHL, [ TDE - ]	2				
	WHL, [ TDE ]	2	16	12	16	
	WHL, [ WHL ]	2				
	WHL, [ VVP ]	2				
	WHL, [ UUP ]	2				
	WHL, [ TDE + byte ]	3	17	13	17	
	WHL, [ SP + byte ]	3	18	14	18	
	WHL, [ WHL + byte ]	3	17	13	17	
	WHL, [ UUP + byte ]	3				
	WHL, [ VVP + byte ]	3				
	WHL, imm24 [ DE ]	5	19	15	19	
	WHL, imm24 [ A ]	5				
	WHL, imm24 [ HL ]	5				
	WHL, imm24 [ B ]	5				
	WHL, [ TDE + A ]	2	17	13	17	
	WHL, [ WHL + A ]	2				
	WHL, [ TDE + B ]	2				
	WHL, [ WHL + B ]	2				
	WHL, [ VVP + DE ]	2				
	WHL, [ VVP + HL ]	2				
	WHL, [ TDE + C ]	2				
	WHL, [ WHL + C ]	2				
	[ TDE + ], WHL	2		15	19	
	[ TDE - ], WHL	2				
[ TDE ], WHL	2		12	16		
[ WHL ], WHL	2					
[ VVP ], WHL	2					
[ UUP ], WHL	2					
[ TDE + byte ], WHL	3		13	17		
[ SP + byte ], WHL	3		14	18		

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>MOVG</b>	[ WHL + byte ], WHL	3		13	17	
	[ UUP + byte ], WHL	3		15	19	
	[ VVP + byte ], WHL	3				
	imm24 [ DE ], WHL	5				
	imm24 [ A ], WHL	5				
	imm24 [ HL ], WHL	5				
	imm24 [ B ], WHL	5				
	[ TDE + A ], WHL	2		13	17	
	[ WHL + A ], WHL	2				
	[ TDE + B ], WHL	2				
	[ WHL + B ], WHL	2				
	[ VVP + DE ], WHL	2				
	[ VVP + HL ], WHL	2				
	[ TDE + C ], WHL	2				
	[ WHL + C ], WHL	2				

## (4) 8ビット・データ交換命令 : XCH

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
XCH	r, r'	2/3		4		
	A, r	1/2		4/5		
	A, saddr2	2		5	13	
	r, saddr	3		6		14
	r, sfr	3				14
	saddr, saddr'	4		8		24
	r, !addr16	4		11		15
	r, !!addr24	5				
	A, [ saddrp ]	2/3		8/9		10/11
	A, [ %saddrg ]	3/4		17/18		21/22
	A, [ TDE + ]	2		14		18
	A, [ WHL + ]	2				
	A, [ TDE - ]	2				
	A, [ WHL - ]	2				
	A, [ TDE ]	2		12		16
	A, [ WHL ]	2				
	A, [ VVP ]	2				
	A, [ UUP ]	2				
	A, [ TDE + byte ]	3		13		17
	A, [ SP + byte ]	3		14		18
	A, [ WHL + byte ]	3		13		17
	A, [ UUP + byte ]	3				
	A, [ VVP + byte ]	3				
	A, imm24 [ DE ]	5		15		19
	A, imm24 [ A ]	5				
	A, imm24 [ HL ]	5				
	A, imm24 [ B ]	5				
	A, [ TDE + A ]	2		13		17
	A, [ WHL + A ]	2				
	A, [ TDE + B ]	2				
A, [ WHL + B ]	2					
A, [ VVP + DE ]	2					
A, [ VVP + HL ]	2					
A, [ TDE + C ]	2					
A, [ WHL + C ]	2					

## (5) 16ビット・データ交換命令：XCHW

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
XCHW	rp, rp'	2		4		
	AX, saddrp2	2		5	13	
	rp, saddrp	3		6	14	
	rp, sfrp	3			14	
	AX, [ saddrp ]	3/4		13/14	17/18	
	AX, [ %saddrp ]	3/4		17/18	21/22	
	AX, !addr16	4		3	3	
	AX, !!addr24	5		4	4	
	saddrp, saddrp'	4		8	24	
	AX, [ TDE + ]	2		14	18	
	AX, [ WHL + ]	2				
	AX, [ TDE - ]	2				
	AX, [ WHL - ]	2				
	AX, [ TDE ]	2		12	16	
	AX, [ WHL ]	2				
	AX, [ VVP ]	2				
	AX, [ UUP ]	2				
	AX, [ TDE + byte ]	3		13	17	
	AX, [ SP + byte ]	3		14	18	
	AX, [ WHL + byte ]	3		13	17	
	AX, [ UUP + byte ]	3				
	AX, [ VVP + byte ]	3				
	AX, imm24 [ DE ]	5		15	19	
	AX, imm24 [ A ]	5				
	AX, imm24 [ HL ]	5				
	AX, imm24 [ B ]	5				
	AX, [ TDE + A ]	2		13	17	
	AX, [ WHL + A ]	2				
	AX, [ TDE + B ]	2				
	AX, [ WHL + B ]	2				
	AX, [ VVP + DE ]	2				
	AX, [ VVP + HL ]	2				
AX, [ TDE + C ]	2					
AX, [ WHL + C ]	2					

## (6) 8ビット演算命令 : ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP

(1/4)

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
ADD	A, #byte	2		2		
ADDC	r, #byte	3		4		
SUB	saddr, #byte	3/4		6/7	12/13	
SUBC	sfr, #byte	4			13	
AND	r, r'	2/3		3/4		
OR	A, saddr2	4		3	7	
XOR	r, saddr	3		4	8	
	saddr, r	3		8	14	
	r, sfr	3			8	
	sfr, r	3			14	
	saddr, saddr'	4		8	18	
	A, [ saddrp ]	3/4	11/12	9/10	11/12	
	A, [ %saddrg ]	3/4	15/16	13/14	15/16	
	[ saddrp ], A	3/4		11/12	15/16	
	[ %saddrg ], A	3/4		15/16	10/20	
	A, !addr16	4	10	8	10	
	A, !!addr24	5	11	9	11	
	!addr16, A	4		10	14	
	!!addr24, A	5		11	15	
	A, [ TDE + ]	1	11	9	11	
	A, [ WHL + ]	1				
	A, [ TDE - ]	1				
	A, [ WHL - ]	1				
	A, [ TDE ]	1	10	8	10	
	A, [ WHL ]	1				
	A, [ VVP ]	2				
	A, [ UUP ]	2				
	A, [ TDE + byte ]	3	12	10	12	
	A, [ SP + byte ]	3				
	A, [ WHL + byte ]	3				
	A, [ UUP + byte ]	3				
	A, [ VVP + byte ]	3				
	A, imm24 [ DE ]	5	13	11	13	
	A, imm24 [ A ]	5				
	A, imm24 [ HL ]	5				
	A, imm24 [ B ]	5				

ニモニック	オペランド	バイト	クロック数						
			内部ROM	IRAM	PRAM/EMEM/SFR	その他			
<b>ADD</b>	A, [ TDE + A ]	2	11	9	11				
<b>ADDC</b>	A, [ WHL + A ]	2							
<b>SUB</b>	A, [ TDE + B ]	2	11	9	11				
<b>SUBC</b>	A, [ WHL + B ]	2							
<b>AND</b>	A, [ VVP + DE ]	2	11	9	11				
<b>OR</b>	A, [ VVP + HL ]	2							
<b>XOR</b>	A, [ TDE + C ]	2	11	9	11				
	A, [ WHL + C ]	2							
	[ TDE + ], A	1	内部ROM	10	14				
	[ WHL + ], A	1							
	[ TDE - ], A	1							
	[ WHL - ], A	1							
	[ TDE ], A	1							
	[ WHL ], A	1							
	[ VVP ], A	2							
	[ UUP ], A	2							
	[ TDE + byte ], A	3					13	17	
	[ SP + byte ], A	3							
	[ WHL + byte ], A	3							
	[ UUP + byte ], A	3							
	[ VVP + byte ], A	3							
	[ WHL + byte ], A	3							
	imm24 [ DE ], A	5	14	18					
	imm24 [ A ], A	5							
	imm24 [ HL ], A	5							
	imm24 [ B ], A	5							
	[ TDE + A ], A	2	12	16					
	[ WHL + A ], A	2							
	[ TDE + B ], A	2							
	[ WHL + B ], A	2							
	[ VVP + DE ], A	2							
	[ VVP + HL ], A	2							
	[ TDE + C ], A	2							
	[ WHL + C ], A	2							

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
CMP	A, #byte	2		2		
	r, #byte	3		4		
	saddr, #byte	3/4		4/5	8/9	
	sfr, #byte	4			9	
	r, r'	2/3		3/4		
	A, saddr2	4		3	7	
	r, saddr	3		4	8	
	saddr, r	3		6	10	
	r, sfr	3			9	
	sfr, r	3			10	
	saddr, saddr'	4		6	14	
	A, [ saddrp ]	3/4	11/12	9/10	11/12	
	A, [ %saddrg ]	3/4	15/16	13/14	15/16	
	[ saddrp ], A	3/4	11/12	9/10	11/12	
	[ %saddrg ], A	3/4	15/16	13/14	15/16	
	A, !addr16	4	10	8	10	
	A, !!addr24	5	11	9	11	
	!addr16, A	4	10	8	10	
	!!addr24, A	5	11	9	11	
	A, [ TDE + ]	1	11	9	11	
	A, [ WHL + ]	1				
	A, [ TDE - ]	1				
	A, [ WHL - ]	1				
	A, [ TDE ]	1	10	8	10	
	A, [ WHL ]	1				
	A, [ VVP ]	2				
	A, [ UUP ]	2				
	A, [ TDE + byte ]	3	12	10	12	
	A, [ SP + byte ]	3				
	A, [ WHL + byte ]	3				
A, [ UUP + byte ]	3					
A, [ VVP + byte ]	3					
A, imm24 [ DE ]	5	13	11	13		
A, imm24 [ A ]	5					
A, imm24 [ HL ]	5					
A, imm24 [ B ]	5					

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
CMP	A, [ TDE + A ]	2	11	9	11	
	A, [ WHL + A ]	2				
	A, [ TDE + B ]	2				
	A, [ WHL + B ]	2				
	A, [ VVP + DE ]	2				
	A, [ VVP + HL ]	2				
	A, [ TDE + C ]	2				
	A, [ WHL + C ]	2				
	[ TDE + ], A	1	10	8	10	
	[ WHL + ], A	1				
	[ TDE - ], A	1				
	[ WHL - ], A	1				
	[ TDE ], A	1				
	[ WHL ], A	1				
	[ VVP ], A	2				
	[ UUP ], A	2				
	[ TDE + byte ], A	3	13	11	13	
	[ SP + byte ], A	3				
	[ WHL + byte ], A	3				
	[ UUP + byte ], A	3				
	[ VVP + byte ], A	3				
	imm24 [ DE ], A	5	14	12	14	
	imm24 [ A ], A	5				
	imm24 [ HL ], A	5				
	imm24 [ B ], A	5				
	[ TDE + A ], A	2	12	10	12	
	[ WHL + A ], A	2				
	[ TDE + B ], A	2				
	[ WHL + B ], A	2				
	[ VVP + DE ], A	2				
	[ VVP + HL ], A	2				
	[ TDE + C ], A	2				
[ WHL + C ], A	2					

(7) 16ビット演算命令 : ADDW, SUBW, CMPW

ニモニク	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>ADDW</b> <b>SUBW</b>	AX, #word	3		3		
	rp, #word	4		5		
	rp, rp'	2		3		
	AX, saddrp2	2			7	
	rp, saddrp	3		5	9	
	saddrp, rp	3		8	14	
	rp, sfrp	3			9	
	sfrp, rp	3			13	
	saddrp, #word	4/5		7/8		
	sfrp, #word	5			14	
	saddrp, saddrp'	4		8	20	
	<b>CMPW</b>	AX, #word	3		3	
rp, #word		4		5		
rp, rp'		2		3		
AX, saddrp2		2			7	
rp, saddrp		3		5	9	
saddrp, rp		3				
rp, sfrp		3				
sfrp, rp		3				
saddrp, #word		4/5		5/6	9	
sfrp, #word		5			10	
saddrp, saddrp'		4		6		

(8) 24ビット演算命令 : ADDG, SUBG

ニモニク	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>ADDG</b> <b>SUBG</b>	rg, rg'	2		6		
	rg, #imm24	5		8		
	WHL, saddrg	3		13	19	

(9) 乗除算命令 : MULU, MULUW, MULW, DIVUW, DIVUX

ニモニク	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>MULU</b>	r	2/3		11/12		
<b>MULUW</b>	rp	2		15		
<b>MULW</b>	rp	2		14		
<b>DIVUW</b>	r	2/3		23/24		
<b>DIVUX</b>	rp	2		43		

(10) 特殊演算命令 : MACW, MACSW, SACW

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MACW	byte	3		5 + 21n		
MACSW	byte	3		5 + 21n		
SACW	[ TDE + ], [ WHL + ]	4		4 + 19n	4 + 23n	

(11) 増減命令 : INC, DEC, INCW, DECW, INCG, DECG

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
INC	r	1/2		2/3		
DEC	saddr	2/3		5/6	11/12	
INCW	rp	2/1		3/2		
DECW	saddrp	3/4		6/7	12/13	
INCG	rg	2		4		
DECG						

(12) 補正命令 : ADJBA, ADJBS, CVTBW

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
ADJBA		2		5		
ADJBS		2		5		
CVTBW		1		3		

(13) シフト・ローテート命令 : ROR, ROL, RORC, ROLC, SHR, SHL, SHRW, SHLW, ROR4, ROL4

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
ROR ROL RORC ROLC SHR SHL	r, n	2/3		5 + n/6 + n		
SHRW SHLW	rp, n	2		5 + n		
ROR4 ROL4	mem3	2		11	15	

## (14) ビット操作命令 : MOV1, AND1, OR1, XOR1, NOT1, SET1, CLR1

(1/2)

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>MOV1</b>	CY, saddr. bit	3/4		6/7	10/11	
	CY, sfr. bit	3			10	
	CY, X. bit	2		5		
	CY, A. bit	2				
	CY, PSWL. bit	2			5	
	CY, PSWH. bit	2				
	CY, [ TDE ] . bit	2	11	9	11	
	CY, [ WHL ] . bit	2				
	CY, !addr16. bit	5	16	14	16	
	CY, !!addr24. bit	6				
	saddr. bit, CY	3/4		5/6	13/14	
	sfr. bit, CY	3			13	
	X. bit, CY	2		6		
	A. bit, CY	2				
	PSWL. bit, CY	2			8	
	PSWH. bit, CY	2			7	
	[ TDE ] . bit, CY	2		10	14	
	[ WHL ] . bit, CY	2				
	!addr16. bit, CY	5		13	15	
	!!addr24. bit, CY	6				
<b>AND1</b> <b>OR1</b>	CY, saddr. bit	3/4		6/7	10/11	
	CY, /saddr. bit	3/4				
	CY, sfr. bit	3			10	
	CY, /sfr. bit	3				
	CY, X. bit	2		5		
	CY, /X. bit	2				
	CY, A. bit	2				
	CY, /A. bit	2				
	CY, PSWL. bit	2				
	CY, /PSWL. bit	2				
	CY, PSWH. bit	2				
	CY, /PSWH. bit	2				
	CY, [ TDE ] . bit	2	11	9	11	
	CY, / [ TDE ] . bit	2				
	CY, [ WHL ] . bit	2				
	CY, / [ WHL ] . bit	2				
	CY, !addr16. bit	5	16	14	16	
	CY, /!addr16. bit	5				
	CY, !!addr24. bit	6				
	CY, /!!addr24. bit	6				

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>XOR1</b>	CY, saddr. bit	3/4		6/7	10/11	
	CY, /sfr. bit	3			10	
	CY, X. bit	2		5		
	CY, A. bit	2				
	CY, PSWL. bit	2			5	
	CY, PSWH. bit	2				
	CY, [ TDE ] . bit	2	11	9	11	
	CY, [ WHL ] . bit	2				
	CY, !addr16. bit	5	16	14	16	
CY, !!addr24. bit	6					
<b>NOT1</b>	saddr. bit	3/4		5/6	13/14	
	sfr. bit	3			13	
	X. bit	2		5		
	A. bit	2				
	PSWL. bit	2			7	
	PSWH. bit	2			6	
	[ TDE ] . bit	2		10	14	
	[ WHL ] . bit	2				
	!addr16. bit	5		13	15	
	!!addr24. bit	6				
	CY	1			2	
<b>SET1 CLR1</b>	saddr. bit	2/3		4/5	12/13	
	sfr. bit	3			13	
	X. bit	2		5		
	A. bit	2				
	PSWL. bit	2			7	
	PSWH. bit	2			6	
	[ TDE ] . bit	2		10	14	
	[ WHL ] . bit	2				
	!addr16. bit	5		13	15	
	!!addr24. bit	6				
	CY	1			2	

(15) スタック操作命令 : PUSH, PUSHU, POP, POPU, MOVG, ADDWG, SUBWG, INCG, DECG

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
PUSH	PSW	1		5	7	
	sfrp	3		10	14	
	sfr	3				
	post	2		4 + 5n	4 + 7n	
	rg	2		12	16	
PUSHU	post	2		6 + 5n	6 + 7n	
POP	PSW	1	8	7	9	
	sfrp	3	15	14	16	
	sfr	3				
	post	2	4 + 8n	4 + 6n	4 + 8n	
	rg	2	17	13	17	
POPU	post	2	7 + 8n	7 + 6n	7 + 8n	
MOVG	SP, #imm24	5				5
	SP, WHL	2				
	WHL, SP	2				
ADDWG SUBWG	SP, #word	4				5
INCG DECG	SP	2				5

(16) コール・リターン命令 : CALL, CALLF, CALLT, BRK, BRKCS, RET, RETI, RETB, RETCS, RETCSB

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
CALL	!addr16	3		19	23	
	!!addr20	4		22	26	
	rp	2		20	24	
	rg	2		22	26	
	[ rp ]	2	30 <sup>注</sup>	24	30	
	[ rg ]	2	37 <sup>注</sup>	29	37	
	\$ !addr20	3		19	23	
CALLF	!addr11	2		19	23	
CALLT	[ addr5 ]	1	28 <sup>注</sup>	22	28	
BRK		1		23	29	
BRKCS	RBn	2				13
RET		1	21	17	21	
RETI		1	22	18	22	
RETB		1	21	17	21	
RETCS	!addr16	3				14
RETCSB	!addr16	4				14

注 スタックがPRAM/EMEMの場合

(17) 無条件分岐命令 : BR

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
BR	!addr16	3				11
	!!addr20	4				12
	rp	2				11
	rg	2				12
	[ rp ]	2	16	14	16	
	[ rg ]	2	22	18	22	
	\$ addr20	2				10
	\$ !addr20	3				11

(18) 条件付き分岐命令 : BNZ, BNE, BZ, BE, BNC, BNL, BC, BL, BNV, BPO, BV, BPE, BP, BN, BLT, BGE, BLE, BGT, BNH, BH, BF, BT, BTCLR, BFSET, DBNZ

( 1/3 )

二モニック	オペランド	バイト	クロック数				
			分岐 しない	分岐する			
				内部ROM	IRAM	PRAM/EMEM/SFR	その他
BNZ	\$ addr20	2	3				10
BNE							
BZ	\$ addr20	2	3				10
BE							
BNC	\$ addr20	2	3				10
BNL							
BC	\$ addr20	2	3				10
BL							
BNV	\$ addr20	2	3				10
BPO							
BV	\$ addr20	2	3				10
BPE							
BP	\$ addr20	2	3				10
BN							
BLT	\$ addr20	3	4				11
BGE	\$ addr20	3	4				11
BLE	\$ addr20	3	4				11
BGT	\$ addr20	3	4				11
BNH	\$ addr20	3	4				11
BH	\$ addr20	3	4				11

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
BF	saddr. bit, \$ addr20	4/5		14/15	18	
				7/8	11	
	sfr. bit, \$ addr20	4			18	
					11	
	X. bit, \$ addr20	3		13		
				6		
	A. bit, \$ addr20	3		13		
				6		
	PSWL. bit, \$ addr20	3			13	
					6	
	PSWH. bit, \$ addr20	3			13	
					6	
mem2. bit, \$ addr20	3	19	17	19		
		12	10	12		
!addr16. bit, \$ addr20	6		22	24		
			15	17		
!!addr24. bit, \$ addr20	7		22	24		
			15	17		
BT	saddr. bit, \$ addr20	3/4		13/14	17	
				6/7	10	
	sfr. bit, \$ addr20	4			18	
					11	
	X. bit, \$ addr20	3		13		
				6		
	A. bit, \$ addr20	3		13		
				6		
	PSWL. bit, \$ addr20	3			13	
					6	
	PSWH. bit, \$ addr20	3			13	
					6	
mem2. bit, \$ addr20	3	19	17	19		
		12	10	12		
!addr16. bit, \$ addr20	6		22	24		
			15	17		
!!addr24. bit, \$ addr20	7		22	24		
			15	17		

備考 クロック数は次の場合により異なるため、オペランドごとに上下2段に分けて記述しています。

上段：分岐する（内部ROM高速フェッチ時）

下段：分岐しない

ニモニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>BTCLR</b> <b>BFSET</b>	saddr. bit, \$ addr20	4/5		16/17	24	
				7/8	15	
	sfr. bit, \$ addr20	4			24	
					15	
	X. bit, \$ addr20	3		15		
				6		
	A. bit, \$ addr20	3		15		
				6		
	PSWL. bit, \$ addr20	3			15	
					6	
	PSWH. bit, \$ addr20	3			16	
					6	
mem2. bit, \$ addr20	3		21	25		
			12	16		
!addr16. bit, \$ addr20	6		24	26		
			15	17		
!!addr24. bit, \$ addr20	7		24	26		
			15	17		
<b>DBNZ</b>	B, \$ addr20	2	12			
			4			
	C, \$ addr20	2	12			
			4			
	saddr, \$ addr20	3	21	17	21	
			5	5	5	
4		22	18	22		
		6	6	6		

備考 クロック数は次の場合により異なるため、オペランドごとに上下2段に分けて記述しています。

上段：分岐する（内部ROM高速フェッチ時）

下段：分岐しない

## (19) CPU制御命令：MOV, LOCATION, SEL, SWRS, NOP, EI, DI

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
MOV	STBC, #byte	4				13
	WDM, #byte	4				
LOCATION	locaddr	4				13
SEL	RBn	2				3
	RBn, ALT	2				
SWRS		2				4
NOP		1				2
EI		1				2
DI		1				2

## (20) 特殊命令：CHKL, CHKLA

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
CHKL	sfr	3			14	
CHKLA	sfr	3			14	

- ★ 注意  $\mu$  PD784216A, 784216AY, 784218A, 784218AY, 784225, 784225Y, 784938Aサブシリーズの製品は、CHKL, CHKLAをサポートしていません。これらの命令は、使用しないでください。

これらの命令を実行した場合、次のような動作をします。

- CHKL命令・・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Zフラグがセット(1)されます。
- CHKLA命令・・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Aレジスタにその結果を格納するとともに、Zフラグをセット(1)します。

(21) スtring命令 : MOVTLBW, MOVVM, XCHM, MOVVK, XCHBK, CMPME, CMPMNE, CMPMC, CMPMNC, CMPBKE, CMPBKNE, CMPBKC, CMPBKNC

二モニック	オペランド	バイト	クロック数			
			内部ROM	IRAM	PRAM/EMEM/SFR	その他
<b>MOVTLBW</b>	!addr16, byte	4		7 + 5n		
<b>MOVVM</b>	[ TDE + ], A	2		3 + 8n	3 + 10n	
	[ TDE - ], A	2				
<b>XCHM</b>	[ TDE + ], A	2		3 + 14n	3 + 20n	
	[ TDE - ], A	2				
<b>MOVVK</b>	[ TDE + ], [ WHL + ]	2	3 + 17n <sup>注1</sup>	3 + 13n <sup>注2</sup>	3 + 17n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				
<b>XCHBK</b>	[ TDE + ], [ WHL + ]	2		3 + 21n <sup>注2</sup>	3 + 29n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				
<b>CMPME</b>	[ TDE + ], A	2	3 + 12n	3 + 10n	3 + 12n	
	[ TDE - ], A	2				
<b>CMPMNE</b>	[ TDE + ], A	2	3 + 12n	3 + 10n	3 + 12n	
	[ TDE - ], A	2				
<b>CMPMC</b>	[ TDE + ], A	2	3 + 12n	3 + 10n	3 + 12n	
	[ TDE - ], A	2				
<b>CMPMNC</b>	[ TDE + ], A	2	3 + 12n	3 + 10n	3 + 12n	
	[ TDE - ], A	2				
<b>CMPBKE</b>	[ TDE + ], [ WHL + ]	2	3 + 19n <sup>注1</sup>	3 + 15n <sup>注2</sup>	3 + 19n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				
<b>CMPBKNE</b>	[ TDE + ], [ WHL + ]	2	3 + 19n <sup>注1</sup>	3 + 15n <sup>注2</sup>	3 + 19n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				
<b>CMPBKC</b>	[ TDE + ], [ WHL + ]	2	3 + 19n <sup>注1</sup>	3 + 15n <sup>注2</sup>	3 + 19n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				
<b>CMPBKNC</b>	[ TDE + ], [ WHL + ]	2	3 + 19n <sup>注1</sup>	3 + 15n <sup>注2</sup>	3 + 19n <sup>注3</sup>	
	[ TDE - ], [ WHL - ]	2				

注1 . WHLレジスタで指定されるメモリが内部ROMで , TDEレジスタで指定されるメモリがPRAM/EMEMの場合

2 . 転送元 , 転送先のメモリが両方ともIRAMの場合

3 . 転送元 , 転送先のメモリが両方ともRRAM/EMEMの場合

〔メモ〕

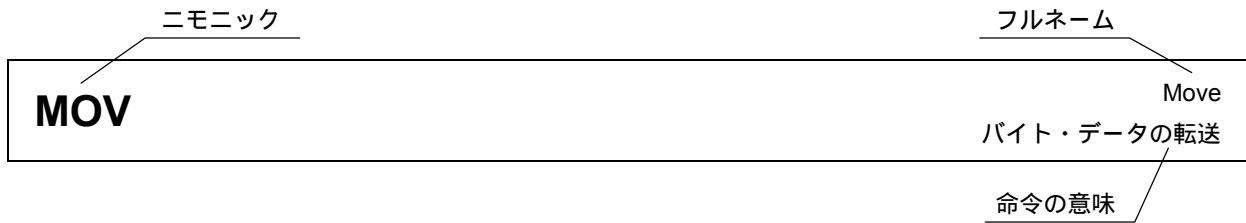
## 第7章 命令の説明

この章では、78K<sup>i</sup> シリーズ製品の命令の説明をします。各命令は、二モニック単位に、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のページに示します。

なお、命令のバイト数、命令コードについては、**第6章 命令セット**を参照してください。

# 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、第6章 命令セットを参照してください。

ニモニック	オペランド
MOV	r, #byte
	~ saddr, #byte
	A, saddr2
	~ saddr2, A
	A, men

ニモニック	オペランド
MOV	[ saddrp ], A
	~ [ %saddrg ], A
	men, A
	~ A, r3
	r3, 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。

S	Z	AC	P/V	CY

## 凡 例

記 号	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
x	結果に従ってセット/クリアされる
P	P/Vフラグがパリティ・フラグとして動作する
V	P/Vフラグがオーバーフロー・フラグとして動作する
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

## 7.1 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 ... 290

# 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
	!addr16, #byte
	!!addr24, #byte
	r, r
	A, r
	A, saddr2
	r, saddr
	saddr2, A
	saddr, r
	A, sfr
	r, sfr
	sfr, A
	sfr, r
	saddr, saddr
	r, !addr16
	!addr16, r

二モニック	オペランド (dst, src)
MOV	r, !!addr24
	!!addr24, r
	A, [ saddrp ]
	A, [ %saddrg ]
	A, mem
	[ saddrp ] , A
	[ %saddrg ] , A
	mem, A
	PSWL, #byte
	PSWH, #byte
	PSWL, A
	PSWH, A
	A, PSWL
	A, PSWH
	r3, #byte
	A, r3
	r3, A

【フラグ】

PSWL, #byteとPSWL, Aの

オペランドの場合

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

左記以外

S	Z	AC	P/V	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 PSWL, #byte命令, MOV PSWL, A命令と次に続く命令の間では、すべての割り込みまたはマクロ・サービスを受け付けません。

r3 (T, U, V, Wレジスタ) をオペランドとする命令は、78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合に、アドレスの上位8ビットを設定するときだけに使用するようになっています。また、可能であれば、r3を直接指定しなくてもよいようにプログラムを修正してください。

**【記 述 例】**

**MOV A, #4DH ; Aレジスタに4DHを転送**

## 7.2 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 ... 293

**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
	!addr16, #word
	!!addr24, #word
	rp, rp'
	AX, saddrp2
	rp, saddrp
	saddr2, AX
	saddrp, rp
	AX, sfrp
	rp, sfrp
	sfrp, AX

二モニック	オペランド (dst, src)
MOVW	sfrp, rp
	saddrp, saddrp'
	rp, !addr16
	!addr16, rp
	rp, !!addr24
	!!addr24, rp
	AX, [ saddrp ]
	AX, [ %saddrg ]
	AX, mem
	[ saddrg ] , AX
	[ %saddrg ] , AX
	mem, AX

【フラグ】

S	Z	AC	P/V	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。

78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合、次の全条件に該当するときは、次の点に留意してください。

**【条 件】**

- ・オペランドにrpを指定する命令を使用する
- ・rpとしてDE, HL, VP, UPを実際に記述する
- ・DE, HL, VP, UPをアドレスのポインタとして使用する

**【留 意 点】**

アドレス・ポインタの上位8ビットを示すT, W, V, Uレジスタの内容が下位16ビットを示すDE, HL, VP, UPと整合がとれていること, また, プログラムの修正が可能であれば24ビット操作命令を使用するように変更してください。

**【記 述 例】**

**MOVW AX, [ WHL ] ; WHLレジスタで示されるメモリの内容をAXレジスタへ転送**

## 7.3 24ビット・データ転送命令

24ビット・データ転送命令には、次の命令があります。

MOVG ... 296

**MOVG**Move G<sup>注</sup>  
24ビット・データの転送【命令形式】 **MOVG dst, src**

注 Gは、24ビット・データを操作することを表す文字です

【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)
<b>MOVG</b>	rg, #imm24
	rg, rg'
	rg, !!addr24
	!!addr24, rg
	rg, saddrg
	saddrg, rg
	WHL, [ %saddrg ]
	[ %saddrg ] , WHL
	WHL, mem1
	mem1, WHL

【フラグ】

S	Z	AC	P/V	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】

**MOVG VVP, SADG** ; ショート・ダイレクト・アドレッシングでアクセス可能なSADG番地にある24ビット長のデータをVVPレジスタへ転送

## 7.4 8ビット・データ交換命令

8ビット・データ交換命令には、次の命令があります。

XCH ... 298

**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】

二モニック	オペランド (dst, src)
XCH	r, r
	A, r
	A, saddr2
	r, saddr
	r, sfr
	saddr, saddr
	r, !addr16
	r, !!addr24
	A, [ saddrp ]
	A, [ %saddrg ]
	A, mem

## 【フラグ】

S	Z	AC	P/V	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 B, D ; Bレジスタの内容とDレジスタの内容を交換

## 7.5 16ビット・データ交換命令

16ビット・データ交換命令には、次の命令があります。

XCHW ... 300

**XCHW**Exchange  
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】

二モニック	オペランド (dst, src)
XCHW	rp, rp
	AX, saddrp2
	rp, saddrp
	rp, sfrp
	AX, [ saddrp ]
	AX, [%saddrg ]
	AX, !addr16
	AX, !!addr24
	saddrp, saddrp'
	AX, mem

【フラグ】

S	Z	AC	P/V	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。

78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合、次の全条件に該当するときは、次の点に留意してください。

【条件】

- ・オペランドにrpを指定する命令を使用する
- ・rpとしてDE, HL, VP, UPを実際に記述する
- ・DE, HL, VP, UPをアドレスのポインタとして使用する

【留意点】

アドレス・ポインタの上位8ビットを示すT, W, V, Uレジスタの内容が、下位16ビットを示すDE, HL, VP, UPと整合がとれていること、また、プログラムの修正が可能であれば24ビット操作命令を使用するように変更してください。

【記述例】

XCHW AX, mem ; AXレジスタの内容とメモリ・アドレッシングによりアドレスされるメモリの内容を交換

## 7.6 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 ... 302  
ADDC ... 304  
SUB ... 306  
SUBC ... 308  
CMP ... 310  
AND ... 312  
OR ... 313  
XOR ... 314

**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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
ADD	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

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

【説明】

第1オペランドで指定されるディスティネーション・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フラグとディスティネーション・オペランド (dst) へ格納します。

加算の結果、dstのビット7がセット (1) された場合は、Sフラグはセット (1) 、その他の場合はクリア (0) されます。

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

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。

加算の結果、ビット6からビット7へのキャリーが発生し、かつ、ビット7からのキャリーが発生しない場合 (2の補数形式での演算によるオーバーフロー発生時) 、またはビット6からビット7へのキャリーが発生せず、かつ、ビット7からのキャリーが発生した場合 (2の補数形式での演算によるアンダフロー発生時) 、P/Vフラグはセット (1) され、その他の場合はクリア (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。

**【記 述 例】**

**ADD CR11, #56H ; CR11レジスタの値と56Hを加算し、結果をCR11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)
<b>ADDC</b>	A, #byte
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
<b>ADDC</b>	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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のビット7がセット (1) された場合は、Sフラグはセット (1) 、その他の場合はクリア (0) されます。

加算の結果、dstが0になった場合、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) されず。

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。

加算の結果、ビット6からビット7へのキャリーが発生し、かつ、ビット7からのキャリーが発生しない場合 (2の補数形式での演算によるオーバーフロー発生時) 、またはビット6からビット7へのキャリーが発生せず、かつ、ビット7からのキャリーが発生した場合 (2の補数形式での演算によるアンダフロー発生時) 、P/Vフラグはセット (1) され、その他の場合はクリア (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。

**【記 述 例】**

**ADDC A, 12345H [ B ]** ; Aレジスタと ( 12345H + ( 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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
SUB	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

## 【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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のビット7がセット (1) された場合は、S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、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。

減算の結果、ビット6からビット7へのボローが発生し、かつ、ビット7でボローが発生しない場合 (2の補数形式での演算によるアンダフロー発生時) , または、ビット6からビット7へのボローが発生せず、かつ、ビット7でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時) , P/Vフラグはセット (1) され、その他の場合はクリア (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。

**【記 述 例】**

**SUB D, L ; DレジスタからL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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
SUBC	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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フラグを減算して、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。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のビット7がセット (1) された場合は、S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。

減算の結果、ビット6からビット7へのボローが発生し、かつ、ビット7でボローが発生しない場合 (2の補数形式での演算によるアンダフロー発生時) , または、ビット6からビット7へのボローが発生せず、かつ、ビット7でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時) , P/Vフラグはセット (1) され、その他の場合はクリア (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。

**【記 述 例】**

**SUBC A, [ TDE + ] ;** AレジスタからTDE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（減算後，TDEレジスタをインクリメント）

**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)
<b>CMP</b>	A, #byte
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
<b>CMP</b>	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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S, Z, AC, P/V, CYの各フラグだけを変化させます。

減算の結果、ビット7がセット (1) された場合は、S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、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。

減算の結果、ビット7でボローが発生し、かつ、ビット6でボローが発生しない場合 (2の補数形式での演算によるアンダフロー発生時)、または、ビット7でボローが発生せず、かつ、ビット6でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (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。

**【記 述 例】**

**CMP SADG1, SADG2** ; ショート・ダイレクト・アドレッシングでアクセス可能なSADG1番地の内容から、ショート・ダイレクト・アドレッシングでアクセス可能なSADG2番地の内容を減算し、フラグだけを変化させる（SADG1番地の内容とSADG2番地の内容の比較）

**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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
AND	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x		P	

【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、dstのビット7がセット(1)された場合は、Sフラグはセット(1)、その他の場合はクリア(0)されます。

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。

論理積をとった結果、dstの中のセット(1)されたビット数が偶数の場合は、P/V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】

**AND SADG, #11011100B** ; ショート・ダイレクト・アドレッシングでアクセス可能なSADG番地の内容と11011100Bのビットごとの論理積をとり、結果をSADG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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
OR	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x		P	

【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、dstのビット7がセット (1) された場合は、Sフラグはセット (1)、その他の場合はクリア (0) されます。

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。

論理和をとった結果、dstの中のセット (1) されたビット数が偶数の場合は、P/V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, !!12345H ; Aレジスタと12345H番地の内容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
	r, #byte
	saddr, #byte
	sfr, #byte
	r, r'
	A, saddr2
	r, saddr
	saddr, r
	r, sfr
	sfr, r
	saddr, saddr'

二モニック	オペランド (dst, src)
XOR	A, [ saddrp ]
	A, [ %saddrg ]
	[ saddrp ] , A
	[ %saddrg ] , A
	A, !addr16
	A, !!addr24
	!addr16, A
	!!addr24, A
	A, mem
	mem, A

【フラグ】

S	Z	AC	P/V	CY
x	x		P	

【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。

排他的論理和をとった結果、dstのビット7がセット (1) された場合は、Sフラグはセット (1)、その他の場合はクリア (0) されます。

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。

排他的論理和をとった結果、dstの中のセット (1) されたビット数が偶数の場合は、P/V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 C, P2 ; CレジスタとP2レジスタのビットごとの排他的論理和をとり、結果をCレジスタへ格納

## 7.7 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 ... 316

SUBW ... 318

CMPW ... 320

**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
	rp, #word
	rp, rp'
	AX, saddrp2
	rp, saddrp
	saddrp, rp
	rp, sfrp
	sfrp, rp
	saddrp, #word
	sfrp, #word
	saddrp, saddrp'

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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のビット15がセット (1) された場合は、Sフラグはセット (1)、その他の場合はクリア (0) されます。

加算の結果、dstが0になった場合、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) されません。

加算の結果、ACフラグは不定となります。

加算の結果、ビット14からビット15へのキャリーが発生し、かつ、ビット15からのキャリーが発生しない場合 (2の補数形式での演算によるオーバーフロー発生時)、または、ビット14からビット15へのキャリーが発生せず、かつ、ビット15からのキャリーが発生した場合 (2の補数形式での演算によるアンダフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (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。

78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合、次の全条件に該当するときは、次の点に留意してください。

**【条 件】**

- ・オペランドにrpを指定する命令を使用する
- ・rpとしてDE, HL, VP, UPを実際に記述する
- ・DE, HL, VP, UPをアドレスのポインタとして使用する

**【留 意 点】**

アドレス・ポインタの上位8ビットを示すT, W, V, Uレジスタの内容が、下位16ビットを示すDE, HL, VP, UPと整合がとれていること、また、プログラムの修正が可能であれば24ビット操作命令を使用するように変更してください。

**【記 述 例】**

**ADDW BC, #0ABCDH ; BCレジスタと0ABCDHを加算し、結果をBC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】

二モニック	オペランド (dst, src)
SUBW	AX, #word
	rp, #word
	rp, rp'
	AX, saddrp2
	rp, saddrp
	saddrp, rp
	rp, sfrp
	sfrp, rp
	saddrp, #word
	sfrp, #word
	saddrp, saddrp'

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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のビット15がセット (1) された場合は、S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、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、ACフラグは不定となります。

減算の結果、ビット14からビット15へのボローが発生し、かつ、ビット15でボローが発生しない場合 (2の補数形式での演算によるアンダフロー発生時)、または、ビット14からビット15へのボローが発生せず、かつ、ビット15でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (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。

78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合，次の全条件に該当するときは，次の点に留意してください。

**【条 件】**

- ・オペランドにrpを指定する命令を使用する
- ・rpとしてDE, HL, VP, UPを実際に記述する
- ・DE, HL, VP, UPをアドレスのポインタとして使用する

**【留 意 点】**

アドレス・ポインタの上位8ビットを示すT, W, V, Uレジスタの内容が，下位16ビットを示すDE, HL, VP, UPと整合がとれていること，また，プログラムの修正が可能であれば24ビット操作命令を使用するように変更してください。

**【記 述 例】**

**SUBW CR01, AX ; CR01レジスタの内容からAXレジスタの内容を減算し，結果をCR01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)
<b>CMPW</b>	AX, #word
	rp, #word
	rp, rp'
	AX, saddrp2
	rp, saddrp
	saddrp, rp
	rp, sfrp
	sfrp, rp
	saddrp, #word
	sfrp, #word
	saddrp, saddrp'

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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。減算の結果はどこでも格納せずにZ, AC, CYの各フラグだけを変化させます。

減算の結果、ビット15がセット (1) された場合は、Sフラグはセット (1)、その他の場合はクリアされます。

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、ACフラグは不定となります。

減算の結果、ビット14からビット15へのボローが発生し、かつ、ビット15でボローが発生しない場合 (2の補数形式での演算によるアングフロー発生時)、または、ビット14からビット15へのボローが発生せず、かつ、ビット15でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (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。

78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合、次の全条件に該当するときは、次の点に留意してください。

## 【条 件】

- ・オペランドにrpを指定する命令を使用する
- ・rpとしてDE, HL, VP, UPを実際に記述する
- ・DE, HL, VP, UPをアドレスのポインタとして使用する

## 【留 意 点】

アドレス・ポインタの上位8ビットを示すT, W, V, Uレジスタの内容が, 下位16ビットを示すDE, HL, VP, UPと整合がとれていること, また, プログラムの修正が可能であれば24ビット操作命令を使用するように変更してください。

## 【記 述 例】

**CMPW AX, SADG**; AXレジスタからショート・ダイレクト・アドレッシングでアクセス可能なSADG番地のワード・データを減算し, フラグだけを変化させる (AXレジスタとSADG番地のワード・データとの比較)

## 7.8 24ビット演算命令

24ビット演算命令には、次の命令があります。

ADDG ... 323

SUBG ... 324

**ADDG**Add G<sup>注</sup>

24ビット・データの加算

【命令形式】 **ADDG dst, src**注 Gは24ビット・データを操作  
することを表す文字です【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)
ADDG	rg, rg'
	rg, #imm24
	WHL, saddrg

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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) を加算します。加算の結果はdstへ格納され、S, Z, AC, P/V, CYの各フラグを変化させます。

加算の結果、dstのビット23がセット (1) された場合は、Sフラグはセット (1)、その他の場合はクリア (0) されます。

加算の結果、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。

加算の結果、ACフラグは不定となります。

加算の結果、ビット22からビット23へのキャリーが発生し、かつ、ビット23からのキャリーが発生しない場合 (2の補数形式での演算によるオーバフロー発生時)、または、ビット22からビット23へのキャリーが発生せず、かつ、ビット23からのキャリーが発生した場合 (2の補数形式での演算によるアンダフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (0) されます。

加算の結果、ビット23からのキャリーが発生した場合、CYフラグはセット (1)、その他の場合はクリア (0) されます。

【記述例】

**ADDG TDE, VVP** ; TDEレジスタとVVPレジスタを加算し、結果をTDEレジスタへ格納

**SUBG**Subtract G<sup>注</sup>  
24ビット・データの減算

【命令形式】 SUBG dst, src

注 Gは24ビット・データを操作  
することを表す文字です

【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)
SUBG	rg, rg'
	rg, #imm24
	WHL, saddrg

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	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dstへ格納し、S、Z、AC、P/V、CYの各フラグを変化させます。

減算の結果、dstのビット23がセット (1) された場合は、S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、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、ACフラグは不定となります。

減算の結果、ビット23からビット22へのボローが発生し、かつ、ビット23でボローが発生しない場合 (2の補数形式での演算によるアンダフロー発生時)、または、ビット23からビット22へのボローが発生せず、かつ、ビット23でボローが発生した場合 (2の補数形式での演算によるオーバフロー発生時)、P/Vフラグはセット (1) され、その他の場合はクリア (0) されます。

減算の結果、ビット23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。

【記述例】

**SUBG UUP, #543210H**; UUPレジスタの内容から543210Hを減算し、結果をUUPレジスタへ格納

## 7.9 乗除算命令

乗除算命令には、次の命令があります。

MULU ... 326  
MULUW ... 327  
MULW ... 328  
DIVUW ... 329  
DIVUX ... 330

**MULU**Multiply 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	r

【フラグ】

S	Z	AC	P/V	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レジスタへ格納します。

【記述例】

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

**MULUW**Multiply Unsigned Word  
ワード・データの符号なし乗算

【命令形式】 MULUW src

【オペレーション】 AX (上位) , src (下位)  $AX \times src$ 

【オペランド】

二モニック	オペランド (src)
MULUW	rp

【フラグ】

S	Z	AC	P/V	CY

【説明】

AX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、結果の上位16ビットをAXレジスタへ、下位16ビットをソース・オペランドへ格納します。

ソース・オペランド (src) としてAXレジスタを指定した場合、乗算結果の上位16ビットがAXレジスタに格納され、下位16ビットはどこにも格納されません。

【記述例】

**MULUW HL** ; AXレジスタの内容とHLレジスタの内容を乗算し、結果をAXレジスタとHLレジスタへ格納

**MULW**Multiply Signed Word  
ワード・データの符号付き乗算

【命令形式】 MULW src

【オペレーション】 AX (上位) , src (下位)  $AX \times src$ 

【オペランド】

二モニック	オペランド ( src )
MULW	rp

【フラグ】

S	Z	AC	P/V	CY

【説明】

AXレジスタの内容とソース・オペランド ( src ) のデータを符号付きのデータとして乗算し、結果の上位16ビットをAXレジスタへ、下位16ビットをソース・オペランドへ格納します。

ソース・オペランド ( src ) としてAXレジスタを指定した場合、乗算結果の上位16ビットがAXレジスタに格納され、下位16ビットはどこにも格納されません。

【記述例】

**MULW HL ;** AXレジスタの内容とHLレジスタの内容を乗算し、結果をAXレジスタとHL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】

二モニック	オペランド (dst)
DIVUW	r

【フラグ】

S	Z	AC	P/V	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います。

0で割ったとき (dst = 0) は、次のようになります。

- ・ AX (商) = FFFFH
- ・ dst (余り) = もとのXレジスタの値

デスティネーション・オペランド (dst) としてAレジスタを指定した場合、Aレジスタに余りが格納され、Xレジスタには商の下位8ビットが格納されます。

デスティネーション・オペランド (dst) としてXレジスタを指定した場合、Aレジスタに商の上位8ビットが格納され、Xレジスタには余りが格納されます。

【記述例】

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

**DIVUX**Divide Unsigned Word Expansion Word  
ダブル・ワード・データの符号なし除算【命令形式】 **DIVUX dst**【オペレーション】 **AXDE (商) , dst (余り) AXDE ÷ dst**

【オペランド】

二モニック	オペランド (dst)
<b>DIVUX</b>	rp

【フラグ】

S	Z	AC	P/V	CY

【説明】

AXレジスタの内容を上位16ビットとし、DEレジスタの内容を下位16ビットとして32ビット・データをデスティネーション・オペランド (dst) の内容で除算し、商の上位16ビットをAXレジスタへ、下位16ビットをDEレジスタへ、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レジスタとDEレジスタで示される32ビット・データの内容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います。

0で割ったとき (dst = 0) は、次のようになります。

- ・ AXDE (商) = FFFFFFFFH
- ・ dst (余り) = もとのDEレジスタの値

デスティネーション・オペランド (dst) として、AXレジスタを指定した場合、AXレジスタに余りが格納され、DEレジスタには商の下位16ビットが格納されます。

デスティネーション・オペランド (dst) として、DEレジスタを指定した場合、AXレジスタに商の上位16ビットが格納され、DEレジスタには余りが格納されます。

【記述例】

**DIVUX BC** ; AXDEレジスタの内容をBCレジスタの内容で除算し、商の上位16ビットをAXレジスタへ、下位16ビットをDEレジスタへ、余りをBCレジスタへ格納

## 7.10 特殊演算命令

特殊演算命令には、次の命令があります。

MACW ... 332

MACSW ... 335

SACW ... 338

**MACW**Multiply and Accumulate Word  
ワード・データ積和演算

【命令形式】 MACW byte

【オペレーション】 AXDE (B) × (C) + AXDE, B B+2, C C+2, byte byte - 1  
End if (byte = 0 or P/V = 1)

【オペランド】

二モニック	オペランド
MACW	byte

【フラグ】

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

【説明】

Bレジスタでアドレスされる2バイトのエリアの内容と、Cレジスタでアドレスされる2バイトのエリアの内容との符号付き乗算を行い、その結果とAXDEレジスタの内容を2進加算します。

加算の結果をAXDEレジスタに格納したあと、BレジスタとCレジスタの内容を+2します。

以上の動作をオペランドに記述された8ビット・イミディエト・データの回数分繰り返します。

加算の結果、オーバーフローまたはアンダフローが発生した場合、AXDEレジスタの値は不定になります。

また、Bレジスタ、Cレジスタはオーバーフローする直前の値を保持しています。

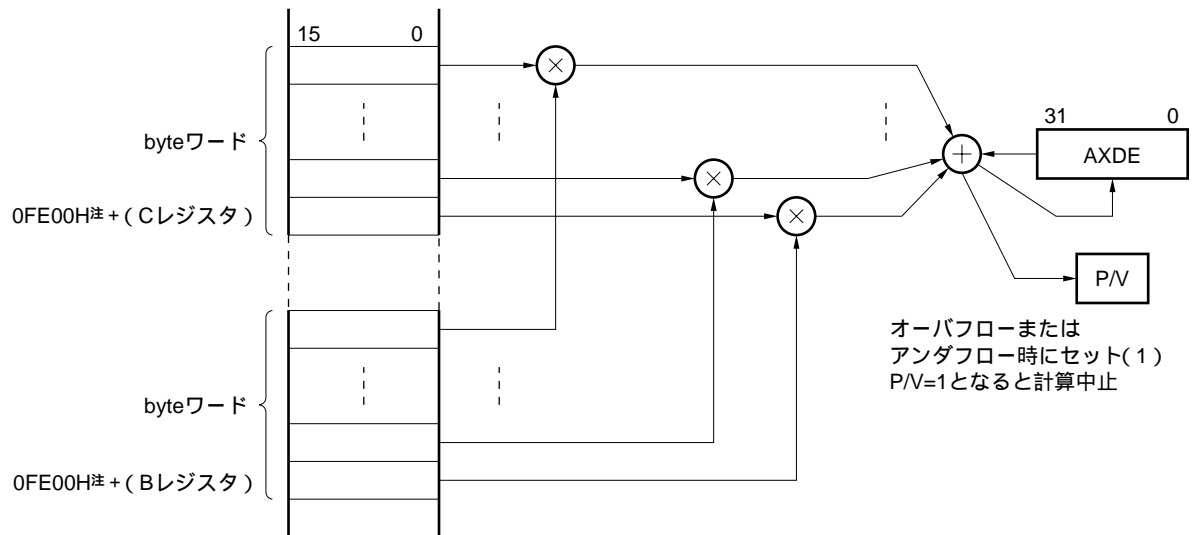
MACW命令でアドレスされる領域は、LOCATION 0H命令を実行した場合は0FE00H-0FEFFHに、LOCATION 0FH命令を実行した場合は0FFE00H-0FFEFFH番地に限定されます。BレジスタおよびCレジスタでは、アドレスの下位1バイトを指定します。また、FE80H-FEFFFH (LOCATION 0FH命令実行時は、FFE80H-FFEFFFH) 番地は汎用レジスタと兼用しています。

MACW命令実行中は、割り込み、マクロ・サービスを受け付けません。

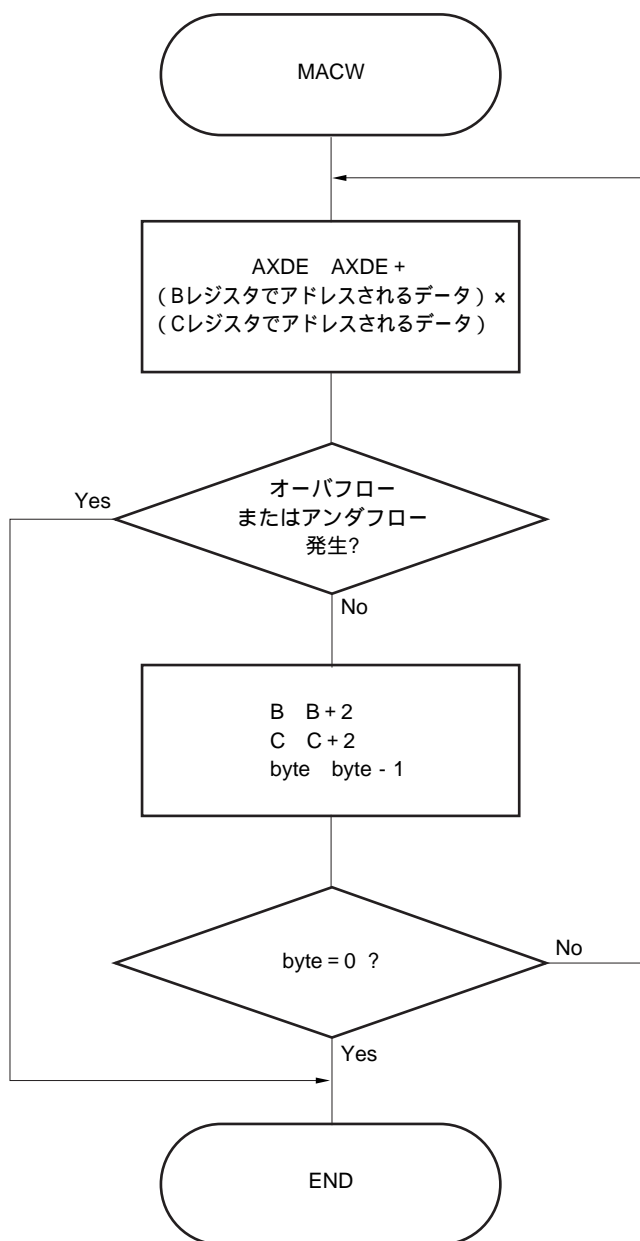
MACW命令は、AXDEレジスタ・ペアの値を自動的にクリアしませんので、必要な場合は、プログラムでクリアするようにしてください。

演算の結果、S, Z, AC, CYのフラグは不定となります。

オーバーフローまたはアンダフローが発生するとP/Vフラグはセット(1)され、その他の場合はクリア(0)されます。



**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時は0FFE00H  
 なお、 $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。



**【記 述 例】**

**MACW 5 ; 積和演算を5回実行**

**MACSW**Multiply and Accumulate with Saturation Word  
飽和機能付き積和命令

【命令形式】 MACSW byte

【オペレーション】  $AXDE (B) \times (C) + AXDE, B \ B + 2, C \ C + 2, \text{byte} \ \text{byte} - 1$  if  
 byte = 0 then End, if P/V = 1, then if overflow AXDE 7FFFFFFFH,  
 end, if underflow AXDE 80000000H, end

【オペランド】

二モニック	オペランド (\$addr16)
MACSW	byte

【フラグ】

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

【説明】

Bレジスタでアドレスされる2バイトのエリアの内容と、Cレジスタでアドレスされる2バイトのエリアの内容との符号付き乗算を行い、その結果とAXDEレジスタの内容を2進加算します。

加算の結果をAXDEレジスタに格納したあと、BレジスタとCレジスタの内容を+2します。

以上の動作をオペランドに記述された8ビット・イミディエイト・データの回数分繰り返します。

加算の結果、オーバーフローが発生した場合、P/Vフラグがセット(1)され、AXDEレジスタは7FFFFFFFHになります。アンダフローが発生した場合、P/Vフラグがセット(1)され、AXDEレジスタは80000000Hになります。Bレジスタ、Cレジスタの値は、オーバーフローまたはアンダフローする直前の値を保持しています。

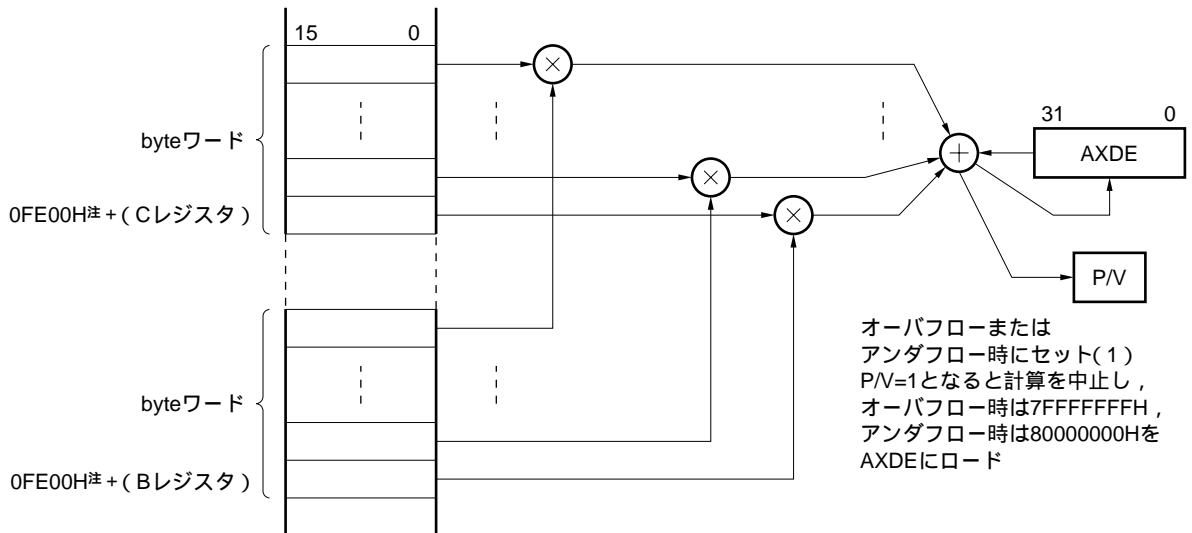
MACSW命令でアドレスされる領域は、LOCATION 0H命令を実行した場合は0FE00H-0FEFFHに、LOCATION 0FH命令を実行した場合は0FFE00H-0FFEFFH番地に限定されます。BレジスタおよびCレジスタでは、アドレスの下位1バイトを指定します。また、FE80H-FEFFFH (LOCATION 0FH命令実行時は、FFE80H-FEFFFH) 番地は汎用レジスタと兼用しています。

MACSW命令実行中は、割り込み、マクロ・サービスを受け付けません。

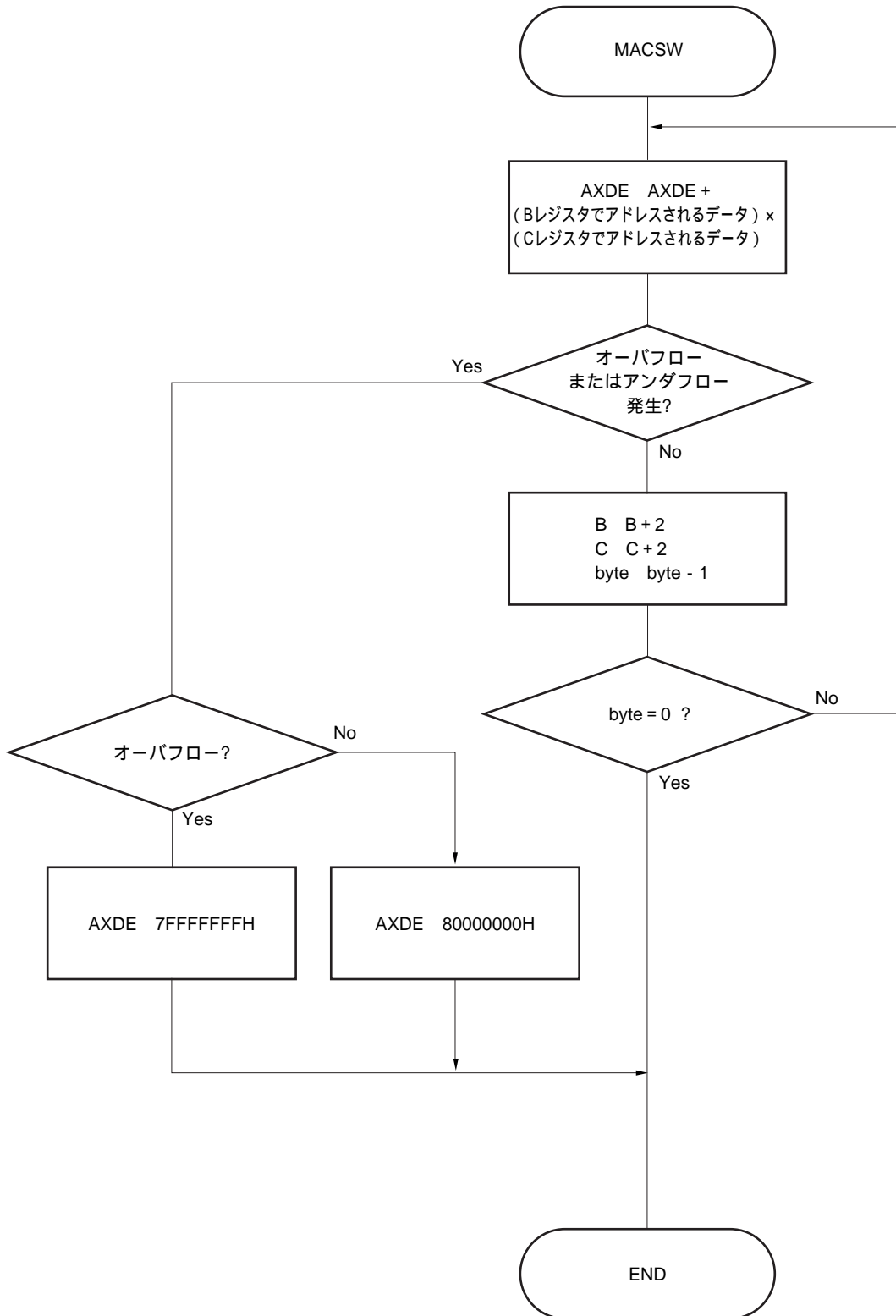
MACSW命令は、AXDEレジスタ・ペアの値を自動的にクリアしませんので、必要な場合は、プログラムでクリアするようにしてください。

演算の結果、S, Z, AC, CYのフラグは不定となります。

オーバーフローまたはアンダフローが発生するとP/Vフラグはセット(1)され、その他の場合はクリア(0)されます。



**注** LOCATION 0H命令実行時。LOCATION 0FH命令実行時は0FFE00H  
 なお、 $\mu$  PD784915サブシリーズは、LOCATION 0H命令固定です。



【記述例】

MACSW 6 ; 積和演算を6回実行

**SACW**

Subtract, Absolute and Accumulate word

関連命令

【命令形式】 SACW [ TDE + ], [ WHL + ]

【オペレーション】 AX | ( TDE ) - ( WHL ) | + AX, TDE TDE + 2, WHL WHL + 2,  
C C - 1, end if ( C = 0 or CY = 1 )

【オペランド】

二モニック	オペランド ( \$addr16 )
SACW	[ TDE + ], [ WHL + ]

【フラグ】

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

【説明】

TDEレジスタとWHLレジスタで、それぞれアドレスされる16ビット・データどうしの減算を行い、その結果の絶対値をAXレジスタの内容と加算し、その結果をAXレジスタに格納します。

上記の演算を1回行うたびに、TDE, WHLレジスタの内容を+2し、Cレジスタの内容を-1します。

上記の動作をCレジスタが0になるまで、または加算の結果、ビット15からのキャリーが発生するまで繰り返します。

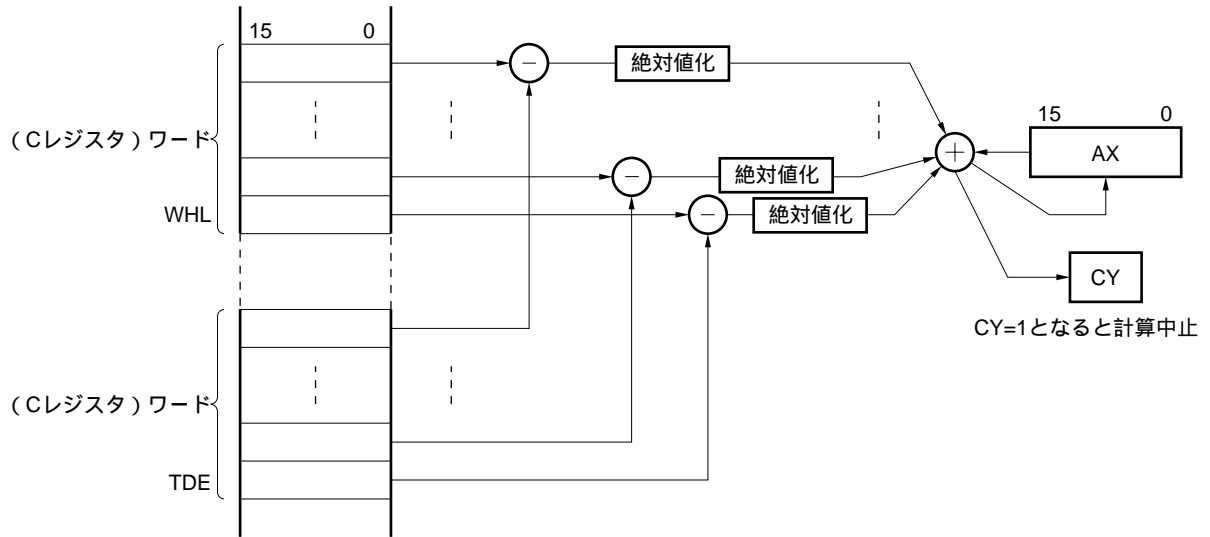
加算の結果、15ビットからのキャリーが発生して繰り返しを止めた場合は、TDE, WHLレジスタはキャリーが発生する直前の値に+2をした値を保持しています。また、Cレジスタは、キャリーが発生する直前の値を保持しています。

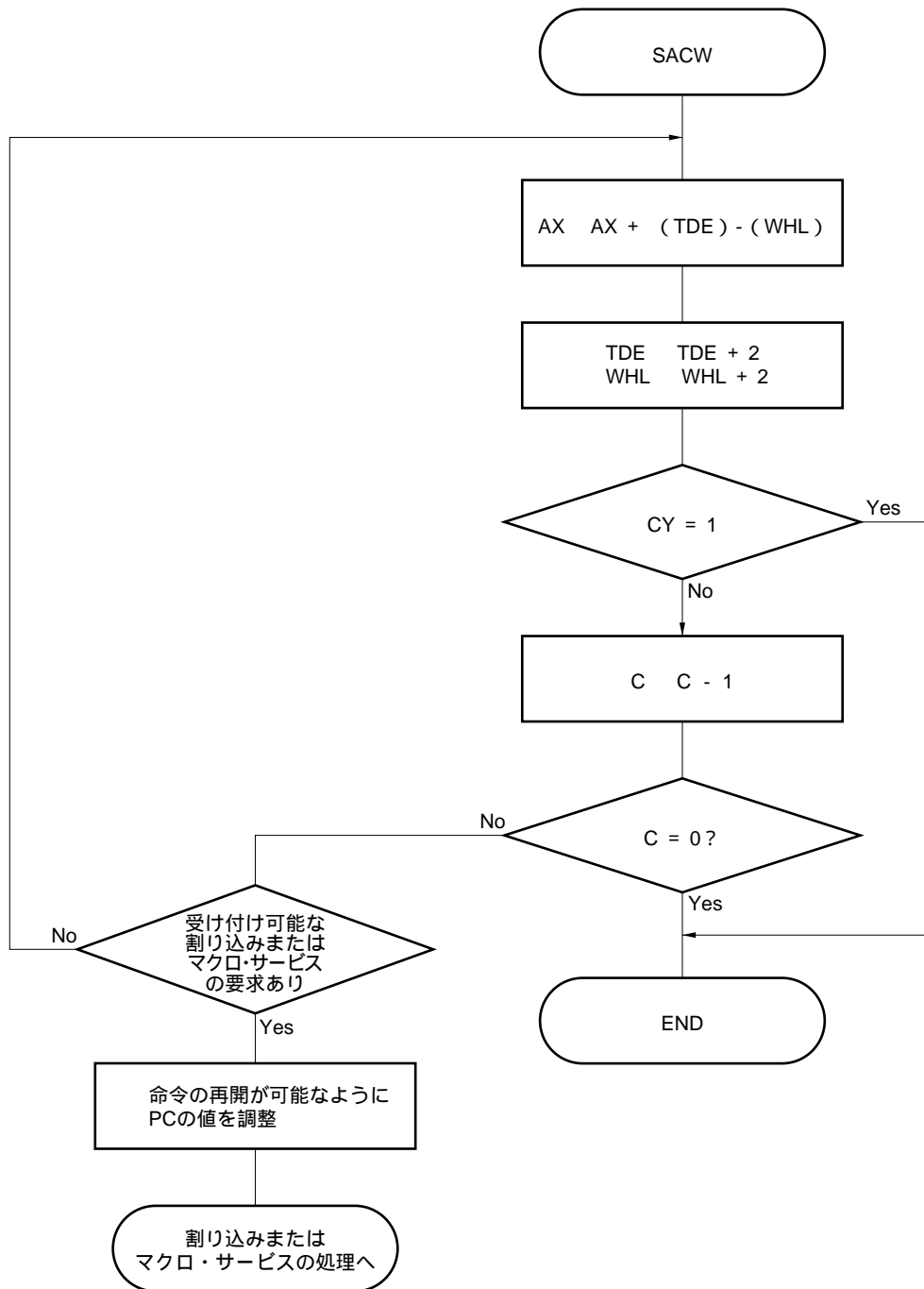
SACW命令実行中に受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、一連の演算処理の前に割り込みまたはマクロ・サービスの処理を行います。割り込みを受け付けた場合、スタックに退避されるプログラム・カウンタ ( PC ) の値は、SACW命令の先頭番地となります。したがって、割り込みからの復帰時には、中断されたSACW命令の続きを実行可能です。

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

S, Z, AC, P/Vフラグの内容は不定となります。

SACW命令は、AXレジスタの内容を自動的にクリアしませんので、必要な場合はプログラムでクリアしてください。





【記述例】

SACW [ TDE + ], [ WHL + ] ; 相関命令を実行

## 7.11 増減命令

増減命令には、次の命令があります。

INC ... 342  
DEC ... 343  
INCW ... 344  
DECW ... 345  
INCG ... 346  
DECG ... 347

**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】

二モニック	オペランド (dst)
INC	r
	saddr

【フラグ】

S	Z	AC	P/V	CY
x	x	x	V	

【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) されます。

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ト・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)。

インクリメントの結果, dstのビット7がセット (1) された場合は, Sフラグはセット (1) , その他の場合はクリア (0) されます。

インクリメントの結果, ビット6からビット7へのキャリーが発生し, かつ, ビット7からのキャリーが発生しない場合 (2の補数形式での演算によるオーバーフロー発生時) , P/Vフラグはセット (1) され, その他の場合はクリア (0) されます。

【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】

S	Z	AC	P/V	CY
x	x	x	V	

【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) されます。

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)。

デクリメントの結果、dstのビット7がセット (1) された場合は、Sフラグはセット (1) , その他の場合はクリア (0) されます。

デクリメントの結果、ビット6からビット7へのボローが発生し、かつ、ビット7でボロー発生しない場合 (2の補数形式での演算によるアンダフロー発生時) , P/Vフラグはセット (1) され、その他の場合はクリア (0) されます。

dstがBレジスタ、Cレジスタ、またはsaddrの場合でS、Z、AC、P/Vの各フラグを変化させたくない場合、DBNZ命令を使用することができます。

【記述例】

DEC SAD1 ; ショート・ダイレクト・アドレッシングでアクセス可能なSAD1番地の内容をデクリメント

**INCW**Increment 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】

二モニック	オペランド (dst)
INCW	rp
	saddrp

【フラグ】

S	Z	AC	P/V	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、使用するレジスタのインクリメントに使用することが多いため、S、Z、AC、P/V、CYの各フラグを変化させません。

オペランドとしてrpを指定した場合で78K/0、78K/1、78K/2、78K/3 シリーズ用のプログラムを流用する場合に、レジスタ・インダイレクト・アドレッシングおよびベース・アドレッシング、ベースト・インデクス・アドレッシング (78K/0、78K/1 シリーズのみ) のベース・レジスタとしてHL、DE、VP、UPの各レジスタ (VP、UPは78K/2 シリーズのみ) を使用していた場合、アドレスの上位8ビットを示すT、W、V、Uレジスタの内容が下位16ビットを示すDE、HL、VP、UPレジスタと整合がとれていることを確認してください。また、プログラムの修正が可能であれば、24ビット操作命令 (INCG命令) を使用するよう変更してください。

【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】

二モニック	オペランド (dst)
DECW	rp
	saddrp

【フラグ】

S	Z	AC	P/V	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、使用するレジスタのデクリメントに使用することが多いため、S, Z, AC, P/V, CYの各フラグを変化させません。

オペランドとしてrpを指定した場合で78K/0, 78K/ , 78K/ , 78K/ シリーズ用のプログラムを流用する場合に、レジスタ・インダイレクト・アドレッシングおよびベース・アドレッシング、ベースト・インデクス・アドレッシング (78K/0, 78K/ シリーズのみ) のベース・レジスタとしてHL, DE, VP, UPの各レジスタ (VP, UPは78K/ シリーズのみ) を使用していた場合、アドレスの上位8ビットを示すT, W, V, Uレジスタの内容が下位16ビットを示すDE, HL, VP, UPレジスタと整合がとれていることを確認してください。また、プログラムの修正が可能であれば、24ビット操作命令 (INCG命令) を使用するよう変更してください。

【記述例】

DECW DE ; DEレジスタをデクリメント

**INCG**Increment G<sup>注</sup>

24ビット・データのインクリメント

【命令形式】 INCG dst

注 Gは24ビット・データを操作  
することを表す文字です

【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)
INCG	rg

【フラグ】

S	Z	AC	P/V	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) のデクリメントに使用することが多いため、S, Z, AC, P/V, CYの各フラグを変化させません。

【記述例】

INCG UUP ; UUPレジスタをインクリメント

**DECG**Decrement G<sup>注</sup>  
24ビット・データのデクリメント

【命令形式】 DECG dst

注 Gは24ビット・データを操作  
することを表す文字です

【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)
DECG	rg
	SP

【フラグ】

S	Z	AC	P/V	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、S, Z, AC, P/V, CYの各フラグを変化させません。

【記述例】

**DECG VVP ;** VVPレジスタをデクリメント

## 7.12 補正命令

補正命令には、次の命令があります。

ADJBA ... 349  
ADJBS ... 350  
CVTBW ... 351

**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】

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

【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。

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

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）されません。

補正の結果、Aレジスタのビット7が1の場合、Sフラグはセット（1）され、その他の場合は、クリア（0）されます。

補正の結果、Aレジスタの中のセット（1）されたビット数が偶数の場合は、P/Vフラグはセット（1）され、その他の場合は、クリア（0）されます。

【記述例】

ADJBA ; Aレジスタの内容を10進補正

**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】**

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

**【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。

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 1
	CY = 1	A A - 01100110B, CY 1, AC 1

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。  
補正の結果、Aレジスタのビット7が1の場合、Sフラグはセット（1）され、その他の場合は、クリア（0）

されます。  
補正の結果、Aレジスタの中のセット（1）されたビット数が偶数の場合は、P/Vフラグはセット（1）さ

**【記述例】**

ADJBS ; Aレジスタの内容を10進補正

**CVTBW**Convert Byte to Word  
バイト・データからワード・データへの変換

【命令形式】 CVTBW

【オペレーション】  $A_7 = 0$ のとき, X A, A 00H  
 $A_7 = 1$ のとき, X A, A FFH

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY

【説明】

Aレジスタ内の符号付き8ビット・データを, AXレジスタ内の符号付き16ビット・データに拡張します。  
この命令では, S, Z, AC, P/V, CYの各フラグを変化させません。

【記述例】

**CVTBW**; Aレジスタの8ビット符号付きデータを符号付きの16ビット・データに拡張してAXレジスタに格納

## 7.13 シフト・ローテート命令

シフト・ローテート命令には、次の命令があります。

ROR	...	353
ROL	...	354
RORC	...	355
ROLC	...	356
SHR	...	357
SHL	...	358
SHRW	...	359
SHLW	...	360
ROR4	...	361
ROL4	...	362

**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) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
ROR	r, n

【フラグ】

S	Z	AC	P/V	Y
			P	x

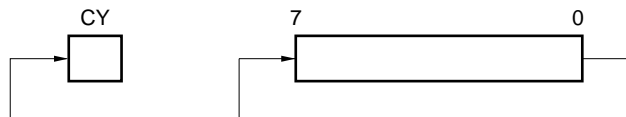
【説明】

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、第2オペランドで指定される cnt 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。

第2オペランド ( cnt ) が0の場合は、何も処理を行いません ( S, Z, AC, P/V, CYの各フラグも変化しません )。右方向への回転の結果、dstの中のセット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 ) , その他の場合はクリア ( 0 ) されます。

回転動作の結果によらず、S, Z, ACの各フラグは変化しません。



【記述例】

ROR R5, 4 ; R5レジスタの内容を右へ4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, dst<sub>0</sub> dst<sub>7</sub>, dst<sub>m+1</sub> dst<sub>m</sub>) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
ROL	r, n

【フラグ】

S	Z	AC	P/V	CY
			P	x

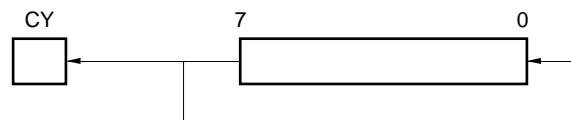
【説明】

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、第2オペランドで指定される cnt 回だけ右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。

第2オペランド ( cnt ) が0の場合は、何も処理を行いません ( S, Z, AC, P/V, CY の各フラグも変化しません )。左方向への回転の結果、dst 中のセット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 )、その他の場合はクリア ( 0 ) されます。

回転動作の結果によらず、S, Z, AC の各フラグは変化しません。



【記述例】

ROL L, 2 ; Lレジスタの内容を左へ2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) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド (dst, cnt)
RORC	r, n

【フラグ】

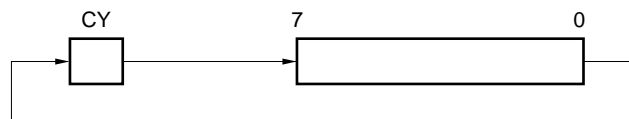
S	Z	AC	P/V	CY
			P	X

【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、第2オペランドで指定されるcnt回だけ右方向へ回転させます。

第2オペランド (cnt) が0の場合は、何も処理を行いません (S, Z, AC, P/V, CYの各フラグも変化しません)。右方向への回転の結果、dstの中のセット (1) されたビット数が偶数の場合は、P/Vフラグはセット (1)、その他の場合はクリア (0) されます。

回転動作の結果によらず、S, Z, ACの各フラグは変化しません。



【記述例】

RORC B, 1 ; B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) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド (dst, cnt)
ROLC	r, n

【フラグ】

S	Z	AC	P/V	CY
			P	x

【説明】

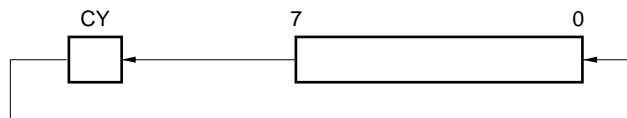
第1オペランドで指定されるデスティネーション・オペランド (dst) の内容を、CYフラグを含め、第2オペランドで指定されるcnt回だけ左方向へ回転させます。

第2オペランド (cnt) が0の場合は、何も処理を行いません (S, Z, AC, P/V, CYの各フラグも変化しません)。

1ビットだけ左へ回転させたい場合は、ADDCr, rを使用すると実行時間を短くすることができます。

左方向への回転の結果、dstの中のセット (1) されたビット数が偶数の場合は、P/Vフラグはセット (1)、その他の場合はクリア (0) されます。

回転動作の結果によらず、S, Z, ACの各フラグは変化しません。



【記述例】

ROLC R7, 3 ; R7レジスタの内容を、CYフラグを含めて3ビット左へ回転

**SHR**Shift Right ( Logical )  
バイト・データの右方向の論理シフト

【命令形式】 SHR dst, cnt

【オペレーション】 (CY dst0, dst7 0, dstm-1 dstm) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
SHR	r, n

【フラグ】

S	Z	AC	P/V	CY
x	x	0	P	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、第2オペランドで指定されるcnt回だけ右方向へシフトします。

1ビット、シフトするごとにMSB ( ビット7 ) には0がシフト・インされます。

cntが1以上のとき、Sフラグはクリア ( 0 ) されます。

シフト動作の結果、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 ) されます。

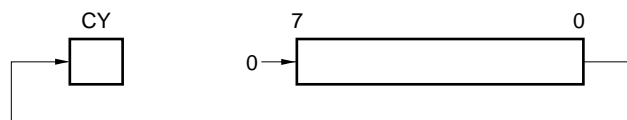
シフト動作の結果によらず、ACフラグは必ず0になります。

シフト動作の結果、dstの中のセット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 ) 、その他の場合はクリア ( 0 ) されます。

シフト動作の結果、LSBB ( ビット0 ) からシフト・アウトされた最後のデータがCYフラグにセットされます。

cntが0の場合は、何も処理を行いません ( S, Z, AC, P/V, CYの各フラグも変化しません ) 。

この命令は、デスティネーション・オペランド ( dst ) を符号なしのデータとして $2^{cnt}$ で除算したのと同じ結果になります。



【記述例】

SHR H, 2 ; Hレジスタの内容を2ビット右へシフト

**SHL**

Shift Left ( Logical )

バイト・データの左方向の論理シフト

【命令形式】 SHL dst, cnt

【オペレーション】 (CY dst7, dst0 0, dstm+1 dstm) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
SHL	r, n

【フラグ】

S	Z	AC	P/V	CY
x	x	0	P	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、第2オペランドで指定されるcnt回だけ左方向へシフトします。

1ビット、シフトするごとにLSB ( ビット0 ) には0がシフト・インされます。

シフト動作の結果、dstのビット7が1の場合、Sフラグはセット ( 1 ) 、0の場合はクリア ( 0 ) されます。

シフト動作の結果、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 ) されます。

シフト動作の結果によらず、ACフラグは必ず0になります。

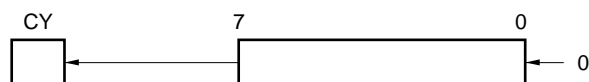
シフト動作の結果、dstの中のセット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 ) 、その他の場合はクリア ( 0 ) されます。

シフト動作の結果、LSB ( ビット0 ) からシフト・アウトされた最後のデータがCYフラグにセットされます。

cntが0の場合は、何も処理を行いません ( S, Z, AC, P/V, CYの各フラグも変化しません ) 。

1ビットだけ左方向へシフトしたい場合は、ADDR, r命令を使用すると実行時間を短くできます。

この命令は、デスティネーション・オペランド ( dst ) を $2^{cnt}$ で乗算したのと同じ結果になります ( 乗算の結果が8ビット以下の場合 ) 。



【記述例】

SHL E, 1 ; Eレジスタの内容を1ビット左へシフト

**SHRW**Shift Right ( Logical ) Word  
ワード・データの右方向の論理シフト

【命令形式】 SHRW dst, cnt

【オペレーション】 (CY dst<sub>0</sub>, dst<sub>15</sub> 0, dst<sub>m-1</sub> dst<sub>m</sub>) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
SHRW	rp, n

【フラグ】

S	Z	AC	P/V	CY
x	x	0	P	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 ) を、第2オペランドで指定される cnt 回だけ右方向へシフトします。

1ビット、シフトするごとにMSB ( ビット15 ) には0がシフト・インされます。

cntが1以上のとき、Sフラグはクリア ( 0 ) されます。

シフト動作の結果、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 ) されます。

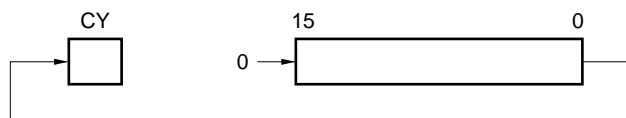
シフト動作の結果によらず、ACフラグは必ず0になります。

シフト動作の結果、dstの中の下位8ビットについて、セット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 ) 、その他の場合はクリア ( 0 ) されます。

シフト動作の結果、LSB ( ビット0 ) からシフト・アウトされた最後のデータがCYフラグにセットされます。

cntが0の場合は、何も処理を行いません ( S, Z, AC, P/V, CYの各フラグも変化しません ) 。

この命令は、デスティネーション・オペランド ( dst ) を符号なしのデータとして $2^{cnt}$ で除算したのと同じ結果になります。



【記述例】

SHRW AX, 3 ; AXレジスタの内容を3ビット右へシフト ( AXレジスタの内容を1/8にする )

**SHLW**Shift Left ( Logical ) Word  
ワード・データの左方向の論理シフト

【命令形式】 SHLW dst, cnt

【オペレーション】 (CY dst15, dst0 0, dstm+1 dstm) × cnt回 cnt = 0 - 7

【オペランド】

二モニック	オペランド ( dst, cnt )
SHLW	rp, n

【フラグ】

S	Z	AC	P/V	CY
x	x	0	P	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、第2オペランドで指定されるcnt回だけ左方向へシフトします。

1ビット、シフトするごとにLSB ( ビット0 ) には0がシフト・インされます。

シフト動作の結果、dstのビット15が1の場合、Sフラグはセット ( 1 ) , 0の場合はクリア ( 0 ) されます。

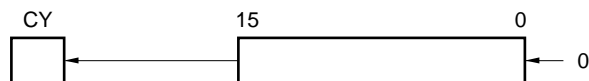
シフト動作の結果、0になればZフラグはセット ( 1 ) , その他の場合はクリア ( 0 ) されます。

シフト動作の結果によらず、ACフラグは必ず0になります。

シフト動作の結果、dstの中の下位8ビットについて、セット ( 1 ) されたビット数が偶数の場合は、P/Vフラグはセット ( 1 ) , その他の場合はクリア ( 0 ) されます。

シフト動作の結果、LSB ( ビット0 ) からシフト・アウトされた最後のデータがCYフラグにセットされます。

cntが0の場合は、何も処理を行いません ( S, Z, AC, P/V, CYの各フラグも変化しません ) 。



【記述例】

SHLW E, 1 ; Eレジスタの内容を1ビットだけ左へシフト

# 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】  $A_{3-0}$  ( $dst$ )<sub>3-0</sub>, ( $dst$ )<sub>7-4</sub>  $A_{3-0}$ , ( $dst$ )<sub>3-0</sub>  $dst_{7-4}$

【オペランド】

二モニック	オペランド (dst)
ROR4	mem3

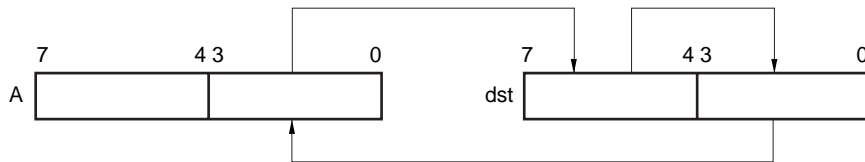
【フラグ】

S	Z	AC	P/V	CY

【説明】

Aレジスタの低位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 [WHL] ; AレジスタとWHLレジスタで指定されるメモリの内容を右方向へディジット・ローテートする

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

# ROL4

Rotate Left Dight  
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】  $A_{3-0} (dst)_{7-4}, (dst)_{3-0} A_{3-0}, (dst)_{7-4} dst_{3-0}$

【オペランド】

二モニック	オペランド (dst)
ROL4	mem3

【フラグ】

S	Z	AC	P/V	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 [TDE] ; AレジスタとTDEレジスタで指定されるメモリの内容を左方向へディジット・ローテートする

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

## 7.14 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 ... 364

AND1 ... 366

OR1 ... 368

XOR1 ... 370

NOT1 ... 371

SET1 ... 372

CLR1 ... 373

**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】

二モニック	オペランド (dst, src)
MOV1	CY, saddr.bit
	CY, sfr.bit
	CY, X.bit
	CY, A.bit
	CY, PSWL.bit
	CY, PSWH.bit
	CY, mem2.bit
	CY, !addr16.bit
	CY, !!addr24.bit
	saddr.bit, CY
	sfr.bit, CY
	X.bit, CY
	A.bit, CY
	PSWL.bit, CY
	PSWH.bit, CY
	mem2.bit, CY
	!addr16.bit, CY
	!!addr24.bit, CY

【フラグ】

dstがPSWL.bit

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

dstがCY

S	Z	AC	P/V	CY
				x

上記以外

S	Z	AC	P/V	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    AND1 dst, /src**【オペレーション】 **dst dst src    dst dst  $\overline{\text{src}}$** 

## 【オペランド】

二モニック	オペランド (dst, src)
AND1	CY, saddr.bit
	CY, /saddr.bit
	CY, sfr.bit
	CY, /sfr.bit
	CY, X.bit
	CY, /X.bit
	CY, A.bit
	CY, /A.bit
	CY, PSWL.bit
	CY, /PSWL.bit
	CY, PSWH.bit
	CY, /PSWH.bit
	CY, mem2.bit
	CY, /mem2.bit
	CY, !addr16.bit
	CY, /!addr16.bit
	CY, !!addr24.bit
	CY, /!!addr24.bit

## 【フラグ】

S	Z	AC	P/V	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。

第2オペランドの直前に “ / ” がある場合は、ソース・オペランド (src) の論理否定を行ったデータと論理積をとります。

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, SADR.3** ; ショート・ダイレクト・アドレッシング可能なSADR番地のビット3とCYフラグの論理積をとり、結果をCYフラグに格納

**AND1 CY, /PSW.6** ; PSWのビット6 (Zフラグ) の論理否定をとった結果と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    OR1 dst, /src

【オペレーション】 dst dst src    dst dst  $\overline{\text{src}}$ 

## 【オペランド】

二モニック	オペランド (dst, src)
OR1	CY, saddr.bit
	CY, /saddr.bit
	CY, sfr.bit
	CY, /sfr.bit
	CY, X.bit
	CY, /X.bit
	CY, A.bit
	CY, /A.bit
	CY, PSWL.bit
	CY, /PSWL.bit
	CY, PSWH.bit
	CY, /PSWH.bit
	CY, mem2.bit
	CY, /mem2.bit
	CY, !addr16.bit
	CY, /!addr16.bit
	CY, !!addr24.bit
	CY, /!!addr24.bit

## 【フラグ】

S	Z	AC	P/V	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。

第2オペランドの直前に “ / ” がある場合は、ソース・オペランド (src) の論理否定を行ったデータと論理和をとります。

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

**OR1 CY, /X.0** ; Xレジスタのビット0の論理否定とCYフラグの論理和の結果を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】

二モニック	オペランド (dst, src)
XOR1	CY, saddr.bit
	CY, sfr.bit
	CY, X.bit
	CY, A.bit
	CY, PSWL.bit
	CY, PSWH.bit
	CY, mem2.bit
	CY, !addr16.bit
	CY, !!addr24.bit

【フラグ】

S	Z	AC	P/V	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

# 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	saddr.bit
	sfr.bit
	X.bit
	A.bit
	PSWL.bit
	PSWH.bit
	mem2.bit
	!addr16.bit
	!!addr24.bit
	CY

【フラグ】

dstがPSWL.bit

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

dstがCY

S	Z	AC	P/V	CY
				x

上記以外

S	Z	AC	P/V	CY

【説明】

デスティネーション・オペランド ( dst ) で指定されるビットの論理否定をとり、結果をデスティネーション・オペランド ( dst ) へ格納します。

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】

NOT1 A.2 ; Aレジスタのビット2を反転

# 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
	X.bit
	A.bit
	PSWL.bit
	PSWH.bit
	mem2.bit
	!addr16.bit
	!!addr24.bit
	CY

【フラグ】

dstがPSWL.bit

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

dstがCY

S	Z	AC	P/V	CY
				1

上記以外

S	Z	AC	P/V	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 BITSYM ; ショート・ダイレクト・アドレッシング可能な領域に配置されているビットの内容をセット ( 1 )

**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】

二モニック	オペランド ( dst )
CLR1	saddr.bit
	sfr.bit
	X.bit
	A.bit
	PSWL.bit
	PSWH.bit
	mem2.bit
	!addr16.bit
	!!addr24.bit
	CY

【フラグ】

dstがPSWL.bit

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

dstがCY

S	Z	AC	P/V	CY
				0

上記以外

S	Z	AC	P/V	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 )

## 7.15 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 ... 375

PUSHU ... 377

POP ... 378

POPU ... 380

MOVG ... 381

ADDWG ... 382

SUBWG ... 383

INCG SP ... 384

DECG SP ... 385

**PUSH**Push  
プッシュ

【命令形式】 PUSH src

【オペレーション】<sup>注</sup> (1) srcがPSW, sfrpの場合(SP - 2) src ,  
SP SP - 2

(2) srcがsfrの場合

(SP - 1) src  
SP SP - 1

(3) srcがrgの場合

(SP - 3) src ,  
SP SP - 3

(4) srcがpostの場合

{ (SP - 2) post, SP SP - 2 } × n回

注 詳細については、第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

【オペランド】

二モニック	オペランド (src)
PUSH	PSW
	sfrp
	sfr
	post
	rg

【フラグ】

S	Z	AC	P/V	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。

ソース・オペランドとしてpostが指定された場合には、次のレジスタの任意の組み合わせを命令でスタックへ退避できます。

AX (RP0), BC (RP1), RP2, RP3, UP, VP, DE, HL

また、このときの退避の順序は、上記の右側にあるレジスタから行われます。

VP, UP, DE, HLの各レジスタについては、78K/0, 78K/ , 78K/ , 78K/ のプログラムを流用するときだけ使用してください。その他の場合は、UUP, VVP, TDE, WHLレジスタとして個別にスタックへの退避を指定してください。また、78K/0, 78K/ , 78K/ , 78K/ 用のプログラムを流用する場合も、できる限り、UUP, VVP, TDE, WHLレジスタとして個別にスタックへの退避を指定してください。

ソース・オペランド (src) の退避後、スタック・ポインタ (SP) は退避したデータのバイト数だけデクリメントされます。

**【記述例】**

**PUSH AX, BC, RP2, RP3 ; AX, BC, RP2, RP3の各レジスタの内容をスタックへ退避**

**PUSHU**

Push to User Stack  
レジスタのユーザ・スタックへのプッシュ

【命令形式】 **PUSHU src**

【オペレーション】<sup>注</sup> { (UUP - 1) post, UUP UUP - 2 } × n回  
(nは、postとして記述したレジスタ・ペアの数)

注 詳細については、第3章 レジスタの図3-4 スタック領域へ退避されるデータおよび図3-5 スタック領域から復帰されるデータを参照してください。

【オペランド】

二モニック	オペランド (src)
PUSHU	post

【フラグ】

S	Z	AC	P/V	CY

【説明】

ソース・オペランド (src) で指定される16ビット・レジスタ・ペアの内容をユーザ・スタック・ポイント (UUP) でアドレスされるメモリに退避し、その後、UUPをデクリメントします。

ソース・オペランド (src) として、postには次のレジスタの任意の組み合わせを記述可能です。

AX (RP0), BC (RP1), RP2, RP3, VP, PSW, DE, HL

また、このときの退避の順序は、上記の右側にあるレジスタから行われます。

【記述例】

**PUSHU BC, PSW** ; BCレジスタとPSWの内容をスタックに退避

# POP

Pop  
ポップ

【命令形式】 POP dst

【オペレーション】<sup>注</sup> (1) dstがPSW, sfrpの場合

dst (SP)  
SP SP + 2

(2) dstがsfrの場合

dst (SP),  
SP SP + 1

(3) (dst) がrgの場合

dst (SP),  
SP SP + 3

(4) dstがpostの場合

{ post (SP), SP SP + 2 } × n回

注 詳細については、第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

【オペランド】

二モニック	オペランド (dst)
POP	PSW
	sfrp
	sfr
	post
	rg

【フラグ】

dstがPSW

S	Z	AC	P/V	CY
R	R	R	R	R

その他

S	Z	AC	P/V	CY

**【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。  
デスティネーション・オペランド (dst) 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。  
デスティネーション・オペランド (dst) としてpostが指定された場合には、次のレジスタの任意の組み合わせへ1命令でスタックから復帰できます。

AX (RP0), BC (RP1), RP2, RP3, VP (RP4), UP (RP5), DE (RP6), HL (RP7)

また、このときの復帰の順序は、上記の左側にあるレジスタから行われます。

UP, VP, DE, HLの各レジスタについては、78K/0, 78K/ , 78K/ , 78K/ 用のプログラムを流用するときだけに使用してください。その他の場合は、UUP, VVP, TDE, WHLレジスタとして個別にスタックからの復帰を指定してください。また、78K/0, 78K/ , 78K/ , 78K/ 用のプログラムを流用する場合も、できるかぎりUUP, VVP, TDE, WHLレジスタとして個別にスタックへの退避を指定してください。

スタックからのデータの復帰後、スタック・ポインタ (SP) は復帰したデータのバイト数だけインクリメントされます。

**【記述例】**

**POP IMK0L ; IMK0Lレジスタへスタックのデータを復帰**

**POPU**Pop from User Stack  
レジスタのユーザ・スタックからのポップ**【命令形式】** POPU dst**【オペレーション】**注 {post (UUP), UUP UUP + 2} × n回  
(nは, postとして記述したレジスタ・ペアの数)

注 詳細については, 第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

**【オペランド】**

二モニック	オペランド (dst)
POPU	post

**【フラグ】**

S	Z	AC	P/V	CY

**【説明】**

ユーザ・スタック・ポインタ (UUP) でアドレスされるメモリ (スタック) の内容を, デスティネーション・オペランド (dst) により指定されるレジスタ・ペアへ復帰し, その後, UUPをインクリメントします。

デスティネーション・オペランド (dst) として, postには次のレジスタの任意の組み合わせを記述可能です。

AX (RP0), BC (RP1), RP2, RP3, VP (RP4), PSW, DE (RP6), HL (RP7)

また, このときの復帰の順序は, 上記の左側にあるレジスタから行われます。

**【記述例】**

**POPU AX, BC ; AX, BCレジスタへスタックのデータを復帰**



**ADDWG**

Add Word to G<sup>注</sup>  
24ビット・ワード・データの加算

【命令形式】 ADDWG dst, src

注 Gは、24ビット・データを操作することを表す文字です

【オペレーション】 SP SP + word

【オペランド】

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

【フラグ】

S	Z	AC	P/V	CY

【説明】

スタック・ポインタ (SP) の内容に符号なしの16ビット・イミディエト・データを加算し、その結果をスタック・ポインタ (SP) に格納します。

この命令は、高級言語などで一時的な変数の記憶場所として確保したメモリ領域を解放するときなどに使用します。

【記述例】

ADDWG SP, #30H ; SPに30Hを加算し、その結果をSPへ格納

**SUBWG**

Subtract Word from G<sup>注</sup>  
24ビット・ワード・データの減算

【命令形式】 SUBWG dst, src

注 Gは、24ビット・データを操作することを表す文字です

【オペレーション】 SUBWG SP SP - 1

【オペランド】

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

【フラグ】

S	Z	AC	P/V	CY

【説明】

スタック・ポインタ (SP) の値から符号なしの16ビット・イミディエト・データを減算し、その結果をSPに格納します。

この命令は、高級言語などで一時的な変数の領域を確保するために使用します。

【記述例】

SUBWG SP, #50H ; SPから50Hを減算し、その結果をSPへ格納。これにより、一時的な変数の領域を50Hバイト確保

**INCG SP**Increment G<sup>注</sup>

スタック・ポインタの24ビット・データのインクリメント

【命令形式】 INCG SP

注 Gは、24ビット・データを操作することを表す文字です

【オペレーション】 SP SP+1

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY

【説明】

SP (スタック・ポインタ) の内容を +1する命令です。

【記述例】

INCG SP

**DECG SP**Decrement G<sup>注</sup>

スタック・ポインタの24ビット・データのデクリメント

【命令形式】 DECG SP

注 Gは、24ビット・データを操作することを表す文字です

【オペレーション】 SP SP - 1

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY

【説明】

SP (スタック・ポインタ) の内容を - 1する命令です。

【記述例】

DECG SP

## 7.16 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 ... 387  
CALLF ... 389  
CALLT ... 390  
BRK ... 391  
BRKCS ... 392  
RET ... 394  
RETI ... 395  
RETB ... 396  
RETCS ... 397  
RETCSB ... 399

**CALL**Call  
サブルーチン・コール

【命令形式】 CALL target

【オペレーション】<sup>注</sup> (SP - 3) (PC + n),

SP SP - 3

PC target

n : 命令のバイト数

注 詳細については、第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

【オペランド】

二モニック	オペランド (target)
CALL	!addr16
	!!addr20
	rp
	rg
	[ rp ]
	[ rg ]
	!\$!addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

16ビット、20ビットの絶対アドレス、16ビット相対アドレス、またはレジスタ直接アドレス、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 + n) 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。

オペランドとして!addr16, rp, [ rp ] を指定した場合、分岐先のアドレスはベース領域 (0-FFFFH) に限定されてしまいます ([ rp ] の場合は、分岐先テーブルもベース領域に限定されます)。実行時間の短縮やオブジェクト・サイズの圧縮などがどうしても必要な場合と、78K/0, 78K/1, 78K/2, 78K/3 シリーズ用のプログラムを流用する場合で、プログラムの修正が困難な場合にのみ使用してください。これらの命令を使用したプログラムは、そのプログラムを流用しようとしたときに修正が必要になる可能性があります。

NEC製のアセンブラ (RA78K4) では、CALL addrと記述することにより、CALL !addr16, CALL !!addr20, CALL !\$!addr20の中から最適と推定できるオブジェクト・コードを自動的に選択して生成することができます。

**【記 述 例】**

**CALL !!13059H ; 013059Hへサブルーチン・コール**

**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】<sup>注</sup> (SP - 3) (PC + 2),  
 SP SP - 3,  
 PC target

注 詳細については、第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

【オペランド】

二モニック	オペランド (target)
CALLF	!addr11

【フラグ】

S	Z	AC	P/V	CY

【説明】

00800H-00FFFH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、00800H-00FFFH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00800H-00FFFH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。

【記述例】

CALLF !0C2AH ; 00C2AH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】**注 (SP - 3) (PC + 1),

SP SP - 3,

PC<sub>HW</sub> 0PC<sub>H</sub> (addr5 + 1)PC<sub>L</sub> (addr5)

注 詳細については、第3章 レジスタの図3 - 4 スタック領域へ退避されるデータおよび図3 - 5 スタック領域から復帰されるデータを参照してください。

**【オペランド】**

二モニック	オペランド ( [ addr5 ] )
CALLT	[ addr5 ]

**【フラグ】**

S	Z	AC	P/V	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 (アドレスの上位ビットは 000000001B に固定で、次の5ビットを addr5 で指定し、最下位ビットは 0 固定です) のワード・データで示されるアドレスへ分岐します。

この命令で分岐可能なサブルーチンの先頭アドレスは、ベース領域 (0-FFFFH) に限定されます。

**【記述例】**

CALLT [ 60H ] ; 00060H, 00061H 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 - 2) PSW,  
(SP - 4) PC + 1,  
IE 0,  
SP SP - 4,  
PC<sub>HW</sub> 0,  
PC<sub>LW</sub> (003EH)

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	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(0003EH)のワード・データで指示されるアドレスへ分岐します(分岐先のアドレスは、ベース領域(0-FFFFH)に限定されます)。

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。

【記述例】

BRK

**BRKCS**Break Context Switch  
ソフトウェア・コンテキスト・スイッチ

【命令形式】 BRKCS RBn

【オペレーション】 PC<sub>LW</sub> RP2,  
 RP3 PSW, PC<sub>15-19</sub>  
 PC<sub>15-19</sub> 0  
 RBS2 - 0 n,  
 RSS 0,  
 IE 0 (n = 0 - 7)

【オペランド】

モニタ	オペランド
BRKCS	RBn

【フラグ】

S	Z	AC	P/V	CY

【説明】

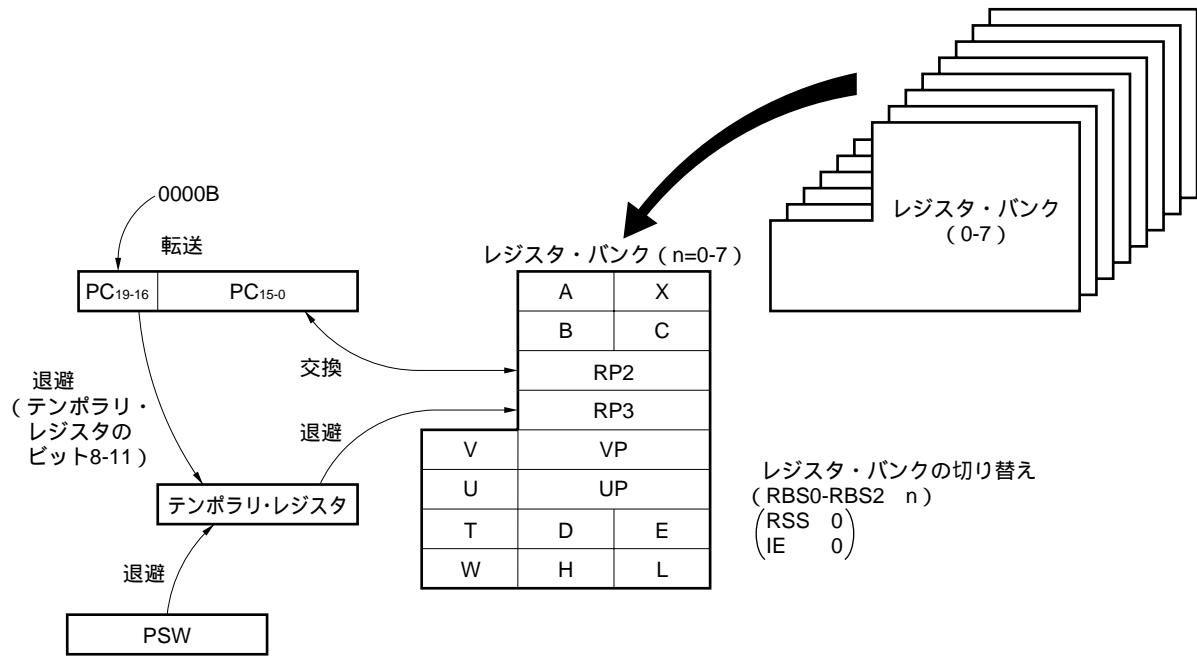
ソフトウェア割り込み命令です。

オペランドに記述したレジスタ・バンクnを選択し、そのレジスタ・バンク内のRP2の内容とプログラム・カウンタ(PC)の下位16ビットの内容とを交換し、プログラム・ステータス・ワード(PSW)とPCの上位4ビットの内容をRP3へ退避し、PCの上位4ビットを0にして、そのアドレスに分岐します。その後、RSSフラグ、IEフラグを0にクリアします。

この命令で分岐可能なアドレスは、ベース領域(0-FFFFH)に限定されます。

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

この命令で起動されるソフトウェア割り込みプログラム中では、RP2、RP3の内容を変化させないでください。RP2、RP3を使用する場合には、スタックなどに退避しておき、RETCSB命令の実行前までに元の値に戻す必要があります。



【記述例】

BRKCS RB3 ; レジスタ , バンク3を選択し , レジスタ・バンク3のRP2で示されるアドレスから命令を実行

**RET**Return  
サブルーチンからの復帰

【命令形式】 RET

【オペレーション】<sup>注</sup> PC (SP),  
SP SP+3

注 詳細については、第3章 レジスタの図3-4 スタック領域へ退避されるデータおよび図3-5 スタック領域から復帰されるデータを参照してください。

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	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】<sup>注</sup> PC (SP) ,  
PSW (SP+2) ,  
SP SP+4  
ISPR内でセット(1)されている優先順位が一番高いビットをクリア(0)

注 詳細については、第3章 レジスタの図3-4 スタック領域へ退避されるデータおよび図3-5 スタック領域から復帰されるデータを参照してください。

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY
R	R	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、ISPRレジスタ中のセット(1)されているフラグのうち、優先順位が一番高いフラグをクリア(0)し、割り込み処理ルーチンからリターンします。

BRK命令、BRKCS命令、オペランド・エラーによるソフトウェア割り込みからの復帰、およびコンテキスト・スイッチングを使用した割り込みからの復帰には使用できません。

**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】<sup>注</sup> PC (SP) ,  
PSW (SP+2) ,  
SP SP+4

注 詳細については、第3章 レジスタの図3-4 スタック領域へ退避されるデータおよび図3-5 スタック領域から復帰されるデータを参照してください。

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY
R	R	R	R	R

【説明】

BRK命令、オペランド・エラー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。  
BRKCS命令およびハードウェア割り込みからの復帰には使用できません。

**RETCS**Return from Context Switch  
ハードウェア・コンテキスト・スイッチからの復帰

【命令形式】 RETCS traget

【オペレーション】 PC<sub>LW</sub> RP2,  
PC<sub>15-19</sub> RP2<sub>8-11</sub>  
RP2 addr16,  
PSW RP3  
ISPR内でセット(1)されている優先順位が一番高いビットをクリア(0)

【オペランド】

二モニック	オペランド
RETCS	!addr16

【フラグ】

S	Z	AC	P/V	CY
R	R	R	R	R

【説明】

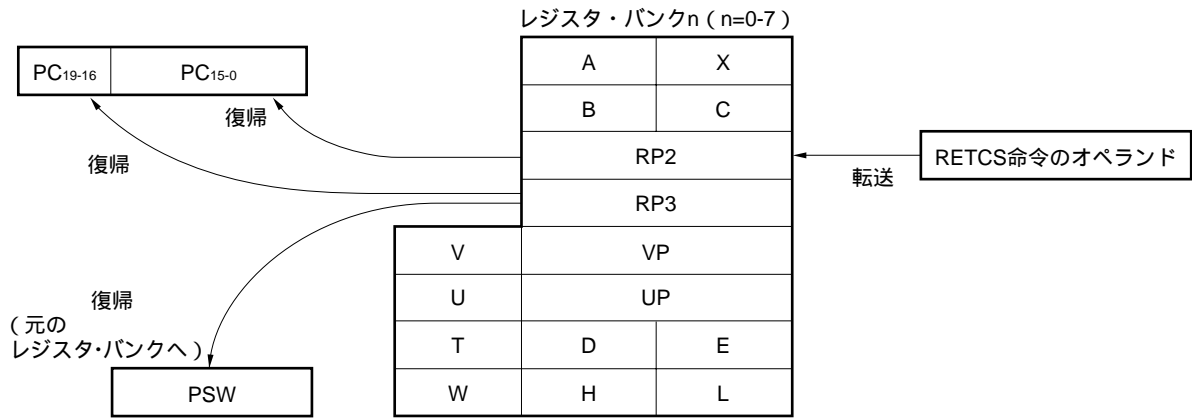
この命令実行時に指定されているレジスタ・バンク内のRP2, RP3の内容を, プログラム・カウンタ(PC)とプログラム・ステータス・ワード(PSW)に転送し, ISPRレジスタのセット(1)されているフラグのうち, 優先順位が一番高いフラグをクリア(0)し, 割り込み処理ルーチンから復帰します。

その後, オペランドで指定されるデータがRP2に転送されます。

RETCS命令は, 割り込み要求発生に伴うコンテキスト・スイッチングに対して有効で, コンテキスト・スイッチングによる分岐処理からの復帰時に使用します。オペランドに記述したaddr16は, 再びコンテキスト・スイッチング機能により, 同じレジスタ・バンクが指定された場合の分岐先アドレスとなります(指定できる分岐先アドレスは, ベース領域に限られます)。

この命令は, BRK, BRKCS命令, オペランド・エラーによるソフトウェア割り込みからの復帰, およびベクタ割り込みからの復帰には使用できません。

この命令の実行前に, RP2, RP3の内容が割り込み受け付け直後と同じ値になっている必要があります。



【記述例】

**RETCS !03456H** ; コンテキスト・スイッチングによる割り込みから復帰し、次の割り込み受け付け時のアドレスを03456Hにする

**RETCSB**Return from Context Switch Break  
ソフトウェア・コンテキスト・スイッチからの復帰

【命令形式】 RETCSB target

【オペレーション】 PC<sub>LW</sub> RP2,  
PC<sub>15-19</sub> RP3<sub>8-11</sub>  
RP2 addr16,  
PSW RP3

【オペランド】

モニタック	オペランド
RETCSB	!addr16

【フラグ】

S	Z	AC	P/V	CY
R	R	R	R	R

【説明】

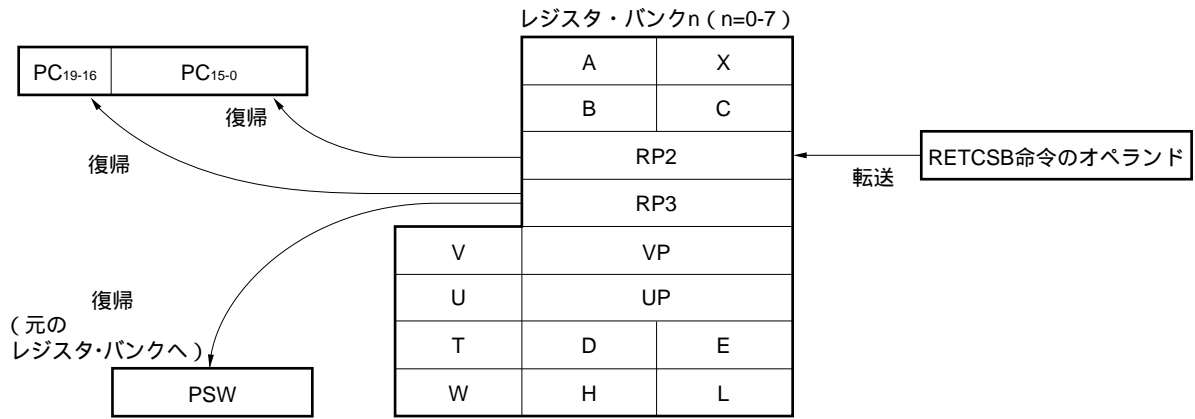
この命令実行時に指定されているレジスタ・バンク内のRP2, RP3の内容を、プログラム・カウンタ (PC) とプログラム・ステータス・ワード (PSW) に転送し、割り込み処理ルーチンから復帰します。

その後、オペランドで指定されるデータがRP2に転送されます。

RETCS命令は、BRKCS命令によるコンテキスト・スイッチングに対して有効で、コンテキスト・スイッチングによる分岐処理からの復帰時に使用します。オペランドに記述したaddr16は、再びコンテキスト・スイッチング機能により、同じレジスタ・バンクが指定された場合の分岐先アドレスとなります (指定できる分岐先アドレスは、ベース領域に限られます)。

この命令は、BRK命令、オペランド・エラーによるソフトウェア割り込みからの復帰、およびハードウェア割り込みからの復帰には使用できません。

この命令の実行前に、RP2, RP3の内容が割り込み受け付け直後と同じ値になっている必要があります。



【記述例】

RETCSB !0ABCDH ; BRKCS命令による割り込みからの復帰

## 7.17 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 ... 402

**BR**Branch  
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】

二モニック	オペランド (target)
BR	!addr16
	!!addr20
	rp
	rg
	[ rp ]
	[ rg ]
	\$addr20
	!\$addr20

【フラグ】

S	Z	AC	P/V	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) のデータをPCに転送し、分岐します。

オペランドとして!addr16, rp, [ rp ] を指定した場合、分岐先のアドレスはベース領域 (0-FFFFH) に限定されてしまいます ( [ rp ] の場合は、分岐先テーブルもベース領域に限定されます)。実行時間の短縮やオブジェクト・サイズの圧縮などがどうしても必要な場合と、78K0, 78K1, 78K2, 78K3 シリーズ用のプログラムを終了する場合で、プログラムの修正が困難な場合にのみ使用してください。これらの命令を使用したプログラムは、そのプログラムを流用しようとしたときに修正が必要になる可能性があります。NEC製のアセンブラRA78K4では、BR addrと記述することにより、BR \$addr20, BR !\$addr20, BR !addr16, BR !!addr20の中から最適と推定できるオブジェクトを自動的に選択して生成することができます。

【記述例】

BR TDE ; TDEレジスタの内容をアドレスとして分岐

## 7.18 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。

BNZ ...	404
BNE ...	404
BZ ...	405
BE ...	405
BNC ...	406
BNL ...	406
BC ...	407
BL ...	407
BNV ...	408
BPO ...	408
BV ...	409
BPE ...	409
BP ...	410
BN ...	411
BLT ...	412
BGE ...	413
BLE ...	414
BGT ...	415
BNH ...	416
BH ...	417
BF ...	418
BT ...	419
BTCLR ...	420
BFSET ...	421
DBNZ ...	422

**BNZ**  
**BNE**Branch if Not Zero/Not Equal  
ゼロ・フラグによる条件分岐 (Z = 0)

【命令形式】 **BNZ \$addr20**  
**BNE \$addr20**

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

## 【オペランド】

二モニック	オペランド (\$addr20)
<b>BNZ</b>	\$addr20
<b>BNE</b>	

## 【フラグ】

S	Z	AC	P/V	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。

BNZ命令とBNE命令の動作は同じです。使い分けは次のとおりです。

- ・BNZ命令：加算，減算，増減命令，8ビット論理演算，シフト・ローテート命令の結果が0になったかどうかを調べる。
- ・BNE命令：比較命令のあとで、一致したかどうかを調べる。

2の補数形式のデータの加算時に8ビットの場合、- 80H, 16ビットの場合、- 8000Hどうしを加算すると、Z = 1となってしまいます。2の補数形式のデータの加算結果で0か0でないかを判断する場合は、事前にオーバーフロー・フラグ (V) でオーバーフローがないかを確認してください。

## 【記述例】

**CMP A, #55H**

**BNE \$0A39H** ; Aレジスタが0055Hでないとき、00A39Hへ分岐

ただし、BNE命令の先頭アドレスは、009B8H-00AB7H番地にあるものとする

**BZ**  
**BE**Branch if Zero/Equal than  
ゼロ・フラグによる条件分岐 (Z = 1)

【命令形式】 **BZ \$addr20**  
**BE \$addr20**

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

【オペランド】

二モニック	オペランド (\$addr20)
<b>BZ</b>	\$addr20
<b>BE</b>	

【フラグ】

S	Z	AC	P/V	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。

BZ命令とBE命令の動作は同一です。使い分けは次のとおりです。

- ・ BZ命令：加算，減算，増減命令，8ビット論理演算，シフト・ローテート命令の結果が0になったかどうかを調べる。
- ・ BE命令：比較命令のあとで，一致したかどうかを調べる。

2の補数形式のデータの加算時に8ビットの場合，- 80H, 16ビットの場合，- 8000Hどうしを加算すると，Z = 1となってしまいます。2の補数形式のデータの加算結果で0か0でないかを判断する場合は，事前にオーバーフロー・フラグ (V) でオーバーフローがないかを確認してください。

【記述例】

**DEC B**

**BZ \$3C5H** ; Bレジスタが0なら003C5Hに分岐

ただし，BZ命令の先頭アドレスは，00344H-00443H番地にあるものとする

**BNC**  
**BNL**Branch if Not Carry/Less than  
キャリー・フラグによる条件分岐 (CY = 0)

【命令形式】 **BNC \$addr20**  
**BNL \$addr20**

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

【オペランド】

二モニック	オペランド (\$addr20)
<b>BNC</b>	\$addr20
<b>BNL</b>	

【フラグ】

S	Z	AC	P/V	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の場合は、何も処理を行わず、次に続く命令を実行します。

BNC命令とBNL命令の動作は同一です。使い分けは次のとおりです。

- ・ BNC命令：加算命令、シフト・ローテート命令のあとでキャリーが発生したかどうかを調べる。  
ビット操作の結果を判定。
- ・ BNL命令：減算命令のあとでボローが発生したかどうかを調べる。  
符号なしデータに対する比較命令のあとで比較命令の第1オペランドが小さくないかどうかを調べる。

【記述例】

**CMP A, B** ; Aレジスタの内容とBレジスタの内容の大小を比較

**BNL \$1500H** ; Aレジスタの内容がBレジスタの内容より小さくなければ01500Hへ分岐

ただし、BNL命令の先頭アドレスは、0147FH-0157EH番地にあるものとする

**BC**  
**BL**Branch if Carry/Less than  
キャリー・フラグによる条件分岐 (CY = 1)【命令形式】 BC \$addr20  
BL \$addr20

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

【オペランド】

モニタ	オペランド (\$addr20)
BC	\$addr20
BL	

【フラグ】

S	Z	AC	P/V	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の場合は、何も処理を行わず、次に続く命令を実行します。

BC命令とBL命令の動作は同一です。使い分けは次のとおりです。

- ・ BC命令：加算命令、シフト・ローテート命令のあとでキャリーが発生したかどうかを調べる。  
ビット操作の結果を判定。
- ・ BL命令：減算命令のあとでボローが発生したかどうかを調べる。  
符号なしデータに対する比較命令のあとで比較命令の第1オペランドが小さいかどうかを調べる。

【記述例】

BC \$300H ; CY = 1なら00300Hへ分岐

ただし、BC命令の先頭アドレスは、0027FH-0037EH番地にあるものとする

**BNV**  
**BPO**Branch if No Overflow/Branch if Parity Odd  
パリティ / オーバフロー・フラグによる条件分岐 (P/V = 0)【命令形式】 **BNV \$addr20**  
**BPO \$addr20**【オペレーション】 **PC PC + 2 + jdisp8 if P/V = 0**

## 【オペランド】

モニタック	オペランド (\$addr20)
<b>BNV</b>	\$addr20
<b>BPO</b>	

## 【フラグ】

S	Z	AC	P/V	CY

## 【説明】

P/V = 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。

P/V = 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。

BNV命令とBPO命令の動作は同一です。使い分けは次のとおりです。

- ・BNV命令：2の補数形式のデータの演算後などに、結果がオーバフローもアンダフローもしていないことを調べる。
- ・BPO命令：論理演算命令またはシフト・ローテート命令などの実行結果のパリティが奇数であることを調べる。

## 【記述例】

**ADD B, C** ; BレジスタとCレジスタの内容 (2の補数形式のデータ) を加算

**BNV \$560H** ; 加算結果にオーバフローがなければ560Hへ分岐

ただし、BNV命令の先頭アドレスは、004DFH-05DEH番地にあるものとする

**BV**  
**BPE**Branch if Overflow/Branch if Parity Even  
パリティ / オーバフロー・フラグによる条件分岐 (P/V = 1)

【命令形式】 **BV \$addr20**  
**BPE \$addr20**

【オペレーション】 **PC PC + 2 + jdisp8 if P/V = 1**

## 【オペランド】

モニタ	オペランド (\$addr20)
<b>BV</b>	\$addr20
<b>BPE</b>	

## 【フラグ】

S	Z	AC	P/V	CY

## 【説明】

P/V = 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。

P/V = 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。

BV命令とBPE命令の動作は同一です。使い分けは次のとおりです。

- ・BV命令 : 2の補数形式のデータの演算後などに、結果がオーバフローもアンダフローしていることを調べる。
- ・BPE命令 : 論理演算命令またはシフト・ローテート命令などの実行結果のパリティが偶数であることを調べる。

## 【記述例】

**OR D, #055H** ; Dレジスタの内容と055Hのビットごとの論理和をとる

**BPE \$841EH** ; 論理和をとった結果、パリティが偶数の場合に841EHに分岐する

ただし、この命令の先頭アドレスは、839DH-849CH番地にあるものとする

**BP**Branch if Positive  
サイン・フラグによる条件分岐 (S = 0)

【命令形式】 PB \$addr20

【オペレーション】 PC PC + 2 + jdisp8 if S = 0

【オペランド】

二モニック	オペランド (\$addr20)
BP	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

S = 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。

S = 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。

この命令は、2の補数形式のデータの演算後に結果が正（0を含む）であることを調べるのに使用します。ただし、演算結果がオーバーフローまたはアンダフローしている場合には、正しい判定ができません（事前に、BV命令またはBNV命令でオーバーフローまたはアンダフローしていないことを確認してから使用するか、BGE命令を使用してください）。

【記述例】

**BV \$OVER** ; 演算結果がオーバーフローまたはアンダフローしていれば、OVER番地へ分岐

**BP \$TARGET** ; 演算結果が正（0を含む）ならば、TARGET番地へ分岐

ただし、TARGET番地は、BP命令の先頭アドレスの - 126 ~ + 129番地の間にあるものとする

**BN**Branch if Negative  
サイン・フラグによる条件分岐 (S = 1)

【命令形式】 BN \$addr20

【オペレーション】 PC PC + 2 + jdisp8 if S = 1

【オペランド】

二モニック	オペランド (\$addr20)
BN	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

S = 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。

S = 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。

この命令は、2の補数形式のデータの演算後に結果が負であることを調べるのに使用します。ただし、演算結果がオーバフローまたはアンダフローしている場合には、正しい判定ができません（事前に、BV命令またはBNV命令でオーバフローまたはアンダフローしていないことを確認してから使用するか、BLT命令を使用してください）。

【記述例】

BN #TARGET ; 演算結果が負の場合にTARGET番地へ分岐

**BLT**Branch if less than  
数値の大小による条件分岐 (...より小さい)

【命令形式】 BLT \$addr20

【オペレーション】 PC ← PC + 3 + jdisp8 if P/V - S = 1

【オペランド】

二モニック	オペランド (\$addr20)
BLT	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

P/V - S = 1の場合に、オペランドで指定したアドレスへ分岐します。

P/V - S = 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。

この命令は、2の補数形式のデータの大小判定や、演算結果が負であることを調べるのに使用します。大小判定は、直前に実行したCMP命令の第1オペランドが第2オペランドより小さいかを調べます。また、演算結果が、アンダフローした場合も含めて負であることを調べるのに使用します。

【記述例】

CMPW AX, #3456H

BLT \$8123H ; AXレジスタの内容が、3456Hより小さい場合に8123H番地へ分岐  
ただし、BLT命令の先頭アドレスは、80A1H-81A0H番地にあるものとする

**BGE**Branch if Greater than/Equal  
数値の大小による条件分岐 (...以上)

【命令形式】 BGE \$addr20

【オペレーション】 PC PC + 3 + jdisp8 if P/V - S = 0

【オペランド】

二モニック	オペランド (\$addr20)
BGE	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

P/V - S = 0の場合に、オペランドで指定したアドレスへ分岐します。

P/V - S = 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。

この命令は、2の補数形式のデータの大小判定や、演算結果が0または正であることを調べるのに使用します。大小判定は、直前に実行したCMP命令の第1オペランドが第2オペランドと同じか大きいかを調べます。また、演算結果が、オーバーフローした場合も含めて0以上であることを調べるのに使用します。

【記述例】

ADDW AX, BC

BGE \$23456H ; 直前の加算命令の結果が0以上の場合に、23456H番地へ分岐

ただし、BGE命令の先頭アドレスは、233D4H-234D3H番地にあるものとする

**BLE**Branch if less than/Equal  
数値の大小による条件分岐 (...以下)

【命令形式】 BLE \$addr20

【オペレーション】 PC PC + 3 + jdisp8 if (P/V - S) Z = 1

【オペランド】

二モニック	オペランド (\$addr20)
BLE	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

(P/V - S) Z = 1の場合に、オペランドで指定したアドレスへ分岐します。

(P/V - S) 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。

この命令は、2の補数形式のデータの大小判定や、演算結果が0を含めて負であることを調べるのに使用します。大小判定は、直前に実行したCMP命令の第1オペランドが第2オペランドと同じか小さいかを調べます。また、演算結果が、アンダフローした場合も含めて負であることを調べるのにも使用します。

【記述例】

SUB H, L

BLE \$789ABH ; 直前の減算命令の結果が、アンダフローした場合も含めて0以下である場合に789ABHへ分岐

ただし、BLE命令の先頭アドレスは、78929H-789ABH番地にあるものとする

**BGT**Branch if Greater than  
数値の大小による条件分岐 (...より大きい)

【命令形式】 BGT \$addr20

【オペレーション】 PC PC + 3 + jdisp8 if (P/V - S) Z = 0

【オペランド】

二モニック	オペランド (\$addr20)
BGT	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

(P/V - S) Z = 0の場合に、オペランドで指定したアドレスへ分岐します。

(P/V - S) 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。

この命令は、2の補数形式のデータの大小判定や、演算結果が0より大きいことを調べるのに使用します。

大小判定は、直前に実行したCMP命令の第1オペランドが第2オペランドより大きいかを調べます。また、演算結果が、オーバーフローした場合も含めて0より大きいことを調べるのに使用します。

【記述例】

CMP A, E

BGT \$0CFFEDH ; Aレジスタの内容が、Bレジスタの内容より大きい場合に、0CFFEDH番地に分岐  
ただし、BGT命令の先頭アドレスは、0CFF6BH-0D006DH番地にあるものとする

**BNH**Branch if Not Higher than  
数値の大小による条件分岐 (...より大きくない)

【命令形式】 BNH \$addr20

【オペレーション】 PC PC + 3 + jdisp8 if Z CY = 1

【オペランド】

二モニック	オペランド (\$addr20)
<b>BNH</b>	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

Z CY = 1の場合に、オペランドで指定されるアドレスへ分岐します。

Z CY = 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。

この命令は、符号なしデータの大小判定をする場合に使用します。直前に実行したCMP命令の第1オペランドが第2オペランドより大きくない(第1オペランドが第2オペランドと同じか小さい)ことを調べます。

【記述例】

**CMPW RP2, #8921H**

**BNH \$TARGET** ; RP2レジスタの内容が、8921Hより大きくない(9821Hと同じか小さい)ときに、TARGET番地へ分岐

ただし、BNH命令の先頭アドレスは、TARGET番地へ分岐可能なアドレスにあるものとする

**BH**Branch if Higher than  
数値の大小による条件分岐 (...より大きい)

【命令形式】 BH \$addr20

【オペレーション】 PC PC + 3 + jdisp8 if Z CY = 0

【オペランド】

二モニック	オペランド (\$addr20)
BH	\$addr20

【フラグ】

S	Z	AC	P/V	CY

【説明】

Z CY = 0の場合に、オペランドで指定されるアドレスへ分岐します。

Z CY = 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。

この命令は、符号なしデータの大小を判定するのに使用します。直前のCMP命令の第1オペランドが第2オペランドより大きいことを調べます。

【記述例】

CMP B, C

BH \$356H ; Bレジスタの内容がCレジスタの内容より大きい場合、356Hへ分岐

ただし、BH命令の先頭アドレスは、2D4H-3D3H番地にあるものとする

**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, \$addr20

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

【オペランド】

二モニック	オペランド ( bit, \$addr20 )	b ( バイト数 )
BF	saddr.bit, \$addr20	4/5
	sfr.bit, \$addr20	4
	X.bit, \$addr20	3
	A.bit, \$addr20	3
	PSWL.bit, \$addr20	3
	PSWH.bit, \$addr20	3
	mem2.bit, \$addr20	3
	!addr16.bit, \$addr20	6
	!!addr24.bit, \$addr20	7

【フラグ】

S	Z	AC	P/V	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 ( \$addr20 ) 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、01549H番地へ分岐

ただし、BF命令の先頭アドレスは、014C6H-015C5H番地にあるものとする

**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, \$addr20

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

【オペランド】

二モニック	オペランド ( bit, \$addr20 )	b ( バイト数 )
BT	saddr.bit, \$addr20	3/4
	sfr.bit, \$addr20	4
	X.bit, \$addr20	3
	A.bit, \$addr20	3
	PSWL.bit, \$addr20	3
	PSWH.bit, \$addr20	3
	mem2.bit, \$addr20	3
	!addr16.bit, \$addr20	6
	!!addr24.bit, \$addr20	7

【フラグ】

S	Z	AC	P/V	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 0FE47H.3, \$55CH ; 0FE47H番地のビット3が1のとき、0055CHへ分岐

ただし、BT命令の先頭アドレスは、004D9H-005D8H番地にあるものとする

**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, \$addr20

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

【オペランド】

二モニック	オペランド ( bit, \$addr20 )	b ( バイト数 )
BTCLR	saddr.bit, \$addr20	4/5
	sfr.bit, \$addr20	4
	X.bit, \$addr20	3
	A.bit, \$addr20	3
	PSWL.bit, \$addr20	3
	PSWH.bit, \$addr20	3
	mem2.bit, \$addr20	3
	!addr16.bit, \$addr20	6
	!!addr24.bit, \$addr20	7

【フラグ】

bitがPSWL.bit

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

左記以外

S	Z	AC	P/V	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 ( 0 ) して、00356H番地へ分岐

ただし、BTCLR命令の先頭アドレスは、002D4H-003D3H番地にあるものとする

**BFSET**

Branch if False and Set

ビット・テストによる条件分岐とセット (バイト・データのビット = 0)

【命令形式】 BFSET bit, \$addr20

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

【オペランド】

二モニック	オペランド ( bit, \$addr20 )	b ( バイト数 )
BFSET	saddr.bit, \$addr20	4/5
	sfr.bit, \$addr20	4
	X.bit, \$addr20	3
	A.bit, \$addr20	3
	PSWL.bit, \$addr20	3
	PSWH.bit, \$addr20	3
	mem2.bit, \$addr20	3
	!addr16.bit, \$addr20	6
	!!addr24.bit, \$addr20	7

【フラグ】

bitがPSWL.bit

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

左記以外

S	Z	AC	P/V	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1オペランド ( bit ) の内容をセット ( 1 ) 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 ) されているときは、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セット ( 1 ) されます。

【記述例】

**BFSET A.6, \$3FFE1H** ; Aレジスタのビット6が0の場合、Aレジスタのビット6をセット ( 1 ) して、3FFE1H番地へ分岐

ただし、BFSET命令の先頭アドレスは、3FF5FH-4005EH番地にあるものとする

**DBNZ**

Decrement and Branch if Not Zero

条件ループ (dst 0)

【命令形式】 DBNZ dst, \$addr20

【オペレーション】 dst dst - 1 ,  
 then PC PC + b + jdisp8 if dst 0

【オペランド】

モニック	オペランド (dst, \$addr20)	b (バイト数)
DBNZ	B, \$addr20	2
	C, \$addr20	2
	saddr, \$addr20	3/4

【フラグ】

S	Z	AC	P/V	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 (\$addr20) 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001215Hへ分岐

ただし、DBNZ命令の先頭アドレスは、001194H-001293H番地にあるものとする

## 7.19 CPU制御命令

CPU制御命令には、次の命令があります。

MOV STBC, #byte ... 424

MOV WDM, #byte ... 425

LOCATION ... 426

SEL RBn ... 427

SEL RBn, ALT ... 428

SWRS ... 429

NOP ... 430

EI ... 431

DI ... 432

**MOV STBC, #byte**Move  
スタンバイ・モードの設定

【命令形式】 MOV STBC #byte

【オペレーション】 STBC byte

【オペランド】

二モニック	オペランド
MOV	STBC, #byte

【フラグ】

S	Z	AC	P/V	CY

【説明】

この命令はスタンバイ・コントロール・レジスタ（STBC）への専用の書き込み命令です。第2オペランドで指定されるイミディエト・データをSTBCへ書き込みます。STBCへの書き込みは、この命令によってのみ可能です。

この命令は特殊なフォーマットをしており、命令コードでは、書き込みを行うイミディエト・データのほかに、その値の論理否定をとったデータを用意する必要があります（下図参照）。

（NEC製のアセンブラ（RA78K4）では、自動的に生成します）。

・命令コードのフォーマット

0	0	0	0	1	0	0	1
1	1	0	0	0	0	0	0
←	byte		→				
←	byte		→				

CPUは、書き込みを行いたいイミディエト・データと論理否定をとったデータを検査し、正しい場合のみ書き込みを行います。正しくない場合は、書き込みを行わずにオペランド・エラー割り込みを発生します。

【記述例】

**MOV STBC, #2 ; STBCに2を書き込む（STOPモードにする）**

**MOV WDM, #byte**Move  
ウォッチドッグ・タイマの設定

【命令形式】 MOV WDM #byte

【オペレーション】 WDM byte

【オペランド】

二モニック	オペランド
MOV	WDM, #byte

【フラグ】

S	Z	AC	P/V	CY

【説明】

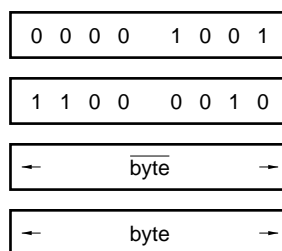
この命令はウォッチドッグ・タイマ・モード・レジスタ (WDM) への専用の書き込み命令です。第2オペランドで指定されるイミディエト・データをWDMへ書き込みます。WDMへの書き込みは、この命令によってのみ可能です。

この命令は、ウォッチドッグ・タイマを内蔵している製品でしか使用できません。ウォッチドッグ・タイマを内蔵しているかないかは、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。

この命令は特殊なフォーマットをしており、命令コードでは、書き込みを行うイミディエト・データのほかに、その値の論理否定をとったデータを用意する必要があります(下図参照)。

(NEC製のアセンブラ (RA78K4) では、自動的に生成します)。

・命令コードのフォーマット



CPUは、書き込みを行いたいイミディエト・データと論理否定をとったデータを検査し、正しい場合のみ書き込みを行います。正しくない場合は、書き込みを行わずにオペランド・エラー割り込みを発生します。

【記述例】

MOV WDM, #0C0H ; WDMに0C0Hを書き込む

# LOCATION

Location  
ロケーション

【命令形式】 LOCATION locaddr

【オペレーション】 SFR, 内部データ領域の配置アドレスの上位ワード指定

【オペランド】

二モニック	オペランド
LOCATION	locaddr

【フラグ】

S	Z	AC	P/V	CY

【説明】

この命令は、内部データ領域（内部RAMおよび特殊機能レジスタ（SFR））のアドレスを指定するための命令です。0が指定された場合、内部データ領域の最大のアドレスは、0FFFFHになります。0FHが指定された場合、内部データ領域の最大のアドレスは、0FFFFFFHになります。

この命令と、次に続く命令の間では、割り込みおよびマクロ・サービスを受け付けません。

この命令は、リセット解除直後に必ず実行してください。すなわち、リセット・ベクタで指定されるアドレスに、必ずこの命令を配置してください。なお、この命令は、2回以上使用することはできません。また、2回以上実行しても、2回目以降の実行では、内部データ領域のアドレスを変更することはできません。この命令のオペランドは、次のようにコード化されます。

locaddr	オペランドのコード
0H	01FEH
0FH	00FFH

これ以外の値では、この命令の実行は無視されます。さらに、オペランドの上位バイトと下位バイトの排他的論理和をとった結果が0FFHではない場合には、オペランド・エラー割り込みを発生します。

【記述例】

LOCATION 0FH ; 内部データ領域の最大アドレスを0FFFFFFHに指定

**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】 RSS 0, RBS2 - 0 n ; (n = 0 - 3)

【オペランド】

二モニック	オペランド (RBn)
SEL	RBn

【フラグ】

S	Z	AC	P/V	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-RB7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

**SEL RBn, ALT**Select Register Bank  
レジスタ・バンクの選択

【命令形式】 SEL RBn, ALT

【オペレーション】 RSS1 1, RBS2 - 0 n ; (n = 0 - 3)

【オペランド】

二モニック	オペランド
SEL	RBn, ALT

【フラグ】

S	Z	AC	P/V	CY

【説明】

第1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、さらに、レジスタ選択フラグ (RSS) をセット (1) します。

RBnには、RB0-RB7まであります。

この命令は、78K/ シリーズとの互換性を保つために用意してある命令です。78K/ シリーズ用のプログラムを流用する場合のみ、使用することができます。78K/ シリーズ以外の78Kシリーズのプログラムを流用する場合、および新規に作成するプログラムでは使用しないでください。

**SWRS**Switch Register Set  
レジスタ・ビットの切り替え

【命令形式】 SWRS

【オペレーション】 RSS  $\overline{\text{RSS}}$ 

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY

【説明】

レジスタ・セット選択フラグ (RSS) の内容を反転します。

この命令は、78K $\mu$  シリーズとの互換性を保つために用意してある命令です。78K $\mu$  シリーズ用のプログラムを流用する場合のみ、使用することができます。78K $\mu$  シリーズ以外の78Kシリーズのプログラムを流用する場合、および新規に作成するプログラムでは使用しないでください。

**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】

S	Z	AC	P/V	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 (Enable interrupt)

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	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）。この命令の実行後、一定の間では、すべての割り込みまたはマクロ・サービスを受け付けません。詳細については、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。

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 ハードウェア編**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 (Disable interrupt)

【オペランド】

なし

【フラグ】

S	Z	AC	P/V	CY

【説明】

マスク可能割り込みのベクタ割り込みによる受け付けを禁止にします（割り込み許可フラグ（IE）をクリア（0）します）。

この命令の実行後、一定の間では、すべての割り込みまたはマクロ・サービスを受け付けません。詳細については、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。

割り込みの処理の詳細については、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。

## 7.20 特殊命令

特殊命令には、次の命令があります。

CHKL ... 434

CHKLA ... 436

**CHKL**Check Level  
端子の出力レベルのチェック

【命令形式】 CHKL sfr

【オペレーション】 (端子レベル) - (出力ラッチ)

【オペランド】

二モニック	オペランド
CHKL	sfr

【フラグ】

S	Z	AC	P/V	CY
x	x		P	

【説明】

出力端子の端子レベルと出力バッファの前段の信号レベルとの排他的論理和をとります。

排他的論理和の結果、ビット7がセット(1)されている場合、Sフラグをセット(1)し、ビット7がクリア(0)されている場合は、Sフラグをクリア(0)します。

排他的論理和の結果、全ビットが0の場合には、Zフラグをセット(1)し、0でないビットがある場合は、Zフラグをクリア(0)します。

排他的論理和の結果、データ中のセット(1)されているビットの数が偶数の場合、P/Vフラグをセット(1)し、奇数の場合、クリア(0)します。

この命令は、何らかの原因で発生した、出力端子の端子レベルと出力バッファの前段の信号レベルが異なるという異常な状態を検出するための命令です。正常動作時には、必ずZフラグがセット(1)されます。

この命令を実行する場合には、ポート・リード・コントロール・レジスタ (PRDC) を持っている製品では、PRDCレジスタのPRDC0ビットがクリア(0)されている必要があります。PRDC0ビットがセット(1)されていると、異常状態を検出できません。

コントロール出力として使用している端子を含むポートについて、この命令を実行する場合には、コントロール出力として使用している端子のポートとしての入出力モードは、入力モードに必ず指定してください。コントロール出力として使用している端子のポートとしての入出力モードが出力モードになっている場合、正常動作していても異常動作と判断される場合があります。

ポートとしての入出力モードが入力モードに指定されている端子については、この命令では必ず正常と判断してしまいます。

## 【記 述 例】

**CHKL P0**

**BNZ \$ERROR** ; ポート0の端子レベルと出力バッファの前段の信号レベルが一致していることを調べ、一致していない場合には、ERROR番地へ分岐

**注意**  $\mu$  PD784216A, 784216AY, 784218A, 784218AY, 784225, 784225Y, 784938Aサブシリーズの製品は、CHKLをサポートしていません。この命令は、使用しないでください。

この命令を実行した場合、次のような動作をします。

- ・ 出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Zフラグがセット(1)されます。

**CHKLA**

Check Level and Transfer to Register  
端子の出力レベルのチェックとレジスタへの転送

【命令形式】 CHKLA sfr

【オペレーション】 A (端子レベル) - (出力ラッチ)

【オペランド】

二モニック	オペランド
CHKLA	sfr

【フラグ】

S	Z	AC	P/V	CY
x	x		P	

【説明】

出力端子の端子レベルと出力バッファの前段の信号レベルとの排他的論理和をとり、結果をAレジスタへ格納します。

排他的論理和の結果、ビット7がセット(1)されている場合、Sフラグをセット(1)し、ビット7がクリア(0)されている場合は、Sフラグをクリア(0)します。

排他的論理和の結果、全ビットが0の場合には、Zフラグをセット(1)し、0でないビットがある場合は、Zフラグをクリア(0)します。

排他的論理和の結果、データ中のセット(1)されているビットの数が偶数の場合、P/Vフラグをセット(1)し、奇数の場合、クリア(0)します。

この命令は、何らかの原因で発生した、出力端子の端子レベルと出力バッファの前段の信号レベルが異なるという異常な状態を検出するための命令です。正常動作時には、必ずZフラグがセット(1)されます。

この命令を実行する場合には、ポート・リード・コントロール・レジスタ (PRDC) を持っている製品では、PRDCレジスタのPRDC0ビットがクリア(0)されている必要があります。PRDC0ビットがセット(1)されていると、異常状態を検出できません。

コントロール出力として使用している端子を含むポートについて、この命令を実行する場合には、コントロール出力として使用している端子のポートとしての入出力モードは、入力モードに必ず指定してください。コントロール出力として使用している端子のポートとしての入出力モードが出力モードになっている場合、正常動作していても異常動作と判断される場合があります。

ポートとしての入出力モードが入力モードに指定されている端子については、この命令では必ず正常と判断してしまいます。

## 【記 述 例】

**CHKLA P3** ; ポート3の端子レベルと出力バッファの前段の信号レベルが一致していることを調べ、結果をAレジスタへ格納

**注意**  $\mu$  PD784216A, 784216AY, 784218A, 784218AY, 784225, 784225Y, 784938Aサブシリーズの製品は、CHKLAをサポートしていません。この命令は、使用しないでください。

この命令を実行した場合、次のような動作をします。

- ・出力端子の端子レベルを2度読み込んで、それらの排他的論理和をとります。そのため、この命令によりチェックする端子がポートの出力モードとして使用されている場合は、排他的論理和の結果は必ず全ビットが0になり、Aレジスタにその結果を格納するとともに、Zフラグをセット(1)します。

## 7.21 ストリング命令

ストリング命令には、次の命令があります。

MOVTBLW	...	439
MOVM	...	441
XCHM	...	443
MOVBK	...	445
XCHBK	...	447
CMPME	...	449
CMPMNE	...	452
CMPMC	...	455
CMPMNC	...	458
CMPBKE	...	461
CMPBKNE	...	464
CMPBKC	...	467
CMPBKNC	...	470

**MOVTBLW**Move Table Word  
テーブル・ワードの転送【命令形式】 **MOVTBLW !addr8, byte**

【オペレーション】 **(addr8 + 2) (addr8),**  
**byte byte - 1,**  
**addr8 addr8 - 2,**  
**End if byte = 0**

【オペランド】

二モニック	オペランド
<b>MOVTBLW</b>	<b>!addr16, byte</b>

【フラグ】

S	Z	AC	P/V	CY

【説明】

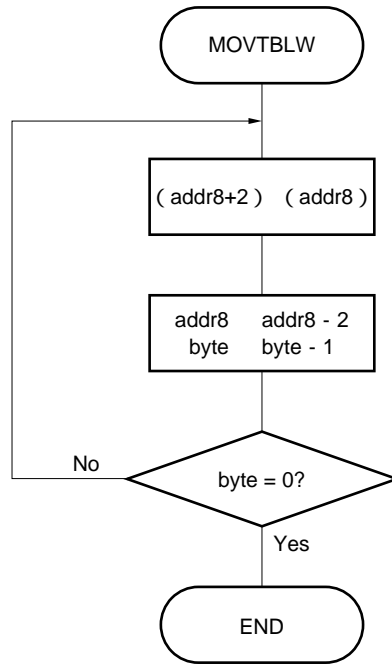
第1オペランドで指定される16ビット・イミディエト・データでアドレスされるメモリの内容を、+2したアドレスに転送します。その後、addr8を-2します。以上の動作を第2オペランドに記述された8ビット・イミディエト・データの回数分繰り返します。

この命令は、MACW、MACSW命令で使用するデータ・テーブルをシフトするのに使用します。

第1オペランドの!addr8には、転送を行いたいデータの最上位のデータのアドレスをラベルまたは数値で直接記述します。

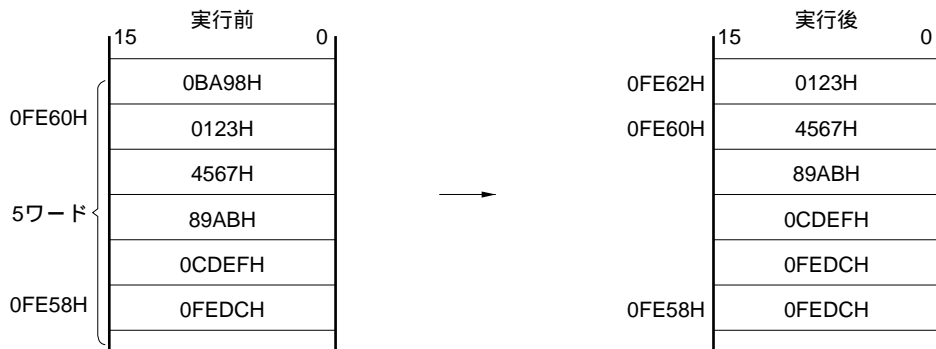
第1オペランドに記述できるアドレスは、LOCATION 0H命令を実行した場合は00FE00H-00FEFFHに、LOCATION 0FH命令を実行した場合は0FFE00H-0FFEFFHに限定されます。

**備考**  $\mu$ PD784915サブシリーズは、LOCATION 0H命令固定です。



【記述例】

**MOVTLW !0FFE60H, 5 ; 0FFE58H-0FFE60Hのデータを, 0FFE5AH-0FFE62Hへ転送**



**MOVM**Move Multiple Byte  
固定のバイト・データのブロック転送

【命令形式】 **MOVM [TDE + ], A**  
**MOVM [TDE - ], A**

【オペレーション】 (TDE) A, TDE TDE + 1, C C - 1 End if C = 0  
(TDE) A, TDE TDE - 1, C C - 1 End if C = 0

## 【オペランド】

二モニック	オペランド
MOVM	[TDE + ], A
	[TDE - ], A

## 【フラグ】

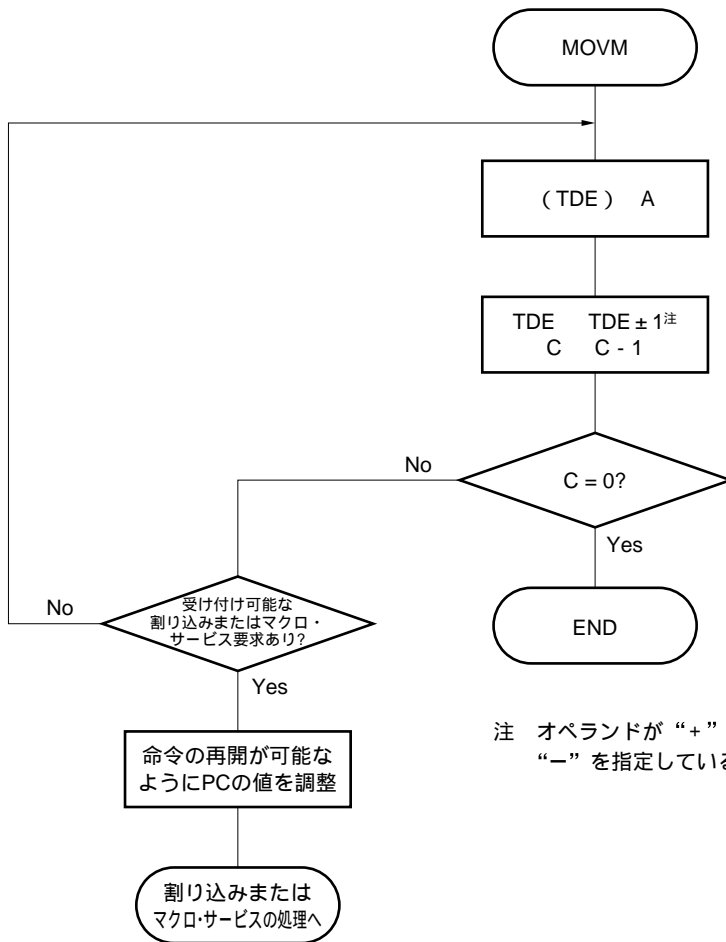
S	Z	AC	P/V	CY

## 【説明】

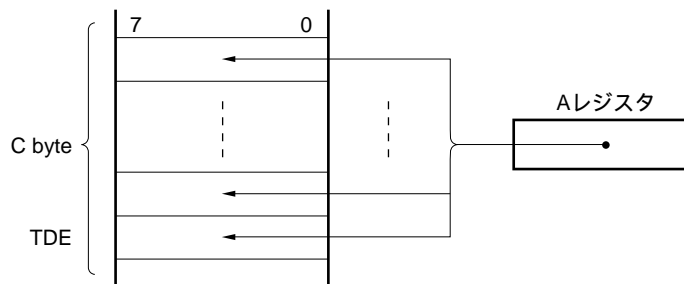
Aレジスタの内容をTDEレジスタでアドレスされるメモリに転送し、TDEレジスタの内容をインクリメント/デクリメントします。その後、Cレジスタの内容をデクリメントし、Cレジスタの内容が0になるまで以上の動作を繰り返します。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていない場合は、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

この命令は、主にメモリの一定領域を特定の値で初期化する場合などに使用します。なお、複数バイトのデータで初期化を行う場合には、MOV BK命令を使用します。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記述例】

```

MOV C, #00H      ; C 00H
MOV A, #00H      ; A 00H
MOVG TDE, #0FE00H ; TDE FE00H
MOV M [ TDE + ], A ; FE00H-FEFFFHのRAMをクリア
  
```

**XCHM**Exchange Multiple Byte  
固定のバイト・データのブロック交換

【命令形式】 XCHM [ TDE + ], A  
XCHM [ TDE - ], A

【オペレーション】 (TDE) A, TDE TDE + 1, C C - 1 End if C = 0  
(TDE) A, TDE TDE - 1, C C - 1 End if C = 0

## 【オペランド】

二モニック	オペランド
XCHM	[ TDE + ], A
	[ TDE - ], A

## 【フラグ】

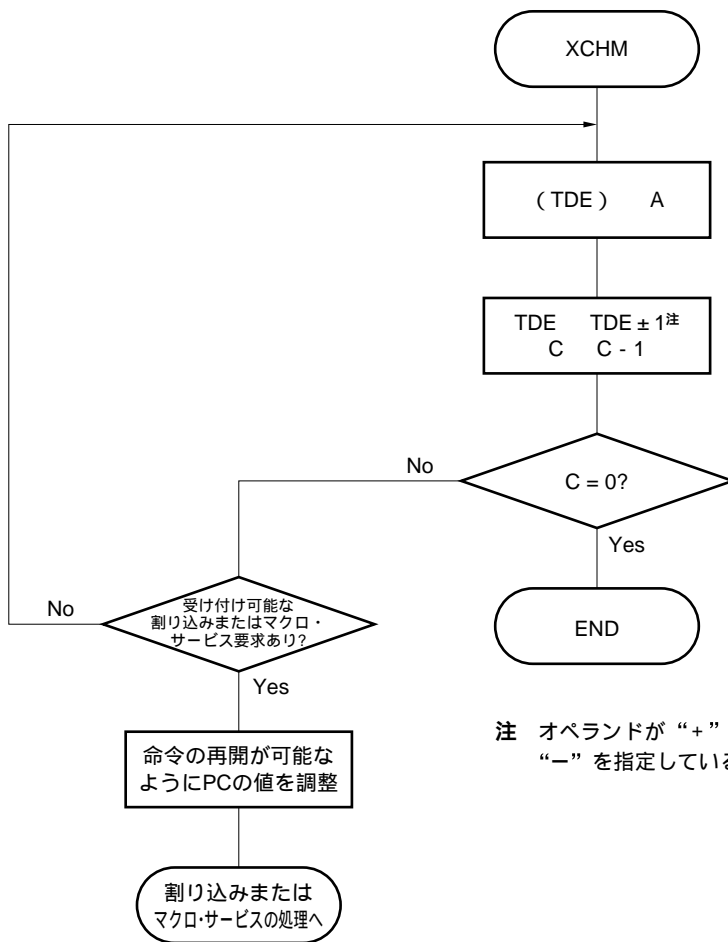
S	Z	AC	P/V	CY

## 【説明】

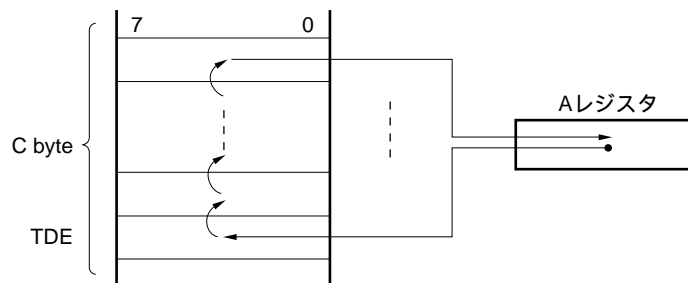
Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容とを交換し、TDEレジスタの内容をインクリメント/デクリメントします。その後、Cレジスタの内容をデクリメントし、Cレジスタの内容が0になるまで以上の動作を繰り返します。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていないならば、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

この命令は、主にメモリ中のデータを1バイト移動させるのに使用します。上位アドレス側に移動させる場合は、XCHM [ TDE + ], Aを、下位アドレス側に移動させる場合はXCHM [ TDE - ], Aを使用します。なお、2バイト移動させる場合は、MOVBK命令を使用します。



注 オペランドが “+” を指定しているときは +1 ,  
“−” を指定しているときは −1



【記述例】

MOV C, #10H ; C 10H

MOV A, #00H ; A 00H

MOVG TDE, #3050H ; TDE 3050H

XCHM [ TDE + ], A ; 3050H-305FHのメモリの内容を1番地ずつ後ろのアドレスにシフト ( 3050H番地の内容は0になる )

**MOVBK**Move Block Byte  
バイト・データのブロック転送

【命令形式】 **MOVBK [ TDE + ], [ WHL + ]**  
**MOVBK [ TDE - ], [ WHL - ]**

【オペレーション】 ( TDE ) ( WHL ), TDE TDE + 1, WHL WHL + 1 C C - 1  
 End if C = 0  
 ( TDE ) ( WHL ), TDE TDE - 1, WHL WHL - 1 C C - 1  
 End if C = 0

## 【オペランド】

二モニック	オペランド
MOVBK	[ TDE + ], [ WHL + ]
	[ TDE - ], [ WHL - ]

## 【フラグ】

S	Z	AC	P/V	CY

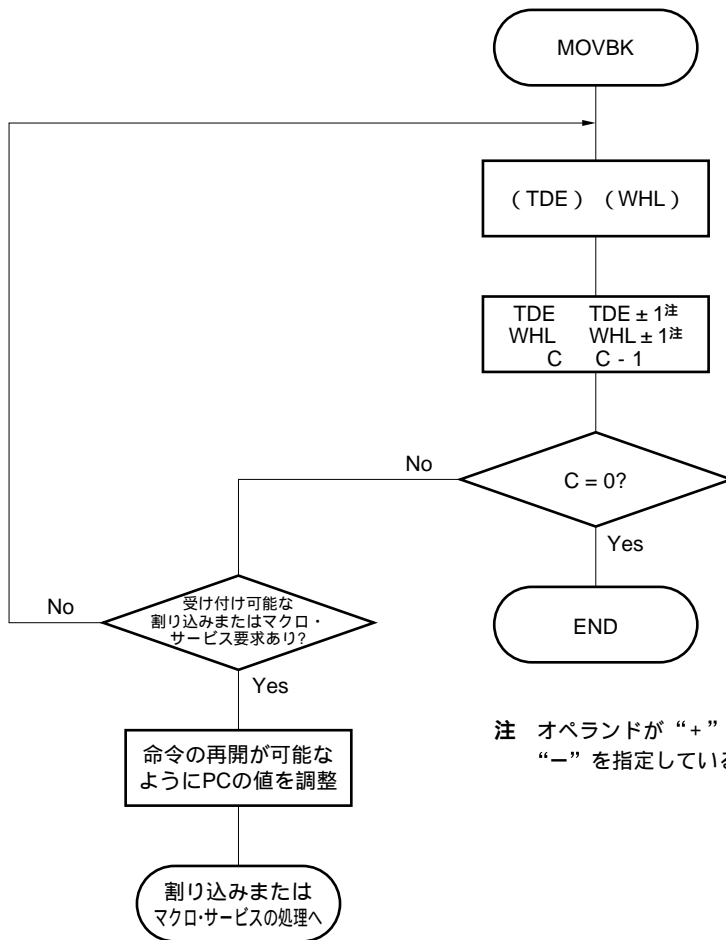
## 【説明】

WHLレジスタでアドレスされるメモリの内容を、TDEレジスタでアドレスされるメモリに転送し、TDE、WHLレジスタの内容をインクリメント/デクリメントします。その後、Cレジスタの内容をデクリメントし、Cレジスタの内容が0になるまで以上の動作を繰り返します。

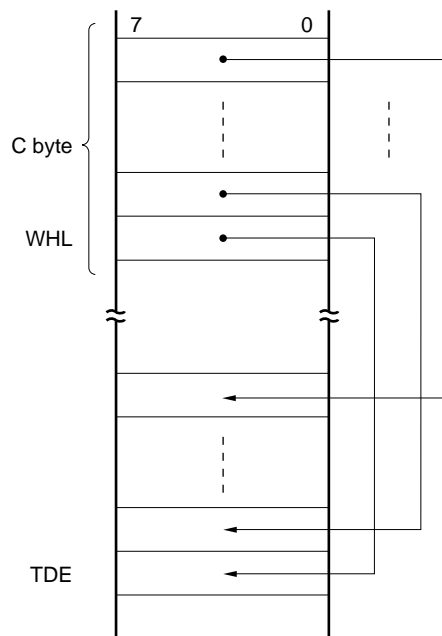
この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていない場合は、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

転送元のデータ領域と転送先のデータ領域が重なっている場合には、次のようにします。

- ・転送元の最下位アドレスが、転送先の最上位アドレスより小さい場合は、TDE, WHLレジスタともに初期値をそれぞれの最下位アドレスとし、MOVBK [ TDE + ], [ WHL + ]を使用します。
- ・転送元の最上位アドレスが、転送先の最下位アドレスより大きい場合は、TDE, WHLレジスタともに初期値をそれぞれの最上位アドレスとし、MOVBK [ TDE - ], [ WHL - ]を使用します。



注 オペランドが “+” を指定しているときは +1 ,  
“-” を指定しているときは -1



【記述例】

```

MOV C, #10H           ; C 10H
MOVG TDE, #3000H     ; TDE 3000H
MOVG WHL, #5000H     ; WHL 5000H
MOVBK [ TDE + ], [ WHL + ] ; 5000H-500FHのメモリの内容を, 3000H-300FHのメモリに転送
  
```

**XCHBK**Exchange Block Byte  
バイト・データのブロック交換

【命令形式】 XCHBK [ TDE + ], [ WHL + ]  
XCHBK [ TDE - ], [ WHL - ]

【オペレーション】 ( TDE ) ( WHL ), TDE TDE + 1,  
WHL WHL + 1 C C - 1 End if C = 0  
( TDE ) ( WHL ), TDE TDE - 1,  
WHL WHL - 1 C C - 1 End if C = 0

## 【オペランド】

二モニック	オペランド
XCHBK	[ TDE + ], [ WHL + ]
	[ TDE - ], [ WHL - ]

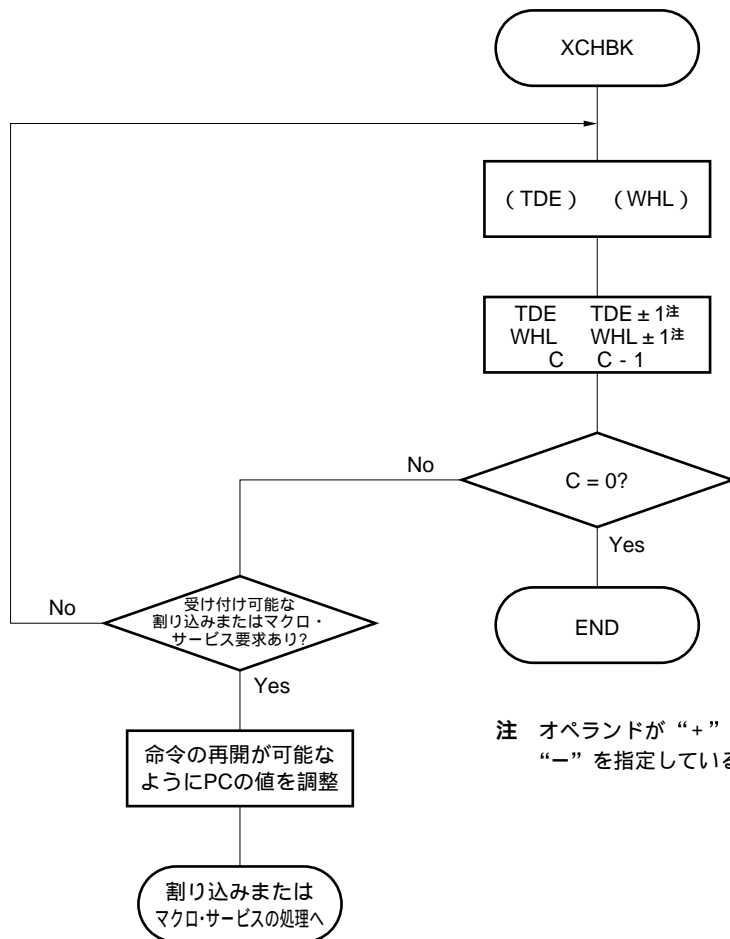
## 【フラグ】

S	Z	AC	P/V	CY

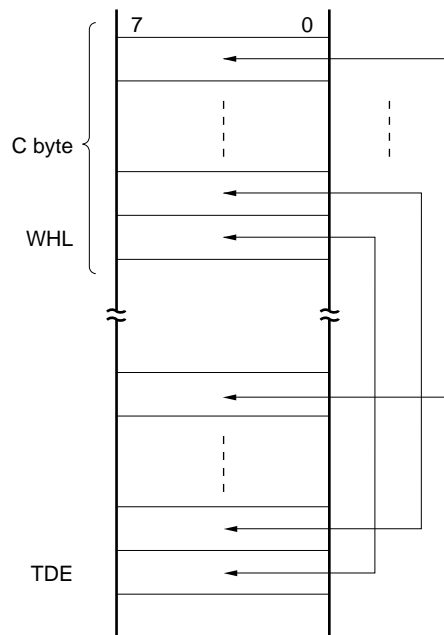
## 【説明】

WHLレジスタでアドレスされるメモリの内容とTDEレジスタでアドレスされるメモリの内容とを交換し、WHL, TDEレジスタの内容をインクリメント/デクリメントします。その後、Cレジスタの内容をデクリメントし、Cレジスタの内容が0になるまで以上の動作を繰り返します。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていないならば、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記述例】

MOV C, #80H

MOVG TDE, #3456H

MOVG WHL, #1FF96H

XCHBK [ TDE + ], [ WHL + ] ; 3456H番地からの80Hバイトのデータと1FF96H番地からのデータを交換

**CMPME**Compare Multiple Equal Byte  
固定のバイト・データとのブロック比較（一致検出）

【命令形式】 **CMPME [ TDE + ], A**  
**CMPME [ TDE - ], A**

【オペレーション】 (TDE) - A, TDE TDE + 1, C C - 1 End if C = 0 or Z = 0  
(TDE) - A, TDE TDE - 1, C C - 1 End if C = 0 or Z = 0

## 【オペランド】

二モニック	オペランド
CMPME	[ TDE + ], A
	[ TDE - ], A

## 【フラグ】

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

## 【説明】

Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDEレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較結果が不一致であるか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていないければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

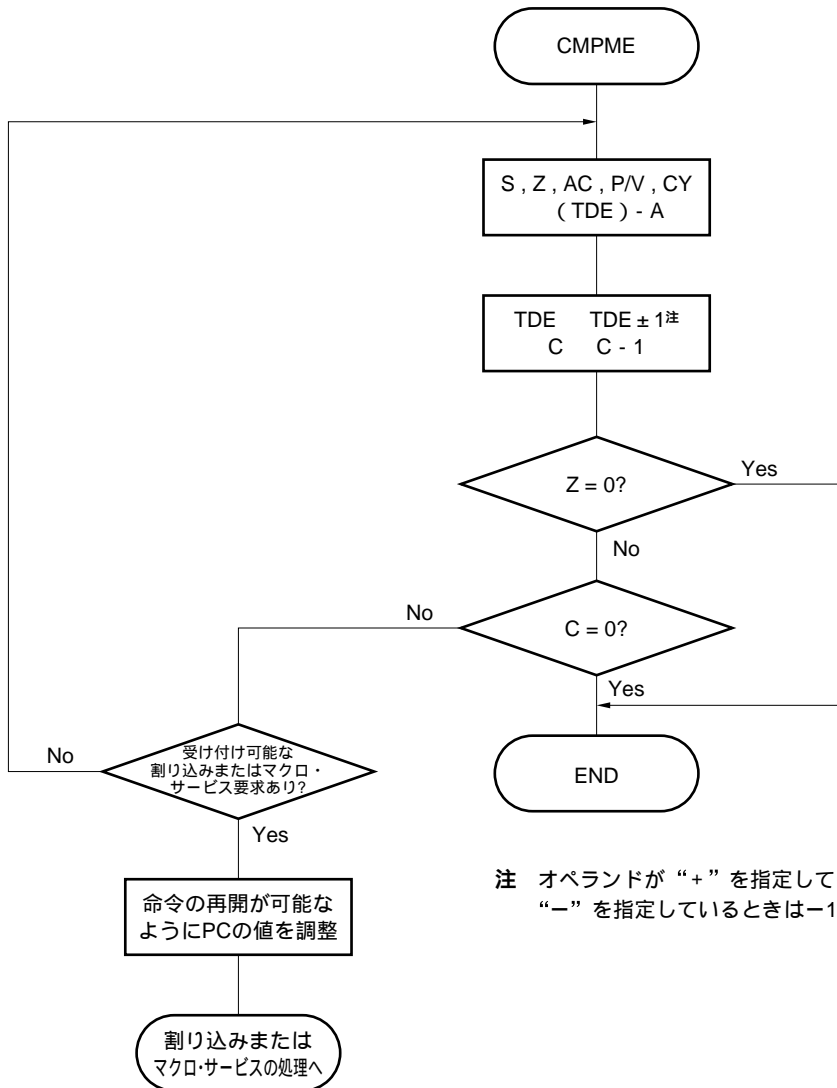
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

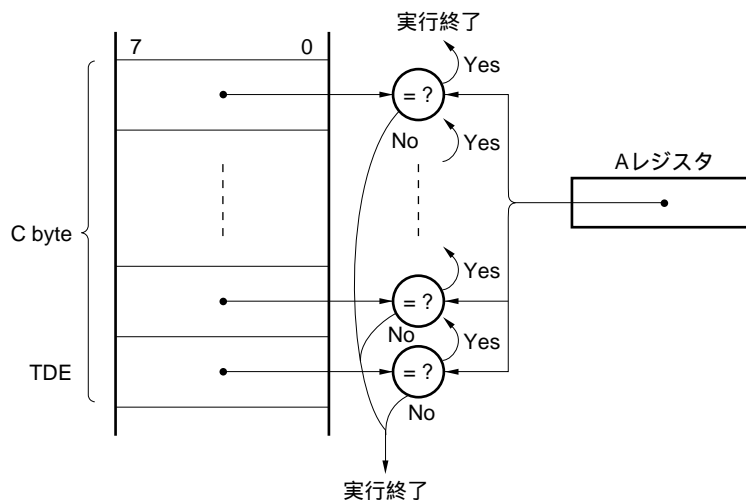
減算の結果、ビット4からビット3へのボローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



## 【記 述 例】

MOV C, #20H

MOVG TDE, #56283H

MOV A, #00H

CMPME [ TDE + ], A ; 56283H番地からの20Hバイトのデータがすべて00Hであるか否かを示す

BNZ \$JMP ; 00Hでないデータがあった場合, JMP番地へ分岐

**CMPMNE**Compare Multiple Not Equal Byte  
固定のバイト・データとのブロック比較（不一致検出）

【命令形式】 **CMPMNE [ TDE + ], A**  
**CMPMNE [ TDE - ], A**

【オペレーション】 (TDE) - A, TDE TDE + 1, C C - 1 End if C = 0 or Z = 1  
(TDE) - A, TDE TDE - 1, C C - 1 End if C = 0 or Z = 1

## 【オペランド】

二モニック	オペランド
CMPMNE	[ TDE + ], A
	[ TDE - ], A

## 【フラグ】

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

## 【説明】

Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDEレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較結果が一致するか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていないければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

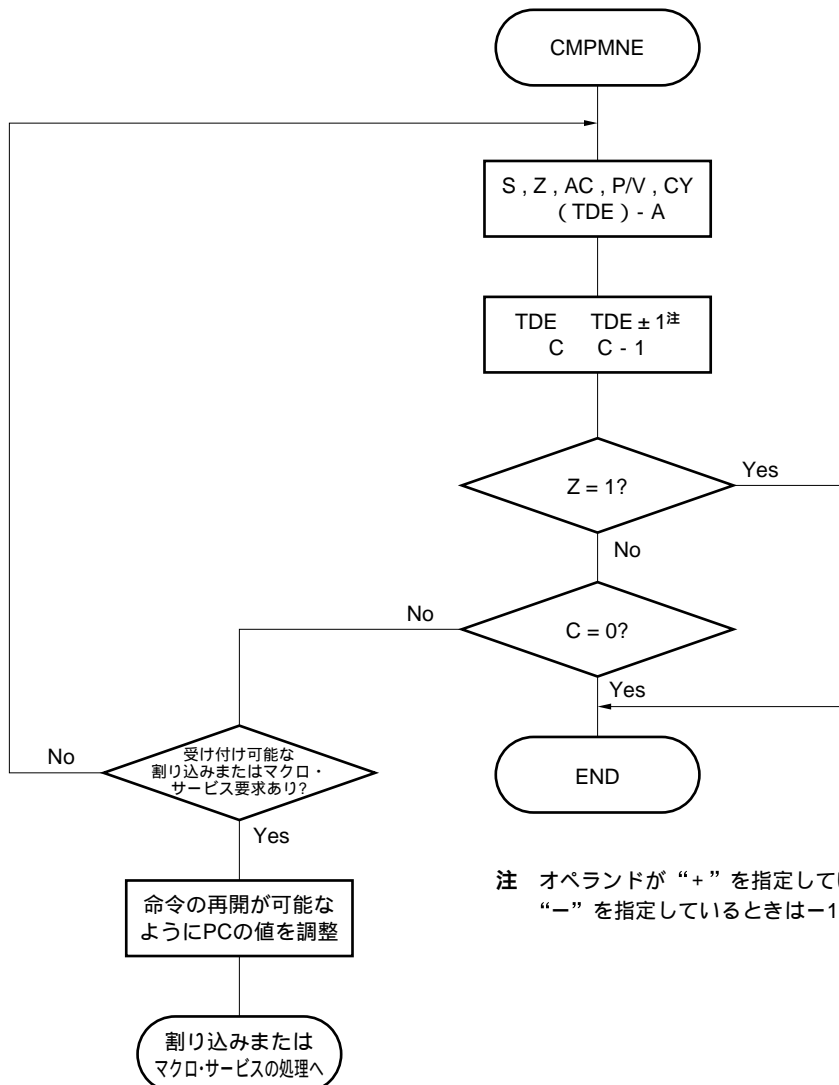
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

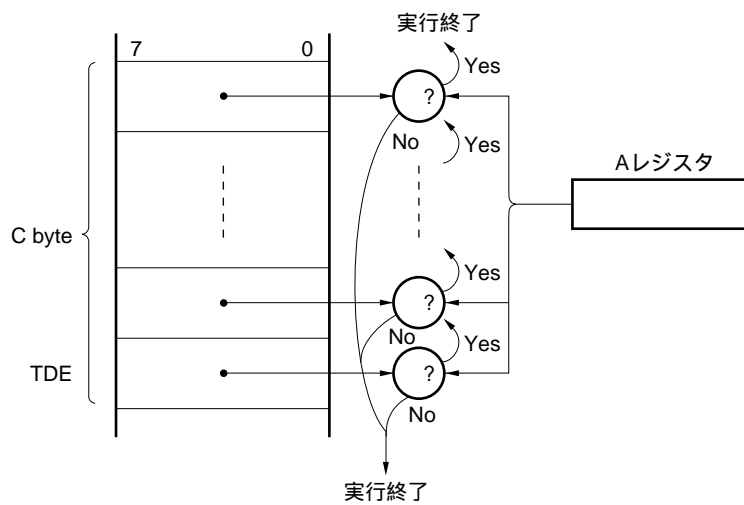
減算の結果、ビット4からビット3へのボローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーパフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



## 【記 述 例】

**MOV C, #00H** ; C 00H

**MOVG TDE, #3000H** ; TDE 3000H

**CMPMNE [ TDE + ], A**

**BZ \$IMP** ; 3000H-30FFHにAレジスタと同じ値があれば、レーベルIMPで示されるアドレスに分岐します。

**CMPMC**Compare Multiple Carry Byte  
固定のバイト・データとのブロック比較（大小比較）

【命令形式】 **CMPMC [ TDE + ], A**  
**CMPMC [ TDE - ], A**

【オペレーション】 (TDE) - A, TDE TDE + 1, C C - 1 End if C = 0 or CY = 0  
(TDE) - A, TDE TDE - 1, C C - 1 End if C = 0 or CY = 0

## 【オペランド】

二モニック	オペランド
CMPMC	[ TDE + ], A
	[ TDE - ], A

## 【フラグ】

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

## 【説明】

Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDEレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較した結果、TDEレジスタでアドレスされるメモリの内容がAレジスタの内容以上の値か、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていなければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

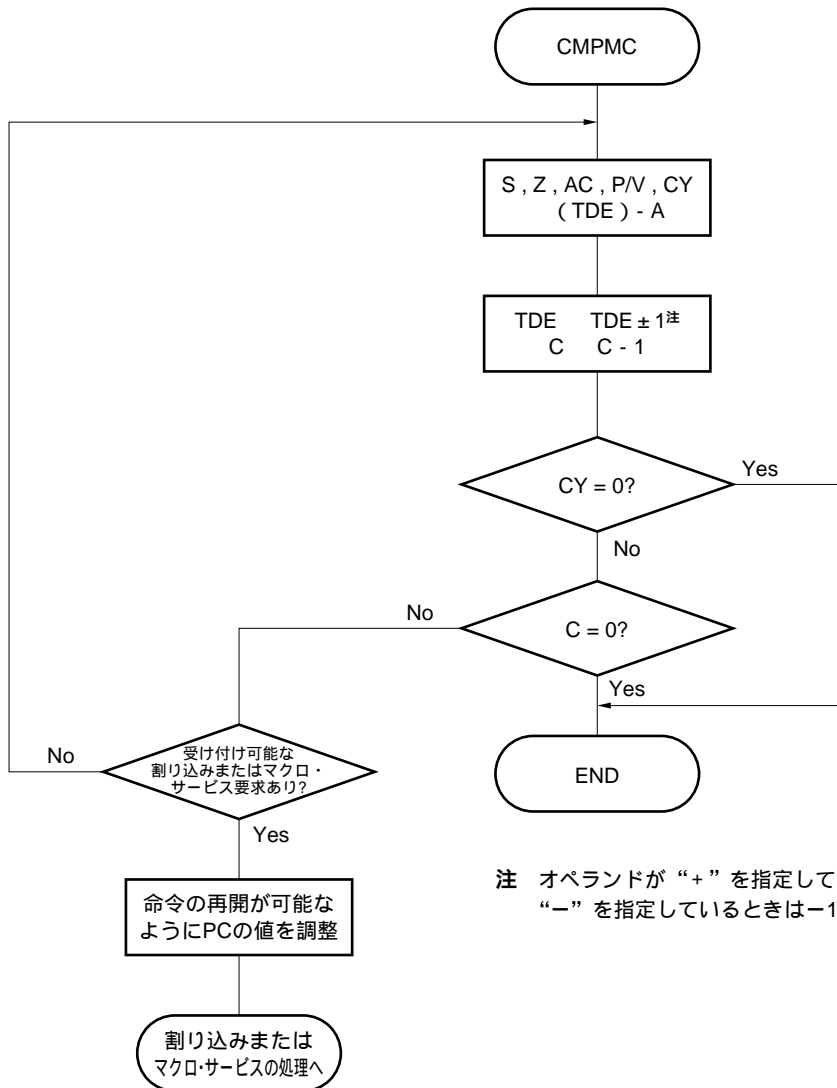
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

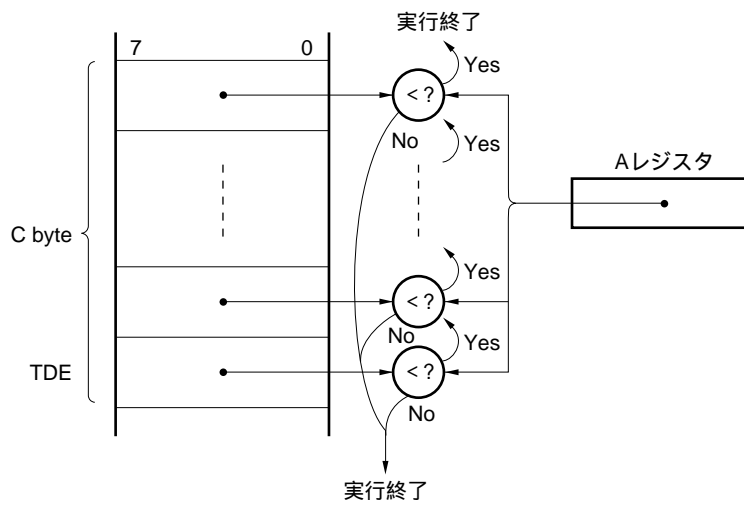
減算の結果、ビット4からビット3へのボローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



**【記 述 例】**

**MOV C, #10H**

**MOV A, #80H**

**MOVG TDE, #567800H**

**CMPMC [ TDE + ], A**

**BNC \$BIG** ; 567800H番地からの10Hバイトのデータ中に, 80H以上のデータがある場合, BIG番地へ分岐

**CMPMNC**

Compare Multiple Not Carry Byte

固定のバイト・データとのブロック比較（大小比較）

【命令形式】 **CMPMNC [ TDE + ], A**  
**CMPMNC [ TDE - ], A**

【オペレーション】 (TDE) - A, TDE TDE + 1, C C - 1 End if C = 0 or CY = 1  
 (TDE) - A, TDE TDE - 1, C C - 1 End if C = 0 or CY = 1

## 【オペランド】

二モニック	オペランド
CMPMNC	[ TDE + ], A
	[ TDE - ], A

## 【フラグ】

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

## 【説明】

Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDEレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較した結果、Aレジスタの内容が大きいか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、Aレジスタの内容とTDEレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, Cの各レジスタの内容が変更されていないければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

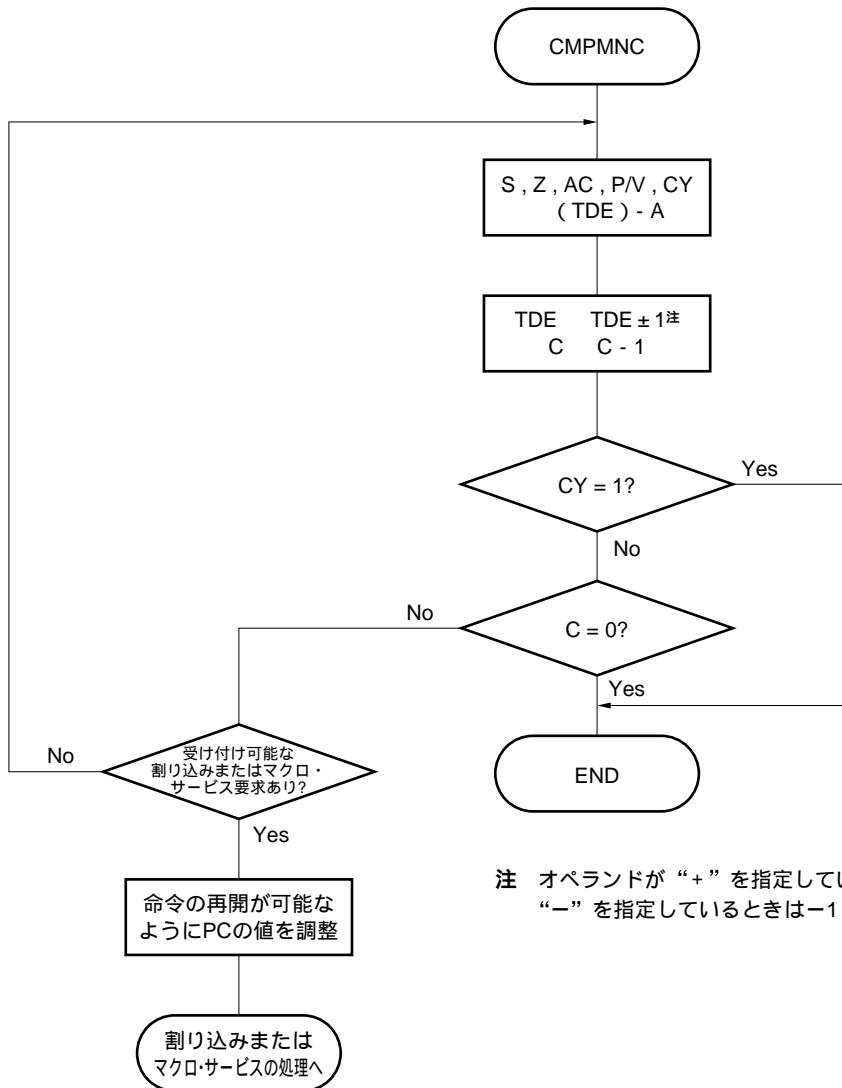
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

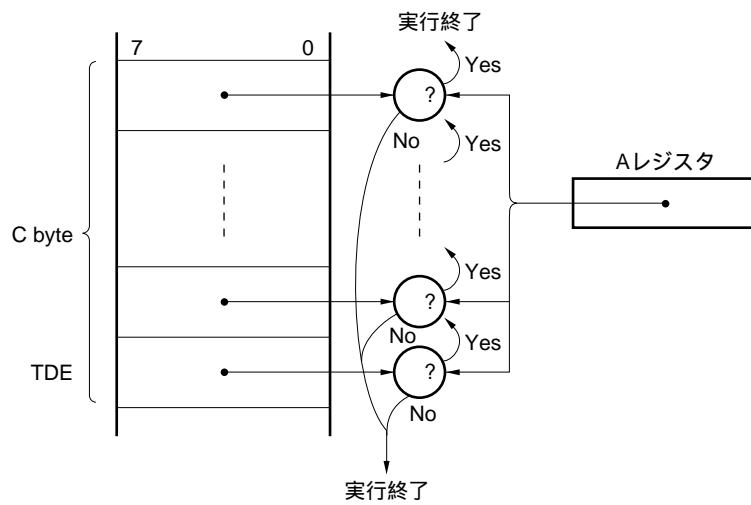
減算の結果、ビット4からビット3へのボローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



## 【記 述 例】

**MOV C, #00H** ; C 00H

**MOVG TDE, #8000H** ; TDE 8000H

**CMPMNC [ TDE + ], A**

**BC \$JMP** ; 8000H-80FFHにAレジスタの内容より大きい値があれば、レーベルJMPで示されるアドレスに分岐

**CMPBKE**Compare Block Equal Byte  
バイト・データとのブロック比較（一致検出）

【命令形式】 **CMPBKE [TDE + ], [WHL + ]**  
**CMPBKE [TDE - ], [WHL - ]**

【オペレーション】 (TDE) - (WHL), TDE TDE + 1, WHL WHL + 1, C C - 1  
End if C = 0 or Z = 0  
(TDE) - (WHL), TDE TDE - 1, WHL WHL - 1, C C - 1  
End if C = 0 or Z = 0

## 【オペランド】

モニタック	オペランド
CMPBKE	[TDE + ], [WHL + ]
	[TDE - ], [WHL - ]

## 【フラグ】

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

## 【説明】

WHLレジスタでアドレスされるメモリの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDE, WHLレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較結果が不一致であるか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、TDE, WHLレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていなければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

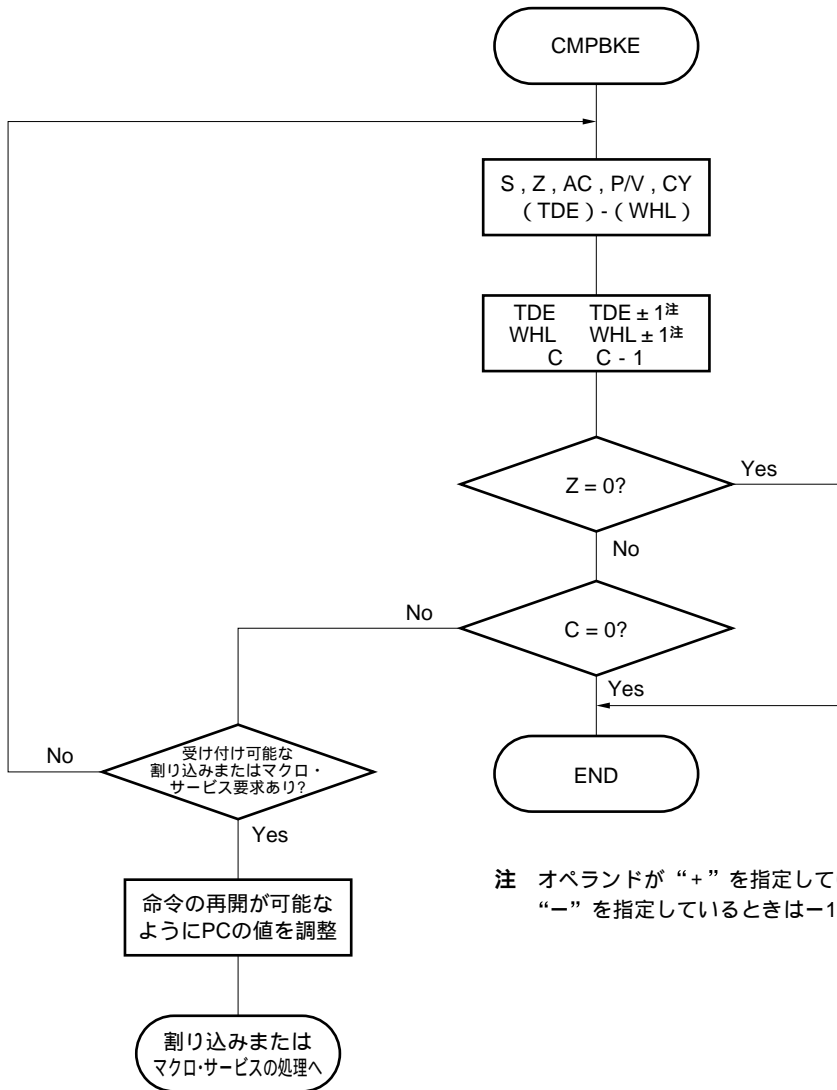
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

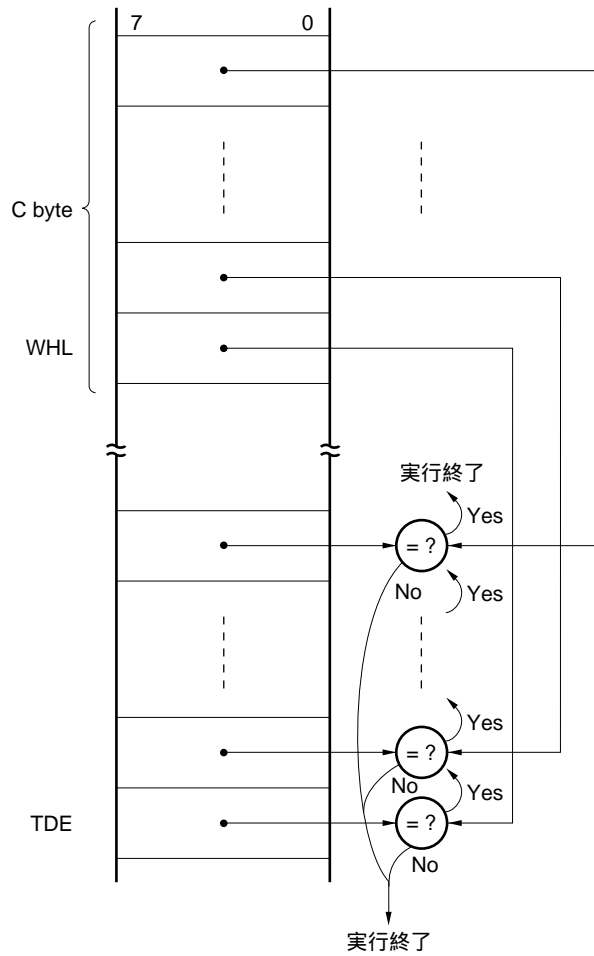
減算の結果、ビット4からビット3へのポローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記 述 例】

MOV C, #40H

MOVG TDE, #342156H

MOVG WHL, #3421AAH

CMPBKE [ TDE + ], [ WHL + ]

BNE \$DIFF

; 342156H番地からの40Hバイトのデータと3421AAH番地からのデータを比較し、異なるデータがあれば、DIFF番地へ分岐

**CMPBKNE**Compare Block Not Equal Byte  
バイト・データとのブロック比較（不一致検出）

【命令形式】 **CMPBKNE [ TDE + ], [ WHL + ]**  
**CMPBKNE [ TDE - ], [ WHL - ]**

【オペレーション】 ( TDE ) - ( WHL ), TDE TDE + 1, WHL WHL + 1, C C - 1  
End if C = 0 or Z = 1  
( TDE ) - ( WHL ), TDE TDE - 1, WHL WHL - 1, C C - 1  
End if C = 0 or Z = 1

## 【オペランド】

二モニック	オペランド
CMPBKNE	[ TDE + ], [ WHL + ]
	[ TDE - ], [ WHL - ]

## 【フラグ】

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

## 【説明】

WHLレジスタでアドレスされるメモリの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDE, WHLレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較結果が一致するか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、TDE, WHLレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていなければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

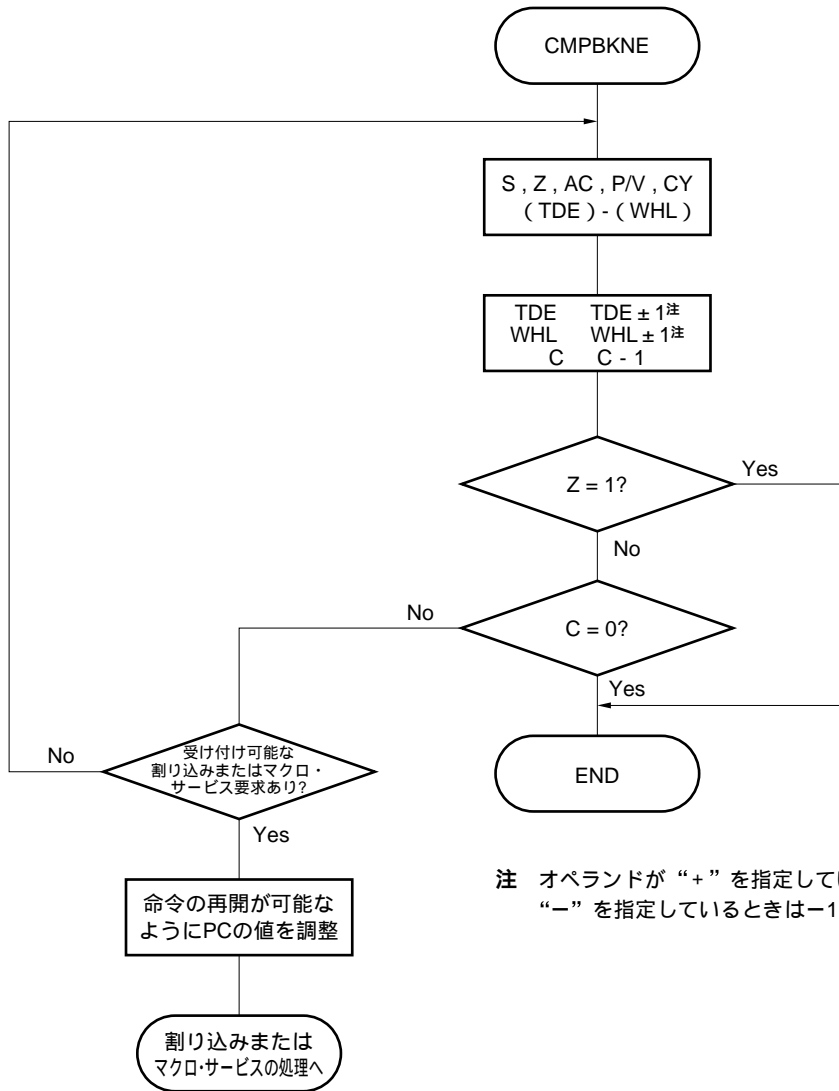
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

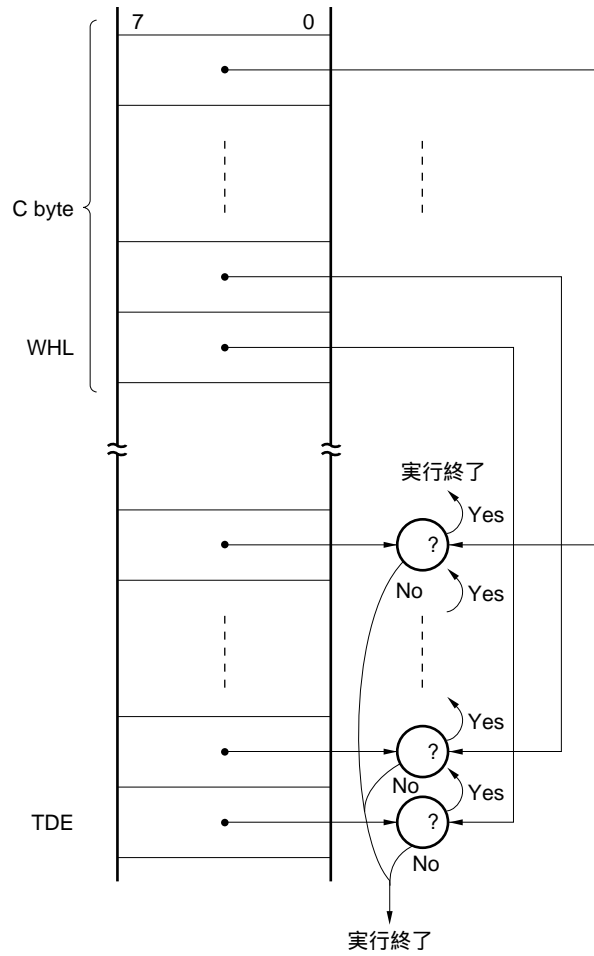
減算の結果、ビット4からビット3へのポローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記 述 例】

MOV C, #5H

MOVG TDE, #0FFC50H

MOVG WHL, #0FC50H

CMPBKNE [ TDE + ], [ WHL + ]

BE \$FIND

; 0FFC50H番地からの5バイトのデータと0FC50H番地からのデータを比較し、一致するデータがあれば、FIND番地へ分岐

**CMPBKC**Compare Block Carry Byte  
バイト・データのブロック比較（大小検出）

【命令形式】 **CMPBKC** [ TDE + ], [ WHL + ]  
**CMPBKC** [ TDE - ], [ WHL - ]

【オペレーション】 ( TDE ) - ( WHL ), TDE TDE + 1, WHL WHL + 1, C C - 1  
 End if C = 0 or CY = 0  
 ( TDE ) - ( WHL ), TDE TDE - 1, WHL WHL - 1, C C - 1  
 End if C = 0 or CY = 0

## 【オペランド】

二モニック	オペランド
<b>CMPBKC</b>	[ TDE + ], [ WHL + ]
	[ TDE - ], [ WHL - ]

## 【フラグ】

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

## 【説明】

WHLレジスタでアドレスされるメモリの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDE, WHLレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較した結果、TDEレジスタでアドレスされるメモリの内容がWHLレジスタでアドレスされるメモリの内容以上か、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、TDE, WHLレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていなければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

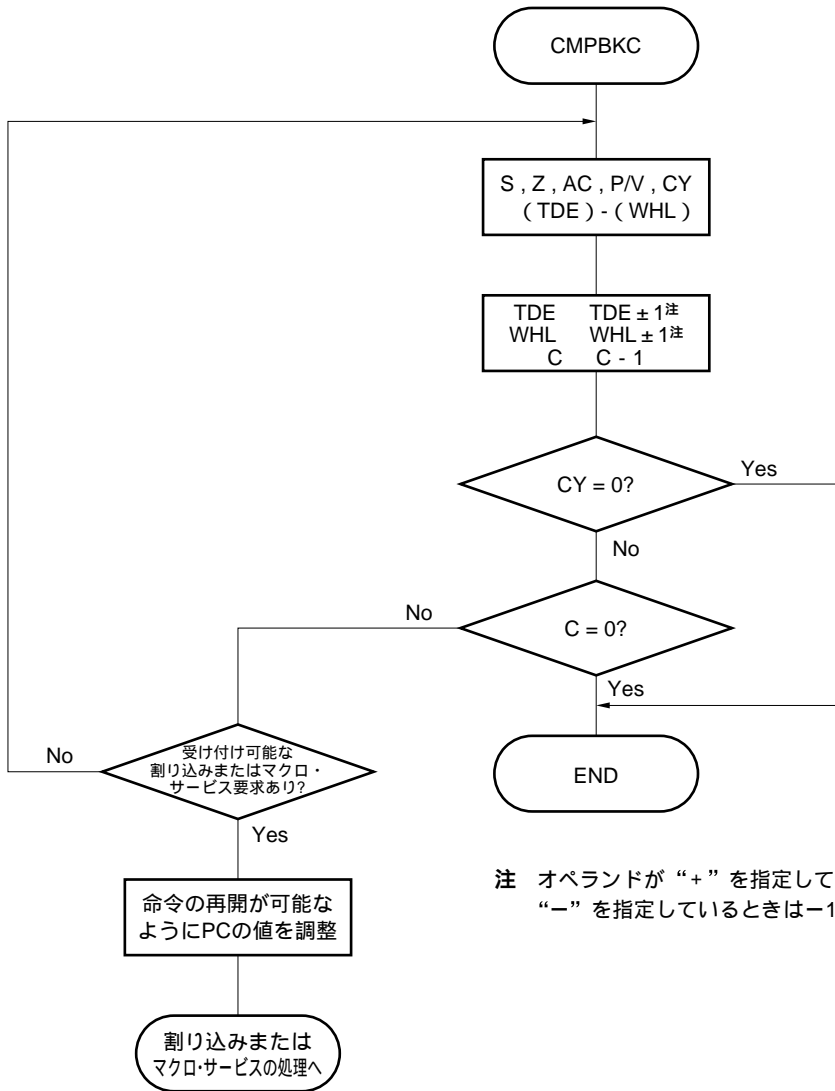
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

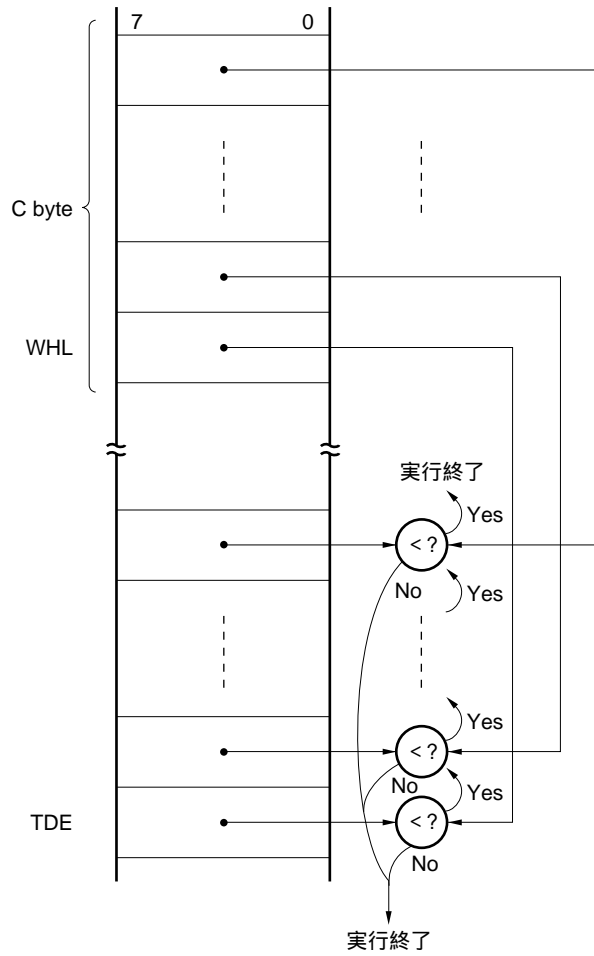
減算の結果、ビット4からビット3へのポローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記述例】

MOV C, #3H

MOVG TDE, #0E8762H

MOVG WHL, #03502H

CMPBKC [ TDE - ], [ WHL - ]

BNC \$BIG

; 0E8760H番地からの3バイト・データと03500H番地からの3バイト・データを比較し、その結果が同じか、または0E8760H番地からの3バイト・データの方が大きいとき、BIG番地へ分岐

**CMPBKNC**Compare Block Not Carry Byte  
固定のバイト・データのブロック比較（大小比較）

【命令形式】 **CMPBKNC [ TDE + ], [ WHL + ]**  
**CMPBKNC [ TDE - ], [ WHL - ]**

【オペレーション】 ( TDE ) - ( WHL ), TDE TDE + 1, WHL WHL + 1, C C - 1  
End if C = 0 or CY = 1  
( TDE ) - ( WHL ), TDE TDE - 1, WHL WHL - 1, C C - 1  
End if C = 0 or CY = 1

## 【オペランド】

モニック	オペランド
<b>CMPBKNC</b>	[ TDE + ], [ WHL + ]
	[ TDE - ], [ WHL - ]

## 【フラグ】

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

## 【説明】

WHLレジスタでアドレスされるメモリの内容とTDEレジスタでアドレスされるメモリの内容とを比較し、TDE, WHLレジスタの内容をインクリメント/デクリメントし、Cレジスタの内容をデクリメントします。比較した結果、WHLレジスタでアドレスされるメモリの内容が大きいか、またはCレジスタの内容が0になるまで以上の動作を繰り返します。

命令の実行により、TDE, WHLレジスタでアドレスされるメモリの内容は変化しません。

この命令の実行中に、受け付け可能な割り込みまたはマクロ・サービスの要求が発生した場合、この命令の実行を中断して割り込みまたはマクロ・サービスを受け付けます。割り込みを受け付けた場合、割り込み処理プログラム中でスタックまたはRP2, R7に退避された戻りアドレスおよび、この命令で使用しているTDE, WHL, Cの各レジスタの内容が変更されていなければ、割り込みからの復帰により、中断したこの命令の実行を再開します。マクロ・サービスを受け付けた場合、マクロ・サービスの終了後、この命令の実行を再開します。

S, Z, AC, P/V, CYの各フラグは、この命令によって最後に実行した比較動作（減算）にしたがって変化します。

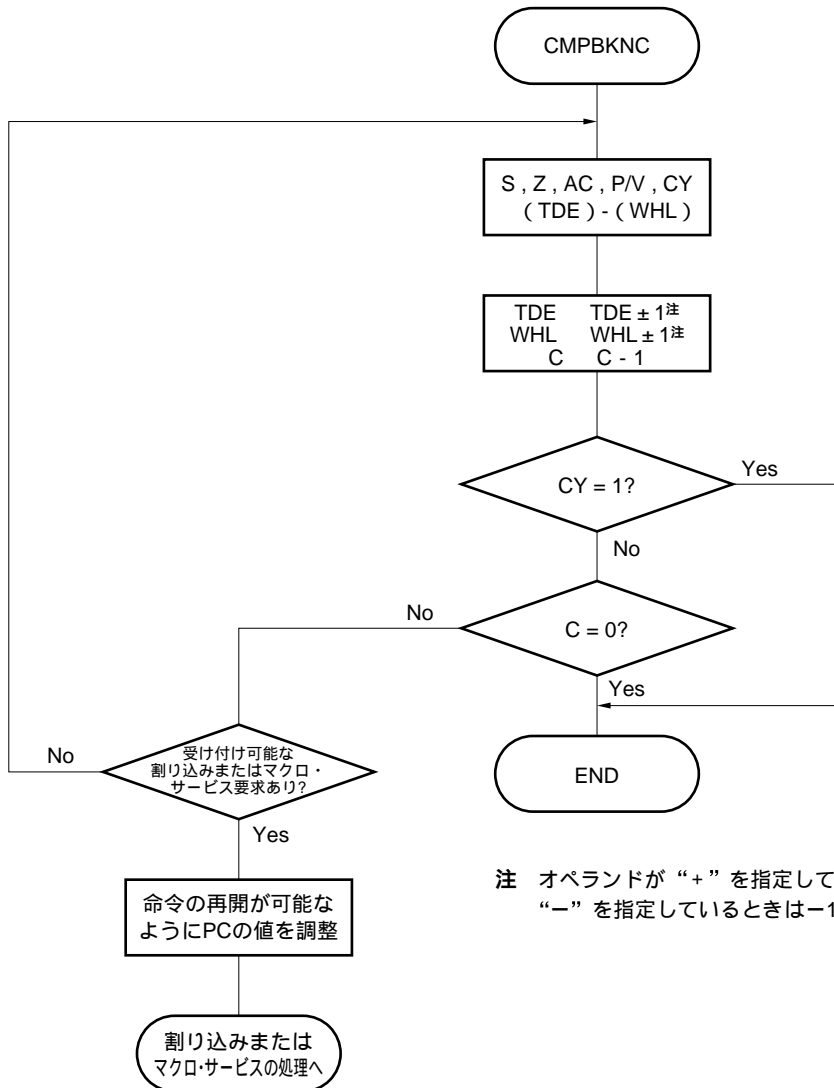
減算の結果、ビット7がセット（1）された場合は、Sフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、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。

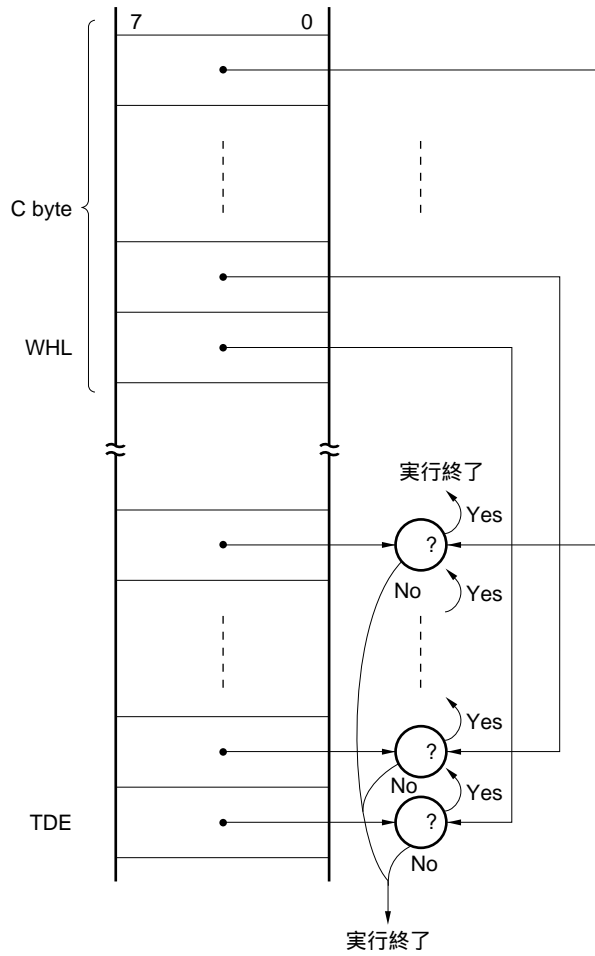
減算の結果、ビット4からビット3へのポローが発生した場合、ACフラグはセット（1）、その他の場合はクリア（0）されます。

減算の結果、ビット6でボローが発生し、かつ、ビット7でボローが発生しない場合（2の補数形式での演算によるアンダフロー発生時）、または、ビット6でボローが発生せず、かつ、ビット7でボローが発生した場合（2の補数形式での演算によるオーバフロー発生時）、P/Vフラグはセット（1）され、その他の場合はクリア（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。



注 オペランドが“+”を指定しているときは+1，“-”を指定しているときは-1



【記 述 例】

MOV C, #4H

MOVG TDE, #05503H

MOVG WHL, #0FFC03H

CMPBKNC [ TDE - ], [ WHL - ]

BC \$LITTLE

; 05500H番地からの4バイト・データと0FFC00H番地からのデータを比較し, 05500H番地からのデータの方が小さいとき, LITTLE番地へ分岐

## 第8章 開発ツール

78K/ シリーズの製品を使用するために必要なツールの概要を示します。

詳細については、各デバイスの**ユーザーズ・マニュアル ハードウェア編**または**開発ツール セレクション・ガイド (U11069J)**を参照してください。

## 8.1 開発ツール

応用システムのプログラム開発のために次の開発ツールを用意しています。

表8-1 開発ツールの種類と機能 (1/2)

開発ツール	機能	
ハードウェア	インサーキット・エミュレータ (IE-784000-R) (IE-78K4-NS)	78K/ シリーズの応用システム開発において、ハードウェアやプログラムのデバッグ用として使用する開発ツールです。 パーソナル・コンピュータ (PC-9800シリーズ, IBM PC/AT™)などをホスト・マシンとして使用した場合、シンボリック・デバッグおよびオブジェクト・ファイル、シンボル・ファイル転送などの各種機能によって効率の良いデバッグが可能です。 RS-232-Cのシリアル・インタフェースを内蔵しており、PROMプログラマ (PG-1500) などとの接続も可能です。
	エミュレーション・ボード (IE-78 x x x -R-EM) (IE-78 x x x -R-EM-A) (IE-78 x x x -R-EM)	対象デバイスごとに異なる周辺ハードウェアをエミュレートするためのボードです。通常は、IE本体に組み込まれています。
	I/Oエミュレーション・ボード (IE-78 x x x -R-EM1) (IE-78 x x x -R-EM1) (IE-78 x x x -NS-EM1)	対象デバイスごとに異なる周辺ハードウェアをエミュレートするためのボードです。エミュレーション・ボードと組み合わせて使用します。I/Oエミュレーション・ボードが必要となる対象デバイスは製品により異なります。
	エミュレーション・プローブ (EP-78 x x x .....) (NP- x x x .....)	IE本体と応用システムを接続するプローブです。各対象デバイスのパッケージごとに用意しています。
	変換ソケット (EV-9200 x x - x x)	QFP用のエミュレーション・プローブを応用システムに接続する際に使用するソケットです。QFP用のエミュレーション・プローブに標準添付しています。応用システムの基板に実装して使用します。
	プログラマ・アダプタ (PA-78P x x x .....)	PROM内蔵品のプログラマ時に使用するPROMプログラマ (PG-1500) 用のアダプタです。 対象デバイスのパッケージごとに用意しています。
	治具 (EV-9900)	WQFNパッケージの製品をEV-9200 x x - x x から外すときに使用する治具です。

表8-1 開発ツールの種類と機能 (2/2)

	開発ツール	機能
ソフトウェア	統合ディバッガ (ID78K4)	78K/ シリーズ用のインサーキット・エミュレータをコントロールするためのプログラムです。デバイス・ファイルと組み合わせて使用します。従来品のIEコントローラと比較して、C言語、構造化アセンブリ言語、アセンブリ言語で書かれたソース・プログラム・レベルでのディバグや、ホスト・マシンの画面を分割し、さまざまな情報を同時に表示することで効率の良いディバグができます。
	デバイス・ファイル	統合ディバッガと組み合わせて使用します。78K/ シリーズのディバグを行う場合に必要です。
	インサーキット・エミュレータ・コントロール・プログラム (IEコントローラ)	IEとホスト・マシンを接続して効率よくディバグを行うためのソフトウェアです。各ファイル(オブジェクト、シンボル)の転送や、オンライン・アSEMBル、逆アSEMBル、ブレーク条件(イベント)設定などIEの持つ機能を十分に発揮させます。
	リロケータブル・アセンブラ <sup>注1</sup>	ニモニックで書かれたプログラムをマイコンの実行可能なオブジェクト・コードに変換するプログラムです。 このほかに、シンボル・テーブルの生成、分岐命令の最適化処理などを自動的に行う機能を備えています。
	構造化アセンブラ・プリプロセッサ	アセンブラに構造化プログラミング手法を導入したものです。 アセンブラの持つスピードとサイズを犠牲にすることなく、C言語風の制御構造の機能を記述できます。
	Cコンパイラ <sup>注1</sup>	高級言語のCで書かれたプログラムをマイコンの実行可能なオブジェクト・コードに変換するプログラムです。
	Cライブラリ・ソース	Cコンパイラに付属しているライブラリのソース・プログラムです。ライブラリを改良する(よりユーザの仕様にあわせる)場合に必要となります。
	システム・シミュレータ (SM78K4) <sup>注2</sup>	78K/ シリーズ用のソフトウェア開発サポート・ツールです。ホスト・マシン上で、ターゲット・システムの動作をシミュレーションしながら、Cソース・レベルまたはアセンブラ・レベルでディバグが可能です。SM78K4を使用することにより、アプリケーションの論理検証、性能検証をハードウェア開発から独立して行うことができ、開発効率やソフトウェア品質の向上が図れます。

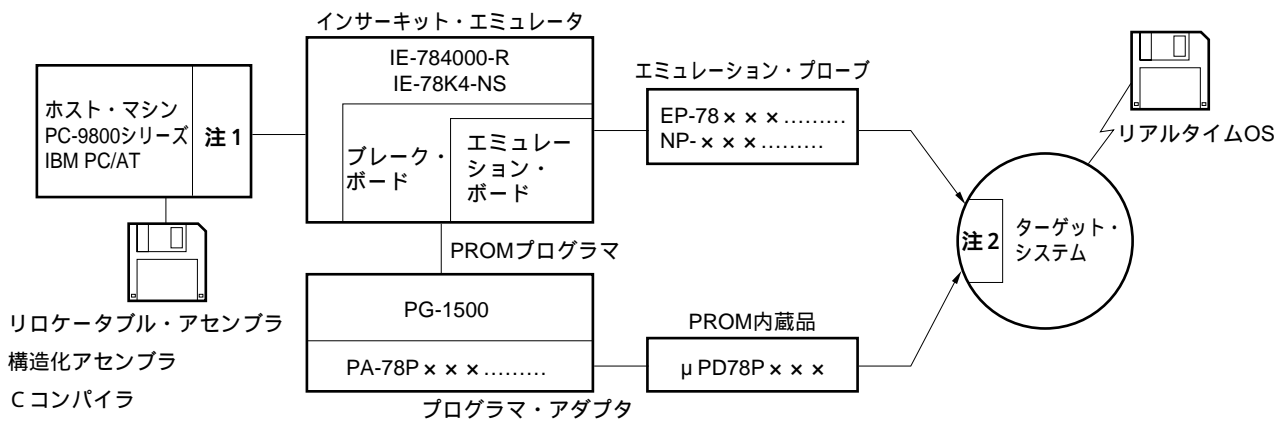
注1. 78K/ シリーズは、デバイス・ファイルと組み合わせて使用します。

2. デバイス・ファイルと組み合わせて使用します。

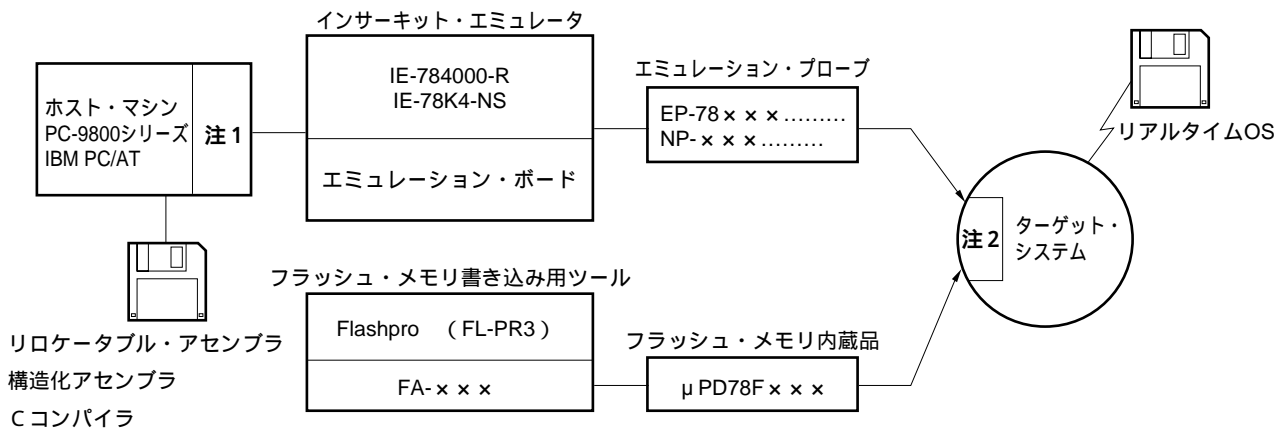
備考 ソフトウェアはすべてMS-DOS<sup>TM</sup>、PC DOS<sup>TM</sup>の各OSをサポートします。

図8-1 開発ツールの構成

(PROM内蔵品の場合)



(フラッシュ・メモリ内蔵品の場合)



注1. 統合ディバッガ, デバイス・ファイル

2. エミュレーション・プローブをターゲット・システムに接続するための変換ソケット (製品名がEV-9200ではじまる製品)

備考 各製品名の記号は, 次のとおりです。

- IE : インサーキット・エミュレータ
- EP : エミュレーション・プローブ
- NP : エミュレーション・プローブ (株内藤電誠町田製作所製)
- PA : PROMプログラマ・アダプタ
- FA : フラッシュ・メモリ書き込み用アダプタ
- xxx... : 対象デバイスやパッケージにより異なる品名

## 8.2 PROM書き込み用ツール

### (1) ハードウェア

PG-1500	付属ボードおよび別売のPROMプログラマ・アダプタを接続することにより、PROM内蔵のシングルチップ・マイクロコンピュータを、スタンド・アロンまたはホスト・マシンからの操作によりプログラミングできるPROMプログラマです。 また、256Kビットから4Mビットまでの代表的なPROMをプログラミングすることもできます。
PROMプログラマ・アダプタ	PROMを内蔵する製品ごとに用意しているアダプタです。 PROMプログラマと組み合わせて使用します。 具体的な品名については、各デバイスのユーザーズ・マニュアル ハードウェア編を参照してください。

### (2) ソフトウェア

PG-1500コントローラ	PG-1500とホスト・マシンをシリアルおよびパラレル・インタフェースで接続し、ホスト・マシン上でPG-1500を制御します。
---------------	---

## 8.3 フラッシュ・メモリ書き込み用ツール

Flashpro (型番FL-PR3) フラッシュ・ライター	フラッシュ・メモリ内蔵マイコン専用フラッシュ・ライターです。
フラッシュ・メモリ書き込み 用アダプタ	対象製品にあわせて結線が必要です。 具体的な品名については、各デバイスのユーザーズ・マニュアル ハードウェア編を参照してください。

**備考** 株式会社内藤電誠町田製作所（（044）822-3813）の製品です。ご購入の際は、NEC特約店にご相談ください。

[メモ]

## 第9章 組み込み用ソフトウェア

### 9.1 リアルタイムOS

RX78K/ <sup>注</sup> リアルタイムOS	RX78K/ は、リアルタイム性の要求される制御分野を対象として、マルチタスク環境を実現することを目的としています。CPUのアイドル時間を他の処理に割り当て、システム全体としての性能の向上を図ることができます。 RX78K/ では、 $\mu$ ITRON仕様に準拠したシステム・コール(31個)を提供しています。また、RX78K/ のニュークリアスと複数の情報テーブルを作成するためのツール(コンフィギュレータ)を提供しています。 別売のアセンブラ・パッケージ(RA78K4)およびデバイス・ファイルと組み合わせて使用します。 <b>PC環境で使用する場合の注意</b> リアルタイムOSはDOSベースのアプリケーションです。Windows上ではDOSプロンプトで使用してください。
MX78K4 OS	$\mu$ ITRON仕様サブセットのOSです。MX78K4のニュークリアスを添付しています。タスク処理、イベント管理、時間管理を行います。タスク管理ではタスクの実行順序を制御し、次に実行するタスクへの切り替え処理を行います。 <b>PC環境で使用する場合の注意</b> MX78K4はDOSベースのアプリケーションです。Windows上ではDOSプロンプトで使用してください。

注 RX78K/ を購入する場合、事前に購入申込書にご記入のうえ、使用許諾契約書を締結してください。

〔メモ〕

## 付録A 命令索引 (二モニック : 機能別)

### 【8ビット・データ転送命令】

MOV ... 290

### 【16ビット・データ転送命令】

MOVW ... 293

### 【24ビット・データ転送命令】

MOVG ... 296

### 【8ビット・データ交換命令】

XCH ... 298

### 【16ビット・データ交換命令】

XCHW ... 300

### 【8ビット演算命令】

ADD ... 302

ADDC ... 304

SUB ... 306

SUBC ... 308

CMP ... 310

AND ... 312

OR ... 313

XOR ... 314

### 【16ビット演算命令】

ADDW ... 316

SUBW ... 318

CMPW ... 320

### 【24ビット演算命令】

ADDG ... 323

SUBG ... 324

### 【乗除算命令】

MULU ... 326

MULUW ... 327

MULW ... 328

DIVUW ... 329

DIVUX ... 330

### 【特殊演算命令】

MACW ... 332

MACSW ... 335

SACW ... 338

### 【増減命令】

INC ... 342

DEC ... 343

INCW ... 344

DECW ... 345

INCG ... 346

DECG ... 347

### 【補正命令】

ADJBA ... 349

ADJBS ... 350

CVTBW ... 351

### 【シフト・ローテート命令】

ROR ... 353

ROL ... 354

RORC ... 355  
 ROLC ... 356  
 SHR ... 357  
 SHL ... 358  
 SHRW ... 359  
 SHLW ... 360  
 ROR4 ... 361  
 ROL4 ... 362

**【ビット操作命令】**

MOV1 ... 364  
 AND1 ... 366  
 OR1 ... 368  
 XOR1 ... 370  
 NOT1 ... 371  
 SET1 ... 372  
 CLR1 ... 373

**【スタック操作命令】**

PUSH ... 375  
 PUSHU ... 377  
 POP ... 378  
 POPU ... 380  
 MOVG ... 381  
 ADDWG ... 382  
 SUBWG ... 383  
 INCG SP ... 384  
 DECG SP ... 385

**【コール・リターン命令】**

CALL ... 387  
 CALLF ... 389  
 CALLT ... 390  
 BRK ... 391  
 BRKCS ... 392  
 RET ... 394  
 RETI ... 395  
 RETB ... 396  
 RETCS ... 397  
 RETCSB ... 399

**【無条件分岐命令】**

BR ... 402

**【条件付き分岐命令】**

BNZ ... 404  
 BNE ... 404  
 BZ ... 405  
 BE ... 405  
 BNC ... 406  
 BNL ... 406  
 BC ... 407  
 BL ... 407  
 BNV ... 408  
 BPO ... 408  
 BV ... 409  
 BPE ... 409  
 BP ... 410  
 BN ... 411  
 BLT ... 412  
 BGE ... 413  
 BLE ... 414  
 BGT ... 415  
 BNH ... 416  
 BH ... 417  
 BF ... 418  
 BT ... 419  
 BTCLR ... 420  
 BFSET ... 421  
 DBNZ ... 422

**【CPU制御命令】**

MOV STBC, #byte ... 424  
 MOV WDM, #byte ... 425  
 LOCATION ... 426  
 SEL RBn ... 427  
 SEL RBn, ALT ... 428  
 SWRS ... 429  
 NOP ... 430  
 EI ... 431  
 DI ... 432

**【特殊命令】**

CHKL ... 434

CHKLA ... 436

**【ストリング命令】**

MOVTBLW ... 439

MOVM ... 441

XCHM ... 443

MOVBK ... 445

XCHBK ... 447

CMPME ... 449

CMPMNE ... 452

CMPMC ... 455

CMPMNC ... 458

CMPBKE ... 461

CMPBKNE ... 464

CMPBKC ... 467

CMPBKNC ... 470

[メモ]

## 付録B 命令索引 (ニモニク : アルファベット順)

### 【A】

ADD ... 302  
ADDC ... 304  
ADDG ... 323  
ADDW ... 316  
ADDWG ... 382  
ADJBA ... 349  
ADJBS ... 350  
AND ... 312  
AND1 ... 366

### 【B】

BC ... 407  
BE ... 405  
BF ... 418  
BFSET ... 421  
BGE ... 413  
BGT ... 415  
BH ... 417  
BL ... 407  
BLE ... 414  
BLT ... 412  
BN ... 411  
BNC ... 406  
BNE ... 404  
BNH ... 416  
BNL ... 406  
BNV ... 408  
BNZ ... 404  
BP ... 410  
BPE ... 409  
BPO ... 408  
BR ... 402  
BRK ... 391  
BRKCS ... 392  
BT ... 419

BTCLR ... 420  
BV ... 409  
BZ ... 405

### 【C】

CALL ... 387  
CALLF ... 389  
CALLT ... 390  
CHKL ... 434  
CHKLA ... 436  
CLR1 ... 373  
CMP ... 310  
CMPBKC ... 467  
CMPBKE ... 461  
CMPBKNC ... 470  
CMPBKNE ... 464  
CMPMC ... 455  
CMPME ... 449  
CMPMNC ... 458  
CMPMNE ... 452  
CMPW ... 320  
CVTBW ... 351

### 【D】

DBNZ ... 422  
DEC ... 343  
DECG ... 347  
DECG SP ... 385  
DECW ... 345  
DI ... 432  
DIVUW ... 329  
DIVUX ... 330

### 【E】

EI ... 431

**【I】**

INC ... 342  
 INCG ... 346  
 INCG SP ... 384  
 INCW ... 344

**【L】**

LOCATION ... 426

**【M】**

MACSW ... 335  
 MACW ... 332  
 MOV ... 290  
 MOVBK ... 445  
 MOVG ... 296, 381  
 MOVN ... 441  
 MOV STBC, # byte ... 424  
 MOVTBLW ... 439  
 MOVW ... 293  
 MOV WDM, # byte ... 425  
 MOV1 ... 364  
 MULU ... 326  
 MULUW ... 327  
 MULW ... 328

**【N】**

NOP ... 430  
 NOT1 ... 371

**【O】**

OR ... 313  
 OR1 ... 368

**【P】**

POP ... 378  
 POPU ... 380  
 PUSH ... 375

PUSHU ... 377

**【R】**

ROL ... 354  
 ROLC ... 356  
 ROL4 ... 362  
 ROR ... 353  
 RORC ... 355  
 ROR4 ... 361  
 RET ... 394  
 RETB ... 396  
 RETCS ... 397  
 RETCSB ... 399  
 RETI ... 395

**【S】**

SACW ... 338  
 SEL RBn ... 427  
 SEL RBn, ALT ... 428  
 SET1 ... 372  
 SHL ... 358  
 SHLW ... 360  
 SHR ... 357  
 SHRW ... 359  
 SUB ... 306  
 SUBC ... 308  
 SUBG ... 324  
 SUBW ... 318  
 SUBWG ... 383  
 SWRS ... 429

**【X】**

XCH ... 298  
 XCHBK ... 447  
 XCHM ... 443  
 XCHW ... 300  
 XOR ... 314  
 XOR1 ... 370

## 付録C 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。

( 1/3 )

版 数	前版からの主な改版内容	適用箇所
第2版	ビット操作目例に次の命令を追加 MOV1 CY, !addr16.bit CY, !!addr24.bit !addr16.bit, CY !!addr24.bit, CY AND1, OR1 CY, !addr16.bit CY, !!addr24.bit CY,!addr16.bit CY,!addr24.bit XOR1 CY, !addr16.bit CY, !!addr24.bit NOT1, SET1, CLR1 !addr16.bit !!addr24.bit 条件付き分岐命令に次の命令を追加 BF, BT, BFSET, BTCLR !addr16.bit, \$ addr20 !!addr24.bit, \$ addr20	第6章 命令セット
第3版	μ PD784915サブシリーズに関する記述を追加 μ PD784026サブシリーズにμ PD784020を追加	全般
	5.2.10 ショート・ダイレクト24ビット・メモリ・インダイレクト・アドレッシングの表現形式を次のように訂正 [ %saddrp ] [ %saddrg ]	第5章 アドレッシング
	6.1 凡例の(1)オペランドと表現形式と記述方法にsaddrg1, saddrg2を追加 MOVGのオペランドを次のように訂正 [ TDE + HL ], WHL [ TDE + C ], WHL 6.5 命令のクロックを追加	第6章 命令セット
	IBM PC/ATの供給媒体に3.5インチ2HCまたは3.5インチ2HDを追加 統合ディバッガのオーダ名称を次のように変更 μ S5A10ID78K4 μ SAA10ID78K4 μ S5A13ID78K4 μ SAA13ID78K4 μ S7B10ID78K4 μ SBB10ID78K4	第8章 開発ツール

版数	前版からの主な改版内容	適用箇所
第4版	<p>μ PD784021に、GKパッケージ（80ピン・プラスチックTQFP、ファインピッチ、12mm）を追加</p> <p>μ PD784038, 784038Yサブシリーズに関する記述を追加</p> <p>μ PD784046サブシリーズに関する記述を追加</p> <p>μ PD784208, 784208Yサブシリーズに関する記述を追加</p> <p>RETCS命令に、μ PD784208, 784208Yサブシリーズにはありませんという注を追加</p>	全般
	フラッシュ・メモリに関する記述を追加	第8章 開発ツール
	MX78K4に関する記述を追加	第9章 組み込み用ソフトウェア
第5版	<p>μ PD784038/Yサブシリーズに新製品（μ PD784031/Y）およびパッケージ（80ピン・プラスチックQFP（14mm, 1.4mm厚））を追加</p> <p>μ PD784046サブシリーズに新製品（μ PD784054）に関する記述を追加</p> <p>μ PD784208サブシリーズに関する記述を追加</p> <p>μ PD784216/Yサブシリーズに関する記述を追加</p> <p>μ PD784915サブシリーズに新製品（μ PD784915A, 784916A）に関する記述を追加</p> <p>μ PD784908サブシリーズに関する記述を追加</p> <p>μ PD78F4943サブシリーズに関する記述を追加</p>	全般
	RETCS命令に、μ PD784208, 784208Yサブシリーズにはありませんという注を削除	第6章 命令セット
	<p>RETI命令の「最上位の/最も上位側にある」という記述を、「優先順位が一番高い」という記述に変更</p> <p>RETCS命令のμ PD784208, 784208Yサブシリーズにはありませんという注を削除、命令形式にtargetを追加、「最上位の/もっと上位側にある」という記述を「優先順位が一番高い」という記述に変更</p> <p>RETCSB命令の命令形式にtargetを追加</p>	第7章 命令の説明
	フラッシュ・メモリに関する記述を修正	第8章 開発ツール
第6版	<p>μ PD784218, 784218Y, 784225, 784225Y, 784928, 784928Yサブシリーズおよびμ PD784943を追加</p> <p>次の製品が開発中 開発済み</p> <p>μ PD784037, 784038, 78P4038</p> <p>μ PD784031Y, 784035Y, 784036Y, 784037Y, 784038Y, 78P4038Y</p> <p>μ PD784215, 784216</p> <p>μ PD784215Y, 784216Y</p> <p>μ PD784915A, 784916A</p> <p>μ PD784214, 784215, 784216, 784214Y, 784215Y, 784216YのGC-7EAパッケージをGC-8EUパッケージに変更</p> <p>μ PD784915サブシリーズには、LOCATION 0FH命令がなく、LOCATION 0命令固定であることを記述</p> <p>μ PD784216, 784216Y, 784218, 784218Y, 784225, 784225Yサブシリーズには、特殊命令（CHKL, CHKLA）がないことを注意で追加</p> <p>μ PD78F4943サブシリーズをμ PD784943サブシリーズに変更</p>	全般

版 数	前版からの主な改版内容	適用箇所
第7版	<p><math>\mu</math> PD784937, 784955サブシリーズを追加, <math>\mu</math> PD784943サブシリーズを削除 次の製品が開発中 開発済み</p> <p><math>\mu</math> PD784031 ( A ), 784035 ( A ), 784036 ( A ), <math>\mu</math> PD784044 ( A ), 784044 ( A1 ), 784044 ( A2 ), 784046 ( A ), 784046 ( A1 ), 784046 ( A2 ), <math>\mu</math> PD784054 ( A ), 784054 ( A1 ), 784054 ( A2 ), <math>\mu</math> PD784214, 784214Y, <math>\mu</math> PD784915B, 784916B, <math>\mu</math> PD784927, 78F4928, 784927Y, 78F4928Y <math>\mu</math> PD78F4216, 78F4216YのGC-7EAパッケージをGC-8EUパッケージに変更 <math>\mu</math> PD784908サブシリーズの電源電圧変更 マスク品 ( <math>\mu</math> PD784907, 784908 ) ... ( <math>V_{DD} = 4.5 \sim 5.5 V</math> ) ( <math>V_{DD} = 3.5 \sim 5.5 V</math> ) PROM品 ( <math>\mu</math> PD78P4908 ) ... ( <math>V_{DD} = 4.5 \sim 5.5 V</math> ) ( <math>V_{DD} = 4.0 \sim 5.5 V</math> )</p>	全般
	特殊命令 ( CHKL, CHKLA ) の注意を変更	第6章 命令セット
	POP命令のオペレーションの順番を修正 CHKL命令に注意を追加 CHKLA命令に注意を追加	第7章 命令の説明
	フォーマットを修正 PC環境に関する記述を追加	第8章 開発のツール 第9章 組み込み用ソフトウェア
第8版	<p><math>\mu</math> PD784216A, 784216AY, 784218A, 784218AY, 784938A, 784956A, 784976Aサブ シリーズを追加, <math>\mu</math> PD784216, 784216Y, 784218, 784218Y, 784937, 784955サブ シリーズを削除, <math>\mu</math> PD784928, 784928Yを追加, <math>\mu</math> PD784915, 784915A, 784916A を削除 次の製品が開発中 開発済み</p> <p><math>\mu</math> PD784224, 784225, 78F4225, 784224Y, 784225Y, 78F4225Y, 784907, 784908, 78P4908</p>	全般
	<p>6. 4. 1 命令コードの記号 ( 8 ) post byteの条件を修正</p> <p>6. 4. 2 命令コード一覧 ( 1 ) 8ビット・データ転送命令 : MOV オペランド「MOV saddr2, A」の命令コードを修正</p>	第6章 命令セット

---

## — 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 : s-info@saed.tmg.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

名古屋 (052)222-2375

大阪 (06)6945-3178, 3200,  
3208, 3212

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

広島 (082)242-5504

高崎 (027)326-1303

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。

### 【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/ シリーズ ユーザーズ・マニュアル 命令編

( U10905JJ8V1UM00 ( 第8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