

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

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

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

2010年4月1日  
ルネサスエレクトロニクス株式会社

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

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

## ご注意書き

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

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

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

# ユーザーズ・マニュアル

## μPD750068

4ビット・シングルチップ・マイクロコンピュータ

---

μPD750064

μPD750066

μPD750068

μPD75P0076

[メモ]

## 目次要約

第1章	概 説	...	21
第2章	端子機能	...	29
第3章	アーキテクチャの特徴とメモリ・マップ	...	45
第4章	内部CPU機能	...	73
第5章	周辺ハードウェア機能	...	95
第6章	割り込み機能とテスト機能	...	211
第7章	スタンバイ機能	...	243
第8章	リセット機能	...	257
第9章	PROM (プログラム・メモリ) の書き込みとベリファイ	...	261
第10章	マスク・オプション	...	267
第11章	命令セット	...	269
付録A	$\mu$ PD75068, 750068, 75P0076の機能一覧表	...	351
付録B	開発ツール	...	353
付録C	マスクROM発注の手順	...	359
付録D	命令索引	...	361
付録E	ハードウェア索引	...	367
付録F	改版履歴	...	373

## 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イやマガジン・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。

MS-DOS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。

IBM DOS, PC/AT, PC DOSは、米国IBM社の商標です。

本製品が外国為替および外国貿易管理法の規定による規制貨物等（または役務）に該当するか否かは、ユーザー（仕様を決定した者）が判定してください。

- 本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。
  - 文書による当社の承諾なしに本資料の転載複製を禁じます。
  - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
  - 本資料に記載された回路、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	μ PD750064, 750068, 75P0076を開発中から開発済みに変更
	μ PD750066を追加
	データ・バス端子 (D0-D7) を追加
	外部クロック使用時, XT2をオープンからXT1の逆相入力に変更
p.43	2.4 未使用端子の処理についてを変更
p.123	表5 - 5 システム・クロック, CPUクロックの切り替えに要する最大時間の表と注意を修正
p.250	7.4 マスク・オプションの選択を追加
p.263	9.2 プログラム・メモリの書き込み手順を修正
p.264	9.3 プログラム・メモリの読み出し手順を修正
p.268	10.3 サブシステム・クロックのフィードバック抵抗のマスク・オプションを追加
p.298	11.3 各命令の命令コードの命令一覧表を修正
p.356	付録B 開発ツールでサポートするOSをバージョン・アップ
p.359	付録C マスクROM発注の手順の発注用媒体を変更
p.373	付録F 改版履歴を追加

本文欄外の★印は、本版で改訂された主な箇所を示しています。

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

# はじめに

**ご利用対象者** このマニュアルは、 $\mu$ PD750064, 750066, 750068, 75P0076の機能を理解し、それを用いたアプリケーション・システムを設計するユーザのエンジニアを対象としています。

**目的** このマニュアルは、次の構成に示す $\mu$ PD750064, 750066, 750068, 75P0076の持つハードウェア機能をユーザに理解していただくことを目的としています。

**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おります。

- ・概説
- ・端子機能
- ・アーキテクチャの特徴とメモリ・マップ
- ・内部CPU機能
- ・周辺ハードウェア機能
- ・割り込み機能とテスト機能
- ・スタンバイ機能
- ・リセット機能
- ・PROMの書き込みとベリファイ
- ・マスク・オプション
- ・命令セット

**読み方** このマニュアルの読者は、電気、論理回路、マイクロコンピュータの一般的な知識を必要とします。

・ **$\mu$ PD75068をすでに経験しているユーザ**

付録A  $\mu$ PD75068, 750068, 75P0076の機能一覧表で違っている部分を確認し、それらの説明を読んでください。

・ **$\mu$ PD750064, 750066, 750068, 75P0076のマニュアルとしてお使いになるユーザ**

このマニュアルは、特に機能面で違いがないかぎり $\mu$ PD750068を代表品種として説明してあります。1.3  $\mu$ PD750068サブシリーズ製品間の違いで機能の違いを確認したうえで、 $\mu$ PD750068をそれぞれの製品に読み替えて使用してください。

・**ニモニックが分かっているときの命令機能を調べるとき**

付録D 命令索引を利用してください。

・**特定の内蔵回路などの機能を調べたいとき**

付録E ハードウェア索引を利用してください。

- ・一通り  $\mu$ PD750064, 750066, 750068, 75P0076の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。

凡 例	データ表記の重み	: 左が上位桁, 右が下位桁
	アクティブ・ロウの表記	: $\overline{\text{xxx}}$ (端子, 信号名称に上線)
	メモリ・マップのアドレス	: 上部 - 下位, 下部 - 上位
	注	: 本文中につけた注の説明
	注意	: 気をつけて読んでいただきたい内容
	備考	: 本文の補足説明
	数の表記	: 2進数... xxxxB 10進数... xxx 16進数... xxxH

**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。

#### デバイス関連資料一覧

資料名	資料番号	
	和文	英文
μPD750064, 750066, 750068, 750064 (A), 750066 (A), 750068 (A) データ・シート	U10165J	U10165E
μPD75P0076 データ・シート	U10232J	U10232E
μPD750068 ユーザーズ・マニュアル	U10670(このマニュアル)	U10670E
μPD750068 インストラクション活用表	IEM-5606	-
75XLシリーズ セレクション・ガイド	U10453J	U10453E

#### 開発ツール関連資料一覧

	資料名	資料番号		
		和文	英文	
ハードウェア	IE-75000-R/IE-75001-R ユーザーズ・マニュアル	EEU-846	EEU-1416	
	IE-75300-R-EM ユーザーズ・マニュアル	U11354J	U11354E	
	EP-750068CU/GT-R ユーザーズ・マニュアル	U10950J	U10950E	
	PG-1500 ユーザーズ・マニュアル	U11940J	U11940E	
ソフトウェア	RA75X アセンブラ・パッケージ ユーザーズ・マニュアル	操作編	U12622J	U12622E
		言語編	U12385J	U12385E
	PG-1500コントローラ ユーザーズ・マニュアル	PC-9800シリーズ (MS-DOS) ベース	EEU-704	EEU-1291
		IBM PCシリーズ (PC DOS) ベース	EEU-5008	U10540E

#### その他のドキュメント

資料名	資料番号	
	和文	英文
SEMICONDUCTORS SELECTION GUIDE Products & Packages (CD-ROM)	X13769X	
半導体デバイス 実装マニュアル	C10535J	C10535E
NEC半導体デバイスの品質水準	C11531J	C11531E
NEC半導体デバイスの信頼性品質管理	C10983J	C10983E
静電気放電 (ESD) 破壊対策ガイド	C11892J	C11892E
半導体品質 / 信頼性ハンドブック	C12769J	-
マイクロコンピュータ関連製品ガイド 社外メーカ編	U11416J	-

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

(メモ)

# 目 次

<b>第1章 概 説</b>	...	21
1.1 機能概要	...	22
1.2 オーダ情報	...	24
1.3 $\mu$ PD750068サブシリーズ製品間の違い	...	24
1.4 ブロック図	...	25
1.5 端子接続図 (Top View)	...	26
<b>第2章 端子機能</b>	...	29
2.1 $\mu$ PD750068の端子機能一覧	...	29
2.2 端子機能の説明	...	33
2.2.1 P00-P03 (PORT0), P10-P13 (PORT1), P110-P113 (PORT11)	...	33
2.2.2 P20-P23 (PORT2), P30-P33 (PORT3), P40-P43 (PORT4), P50-P53 (PORT5), P60-P63 (PORT6)	...	34
2.2.3 TI0, TI1	...	35
2.2.4 PTO0, PTO1	...	35
2.2.5 PCL	...	35
2.2.6 BUZ	...	35
2.2.7 $\overline{\text{SCK}}$ , SO/SB0, SI/SB1	...	35
2.2.8 INT4	...	36
2.2.9 INT0, INT1	...	36
2.2.10 INT2	...	36
2.2.11 KR0-KR3	...	37
2.2.12 AN0-AN7	...	37
2.2.13 AVREF	...	37
2.2.14 AVss	...	37
2.2.15 X1, X2	...	37
2.2.16 XT1, XT2	...	38
2.2.17 $\overline{\text{RESET}}$	...	38
2.2.18 MD0-MD3 ( $\mu$ PD75P0076のみ)	...	38
2.2.19 D0-D7 ( $\mu$ PD75P0076のみ)	...	38
2.2.20 IC ( $\mu$ PD750064, 750066, 750068のみ)	...	39
2.2.21 VPP ( $\mu$ PD75P0076のみ)	...	39
2.2.22 VDD	...	39
2.2.23 Vss	...	39
2.3 端子の入出力回路	...	40
2.4 未使用端子の処理について	...	43
<b>第3章 アーキテクチャの特徴とメモリ・マップ</b>	...	45
3.1 データ・メモリのバンク構成とアドレッシング・モード	...	45

- 3.1.1 データ・メモリのバンク構成 ... 45
- 3.1.2 データ・メモリのアドレッシング・モード ... 47
- 3.2 汎用レジスタのバンク構成 ... 60
- 3.3 メモリ・マップトI/O ... 65

## 第4章 内部CPU機能 ... 73

- 4.1 Mk モードとMk モードの切り替え機能 ... 73
  - 4.1.1 Mk モードとMk モードの違い ... 73
  - 4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法 ... 74
- 4.2 プログラム・カウンタ (PC) ... 75
- 4.3 プログラム・メモリ (ROM) ... 76
- 4.4 データ・メモリ (RAM) ... 81
  - 4.4.1 データ・メモリの構成 ... 81
  - 4.4.2 データ・メモリのバンクの指定 ... 82
- 4.5 汎用レジスタ ... 85
- 4.6 アキュムレータ ... 86
- 4.7 スタック・ポインタ (SP) , スタック・バンク選択レジスタ (SBS) ... 86
- 4.8 プログラム・ステータス・ワード (PSW) ... 90
- 4.9 バンク選択レジスタ (BS) ... 94

## 第5章 周辺ハードウェア機能 ... 95

- 5.1 デジタル入出力ポート ... 95
  - 5.1.1 デジタル入出力ポートの種類, 特徴, 構成 ... 96
  - 5.1.2 入出力モードの設定 ... 101
  - 5.1.3 デジタル入出力ポート操作命令 ... 103
  - 5.1.4 デジタル入出力ポートの動作 ... 106
  - 5.1.5 プルアップ抵抗の内蔵 ... 108
  - 5.1.6 デジタル入出力ポートの入出力タイミング ... 110
- 5.2 クロック発生回路 ... 112
  - 5.2.1 クロック発生回路の構成 ... 112
  - 5.2.2 クロック発生回路の機能と動作 ... 113
  - 5.2.3 システム・クロックとCPUクロックの設定 ... 123
  - 5.2.4 クロック出力回路 ... 125
- 5.3 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ ... 128
  - 5.3.1 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマの構成 ... 128
  - 5.3.2 ベーシック・インターバル・タイマ・モード・レジスタ (BTM) ... 129
  - 5.3.3 ウォッチドッグ・タイマ許可フラグ (WDTM) ... 131
  - 5.3.4 ベーシック・インターバル・タイマの動作 ... 132
  - 5.3.5 ウォッチドッグ・タイマの動作 ... 133
  - 5.3.6 その他の機能 ... 135
- 5.4 時計用タイマ ... 138
  - 5.4.1 時計用タイマの構成 ... 139
  - 5.4.2 時計モード・レジスタ ... 140
- 5.5 タイマ/イベント・カウンタ ... 142

5.5.1	タイマ/イベント・カウンタの構成 ...	142
5.5.2	8ビット・タイマ/イベント・カウンタ・モードの動作 ...	150
5.5.3	16ビット・タイマ/イベント・カウンタ・モードの動作 ...	157
5.5.4	タイマ/イベント・カウンタ応用上の注意 ...	166
<b>5.6</b>	<b>シリアル・インタフェース ...</b>	<b>171</b>
5.6.1	シリアル・インタフェースの機能 ...	171
5.6.2	シリアル・インタフェースの構成 ...	171
5.6.3	レジスタの機能 ...	175
5.6.4	動作停止モード ...	180
5.6.5	3線式シリアルI/Oモードの動作 ...	182
5.6.6	2線式シリアルI/Oモードの動作 ...	192
5.6.7	SCK端子出力の操作 ...	199
<b>5.7</b>	<b>A/Dコンバータ ...</b>	<b>200</b>
5.7.1	A/Dコンバータの構成 ...	200
5.7.2	A/Dコンバータの動作 ...	203
5.7.3	スタンバイ・モード時の注意 ...	206
5.7.4	その他, 使用上の注意 ...	206
<b>5.8</b>	<b>ビット・シーケンシャル・バッファ ...</b>	<b>208</b>
<b>第6章</b>	<b>割り込み機能とテスト機能 ...</b>	<b>211</b>
6.1	割り込み制御回路の構成 ...	211
6.2	割り込みソースの種類とベクタ・テーブル ...	213
6.3	割り込み機能を制御する各種ハードウェア ...	215
6.4	割り込みシーケンス ...	223
6.5	多重割り込み処理制御 ...	224
6.6	ベクタ・アドレス共用割り込み処理 ...	226
6.7	割り込み処理までのマシン・サイクル ...	228
6.8	割り込みの効果的な使い方 ...	230
6.9	割り込みの応用 ...	230
6.10	テスト機能 ...	238
6.10.1	テスト・ソースの種類 ...	238
6.10.2	テスト機能を制御する各種ハードウェア ...	238
<b>第7章</b>	<b>スタンバイ機能 ...</b>	<b>243</b>
7.1	スタンバイ・モードの設定および動作状態 ...	245
7.2	スタンバイ・モードの解除 ...	247
7.3	スタンバイ・モード解除後の動作 ...	250
7.4	マスク・オプションの選択 ...	250
7.5	スタンバイ・モードの応用 ...	251
<b>第8章</b>	<b>リセット機能 ...</b>	<b>257</b>
<b>第9章</b>	<b>PROM (プログラム・メモリ) の書き込みとベリファイ ...</b>	<b>261</b>
9.1	プログラム・メモリの書き込み/ベリファイ時の動作モード ...	262
9.2	プログラム・メモリの書き込み手順 ...	263

9.3	プログラム・メモリの読み出し手順 ...	264
9.4	ワン・タイムPROMのスクリーニングについて ...	265
<b>第10章</b>	<b>マスク・オプション</b> ...	267
10.1	端子 ...	267
10.2	スタンバイ機能のマスク・オプション ...	267
10.3	サブシステム・クロックのフィードバック抵抗のマスク・オプション ...	268
<b>第11章</b>	<b>命令セット</b> ...	269
11.1	特徴的な命令 ...	269
11.1.1	GETI命令 ...	269
11.1.2	ビット操作命令 ...	270
11.1.3	たてづみ命令 ...	270
11.1.4	進数補正命令 ...	271
11.1.5	スキップ命令とスキップに要するマシン・サイクル数 ...	272
11.2	命令セットとそのオペレーション ...	272
11.3	各命令の命令コード ...	293
11.4	命令機能と応用 ...	299
11.4.1	転送命令 ...	300
11.4.2	テーブル参照命令 ...	309
11.4.3	ビット転送命令 ...	313
11.4.4	演算命令 ...	314
11.4.5	アキュムレータ操作命令 ...	322
11.4.6	増減命令 ...	323
11.4.7	比較命令 ...	325
11.4.8	キャリー・フラグ操作命令 ...	327
11.4.9	メモリ・ビット操作命令 ...	328
11.4.10	分岐命令 ...	332
11.4.11	サブルーチン・スタック制御命令 ...	337
11.4.12	割り込み制御命令 ...	342
11.4.13	入出力命令 ...	343
11.4.14	CPU制御命令 ...	345
11.4.15	特殊命令 ...	346
<b>付録A</b>	<b><math>\mu</math>PD75068 , 750068 , 75P0076の機能一覧表</b> ...	351
<b>付録B</b>	<b>開発ツール</b> ...	353
<b>付録C</b>	<b>マスクROM発注の手順</b> ...	359
<b>付録D</b>	<b>命令索引</b> ...	361
D.1	命令索引(機能別) ...	361
D.2	命令索引(アルファベット順) ...	364
<b>付録E</b>	<b>ハードウェア索引</b> ...	367

- E.1 ハードウェア索引 (50音順) ... 367
- E.2 ハードウェア索引 (アルファベット順) ... 370

**付録 F 改版履歴** ... 373

## 図の目次 (1/3)

図番号	タイトル, ページ
3 - 1	MBE = 0のモードとMBE = 1のモードの使い分け ... 46
3 - 2	データ・メモリの構成と, 各アドレッシング・モードのアドレッシング範囲 ... 48
3 - 3	スタティックRAMのアドレス更新方法 ... 54
3 - 4	レジスタ・バンクの使い分け例 ... 61
3 - 5	汎用レジスタの構成 (4ビット処理の場合) ... 63
3 - 6	汎用レジスタの構成 (8ビット処理の場合) ... 64
3 - 7	μPD750068 I/Oマップ ... 67
4 - 1	スタック・バンク選択レジスタのフォーマット ... 74
4 - 2	プログラム・カウンタの構成 ... 75
4 - 3	プログラム・メモリ・マップ ... 77
4 - 4	データ・メモリ・マップ ... 83
4 - 5	汎用レジスタの構成 ... 85
4 - 6	レジスタ・ペアの構成 ... 85
4 - 7	アキュムレータ ... 86
4 - 8	スタック・ポインタおよびスタック・バンク選択レジスタの構成 ... 87
4 - 9	スタック・メモリへ退避されるデータ (Mk モード) ... 88
4 - 10	スタック・メモリから復帰されるデータ (Mk モード) ... 88
4 - 11	スタック・メモリへ退避されるデータ (Mk モード) ... 89
4 - 12	スタック・メモリから復帰されるデータ (Mk モード) ... 89
4 - 13	プログラム・ステータス・ワードの構成 ... 90
4 - 14	バンク選択レジスタの構成 ... 94
5 - 1	デジタル・ポートのデータ・メモリ・アドレス ... 95
5 - 2	ポート0, 1の構成 ... 97
5 - 3	ポート3, 6の構成 ... 98
5 - 4	ポート2の構成 ... 99
5 - 5	ポート4, 5の構成 ... 100
5 - 6	ポート11の構成 ... 100
5 - 7	各ポート・モード・レジスタのフォーマット ... 102
5 - 8	プルアップ抵抗指定レジスタのフォーマット ... 109
5 - 9	デジタル入出力ポートの入出力タイミング ... 110
5 - 10	ソフトウェアによる内蔵プルアップ抵抗のONタイミング ... 111
5 - 11	クロック発生回路のブロック図 ... 112

## 図の目次 (2/3)

図番号	タイトル, ページ
5 - 12	プロセッサ・クロック・コントロール・レジスタのフォーマット ... 115
5 - 13	システム・クロック・コントロール・レジスタのフォーマット ... 116
5 - 14	メイン・システム・クロック発振回路の外付け回路 ... 117
5 - 15	サブシステム・クロック発振回路の外付け回路 ... 117
5 - 16	発振子の接続回路の悪い例 ... 118
5 - 17	サブシステム・クロック発振回路 ... 121
5 - 18	サブ発振回路コントロール・レジスタ (SOS) のフォーマット ... 122
5 - 19	システム・クロックとCPUクロックの切り替え ... 124
5 - 20	クロック出力回路のブロック図 ... 125
5 - 21	クロック出力モード・レジスタのフォーマット ... 126
5 - 22	リモコン波形出力応用例 ... 127
5 - 23	ベーシック・インターバル・タイマ/ウォッチドッグ・タイマのブロック図 ... 128
5 - 24	ベーシック・インターバル・タイマ・モード・レジスタのフォーマット ... 130
5 - 25	ウォッチドッグ・タイマ許可フラグ (WDTM) のフォーマット ... 131
5 - 26	時計用タイマのブロック図 ... 139
5 - 27	時計モード・レジスタのフォーマット ... 141
5 - 28	タイマ/イベント・カウンタ (チャンネル0) のブロック図 ... 143
5 - 29	タイマ/イベント・カウンタ (チャンネル1) のブロック図 ... 144
5 - 30	タイマ/イベント・カウンタ・モード・レジスタ (チャンネル0) のフォーマット ... 146
5 - 31	タイマ/イベント・カウンタ・モード・レジスタ (チャンネル1) のフォーマット ... 147
5 - 32	タイマ/イベント・カウンタ出力許可フラグのフォーマット ... 149
5 - 33	タイマ/イベント・カウンタ・モード・レジスタの設定 (8ビット時) ... 151
5 - 34	タイマ/イベント・カウンタ出力許可フラグの設定 ... 152
5 - 35	タイマ/イベント・カウンタ動作時の構成 ... 155
5 - 36	カウント動作時のタイミング ... 155
5 - 37	タイマ/イベント・カウンタ・モード・レジスタの設定 ... 158
5 - 38	タイマ/イベント・カウンタ出力許可フラグの設定 ... 159
5 - 39	タイマ/イベント・カウンタ動作時の構成 ... 162
5 - 40	カウント動作時のタイミング ... 162
5 - 41	シリアル・インタフェースのブロック図 ... 172
5 - 42	シリアル動作モード・レジスタ (CSIM) のフォーマット ... 175
5 - 43	シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) のフォーマット ... 178
5 - 44	シフト・レジスタの周辺の構成 ... 179
5 - 45	3線式シリアルI/Oのシステム構成例 ... 182

## 図の目次 (3/3)

図番号	タイトル, ページ
5 - 46	3線式シリアルI/Oモードのタイミング ... 185
5 - 47	RELT, CMDTの動作 ... 186
5 - 48	転送ビットの切り替え回路 ... 187
5 - 49	2線式シリアルI/Oのシステム構成例 ... 192
5 - 50	2線式シリアルI/Oモードのタイミング ... 195
5 - 51	RELT, CMDTの動作 ... 196
5 - 52	$\overline{\text{SCK}}$ /P01端子の構成 ... 199
5 - 53	A/Dコンバータのブロック図 ... 200
5 - 54	A/D変換モード・レジスタのフォーマット ... 202
5 - 55	A/D変換のタイミング・チャート ... 205
5 - 56	アナログ入力電圧とA/D変換結果の関係 (理想的な場合) ... 206
5 - 57	アナログ入力端子の処理 ... 207
5 - 58	ビット・シーケンシャル・バッファのフォーマット ... 208
6 - 1	割り込み制御回路ブロック図 ... 212
6 - 2	割り込みベクタ・テーブル ... 214
6 - 3	割り込みプライオリティ選択レジスタ ... 217
6 - 4	INT0, INT1, INT4の構成 ... 219
6 - 5	ノイズ除去回路の入出力タイミング ... 220
6 - 6	エッジ検出モード・レジスタのフォーマット ... 221
6 - 7	割り込み処理シーケンス ... 223
6 - 8	高位の割り込みによる多重割り込み ... 224
6 - 9	割り込みステータス・フラグの変更による多重割り込み ... 225
6 - 10	INT2, KR0-KR3のブロック図 ... 240
6 - 11	INT2エッジ検出モード・レジスタ (IM2) のフォーマット ... 241
7 - 1	スタンバイ・モードの解除動作 ... 247
7 - 2	STOPモード解除時のウェイト時間 ... 249
8 - 1	リセット機能の構成 ... 257
8 - 2	$\overline{\text{RESET}}$ 信号発生によるリセット動作 ... 257

## 表の目次 (1/2)

表番号	タイトル, ページ
2 - 1	デジタル入出力ポートの端子機能一覧 ... 29
2 - 2	ポート端子以外の端子機能一覧 ... 31
2 - 3	未使用端子の処理一覧 ... 43
3 - 1	アドレッシング・モード ... 49
3 - 2	RBE, RBSと選択されるレジスタ・バンク ... 60
3 - 3	通常ルーチンと割り込みルーチンでのレジスタ・バンクの使い分けの例 ... 60
3 - 4	周辺ハードウェア操作時に適用可能なアドレッシング・モード ... 65
4 - 1	Mk モードとMk モードの違い ... 73
4 - 2	SBSで選択されるスタック・エリア ... 86
4 - 3	スタック動作時に退避 / 復帰されるPSWのフラグ ... 90
4 - 4	キャリー・フラグ操作命令 ... 91
4 - 5	割り込みステータス・フラグの指示内容 ... 92
4 - 6	RBE, RBSと選択されるレジスタ・バンク ... 94
5 - 1	デジタル・ポートの種類と特徴 ... 96
5 - 2	入出力端子操作命令一覧表 ... 105
5 - 3	入出力ポート操作時の動作 ... 107
5 - 4	プルアップ抵抗内蔵の指定法 ... 108
5 - 5	システム・クロック, CPUクロックの切り替えに要する最大時間 ... 123
5 - 6	使用可能モード一覧表 ... 142
5 - 7	分解能と最長設定時間 ( 8ビット・タイマ時 ) ... 153
5 - 8	分解能と最長設定時間 ( 16ビット・タイマ時 ) ... 160
5 - 9	シリアル・クロックの選択と応用 ( 3線式シリアルI/Oモード時 ) ... 186
5 - 10	シリアル・クロックの選択と応用 ( 2線式シリアルI/Oモード時 ) ... 196
5 - 11	SCC, PCCの設定 ... 205
6 - 1	割り込みソースの種類 ... 213
6 - 2	割り込み要求フラグのセット信号 ... 216
6 - 3	IST1, IST0と割り込み処理状態 ... 222
6 - 4	共用割り込みの判別 ... 226
6 - 5	テスト・ソースの種類 ... 238
6 - 6	テスト要求フラグのセット信号 ... 238

## 表の目次 (2/2)

表番号	タイトル, ページ
7 - 1	スタンバイ・モード時の各動作状態 ... 245
7 - 2	BTMによるウェイト時間の選択 ... 249
8 - 1	各ハードウェアのリセット後の状態 ... 258
9 - 1	プログラム・メモリ書き込み/ベリファイ時の使用端子 ... 261
9 - 2	動作モード ... 262
10 - 1	端子のマスク・オプションの選択 ... 267
11 - 1	ビット操作アドレッシングの種類と指定可能な範囲 ... 270

# 第1章 概 説

μPD750064, 750066, 750068, 75P0076は、豊富な製品展開を誇る75Xシリーズの後継品種75XLシリーズの4ビット・シングルチップ・マイクロコンピュータです。このμPD750064, 750066, 750068, 75P0076を総称してμPD750068サブシリーズといいます。

μPD750068は、従来品であるμPD750068に比べて、ROM容量を拡大、CPU機能拡張、低電圧1.8Vでの高速動作を可能にした製品です。

小型のプラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ) を用意しています。

以下に特徴を示します。

- ・低電圧動作可能： $V_{DD}=1.8 \sim 5.5$  V
- ・高速動作とパワー・セーブに有利な命令実行時間可変機能
  - 0.95  $\mu$ s, 1.91  $\mu$ s, 3.81  $\mu$ s, 15.3  $\mu$ s (4.19 MHz動作時)
  - 0.67  $\mu$ s, 1.33  $\mu$ s, 2.67  $\mu$ s, 10.7  $\mu$ s (6.0 MHz動作時)
  - 122  $\mu$ s (32.768 kHz動作時)
- ・タイマを4チャンネル内蔵
- ・低電圧動作可能なA/Dコンバータ内蔵 (8ビット分解能×8チャンネル, 逐次比較型)
- ・小型パッケージを採用 (42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ))

μPD75P0076は、電氣的書き込み可能なワン・タイムPROMを備えた製品で、μPD750068とピン・コンパチブルです。システム開発時の試作用、または少量生産用に適しています。

## 応用分野

- ・コードレス電話機
- ・AV機器
- ・家電
- ・OA                      など

**備考** このマニュアルでは、特に機能面での違いがないかぎりμPD750068を代表品種として記述してあります。μPD750064, 750066, 75P0076のマニュアルとして使用する場合は、μPD750068をそれぞれの製品に読み替えて使用してください。

## 1.1 機能概要

## 機能概要 (1/2)

項 目		機 能	
命令実行時間		<ul style="list-style-type: none"> <li>・ 0.95, 1.91, 3.81, 15.3 <math>\mu</math>s (メイン・システム・クロック : 4.19 MHz動作時)</li> <li>・ 0.67, 1.33, 2.67, 10.7 <math>\mu</math>s ( " : 6.0 MHz動作時)</li> <li>・ 122 <math>\mu</math>s (サブシステム・クロック : 32.768 kHz動作時)</li> </ul>	
内蔵メモリ	ROM	4096 x 8 ビット ( $\mu$ PD750064 )	
		6144 x 8 ビット ( $\mu$ PD750066 )	
		8192 x 8 ビット ( $\mu$ PD750068 )	
		16384 x 8 ビット ( $\mu$ PD75P0076 )	
	RAM	512 x 4 ビット	
汎用レジスタ		<ul style="list-style-type: none"> <li>・ 4 ビット操作時 : 8 個 x 4 バンク</li> <li>・ 8 ビット操作時 : 4 個 x 4 バンク</li> </ul>	
入出力ポート	CMOS入力	12本	ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能 : 7 本 アナログ入力端子と兼用 : 4 本
	CMOS入出力	12本	ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能 : 12本 アナログ入力端子と兼用 : 4 本
	N-chオープン・ドレイン入出力	8 本	13 V耐圧, マスク・オプションによるプルアップ抵抗内蔵可能 <sup>注</sup>
	合計	32本	
タイマ		4 チャンネル <ul style="list-style-type: none"> <li>・ 8 ビット・タイマ/イベント・カウンタ : 2 チャンネル ( 16 ビット・タイマ/イベント・カウンタとして使用可能 )</li> <li>・ ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ : 1 チャンネル</li> <li>・ 時計用タイマ : 1 チャンネル</li> </ul>	
シリアル・インタフェース		<ul style="list-style-type: none"> <li>・ 3 線式シリアルI/Oモード...MSB/LSB先頭切り替え</li> <li>・ 2 線式シリアルI/Oモード</li> </ul>	
A/Dコンバータ		8 ビット分解能 x 8 チャンネル ( 1.8 V $V_{REF}$ $V_{DD}$ )	
ビット・シーケンシャル・バッファ ( BSB )		16ビット	
クロック出力 ( PCL )		<ul style="list-style-type: none"> <li>・ , 1.05 MHz, 262 kHz, 65.5 kHz (メイン・システム・クロック : 4.19 MHz動作時)</li> <li>・ , 1.5 MHz, 375 kHz, 93.8 kHz ( " : 6.0 MHz動作時)</li> </ul>	
ブザー出力 ( BUZ )		<ul style="list-style-type: none"> <li>・ 2, 4, 32 kHz (メイン・システム・クロック : 4.19 MHz動作時 またはサブシステム・クロック : 32.768 kHz動作時)</li> <li>・ 2.93, 5.86, 46.9 kHz (メイン・システム・クロック : 6.0 MHz動作時)</li> </ul>	

注  $\mu$ PD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

## 機能概要 (2/2)

項 目	機 能
ベクタ割り込み	外部：3本，内部：4本
テスト入力	外部：1本，内部：1本
システム・クロック発振回路	・メイン・システム・クロック発振用セラミック/クリスタル発振回路 ・サブシステム・クロック発振用クリスタル発振回路
スタンバイ機能	STOPモード/HALTモード
電源電圧	$V_{DD} = 1.8 \sim 5.5 \text{ V}$
パッケージ	・42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ) ・42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)

## 1.2 オーダ情報

オーダ名称	パッケージ	内蔵ROM
μ PD750064CU- x x x	42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ)	マスクROM
μ PD750064GT- x x x	42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)	"
μ PD750066CU- x x x	42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ)	"
μ PD750066GT- x x x	42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)	"
μ PD750068CU- x x x	42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ)	"
μ PD750068GT- x x x	42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)	"
μ PD75P0076CU	42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ)	ワン・タイムPROM
μ PD75P0076GT	42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)	"

備考 x x xはROMコード番号です。

## 1.3 μ PD750068サブシリーズ製品間の違い

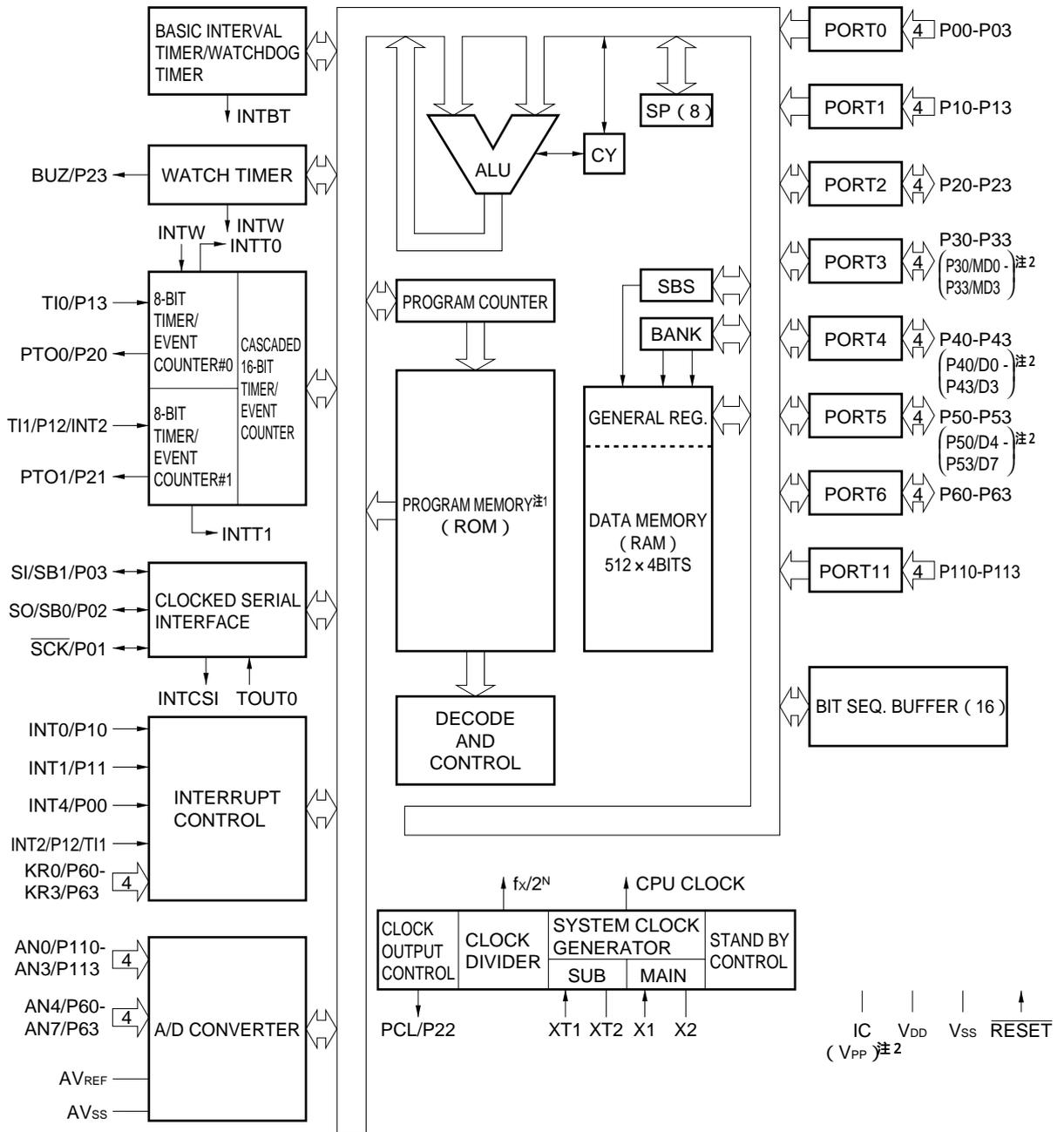
項 目		μ PD750064	μ PD750066	μ PD750068	μ PD75P0076
プログラム・カウンタ		12ビット	13ビット		14ビット
プログラム・メモリ (バイト)		マスクROM 4096	マスクROM 6144	マスクROM 8192	ワン・タイムPROM 16384
データ・メモリ (x 4ビット)		512			
マスク・オプション	ポート4, 5のプルアップ抵抗	あり (内蔵する / しないの指定可能)			なし (内蔵不可)
	RESET時のウェイト時間	あり (2 <sup>17</sup> /fx, 2 <sup>15</sup> /fxの2つから選択可能) <sup>注</sup>			なし (2 <sup>15</sup> /fx固定) <sup>注</sup>
	サブシステム・クロックのフィードバック抵抗	あり (使用する / しないの指定可能)			なし (使用する)
端子接続	6-9番ピン	P33-P30			P33/MD3-P30/MD0
	20番ピン	IC			V <sub>PP</sub>
	34-37番ピン	P53-P50			P53/D7-P50/D4
	38-41番ピン	P43-P40			P43/D3-P40/D0
その他	回路の規模やマスク・レイアウトが異なるため、ノイズ耐量、ノイズ輻射などが異なります。				

注 2<sup>17</sup>/fxは、6.0 MHz動作時：21.8 ms, 4.19 MHz動作時：31.3 msです。

2<sup>15</sup>/fxは、6.0 MHz動作時：5.46 ms, 4.19 MHz動作時：7.81 msです。

注意 PROMとマスクROMでは、ノイズ耐量やノイズ輻射が異なります。試作から量産の過程でPROM品からマスクROM品への置き換えを検討される場合は、マスクROMのCS品 (ES品ではなく) で十分な評価を行ってください。

## 1.4 ブロック図



注1．ROMの容量は、品種により異なります。

2．( )内は、μPD75P0076の場合です。

## 1.5 端子接続図 (Top View)

42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ)

μ PD750064CU- x x x

μ PD750066CU- x x x

μ PD750068CU- x x x

μ PD75P0076CU

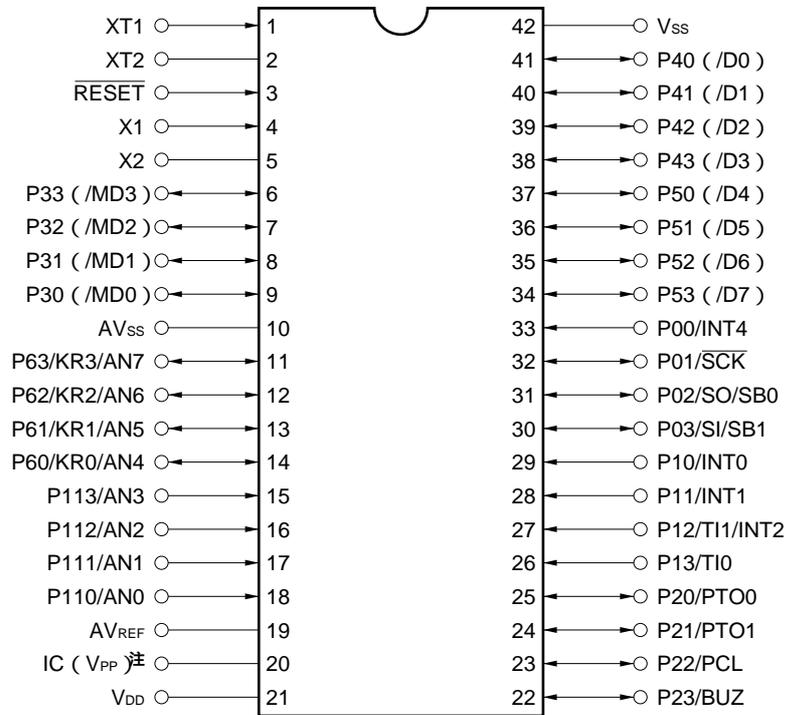
42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)

μ PD750064GT- x x x

μ PD750066GT- x x x

μ PD750068GT- x x x

μ PD75P0076GT



注 IC (VPP) は、VDDに直接接続してください。

備考 ( )内は、μ PD75P0076の場合です。

P00-P03	: Port0	PTO0, PTO1	: Programmable Timer Output 0, 1
P10-P13	: Port1	BUZ	: Buzzer Clock
P20-P23	: Port2	PCL	: Programmable Clock
P30-P33	: Port3	INT0, INT1, INT4	: External Vectored Interrupt 0, 1, 4
P40-P43	: Port4	INT2	: External Test Input 2
P50-P53	: Port5	X1, X2	: Main System Clock Oscillation 1, 2
P60-P63	: Port6	XT1, XT2	: Subsystem Clock Oscillation 1, 2
P110-P113	: Port11	AN0-AN7	: Analog Input 0-7
KR0-KR3	: Key Return 0-3	AV <sub>REF</sub>	: Analog Reference
$\overline{\text{SCK}}$	: Serial Clock	AV <sub>SS</sub>	: Analog Ground
SI	: Serial Input	V <sub>DD</sub>	: Positive Power Supply
SO	: Serial Output	V <sub>SS</sub>	: Ground
SB0, SB1	: Serial Data Bus 0, 1	IC	: Internally Connected
$\overline{\text{RESET}}$	: Reset Input	MD0-MD3	: Mode Selection 0-3
TI0, TI1	: Timer Input 0, 1	D0-D7	: Data Bus 0-7
		V <sub>PP</sub>	: Programming Power Supply

(メモ)

## 第2章 端子機能

### 2.1 μPD750068の端子機能一覧

表2-1 デジタル入出力ポートの端子機能一覧(1/2)

端子名称	入出力	兼用端子	機能	8ビットI/O	リセット時	入出力回路 TYPE <sup>注1</sup>
P00	入力	INT4	4ビット入力ポート(PORT0)。 P01-P03は3ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。	×	入力	ⓑ
P01	入出力	SCK				Ⓕ-A
P02	入出力	SO/SB0				Ⓕ-B
P03	入出力	SI/SB1				Ⓜ-C
P10	入力	INT0	4ビット入力ポート(PORT1)。 4ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。 P10/INT0はノイズ除去回路を選択可能。	×	入力	ⓑ-C
P11		INT1				
P12		TI1/INT2				
P13		TI0				
P20	入出力	PTO0	4ビット入出力ポート(PORT2)。 4ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。	×	入力	E-B
P21		PTO1				
P22		PCL				
P23		BUZ				
P30	入出力	(MD0) <sup>注3</sup>	プログラマブル4ビット入出力ポート(PORT3)。 ビット単位で入力/出力指定可能。 4ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。	×	入力	E-B
P31		(MD1) <sup>注3</sup>				
P32		(MD2) <sup>注3</sup>				
P33		(MD3) <sup>注3</sup>				
P40 <sup>注2</sup>	入出力	(D0) <sup>注3</sup>	N-chオープン・ドレイン4ビット入出力ポート(PORT4)。オープン・ドレイン時、13V耐圧。 ビット単位でプルアップ抵抗を内蔵可能(マスク・オプション) <sup>注4</sup> 。 プログラム・メモリ(PROM)書き込み/ベリファイ時のデータ入出力端子(下位4ビット)。		ハイ・レベル (プルアップ抵抗内蔵時)またはハイ・インピーダンス	M-D (M-E) <sup>注3</sup>
P41 <sup>注2</sup>		(D1) <sup>注3</sup>				
P42 <sup>注2</sup>		(D2) <sup>注3</sup>				
P43 <sup>注2</sup>		(D3) <sup>注3</sup>				

注1. 印はシュミット・トリガ入力を示します。

2. マスク・オプションによるプルアップ抵抗を内蔵しない場合(N-chオープン・ドレイン入力ポートとして使用している場合)、入力命令、ビット操作命令を行ったときにロウ・レベル入力リーク電流が増加します。
3. ( )内はμPD75P0076の場合。
4. μPD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

表2-1 デジタル入出力ポートの端子機能一覧(2/2)

端子名称	入出力	兼用端子	機 能	8ビット I/O	リセット時	入出力 回 路 TYPE <sup>注1</sup>
P50 <sup>注2</sup>	入出力	(D4) <sup>注3</sup>	N-chオープン・ドレイン4ビット入出力ポート (PORT5)。オープン・ドレイン時、13V耐圧。 ビット単位でプルアップ抵抗を内蔵可能(マスク・ オプション) <sup>注4</sup> 。 プログラム・メモリ(PROM)書き込み/ベリファイ 時のデータ入出力端子(上位4ビット)。		ハイ・レベル (プルアップ抵抗 内蔵時)または ハイ・イン ピーダンス	M-D (M-E) <sup>注3</sup>
P51 <sup>注2</sup>		(D5) <sup>注3</sup>				
P52 <sup>注2</sup>		(D6) <sup>注3</sup>				
P53 <sup>注2</sup>		(D7) <sup>注3</sup>				
P60	入出力	KR0/AN4	プログラマブル4ビット入出力ポート(PORT6)。 ビット単位で入力/出力指定可能。 4ビット単位で、ソフトウェアにより内蔵プルアッ プ抵抗の接続を指定可能。	x	入 力	⓪-D
P61		KR1/AN5				
P62		KR2/AN6				
P63		KR3/AN7				
P110	入 力	AN0	4ビット入力ポート(PORT11)。	x	入 力	Y-A
P111		AN1				
P112		AN2				
P113		AN3				

注1 . 印はシュミット・トリガ入力を示します。

- 2 . マスク・オプションによるプルアップ抵抗を内蔵しない場合(N-chオープン・ドレイン入力ポートとして使  
用している場合), 入力命令, ビット操作命令を行ったときにロウ・レベル入力リーク電流が増加します。
- 3 . ( ) 内はμPD75P0076の場合。
- 4 . μPD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

表2-2 ポート端子以外の端子機能一覧(1/2)

端子名称	入出力	兼用端子	機 能		リセット時	入出力 回路 TYPE <sup>注</sup>
Ti0	入 力	P13	タイマ/イベント・カウンタへの外部イベント・パルス入力。		入 力	Ⓑ-C
Ti1		P12/INT2				
PTO0	出 力	P20	タイマ/イベント・カウンタ出力。		入 力	E-B
PTO1		P21				
PCL		P22	クロック出力。			
BUZ		P23	任意の周波数出力(ブザー用またはシステム・クロックのトリミング用)。			
$\overline{\text{SCK}}$	入出力	P01	シリアル・クロック入出力。		入 力	Ⓕ-A
SO/SB0		P02	シリアル・データ出力。 シリアル・データ・バス入出力。			Ⓕ-B
SI/SB1		P03	シリアル・データ入力。 シリアル・データ・バス入出力。			Ⓜ-C
INT4	入 力	P00	エッジ検出ベクタ割り込み入力(立ち上がりおよび立ち下がりの両エッジ検出)。		入 力	Ⓑ
INT0	入 力	P10	エッジ検出ベクタ割り込み入力 (検出エッジ選択可能)。	ノイズ除去回路付き /非同期選択可	入 力	Ⓑ-C
INT1		P11	INT0/P10はノイズ除去回路を選択可能。	非同期		
INT2		P12/Ti1	立ち上がりエッジ検出テストブル入力。	非同期		
KR0-KR3	入 力	P60/AN4-P63/AN7	立ち下がりエッジ検出テストブル入力。		入 力	Ⓜ-D
AN0-AN3	入 力	P110-P113	アナログ信号入力。		入 力	Y-A
AN4-AN7		P60/KR0- P63/KR3				Ⓜ-D
AV <sub>REF</sub>	-	-	A/Dコンバータ基準電圧。		-	Z-N
AV <sub>SS</sub>	-	-	A/Dコンバータ基準GND電位。		-	Z-N
X1	入 力	-	メイン・システム・クロック発振用クリスタル/セラミック接続端子。外部クロックの場合、X1へ入力しX2へその逆相を入力。		-	-
X2	-					
XT1	入 力	-	サブシステム・クロック発振用クリスタル接続端子。外部クロックの場合、XT1へ入力し、XT2へその逆相を入力。XT1は1ビット入力(テスト)端子として使用可能。		-	-
XT2	-					
$\overline{\text{RESET}}$	入 力	-	システム・リセット入力(ロウ・レベル・アクティブ)。		-	Ⓑ

注 印はシュミット・トリガ入力を示します。

表2 - 2 ポート端子以外の端子機能一覧(2/2)

端子名称	入出力	兼用端子	機 能	リセット時	入出力 回 路 TYPE
MD0-MD3	入 力	P30-P33	μ PD75P0076のみ内蔵。 プログラム・メモリ (PROM) 書き込み / ベリファイ 時のモード選択。	入 力	E-B
D0-D3	入出力	P40-P43	μ PD75P0076のみ内蔵。 プログラム・メモリ (PROM) 書き込み / ベリファイ 時のデータ・バス端子。	入 力	M-E
D4-D7		P50-P53			
IC	-	-	Internally Connected. $V_{DD}$ に直接接続してください。	-	-
$V_{PP}$	-	-	μ PD75P0076のみ内蔵。 プログラム・メモリ (PROM) 書き込み / ベリファイ 時のプログラム電圧印加。 通常動作時は $V_{DD}$ と直接接続。 PROM書き込み / ベリファイ時は + 12.5 Vを印加。	-	-
$V_{DD}$	-	-	正電源。	-	-
$V_{SS}$	-	-	グランド電位。	-	-

## 2.2 端子機能の説明

### 2.2.1 P00-P03 (PORT0) ...INT4, $\overline{SCK}$ , SO/SB0, SI/SB1兼用入力 P10-P13 (PORT1) ...INT0, INT1, TI1/INT2, TI0兼用入力 P110-P113 (PORT11) ...AN0-AN3兼用入力

4ビットの入力ポートです。入力ポートとしての機能のほかに、次のような機能を兼用しています。

ポート0：ベクタ割り込み入力 (INT4)

シリアル・インタフェース入出力 ( $\overline{SCK}$ , SO/SB0, SI/SB1)

ポート1：ベクタ割り込み入力 (INT0, INT1)

エッジ検出テスト入力 (INT2)

タイマ/イベント・カウンタへの外部イベント・パルス入力 (TI0, TI1)

ポート11：A/Dコンバータへのアナログ信号入力 (AN0-AN3)

ポート0は、シリアル・インタフェース機能使用時は、動作モードによって兼用端子が出力機能を持ちません。

ポート0およびポート1の各端子は、ノイズによる誤動作を防ぐためにシュミット・トリガ入力になっています。さらに、P10はノイズ除去回路を選択できます。(詳細は6.3(3) INT0, INT1, INT4のハードウェアを参照)。

ポート0は3ビット単位 (P01-P03) で、ポート1は4ビット単位 (P10-P13) で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定することができます。これらは、プルアップ抵抗指定レジスタ・グループA (POGA) を操作して指定します。

$\overline{RESET}$ 信号発生により、入力モードになります。

2.2.2 P20-P23 (PORT2) ...PTO0, PTO1, PCL, BUZ兼用入出力  
 P30-P33 (PORT3) ...MD0-MD3兼用入出力<sup>注</sup>  
 P40-P43 (PORT4) ...D0-D3兼用入出力<sup>注</sup>  
 P50-P53 (PORT5) ...N-chオープン・ドレーン中耐圧(13V) ,  
 D4-D7兼用入出力<sup>注</sup>  
 P60-P63 (PORT6) ...KR0-KR3, AN4-AN7兼用入出力

出力ラッチ付き4ビットの入出力ポートです。入出力ポートの機能のほかに、次のような機能を兼用しています。

ポート2	: タイマ/イベント・カウンタ出力 (PTO0, PTO1) クロック出力 (PCL) 任意の周波数出力 (BUZ)
ポート3	: プログラム・メモリ (PROM) 書き込み/ベリファイ時のモード選択 (MD0-MD3) <sup>注</sup>
ポート4	: プログラム・メモリ (PROM) 書き込み/ベリファイ時のデータ・バス (D0-D3) <sup>注</sup>
ポート5	: プログラム・メモリ (PROM) 書き込み/ベリファイ時のデータ・バス (D4-D7) <sup>注</sup>
ポート6	: キー割り込み入力 (KR0-KR3) A/Dコンバータへのアナログ信号入力 (AN4-AN7)

注  $\mu$ PD75P0076のみ兼用している機能です。

ポート4, 5はN-chオープン・ドレーン中耐圧(13V)です。

おのこのポートの入出力のモード選択はポート・モード・レジスタにより設定します。ポート2, 4, 5は4ビット単位で、ポート3, 6はビット単位で指定できます。

また、ポート2, 3, 6は、ソフトウェアにより内蔵プルアップ抵抗の接続を4ビット単位で指定可能で、プルアップ抵抗指定レジスタ・グループA (POGA) を操作して行います。ポート4, 5はマスク・オプションにより、ビット単位でプルアップ抵抗内蔵指定ができます。ただし、 $\mu$ PD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

ポート4, 5はペアで8ビット単位の入出力ができます。 $\overline{\text{RESET}}$ 信号発生により、ポート2, 3, 6は入力モード(ハイ・インピーダンス)に、ポート4, 5はハイ・レベル(マスク・オプションのプルアップ抵抗内蔵時)またはハイ・インピーダンスになります。

### 2.2.3 TIO, TI1...ポート1兼用入力

タイマ/イベント・カウンタ0, 1の外部イベント・パルス入力端子です。

タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1) で、カウント・パルス (CP) に外部イベント・パルス入力を選択することにより用いることができます。

TIO, TI1はシュミット・トリガ入力になっています。

詳細は5.5.1(1)タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1) を参照してください。

### 2.2.4 PTO0, PTO1...ポート2兼用出力

タイマ/イベント・カウンタ0, 1の出力端子です。方形波パルスを出力します。タイマ/イベント・カウンタの信号を出力する場合は、出力ラッチを“0”にクリアし、ポート・モード・レジスタのポート2のビットを出力モード“1”にセットします。

タイマ・スタート命令により、TOOUT F/Fの出力は“0”にクリアされます。

詳細は5.5.2(3)8ビット・タイマ/イベント・カウンタ動作を参照してください。

### 2.2.5 PCL...ポート2兼用出力

プログラマブル・クロック出力端子です。周辺LSI (スレーブのマイクロコンピュータなど) にクロックを供給するときに使用します。 $\overline{\text{RESET}}$ 信号発生により、クロック出力モード・レジスタ (CLOM) が“0”にクリアされクロックは出力禁止状態になり、通常のポートとしての動作モードになります。

詳細は5.2.4 クロック出力回路を参照してください。

### 2.2.6 BUZ...ポート2兼用出力

周波数出力端子です。任意の周波数 (2 kHz, 4 kHz, 32 kHz : メイン・システム・クロック4.19 MHz動作時、またはサブシステム・クロック32.768 kHz動作時) を出力することにより、ブザー音発生やシステム・クロック発振周波数のトリミングに使用します。P23端子と兼用となっており、時計モード・レジスタ (WM) のビット7 (WM7) が“1”にセットされているときのみ有効となります。

$\overline{\text{RESET}}$ 信号発生により、WM7は“0”にクリアされ、通常のポートとしての動作モードになります。

詳細は5.4.2 時計モード・レジスタを参照してください。

### 2.2.7 $\overline{\text{SCK}}$ , SO/SB0, SI/SB1...ポート0兼用入出力

シリアル・インタフェースのための入出力端子です。シリアル動作モード・レジスタ (CSIM) の設定に従って動作します。3線式シリアルI/Oモードを選択した場合、 $\overline{\text{SCK}}$ はCMOS入出力、SOはCMOS出力、SIはCMOS入力として機能します。また、2線式シリアルI/Oモードを選択した場合、 $\overline{\text{SCK}}$ はCMOS入出力、SB1 (SB0) はN-chオープン・ドレイン入出力として機能します。

$\overline{\text{RESET}}$ 信号発生によって、シリアル・インタフェース動作は停止し、入力ポートとなります。

いずれの端子とも、シュミット・トリガ入力になっています。

詳細は5.6 シリアル・インタフェースを参照してください。

### 2.2.8 INT4...ポート0 兼用入力

立ち上がりおよび立ち下りの両エッジ・アクティブの外部ベクタ割り込み入力端子です。この端子に入力された信号がロウ・レベルからハイ・レベルに変化したときと、ハイ・レベルからロウ・レベルに変化したときに割り込み要求フラグがセットされます。

INT4は非同期入力であり、CPUの動作クロックに関係なく一定のハイ・レベル幅またはロウ・レベル幅を持つ信号が入力されると受け付けられます。

INT4はSTOPモード、HALTモードの解除にも利用できます。また、シュミット・トリガ入力になっています。

### 2.2.9 INT0, INT1...ポート1 兼用入力

エッジ検出ベクタ割り込み入力端子で、INT0はノイズ除去回路を選択できます。エッジ検出モード・レジスタ(IM0, IM1)により、検出エッジの選択ができます。

#### (1) INT0 (IM0のビット0, 1)

- (a) 立ち上がりエッジ・アクティブ
- (b) 立ち下りエッジ・アクティブ
- (c) 立ち上がりおよび立ち下りの両エッジ・アクティブ
- (d) 外部割り込み信号入力禁止

#### (2) INT1 (IM1のビット0)

- (a) 立ち上がりエッジ・アクティブ
- (b) 立ち下りエッジ・アクティブ

INT0, INT1は非同期入力であり、CPUの動作クロックに関係なく一定のハイ・レベル幅を持つ入力があれば受け付けられます。また、INT0はソフトウェアにより、ノイズ除去回路付きにすることができ、ノイズ除去を行うサンプリング・クロックを2段階に変えることができます。この場合、CPU動作クロックによって受け付けられる信号の幅は異なります。

$\overline{\text{RESET}}$ 信号発生により、IM0, IM1は“0”にクリアされ、立ち上がりエッジ・アクティブが選択されます。

INT0, INT1ともに、STOPモード、HALTモードの解除に利用できます。ただし、ノイズ除去回路選択時には、INT0によるSTOPモード、HALTモードの解除はできません。

INT0, INT1は、シュミット・トリガ入力になっています。

### 2.2.10 INT2...ポート1 兼用入力

立ち上がりエッジ・アクティブの外部テスト入力端子です。エッジ検出モード・レジスタ(IM2)によってINT2が選択されているとき、この端子に入力された信号がロウ・レベルからハイ・レベルに変化したとき、内部テスト・フラグ(IRQ2)がセットされます。

INT2は非同期入力であり、CPUの動作クロックに関係なく一定のハイ・レベル幅を持つ入力があれば受け付けられます。

$\overline{\text{RESET}}$ 信号発生により、IM2は“0”にクリアされ、テスト・フラグ（IRQ2）は、INT2端子の立ち上がりエッジ入力によってセットされます。

INT2はSTOPモード、HALTモードの解除にも利用できます。また、シュミット・トリガ入力になっています。

### 2.2.11 KR0-KR3...ポート6兼用入力

キー割り込み入力端子です。KR0-KR3は、並列立ち下がりエッジ検出割り込み入力端子です。

エッジ検出モード・レジスタ（IM2）により、割り込みソースを“KR2とKR3”または“KR0-KR3”の2つから選択できます。。

$\overline{\text{RESET}}$ 信号発生により、ポート6の入力モードになります。

### 2.2.12 AN0-AN3...ポート11兼用入力 AN4-AN7...ポート6兼用入力

A/Dコンバータへの8本のアナログ信号入力端子です。

### 2.2.13 AVREF

A/Dコンバータの基準電圧供給端子です。

### 2.2.14 AVss

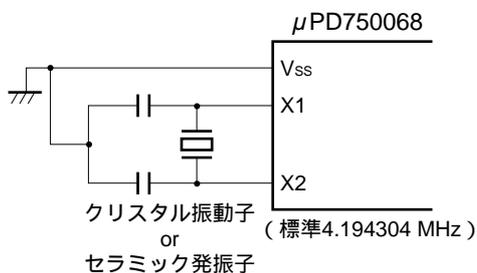
A/DコンバータのGND端子です。常にVssと同じ電位にしてください。

### 2.2.15 X1, X2

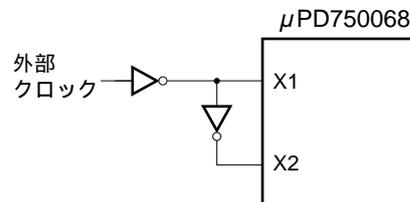
メイン・システム・クロック発振用クリスタル/セラミック接続端子です。

外部クロックを入力することも可能です。

(a) クリスタル/セラミック発振



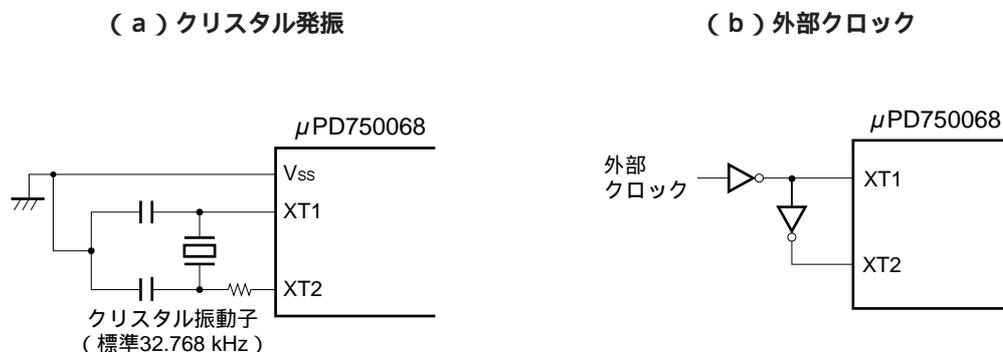
(b) 外部クロック



### 2.2.16 XT1, XT2

サブシステム・クロック発振用クリスタル接続端子です。

外部クロックを入力することも可能です。



**備考** サブシステム・クロックを使用しない場合については、5.2.2(6) サブ発振回路コントロール・レジスタ(SOS)を参照してください。

### 2.2.17 $\overline{\text{RESET}}$

ロウ・レベル・アクティブのリセット入力端子です。

$\overline{\text{RESET}}$ 入力は非同期入力であり、動作クロックに無関係に一定のロウ・レベル幅を持つ信号が入力されると、 $\overline{\text{RESET}}$ 信号が発生し、すべての動作に優先してシステム・リセットがかかります。

通常のCPUのイニシャライズ/スタートのほかに、STOPモード、HALTモードの解除にも使用します。

$\overline{\text{RESET}}$ 入力はシュミット・トリガ入力になっています。

### 2.2.18 MD0-MD3 (μPD75P0076のみ)

μPD75P0076のみMD0-MD3を内蔵しています。

プログラム・メモリ(ワン・タイムPROM)への書き込み/ベリファイ時のモードの選択端子です。

### 2.2.19 D0-D7 (μPD75P0076のみ)

μPD75P0076のみD0-D7を内蔵しています。

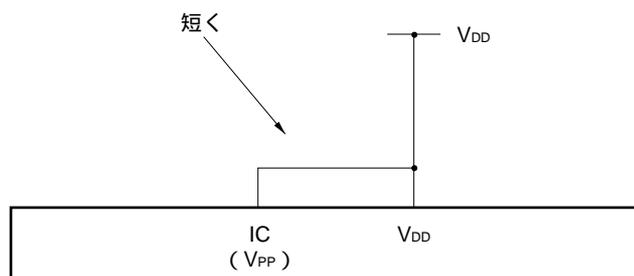
プログラム・メモリ(ワン・タイムPROM)への書き込み/ベリファイ時のデータ・バス端子です。

### 2.2.20 IC ( $\mu$ PD750064, 750066, 750068のみ )

IC ( Internally Connected ) 端子は、当社出荷時に  $\mu$ PD750068を検査するためのテスト・モードに設定するための端子です。通常動作時には、IC端子は $V_{DD}$ 端子に直接接続し、その配線長を極力短くしてください。

IC端子と $V_{DD}$ 端子間の配線の引き回しが長い場合や、IC端子に外来ノイズが加わった場合などで、IC端子と $V_{DD}$ 端子間に電位差が生じたときには、お客様のプログラムが正常に動作しないことがあります。

IC端子は $V_{DD}$ に直接接続してください。



### 2.2.21 $V_{PP}$ ( $\mu$ PD75P0076のみ )

プログラム・メモリ (ワン・タイムPROM) 書き込み / ベリファイ時のプログラム電圧入力端子です。

通常動作時は $V_{DD}$ に直接接続してください (上図参照)。PROMへの書き込み / ベリファイ時は12.5 Vを印加します。

### 2.2.22 $V_{DD}$

正電源供給端子です。

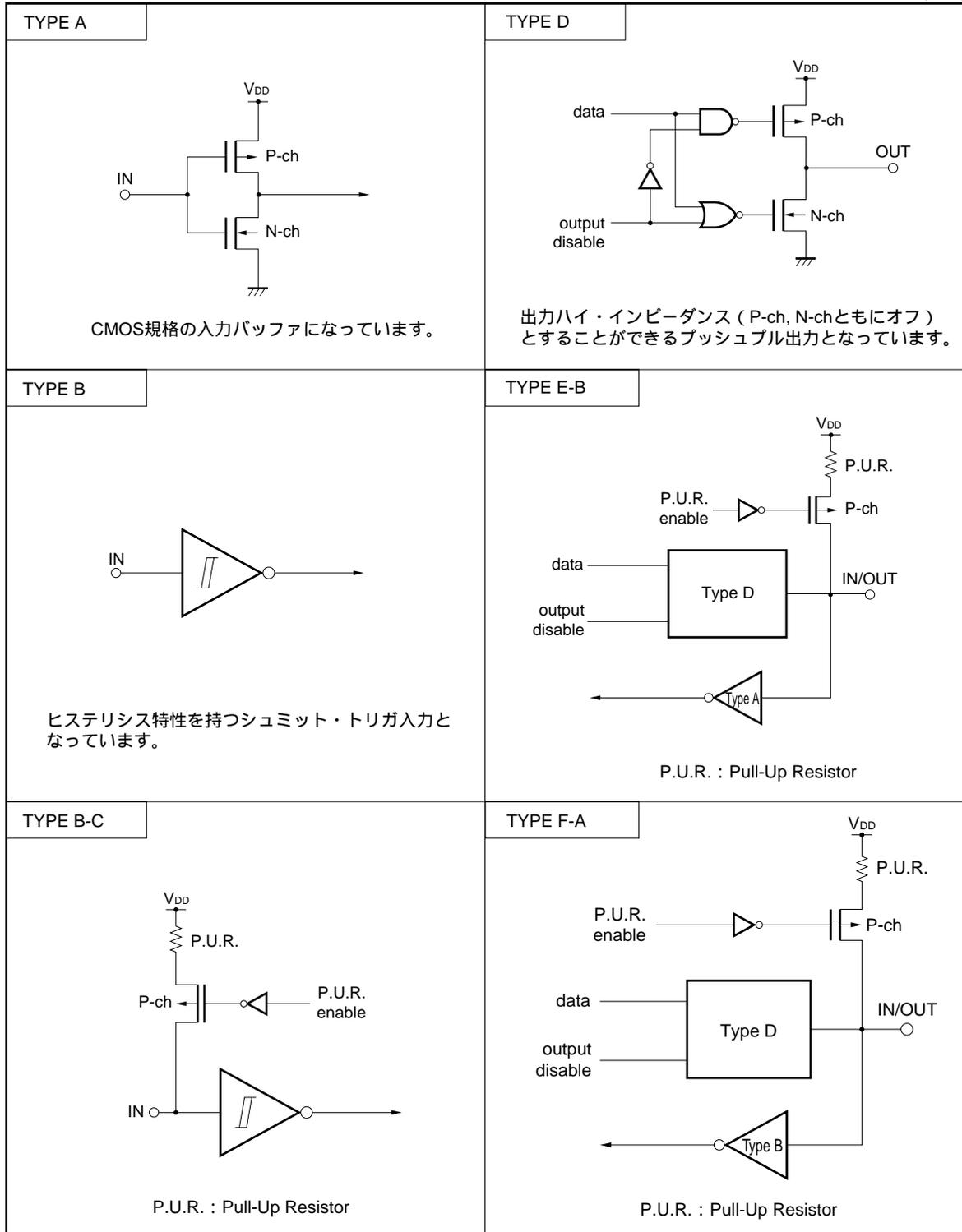
### 2.2.23 $V_{SS}$

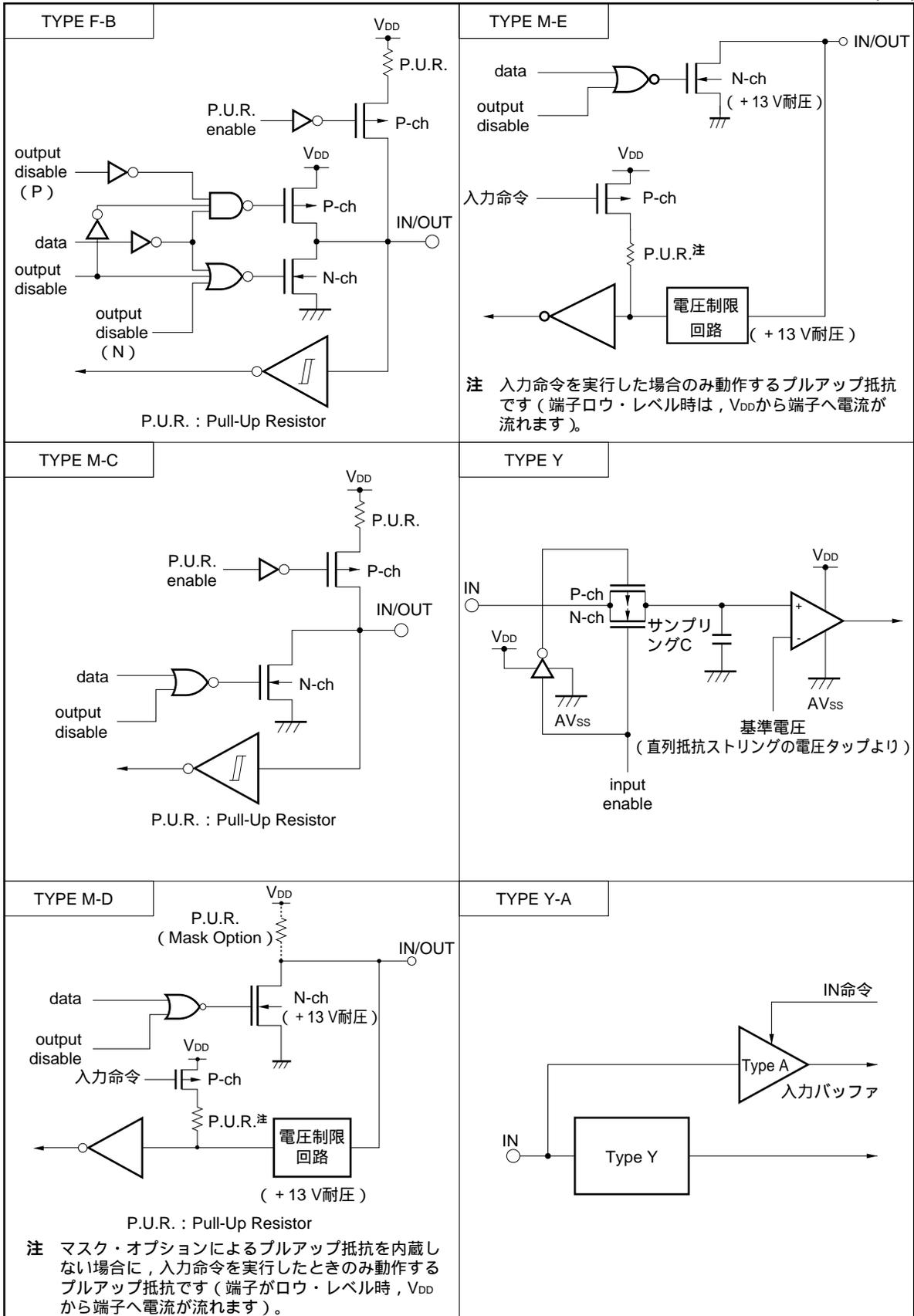
GND電位です。

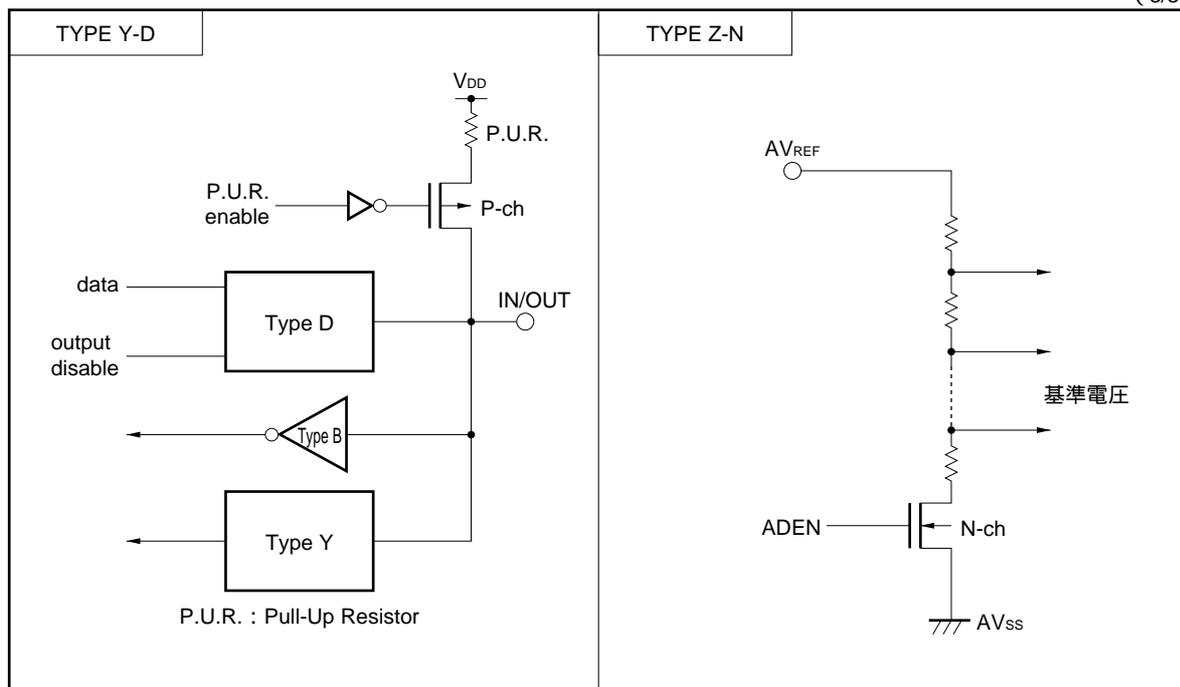
## 2.3 端子の入出力回路

μ PD750068の各端子の入出力回路を一部簡略した形式を用いて示します。

(1/3)







## 2.4 未使用端子の処理について

表2-3 未使用端子の処理一覧

端子	推奨接続方法
P00/INT4	V <sub>SS</sub> またはV <sub>DD</sub> に接続
P01/ $\overline{\text{SCK}}$	個別に抵抗を介して、V <sub>SS</sub> またはV <sub>DD</sub> に接続
P02/SO/SB0	
P03/SI/SB1	V <sub>SS</sub> に接続
P10/INT0, P11/INT1	V <sub>SS</sub> またはV <sub>DD</sub> に接続
P12/TI1/INT2	
P13/TI0	
P20/PTO0	入力状態：個別に抵抗を介して、V <sub>SS</sub> またはV <sub>DD</sub>
P21/PTO1	に接続
P22/PCL	出力状態：オープン
P23/BUZ	
P30 (/MD0) 注1	
P31 (/MD1) 注1	
P32 (/MD2) 注1	
P33 (/MD3) 注1	
P40 (/D0) 注1	V <sub>SS</sub> に接続 (マスク・オプションのプルアップ
P41 (/D1) 注1	抵抗を接続しないでください)
P42 (/D2) 注1	
P43 (/D3) 注1	
P50 (/D4) 注1	
P51 (/D5) 注1	
P52 (/D6) 注1	
P53 (/D7) 注1	
P60/KR0/AN4-P63/KR3/AN7	入力状態：個別に抵抗を介して、V <sub>SS</sub> またはV <sub>DD</sub> に接続 出力状態：オープン
P110/AN0-P113/AN3	V <sub>SS</sub> またはV <sub>DD</sub> に直接接続
AV <sub>REF</sub>	V <sub>SS</sub> に接続
AV <sub>SS</sub>	
XT1注2	V <sub>SS</sub> またはV <sub>DD</sub> に接続
XT2注2	オープン
IC (V <sub>PP</sub> ) 注1	V <sub>DD</sub> に直接接続してください。

注1．( )内はμPD75P0076のみの端子です。

- 2．サブシステム・クロックを使用しない場合は、SOS.0 = 1 (内蔵フィードバック抵抗を使用しない)を選択してください。

[メモ]

## 第3章 アーキテクチャの特徴とメモリ・マップ

μPD750068に採用されている75XLのアーキテクチャは以下の特徴を持っています。

- ・内蔵RAM最大4Kワード×4ビット(12ビット・アドレス)
- ・周辺ハードウェアの拡張性

この優れた特徴を実現するため、以下の方式を採用しています。

- (1) データ・メモリのバンク構成
- (2) 汎用レジスタのバンク構成
- (3) メモリ・マップI/O

本章では、これらのおのおのについて解説します。

### 3.1 データ・メモリのバンク構成とアドレッシング・モード

#### 3.1.1 データ・メモリのバンク構成

μPD750068ではデータ・メモリ空間の000H-1FFH番地にスタティックRAMが内蔵されています。また、周辺ハードウェア(入出力ポートやタイマなど)がF80H-FFFH番地に割り付けられています。

μPD750068は、12ビット・アドレス(4Kワード×4ビット)のデータ・メモリ空間をアドレッシングするために、下位8ビット・アドレスは命令によって直接あるいは間接に指定し、上位4ビット・アドレスをメモリ・バンクによって指定するメモリ・バンク構成となっています。

このメモリ・バンク(MB)を指定するために、以下のハードウェアが内蔵されています。

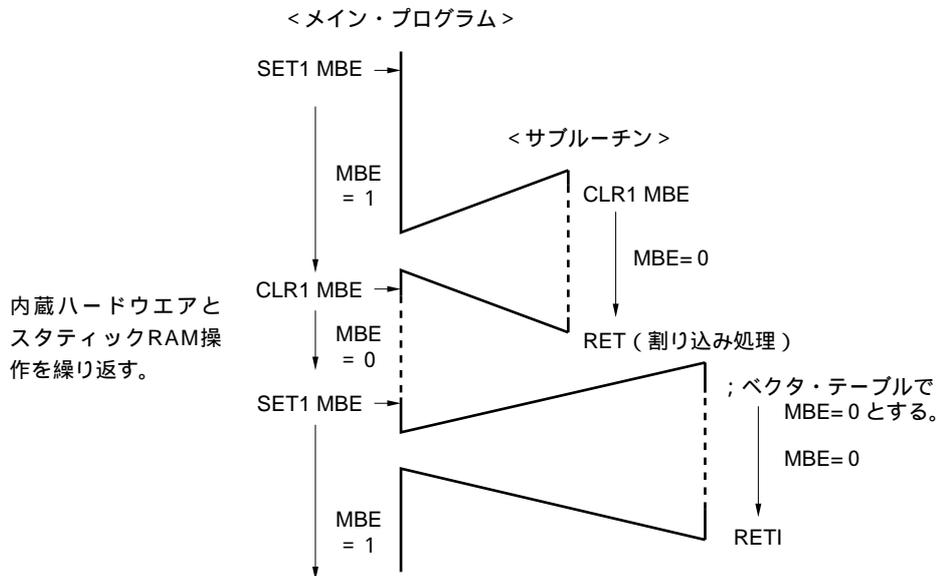
- ・メモリ・バンク許可フラグ(MBE)
- ・メモリ・バンク選択レジスタ(MBS)

MBSはメモリ・バンクを選択するレジスタで、メモリ・バンク0, 1, 15が設定できます。MBEはMBSで選択されたメモリ・バンクを有効とするか否かを決定するフラグです。図3-1に示すようにMBEが0の場合には指定されるメモリ・バンク(MB)はMBSにかかわらず固定されますが、MBEが1の場合にはMBSの設定によってメモリ・バンクを切り替え、データ・メモリ空間を拡張することができます。

データ・メモリ空間のアドレッシングにあたっては、通常はMBE=1とし、MBSで指定されたメモリ・バンクのデータ・メモリを操作しますが、プログラムの各処理でMBE=0のモードとMBE=1のモードを使い分けることにより、効率よくプログラムすることができます。

	適合するプログラムの処理	効果
MBE = 0 のモード	割り込み処理	MBSの退避 / 復帰が不要になる
	内蔵ハードウェア操作とスタティックRAM操作を繰り返す処理	MBSの変更が不要になる
	サブルーチン処理	MBSの退避 / 復帰が不要になる
MBE = 1 のモード	通常のプログラム処理	

図3 - 1 MBE = 0のモードとMBE = 1のモードの使い分け



備考 ——— MBE = 1 のとき, - - - - - MBE = 0 のとき。

MBEは、サブルーチン処理時に自動的に退避 / 復帰されますから、サブルーチン処理中でも自由に変更できます。また、割り込み処理時には、自動的に退避 / 復帰されるとともに、割り込みベクタ・テーブルの設定によって、割り込み処理開始と同時に割り込み処理中のMBEを指定することができます。したがって、高速の割り込み処理に有利です。

MBSをサブルーチン処理あるいは割り込み処理で変更する場合は、PUSH/POP命令によって退避 / 復帰します。

MBEの設定は、SET1命令 / CLR1命令で行います。MBSの設定は、SEL命令により行います。

例1 . MBEをクリアし、メモリ・バンクを固定する。

```
CLR1    MBE    ; MBE 0
```

2 . メモリ・バンク 1 を選択する。

```
SET1    MBE    ; MBE 1
```

```
SEL     MB1    ; MBS 1
```

### 3.1.2 データ・メモリのアドレッシング・モード

μ PD750068に採用されている75XLシリーズのアーキテクチャでは、データ・メモリ空間を、処理するデータのビット長ごとに効率よくアドレッシングするために、表3 - 1にまとめたように7種類のアドレッシング・モードが用意されており、効率のよいプログラミングが可能となります。

#### (1) 1ビット直接アドレッシング (mem.bit)

全データ・メモリ空間の各ビットを、命令のオペランドで直接に指定するアドレッシング・モードです。

指定されるメモリ・バンク (MB) は、MBE = 0 のモードでは、オペランドで指定されたアドレスが00H-7FHの場合MB = 0に、アドレスが80H-FFHの場合MB = 15に固定されます。したがって、MBE = 0のモードでは000H-07FHのデータ・エリアと、F80H-FFFFHの周辺ハードウェア・エリアの両方をアドレッシングすることができます。

MBE = 1のモードでは、MB = MBSとなり、指定できるデータ・メモリ空間を拡張することができます。

このアドレッシング・モードは、ビットのセット、リセット命令 (SET1/CLR1)、ビットのテスト命令 (SKT/SKF) の4つの命令に適用できます。

**例** FLAG1をセット、FLAG2をリセット、FLAG3が0であるかテストする。

```
FLAG1 EQU 03FH.1 ; 3FH番地ビット1
```

```
FLAG2 EQU 087H.2 ; 87H番地ビット2
```

```
FLAG3 EQU 0A7H.0 ; A7H番地ビット0
```

```
SET1 MBE ; MBE 1
```

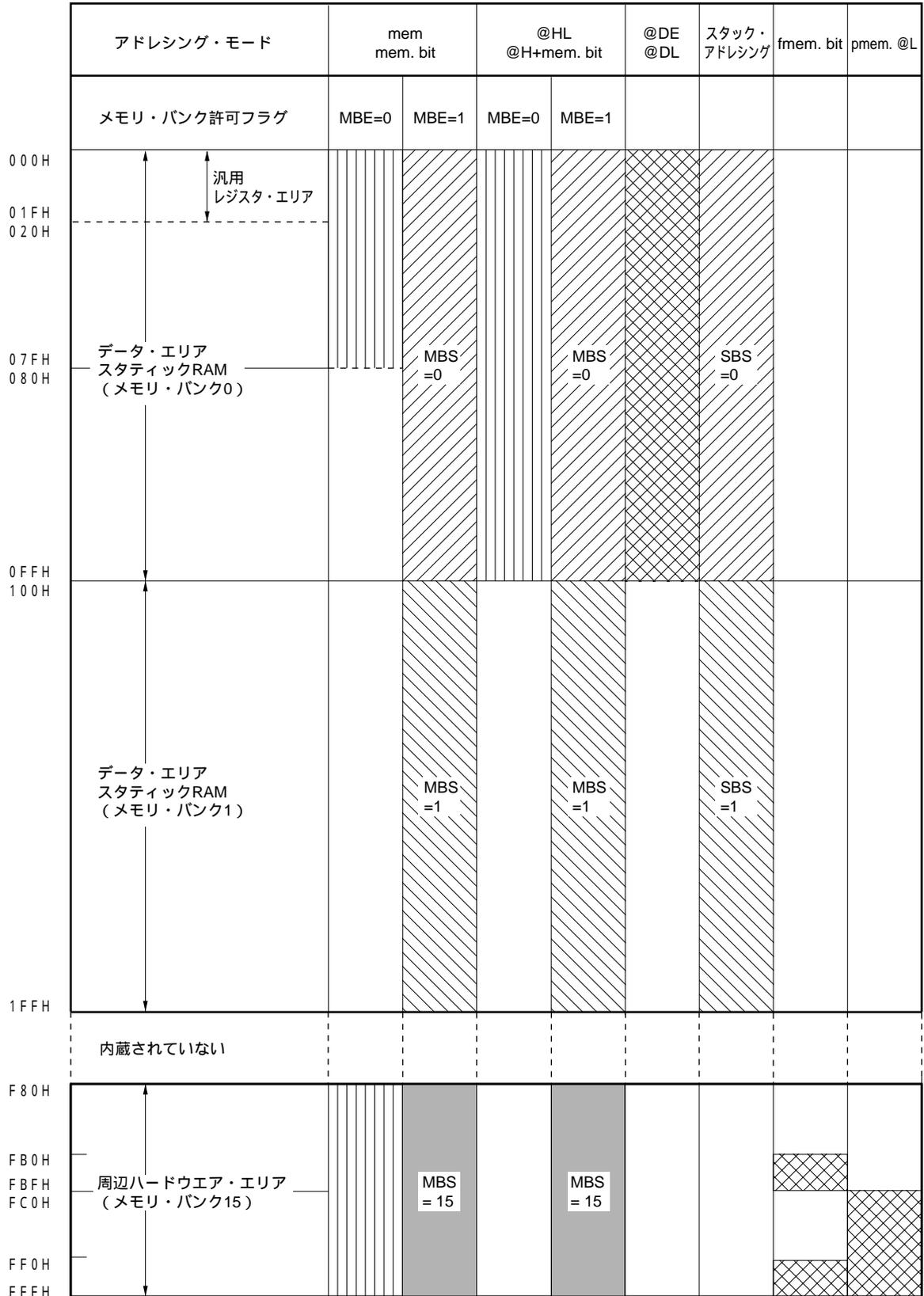
```
SEL MB0 ; MBS 0
```

```
SET1 FLAG1 ; FLAG1 1
```

```
CLR1 FLAG2 ; FLAG2 0
```

```
SKF FLAG3 ; FLAG3 = 0 ?
```

図3 - 2 データ・メモリの構成と、各アドレッシング・モードのアドレッシング範囲



備考 - : don't care

表3 - 1 アドレッシング・モード

アドレッシング・モード	表現形式	指定されるアドレス
1ビット 直接アドレッシング	mem.bit	MBとmemで示されるアドレスの, bitで示されるビット。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB=MBS
4ビット 直接アドレッシング	mem	MBとmemで示されるアドレス。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB = MBS
8ビット 直接アドレッシング		MBとmemで示されるアドレス (memは偶数アドレス)。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB = MBS
4ビット・レジスタ 間接アドレッシング	@HL	MBとHLで示されるアドレス。 ただし, MB = MBE・MBS
	@HL + @HL -	MBとHLで示されるアドレス。ただし, MB = MBE・MBS HL + は, アドレッシングのあとLレジスタを自動的にインクリメント。 HL - は, アドレッシングのあとLレジスタを自動的にデクリメント。
	@DE	メモリ・バンク0の, DEで示されるアドレス。
	@DL	メモリ・バンク0の, DLで示されるアドレス。
8ビット・レジスタ 間接アドレッシング	@HL	MBとHLで示されるアドレス (Lレジスタの内容は偶数)。 ただし, MB = MBE・MBS
ビット操作 アドレッシング	fmem.bit	fmemで示されるアドレスの, bitで示されるビット。 fmem = FB0H-FBFH (割り込み関係のハードウェア) FF0H-FFFH (I/Oポート)
	pmem.@L	pmemの上位10ビットとLレジスタの上位2ビットで示されるアドレスの, Lレジスタの下位2ビットで示されるビット。 ただし, pmem = FC0H-FFFH
	@H + mem.bit	MBとHそしてmemの下位4ビットで示されるアドレスの, bitで示される ビット。 ただし, MB = MBE・MBS
スタック・アドレッシング	-	SBSで選択されるメモリ・バンク0, 1の, SPで示されるアドレス。

## (2) 4ビット直接アドレッシング(mem)

全データ・メモリ空間を4ビット単位で、命令のオペランドによって直接に指定するアドレッシング・モードです。

1ビット直接アドレッシングと同様に、MBE = 0のモードでは、指定できる領域が、000H-07FHのデータ・エリアとF80H-FFFHの周辺ハードウェア・エリアに固定されます。MBE = 1のモードでは、MB = MBSとなって、指定できるデータ・メモリ空間は全空間に拡張されます。

このアドレッシング・モードは、MOV/XCH/INCS/IN/OUT命令に適用されます。

**注意** 例1のように、入出力ポートに関連するデータをバンク1のスタティックRAMに格納するとプログラムの効率が悪くなってしまいます。入出力ポートと関連したデータはバンク0の00H-7FH番地に格納するようにすれば、例2のようにMBSを変更せずにプログラムすることができます。

例1. 「BUFF」のデータをポート5に出力する。

```

BUFF EQU 11AH      ; 「BUFF」は11AH番地
SET1 MBE          ; MBE 1
SEL MB1          ; MBS 1
MOV A, BUFF      ; A (BUFF)
SEL MB15         ; MBS 15
OUT PORT5, A    ; PORT5 A

```

2. ポート4を入力し、「DATA1」に格納する。

```

DATA1 EQU 5FH     ; 「DATA1」は5FH番地
CLR1 MBE          ; MBE 0
IN A, PORT4      ; A PORT4
MOV DATA1, A    ; (DATA1) A

```

**(3) 8ビット直接アドレッシング (mem)**

全データ・メモリ空間を8ビット単位で、命令のオペランドによって直接に指定するアドレッシング・モードです。

オペランドで指定できるアドレスは偶数アドレスであり、オペランドで指定されたアドレスの4ビット・データと、プラス1されたアドレスの4ビット・データがペアとなって、8ビット・アキュムレータ (XAレジスタ・ペア) との間で8ビット処理されます。

指定されるメモリ・バンクは、4ビット直接アドレッシングと同じです。

このアドレッシング・モードは、MOV/XCH/IN/OUT命令に適用されます。

**例1** .ポート4, 5の8ビット・データを20H, 21H番地に転送する。

```
DATA EQU 020H
      CLR1 MBE ; MBE 0
      IN   XA, PORT4 ; X   ポート5 , A   ポート4
      MOV  DATA, XA ; (21H) X , (20H) A
```

**2** .シリアル・インタフェースのシフト・レジスタ (SIO) に入力された8ビット・データを取り込むと同時に、転送データをセットし、転送スタート指示する。

```
SEL  MB15 ; MBS 15
XCH  XA, SIO ; XA (SIO)
```

## (4) 4ビット・レジスタ間接アドレッシング (@rpa)

データ・メモリ空間を命令のオペランドで指定されたデータ・ポインタ（汎用レジスタのレジスタ・ペア）によって4ビット単位で間接に指定するアドレッシング・モードです。

データ・ポインタとしては、MB = MBE・MBSの指定によって全データ・メモリ空間を指定できるHLレジスタ・ペアとMBE、MBSの指定にかかわらず常にメモリ・バンク0に固定されているDEレジスタ・ペア/DLレジスタ・ペアの3種類があります。使用するデータ・メモリのバンクによってデータ・ポインタを使い分けることによって効率のよいプログラムが可能となります。

HLレジスタ・ペアを指定した場合には、命令実行と同時にLレジスタを+1または-1するオートインクリメント/オートデクリメント・モードがあり、プログラム・ステップの短縮につながります。

**例** 50H-57Hのデータを110H-117Hに転送する。

```

DATA1 EQU 57H
DATA2 EQU 117H
SET1 MBE
SEL MB1
MOV D, #DATA1 SHR 4
MOV HL, #DATA2 AND 0FFH ; HL 17H
LOOP: MOV A, @DL ; A (DL)
XCH A, @HL ; A (HL)
DECS L ; L L - 1
BR LOOP

```

データ・ポインタとしてHLレジスタ・ペアを使用するアドレッシング・モードは、データの転送、演算、比較、入出力等に幅広く適用されます。DE、DLレジスタ・ペアを使用するアドレッシング・モードは、MOV/XCH命令に適用されます。

また、汎用レジスタあるいはレジスタ・ペアの増減命令と組み合わせることによって、図3-3に示されるようにデータ・メモリ空間のアドレスを自由自在に更新することができます。

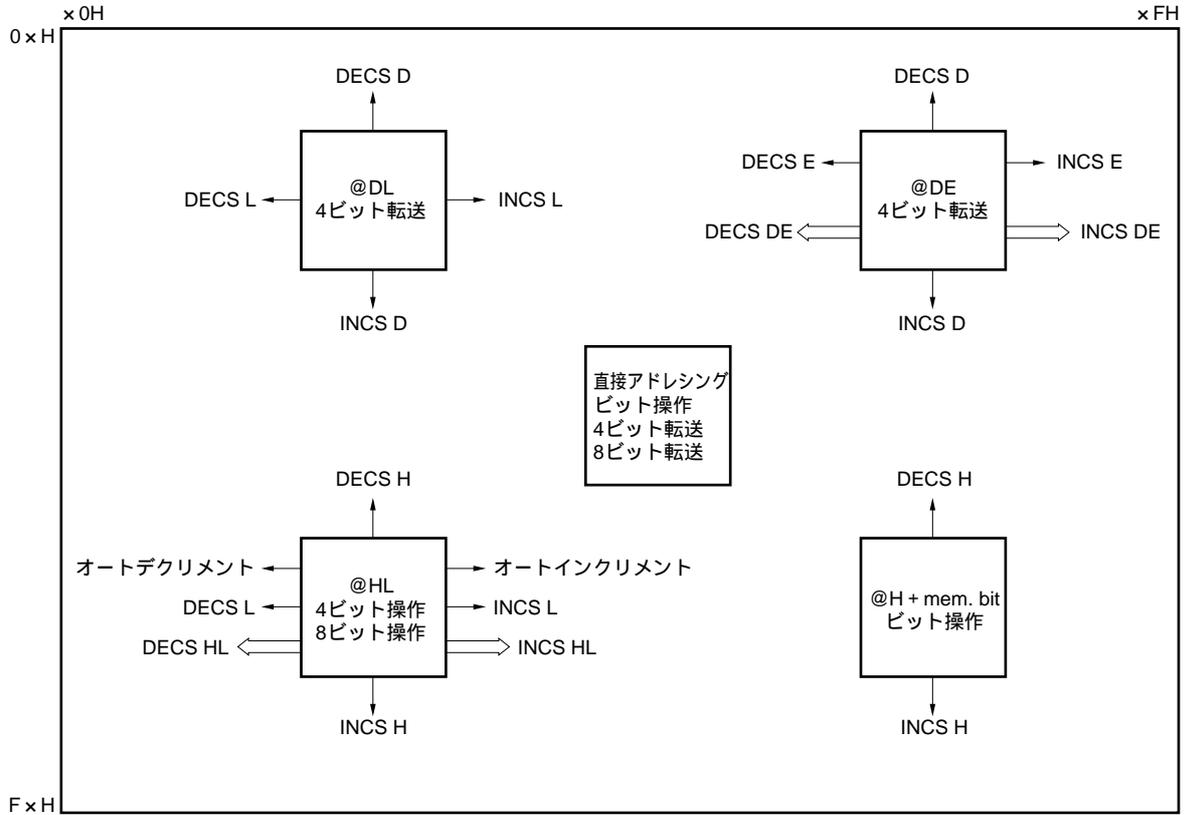
例1 . 50H-57Hのデータと110H-117Hのデータを比較する。

```
DATA1 EQU 57H
DATA2 EQU 117H
SET1 MBE
SEL MB1
MOV D, #DATA1 SHR4
MOV HL, #DATA2 AND 0FFH
LOOP: MOV A, @DL
      SKE A, @HL ; A = (HL) ?
      BR NO ; NO
      DECS L ; YES, L = L - 1
      BR LOOP
```

2 . 00H-FFHのデータ・メモリを0クリアする。

```
CLR1 RBE
CLR1 MBE
MOV XA, #00H
MOV HL, #04H
LOOP: MOV @HL, A ; (HL) = A
      INCS L ; L = L + 1
      BR LOOP
      INCS H ; H = H + 1
      BR LOOP
```

図3 - 3 スタティックRAMのアドレス更新方法



## (5) 8ビット・レジスタ間接アドレッシング (@HL)

全データ・メモリ空間を8ビット単位で、データ・ポインタ(HLレジスタ・ペア)により間接に指定するアドレッシング・モードです。

データ・ポインタのビット0(Lレジスタのビット0)を0にしたアドレスの4ビット・データとプラス1されたアドレスの4ビット・データがペアとなって、8ビット・アキュムレータ(XAレジスタ)との間で、8ビット処理されます。

指定されるメモリ・バンクは4ビット・レジスタ間接アドレッシングでHLレジスタを指定した場合と同じで、MB = MBE・MBSとなります。このアドレッシング・モードは、MOV/XCH/SKE命令に適用されません。

**例1** . タイマ/イベント・カウンタ0のカウンタ・レジスタ(T0)値が30H, 31H番地のデータと等しいかどうか比較する。

```

DATA    EQU    30H
        CLR1   MBE
        MOV    HL, #DATA
        MOV    XA, T0      ; XA カウンタ・レジスタ 0
        SKE   A, @HL      ; A = (HL) ?
        BR    NO
        INCS  L
        MOV    A, X        ; A = X
        SKE   A, @HL      ; A = (HL) ?

```

**2** . 00H-FFHのデータ・メモリを0クリアする。

```

        CLR1   RBE
        CLR1   MBE
        MOV    XA, #00H
        MOV    HL, #04H
LOOP:   MOV    @HL, A      ; (HL) = A
        INCS  L
        BR    LOOP
        INCS  H
        BR    LOOP

```

(6) ビット操作アドレッシング

全データ・メモリ空間の各ビットに対し、ビット操作（ブーリアン処理、ビット転送等）を行うためのアドレッシング・モードです。

1ビット直接アドレッシングがビットのセット/リセット/テスト命令にしか適用できないのに対し、このアドレッシングではAND1/OR1/XOR1命令によるブーリアン処理、SKTCLR命令によるテスト&リセットなど、多彩なビット操作が可能になります。

ビット操作アドレッシングには、次に示す3つの方法があり、使用するデータ・メモリ・アドレスによって使い分けることができます。

(a) 特定アドレス・ビット直接アドレッシング (fmem.bit)

このアドレッシング・モードは、周辺ハードウェアのうち入出力ポートや割り込み関係のフラグなど、特にビット操作をよく使うハードウェアをメモリ・バンクの設定に無関係にいつでも操作できるようにするためのアドレッシングです。したがって、このアドレッシングを適用できるデータ・メモリ・アドレスは、入出力ポートをマッピングしてあるFF0H-FFFHと、割り込み関係のハードウェアをマッピングしてあるFB0H-FBFHとなっています。この2つのデータ・メモリ領域にあるハードウェアは、MBSおよびMBEの設定にかかわらず、いつでも自由に直接アドレッシングでビット操作することができます。

例1. タイマ0 割り込み要求フラグ (IRQT0) をテストし、セットされていたら要求フラグをクリアし、P63をリセットする。

```
SKTCLR  IRQT0    ; IRQT0=1?
BR      NO      ; NO
CLR1    PORT6.3 ; YES
```

2. P30とP41の端子がともに1であれば、P53をリセットする。



```
( i )   SET1  CY          ; CY  1
        AND1  CY, PORT3.0 ; CY  P30
        AND1  CY, PORT4.1 ; CY  P41
        SKT   CY          ; CY=1?
        BR    SETP
        CLR1  PORT5.3     ; P53  0
        ⋮
        SETP : SET1  PORT5.3 ; P53  1
```

```

        :
(ii)   SKT   PORT3.0      ; P30=1?
        BR    SETP
        SKT   PORT4.1      ; P41=1?
        BR    SETP
        CLR1  PORT5.3      ; P53  0
        :
        SETP : SET1  PORT5.3      ; P53  1
    
```

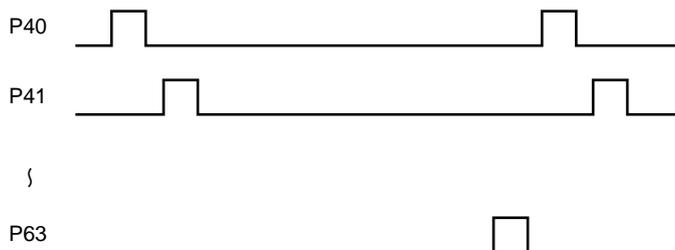
**(b) 特定アドレス・ビット・レジスタ間接アドレッシング (pmem.@L)**

このアドレッシング・モードは周辺ハードウェアのうち入出力ポート等の各ビットをレジスタ間接指定し、連続的に操作するためのアドレッシングです。このアドレッシングを適用できるデータ・メモリ・アドレスはFC0H-FFFHとなっています。

このアドレッシング・モードは、データ・メモリ・アドレス12ビットの上位10ビット・アドレスをオペランドで直接指定し、下位2ビット・アドレスとビット・アドレスはLレジスタによって間接指定します。したがって、Lレジスタの指定によって16ビット（4ポート）を連続的に操作できます。

このアドレッシングも、MBE、MBSの設定に無関係にいつでもビット操作することができます。

**例** ポート4からポート6の各ビットに順にパルス出力する。



```

LOOP2 : MOV   L, #0
LOOP1 : SET1  PORT4. @L    ; ポート4-6のビット (L1-0)  1
        CLR1  PORT4. @L    ; ポート4-6のビット (L1-0)  0
        INCS  L
        NOP
        SKE   L, #0CH
        BR    LOOP1
        BR    LOOP2
    
```

(c) 特殊1ビット直接アドレッシング (@H+mem.bit)

このアドレッシング・モードは、全データ・メモリ空間の各ビットに対し、ビット操作を可能にするためのアドレッシングです。

このアドレッシングでは、MB=MBE・MBSで指定されるメモリ・バンクのデータ・メモリ・アドレスの上位4ビット・アドレスをHレジスタで間接指定し、下位4ビット・アドレスとビット・アドレスはオペランドで直接指定します。このアドレッシングによってすべてのデータ・メモリの各ビットに対し多彩なビット操作が可能となります。

**例** 30H番地のビット3 (FLAG1) と31H番地のビット0 (FLAG2) がともに0かともに1なら32H番地のビット2 (FLAG3) をリセットする。



```

FLAG1 EQU 30H.3
FLAG2 EQU 31H.0
FLAG3 EQU 32H.2
SEL MB0
MOV H, #FLAG1 SHR 6
CLR1 CY ; CY 0
OR1 CY, @H+FLAG1 ; CY CY FLAG1
XOR1 CY, @H+FLAG2 ; CY CY ∨ FLAG2
SET1 @H+FLAG3 ; FLAG3 1
SKT CY ; CY=1?
CLR1 @H+FLAG3 ; FLAG3 0
  
```

**(7) スタック・アドレッシング**

このアドレッシング・モードは、割り込み処理時、サブルーチン処理時の退避 / 復帰動作のためのアドレッシングです。

このアドレッシングでは、データ・メモリ・バンク 0 の、スタック・ポインタ (8 ビット) で示されるアドレスが指定されます。

このアドレッシングは、割り込み処理時、サブルーチン処理時のほか PUSH/POP 命令によるレジスタの退避 / 復帰にも適用されます。

**例 1 .** サブルーチン処理でレジスタを退避 / 復帰する。

```
SUB :  PUSH  XA
      PUSH  HL
      PUSH  BS ; MBSとRBSの退避
      :
      POP   BS
      POP   HL
      POP   XA
      RET
```

**2 .** HLレジスタ・ペアの内容をDEレジスタ・ペアに転送する。

```
PUSH  HL
POP   DE ; DE HL
```

**3 .** [XABC] のレジスタで示されるアドレスへ分岐する。

```
PUSH  BC
PUSH  XA
RET           ; XABC番地へ分岐
```

### 3.2 汎用レジスタのバンク構成

μPD750068は、X、A、B、C、D、E、H、Lの8つの汎用レジスタを1バンクとして、4つのレジスタ・バンクを内蔵しています。この汎用レジスタ・エリアはデータ・メモリのメモリ・バンク0の00H-1FH番地にマッピングされています(図3-5 汎用レジスタの構成(4ビット処理の場合)参照)。この汎用レジスタのバンクを指定するためにレジスタ・バンク許可フラグ(RBE)とレジスタ・バンク選択レジスタ(RBS)が内蔵されています。RBSはレジスタ・バンクを選択するためのレジスタで、RBEはRBSで選択されたレジスタ・バンクを有効とするか否かを決定するフラグです。命令実行の際に有効となるレジスタ・バンク(RB)は、次のようになります。

$$RB = RBE \cdot RBS$$

表3-2 RBE, RBSと選択されるレジスタ・バンク

RBE	RBS				レジスタ・バンク
	3	2	1	0	
0	0	0	x	x	バンク0に固定
1	0	0	0	0	バンク0を選択
			0	1	バンク1 "
			1	0	バンク2 "
			1	1	バンク3 "

└─ 0に固定

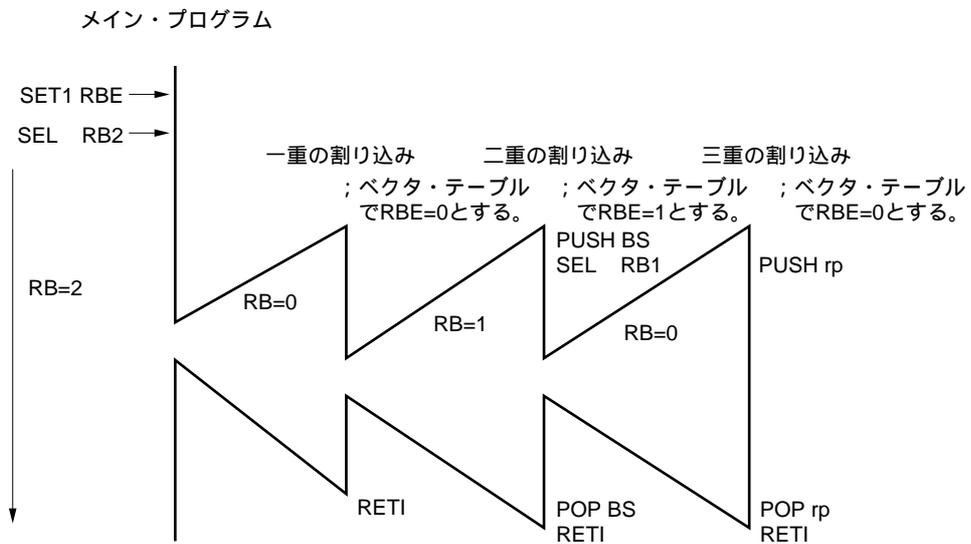
備考 x : don't care

RBEは、サブルーチン処理時に自動的に退避/復帰されますので、サブルーチン処理中に自由に設定することができます。また、割り込み処理時は、自動的に退避/復帰されるとともに、割り込みベクタ・テーブルの設定によって、割り込み処理開始と同時に、割り込み処理中のRBEを設定することができます。したがって、表3-3に示すように、通常処理と割り込み処理で、レジスタ・バンクを使い分ければ、一重割り込みでは、汎用レジスタの退避/復帰は不要、二重割り込みでは、RBSの退避/復帰のみとなり、割り込み処理の高速化が図れます。

表3-3 通常ルーチンと割り込みルーチンでのレジスタ・バンクの使い分けの例

通常の処理	RBE=1とし、レジスタ・バンク2、3を使用する。
1重割り込み処理	RBE=0とし、レジスタ・バンク0を使用する。
2重割り込み処理	RBE=1とし、レジスタ・バンク1を使用する。 (このとき、RBSの退避/復帰が必要)
3重以上の割り込み処理	PUSH, POPでレジスタ退避をする。

図3 - 4 レジスタ・バンクの使い分け例



RBSを、サブルーチン処理あるいは割り込み処理で変更する場合は、PUSH/POP命令によって退避 / 復帰します。

RBEの設定は、SET1/CLR1命令によって行います。RBSの設定は、SEL命令により行います。

例 SET1 RBE ;RBE 1  
CLR1 RBE ;RBE 0  
SEL RB0 ;RBS 0  
SEL RB3 ;RBS 3

μPD750068に内蔵されている汎用レジスタ・エリアは、4ビット・レジスタとしての使用のほかに、レジスタ・ペアによる8ビット・レジスタとして使用ができ、8ビット・マイコンに匹敵する転送、演算、比較、増減命令によって、汎用レジスタ中心のプログラミングが可能となります。

(1) 4ビット・レジスタとして使用する場合

汎用レジスタ・エリアを4ビット・レジスタとして使用する場合には、図3-5に示すように、RB=RBE・RBSで指定されたレジスタ・バンクのX, A, B, C, D, E, H, L, 計8個の汎用レジスタを使うことができます。このうちAレジスタは4ビット・アキュムレータとして、4ビット・データの転送、演算、比較等に中心的な働きをします。ほかの汎用レジスタは、アキュムレータとの転送、比較、増減ができます。

(2) 8ビット・レジスタとして使用する場合

汎用レジスタ・エリアを8ビット・レジスタとして使用する場合には、図3-6に示すように、RB=RBE・RBSで指定されたレジスタ・バンクのレジスタ・ペアをXA, BC, DE, HLとし、レジスタ・バンク(RB)のビット0を反転したレジスタ・バンクのレジスタ・ペアをXA', BC', DE', HL'として計8個の8ビット・レジスタを使うことができます。このうちXAレジスタ・ペアは、8ビット・アキュムレータとして、8ビット・データの転送、演算・比較等に中心的な働きをします。ほかのレジスタ・ペアは、アキュムレータとの転送、演算、比較、増減ができます。また、HLレジスタ・ペアは、主にデータ・ポインタとして機能します。DE, DLレジスタ・ペアも補助的なデータ・ポインタとして機能します。

```

例1 .INCS HL ; HL HL + 1, HL=00Hでスキップ
      ADDS XA, BC ; XA XA + BC, キャリーでスキップ
      SUBC DE', XA ; DE' DE' - XA - CY
      MOV XA, XA' ; XA XA'
      MOVT XA, @PCDE ; XA (PC12-8 + DE) ROM, テーブル参照
      SKE XA, BC ; XA=BCならスキップ
    
```

2. タイマ/イベント・カウンタ0のカウンタ・レジスタ(T0)の値がBC'レジスタ・ペアの値より大きいかどうかをテストし大きくなるまで待つ。

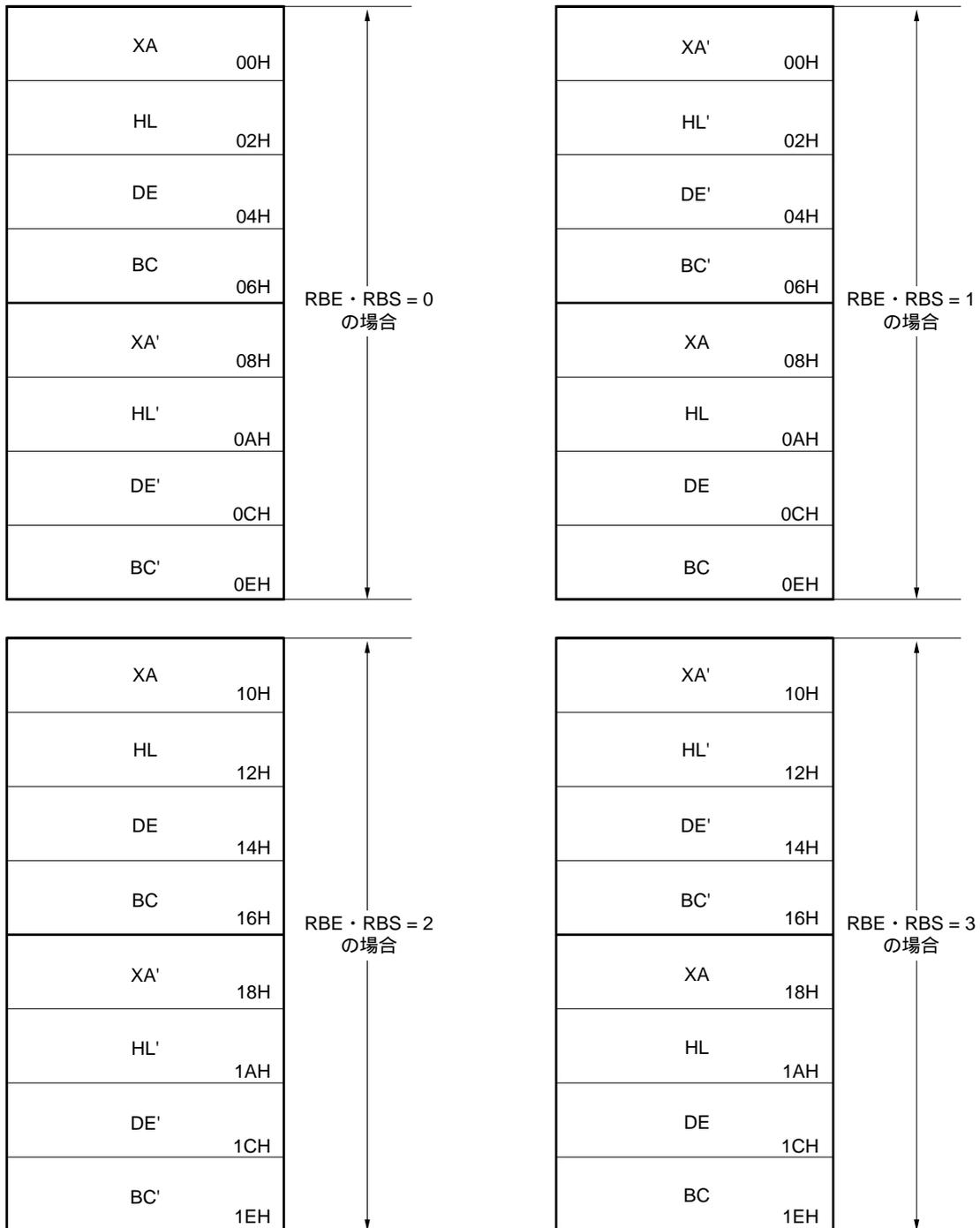
```

      CLR1 MBE
NO:   MOV XA, T0 ; カウンタ・レジスタ読み取り
      SUBS XA, BC' ; XA BC'?
      BR YES ; YES
      BR NO ; NO
    
```

図3 - 5 汎用レジスタの構成 (4ビット処理の場合)

X	01H	A	00H	レジスタ・バンク0 (RBE・RBS = 0)
H	03H	L	02H	
D	05H	E	04H	
B	07H	C	06H	
X	09H	A	08H	レジスタ・バンク1 (RBE・RBS = 1)
H	0BH	L	0AH	
D	0DH	E	0CH	
B	0FH	C	0EH	
X	11H	A	10H	レジスタ・バンク2 (RBE・RBS = 2)
H	13H	L	12H	
D	15H	E	14H	
B	17H	C	16H	
X	19H	A	18H	レジスタ・バンク3 (RBE・RBS = 3)
H	1BH	L	1AH	
D	1DH	E	1CH	
B	1FH	C	1EH	

図3 - 6 汎用レジスタの構成 (8ビット処理の場合)



### 3.3 メモリ・マップトI/O

μPD750068は、図3 - 2 に示したようにデータ・メモリ空間上のF80H-FFFH番地に入出力ポートやタイマなどの周辺ハードウェアをマッピングしたメモリ・マップトI/Oを採用しています。このため、周辺ハードウェアを制御する特別な命令はなく、すべてメモリ操作命令により制御します（一部、プログラムをわかりやすくするためのハードウェア制御の二モニックが用意してあります）。

周辺ハードウェアを操作する場合は表3 - 4のようなアドレッシング・モードが使用できます。

表3 - 4 周辺ハードウェア操作時に適用可能なアドレッシング・モード

	適用可能なアドレッシング・モード	適用可能なハードウェア
ビット操作	MBE = 0, または ( MBE = 1, MBS = 15 ) として 直接アドレッシングmem.bitで指定	ビット操作可能なすべての ハードウェア
	MBE, MBSにかかわらず, 直接アドレッシングfmem.bitで指定	IST1, IST0, MBE, RBE IE x x x , IRQ x x x , PORTn. x
	MBE, MBSにかかわらず, 間接アドレッシングpmem.@Lで指定	BSBn. x PORTn. x
4ビット操作	MBE = 0, または ( MBE = 1, MBS = 15 ) として 直接アドレッシングmemで指定	4ビット操作可能なすべての ハードウェア
	( MBE = 1, MBS = 15 ) として レジスタ間接アドレッシング@HLで指定	
8ビット操作	MBE = 0, または ( MBE = 1, MBS = 15 ) として直接アドレ ッシングmemで指定。ただし, memは偶数アドレス	8ビット操作可能なすべての ハードウェア
	MBE = 1, MBS = 15としてレジスタ間接アドレッシング@HLで 指定。ただし, Lレジスタの内容は偶数	

例 CLR1 MBE ; MBE=0  
 SET1 TM0.3 ; タイマ0のスタート  
 EI IE0 ; INT0の許可  
 DI IE1 ; INT1の禁止  
 SKTCLR IRQ2 ; INT2要求フラグのテスト・アンド・クリア  
 SET1 PORT4.@L ; ポート4のセット  
 IN A, PORT0 ; A ポート0  
 OUT PORT4, XA ; ポート5, 4 XA

図3 - 7に $\mu$ PD750068のI/Oマップをまとめます。

同図における項目の意味は次のようになっています。

略号.....内蔵されたハードウェアのアドレスを示すネームです。

命令のオペランド欄に記述できます。

R/W.....該当するハードウェアが、読み出し / 書き込み可能かどうかを示します。

R/W : 読み出し (Read) / 書き込み (Write) 可能

R : 読み出し (Read) のみ

W : 書き込み (Write) のみ

操作可能なビット数.....該当するハードウェアを操作する場合に、適用可能な処理ビット数を示します。

: その欄で使用している単位 (1/4/8ビット) でのビットの操作が可能。

: 一部のビットのみ操作が可能。操作の可能なビットについては備考欄を参照してください。

- : その欄で使用している単位 (1/4/8ビット) でのビットの操作が不可能。

ビット操作アドレッシング.....該当するハードウェアをビット操作する場合に、適用可能なビット操作アドレッシングを示します。

図3 - 7 μPD750068 I/Oマップ (1/5)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
F80H	スタック・ポインタ (SP)				R/W	-	-		-	ビット0は0に固定
F82H	レジスタ・バンク選択レジスタ (RBS)				R	-			-	注1
F83H	バンク選択レジスタ (BS)					-				
F84H	メモリ・バンク選択レジスタ (MBS)									
F84H	スタック・バンク選択レジスタ (SBS)				R/W	-		-	-	
F85H	ベーシック・インターバル・タイマ・モード・レジスタ (BTM)				W			-	mem.bit	ビット3のみ ビット操作可能
F86H	ベーシック・インターバル・タイマ (BT)				R	-	-		-	
F8BH	WDTM <sup>注2</sup>				W		-	-	mem.bit	ビット3のみ ビット操作可能

注1 . 4ビット操作では、RBSおよびMBSとして別々に操作可能。

8ビット操作では、BSとして操作可能。

MBS, RBSへの書き込みは、それぞれSEL MBn, SEL RBn命令で行ってください。

2 . WDTM : ウォッチドッグ・タイマ許可フラグ (W) ; 1度セットすると命令ではクリアできません。

図3 - 7 μPD750068 I/Oマップ (2/5)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
F98H	時計モード・レジスタ (WM)				R/W	(R)	-		mem.bit	ビット3のみ ビット操作可能
F99H						-	-		-	

FA0H	タイマ/イベント・カウンタ0モード・ レジスタ (TM0)				R/W	(W)	-		mem.bit	ビット3のみ ビット操作可能
						-	-		-	
FA2H	TOE0 <sup>注1</sup>				W		-	-	mem.bit	
FA4H	タイマ/イベント・カウンタ0カウント・ レジスタ (T0)				R	-	-		-	
FA6H	タイマ/イベント・カウンタ0モジュロ・ レジスタ (TMOD0)				R/W	-	-		-	
FA8H	タイマ/イベント・カウンタ1モード・ レジスタ (TM1)				R/W	(W)	-		mem.bit	ビット3のみ ビット操作可能
						-	-		-	
FAAH	TOE1 <sup>注2</sup>				W		-	-	mem.bit	
FACH	タイマ/イベント・カウンタ1カウント・ レジスタ (T1)				R	-	-		-	
FAEH	タイマ/イベント・カウンタ1モジュロ・ レジスタ (TMOD1)				R/W	-	-		-	

注1 . TOE0 : タイマ/イベント・カウンタ0出力許可フラグ (W)

2 . TOE1 : タイマ/イベント・カウンタ1出力許可フラグ (W)

図3 - 7 μPD750068 I/Oマップ (3/5)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
FB0H	IST1	IST0	MBE	RBE	R/W	(R/W)	(R/W)	(R)	fmem.bit	8ビット操作は Rのみ可能
	プログラム・ステータス・ワード (PSW)					注2	-			
	CY <sup>注1</sup>	SK2 <sup>注1</sup>	SK1 <sup>注1</sup>	SK0 <sup>注1</sup>						
FB2H	割り込みプライオリティ選択レジスタ (IPS)				R/W	-		-		注3
FB3H	プロセッサ・クロック・コントロール・レジスタ (PCC)				R/W	-		-		注4
FB4H	INT0エッジ検出モード・レジスタ (IM0)				R/W	-		-	-	
FB5H	INT1エッジ検出モード・レジスタ (IM1)				R/W	-		-	-	ビット0のみ ビット操作可能
FB6H	INT2エッジ検出モード・レジスタ (IM2)				R/W	-		-	-	ビット0, 1のみ ビット操作可能
FB7H	システム・クロック・コントロール・レジスタ (SCC)				R/W	(R/W)	(R)	-	-	ビット0, 3のみ ビット操作可能
FB8H	INTAレジスタ (INTA) IE4 IRQ4 IEBT IRQB				R/W			-	fmem.bit	
FBAH	INTCレジスタ (INTC) IEW IRQW				R/W			-		
FBCH	INTEレジスタ (INTE) IET1 IRQT1 IET0 IRQT0				R/W			-		
FBDH	INTFレジスタ (INTF) IECSI IRQCSI				R/W			-		
FBEH	INTGレジスタ (INTG) IE1 IRQ1 IE0 IRQ0				R/W			-		
FBFH	INTHレジスタ (INTH) IE2 IRQ2				R/W			-		

FC0H	ビット・シーケンシャル・バッファ0 (BSB0)	R/W				mem.bit	
FC1H	" 1 (BSB1)	R/W				pmem.@L	
FC2H	" 2 (BSB2)	R/W					
FC3H	" 3 (BSB3)	R/W					
FCFH	サブ発振回路コントロール・レジスタ (SOS)	R/W	-		-	-	

備考1 . IE x x x は割り込み許可フラグ

2 . IRQ x x x は割り込み要求フラグ

注1 . 予約語としては登録されていません。

2 . CYへの書き込みは、CY操作命令で行ってください。

3 . ビット3のみEI/DI命令による操作可能。

4 . ビット3, 2はSTOP/HALT命令実行時ビット操作可能。

図3-7 μPD750068 I/Oマップ(4/5)

アドレス	ハードウェア名称(略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
FD0H	クロック出力モード・レジスタ(CLOM)				R/W	-		-	-	
FD8H	SOC	EOC	-	-	R/W		-		mem.bit	EOC...R SOC, ADEN...W
	A/D変換モード・レジスタ(ADM)				R/W		-			
	ADEN	ADM6 <sup>注</sup>	ADM5 <sup>注</sup>	ADM4 <sup>注</sup>	R/W		-			
FDAH	SAレジスタ(SA)				R	-	-			
FDCH	PO3 <sup>注</sup>	PO2 <sup>注</sup>	PO1 <sup>注</sup>	PO0 <sup>注</sup>	R/W	-	-		-	
	プルアップ抵抗指定レジスタ・グループA(POGA)									
	-	PO6 <sup>注</sup>	-	-						

FE0H	CSIM3 <sup>注</sup>	CSIM2 <sup>注</sup>	CSIM1 <sup>注</sup>	CSIM0 <sup>注</sup>	W	-	-		-	
	シリアル動作モード・レジスタ(CSIM)									
	CSIE			CSIM4 <sup>注</sup>						
FE2H			CMDT	RELT	W		-	-	mem.bit	
	SBIコントロール・レジスタ(SBIC)									
FE4H	シリアルI/Oシフト・レジスタ(SIO)				R/W	-	-		-	
FE8H	PM33 <sup>注</sup>	PM32 <sup>注</sup>	PM31 <sup>注</sup>	PM30 <sup>注</sup>	R/W	-	-		-	
	ポート・モード・レジスタ・グループA(PMGA)									
	PM63 <sup>注</sup>	PM62 <sup>注</sup>	PM61 <sup>注</sup>	PM60 <sup>注</sup>						
FECH	-	PM2 <sup>注</sup>	-	-	R/W	-	-		-	
	ポート・モード・レジスタ・グループB(PMGB)									
	-	-	PM5 <sup>注</sup>	PM4 <sup>注</sup>						

注 予約語としては登録されていません。

図3 - 7 μPD750068 I/Oマップ (5/5)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
FF0H	ポート0			SCKP (PORT0)	R			-	fmem.bit	注1
FF1H	ポート1 (PORT1)				R				pmem.@L	
FF2H	ポート2 (PORT2)				R/W			-		
FF3H	ポート3 (PORT3)				R/W					
FF4H	ポート4 (PORT4)				R/W					
FF5H	ポート5 (PORT5)				R/W					
FF6H <sup>注2</sup>	KR3	KR2	KR1	KR0 ポート6 (PORT6)	R/W			-		
FFBH	ポート11 (PORT11)				R					

注1．ビット1はシリアル動作許可モード時のみR/W可能。ビット0，2，3と4ビット操作時はRのみ可能。

2．KR0-KR3は1ビット単位での読み出し (Read) のみ可能です。

(メモ)

## 第4章 内部CPU機能

### 4.1 Mk モードとMk モードの切り替え機能

#### 4.1.1 Mk モードとMk モードの違い

μPD750068のCPUはMk モードとMk モードの2つのモードを持ち、どちらを使用するかを選択ができます。モードの切り替え操作は、スタック・バンク選択レジスタ(SBS)のビット3で行います。

- ・Mk モード：μPD75068と上位互換性があります。  
ROM容量が16 Kバイトまでの75XL CPUで使用できます。
- ・Mk モード：μPD75068との互換性がありません。  
ROM容量が16 Kバイト以上の製品も含め、75XL CPU全部で使用できます。

表4-1 Mk モードとMk モードの違い

	Mk モード	Mk モード
サブルーチン命令の スタック・バイト数	2バイト	3バイト
BRA !addr1命令 CALLA !addr1命令	なし	あり
CALL !addr命令	3マシン・サイクル	4マシン・サイクル
CALLF !faddr命令	2マシン・サイクル	3マシン・サイクル

**注意** Mk モードは、75Xおよび75XLシリーズにおいて、16 Kバイトを越えるプログラム領域をサポートするモードです。このモードにより、16 Kバイトを越える製品とのソフトウェア上の互換性を高めることができます。

なお、Mk モードを選択すると、サブルーチン・コール命令実行時のスタック・バイト数がMk モードに比べ1スタックごとに1バイト分使用エリアが増えます。また、CALL !addr、CALLF !faddr命令使用時は、マシン・サイクルがそれぞれ1マシン・サイクル分長くかかります。したがって、ソフトウェアの互換性よりRAMの使用効率や処理能力を重視する場合は、Mk モードを使用してください。

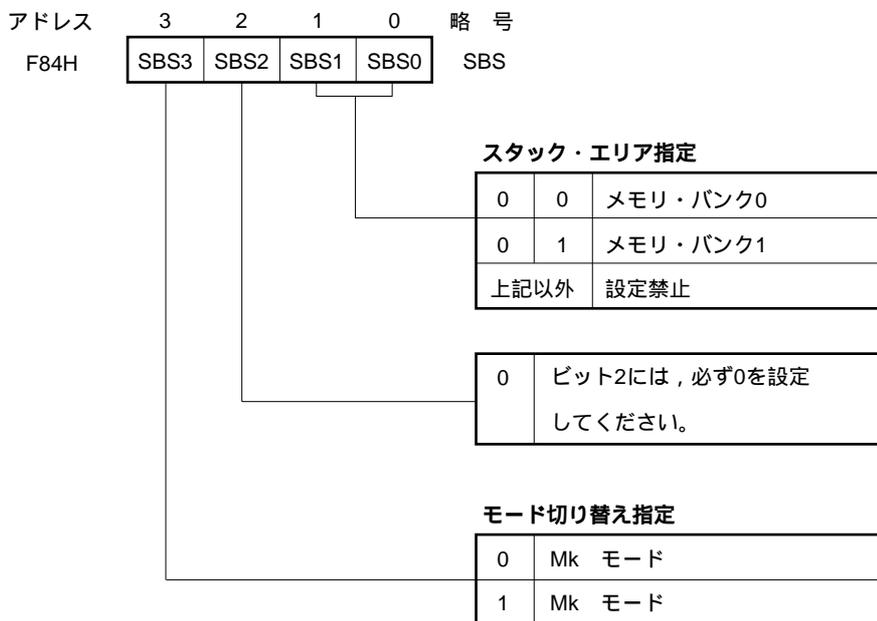
### 4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法

Mk モードとMk モードの切り替えは、スタック・バンク選択レジスタによって行います。図4 - 1にそのフォーマットを示します。

スタック・バンク選択レジスタは、4ビット・メモリ操作命令により設定します。Mk モードを使用する場合は、プログラムの初期で必ずスタック・バンク選択レジスタを100×B<sup>※</sup>にイニシャライズしてください。またMk モードを使用する場合は、必ず000×B<sup>※</sup>にイニシャライズしてください。

注 ※には希望の値を設定してください。

図4 - 1 スタック・バンク選択レジスタのフォーマット



注意 SBS.3は $\overline{\text{RESET}}$ 信号発生後“1”になるので、CPUはMk モードで動作します。Mk モードの命令を使用する場合は、SBS.3を“0”にし、Mk モードに設定してから使用してください。

## 4.2 プログラム・カウンタ (PC)

.....12ビット (  $\mu$ PD750064 )

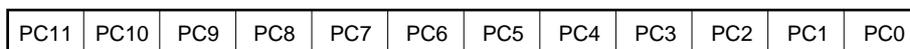
13ビット (  $\mu$ PD750066 , 750068 )

14ビット (  $\mu$ PD75P0076 )

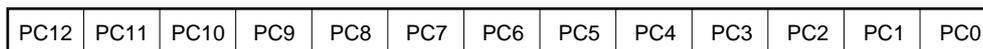
プログラム・メモリのアドレスを保持するバイナリ・カウンタです。

図4 - 2 プログラム・カウンタの構成

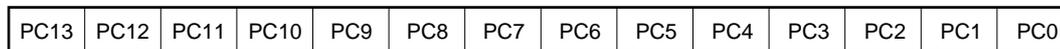
(a)  $\mu$ PD750064



(b)  $\mu$ PD750066, 750068



(c)  $\mu$ PD75P0076



プログラム・カウンタは、通常、命令を1つ実行するごとにその命令のバイト数に応じて自動的にインクリメントされます。

分岐命令 (BR, BRA, BRCB) 実行時には、分岐先アドレスを示すイミディエト・データやレジスタ・ペアの内容がPCの全部または一部のビットにロードされます。

サブルーチン・コール命令 (CALL, CALLA, CALLF) 実行時およびベクタ割り込み発生時には、そのときのPC内容 (すでに次の命令をフェッチするためにインクリメントされているリターン・アドレス) がスタック・メモリ (スタック・ポインタにより指定されたデータ・メモリ) に退避されたあと、それぞれのジャンプ先のアドレスがロードされます。

リターン命令 (RET, RETS, RETI) 実行時には、スタック・メモリの内容がPCにセットされます。

RESET信号発生により、プログラム・カウンタ (PC) の内容はプログラム・メモリの0000H, 0001H番地の内容でイニシャライズされ、その内容により任意のアドレスからプログラムをスタートできます。

$\mu$ PD750064 : PC<sub>11-8</sub> (0000H)<sub>b-0</sub>, PC<sub>7-0</sub> (0001H)<sub>f-0</sub>  
 $\mu$ PD750066, 750068 : PC<sub>12-8</sub> (0000H)<sub>f-0</sub>, PC<sub>7-0</sub> (0001H)<sub>f-0</sub>  
 $\mu$ PD75P0076 : PC<sub>13-8</sub> (0000H)<sub>b-0</sub>, PC<sub>7-0</sub> (0001H)<sub>f-0</sub>

### 4.3 プログラム・メモリ (ROM)

.....	4096 × 8 ビット ( μPD750064 )
.....	6144 × 8 ビット ( μPD750066 )
.....	8192 × 8 ビット ( μPD750068 )
.....	16384 × 8 ビット ( μPD75P0076 )

プログラム・メモリは、プログラム、割り込みベクタ・テーブル、GETI命令の参照テーブルおよびテーブル・データなどを格納するメモリです。

プログラム・メモリは、プログラム・カウンタによってアドレスされます。また、テーブル参照命令 (MOVT) によってテーブル・データを参照することができます。

分岐命令、サブルーチン・コール命令による分岐可能アドレス範囲は図4-3に示すとおりです。相対分岐命令 (BR \$addr命令) ではブロック境界に関係なく、[ PCの内容 - 15 ~ - 1, + 2 ~ + 16 ] アドレスへ分岐できます。

プログラム・メモリのアドレスは次の範囲です。

- 0000H-0FFFH : μPD750064
- 0000H-17FFH : μPD750066
- 0000H-1FFFH : μPD750068
- 0000H-3FFFH : μPD75P0076

次に示すアドレスには特別な機能を割り付けています。また、0000H, 0001Hを除くすべての領域は、通常のプログラム・メモリとして使用できます。

0000H, 0001H番地

RESET信号発生でのプログラム・スタート・アドレスとRBE, MBEの設定値を書くベクタ・テーブルです。任意の番地からリセット・スタートできます。

0002H-000DH番地

各ベクタ割り込みによるプログラム・スタート・アドレスとRBE, MBEの設定値を書くベクタ・テーブルです。任意の番地から割り込み処理をスタートできます。

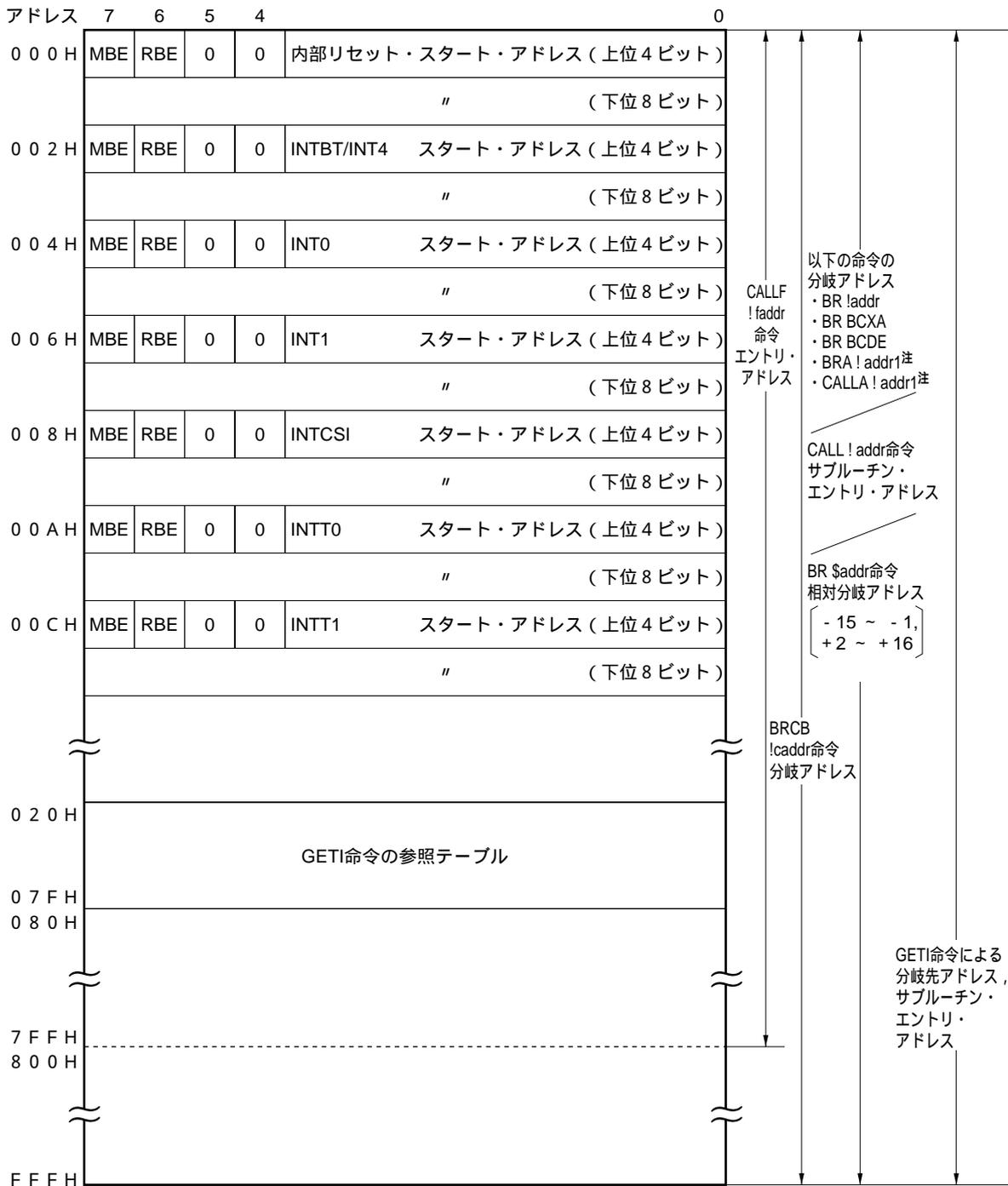
0020H-007FH番地

GETI命令<sup>注</sup>で参照するテーブル領域です。

**注** GETI命令は、任意の2バイト/3バイト命令、または1バイト命令2つを1バイトで実現するための命令で、プログラム・ステップ数を縮小するために使用します (11.1.1 GETI命令参照)。

図4 - 3 プログラム・メモリ・マップ (1/4)

( a ) μPD750064

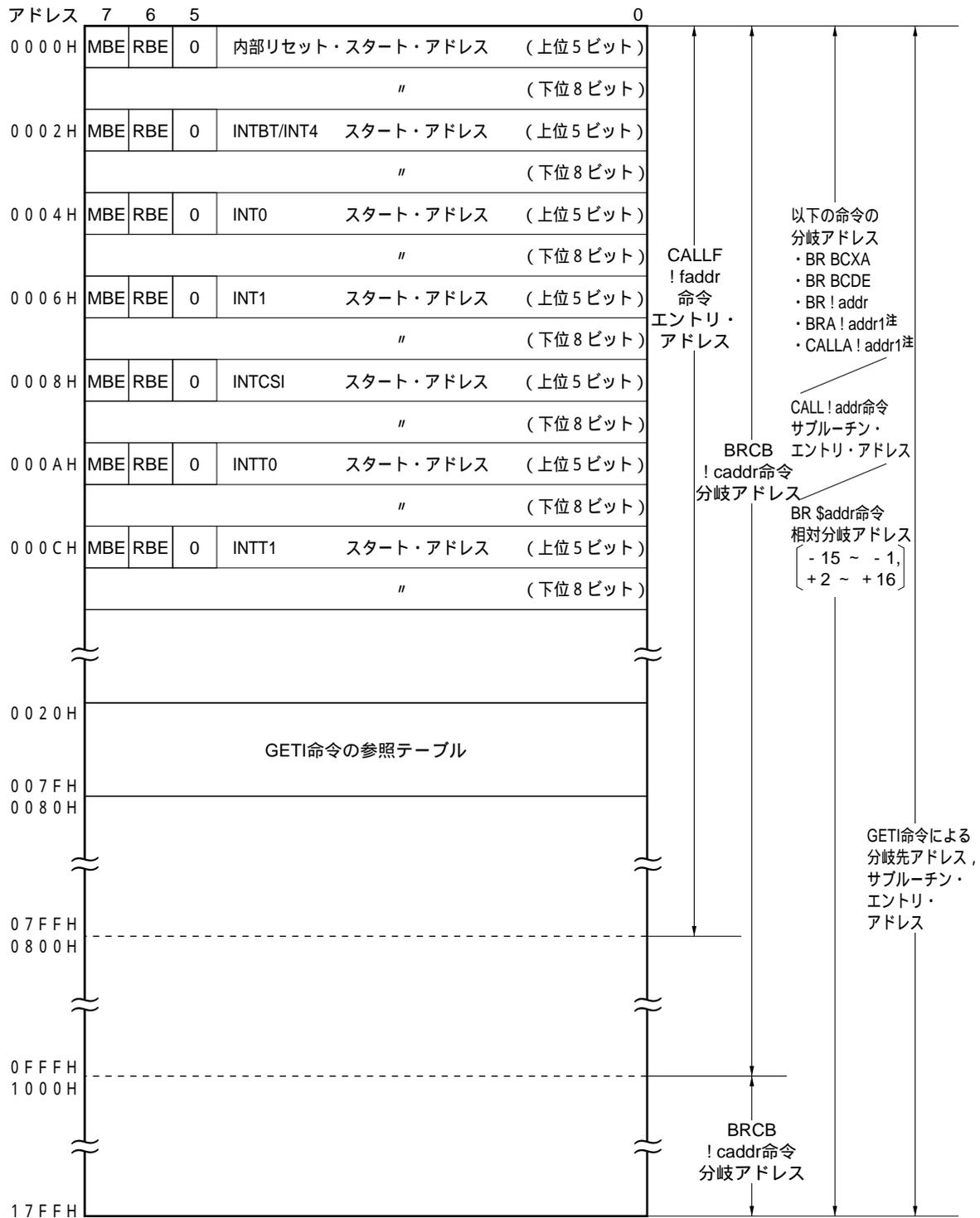


注 Mk モードでのみ使用できます。

備考 上記の命令以外では, BR PCDE, BR PCXA命令により, PCの下位8ビットのみ変更したアドレスへ分岐することができます。

図4 - 3 プログラム・メモリ・マップ (2/4)

(b)  $\mu$ PD750066

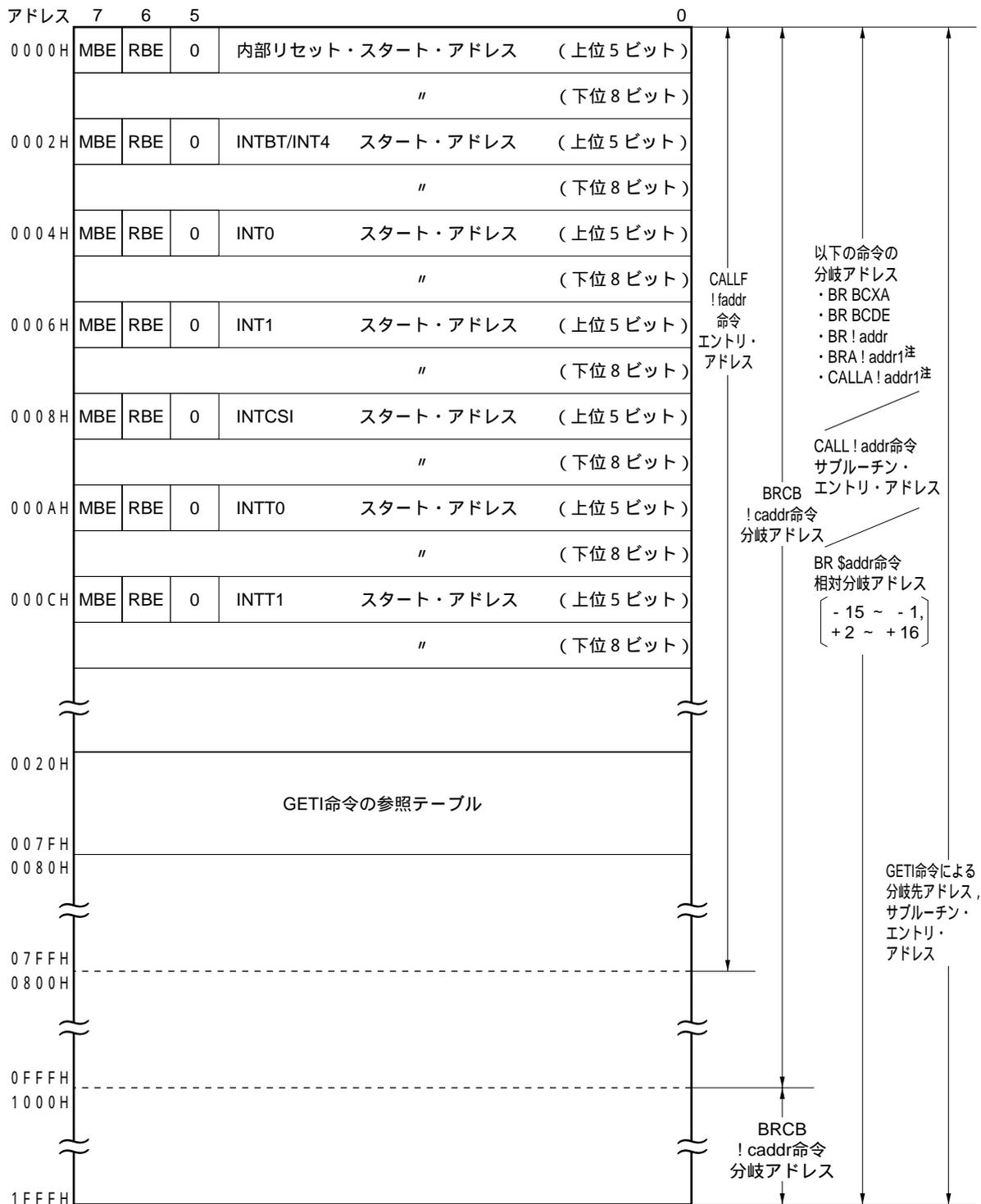


注 Mk モードでのみ使用できます。

備考 上記の命令以外では、BR PCDE、BR PCXA命令により、PCの下位8ビットのみ変更したアドレスへ分岐することができます。

図4-3 プログラム・メモリ・マップ(3/4)

(c) μPD750068

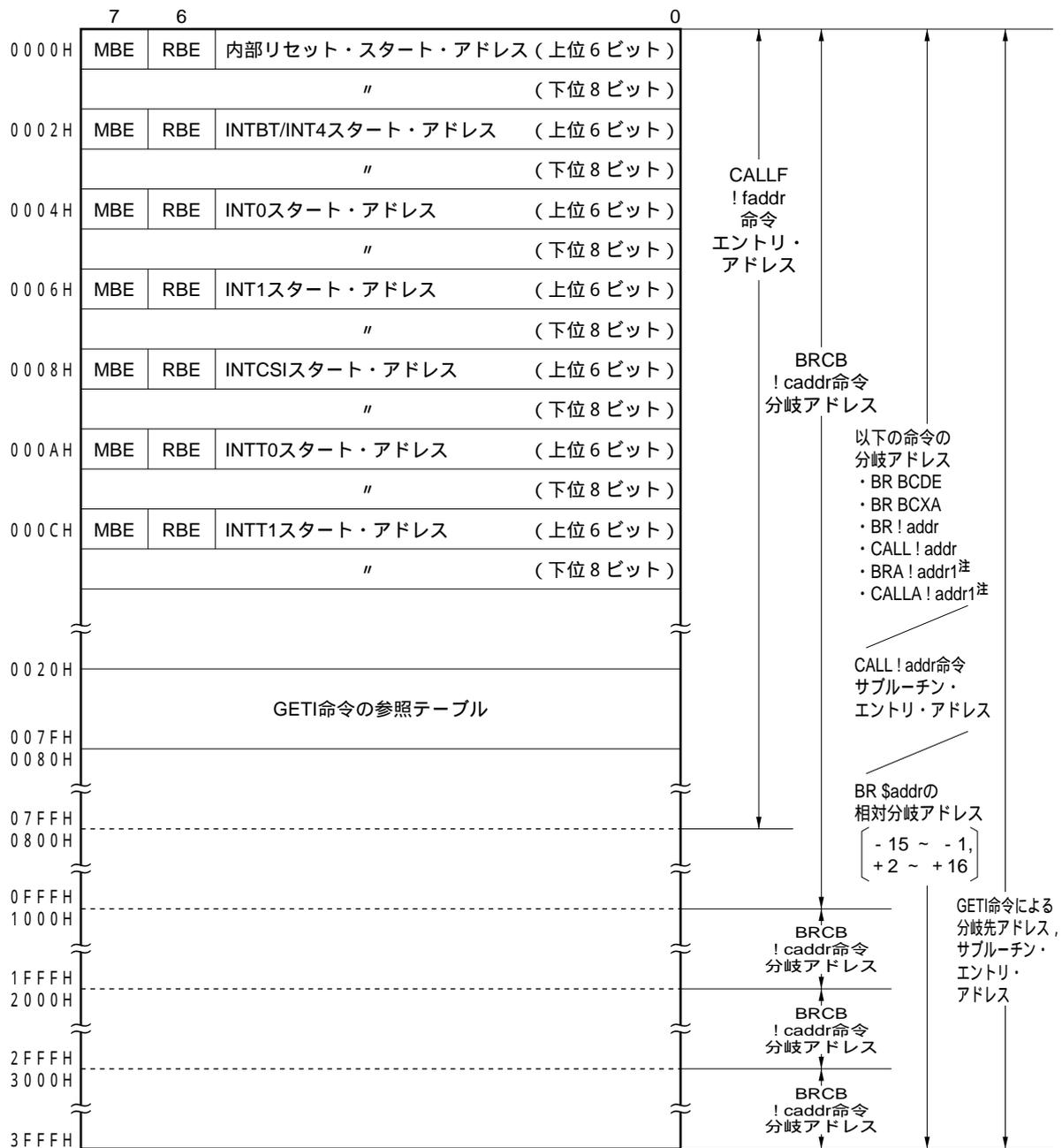


注 Mk モードでのみ使用できます。

備考 上記の命令以外では、BR PCDE, BR PCXA命令により、PCの下位8ビットのみ変更したアドレスへ分岐することができます。

図4-3 プログラム・メモリ・マップ(4/4)

(d)  $\mu$ PD75P0076



注 Mk モードでのみ使用できます。

備考 上記の命令以外では、BR PCDE、BR PCXA命令により、PCの下位8ビットのみ変更したアドレスへ分岐することができます。

## 4.4 データ・メモリ (RAM) .....512ワード × 4ビット

データ・メモリは、図4 - 4のようにデータ・エリアと周辺ハードウェア・エリアによって構成されています。

データ・メモリは、256ワード×4ビットを1バンクとするバンク構成になっており、次のようなメモリ・バンクがあります。

- ・メモリ・バンク0, 1 (データ・エリア)
- ・メモリ・バンク15 (周辺ハードウェア・エリア)

### 4.4.1 データ・メモリの構成

#### (1) データ・エリア

データ・エリアは、スタティックRAMで構成され、データのストアやサブルーチン、割り込み実行時のスタック・メモリとして使用します。また、スタンバイ・モードによりCPUの動作を停止させたときでも、バッテリー・バックアップなどでメモリ内容の長時間保持が可能です。操作はメモリ操作命令で行います。

スタティックRAMは、メモリ・バンク0, 1にそれぞれ256ワード×4ビットずつマッピングされています。バンク0, 1は、データ・エリアとしてマッピングされていますが、それ以外にも汎用レジスタ・エリア (000H-01FH) とスタック・エリア<sup>注</sup> (000H-1FFH) として使用できます。

スタティックRAMは、1アドレスが4ビット構成です。しかし、8ビット・メモリ操作命令により8ビット単位で操作することも、ビット操作命令によりビット単位に操作することもできます。8ビット操作命令では、偶数アドレスを指定してください。

**注** スタック・エリアとして、メモリ・バンク0, 1のうち1つを選択できます。

#### ・汎用レジスタ・エリア

操作は汎用レジスタ操作命令、メモリ操作命令のいずれによっても可能です。最大8本の4ビット・レジスタを使用できます。汎用レジスタ8本のうちプログラムで使用しない部分については、データ・エリアまたはスタック・エリアとして使用できます (4.5 汎用レジスタ参照)。

#### ・スタック・エリア

スタック・エリアは命令によって設定し、サブルーチン実行時や、割り込み処理実行時の退避エリアとして使用できます (4.7 スタック・ポインタ (SP), スタック・バンク選択レジスタ (SBS) 参照)。

## (2) 周辺ハードウェア・エリア

周辺ハードウェア・エリアは、メモリ・バンク15のF80H-FFFFHにマッピングされています。

操作はスタティックRAMと同様にメモリ操作命令で行います。ただし、周辺ハードウェアは、操作可能なビット単位が各アドレスごとに異なります。周辺ハードウェアの割り付けられていないアドレスにはデータ・メモリは内蔵されていないためアクセスできません。

### 4.4.2 データ・メモリのバンクの指定

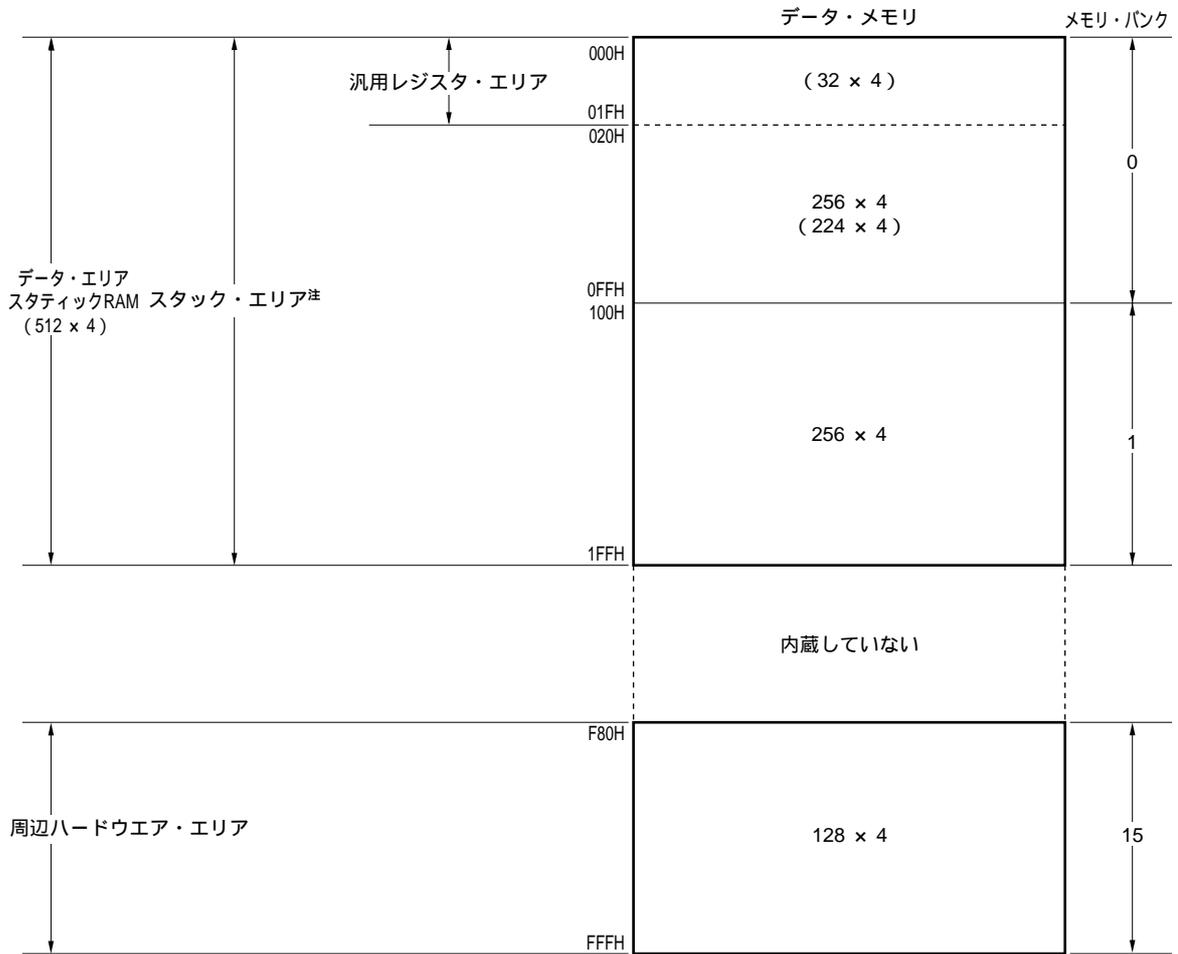
メモリ・バンクは、メモリ・バンク許可フラグ (MBE) によってバンク指定を許可したとき (MBE = 1)、4ビット・メモリ・バンク選択レジスタ (MBS) によって指定されます (MBS = 0, 1, 15)。バンク指定を禁止したとき (MBE = 0)、メモリ・バンクはそのときのアドレッシング・モードによってバンク0またはバンク15が自動的に指定されます。また、バンク内のアドレスは、8ビット・イミューディエト・データやレジスタ・ペアなどによってアドレスされます。

メモリ・バンクの選択とアドレッシングの詳細については、3.1 データ・メモリのバンク構成とアドレッシング・モードを参照してください。

データ・メモリの特定エリアの使用方法については、それぞれ下記を参照してください。

- ・汎用レジスタ・エリア ..... 4.5 汎用レジスタ
- ・スタック・エリア ..... 4.7 スタック・ポインタ (SP),  
スタック・バンク選択レジスタ (SBS)
- ・周辺ハードウェア・エリア ..... 第5章 周辺ハードウェア機能

図4 - 4 データ・メモリ・マップ



注 スタック・エリアとして、メモリ・バンク0，1のうち1つを選択できます。

データ・メモリは、リセット時には不定となっています。このため、通常プログラムの初期でゼロにイニシャライズします（RAMクリア）。これを行わないと思わぬバグ発生の原因となりますので必ず行ってください。

**例** 000H-1FFH番地のRAMをクリアする。

```

SET1    MBE
SEL     MB0
MOV     XA, #00H
MOV     HL, #04H
RAMC0:  MOV   @HL, A      ; 04H-FFHをクリア注
        INCS  L          ; L  L + 1
        BR    RAMC0
        INCS  H          ; H  H + 1
        BR    RAMC0
SEL     MB1
RAMC1:  MOV   @HL, A      ; 100H-1FFHをクリア
        INCS  L          ; L  L + 1
        BR    RAMC1
        INCS  H          ; H  H + 1
        BR    RAMC1

```

**注** 000H-003Hのデータ・メモリは、汎用レジスタXA, HLとして使用しているため、クリアしません。

## 4.5 汎用レジスタ..... 8 × 4 ビット × 4 バンク

汎用レジスタはデータ・メモリの特定番地にマッピングされており，4ビット・レジスタ8個（B, C, D, E, H, L, X, A）を1バンクとして4バンクのレジスタがあります。

命令実行時に有効となるレジスタ・バンク（RB）は次の式により決まります。

$$RB = RBE \cdot RBS \quad (RBS = 0-3)$$

汎用レジスタは，それぞれ4ビット単位で操作されるほか，BC, DE, HL, XAがレジスタ・ペアを形成し，8ビット操作に使用されます。また，DE, HLのほかDLもレジスタ・ペアを形成し，この3組はデータ・ポイントとして使用することができます。

8ビット操作時は，BC, DE, HL, XAのほか，レジスタ・バンク（RB）のビット0を反転したレジスタ・バンク（0 1, 2 3）のレジスタ・ペアをBC', DE', HL', XA'として使用できます（3.2 汎用レジスタのバンク構成参照）。

汎用レジスタ・エリアは，レジスタとして使用する，しないにかかわらず，通常のRAMとしてアドレス指定し，アクセスすることができます。

図4 - 5 汎用レジスタの構成

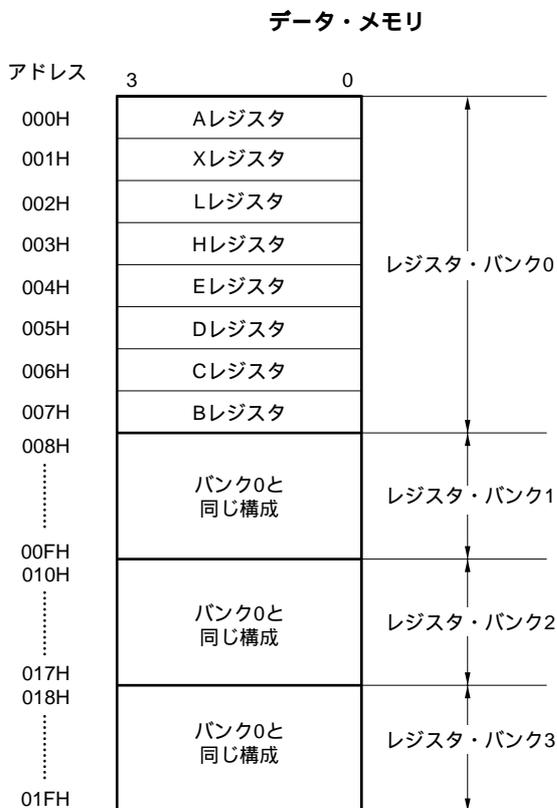
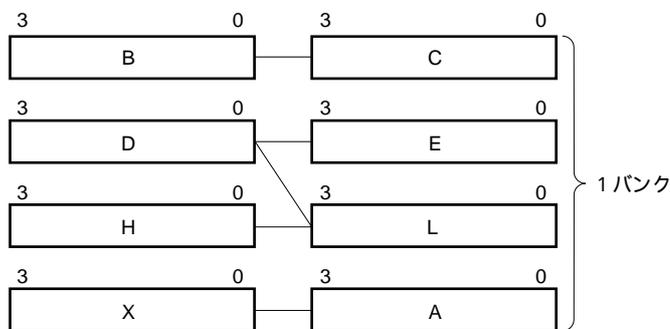


図4 - 6 レジスタ・ペアの構成

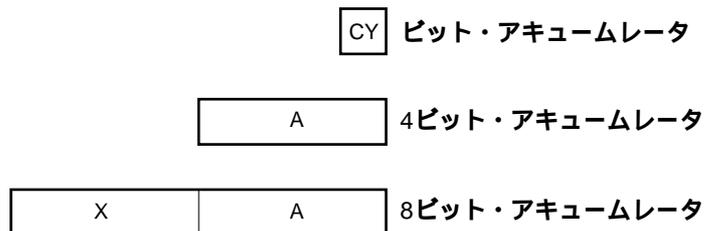


## 4.6 アキュームレータ

μPD750068では、Aレジスタ、XAレジスタ・ペアがアキュームレータとして機能し、4ビット・データ処理命令ではAレジスタが中心となり、8ビット・データ処理命令ではXAレジスタ・ペアが中心となって実行されます。

ビット操作命令では、キャリー・フラグ(CY)がビット・アキュームレータとして機能します。

図4 - 7 アキュームレータ



## 4.7 スタック・ポインタ (SP) , スタック・バンク選択レジスタ (SBS)

μPD750068では、スタティックRAMをスタック・メモリ (LIFO形式) として用いており、このスタック・エリアの先頭アドレス情報を保持している8ビット・レジスタがスタック・ポインタ (SP) です。

スタック・エリアは、メモリ・バンク0, 1の000H-1FFH番地です。2ビットのSBSによって1つのメモリ・バンクを指定します (表4 - 2参照)。

表4 - 2 SBSで選択されるスタック・エリア

SBS		スタック・エリア
SBS1	SBS0	
0	0	メモリ・バンク0
0	1	メモリ・バンク1
上記以外		設定禁止

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

各スタック動作によって退避 / 復帰されるデータは図4 - 9 ~ 4 - 12のようになります。

SPは、8ビット・メモリ操作命令、SBSは4ビット・メモリ操作命令によって初期値を設定し、スタック・エリアを決定します。また内容の読み出しもできます。

SPの初期値を00Hに設定すると、SBSで指定したメモリ・バンク (n) の最上位アドレス (nFFH) からスタックされます。

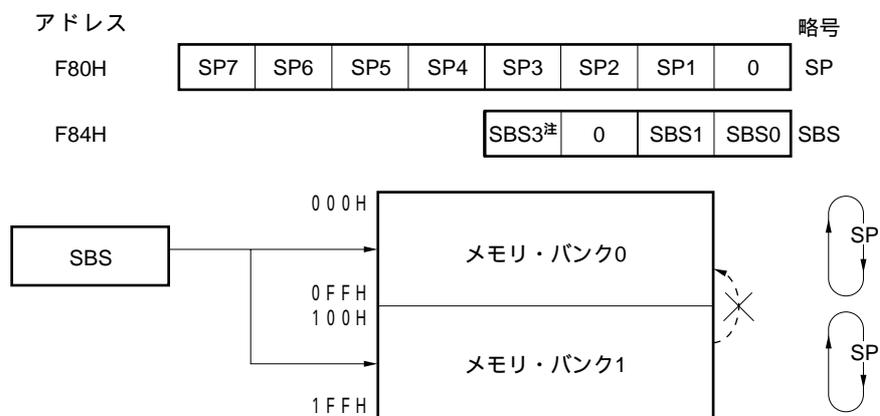
備考 n=0, 1

スタック・エリアはSBSで指定したメモリ・バンク内に限られ、n00H番地からさらにスタック動作を行うと、同一バンク内のnFFHへ戻ります。SBSを書き換えずにメモリ・バンクの境界を越えてスタックすることはできません。

RESET信号発生により、SPの内容は不定に、SBSの内容は1000Bになりますので、プログラムの初期で必ず希望の値にイニシャライズしてください。

備考 n = 0, 1

図4 - 8 スタック・ポインタおよびスタック・バンク選択レジスタの構成



注 SBS3でMk モードとMk モードの切り替えができます。スタック・バンク選択機能は、Mk モード時とMk モード時の両方で使用することができます（詳しくは4.1 Mk モードとMk モードの切り替え機能を参照してください）。

例 SPのイニシャライズ

スタック・エリアをメモリ・バンク1とし、1FFH番地からスタック動作させる場合

```

SEL    MB15      ;またはCLR1 MBE
MOV    A, #1
MOV    SBS, A    ;メモリ・バンク1をスタック・エリアに指定
MOV    XA, #00H
MOV    SP, XA    ;SP 00H
    
```

図4 - 9 スタック・メモリへ退避されるデータ (Mk モード)

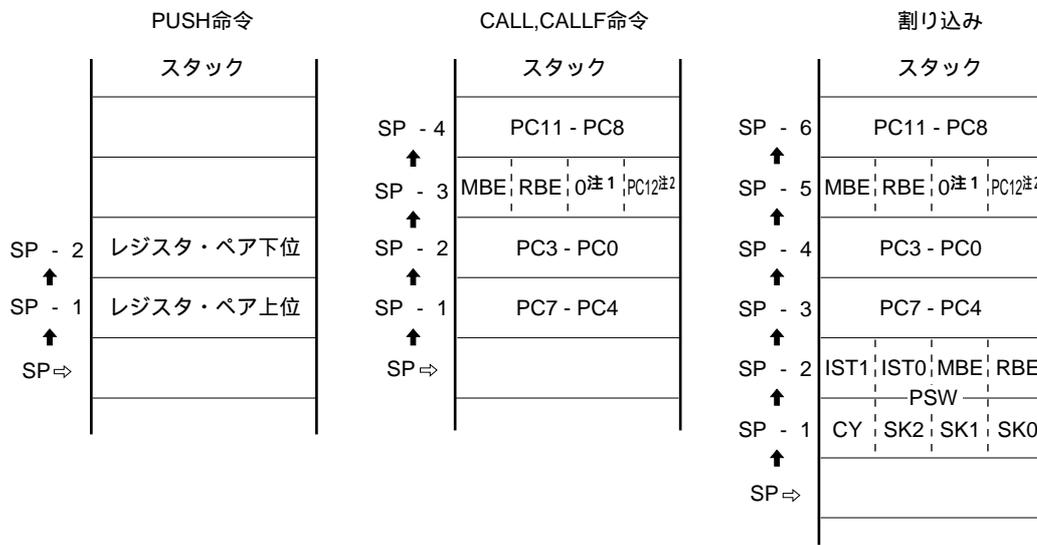
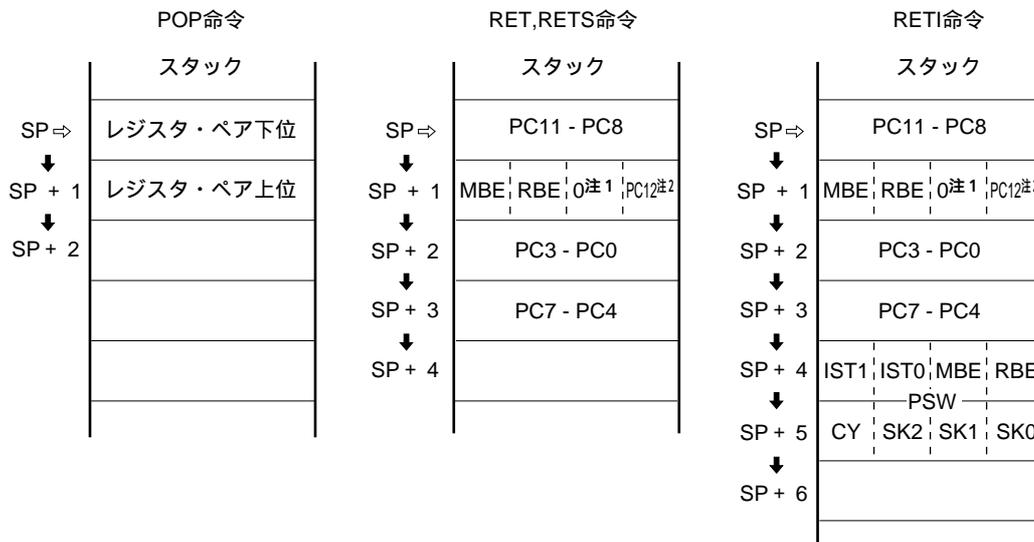


図4 - 10 スタック・メモリから復帰されるデータ (Mk モード)



注1 . μPD75P0076の場合は , PC13が入ります。

2 . μPD750064の場合は , 0 が入ります。

図4 - 11 スタック・メモリへ退避されるデータ (Mk モード)

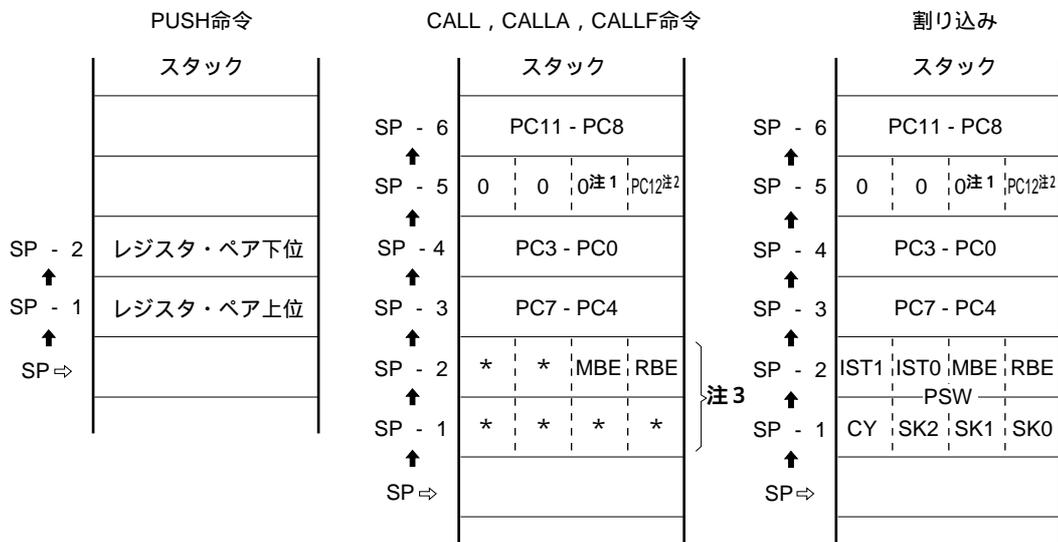
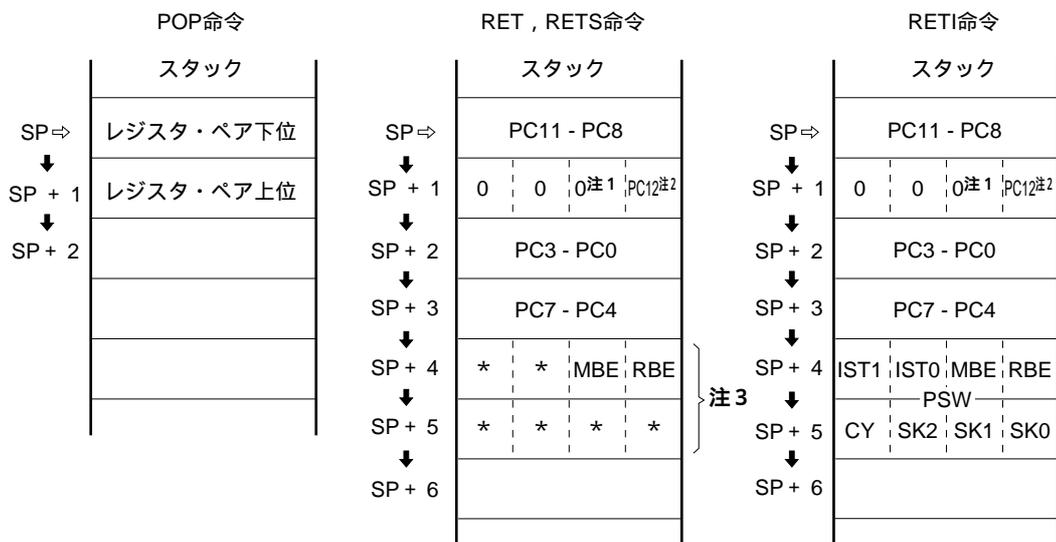


図4 - 12 スタック・メモリから復帰されるデータ (Mk モード)



- 注1 . μPD75P0076の場合は, PC13が入ります。
- 2 . μPD750064の場合は, 0が入ります。
- 3 . MBE, RBE以外のPSWIは退避 / 復帰しません。

備考 \*は不定。

## 4.8 プログラム・ステータス・ワード (PSW) ..... 8 ビット

プログラム・ステータス・ワード (PSW) は、プロセッサ動作と密接に関係する各種のフラグで構成されています。

PSWはデータ・メモリ空間のFB0H, FB1H番地にマッピングされており、FB0H番地の4ビットは、メモリ操作命令で操作できます。

図4-13 プログラム・ステータス・ワードの構成

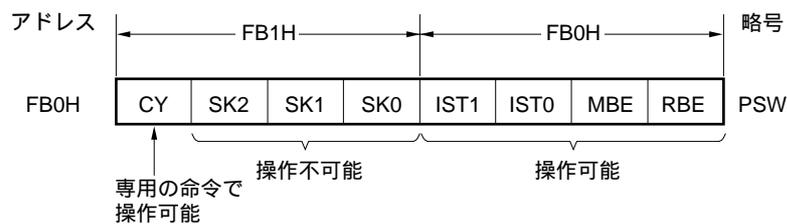


表4-3 スタック動作時に退避/復帰されるPSWのフラグ

		退避 / 復帰されるフラグ
退 避	CALL, CALLA, CALLF命令実行時	MBE, RBEを退避
	ハードウェア割り込み時	PSW全ビットを退避
復 帰	RET, RETS命令実行時	MBE, RBEを復帰
	RETI命令実行時	PSW全ビットを復帰

### (1) キャリー・フラグ (CY)

キャリー・フラグは、キャリー付き演算命令 (ADDC, SUBC) 実行時のオーバフロー、アンダフロー発生情報を記憶する1ビット・フラグです。

また、キャリー・フラグは、ビット・アキュムレータの機能をもっており、ビット・アドレス指定のデータ・メモリとの間でブール代数演算を行い結果をストアすることができます。

キャリー・フラグの操作は、他のPSWのビットとは無関係に専用の命令で行います。

RESET信号発生により、キャリー・フラグは不定となります。

表4 - 4 キャリー・フラグ操作命令

	命令 (二モニック)	キャリー・フラグの動作, 処理
キャリー・フラグ操作専用命令	SET1 CY	CYのセット ( 1 )
	CLR1 CY	CYのクリア ( 0 )
	NOT1 CY	CYの内容の反転
	SKT CY	CYの内容が1ならばスキップ
ビット転送命令	MOV1 mem*.bit, CY	指定ビットへCYの内容を転送
	MOV1 CY, mem*.bit	指定ビットの内容をCYへ転送
ビット・ブーリアン命令	AND1 CY, mem*.bit	指定ビットの内容とCYの内容とでAND/OR/XORをとり, 結果をCYにセット
	OR1 CY, mem*.bit	
	XOR1 CY, mem*.bit	
割り込み処理	割り込み実行時	ほかのPSWビットと8ビット並列にスタック・メモリへ退避
	RETI	スタック・メモリから, ほかのPSWと並列に復帰

**備考** mem\*.bitは次の3つのビット操作アドレッシングを示します。

fmem.bit  
 pmem.@L  
 @H + mem.bit

**例** 3FH番地のビット3とP33のANDをとり, P50に出力する。

```
MOV    H, #3H           ; Hレジスタに上位4ビット・アドレス・セット
MOV1   CY, @H + 0FH.3   ; CY 3FHのビット3
AND1   CY, PORT3.3      ; CY CY P33
MOV1   PORT5.0, CY      ; P50 CY
```

#### (2) スキップ・フラグ (SK2, SK1, SK0)

スキップ・フラグは, スキップ状態を記憶するフラグで, CPUが命令を実行することにより自動的にセット/リセットされます。

ユーザがオペランドとして直接操作することはできません。

**(3) 割り込みステータス・フラグ (IST1, IST0)**

割り込みステータス・フラグは、現在実行中の処理のステータスを記憶する2ビットのフラグです（詳細は表6-3 IST1, IST0と割り込み処理状態参照）。

表4-5 割り込みステータス・フラグの指示内容

IST1	IST0	実行中の処理のステータス	処理内容と割り込み制御
0	0	ステータス0	通常のプログラムを処理中。 すべての割り込みを受け付け可能。
0	1	ステータス1	低位の、または高位の割り込み処理中。 高位の割り込みのみ受け付け可能。
1	0	ステータス2	高位の割り込み処理中。 すべての割り込みの受け付けを禁止。
1	1	-	設定禁止。

割り込みプライオリティ・コントロール回路（図6-1 割り込み制御回路ブロック図参照）は、このフラグ内容を判別して多重割り込みの制御をします。

IST1, 0の内容は、割り込みが受け付けられればPSWの一部としてスタック・メモリに退避されたあと自動的に一段階上のステータスに変更され、RETI命令により割り込みに入る前の値が復帰します。

割り込みステータス・フラグはメモリ操作命令によって操作可能で、実行中の処理ステータスをプログラム制御で変更することもできます。

**注意** このフラグを操作する場合は、操作前に必ずDI命令を実行して割り込みを禁止し、操作後にEI命令を実行して割り込みを許可するようにしてください。

**(4) メモリ・バンク許可フラグ (MBE)**

データ・メモリ・アドレス12ビットのうち、上位4ビットのアドレス情報発生モードを指定する1ビットのフラグです。

MBEは、メモリ・バンクの設定にかかわらず、いつでもビット操作命令によりセット/リセットすることができます。

“1”に設定すると、データ・メモリ・アドレス空間は拡張され、すべてのデータ・メモリ空間がアドレス可能となります。

“0”にリセットすると、データ・メモリ・アドレス空間は、MBSにかかわらず固定されます（図3-2 データ・メモリの構成と、各アドレス・モードのアドレッシング範囲参照）。

RESET信号発生により、プログラム・メモリ0番地のビット7の内容がセットされて、自動的に初期設定されます。

ベクタ割り込み処理時は、該当するベクタ・アドレス・テーブルのビット7の内容がセットされて、割り込みサービス時のMBEの状態が自動的に設定されます。

通常、割り込み処理ではMBE = 0とし、メモリ・バンク0のスタティックRAMを使用します。

#### (5) レジスタ・バンク許可フラグ (RBE)

汎用レジスタのレジスタ・バンク構成を、拡張するか否かを制御する1ビットのフラグです。

RBEは、メモリ・バンクの設定にかかわらず、いつでも、ビット操作命令によりセット/リセットすることができます。

“1”にセットすると、レジスタ・バンク選択レジスタ (RBS) の内容により汎用レジスタを、レジスタ・バンク0-3の4組のうちから1組選択できます。

“0”にリセットすると、レジスタ・バンク選択レジスタ (RBS) の内容にかかわらず、常にレジスタ・バンク0が汎用レジスタとして選択されます。

$\overline{\text{RESET}}$ 信号発生により、プログラム・メモリ0番地のビット6の内容がセットされて自動的に初期設定されます。

ベクタ割り込み発生時は、該当するベクタ・アドレス・テーブルのビット6の内容がセットされて、割り込みサーブス時のRBEの状態が自動的に設定されます。通常、割り込み処理ではRBE=0とし、4ビット処理ではレジスタ・バンク0を、8ビット処理ではレジスタ・バンク0, 1を使用します。

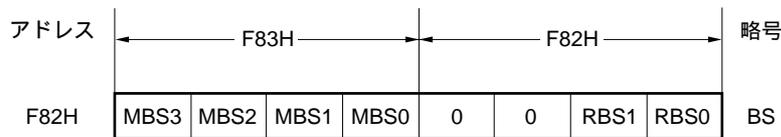
## 4.9 バンク選択レジスタ (BS)

バンク選択レジスタ (BS) は、レジスタ・バンク選択レジスタ (RBS) とメモリ・バンク選択レジスタ (MBS) とで構成され、それぞれ使用するレジスタ・バンクとメモリ・バンクを指定します。

RBS, MBSは、それぞれSEL RBn, SEL MBn命令で設定されます。

BSは、PUSH BS/POP BS命令により、8ビット単位でスタック・エリアへ退避/復帰することができます。

図4 - 14 バンク選択レジスタの構成



### (1) メモリ・バンク選択レジスタ (MBS)

メモリ・バンク選択レジスタは、データ・メモリ・アドレス (12ビット) 上位4ビット・アドレス情報を記憶する4ビット・レジスタで、このレジスタの内容によりアクセスするメモリ・バンクを指定します。ただし、μPD750068では、バンク0, 1, 15のみ指定できます。

MBSは、SEL MBn命令により設定します (n=0, 1, 15)。

MBEとMBSの設定に対するアドレス範囲は図3 - 2に示すようになります。

$\overline{\text{RESET}}$ 信号発生により、MBSは“0”にイニシャライズされます。

### (2) レジスタ・バンク選択レジスタ (RBS)

レジスタ・バンク選択レジスタは、汎用レジスタとして使用するレジスタ・バンクを指定するレジスタで、バンク0からバンク3まで設定できます。

RBSは、SEL RBn命令により設定します (n=0-3)。

$\overline{\text{RESET}}$ 信号発生により、RBSは“0”にイニシャライズされます。

表4 - 6 RBE, RBSと選択されるレジスタ・バンク

RBE	RBS				レジスタ・バンク
	3	2	1	0	
0	0	0	x	x	バンク0に固定
1	0	0	0	0	バンク0を選択
			0	1	バンク1 "
			1	0	バンク2 "
			1	1	バンク3 "

0に固定

x : don't care

## 第5章 周辺ハードウェア機能

### 5.1 デジタル入出力ポート

μPD750068ではメモリ・マップトI/Oが採用されており、すべての入出力ポートはデータ・メモリ空間上にマッピングされています。

図5 - 1 デジタル・ポートのデータ・メモリ・アドレス

アドレス	3	2	1	0	
FF0H	P03	P02	P01	P00	PORT0
FF1H	P13	P12	P11	P10	PORT1
FF2H	P23	P22	P21	P20	PORT2
FF3H	P33	P32	P31	P30	PORT3
FF4H	P43	P42	P41	P40	PORT4
FF5H	P53	P52	P51	P50	PORT5
FF6H	P63	P62	P61	P60	PORT6
FFBH	P113	P112	P111	P110	PORT11

入出力ポートの操作命令は表5 - 2のようになっており、PORT4、PORT5については4ビット入出力のほか8ビット入出力、ビット操作が可能で、きわめて多様な制御ができます。

**例1** . P13の状態をテストして、その結果によって異なった値をポート4、5に出力。

SKT PORT1.3 ; ポート1のビット3が1ならスキップ。

MOV XA, #18H ; XA 18H }  
MOV XA, #14H ; XA 14H } たてづみ

SEL MB15 ; または、CLR1 MBE

OUT PORT4, XA ; ポート5, 4 XA

2 . SET1 PORT4.@L ; ポート4, 5のうち、Lレジスタで指定されるビットを“1”にセット。

### 5.1.1 デジタル入出力ポートの種類、特徴、構成

デジタル入出力ポートには表5-1のような種類があります。

各ポートの構成は図5-2～5-6のようになっています。

表5-1 デジタル・ポートの種類と特徴

ポート (端子名)	機能	動作・特徴	備考
PORT0 (P00-P03)	4ビット入力	シリアル・インタフェース機能使用時は、動作モードによって兼用端子が出力機能を持ちます。	INT4, $\overline{SCK}$ , SO/SB0, SI/SB1と端子を兼用。
PORT1 (P10-P13)		4ビット入力専用ポート。	INT0-INT2/TI1, TI0と端子を兼用。
PORT2 (P20-P23)	4ビット入出力	4ビット単位で入力または出力モードに設定可能。	PTO0, PTO1, PCL, BUZと端子を兼用。
PORT3 (P30-P33)		1ビット単位で入力または出力モードに設定可能。	MD0-MD3 <sup>注1</sup> と端子を兼用。
PORT4 (P40-P43)	4ビット入出力 (N-chオープン・ドレーン13V耐圧)	4ビット単位で入力または出力モードに設定可能。マスク・オプション <sup>注2</sup> で、1ビット単位でプルアップ抵抗内蔵の指定可能。ペアとなって8ビット単位でデータの入出力が可能。	D0-D3 <sup>注1</sup> と端子を兼用。
PORT5 (P50-P53)			D4-D7 <sup>注1</sup> と端子を兼用。
PORT6 (P60-P63)	4ビット入出力	1ビット単位で入力または出力モードに設定可能。	KR0-KR3, AN4-AN7と端子を兼用。
PORT11 (P110-P113)	4ビット入力	4ビット入力専用ポート。	AN0-AN3と端子を兼用。

注1 .  $\mu$  PD75P0076のときのみ兼用している端子です。

2 .  $\mu$  PD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

P10は、外部ベクタ割り込み入力端子と兼用になっており、ノイズ除去回路付き入力となっています（詳細は6.3 割り込み機能を制御する各種ハードウェアを参照）。

$\overline{RESET}$ 信号発生により、各ポート・レジスタは全ビット“0”にクリアされるため、出力バッファはオフし、全ポートは入力モードになります。

図5-2 ポート0, 1の構成

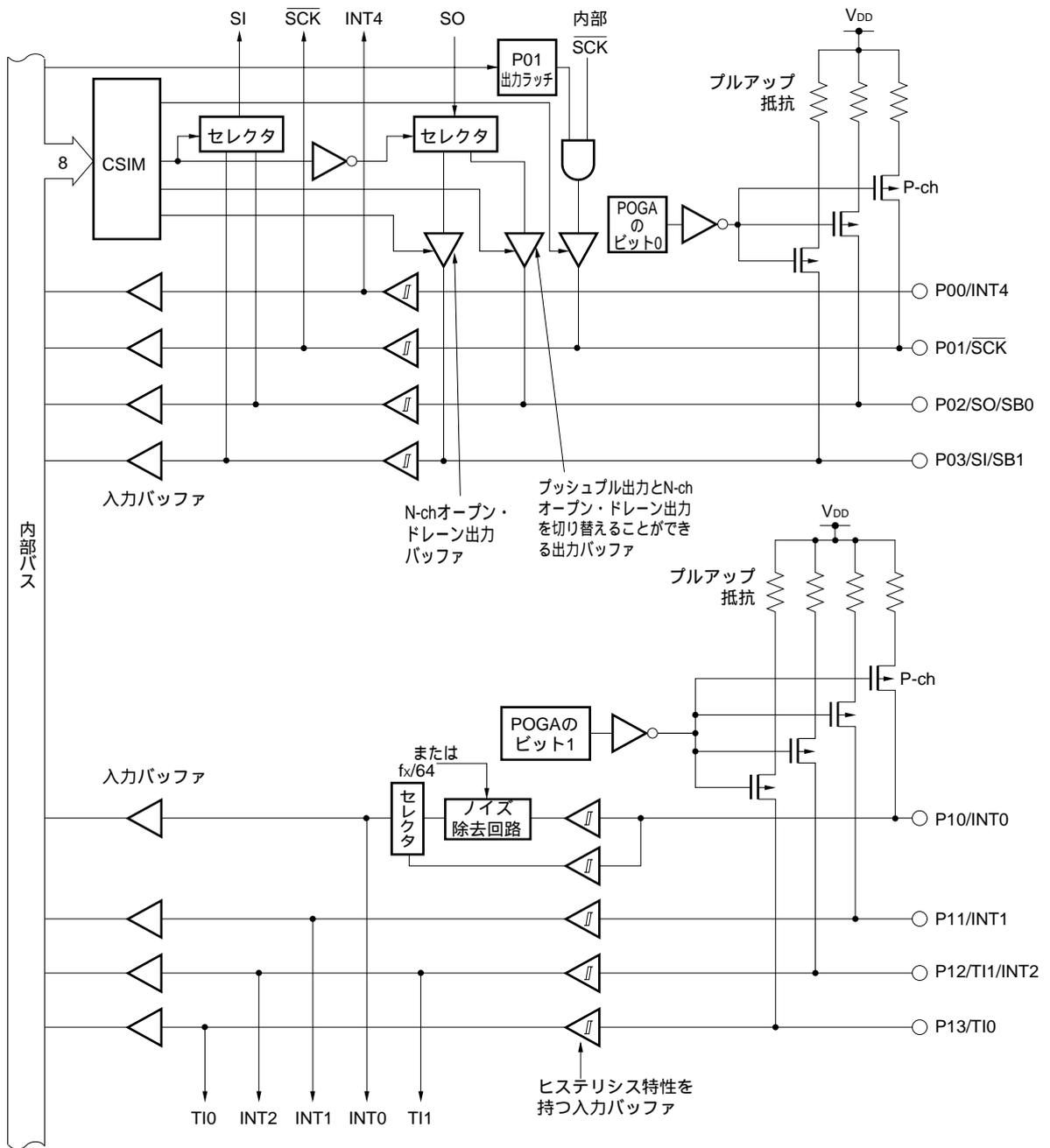


図5-3 ポート3, 6の構成

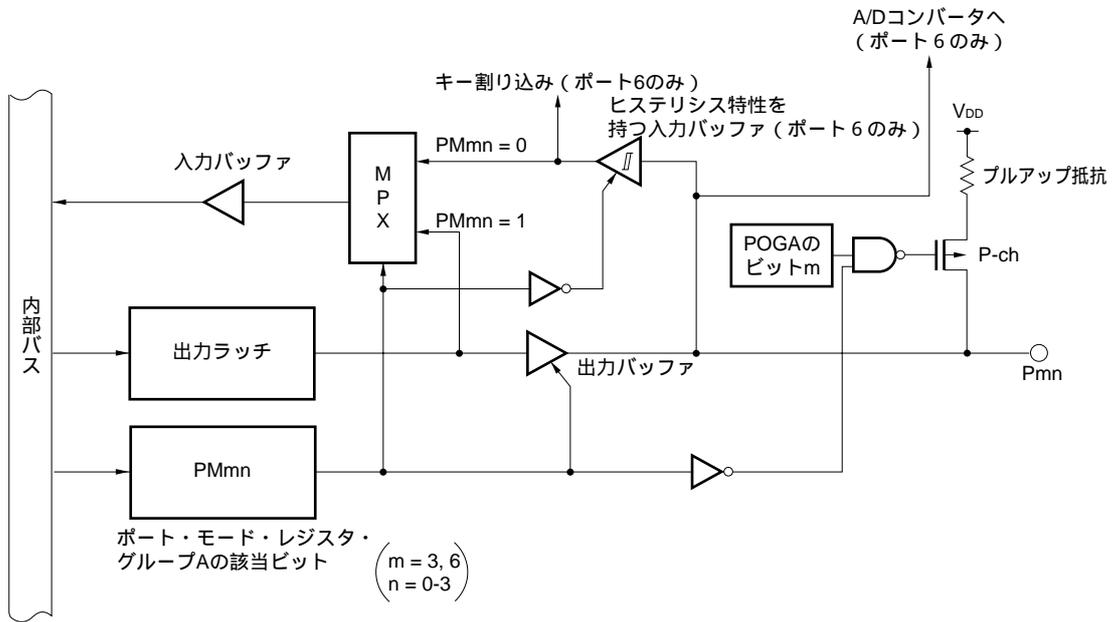


図5 - 4 ポート2の構成

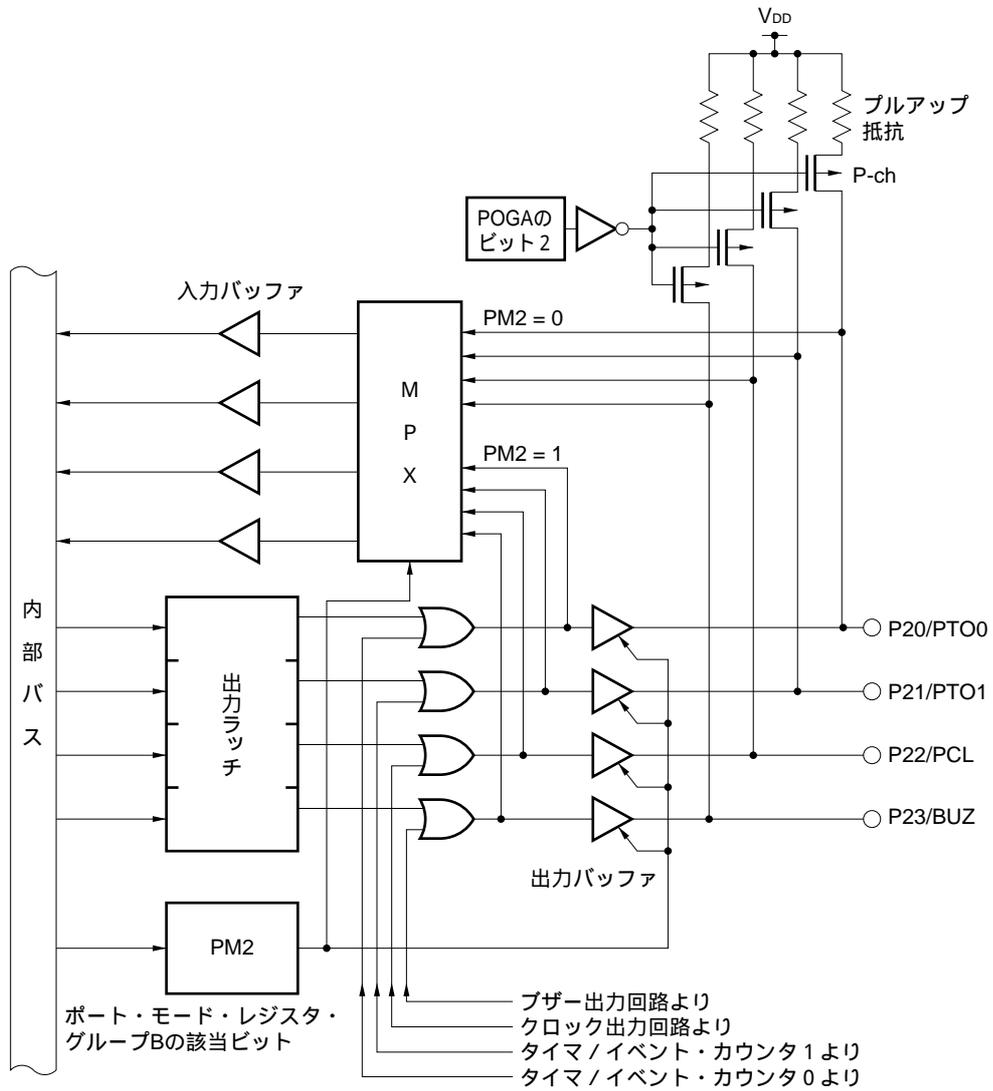


図5-5 ポート4, 5の構成

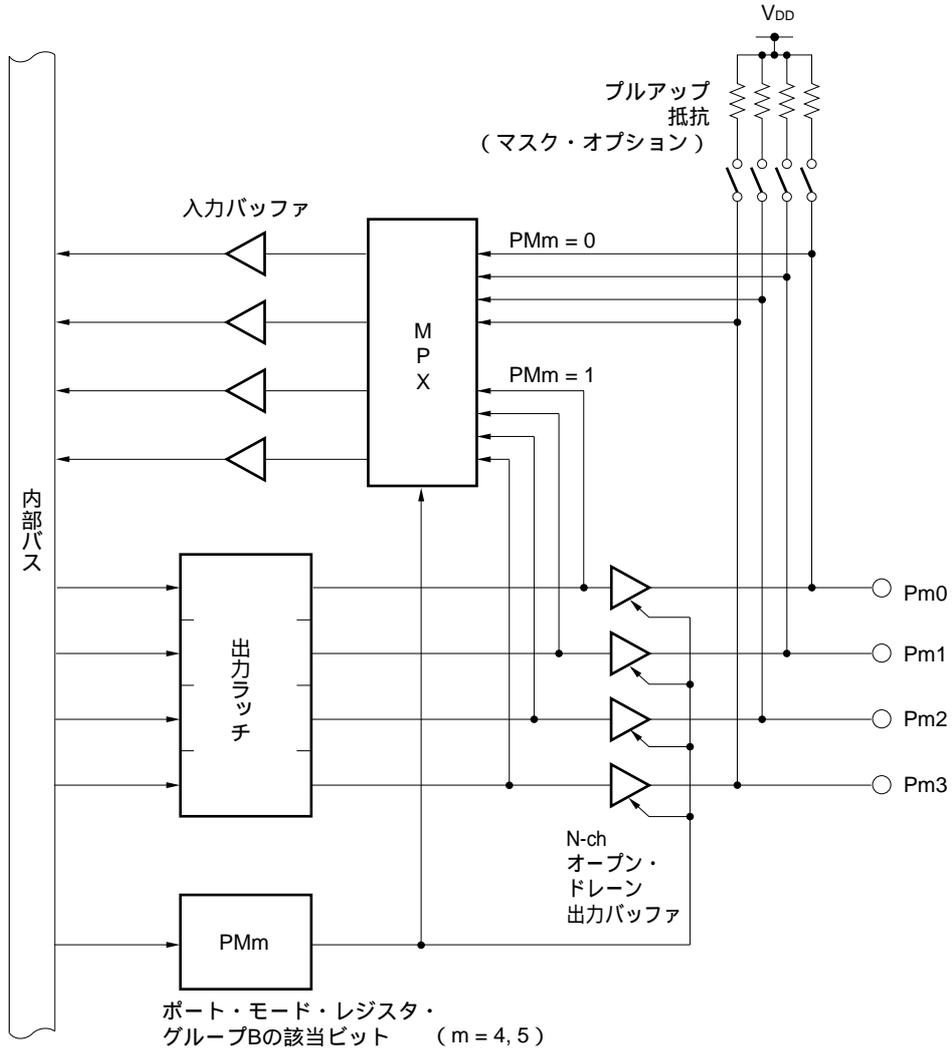
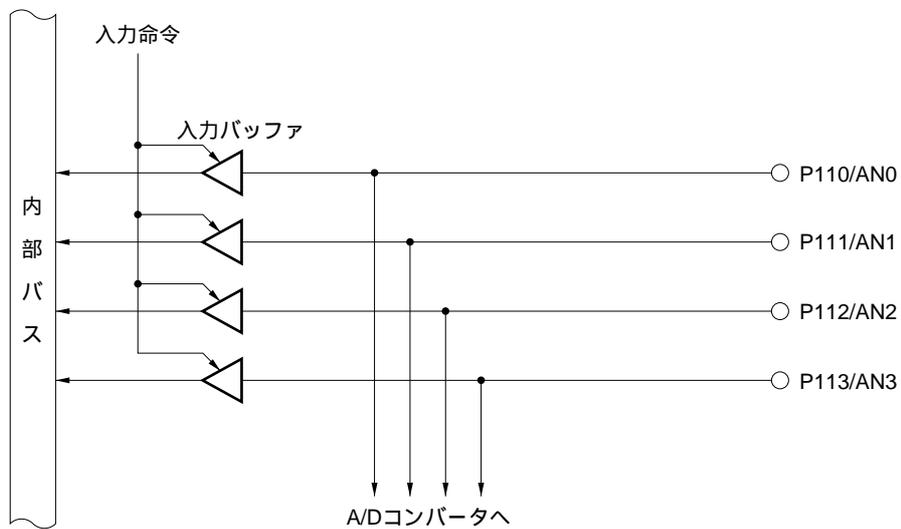


図5-6 ポート11の構成



## 5.1.2 入出力モードの設定

各入出力ポートの入出力モードは、図5-7のようにポート・モード・レジスタにより設定されます。ポート3, 6はポート・モード・レジスタ・グループA (PMGA) によって、ビットごとに入出力を指定できます。ポート2, 4, 5はポート・モード・レジスタ・グループB (PMGB) によって、4ビット単位で入出力を指定します。

各ポートは、対応するポート・モード・レジスタのビットが“0”のとき入力ポートとして働き、“1”のとき出力ポートとして働きます。

ポート・モード・レジスタの設定により出力モードが選択されると同時に出力ラッチの内容が、出力端子に出力されますから、出力モードを設定する前にあらかじめ、出力ラッチの内容を必要な値に書き換えておく必要があります。

ポート・モード・レジスタ・グループA, Bは、それぞれ8ビット・メモリ操作命令で設定します。

$\overline{\text{RESET}}$ 信号発生により、各ポート・モード・レジスタは全ビット0にクリアされるため出力バッファはオフし、全ポートは入力モードになります。

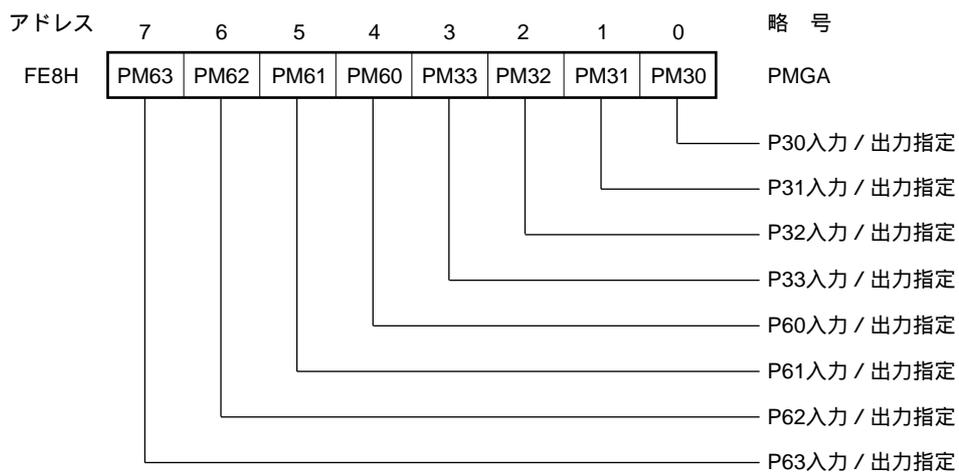
**例** P30, 31, 62, 63を入力, P32, 33, 60, 61を出力端子として使う。

```
CLR1    MBE           ;またはSEL MB15
MOV     XA, #3CH
MOV     PMGA, XA
```

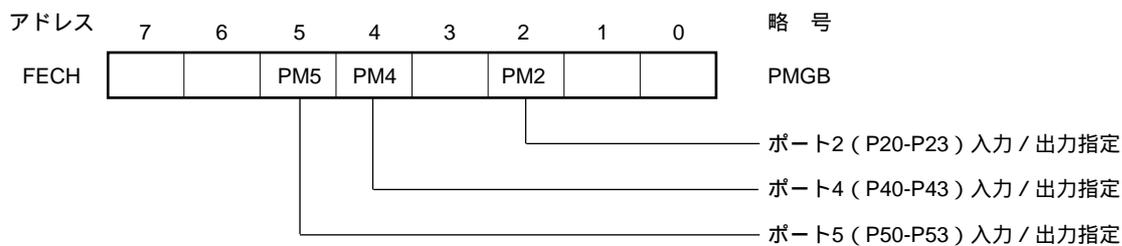
図5 - 7 各ポート・モード・レジスタのフォーマット

	指定内容
0	入力モード (出力バッファ・オフ)
1	出力モード (出力バッファ・オン)

ポート・モード・レジスタ・グループA



ポート・モード・レジスタ・グループB



### 5.1.3 デジタル入出力ポート操作命令

μPD750068に内蔵されている入出力ポートはすべてデータ・メモリ空間にマッピングされていますから、データ・メモリ操作命令はすべて適用することができます。データ・メモリ操作命令のうち、特に入出力端子の操作に有効と思われる命令とその適用範囲を表5 - 2にまとめます。

#### (1) ビット操作命令

デジタル入出力ポートPORT0-6, 11は、特定アドレス・ビット直接アドレッシング (fmem. bit) と特定アドレス・ビット・レジスタ間接アドレッシング (pmem. @L) が適用できますから、MBE, MBSの指定にかかわらずいつでも、自由に、ポートのビット操作が可能です。

例 P50とP41のORをとってP61に出力する。

```
MOV1    CY, PORT5.0    ; CY P50
OR1     CY, PORT4.1    ; CY CY P41
MOV1    PORT6.1, CY    ; P61 CY
```

#### (2) 4ビット操作命令

IN/OUT命令のほか、MOV, XCH, ADDS, INCSなど4ビット・メモリ操作命令がすべて使用できますが、命令の実行に先立ってメモリ・バンク15を選択しておく必要があります。

例1 . ポート3に、アキュムレータの内容を出力する。

```
SET1    MBE
SEL     MB15           ; またはCLR1 MBE
OUT     PORT3, A
```

2 . ポート5に出力しているデータにアキュムレータの値を加えて出力する。

```
SET1    MBE
SEL     MB15
MOV     HL, #PORT5
ADDS   A, @HL         ; A A + PORT5
NOP
MOV     @HL, A        ; PORT5 A
```

3 . ポート4のデータがアキュムレータの値より大きいかどうかテストする。

```
SET1    MBE
SEL     MB15
MOV     HL, #PORT4
SUBS   A, @HL         ; A < PORT4
BR     NO             ; NO
                     ; YES
```

**(3) 8ビット操作命令**

8ビット操作が可能なポート4, 5に対しIN/OUT命令のほか, MOV/XCH/SKE命令が使用できません。4ビット操作時と同様に, あらかじめ, メモリ・バンク15を選択しておく必要があります。

**例** ポート4, 5から入力された8ビット・データで指定される出力ポートにBCレジスタ・ペアのデータを出力する。

```
SET1      MBE
SEL       MB15
IN        XA, PORT4          ; XA ポート5, 4
MOV       HL, XA             ; HL XA
MOV       XA, BC             ; XA BC
MOV       @HL, XA           ; ポート(L) XA
```

表5 - 2 入出力端子操作命令一覧表

		PORT							
		0	1	2	3	4	5	6	11
IN	A, PORTn	注1							
IN	XA, PORTn	注1	-	-				-	
OUT	PORTn, A	注1	-						-
OUT	PORTn, XA	注1	-	-				-	
MOV	A, PORTn	注1							
MOV	XA, PORTn	注1	-	-				-	
MOV	PORTn, A	注1	-						-
MOV	PORTn, XA	注1	-	-				-	
XCH	A, PORTn	注1							
XCH	XA, PORTn	注1	-	-				-	
MOV1	CY, PORTn.bit								
MOV1	CY, PORTn.@L	注2							
MOV1	PORTn.bit, CY		-	-					
MOV1	PORTn.@L, CY	注2	-	-					
INCS	PORTn	注1							
SET1	PORTn.bit		-						-
SET1	PORTn.@L	注2	-						-
CLR1	PORTn.bit		-						-
CLR1	PORTn.@L	注2	-						-
SKT	PORTn.bit								
SKT	PORTn.@L	注2							
SKF	PORTn.bit								
SKF	PORTn.@L	注2							
SKTCLR	PORTn.bit								
SKTCLR	PORTn.@L	注2							
AND1	CY, PORTn.bit								
AND1	CY, PORTn.@L	注2							
OR1	CY, PORTn.bit								
OR1	CY, PORTn.@L	注2							
XOR1	CY, PORTn.bit								
XOR1	CY, PORTn.@L	注2							

注1 . 実行前に、あらかじめMBE=0、または(MBE=1, MBS=15)としておく必要があります。

2 . アドレスの下位2ビットとビット・アドレスを、Lレジスタで間接指定します。

### 5.1.4 デジタル入出力ポートの動作

デジタル入出力ポートに対し、データ・メモリ操作命令を実行したときの各ポートおよび、各端子の動作は、入出力モードの設定によって異なります（表5-3参照）。これは入出力ポートの構成からもわかるように、内部バスに取り込まれるデータが、入力モードでは各端子のデータ、出力モードでは出力ラッチのデータになっているためです。

#### （1）入力モードに設定されているときの動作

SKT命令のようなテスト命令、MOV1命令によるビット入力命令、あるいは4、8ビットで、ポート・データを内部バスに取り込む命令（IN、MOV、演算命令、比較命令）が実行されたときは、各端子のデータが操作されます。

4、8ビットでアキュムレータの内容をポートに転送する命令（OUT、MOV命令）が実行されたときは、アキュムレータのデータが出力ラッチにラッチされます。出力バッファはオフしたままです。

XCH命令が実行されたときは、各端子のデータがアキュムレータに入力され、アキュムレータのデータは出力ラッチにラッチされます。出力バッファはオフしたままです。

INCS命令を実行すると、各端子のデータ（4ビット）に+1したデータが、出力ラッチにラッチされます。出力バッファはオフしたままです。

SET1/CLR1/MOV1/SKTCLR命令のように、ビット単位でデータ・メモリを書き換える命令を実行したときは、指定したビットの出力ラッチは、命令の指示どおりに書き換えることができますが、その他のビットの出力ラッチの内容が不定となります。

#### （2）出力モードに設定されているときの動作

テスト命令、ビット入力命令、4、8ビットでポート・データを内部バスに取り込む命令を実行すると、出力ラッチの内容が操作されます。

4、8ビットでアキュムレータの内容を転送する命令が実行されたときは、出力ラッチのデータが書き換えられると同時に端子から出力されます。

XCH命令が実行されたときは、出力ラッチの内容がアキュムレータに転送され、アキュムレータの内容は、出力ラッチにラッチされ、端子から出力されます。

INCS命令を実行すると、出力ラッチの内容を+1したデータが出力ラッチにラッチされ、端子から出力されます。

ビット出力命令を実行すると、指定された出力ラッチのビットが書き換えられ、端子から出力されません。

表5 - 3 入出力ポート操作時の動作

実行する命令	ポートおよび各端子の動作	
	入力モード	出力モード
SKT <input type="checkbox"/> SKF <input type="checkbox"/>	端子のデータをテスト	出力ラッチのデータをテスト
MOV1 CY, <input type="checkbox"/>	端子のデータをCYへ転送	出力ラッチのデータをCYへ転送
AND1 CY, <input type="checkbox"/> OR1 CY, <input type="checkbox"/> XOR1 CY, <input type="checkbox"/>	端子のデータとCYの間で演算	出力ラッチのデータとCYの間で演算
IN A, PORTn IN XA, PORTn MOV A, PORTn MOV XA, PORTn MOV A, @HL MOV XA, @HL	端子のデータをアキュムレータへ転送	出力ラッチのデータをアキュムレータへ転送
ADDS A, @HL ADDC A, @HL SUBS A, @HL SUBC A, @HL AND A, @HL OR A, @HL XOR A, @HL	端子のデータとアキュムレータの間で演算	出力ラッチのデータとアキュムレータの間で演算
SKE A, @HL SKE XA, @HL	端子のデータとアキュムレータを比較	出力ラッチのデータとアキュムレータを比較
OUT PORTn, A OUT PORTn, XA MOV PORTn, A MOV PORTn, XA MOV @HL, A MOV @HL, XA	出力ラッチにアキュムレータのデータを転送（出力バッファはオフのまま）	出力ラッチにアキュムレータのデータを転送し、端子から出力
XCH A, PORTn XCH XA, PORTn XCH A, @HL XCH XA, @HL	端子のデータをアキュムレータに転送し、アキュムレータのデータは出力ラッチに転送（出力バッファはオフのまま）	出力ラッチとアキュムレータの間でデータを交換
INCS PORTn INCS @HL	端子のデータを+ 1したデータを出力ラッチにラッチ	出力ラッチの内容を+ 1
SET1 <input type="checkbox"/> CLR1 <input type="checkbox"/> MOV1 <input type="checkbox"/> , CY SKTCLR <input type="checkbox"/>	指定したビットの出力ラッチは命令の指示通り書き替わるが、その他のビットの出力ラッチは不定	出力端子の状態を命令に従って変更

備考  : PORTn. bitとPORTn. @Lの2つのアドレッシング・モードを示します。

### 5.1.5 ブルアップ抵抗の内蔵

μPD750068の各ポート端子にはブルアップ抵抗を内蔵することができます（ただし、P00端子を除く）。ブルアップ抵抗の内蔵の指定はソフトウェアによる端子と、マスク・オプションによる端子があります。

各ポート端子の指定法を表5 - 4に示します。また、ソフトウェアによる内蔵ブルアップ抵抗の接続指定は、図5 - 8に示したフォーマットに基づいて行います。

ポート3, 6に対する内蔵ブルアップ抵抗の接続指定は、入力モードに指定されている端子にのみ有効です。出力モードに指定された端子は、POGAの設定にかかわらず、内蔵ブルアップ抵抗の接続を指定できません。

表5 - 4 ブルアップ抵抗内蔵の指定法

ポート（端子名）	ブルアップ抵抗内蔵の指定法	指定ビット
ポート0（P01-P03） <sup>注</sup>	3ビット単位でソフトウェアにより接続を指定	POGA.0
ポート1（P10-P13）	4ビット単位でソフトウェアにより接続を指定	POGA.1
ポート2（P20-P23）		POGA.2
ポート3（P30-P33）		POGA.3
ポート4（P40-P43）	1ビット単位でマスク・オプションにより内蔵指定	-
ポート5（P50-P53）		
ポート6（P60-P63）	4ビット単位でソフトウェアにより接続を指定	POGA.6

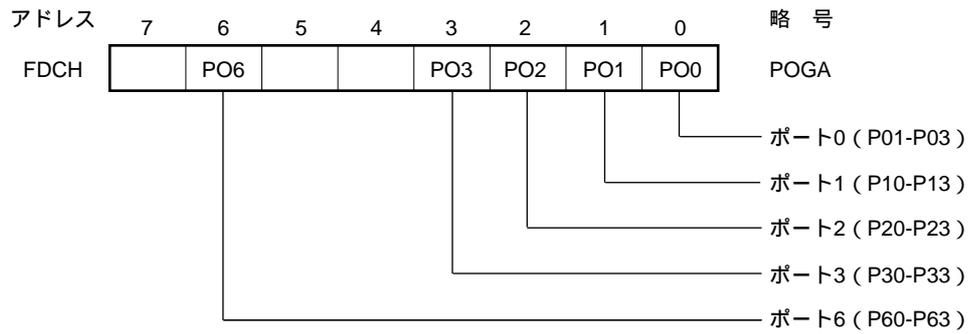
注 P00端子には内蔵ブルアップ抵抗の接続を指定できません。

備考 μPD75P0076には、マスク・オプションによるブルアップ抵抗の内蔵機能はありません。

図5 - 8 プルアップ抵抗指定レジスタのフォーマット

	指定内容
0	内蔵プルアップ抵抗の接続を指定しない
1	内蔵プルアップ抵抗の接続を指定する

プルアップ抵抗指定レジスタ・グループA



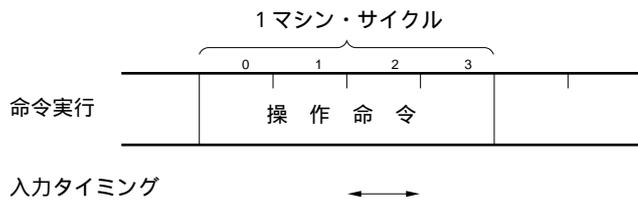
### 5.1.6 デジタル入出力ポートの入出力タイミング

出力ラッチにデータが出力されるタイミングおよび端子のデータまたは、出力ラッチのデータが内部バスに取り込まれるタイミングは図5 - 9 に示すようになっています。

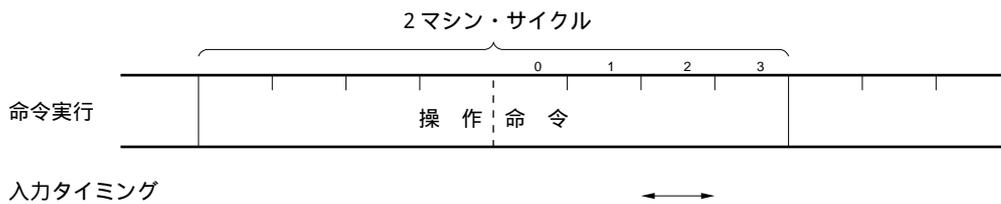
また、ソフトウェアにより内蔵プルアップ抵抗の接続を指定した場合のONタイミングを図5 - 10 に示します。

図5 - 9 デジタル入出力ポートの入出力タイミング

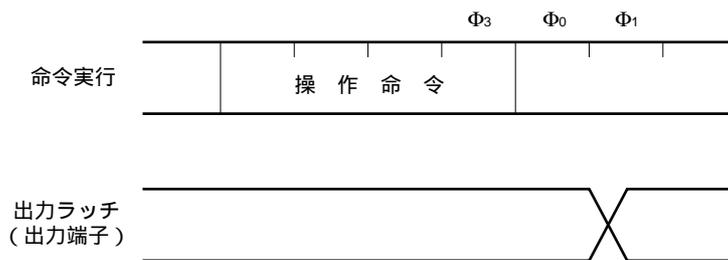
( a ) 1 マシン・サイクル命令によってデータが取り込まれる場合



( b ) 2 マシン・サイクル命令によってデータが取り込まれる場合



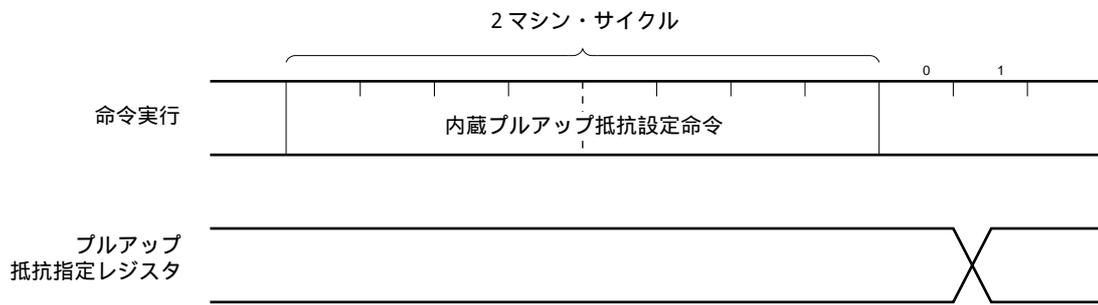
( c ) 1 マシン・サイクル命令によってデータがラッチされる場合



( d ) 2 マシン・サイクル命令によってデータがラッチされる場合



図5 - 10 ソフトウェアによる内蔵プルアップ抵抗のONタイミング



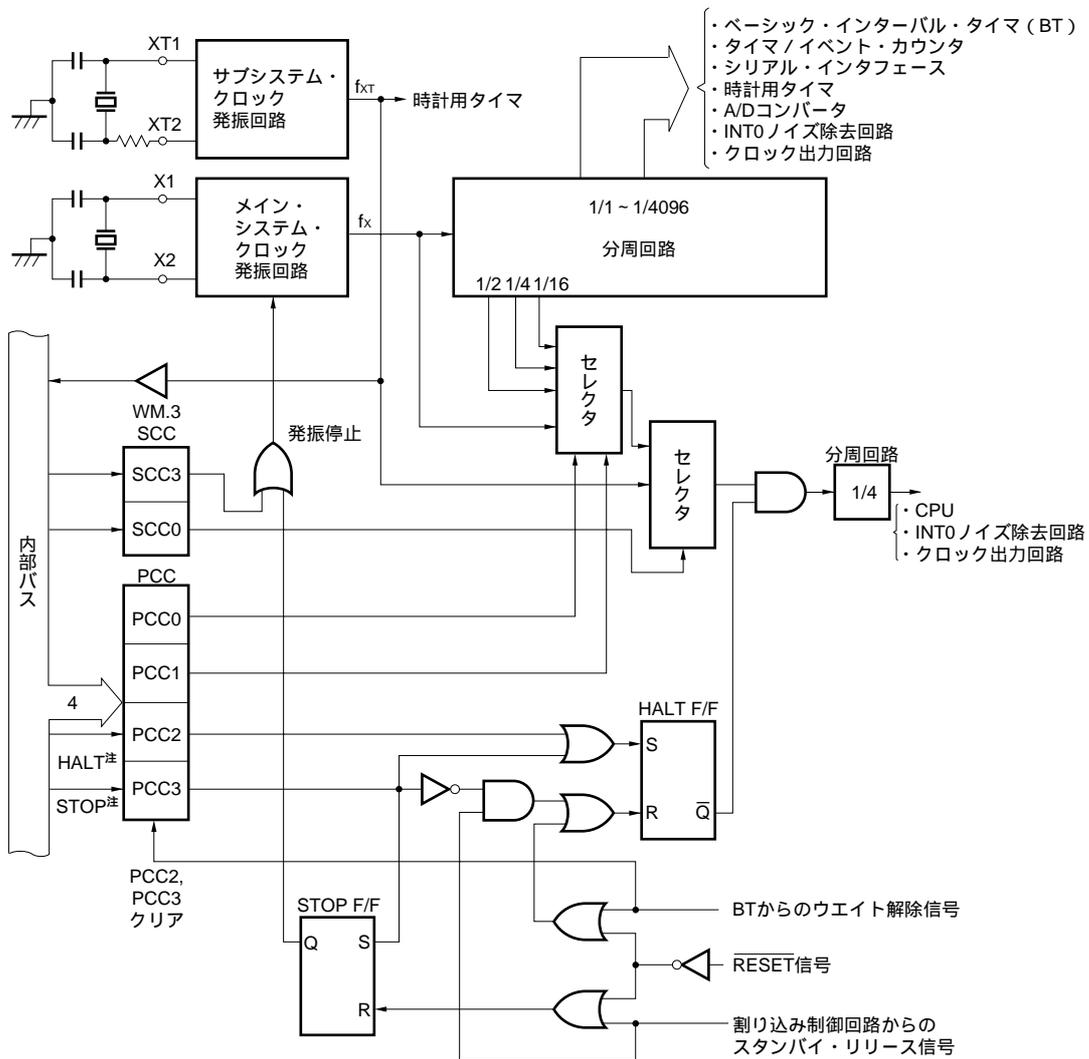
## 5.2 クロック発生回路

クロック発生回路はCPUおよび周辺ハードウェアに各種クロックを供給し、CPUの動作モードを制御する回路です。

### 5.2.1 クロック発生回路の構成

クロック発生回路は図5-11のように構成されています。

図5-11 クロック発生回路のブロック図



注 命令の実行

備考1 .  $f_x$  = メイン・システム・クロック周波数

2 .  $f_{XT}$  = サブシステム・クロック周波数

3 .  $f_{CPU}$  = CPUクロック

4 . PCC : プロセッサ・クロック・コントロール・レジスタ

5 . SCC : システム・クロック・コントロール・レジスタ

6 .  $t_{CY}$  の1クロック・サイクルは命令の1マシン・サイクルです。

## 5.2.2 クロック発生回路の機能と動作

クロック発生回路は次に示す各種のクロックを発生し、かつ、スタンバイ・モード等のCPUの動作モードを制御します。

メイン・システム・クロック  $f_x$   
 サブシステム・クロック  $f_{xt}$   
 CPUクロック  
 周辺ハードウェアへのクロック

クロック発生回路の動作はプロセッサ・クロック・コントロール・レジスタ (PCC) とシステム・クロック・コントロール・レジスタ (SCC) により決定され、以下のような機能、動作となります。

- ( a )  $\overline{\text{RESET}}$ 信号発生によりメイン・システム・クロックの最低速モード ( 10.7  $\mu\text{s}$  : 6.00 MHz動作時 ) が選択されます ( PCC = 0, SCC = 0 )。
- ( b ) メイン・システム・クロックを選択した状態でPCCの設定により4段階のCPUクロックを選択することができます ( 0.67, 1.33, 2.67, 10.7  $\mu\text{s}$  : 6.00 MHz動作時 )。
- ( c ) メイン・システム・クロックを選択した状態でSTOPモード, HALTモードの2つのスタンバイ・モードが使用できます。
- ( d ) SCCによりサブシステム・クロックを選択し, 超低速, 低消費電流 ( 122  $\mu\text{s}$  : 32.768 kHz動作時 ) で動作することができます。この場合PCCの設定値はCPUクロックに影響を与えません。
- ( e ) サブシステム・クロックを選択した状態で, SCCによりメイン・システム・クロックの発振を停止することができます。また, HALTモードを使用することもできます。ただし, STOPモードは使用できません ( サブシステム・クロックの発振を停止させることはできません )。
- ( f ) 周辺ハードウェアへのクロックはメイン・システム・クロックを分周して供給されますが, 時計用タイマにだけはサブシステム・クロックを直接供給することができます。このため, スタンバイ状態でも, 時計機能と時計用タイマからのクロックで動作するプザー出力機能は動作を継続することができます。
- ( g ) サブシステム・クロックを選択したときには時計用タイマは正常動作を継続することができます。また, シリアル・インタフェースおよびタイマ/イベント・カウンタは, クロックとして外部クロックまたは時計用タイマを選択した場合, 動作を継続することができます。しかし, ほかのハードウェアはメイン・システム・クロックによって動作しますので, メイン・システム・クロックを停止させたときは使用することはできません。

**(1) プロセッサ・クロック・コントロール・レジスタ (PCC)**

PCCは、下位2ビットでCPUクロックの選択、上位2ビットでCPU動作モードの制御を行う4ビット・レジスタです(図5-12参照)。

ビット3またはビット2の片方のみが“1”にセットされると、スタンバイ・モードに設定されます。スタンバイ・リリース信号によって解除されると、両ビットは自動的にクリアされ通常の動作モードとなります(詳細は第7章 **スタンバイ機能**を参照)。

PCCの下位2ビットは、4ビット・メモリ操作命令により設定します(上位2ビットは“0”としてください)。

また、ビット3とビット2は、それぞれSTOP命令、HALT命令により“1”にセットします。

STOP命令、HALT命令は、MBEの内容に関係なく常に行うことができます。

CPUクロックの選択はメイン・システム・クロックで動作しているときのみ可能です。サブシステム・クロックで動作させる場合は、PCCの下位2ビットは無効となり、 $f_{XT}/4$ に固定されます。また、STOP命令もメイン・システム・クロックで動作しているときのみ実行可能です。

**例1** . マシン・サイクルを最高速モード(0.67  $\mu$ s : 6.00 MHz動作時)にする。

```
SEL      MB15
MOV      A, #0011B
MOV      PCC, A
```

**2** . マシン・サイクルを1.91  $\mu$ s ( $f_x = 4.19$  MHz動作時)にする。

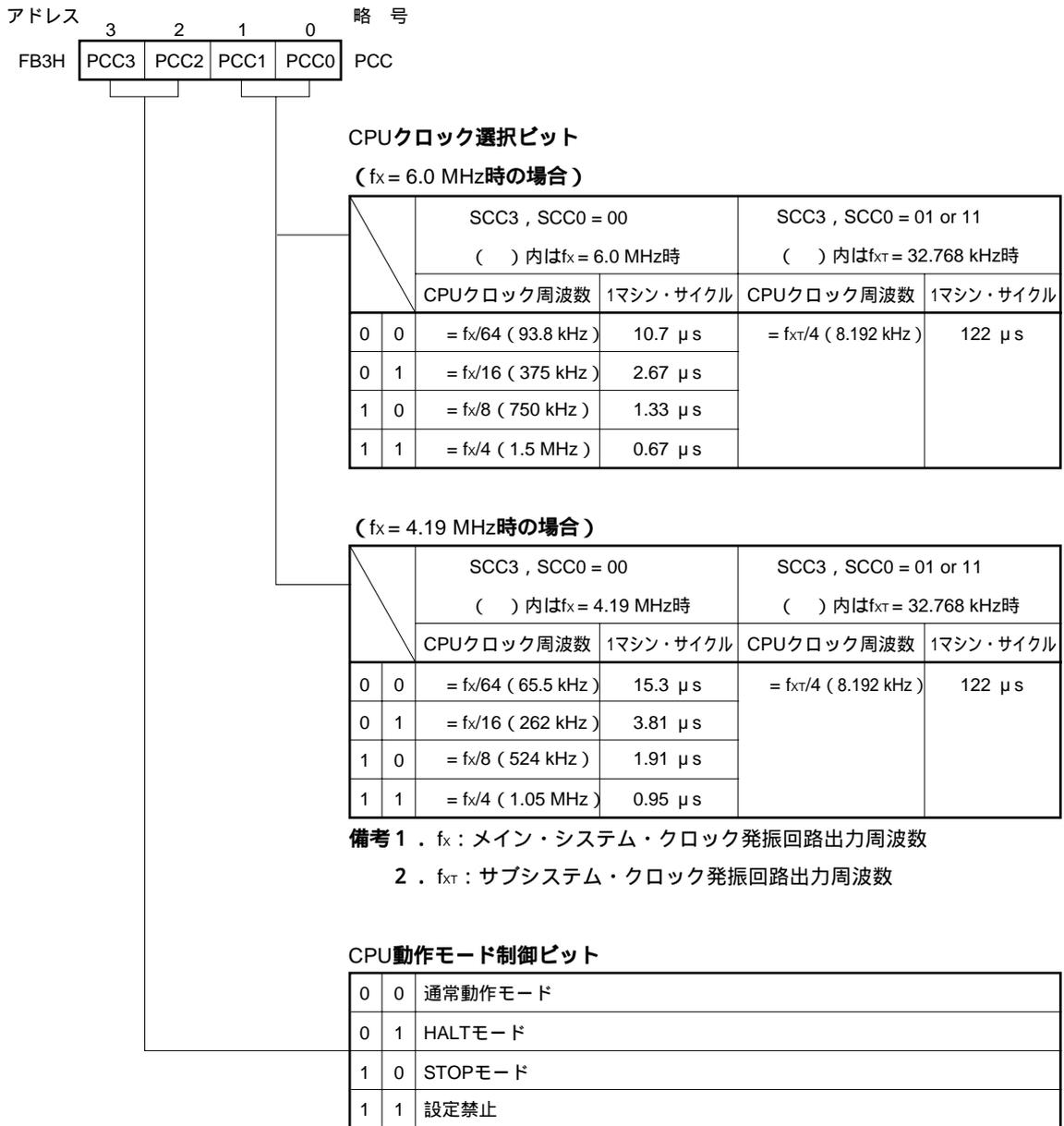
```
SEL      MB15
MOV      A, #0010B
MOV      PCC, A
```

**3** . STOPモードに設定する(STOP命令、およびHALT命令のあとには必ずNOP命令を書いてください)。

```
STOP
NOP
```

$\overline{\text{RESET}}$ 信号発生によりPCCは“0”にクリアされます。

図5 - 12 プロセッサ・クロック・コントロール・レジスタのフォーマット



(2) システム・クロック・コントロール・レジスタ (SCC)

SCCは最下位ビットでCPUクロック の選択, 最上位ビットでメイン・システム・クロックの発振停止制御を行う4ビット・レジスタです(図5-13参照)。

SCCのビット0とビット3は同じデータ・メモリ・アドレスに存在しますが, 両方のビットを同時に変更することはできません。したがって, SCCのビット0とビット3の設定はビット操作命令によって行います。なお, SCCのビット0とビット3はMBEの内容に関係なく常にビット操作できます。

SCCのビット3のセットによるメイン・システム・クロックの発振停止は, サブシステム・クロックで動作しているときにかぎり許されます。メイン・システム・クロックで動作しているときの発振停止はSTOP命令により行います。

RESET信号発生によりSCCは“0”にクリアされます。

図5-13 システム・クロック・コントロール・レジスタのフォーマット



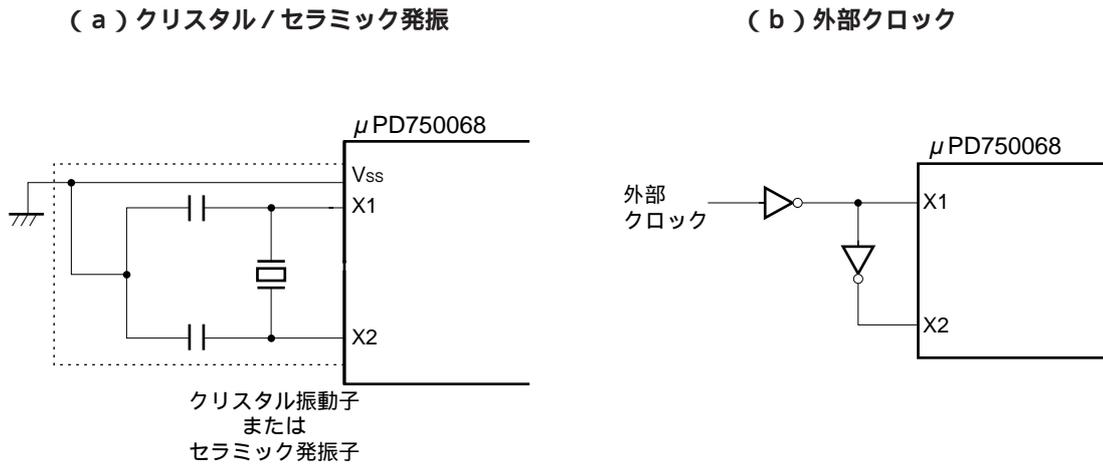
- 注意1. システム・クロックの変更には最大1/f<sub>XT</sub>の時間が必要です。したがって, メイン・システム・クロックの発振を停止させる場合は, サブシステム・クロック変更後, 表5-5に示すマシン・サイクル以上経過後, SCC. 3を1に設定してください。
2. メイン・システム・クロックで動作しているときにSCC. 3をセットして発振を停止させても正常なSTOPモードには入りません。
3. SCC. 3に“1”をセットすると, X2端子が内部でV<sub>DD</sub>に50k (TYP.)の抵抗でプルアップされます。

(3) システム・クロック発振回路

(i) メイン・システム・クロック発振回路はX1, X2端子に接続されたクリスタル振動子またはセラミック発振子によって発振します(標準: 4.194304 MHz)。

また, 外部クロックを入力することもできます。

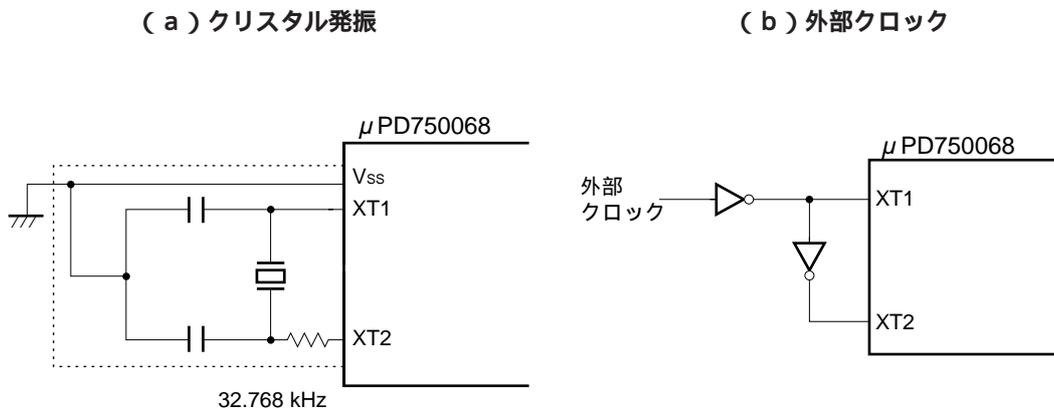
図5-14 メイン・システム・クロック発振回路の外付け回路



(ii) サブシステム・クロック発振回路はXT1, XT2端子に接続されたクリスタル振動子(標準: 32.768 kHz)によって発振します。

また, 外部クロックを入力することもできます。

図5-15 サブシステム・クロック発振回路の外付け回路



注意1．STOPモードに設定すると、X2端子が内部で $V_{DD}$ に50 k（TYP.）の抵抗でプルアップされます。

2．メイン・システム・クロックおよびサブシステム・クロック発振回路を使用する場合は、配線容量などの影響を避けるために、図5 - 14、5 - 15の破線の部分を次のように配線してください。

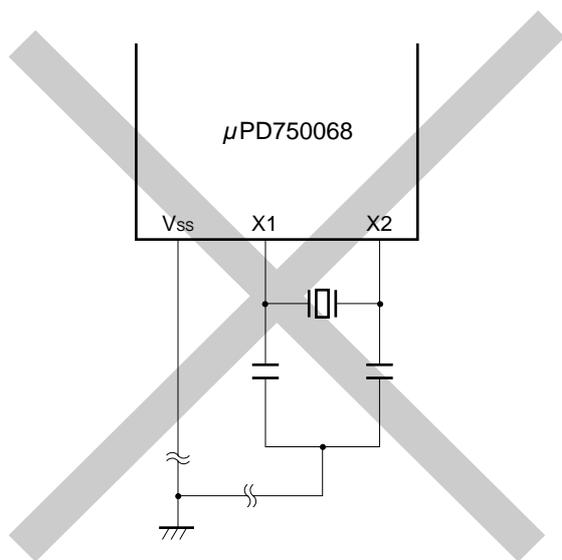
- ・配線は極力短くする。
- ・他の信号線と交差させない。
- ・変化する大電流が流れる線と接近させない。
- ・発振回路のコンデンサの接続点は、常に $V_{SS}$ と同電位となるようにする。大電流が流れるグランド・パターンには接続しない。
- ・発振回路から信号を取り出さない。

サブシステム・クロック発振回路は、低消費電流にするために増幅度の低い回路になっており、ノイズに対する誤動作がメイン・システム・クロック発振回路よりも起こりやすくなっています。したがって、サブシステム・クロックを使用する場合は、配線方法について特にご注意ください。

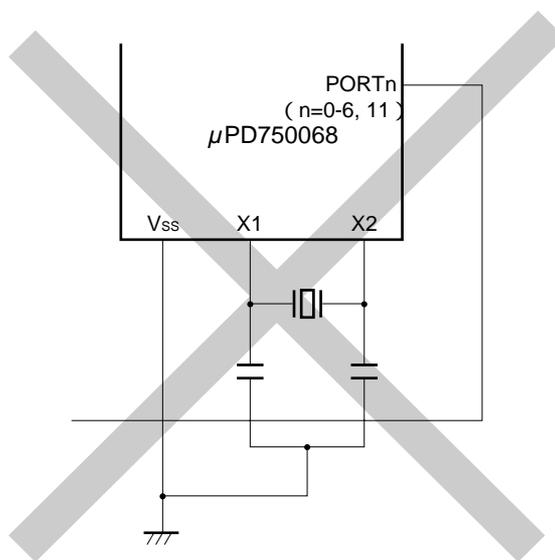
図5 - 16に発振子の接続回路の悪い例を示します。

図5 - 16 発振子の接続回路の悪い例（1/2）

( a ) 接続回路の配線が長い



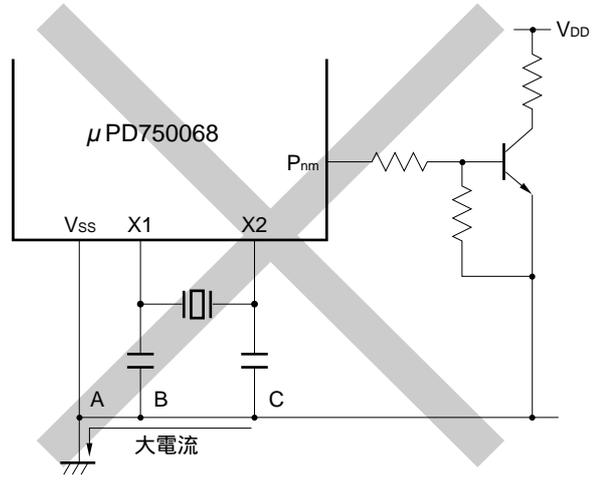
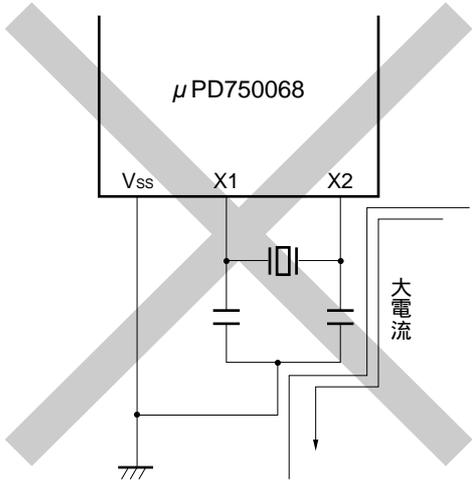
( b ) 信号線が交差している



備考 サブシステム・クロックをご使用の場合は、X1, X2をXT1, XT2と読み替えてください。また、XT2側に直列に抵抗を挿入してください。

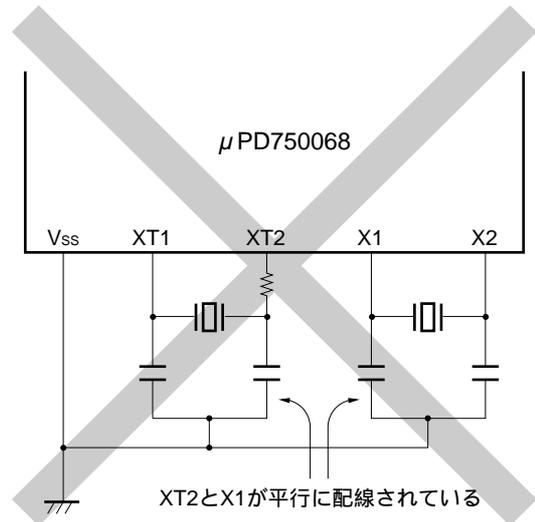
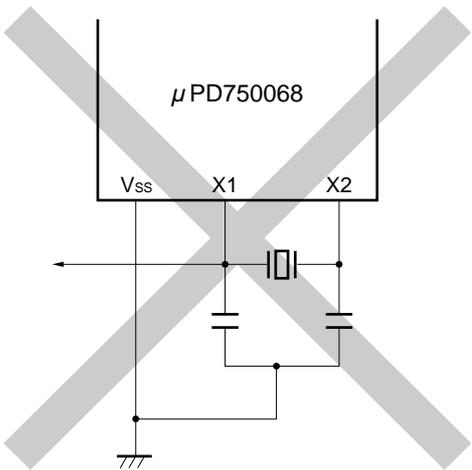
図5-16 発振子の接続回路の悪い例(2/2)

- (c) 変化する大電流が信号線に近接している (d) 発振回路部のグランド・ライン上に電流が流れる  
(A点, B点, C点の電位が変動する)



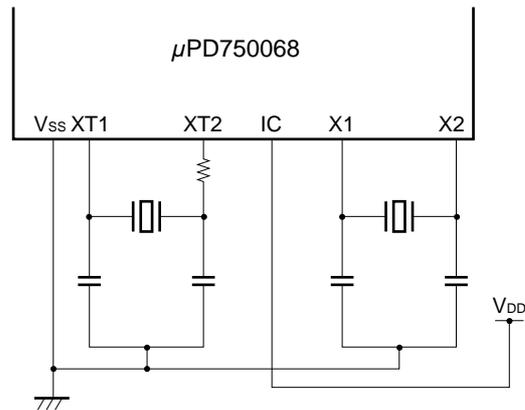
(e) 信号を取り出している

(f) メイン・システム・クロックとサブシステム・クロックの信号線が平行かつ隣接している



備考 サブシステム・クロックをご使用の場合は, X1, X2をXT1, XT2と読み替えてください。また, XT2側に直列に抵抗を挿入してください。

注意2 . 図5 - 16 ( f ) ではXT2とX1が平行に配線されています。このためX1のクロストーク・ノイズがXT2に相乗し誤動作を引き起こすことがあります。  
これを避けるために、XT2とX1の配線を平行にしないととも、XT2, X1の間にあるIC端子をV<sub>DD</sub>に接続してください。



#### (4) 分周回路

分周回路は、メイン・システム・クロック発振回路出力 ( $f_x$ ) を分周して、各種クロックを生成します。

(5) サブシステム・クロック発振回路の制御機能

μPD750068のサブシステム・クロック発振回路には次の2つの制御機能があります。

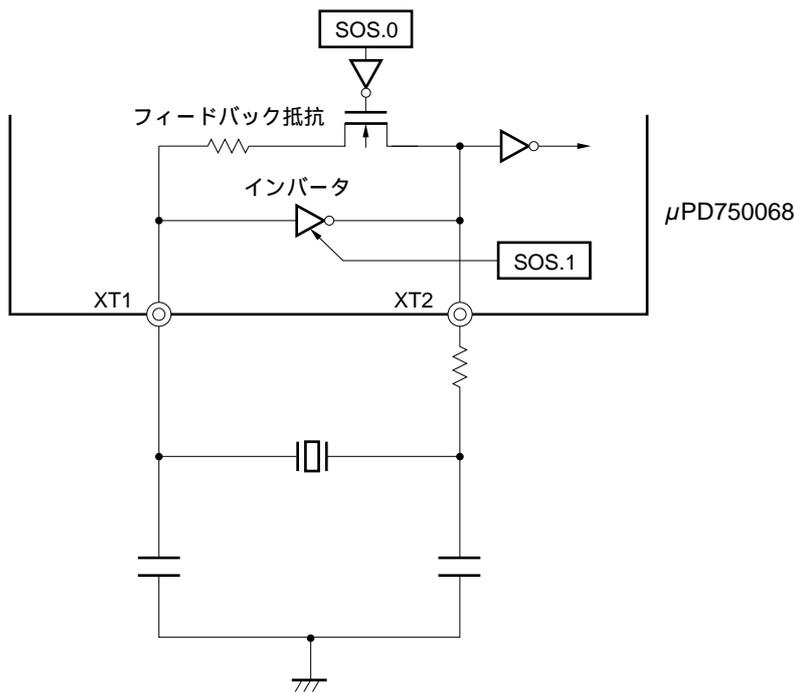
内蔵フィードバック抵抗を使用するか使用しないかをソフトウェアで選択する機能<sup>注</sup>

内蔵インバータのドライブ電流を下げて、動作電源電圧が高いとき ( $V_{DD} = 2.7V$ ) に消費電流を抑える機能

**注** サブシステム・クロックを使用しない場合は、ソフトウェアでSOS.0=1 (内蔵フィードバック抵抗を使用しない) とし、XT1を $V_{SS}$ または $V_{DD}$ に接続、XT2はオープンにしてください。サブシステム・クロック発振回路で消費する電流を抑えることができます。

それぞれの機能は、サブ発振回路コントロール・レジスタ (SOS) のビット0, 1を切り替えることで使用できます (図5-17参照)。

図5-17 サブシステム・クロック発振回路



(6) サブ発振回路コントロール・レジスタ (SOS)

SOSレジスタは内蔵フィードバック抵抗を使用するかどうかの選択と、内蔵インバータのドライブ電流の制御をするレジスタです (図5-18参照)。

RESET信号の発生により全ビットが0にクリアされます。次にSOSレジスタのフラグの機能概要を示します。

(a) SOS. 0 (フィードバック抵抗カット・フラグ)

μPD750068ではSOS. 0の切り替えによって、ソフトウェアで、内蔵フィードバック抵抗を使用するかしないかの選択ができます。

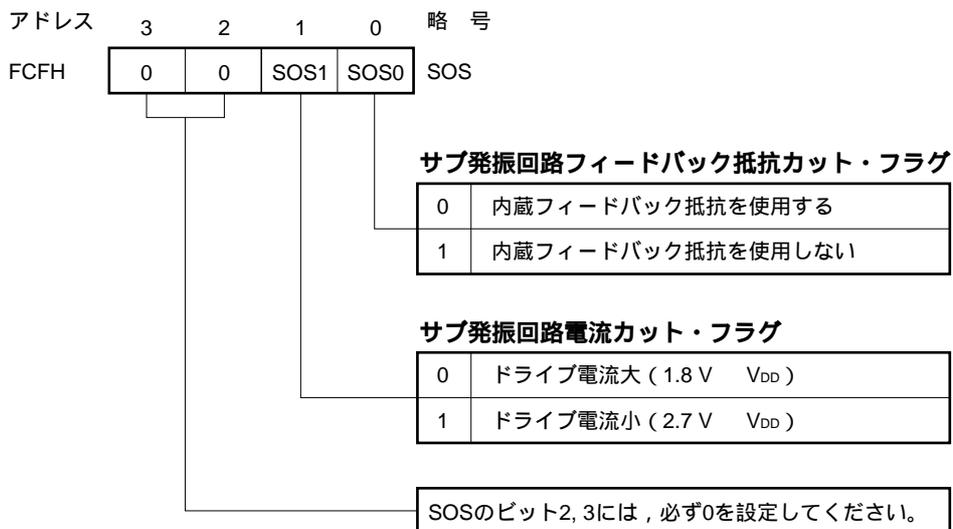
発振子を使用しない場合に“1”を設定すると、フィードバック回路をオフして消費電流を下げることができます。発振子を使用する場合には、必ず“0”（フィードバック回路オン）に設定してください。

(b) SOS. 1 (ドライブ能力の切り替えフラグ)

μPD750068のサブシステム・クロック発振回路の内蔵インバータは、低電圧対応 ( $V_{DD} = 1.8\text{ V}$  使用可能) のためドライブ電流が大きくなっています。このままですと、高い電源電圧で使うとき ( $V_{DD} = 2.7\text{ V}$ )、電源電流が大きくなってしまいます。このときSOS. 1に“1”をセットすることにより、インバータのドライブ電流を下げ、電源電流を下げるができます。

ただし、 $V_{DD} = 2.7\text{ V}$ 未満の場合、“1”に設定すると、ドライブ電流不足で発振が停止することがあります。 $V_{DD} = 2.7\text{ V}$ 未満で使用する場合は、必ず“0”に設定してください。

図5 - 18 サブ発振回路コントロール・レジスタ (SOS) のフォーマット



**備考** サブシステム・クロックを使用する必要のない場合は、XT1, XT2端子とSOSレジスタを次のように処置してください。

XT1 :  $V_{SS}$ または $V_{DD}$ に接続

XT2 : オープン

SOS : 00x1B ( x : don't care )

### 5.2.3 システム・クロックとCPUクロックの設定

#### (1) システム・クロックとCPUクロックの切り替えに要する時間

システム・クロックとCPUクロックはSCCの最下位ビットおよびPCCの下位2ビットにより切り替えることができますが、この切り替えは、レジスタの書き換え後すぐには行われず、一定のマシン・サイクルの間は切り替え前のクロックで動作します。したがって、メイン・システム・クロックの発振を停止させる場合には、この切り替え時間経過後、STOP命令の実行またはSCCのビット3のセットを行う必要があります。

表5-5 システム・クロック，CPUクロックの切り替えに要する最大時間

切り替え前の設定値			切り替え後の設定値														
SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0	SCC0	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	x	x
0	0	0	/			1マシン・サイクル			1マシン・サイクル			1マシン・サイクル			$\frac{f_x}{64f_{XT}}$ マシン・サイクル (3マシン・サイクル)		
	0	1				4マシン・サイクル			4マシン・サイクル			4マシン・サイクル			$\frac{f_x}{16f_{XT}}$ マシン・サイクル <sup>注</sup> (12マシン・サイクル)		
	1	0				8マシン・サイクル			8マシン・サイクル			8マシン・サイクル			$\frac{f_x}{8f_{XT}}$ マシン・サイクル (23マシン・サイクル)		
	1	1				16マシン・サイクル			16マシン・サイクル			16マシン・サイクル			$\frac{f_x}{4f_{XT}}$ マシン・サイクル (46マシン・サイクル)		
1	x	x	1マシン・サイクル			1マシン・サイクル <sup>注</sup>			1マシン・サイクル			1マシン・サイクル					

注 ツールではエミュレーションできません。

**注意** 発振子の周囲温度，負荷容量の性能のばらつきなどの条件により， $f_x$ ， $f_{XT}$ の値は変化します。特に $f_x$ が公称値より高い場合や， $f_{XT}$ が公称値より低い場合には，表中の $\frac{f_x}{64f_{XT}}$ ， $\frac{f_x}{16f_{XT}}$ ， $\frac{f_x}{8f_{XT}}$ ， $\frac{f_x}{4f_{XT}}$ の式で求められるマシン・サイクルは， $f_x$ ， $f_{XT}$ の公称値で求めたマシン・サイクルより大きくなります。したがって，システム・クロック，CPUクロックの切り替えに必要なウエイト時間を設定するときは， $f_x$ ， $f_{XT}$ の公称値で求めたマシン・サイクルよりも長くしてください。

備考1．( )内は， $f_x = 6.0$  MHz， $f_{XT} = 32.768$  kHzの場合。

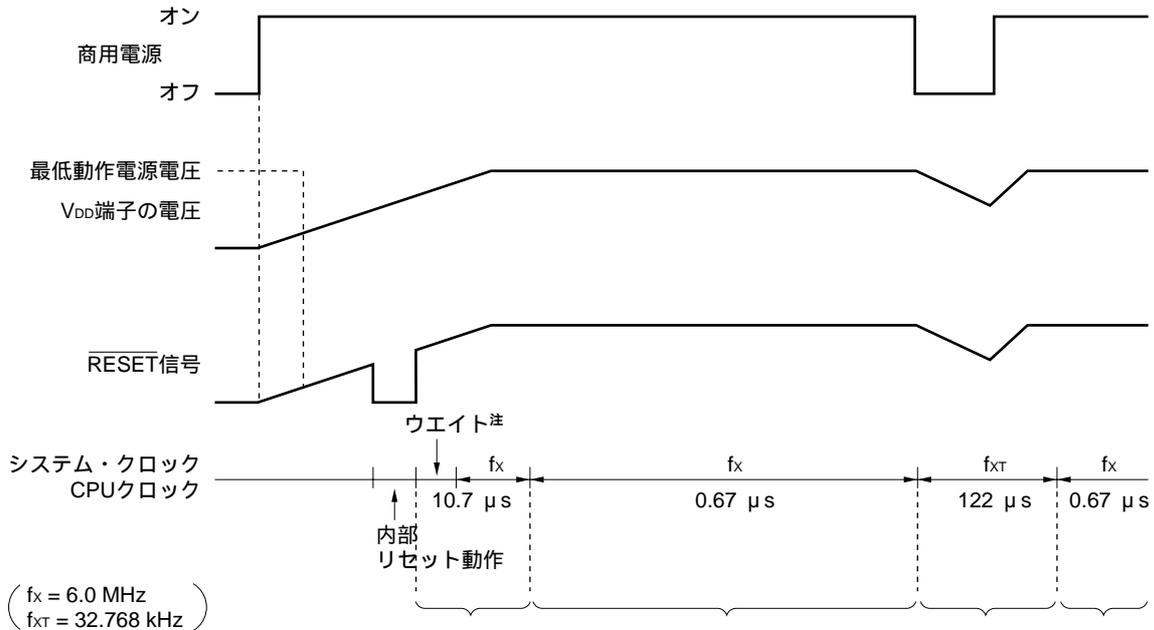
2．x : don't care

3．CPUクロック は，内部CPUに供給されるクロックで，この逆数が最小命令時間（このマニュアルでは1マシン・サイクルと定義する）となります。

(2) システム・クロックとCPUクロックの切り替え手順

システム・クロックとCPUクロックの切り替えを図5 - 19に従って説明します。

図5 - 19 システム・クロックとCPUクロックの切り替え



RESET信号発生により、発振安定時間確保のためのウエイト時間<sup>注</sup>後、CPUはメイン・システム・クロックの最低速（10.7 μs：6.0 MHz動作時，15.3 μs：4.19 MHz動作時）で動作をスタートします。

VDD端子の電圧が、最高速で動作できる電圧まで上昇するのに十分な時間経過後、PCCを書き換え最高速で動作します。

商用電源がオフしたことを割り込み入力（INT4を使うと効果的です）により検出し、SCCのビット0を“1”にしサブシステム・クロックで動作します（このとき、あらかじめ、サブシステム・クロックの発振が開始していなければなりません）。そしてサブシステム・クロックに切り替わるのに必要な時間（46マシン・サイクル）経過後、SCCのビット3を“1”にセットしてメイン・システム・クロックの発振を停止します。

商用電源が復帰したことを割り込みなどで検出したあと、SCCのビット3を“0”にクリアして、メイン・システム・クロックの発振を開始し、発振が安定するのに必要な時間経過後、SCCのビット0を“0”にクリアし最高速で動作します。

注 マスク・オプションにより、次の2つの時間を選択することができます。

$2^{17}/f_x$ （21.8 ms：6.0 MHz動作時，31.3 ms：4.19 MHz動作時）

$2^{15}/f_x$ （5.46 ms：6.0 MHz動作時，7.81 ms：4.19 MHz動作時）

ただし、μPD75P0076にはマスク・オプションがなく、 $2^{15}/f_x$ に固定されています。

## 5.2.4 クロック出力回路

### (1) クロック出力回路の構成

クロック出力回路は、図5-20のように構成されています。

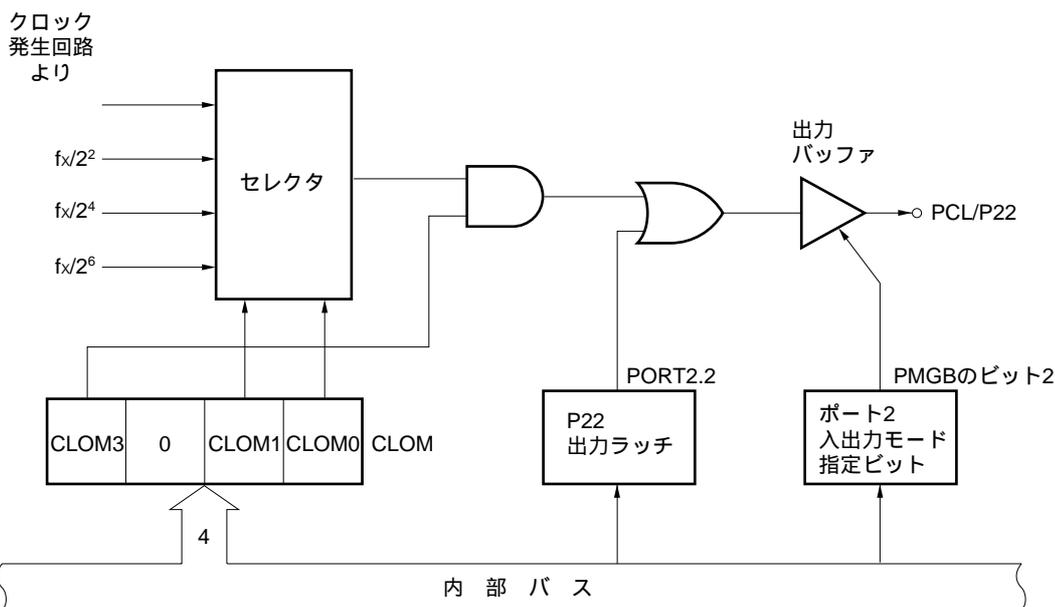
### (2) クロック出力回路の機能

クロック出力回路は、P22/PCL端子からクロック・パルスを出力するための回路で、リモコン波形出力への応用や周辺LSIにクロック・パルスを供給する場合などに利用します。

クロック・パルスを出力するときは、次の手順で行います。

- (a) クロック出力周波数を選択する。クロックの出力は禁止。
- (b) P22の出力ラッチに0を書き込む。
- (c) ポート2の入出力モードを出力に設定。
- (d) クロックの出力を許可。

図5-20 クロック出力回路のブロック図



**備考** クロックの出力の許可/禁止を切り替えるときに、幅の短いパルスが出力されないように考慮されています。

(3) クロック出力モード・レジスタ (CLOM)

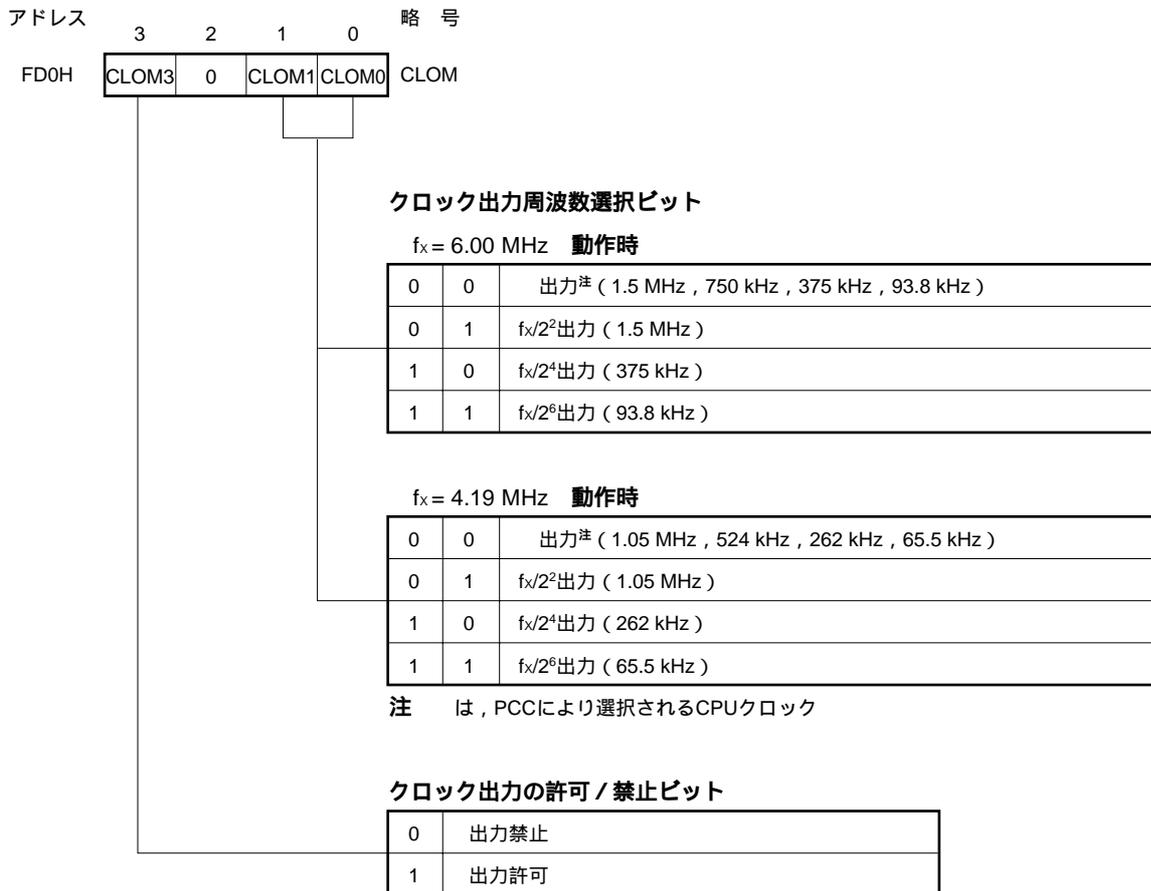
CLOMは、クロック出力を制御する4ビット・レジスタです。

CLOMは、4ビット・メモリ操作命令で設定します。

例 CPUクロック をPCL/P22端子より出力  
 SEL MB15 ;またはCLR1 MBE  
 MOV A, #1000B  
 MOV CLOM, A

RESET信号発生によりCLOMは“0”にクリアされ、クロック出力は禁止状態になります。

図5 - 21 クロック出力モード・レジスタのフォーマット



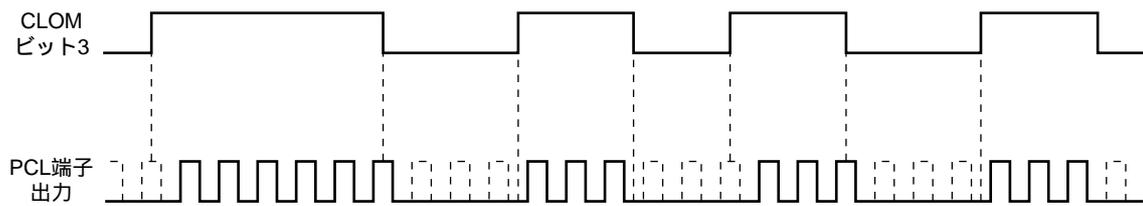
注意 CLOMのビット2には、必ず0を設定してください。

#### (4) リモコン波形出力への応用例

μPD750068のクロック出力機能は、リモコン波形出力に応用可能です。リモコン波形出力のキャリア周波数は、クロック出力モード・レジスタのクロック周波数選択ビットにより選択します。パルスの出力の許可/禁止はクロック出力の許可/禁止ビットをソフトウェアで制御することにより行います。

クロック出力の許可/禁止切り替え時に、幅の狭いパルスは出力されないよう考慮されています。

図5 - 22 リモコン波形出力応用例



## 5.3 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ

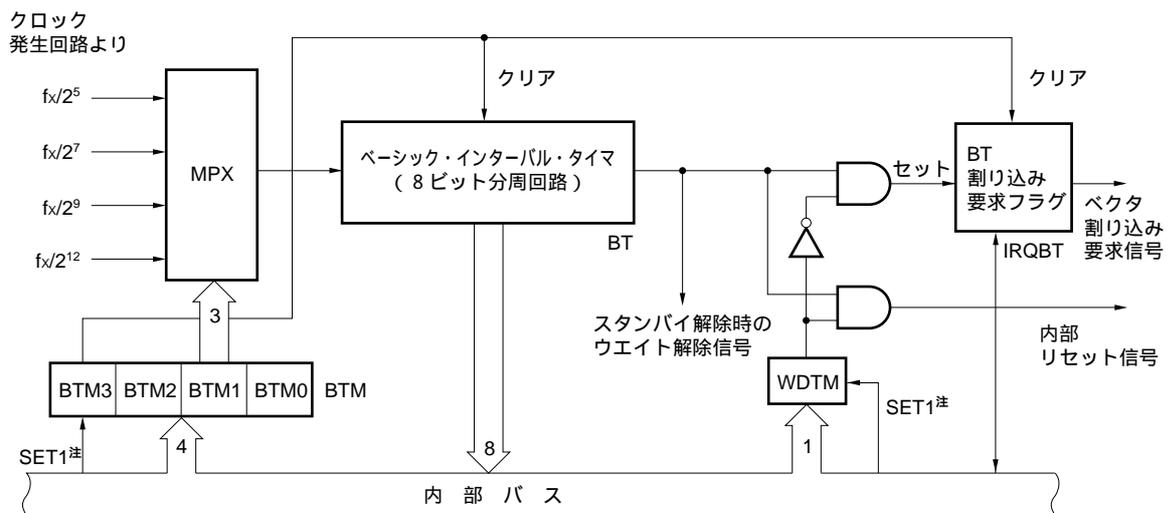
μPD750068は、8ビット・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマを備えており、次のような機能があります。

- (a) 基準時間割り込みを発生するインターバル・タイマ動作
- (b) プログラムの暴走を検出し、CPUをリセットするウォッチドッグ・タイマ動作
- (c) スタンバイ・モード解除時のウエイト時間の選択とカウント
- (d) カウント内容の読み出し

### 5.3.1 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマの構成

ベーシック・インターバル・タイマ/ウォッチドッグ・タイマは、図5-23のように構成されています。

図5-23 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマのブロック図



注 命令の実行

### 5.3.2 ベーシック・インターバル・タイマ・モード・レジスタ (BTM)

BTMは、ベーシック・インターバル・タイマ (BT) の動作を制御する4ビット・レジスタです。

BTMは、4ビット・メモリ操作命令により設定します。

ビット3は、ビット操作命令により単独で操作できます。

**例** 割り込み発生インターバルを1.37 ms (6.00 MHz) に設定

```
SEL    MB15          ;またはCLR1 MBE
CLR1   WDTM
MOV    A, #1111B
MOV    BTM, A       ;BTM 1111B
```

ビット3を“1”に設定すると、BTの内容がクリアされると同時に、ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ割り込み要求フラグ (IRQBT) もクリアされます (ベーシック・インターバル・タイマ/ウォッチドッグ・タイマのスタート)。

$\overline{\text{RESET}}$ 信号発生により内容は“0”にクリアされ、割り込み要求信号の発生インターバル時間は最長に設定されます。

図5 - 24 ベーシック・インターバル・タイマ・モード・レジスタのフォーマット



### 5.3.3 ウォッチドッグ・タイマ許可フラグ (WDTM)

WDTMは、オーバフローによりリセット信号を発生することを許可するフラグです。

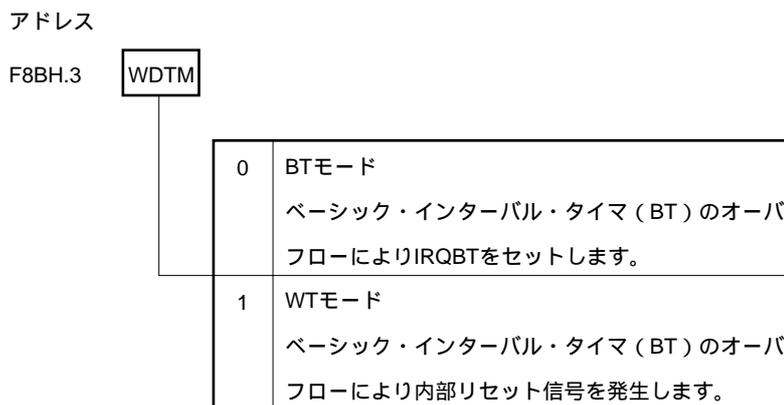
WDTMは、ビット操作命令により設定します。1度セットすると、命令ではクリアできません。

例 ウォッチドッグ・タイマ機能の設定

```
SEL      MB15          ;またはCLR1 MBE
SET1     WDTM
      :
SET1     BTM.3        ;BTMのビット3を“1”にセット
```

$\overline{\text{RESET}}$ 信号発生により内容は0にクリアされます。

図5 - 25 ウォッチドッグ・タイマ許可フラグ (WDTM) のフォーマット



### 5.3.4 ベーシック・インターバル・タイマの動作

WDTMに“0”をセットすると、ベーシック・インターバル・タイマ(BT)のオーバフローにより割り込み要求フラグ(IRQBT)をセットし、インターバル・タイマとして動作します。BTは、クロック発生回路からのクロックによって常にインクリメントされ、カウント動作を停止することはできません。

割り込み発生インターバルは、BTMの設定により4通りの時間が選択できます(図5-24参照)。

BTMのビット3を“1”にセットすることにより、BTとIRQBTをクリアすることができます(インターバル・タイマとしてのスタート指示)。

BTは、8ビット操作命令でカウント状態を読み出すことができます。なお、データの書き込みはできません。

タイマ動作は、次のように行ってください(、の設定は同時に行ってもかまいません)。

BTMにインターバル時間をセットする。

BTMのビット3に“1”をセットする。

**例** 1.37 ms (6.00 MHz動作時) ごとに割り込みを発生する。

```
SET1  MBE
SEL   MB15
MOV   A, #1111B
MOV   BTM, A           ; 時間設定とスタート
EI    ; 割り込みを許可
EI    IEBT             ; BT 割り込みを許可
```

### 5.3.5 ウォッチドッグ・タイマの動作

ベーシック・インターバル・タイマ/ウォッチドッグ・タイマは、WDTMに“1”をセットすると、ベーシック・インターバル・タイマ（BT）のオーパフローにより内部リセット信号を発生するウォッチドッグ・タイマとして動作します。ただし、STOP命令解除後の発振ウエイト時間によるタイマのオーパフローでは、リセットはかかりません（なお、WDTMは一度“1”にセットすると、リセット以外にクリアすることはできません）。BTは、クロック発生回路からのクロックによって常にインクリメントされ、カウント動作を停止することはできません。

ウォッチドッグ・タイマ・モードでは、BTのオーパフローするインターバル時間を利用して、プログラムの暴走を検出します。このインターバル時間は、BTMのビット2-0の設定により4通りの時間が選択できます（図5-24参照）。これらの中からユーザ・システムに応じて暴走検出に必要な時間を決めてください。インターバル時間を設定しておいて、プログラムをその時間内に実行できる単位に分割し、それぞれの単位の最後でBTをクリアする命令を実行させるようにします。そうすると、設定時間内にこのBTクリアを実行する命令にたどりつかなければ（順調にプログラムの実行が進んでいなければ＝暴走）BTはオーパフローし、内部リセット信号が発生してプログラムを強制終了させてしまいます。この結果、内部リセットがかかったということはプログラムの暴走が起きたことを示し、その検出ができたことになります。

ウォッチドッグ・タイマの設定手順は、次のように行ってください（、の設定は同時に行ってもかまいません）。

BTMにインターバル時間をセットする。	} 初期設定
BTMのビット3に“1”をセットする。	
WDTMに“1”をセットする。	

～ を設定したあとは、インターバル時間以内にBTMのビット3に“1”をセットする。

例 5.46 msのウォッチドッグ・タイマとして使用する（6.00 MHz動作時）。

プログラムをBTMの設定時間（5.46 ms）以内に処理が終了するいくつかのモジュールに分割し、各モジュールの終わりでBTをクリアする。暴走した場合、BTが設定時間内にクリアされないためオーバフローしてしまい、内部リセット信号が発生する。

初期設定：

```

SET1    MBE
SEL     MB15
MOV     A, #1101B
MOV     BTM, A      ; 時間設定とスタート
SET1    WDTM       ; ウォッチドッグ・タイマを許可
      ⋮
    
```

（以後、5.46 msごとにBTMのビット3に“1”をセットする。）

モジュール1：

```

      ⋮
      ⋮
      ⋮
SET1    MBE
SEL     MB15
SET1    BTM.3
    
```

5.46 ms以内で  
処理完了

モジュール2：

```

      ⋮
      ⋮
      ⋮
SET1    MBE
SEL     MB15
SET1    BTM.3
    
```

5.46 ms以内で  
処理完了

⋮

### 5.3.6 その他の機能

ベーシック・インターバル・タイマ/ウォッチドッグ・タイマは、ベーシック・インターバル・タイマ動作、ウォッチドッグ・タイマ動作に関係なく、次の機能があります。

スタンバイ・モード解除後ウエイト時間の選択とカウント  
 カウント内容の読み出し

#### (1) STOPモード解除後のウエイト時間の選択とカウント

STOPモードの解除時には、システム・クロックの発振が安定するまでの発振安定時間を得るため、ベーシック・インターバル・タイマ (BT) のオーバフローまでの間CPUの動作を停止するウエイト機能があります。

$\overline{\text{RESET}}$ 信号発生後のウエイト時間は、マスク・オプションにより固定されますが、割り込み発生によるSTOPモード解除時は、BTMの設定によりウエイト時間を選択できます。その場合のウエイト時間は図5 - 24のインターバル時間と同じになります。BTMへの設定は、STOPモードに設定する前に行います (詳しくは第7章 **スタンバイ機能**を参照)。

**例** STOPモードを割り込みで解除したときのウエイト時間を5.46 msに設定する (6.00 MHz動作時)。

```
SET1  MBE
SEL   MB15
MOV   A, #1101B
MOV   BTM, A      ; 時間設定
STOP                ; STOPモード設定
NOP
```

#### (2) カウント動作の読み出し

ベーシック・インターバル・タイマ (BT) は、8ビット操作命令でカウント状態を読み出すことができます。なお、データの書き込みはできません。

**注意** BTのカウント内容を読み出す場合は、カウント更新中の不安定なデータを読み出ししてしまうのを防ぐため、読み出し命令は2回実行してください。そして、読み出した2つの内容を比較して、妥当な値であればあとの方を読み出し結果とし、まったく異なれば初めからやり直すようにしてください。

例1 . BTのカウンタ内容の読み出し。

```
SET1 MBE
SEL MB15
MOV HL, #BT ; BTのアドレスをHLにセット
LOOP: MOV XA, @HL ; 1回目の読み出し
MOV BC, XA
MOV XA, @HL ; 2回目の読み出し
SKE XA, BC
BR LOOP
```

例2 . INT4割り込み（両エッジ検出）へ入力されるパルスのハイ・レベルの幅を設定する（パルス幅はBTの設定値を越えないものとする。また、BTMの設定値は5.46 ms以上（6.00 MHz動作時）とする）。

< INT4割り込みルーチン（MBE = 0） >

```
LOOP:  MOV   XA, BT      ; 1回目の読み出し
        MOV   BC, XA     ; データの格納
        MOV   XA, BT     ; 2回目の読み出し
        SKE   A, C
        BR    LOOP
        MOV   A, X
        SKE   A, B
        BR    LOOP
        SKT   PORT0.0   ; P00 = 1?
        BR    AA        ; NO
        MOV   XA, BC     ; データをデータ・メモリに格納する
        MOV   BUFF, XA
        CLR1  FLAG      ; データあり。フラグをクリア
        RETI
AA:     MOV   HL, #BUFF
        MOV   A, C
        SUBC  A, @HL
        INCS  L
        MOV   C, A
        MOV   A, B
        SUBC  A, @HL
        MOV   B, A
        MOV   XA, BC
        MOV   BUFF, XA  ; データを格納する
        SET1  FLAG      ; データあり。フラグをセット
        RETI
```

## 5.4 時計用タイマ

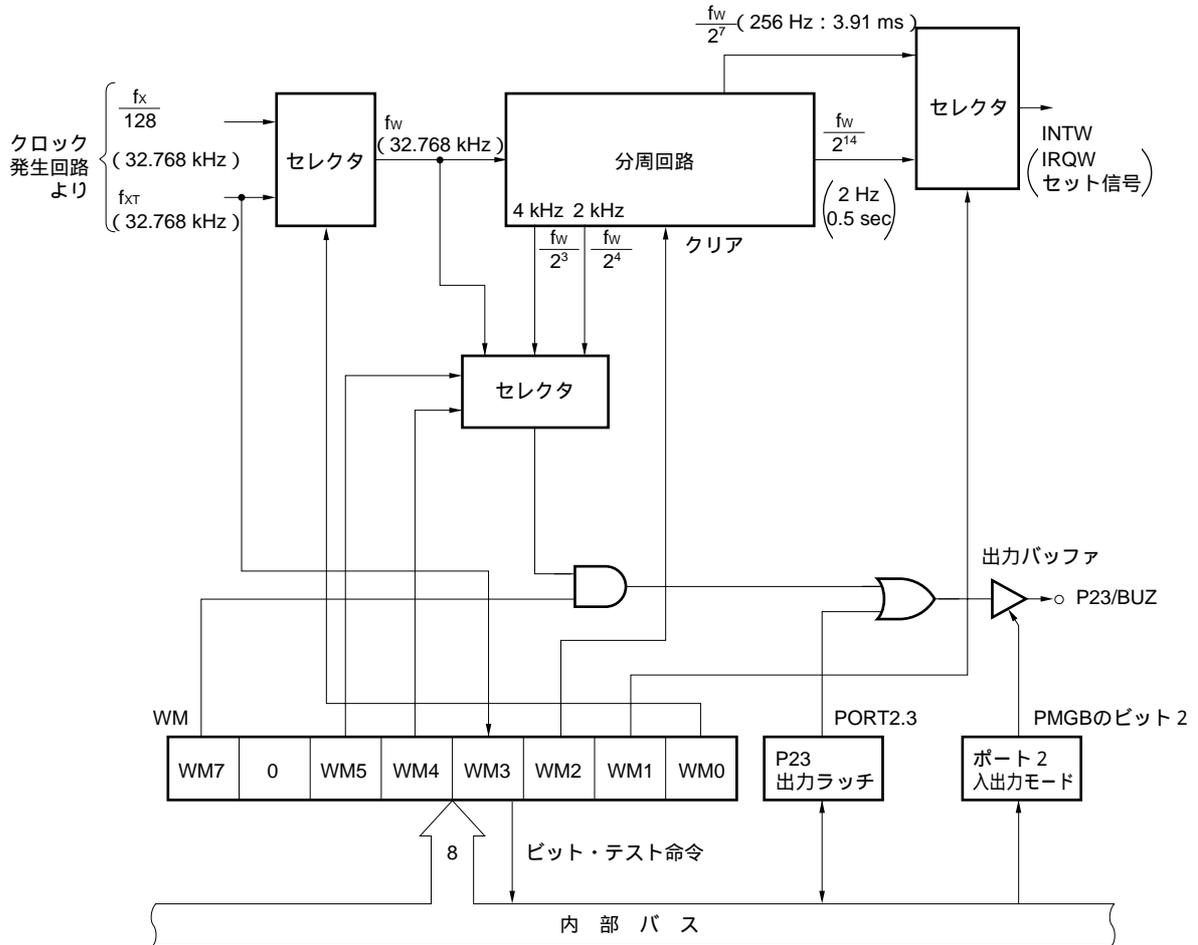
μPD750068は時計用タイマを1チャンネル内蔵しています。時計用タイマには次のような機能があります。

- (a) 0.5秒の時間間隔でテスト・フラグ (IRQW) をセットします。  
IRQWによりスタンバイ・モードの解除ができます。
- (b) メイン・システム・クロック (4.194304 MHz) とサブシステム・クロック (32.768 kHz) のいずれでも0.5秒の時間間隔を作ることができます。
- (c) 早送りモードにより128倍 (3.91 ms) の時間間隔となり、プログラムのディバグや検査に便利です。
- (d) 任意の周波数 (2.048, 4.096, 32.768 kHz) をP23/BUZ端子に出力することができ、ブザー音発生や、システム・クロック発振周波数のトリミングに使用できます。
- (e) 分周回路のクリアをすることによって、時計をゼロ秒スタートできます。
- (f) 0.5秒のクロックをタイマ/イベント・カウンタのクロック・ソースとして使用すると、最長約9時間 (タイマ0, 1 使用時) まで、スタンバイ・モードを継続し、超低消費モードにすることができます。

### 5.4.1 時計用タイマの構成

時計用タイマは図5 - 26のように構成されています。

図5 - 26 時計用タイマのブロック図



( )内は $f_x = 4.194304$  MHz,  $f_{XT} = 32.768$  kHzの場合です。

## 5.4.2 時計モード・レジスタ

時計モード・レジスタ (WM) は、時計用タイマを制御する 8 ビット・レジスタです。図 5 - 27にそのフォーマットを示します。

時計モード・レジスタは、ビット 3を除き 8 ビット操作命令により設定されます。ビット 3は、XT1端子の入力レベルのテスト用のビットで、ビット・テストによりXT1端子への入力レベルをテストすることができます。データを書き込むことはできません。

$\overline{\text{RESET}}$ 信号発生により、ビット 3を除く全ビットは“ 0 ”にクリアされます。

**例** メイン・システム・クロック (4.19 MHz) で時間を作る。ブザー出力を許可。

```
CLR1  MBE
MOV   XA, #84H
MOV   WM, XA    ; WMセット
```

図5 - 27 時計モード・レジスタのフォーマット

アドレス	7	6	5	4	3	2	1	0	略号
F98H	WM7	0	WM5	WM4	WM3	WM2	WM1	WM0	WM

**BUZ出力 許可/禁止ビット**

WM7	0	BUZ出力禁止
	1	BUZ出力許可

**BUZ出力周波数選択ビット**

WM5	WM4	BUZ出力周波数
0	0	$\frac{f_w}{2^4}$ (2.048 kHz)
0	1	$\frac{f_w}{2^3}$ (4.096 kHz)
1	0	設定禁止
1	1	$f_w$ (32.768 kHz)

**XT1端子への入力レベル (ビット・テストのみ可能)**

WM3	0	XT1端子への入力は、ロウ・レベル
	1	XT1端子への入力は、ハイ・レベル

**時計動作 許可/禁止ビット**

WM2	0	時計動作停止 (分周回路クリア)
	1	時計動作可能

**動作モード選択ビット**

WM1	0	通常時計モード ( $\frac{f_w}{2^{14}}$ : 0.5秒でIRQWをセット)
	1	早送り時計モード ( $\frac{f_w}{2^7}$ : 3.91 msでIRQWをセット)

**カウント・クロック ( $f_w$ ) の選択ビット**

WM0	0	システム・クロックの分周出力 : $\frac{f_x}{128}$ を選択
	1	サブシステム・クロック : $f_{xt}$ を選択

備考 ( ) 内は、 $f_w = 32.768$  kHz時

## 5.5 タイマ/イベント・カウンタ

μPD750068は、タイマ/イベント・カウンタを2チャンネル内蔵しています。タイマ/イベント・カウンタには次のような機能があります。

- ( a ) プログラマブル・インターバル・タイマ動作
- ( b ) PTO<sub>n</sub>端子への任意の周波数の方形波出力
- ( c ) イベント・カウンタ動作
- ( d ) TIn端子入力をN分周してPTO<sub>n</sub>端子へ出力（分周回路動作）
- ( e ) シリアル・インタフェース回路へのシフト・クロック供給
- ( f ) カウント値の読み出し機能

**備考** n = 0, 1

また、タイマ/イベント・カウンタの動作は、モード・レジスタの設定によって、次に示す2種類のモードでの動作が可能です。

表5 - 6 使用可能モード一覧表

モード	チャンネル	
	チャンネル0	チャンネル1
8ビット・タイマ/イベント・カウンタ・モード		
16ビット・タイマ/イベント・カウンタ・モード		

### 5.5.1 タイマ/イベント・カウンタの構成

タイマ/イベント・カウンタは図5 - 28, 5 - 29のように構成されています。

図5-28 タイマ/イベント・カウンタ(チャンネル0)のブロック図

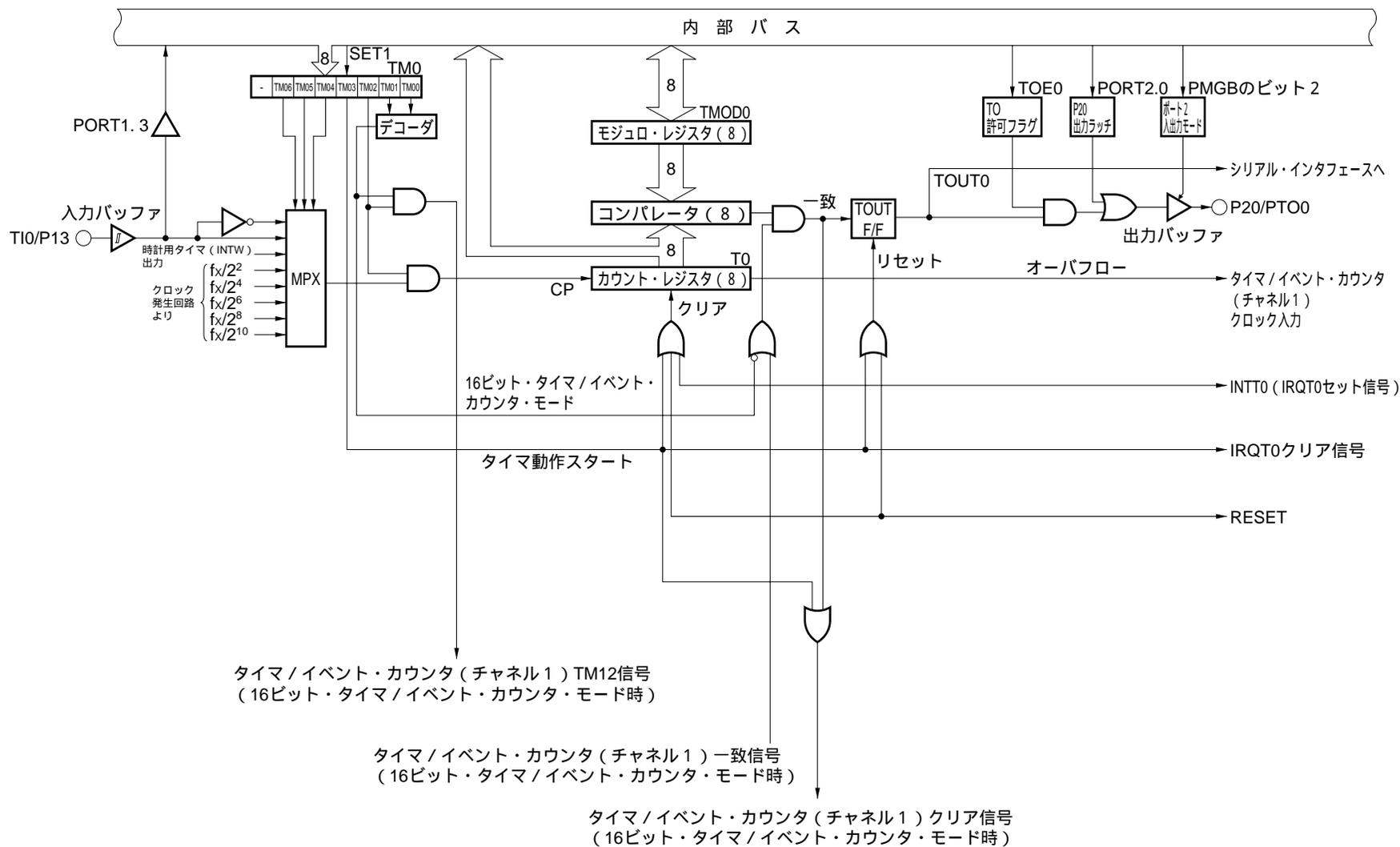
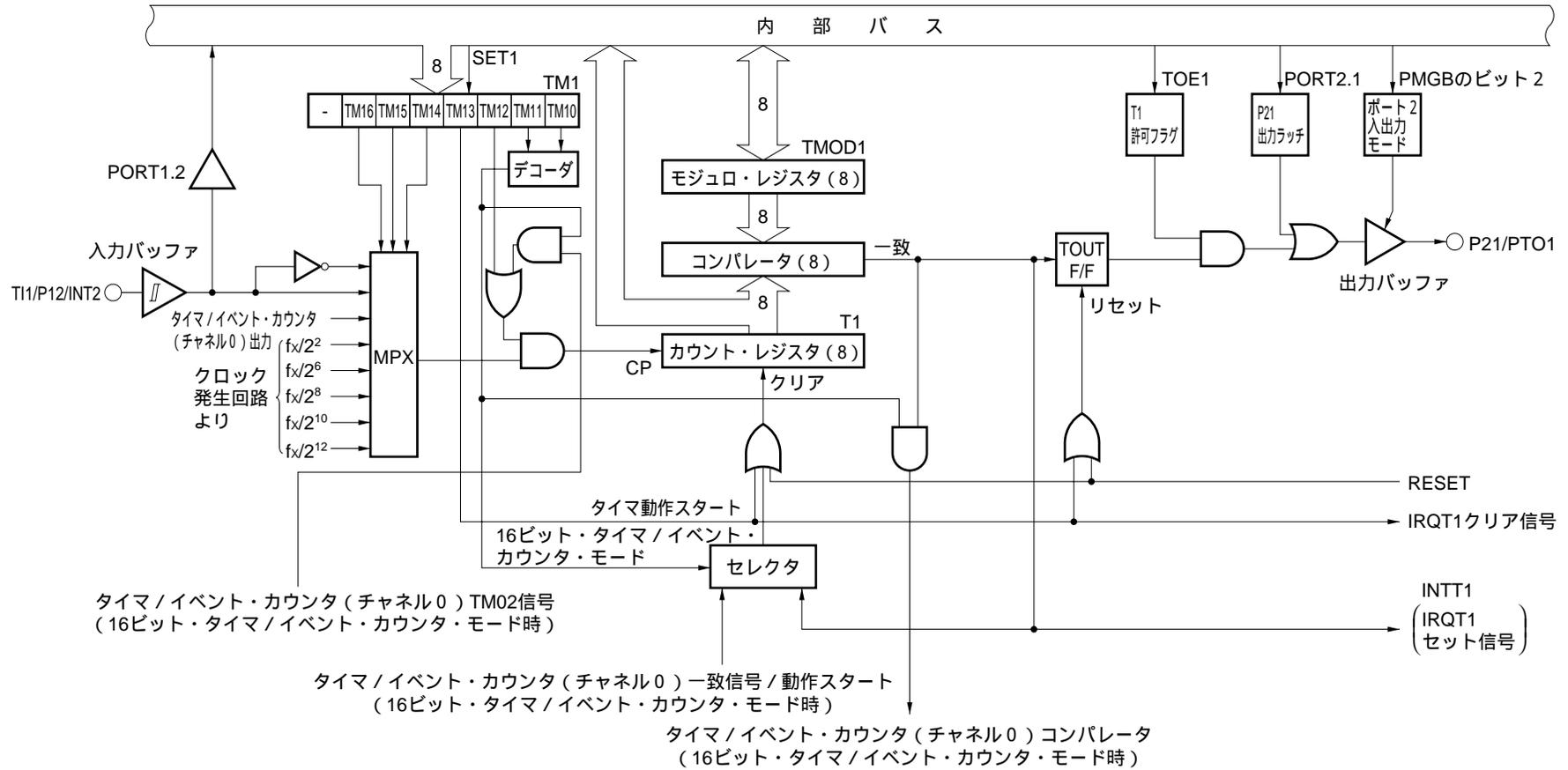


図5 - 29 タイマ/イベント・カウンタ (チャンネル1) のブロック図



## (1) タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1)

タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1) は、タイマ/イベント・カウンタを制御する 8 ビット・レジスタです。

図 5 - 30, 5 - 31にそのフォーマットを示します。

タイマ/イベント・カウンタ・モード・レジスタは 8 ビット・メモリ操作命令により設定します。

ビット 3 は、タイマ・スタート・ビットで、1 ビット単独で操作することができ、タイマ動作を開始すると自動的に“ 0 ” にリセットされます。

$\overline{\text{RESET}}$ 信号発生により、タイマ/イベント・カウンタ・モード・レジスタは全ビット“ 0 ” にクリアされます。

**例 1** . CP = 5.86 kHz (6.00 MHz動作時) のインターバル・タイマ・モードでタイマ 0 をスタート。

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #01001100B
MOV    TM0, XA             ;TM0 4CH
```

**2** . タイマ/イベント・カウンタ・モード・レジスタの設定に従い、タイマを再スタート。

```
SEL    MB15                ;またはCLR1 MBE
SET1   TMn.3              ;TMn. bit3 1 (n = 0, 1)
```

図5 - 30 タイマ/イベント・カウンタ・モード・レジスタ (チャンネル0) のフォーマット (1/2)

アドレス	7	6	5	4	3	2	1	0	略号
FA0H	-	TM06	TM05	TM04	TM03	TM02	TM01	TM00	TM0

**カウント・パルス (CP) 選択ビット**

$f_x = 6.00 \text{ MHz}$ 時の場合

TM06	TM05	TM04	カウント・パルス (CP)
0	0	0	T10立ち上がりエッジ
0	0	1	T10立ち下がりエッジ
0	1	0	$f_w/2^{14}$ または $f_w/2^7$ (時計用タイマ (INTW) 出力) 注
0	1	1	$f_x/2^2$ (1.5 MHz)
1	0	0	$f_x/2^{10}$ (5.86 kHz)
1	0	1	$f_x/2^8$ (23.4 kHz)
1	1	0	$f_x/2^6$ (93.8 kHz)
1	1	1	$f_x/2^4$ (375 kHz)

注  $f_w = f_{XT}$ または $f_x/2^7$

$f_x = 4.19 \text{ MHz}$ 時の場合

TM06	TM05	TM04	カウント・パルス (CP)
0	0	0	T10立ち上がりエッジ
0	0	1	T10立ち下がりエッジ
0	1	0	$f_w/2^{14}$ または $f_w/2^7$ (時計用タイマ (INTW) 出力) 注
0	1	1	$f_x/2^2$ (1.05 MHz)
1	0	0	$f_x/2^{10}$ (4.10 kHz)
1	0	1	$f_x/2^8$ (16.4 kHz)
1	1	0	$f_x/2^6$ (65.5 kHz)
1	1	1	$f_x/2^4$ (262 kHz)

注  $f_w = f_{XT}$ または $f_x/2^7$

**タイマ・スタート指示ビット**

TM03	“ 1 ” を書き込むことによって、カウンタ、IRQT0フラグをクリア。 ビット2が“ 1 ” にセットされていれば、カウント動作をスタート。
------	--

**動作モード**

TM02	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

図5 - 30 タイマ/イベント・カウンタ・モード・レジスタ (チャンネル0) のフォーマット (2/2)

動作モード選択ビット

TM01	TM00	モード
0	0	8ビット・タイマ/イベント・カウンタ・モード
1	0	16ビット・タイマ/イベント・カウンタ・モード
上記以外		設定禁止

図5 - 31 タイマ/イベント・カウンタ・モード・レジスタ (チャンネル1) のフォーマット (1/2)

アドレス	7	6	5	4	3	2	1	0	略号
FA8H	-	TM16	TM15	TM14	TM13	TM12	TM11	TM10	TM1

カウント・パルス (CP) 選択ビット

$f_x = 6.00 \text{ MHz}$ 時の場合

TM16	TM15	TM14	カウント・パルス (CP)
0	0	0	T11立ち上がりエッジ
0	0	1	T11立ち下がりエッジ
0	1	0	タイマ/イベント・カウンタ・チャンネル0のオーバフロー
0	1	1	$f_x/2^2$ (1.5 MHz)
1	0	0	$f_x/2^{12}$ (1.46 kHz)
1	0	1	$f_x/2^{10}$ (5.86 kHz)
1	1	0	$f_x/2^8$ (23.4 kHz)
1	1	1	$f_x/2^6$ (93.8 kHz)

$f_x = 4.19 \text{ MHz}$ 時の場合

TM16	TM15	TM14	カウント・パルス (CP)
0	0	0	T11立ち上がりエッジ
0	0	1	T11立ち下がりエッジ
0	1	0	タイマ/イベント・カウンタ・チャンネル0のオーバフロー
0	1	1	$f_x/2^2$ (1.05 MHz)
1	0	0	$f_x/2^{12}$ (1.02 kHz)
1	0	1	$f_x/2^{10}$ (4.10 kHz)
1	1	0	$f_x/2^8$ (16.4 kHz)
1	1	1	$f_x/2^6$ (65.5 kHz)

図5 - 31 タイマ/イベント・カウンタ・モード・レジスタ(チャンネル1)のフォーマット(2/2)

**タイマ・スタート指示ビット**

TM13	“ 1 ” を書き込むことによって、カウンタ、IRQT1フラグをクリア。 ビット2が“ 1 ” にセットされていれば、カウント動作をスタート。
------	--

**動作モード**

TM12	カウント動作
0	停止(カウント内容保持)
1	カウント動作

**動作モード選択ビット**

TM11	TM10	モード
0	0	8ビット・タイマ/イベント・カウンタ・モード
1	0	16ビット・タイマ/イベント・カウンタ・モード
上記以外		設定禁止

(2) タイマ/イベント・カウンタ出力許可フラグ (TOE0, TOE1)

タイマ/イベント・カウンタ出力許可フラグ (TOE0, TOE1) は、タイマ・アウトF/F (TOUT F/F) の状態でPTO0, PTO1端子へ出力許可/禁止を制御するフラグです。

タイマ・アウトF/Fは、コンパレータからの一致信号で反転するF/Fです。タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1) のビット3を“1”に設定すると、タイマ・アウトF/Fは、“0”にクリアされます。

$\overline{\text{RESET}}$ 信号発生により、TOE0, TOE1とタイマ・アウトF/Fは“0”にクリアされます。

図5 - 32 タイマ/イベント・カウンタ出力許可フラグのフォーマット



## 5.5.2 8ビット・タイマ/イベント・カウンタ・モードの動作

8ビット・タイマ/イベント・カウンタとして使用するモードです。8ビット長でのプログラマブル・インターバル・タイマ動作およびイベント・カウンタ動作を行います。

### (1) レジスタの設定

8ビット・タイマ/イベント・カウンタ・モードは、次の3つのレジスタと1つのフラグを使用します。

タイマ/イベント・カウンタ・モード・レジスタ (TMn)  
タイマ/イベント・カウンタ・カウント・レジスタ (Tn)  
タイマ/イベント・カウンタ・モジュロ・レジスタ (TMODn)  
タイマ/イベント・カウンタ出力許可フラグ (TOEn)

#### (a) タイマ/イベント・カウンタ・モード・レジスタ (TMn)

8ビット・タイマ/イベント・カウンタ・モードを使用する場合はTMnを図5-33のように設定します (TMnのフォーマットについては図5-30, 5-31を参照してください)。

TMnは、8ビット操作命令により操作します。ビット3は、タイマ・スタート指示ビットで、ビット単位の操作もでき、タイマが動作を開始すると自動的に0クリアされます。

TMnは、内部リセット信号発生時に00Hにクリアされます。

**備考** n = 0, 1

図5 - 33 タイマ/イベント・カウンタ・モード・レジスタの設定（8ビット時）（1/2）

(a) タイマ/イベント・カウンタ（チャンネル0）の場合

アドレス	7	6	5	4	3	2	1	0	略号
FA0H		TM06	TM05	TM04	TM03	TM02	TM01	TM00	TM0

カウント・パルス（CP）選択ビット

TM06	TM05	TM04	カウント・パルス（CP）
0	0	0	T10立ち上がりエッジ
0	0	1	T10立ち下がりエッジ
0	1	0	$f_w/2^{14}$ または $f_w/2^7$ （時計用タイマ（INTW）出力）注
0	1	1	$f_x/2^2$
1	0	0	$f_x/2^{10}$
1	0	1	$f_x/2^8$
1	1	0	$f_x/2^6$
1	1	1	$f_x/2^4$

注  $f_w = f_{XT}$ または $f_x/2^7$

タイマ・スタート指示ビット

TM03	“1”を書き込むことによって、カウンタ、IRQTOフラグをクリア。 ビット2が“1”にセットされていれば、カウント動作をスタート。
------	--

動作モード

TM02	カウント動作
0	停止（カウント内容保持）
1	カウント動作

動作モード選択ビット

TM01	TM00	モード
0	0	8ビット・タイマ/イベント・カウンタ・モード

図5 - 33 タイマ/イベント・カウンタ・モード・レジスタの設定（8ビット時）（2/2）

(b) タイマ/イベント・カウンタ（チャンネル1）の場合

アドレス	7	6	5	4	3	2	1	0	略号
FA8H		TM16	TM15	TM14	TM13	TM12	TM11	TM10	TM1

カウント・パルス（CP）選択ビット

TM16	TM15	TM14	カウント・パルス（CP）
0	0	0	T11立ち上がりエッジ
0	0	1	T11立ち下がりエッジ
0	1	0	タイマ/イベント・カウンタ・チャンネル0のオーバーフロー
0	1	1	$f_x/2^2$
1	0	0	$f_x/2^{12}$
1	0	1	$f_x/2^{10}$
1	1	0	$f_x/2^8$
1	1	1	$f_x/2^6$

タイマ・スタート指示ビット

TM13	“ 1 ” を書き込むことによって、カウンタ、IRQT1フラグをクリア。 ビット2が“ 1 ” にセットされていれば、カウント動作をスタート。
------	--

動作モード

TM12	カウント動作
0	停止（カウント内容保持）
1	カウント動作

動作モード選択ビット

TM11	TM10	モード
0	0	8ビット・タイマ/イベント・カウンタ・モード

図5 - 34 タイマ/イベント・カウンタ出力許可フラグの設定



(2) タイマ/イベント・カウンタの時間設定

[ タイマ設定時間 ] (周期) は, [ モジユロ・レジスタの内容 + 1 ] をモード・レジスタの設定により選択された [ カウント・パルス (CP) 周波数 ] で割った値となります。

$$T(\text{sec}) = \frac{n+1}{f_{CP}} = (n+1) \cdot (\text{分解能})$$

T (sec) : タイマ設定時間 (秒)

f<sub>CP</sub> (Hz) : CP周波数 (Hz)

n : モジユロ・レジスタの内容 (n = 0)

タイマは一度設定すれば, その間隔で毎回割り込み要求フラグ (IRQT0, IRQT1) をセットします。

表5 - 7にタイマ/イベント・カウンタの各カウント・パルスでの分解能と最長設定時間 (モジユロ・レジスタにFFHをセットしたときの時間) を示します。

表5 - 7 分解能と最長設定時間 (8ビット・タイマ時)

(a) タイマ/イベント・カウンタ (チャンネル0) の場合

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM06	TM05	TM04	分解能	最長設定時間	分解能	最長設定時間
0	1	0	0.35 s (2.73 ms) <sup>注</sup>	89.5 s (0.7 s) <sup>注</sup>	0.5 s (3.91 ms) <sup>注</sup>	128 s (1.00 s) <sup>注</sup>
0	1	1	677 ns	171 μs	954 ns	244 μs
1	0	0	171 μs	43.7 ms	244 μs	62.5 ms
1	0	1	42.7 μs	10.9 ms	61.0 μs	15.6 ms
1	1	0	10.7 μs	2.73 ms	15.3 μs	3.91 ms
1	1	1	2.67 μs	683 μs	3.81 μs	977 μs

注 f<sub>w</sub> = 32.768 kHz, WM1 = 0 のとき。( ) 内はWM1 = 1 のとき。

(b) タイマ/イベント・カウンタ (チャンネル1) の場合

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM16	TM15	TM14	分解能	最長設定時間	分解能	最長設定時間
0	1	1	677 ns	171 μs	954 ns	244 μs
1	0	0	683 μs	175 ms	977 μs	250 ms
1	0	1	171 μs	43.7 ms	244 μs	62.5 ms
1	1	0	42.7 μs	10.9 ms	61.0 μs	15.6 ms
1	1	1	10.7 μs	2.73 ms	15.3 μs	3.91 ms

### (3) 8ビット・タイマ/イベント・カウンタ動作

8ビット・タイマ/イベント・カウンタ動作は、次のようになります。

図5 - 35にタイマ/イベント・カウンタ動作時の構成を示します。

タイマ/イベント・カウンタ・モード・レジスタ (TMn) の設定によりカウント・パルス (CP) が選択され、タイマ/イベント・カウンタ・カウント・レジスタ (Tn) に入力されます。

Tnとタイマ/イベント・カウンタ・モジュロ・レジスタ (TMODn) の内容を比較し、一致すると一致信号を発生し、割り込み要求フラグ (IRQTn) をセットします。同時にタイマ・アウト・フリップフロップ (TOUT F/F) は反転します。

8ビット・タイマ/イベント・カウンタ動作のタイミングは、図5 - 36のようになります。

8ビット・タイマ/イベント・カウンタ動作は、通常、次の手順により開始します。

TMODnにカウント数を設定します。

TMnに動作モード、カウント・パルス、スタート指示を設定します。

**注意** タイマ/イベント・カウンタ・モジュロ・レジスタ (TMODn) には00H以外の値を設定してください。

タイマ/イベント・カウンタ出力端子 (PTOn) を使用する場合は、兼用端子のP2nを次のように設定してください。

P2nの出力ラッチをクリアする。

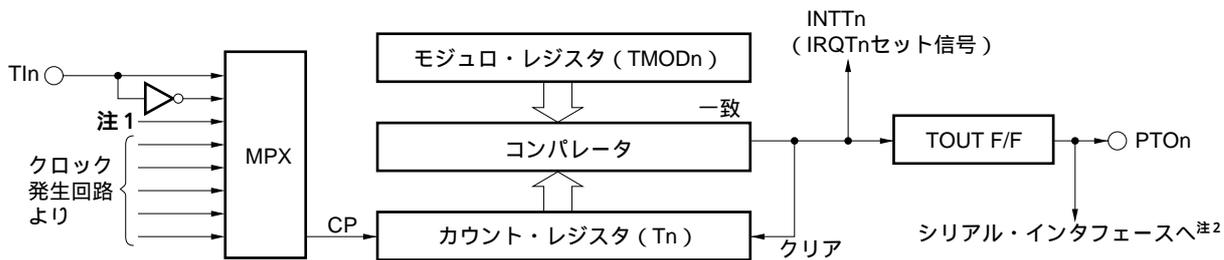
ポート2を出力モードに設定する。

ポート2の内蔵プルアップ抵抗を接続しない状態にする。

タイマ/イベント・カウンタ出力許可フラグ (TOEn) を1にする。

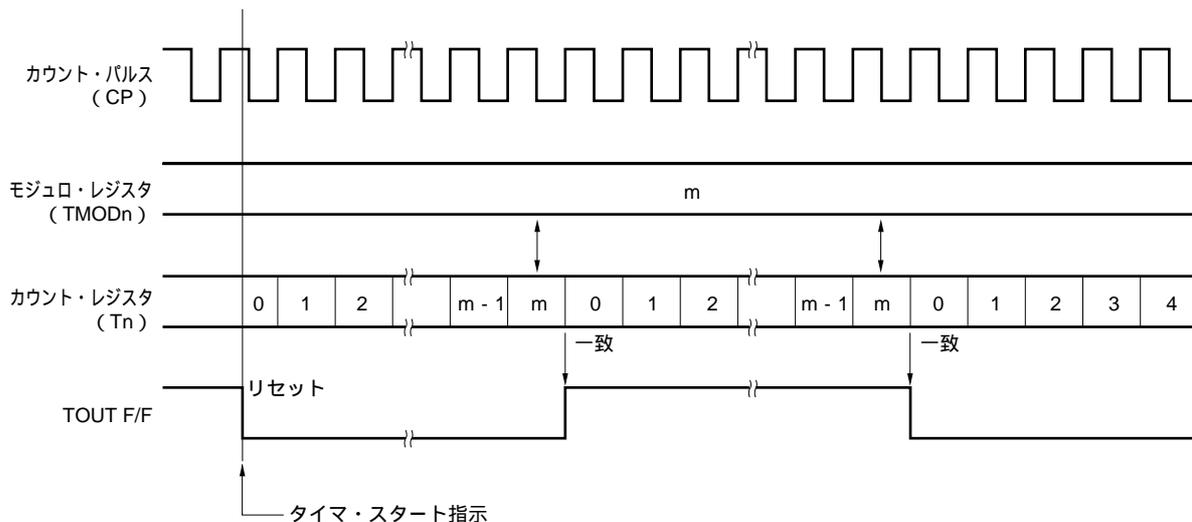
**備考** n = 0, 1

図5 - 35 タイマ/イベント・カウンタ動作時の構成



- 注1 . 時計用タイマ (INTW) 出力 : チャンネル0 のとき  
 タイマ/イベント・カウンタ (チャンネル0) 出力 : チャンネル1 のとき
- 2 . シリアル・インタフェースへの信号は, タイマ/イベント・カウンタのチャンネル0 のみ出力できます。

図5 - 36 カウント動作時のタイミング



備考 m : モジュロ・レジスタの設定値  
 n = 0, 1

## (4) 8ビット・タイマ/イベント・カウンタ・モードの応用

(a) 50 ms間隔で割り込みを発生するインターバル・タイマとして応用します。

タイマ/イベント・カウンタ・モード・レジスタ (TM0) の上位4ビットは0100Bとし、最長設定時間62.5 ms (fx = 4.19 MHz時) を選択します。

TM0の下位4ビットは1100Bとします。

タイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0) の設定値は次のとおりです。

$$\frac{50 \text{ ms}}{244 \mu\text{s}} = 205, 205 - 1 = \text{CCH}$$

**プログラム例**

```
SEL    MB15           ;またはCLR1 MBE
MOV    XA, #0CCH
MOV    TMOD0, XA     ;モジュロを設定
MOV    XA, #01001100B
MOV    TM0, XA       ;モードを設定, タイマ・スタート
EI
EI     IET0          ;タイマの割り込みを許可
```

**備考** この応用の場合, TI0端子は入力端子として使用できます。

(b) TIn端子から入力されるパルス数が100になったら割り込みを発生します (パルスはハイ・アクティブ)。

タイマ/イベント・カウンタ・モード・レジスタ (TMn) の上位4ビットは0000Bで立ち上がりエッジを選択します。

TMnの下位4ビットは1100Bとします。

タイマ/イベント・カウンタ・モジュロ・レジスタ (TMODn) は100 - 1 = 63Hです。

**プログラム例**

```
SEL    MB15           ;またはCLR1 MBE
MOV    XA, #100 - 1
MOV    TMODn, XA     ;モジュロを設定
MOV    XA, #00001100B
MOV    TMn, XA       ;モードを設定, カウント・スタート
EI
EI     IETn          ;INTTnを許可
```

**備考** n = 0, 1

### 5.5.3 16ビット・タイマ/イベント・カウンタ・モードの動作

16ビット・タイマ/イベント・カウンタとして使用するモードです。16ビット長でのプログラブル・インターバル・タイマ動作およびイベント・カウンタ動作を行います。

16ビット・タイマ/イベント・カウンタ・モードを使用する場合は、タイマ・イベント・カウンタのチャンネル0とチャンネル1を組み合わせて使用します。

#### (1) レジスタの設定

16ビット・タイマ/イベント・カウンタ・モードは、次の6つのレジスタと1つのフラグを使用します。

タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1)

タイマ/イベント・カウンタ・カウント・レジスタ (T0, T1)

タイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0, TMOD1)

タイマ/イベント・カウンタ出力許可フラグ (TOE0)

#### (a) タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1)

16ビット・タイマ/イベント・カウンタ・モードを使用する場合はTM0, TM1を図5 - 37のように設定します (TM0, TM1のフォーマットについては図5 - 30 **タイマ/イベント・カウンタ・モード・レジスタ (チャンネル0) のフォーマット**, 図5 - 31 **タイマ/イベント・カウンタ・モード・レジスタ (チャンネル1) のフォーマット**を参照してください)。

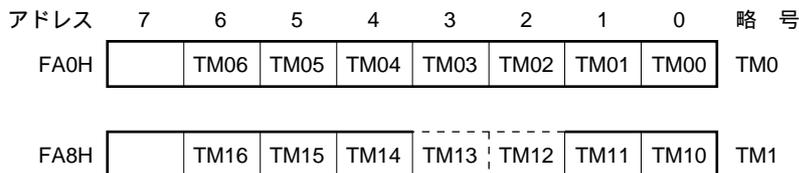
TM0, TM1は、8ビット操作命令により操作します。TM0のビット3 (TM03) は、タイマ・スタート指示ビットで、ビット単位の操作もでき、タイマが動作を開始すると自動的に0クリアされます。

TM0, TM1は、内部リセット信号発生時に00Hにクリアされます。

実線で示されるフラグは、16ビット・タイマ/イベント・カウンタ・モードにおける使用ビットを表します。

点線で示されるフラグは、16ビット・タイマ/イベント・カウンタ・モードでは使用しないでください (0を設定してください)。

図5 - 37 タイマ/イベント・カウンタ・モード・レジスタの設定



**カウント・パルス (CP) 選択ビット**

TMn6	TMn5	TMn4	TM0	TM1
0	0	0	T10立ち上がりエッジ	T11立ち上がりエッジ
0	0	1	T10立ち下がりエッジ	T11立ち下がりエッジ
0	1	0	$f_w/2^{14}$ または $f_w/2^7$ (INTW出力) 注	カウント・レジスタ (T0) のオーバフロー
0	1	1	$f_x/2^2$	$f_x/2^2$
1	0	0	$f_x/2^{10}$	$f_x/2^{12}$
1	0	1	$f_x/2^8$	$f_x/2^{10}$
1	1	0	$f_x/2^6$	$f_x/2^8$
1	1	1	$f_x/2^4$	$f_x/2^6$

**タイマ・スタート指示ビット**

TM03	“ 1 ” を書き込むことによって、カウンタ、IRQT0フラグをクリア。 ビット2が “ 1 ” にセットされていれば、カウント動作をスタート。
------	---

**動作モード**

TM02	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

**動作モード選択ビット**

TM11	TM10	TM01	TM00	モード
1	0	1	0	16ビット・タイマ/イベント・カウンタ・モード

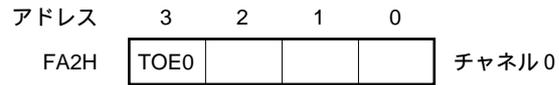
注  $f_w = f_{XT}$ または $f_x/2^7$

**注意** 16ビット・タイマ/イベント・カウンタ・モード時は、TM1 = 00100010Bに設定してください。

(b) タイマ/イベント・カウンタ出力許可フラグ (TOE0)

16ビット・タイマ/イベント・カウンタ出力を行う場合は、TOE0を図5 - 38のように設定します。

図5 - 38 タイマ/イベント・カウンタ出力許可フラグの設定



タイマ/イベント・カウンタ出力許可フラグ (W)

TOE0	タイマ出力
0	禁止 (ロウ・レベルを出力する)
1	許可

(2) 16ビット・タイマ/イベント・カウンタの時間設定

[ タイマ設定時間 ] (周期) は, [ モジユロ・レジスタの内容 + 1 ] をモード・レジスタの設定により選択された [ カウント・パルス (CP) 周波数 ] で割った値となります。

$$T(\text{sec}) = \frac{n+1}{f_{\text{CP}}} = (n+1) \cdot (\text{分解能})$$

T (sec) : タイマ設定時間 (秒)

f<sub>CP</sub> (Hz) : CP周波数 (Hz)

n : モジユロ・レジスタの内容 (n = 0)

タイマは一度設定すれば, その間隔で毎回割り込み要求フラグ (IRQT0) をセットします。

表5 - 8に16ビット・タイマ/イベント・カウンタの各カウント・パルスでの分解能と最長設定時間 (モジユロ・レジスタ0, 1に各々FFHをセットしたときの時間) を示します。

表5 - 8 分解能と最長設定時間 (16ビット・タイマ時)

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM06	TM05	TM04	分解能	最長設定時間	分解能	最長設定時間
0	1	0	0.35 s (2.73 ms) <sup>注</sup>	22906 s (179 s) <sup>注</sup>	0.5 s (3.91 ms) <sup>注</sup>	32768 s (256 s) <sup>注</sup>
0	1	1	677 ns	43.7 ms	954 ns	62.5 ms
1	0	0	171 μs	11.2 s	244 μs	16.0 s
1	0	1	42.7 μs	2.80 s	61.0 μs	4.0 s
1	1	0	10.7 μs	699 ms	15.3 μs	1.0 s
1	1	1	2.67 μs	175 ms	3.81 μs	250 ms

注 f<sub>w</sub> = 32.768 kHz, WM1 = 0 のとき。 ( ) 内はWM1 = 1 のとき。

注意1 . 16ビット・タイマ/イベント・カウンタ・モード時は, TM1 = 00100010Bに設定してください。

2 . 分解能は, タイマ・チャンネル0のCPで決定されます。

### (3) 16ビット・タイマ/イベント・カウンタ動作

タイマ/イベント・カウンタ動作は、次のようになります。

図5 - 39にタイマ/イベント・カウンタ動作時の構成を示します。

タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1) の設定によりカウント・パルス (CP) が選択され、タイマ/イベント・カウンタ・カウント・レジスタ (T0) に入力されます。タイマ/イベント・カウンタ・カウント・レジスタ (T1) には、T0のオーバフローが入力されません。

T0とタイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0) の内容を比較し、一致すると一致信号を発生します。

T1とタイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD1) の内容を比較し、一致すると一致信号を発生します。

と の一致信号が重複すると、割り込み要求フラグ (IRQT0) をセットします。同時にタイマ・アウト・フリップフロップ (TOUF/F) は反転します。

16ビット・タイマ/イベント・カウンタ動作のタイミングは、図5 - 40のようになります。

16ビット・タイマ/イベント・カウンタ動作は、通常、次の手順により開始します。

TMOD1に16ビット幅で示されるカウント数のうち、上位8ビット分を設定します。

TMOD2に16ビット幅で示されるカウント数のうち、下位8ビット分を設定します。

TM1 に00100010Bを設定します。

TM0 に動作モード、カウント・パルス、スタート指示を設定します。

**注意1．タイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0) には00H以外の値を設定してください。**

**2．タイマ/イベント・カウンタ割り込み許可フラグ (IET1) は0 (禁止) にしてください。**

タイマ/イベント・カウンタ出力端子 (PTO0) を使用する場合は、兼用端子のP20を次のように設定してください。

P20の出力ラッチをクリアする。

ポート2を出力モードに設定する。

ポート2の内蔵プルアップ抵抗を接続しない状態にする。

タイマ/イベント・カウンタ出力許可フラグ (TOE0) を1にする。

図5 - 39 タイマ/イベント・カウンタ動作時の構成

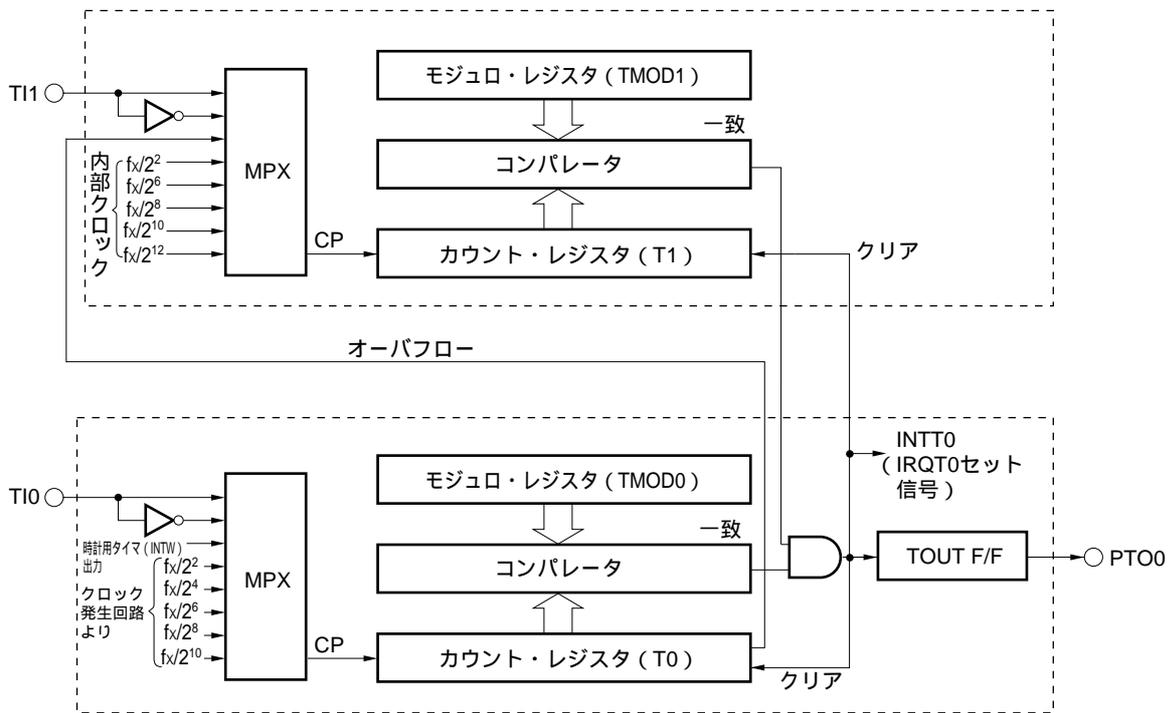
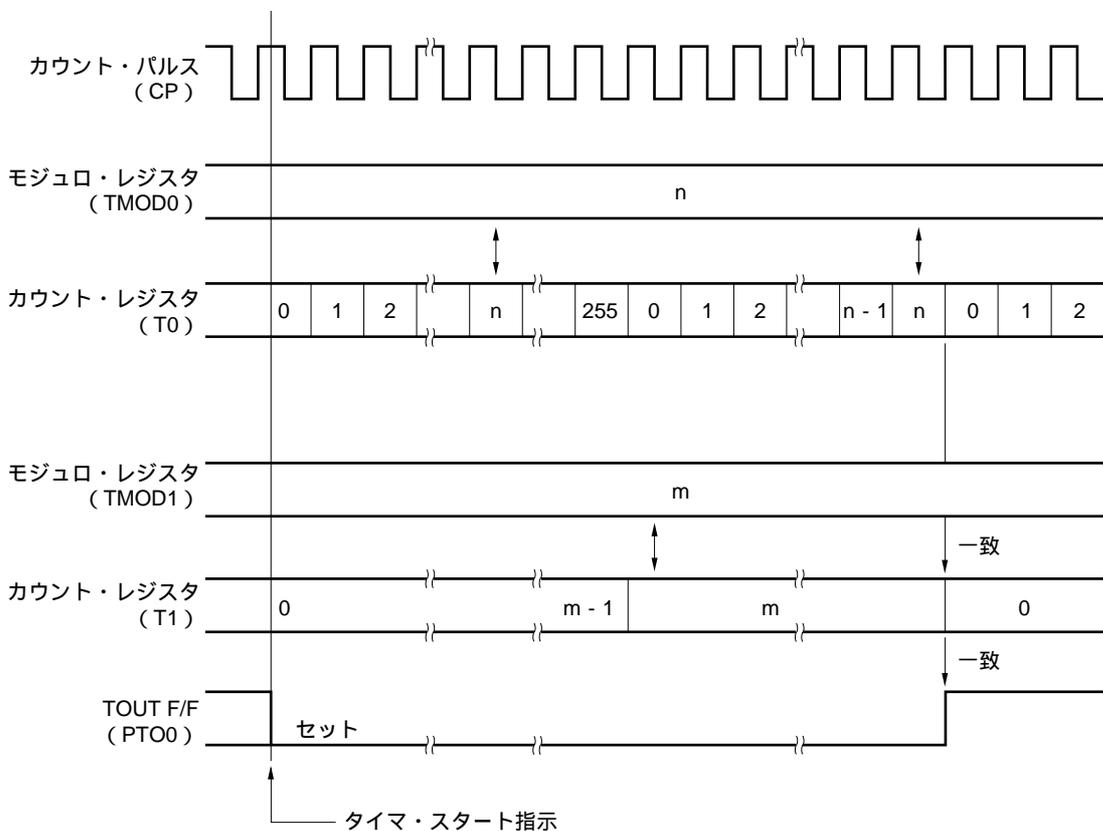


図5 - 40 カウント動作時のタイミング



**備考** m : モジュール・レジスタ (TMOD1) の設定値  
 n : モジュール・レジスタ (TMOD0) の設定値

## (4) 16ビット・タイマ/イベント・カウンタ・モードの応用

(a) 5 sec間隔で割り込みを発生するインターバル・タイマとして応用します。

タイマ/イベント・カウンタ・モード・レジスタ (TM1) の上位4ビットは0010Bとし、タイマ/イベント・カウンタ・カウント・レジスタ (T0) のオーバーフローを選択します。

TM0の上位4ビットは0100Bとし、最長設定時間16.0 secを選択します。

TM1の下位4ビットは0010Bとし、16ビット・タイマ/イベント・カウンタ・モードを選択します。

TM0の下位4ビットは1110Bとし、16ビット・タイマ/イベント・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0, TMOD1) の設定値は、次のとおりです。

$$\frac{5 \text{ sec}}{244 \mu\text{s}} = 20491.8, 20492 - 1 = 500\text{BH}$$

## &lt;プログラム例&gt;

```

SEL    MB15           ;またはCLR1 MBE
MOV    XA, #050H
MOV    TMOD1, XA     ;モジュロを設定 (上位8ビット分)
MOV    XA, #00BH
MOV    TMOD0, XA     ;モジュロを設定 (下位8ビット分)
MOV    XA, #00100010B
MOV    TM1, XA       ;モードを設定
MOV    XA, #01001110B
MOV    TM0, XA       ;モードを設定, タイマ・スタート
DI     IET0          ;タイマ (チャンネル1) の割り込みを禁止
EI     IET0          ;割り込みを許可
EI     IET0          ;タイマ (チャンネル0) 割り込みを許可

```

**備考** この応用の場合、TI0, TI1端子は入力端子として使用できます。

(b) TI0端子から入力されるパルス数が1000になると割り込みを発生します(パルスはハイ・アクティブ)。

タイマ/イベント・カウンタ・モード・レジスタ(TM1)の上位4ビットは0010Bとし、タイマ/イベント・カウンタ・カウント・レジスタ(T0)のオーバーフローを選択します。

TM0の上位4ビットは0000Bとし、TI0入力立ち上がりエッジを選択します。

TM1の下部4ビットは0010Bとし、16ビット・タイマ/イベント・カウンタ・モードを選択します。

TM0の下部4ビットは1110Bとし、16ビット・タイマ/イベント・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ/イベント・カウンタ・モジュロ・レジスタ(TM0D0, TM0D1)の設定値は、 $1000 - 1 = 999 = 03E7H$ となり、TM0D0には03Hを設定し、TM0D1にはE7Hを設定します。

#### <プログラム例>

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #003H
MOV    TM0D1, XA          ;モジュロを設定(上位8ビット分)
MOV    XA, #0E7H
MOV    TM0D0, XA          ;モジュロを設定(下部8ビット分)
MOV    XA, #00100010B
MOV    TM1, XA            ;モードを設定
MOV    XA, #00001110B
MOV    TM0, XA            ;モードを設定, タイマ・スタート
DI     IET1                ;タイマ(チャンネル1)の割り込みを禁止
EI
EI     IET0                ;タイマ(チャンネル0)の割り込みを許可
```

**備考** この応用の場合、TI1端子は入力端子として使用できます。

(c) 時計用タイマ (INTW) 出力を使って、9 時間間隔で割り込みを発生するインターバル・タイマとして応用します。

タイマ/イベント・カウンタ・モード・レジスタ (TM1) の上位 4 ビットは 0010B とし、タイマ/イベント・カウンタ・カウント・レジスタ (T0) のオーバフローを選択します。

TM0 の上位 4 ビットは 0010B とし、最長設定時間 32768 sec を選択します (fw = 32.768 kHz, WM1 = 0 に設定する)。

TM1 の下位 4 ビットは 0010B とし、16 ビット・タイマ/イベント・カウンタ・モードを選択します。

TM0 の下位 4 ビットは 1110B とし、16 ビット・タイマ/イベント・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ/イベント・カウンタ・モジュロ・レジスタ (TMOD0, TMOD1) の設定値は次のとおりです。

$$\frac{9 \text{ 時間 (32400 s)}}{0.5 \text{ s}} = 64800, 64800 - 1 = \text{FD1FH}$$

#### < プログラム例 >

```

SEL   MB15           ; または CLR1 MBE
MOV   XA, #0FDH
MOV   TMOD1, XA      ; モジュロを設定 (上位 8 ビット分)
MOV   XA, #01FH
MOV   TMOD0, XA      ; モジュロを設定 (下位 8 ビット分)
MOV   XA, #00100010B
MOV   TM1, XA        ; モードを設定
MOV   XA, #00101110B
MOV   TM0, XA        ; モードを設定, タイマ・スタート
DI    IET1           ; タイマ (チャンネル 1) の割り込みを禁止
EI
EI    IET0           ; タイマ (チャンネル 0) の割り込みを許可

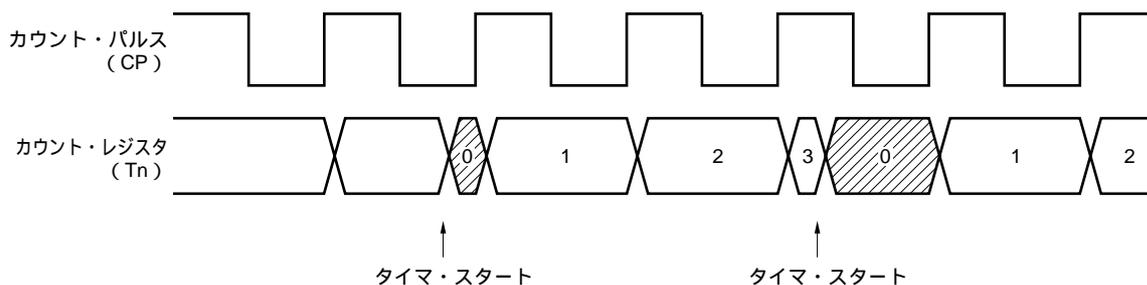
```

**備考** この応用の場合、TI0、TI1 端子は入力端子として使用できます。

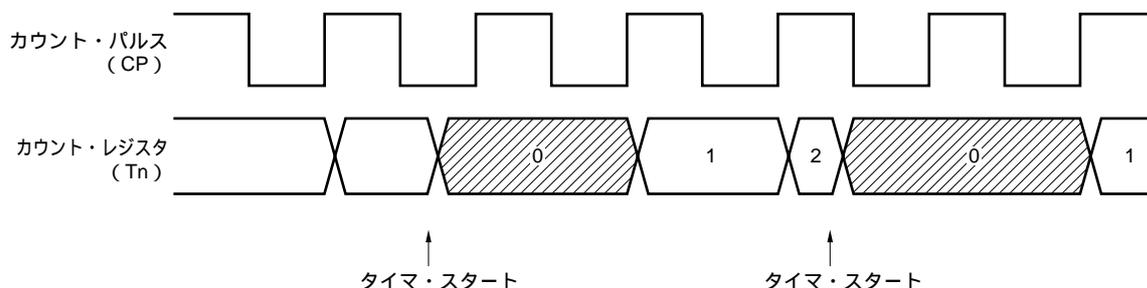
### 5.5.4 タイマ/イベント・カウンタ応用上の注意

#### (1) タイマ・スタート時の誤差

タイマ・スタート (TMnのビット3を“1”にセット) 後、一致信号が発生するまでの時間は (モジュロ・レジスタの内容 + 1) × 分解能の式で求めた値に対し、最大でカウント・パルス (CP) 1クロック分の誤差が生じます。これは次の図に示すように、CPに対してカウント・レジスタTnのクリアが非同期で行われるためです。



またCPの周波数が1マシン・サイクル以上の場合、タイマ・スタート (TMnのビット3を“1”にセット) 後、一致信号が発生するまでの時間は、(モジュロ・レジスタの内容 + 1) × 分解能の式で求めた値に対し、最大でCP2クロック分の誤差が生じます。これは、次の図に示すようにCPに対してTnのクリアがCPUクロックを基に非同期で行われるためです。



備考 n = 0, 1

## (2) タイマ・スタート時の注意

通常タイマ・スタート (TMnのビット3を“1”にセット) によってカウント・レジスタTnと割り込み要求フラグIRQTnはクリアされますが、タイマが動作モードにあり、IRQTnのセットと、タイマ・スタートが同一タイミングで発生した場合には、IRQTnをクリアできない場合が生じます。これは、IRQTnをベクタ割り込みとして使用しているときは問題ありませんが、IRQTnをテストする応用の場合には、タイマ・スタートを行ったにもかかわらず、IRQTnがセットされていることになり不都合を生じます。したがって、IRQTnが立つ可能性のあるタイミングでタイマ・スタートする場合にはいったん、タイマを停止 (TMnのビット2を“0”にする) させてから再スタートするか、タイマ・スタートを2回行うようにしてください。

## 例 IRQTnが立つ可能性のあるタイミングでのタイマ・スタート

```

SEL      MB15
MOV      XA, #0
MOV      TMn, XA          ; タイマを停止
MOV      XA, #4CH
MOV      TMn, XA          ; 再スタート
または
SEL      MB15
SET1     TMn.3
SET1     TMn.3          ; 再スタート

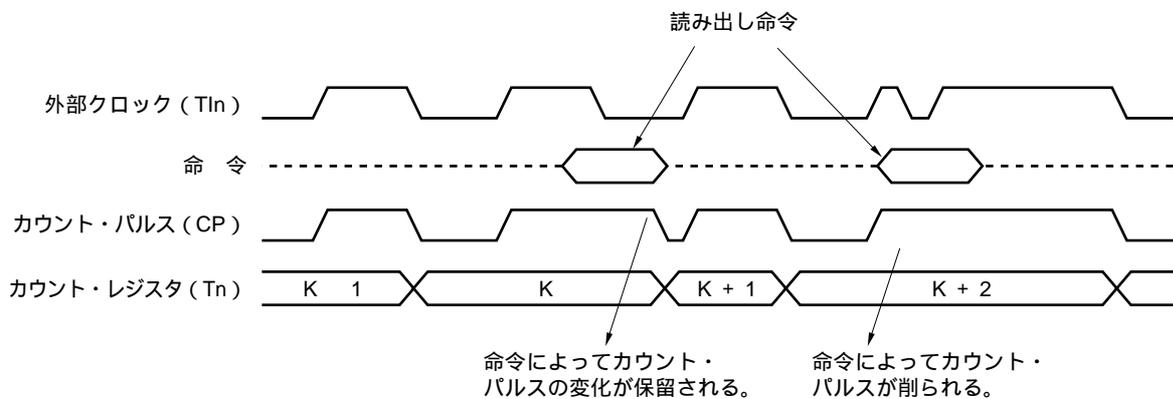
```

備考 n = 0, 1

(3) カウント・レジスタ読み出し時の誤差

カウント・レジスタ (Tn) の内容は、8 ビット・データ・メモリ操作命令によっていつでも読み出し可能ですが、この命令の動作中、カウント・パルス (CP) の変化が保留され、Tnが変化しないようになっています。このため、CPの信号源をTIn入力にした場合、命令実行時間だけCPが削られます (内部クロックをCPとした場合は、命令と同期するのでこの現象は起こりません)。

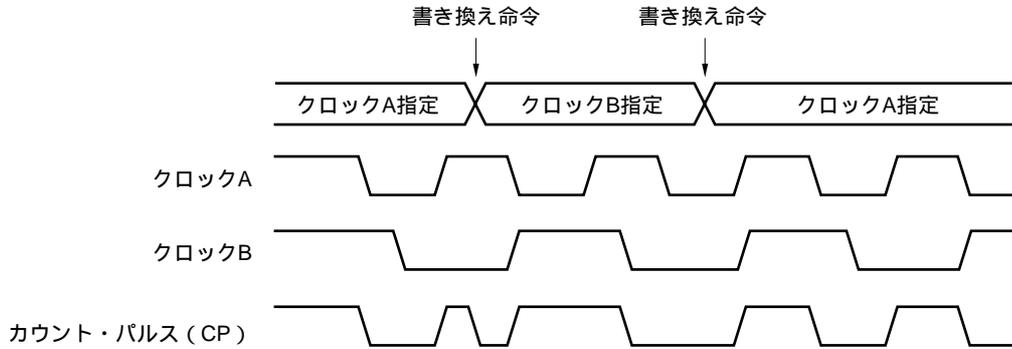
したがって、TIn入力をCPとして入力し、Tnの内容を読み取る場合は、CPが削られてもミス・カウントしないパルス幅の信号を入力しなければなりません。すなわち、読み出し命令によってカウントが保留される期間は、1 マシン・サイクルとなっていますから、TInに inputsするパルスはそれ以上の幅が必要となります。



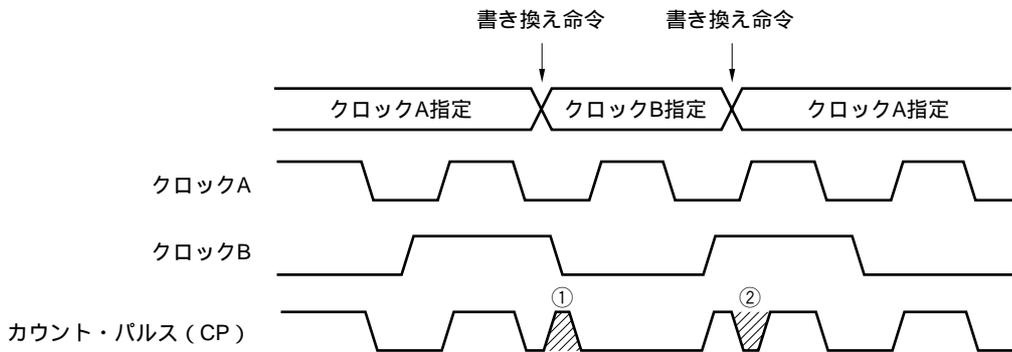
備考 n = 0, 1

(4) カウント・パルス変更時の注意

タイマ/イベント・カウンタ・モード・レジスタ (TMn) を書き換えてカウント・パルス (CP) を変更した場合、その指定は命令実行直後から有効になります。



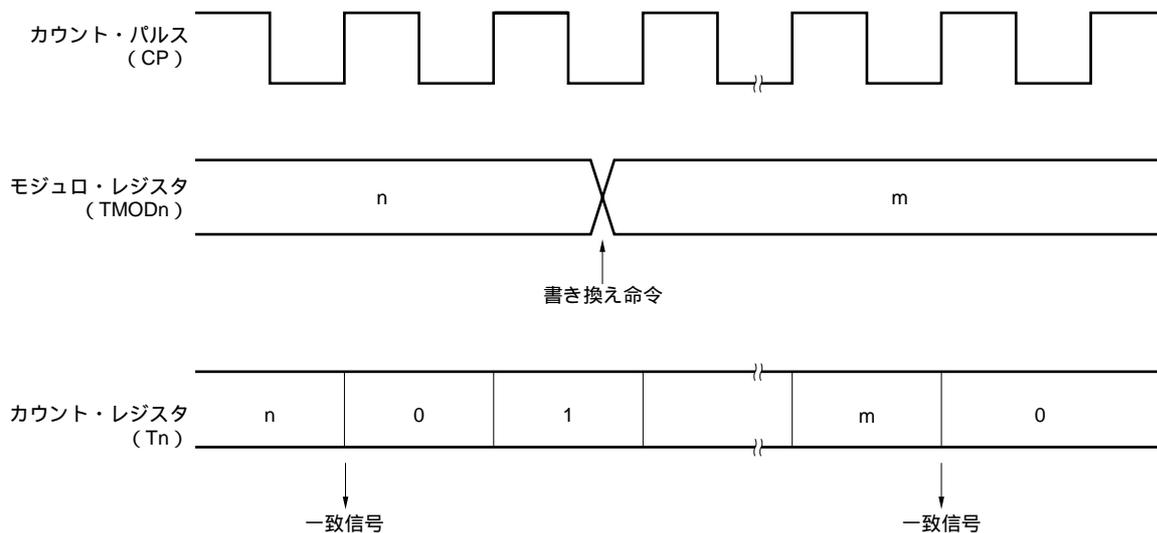
CP変更時のクロックの組み合わせにより、次図のようにヒゲ状 (または ) のCPが発生すること也被考えられます。この場合、ミス・カウントあるいはカウント・レジスタ (Tn) の内容が破壊される可能性がありますので、CPを変更する場合は必ず、TMnのビット3を“1”にして、タイマの再スタートも同時に行ってください。



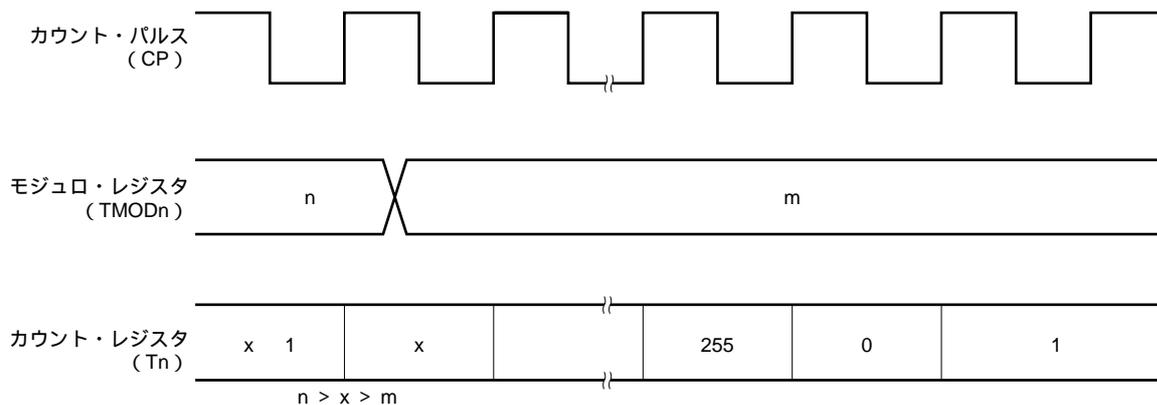
備考 n = 0, 1

(5) モジュロ・レジスタ変更後の動作

モジュロ・レジスタ (TMODn) の変更は、8ビット・データ・メモリ操作命令を実行した時点で行われます。



TMODnの変更後の値が、カウンタ・レジスタ (Tn) の値よりも小さい場合には、Tnはカウントを継続しオーバーフローして0から再カウントします。したがって、TMODnの変更後の値 ( $m$ ) が変更前の ( $n$ ) より小さい場合には、TMODnを変更後、タイマを再スタートさせる必要があります。



## 5.6 シリアル・インタフェース

### 5.6.1 シリアル・インタフェースの機能

μPD750068は、クロック同期式8ビット・シリアル・インタフェースを内蔵しており、次の3種類のモードがあります。

以下に、各モードの機能概要を示します。

#### (1) 動作停止モード

シリアル転送を行わないときに使用するモードです。消費電力を低減することができます。

#### (2) 3線式シリアルI/Oモード

シリアル・クロック ( $\overline{\text{SCK}}$ )、シリアル出力 (SO)、シリアル入力 (SI) の3本のラインによって8ビット長の転送を行います。

3線式シリアルI/Oモードは同時送受信動作が可能なので、データ転送の処理時間は速くなります。

シリアル転送する8ビット・データの先頭ビットをMSBか、またはLSBかに切り替えることができるので、いずれの先頭ビットのデバイスとも接続が可能です。

3線式シリアルI/Oモードでは、75XLシリーズ、75Xシリーズ、78Kシリーズ、各種周辺I/Oデバイスとの接続が可能です。

#### (3) 2線式シリアルI/Oモード

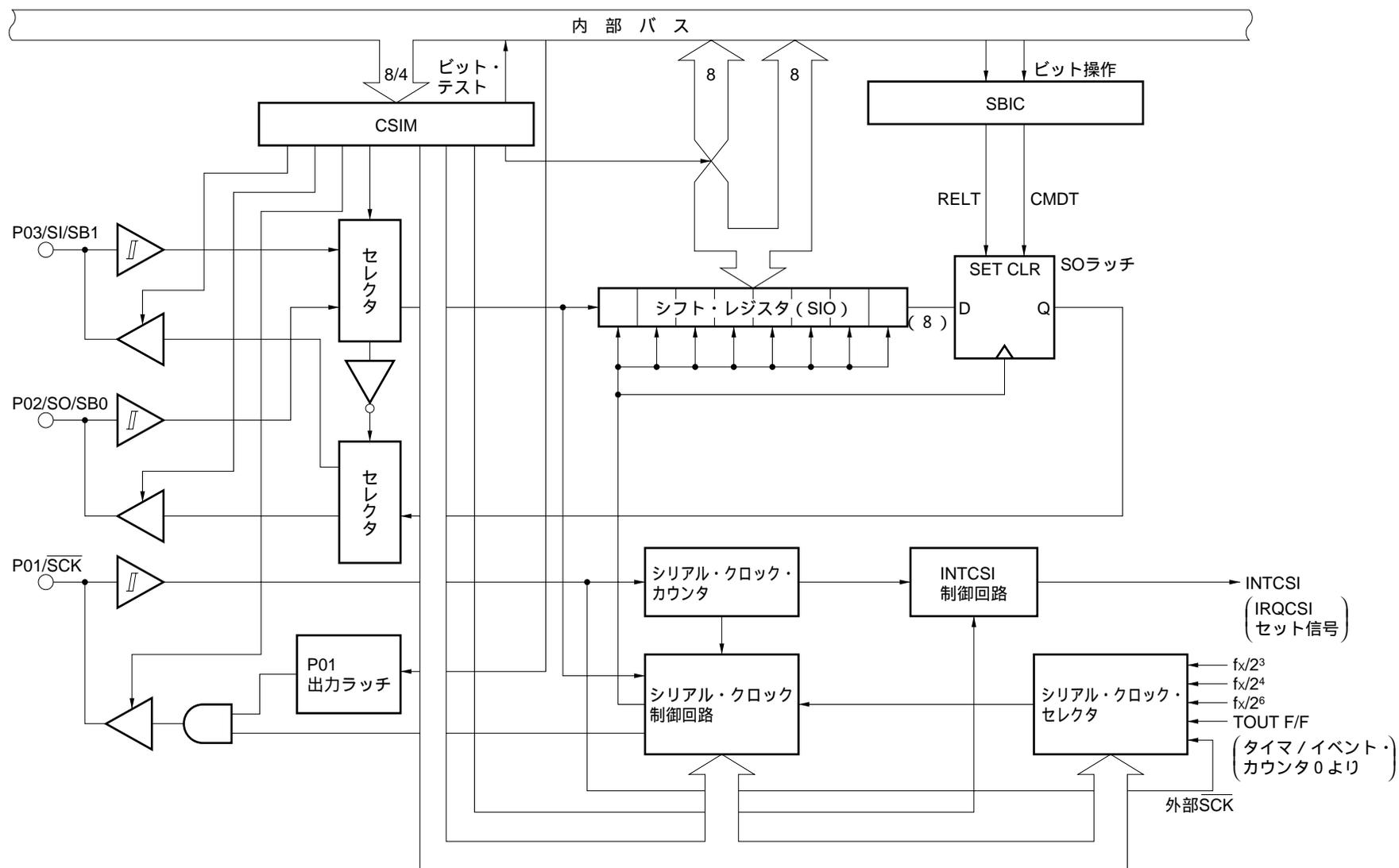
シリアル・クロック ( $\overline{\text{SCK}}$ )、シリアル・データ・バス (SB0、またはSB1) の2本のラインによって8ビット長のデータの転送を行います。2本のラインへの出力レベルをソフトウェアで操作することにより、複数のデバイスと通信できます。

また、 $\overline{\text{SCK}}$ とSB0 (または、SB1) はソフトウェアによる出力レベルの操作が可能のため、任意の転送フォーマットに対応することができます。したがって、従来、複数のデバイスを接続する場合に必要なハンドシェイクのためのラインを削減することができ、入出力ポートの有効活用ができます。

### 5.6.2 シリアル・インタフェースの構成

図5-41にシリアル・インタフェースのブロック図を示します。

図5 - 41 シリアル・インタフェースのブロック図



**(1) シリアル動作モード・レジスタ (CSIM)**

シリアル・インタフェースの動作モード、シリアル・クロック、ウエイク・アップ機能などを指定する8ビット・レジスタです(詳細は、5.6.3(1)シリアル動作モード・レジスタ(CSIM)を参照)。

**(2) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)**

シリアル・バスの状態を制御するビットと、シリアル・バスからの入力データの各種状態を示すフラグから構成される8ビット・レジスタです(詳細は、5.6.3(2)シリアル・バス・インタフェース・コントロール・レジスタ(SBIC)を参照)。

**(3) シフト・レジスタ (SIO)**

8ビットのシリアル・データをパラレル・データに、8ビットのパラレル・データをシリアル・データに変換するレジスタです。シリアル・クロックに同期して送受信動作(シフト動作)を行います。SIOに対する書き込みにより、実際の送受信動作が制御されます(詳細は、5.6.3(3)シフト・レジスタ(SIO)を参照)。

**(4) SOラッチ**

SO/SB0, SI/SB1端子レベルを保持するラッチです。ソフトウェアにより直接制御することもできます(詳細は、5.6.3(2)シリアル・バス・インタフェース・コントロール・レジスタ(SBIC)を参照)。

**(5) シリアル・クロック・セレクタ**

使用するシリアル・クロックを選択します。

**(6) シリアル・クロック・カウンタ**

送受信動作時に出力されるシリアル・クロック、および入力されるシリアル・クロックをカウントし、8ビット・データの送受信が行われたことを調べます。

**(7) INTCSI制御回路**

割り込み要求の発生を制御します。次の場合に割り込み要求(INTCSI)を発生し、割り込み要求フラグ(IRQCSI)をセットします(図6-1 割り込み制御回路ブロック図を参照)。

**3線式および2線式シリアルI/Oモード時**

シリアル・クロックを8発カウントするたびに割り込み要求を発生します。

**(8) シリアル・クロック制御回路**

シフト・レジスタへのシリアル・クロックの供給を制御します。また、内部システム・クロック使用時は、 $\overline{\text{SCK}}$ 端子へ出力するクロックの制御も行います。

**(9) P01出力ラッチ**

シリアル・クロックが8発終了したあとに、ソフトウェアでシリアル・クロックを発生するためのラッチです。

リセット入力により“1”に設定されます。

シリアル・クロックとして内部システム・クロックを選択する場合は、P01出力ラッチを“1”に設定してください。

### 5.6.3 レジスタの機能

#### (1) シリアル動作モード・レジスタ (CSIM)

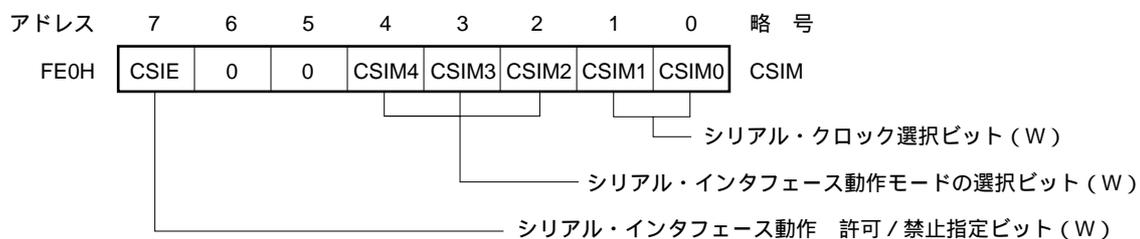
図5 - 42にシリアル動作モード・レジスタ (CSIM) のフォーマットを示します。

CSIMは、シリアル・インタフェースの動作モード、シリアル・クロックなどを指定する8ビット・レジスタです。

操作は、8ビット・メモリ操作命令で行います。ビット7については、1ビット単位の操作もできます。ビット操作の場合はビットの名称を用いて操作します。

$\overline{\text{RESET}}$ 信号発生により、全ビット0にクリアされます。

図5 - 42 シリアル動作モード・レジスタ (CSIM) のフォーマット (1/3)



**注意** CSIMのビット6, 5には必ず“0”を設定してください。

**備考** (W) 書き込みのみ可能

図5 - 42 シリアル動作モード・レジスタ (CSIM) のフォーマット (2/3)

シリアル・インタフェース動作 許可/禁止指定ビット (W)

	シフト・レジスタの動作	シリアル・クロック・カウンタ	IRQCSIフラグ	SO/SB0, SI/SB1端子	
CSIE	0	シフト動作禁止	クリア	保持	ポート0機能専用
	1	シフト動作可能	カウント動作	セット可能	各モードでの機能とポート0兼用

シリアル・インタフェース動作モードの選択ビット (W)

CSIM4	CSIM3	CSIM2	動作モード	シフト・レジスタのビット順	SO/SB0/P02端子機能	SI/SB1/P03端子機能
×	0	0	3線式シリアル I/Oモード	SIO <sub>7-0</sub> XA (MSB先頭で転送)	SO (CMOS出力)	SI (CMOS入力)
		1		SIO <sub>0-7</sub> XA (LSB先頭で転送)		
0	1	1	2線式シリアル I/Oモード	SIO <sub>7-0</sub> XA (MSB先頭で転送)	SB0 (N-chオープン・ ドレイン入出力)	P03 (CMOS入力)
1					P02 (CMOS入力)	SB1 (N-chオープン・ ドレイン入出力)
上記以外			設定禁止			

備考 × : don't care

シリアル・クロック選択ビット (W)

CSIM1	CSIM0	シリアル・クロック		SCK端子
		3線式シリアルI/Oモード	2線式シリアルI/Oモード	モード
0	0	SCK端子への外部からの入力クロック		入力
0	1	タイマ/イベント・カウンタ0出力 (TO0)		出力
1	0	f <sub>x</sub> /2 <sup>4</sup> (375 kHz : 6.0 MHz動作時, 262 kHz : 4.19 MHz動作時)		出力
1	1	f <sub>x</sub> /2 <sup>3</sup> (750 kHz : 6.0 MHz動作時, 524 kHz : 4.19 MHz動作時)		

図5 - 42 シリアル動作モード・レジスタ (CSIM) のフォーマット (3/3)

備考1 . CSIE, CSIM3, CSIM2の設定により各モードを選択できます。

CSIE	CSIM3	CSM2	動作モード
0	x	x	動作停止モード
1	0	x	3線式シリアル/Oモード
1	1	1	2線式シリアル/Oモード

2 . P01/SCK端子はCSIE, CSIM1, CSIM0の設定により次の状態になります。

CSIE	CSIM1	CSIM0	P01/SCK端子の状態
0	0	0	入力ポート (P01)
1	0	0	ハイ・インピーダンス (SCK入力)
0	0	1	ハイ・レベル出力
0	1	0	
0	1	1	
1	0	1	シリアル・クロック出力
1	1	0	(ハイ・レベル出力: シリアル転送終了時)
1	1	1	

3 . シリアル転送中にCSIEをクリアするときは、次の手順で行ってください。

割り込み許可フラグ (IECSI) をクリアして割り込み禁止状態にする。

CSIEをクリアする。

割り込み要求フラグ (IRQCSI) をクリアする。

例1 . シリアル・クロックにfx/2<sup>4</sup>を選択し、各シリアル転送終了時にシリアル割り込みIRQCSIを発生させ、3線式シリアル/OモードでMSB先頭のシリアル転送を行うモードを選択する。

```
SEL    MB15          ;またはCLR1 MBE
MOV    XA, #10000010B
MOV    CSIM, XA      ;CSIM  10000010B
```

2 . CSIMの内容に従うシリアル転送を可能な状態にする。

```
SEL    MB15          ;またはCLR1 MBE
SET1   CSIE
```

(2) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

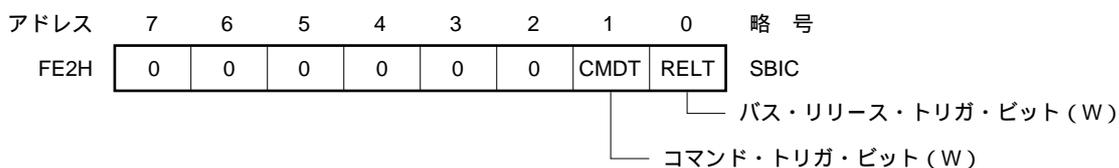
図5 - 43にシリアル・バス・インタフェース・コントロール・レジスタ (SBIC) のフォーマットを示します。

SBICはシリアル・バスを制御する8ビット・レジスタです。

操作は、ビット操作命令で行います。4ビットまたは8ビット・メモリ操作命令で操作することはありません。

RESET信号発生により全ビット0にクリアされます。

図5 - 43 シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) のフォーマット



備考 (W) 書き込みのみ可能

コマンド・トリガ・ビット (W)

CMDT	コマンド信号 (CMD) のトリガ出力制御ビットです。セット (CMDT=1) することによりSOラッチがクリア (0) され、その後、CMDTビットは自動的にクリア (0) されます。
------	---

注意 CMDTビットのセットはシリアル転送中には行わないでください。必ず転送開始前、または転送終了後に行ってください。

バス・リリース・トリガ・ビット (W)

RELT	バス・リリース信号 (REL) のトリガ出力制御ビットです。セット (RELT=1) することによりSOラッチがセット (1) され、その後、RELTビットは自動的にクリア (0) されます。
------	--

注意 RELTビットのセットはシリアル転送中には行わないでください。必ず転送開始前、または転送終了後に行ってください。

### (3) シフト・レジスタ (SIO)

図5 - 44にシフト・レジスタ周辺の構成を示します。SIOはパラレル-シリアル変換を行い、シリアル・クロックに同期してシリアル送受信（シフト動作）を行うための8ビット・レジスタです。

シリアル転送の開始はSIOにデータを書き込むことにより開始します。

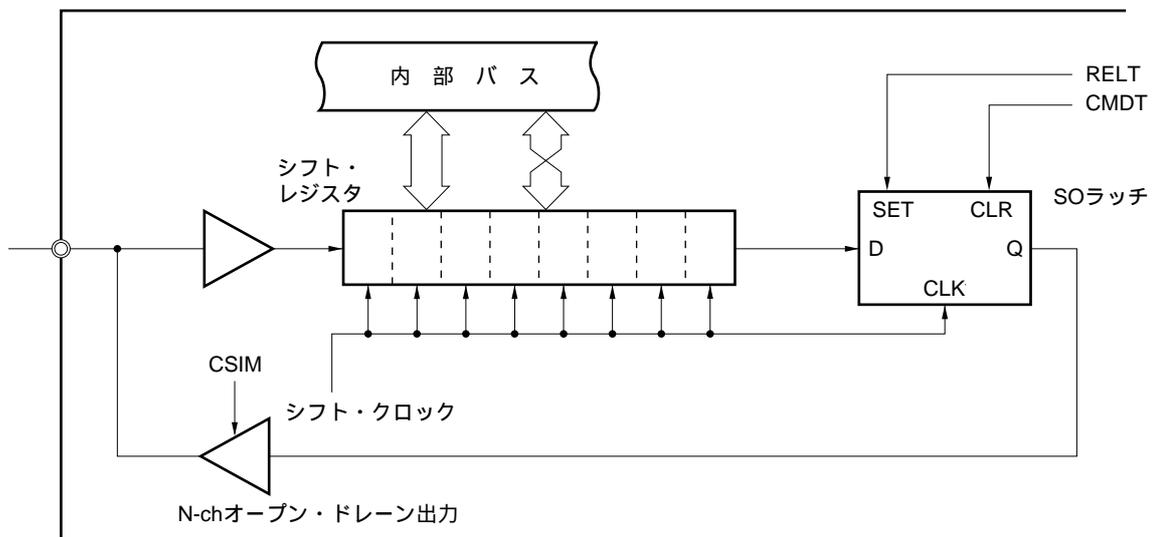
送信時はSIOに書き込まれたデータが、シリアル出力（SO）またはシリアル・データ・バス（SB0/SB1）に出力されます。受信時はデータが、シリアル入力（SI）またはSB0/SB1からSIOに読み込まれます。

8ビット操作命令で読み出し、書き込みが可能です。

動作中にRESET信号を発生した場合、SIOの値は不定となります。スタンバイ・モード時にRESET信号を発生した場合、SIOの値は保持されます。

8ビット送信 / 受信終了後、シフト操作は停止します。

図5 - 44 シフト・レジスタの周辺の構成



SIOの読み出しおよびシリアル転送スタート（書き込み）可能なタイミングは次の場合です。

シリアル・インタフェース動作許可 / 禁止ビット（CSIE）= 1のとき。ただし、シフト・レジスタにデータを書き込んだあと、CSIEを“1”に設定した場合を除く。

8ビット・シリアル転送後で、シリアル・クロックがマスクされた状態のとき。

SCKがハイ・レベルのとき。

SIOへのデータの書き込み / 読み出しは、必ずSCKがハイ・レベルの状態で行ってください。

2線式シリアルI/Oモードのデータ・バス構成は、入力端子と出力端子が兼用です。出力端子はN-chオープン・ドレイン構成です。したがって、これから受信を行おうとするデバイスは、SIOにFFHを設定してください。

### 5.6.4 動作停止モード

動作停止モードは、シリアル転送を行わないときに使用するモードです。消費電力を低減することができます。

このモードでは、シフト・レジスタはシフト動作を行わないので、通常の8ビット・レジスタとして使用することができます。

リセット信号入力時は、動作停止モードになります。P02/SO/SB0端子，P03/SI/SB1端子は，入力ポートに固定されます。また，P01/SCKは，シリアル動作モード・レジスタの設定により，入力ポートとして使用することができます。

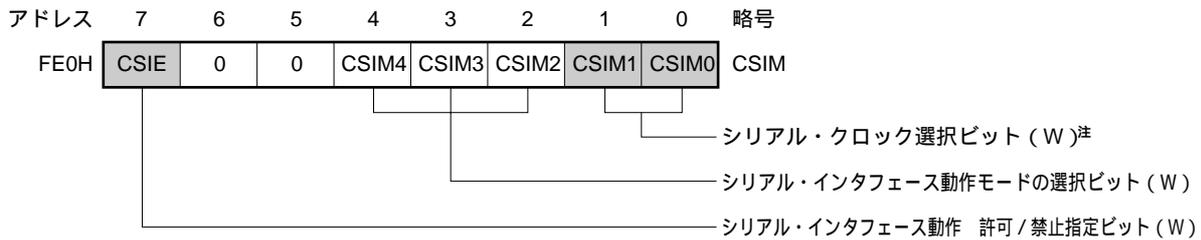
#### 〔レジスタの設定〕

動作停止モードの設定は，シリアル動作モード・レジスタ（CSIM）で行います。（CSIMのフォーマットについては，5.6.3（1）シリアル動作モード・レジスタ（CSIM）を参照）。

CSIMの操作は，8ビット操作命令により行いますが，CSIEはビット操作も可能です。また，ビットの名称を用いて操作することができます。

リセット入力によりCSIMは00Hになります。

■ は動作停止モードにおける使用ビットを示します。



注 P01/SCK端子の状態を選択可能

備考 (W) 書き込みのみ可能

#### シリアル・インタフェース動作 許可/禁止指定ビット (W)

		シフト・レジスタの動作	シリアル・クロック・カウンタ	IRQCSIフラグ	SO/SB0, SI/SB1端子
CSIE	0	シフト動作禁止	クリア	保持	ポート0 機能専用

## シリアル・クロック選択ビット (W)

P01/SCK端子は、CSIM0, CSIM1の設定により以下のような状態になります。

CSIM1	CSIM0	P01/SCK端子の状態
0	0	ハイ・インピーダンス
0	1	ハイ・レベル
1	0	
1	1	

シリアル転送中にCSIEをクリアするときは次の手順で行ってください。

割り込み許可フラグ (IECSI) をクリアして割り込み禁止状態にする。

CSIEをクリアする。

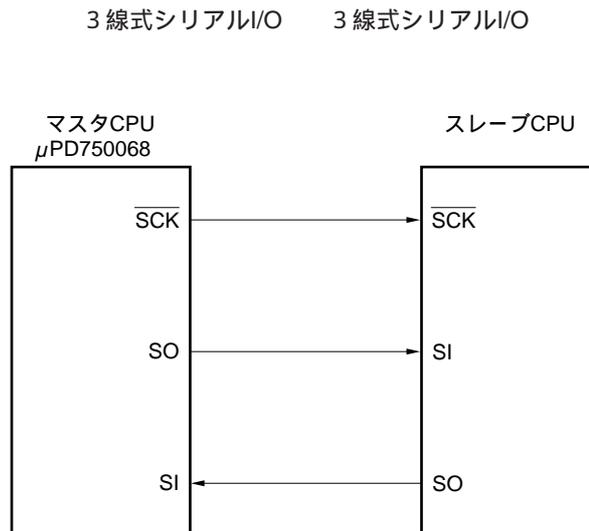
割り込み要求フラグ (IRQCSI) をクリアする。

### 5.6.5 3線式シリアルI/Oモードの動作

3線式シリアルI/Oモードは、他の75XLシリーズや75Xシリーズ、78Kシリーズ、各種周辺I/Oデバイスなどで用いられている方式と接続可能なモードです。

シリアル・クロック ( $\overline{\text{SCK}}$ )、シリアル出力 (SO)、シリアル入力 (SI) の3本のラインで通信を行います。

図5 - 45 3線式シリアルI/Oのシステム構成例



備考 μPD750068をスレーブCPUとして使用することもできます。

#### (1) レジスタの設定

3線式シリアルI/Oモードを使用するときは、次の2つのレジスタの設定により行います。

シリアル動作モード・レジスタ (CSIM)

シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

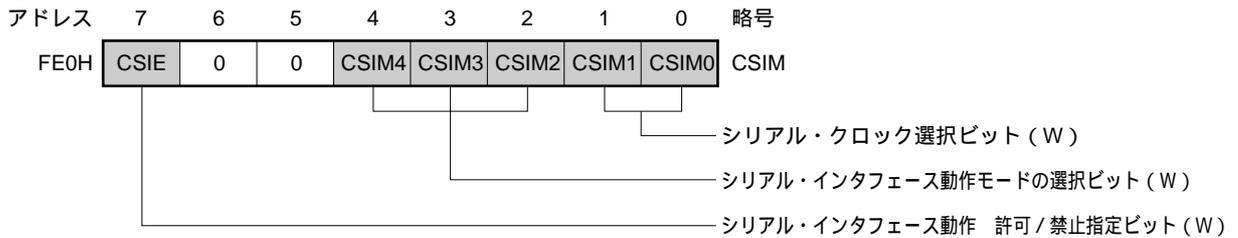
(a) シリアル動作モード・レジスタ (CSIM)

3線式シリアルI/Oモードを使用する場合は、CSIMを下の図のように設定します (CSIMのフォーマットについては5.6.3 (1) シリアル動作モード・レジスタ (CSIM) を参照)。

操作は、8ビット操作命令により行います。ビット7についてはビット操作も可能です。

リセット入力によりCSIMは00Hになります。

■ は3線式シリアルI/Oモードにおける使用ビットを示します。



備考 (W) 書き込みのみ可能

シリアル・インタフェース動作 許可/禁止指定ビット (W)

		シフト・レジスタの動作	シリアル・クロック・カウンタ	IRQCSIフラグ	SO/SB0, SI/SB1端子
CSIE	1	シフト動作可能	カウンタ動作	セット可能	各モードでの機能とポート0兼用

シリアル・インタフェース動作モードの選択ビット (W)

CSIM4	CSIM3	CSIM2	シフト・レジスタのビット順	SO端子機能	SI端子機能
x	0	0	SIO <sub>7-0</sub> XA (MSB先頭で転送)	SO (CMOS出力)	SI (CMOS入力)
		1	SIO <sub>0-7</sub> XA (LSB先頭で転送)		

備考 x : don't care

シリアル・クロック選択ビット (W)

CSIM1	CSIM0	シリアル・クロック	SCK端子モード
0	0	SCK端子への外部からの入力クロック	入 力
0	1	タイマ/イベント・カウンタ0出力 (TO0)	出 力
1	0	$f_x/2^4$ (375 kHz : 6.0 MHz動作時, 262 kHz : 4.19 MHz動作時)	
1	1	$f_x/2^3$ (750 kHz : 6.0 MHz動作時, 524 kHz : 4.19 MHz動作時)	

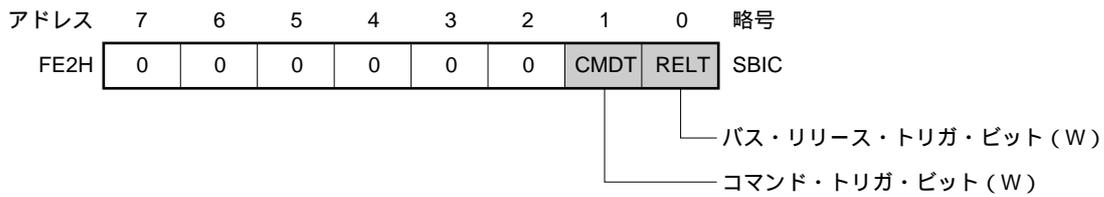
(b) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

3線式シリアルI/Oモードを使用する場合は、SBICを下の図のように設定します (SBICのフォーマットについては5.6.3 (2) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) を参照)。

操作はビット操作命令により行います。

リセット入力によりSBICは00Hになります。

■ は3線式シリアルI/Oモードにおける使用ビットを示します。



備考 (W) 書き込みのみ可能

コマンド・トリガ・ビット (W)

CMDT	コマンド信号 (CMD) のトリガ出力制御ビットです。セット (CMDT=1) することによりSOラッチがクリア (0) され、その後、CMDTビットは自動的にクリア (0) されます。
------	---

バス・リリース・トリガ・ビット (W)

RELT	バス・リリース信号 (REL) のトリガ出力制御ビットです。セット (RELT=1) することによりSOラッチがセット (1) され、その後、RELTビットは自動的にクリア (0) されます。
------	--

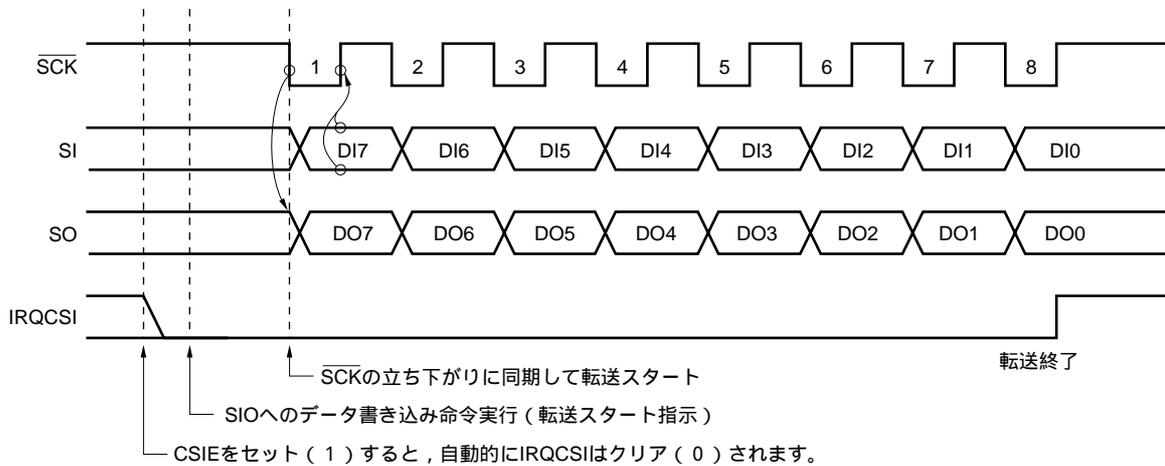
(2) 通信動作

3線式シリアルI/Oモードは、8ビット単位でデータの送受信を行います。データは、シリアル・クロックに同期して1ビットごとに送受信を行います。

シフト・レジスタのシフト動作は、シリアル・クロック ( $\overline{SCK}$ ) の立ち下がりに同期して行われます。そして、送信データがSOラッチに保持され、SO端子から出力されます。また、 $\overline{SCK}$ の立ち上がりで、SI端子に入力された受信データがシフト・レジスタにラッチされます。

8ビット転送終了によりシフト・レジスタの動作は自動的に停止し、割り込み要求フラグ (IRQCSI) がセットされます。

図5 - 46 3線式シリアルI/Oモードのタイミング



SO端子はCMOS出力となり、SOラッチの状態を出力しますので、RELTビット、CMDTビットのセットによって、SO端子出力状態を操作することができます。

ただし、シリアル転送中にはこの操作を行わないでください。

$\overline{SCK}$ 端子は、出力モード (内部システム・クロックのモード) 時にはP01出力ラッチを操作すれば、出力状態を制御できます (5.6.7  $\overline{SCK}$ 端子出力の操作参照)。

(3) シリアル・クロックの選択

シリアル・クロックの選択は、シリアル動作モード・レジスタ (CSIM) のビット0, 1の設定により行います。次の4種類のクロックを選択できます。

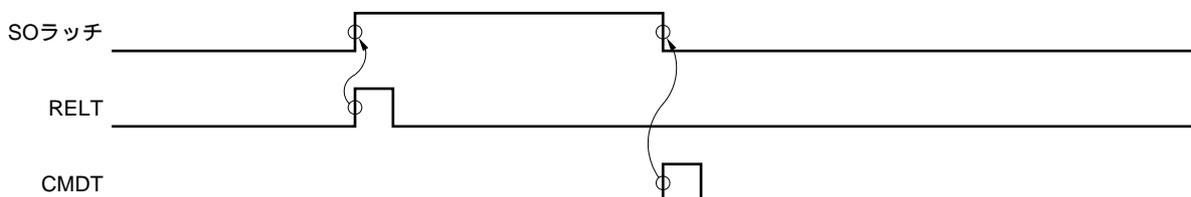
表5 - 9 シリアル・クロックの選択と応用 (3線式シリアルI/Oモード時)

モード・レジスタ		シリアル・クロック		シフト・レジスタのR/W, およびシリアル転送スタートの可能なタイミング	応 用
CSIM 1	CSIM 0	ソース	シリアル・クロック のマスク		
0	0	外 部 $\overline{\text{SCK}}$	8ビット・データ転送終了により, 自動的にマスクする。	動作可能モード時 (CSIE=1) 8ビット・シリアル転送後で, シリアル・クロックがマスクされた状態のとき $\overline{\text{SCK}}$ がハイ・レベルのとき	スレーブCPU
0	1	TOUT F/F			半二重調歩同期転送 (ソフト制御)
1	0	$f_x/2^4$			中速シリアル転送
1	1	$f_x/2^3$			高速シリアル転送

(4) 各種信号

図5 - 47にRELT, CMDTの動作を示します。

図5 - 47 RELT, CMDTの動作



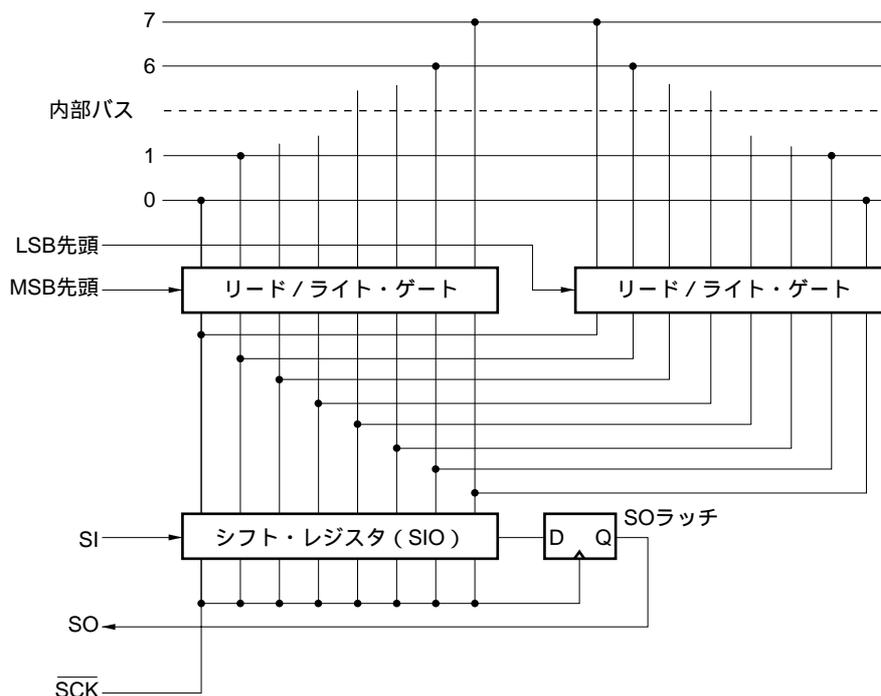
(5) MSB/LSB先頭の切り替え

3線式シリアルI/Oモードは、転送がMSB先頭か、LSBの先頭かを選択できる機能を持っています。

図5-48にシフト・レジスタ（SIO）、および内部バスの構成を示します。図5-48に示すようにMSB/LSBを反転して読み出し/書き込みを行うことができます。

MSB/LSB先頭切り替えはシリアル動作モード・レジスタ（CSIM）のビット2により指定できます。

図5-48 転送ビットの切り替え回路



先頭ビットの切り替えは、シフト・レジスタ（SIO）へのデータ書き込みのビット順を切り替えることによって実現させています。SIOのシフト順は常に同じです。

したがって、MSB/LSBの先頭ビットの切り替えは、シフト・レジスタにデータを書き込む前に行ってください。

### (6) 転送スタート

シリアル転送は、次の2つの条件を満たしたとき、シフト・レジスタ(SIO)に転送データをセットすることで開始します。

シリアル・インタフェース動作 許可/禁止ビット(CSIE)=1

8ビット・シリアル転送後、内部のシリアル・クロックが停止した状態か、または $\overline{\text{SCK}}$ がハイ・レベルの状態

**注意** シフト・レジスタにデータを書き込んだあと、CSIEを“1”にしても、転送はスタートしません。

8ビット転送終了により、シリアル転送は自動的に停止し、割り込み要求フラグ(IRQCSI)をセットします。

**例** HLレジスタで指定されたRAMのデータをSIOに転送すると同時に、SIOのデータをアキュムレータに取り込み、シリアル転送をスタートする。

```
MOV  XA, @HL    ; RAMから送信データを取り出す
SEL  MB15       ; またはCLR1 MBE
XCH  XA, SIO    ; 送信データと受信データを交換し、転送をスタートする。
```

(7) 3線式シリアルI/Oモードの応用

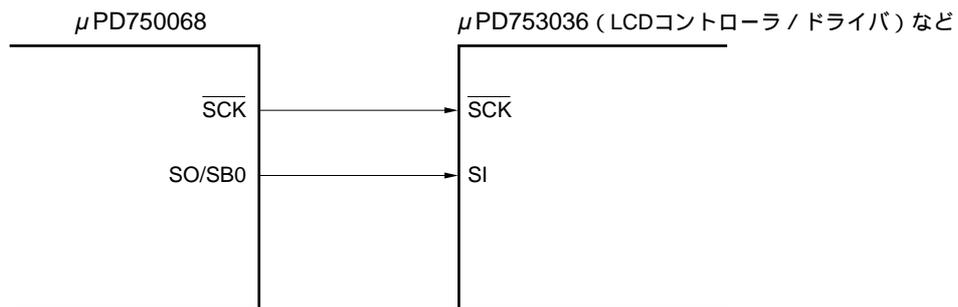
例1 . 転送クロック262 kHz (4.19 MHz動作時) で、データをMSB先頭で転送する (マスタ動作)。

プログラム例

```

CLR1   MBE
MOV    XA, #10000010B
MOV    CSIM, XA           ; 転送モードの設定
MOV    XA, TDATA         ; TDATAは転送データの格納アドレス
MOV    SIO, XA           ; 転送データのセット, 転送スタート
    
```

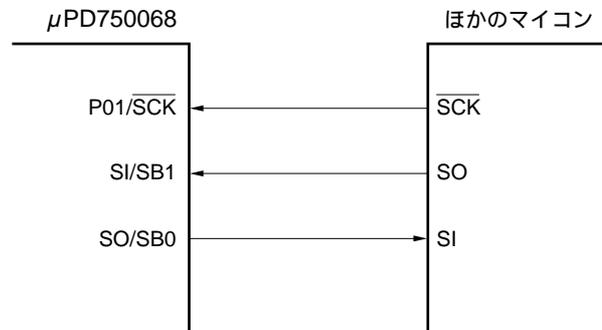
注意 2回目以降はSIOへのデータのセット (MOV SIO, XAまたはXCH XA, SIO) によって転送スタートできます。



この応用の場合、μPD750068のSI/SB1端子は入力として使用できます。

例2．外部クロックでLSB先頭のデータを送受信する（スレーブ動作）。

（この場合は、シフト・レジスタのリード/ライトをMSBとLSBを反転して行う機能を活用します。）



### プログラム例

メイン・ルーチン

```

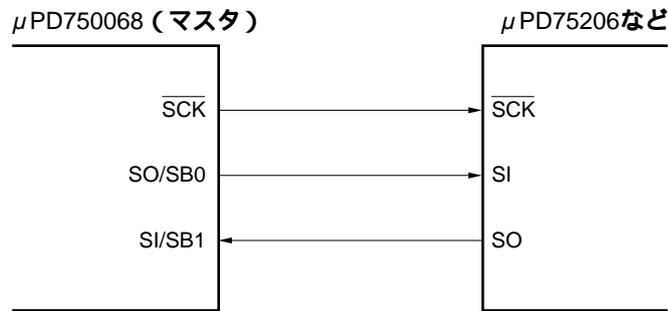
CLR1    MBE
MOV     XA, #84H
MOV     CSIM, XA      ; シリアル動作停止, MSB/LSB反転モード, 外部クロック
MOV     XA, TDATA
MOV     SIO, XA       ; 転送データ・セット, 転送スタート
EI      IECSI
EI
  
```

割り込みルーチン ( MBE=0 )

```

MOV     XA, TDATA
XCH     XA, SIO       ; 受信データ 送信データ, 転送スタート
MOV     RDATA, XA     ; 受信データ・セーブ
RETI
  
```

例3 . 転送クロックで524 kHz (4.19 MHz動作時) を使用し, 高速でデータを送受信する。



プログラム例 . . . . . マスタ側

```

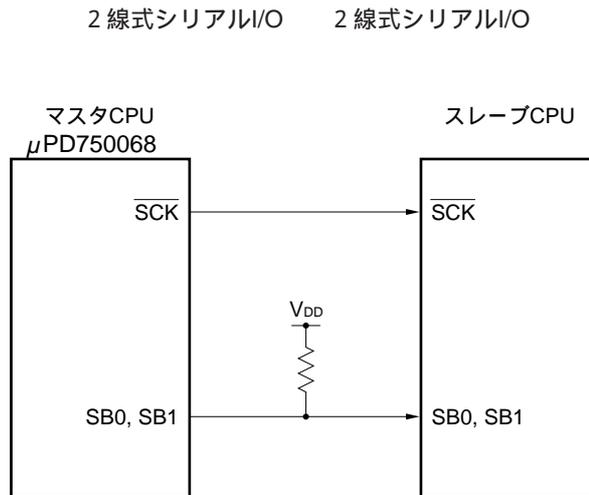
CLR1   MBE
MOV    XA, #10000011B
MOV    CSIM, XA          ; 転送モードの設定
MOV    XA, TDATA
MOV    SIO, XA          ; 転送データをセット, 転送スタート
      ⋮
      ⋮
LOOP : SKTCLR  IRQCSI    ; IRQCSIのテスト
      BR     LOOP
MOV    XA, SIO          ; 受信データ取り込み
  
```

### 5.6.6 2線式シリアルI/Oモードの動作

2線式シリアルI/Oモードは、プログラムにより任意の通信フォーマットに対応可能なモードです。

基本的にはシリアル・クロック ( $\overline{\text{SCK}}$ ) とシリアル・データ入力/出力 (SB0またはSB1) の2本のラインで通信を行います。

図5 - 49 2線式シリアルI/Oのシステム構成例



備考 μPD750068をスレーブCPUとして使用することもできます。

#### (1) レジスタの設定

2線式シリアルI/Oモードを使用するときは、次の2つのレジスタの設定により行います。

シリアル動作モード・レジスタ (CSIM)

シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

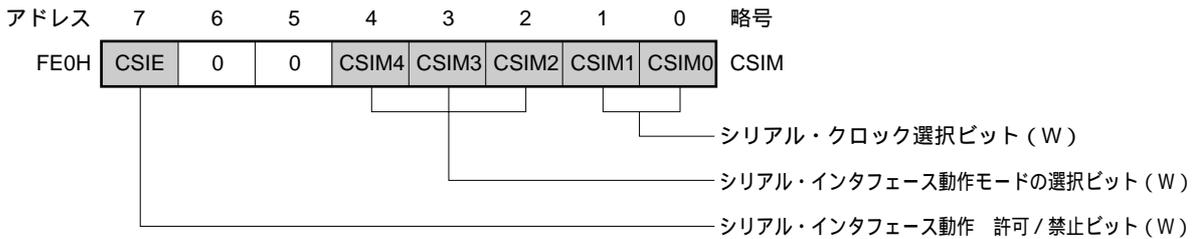
(a) シリアル動作モード・レジスタ (CSIM)

2線式シリアルI/Oモードを使用する場合は、CSIMを下の図のように設定します (CSIMのフォーマットについては5.6.3 (1) シリアル動作モード・レジスタ (CSIM) を参照)。

操作は、8ビット操作命令により行います。ビット7についてはビット操作も可能です。

リセット入力によりCSIMは00Hになります。

■ は2線式シリアルI/Oモードにおける使用ビットを示します。



備考 (W) 書き込みのみ可能

シリアル・インタフェース動作 許可/禁止指定ビット (W)

		シフト・レジスタの動作	シリアル・クロック・カウンタ	IRQCSIフラグ	SO/SB0, SI/SB1端子
CSIE	1	シフト動作可能	カウンタ動作	セット可能	各モードでの機能とポート0兼用

シリアル・インタフェース動作モードの選択ビット (W)

CSIM4	CSIM3	CSIM2	シフト・レジスタのビット順	SB0/P02端子機能	SB1/P03端子機能
0	1	1	SIO <sub>7-0</sub> XA (MSB先頭で転送)	SB0 (N-chオープン・ドレイン) 入出力	P03 (CMOS入力)
1				P02 (CMOS入力)	SB1 (N-chオープン・ドレイン) 入出力

シリアル・クロック選択ビット (W)

CSIM1	CSIM0	シリアル・クロック	SCK端子モード
0	0	SCK端子への外部からの入力クロック	入 力
0	1	タイマ/イベント・カウンタ0出力 (TO0)	出 力
1	0	f <sub>x</sub> /2 <sup>6</sup> (93.8 kHz : 6.0 MHz動作時, 65.5 kHz : 4.19 MHz動作時)	
1	1		

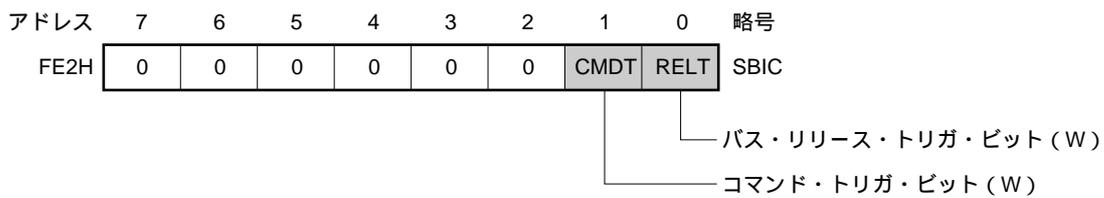
(b) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

2線式シリアルI/Oモードを使用する場合は、SBICを下の図のように設定します (SBICのフォーマットについては、5.6.3 (2) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) を参照)。

操作はビット操作命令により行います。

リセット入力によりSBICは00Hになります。

■ は2線式シリアルI/Oモードにおける使用ビットを示します。



備考 (W) 書き込みのみ可能

コマンド・トリガ・ビット (W)

CMDT	コマンド信号 (CMD) のトリガ出力制御ビットです。セット (CMDT=1) することによりSOラッチがクリア (0) され、その後、CMDTビットは自動的にクリア (0) されます。
------	---

バス・リリース・トリガ・ビット (W)

RELT	バス・リリース信号 (REL) のトリガ出力制御ビットです。セット (RELT=1) することによりSOラッチがセット (1) され、その後、RELTビットは自動的にクリア (0) されます。
------	--

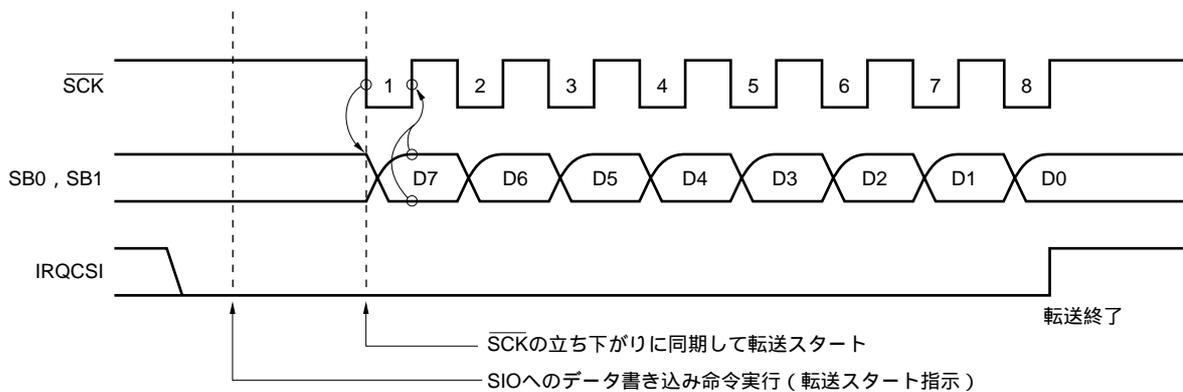
## (2) 通信動作

2線式シリアルI/Oモードでは、8ビット単位でデータの送受信を行います。データは、シリアル・クロックに同期して1ビットごとに送受信を行います。

シフト・レジスタのシフト動作は、シリアル・クロック ( $\overline{SCK}$ ) の立ち下がりに同期して行われます。そして、送信データがSOラッチに保持され、SB0/P02 (またはSB1/P03) 端子からMSBを先頭にして出力されます。また、 $\overline{SCK}$ の立ち上がりで、SB0端子 (またはSB1) から入力された受信データがシフト・レジスタにラッチされます。

8ビット転送終了によりシフト・レジスタの動作は自動的に停止し、割り込み要求フラグ (IRQCSI) がセットされます。

図5 - 50 2線式シリアルI/Oモードのタイミング



SB0 (またはSB1) 端子のシリアル・データ・パスに指定された端子はN-chオープン・ドレイン入出力となりますので、外部でプルアップする必要があります。また、データの受信時にはN-chトランジスタをオフさせる必要があるため、SIOにはあらかじめFFHを書き込んでおきます。

SB0 (またはSB1) 端子はSOラッチの状態を出力しますので、RELTビット、CMDTビットのセットによって、SB0 (またはSB1) 端子の出力状態を操作することができます。

ただし、シリアル転送中にはこの操作を行わないでください。

$\overline{SCK}$ 端子は、出力モード (内部システム・クロックのモード) 時にはP01出力ラッチを操作すれば、出力状態を制御できます (5.6.7  $\overline{SCK}$ 端子出力の操作参照)。

(3) シリアル・クロックの選択

シリアル・クロックの選択は、シリアル動作モード・レジスタ (CSIM) のビット0, 1の設定により行います。次の3種類のクロックを選択できます。

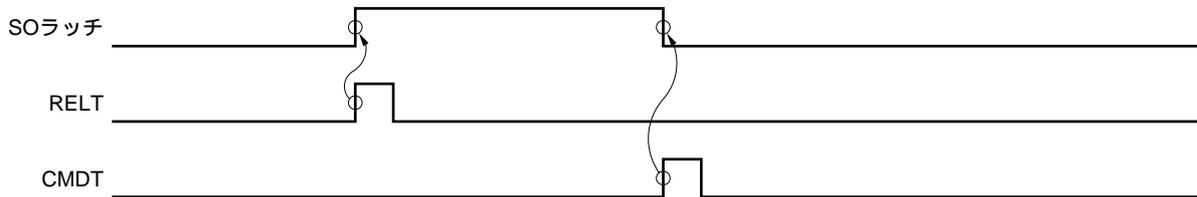
表5 - 10 シリアル・クロックの選択と応用 (2線式シリアルI/Oモード時)

モード・レジスタ		シリアル・クロック		シフト・レジスタのR/W, およびシリアル転送スタートの可能なタイミング	応 用
CSIM 1	CSIM 0	ソース	シリアル・クロック のマスク		
0	0	外 部 $\overline{\text{SCK}}$	8ビット・データ転送終了により, 自動的にマスクする。	動作可能モード時 (CSIE=1) 8ビット・シリアル転送後で, シリアル・クロックがマスクされた状態のとき $\overline{\text{SCK}}$ がハイ・レベルのとき	スレーブCPU
0	1	TOUT F/F			任意速シリアル転送
1	0	$f_x/2^6$			低速シリアル転送
1	1				

(4) 各種信号

図5 - 51にRELT, CMDTの動作を示します。

図5 - 51 RELT, CMDTの動作



### (5) 転送スタート

シリアル転送は、次の2つの条件を満たしたとき、シフト・レジスタ(SIO)に転送データをセットすることで開始します。

シリアル・インタフェース動作 許可/禁止ビット(CSIE)=1

8ビット・シリアル転送後、内部のシリアル・クロックが停止した状態か、または $\overline{\text{SCK}}$ がハイ・レベルの状態

**注意1**．シフト・レジスタにデータを書き込んだあと、CSIEを“1”にしても、転送はスタートしません。

**2**．データ受信時にはN-chトランジスタをオフする必要がありますので、SIOにはあらかじめFFHを書き込んでおいてください。

8ビット転送終了により、シリアル転送は自動的に停止し、割り込み要求フラグ(IRQCSI)をセットします。

### (6) エラーの検出

2線式シリアルI/Oモードでは、送信中のシリアル・バスSB0またはSB1の状態が送信しているデバイスのシフト・レジスタSIOにも取り込まれるため、次の方法によって送信エラーの検出をすることができます。

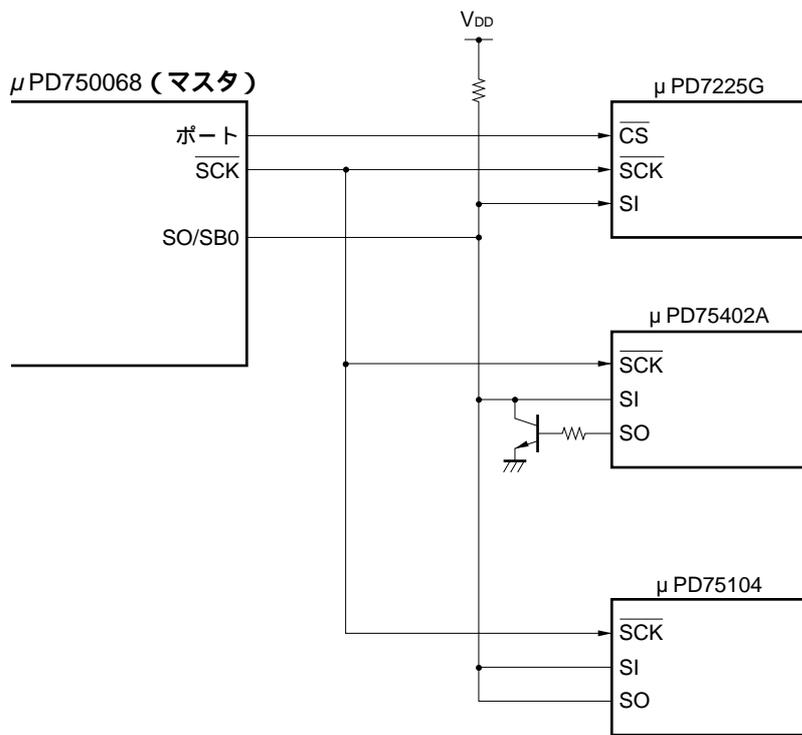
#### 送信開始前と送信終了後のSIOのデータを比較する方法

この場合、2つのデータが異なっていれば送信エラーが発生したと判断します。

(7) 2線式シリアルI/Oモードの応用

シリアル・バスを構成し、複数のデバイスを接続する。

例  $\mu$ PD750068をマスタとし、スレーブに $\mu$ PD75104、 $\mu$ PD75402A、 $\mu$ PD7225Gを接続してシステムを構成する。



$\mu$ PD75104はSI端子とSO端子を接続し、シリアル・データを出力するとき以外は、シリアル動作モード・レジスタを操作し、出力バッファをオフしてバスを開放します。

$\mu$ PD75402AはSO端子をハイ・インピーダンスにできないため、図のようにトランジスタを接続し、オープン・コレクタ出力となるようにします。そしてデータを入力するタイミングでは、あらかじめシフト・レジスタに00Hを書き込んでトランジスタをオフさせます。

各マイコンがいつデータを出力するかは、あらかじめ、取り決めしておきます。

シリアル・クロックはマスタ・マイコンである $\mu$ PD750068が出力し、ほかのスレーブ・マイコンはいずれも外部クロックで動作します。

### 5.6.7 SCK端子出力の操作

SCK/P01端子には、出力ラッチが内蔵されているため、通常のシリアル・クロック以外に、ソフトウェア操作によりスタティック出力も可能です。

また、P01出力ラッチの操作により、SCKの数をソフトウェアで任意に設定することができます（SO/SB0, SI/SB1端子の制御はSBICのRELT, CMDTビットによって行います）。

次に、SCK/P01端子出力の操作方法を示します。

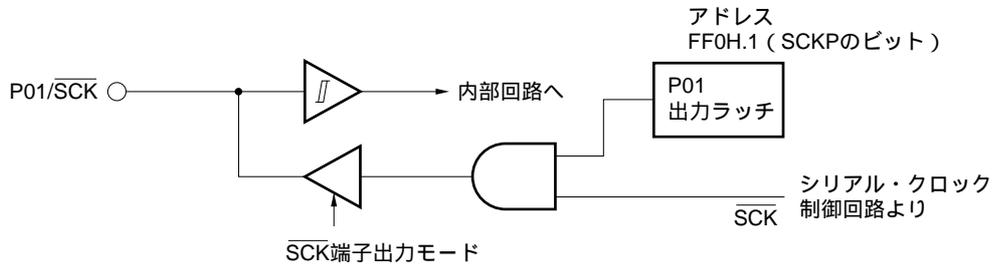
シリアル動作モード・レジスタ（CSIM）を設定します（SCK端子：出力モード）。シリアル転送停止中では、シリアル・クロック制御回路からのSCKが1となっています。

P01出力ラッチを、ビット操作命令により操作します。

**例** ソフトウェアによりSCK/P01端子に1クロック出力する。

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #10000011B     ;SCK (fx/2³) , 出力モード
MOV    CSIM, XA
CLR1   OFF0H. 1          ;SCK/P01  0
SET1   OFF0H. 1          ;SCK/P01  1
```

図5 - 52 SCK/P01端子の構成



P01出力ラッチは、FF0H番地のビット1にマッピングされています。RESET信号発生によりP01出力ラッチは“1”にセットされます。

**注意 1** . 通常のシリアル転送中は、P01出力ラッチは1にセットしておく必要があります。

2 . P01出力ラッチのアドレスは下記のように「PORT0.1」では指定できません。オペランドにはアドレス（OFF0H.1）を直接記述するかSCKPで指定してください。ただし、その命令実行時には、あらかじめMBE = 0または（MBE = 1, MBS = 15）としておく必要があります。

使用不可		使用可	
CLR1	PORT0.1	CLR1	OFF0H.1
SET1	PORT0.1	SET1	OFF0H.1
		CLR1	SCKP
		SET1	SCKP

## 5.7 A/Dコンバータ

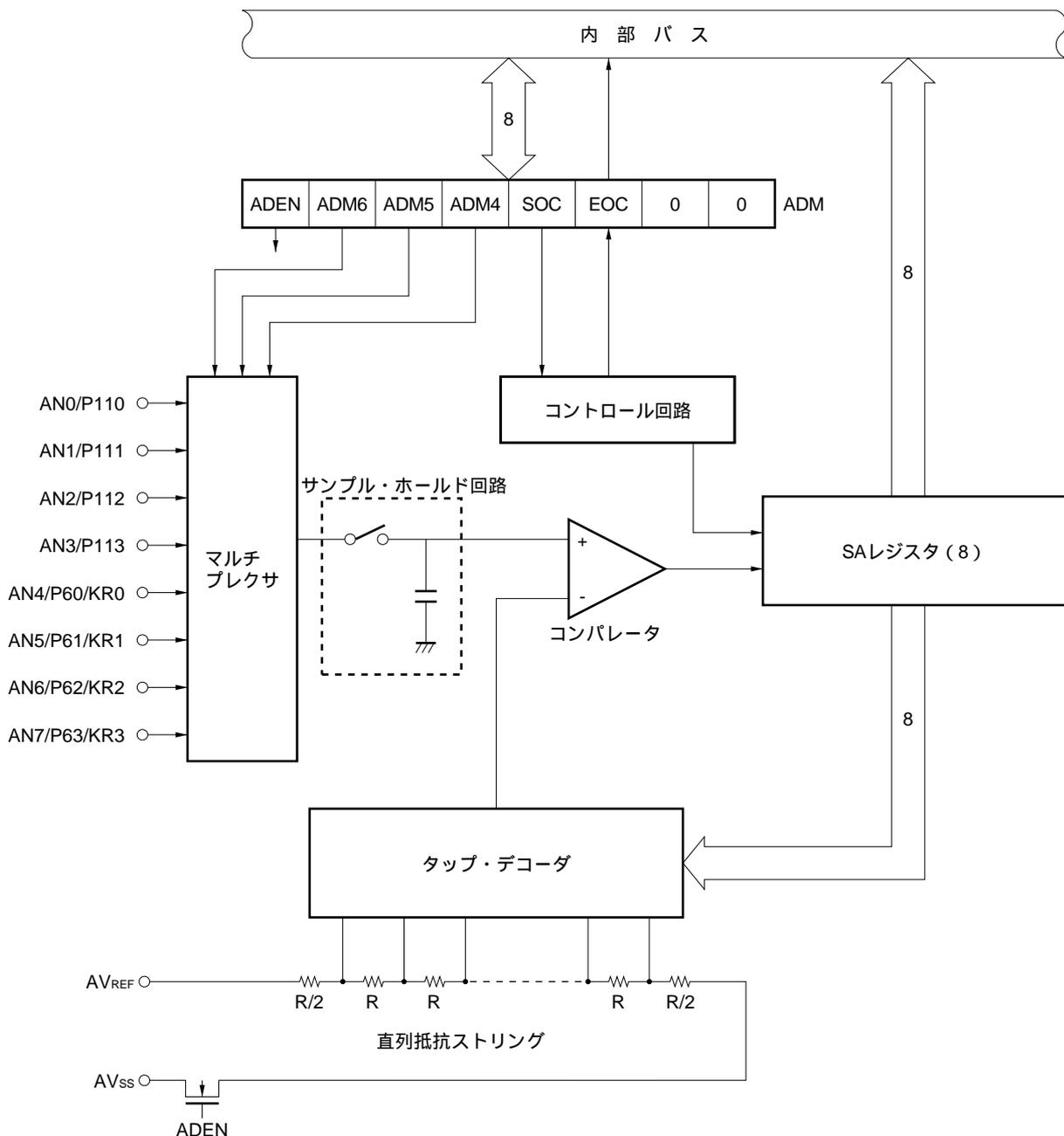
μPD750068は、8チャンネルのアナログ入力（AN0-AN7）を持つ8ビット精度のアナログ/デジタル（A/D）コンバータを内蔵しています。

A/Dコンバータは、逐次比較法を採用しています。

### 5.7.1 A/Dコンバータの構成

A/Dコンバータは、図5-53に示すように構成されています。

図5-53 A/Dコンバータのブロック図



## (1) A/Dコンバータの各端子

### (a) AN0-AN7

A/Dコンバータへの8チャンネルのアナログ信号入力端子です。A/D変換するアナログ信号を入力します。

AN0-AN3とP110-P113, AN4-AN7とP60/KR0-P63/KR3はそれぞれ端子を兼用しています<sup>注</sup>。

A/Dコンバータ内部には、サンプル・ホールド回路が内蔵されており、A/D変換中は、アナログ入力電圧は内部で保持されています。

**注** AN4-AN7を使用する場合は、A/D変換をする前に次の設定が必要です、  
ポート6を入力モードに設定する。  
ポート6の内蔵プルアップ抵抗を接続しない状態にする。  
(詳細は5.1 デジタル入出力ポートを参照)

**注意** AN0-AN7入力電圧は規格の範囲内でご使用ください。特に $V_{DD}$ 以上、 $V_{SS}$ 以下(絶対最大定格の範囲内でも)の電圧が入力されると、そのチャンネルの変換値が不定となり、またほかのチャンネルの変換値にも影響を与えることがあります。

### (b) AVREF

A/Dコンバータの基準電圧を入力する端子です。

AVREF-AVSS間にかかる電圧に基づいて、AN0-AN7に入力される信号をデジタル信号に変換します。

### (c) AVSS

A/DコンバータのGND端子です。常に $V_{SS}$ と同じ電位にしてください。

(2) A/D変換モード・レジスタ (ADM)

ADMは、変換の許可、アナログ入力チャネルの選択、変換のスタート指示、変換終了の検出を行う8ビット・レジスタです。

ADMは、8ビット操作命令により設定します。

ビット2 (EOC)、ビット3 (SOC)、およびビット7 (ADEN) は、ビット単位で操作可能です。

$\overline{\text{RESET}}$ 信号発生により、ADMは04Hにイニシャライズされます (EOCのみ“1”にセットされ、ほかは“0”にクリアされます)。

図5 - 54 A/D変換モード・レジスタのフォーマット

アドレス	7	6	5	4	3	2	1	0	略号
FD8H	ADEN	ADM6	ADM5	ADM4	SOC	EOC	0	0	ADM

A/D変換許可フラグ

ADEN	0	A/Dコンバータを使用しない
	1	A/Dコンバータを使用する

アナログ・チャネル選択ビット

ADM6	ADM5	ADM4	アナログ・チャネル
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

変換スタート指示ビット

SOC	セットすることにより、A/D変換を開始する。 変換終了後、自動的にクリアされる。
-----	---

変換終了検出フラグ

EOC	0	変換中
	1	変換終了

**注意** A/D変換は、SOCのセット後最大 $2^4/f_x$ 秒 (2.67  $\mu$ s :  $f_x = 6.0$  MHz動作時)<sup>注</sup>遅れて開始します (5.7.2 A/Dコンバータの動作参照)。

**注**  $f_x = 4.19$  MHz動作時は3.81  $\mu$ sになります。

### (3) SAレジスタ (SA)

SAレジスタ (Successive Approximation Register) は、A/D変換された結果を格納する8ビット・レジスタです。

SAは、8ビット操作命令により読み出します。読み出し専用で、書き込み操作やビット操作はできません。

$\overline{\text{RESET}}$ 信号発生により、SAは7FHになります。

**注意1** . ADMレジスタのビット3 (SOC) を“1”にセットしてA/D変換を開始すると、SA内の変換結果は破壊され、新たな変換結果が格納されるまでSAは不定となります。

**2** .  $\text{AV}_{\text{REF}}$ 端子にGNDレベルを入力した場合やアナログ入力端子に $\text{AV}_{\text{REF}}$ と $\text{V}_{\text{DD}}$ 間の電位を入力した場合、またADENが0の状態A/D変換を開始した場合には、SAにはFFHが格納されます。

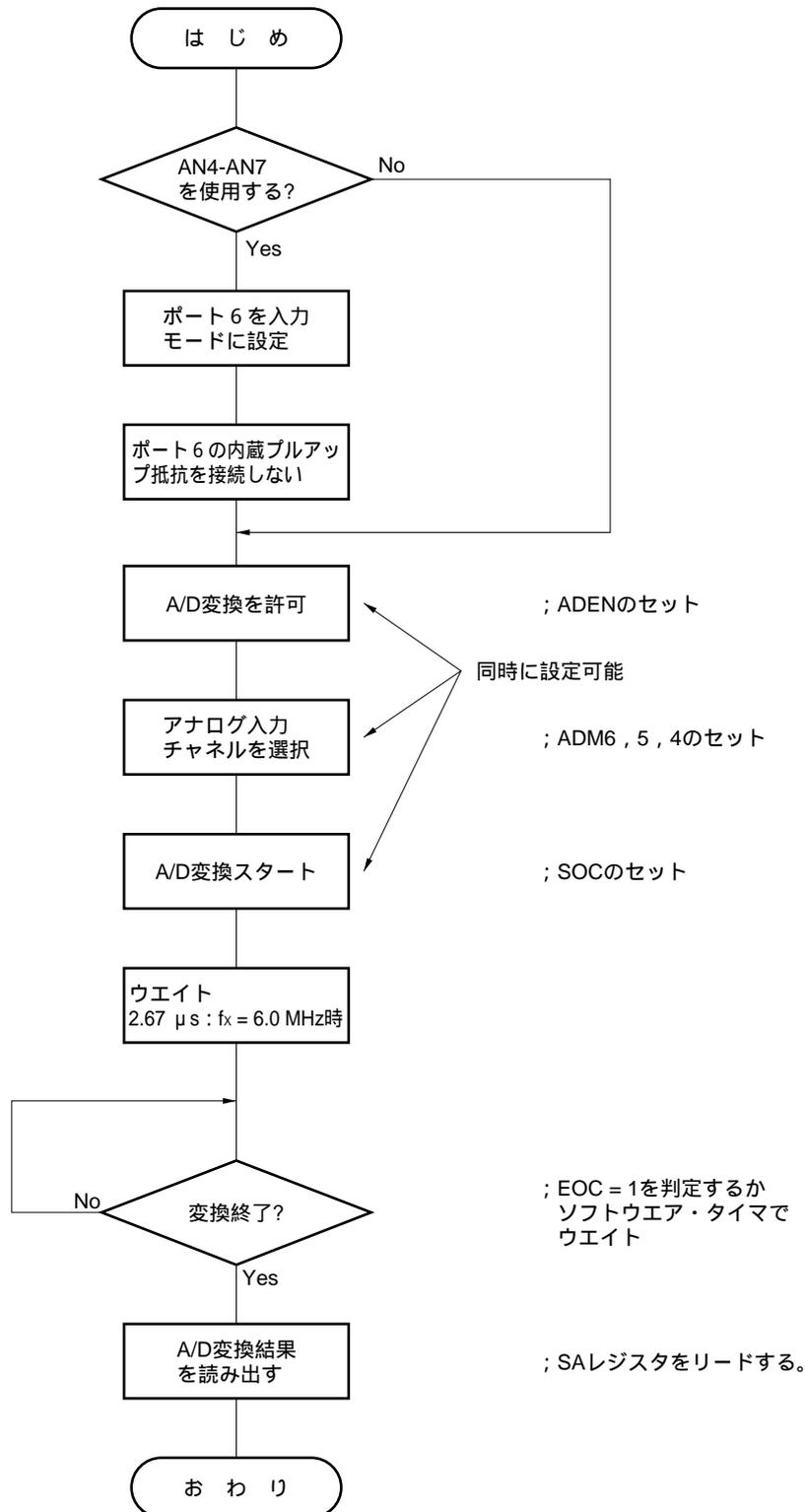
## 5.7.2 A/Dコンバータの動作

A/D変換されるアナログ入力信号は、A/D変換モード・レベルのビット6, 5, 4 (ADM 6, 5, 4) の設定により指定されます。

A/D変換は、ADMのビット7 (ADEN) とビット3 (SOC) を“1”にセットすることにより開始します (ADENのセットは $\overline{\text{RESET}}$ 後のみ必要)。SOCは、セット後、自動的に“0”にクリアされます。A/D変換は、ハードウェアによって逐次比較法で行われ、変換結果の8ビット・データがSAレジスタに格納されます。また、変換終了時にADMのビット2 (EOC) が“1”にセットされます。

A/D変換のタイミング・チャートを図5-55に示します。

A/Dコンバータは、次の手順で操作してください。



**注意** SOCのセット後、A/D変換が開始されてEOCがクリアされるまで最大 $2^4/f_x$  ( $f_x = 6.0 \text{ MHz}$ 時、 $2.67 \mu\text{s}$ )<sup>注</sup>遅れます。このため、EOCのテストはSOCのセット後、表5 - 11に示す時間経過後行ってください。A/D変換時間もあわせて表5 - 11に示します。

**注**  $f_x = 4.19 \text{ MHz}$ 時は $3.81 \mu\text{s}$ になります。

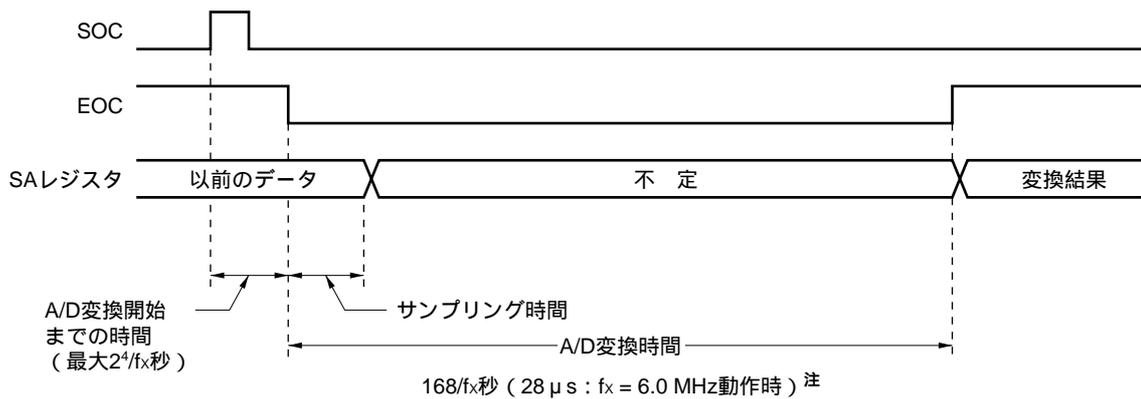
表5 - 11 SCC , PCCの設定

SCC , PCCの設定値				A/D変換時間	SOCのセット後 , EOCの テストまでのウェイト時間	SOCのセット後 , A/D変換 終了までのウェイト時間
SCC3	SCC0	PCC1	PCC0			
0	0	0	0	168/f <sub>x</sub> 秒 (28 μs : f <sub>x</sub> = 6.0 MHz動作時) <sup>注</sup>	ウェイト不要	3マシン・サイクル
		0	1		1マシン・サイクル	11マシン・サイクル
		1	0		2マシン・サイクル	21マシン・サイクル
		1	1		4マシン・サイクル	42マシン・サイクル
0	1	x	x		ウェイト不要	ウェイト不要
1	x	x	x	変換動作停止	-	-

注 f<sub>x</sub> = 4.19 MHz動作時は40.1 μs。

備考 x : don't care

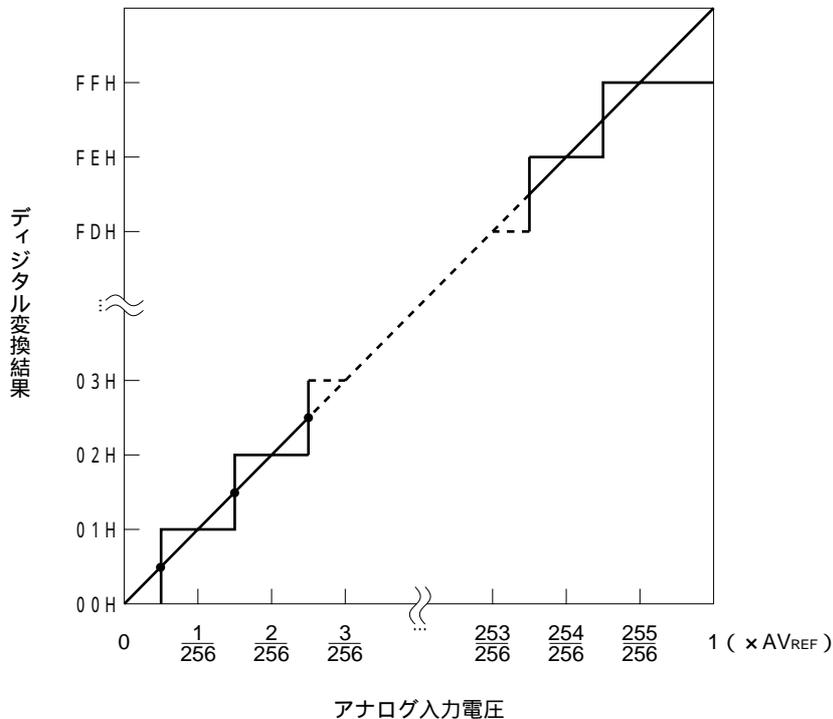
図5 - 55 A/D変換のタイミング・チャート



注 f<sub>x</sub> = 4.19 MHz動作時は40.1 μs。

アナログ入力電圧とA/D変換された8ビット・デジタル・データとの対応は図5 - 56のようになります。

図5 - 56 アナログ入力電圧とA/D変換結果の関係（理想的な場合）



### 5.7.3 スタンバイ・モード時の注意

A/Dコンバータは、メイン・システム・クロックによって動作します。したがって、STOPモード、または、サブシステム・クロックでのHALTモード時には動作は停止します。このときにも、AVREF端子には電流が流れ込みますので、システム全体としての消費電力を少なくするには、この電流をカットする必要があります。そのためには、A/D変換を禁止（ADEN = 0）することで消費電力を小さくすることができます。

### 5.7.4 その他、使用上の注意

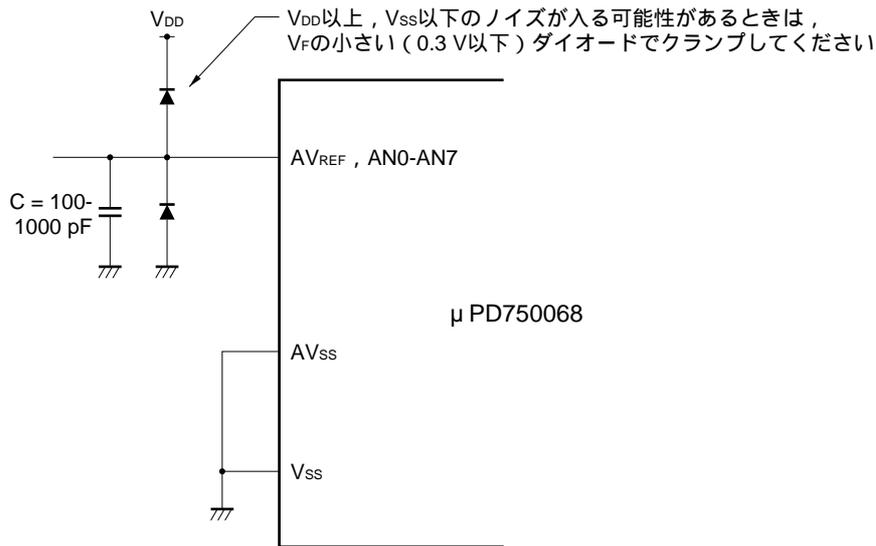
#### (1) AN0-AN7入力範囲について

AN0-AN7入力電圧は規格の範囲内でご使用ください。特に、V<sub>DD</sub>以上、V<sub>SS</sub>以下（絶対最大定格の範囲内でも）の電圧が入力されると、そのチャンネルの変換値は不定となり、また、ほかのチャンネルの変換値にも影響を与えることがあります。

## (2) ノイズ対策について

8ビット精度を保つためには、 $AV_{REF}$ 、 $AN0$ - $AN7$ 端子へのノイズに注意する必要があります。アナログ入力源の出力インピーダンスが高いほど影響が大きくなりますので、ノイズを低減するために図5-57のように、 $C$ を外付けすることを推奨します。

図5-57 アナログ入力端子の処理



## (3) $AN0$ - $AN3$ 端子について

アナログ入力 $AN0$ - $AN3$ は入力ポート(PORT11)端子と兼用になっています。 $AN0$ - $AN3$ のいずれかを選択してA/D変換をする場合、変換中にPORT11の入力命令は実行しないでください。変換精度が低下することがあります。

## (4) $AN4$ - $AN7$ 端子について

アナログ入力 $AN4$ - $AN7$ は、入出力ポート(PORT6)端子かつ $KR0$ - $KR3$ 端子と兼用になっています。 $AN4$ - $AN7$ のいずれかを選択してA/D変換をする場合、まずPORT6を入力モードにしてください。このとき、変換中にPORT6の入出力命令は実行しないでください。また、内蔵プルアップ抵抗の接続を指定しないでください。変換精度が低下することがあります。

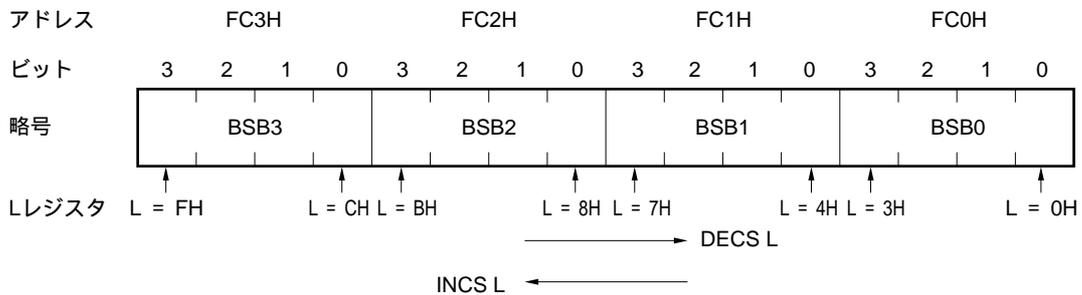
また、A/D変換中の端子に隣接する端子へデジタル・パルス印加すると、カップリング・ノイズによってA/D変換値が期待どおりに得られないことがあります。したがって、A/D変換中の隣接端子へのパルス印加はしないようにしてください。

## 5.8 ビット・シーケンシャル・バッファ.....16ビット

ビット・シーケンシャル・バッファ (BSB) は、ビット操作用の特殊データ・メモリで、特にアドレスおよびビット指定を順次変更してビット操作が容易にできるので、ビット長の長いデータをビット単位で処理するときに便利です。

このデータ・メモリは16ビット構成になっており、ビット操作命令のpmem. @Lアドレッシングが可能で、Lレジスタにより間接ビット指定ができます。この場合、Lレジスタをプログラム・ループ内でインクリメント/デクリメントするだけで順次指定ビットを移動させながら処理を進めることができます。

図5 - 58 ビット・シーケンシャル・バッファのフォーマット



備考1 . pmem. @Lアドレッシングでは、Lレジスタに対応して指定ビットが移動します。

2 . pmem. @Lアドレッシングでは、MBE, MBSの指定にかかわらずに、いつでもBSBを操作できます。

直接アドレッシングでもデータ操作が可能です。1ビット、4ビット、8ビットの直接アドレッシングとpmem. @Lアドレッシングとを合わせて、1ビット・データの連続入力や連続出力などに応用できます。なお、8ビット操作の場合は、BSB0, BSB2を指定して上位、下位8ビットずつを操作します。

例 BUFF1, 2の16ビット・データをポート3のビット0からシリアルに出力する。

```
CLR1  MBE
MOV   XA, BUFF1
MOV   BSB0, XA      ; BSB0, 1をセット
MOV   XA, BUFF2
MOV   BSB2, XA      ; BSB2, 3をセット
MOV   L, #0
LOOP0: SKT   BSB0, @L      ; BSBの指定ビットをテスト
BR    LOOP1
NOP                    ; ダミー ( タイミング調整 )
SET1  PORT3. 0      ; ポート3のビット0をセット
BR    LOOP2
LOOP1: CLR1  PORT3. 0      : ポート3のビット0をクリア
NOP                    ; ダミー ( タイミング調整 )
NOP
LOOP2: INCS  L          ; L L+ 1
BR    LOOP0
RET
```

(メモ)

## 第6章 割り込み機能とテスト機能

μPD750068には、7種類のベクタ割り込みソースと、2種類のテスト入力があり多彩な応用が可能です。また、μPD750068の割り込み制御回路には次のような特色があり、非常に高速な割り込み処理が可能となります。

### (1) 割り込み機能

- (a) 割り込み許可フラグ (IE × × ×) と、割り込みマスタ許可フラグ (IME) により受け付け可否を制御できるハードウェア制御のベクタ割り込み機能。
- (b) 割り込みスタート・アドレスを任意に設定可能。
- (c) 割り込みプライオリティ選択レジスタ (IPS) により優先順位が指定できる多重割り込み機能。
- (d) 割り込み要求フラグ (IRQ × × ×) のテスト機能 (ソフトウェアで割り込み発生の確認可能)。
- (e) スタンバイ・モードの解除 (割り込み許可フラグにより、解除する割り込みの選択可能)。

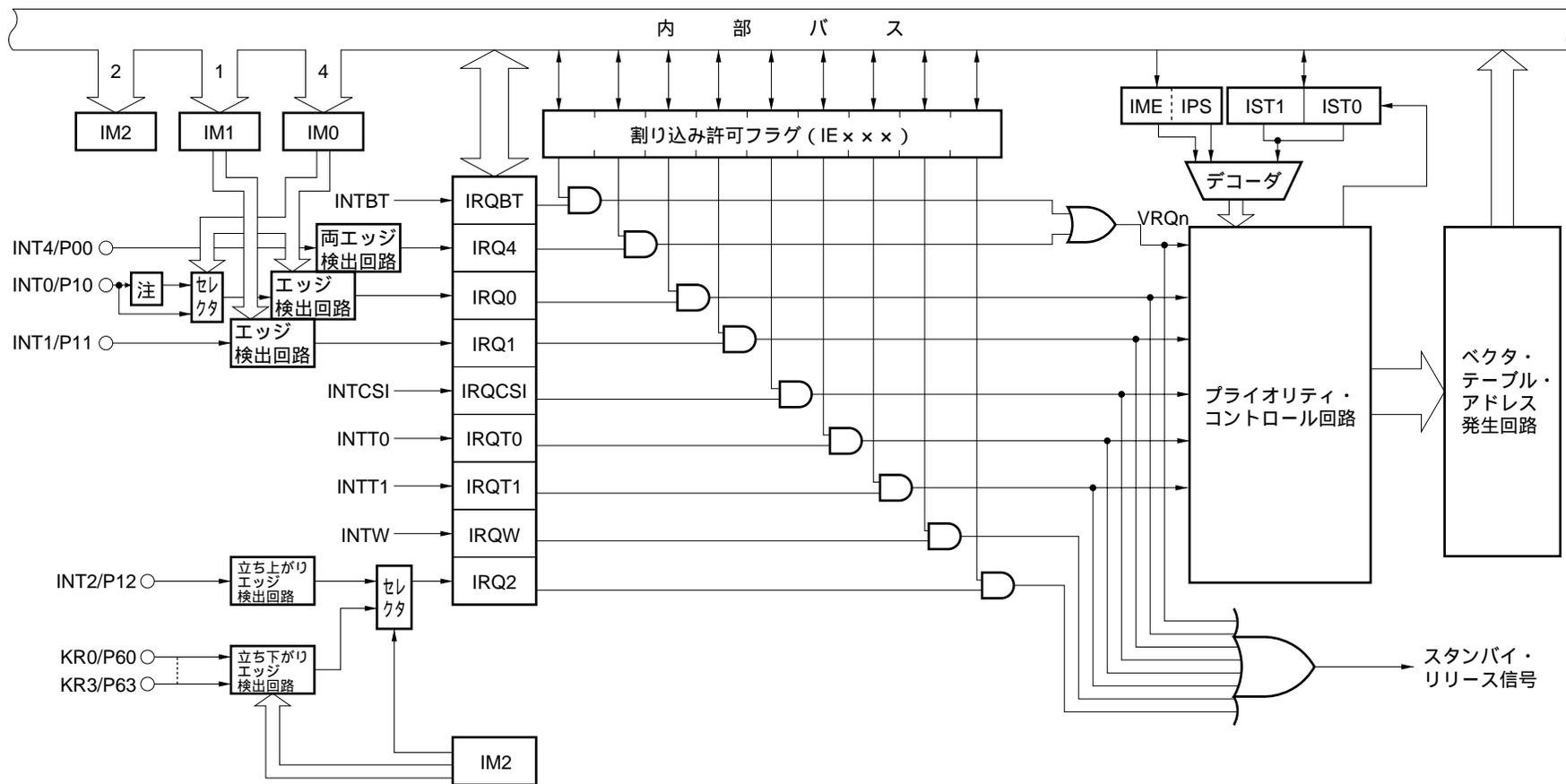
### (2) テスト機能

- (a) ソフトウェアでテスト要求フラグ (IRQ × × ×) 発生の確認可能。
- (b) スタンバイ・モードの解除 (テスト許可フラグにより、解除するテスト・ソースの選択可能)。

## 6.1 割り込み制御回路の構成

割り込み制御回路は図6-1のように構成されており、各ハードウェアはデータ・メモリ空間にマッピングされています。

図 6 - 1 割り込み制御回路ブロック図



注 ノイズ除去回路 (ノイズ除去回路選択時はスタンバイ・リリース不可)

## 6.2 割り込みソースの種類とベクタ・テーブル

μPD750068には、次の7種類の割り込みソースがあり、ソフトウェア制御による多重割り込みが可能です。

表6-1 割り込みソースの種類

割り込みソース		内/外	割り込み優先順位 <sup>注</sup>	ベクタ割り込み要求信号 (ベクタ・テーブル・アドレス)
INTBT	(ベーシック・インターバル・タイマ/ウォッチドッグ・タイマからの基準時間間隔信号)	内	1	VRQ1 (0002H)
INT4	(立ち上がりおよび立ち下りの両エッジ検出)	外		
INT0	(立ち上がり, 立ち下がりエッジ選択)	外	2	VRQ2 (0004H)
INT1		外	3	VRQ3 (0006H)
INTCSI	(シリアル・データ転送終了信号)	内	4	VRQ4 (0008H)
INTT0	(タイマ/イベント・カウンタ0のカウント・レジスタとモジュロ・レジスタとの一致信号)	内	5	VRQ5 (000AH)
INTT1	(タイマ/イベント・カウンタ1のカウント・レジスタとモジュロ・レジスタとの一致信号)	内	6	VRQ6 (000CH)

注 割り込み優先順位は、複数の割り込み要求が同時に発生した場合に、優先される順位です。

図6-2 割り込みベクタ・テーブル

0002H	MBE	RBE	INTBT/INT4	スタート・アドレス	(上位6ビット)
					"
0004H	MBE	RBE	INT0	スタート・アドレス	(上位6ビット)
					"
0006H	MBE	RBE	INT1	スタート・アドレス	(上位6ビット)
					"
0008H	MBE	RBE	INTCSI	スタート・アドレス	(上位6ビット)
					"
000AH	MBE	RBE	INTT0	スタート・アドレス	(上位6ビット)
					"
000CH	MBE	RBE	INTT1	スタート・アドレス	(上位6ビット)
					"

表6-1の中で、割り込み優先順位と書かれているのは、複数の割り込み要求が同時に発生した場合や、複数の割り込み要求が保留されていた場合に、割り込みが実行される順位を示します。

ベクタ・テーブルには割り込み処理の開始アドレスと、割り込み処理中のMBE, RBEの設定値を書き込みます。ベクタ・テーブルの設定は、アセンブラ疑似命令 (VENTn : n = 1-6) によって行います。

例 INTBT/INT4のベクタ・テーブルの設定

VENT1    MBE = 0, RBE = 0, GOTOBT

- アドレス0002のベクタ・テーブル
- 割り込み処理ルーチンでのMBEの設定
- 割り込み処理ルーチンでのRBEの設定
- 割り込み処理ルーチンの開始アドレスを示すシンボル

**注意** VENTn (n = 1-6) 命令のオペランドに記述した内容 (MBE, RBE, 開始アドレス) は、2<sup>n</sup>番地のベクタ・テーブル・アドレスに格納されます。

例 INTBT/INT4とINTT0のベクタ・テーブルの設定。

VENT1            MBE = 0, RBE = 0, GOTOBT ; INTBT/INT4スタート・アドレス  
 VENT5            MBE = 0, RBE = 1, GOTOT0 ; INTT0スタート・アドレス

## 6.3 割り込み機能を制御する各種ハードウェア

### (1) 割り込み要求フラグ，割り込み許可フラグ

割り込み要求フラグ (IRQ × × ×) には，割り込みソースに対応して次に示す7つがあります。

INT0割り込み要求フラグ (IRQ0)    シリアル・インタフェース割り込み要求フラグ (IRQCSI)  
 INT1割り込み要求フラグ (IRQ1)    タイマ/イベント・カウンタ0割り込み要求フラグ (IRQT0)  
 INT4割り込み要求フラグ (IRQ4)    タイマ/イベント・カウンタ1割り込み要求フラグ (IRQT1)  
 BT割り込み要求フラグ (IRQBT)

割り込み要求フラグは，割り込み要求発生で“1”にセットされ，割り込み処理が実行されると自動的に“0”にクリアされます。ただし，IRQBTとIRQ4はベクタ・アドレスを共用しているのでクリア動作が異なります(6.6 ベクタ・アドレス共用割り込み処理参照)。

割り込み許可フラグ (IE × × ×) には，割り込み要求フラグに対応して次に示す7つがあります。

INT0割り込み許可フラグ (IE0)    シリアル・インタフェース割り込み許可フラグ (IECSI)  
 INT1割り込み許可フラグ (IE1)    タイマ/イベント・カウンタ0割り込み許可フラグ (IET0)  
 INT4割り込み許可フラグ (IE4)    タイマ/イベント・カウンタ1割り込み許可フラグ (IET1)  
 BT割り込み許可フラグ (IEBT)

割り込み許可フラグは，内容が“1”のとき割り込みを許可し，“0”のとき禁止します。

割り込み要求フラグがセットされ，割り込み許可フラグが割り込みを許可している場合，ベクタ割り込み要求 (VRQn : n = 1-6) が発生します。この信号はスタンバイ・モードの解除にも使用されます。

割り込み要求フラグと，割り込み許可フラグは，ビット操作命令，および4ビット・メモリ操作命令により操作します。ビット操作命令の場合は，MBEの設定にかかわらず常に直接操作できます。また，割り込み許可フラグはEI IE × × ×命令，DI IE × × ×命令で操作します。割り込み要求フラグのテストには通常SKTCLR命令を使用します。

例 EI        IE0        ; INT0許可  
 DI        IE1        ; INT1禁止  
 SKTCLR    IRQCSI    ; IRQCSIが1ならばスキップしてクリアする。

割り込み要求フラグを命令によりセットすると，割り込みが発生していなくても発生した場合と同様にベクタ割り込みが実行されます。

RESET信号発生により，割り込み要求フラグと割り込み許可フラグは“0”にクリアされ，すべての割り込みは禁止されます。

表 6 - 2 割り込み要求フラグのセット信号

割り込み 要求フラグ	割り込み要求フラグのセット信号	割り込み 許可フラグ
IRQBT	ベーシック・インターバル/ウォッチドッグ・タイマによる基準時間間隔信号でセット。	IEBT
IRQ4	INT4/P00端子入力信号の立ち上がり、立ち下がりどちらのエッジ検出によってもセット。	IE4
IRQ0	INT0/P10端子入力信号のエッジ検出によりセット。検出エッジはINT0エッジ検出モード・レジスタ (IM0) により選択。	IE0
IRQ1	INT1/P11端子入力信号のエッジ検出によりセット。検出エッジはINT1エッジ検出モード・レジスタ (IM1) により選択。	IE1
IRQCSI	シリアル・インタフェースのシリアル・データ転送動作終了信号によりセット。	IECSI
IRQT0	タイマ/イベント・カウンタ0からの一致信号でセット。	IET0
IRQT1	タイマ/イベント・カウンタ1からの一致信号でセット。	IET1

### (2) 割り込みプライオリティ選択レジスタ (IPS)

割り込みプライオリティ選択レジスタは、多重割り込みが可能な高位の割り込みを選択するレジスタで、下位3ビットにより指定します。

ビット3は、すべての割り込みを禁止するかどうかを指定する、割り込みマスタ許可フラグ (IME) となっています。

IPSは4ビット・メモリ操作命令で設定されますが、ビット3はEI/DI命令により、セット/リセットされます。

IPSの下位3ビットの内容を変更する場合は、割り込みが禁止されている (IME = 0) 状態で行う必要があります。

```

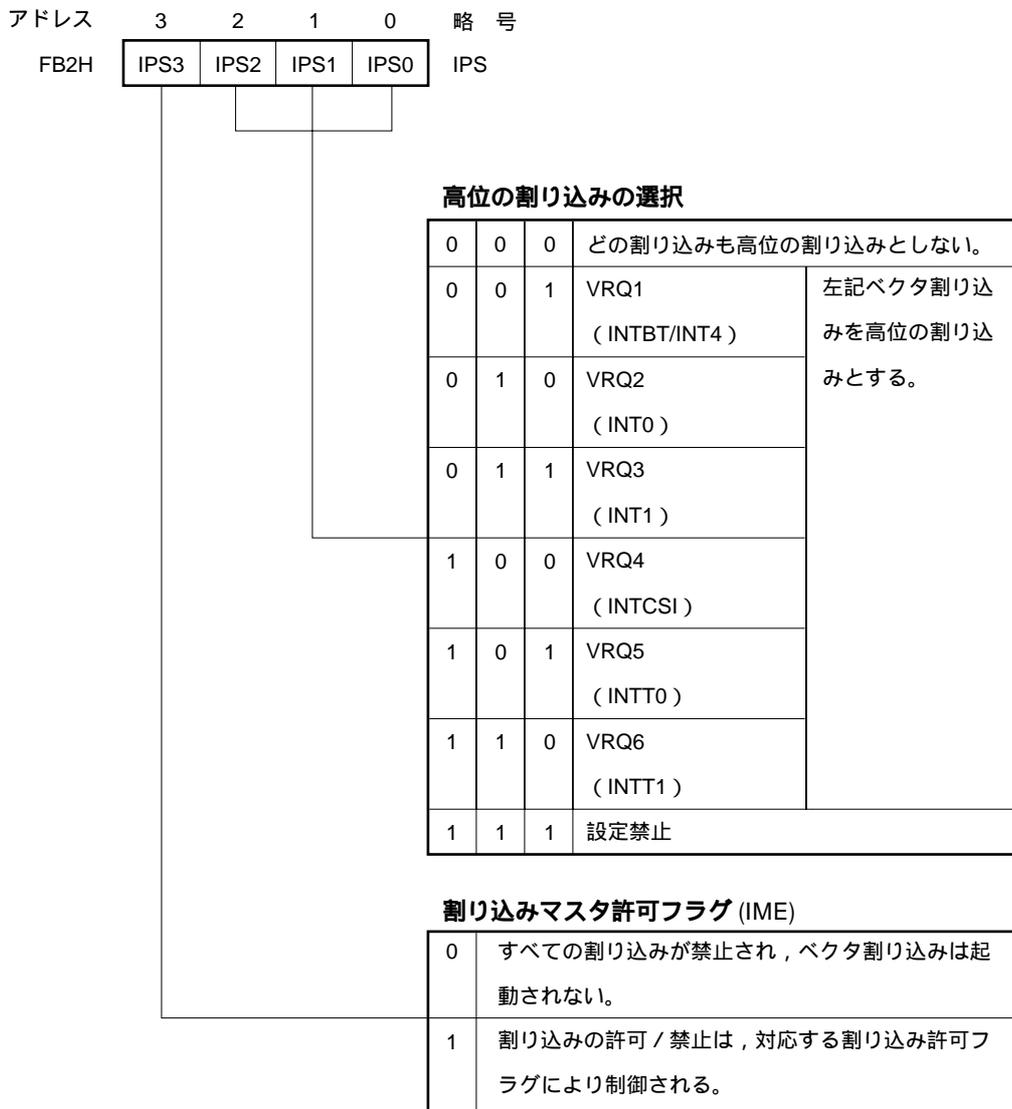
例  DI                      ; 割り込み禁止
    CLR1  MBE
    MOV   A, #1011B
    MOV   IPS, A            ; INT1を高位の割り込みにし、割り込みを許可する

```

RESET信号発生により全ビットが“0”にクリアされます。

**注意** IPSを設定するときは、必ず設定操作前に割り込みを禁止してから設定するようにしてください。

図6 - 3 割り込みプライオリティ選択レジスタ



## (3) INT0, INT1, INT4のハードウェア

(a) INT0の構成を図6-4(a)に示します。検出エッジを立ち上がりとするか、立ち下がりとするかを選択できる外部割り込み入力となっています。

また、INT0は、サンプリング・クロックによるノイズ除去機能を持っています(図6-5 ノイズ除去回路の入出力タイミング参照)。ノイズ除去回路によって、サンプリング・クロックの2周期分<sup>注</sup>より幅の狭いパルスはノイズとして除去されます。ただし、サンプリング・クロックの1周期分以上のパルスは、サンプリングを行うタイミングによっては、割り込み信号として受け付けられる場合もあります(図6-5 (a)参照)。サンプリング・クロックの2周期分以上のパルスは確実に割り込み信号として受け付けることができます。

INT0は、サンプリング・クロックとして  $f_x/64$  の2つを持っており、どちらかを選択して使用できるようになっています。その選択はINT0エッジ検出モード・レジスタ(IM0)のビット3(IM03)で行います(図6-6(a)参照)。

また、検出エッジの選択は、INT0エッジ検出モード・レジスタ(IM0)のビット0, 1(IM00, IM01)で行います。

IM0のフォーマットを図6-6(a)に示します。IM0は4ビット操作命令によって設定します。リセット信号の発生により全ビット“0”にクリアされ、立ち上がりエッジ指定となります。

注 サンプリング・クロックが のとき :  $2tcy$   
サンプリング・クロックが  $f_x/64$  のとき :  $128/f_x$

注意1. INT0/P10端子は、ポートとして入力する場合にも、ノイズ除去回路を通して入力されますので、サンプリング・クロックの2周期分以上のパルスを入力してください。

2. ノイズ除去回路を選択する(IM02 = 0に設定する)と、INT0はクロックによるサンプリングを行うため、スタンバイ・モード時は動作しません(ノイズ除去回路はCPUクロック が供給されていないときは動作しません)。したがって、スタンバイ・モード時にINT0によるスタンバイ・リリースを行う必要がある場合は、ノイズ除去回路を選択しないでください(IM02 = 1に設定してください)。

(b) INT1の構成を図6-4(b)に示します。検出エッジを立ち上がりとするか、立ち下がりとするかを選択できる外部割り込み入力となっています。

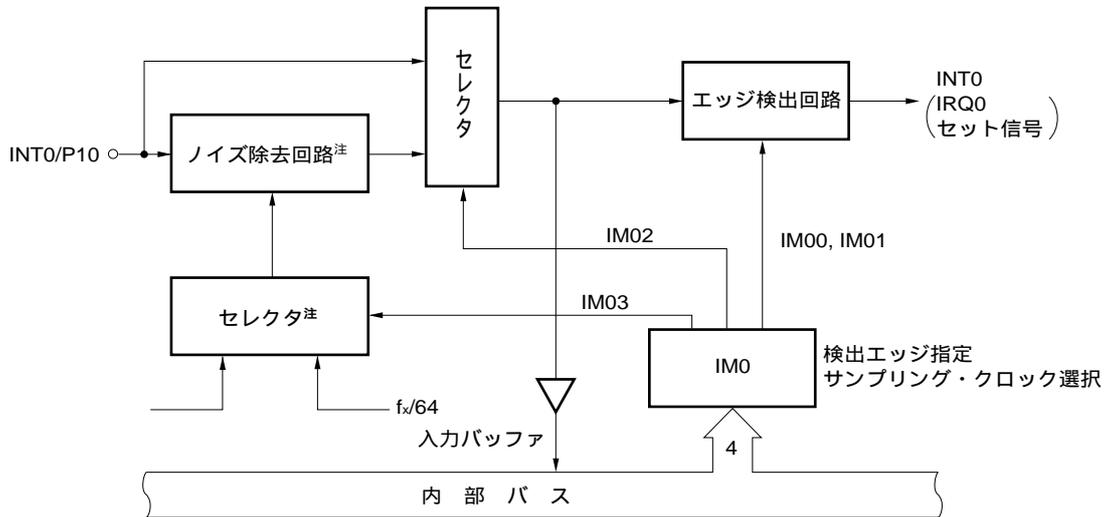
検出エッジの選択は、INT1エッジ検出モード・レジスタ(IM1)によって行います。

IM1のフォーマットを図6-6(b)に示します。IM1は4ビット操作命令によって設定します。リセット信号の発生により全ビット“0”にクリアされ、立ち上がりエッジ指定となります。

(c) INT4の構成を図6-4(c)に示します。立ち上がり、立ち下がりの両エッジを検出できる外部割り込み入力となっています。

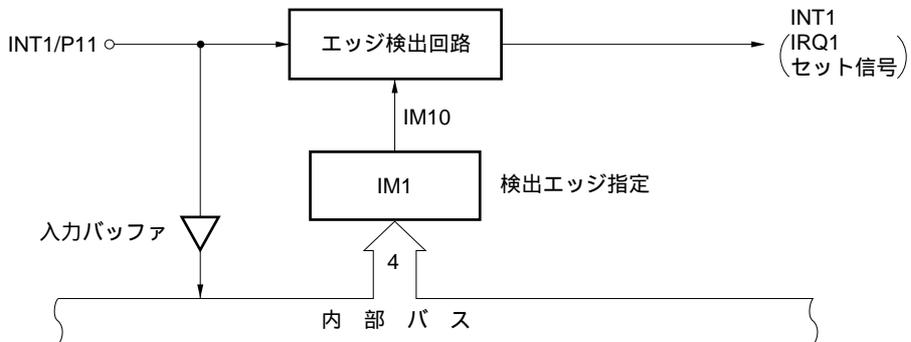
図6-4 INT0, INT1, INT4の構成

(a) INT0のハードウェア



注 fx/64を選択しても、INT0によるHALTモードの解除はできません。

(b) INT1のハードウェア



(c) INT4のハードウェア

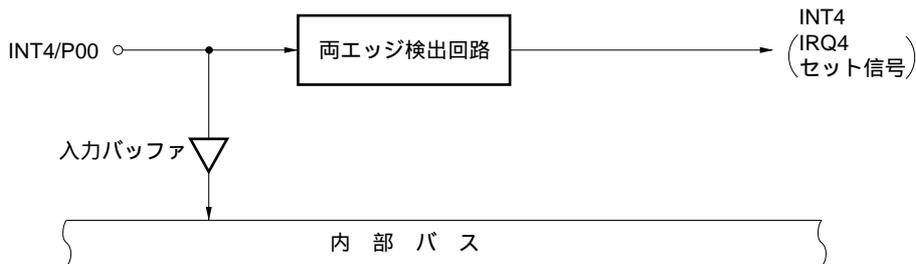
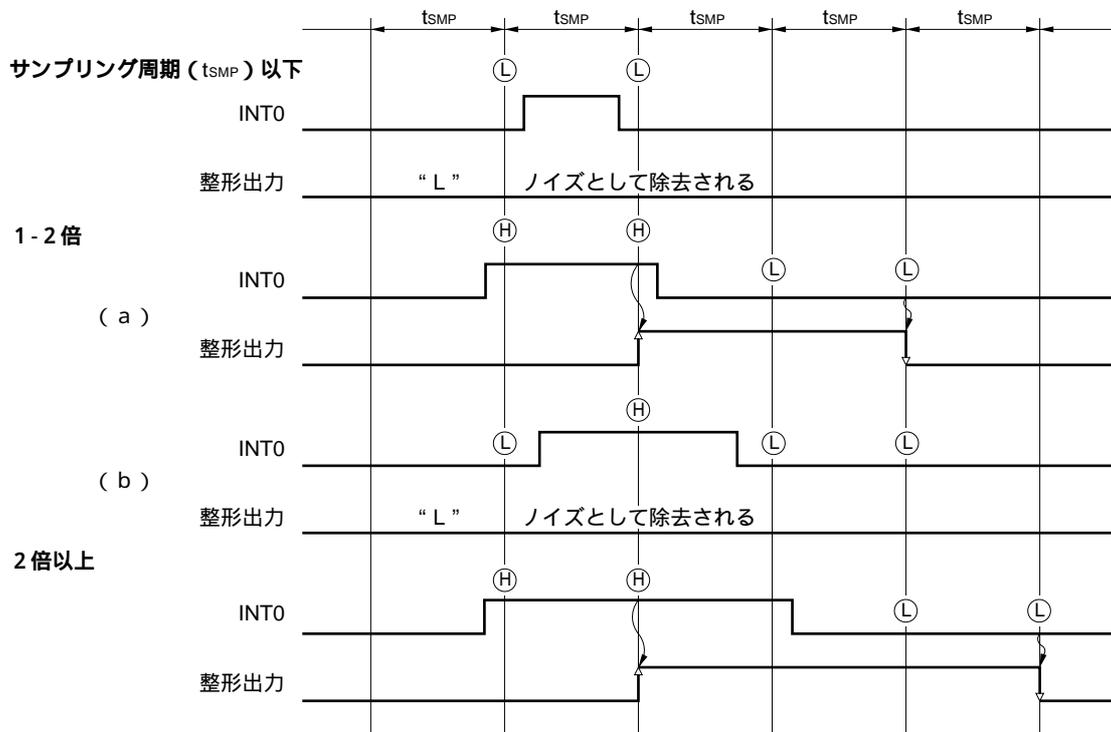


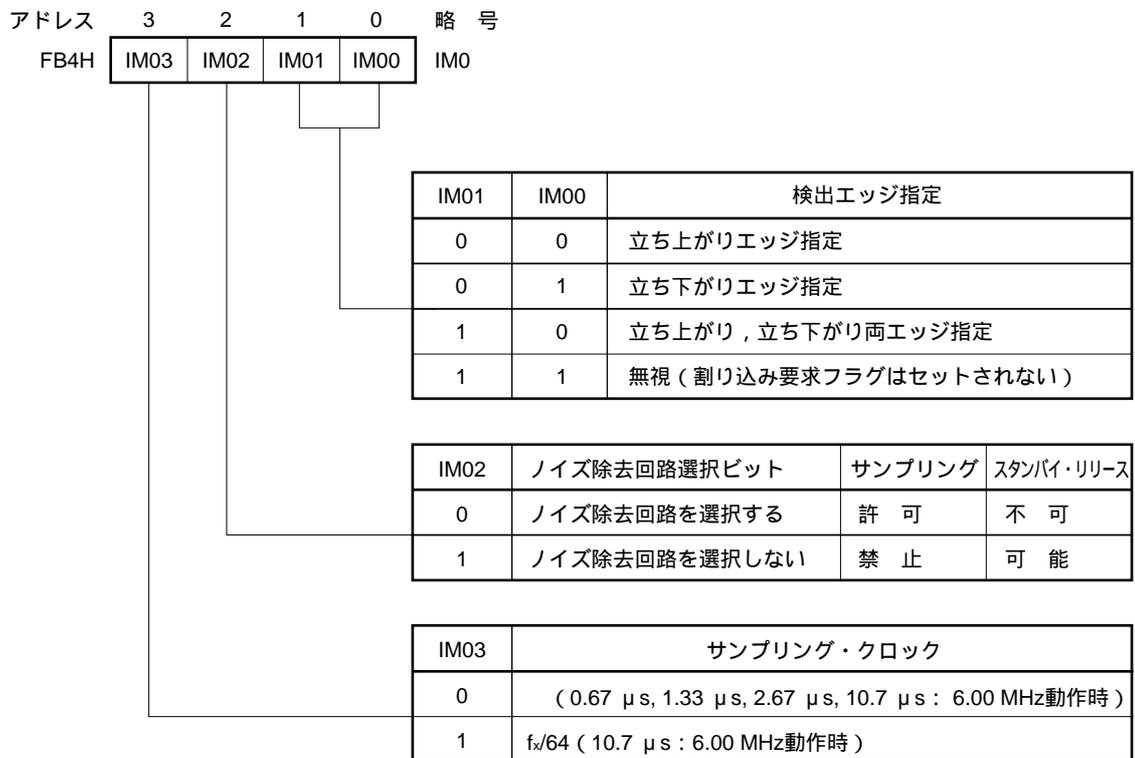
図6 - 5 ノイズ除去回路の入出力タイミング



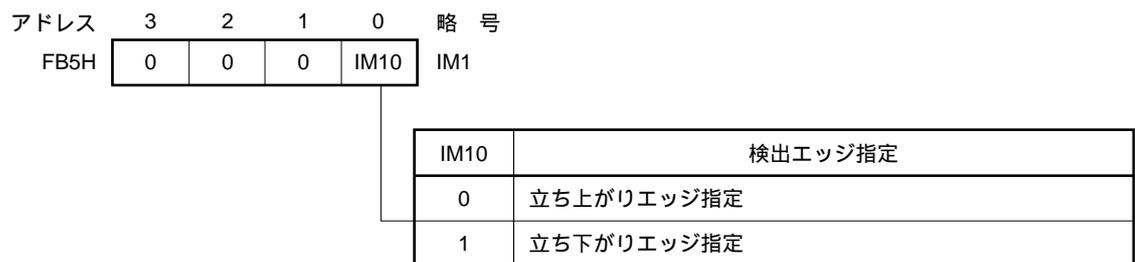
備考  $t_{SMP} = t_{CY}$ または $64/f_x$

図6-6 エッジ検出モード・レジスタのフォーマット

(a) INT0エッジ検出モード・レジスタ (IM0)



(b) INT1エッジ検出モード・レジスタ (IM1)



**注意** エッジ検出モード・レジスタを変更すると、割り込み要求フラグがセットされる場合がありますので、あらかじめ割り込みを禁止してモード・レジスタを変更し、CLR1命令によって割り込み要求フラグをクリアしてから、割り込みを許可してください。なお、IM0の変更で、サンプリング・クロックとして $f_x/64$ を選択した場合には、モード・レジスタ変更後16マシン・サイクル経過してから、割り込み要求フラグをクリアする必要があります。

(4) 割り込みステータス・フラグ

割り込みステータス・フラグ (IST0, IST1) はCPUが現在実行中の処理のステータスを示すフラグでPSWに含まれています。

割り込みプライオリティ・コントロール回路は、このフラグの内容によって表6-3に示されるように、多重割り込みの制御を行います。

IST0, IST1は4ビット操作命令またはビット操作命令により変更することができるため、実行中のステータスを変えて多重割り込みを行うこともできます。IST0, IST1をビット単位で操作する場合は、MBEの設定に関係なくいつでも行うことができます。

なお、IST0, IST1を操作する場合は、操作前に必ずDI命令を実行して割り込みを禁止し、操作後にEI命令を実行して割り込みを許可するようにしてください。

IST1, IST0は、割り込み受け付け時に他のPSWとともにスタック・メモリに退避されたのち、自動的に1つ上位のステータスに変更されます。RETI命令が実行されると、もとのIST1, IST0の値が復帰します。

$\overline{\text{RESET}}$ 信号の発生によりフラグの内容は“0”にクリアされます。

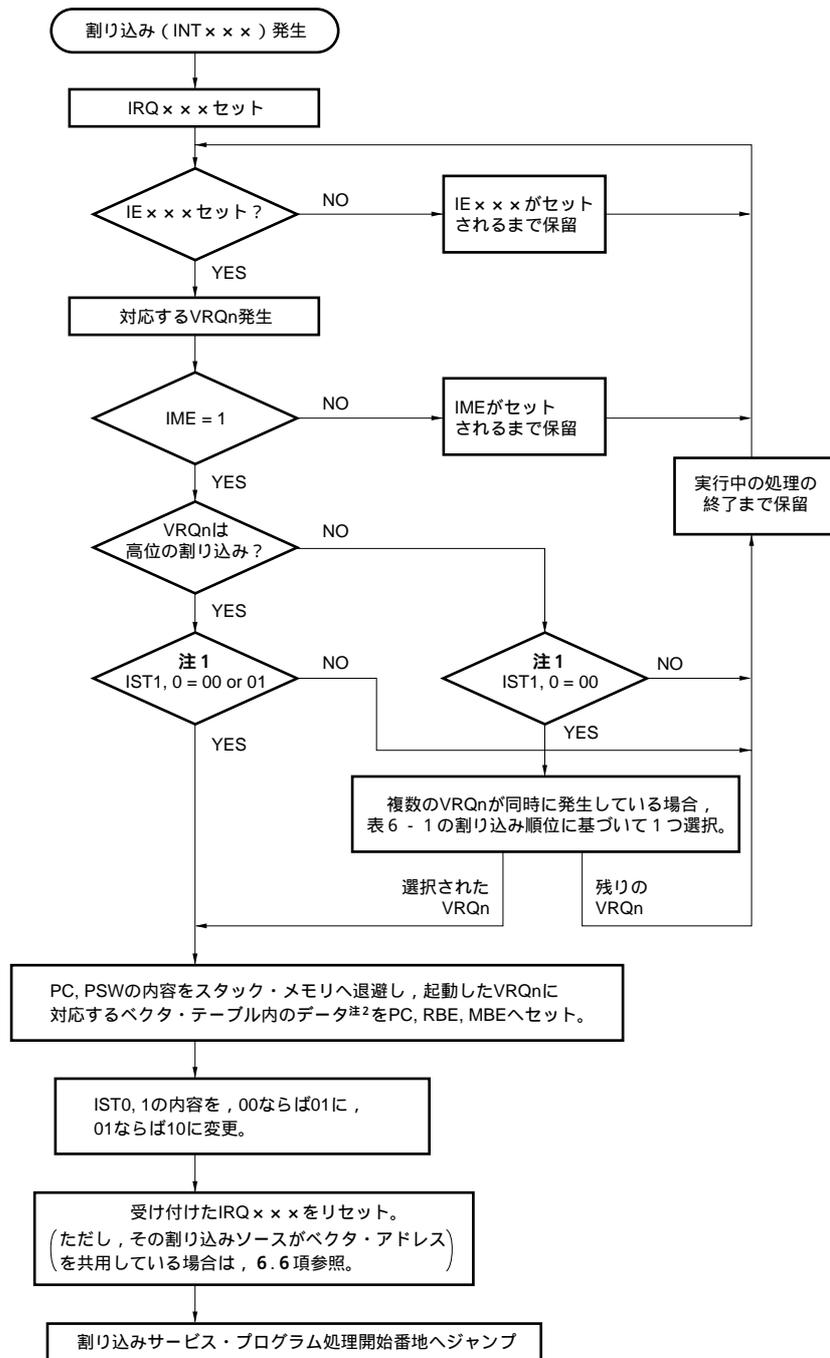
表6-3 IST1, IST0と割り込み処理状態

IST1	IST0	実行中の処理のステータス	CPUの処理内容	受け付け可能な割り込み要求	割り込み受け付け後	
					IST1	IST0
0	0	ステータス0	通常のプログラム処理中	すべての割り込みを受け付け可能	0	1
0	1	ステータス1	低位の割り込み、または高位の割り込み処理中	高位の割り込みのみ受け付け可能	1	0
1	0	ステータス2	高位の割り込み処理中	すべての割り込みの受け付け禁止	-	-
1	1	設定禁止				

## 6.4 割り込みシーケンス

割り込みが発生すると、以下の手順で処理されます。

図 6 - 7 割り込み処理シーケンス



注 1 . IST1, 0 : 割り込みステータス・フラグ (PSWのビット 3 , 2 ; 表 6 - 3 参照)

2 . 各ベクタ・テーブルには、割り込みサービス・プログラムの開始アドレスと、割り込み開始時のMBE, RBEの設定値をストアしておきます。

## 6.5 多重割り込み処理制御

μPD750068は、次の2つの方法により多重割り込みができます。

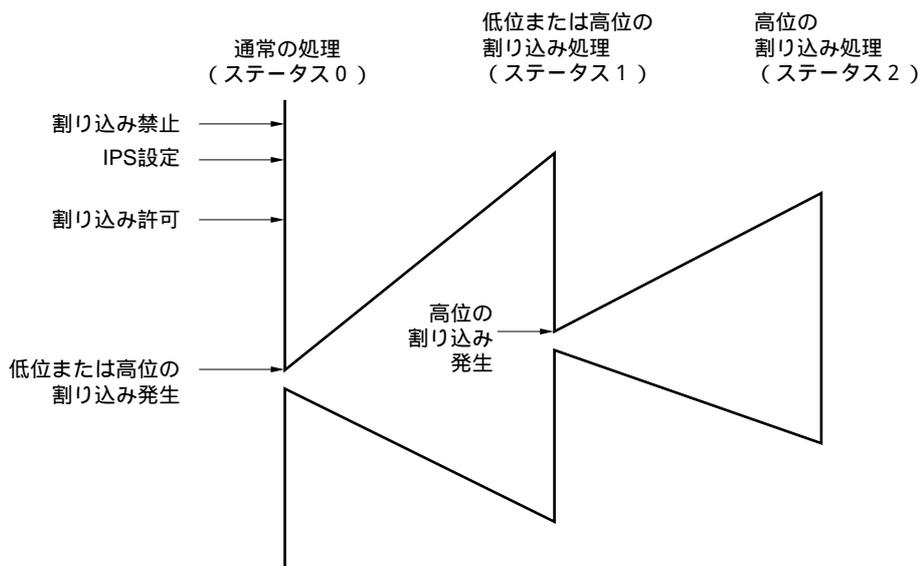
### (1) 高位の割り込みを指定する多重割り込み

この方法は、μPD750068の標準的な多重割り込み方法で、割り込みソースのうち1つを選び、その割り込みの多重割り込み（二重割り込み）を可能にするものです。

すなわち、割り込みプライオリティ選択レジスタ（IPS）によって指定された高位の割り込みは、実行中の処理のステータスが0および1のときに割り込み可能で、それ以外の割り込み（低位の割り込み）はステータス0のときのみ割り込み可能です（図6-8、表6-3参照）。

したがって、使用する割り込みのうち1つだけを多重割り込みできるようにしたい場合には、この方法を使えば、割り込み処理中での割り込み許可、禁止などの操作を行わなくても二重割り込みを実現し、かつ、ネスティング・レベルを2レベルに抑えることができます。

図6-8 高位の割り込みによる多重割り込み



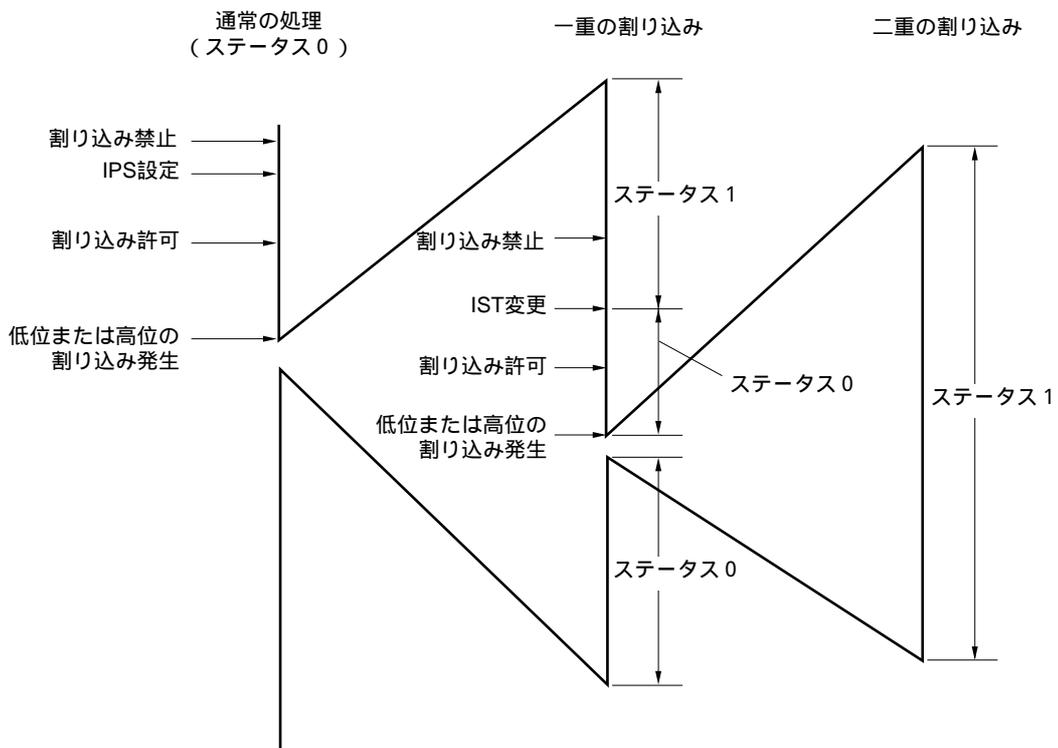
(2) 割り込みステータス・フラグを変更する多重割り込み

割り込みステータス・フラグをプログラムにより変更すれば、多重割り込み可能になります。すなわち、割り込み処理プログラムでIST1, IST0を“0, 0”に変更し、ステータス0にすれば多重割り込みできます。

この方法は2つ以上の割り込みの多重割り込みを許可したい場合、あるいは三重以上の多重割り込みを行いたい場合に使用します。

IST1, IST0の変更は、あらかじめDI命令により割り込みを禁止した状態で行います。

図6 - 9 割り込みステータス・フラグの変更による多重割り込み



## 6.6 ベクタ・アドレス共用割り込み処理

INTBTとINT4の各割り込みソースはベクタ・テーブルを共用しているため、以下に示すように割り込みソースの選択を行います。

### (1) 一方の割り込みのみ使用する場合

ベクタ・テーブルを共用する2種類の割り込みソースのうち、必要とする割り込みソースの割り込み許可フラグを“1”にセット、もう一方の割り込み許可フラグを“0”にクリアします。この場合、許可した ( $IE \times \times \times = 1$ ) 方の割り込みソースにより割り込み要求を発生し、受け付けられるとその割り込み要求フラグはリセットされます。

### (2) 両方の割り込みを使用する場合

2種類の割り込みソースに対応する割り込み許可フラグを共に“1”にセットします。この場合、2種類の割り込みソースの割り込み要求フラグの論理和が割り込み要求となります。

この場合、一方あるいは両方の割り込み要求フラグのセットによる割り込み要求が受け付けられてもどちらの割り込み要求フラグもリセットされません。

したがって、この場合は割り込みサービス・ルーチンでどちらの割り込みソースによる割り込みかを判断する必要があります。それは、割り込みサービス・ルーチンの最初にSKTCLR命令で割り込み要求フラグをチェックすることにより行います。

この割り込み要求フラグのテスト&クリアを行ったとき、両方の要求フラグがセットされていると、一方の要求フラグがクリアされても割り込み要求は残ります。この割り込みが「高位の割り込み」として選択されていた場合には、残った割り込み要求により二重割り込み処理に入ります。

すなわち、テストしない方の割り込み要求が先に処理されることとなります。一方、「低位の割り込み」の場合には、残った割り込みは保留されますから、テストした方の割り込み要求が先に処理されます。したがって、表6-4のように、「高位の割り込み」であるか否かによって共用割り込み判別方法は異なります。

表6-4 共用割り込みの判別

「高位の割り込み」の場合	割り込みを禁止し、優先したい割り込み要求フラグをテストする。
「低位の割り込み」の場合	優先したい割り込みソースの割り込み要求フラグをテストする。

例1 . INTBTとINT4の両方を「高位の割り込み」として使用し，INT4を優先する場合。

```

DI
SKTCLR   IRQ4   ;   IRQ4 = 1 ?
BR       VSUBBT
      ...
EI
RETI
      ...

VSUBBT : CLR1   IRQBT
      ...
EI
RETI
    
```

} INT4の処理  
ルーチン

} INTBTの処理  
ルーチン

2 . INTBTとINT4の両方を「低位の割り込み」として使用しINT4を優先する場合。

```

SKTCLR   IRQ4   ;   IRQ4 = 1 ?
BR       VSUBBT
      ...
RETI
      ...

VSUBBT : CLR1   IRQBT
      ...
RETI
    
```

} INT4の処理  
ルーチン

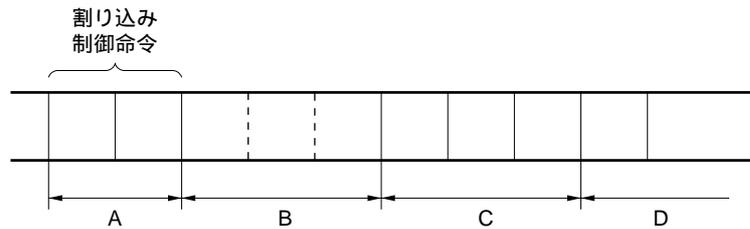
} INTBTの処理  
ルーチン

## 6.7 割り込み処理までのマシン・サイクル

μPD750068では、割り込み要求フラグ（IRQ×××）がセットされてから割り込みルーチンのプログラムを実行するまでのマシン・サイクルは、次のようになります。

### （1）割り込み制御命令実行中にIRQ×××がセットされた場合

割り込み制御命令実行中にIRQ×××がセットされた場合には、続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。



A : IRQ×××セット

B : 続く1命令を実行（1-3マシン・サイクル；命令により異なる）

C : 割り込み処理（3マシン・サイクル）

D : 割り込みルーチンの実行

**注意1**．割り込み制御命令が連続している場合には、最後に実行した割り込み制御命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。

**2**．IRQ×××がセットされるタイミング（上図のA）または、そのあとのタイミングでDI命令を実行すると、セットされたIRQ×××の割り込み要求は次にEI命令を実行するまで保留されます。

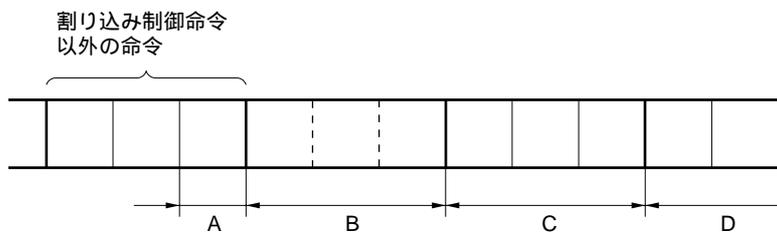
**備考1**．割り込み制御命令とは、割り込み関係のハードウェア（データ・メモリのFB×H番地）を操作する命令です。この命令には、DI命令、EI命令があります。

**2**．割り込み処理の3マシン・サイクルは、割り込み受け付け時に操作されるスタックの操作時間などです。

(2)(1) 以外の命令実行中にIRQ $\times\times\times$ がセットされた場合

(a) 実行中の命令の最終マシン・サイクルでIRQ $\times\times\times$ がセットされた場合

この場合は、実行中の命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。

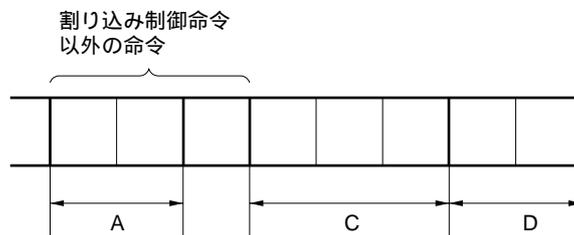


- A : IRQ $\times\times\times$ セット
- B : 続く1命令を実行 (1-3マシン・サイクル; 命令により異なる)
- C : 割り込み処理 (3マシン・サイクル)
- D : 割り込みルーチンの実行

**注意** 続く命令が割り込み制御命令の場合には、最後に実行した割り込み制御命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。またIRQ $\times\times\times$ がセットされたあとに実行される割り込み制御命令がDI命令の場合には、セットされたIRQ $\times\times\times$ の割り込み要求は保留されません。

(b) 実行中の命令の最終マシン・サイクル以前でIRQ $\times\times\times$ がセットされた場合

この場合は、実行中の命令ののち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。



- A : IRQnセット
- C : 割り込み処理 (3マシン・サイクル)
- D : 割り込みルーチンの実行

## 6.8 割り込みの効果的な使い方

割り込み機能は、次のように使用すると効果的です。

### (1) 割り込み処理ルーチンはMBE = 0とする

割り込み処理ルーチン中で使用するデータ・メモリは優先的に00H-7FH番地に割り付け、割り込みベクタ・テーブルでMBE=0を指定すれば、メモリ・バンクを意識しないでプログラムすることができます。

プログラムの都合でやむを得ず、メモリ・バンク1を使用しなければならない場合は、PUSH BS命令によってメモリ・バンク選択レジスタを退避して、メモリ・バンク1を選択します。

### (2) 通常ルーチンと割り込みルーチンでレジスタ・バンクを使い分ける

通常ルーチンではRBE = 1, RBS = 2としてレジスタ・バンク2, 3を使用します。一重の割り込み処理ルーチンではRBE = 0として、レジスタ・バンク0を使用することによってレジスタの退避/復帰は一切必要なくなります。二重の割り込み処理ではRBE = 1として、PUSH BS命令によりレジスタ・バンクを退避し、RBS = 1としてレジスタ・バンク1を使用します。

### (3) ソフトウェア割り込みをディバグに使う

割り込み要求フラグを命令によってセットしても、割り込みが発生したときと同じ動作をします。不特定の割り込みのディバグや割り込みが同時に発生した場合のディバグなどは、命令によって、割り込み要求フラグをセットすることによって効率的に行うことができます。

## 6.9 割り込みの応用

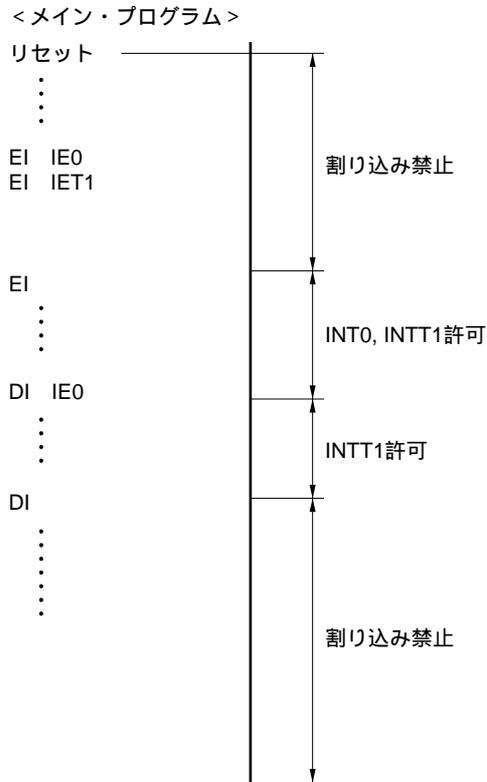
割り込み機能を使用する場合は、まず、メイン・プログラムで次のように設定します。

- (a) 使用する割り込み許可フラグをセット (EI IE × × × 命令)
- (b) INT0, INT1を使用する場合は、アクティブ・エッジの選択 (IM0, IM1セット)。
- (c) 二重割り込み (高位の割り込みによる) を使用する場合は、IPSを設定 (同時にIMEセットもできる)。
- (d) 割り込みマスタ許可フラグをセット (EI命令)。

割り込み処理プログラムでは、ベクタ・テーブルによってMBE, RBEが設定されます。ただし、「高位の割り込み」に指定された割り込みでは、レジスタ・バンクの退避と設定が必要です。

割り込み処理プログラムからの復帰はRETI命令を使用します。

(1) 割り込みの許可・禁止



$\overline{\text{RESET}}$ 信号により，すべての割り込みは禁止。

EI IE x x x 命令により，割り込み許可フラグをセット。

この段階ではまだすべての割り込みは禁止

EI命令により，割り込みマスタ許可フラグをセット。

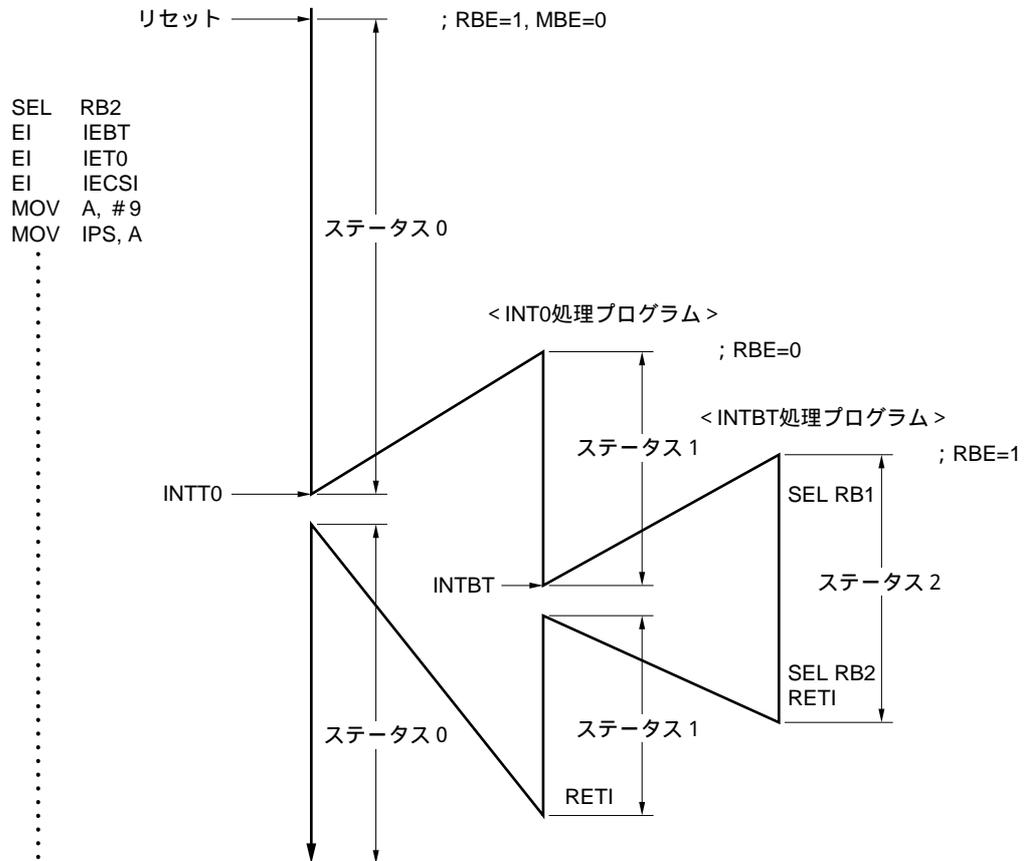
この段階で，INT0, INTT1が許可される。

DI IE x x x 命令により，割り込み許可フラグをクリア，INT0が禁止される。

DI命令によりすべての割り込みが禁止。



(3) 「高位の割り込み」による多重割り込み (INTBTを高位, INTT0, INTCSIを低位の割り込み)



IPSの設定により、INTBTを「高位の割り込み」とし、同時に割り込み許可。

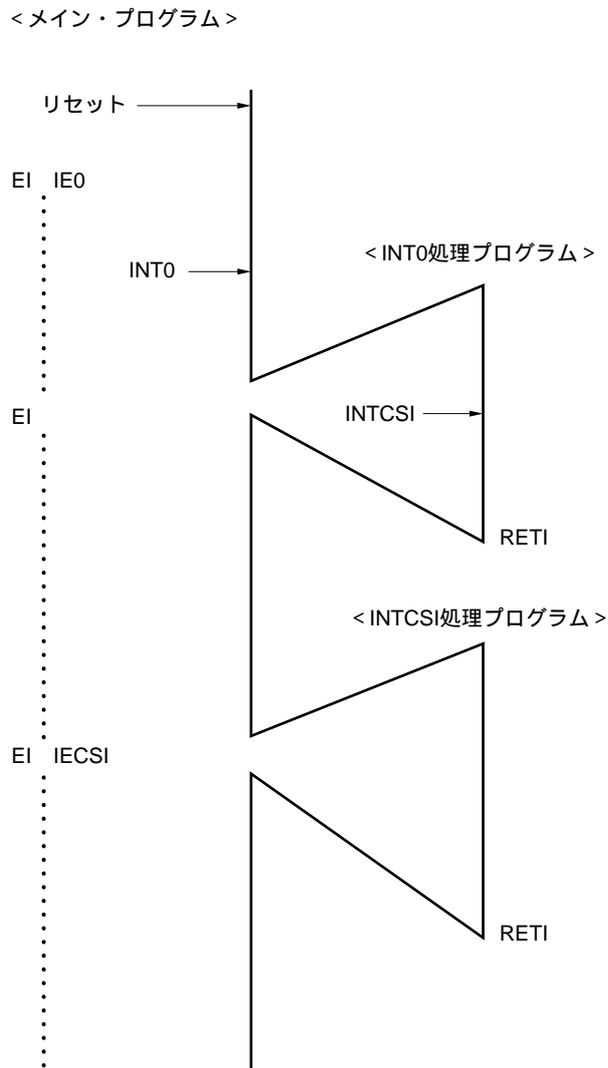
低位の割り込みINTT0発生により、INTT0処理プログラム開始、ステータス1となり低位の割り込みは禁止。RBE = 0としてレジスタ・バンク0を使用。

高位の割り込みINTBT発生により、二重割り込み実行。ステータスとなりすべての割り込み禁止。

RBE = 1, RBS = 1としてレジスタ・バンク1を使用 (使用するレジスタだけをPUSH命令で退避してもよい)。

RBSを2に戻し、リターン。ステータスは1に戻る。

(4) 保留割り込みの実行 割り込み禁止中の割り込み入力



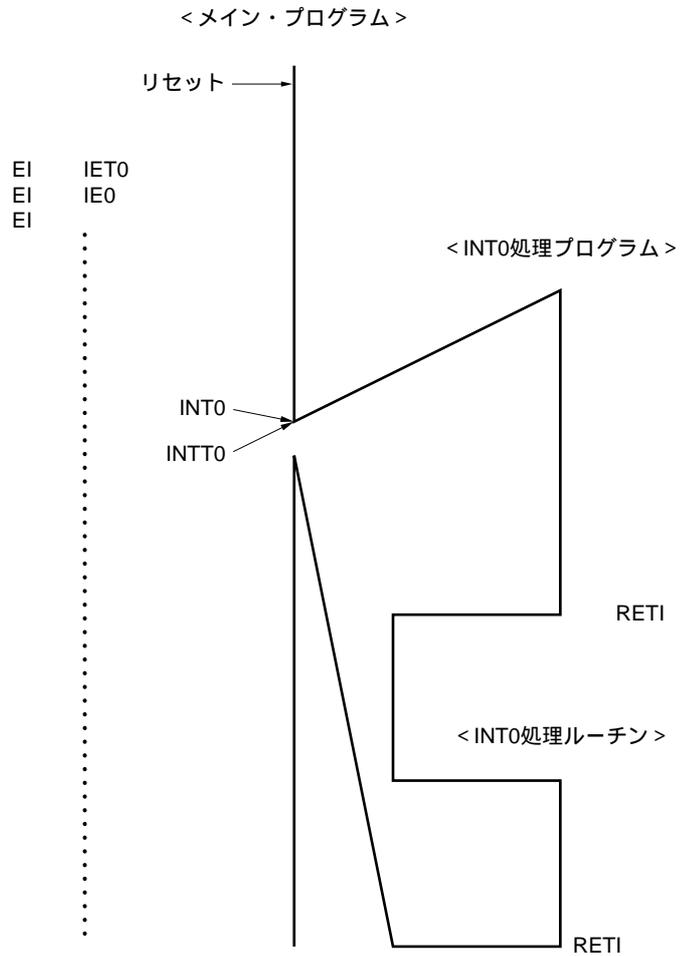
割り込み禁止中にINT0がセットされても要求フラグは保留される。

EI命令で割り込みが許可された時点で、INT0処理プログラム・スタート。

と同様

保留されていたINTCSIが許可された時点で、INTCSI処理プログラム・スタート。

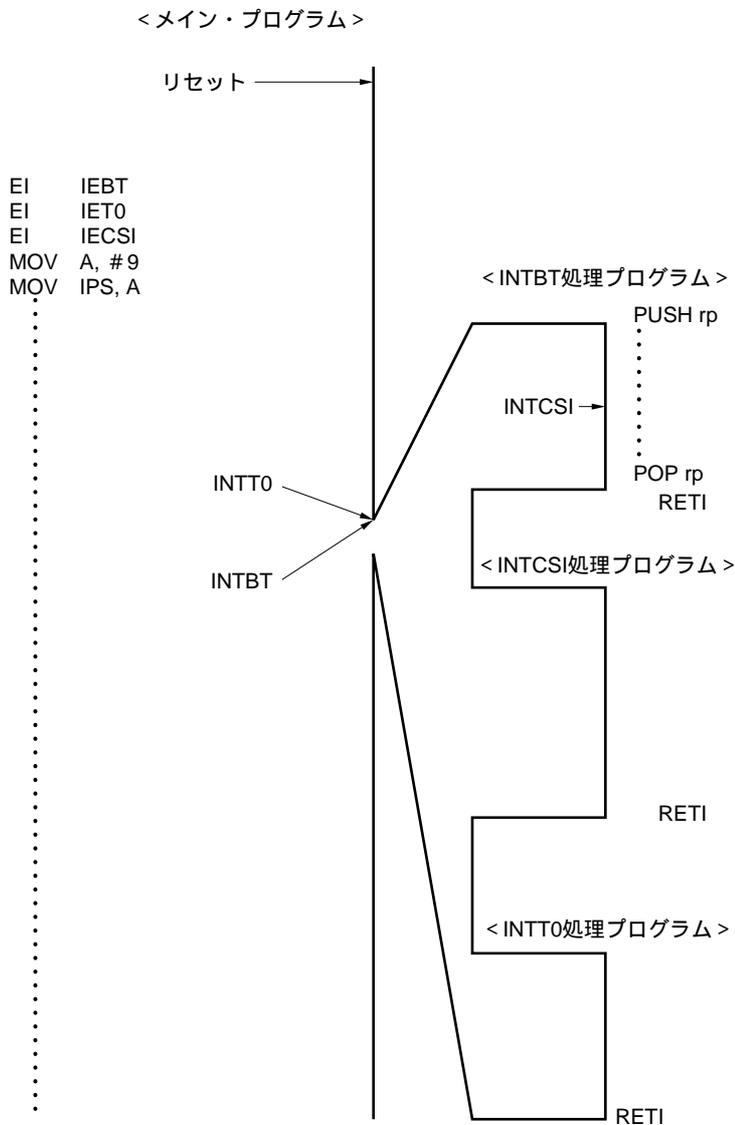
(5) 保留割り込みの実行 低位の割り込みが同時に2つ発生



「低位の割り込み」INT0とINTT0が同時（同一命令実行中）に発生した場合は、割り込み順位の高いINT0を先に実行（INTT0は保留される）。

RETI命令により、INT0処理プログラムが終わると保留されていたINTT0処理プログラムをスタート。

(6) 保留割り込みの実行 割り込み処理中の割り込み発生 (INTBTを高位, INTT0, INTCSIを低位の割り込み)



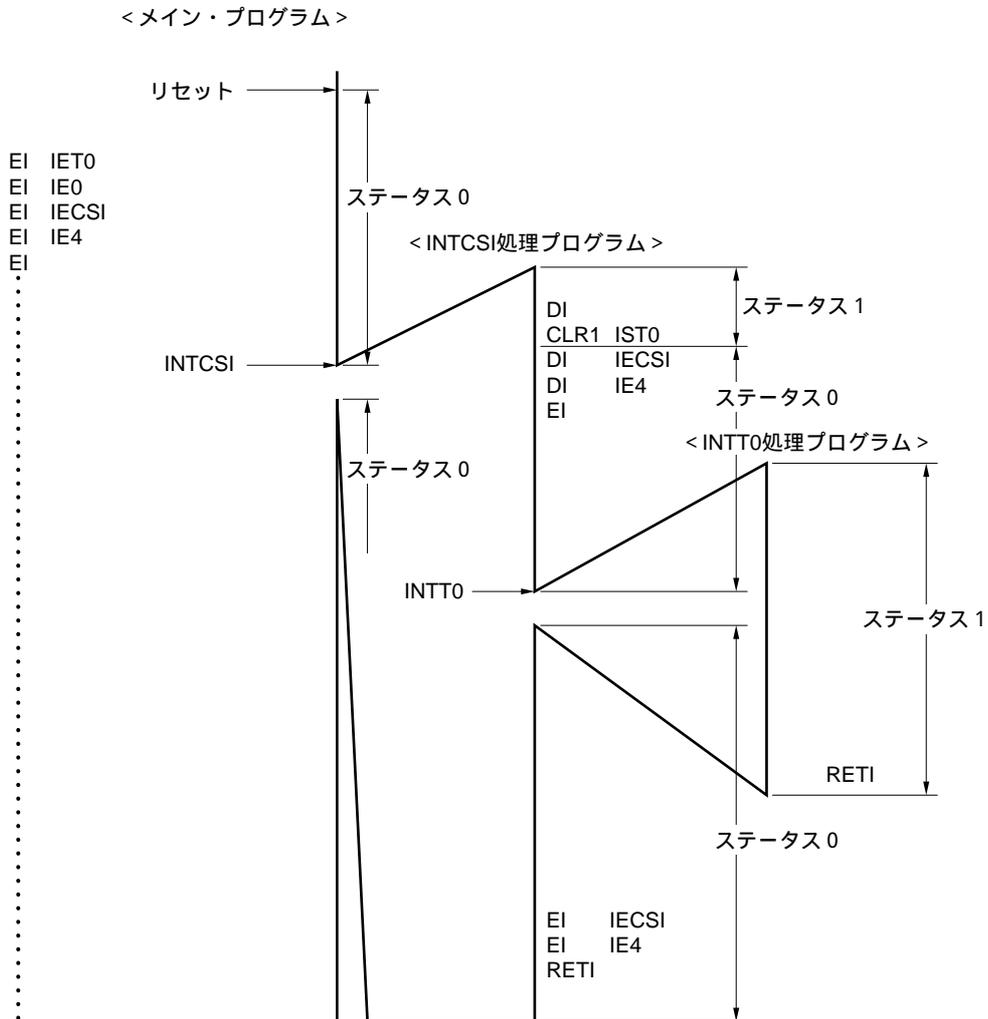
高位の割り込みINTBTと低位の割り込みINTT0が同時に発生すると、高位の割り込み処理を開始（高位の割り込み処理中に、高位の割り込みが発生するおそれがないことが確実ならDI IE × ×は不要）。

高位の割り込み実行中に低位の割り込みが発生すると、割り込みは保留される。

高位の割り込みを終了すると、保留されていた低位の割り込みのうち割り込み順位の高いINTCSI処理が実行される。

INTCSI処理が終了すると保留されていたINTT0が処理される。

(7) 二重割り込みを2つ許可する。 INTT0とINT0は二重割り込みを許可。 INTCSI, INT4は一重割り込み



二重割り込みを許可しない割り込みINTCSIの発生によりINTCSI処理プログラムをスタート。ステータスは1となる。

IST0のクリアにより、ステータスを0にする。二重割り込みを許可しないINTCSIとINT4を禁止。

二重割り込みを許可するINTT0発生により、二重割り込み実行、ステータス1となり、すべての割り込みは禁止される。

INTT0処理終了により、ステータスは0に戻る。

禁止していたINTCSI, INT4を許可して復帰。

## 6.10 テスト機能

### 6.10.1 テスト・ソースの種類

μPD750068には、2種類のテスト・ソースがあります。テスト・ソースのうちINT2は、2種類のエッジ検出テストブル入力を備えています。

表6-5 テスト・ソースの種類

テスト・ソース	内/外
INT2 (INT2端子への入力の立ち上がりエッジ検出またはKR0-KR3への入力のいずれかの立ち下がり検出)	外
INTW (時計用タイマからの信号)	内

### 6.10.2 テスト機能を制御する各種ハードウェア

#### (1) テスト要求フラグ, テスト許可フラグ

テスト要求フラグ (IRQ $\times\times\times$ ) は、テスト要求発生 (INT $\times\times\times$ ) で“1”にセットされます。テスト処理が実行されたら、ソフトウェアで“0”にクリアしてください。

テスト許可フラグ (IE $\times\times\times$ ) は、各テスト要求フラグに対応して個別に備わっており、内容が“1”のときスタンバイ・リリース信号を許可し、“0”のとき禁止します。

テスト要求フラグ, テスト許可フラグがともに“1”にセットされている場合、スタンバイ・リリース信号を発生します。

表6-6にテスト要求フラグのセット信号を示します。

表6-6 テスト要求フラグのセット信号

テスト要求フラグ	テスト要求フラグのセット信号	テスト許可フラグ
IRQW	時計用タイマからの信号でセット。	IEW
IRQ2	INT2/P12端子入力信号の立ち上がりエッジ検出またはKR0/P60-KR3/P63端子への入力のいずれかの立ち下がりエッジ検出によりセット。検出エッジはINT2エッジ検出モード・レジスタ (IM2) により選択。	IE2

**(2) INT2, キー割り込み (KR0-KR3) のハードウェア**

INT2, KR0-KR3の構成を図6 - 10に示します。

次に示す2系統の端子でのエッジ検出により, IRQ2セット信号が出力されます。いずれの端子を選択するかは, INT2エッジ検出モード・レジスタ (IM2) で行います。

**(a) INT2端子入力の立ち上がりエッジ検出**

INT2端子入力の立ち上がりエッジを検出するとIRQ2がセットされます。

**(b) KR0-KR3端子入力のいずれかの立ち下がりエッジ検出 (キー割り込み)**

KR0-KR3のうち, 割り込み入力に使用する端子をINT2エッジ検出モード・レジスタ (IM2) によって選択します。選択された端子の入力のいずれかの立ち下がりエッジを検出するとIRQ2がセットされます。

IM2のフォーマットを図6 - 11に示します。IM2は4ビット操作命令によって設定します。リセット信号の発生により全ビット“0”にクリアされ, INT2の立ち上がりエッジ指定となります。

図 6 - 10 INT2, KR0-KR3のブロック図

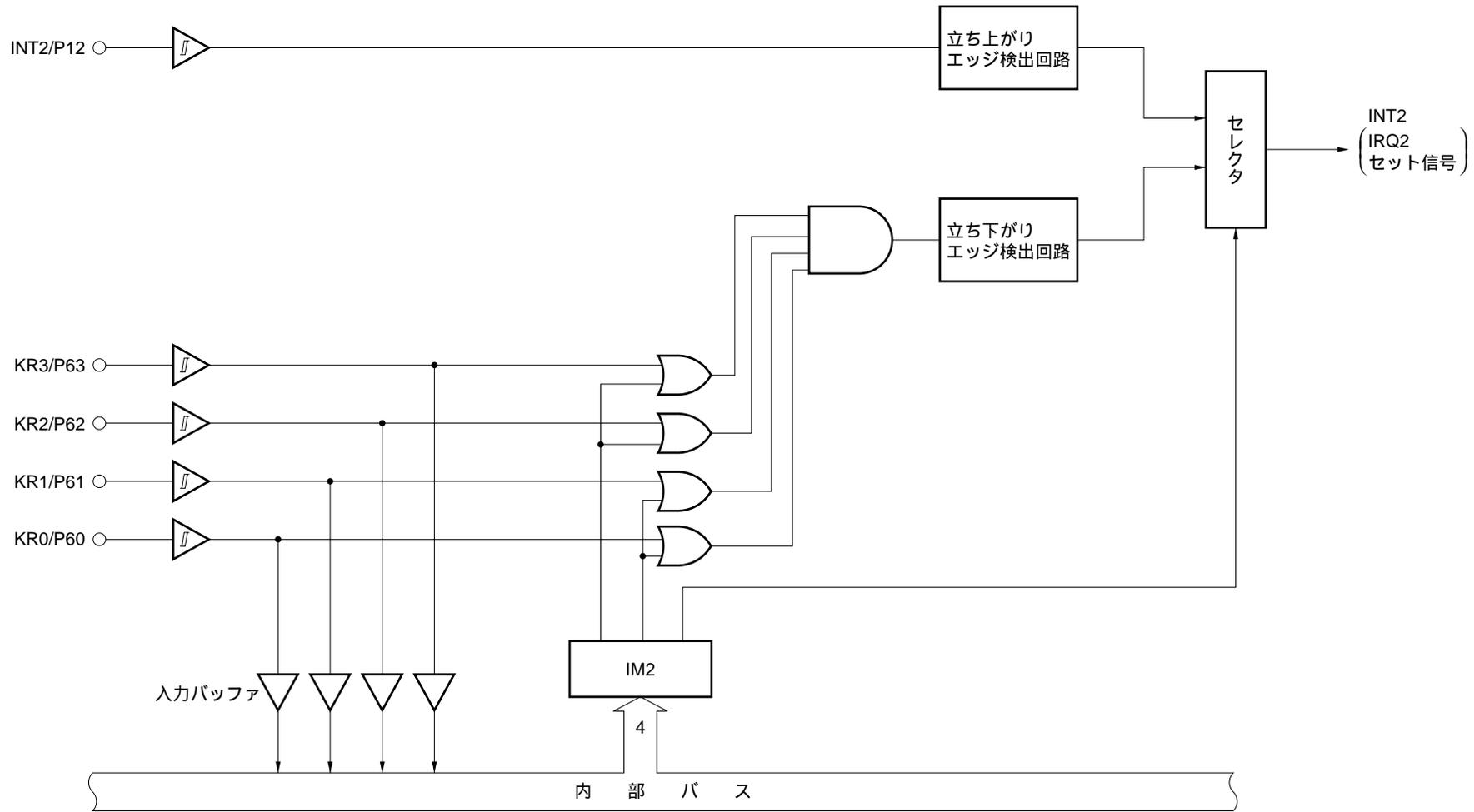


図6 - 11 INT2エッジ検出モード・レジスタ (IM2) のフォーマット



注意1 . エッジ検出モード・レジスタを変更すると、テスト要求フラグがセットされる場合がありますので、あらかじめテスト入力を禁止してモード・レジスタを変更し、CLR1命令によってテスト要求フラグをクリアしてから、テスト入力を許可してください。

2 . 立ち下がりエッジ検出として選択した端子のうち1本でもロウ・レベルが入力されていると、ほかの端子に立ち下がりエッジが入力されてもIRQ2はセットされません。

(メモ)

## 第7章 スタンバイ機能

μPD750068は、システムの消費電力を低減させることのできるスタンバイ機能を持っています。スタンバイ機能には次のような2つのモードがあります。

STOPモード

HALTモード

STOPモード、HALTモードはそれぞれ以下に示すような機能を持っています。

### (1) STOPモード

メイン・システム・クロック発振回路を停止させ、システム全体が停止するモードです。CPUの消費電流は、かなり低減されます。

また、データ・メモリの低電圧 ( $V_{DD} = 1.8\text{ V}$ まで) 保持が可能です。したがって、超低消費電流で、データ・メモリの内容を保持する場合に有効です。

μPD750068のSTOPモードは、割り込み要求によって解除できるため、間欠動作も可能です。しかし、STOPモードの解除時に発振安定時間確保のためのウェイト時間がとられるため、割り込み要求によって、すぐ処理を開始しなければならない場合にはHALTモードを選択してください。

### (2) HALTモード

CPUの動作クロックを停止させるモードです。システム・クロック発振回路の発振は続きます。このモードでは、STOPモードほどの消費電流の低減はできませんが、割り込み要求により、すぐに処理を再開したい場合や、時計動作のような間欠動作をさせたい場合に有効です。

いずれのモードでも、スタンバイ・モードに設定される直前のレジスタ、フラグ、データ・メモリの内容はすべて保持されます。また、入出力ポートの出力ラッチ、出力バッファの状態も保持されますから、入出力ポートの状態を、あらかじめ、システム全体の消費電流が最も小さくなるように処理しておきます。

**使用上の注意を次ページに示します。**

- 注意1 . STOPモードは、メイン・システム・クロックで動作しているときだけ使用できます（サブシステム・クロックの発振を停止させることはできません）。HALTモードは、メイン・システム・クロック、サブシステム・クロックのどちらの動作状態でも使用できます。
- 2 . 時計用タイマの動作クロックがメイン・システム・クロック $f_X$ のとき、STOPモードに設定するとその動作が停止します。
- したがって、動作を継続する場合はSTOPモードを設定する前に動作クロックをサブシステム・クロック $f_{XT}$ に切り替える必要があります。
- 3 . スタンバイ・モードとCPUクロック、システム・クロックを切り替えることで効率の良い低消費電流、低電圧動作が可能ですが、いずれもコントロール・レジスタを操作して新しいクロックを選択してから切り替え後のクロックで動作を開始するまでに、5.2.3 システム・クロックとCPUクロックの設定で述べたような時間が必要です。このため、クロック切り替え機能とスタンバイ・モードを組み合わせる場合には、切り替わりに必要な時間経過後、スタンバイ・モードに設定してください。
- 4 . スタンバイ・モードを使用する場合には、入出力ポートは消費電流が最小となるよう処理してください。
- 特に、入力ポートはオープンにしないでください。必ずロウ・レベルかハイ・レベルを入力してください。

## 7.1 スタンバイ・モードの設定および動作状態

表7-1 スタンバイ・モード時の各動作状態

		STOPモード	HALTモード
設定命令		STOP命令	HALT命令
設定時のシステム・クロック		メイン・システム・クロックの場合のみ設定可	メイン・システム・クロックとサブシステム・クロックのいずれでも設定可
動作状態	クロック発生回路	メイン・システム・クロック発振停止	CPUクロックのみ停止（発振継続）
	ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ	動作停止	メイン・システム・クロック発振時のみ動作（基準時間間隔でIRQBTをセット）
	シリアル・インタフェース	シリアル・クロックに外部SCK入力を選択した場合のみ、動作可能	シリアル・クロックに外部SCK入力を選択した場合、またはメイン・システム・クロック発振時のみ動作
	タイマ/イベント・カウンタ	カウント・クロックにTI0, TI1の端子入力またはf <sub>XT</sub> を選択した時計用タイマを指定した場合のみ、動作可能	カウント・クロックにTI0, TI1の端子入力またはf <sub>XT</sub> を選択した時計用タイマを指定した場合、またはメイン・システム・クロック発振時のみ動作
	時計用タイマ	カウント・クロックにf <sub>XT</sub> を選択した場合動作可能	動作可能
	A/Dコンバータ	動作停止	メイン・システム・クロック発振時のみ動作可能
	外部割り込み	INT1, 2, 4は動作可能 INT0のみ動作不可能 <sup>注</sup>	
	CPU	動作停止	
解除信号		割り込み許可フラグで許可されている動作可能なハードウェアからの割り込み要求信号または $\overline{\text{RESET}}$ 信号発生	

注 エッジ検出モード・レジスタ（IM0）のビット2により、ノイズ除去回路を選択しない場合（IM02 = 1 のとき）のみ動作可能。

STOPモードにはSTOP命令で、HALTモードにはHALT命令で設定します（STOP命令、HALT命令は、それぞれPCCのビット3、ビット2をセットする命令です）。

STOP命令、HALT命令の次には必ずNOP命令を書くようにしてください。

PCCの下位2ビットによってCPU動作クロックを変更する場合、表5-5 システム・クロック、CPUクロックの切り替えに要する最大時間に示すようにPCCを書き換えてから、CPUクロックが変更されるまでには、時間的なずれを生じる場合があります。したがって、スタンバイ・モード前の動作クロックとスタンバイ・モード解除後のCPUクロックを変更するような場合には、PCCを書き換えてからCPUクロックの変更に必要なマシン・サイクル経過後にスタンバイ・モードを設定します。

スタンバイ・モードでは、汎用レジスタ、フラグ類、モード・レジスタ類、出力ラッチなど、スタンバイ・モード中動作を停止しているすべてのレジスタとデータ・メモリのデータが保持されます。

**注意 1** . STOPモードに設定すると、X2端子が内部で $V_{DD}$ に50 k（TYP.）の抵抗でプルアップされません。

**2** . スタンバイ・モードを設定する前に、あらかじめすべての割り込み要求フラグをリセットしてください。

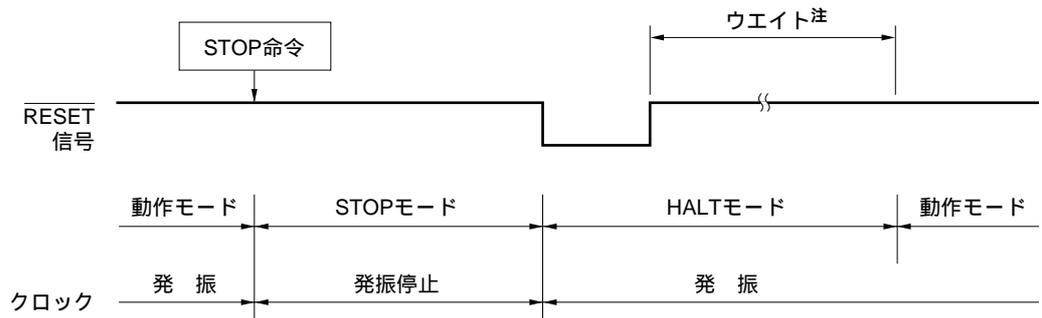
割り込み要求フラグと割り込み許可フラグの両方がセットされている割り込みソースがある場合には、スタンバイ・モードに入ってもただちに解除されます（図6-1 割り込み制御回路ブロック図を参照）。ただし、STOPモードを設定した場合は、STOP命令実行後すぐHALTモードに入り、BTMレジスタによる設定時間だけウエイトしたあと動作モードに戻ります。

## 7.2 スタンバイ・モードの解除

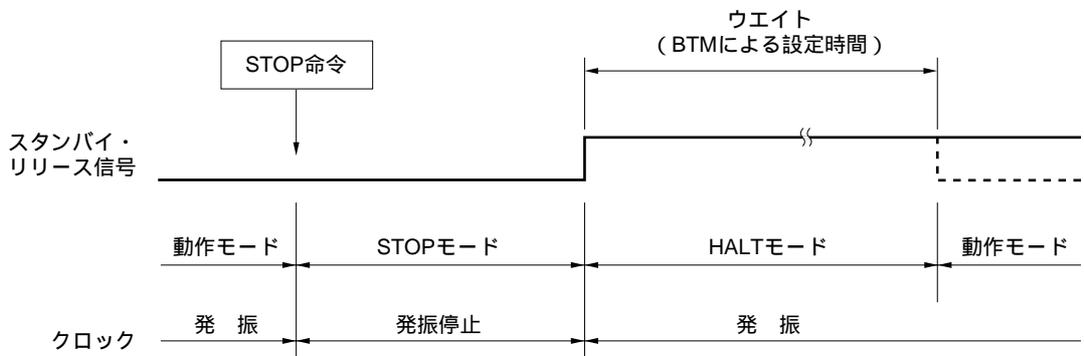
STOPモード，HALTモードいずれも割り込み許可フラグで許可されている割り込み要求信号の発生，および $\overline{\text{RESET}}$ 信号発生により解除されます。各モードの解除動作は図7-1のようになります。

図7-1 スタンバイ・モードの解除動作(1/2)

### (a) STOPモードの $\overline{\text{RESET}}$ 信号発生による解除



### (b) STOPモードの割り込み発生による解除



注 マスク・オプションにより次の2つの時間を選択することができます。

$2^{17}/f_x$  (21.8 ms : 6.0 MHz動作時，31.3 ms : 4.19 MHz動作時)

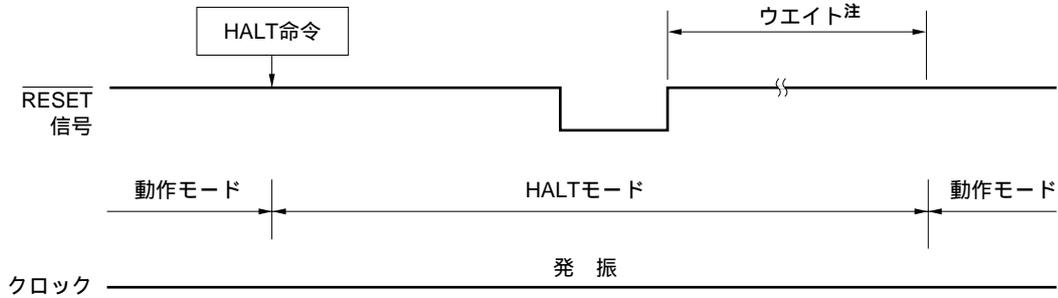
$2^{15}/f_x$  (5.46 ms : 6.0 MHz動作時，7.81 ms : 4.19 MHz動作時)

ただし， $\mu$ PD75P0076にはマスク・オプションがなく，ウエイト時間は $2^{15}/f_x$ に固定されています。

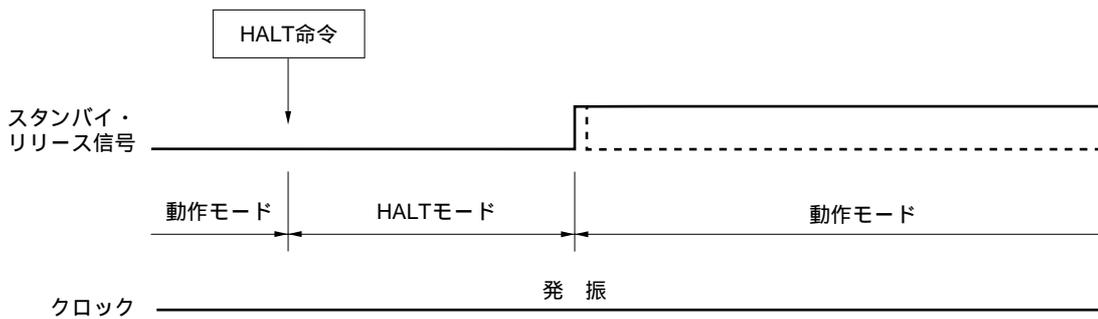
備考 破線は，スタンバイを解除した割り込み要求が受け付けられた場合です。

図7-1 スタンバイ・モードの解除動作 (2/2)

(c) HALTモードのRESET信号発生による解除



(d) HALTモードの割り込み発生による解除



注 マスク・オプションにより次の2つの時間を選択することができます。

$2^{17}/f_x$  (21.8 ms : 6.0 MHz動作時, 31.3 ms : 4.19 MHz動作時)

$2^{15}/f_x$  (5.46 ms : 6.0 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 $\mu$ PD75P0076にはマスク・オプションがなく、ウエイト時間は $2^{15}/f_x$ に固定されています。

備考 破線は、スタンバイを解除した割り込み要求が受け付けられた場合です。

STOPモードを割り込み発生により解除した場合、ウエイト時間はBTMの設定により決まります(表7-2参照)。

発振が安定するまでの時間は、使用する発振子の種類と、STOPモード解除時の電源電圧によって変わってきます。したがって使用状況に応じてウエイト時間を選択し、STOPモードを設定する前に、BTMをセットします。

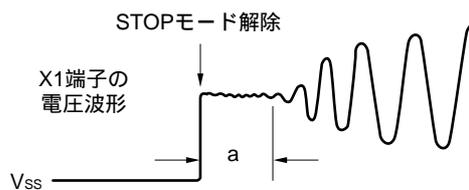
表7 - 2 BTMによるウェイト時間の選択

BTM3	BTM2	BTM1	BTM0	ウェイト時間 <sup>注</sup>	
				$f_x = 6.0 \text{ MHz}$ 時	$f_x = 4.19 \text{ MHz}$ 時
-	0	0	0	約 $2^{20}/f_x$ (約175 ms)	約 $2^{20}/f_x$ (約250 ms)
-	0	1	1	約 $2^{17}/f_x$ (約21.8 ms)	約 $2^{17}/f_x$ (約31.3 ms)
-	1	0	1	約 $2^{15}/f_x$ (約5.46 ms)	約 $2^{15}/f_x$ (約7.81 ms)
-	1	1	1	約 $2^{13}/f_x$ (約1.37 ms)	約 $2^{13}/f_x$ (約1.95 ms)
上記以外				設定禁止	

注 この時間は、STOPモード解除後、発振を開始するまでの時間は含みません。

注意 STOPモード解除時のウェイト時間は、 $\overline{\text{RESET}}$ 信号発生による場合も、割り込み発生による場合もSTOPモード解除後クロック発振を開始するまでの時間（図7 - 2のa）は含みません。

図7 - 2 STOPモード解除時のウェイト時間



## 7.3 スタンバイ・モード解除後の動作

- (1)  $\overline{\text{RESET}}$ 信号発生により解除された場合は、通常のリセット動作を実行します。
- (2) 割り込み発生により解除された場合、CPUが命令実行を再開したときにベクタ割り込みを行うかどうかは、割り込みマスタ許可フラグ (IME) の内容により決まります。

### (a) IME = 0 のとき

スタンバイ・モード解除後、スタンバイ・モード設定の次の命令から実行を再開します。  
割り込み要求フラグは保持されています。

### (b) IME = 1 のとき

スタンバイ・モード解除後、2命令を実行してからベクタ割り込みが実行されます。ただし、INTW, INT2 (テストブル入力) によって解除された場合はベクタ割り込みは発生しませんから、(a)と同様の処理を行います。

## 7.4 マスク・オプションの選択

$\mu$ PD750068のスタンバイ機能では、 $\overline{\text{RESET}}$ 信号発生によるスタンバイ機能解除後のウェイト時間を、次の2種類からマスク・オプションにより選択することができます。

$2^{17}/f_x$  (21.8 ms : 6.0 MHz動作時, 31.3 ms : 4.19 MHz動作時)

$2^{15}/f_x$  (5.46 ms : 6.0 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 $\mu$ PD75P0076にはマスク・オプションはなく、 $2^{15}/f_x$ に固定されています。

## 7.5 スタンバイ・モードの応用

スタンバイ・モードを使用する場合には次のような手順で行います。

割り込み入力あるいはポート入力による電源断などスタンバイ・モード設定要因の検出（電源断検出にはINT4を使用すると効果的です）。

入出力ポートの処理（消費電流が最小となるよう処理する）。

特に、入力ポートはオープンにしないでください。必ずロウ・レベルかハイ・レベルを入力してください。

スタンバイ・モードを解除する割り込みの指定（INT4を使用すると効果的、解除しない割り込み許可フラグはクリアする）。

解除後の動作指定（割り込み処理をするかしないかによりIMEを操作）。

解除後のCPUクロックの指定（切り替える場合は、スタンバイ・モード・セットまでに必要なマシン・サイクル経過するようにする）。

解除時のウェイト時間の選択。

スタンバイ・モード設定（STOP, HALT命令）。

さらにスタンバイ・モードは、システム・クロック切り替え機能と組み合わせることによって、低消費電流、低電圧動作を実現できます。

(1) STOPモードの応用例 (fx = 6.0 MHz動作時)

次の条件でSTOPモードを使用する場合

INT4の立ち下がリエッジ入力によりSTOPモードを設定し、立ち上がりエッジ入力で解除する (INTBTは使用しない)。

入出力ポートは、すべてハイ・インピーダンスとする (ハイ・インピーダンスにおいて消費電流が少なくなるように、外部で端子が処理されている場合)。

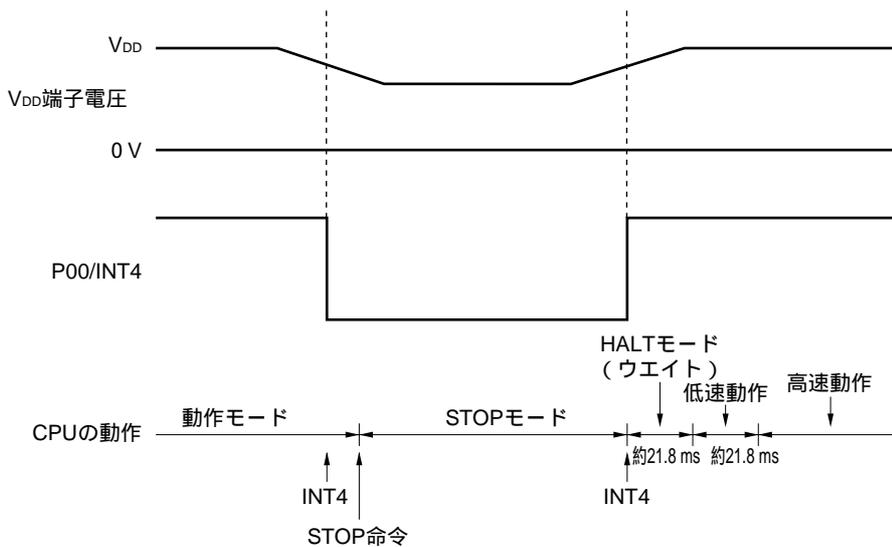
プログラムで使用する割り込みはINT0, INTT0。ただし、これらをSTOP解除には使用しない。解除後も割り込み許可する。

解除後は、最低速のCPUクロックで動作スタートする。

解除時のウェイト時間は約21.8 msとする。

解除後、電源の安定のためにさらに21.8 msウェイトする。またP00/INT4端子は2回チェックして、チャタリングを除去する。

タイミング・チャート



## プログラム例

(INT4処理プログラム, MBE = 0)

```

VSB4 : SKT      PORT0.0      ; P00 = 1?
        BR      PDOWN       ; パワーダウン
        SET1    BTM.3        ; パワーオン
WAIT : SKT      IRQBT        ; 21.8 ms ウェイト
        BR      WAIT
        SKT     PORT0.0      ; チャタリングをチェック
        BR      PDOWN
        MOV     A, #0011B
        MOV     PCC, A       ; 高速モード設定
        [ MOV   XA, #x xH ]   ; ポート・モード・レジスタをセット
        [ MOV   PMGm, XA ]
        EI      IE0
        EI      IET0
        RETI
PDOWN : MOV     A, #0         ; 最低速モード
        MOV     PCC, A
        MOV     XA, #00H
        MOV     PMGA, XA     ; 入出力ポート・ハイ・インピーダンス
        MOV     PMGB, XA
        DI      IE0         ; INT0, INTT0禁止
        DI      IET0
        MOV     A, #1011B
        MOV     BTM, A       ; ウェイト時間 21.8 ms
        NOP
        STOP                    ; STOPモードをセット
        NOP
        RETI

```

(2) HALTモードの応用例 (fx = 6.0 MHz動作時)

次の条件で間欠動作する場合

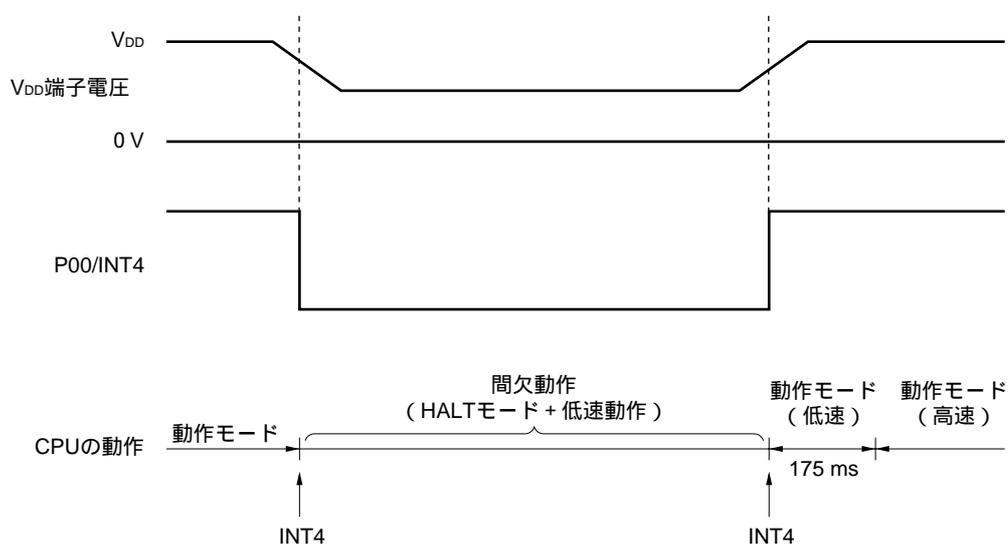
INT4の立ち下がりで、スタンバイ・モードを設定し、立ち上がりエッジ入力で解除する。

スタンバイ・モード中は175 ms間隔で、間欠動作する (INTBT)。

INT4, INTBTは低位の割り込みとする。

スタンバイ・モード中のCPUクロックは最低速とする。

タイミング・チャート



## プログラム例

(初期設定)

```

MOV    A, #0011B
MOV    PCC, A      ; 高速モード
MOV    XA, #05H
MOV    WM, XA     ; サブシステム・クロック
EI     IE4
EI     IEW
EI                      ; 割り込み許可

```

(メイン・ルーチン)

```

SKT    PORT0.0    ; 電源OK?
HALT                    ; パワーダウン・モード
NOP                    ; 電源OK?
SKTCLR IRQW        ; 0.5secフラグあり?
BR     MAIN        ; NO
CALL   WATCH       ; 時計サブルーチン

```

```

MAIN :      .....
           .....
           .....

```

(INT4処理ルーチン)

```

VINT4 : SKT    PORT0.0    ; 電源OK?, MBE = 0
        BR     PDOWN
        CLR1   SCC.3      ; メイン・システム・クロック発振開始
        MOV    A, #1000B
        MOV    BTM, A
WAIT1 : SKT    IRQBT      ; 175 msウェイト
        BR     WAIT1
        SKT    PORT0.0    ; チャタリングをチェック
        BR     PDOWN
        CLR1   SCC.0      ; メイン・システム・クロックに切り替え
        RETI
PDOWN : SET1   SCC.0      ; サブシステム・クロックに切り替え
        MOV    A, #5

```

```

WAIT2 :   INCS      A           ; 46マシン・サイクル以上待機注
          BR        WAIT2
          SET1     SCC.3       ; メイン・システム・クロック発振停止
          RETI
    
```

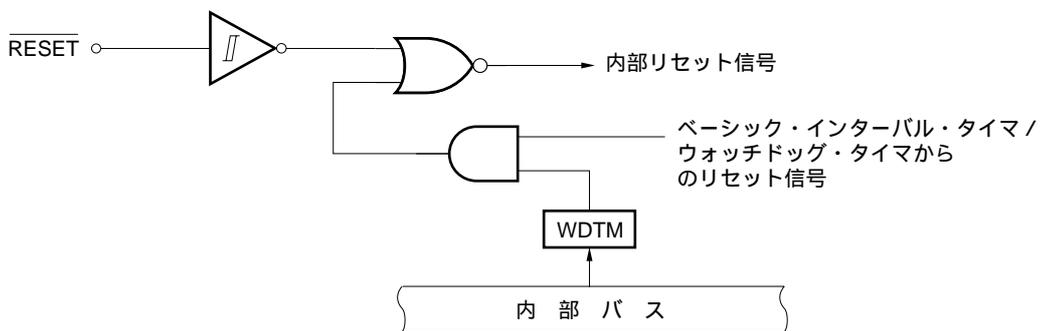
**注** システム・クロックとCPUクロックの切り替えについては、5.2.3 システム・クロックとCPUクロックの設定を参照してください。

**注意** メイン・システム・クロックからサブシステム・クロックに切り替える場合、サブシステム・クロックの発振安定を待ってから切り替えてください。

## 第8章 リセット機能

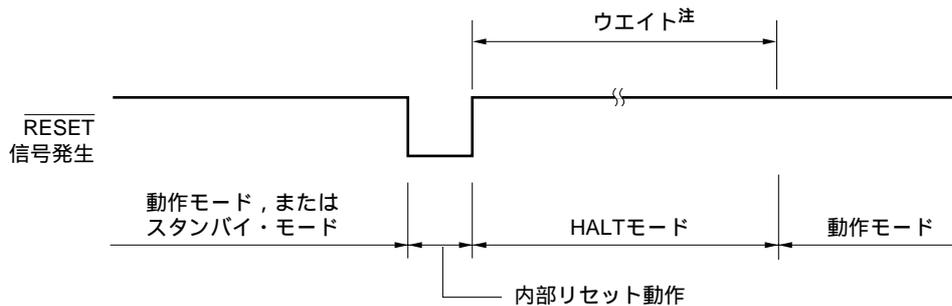
リセット入力には、外部リセット信号 ( $\overline{\text{RESET}}$ ) とベーシック・インターバル・タイマ/ウォッチドッグ・タイマからのリセット信号の2種類があります。どちらか一方のリセット信号が入力されると、内部リセット信号が発生します。図8 - 1にそれぞれの構成を示します。

図8 - 1 リセット機能の構成



$\overline{\text{RESET}}$ 信号発生によって、各ハードウェアは表8 - 1に示すようにイニシャライズされます。リセット動作のタイミングを図8 - 2に示します。

図8 - 2  $\overline{\text{RESET}}$ 信号発生によるリセット動作



注 マスク・オプションにより次の2つの時間を選択することができます。

$2^{17}/f_x$  (21.8 ms : 6.0 MHz動作時, 31.3 ms : 4.19 MHz動作時)

$2^{15}/f_x$  (5.46 ms : 6.0 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 $\mu$ PD75P0076にはマスク・オプションがなく、ウェイト時間は $2^{15}/f_x$ に固定されています。

表8-1 各ハードウェアのリセット後の状態(1/2)

ハードウェア		スタンバイ・モード中の RESET信号発生	動作中のRESET信号発生
プログラム・ カウンタ(PC)	μ PD750064	プログラム・メモリの0000H番地の 下位4ビットをPC11-PC8に, 0001H番地の内容をPC7-PC0に セット	左に同じ
	μ PD750066, 750068	プログラム・メモリの0000H番地の 下位5ビットをPC12-PC8に, 0001H番地の内容をPC7-PC0に セット	
	μ PD75P0076	プログラム・メモリの0000H番地の 下位6ビットをPC13-PC8に, 0001H番地の内容をPC7-PC0に セット	
PSW	キャリー・フラグ(CY)	保持	不定
	スキップ・フラグ(SK0-SK2)	0	0
	割り込みステータス・フラグ(IST0, IST1)	0	0
	バンク許可フラグ(MBE, RBE)	プログラム・メモリの0000H番地の ビット6をRBE, ビット7を MBEにセット	左に同じ
スタック・ポインタ(SP)		不定	不定
スタック・バンク選択レジスタ(SBS)		1000B	1000B
データ・メモリ(RAM)		保持	不定
汎用レジスタ(X, A, H, L, D, E, B, C)		保持	不定
バンク選択レジスタ(MBS, RBS)		0, 0	0, 0
ベーシック・イ ンターバル・タ イマ/ウォッチ ドッグ・タイマ	カウンタ(BT)	不定	不定
	モード・レジスタ(BTM)	0	0
タイマ/イベン ト・カウンタ (T0)	ウォッチドッグ・タイマ許可 フラグ(WDTM)	0	0
	カウンタ(T0)	0	0
タイマ/イベン ト・カウンタ (T1)	モジュロ・レジスタ(TMOD0)	FFH	FFH
	モード・レジスタ(TM0)	0	0
	TOE0, TOUT F/F	0, 0	0, 0
タイマ/イベン ト・カウンタ (T1)	カウンタ(T1)	0	0
	モジュロ・レジスタ(TMOD1)	FFH	FFH
	モード・レジスタ(TM1)	0	0
時計用タイマ	TOE1, TOUT F/F	0, 0	0, 0
	モード・レジスタ(WM)	0	0

表8 - 1 各ハードウェアのリセット後の状態 (2/2)

ハードウェア		スタンバイ・モード中の RESET信号発生	動作中のRESET信号発生
シリアル・ インタフェース	シフト・レジスタ (SIO)	保持	不定
	動作モード・レジスタ (CSIM)	0	0
	SBIコントロール・レジスタ (SBIC)	0	0
クロック発生 回路, クロック出力 回路	プロセッサ・クロック・ コントロール・レジスタ (PCC)	0	0
	システム・クロック・ コントロール・レジスタ (SCC)	0	0
	クロック出力モード・レジスタ (CLOM)	0	0
サブ発振回路コントロール・レジスタ (SOS)		0	0
A/Dコンバータ	モード・レジスタ (ADM)	04H	04H
	SAレジスタ (SA)	7FH	7FH
割り込み機能	割り込み要求フラグ (IRQ x x x)	リセット (0)	リセット (0)
	割り込み許可フラグ (IE x x x)	0	0
	割り込みマスタ許可フラグ (IME)	0	0
	割り込みプライオリティ 選択レジスタ (IPS)	0	0
	INT0, 1, 2モード・レジスタ (IM0, IM1, IM2)	0, 0, 0	0, 0, 0
デジタル・ ポート	出力バッファ	オフ	オフ
	出力ラッチ	クリア (0)	クリア (0)
	入出力モード・レジスタ (PMGA, PMGB)	0	0
	ブルアップ抵抗指定レジスタ (POGA)	0	0
ビット・シーケンシャル・バッファ (BSB0-BSB3)		保持	不定

(メモ)

## 第9章 PROM (プログラム・メモリ) の書き込みとベリファイ

μPD75P0076に内蔵されているプログラム・メモリはワン・タイムPROMです。メモリ容量は次のとおりです。

μPD75P0076 : 16384ワード × 8ビット

このワン・タイムPROMの書き込み/ベリファイのために表9 - 1に示すような端子を使用します。なお、アドレスの入力はなく、代わりにX1端子からクロック入力することで、アドレスを更新します。

表9 - 1 プログラム・メモリ書き込み/ベリファイ時の使用端子

端子名	機能
X1, X2	プログラム・メモリ書き込み/ベリファイ時のアドレス更新のクロックを入力し、X2端子にはその反転した信号を入力します。
MD0/P30-MD3/P33	プログラム・メモリ書き込み/ベリファイ時の動作モードの選択端子。
D0/P40-D3/P43 (下位4ビット), D4/P50-D7/P53 (上位4ビット)	プログラム・メモリ書き込み/ベリファイ時の8ビット・データの入出力端子。
V <sub>DD</sub>	電源電圧の印加端子。 通常動作時は1.8 ~ 5.5 Vを印加。 プログラム・メモリ書き込み/ベリファイ時は+ 6 Vを印加。
V <sub>PP</sub>	プログラム・メモリ書き込み/ベリファイ時の電圧印加端子 (通常はV <sub>DD</sub> 電位)。

注意1 . μPD75P0076は、消去用窓を持っていないので、紫外線消去はできません。

2 . プログラム・メモリの書き込み/ベリファイ時に使用しない端子は、次のように処理します。

- XT2端子以外 ... ブルダウン抵抗を介してV<sub>SS</sub>に接続。
- XT2端子 ... オープン

## 9.1 プログラム・メモリの書き込み/ベリファイ時の動作モード

μPD75P0076は、V<sub>DD</sub>端子に+ 6 V、V<sub>PP</sub>端子に+ 12.5 Vを印加すると、プログラム・メモリ書き込み/ベリファイのモードになります。このモードは、MD0-MD3端子への入力信号により次のような動作モードになります。

表9 - 2 動作モード

動作モードの指定						動作モード
V <sub>DD</sub>	V <sub>PP</sub>	MD0	MD1	MD2	MD3	
+ 6 V	+ 12.5 V	H	L	H	L	プログラム・メモリ・アドレスの0クリア
		L	H	H	H	書き込みモード
		L	L	H	H	ベリファイ・モード
		H	x	H	H	プログラム・インヒビット・モード

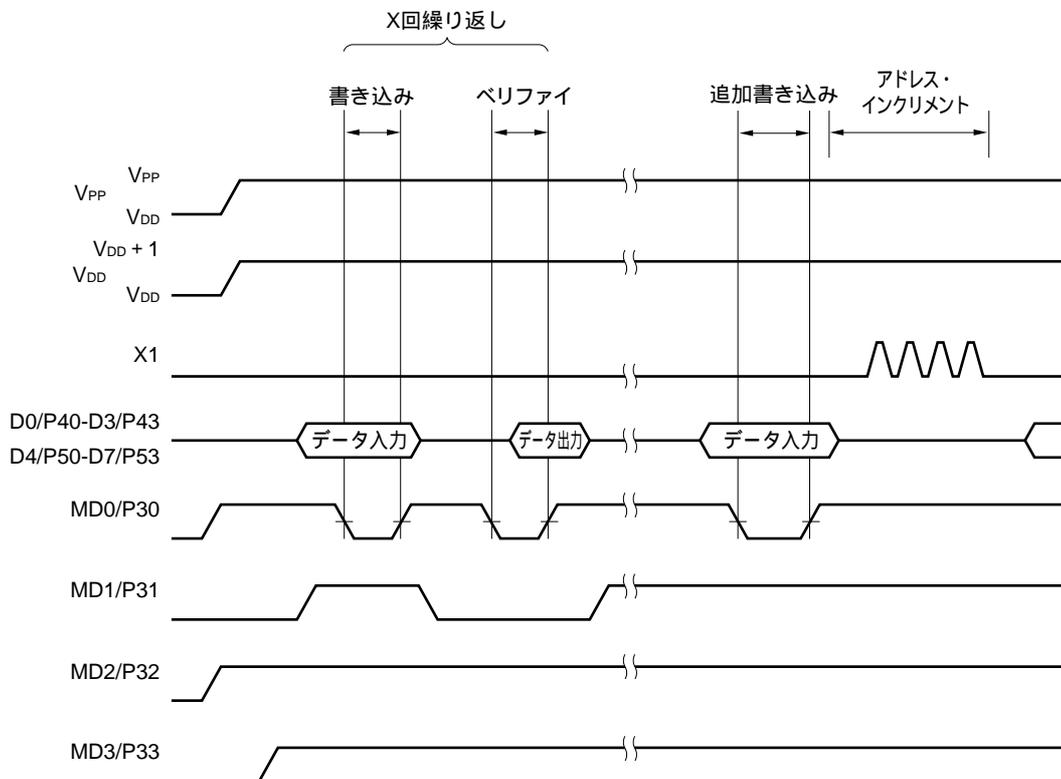
備考 x : LまたはH

## 9.2 プログラム・メモリの書き込み手順

プログラム・メモリ書き込み手順は次のようになっており、高速書き込みが可能です。

- (1) 使用しない端子を抵抗を介してV<sub>SS</sub>にプルダウン。X1端子はロウ・レベル。
- (2) V<sub>DD</sub>, V<sub>PP</sub>端子に5 Vを供給。
- (3) 10  $\mu$ sウエイト。
- (4) プログラム・メモリ・アドレスの0クリア・モード
- (5) V<sub>DD</sub>に+ 6 V, V<sub>PP</sub>に+ 12.5 Vを供給。
- (6) 1 msの書き込みモードでデータを書き込む。
- (7) ベリファイ・モード。書き込めていれば(8)へ、書き込めていなければ(6), (7)を繰り返す。
- (8) ((6), (7)で書き込んだ回数: X)  $\times$  1 msの追加書き込み。
- (9) X1端子にパルスを4発入力することにより、プログラム・メモリ・アドレスを更新(+ 1)。
- (10) (6) ~ (9)を最終アドレスまで繰り返す。
- (11) プログラム・メモリ・アドレスの0クリア・モード。
- (12) V<sub>DD</sub>, V<sub>PP</sub>端子の電圧を5 Vに変更。
- (13) 電源オフ。

(2) ~ (9)の手順を下図に示します。

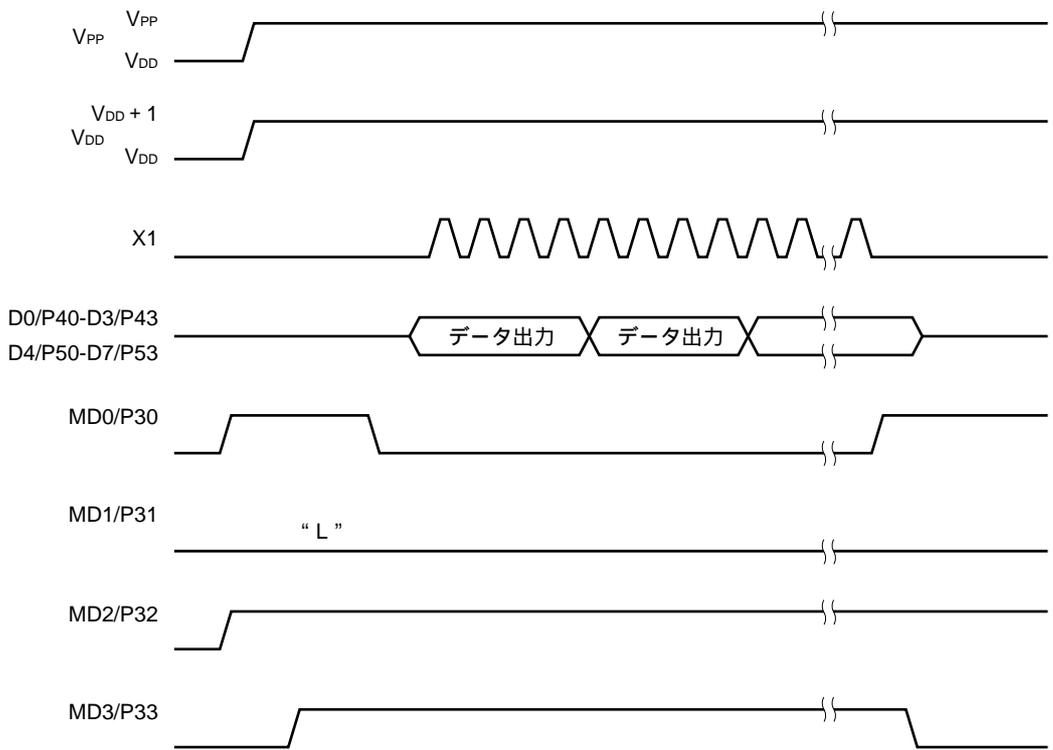


### 9.3 プログラム・メモリの読み出し手順

次の手順によりプログラム・メモリの内容の読み出しができます。読み出しは、ベリファイ・モードで行います。

- (1) 使用しない端子を抵抗を介してV<sub>SS</sub>にプルダウン。X1端子はロウ・レベル。
- (2) V<sub>DD</sub>, V<sub>PP</sub>端子に5 Vを供給。
- (3) 10 μsウエイト。
- (4) プログラム・メモリ・アドレスの0クリア・モード。
- (5) V<sub>DD</sub>に+ 6 V, V<sub>PP</sub>に+ 12.5 Vを供給。
- (6) ベリファイ・モード。X1端子にクロック・パルスを入力すると4 発入力する周期でデータを1 アドレスずつ順次出力。
- (7) プログラム・メモリ・アドレスの0クリア・モード。
- (8) V<sub>DD</sub>, V<sub>PP</sub>の電圧を5 Vに変更。
- (9) 電源オフ。

(2) ~ (7) の手順を下図に示します。



## 9.4 ワン・タイムPROMのスクリーニングについて

ワン・タイムPROM製品は、その構造上、当社にて完全な試験をして出荷することはできません。必要なデータを書き込んだあと、下記の条件で高温保管後、PROMのベリファイを行うスクリーニングを実施することを推奨します。

保管温度	保管時間
125	24時間

(メモ)

## 第10章 マスク・オプション

### 10.1 端子

μPD750068の端子には、次のようなマスク・オプションがあります。

表10 - 1 端子のマスク・オプションの選択

端子	マスク・オプション
P40-P43, P50-P53	ビット単位でプルアップ抵抗の内蔵を指定可能

P40-P43 (ポート4), P50-P53 (ポート5) は、マスク・オプションによりプルアップ抵抗の内蔵を指定することができます。マスク・オプションはビット単位で指定することができます。

ポート4, 5のリセット後の状態は、マスク・オプションによるプルアップ抵抗の内蔵を指定した場合、ハイ・レベルになります。内蔵を指定しない場合は、ハイ・インピーダンスになります。

μPD75P0076にはマスク・オプションによるプルアップ抵抗の内蔵機能はありません。

### 10.2 スタンバイ機能のマスク・オプション

μPD750068のスタンバイ機能では、マスク・オプションによりウェイト時間を選択することができます。これは、 $\overline{\text{RESET}}$ 信号によってスタンバイ機能を解除したあと、通常動作モードに戻るまでの時間です (詳細は7.2 スタンバイ・モードの解除参照)。

ウェイト時間は次の2つの時間から選択することができます。

$2^{17}/f_x$  (21.8 ms :  $f_x = 6.0$  MHz動作時, 31.3 ms :  $f_x = 4.19$  MHz動作時)

$2^{15}/f_x$  (5.46 ms :  $f_x = 6.0$  MHz動作時, 7.81 ms :  $f_x = 4.19$  MHz動作時)

μPD75P0076にはマスク・オプションはなく、ウェイト時間は $2^{15}/f_x$ に固定されています。

### 10.3 サブシステム・クロックのフィードバック抵抗のマスク・オプション

μPD750068のサブシステム・クロックでは、マスク・オプションによりフィードバック抵抗を使用可能にするかどうかを選択できます。

フィードバック抵抗を使用可能にする（ソフトウェアでオン/オフを切り替える）

フィードバック抵抗を使用不可能にする（ハードウェアで切断する）

を選択したあとフィードバック抵抗を使用するには、ソフトウェアでSOS.0を“0”に設定し、フィードバック抵抗をオンにします（詳細は5.2.2（6）サブ発振回路コントロール・レジスタ（SOS）参照）。

サブシステム・クロックを使用する場合は、 を選択してください。

μPD75P0076にはマスク・オプションの設定はなく、フィードバック抵抗は使用可能になっています。

# 第11章 命令セット

μPD750068の命令セットは、75Xシリーズの命令セットを改良・発展させたもので、75Xシリーズからの継承性を保持した命令セットとなっており、次のような特色があります。

- ( 1 ) 多彩な応用ができるビット操作命令
- ( 2 ) 効率のよい4ビット操作命令
- ( 3 ) 8ビット・マイコンに匹敵する8ビット操作命令
- ( 4 ) プログラム・サイズ短縮のためのGETI命令
- ( 5 ) プログラムの効率を上げるたてづみ命令，進数補正命令
- ( 6 ) 連続参照に適したテーブル参照命令
- ( 7 ) 1バイト相対分岐命令
- ( 8 ) わかりやすく整理されたNEC標準二モニック

なお，データ・メモリを操作する場合に適用できるアドレッシング・モードおよび，命令実行時に有効となるレジスタ・バンクについては3.2 汎用レジスタのバンク構成を参照してください。

## 11.1 特徴的な命令

ここでは，μPD750068の命令セットのうち，特徴的な命令の概要を説明します。

### 11.1.1 GETI命令

GETI(ゲットアイ)命令は以下の命令を1バイト命令に変換するための命令です。

- ( a ) 16 Kバイト空間(0000H-3FFFH)のサブルーチン・コール命令
- ( b ) 16 Kバイト空間(0000H-3FFFH)への分岐命令
- ( c ) 任意の2バイト，2マシン・サイクルの命令(ただし，BRCB命令およびCALLF命令を除く)
- ( d ) 1バイト命令2つの組み合わせ

GETI命令では，プログラム・メモリの0020H-007FH番地のテーブルを参照し，参照された2バイトのデータを，( a ) ~ ( d )の命令として実行します。したがって，48通りもの( a ) ~ ( d )の命令が1バイト命令に変換できることになります。

このGETI命令を使って使用頻度の高い( a ) ~ ( d )の命令を1バイトに変換すれば，プログラムのバイト数を大幅に短縮することができます。

### 11.1.2 ビット操作命令

μPD750068は、通常のビット操作命令（セット、クリア命令）に加え、ビット・テスト命令、ビット転送命令、およびビット・ブーリアン命令（AND, OR, XOR）が強化されています。

操作ビットの指定は、ビット操作アドレッシングにより行います。ビット操作アドレッシングには、3種類あり、各アドレッシングにより操作されるビットを表11 - 1 に示します。

表11 - 1 ビット操作アドレッシングの種類と指定可能な範囲

アドレッシング	操作可能な周辺ハードウェア	操作可能なビットのアドレッシング範囲
fmem. bit	RBE, MBE, IST1, IST0, SCC, IE x x x , IRQ x x x	FB0H-FBFH
	PORT0-6, 11	FF0H-FFFH
pmem. @L	BSP0-3, PORT0-6, 11	FC0H-FFFH
@H + mem. bit	ビット操作可能なすべての周辺ハードウェア	MBで指定されるメモリ・バンクのビット操作可能なすべてのビット

備考 1 . x x x : 0, 1, 2, 4, BT, T0, T1, W, CSI

2 . MB = MBE · MBS

### 11.1.3 たてづみ命令

μPD750068には、次の2種類のたてづみ命令が用意されています。

( a ) MOV A, #n4 または MOV XA, #n8

( b ) MOV HL, #n8

「たてづみ」とはこの2種類の命令を、それぞれ連続したアドレスに置くことを示します。

**例** A0 : MOV A, #0  
A1 : MOV A, #1  
XA7 : MOV XA, #07

この例のように、たてづみ命令が並べられていると、最初に行うアドレスがA0の場合はそれに続く2命令が、最初に行うアドレスがA1の場合は続く1命令が、NOP命令に置き換えられて実行されます。すなわち、最初に行われた命令だけが有効で、それに続くたてづみ命令はすべてNOP命令として処理されるわけです。

このたてづみ命令を使うことによって、アキュムレータ（Aレジスタ、レジスタ・ペアXA）への定数設定や、データ・ポインタ（レジスタ・ペアHL）への定数設定を効率よく行うことができます。

### 11.1.4 進数補正命令

応用によっては、4ビット・データの加算または減算（2進数で行われる）の結果を10進数に変換したり、あるいは、時刻のように6進補正したりする必要があります。

このためにμPD750068の命令セットでは、4ビット・データの加算あるいは減算の結果を、任意の進数に補正するための進数補正命令が用意されています。

#### (1) 加算時の進数補正

補正したい進数値をmとすると

```

ADDS  A, #16 - m
ADDC  A, @HL      ; A, CY  A + (HL) + CY
ADDS  A, #m

```

の組み合わせによって、アキュムレータとメモリ（HL）を加算し、加算結果をm進補正します。オーバーフローはキャリー・フラグに残ります。

ADDC A, @HL命令の実行の結果、キャリーが出ると続くADDS A, #n4命令をスキップします。またキャリーが出なければADDS A, #n4命令が実行されますが、このときこの命令のスキップ機能は禁止され、加算の結果キャリーが出ても続く命令をスキップしません。したがってADDS A, #n4命令に続けてプログラムを書くことができます。

例 アキュムレータと、メモリを10進加算する。

```

ADDS  A, #6
ADDC  A, @HL      ; A, CY  A + (HL) + CY
ADDS  A, #10
      ⋮

```

#### (2) 減算時の進数補正

補正したい進数値をmとすると

```

SUBC  A, @HL
ADDS  A, #m

```

の組み合わせにより、アキュムレータからメモリ（HL）を減算し、結果をm進補正します。

アンダフローはキャリー・フラグに残ります。

SUBC A, @HL命令の実行の結果、ボローが出なければ続くADDS A, #n4命令をスキップします。またボローが出るとADDS A, #n4命令が実行されますが、このときこの命令のスキップ機能は禁止され、加算の結果キャリーが出ても続く命令をスキップしません。したがって、ADDS A, #n4命令に続けてプログラムを書くことができます。

### 11.1.5 スキップ命令とスキップに要するマシン・サイクル数

μPD750068の命令セットでは、スキップによって条件判断をして、プログラムを構成するようになっていきます。

スキップ命令を実行したとき、スキップ条件が満足されると、続く1命令を飛ばして(スキップして)1命令あとの命令が実行されます。

スキップが発生したとき、スキップに要するマシン・サイクル数は

(a) スキップ命令に続く命令(スキップされる命令)が、3バイト命令(BR !addr命令, BRA !addr1命令, CALL !addr命令, CALLA !addr1命令)の場合: 2マシン・サイクル

(b) (a)以外の命令の場合: 1マシン・サイクル

## 11.2 命令セットとそのオペレーション

### (1) オペランドの表現形式と記述方法

各命令のオペランド欄には、その命令のオペランド表現形式に対する記述方法に従ってオペランドを記述します(詳細はRA75X **アセンブラ・パッケージ ユーザーズ・マニュアル 言語編**(EEU-730)を参照してください)。記述方法の中で、複数個あるものは、それらの要素の1つを選択します。大文字で書かれた英字および+, -記号はキー・ワードであり、そのまま記述します。

イミューディアット・データの場合は、適当な数値またはラベルを記述します。

ラベルとしてmem, fmem, pmem, bitなどの代わりに、図3-7に書かれた各種レジスタ・フラグの略語を記述できます(ただし、fmem, pmemは記述できるラベルに制限があります。詳しくは、**表3-1 アドレッシング・モード**, **図3-7 μPD750068 I/Oマップ**を参照してください)。

表現形式	記述方法
reg	X, A, B, C, D, E, H, L
reg1	X, B, C, D, E, H, L
rp	XA, BC, DE, HL
rp1	BC, DE, HL
rp2	BC, DE
rp'	XA, BC, DE, HL, XA', BC', DE', HL'
rp'1	BC, DE, HL, XA', BC', DE', HL'
rpa	HL, HL + , HL - , DE, DL
rpa1	DE, DL
n4	4ビット・イミディエト・データまたはレーベル
n8	8ビット・イミディエト・データまたはレーベル
mem	8ビット・イミディエト・データまたはレーベル <sup>注</sup>
bit	2ビット・イミディエト・データまたはレーベル
fmem	FB0H-FBFH, FF0H-FFFHイミディエト・データまたはレーベル
pmem	FC0H-FFFHイミディエト・データまたはレーベル
addr,	000H-FFFHイミディエト・データまたはレーベル ( μ PD750064 )
addr1 ( Mk	0000H-17FFHイミディエト・データまたはレーベル ( μ PD750066 )
モード時のみ )	0000H-1FFFHイミディエト・データまたはレーベル ( μ PD750068 )
	0000H-3FFFHイミディエト・データまたはレーベル ( μ PD75P0076 )
caddr	12ビット・イミディエト・データまたはレーベル
faddr	11ビット・イミディエト・データまたはレーベル
taddr	20H-7FHイミディエト・データ ( ただしbit0 = 0 ) またはレーベル
PORTn	PORT0-PORT6, PORT11
IE x x x	IEBT, IET0, IET1, IE0-IE2, IE4, IECSI, IEW
RBn	RB0-RB3
MBn	MB0, MB1, MB15

注 memは , 8ビット・データ処理の場合は偶数アドレスのみ記述できます。

## (2) オペレーション説明上の凡例

A	: Aレジスタ; 4ビット・アキュムレータ
B	: Bレジスタ
C	: Cレジスタ
D	: Dレジスタ
E	: Eレジスタ
H	: Hレジスタ
L	: Lレジスタ
X	: Xレジスタ
XA	: レジスタ・ペア (XA); 8ビット・アキュムレータ
BC	: レジスタ・ペア (BC)
DE	: レジスタ・ペア (DE)
HL	: レジスタ・ペア (HL)
XA'	: 拡張レジスタ・ペア (XA')
BC'	: 拡張レジスタ・ペア (BC')
DE'	: 拡張レジスタ・ペア (DE')
HL'	: 拡張レジスタ・ペア (HL')
PC	: プログラム・カウンタ
SP	: スタック・ポインタ
CY	: キャリー・フラグ; ビット・アキュムレータ
PSW	: プログラム・ステータス・ワード
MBE	: メモリ・バンク許可フラグ
RBE	: レジスタ・バンク許可フラグ
PORT <sub>n</sub>	: ポート <sub>n</sub> (n = 0-6, 11)
IME	: 割り込みマスタ許可フラグ
IPS	: 割り込みプライオリティ選択レジスタ
IE <sub>x x x</sub>	: 割り込み許可フラグ
RBS	: レジスタ・バンク選択レジスタ
MBS	: メモリ・バンク選択レジスタ
PCC	: プロセッサ・クロック・コントロール・レジスタ
.	: アドレス, ビット区切り
( x x )	: x x でアドレスされる内容
x x H	: 16進データ

(3) アドレッシング・エリア欄の記号説明

* 1	MB = MBE・MBS ( MBS = 0, 1, 15 )		データ・メモリ・ アドレッシング
* 2	MB = 0		
* 3	MBE = 0 : MB = 0 ( 000H-07FH ) MB = 15 ( F80H-FFFFH ) MBE = 1 : MB = MBS ( MBS = 0, 1, 15 )		
* 4	MB = 15, fmem = FB0H-FBFH, FF0H-FFFFH		
* 5	MB = 15, pmem = FC0H-FFFFH		
* 6	μ PD750064	addr, addr1 = 000H-FFFFH	プログラム・メモリ・ アドレッシング
	μ PD750066	addr, addr1 = 0000H-17FFH	
	μ PD750068	addr, addr1 = 0000H-1FFFFH	
	μ PD75P0076	addr, addr1 = 0000H-3FFFH	
* 7	addr, addr1 = ( Current PC ) - 15 ~ ( Current PC ) - 1 ( Current PC ) + 2 ~ ( Current PC ) + 16		
* 8	μ PD750064	caddr = 000H-FFFFH	
	μ PD750066	caddr = 0000H-0FFFH ( PC <sub>12</sub> = 0 ) or 1000H-17FFH ( PC <sub>12</sub> = 1 )	
	μ PD750068	caddr = 0000H-0FFFH ( PC <sub>12</sub> = 0 ) or 1000H-17FFH ( PC <sub>12</sub> = 1 )	
	μ PD75P0076	caddr = 0000H-0FFFH ( PC <sub>13,12</sub> = 00B ) or 1000H-17FFH ( PC <sub>13,12</sub> = 01B ) or 2000H-2FFFH ( PC <sub>13,12</sub> = 10B ) or 3000H-3FFFH ( PC <sub>13,12</sub> = 11B )	
* 9	faddr = 0000H-07FFH		
* 10	taddr = 0020H-007FH		
* 11	Mk モード時のみ  addr1 = 0000H-0FFFH ( μ PD750064 ) 0000H-17FFH ( μ PD750066 ) 0000H-1FFFFH ( μ PD750068 ) 0000H-3FFFH ( μ PD75P0076 )		

備考1 . MBはアクセス可能なメモリ・バンクを示します。

- 2 . \*2ではMBE, MBSに関係なくMB = 0です。
- 3 . \*4, \*5ではMBE, MBSに関係なくMB = 15です。
- 4 . \*6 ~ \*11は, それぞれアドレッシング可能な領域を示します。

**(4) マシン・サイクルの欄の説明**

Sは、スキップ付き命令がスキップ動作をするときに要するマシン・サイクル数を示します。Sの値は次のように変わります。

スキップしないとき .....	S = 0
スキップされる命令が、1バイト命令、または2バイト命令のとき .....	S = 1
スキップされる命令が、3バイト命令 <sup>注</sup> のとき .....	S = 2

**注** 3バイト命令：BR !addr, BRA !addr1, CALL !addr, CALLA !addr1命令

**注意** GETI命令は1マシン・サイクルでスキップされます。

1マシン・サイクルはCPUクロック の1サイクル分 (=  $t_{CY}$ ) に等しく、PCCの設定により4通りの時間が選択できます(図5-12 プロセッサ・クロック・コントロール・レジスタのフォーマット参照)。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
転送	MOV	A, #n4	1	1	A n4		たてづみA
		reg1, #n4	2	2	reg1 n4		
		XA, #n8	2	2	XA n8		たてづみA
		HL, #n8	2	2	HL n8		たてづみB
		rp2, #n8	2	2	rp2 n8		
		A, @HL	1	1	A (HL)	*1	
		A, @HL +	1	2 + S	A (HL), then L L + 1	*1	L = 0
		A, @HL -	1	2 + S	A (HL), then L L - 1	*1	L = FH
		A, @rpa1	1	1	A (rpa1)	*2	
		XA, @HL	2	2	XA (HL)	*1	
		@HL, A	1	1	(HL) A	*1	
		@HL, XA	2	2	(HL) XA	*1	
		A, mem	2	2	A (mem)	*3	
		XA, mem	2	2	XA (mem)	*3	
		mem, A	2	2	(mem) A	*3	
		mem, XA	2	2	(mem) XA	*3	
		A, reg	2	2	A reg		
		XA, rp'	2	2	XA rp'		
	reg1, A	2	2	reg1 A			
	rp'1, XA	2	2	rp'1 XA			
	XCH	A, @HL	1	1	A (HL)	*1	
		A, @HL +	1	2 + S	A (HL), then L L + 1	*1	L = 0
		A, @HL -	1	2 + S	A (HL), then L L - 1	*1	L = FH
		A, @rpa1	1	1	A (rpa1)	*2	
		XA, @HL	2	2	XA (HL)	*1	
		A, mem	2	2	A (mem)	*3	
		XA, mem	2	2	XA (mem)	*3	
		A, reg1	1	1	A reg1		
XA, rp'		2	2	XA rp'			

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
テーブル参照	MOVT	XA, @PCDE	1	3	μ PD750064 XA ( PC <sub>11-8</sub> + DE ) <sub>ROM</sub>		
					μ PD750066, 750068 XA ( PC <sub>12-8</sub> + DE ) <sub>ROM</sub>		
					μ PD75P0076 XA ( PC <sub>13-8</sub> + DE ) <sub>ROM</sub>		
		XA, @PCXA	1	3	μ PD750064 XA ( PC <sub>11-8</sub> + XA ) <sub>ROM</sub>		
					μ PD750066, 750068 XA ( PC <sub>12-8</sub> + XA ) <sub>ROM</sub>		
					μ PD75P0076 XA ( PC <sub>13-8</sub> + XA ) <sub>ROM</sub>		
		XA, @BCDE	1	3	XA ( BCDE ) <sub>ROM</sub> <sup>注</sup>	* 6	
	XA, @BCXA	1	3	XA ( BCXA ) <sub>ROM</sub> <sup>注</sup>	* 6		
ビット転送	MOV1	CY, fmem.bit	2	2	CY ( fmem.bit )	* 4	
		CY, pmem.@L	2	2	CY ( pmem <sub>7-2</sub> + L <sub>3-2.bi</sub> ( L <sub>1-0</sub> ) )	* 5	
		CY, @H + mem.bit	2	2	CY ( H + mem <sub>3-0.bit</sub> )	* 1	
		fmem.bit, CY	2	2	( fmem.bit ) CY	* 4	
		pmem.@L, CY	2	2	( pmem <sub>7-2</sub> + L <sub>3-2.bi</sub> ( L <sub>1-0</sub> ) ) CY	* 5	
		@H + mem.bit, CY	2	2	( H + mem <sub>3-0.bit</sub> ) CY	* 1	
演算	ADDS	A, #n4	1	1 + S	A A + n4		carry
		XA, #n8	2	2 + S	XA XA + n8		carry
		A, @HL	1	1 + S	A A + ( HL )	* 1	carry
		XA, rp'	2	2 + S	XA XA + rp'		carry
		rp'1, XA	2	2 + S	rp'1 rp'1 + XA		carry
	ADDC	A, @HL	1	1	A, CY A + ( HL ) + CY	* 1	
		XA, rp'	2	2	XA, CY XA + rp' + CY		
		rp'1, XA	2	2	rp'1, CY rp'1 + XA + CY		
	SUBS	A, @HL	1	1 + S	A A - ( HL )	* 1	borrow
XA, rp'		2	2 + S	XA XA - rp'		borrow	
rp'1, XA		2	2 + S	rp'1 rp'1 - XA		borrow	

注 μ PD750064を使用する場合は、Bレジスタには“ 0 ”を設定してください。

μ PD750066, 750068を使用する場合は、Bレジスタは下位 1 ビットのみ有効です。

μ PD75P0076を使用する場合は、下位 2 ビットのみ有効です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
演算	SUBC	A, @HL	1	1	A, CY A - (HL) - CY	* 1	
		XA, rp'	2	2	XA, CY XA - rp' - CY		
		rp'1, XA	2	2	rp'1, CY rp'1 - XA - CY		
	AND	A, #n4	2	2	A A n4		
		A, @HL	1	1	A A (HL)	* 1	
		XA, rp'	2	2	XA XA rp'		
		rp'1, XA	2	2	rp'1 rp'1 XA		
	OR	A, #n4	2	2	A A n4		
		A, @HL	1	1	A A (HL)	* 1	
		XA, rp'	2	2	XA XA rp'		
		rp'1, XA	2	2	rp'1 rp'1 XA		
	XOR	A, #n4	2	2	A A $\nabla$ n4		
		A, @HL	1	1	A A $\nabla$ (HL)	* 1	
		XA, rp'	2	2	XA XA $\nabla$ rp'		
		rp'1, XA	2	2	rp'1 rp'1 $\nabla$ XA		
	操作 アキュムレータ	RORC	A	1	1	CY A <sub>0</sub> , A <sub>3</sub> CY, A <sub>n-1</sub> A <sub>n</sub>	
NOT		A	2	2	A $\bar{A}$		
増減	INCS	reg	1	1 + S	reg reg + 1		reg = 0
		rp1	1	1 + S	rp1 rp1 + 1		rp1 = 00H
		@HL	2	2 + S	(HL) (HL) + 1	* 1	(HL) = 0
		mem	2	2 + S	(mem) (mem) + 1	* 3	(mem) = 0
	DECS	reg	1	1 + S	reg reg - 1		reg = FH
		rp'	2	2 + S	rp' rp' - 1		rp' = FFH
比較	SKE	reg, #n4	2	2 + S	Skip if reg = n4		reg = n4
		@HL, #n4	2	2 + S	Skip if (HL) = n4	* 1	(HL) = n4
		A, @HL	1	1 + S	Skip if A = (HL)	* 1	A = (HL)
		XA, @HL	2	2 + S	Skip if XA = (HL)	* 1	XA = (HL)
		A, reg	2	2 + S	Skip if A = reg		A = reg
		XA, rp'	2	2 + S	Skip if XA = rp'		XA = rp'
操作 キャリー・フラグ	SET1	CY	1	1	CY 1		
	CLR1	CY	1	1	CY 0		
	SKT	CY	1	1 + S	Skip if CY = 1		CY = 1
	NOT1	CY	1	1	CY $\bar{CY}$		

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
メモリ・ビット操作	SET1	mem.bit	2	2	( mem.bit ) 1	* 3	
		fmem.bit	2	2	( fmem.bit ) 1	* 4	
		pmem.@L	2	2	( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) ) 1	* 5	
		@H + mem.bit	2	2	( H + mem <sub>3-0</sub> .bit ) 1	* 1	
	CLR1	mem.bit	2	2	( mem.bit ) 0	* 3	
		fmem.bit	2	2	( fmem.bit ) 0	* 4	
		pmem.@L	2	2	( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) ) 0	* 5	
		@H + mem.bit	2	2	( H + mem <sub>3-0</sub> .bit ) 0	* 1	
	SKT	mem.bit	2	2 + S	Skip i( mem.bit ) = 1	* 3	( mem.bit ) = 1
		fmem.bit	2	2 + S	Skip i( fmem.bit ) = 1	* 4	( fmem.bit ) = 1
		pmem.@L	2	2 + S	Skip i( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) ) = 1	* 5	( pmem.@L ) = 1
		@H + mem.bit	2	2 + S	Skip i( H + mem <sub>3-0</sub> .bit ) = 1	* 1	( @H + mem.bit ) = 1
	SKF	mem.bit	2	2 + S	Skip i( mem.bit ) = 0	* 3	( mem.bit ) = 0
		fmem.bit	2	2 + S	Skip i( fmem.bit ) = 0	* 4	( fmem.bit ) = 0
		pmem.@L	2	2 + S	Skip i( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) ) = 0	* 5	( pmem.@L ) = 0
		@H + mem.bit	2	2 + S	Skip i( H + mem <sub>3-0</sub> .bit ) = 0	* 1	( @H + mem.bit ) = 0
	SKTCLR	fmem.bit	2	2 + S	Skip i( fmem.bit ) = 1 and clear	* 4	( fmem.bit ) = 1
		pmem.@L	2	2 + S	Skip i( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) ) = 1 and clear	* 5	( pmem.@L ) = 1
		@H + mem.bit	2	2 + S	Skip i( H + mem <sub>3-0</sub> .bit ) = 1 and clear	* 1	( @H + mem.bit ) = 1
	AND1	CY, fmem.bit	2	2	CY CY ( fmem.bit )	* 4	
		CY, pmem.@L	2	2	CY CY ( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) )	* 5	
		CY, @H + mem.bit	2	2	CY CY ( H + mem <sub>3-0</sub> .bit )	* 1	
	OR1	CY, fmem.bit	2	2	CY CY ( fmem.bit )	* 4	
		CY, pmem.@L	2	2	CY CY ( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) )	* 5	
CY, @H + mem.bit		2	2	CY CY ( H + mem <sub>3-0</sub> .bit )	* 1		
XOR1	CY, fmem.bit	2	2	CY CY ∇ ( fmem.bit )	* 4		
	CY, pmem.@L	2	2	CY CY ∇ ( pmem <sub>7-2</sub> + L <sub>3-2</sub> .bi( L <sub>1-0</sub> ) )	* 5		
	CY, @H + mem.bit	2	2	CY CY ∇ ( H + mem <sub>3-0</sub> .bit )	* 1		

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
分岐	BR	addr	-	-	<p><math>\mu</math>PD750064</p> <p>PC<sub>11-0</sub> addr</p> <p>〔アセンブラにより、次のうち最適な命令を選択します。〕</p> <ul style="list-style-type: none"> <li>・ BR !addr</li> <li>・ BRCB !caddr</li> <li>・ BR \$addr</li> </ul>	* 6	
				<p><math>\mu</math>PD750066, 750068</p> <p>PC<sub>12-0</sub> addr</p> <p>〔アセンブラにより、次のうち最適な命令を選択します。〕</p> <ul style="list-style-type: none"> <li>・ BR !addr</li> <li>・ BRCB !caddr</li> <li>・ BR \$addr</li> </ul>			
				<p><math>\mu</math>PD75P0076</p> <p>PC<sub>13-0</sub> addr</p> <p>〔アセンブラにより、次の命令から最適な命令を選択します。〕</p> <ul style="list-style-type: none"> <li>・ BR !addr</li> <li>・ BRCB !caddr</li> <li>・ BR \$addr</li> </ul>			

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
分岐	BR <sup>注</sup>	addr1	-	-	$\mu$ PD750064 PC <sub>11-0</sub> addr1 [ アセンブラにより、次のうち最適な命令を選択します。 ・ BR ! addr ・ BRA ! addr1 ・ BRCB ! caddr ・ BR \$addr1 ]	* 11	
					$\mu$ PD750066, 750068 PC <sub>12-0</sub> addr1 [ アセンブラにより、次のうち最適な命令を選択します。 ・ BR ! addr ・ BRA ! addr1 ・ BRCB ! caddr ・ BR \$addr1 ]		
					$\mu$ PD75P0076 PC <sub>13-0</sub> addr1 [ アセンブラにより、次の命令から最適な命令を選択します。 ・ BR ! addr ・ BRA ! addr1 ・ BRCB ! caddr ・ BR \$addr1 ]		
		! addr	3	3	$\mu$ PD750064 PC <sub>11-0</sub> addr $\mu$ PD750066, 750068 PC <sub>12-0</sub> addr $\mu$ PD75P0076 PC <sub>13-0</sub> addr	* 6	
		\$addr	1	2	$\mu$ PD750064 PC <sub>11-0</sub> addr $\mu$ PD750066, 750068 PC <sub>12-0</sub> addr $\mu$ PD75P0076 PC <sub>13-0</sub> addr	* 7	

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
分岐	BR <sup>注1</sup>	\$addr1	1	2	μPD750064 PC <sub>11-0</sub> addr1	* 7	
					μPD750066, 750068 PC <sub>12-0</sub> addr1		
					μPD75P0076 PC <sub>13-0</sub> addr1		
	PCDE	2	3	μPD750064 PC <sub>11-0</sub> PC <sub>11-8</sub> + DE			
				μPD750066, 750068 PC <sub>12-0</sub> PC <sub>12-8</sub> + DE			
				μPD75P0076 PC <sub>13-0</sub> PC <sub>13-8</sub> + DE			
	PCXA	2	3	μPD750064 PC <sub>11-0</sub> PC <sub>11-8</sub> + XA			
				μPD750066, 750068 PC <sub>12-0</sub> PC <sub>12-8</sub> + XA			
				μPD75P0076 PC <sub>13-0</sub> PC <sub>13-8</sub> + XA			
	BCDE	2	3	μPD750064 PC <sub>11-0</sub> BCDE <sup>注2</sup>	* 6		
				μPD750066, 750068 PC <sub>12-0</sub> BCDE <sup>注3</sup>			
				μPD75P0076 PC <sub>13-0</sub> BCDE <sup>注4</sup>			
BCXA	2	3	μPD750064 PC <sub>11-0</sub> BCXA <sup>注2</sup>	* 6			
			μPD750066, 750068 PC <sub>12-0</sub> BCXA <sup>注3</sup>				
			μPD75P0076 PC <sub>13-0</sub> BCXA <sup>注4</sup>				

注1 . で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

- 2 . Bレジスタには必ず“ 0 ”を設定してください。
- 3 . Bレジスタは、下位1ビットのみ有効です。
- 4 . Bレジスタは、下位2ビットのみ有効です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
分岐	BRA <sup>注</sup>	! addr1	3	3	$\mu$ PD750064 PC <sub>11-0</sub> addr1 $\mu$ PD750066, 750068 PC <sub>12-0</sub> addr1 $\mu$ PD75P0076 PC <sub>13-0</sub> addr1	* 11	
	BRCB	! caddr	2	2	$\mu$ PD750064 PC <sub>11-0</sub> caddr <sub>11-0</sub> $\mu$ PD750066, 750068 PC <sub>12-0</sub> PC <sub>12</sub> + caddr <sub>11-0</sub> $\mu$ PD75P0076 PC <sub>13-0</sub> PC <sub>13, 12</sub> + caddr <sub>11-0</sub>	* 8	
サブルーチン・スタック制御	CALLA <sup>注</sup>	! addr1	3	3	$\mu$ PD750064 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> addr1, SP SP - 6 $\mu$ PD750066, 750068 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, PC <sub>12</sub> PC <sub>12-0</sub> addr1, SP SP - 6 $\mu$ PD75P0076 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, PC <sub>13, 12</sub> PC <sub>13-0</sub> addr1, SP SP - 6	* 11	

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
サブルーチン・スタック制御	CALL <sup>注</sup>	! addr	3	3	$\mu$ PD750064 (SP - 3) MBE, RBE, 0, 0 (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>11-0</sub> addr, SP SP - 4	* 6	
					$\mu$ PD750066, 750068 (SP - 3) MBE, RBE, 0, PC <sub>12</sub> (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>12-0</sub> addr, SP SP - 4		
					$\mu$ PD75P0076 (SP - 3) MBE, RBE, PC <sub>13, 12</sub> (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>13-0</sub> addr, SP SP - 4		
				4	$\mu$ PD750064 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> addr, SP SP - 6		
					$\mu$ PD750066, 750068 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, PC <sub>12</sub> PC <sub>12-0</sub> addr, SP SP - 6		
					$\mu$ PD75P0076 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, PC <sub>13, 12</sub> PC <sub>13-0</sub> addr, SP SP - 6		

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニク	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
サブルーチン・スタック制御	CALLF <sup>注</sup>	! faddr	2	2	$\mu$ PD750064 (SP - 3) MBE, RBE, 0, 0 (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>11-0</sub> 0 + faddr, SP SP - 4	* 9	
					$\mu$ PD750066, 750068 (SP - 3) MBE, RBE, 0, PC <sub>12</sub> (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>12-0</sub> 00 + faddr, SP SP - 4		
					$\mu$ PD75P0076 (SP - 3) MBE, RBE, PC <sub>13, 12</sub> (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>13-0</sub> 000 + faddr, SP SP - 4		
			3	$\mu$ PD750064 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> 0 + faddr, SP SP - 6			
				$\mu$ PD750066, 750068 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, PC <sub>12</sub> PC <sub>12-0</sub> 00 + faddr, SP SP - 6			
				$\mu$ PD75P0076 (SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, PC <sub>13, 12</sub> PC <sub>13-0</sub> 000 + faddr, SP SP - 6			

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
サブルーチン・スタック制御	RET <sup>注</sup>		1	3	$\mu$ PD750064 PC <sub>11-0</sub> (SP) (SP+3) (SP+2) MBE, RBE, 0, 0 (SP+1), SP SP+4		
					$\mu$ PD750066, 750068 PC <sub>11-0</sub> (SP) (SP+3) (SP+2) MBE, RBE, 0, PC <sub>12</sub> (SP+1), SP SP+4		
					$\mu$ PD75P0076 PC <sub>11-0</sub> (SP) (SP+3) (SP+2) MBE, RBE, PC <sub>13, 12</sub> (SP+1) SP SP+4		
					$\mu$ PD750064 x, x, MBE, RBE (SP+4) 0, 0, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2), SP SP+6		
					$\mu$ PD750066, 750068 x, x, MBE, RBE (SP+4) 0, 0, 0, PC <sub>12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2), SP SP+6		
					$\mu$ PD75P0076 x, x, MBE, RBE (SP+4) 0, 0, PC <sub>13, 12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) SP SP+6		

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	ニモニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
サブルーチン・スタック制御	RETS <sup>注</sup>		1	3+S	$\mu$ PD750064 MBE, RBE, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) SP SP+4 then skip unconditionally		無条件
					$\mu$ PD750066, 750068 MBE, RBE, 0, PC <sub>12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) SP SP+4 then skip unconditionally		
					$\mu$ PD75P0076 MBE, RBE, PC <sub>13, 12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) SP SP+4 then skip unconditionally		
					$\mu$ PD750064 0, 0, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) x, x, MBE, RBE (SP+4) SP SP+6 then skip unconditionally		
					$\mu$ PD750066, 750068 0, 0, 0, PC <sub>12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) x, x, MBE, RBE (SP+4) SP SP+6 then skip unconditionally		
					$\mu$ PD75P0076 0, 0, PC <sub>13, 12</sub> (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) x, x, MBE, RBE (SP+4) SP SP+6 then skip unconditionally		
					$\mu$ PD750064 0, 0, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) x, x, MBE, RBE (SP+4) SP SP+6 then skip unconditionally		

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
サブルーチン・スタック制御	RET <sup>注</sup>		1	3	$\mu$ PD750064 MBE, RBE, 0, 0 (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
					$\mu$ PD750066, 750068 MBE, RBE, 0, PC <sub>12</sub> (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
					$\mu$ PD75P0076 MBE, RBE, PC <sub>13, 12</sub> (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
					$\mu$ PD750064 0, 0, 0, 0 (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
					$\mu$ PD750066, 750068 0, 0, 0, PC <sub>12</sub> (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
					$\mu$ PD75P0076 0, 0, PC <sub>13, 12</sub> (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6		
	PUSH	rp	1	1	(SP - 1) (SP - 2) rp, SP SP - 2		
		BS	2	2	(SP - 1) MBS, (SP - 2) RBS, SP SP - 2		
	POP	rp	1	1	rp (SP + 1) (SP), SP SP + 2		
BS		2	2	MBS (SP + 1), RBS (SP), SP SP + 2			
割り込み制御	EI		2	2	IME (IPS. 3) 1		
		IE x x x	2	2	IE x x x 1		
	DI		2	2	IME (IPS. 3) 0		
		IE x x x	2	2	IE x x x 0		

注  で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
入出力	IN <sup>注</sup>	A, PORTn	2	2	A PORTn (n = 0-6, 11)		
		XA, PORTn	2	2	XA PORTn+1, PORTn (n = 4)		
	OUT <sup>注</sup>	PORTn, A	2	2	PORTn A (n = 2-6)		
		PORTn, XA	2	2	PORTn+1, PORTn XA (n = 4)		
CPU制御	HALT		2	2	Set HALT Mode (PCC. 2 1)		
	STOP		2	2	Set STOP Mode (PCC. 3 1)		
	NOP		1	1	No Operation		
特殊	SEL	RBn	2	2	RBS n (n = 0-3)		
		MBn	2	2	MBS n (n = 0, 1, 15)		

注 IN/OUT命令実行時には、MBE = 0またはMBE = 1、MBS = 15としておく必要があります。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件	
特殊	GETI <sup>注</sup>	taddr	1	3	$\mu$ PD750064 ・ TBR命令のとき $PC_{11-0} (taddr)_{3-0} + (taddr + 1)$	* 10		
					・ TCALL命令のとき $(SP - 4) (SP - 1) (SP - 2) PC_{11-0}$ $(SP - 3) MBE, RBE, 0, 0$ $PC_{11-0} (taddr)_{3-0} + (taddr + 1)$ $SP SP - 4$			
					・ TBR, TCALL命令以外 のとき $(taddr) (taddr + 1)$ の命令実行			参照した命令による
					$\mu$ PD750066, 750068 ・ TBR命令のとき $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$			
・ TCALL命令のとき $(SP - 4) (SP - 1) (SP - 2) PC_{11-0}$ $(SP - 3) MBE, RBE, 0, PC_{12}$ $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$ $SP SP - 4$								
・ TBR, TCALL命令以外 のとき $(taddr) (taddr + 1)$ の命令実行	参照した命令による							
$\mu$ PD75P0076 ・ TBR命令のとき $PC_{13-0} (taddr)_{5-0} + (taddr + 1)$								
・ TCALL命令のとき $(SP - 4) (SP - 1) (SP - 2) PC_{11-0}$ $(SP - 3) MBE, RBE, PC_{13, 12}$ $PC_{13-0} (taddr)_{5-0} + (taddr + 1)$ $SP SP - 4$								
・ TBR, TCALL命令以外 のとき $(taddr) (taddr + 1)$ の命令実行		参照した命令による						

注 TBR, TCALL命令はGETI命令のテーブル定義用アセンブラ疑似命令です。

命令群	ニモニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
特殊	GETI <sup>注1, 2</sup>	taddr	1	3	$\mu$ PD750064 ・ TBR命令のとき $PC_{11-0} (taddr)_{3-0} + (taddr + 1)$	* 10	
				4	・ TCALL命令のとき $(SP - 6) (SP - 3) (SP - 4) PC_{11-0}$ $(SP - 5) 0, 0, 0, 0$ $(SP - 2) x, x, MBE, RBE$ $PC_{11-0} (taddr)_{3-0} + (taddr + 1)$ $SP SP - 6$		
				3	・ TBR, TCALL命令以外のとき $(taddr) (taddr + 1)$ の命令実行		参照した命令による
				3	$\mu$ PD750066, 750068 ・ TBR命令のとき $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$		
				4	・ TCALL命令のとき $(SP - 6) (SP - 3) (SP - 4) PC_{11-0}$ $(SP - 5) 0, 0, 0, PC_{12}$ $(SP - 2) x, x, MBE, RBE$ $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$ $SP SP - 6$		
				3	・ TBR, TCALL命令以外のとき $(taddr) (taddr + 1)$ の命令実行		参照した命令による
				3	$\mu$ PD75P0076 ・ TBR命令のとき $PC_{13-0} (taddr)_{5-0} + (taddr + 1)$		
				4	・ TCALL命令のとき $(SP - 6) (SP - 3) (SP - 4) PC_{11-0}$ $(SP - 5) 0, 0, PC_{13, 12}$ $(SP - 2) x, x, MBE, RBE$ $PC_{13-0} (taddr)_{5-0} + (taddr + 1)$ $SP SP - 6$		
3	・ TBR, TCALL命令以外のとき $(taddr) (taddr + 1)$ の命令実行	参照した命令による					

注1 . TBR, TCALL命令はGETI命令のテーブル定義用アセンブラ疑似命令です。

2 . で示す部分はMk モード時のみ対応可能です。そのほかは、Mk モード時にのみ対応可能です。

### 11.3 各命令の命令コード

(1) 命令コードの記号説明

R <sub>2</sub>	R <sub>1</sub>	R <sub>0</sub>	reg
0	0	0	A
0	0	1	X
0	1	0	L
0	1	1	H
1	0	0	E
1	0	1	D
1	1	0	C
1	1	1	B

↑

reg

↑

reg1

↑

P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>	reg-pair
0	0	0	XA
0	0	1	XA'
0	1	0	HL
0	1	1	HL'
1	0	0	DE
1	0	1	DE'
1	1	0	BC
1	1	1	BC'

↑

rp'

↑

rp'1

↑

Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	addressing
0	0	0	@HL
0	1	0	@HL +
0	1	1	@HL -
1	0	0	@DE
1	0	1	@DL

↑

@rpa

↑

@rpa1

↑

P <sub>2</sub>	P <sub>1</sub>	reg-pair
0	0	XA
0	1	HL
1	0	DE
1	1	BC

↑

rp2

↑

rp1

↑

rp

N <sub>5</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	IE x x x
0	0	0	0	IEBT
0	0	1	0	IEW
0	1	0	0	IET0
0	1	0	1	IECSI
0	1	1	0	IE0
0	1	1	1	IE2
1	0	0	0	IE4
1	1	0	0	IET1
1	1	1	0	IE1

I<sub>n</sub> : n4, n8に対するイミューディエト・データ

D<sub>n</sub> : memに対するイミューディエト・データ

B<sub>n</sub> : bitに対するイミューディエト・データ

N<sub>n</sub> : n, IE x x xに対するイミューディエト・データ

T<sub>n</sub> : taddr ×  $\frac{1}{2}$  に対するイミューディエト・データ

A<sub>n</sub> : [ 分岐先アドレスとの相対アドレス距離 ( 2 ~ 16 ) ] - 1 に対するイミューディエト・データ

S<sub>n</sub> : [ 分岐先アドレスとの相対アドレス距離 ( 15 ~ 1 ) ] の 1 の補数に対するイミューディエト・データ

(2) ビット操作アドレッシングの命令コード

オペランドの欄の \* 1 は、以下の3種類があることを示します。

- {
- fmem. bit
- pmem. @L
- @H + mem. bit
- }

上記アドレッシングに対応する命令コードの2バイト目 \* 2 は、下表のようになります。

* 1	命令コード・2バイト目	アクセス可能なビット
fmem. bit	1 0 B <sub>1</sub> B <sub>0</sub> F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	FB0H-FBFHの操作可能ビット
	1 1 B <sub>1</sub> B <sub>0</sub> F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	FF0H-FFFHの操作可能ビット
pmem. @L	0 1 0 0 G <sub>3</sub> G <sub>2</sub> G <sub>1</sub> G <sub>0</sub>	FC0H-FFFHの操作可能ビット
@H + mem. bit	0 0 B <sub>1</sub> B <sub>0</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	アクセス可能なメモリ・バンクの操作可能ビット

B<sub>n</sub> : bitに対するイミディエト・データ

F<sub>n</sub> : fmemに対するイミディエト・データ

(アドレスの下位4ビットを示す)

G<sub>n</sub> : pmemに対するイミディエト・データ

(アドレスのビット5-2を示す)

D<sub>n</sub> : memに対するイミディエト・データ

(アドレスの下位4ビットを示す)

命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
転送	MOV	A, #n4	0 1 1 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>		
		reg1, #n4	1 0 0 1 1 0 1 0	I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		rp, #n8	1 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 1	I <sub>7</sub> I <sub>6</sub> I <sub>5</sub> I <sub>4</sub> I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	
		A, @rpa1	1 1 1 0 0 Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 1 0 0 0	
		@HL, A	1 1 1 0 1 0 0 0		
		@HL, XA	1 0 1 0 1 0 1 0	0 0 0 1 0 0 0 0	
		A, mem	1 0 1 0 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		XA, mem	1 0 1 0 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		mem, A	1 0 0 1 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		mem, XA	1 0 0 1 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		A, reg	1 0 0 1 1 0 0 1	0 1 1 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		reg1, A	1 0 0 1 1 0 0 1	0 1 1 1 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
	rp'1, XA	1 0 1 0 1 0 1 0	0 1 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
	XCH	A, @rpa1	1 1 1 0 1 Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 0 0 0 1	
		A, mem	1 0 1 1 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		XA, mem	1 0 1 1 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		A, reg1	1 1 0 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
XA, rp'		1 0 1 0 1 0 1 0	0 1 0 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
テーブル参照	MOVT	XA, @PCDE	1 1 0 1 0 1 0 0		
		XA, @PCXA	1 1 0 1 0 0 0 0		
		XA, @BCXA	1 1 0 1 0 0 0 1		
		XA, @BCDE	1 1 0 1 0 1 0 1		
転送ビット	MOV1	CY, [* 1]	1 0 1 1 1 1 0 1	* 2	
		[* 1], CY	1 0 0 1 1 0 1 1	* 2	

命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
演算	ADDS	A, #n4	0 1 1 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>		
		XA, #n8	1 0 1 1 1 0 0 1	l <sub>7</sub> l <sub>6</sub> l <sub>5</sub> l <sub>4</sub> l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 1 0 1 0 0 1 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 0 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 0 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	ADDC	A, @HL	1 0 1 0 1 0 0 1		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	SUBS	A, @HL	1 0 1 0 1 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 1 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	SUBC	A, @HL	1 0 1 1 1 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 1 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 1 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AND	A, #n4	1 0 0 1 1 0 0 1	0 0 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 0 1 0 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 0 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 0 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	OR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 1 0 0 0 0 0		
XA, rp'		1 0 1 0 1 0 1 0	1 0 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
rp'1, XA		1 0 1 0 1 0 1 0	1 0 1 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
XOR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>		
	A, @HL	1 0 1 1 0 0 0 0			
	XA, rp'	1 0 1 0 1 0 1 0	1 0 1 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
	rp'1, XA	1 0 1 0 1 0 1 0	1 0 1 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
ア プ ロ グ レ ー タ	RORC	A	1 0 0 1 1 0 0 0		
	NOT	A	1 0 0 1 1 0 0 1	0 1 0 1 1 1 1 1	

命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
増減	INCS	reg	1 1 0 0 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
		rp1	1 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 0		
		@HL	1 0 0 1 1 0 0 1	0 0 0 0 0 0 1 0	
		mem	1 0 0 0 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
	DECS	reg	1 1 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
		rp'	1 0 1 0 1 0 1 0	0 1 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
比較	SKE	reg, #n4	1 0 0 1 1 0 1 0	l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		@HL, #n4	1 0 0 1 1 0 0 1	0 1 1 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 0 0 0 0 0 0		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 1 0 0 1	
		A, reg	1 0 0 1 1 0 0 1	0 0 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
キャリー・フラグ操作	SET1	CY	1 1 1 0 0 1 1 1		
	CLR1	CY	1 1 1 0 0 1 1 0		
	SKT	CY	1 1 0 1 0 1 1 1		
	NOT1	CY	1 1 0 1 0 1 1 0		
メモリ・ビット操作	SET1	mem.bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 0 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		<input type="checkbox"/> * 1	1 0 0 1 1 1 0 1	* 2	
	CLR1	mem.bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 0 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		<input type="checkbox"/> * 1	1 0 0 1 1 1 0 0	* 2	
	SKT	mem.bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		<input type="checkbox"/> * 1	1 0 1 1 1 1 1 1	* 2	
	SKF	mem.bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		<input type="checkbox"/> * 1	1 0 1 1 1 1 1 0	* 2	
	SKTCLR	<input type="checkbox"/> * 1	1 0 0 1 1 1 1 1	* 2	
	AND1	CY, <input type="checkbox"/> * 1	1 0 1 0 1 1 0 0	* 2	
	OR1	CY, <input type="checkbox"/> * 1	1 0 1 0 1 1 1 0	* 2	
	XOR1	CY, <input type="checkbox"/> * 1	1 0 1 1 1 1 0 0	* 2	

命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
★ 分岐	BR	laddr	1 0 1 0 1 0 1 1	0 0 ←	addr →
		\$addr1 (+16) (-2) (-1) (-15)	0 0 0 0 A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>		
			1 1 1 1 S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>		
		PCDE	1 0 0 1 1 0 0 1	0 0 0 0 0 1 0 0	
		PCXA	1 0 0 1 1 0 0 1	0 0 0 0 0 0 0 0	
		BCDE	1 0 0 1 1 0 0 1	0 0 0 0 0 1 0 1	
		BCXA	1 0 0 1 1 0 0 1	0 0 0 0 0 0 0 1	
分岐	BRA	laddr1	1 0 1 1 1 0 1 0	0 ←	addr1 →
	BRCB	lcaddr	0 1 0 1 ←	caddr →	
サブ ルー チ ン ・ ス タ ック 制 御	CALLA	laddr1	1 0 1 1 1 0 1 1	0 ←	addr1 →
	CALL	laddr	1 0 1 0 1 0 1 1	0 1 ←	addr →
	CALLF	lfaddr	0 1 0 0 0 ←	faddr →	
	RET		1 1 1 0 1 1 1 0		
	RETS		1 1 1 0 0 0 0 0		
	RETI		1 1 1 0 1 1 1 1		
	PUSH	rp	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 1		
		BS	1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 1	
POP	rp	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 0			
	BS	1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 0		
割 り 込 み 制 御	EI		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 0	
		IE x x x	1 0 0 1 1 1 0 1	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
	DI		1 0 0 1 1 1 0 0	1 0 1 1 0 0 1 0	
		IE x x x	1 0 0 1 1 1 0 0	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
入 出 力	IN	A, PORT <sub>n</sub>	1 0 1 0 0 0 1 1	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
		XA, PORT <sub>n</sub>	1 0 1 0 0 0 1 0	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
	OUT	PORT <sub>n</sub> , A	1 0 0 1 0 0 1 1	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
		PORT <sub>n</sub> , XA	1 0 0 1 0 0 1 0	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
C P U 制 御	HALT		1 0 0 1 1 1 0 1	1 0 1 0 0 0 1 1	
	STOP		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 1	
	NOP		0 1 1 0 0 0 0 0		
特 殊	SEL	RB <sub>n</sub>	1 0 0 1 1 0 0 1	0 0 1 0 0 0 N <sub>1</sub> N <sub>0</sub>	
		MB <sub>n</sub>	1 0 0 1 1 0 0 1	0 0 0 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
	GETI	taddr	0 0 T <sub>5</sub> T <sub>4</sub> T <sub>3</sub> T <sub>2</sub> T <sub>1</sub> T <sub>0</sub>		

## 11.4 命令機能と応用

この項では、命令の機能と応用を説明します。μPD750064, 750066, 750068, 75P0076のMk モード, Mk モードでは、使用できる命令、または命令の機能が異なりますので、次の読み方に従って読んでください。

### 読み方

- 次のすべてに共通して使用できます。
 

μ PD750064	}	のMk モード時とMk モード時
μ PD750066		
μ PD750068		
μ PD75P0076		
- μ PD750064, 750066, 750068, 75P0076のMk モード時のみ使用できます。
- μ PD750064, 750066, 750068, 75P0076のMk モード時のみ使用できます。
- / μ PD750064, 750066, 750068, 75P0076のMk モード時とMk モード時に共通して使用することができますが、Mk モードとMk モードでは、機能面で異なる箇所があります。  
Mk モードの場合は【Mk モード】、Mk モードの場合は【Mk モード】に示す説明書きを、それぞれの場合に応じて、読み分けてください。

**備考** この項で示されている機能は、プログラム・カウンタが13ビットのμPD750066, 750068を例にして説明してあります。μPD750064のプログラム・カウンタは12ビット、μPD75P0076のプログラム・カウンタは14ビットですので、それぞれ読み替えてお使いください。

## 11.4.1 転送命令

## ○ MOV A, #n4

機能 : A n4 n4 = l3-0 : 0-FH

4ビット・イミディエト・データn4をAレジスタ(4ビット・アキュムレータ)に転送します。たてづみ効果(グループA)があり, MOV A, #n4またはMOV XA, #n8命令が続けて置かれると実行した命令以降のたてづみ命令はNOPとして処理されます。

## 応用例

(1) アキュムレータに0BHをセットする。

```
MOV A, #0BH
```

(2) ポート3に出力するデータを0-2から選ぶ。

```
A0: MOV A, #0
```

```
A1: MOV A, #1
```

```
A2: MOV A, #2
```

```
OUT PORT3, A
```

## ○ MOV reg1, #n4

機能 : reg1 n4 n4 = l3-0 0-FH

4ビット・イミディエト・データn4をAレジスタreg1(X, H, L, D, E, B, C)に転送します。

## ○ MOV XA, #n8

機能 : XA n8 n8 = l7-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアXAに転送します。たてづみ効果があり, 同じ命令か, またはMOV A, #n4命令が続けて置かれると実行した命令以降のたてづみ命令は, NOPとして処理されます。

## ○ MOV HL, #n8

機能: HL n8 n8 = 17-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアHLに転送します。たてづみ効果があり、同じ命令が続けて置かれると実行した命令以降のたてづみ命令は、NOPとして処理されます。

## ○ MOV rp2, #n8

機能: rp2 n8 n8 = 17-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアrp2 (BC, DE) に転送します。

## ○ MOV A, @HL

機能: A (HL)

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。

## ○ MOV A, @HL +

機能: A (HL), L L + 1  
skip if L = 0H

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。そのあとLレジスタの内容を自動的に+ 1し、その結果が0Hになると、続く1命令をスキップします。

## ○ MOV A, @HL -

機能: A (HL), L L - 1  
skip if L = FH

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。そのあとLレジスタの内容を自動的に- 1し、その結果がFHになると、続く1命令をスキップします。

## ○ MOV A, @rpa1

機能：A (rpa)

rpa = HL + のとき：skip if L = 0

rpa = HL - のとき：skip if L = FH

レジスタ・ペアrpa (HL, HL+ , HL - , DE, DL) でアドレスされるデータ・メモリの内容をAレジスタに転送します。

rpaとしてオートインクリメント (HL+ ) を指定した場合は、データの転送後、Lレジスタの内容を自動的に + 1 し、その結果が0になると続く1命令をスキップします。

rpaとしてオートデクリメント (HL - ) を指定した場合は、データの転送後、Lレジスタの内容を自動的に - 1 し、その結果がFHになると続く1命令をスキップします。

## ○ MOV XA, @HL

機能: A (HL), X (HL + 1)

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送し、メモリの次の番地の内容をXレジスタに転送します。

ただし、Lレジスタの内容が奇数の場合、最下位ビットが無視されたアドレスが転送されます。

### 応用例

3EH, 3FH番地のデータをレジスタ・ペアXAに転送する。

```
MOV HL, #3EH
MOV XA, @HL
```

## ○ MOV @HL, A

機能: (HL) A

Aレジスタの内容をレジスタ・ペアHLでアドレスされるデータ・メモリへ転送します。

## ○ MOV @HL, XA

機能: (HL) A, (HL + 1) X

Aレジスタの内容をレジスタ・ペアHLでアドレスされるデータ・メモリに転送し、Xレジスタの内容をメモリの次の番地に転送します。

ただし、Lレジスタの内容が奇数の場合は、最下位ビットを無視したアドレスが指定されます。

## ○ MOV A, mem

機能: A (mem) mem = D7-0 : 00H-FFH

8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容をAレジスタに転送します。

## ○ MOV XA, mem

機能：A (mem), X (mem + 1) mem = D7-0 : 00H-FEH

8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容をAレジスタに、その次の番地の内容をXレジスタに転送します。

memで指定できるアドレスは偶数アドレスです。

## 応用例

40H, 41H番地のデータをレジスタ・ペアXAに転送する。

```
MOV XA, 40H
```

## ○ MOV mem, A

機能：(mem) A mem = D7-0 : 00H-FFH

Aレジスタの内容を8ビット・イミディエト・データmemでアドレスされるデータ・メモリに転送します。

## ○ MOV mem, XA

機能：(mem) A, (mem + 1) X mem = D7-0 : 00H-FEH

Aレジスタの内容を8ビット・イミディエト・データmemでアドレスされるデータ・メモリに転送し、Xレジスタの内容をメモリの次の番地に転送します。

memで指定できるアドレスは偶数アドレスです。

## ○ MOV A, reg

機能：A reg

レジスタreg ( X, A, H, L, D, E, B, C ) の内容をAレジスタに転送します。

 MOV XA, rp'

機能 : XA rp'

レジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容をレジスタ・ペアXAへ転送します。

応用例

レジスタ・ペアXA'のデータをレジスタ・ペアXAに転送する。

```
MOV XA, XA'
```

 MOV reg1, A

機能 : reg1 A

Aレジスタの内容をレジスタreg1 (X, H, L, D, E, B, C) に転送します。

 MOV rp'1, XA

機能 : rp'1 XA

レジスタ・ペアXAの内容をレジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') に転送します。

## ○ XCH A, @HL

機能: A (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換します。

## ○ XCH A, @HL +

機能: A (HL), L L + 1  
skip if L = 0H

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換します。その後Lレジスタの内容を自動的に+ 1し、その結果が0Hになると、続く1命令をスキップします。

## ○ XCH A, @HL -

機能: A (HL), L L - 1  
skip if L = FH

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換します。その後Lレジスタの内容を自動的に- 1し、その結果がFHになると、続く1命令をスキップします。

## ○ XCH A, @rpa1

**機能** : A (rpa)

rpa = HL + のとき : skip if L = 0

rpa = HL - のとき : skip if L = FH

Aレジスタの内容とレジスタ・ペアrpa (HL, HL+, HL-, DE, DL) でアドレスされるデータ・メモリの内容を交換します。rpaとしてオートインクリメント (HL+) / オートデクリメント (HL-) を指定した場合は、データの交換後、Lレジスタの内容を自動的に+1 / -1し、その結果がHL+の場合は0に、HL-の場合はFHになると続く1命令をスキップします。

### 応用例

データ・メモリの20H-2FH番地のデータと30H-3FH番地のデータを交換する。

```
SEL MB0
MOV D, #2
MOV HL, #30H
LOOP : XCH A, @HL ; A ( 3 x )
      XCH A, @DL ; A ( 2 x )
      XCH A, @HL ; A ( 3 x )
BR LOOP
```

## ○ XCH XA, @HL

**機能** : A (HL), X (HL+1)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換し、Xレジスタの内容とメモリの次の番地の内容を交換します。

ただし、Lレジスタの内容が奇数の場合は、最下位ビットを無視したアドレスが指定されます。

## ○ XCH A, mem

**機能** : A (mem) mem = D7-0 : 00H-FEH

Aレジスタの内容と8ビット・イミディエイト・データmemでアドレスされるデータ・メモリの内容を交換します。

## ○ XCH XA, mem

機能: A (mem), X (mem + 1) mem = D7-0 : 00H-FEH

Aレジスタの内容と8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容を交換し、Xレジスタの内容とメモリの次の番地の内容を交換します。

memで指定できるアドレスは偶数アドレスです。

## ○ XCH A, reg1

機能: A reg1

Aレジスタの内容とレジスタreg1 ( X, H, L, D, E, B, C ) の内容を交換します。

## ○ XCH XA, rp'

機能: XA rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容を交換します。

## 11.4.2 テーブル参照命令

○ MOV<sub>T</sub> XA, @PCDE

**機能:**  $\mu$ PD750066, 750068の場合 XA ROM (PC<sub>12-8</sub> + DE)

プログラム・カウンタ (PC) の下位 8 ビット (PC<sub>7-0</sub>) をレジスタ・ペア DE の内容で置き換えたときにアドレスされるプログラム・メモリ内のテーブル・データの下位 4 ビットを A レジスタに, 上位 4 ビットを X レジスタに転送します。

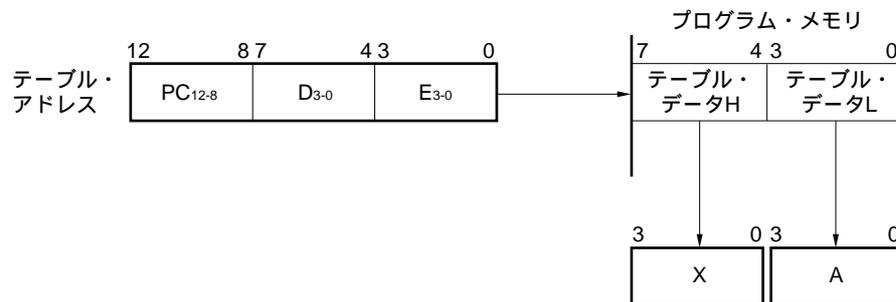
テーブル・アドレスはこの命令実行時のプログラム・カウンタ (PC) の内容によって決定されます。

テーブル領域には, 必要なデータをあらかじめアセンブラ疑似命令 (DB 命令) によりプログラムしておく必要があります。

プログラム・カウンタはこの命令の実行により影響を受けません。

この命令は, テーブル・データを連続して参照する場合に有効です。

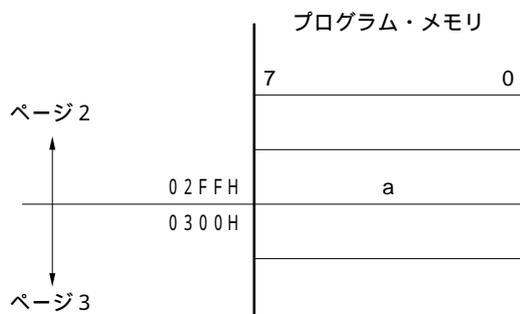
**例**  $\mu$ PD750066, 750068の場合



**備考** この項で示されている機能は, プログラム・カウンタが13ビットの $\mu$ PD750066, 750068を例にして説明してあります。 $\mu$ PD750064のプログラム・カウンタは12ビット,  $\mu$ PD75P0076のプログラム・カウンタは14ビットですので, それぞれ読み替えてお使いください。

**注 意**

MOVT XA, @PCDE命令は、通常、その命令のあるページのテーブル・データを参照しますが、命令が  $x \times FFH$  番地にある場合は、そのページ内のテーブル・データではなく次のページのテーブル・データを参照します。



たとえば、上図のように a の位置に MOVT XA, @PCDE 命令がある場合は、ページ 2 ではなくページ 3 の、レジスタ・ペア DE の内容により指定されるテーブル・データがレジスタ・ペア XA に転送されます。

**応用例**

プログラム・メモリの  $x \times F0H$ - $x \times FFH$  番地の 16 バイトのデータを、データ・メモリの  $30H$ - $4FH$  番地に転送する。

```

SUB:  SEL  MB0
      MOV  HL, #30H      ; HL  30H
      MOV  DE, #0F0H    ; DE  F0H
LOOP: MOVT XA, @PCDE    ; XA  テーブル・データ
      MOV  @HL, XA      ; (HL) XA
      INCS HL           ; HL  HL + 2
      INCS HL
      INCS E            ; E   E + 1
      BR   LOOP
      RET
      ORG   $x \times F0H$ 
      DB    $x \times H, x \times H, \dots$  ; テーブル・データ
    
```

## ○ MOVT XA, @PCXA

**機能:**  $\mu$ PD750066, 750068の場合 XA ROM ( $PC_{12-8} + XA$ )

プログラム・カウンタ (PC) の下位 8 ビット ( $PC_{7-0}$ ) をレジスタ・ペアXAの内容で置き換えたときアドレスされるプログラム・メモリ内のテーブル・データの下位 4 ビットをAレジスタに, 上位 4 ビットをXレジスタに転送します。

テーブル・アドレスはこの命令実行時のPCの内容によって決まります。

テーブル領域には, 必要なデータをあらかじめアセンブラ疑似命令 (DB命令) によりプログラムしておく必要があります。

PCは, この命令の実行により影響を受けません。

### 注 意

MOV<sub>T</sub> XA, @PCDEと同様に, 命令が  $x \times FFH$  番地にある場合は, 次のページのテーブル・データが転送されます。

**備考** この項で示されている機能は, プログラム・カウンタが13ビットの $\mu$ PD750066, 750068を例にして説明してあります。 $\mu$ PD750064のプログラム・カウンタは12ビット,  $\mu$ PD75P0076のプログラム・カウンタは14ビットですので, それぞれ読み替えてお使いください。

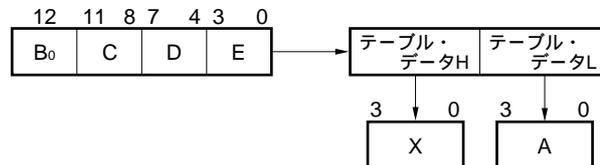
## ○ MOVT XA, @BCDE

機能：μPD750066, 750068の場合 XA (BCDE)<sub>ROM</sub>

レジスタBの下位1ビットとレジスタC, D, Eの内容でアドレスされるプログラム・メモリ内のテーブル・データ(8ビット)の下位4ビットをAレジスタに, 上位4ビットをXレジスタに転送します。

テーブル領域には必要なデータをあらかじめアセンブラ疑似命令(DB命令)によりプログラムしておく必要があります。PCは, この命令実行により影響を受けません。

例



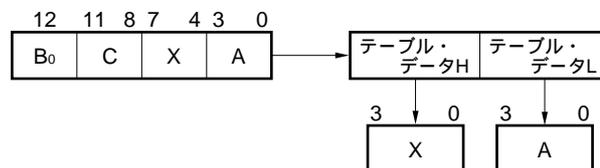
## ○ MOVT XA, @BCXA

機能：μPD750066, 750068の場合 XA (BCXA)<sub>ROM</sub>

レジスタBの下位1ビットとレジスタC, X, Aの内容でアドレスされるプログラム・メモリ内のテーブル・データ(8ビット)の下位4ビットをAレジスタに, 上位4ビットをXレジスタに転送します。

テーブル領域には必要なデータをあらかじめアセンブラ疑似命令(DB命令)によりプログラムしておく必要があります。PCは, この命令実行により影響を受けません。

例



**備考** この項で示されている機能は, プログラム・カウンタが13ビットのμPD750066, 750068を例にして説明してあります。μPD750064のプログラム・カウンタは12ビット, μPD75P0076のプログラム・カウンタは14ビットですので, それぞれ読み替えてお使いください。

## 11.4.3 ビット転送命令

- MOV1 CY, fmem.bit
- MOV1 CY, pmem.@L
- MOV1 CY, @H + mem.bit

**機能:** CY ( オペランドにより指定されたビット )

ビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) により指定されるデータ・メモリ・ビットの内容をキャリー・フラグ ( CY ) に転送します。

- MOV1 fmem.bit, CY
- MOV1 pmem.@L, CY
- MOV1 @H + mem.bit, CY

**機能:** ( オペランドで指定されるビット ) CY

キャリー・フラグ ( CY ) の内容をビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) により指定されるデータ・メモリ・ビットに転送します。

**応用例**

データ・メモリの3FH番地のビット3のフラグを、ポート3のビット2に出力する。

```

FLAG EQU 3FH.3
SEL MB0
MOV H, #FLAG SHR6 ; H FLAGの上位4ビット
MOV1 CY, @H+FLAG ; CY FLAG
MOV1 PORT3.2, CY ; P32 CY

```

## 11.4.4 演算命令

## ○ ADDS A, #n4

機能 : A A + n4 ; Skip if carry. n4 = l3-0 : 0-FH

Aレジスタの内容に4ビット・イミディエト・データn4を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

ADDC A, @HL命令およびSUBC A, @HL命令と組み合わせると、進数補正命令となります(11.1.4 進数補正命令を参照)。

## ○ ADDS XA, #n8

機能 : XA XA + n8 ; Skip if carry. n8 = l7-0 : 00H-FFH

レジスタ・ペアXAの内容に8ビット・イミディエト・データn8を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS A, @HL

機能 : A A + (HL) ; Skip if carry.

Aレジスタの内容にレジスタ・ペアHLでアドレスされるデータ・メモリの内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS XA, rp'

機能 : XA XA + rp' ; Skip if carry.

レジスタ・ペアXAの内容にレジスタ・ペアrp' (XA, HL, DE, BC, XA' HL', DE', BC') の内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS rp'1, XA

**機能** : rp'    rp'1 + XA ; Skip if carry.

レジスタ・ペアrp'1 ( HL, DE, BC, XA', HL', DE', BC' ) の内容にレジスタ・ペアXAの内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

### 応用例

レジスタ・ペアを左シフトする。

```
MOV   XA, rp'1
ADDS  rp'1, XA
NOP
```

## ○ ADDC A, @HL

**機能** : A, CY    A + ( HL ) + CY

Aレジスタの内容にレジスタ・ペアHLでアドレスされるデータ・メモリの内容をキャリー・フラグも含めて2進加算します。加算の結果キャリーが出たらキャリー・フラグがセットされ、キャリーが出ないときはキャリー・フラグはリセットされます。

この命令の次にADDS A, #n4命令が置かれた場合、この命令にキャリーが発生するとADDS A, #n4命令をスキップします。キャリーが発生しないときはADDS A, #n4命令を実行し、ADDS A, #n4命令のスキップ機能を禁止するという機能が生じます。したがって、これらの命令の組み合わせを進数補正に利用できます ( 11.1.4 進数補正命令を参照 ) 。

## ○ ADDC XA, rp'

**機能** : XA, CY    XA + rp' + CY

レジスタ・ペアXAの内容にレジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をキャリー・フラグも含めて2進加算します。加算の結果キャリーが出たとき、キャリー・フラグはセットされ、キャリーが出ないとき、キャリー・フラグはリセットされます。

## ○ ADDC rp'1, XA

機能 : rp'1, CY     $rp'1 + XA + CY$

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容にレジスタ・ペアXAの内容をキャリー・フラグを含めて2進加算します。加算の結果キャリーが出たらキャリー・フラグがセットされます。キャリーが出ないときはキャリー・フラグはリセットされます。

## ○ SUBS A, @HL

機能 : A     $A - (HL)$  ; Skip if borrow

Aレジスタの内容からレジスタ・ペアHLでアドレスされるデータ・メモリの内容を減算し、結果をAレジスタにセットします。減算の結果ボローが出ると続く1命令をスキップします。

キャリー・フラグには影響を与えません。

## ○ SUBS XA, rp'

機能 : XA     $XA - rp'$  ; Skip if borrow

レジスタ・ペアXAの内容からレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容を減算し、結果をレジスタ・ペアXAにセットします。減算の結果ボローが出ると続く1命令をスキップします。

キャリー・フラグには影響を与えません。

## 応用例

データ・メモリとレジスタ・ペアの大小比較をする。

```
MOV   XA, mem
SUBS  XA, rp'
      ; (mem)   rp'
      ; (mem) < rp'
```

## ○ SUBS rp'1, XA

**機能** : rp'1    rp'1 - XA ; Skip if borrow

レジスタ・ペアrp'1 ( HL, DE, BC, XA', HL', DE', BC' ) の内容からレジスタ・ペアXAの内容を減算し、結果を指定されたレジスタ・ペアrp'1にセットします。減算の結果ポローが出ると続く1命令をスキップします。

キャリー・フラグには影響を与えません。

## ○ SUBC A, @HL

**機能** : A, CY    A - (HL) - CY

Aレジスタの内容からレジスタ・ペアHLでアドレスされるデータ・メモリの内容をキャリー・フラグを含めて減算し、結果をAレジスタにセットします。減算の結果ポローが出るとき、キャリー・フラグがセットされ、ポローが出ないときキャリー・フラグはリセットされます。

この命令の次にADDS A, #n4命令が置かれた場合、この命令にポローが出なければADDS A, #n4命令をスキップします。ポローが出るとADDS A, #n4命令を実行し、ADDS A, #n4命令のスキップ機能を禁止するという機能が生じます。したがって、これらの命令の組み合わせを進数補正に利用できます (11.1.4 進数補正命令を参照)。

## ○ SUBC XA, rp'

**機能** : XA, CY    XA - rp' - CY

レジスタ・ペアXAの内容からレジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をキャリー・フラグを含めて減算し、結果をレジスタ・ペアXAにセットします。減算の結果ポローが出るとキャリー・フラグがセットされ、ポローが出ないときはキャリー・フラグはリセットされます。

## ○ SUBC rp'1, XA

機能 : rp'1, CY    rp'1 - XA - CY

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容からレジスタ・ペアXAの内容をキャリー・フラグを含めて減算し、結果を指定されたレジスタ・ペアrp'1にセットされます。減算の結果ポローが出るとキャリー・フラグがセットされ、ポローが出ないときはキャリー・フラグはリセットされます。

## ○ AND A, #n4

機能 : A    A    n4    n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミディエト・データn4との論理積をとり、結果をAレジスタにセットします。

## 応用例

アキュムレータの上位2ビットを0にする。

```
AND A, #0011B
```

## ○ AND A, @HL

機能 : A    A    (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容との論理積をとり、結果をAレジスタにセットします。

## ○ AND XA, rp'

機能 : XA    XA    rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容との論理積をとり、結果をレジスタ・ペアXAにセットします。

## ○ AND rp'1, XA

機能: rp'1 rp'1 XA

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容とレジスタ・ペアXAの内容との論理積をとり、結果を指定されたレジスタ・ペアにセットします。

## ○ OR A, #n4

機能: A A n4 n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミューディエト・データn4との論理和をとり、結果をAレジスタにセットします。

## 応用例

アキュムレータの下位3ビットを1にする。

```
OR A, #0111B
```

## ○ OR A, @HL

機能: A A (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容と論理和をとり、結果をAレジスタにセットします。

## ○ OR XA, rp'

機能: XA XA rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容との論理和をとり、結果をレジスタ・ペアXAにセットします。

## ○ OR rp'1, XA

機能 : rp'1 rp'1 XA

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL' DE', BC') の内容とレジスタ・ペアXAの内容との論理和をとり、結果をレジスタ・ペアrp'1にセットします。

## ○ XOR A, #n4

機能 : A A  $\forall$  n4 n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミューディエト・データn4との排他的論理和をとり、結果をAレジスタにセットします。

## 応用例

アキュムレータの最上位4ビットを反転する。

```
XOR A, #1000B
```

## ○ XOR A, @HL

機能 : A A  $\forall$  (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容と排他的論理和をとり、結果をAレジスタにセットします。

## ○ XOR XA, rp'

機能 : XA XA  $\forall$  rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容との排他的論理和をとり、結果をレジスタ・ペアXAにセットします。

 XOR rp'1, XA

**機能** : rp'1 ← rp'1 ∨ XA

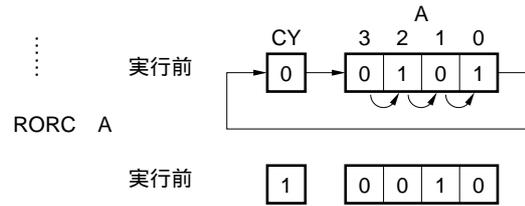
レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容とレジスタ・ペアXAの内容との排他的論理和をとり、結果をレジスタ・ペアrp'1にセットします。

## 11.4.5 アキュムレータ操作命令

## ○ RORC A

機能: CY A<sub>0</sub>, A<sub>n-1</sub> A<sub>n</sub>, A<sub>3</sub> CY (n = 1-3)

Aレジスタ(4ビット・アキュムレータ)の内容をキャリー・フラグを含めて1ビットずつ右回転します。



## ○ NOT A

機能: A  $\bar{A}$

Aレジスタ(4ビット・アキュムレータ)の1の補数をとります(各ビットを反転させます)。

## 11.4.6 増減命令

## ○ INCS reg

機能 : reg reg + 1 ; Skip if reg = 0

レジスタreg ( X, A, H, L, D, E, B, C ) の内容をインクリメントします。インクリメントの結果reg = 0になると続く1命令をスキップします。

## ○ INCS rp1

機能 : rp1 rp1 + 1 ; Skip if rp1 = 00H

レジスタ・ペアrp1 ( HL, DE, BC ) の内容をインクリメントします。インクリメントの結果rp1=00Hになると続く1命令をスキップします。

## ○ INCS @HL

機能 : ( HL ) ( HL ) + 1 ; Skip if ( HL ) = 0

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をインクリメントします。インクリメントの結果そのデータ・メモリの内容が0になると続く1命令をスキップします。

## ○ INCS mem

機能 : ( mem ) ( mem ) + 1 ; Skip if ( mem ) = 0, mem = D7-0 : 00H-FFH

8ビット・イミディエイト・データmemでアドレスされるデータ・メモリの内容をインクリメントします。インクリメントの結果、そのデータ・メモリの内容が0になると続く1命令をスキップします。

## ○ DECS reg

機能 : reg    reg - 1 ; Skip if reg = FH

レジスタreg ( X, A, H, L, D, E, B, C ) の内容をデクリメントします。デクリメントの結果reg = FHになると、続く1命令をスキップします。

## ○ DECS rp'

機能 : rp'    rp' - 1 ; Skip if rp' = FFH

レジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をデクリメントします。デクリメントの結果rp' = FFHになると、続く1命令をスキップします。

## 11.4.7 比較命令

○ SKE reg, #n4

機能 : Skip if reg = n4    n4 = I3-0 : 0-FH

レジスタreg (X, A, H, L, D, E, B, C) の内容と4ビット・イミディエト・データn4が等しければ, 続く1命令をスキップします。

○ SKE @HL, #n4

機能 : Skip if (HL) = n4    n4 = I3-0 : 0-FH

レジスタ・ペアHLでアドレスされるデータ・メモリの内容と4ビット・イミディエト・データn4が等しければ, 続く1命令をスキップします。

○ SKE A, @HL

機能 : Skip if A = (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容が等しければ, 続く1命令をスキップします。

○ SKE XA, @HL

機能 : Skip if A = (HL) and X = (HL + 1)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容が等しく, かつXレジスタの内容とメモリの次の番地の内容が等しければ, 続く1命令をスキップします。

ただし, Lレジスタの内容が奇数の場合, 最下位ビットを無視したアドレスが指定されます。

 SKE A, reg

機能 : Skip if A = reg

Aレジスタの内容とレジスタreg (X, A, H, L, D, E, B, C) の内容が等しければ, 続く1命令をスキップします。

 SKE XA, rp'

機能 : Skip if XA = rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容が等しければ, 続く1命令をスキップします。

## 11.4.8 キャリー・フラグ操作命令

○ SET1 CY

機能: CY 1

キャリー・フラグをセットします。

○ CLR1 CY

機能: CY 0

キャリー・フラグをクリアします。

○ SKT CY

機能: Skip if CY = 1

キャリー・フラグが1のとき、続く1命令をスキップします。

○ NOT1 CY

機能: CY  $\overline{\text{CY}}$

キャリー・フラグを反転させます。0なら1に、1なら0になります。

## 11.4.9 メモリ・ビット操作命令

SET1 mem. bit

機能：(mem.bit) 1 mem = D7-0 : 00H-FFH, bit = B1-0 : 0-3

8ビット・イミディエト・データmemで示されるアドレスの、2ビット・イミディエト・データbitで指定されるビットをセットします。

SET1 fmem. bit

SET1 pmem. @L

SET1 @H + mem. bit

機能：(オペランドで指定されるビット) 1

ビット操作アドレッシング (fmem.bit, pmem.@L, @H + mem.bit) により指定されるデータ・メモリのビットをセットします。

CLR1 mem. bit

機能：(mem.bit) 0 mem = D7-0 : 00H-FFH, bit = B1-0 : 0-3

8ビット・イミディエト・データmemで示されるアドレスの、2ビット・イミディエト・データbitで指定されるビットをクリアします。

CLR1 fmem. bit

CLR1 pmem. @L

CLR1 @H + mem. bit

機能：(オペランドで指定されるビット) 0

ビット操作アドレッシング (fmem.bit, pmem.@L, @H + mem.bit) により指定されるデータ・メモリのビットをクリアします。

## ○ SKT mem. bit

機能 : Skip if ( mem.bit ) = 1

mem = D7-0 : 00H-FFH, bit = B1-0 : 0-3

8ビット・イミディエト・データmemで示されるアドレスの、2ビット・イミディエト・データbitで指定されるビットが1なら、続く1命令をスキップします。

## ○ SKT fmem. bit

## ○ SKT pmem. @L

## ○ SKT @H + mem. bit

機能 : Skip if ( オペランドで指定されるビット ) = 1

ビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) により指定されるデータ・メモリのビットが1ならば、続く1命令をスキップします。

## ○ SKF mem. bit

機能 : Skip if ( mem.bit ) = 0

mem = D7-0 : 00H-FFH, bit = B1-0 : 0-3

8ビット・イミディエト・データmemで示されるアドレスの、2ビット・イミディエト・データbitで指定されるビットが0ならば、続く1命令をスキップします。

## ○ SKF fmem. bit

## ○ SKF pmem. @L

## ○ SKF @H + mem. bit

機能 : Skip if ( オペランドで指定されるビット ) = 0

ビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) により指定されるデータ・メモリのビットの内容が0ならば、続く1命令をスキップします。

- SKTCLR fmem. bit
- SKTCLR pmem. @L
- SKTCLR @H + mem. bit

機能 : Skip if (オペランドで指定されるビット) = 1 then clear

ビット操作アドレッシング (fmem.bit, pmem.@L, @H + mem.bit) で指定されるデータ・メモリのビットが1なら, 続く1命令をスキップして, そのビットを“0”にクリアします。

- AND1 CY, fmem. bit
- AND1 CY, pmem. @L
- AND1 CY, @H + mem. bit

機能 : CY CY (オペランドで指定されるビット)

キャリー・フラグの内容とビット操作アドレッシング (fmem.bit, pmem.@L, @H + mem.bit) で指定されるデータ・メモリの内容との論理積をとり, 結果をキャリー・フラグにセットします。

- OR1 CY, fmem. bit
- OR1 CY, pmem. @L
- OR1 CY, @H + mem. bit

機能 : CY CY (オペランドで指定されるビット)

キャリー・フラグの内容と, ビット操作アドレッシング (fmem.bit, pmem.@L, @H + mem.bit) で指定されるデータ・メモリのビットの内容との論理和をとり, 結果をキャリー・フラグにセットします。

- XOR1 CY, fmem. bit
- XOR1 CY, pmem. @L
- XOR1 CY, @H + mem. bit

**機能** : CY ← CY  $\nabla$  (オペランドで指定されるビット)

キャリー・フラグの内容と、ビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) で指定されるデータ・メモリのビットの内容との排他的論理和をとり、結果をキャリー・フラグにセットします。

## 11.4.10 分岐命令

## ○ BR addr

機能：μPD750068の場合 PC<sub>12-0</sub> addr  
addr = 0000H-1FFFH

イミューディエト・データaddrでアドレスされるアドレスに分岐します。

この命令はアセンブラの疑似命令であり、アセンブル時、アセンブラよりBR !addr命令、BRCB !caddr命令、BR \$addr命令の中から最適な命令に自動的に置き換えられます。

## ○ BR addr1

機能：μPD750068の場合 PC<sub>12-0</sub> addr1  
addr1 = 0000H = 1FFFH

イミューディエト・データaddr1でアドレスされるアドレスに分岐します。

この命令はアセンブラの疑似命令であり、アセンブル時、アセンブラよりBRA !addr1命令、BR !addr命令、BRCB !caddr命令、BR \$addr1命令の中から最適な命令に自動的に置き換えられます。

## ○ BRA !addr1

機能：μPD750068の場合 PC<sub>12-0</sub> addr1

## ○ BR !addr

機能：μPD750068の場合 PC<sub>12-0</sub> addr  
addr = 0000H-1FFFH

イミューディエト・データaddrがプログラム・カウンタ(PC)へ転送され、PCでアドレスされる番地に分岐します。

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHのμPD750068を例にして説明してあります。μPD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、μPD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、μPD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

## ○ BR \$addr

**機能**：μPD750068の場合 PC<sub>12-0</sub> addr

addr = (PC - 15) ~ (PC - 1), (PC + 2) ~ (PC + 16)

現在のアドレスから (-15 ~ -1), (+2 ~ +16) の分岐範囲を持つ相対分岐命令です。ページの境界や、ブロックの境界には影響を受けません。

## ○ BR \$addr1

**機能**：μPD750068の場合 PC<sub>12-0</sub> addr1

addr1 = (PC - 15) ~ (PC - 1), (PC + 2) ~ (PC + 16)

現在のアドレスから (-15 ~ -1), (+2 ~ +16) の分岐範囲を持つ相対分岐命令です。ページの境界や、ブロックの境界には影響を受けません。

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHのμPD750068を例にして説明してあります。μPD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、μPD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、μPD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

## ○ BRCB !caddr

機能： $\mu$ PD750068の場合  $PC_{12-0} = PC_{12} + caddr_{11-0}$   
 $caddr = n000H-nFFFH$   
 $n = PC_{12} = 0, 1$

プログラム・カウンタの下位12ビット ( $PC_{11-0}$ ) を12ビット・イミディエト・データ  $caddr$  で置き換えた番地に分岐します。

$\mu$ PD750064はプログラム・カウンタが11ビット構成なので、この命令で全空間に分岐できます。  
 $\mu$ PD750066, 750068は $PC_{12}$ を、 $\mu$ PD75P0076は $PC_{12,13}$ を変えることはできず、ブロック内に分岐となります。

### 注 意

BRCB !caddr命令は、通常、その命令のあるブロック内に分岐しますが、1バイト目が、0FFE<sub>H</sub>番地、または0FFF<sub>H</sub>番地にある場合は、ブロック0には分岐せずブロック1に分岐しますので注意してください。



上図の a または b の位置に BRCB !caddr 命令がある場合は、ブロック0ではなくブロック1に分岐します。

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、 $addr = 0000H-1FFFH$  の  $\mu$ PD750068 を例にして説明してあります。 $\mu$ PD750064はプログラム・カウンタが12ビット、 $addr = 000H-FFFH$ 、 $\mu$ PD750066はプログラム・カウンタが13ビット、 $addr = 0000H-17FFH$ 、 $\mu$ PD75P0076はプログラム・カウンタが14ビット、 $addr = 0000H-3FFFH$  ですので、それぞれ読み替えてお使いください。

## ○ BR PCDE

**機能:**  $\mu$ PD750068の場合 PC<sub>12-0</sub> PC<sub>12-8</sub> + DE  
PC<sub>7-4</sub> D, PC<sub>3-0</sub> E

プログラム・カウンタの下位8ビット(PC<sub>7-0</sub>)をレジスタ・ペアDEの内容で置き換えた番地に分岐します。プログラム・カウンタの上位のビットには影響を与えません。

### 注 意

BR PCDE命令は、通常、その命令のあるページ内に分岐しますが、命令コードの1バイト目がx xFE番地、またはx xFFH番地にある場合は、そのページ内には分岐せず次のページに分岐します。



たとえば、上図のように a または b の位置に BR PCDE 命令がある場合は、ページ 2 ではなくページ 3 の、レジスタ・ペア DE の内容により指定される下位 8 ビット・アドレスへ分岐します。

## ○ BR PCXA

**機能:**  $\mu$ PD750068の場合 PC<sub>12-0</sub> PC<sub>12-8</sub> + XA  
PC<sub>7-4</sub> X, PC<sub>3-0</sub> A

プログラム・カウンタの下位8ビット(PC<sub>7-0</sub>)をレジスタ・ペアXAの内容で置き換えた番地に分岐します。プログラム・カウンタの上位のビットには影響を与えません。

### 注 意

BR PCDE命令と同様に、1バイト目がx xFEH、またはx xFFH番地にある場合は、同一ページ内には分岐せず、次のページに分岐します。

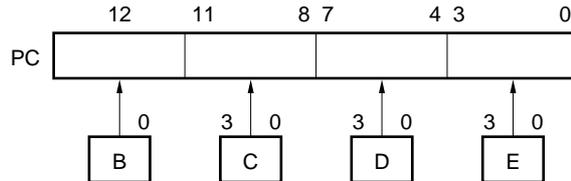
**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHの  $\mu$ PD750068を例にして説明してあります。  $\mu$ PD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、  $\mu$ PD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFH、  $\mu$ PD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

## ○ BR BCDE

機能：μPD750068の場合 PC<sub>12-0</sub> BCDE

例

プログラム・カウンタをレジスタB<sub>0</sub>, C, D, Eの内容で置き換えた番地に分岐します。

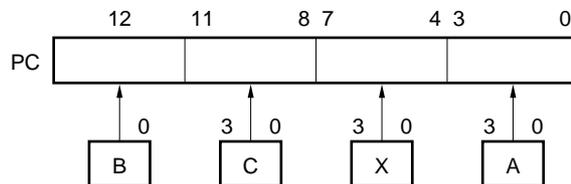


## ○ BR BCXA

機能：μPD750068の場合 PC<sub>12-0</sub> BCXA

例

プログラム・カウンタをレジスタB<sub>0</sub>, C, X, Aの内容で置き換えた番地に分岐します。



## ○ TBR addr

機能：

GETI命令のテーブル定義用アセンブラ疑似命令です。3バイトのBR !addr命令を、1バイトのGETI命令で置き換える場合に使用します。addrには12ビット・アドレス・データを記述します。詳しくは、RA75X **アセンブラ・パッケージ ユーザーズ・マニュアル 言語編 (EEU-730)**を参照してください。

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHのμPD750068を例にして説明してあります。μPD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、μPD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFH、μPD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

## 11.4.11 サブルーチン・スタック制御命令

## ○ CALLA !addr1

機能：μPD750068の場合

(SP - 2) x, x, MBE, RBE, (SP - 3) PC<sub>7-4</sub>(SP - 4) PC<sub>3-0</sub>, (SP - 5) 0, 0, 0, PC<sub>12</sub>(SP - 6) PC<sub>11-8</sub>PC<sub>12-0</sub> addr1, SP SP - 6

## / CALL !addr

機能：μPD750068の場合

【Mk モード】

(SP - 1) PC<sub>7-4</sub>, (SP - 2) PC<sub>3-0</sub>(SP - 3) MBE, RBE, 0, PC<sub>12</sub>(SP - 4) PC<sub>11-8</sub>, PC<sub>12-0</sub> addr, SP SP - 4

addr = 0000H-1FFFFH

【Mk モード】

(SP - 2) x, x, MBE, RBE

(SP - 3) PC<sub>7-4</sub>, (SP - 4) PC<sub>3-0</sub>(SP - 5) 0, 0, 0, PC<sub>12</sub>, (SP - 6) PC<sub>11-8</sub>PC<sub>12-0</sub> addr, SP SP - 6

addr = 0000H-1FFFFH

プログラム・カウンタ（リターン・アドレス）およびMBE,RBEの内容をスタック・ポインタ（SP）でアドレスされるデータ・メモリ（スタック）に退避し，SPをデクリメントしたのち，14ビット・イミューディアット・データaddrでアドレスされる番地へ分岐します。

**備考** この項で示されている機能は，プログラム・カウンタが13ビット，addr = 0000H-1FFFFHのμPD750068を例にして説明してあります。μPD750064はプログラム・カウンタが12ビット，addr = 000H-FFFFH，μPD750066はプログラム・カウンタが13ビット，addr = 0000H-17FFFH，μPD75P0076はプログラム・カウンタが14ビット，addr = 0000H-3FFFFHですので，それぞれ読み替えてお使いください。

## ① CALLF !faddr

**機能**：μPD750068の場合

**【Mk モード】**

(SP - 1) PC<sub>7-4</sub>, (SP - 2) PC<sub>3-0</sub>

(SP - 3) MBE, RBE, 0, PC<sub>12</sub>

(SP - 4) PC<sub>11-8</sub>, SP - 4

PC<sub>12-0</sub> 00 + faddr

faddr = 0000H-07FFH

**【Mk モード】**

(SP - 2) x, x, MBE, RBE

(SP - 3) PC<sub>7-4</sub>, (SP - 4) PC<sub>3-0</sub>

(SP - 5) 0, 0, 0, PC<sub>12</sub>, (SP - 6) PC<sub>11-8</sub>

SP - 6

PC<sub>12-0</sub> 00 + faddr

faddr = 0000H-07FFH

プログラム・カウンタ (PC ; リターン・アドレス) および MBE, RBE の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, SP をデクリメントしたのち, 11 ビット・イミディエト・データ faddr でアドレスされる番地へ分岐します。コールできる範囲は 0000H-07FFH (0-2047) 番地に限られます。

## ○ TCALL !addr

**機能**

GETI 命令のテーブル定義用アセンブラ疑似命令です。3 バイトの CALL !addr 命令を, 1 バイトの GETI 命令で置き換える場合に使用します。addr には 12 ビット・アドレス・データを記述します。詳しくは, RA75X **アセンブラ・パッケージ ユーザーズ・マニュアル 言語編 (EEU-730)** を参照してください。

**備考** この項で示されている機能は, プログラム・カウンタが 13 ビット, addr = 0000H-1FFFH の μPD750068 を例にして説明してあります。μPD750064 はプログラム・カウンタが 12 ビット, addr = 000H-FFFH, μPD750066 はプログラム・カウンタが 13 ビット, addr = 0000H-17FFFH, μPD75P0076 はプログラム・カウンタが 14 ビット, addr = 0000H-3FFFH です。で, それぞれ読み替えてお使いください。

## ① RET

**機能:**  $\mu$ PD750068の場合

【Mk モード】 PC<sub>11-8</sub> (SP), MBE, RBE, 0, PC<sub>12</sub> (SP + 1)  
 PC<sub>3-0</sub> (SP + 2)  
 PC<sub>7-4</sub> (SP + 3), SP SP + 4

【Mk モード】 PC<sub>11-8</sub> (SP), 0, 0, 0, PC<sub>12</sub> (SP + 1)  
 PC<sub>3-0</sub> (SP + 2), PC<sub>7-4</sub> (SP + 3),  
 x, x, MBE, RBE (SP + 4), SP SP + 6

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をプログラム・カウンタ (PC), メモリ・バンク許可フラグ (MBE), およびレジスタ・バンク許可フラグ (RBE) にリストアし, そのあとSPの内容をインクリメントします。

**注意**

プログラム・ステータス・ワード (PSW) は, MBE, RBE以外リストアされません。

## ① RETS

**機能:**  $\mu$ PD750068の場合

【Mk モード】 PC<sub>11-8</sub> (SP), MBE, 0, 0, PC<sub>12</sub> (SP + 1)  
 PC<sub>3-0</sub> (SP + 2), PC<sub>7-4</sub> (SP + 3), SP SP + 4  
 Then skip unconditionally

【Mk モード】 PC<sub>11-8</sub> (SP), 0, 0, 0, PC<sub>12</sub> (SP + 1)  
 PC<sub>3-0</sub> (SP + 2), PC<sub>7-4</sub> (SP + 3)  
 x, x, MBE, RBE (SP + 4), SP SP + 6  
 Then skip unconditionally

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をプログラム・カウンタ (PC), メモリ・バンク許可フラグ (MBE), およびレジスタ・バンク許可フラグ (RBE) にリストアし, SPの内容をインクリメント後, 無条件にスキップします。

**注意**

プログラム・ステータス・ワード (PSW) は, MBE, RBE以外リストアされません。

**備考** この項で示されている機能は, プログラム・カウンタが13ビット, addr = 0000H-1FFFHの  $\mu$ PD750068を例にして説明してあります。  $\mu$ PD750064はプログラム・カウンタが12ビット, addr = 000H-FFFH,  $\mu$ PD750066はプログラム・カウンタが13ビット, addr = 0000H-17FFFH,  $\mu$ PD75P0076はプログラム・カウンタが14ビット, addr = 0000H-3FFFHですので, それぞれ読み替えてお使いください。

## ① RETI

**機能：**  $\mu$  PD750068の場合

【Mk モード】  $PC_{11-8}$  (SP), MBE, RBE, 0,  $PC_{12}$  (SP + 1),  
 $PC_{3-0}$  (SP + 2),  $PC_{7-4}$  (SP + 3)  
 PSWL (SP + 4), PSWH (SP + 5)  
 SP SP + 6

【Mk モード】  $PC_{11-8}$  (SP), 0, 0, 0,  $PC_{12}$  (SP + 1)  
 $PC_{3-0}$  (SP + 2),  $PC_{7-4}$  (SP + 3)  
 PSWL (SP + 4), PSWH (SP + 5)  
 SP SP + 6

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をプログラム・カウンタ (PC), およびプログラム・ステータス・ワードにリストアし, そのあとSPの内容をインクリメントします。

割り込み処理ルーチンからの復帰時に使用します。

**備考** この項で示されている機能は, プログラム・カウンタが13ビット,  $addr = 0000H-1FFFH$ の  $\mu$  PD750068を例にして説明してあります。  $\mu$  PD750064はプログラム・カウンタが12ビット,  $addr = 000H-FFFH$ ,  $\mu$  PD750066はプログラム・カウンタが13ビット,  $addr = 0000H-17FFFH$ ,  $\mu$  PD75P0076はプログラム・カウンタが14ビット,  $addr = 0000H-3FFFH$ ですので, それぞれ読み替えてお使いください。

## ○ PUSH rp

**機能:** (SP - 1) rp<sub>H</sub>, (SP - 2) rp<sub>L</sub>, SP SP - 2

レジスタ・ペアrp (XA, HL, DE, BC) の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, そのあとSPをデクリメントします。

レジスタ・ペアの上位側 (rp<sub>H</sub>: X, H, D, B) は (SP - 1) で, 下位側 (rp<sub>L</sub>: A, L, E, C) は (SP - 2) でアドレスされるスタックに退避されます。

## ○ PUSH BS

**機能:** (SP - 1) MBS, (SP - 2) RBS, SP SP - 2

メモリ・バンク選択レジスタ (MBS), およびレジスタ・バンク選択レジスタ (RBS) の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, そのあとSPをデクリメントします。

## ○ POP rp

**機能:** rp<sub>L</sub> (SP), rp<sub>H</sub> (SP + 1), SP SP + 2

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をレジスタ・ペアrp (XA, HL, DE, BC) へ復帰し, そのあとSPをインクリメントします。

レジスタ・ペアの下位側 (rp<sub>L</sub>: A, L, E, C) には (SP) の内容が, 上位側 (rp<sub>H</sub>: X, H, D, B) には (SP + 1) の内容がリストアされます。

## ○ POP BS

**機能:** RBS (SP), MBS (SP + 1), SP SP + 2

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をレジスタ・バンク選択レジスタ (RBS), メモリ・バンク選択レジスタ (MBS) へ復帰し, そのあとSPをインクリメントします。

## 11.4.12 割り込み制御命令

○ EI

機能：IME (IPS.3) 1

割り込みマスタ許可フラグ (割り込みプライオリティ選択レジスタのビット3) を, “ 1 ” にセットし, 割り込みを許可します。割り込み受け付けの可否は, 各割り込み許可フラグにより制御されます。

○ EI IE × × ×

機能：IE × × × 1 × × × = N<sub>5</sub>, N<sub>2-0</sub>

割り込み許可フラグ (IE × × ×) を “ 1 ” にセットし, 割り込みを受け付け可能状態にします。  
( × × × = BT, CSI, T0, T1, W, 0, 1, 2, 4)

○ DI

機能：IME (IPS.3) 0

割り込みマスタ許可フラグ (割り込みプライオリティ選択レジスタのビット3) を “ 0 ” にリセットし, 各割り込み許可フラグの内容に関係なく, すべての割り込みを禁止します。

○ DI IE × × ×

機能：IE × × × 0 × × × = N<sub>5</sub>, N<sub>2-0</sub>

割り込み許可フラグ (IE × × ×) を “ 0 ” にリセットし, 割り込みの受け付けを禁止します。  
( × × × = BT, CSI, T0, T1, W, 0, 1, 2, 4)

## 11.4.13 入出力命令

 IN A, PORTn

機能 : A PORTn n = N<sub>3-0</sub> : 0-6, 11

PORTn (n = 0-6, 11) で指定されるポートの内容をAレジスタに転送します。

注 意

この命令実行時には、MBE = 0 とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。n には0-6, 11のみ指定できます。

入出力モードの指定により、出力ラッチのデータ (出力モード) または、端子のデータ (入力モード) が取り込まれます。

 IN XA, PORTn

機能 : A PORTn, X PORTn+1 n = N<sub>3-0</sub> : 4

PORTn (n = 4) で指定されるポートの内容をAレジスタに転送し、次のポートの内容をXレジスタに転送します。

注 意

n としては、4のみ指定できます。この命令実行時には、MBE = 0 とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。

入出力モードの指定により、出力ラッチのデータ (出力モード) または、端子のデータ (入力モード) が取り込まれます。

## ○ OUT PORTn, A

機能: PORTn A n = N<sub>3-0</sub> : 2-6

Aレジスタの内容をPORTn (n = 2-6) で指定されるポートの出力ラッチへ転送します。

### 注 意

この命令実行時には、MBE = 0 とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。

nには2-6のみ指定できます。

## ○ OUT PORTn, XA

機能: PORTn A, PORTn+1 X n = N<sub>3-0</sub> : 4

Aレジスタの内容をPORTn (n = 4) で指定されるポートの出力ラッチへ転送し、Xレジスタの内容を次のポートの出力ラッチへ転送します。

### 注 意

この命令実行時には、MBE = 0 とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。

nとしては、4のみ指定できます。

## 11.4.14 CPU制御命令

 HALT

機能：PCC.2 1

HALTモードをセットします（プロセッサ・クロック・コントロール・レジスタのビット2をセットする命令です）。

**注 意**

HALT命令に続く1命令は必ずNOP命令にしてください。

 STOP

機能：PCC.3 1

STOPモードをセットします（プロセッサ・クロック・コントロール・レジスタのビット3をセットする命令です）。

**注 意**

STOP命令に続く1命令は必ずNOP命令にしてください。

 NOP

機能：何もせず1マシン・サイクル費やします。

## 11.4.15 特殊命令

○ SEL R<sub>Bn</sub>機能：RBS n n = N<sub>1-0</sub> : 0-3

2ビット・イミディエト・データ n をレジスタ・バンク選択レジスタ (RBS) にセットします。

○ SEL M<sub>Bn</sub>機能：MBS n n = N<sub>3-0</sub> : 0, 1, 15

4ビット・イミディエト・データ n をメモリ・バンク選択レジスタ (MBS) に転送します。

## / GETI taddr

機能：μ PD750068の場合

taddr = T<sub>5-0</sub>, 0 : 20H-7FH

【Mk モード】

TBR命令で定義したテーブルを参照した場合

PC<sub>12-0</sub> (taddr)<sub>4-0</sub> + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合

(SP - 1) PC<sub>7-4</sub>, (SP - 2) PC<sub>3-0</sub>(SP - 3) MBE, RBE, 0, PC<sub>12</sub>(SP - 4) PC<sub>11-8</sub>PC<sub>12-0</sub> (taddr)<sub>4-0</sub> + (taddr + 1)

SP SP - 4

TBR, TCALL命令以外で定義したテーブルを参照した場合

(taddr)(taddr + 1) を命令コードとする命令を実行

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHのμ PD750068を例にして説明してあります。μ PD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、μ PD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、μ PD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

## 【Mk モード】

TBR命令で定義したテーブルを参照した場合<sup>注</sup>

PC<sub>12-0</sub> (taddr)<sub>4-0</sub> + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合<sup>注</sup>

(SP - 2) x, x, MBE, RBE

(SP - 3) PC<sub>7-4</sub>, (SP - 4) PC<sub>3-0</sub>

(SP - 5) 0, 0, 0, PC<sub>12</sub>, (SP - 6) PC<sub>11-8</sub>

PC<sub>12-0</sub> (taddr)<sub>4-0</sub> + (taddr + 1), SP SP - 6

TBR, TCALL命令以外で定義したテーブルを参照した場合

(taddr)(taddr + 1)を命令コードとする命令を実行

**注** TBR, TCALL命令で指定されるアドレスは0000H-3FFFHに限ります。

(taddr), (taddr + 1)で指定されるプログラム・メモリ・アドレスにある2バイトのデータを参照し、命令として実行します。

参照テーブルの領域は0020H-007FHで、あらかじめデータを書き込んでおきます。データの書き込みは、1バイト命令、2バイト命令の場合は、二モニックをそのまま記述します。

3バイト・コール命令および3バイト分岐命令の場合は、アセンブラ疑似命令(TCALL, TBR)により記述します。

taddrとして指定できるアドレスは偶数アドレスのみです。

**備考** この項で示されている機能は、プログラム・カウンタが13ビット、addr = 0000H-1FFFHのμPD750068を例にして説明してあります。μPD750064はプログラム・カウンタが12ビット、addr = 000H-FFFH、μPD750066はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、μPD75P0076はプログラム・カウンタが14ビット、addr = 0000H-3FFFHですので、それぞれ読み替えてお使いください。

**注 意**

参照テーブルにセットできる2バイト命令は、2マシン・サイクルの命令に限られます(ただし、BRCB命令およびCALLF命令を除く)。また、1バイト命令2つをセットする場合は、次表のような組み合わせに限られます。

1バイト目の命令	2バイト目の命令															
MOV A, @HL MOV @HL, A XCH A, @HL	<table border="0"> <tr><td>[</td><td>INCS</td><td>L</td></tr> <tr><td></td><td>DECS</td><td>L</td></tr> <tr><td>[</td><td>INCS</td><td>H</td></tr> <tr><td></td><td>DECS</td><td>H</td></tr> <tr><td></td><td>INCS</td><td>HL</td></tr> </table>	[	INCS	L		DECS	L	[	INCS	H		DECS	H		INCS	HL
[	INCS	L														
	DECS	L														
[	INCS	H														
	DECS	H														
	INCS	HL														
MOV A, @DE XCH A, @DE	<table border="0"> <tr><td>[</td><td>INCS</td><td>E</td></tr> <tr><td></td><td>DECS</td><td>E</td></tr> <tr><td>[</td><td>INCS</td><td>D</td></tr> <tr><td></td><td>DECS</td><td>D</td></tr> <tr><td></td><td>INCS</td><td>DE</td></tr> </table>	[	INCS	E		DECS	E	[	INCS	D		DECS	D		INCS	DE
[	INCS	E														
	DECS	E														
[	INCS	D														
	DECS	D														
	INCS	DE														
MOV A, @DL XCH A, @DL	<table border="0"> <tr><td>[</td><td>INCS</td><td>L</td></tr> <tr><td></td><td>DECS</td><td>L</td></tr> <tr><td>[</td><td>INCS</td><td>D</td></tr> <tr><td></td><td>DECS</td><td>D</td></tr> </table>	[	INCS	L		DECS	L	[	INCS	D		DECS	D			
[	INCS	L														
	DECS	L														
[	INCS	D														
	DECS	D														

PCは、GETI命令実行中はインクリメントは行いませんので、参照命令実行後、GETI命令の次の番地から処理を続けます。

GETI命令の前の命令がスキップ機能を持つ場合、GETI命令は他の1バイト命令と同様にスキップされます。また、GETI命令で参照した命令がスキップ機能を持つ場合、GETI命令の次に続く命令はスキップされます。

GETI命令でたてづみ効果を持つ命令を参照した場合は次のように実行されます。

GETI命令の前の命令も同じグループのたてづみ効果を持つ命令の場合、GETI命令を実行するとたてづみ効果は消え、参照した命令はスキップされません。

GETI命令の次の命令も同じグループのたてづみ効果を持つ命令の場合、参照した命令により生ずるたてづみ効果は有効で、次に続く命令はスキップされます。

## 応用例

```

{
MOV  HL, #00H
MOV  XA, #FFH
CALL SUB1
BR   SUB2
} をGETI命令に置き換える。

      ORG    20H
HL00: MOV    HL, #00H
XAFF: MOV    XA, #FFH
CSUB1: TCALL SUB1
BSUB2: TBR   SUB2
      ⋮
      GETI   HL00      ; MOV  HL, #00H
      ⋮
      GETI   BSUB2     ; BR   SUB2
      ⋮
      GETI   CSUB1     ; CALL SUB1
      ⋮
      GETI   XAFF      ; MOV  XA, #FFH
```

(メモ)

# 付録A $\mu$ PD75068, 750068, 75P0076の機能一覧表

(1/2)

項 目		$\mu$ PD75068	$\mu$ PD750068	$\mu$ PD75P0076
プログラム・メモリ		マスクROM 0000H-1F7FH (8064×8ビット)	マスクROM 0000H-1FFFH (8192×8ビット)	ワン・タイムPROM 0000H-3FFFH (16384×8ビット)
データ・メモリ		000H-1FFFH (512×4ビット)		
CPU		75X Standard CPU	75XL CPU	
汎用レジスタ		4ビット×8または8ビット×4 (4ビット×8または8ビット×4) 4バンク		
命令実行時間	メイン・システム・クロック 選択時	0.95, 1.91, 15.3 $\mu$ s (4.19 MHz動作時)	<ul style="list-style-type: none"> <li>0.67, 1.33, 2.67, 10.7 <math>\mu</math>s (6.0 MHz動作時)</li> <li>0.95, 1.91, 3.81, 15.3 <math>\mu</math>s (4.19 MHz動作時)</li> </ul>	
	サブシステム・クロック 選択時	122 $\mu$ s (32.768 kHz動作時)		
入出力ポート	CMOS入力	12本 (ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能: 7本)		
	CMOS入出力	12本 (ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能)		
	N-chオープン・ドレイン 入出力	8本 (マスク・オプションによるプルアップ抵抗内蔵可能) 10 V耐圧	8本 (マスク・オプションによるプルアップ抵抗内蔵可能) 13 V耐圧	8本 (マスク・オプションなし) 13 V耐圧
	合計	32本		
タイマ		<ul style="list-style-type: none"> <li>3チャンネル</li> <li>・8ビット・タイマ/イベント・カウンタ</li> <li>・8ビット・ベーシック・インターバル・タイマ</li> <li>・時計用タイマ</li> </ul>	<ul style="list-style-type: none"> <li>4チャンネル</li> <li>・8ビット・タイマ/イベント・カウンタ0 (時計用タイマ出力追加)</li> <li>・8ビット・タイマ/イベント・カウンタ1 (16ビット・タイマ/イベント・カウンタとして使用可能)</li> <li>・8ビット・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ</li> <li>・時計用タイマ</li> </ul>	
A/Dコンバータ		<ul style="list-style-type: none"> <li>・8ビット分解能×8チャンネル (逐次比較型)</li> <li>・<math>V_{DD} = 2.7</math> Vから動作可能</li> </ul>	<ul style="list-style-type: none"> <li>・8ビット分解能×8チャンネル (逐次比較型)</li> <li>・<math>V_{DD} = 1.8</math> Vから動作可能</li> </ul>	
クロック出力 (PCL)		<ul style="list-style-type: none"> <li>・, 524, 262, 65.5 kHz</li> <li>(メイン・システム・クロック: 4.19 MHz動作時)</li> </ul>	<ul style="list-style-type: none"> <li>・, 1.05 MHz, 262 kHz, 65.5 kHz</li> <li>(メイン・システム・クロック: 4.19 MHz動作時)</li> <li>・, 1.5 MHz, 375 kHz, 93.8 kHz</li> <li>(メイン・システム・クロック: 6.0 MHz動作時)</li> </ul>	

(2/2)

項 目	μPD75068	μPD750068	μPD75P0076
ブザー出力 (BUZ)	2, 4, 32 kHz(メイン・システム・クロック : 4.19 MHz 動作時またはサブシステム・クロック : 32.768 kHz動作時)	・ 2, 4, 32 kHz(メイン・システム・クロック : 4.19 MHz 動作時またはサブシステム・クロック : 32.768 kHz動作時) ・ 2.93, 5.86, 46.9 kHz(メイン・システム・クロック : 6.0 MHz動作時)	
シリアル・インタフェース	3種類のモードに対応可能 ・ 3線式シリアルI/Oモード ...MSB/LSB先頭切り替え可能 ・ 2線式シリアルI/Oモード ・ SBIモード	2種類のモードに対応可能 ・ 3線式シリアルI/Oモード...MSB/LSB先頭切り替え可能 ・ 2線式シリアルI/Oモード	
ベクタ割り込み	外部 : 3本, 内部 : 3本	外部 : 3本, 内部 : 4本	
テスト入力	外部 : 1本, 内部 : 1本		
電源電圧	V <sub>DD</sub> = 2.7 ~ 6.0 V	V <sub>DD</sub> = 1.8 ~ 5.5 V	
動作周囲温度	T <sub>A</sub> = - 40 ~ + 85		
パッケージ	・ 42ピン・プラスチック・シュリンクDIP (600 mil) ・ 44ピン・プラスチックQFP (10 mm)	・ 42ピン・プラスチック・シュリンクDIP (600 mil, 1.778 mmピッチ) ・ 42ピン・プラスチック・シュリンクSOP (375 mil, 0.8 mmピッチ)	

## 付録 B 開発ツール

μ PD750068を使用するシステム開発のために次のような開発ツールを用意しております。75XLシリーズでは、シリーズ共通のリロケータブル・アセンブラを品種ごとのデバイス・ファイルと組み合わせて使用します。

### 言語プロセッサ

RA75X リロケータブル・ アセンブラ	ホスト・マシン			オーダ名称 (品名)
		OS	供給媒体	
	PC-9800シリーズ	MS-DOS™	3.5インチ2HD	μ S5A13RA75X
		<div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 2px;">                     Ver.3.30                      }                      Ver.6.2<sup>注</sup> </div>	5インチ2HD	μ S5A10RA75X
	IBM PC/AT™ およびその互換機	「IBM PC用のOSにつ いて」参照	3.5インチ2HC	μ S7B13RA75X
			5インチ2HC	μ S7B10RA75X

デバイス・ファイル	ホスト・マシン			オーダ名称 (品名)
		OS	供給媒体	
	PC-9800シリーズ	MS-DOS	3.5インチ2HD	μ S5A13DF750068
		<div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 2px;">                     Ver. 3.30                      }                      Ver. 6.2<sup>注</sup> </div>	5インチ2HD	μ S5A10DF750068
	IBM PC/AT およびその互換機	「IBM PC用のOSにつ いて」参照	3.5インチ2HC	μ S7B13DF750068
			5インチ2HC	μ S7B10DF750068

**注** Ver. 5.00以降にはタスク・スワップ機能がありますが、このソフトウェアではタスク・スワップ機能は使用できません。

**備考** アセンブラ、デバイス・ファイルの動作は、上記のホスト・マシンとOS上でのみ保証されます。

PROM書き込み用ツール

ハードウェア	PG-1500	付属のボードおよび別売りのプログラマ・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、256 Kビットから4 M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。			
	PA-75P0076CU	μ PD75P0076CU, 75P0076GT用の専用PROMプログラマ・アダプタで、PG-1500に接続して使用します。			
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。			
		ホスト・マシン		オーダ名称(品名)	
			OS		供給媒体
		PC-9800シリーズ	MS-DOS	3.5インチ2HD	μ S5A13PG1500
			[ Ver.3.30 ] } [ Ver.6.2 <sup>注</sup> ]	5インチ2HD	μ S5A10PG1500
IBM PC/AT およびその互換機	「IBM PC用のOSについて」参照	3.5インチ2HD	μ S7B13PG1500		
		5インチ2HC	μ S7B10PG1500		

**注** Ver.5.00以降にはタスク・スワップ機能がありますが、このソフトウェアではタスク・スワップ機能は使用できません。

**備考** PG-1500コントローラの動作は、上記のホスト・マシンとOS上でのみ保証されます。

ディバグ用ツール

μPD750068のプログラム・ディバグ用ツールとしてインサーキット・エミュレータ (IE-75000-R, IE-75001-R) を用意しています。

それぞれのシステム構成を次に示します。

ハードウェア	IE-75000-R <sup>注1</sup>	IE-75000-Rは、75Xシリーズ、75XLシリーズを使用する応用システムを開発する際に、ハードウェア、ソフトウェアのディバグを行うためのインサーキット・エミュレータです。μPD750068サブシリーズを開発する場合、IE-75000-Rと別売りのエミュレーション・ボードIE-75300-R-EM、およびエミュレーション・プローブを組み合わせで使用します。 ホスト・マシン、PROMプログラムと接続して効率的にディバグを行うことができます。 なお、IE-75000-R内にはエミュレーション・ボードIE-75000-R-EMが含まれており、接続されています。			
	IE-75001-R	IE-75001-Rは、75Xシリーズ、75XLシリーズを使用する応用システムを開発する際に、ハードウェア、ソフトウェアのディバグを行うためのインサーキット・エミュレータです。μPD750068サブシリーズを開発する場合、IE-75001-Rと別売りのエミュレーション・ボードIE-75300-R-EM、およびエミュレーション・プローブを組み合わせで使用します。 ホスト・マシン、PROMプログラムを接続して効率的にディバグを行うことができます。			
	IE-75300-R-EM	μPD750068サブシリーズを使用する応用システムの評価を行うためのエミュレーション・ボードです。IE-75000-RまたはIE-75001-Rと組み合わせで使用します。			
	EP-750068CU-R	μPD750068CU用のエミュレーション・プローブです。 IE-75000-RまたはIE-75001-Rと、IE-75300-R-EMに接続して使用します。			
	EP-750068GT-R	μPD750068GT用のエミュレーション・プローブです。 IE-75000-RまたはIE-75001-Rと、IE-75300-R-EMに接続して使用します。			
	EV-9500GT-42	ターゲット・システムとの接続を容易にするフレキシブル基板EV-9500GT-42を添付しています。			
ソフトウェア	IEコントロール・プログラム	IE-75000-RまたはIE-75001-Rとホスト・マシンをRS-232-C、およびセントロニクスI/Fで接続し、ホスト・マシン上でIE-75000-RまたはIE-75001-Rを制御します。			
		ホスト・マシン	オーガ名称 (品名)		
			OS	供給媒体	
		PC-9800シリーズ	MS-DOS	3.5インチ2HD	μS5A13IE75X
			<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">Ver.3.30</td> </tr> <tr> <td style="text-align: center;">}</td> </tr> <tr> <td style="text-align: center;">Ver.6.2<sup>注2</sup></td> </tr> </table>	Ver.3.30	}
Ver.3.30					
}					
Ver.6.2 <sup>注2</sup>					
IBM PC/AT およびその互換機	「IBM PC用のOSについて」参照	3.5インチ2HC	μS7B13IE75X		
		5インチ2HC	μS7B10IE75X		

注1. 保守品です。

2. Ver.5.00以降にはタスク・スワップ機能がありますが、このソフトウェアではタスク・スワップ機能は使用できません。

備考 IEコントロール・プログラムの動作は、上記のホスト・マシンとOS上でのみ保証されます。

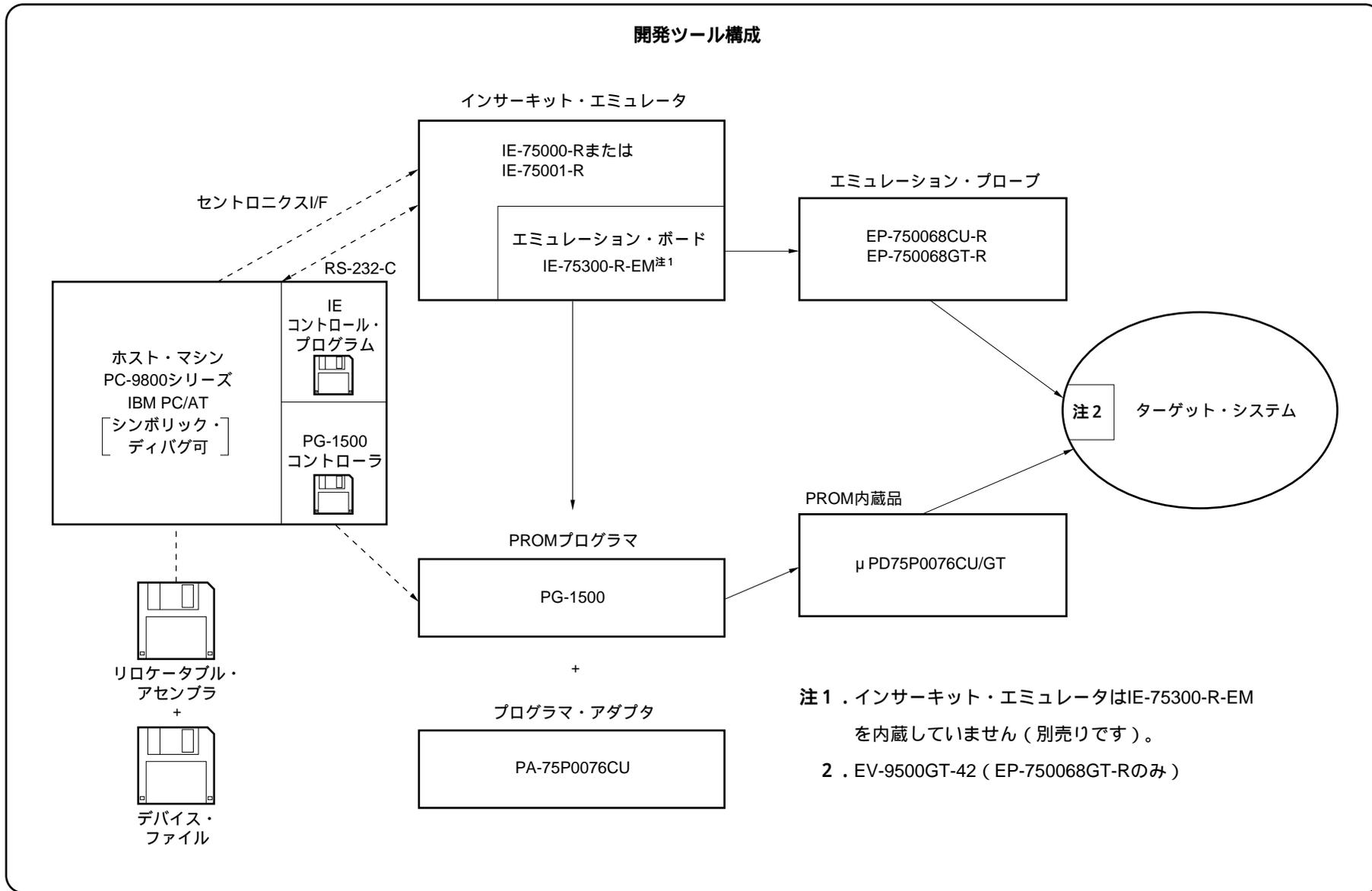
## IBM PC用のOSについて

IBM PC用のOSとして、次のものがサポートされています。

OS	バージョン
PC DOS™	Ver.5.02 ~ Ver.6.3 J6.1/V <sup>注</sup> ~ J6.3/V <sup>注</sup>
MS-DOS	Ver.5.0 ~ Ver.6.22 5.0/V <sup>注</sup> ~ 6.2/V <sup>注</sup>
IBM DOS™	J5.02/V <sup>注</sup>

**注** 英語モードのみサポートしています。

**注意** Ver.5.0以降にはタスク・スワップ機能がありますが、このソフトウェアでは、タスク・スワップ機能は使用できません。



(メモ)

## 付録C マスクROM発注の手順

プログラム開発が完了して、マスクROMを発注する場合には、次のような手順となります。

### マスクROM発注の予約

特約店あるいは、当社販売部門を通じて、マスクROM発注の予定を連絡してください（あらかじめご連絡をいただかないと処理が遅れる場合があります）。

### 発注用媒体の作成

マスクROM発注用の媒体には、次の3つがあります。

- ・ UV-EPROM<sup>注</sup>
- ・ 3.5インチのIBMフォーマットのフロッピー・ディスク（海外のみ）
- ・ 5インチのIBMフォーマットのフロッピー・ディスク（海外のみ）

**注** UV-EPROMで発注する場合には、同じ内容のUV-EPROMを3個作成してください。  
また、マスク・オプション・データはマスク・オプション情報書に記入してください。

### 必要書類の作成

マスクROM発注にあたって、下記の書類に記入してください。

- ・ マスク式ROM発注書
- ・ マスク式ROM発注チェック・シート
- ・ マスク・オプション情報書

### 発 注

で作成した媒体と で記入した書類をまとめて、発注予約日までに特約店あるいは販売部門に提出していただきます。

**注意** 詳しくはインフォメーション資料「ROMコードの発注方法」（資料番号 C10302J）をご覧ください。

(メモ)

## 付録D 命令索引

### D.1 命令索引（機能別）

#### [ 転送命令 ]

MOV A, #n4 ... 277, 300  
MOV reg1, #n4 ... 277, 300  
MOV XA, #n8 ... 277, 300  
MOV HL, #n8 ... 277, 301  
MOV rp2, #n8 ... 277, 301  
MOV A, @HL ... 277, 301  
MOV A, @HL + ... 277, 301  
MOV A, @HL - ... 277, 301  
MOV A, @rpa1 ... 277, 302  
MOV XA, @HL ... 277, 303  
MOV @HL, A ... 277, 303  
MOV @HL, XA ... 277, 303  
MOV A, mem ... 277, 303  
MOV XA, mem ... 277, 304  
MOV mem, A ... 277, 304  
MOV mem, XA ... 277, 304  
MOV A, reg ... 277, 304  
MOV XA, rp' ... 277, 305  
MOV reg1, A ... 277, 305  
MOV rp'1, XA ... 277, 305  
XCH A, @HL ... 277, 306  
XCH A, @HL + ... 277, 306  
XCH A, @HL - ... 277, 306  
XCH A, @rpa1 ... 277, 307  
XCH XA, @HL ... 277, 307  
XCH A, mem ... 277, 307  
XCH XA, mem ... 277, 308  
XCH A, reg1 ... 277, 308  
XCH XA, rp' ... 277, 308

#### [ テーブル参照命令 ]

MOVT XA, @PCDE ... 278, 309  
MOVT XA, @PCXA ... 278, 311  
MOVT XA, @BCDE ... 278, 312  
MOVT XA, @BCXA ... 278, 312

#### [ ビット転送命令 ]

MOV1 CY, fmem.bit ... 278, 313  
MOV1 CY, pmem.@L ... 278, 313  
MOV1 CY, @H + mem.bit ... 278, 313  
MOV1 fmem.bit, CY ... 278, 313  
MOV1 pmem.@L, CY ... 278, 313  
MOV1 @H + mem.bit, CY ... 278, 313

#### [ 演算命令 ]

ADDS A, #n4 ... 278, 314  
ADDS XA, #n8 ... 278, 314  
ADDS A, @HL ... 278, 314  
ADDS XA, rp' ... 278, 314  
ADDS rp'1, XA ... 278, 315  
ADDC A, @HL ... 278, 315  
ADDC XA, rp' ... 278, 315  
ADDC rp'1, XA ... 278, 316  
SUBS A, @HL ... 278, 316  
SUBS XA, rp' ... 278, 316  
SUBS rp'1, XA ... 278, 317  
SUBC A, @HL ... 279, 317  
SUBC XA, rp' ... 279, 317  
SUBC rp'1, XA ... 279, 318  
AND A, #n4 ... 279, 318  
AND A, @HL ... 279, 318

AND XA, rp' ... 279, 318  
 AND rp'1, XA ... 279, 319  
 OR A, #n4 ... 279, 319  
 OR A, @HL ... 279, 319  
 OR XA, rp' ... 279, 319  
 OR rp'1, XA ... 279, 320  
 XOR A, #n4 ... 279, 320  
 XOR A, @HL ... 279, 320  
 XOR XA, rp' ... 279, 320  
 XOR rp'1, XA ... 279, 321

#### [ アキュムレータ操作命令 ]

RORC A ... 279, 322  
 NOT A ... 279, 322

#### [ 増減命令 ]

INCS reg ... 279, 323  
 INCS rp1 ... 279, 323  
 INCS @HL ... 279, 323  
 INCS mem ... 279, 323  
 DECS reg ... 279, 324  
 DECS rp' ... 279, 324

#### [ 比較命令 ]

SKE reg, #n4 ... 279, 325  
 SKE @HL, #n4 ... 279, 325  
 SKE A, @HL ... 279, 325  
 SKE XA, @HL ... 279, 325  
 SKE A, reg ... 279, 326  
 SKE XA, rp' ... 279, 326

#### [ キャリー・フラグ操作命令 ]

SET1 CY ... 279, 327  
 CLR1 CY ... 279, 327  
 SKT CY ... 279, 327  
 NOT1 CY ... 279, 327

#### [ メモリ・ビット操作命令 ]

SET1 mem.bit ... 280, 328  
 SET1 fmem.bit ... 280, 328  
 SET1 pmem.@L ... 280, 328  
 SET1 @H+mem.bit ... 280, 328  
 CLR1 mem.bit ... 280, 328  
 CLR1 fmem.bit ... 280, 328  
 CLR1 pmem.@L ... 280, 328  
 CLR1 @H+mem.bit ... 280, 328  
 SKT mem.bit ... 280, 329  
 SKT fmem.bit ... 280, 329  
 SKT pmem.@L ... 280, 329  
 SKT @H+mem.bit ... 280, 329  
 SKF mem.bit ... 280, 329  
 SKF fmem.bit ... 280, 329  
 SKF pmem.@L ... 280, 329  
 SKF @H+mem.bit ... 280, 329  
 SKTCLR fmem.bit ... 280, 330  
 SKTCLR pmem.@L ... 280, 330  
 SKTCLR @H+mem.bit ... 280, 330  
 AND1 CY, fmem.bit ... 280, 330  
 AND1 CY, pmem.@L ... 280, 330  
 AND1 CY, @H+mem.bit ... 280, 330  
 OR1 CY, fmem.bit ... 280, 330  
 OR1 CY, pmem.@L ... 280, 330  
 OR1 CY, @H+mem.bit ... 280, 330  
 XOR1 CY, fmem.bit ... 280, 331  
 XOR1 CY, pmem.@L ... 280, 331  
 XOR1 CY, @H+mem.bit ... 280, 331

#### [ 分岐命令 ]

BR addr ... 281, 332  
 BR addr1 ... 282, 332  
 BR !addr ... 282, 332  
 BR \$addr ... 282, 333  
 BR \$addr1 ... 283, 333  
 BR PCDE ... 283, 335

BR PCXA ... 283, 335  
BR BCDE ... 283, 336  
BR BCXA ... 283, 336  
BRA !addr1 ... 284, 332  
BRCB !caddr ... 284, 334  
TBR addr ... 336

**[ 特殊命令 ]**

SEL RBn ... 290, 346  
SEL MBn ... 290, 346  
GETI taddr ... 291, 346

**[ サブルーチン・スタック制御命令 ]**

CALLA !addr1 ... 284, 337  
CALL !addr ... 285, 337  
CALLF !faddr ... 286, 338  
TCALL !addr ... 338  
RET ... 287, 339  
RETS ... 288, 339  
RETI ... 289, 340  
PUSH rp ... 289, 341  
PUSH BS ... 289, 341  
POP rp ... 289, 341  
POP BS ... 289, 341

**[ 割り込み制御命令 ]**

EI ... 289, 342  
EI IE x x x ... 289, 342  
DI ... 289, 342  
DI IE x x x ... 289, 342

**[ 入出力命令 ]**

IN A, PORTn ... 290, 343  
IN XA, PORTn ... 290, 343  
OUT PORTn, A ... 290, 344  
OUT PORTn, XA ... 290, 344

**[ CPU制御命令 ]**

HALT ... 290, 345  
STOP ... 290, 345  
NOP ... 290, 345

## D.2 命令索引 (アルファベット順)

## [ A ]

ADDC A, @HL ... 278, 315  
 ADDC rp'1, XA ... 278, 316  
 ADDC XA, rp' ... 278, 315  
 ADDS A, #n4 ... 278, 314  
 ADDS A, @HL ... 278, 314  
 ADDS rp'1, XA ... 278, 315  
 ADDS XA, rp' ... 278, 314  
 ADDS XA, #n8 ... 278, 314  
 AND A, #n4 ... 279, 318  
 AND A, @HL ... 279, 318  
 AND rp'1, XA ... 279, 319  
 AND XA, rp' ... 279, 318  
 AND1 CY, fmem.bit ... 280, 330  
 AND1 CY, pmem.@L ... 280, 330  
 AND1 CY, @H+ mem.bit ... 280, 330

## [ B ]

BR addr ... 281, 332  
 BR addr1 ... 282, 332  
 BR BCDE ... 283, 336  
 BR BCXA ... 283, 336  
 BR PCDE ... 283, 335  
 BR PCXA ... 283, 335  
 BR !addr ... 282, 332  
 BR \$addr ... 282, 333  
 BR \$addr1 ... 283, 333  
 BRA !addr1 ... 284, 332  
 BRCB !caddr ... 284, 334

## [ C ]

CALL !addr ... 285, 337  
 CALLA !addr1 ... 284, 337  
 CALLF !faddr ... 286, 338  
 CLR1 CY ... 279, 327  
 CLR1 fmem.bit ... 280, 328

CLR1 mem.bit ... 280, 328  
 CLR1 pmem.@L ... 280, 328  
 CLR1 @H+ mem.bit ... 280, 328

## [ D ]

DECS reg ... 279, 324  
 DECS rp' ... 279, 324  
 DI ... 289, 342  
 DI IE x x x ... 289, 342

## [ E ]

EI ... 289, 342  
 EI IE x x x ... 289, 342

## [ G ]

GETI taddr ... 291, 346

## [ H ]

HALT ... 290, 345

## [ I ]

IN A, PORTn ... 290, 343  
 IN XA, PORTn ... 290, 343  
 INCS mem ... 279, 323  
 INCS reg ... 279, 323  
 INCS rp1 ... 279, 323  
 INCS @HL ... 279, 323

## [ M ]

MOV A, mem ... 277, 303  
 MOV A, reg ... 277, 304  
 MOV A, #n4 ... 277, 300  
 MOV A, @HL ... 277, 301  
 MOV A, @HL+ ... 277, 301  
 MOV A, @HL- ... 277, 301  
 MOV A, @rpa1 ... 277, 302

MOV	HL, #n8 ...	277, 301	OUT	PORTn, XA ...	290, 344
MOV	mem, A ...	277, 304			
MOV	mem, XA ...	277, 304			
MOV	reg1, A ...	277, 305			
MOV	reg1, #n4 ...	277, 300			
MOV	rp'1, XA ...	277, 305			
MOV	rp2, #n8 ...	277, 301			
MOV	XA, mem ...	277, 304			
MOV	XA, rp' ...	277, 305			
MOV	XA, #n8 ...	277, 300			
MOV	XA, @HL ...	277, 303			
MOV	@HL, A ...	277, 303			
MOV	@HL, XA ...	277, 303			
MOVT	XA, @BCDE ...	278, 312			
MOVT	XA, @BCXA ...	278, 312			
MOVT	XA, @PCDE ...	278, 309			
MOVT	XA, @PCXA ...	278, 311			
MOV1	CY, fmem.bit ...	278, 313			
MOV1	CY, pmem.@L ...	278, 313			
MOV1	CY, @H+mem.bit ...	278, 313			
MOV1	fmem.bit, CY ...	278, 313			
MOV1	pmem.@L, CY ...	278, 313			
MOV1	@H+mem.bit, CY ...	278, 313			
<b>[ N ]</b>					
NOP	...	290, 345			
NOT	A ...	279, 322			
NOT1	CY ...	279, 327			
<b>[ O ]</b>					
OR	A, #n4 ...	279, 319			
OR	A, @HL ...	279, 319			
OR	rp'1, XA ...	279, 320			
OR	XA, rp' ...	279, 319			
OR1	CY, fmem.bit ...	280, 330			
OR1	CY, pmem.@L ...	280, 330			
OR1	CY, @H+mem.bit ...	280, 330			
OUT	PORTn, A ...	290, 344			
			<b>[ P ]</b>		
			POP	BS ...	289, 341
			POP	rp ...	289, 341
			PUSH	BS ...	289, 341
			PUSH	rp ...	289, 341
			<b>[ R ]</b>		
			RET	...	287, 339
			RETI	...	289, 340
			RETS	...	288, 339
			RORC	A ...	279, 322
			<b>[ S ]</b>		
			SEL	MBn ...	290, 346
			SEL	RBn ...	290, 346
			SET1	CY ...	279, 327
			SET1	fmem.bit ...	280, 328
			SET1	mem.bit ...	280, 328
			SET1	pmem.@L ...	280, 328
			SET1	@H+mem.bit ...	280, 328
			SKE	A, reg ...	279, 326
			SKE	A, @HL ...	279, 325
			SKE	reg, #n4 ...	279, 325
			SKE	XA, rp' ...	279, 326
			SKE	XA, @HL ...	279, 325
			SKE	@HL, #n4 ...	279, 325
			SKF	fmem.bit ...	280, 329
			SKF	mem.bit ...	280, 329
			SKF	pmem.@L ...	280, 329
			SKF	@H+mem.bit ...	280, 329
			SKT	CY ...	279, 327
			SKT	fmem.bit ...	280, 329
			SKT	mem.bit ...	280, 329
			SKT	pmem.@L ...	280, 329
			SKT	@H+mem.bit ...	280, 329
			SKTCLR	fmem.bit ...	280, 330

SKTCLR pmem.@L ... 280, 330  
SKTCLR @H+mem.bit ... 280, 330  
STOP ... 290, 345  
SUBC A, @HL ... 279, 317  
SUBC rp'1, XA ... 279, 318  
SUBC XA, rp' ... 279, 317  
SUBS A, @HL ... 278, 316  
SUBS rp'1, XA ... 278, 317  
SUBS XA, rp' ... 278, 316

**[ T ]**

TBR addr ... 336  
TCALL !addr ... 338

**[ X ]**

XCH A, mem ... 277, 307  
XCH A, reg1 ... 277, 308  
XCH A, @HL ... 277, 306  
XCH A, @HL+ ... 277, 306  
XCH A, @HL- ... 277, 306  
XCH A, @rpa1 ... 277, 307  
XCH XA, mem ... 277, 308  
XCH XA, rp' ... 277, 308  
XCH XA, @HL ... 277, 307  
XOR A, #n4 ... 279, 320  
XOR A, @HL ... 279, 320  
XOR rp'1, XA ... 279, 321  
XOR XA, rp' ... 279, 320  
XOR1 CY, fmem.bit ... 280, 331  
XOR1 CY, pmem.@L ... 280, 331  
XOR1 CY, @H+mem.bit ... 280, 331

## 付録E ハードウェア索引

### E.1 ハードウェア索引 (50音順)

#### [あ行]

INTAレジスタ (INTA) ...	69
INTCレジスタ (INTC) ...	69
INTEレジスタ (INTE) ...	69
INTFレジスタ (INTF) ...	69
INTGレジスタ (INTG) ...	69
INTHレジスタ (INTH) ...	69
INT0エッジ検出モード・レジスタ (IM0) ...	221
INT1エッジ検出モード・レジスタ (IM1) ...	221
INT2エッジ検出モード・レジスタ (IM2) ...	241
INT0割り込み許可フラグ (IE0) ...	215
INT1割り込み許可フラグ (IE1) ...	215
INT4割り込み許可フラグ (IE4) ...	215
INT0割り込み要求フラグ (IRQ0) ...	215
INT1割り込み要求フラグ (IRQ1) ...	215
INT4割り込み要求フラグ (IRQ4) ...	215
ウォッチドッグ・タイマ許可フラグ (WDTM) ...	131
A/D変換許可フラグ (ADEN) ...	202
A/D変換モード・レジスタ (ADM) ...	202
SAレジスタ (SA) ...	203

#### [か行]

キー割り込み0-3 (KR0-KR3) ...	239
キャリー・フラグ (CY) ...	90
クロック出力モード・レジスタ (CLOM) ...	126
コマンド・トリガ・ビット (CMDT) ...	178

#### [さ行]

サブ発振回路コントロール・レジスタ (SOS) ...	121
システム・クロック・コントロール・レジスタ (SCC) ...	116
シフト・レジスタ (SIO) ...	179

シリアル・インタフェース動作 許可/禁止指定ビット (CSIE) ...	176
シリアル・インタフェース割り込み許可フラグ (IECSI) ...	215
シリアル・インタフェース割り込み要求フラグ (IRQCSI) ...	215
シリアル動作モード・レジスタ (CSIM) ...	175
シリアル・バス・インタフェース・コントロール・レジスタ (SBIC) ...	178
スキップ・フラグ0-2 (SK0-SK2) ...	90
スタック・バンク選択レジスタ (SBS) ...	74, 86
スタック・ポインタ (SP) ...	86

### 【た行】

タイマ/イベント・カウンタ0カウント・レジスタ (T0) ...	68
タイマ/イベント・カウンタ1カウント・レジスタ (T1) ...	68
タイマ/イベント・カウンタ0出力許可フラグ (TOE0) ...	149
タイマ/イベント・カウンタ1出力許可フラグ (TOE1) ...	149
タイマ/イベント・カウンタ0モード・レジスタ (TM0) ...	146
タイマ/イベント・カウンタ1モード・レジスタ (TM1) ...	147
タイマ/イベント・カウンタ0モジュロ・レジスタ (TMOD0) ...	68
タイマ/イベント・カウンタ1モジュロ・レジスタ (TMOD1) ...	68
タイマ/イベント・カウンタ0割り込み許可フラグ (IET0) ...	215
タイマ/イベント・カウンタ1割り込み許可フラグ (IET1) ...	215
タイマ/イベント・カウンタ0割り込み要求フラグ (IRQT0) ...	215
タイマ/イベント・カウンタ1割り込み要求フラグ (IRQT1) ...	215
時計モード・レジスタ (WM) ...	140

### 【は行】

バス・リリース・トリガ・ビット (RELT) ...	178
バンク選択レジスタ (BS) ...	94
BT割り込み許可フラグ (IEBT) ...	215
BT割り込み要求フラグ (IRQBT) ...	215
ビット・シーケンシャル・バッファ0-3 (BSB0-BSB3) ...	208
プルアップ抵抗指定レジスタ・グループA (POGA) ...	109
プログラム・カウンタ (PC) ...	75
プログラム・ステータス・ワード (PSW) ...	90
プロセッサ・クロック・コントロール・レジスタ (PCC) ...	114
ベーシック・インターバル・タイマ (BT) ...	128
ベーシック・インターバル・タイマ・モード・レジスタ (BTM) ...	129
変換終了検出フラグ (EOC) ...	202

変換スタート指示ビット (SOC) ...	202
ポート (PORT0-6, 11) ...	96
ポート・モード・レジスタ・グループ A , B (PMGA, PMGB) ...	102

**[ま行]**

メモリ・バンク許可フラグ (MBE) ...	92
メモリ・バンク選択レジスタ (MBS) ...	94

**[ら行]**

レジスタ・バンク許可フラグ (RBE) ...	93
レジスタ・バンク選択レジスタ (RBS) ...	94

**[わ行]**

割り込みステータス・フラグ 0 , 1 (IST0, IST1) ...	222
割り込みプライオリティ選択レジスタ (IPS) ...	216
割り込みマスタ許可フラグ (IME) ...	217

## E.2 ハードウェア索引 (アルファベット順)

### [ A ]

ADEN ... 202  
ADM ... 202

### [ B ]

BS ... 94  
BSB0-BSB3 ... 208  
BT ... 132  
BTM ... 129

### [ C ]

CLOM ... 126  
CMDT ... 178  
CSIE ... 176  
CSIM ... 175  
CY ... 90

### [ E ]

EOC ... 202

### [ I ]

IE0 ... 215  
IE1 ... 215  
IE2 ... 238  
IE4 ... 215  
IEBT ... 215  
IECSI ... 215  
IET0 ... 215  
IET1 ... 215  
IEW ... 238  
IM0 ... 221  
IM1 ... 221  
IM2 ... 241  
IME ... 217  
INTA ... 69

INTC ... 69

INTE ... 69

INTF ... 69

INTG ... 69

INTH ... 69

IPS ... 216

IRQ0 ... 215

IRQ1 ... 215

IRQ2 ... 238

IRQ4 ... 215

IRQBT ... 215

IRQCSI ... 215

IRQT0 ... 215

IRQT1 ... 215

IRQW ... 238

IST0, IST1 ... 222

### [ K ]

KR0-KR3 ... 239

### [ M ]

MBE ... 92

MBS ... 94

### [ P ]

PC ... 75

PCC ... 114

PMGA, PMGB ... 102

POGA ... 109

PORT0-PORT6, PORT11 ... 96

PSW ... 90

### [ R ]

RBE ... 93

RBS ... 94

RELT ... 178

**[ S ]**

SA ... 203

SBIC ... 178

SBS ... 74, 86

SCC ... 116

SIO ... 179

SK0-SK2 ... 90

SOC ... 202

SOS ... 121

SP ... 86

**[ T ]**

T0, T1 ... 68

TM0 ... 146

TM1 ... 147

TMOD0, TMOD1 ... 68

TOE0, TOE1 ... 149

**[ W ]**

WDTM ... 131

WM ... 140

(メモ)

## 付録 F 改版履歴

これまでの改版履歴を次に示します。なお、適用箇所は各版での章を示します。

版数	内 容	適用箇所
第 2 版	μ PD750064, 750068, 75P0076を開発中から開発済みに変更	全 般
	μ PD750066を追加	
	データ・バス端子 (D0-D7) を追加	
	外部クロック使用時, XT2をオープンからXT1の逆相入力に変更	
	未使用端子の処理についてを変更	第 2 章 端子機能
	システム・クロック, CPUクロックの切り替えに要する最大時間の表と注意を修正	第 5 章 周辺ハードウェア機能
	マスク・オプションの選択を追加	第 7 章 スタンバイ機能
	プログラム・メモリの書き込み手順を修正	第 9 章 PROM (プログラム・メモリ) の書き込みとベリファイ
	プログラム・メモリの読み出し手順を修正	
	サブシステム・クロックのフィードバック抵抗のマスク・オプションを追加	第 10 章 マスク・オプション
	各命令の命令コードの命令一覧表の一部を修正	第 11 章 命令セット
	開発ツールでサポートするOSをバージョン・アップ	付録 B 開発ツール
	マスクROM発注用の媒体を変更	付録 C マスクROM発注の手順

— お問い合わせ先 —

【技術的なお問い合わせ先】

N E C 半導体テクニカルホットライン (インフォメーションセンター)  
 (電話 : 午前 9:00 ~ 12:00 , 午後 1:00 ~ 5:00)

電話 : 044-548-8899  
 FAX : 044-548-7900  
 E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

半導体第一販売事業部								
半導体第二販売事業部	〒108-8001	東京都港区芝5-7-1	(日本電気本社ビル)					(03)3454-1111
半導体第三販売事業部								
中部支社	半導体第一販売部 半導体第二販売部	〒460-8525	愛知県名古屋市中区錦1-17-1	(日本電気中部ビル)				(052)222-2170 (052)222-2190
関西支社	半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540-8551	大阪府大阪市中央区城見1-4-24	(日本電気関西ビル)				(06)6945-3178 (06)6945-3200 (06)6945-3208
北海道支社	札幌	(011)231-0163	甲府支店	甲府	(055)224-4141	京都支社	京都	(075)344-7824
東北支社	仙台	(022)267-8740	長野支店	松本	(0263)35-1662	神戸支社	神戸	(078)333-3854
岩手支店	盛岡	(019)651-4344	静岡支店	静岡	(054)254-4794	中国支社	広島	(082)242-5504
郡山支店	郡山	(024)923-5511	立川支店	立川	(042)526-5981,6167	鳥取支店	鳥取	(0857)27-5311
長岡支店	長岡	(0258)36-2155	埼玉支店	大宮	(048)649-1415	岡山支店	岡山	(086)225-4455
水戸支店	水戸	(029)226-1717	千葉支店	千葉	(043)238-8116	四国支社	松山	(089)945-4149
群馬支店	高崎	(027)326-1255	神奈川支店	横浜	(045)682-4524	九州支社	福岡	(092)261-2806
太田支店	太田	(0276)46-4011	三重支店	津	(059)225-7341			
宇都宮支店	宇都宮	(028)621-2281	北陸支店	金沢	(076)232-7303			

## 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] μPD750068 ユーザーズ・マニュアル

(U10670JJ2V1UM00 (第2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, NEC半導体ソリューション技術本部員,  
その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡ししてください。

NEC半導体テクニカルホットライン

FAX: (044) 548-7900