

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

μPD750008

4ビット・シングルチップ・マイクロコンピュータ

μPD750004

μPD750006

μPD750008

μPD75P0016

[メモ]

目次要約

第1章	概 説	...	1
第2章	端子機能	...	9
第3章	アーキテクチャの特徴とメモリ・マップ	...	21
第4章	内部CPU機能	...	47
第5章	周辺ハードウェア機能	...	71
第6章	割り込み機能とテスト機能	...	213
第7章	スタンバイ機能	...	247
第8章	リセット機能	...	261
第9章	PROM (プログラム・メモリ) の書き込みとベリファイ	...	265
第10章	マスク・オプション	...	271
第11章	命令セット	...	273
付録A	μ PD75008, 750008, 75P0016の機能一覧表	...	347
付録B	開発ツール	...	349
付録C	マスクROM発注の手順	...	357
付録D	命令索引	...	359
付録E	ハードウェア索引	...	365
付録F	改版履歴	...	369

CMOSデバイスの一般的注意事項

静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

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

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

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

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

初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

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

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

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	44ピン・プラスチックQFPのパッケージをμPD750008GB-x x x-3B4からμPD750008GB-x x x-3BS-MTXに変更
	μPD75P0016を開発中から開発済みに変更
	ポート4, 5のN-chオープン・ドレイン時, 入力耐圧を12 Vから13 Vに変更
	TM1のビット7を“0”固定に変更
はじめに	関連資料に英文版の資料番号を追加
p. 4	1.3 サブシリーズ製品間の比較の表のフォーマットを変更
p.47	4.1.1 Mk モードとMk モードの違いに, Mk モードを使用する場合の注意を追加
p.99	5.2.2(6) サブ発振回路コントロール・レジスタ(SOS)にフィードバック抵抗のマスク・オプションの説明を追加
p.217	6.3 割り込み機能を制御する各種ハードウェアに割り込み許可フラグについての説明を追加
p.230	6.6 ベクタ・アドレス共用割り込み処理に表6-4 共用割り込みの判別を追加
p.269	9.4 ワン・タイムPROMのスクリーニングについてを追加
p.271	第10章 マスク・オプションを追加
p.273-346	第11章 命令セットのオペランドの@rpaを@rpa1に変更
p.277	11.2 命令セットとそのオペレーションの, オペランドの表現形式と記述方法の表に@rpa1を追加
p.301	11.3 各命令の命令コードの命令一覧表(オペランド: BCDE)を訂正
p.302	11.4 命令機能と応用の見出しを, 11.2 命令セットとそのオペレーションに合わせて修正
p.349	付録B 開発ツール サポートするOSのバージョンをアップ
p.369	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、 μ PD750004, 750006, 750008, 75P0016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 μ PD750004, 750006, 750008, 75P0016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。

概要

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。

μ PD75008をすでに経験しているユーザ

付録A μ PD75008, 750008, 75P0016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。

二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 μ PD750004, 750006, 750008, 75P0016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
	重要事項, 強調	: 太字で表記
	数の表記	: 2進数 xxx または xxx B 10進数 xxx 16進数 xxx H

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

デバイス関連資料一覧

資料名	資料番号	
	和文	英文
μ PD750004,750006,750008 データ・シート	U10738J	U10738E
μ PD75P0016 データ・シート	U10328J	U10328E
μ PD750008 ユーザーズ・マニュアル	U10740J (このマニュアル)	U10740E
μ PD750008 インストラクション活用表	U11456J	-
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-75008GB-R ユーザーズ・マニュアル	-	U12928E	
	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 社外メーカー編	C11416J	-

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

第1章 概 説 ...	1
1.1 機能概要 ...	3
1.2 オーダ情報 ...	4
1.3 サブシリーズ製品間の比較 ...	4
1.4 ブロック図 ...	5
1.5 端子接続図 (Top View) ...	6
第2章 端子機能 ...	9
2.1 μ PD750008の端子機能一覧 ...	9
2.2 端子機能の説明 ...	12
2.2.1 P00-P03 (PORT0) ...	12
P10-P13 (PORT1) ...	12
2.2.2 P20-P23 (PORT2) ...	12
P30-P33 (PORT3) ...	12
P40-P43 (PORT4), P50-P53 (PORT5) ...	12
P60-P63 (PORT6), P70-P73 (PORT7) ...	12
2.2.3 P80, P81 (PORT8) ...	13
2.2.4 TI0 ...	13
2.2.5 PTO0, PTO1 ...	13
2.2.6 PCL ...	14
2.2.7 BUZ ...	14
2.2.8 \overline{SCK} , SO/SB0, SI/SB1 ...	14
2.2.9 INT4 ...	14
2.2.10 INT0, INT1 ...	14
2.2.11 INT2 ...	15
2.2.12 KR0-KR3 ...	15
KR4-KR7 ...	15
2.2.13 X1, X2 ...	16
2.2.14 XT1, XT2 ...	16
2.2.15 \overline{RESET} ...	16
2.2.16 V _{DD} ...	17
2.2.17 V _{SS} ...	17
2.2.18 IC (μ PD750004, 750006, 750008のみ) ...	17
2.2.19 V _{PP} (μ PD75P0016のみ) ...	17
2.2.20 MD0-MD3 (μ PD75P0016のみ) ...	17
2.3 端子の入出力回路 ...	18
2.4 未使用端子の処理について ...	20
第3章 アーキテクチャの特徴とメモリ・マップ ...	21
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 ...	21

- 3.1.1 データ・メモリのバンク構成 ... 21
- 3.1.2 データ・メモリのアドレッシング・モード ... 23
- 3.2 汎用レジスタのバンク構成 ... 36
- 3.3 メモリ・マップトI/O ... 41

第4章 内部CPU機能 ... 47

- 4.1 Mk モードとMk モードの切り替え機能 ... 47
 - 4.1.1 Mk モードとMk モードの違い ... 47
 - 4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法 ... 48
- 4.2 プログラム・カウンタ (PC) ... 49
- 4.3 プログラム・メモリ (ROM) ... 50
- 4.4 データ・メモリ (RAM) ... 56
 - 4.4.1 データ・メモリの構成 ... 56
 - 4.4.2 データ・メモリのバンクの指定 ... 57
- 4.5 汎用レジスタ ... 60
- 4.6 アキュムレータ ... 62
- 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) ... 62
- 4.8 プログラム・ステータス・ワード (PSW) ... 66
- 4.9 バンク選択レジスタ (BS) ... 69

第5章 周辺ハードウェア機能 ... 71

- 5.1 デジタル入出力ポート ... 71
 - 5.1.1 デジタル入出力ポートの種類, 特徴, 構成 ... 72
 - 5.1.2 入出力モードの設定 ... 78
 - 5.1.3 デジタル入出力ポート操作命令 ... 80
 - 5.1.4 デジタル入出力ポートの動作 ... 83
 - 5.1.5 プルアップ抵抗の内蔵 ... 85
 - 5.1.6 デジタル入出力ポートの入出力タイミング ... 86
- 5.2 クロック発生回路 ... 89
 - 5.2.1 クロック発生回路の構成 ... 89
 - 5.2.2 クロック発生回路の機能と動作 ... 91
 - 5.2.3 システム・クロックとCPUクロックの設定 ... 101
 - 5.2.4 クロック出力回路 ... 104
- 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 ... 107
 - 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 ... 107
 - 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) ... 108
 - 5.3.3 ウォッチドッグ・タイマ許可フラグ (WDTM) ... 110
 - 5.3.4 ベーシック・インターバル・タイマの動作 ... 110
 - 5.3.5 ウォッチドッグ・タイマの動作 ... 111
 - 5.3.6 その他の機能 ... 113
- 5.4 時計用タイマ ... 116
 - 5.4.1 時計用タイマの構成 ... 117
 - 5.4.2 時計モード・レジスタ ... 118
- 5.5 タイマ/イベント・カウンタ ... 120
 - 5.5.1 タイマ/イベント・カウンタの構成 ... 120

5.5.2	8ビット・タイマ/イベント・カウンタ・モードの動作	...	127
5.5.3	タイマ/イベント・カウンタ応用上の注意	...	135
5.6	シリアル・インタフェース	...	138
5.6.1	シリアル・インタフェースの機能	...	138
5.6.2	シリアル・インタフェースの構成	...	139
5.6.3	レジスタの機能	...	143
5.6.4	動作停止モード	...	153
5.6.5	3線式シリアルI/Oモードの動作	...	155
5.6.6	2線式シリアルI/Oモードの動作	...	166
5.6.7	SBIモードの動作	...	174
5.6.8	SCK端子出力の操作	...	209
5.7	ビット・シーケンシャル・バッファ	...	210
第6章 割り込み機能とテスト機能 ... 213			
6.1	割り込み制御回路の構成	...	213
6.2	割り込みソースの種類とベクタ・テーブル	...	215
6.3	割り込み機能を制御する各種ハードウェア	...	217
6.4	割り込みシーケンス	...	226
6.5	多重割り込み処理制御	...	228
6.6	ベクタ・アドレス共用割り込み処理	...	230
6.7	割り込み処理までのマシン・サイクル	...	232
6.8	割り込みの効果的な使い方	...	234
6.9	割り込みの応用	...	234
6.10	テスト機能	...	242
6.10.1	テスト・ソースの種類	...	242
6.10.2	テスト機能を制御する各種ハードウェア	...	242
第7章 スタンバイ機能 ... 247			
7.1	スタンバイ・モードの設定および動作状態	...	249
7.2	スタンバイ・モードの解除	...	251
7.3	スタンバイ・モード解除後の動作	...	254
7.4	マスク・オプションの選択	...	254
7.5	スタンバイ・モードの応用	...	255
第8章 リセット機能 ... 261			
第9章 PROM (プログラム・メモリ) の書き込みとベリファイ ... 265			
9.1	プログラム・メモリ書き込み/ベリファイ時の動作モード	...	265
9.2	プログラム・メモリ書き込みの手順	...	266
9.3	プログラム・メモリ読み出しの手順	...	268
9.4	ワン・タイムPROMのスクリーニングについて	...	269
第10章 マスク・オプション ... 271			
10.1	端子	...	271

- 10.2 スタンバイ機能のマスク・オプション ... 271
- 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 ... 272

第11章 命令セット ... 273

- 11.1 特徴的な命令 ... 273
 - 11.1.1 GETI命令 ... 273
 - 11.1.2 ビット操作命令 ... 274
 - 11.1.3 たてづみ命令 ... 274
 - 11.1.4 進数補正命令 ... 275
 - 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 ... 276
- 11.2 命令セットとそのオペレーション ... 277
- 11.3 各命令の命令コード ... 296
- 11.4 命令機能と応用 ... 302
 - 11.4.1 転送命令 ... 303
 - 11.4.2 テーブル参照命令 ... 310
 - 11.4.3 ビット転送命令 ... 314
 - 11.4.4 演算命令 ... 315
 - 11.4.5 アキュムレータ操作命令 ... 322
 - 11.4.6 増減命令 ... 322
 - 11.4.7 比較命令 ... 323
 - 11.4.8 キャリー・フラグ操作命令 ... 325
 - 11.4.9 メモリ・ビット操作命令 ... 325
 - 11.4.10 分岐命令 ... 329
 - 11.4.11 サブルーチン・スタック制御命令 ... 334
 - 11.4.12 割り込み制御命令 ... 340
 - 11.4.13 入出力命令 ... 341
 - 11.4.14 CPU制御命令 ... 342
 - 11.4.15 特殊命令 ... 343

付録A μ PD75008, 750008, 75P0016の機能一覧表 ... 347

付録B 開発ツール ... 349

付録C マスクROM発注の手順 ... 357

付録D 命令索引 ... 359

- D.1 命令索引(機能別) ... 359
- D.2 命令索引(アルファベット順) ... 362

付録E ハードウェア索引 ... 365

- E.1 ハードウェア索引(50音順) ... 365
- E.2 ハードウェア索引(アルファベット順) ... 367

付録F 改版履歴 ... 369

図 の 目 次 (1/4)

図番号	タイトル, ページ
2 - 1	端子の入出力回路一覧 ... 18
3 - 1	MBE = 0のモードとMBE = 1のモードの使い分け ... 22
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 ... 24
3 - 3	スタティックRAMのアドレス更新方法 ... 30
3 - 4	レジスタ・バンクの使い分け例 ... 37
3 - 5	汎用レジスタの構成 (4ビット処理の場合) ... 39
3 - 6	汎用レジスタの構成 (8ビット処理の場合) ... 40
3 - 7	μPD750008 I/Oマップ ... 42
4 - 1	スタック・バンク選択レジスタのフォーマット ... 48
4 - 2	プログラム・カウンタの構成... 49
4 - 3	プログラム・メモリ・マップ (μPD750004) ... 52
4 - 4	プログラム・メモリ・マップ (μPD750006) ... 53
4 - 5	プログラム・メモリ・マップ (μPD750008) ... 54
4 - 6	プログラム・メモリ・マップ (μPD75P0016) ... 55
4 - 7	データ・メモリ・マップ ... 58
4 - 8	汎用レジスタの構成 ... 61
4 - 9	レジスタ・ペアの構成 ... 61
4 - 10	アキュムレータ ... 62
4 - 11	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 ... 63
4 - 12	スタック・メモリへ退避されるデータ (Mk モード) ... 64
4 - 13	スタック・メモリから復帰されるデータ (Mk モード) ... 64
4 - 14	スタック・メモリへ退避されるデータ (Mk モード) ... 65
4 - 15	スタック・メモリから復帰されるデータ (Mk モード) ... 65
4 - 16	プログラム・ステータス・ワードの構成 ... 66
4 - 17	バンク選択レジスタの構成 ... 69
5 - 1	デジタル・ポートのデータ・メモリ・アドレス ... 71
5 - 2	ポート0, 1の構成 ... 73
5 - 3	ポート2, 7の構成 ... 74
5 - 4	ポート3n , ポート6nの構成 (n = 0 - 3) ... 75
5 - 5	ポート4, 5の構成 ... 76
5 - 6	ポート8の構成 ... 77

図 の 目 次 (2/4)

図番号	タイトル, ページ
5 - 7	各ポート・モード・レジスタのフォーマット ... 79
5 - 8	プルアップ抵抗指定レジスタのフォーマット ... 86
5 - 9	デジタル入出力ポートの入出力タイミング ... 87
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 ... 88
5 - 11	クロック発生回路のブロック図 ... 89
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 ... 93
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 ... 94
5 - 14	メイン・システム・クロック発振回路の外付け回路 ... 95
5 - 15	サブシステム・クロック発振回路の外付け回路 ... 95
5 - 16	発振子の接続回路の悪い例 ... 96
5 - 17	サブシステム・クロック発振回路 ... 98
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 ... 100
5 - 19	システム・クロックとCPUクロックの切り替え ... 102
5 - 20	クロック出力回路の構成 ... 104
5 - 21	クロック出力モード・レジスタのフォーマット ... 105
5 - 22	リモコン出力応用例 ... 106
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 ... 107
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 ... 109
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 ... 110
5 - 26	時計用タイマのブロック図 ... 117
5 - 27	時計モード・レジスタのフォーマット ... 119
5 - 28	タイマ/イベント・カウンタのブロック図 (チャンネル0) ... 121
5 - 29	タイマ・カウンタのブロック図 (チャンネル1) ... 122
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 ... 124
5 - 31	タイマ・カウンタ・モード・レジスタ (チャンネル1) のフォーマット ... 125
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 ... 126
5 - 33	タイマ/イベント・カウンタ・モード・レジスタの設定 ... 128
5 - 34	タイマ/イベント・カウンタ出力許可フラグの設定 ... 130
5 - 35	タイマ/イベント・カウンタ動作時の構成 ... 133
5 - 36	カウント動作時のタイミング ... 133
5 - 37	カウント・レジスタのクリア・タイミング ... 135
5 - 38	SBIのシステム構成例 ... 139
5 - 39	シリアル・インタフェースのブロック図 ... 140
5 - 40	シリアル動作モード・レジスタ (CSIM) のフォーマット ... 143

図 の 目 次 (3/4)

図番号	タイトル, ページ
5 - 41	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 ... 148
5 - 42	シフト・レジスタの周辺の構成 ... 151
5 - 43	3線式シリアルI/Oのシステム構成例 ... 155
5 - 44	3線式シリアルI/Oモードのタイミング ... 159
5 - 45	RELT, CMDTの動作 ... 160
5 - 46	転送ビットの切り替え回路 ... 161
5 - 47	2線式シリアルI/Oのシステム構成例 ... 166
5 - 48	2線式シリアルI/Oモードのタイミング ... 170
5 - 49	RELT, CMDTの動作 ... 171
5 - 50	SBIのシステム構成例 ... 175
5 - 51	SBI転送のタイミング ... 177
5 - 52	バス・リリース信号 ... 178
5 - 53	コマンド信号 ... 178
5 - 54	アドレス ... 179
5 - 55	アドレスによるスレーブの選択 ... 179
5 - 56	コマンド ... 180
5 - 57	データ ... 180
5 - 58	アクノリッジ信号 ... 181
5 - 59	ビジー信号, レディ信号 ... 182
5 - 60	RELT, CMDT, RELD, CMDD (マスタ) の動作 ... 187
5 - 61	RELT, CMDT, RELD, CMDD (スレーブ) の動作 ... 188
5 - 62	ACKTの動作 ... 188
5 - 63	ACKEの動作 ... 189
5 - 64	ACKDの動作 ... 190
5 - 65	BSYEの動作 ... 191
5 - 66	端子構成図 ... 194
5 - 67	マスタ・デバイスからスレーブ・デバイス (WUP = 1) へのアドレス送信動作 ... 196
5 - 68	マスタ・デバイスからスレーブ・デバイスへのコマンド送信動作 ... 197
5 - 69	マスタ・デバイスからスレーブ・デバイスへのデータ送信動作 ... 198
5 - 70	スレーブ・デバイスからマスタ・デバイスへのデータ送信動作 ... 199
5 - 71	シリアル・バス構成例 ... 202
5 - 72	READコマンドの転送フォーマット ... 204
5 - 73	WRITE, ENDコマンドの転送フォーマット ... 204
5 - 74	STOPコマンドの転送フォーマット ... 205

図 の 目 次 (4/4)

図番号	タイトル, ページ
5 - 75	STATUSコマンドの転送フォーマット ... 206
5 - 76	STATUSコマンドのステータス・フォーマット ... 206
5 - 77	RESETコマンドの転送フォーマット ... 207
5 - 78	CHGMSTコマンドの転送フォーマット ... 207
5 - 79	エラー発生時のマスタ, スレーブの動作 ... 208
5 - 80	$\overline{\text{SCK}}$ /P01端子の構成 ... 209
5 - 81	ビット・シーケンシャル・パツファのフォーマット ... 210
6 - 1	割り込み制御回路ブロック図 ... 214
6 - 2	割り込みベクタ・テーブル ... 215
6 - 3	割り込みプライオリティ選択レジスタ ... 220
6 - 4	INT0, INT1, INT4の構成 ... 222
6 - 5	ノイズ除去回路の入出力タイミング ... 223
6 - 6	エッジ検出モード・レジスタのフォーマット ... 224
6 - 7	割り込みシーケンス ... 226
6 - 8	高位の割り込みによる多重割り込み ... 228
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 ... 229
6 - 10	INT2, KR0-KR7のブロック図 ... 244
6 - 11	INT2エッジ検出モード・レジスタ (IM2) のフォーマット ... 245
7 - 1	スタンバイ・モードの解除動作 ... 251
7 - 2	STOPモード解除時のウエイト時間 ... 253
8 - 1	リセット機能の構成 ... 261
8 - 2	$\overline{\text{RESET}}$ 信号発生によるリセット動作 ... 261
B - 1	EV-9200G-44 外形図 (参考) (単位: mm) ... 354
B - 2	EV-9200G-44 基板取り付け推奨パターン (参考) (単位: mm) ... 355

表 の 目 次 (1/2)

表番号	タイトル, ページ
1 - 1	各製品の特徴 ... 1
2 - 1	デジタル入出力ポートの端子機能一覧 ... 9
2 - 2	ポート端子以外の端子機能一覧 ... 11
2 - 3	未使用端子の処理一覧 ... 20
3 - 1	アドレッシング・モード ... 25
3 - 2	RBE, RBSと選択されるレジスタ・バンク ... 36
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 ... 36
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 ... 41
4 - 1	Mk モードとMk モードの違い ... 47
4 - 2	SBSで選択されるスタック・エリア ... 62
4 - 3	スタック動作時に退避 / 復帰されるPSWのフラグ ... 66
4 - 4	キャリア・フラグ操作命令 ... 67
4 - 5	割り込みステータス・フラグの指示内容 ... 68
4 - 6	RBE, RBSと選択されるレジスタ・バンク ... 70
5 - 1	デジタル・ポートの種類と特徴 ... 72
5 - 2	入出力端子操作命令一覧表 ... 82
5 - 3	入出力ポート操作時の動作 ... 84
5 - 4	プルアップ抵抗内蔵の指定方法 ... 85
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 ... 101
5 - 6	分解能と最長設定時間 ... 131
5 - 7	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) ... 160
5 - 8	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) ... 171
5 - 9	シリアル・クロックの選択と応用 (SBIモード時) ... 187
5 - 10	SBIにおける各種の信号 ... 192
6 - 1	割り込みソースの種類 ... 215
6 - 2	割り込み要求フラグのセット信号 ... 218
6 - 3	IST1, IST0と割り込み処理状態 ... 225
6 - 4	共用割り込みの判別 ... 230
6 - 5	テスト・ソースの種類 ... 242

表 の 目 次 (2/2)

表番号	タイトル, ページ
6 - 6	テスト要求信号のセット信号 ... 242
7 - 1	スタンバイ・モード時の各動作状態 ... 249
7 - 2	BTMによるウエイト時間の選択 ... 253
8 - 1	各ハードウェアのリセット後の状態 ... 262
10 - 1	端子のマスク・オプションの選択 ... 271

第 1 章 概 説

μPD750004, 750006, 750008, 75P0016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μPD750004, 750006, 750008, 75P0016を総称して、μPD750008サブシリーズといたします。

75XLシリーズは、75XシリーズのCPUを継承し、幅広い動作電圧と高速動作を実現しました。従来製品と上位互換性があるだけでなく、電池を使用するアプリケーションなどにも最適です。

次に示すような特徴を持っています。

- ・低電圧動作可能... $V_{DD} = 2.2 \sim 5.5 \text{ V}$
- ・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 μs , 1.91 μs , 3.81 μs , 15.3 μs , (4.19 MHz動作時)
 - 0.67 μs , 1.33 μs , 2.67 μs , 10.7 μs , (6.0 MHz動作時)
 - 122 μs , (32.768 kHz動作時)
- ・強化されたタイマ機能：4チャンネル
- ・従来品のμPD75008の機能、インストラクションを継承しているため置き換えが容易

プログラム・メモリのサイズと種類によって、表1 - 1のような製品群を用意しております。

表 1 - 1 各製品の特徴

品 名	プログラム・メモリ (ROM)	備 考
μPD750004	4096 × 8 ビット	マスクROM
μPD750006	6144 × 8 ビット	
μPD750008	8192 × 8 ビット	
μPD75P0016	16384 × 8 ビット	ワン・タイムPROM

μPD75P0016は、電气的書き込み可能なワン・タイムPROMを備えた製品で、μPD750004, 750006, 750008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

- ・家電分野

VTR, オーディオ (CDプレーヤー), リモコン, etc.

- ・その他

電話機, カメラ, etc.

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 μ PD750008を代表品種として記述してあります。 μ PD750004, 750006, 75P0016のマニュアルとしてお使いの場合は, μ PD750008を μ PD750004, 750006, 75P0016と読み替えてお使いください。

1.1 機能概要

項 目	機 能			
命令実行時間	<ul style="list-style-type: none"> ・ 0.95, 1.91, 3.81, 15.3 μs (メイン・システム・クロック : 4.19 MHz動作時) ・ 0.67, 1.33, 2.67, 10.7 μs (メイン・システム・クロック : 6.0 MHz動作時) ・ 122 μs (サブシステム・クロック : 32.768 kHz動作時) 			
内蔵メモリ	ROM	4096 \times 8 ビット (μ PD750004)		
		6144 \times 8 ビット (μ PD750006)		
		8192 \times 8 ビット (μ PD750008)		
		16384 \times 8 ビット (μ PD75P0016)		
	RAM	512 \times 4 ビット		
汎用レジスタ	<ul style="list-style-type: none"> ・ 4 ビット操作時 : 8 個 \times 4 バンク ・ 8 ビット操作時 : 4 個 \times 4 バンク 			
入出力ポート	34本	8本	CMOS入力端子	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 : 25
		18本	CMOS入出力端子 LED直接駆動可能 : 4	
	8本	N-chオープン・ドレイン 入出力端子 LED直接駆動可能 : 8	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 ^注	
タイマ	4チャンネル	<ul style="list-style-type: none"> ・ タイマ/イベント・カウンタ : 1チャンネル ・ タイマ・カウンタ : 1チャンネル ・ 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 ・ 時計用タイマ : 1チャンネル 		
シリアル・インタフェース	<ul style="list-style-type: none"> ・ 3線式シリアルI/Oモード (MSB/LSB先頭切り替え) ・ 2線式シリアルI/Oモード ・ SBIモード 			
ビット・シーケンシャル・バッファ	16ビット			
クロック出力機能	<ul style="list-style-type: none"> ・ Φ, 524, 262, 65.5 kHz (メイン・システム・クロック : 4.19 MHz動作時) ・ Φ, 750, 375, 93.7 kHz (メイン・システム・クロック : 6.0 MHz動作時) 			
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	<ul style="list-style-type: none"> ・ 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/HALTモード			
動作周囲温度	$T_A = -40 \sim +85$			
電源電圧	$V_{DD} = 2.2 \sim 5.5$ V			
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)			

注 μ PD75P0016にはマスク・オプションによるプルアップ抵抗がなく, 常にオープンとなっています。

1.2 オーダ情報

オーダ名称	パッケージ	内蔵ROM
μPD750004CU-x x x	42ピン・プラスチック・シュリンクDIP (600 mil)	マスクROM
μPD750004GB-x x x-3BS-MTX ^注	44ピン・プラスチックQFR (10 mm)	"
μPD750006CU-x x x	42ピン・プラスチック・シュリンクDIP (600 mil)	"
μPD750006GB-x x x-3BS-MTX ^注	44ピン・プラスチックQFP (10 mm)	"
μPD750008CU-x x x	42ピン・プラスチック・シュリンクDIP (600 mil)	"
μPD750008GB-x x x-3BS-MTX ^注	44ピン・プラスチックQFP (10 mm)	"
μPD75P0016CU	42ピン・プラスチック・シュリンクDIP (600 mil)	ワン・タイムPROM
μPD75P0016GB-3BS-MTX ^注	44ピン・プラスチックQFP (10 mm)	"

注 1996年4月1日以降のコード受注分より受け付けます。

備考 x x xはROMコード番号です。

1.3 サブシリーズ製品間の比較

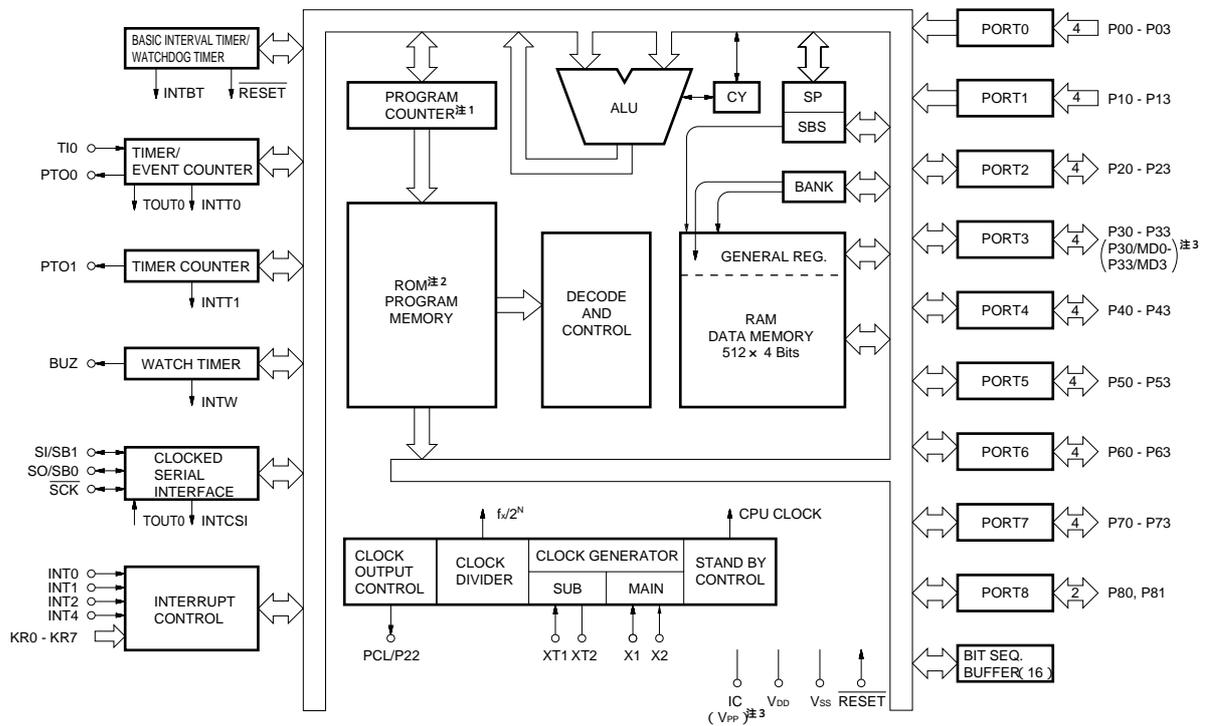
項 目	μPD750004	μPD750006	μPD750008	μPD75P0016
プログラム・カウンタ	12ビット	13ビット		14ビット
プログラム・メモリ (バイト)	マスクROM 4096	マスクROM 6144	マスクROM 8192	ワン・タイムPROM 16384
データ・メモリ (x 4ビット)	512			
マスク・オプション	PORT4, 5のプルアップ抵抗	あり (内蔵する / しないの指定可能)		なし (内蔵不可)
	RESET 時のウエイト時間	あり (2 ¹⁷ /fx, 2 ¹⁵ /fxの2つから選択可能) ^注		なし (2 ¹⁵ /fx固定)
	サブシステム・クロックのフィードバック抵抗	あり (使用可能にする / しないの選択可能)		なし (使用可能)
端子接続	6-9番ピン (CU)	P33-30		P33/MD3-P30/MD0
	23-26番ピン (GB)			
	20番ピン (CU)	IC		V _{PP}
	38番ピン (GB)			
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 2¹⁷/fxは、6.0 MHz動作時：21.8 ms，4.19 MHz動作時：31.3 msです。

2¹⁵/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などが異なります。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品の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 . μ PD750004は12ビット , μ PD750006, 750008は13ビット, μ PD75P0016は14ビット構成です。

2 . ROMの容量は , 品種により異なります。

3 . μ PD75P0016の場合です。

1.5 端子接続図 (Top View)

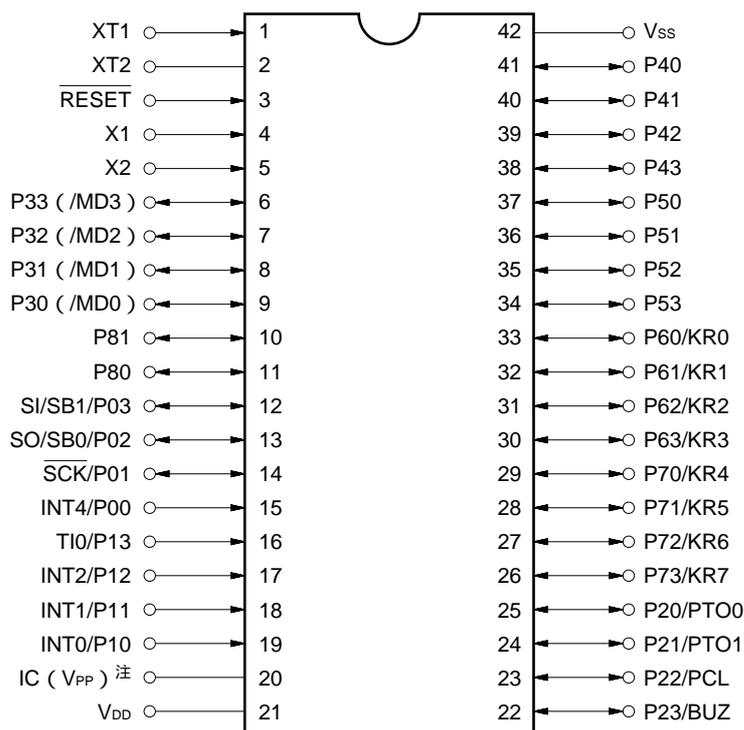
(1) 42ピン・プラスチック・シュリンクDIP (600 mil)

μ PD750004CU-x x x

μ PD750006CU-x x x

μ PD750008CU-x x x

μ PD75P0016CU



注 IC (VPP) はVDDに直接接続してください。

備考 ()内はμ PD75P0016のとき

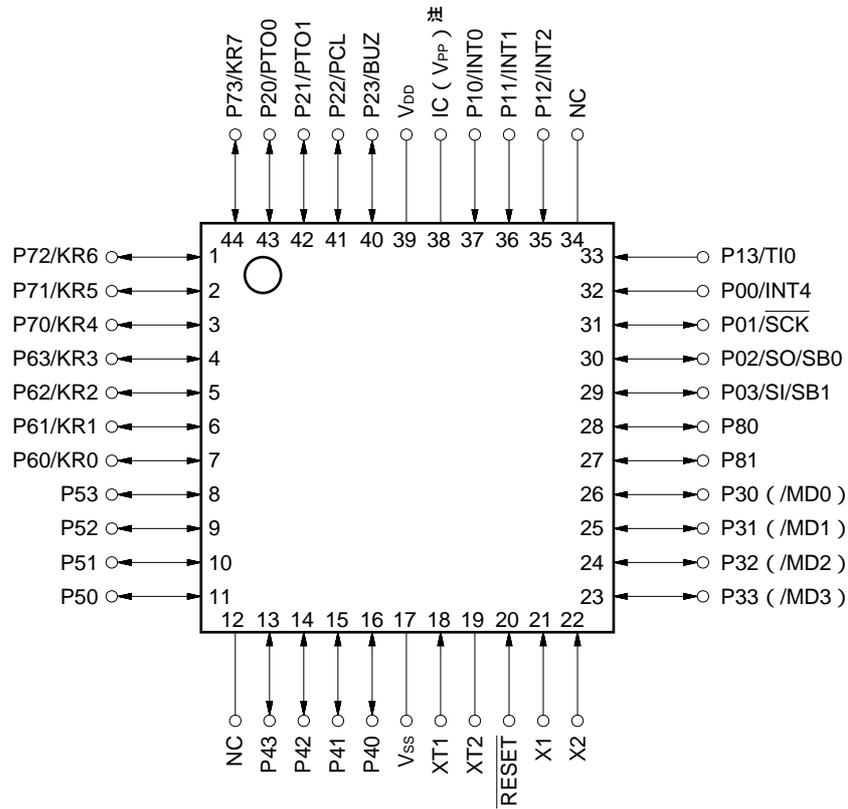
(2) 44ピン・プラスチックQFP (10 mm)

μ PD750004GB- x x x -3BS-MTX

μ PD750006GB- x x x -3BS-MTX

μ PD750008GB- x x x -3BS-MTX

μ PD75P0016GB-3BS-MTX



注 IC (V_{PP}) はV_{DD}に直接接続してください。

備考 ()内はμ PD75P0016のとき

端子名称

P00-03 : Port0	$\overline{\text{RESET}}$: Reset Input
P10-13 : Port1	TI0 : Timer Input0
P20-23 : Port2	PTO0, 1 : Programmable Timer Output0, 1
P30-33 : Port3	BUZ : Buzzer Clock
P40-43 : Port4	PCL : Programmable Clock
P50-53 : Port5	INT0, 1, 4 : External Vectored Interrupt0, 1, 4
P60-63 : Port6	INT2 : External Test Input2
P70-73 : Port7	X1, 2 : Main System Clock Oscillation1, 2
P80-81 : Port8	XT1, 2 : Subsystem Clock Oscillation1, 2
KR0-7 : Key Return	NC : No Connection
$\overline{\text{SCK}}$: Serial Clock	IC : Internally Connected
SI : Serial Input	V _{DD} : Positive Power Supply
SO : Serial Output	V _{SS} : Ground
SB0, 1 : Serial Bus0, 1	V _{PP} : Programming Power Supply
	MD0-MD3 : Mode Selection0-3

第2章 端子機能

2.1 μPD750008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 ^{注1}
P00	入力	INT4	4ビット入力ポート(PORT0)。	×	入力	B
P01	入出力	\overline{SCK}	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。			F-A
P02	入出力	SO/SB0				F-B
P03	入出力	SI/SB1				M-C
P10	入力	INT0	4ビット入力ポート(PORT1)。	×	入力	B-C
P11		INT1	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。			
P12		INT2				
P13		TI0	PI0/INT0のみノイズ除去機能付き。			
P20	入出力	PTO0	4ビット入出力ポート(PORT2)。	×	入力	E-B
P21		PTO1	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。			
P22		PCL				
P23		BUZ				
P30 ^{注2}	入出力	(MD0) ^{注3}	プログラマブル4ビット入出力ポート(PORT3)。	×	入力	E-B
P31 ^{注2}		(MD1) ^{注3}				
P32 ^{注2}		(MD2) ^{注3}	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。			
P33 ^{注2}		(MD3) ^{注3}				
P40-P43 ^{注2,4}	入出力	-	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) ^{注5} 。 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) ^{注3}

注1. 印はシュミット・トリガ入力を示します。

2. LEDを直接駆動可能。

3. ()内は、μPD75P0016の場合です。

4. 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。

5. μPD75P0016にはマスク・オプションによるプルアップ抵抗がなく、常にオープンとなっています。

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 ^{注1}
P50-P53 ^{注2, 3}	入出力	-	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) ^{注4} 。 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) ^{注5}
P60	入出力	KR0	プログラマブル4ビット入出力ポート(PORT6)。 ビット単位で入出力指定可能。 4ビット単位で, ソフトウエアにより内蔵プルアップ抵抗の接続を指定可能。		入力	F-A
P61		KR1				
P62		KR2				
P63		KR3				
P70	入出力	KR4	4ビット入出力ポート(PORT7)。 4ビット単位で, ソフトウエアにより内蔵プルアップ抵抗の接続を指定可能。		入力	F-A
P71		KR5				
P72		KR6				
P73		KR7				
P80	入出力	-	2ビット入出力ポート(PORT8)。 2ビット単位で, ソフトウエアにより内蔵プルアップ抵抗の接続を指定可能。	x	入力	E-B
P81		-				

注1 . 印はシュミット・トリガ入力を示します。

2 . LEDを直接駆動可能。

3 . 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。

4 . μPD75P0016にはマスク・オプションによるプルアップ抵抗がなく, 常にオープンとなっています。

5 . ()内はμPD75P0016の場合です。

表2-2 ポート端子以外の端子機能一覧

端子名称	入出力	兼用端子	機能		リセット時	入出力回路 TYPE ^{注1}
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。		-	B-C
PTO0	入出力	P20	タイマ/イベント・カウンタ出力。		入力	E-B
PTO1		P21	タイマ・カウンタ出力。			
PCL	入出力	P22	クロック出力。		入力	E-B
BUZ	入出力	P23	固定周波数出力（ブザー用またはシステム・クロックのトリミング用）。		入力	E-B
$\overline{\text{SCK}}$	入出力	P01	シリアル・クロック入出力。		入力	F-A
SO/SB0	入出力	P02	シリアル・データ出力。 シリアル・データ・バス入出力。		入力	F-B
SI/SB1	入出力	P03	シリアル・データ入力。 シリアル・データ・バス入出力。		入力	M-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エッジ検出の両方が有効）。		-	B
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）。 INT0/P10はノイズ除去機能付き。	クロック同期式	-	B-C
INT1		P11		非同期		
INT2	入力	P12	立ち上がりエッジ検出テストブル入力。	非同期	-	B-C
KR0-KR3	入出力	P60-P63	並列立ち下がりエッジ検出テストブル入力。		入力	F-A
KR4-KR7	入出力	P70-P73	並列立ち下がりエッジ検出テストブル入力。		入力	F-A
X1, X2	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。		-	-
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はオープンとする。		-	-
XT2	-					
$\overline{\text{RESET}}$	入力	-	システム・リセット入力。		-	B
IC ^{注2}	-	-	Internally Connected. V _{DD} に直接接続。		-	-
V _{DD}	-	-	正電源。		-	-
V _{SS}	-	-	GND電位。		-	-
V _{PP} ^{注3}	-	-	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。PROM書き込み/ベリファイ時は +12.5 Vを印加。V _{DD} に直接接続。		-	-
MD0-MD3 ^{注3}	入出力	P30-P33	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
NC	-	-	No Connection.		-	-

注1 . 印はシュミット・トリガ入力を示します。

2 . μ PD75P0016の場合は、V_{PP}として使用します。

3 . μ PD75P0016のみ内蔵。

2.2 端子機能の説明

2.2.1 P00-P03 (PORT0) ...INT4, $\overline{\text{SCK}}$, SO/SB0, SI/SB1兼用入力 P10-P13 (PORT1) ...INT0-INT2, TI0兼用入力

4ビット入力ポート：ポート0，ポート1の入力端子です。

ポート0，ポート1は入力ポートの機能のほかに，次のような機能があります。

(1) **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 ($\overline{\text{SCK}}$, SO/SB0, SI/SB1)

(2) **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)

ポート0，ポート1の各端子の状態は兼用端子の動作にかかわらず，常に入力ができます。

ポート0の入力，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・トリガ入力となっています。さらに，P10はノイズ除去回路が付いています (6.3(3)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。これらの指定は，プルアップ抵抗指定レジスタ・グループA (POGA) を操作して行います。

$\overline{\text{RESET}}$ 信号発生により，いずれの端子も入力ポートとなります。

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^注 P40-P43 (PORT4) , P50-P53 (PORT5) ... N-chオープン・ドレイン中耐圧 (13 V) 大電流出力 P60-P63 (PORT6) , P70-P73 (PORT7) ... 3ステート入出力

出力ラッチ付き4ビット入出力ポート：ポート2-7の入出力端子です。

ポートn (n=2, 3, 6, 7) には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。

- (1) **ポート2** : タイマ/イベント・カウンタ出力 (PTO0)
 タイマ・カウンタ出力 (PTO1)
 クロック出力 (PCL)
 固定周波数出力 (BUZ)
- (2) **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)^注
- (3) **ポート6, ポート7** : キー割り込み入力 (KR0-KR3, KR4-KR7)

注 μ PD75P0016のみMD0-MD3を兼用しています。

ポート3は大電流出力となっており、LEDを直接駆動可能です。ポート4, ポート5はN-chオープン・ドレイン中耐圧(13V)大電流出力で、LEDを直接駆動可能です。

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m (m = 2, 4, 5, 7) は4ビット単位で入出力を選択できます。ポート3, ポート6はビット単位で入出力を選択できます。

ポートnはソフトウェアによる内蔵プルアップ抵抗の接続指定が4ビット単位で可能で、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, ポート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の内蔵指定ができます。

ポート4, ポート5およびポート6, ポート7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、いずれのポートも出力ラッチはクリアされ、ポートnは入力モード(出力ハイ・インピーダンス)に、ポート4, ポート5はハイ・レベル(プルアップ抵抗の内蔵時)またはハイ・インピーダンスになります。

2.2.3 P80, P81 (PORT8)

出力ラッチ付き2ビット入出力ポート: ポート8の入出力端子です。

ポート8は、ソフトウェアによる内蔵プルアップ抵抗の接続指定が2ビット単位で可能で、プルアップ抵抗指定レジスタ・グループB (POGB) を操作することにより行います。

2.2.4 TI0...ポート1兼用入出力

プログラマブル・タイマ/イベント・カウンタの外部イベント・パルス入力端子です。

TI0はシュミット・トリガ入力になっています。

2.2.5 PTO0, PTO1...ポート2兼用出力

プログラマブル・タイマ/イベント・カウンタ, およびプログラマブル・タイマ・カウンタの出力端子です。方形波パルスを出力します。プログラマブル・タイマ/イベント・カウンタ, およびプログラマブル・タイマ・カウンタの信号を出力する場合は、P20またはP21の出力ラッチをクリア(0)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)にします。

タイマ・スタート命令により、出力はクリア(0)されます。

2.2.6 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、A/Dコンバータなど）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・モード・レジスタ（CLOM）がクリア（0）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。

2.2.7 BUZ...ポート2 兼用出力

任意の周波数（2.048, 4.096, 32.768 kHz）を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（WM.7）を命令でセット（1）にすることによってブザー出力として動作させることができます。

$\overline{\text{RESET}}$ 信号発生により、通常のポートとしての動作モードになります（詳細は5.4.2 時計モード・レジスタを参照）。

2.2.8 $\overline{\text{SCK}}$, SO/SB0, SI/SB1...ポート0 兼用, 3ステート入出力

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）の設定にしたがって動作します。

$\overline{\text{RESET}}$ 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。

2.2.9 INT4...ポート0 兼用入力

立ち上がりおよび立ち下がり両エッジ・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リクエスト・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・レベル幅を有する信号が入力されると受け付けられます。

INT4はSTOP, 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。

2.2.10 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はノイズ除去機能をもっています。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) 立ち下がりエッジ・アクティブ
- (c) 立ち上がりおよび立ち下がり両エッジ・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) 立ち下がりエッジ・アクティブ

INT0はノイズ除去機能付きで、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。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。

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・レベル幅を有する入力があれば受け付けられます。

$\overline{\text{RESET}}$ 信号発生により、IM0, IM1はクリア(0)され、立ち上がりエッジ・アクティブが選択されま

す。

INT1はSTOPモード、HALTモードの解除にも利用できますが、INT0はSTOPモード、HALTモードの解除に利用することはできません。

INT0, INT1は、シュミット・トリガ入力となっています。

2.2.11 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・レベル幅を有する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はクリア(0)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。

2.2.12 KR0-KR3...ポート6兼用入力 KR4-KR7...ポート7兼用入力

キー割り込み入力端子です。KR0-KR7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)にしたがって割り込みフォーマットを指定できます。

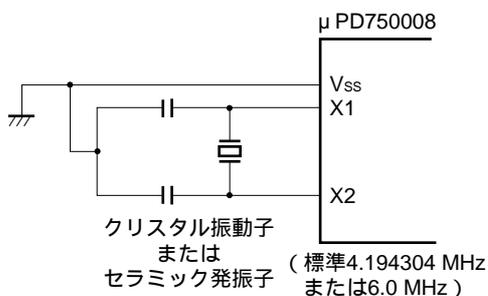
$\overline{\text{RESET}}$ 信号発生により、ポート6、ポート7の入力モードになります。

2.2.13 X1, X2

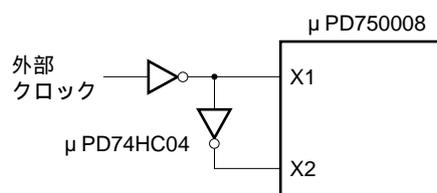
メイン・システム・クロック発振用クリスタル/セラミック接続端子です。

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

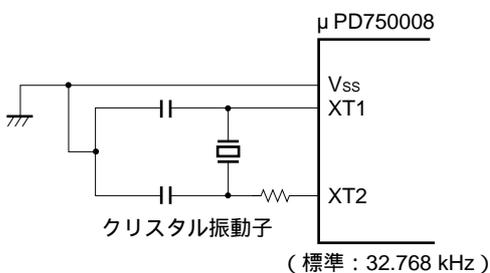


2.2.14 XT1, XT2

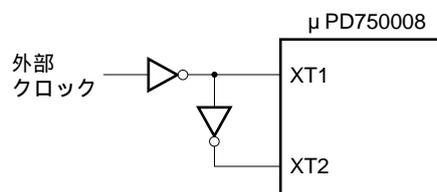
サブシステム・クロック発振用クリスタル接続端子です。

外部クロックを入力することも可能です。

(a) クリスタル発振



(b) 外部クロック



2.2.15 $\overline{\text{RESET}}$

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・レベル幅を有する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、スタンバイ (STOP/HALT) モードの解除にも使用されます。

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

2.2.16 VDD

正電源供給端子です。

2.2.17 VSS

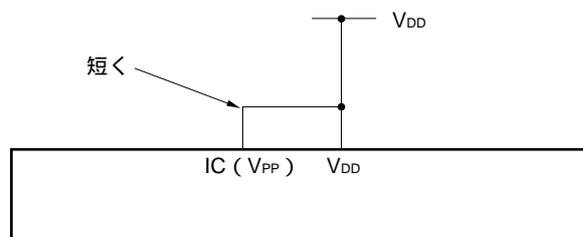
GND電位です。

2.2.18 IC (μ PD750004, 750006, 750008のみ)

IC (Internally Connected) 端子は、当社出荷時に μ PD750008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VDD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VDD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VDD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端子をVDD端子に直接接続してください



2.2.19 VPP (μ PD75P0016のみ)

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端子です。通常動作時は、VDDと直接接続します (上図参照)。PROM書き込み / ベリファイ時は、+ 12.5 Vを印加します。

2.2.20 MD0-MD3 (μ PD75P0016のみ) ...ポート3兼用入出力

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

2.3 端子の入出力回路

μPD750008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。

図2 - 1 端子の入出力回路一覧 (1/2)

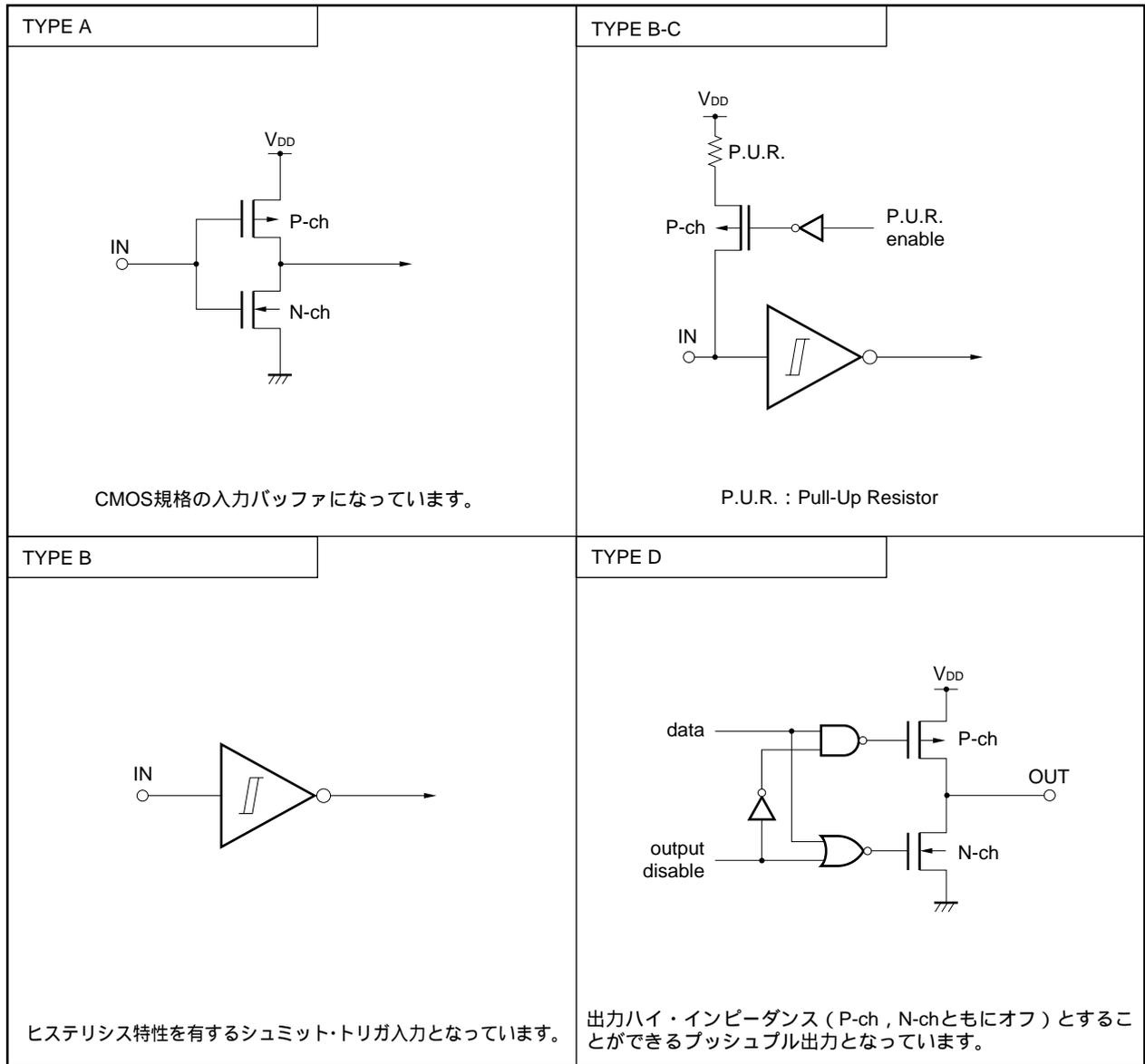
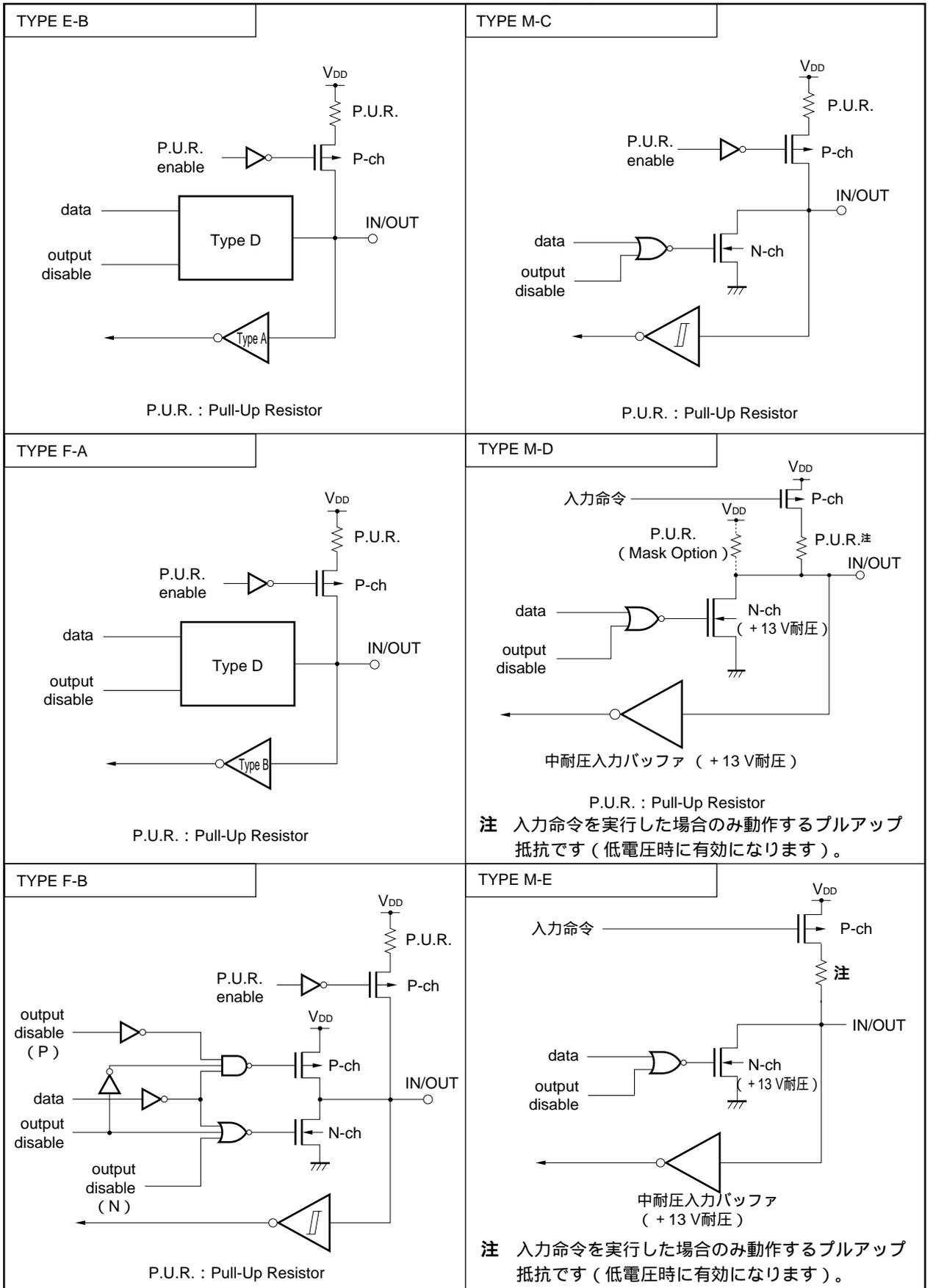


図 2 - 1 端子の入出力回路一覧 (2/2)



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 _{SS} に接続
P01/ $\overline{\text{SCK}}$	V _{SS} またはV _{DD} に接続
P02/SO/SB0	
P03/SI/SB1	
P10/INT0-P12/INT2	V _{SS} に接続
P13/TI0	
P20/PTO0	入力状態：個別に抵抗を介してV _{SS} またはV _{DD} に接続 出力状態：オープン
P21/PTO1	
P22/PCL	
P23/BUZ	
P3 α /MD0)-P3 α /MD3) ^注	
P40-P43	
P50-P53	
P60-P63	
P70-P73	
P80-P81	
XT1	V _{SS} またはV _{DD} に接続
XT2	オープン
IC (V _{PP}) ^注	V _{DD} に直接接続

注 ()内は μ PD75P0016の場合です。

第3章 アーキテクチャの特徴とメモリ・マップ

μPD750008に採用されている75XLシリーズのアーキテクチャでは

- ・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

などの優れた特徴を実現するために以下の方法を採用しています。

- （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

この章では、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

μPD750008ではデータ・メモリ空間の000H-1FFH番地にスタティックRAMが512ワード × 4ビット内蔵されています。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。この12ビット・アドレス（4K × 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、μPD750008では下位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、次の2つのハードウェアが内蔵されています。

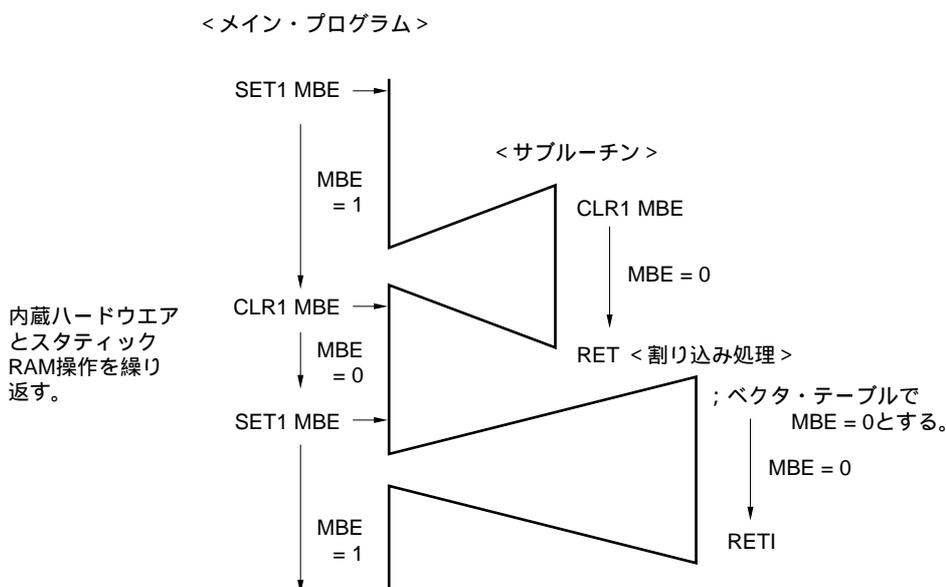
- ・メモリ・バンク許可フラグ（MBE）
- ・メモリ・バンク選択レジスタ（MBS）

MBSはメモリ・バンクを選択するレジスタで、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



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から高速の割り込み処理に有利となっています。

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

μPD750008に採用されている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図3-2, 3-3, 表3-1にまとめたような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のスタティックRAM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

アドレッシング・モード	mem mem.bit		@HL @H + mem.bit		@DE @DL	スタック・アドレッシング	fmem.bit	pmem.@L
	MBE =0	MBE =1	MBE =0	MBE =1	-	-	-	-
メモリ・バンク許可フラグ								
000H ↑ 汎用レジスタ・エリア ↓ 01FH 020H 07FH		MBS =0		MBS =0		SBS =0		
データ・エリア スタティックRAM (メモリ・バンク0)								
0FFH ↑ 100H		MBS =1		MBS =1		SBS =1		
データ・エリア スタティックRAM (メモリ・バンク1)								
1FFH ↓								
内蔵されていない								
F80H								
↑ 周辺ハードウェア・エリア (メモリ・バンク15) ↓ FC0H		MBS =15		MBS =15				
データ・エリア スタティックRAM (メモリ・バンク15)								
FFFH								

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。
	@HL +	ただし, MB = MBE · MBS
	@HL -	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。 ただし, fmem = $\begin{cases} \text{FB0H-FBFH (割り込み関係のハードウェア)} \\ \text{FF0H-FFFH (I/Oポート)} \end{cases}$
	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のスタティックRAM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           ; MBE 1
      SEL MB1           ; MBS 1
      MOV D, #DATA1 SHR 4 ; D 5
      MOV HL, #DATA2 AND 0FFH ; HL 17H
LOOP : MOV A, @DL       ; A (DL)
      XCH A, @HL -      ; A (HL), 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。

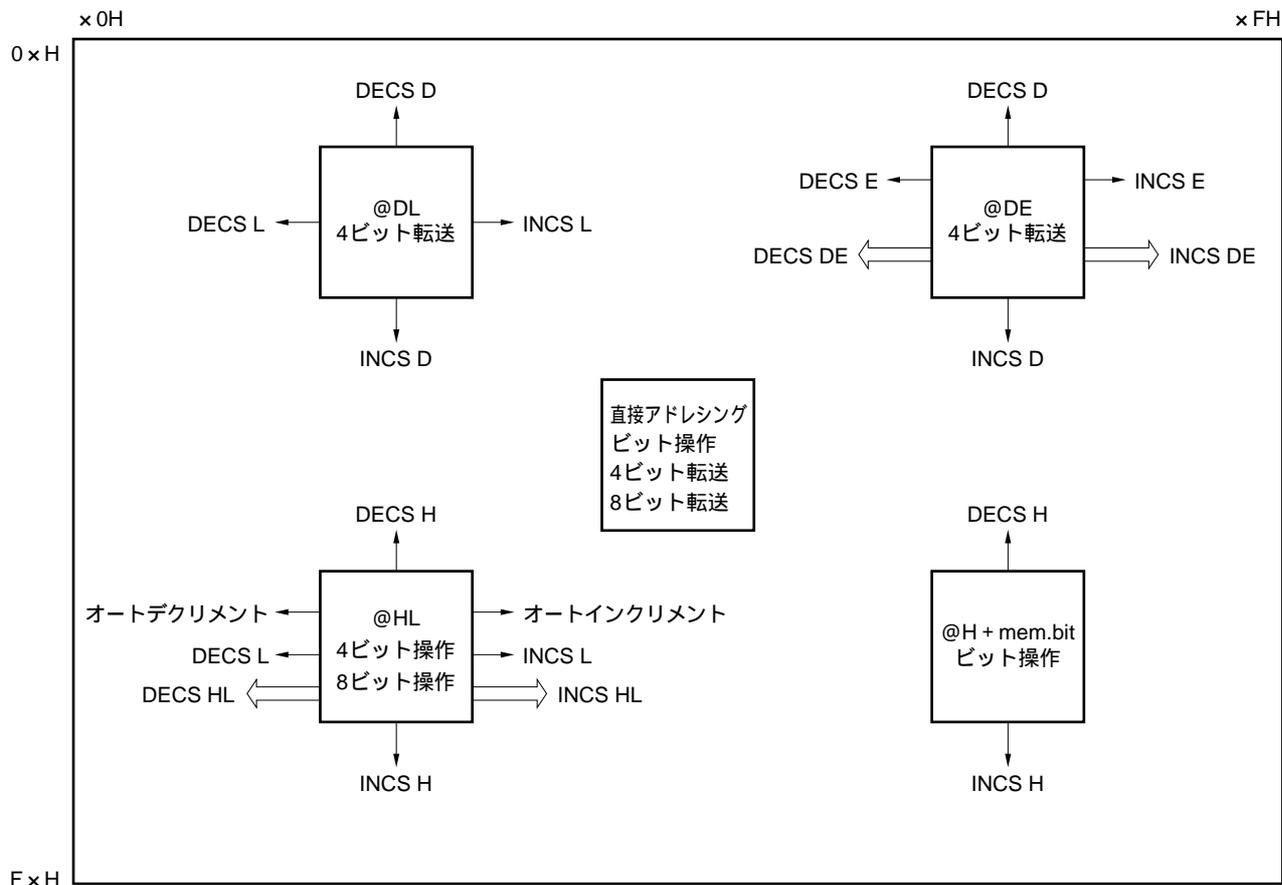
例1 . 50H-57Hのデータと110H-117Hのデータを比較する。

```
DATA1 EQU 57H
DATA2 EQU 117H
SET1 MBE
SEL MB1
MOV D, #DATA1 SHR 4
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 HL ; HL HL + 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命令に適用されます。

例1．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 XA, @HL ; XA = (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, XA ; (HL) XA
      INCS HL
      INCS HL
      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、MOV1命令によるビット転送、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 . P02端子入力を反転してP33端子から出力する。

```
MOV1  CY, PORT0.2
NOT1  CY
MOV1  PORT3.3, CY
```

2 . タイマ0 割り込みリクエスト・フラグ (IRQT0) をテストし、セットされていたらリクエスト・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
```

3 . P30とP41の端子がともに1であれば、P53をリセットする。



```
MOV1  CY, PORT3.0  ; CY  P30
AND1  CY, PORT4.1  ; CY  P41
NOT1  CY           ; CY   $\overline{CY}$ 
MOV1  PORT5.3, CY  ; P53  CY
```

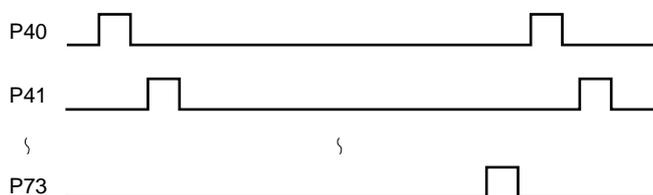
(b) 特定アドレス・ビット・レジスタ間接アドレッシング (pmem.@L)

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7の各ビットに順にパルス出力する。



```

MOV    L, #0
LOOP:  SET1   PORT4.@L; ポート4-7のビット(L1-0)  1
      CLR1   PORT4.@L;           "              0
      INCS   L
      NOP
      BR    LOOP
    
```

(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
MOV1 CY, @H + FLAG1 ; CY FLAG1
XOR1 CY, @H + FLAG2 ; CY CY ∨ FLAG2
MOV1 @H + FLAG3, CY ; FLAG3 CY
    
```

(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

μPD750008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

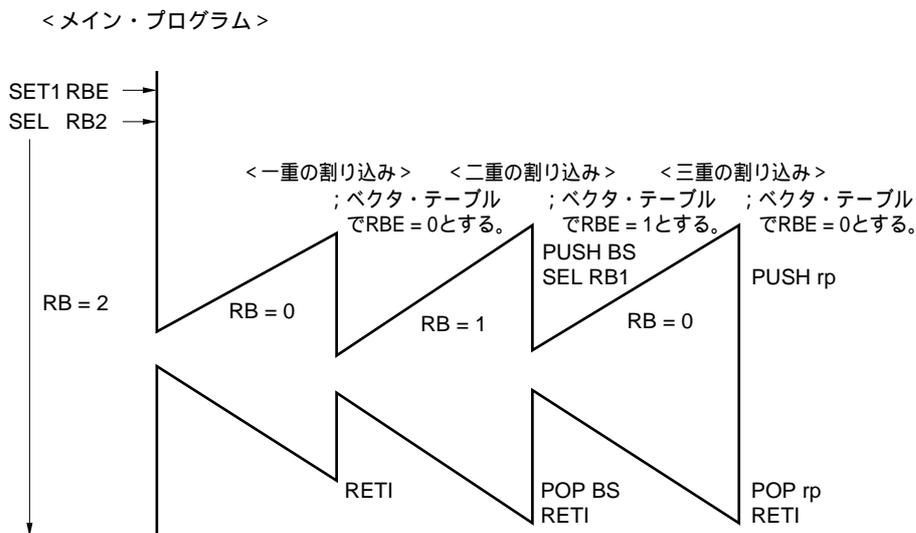
備考 x : don't care

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/復帰をする。

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

μPD750008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，比較など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、 $RB = RBE \cdot RBS$ 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・比較など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

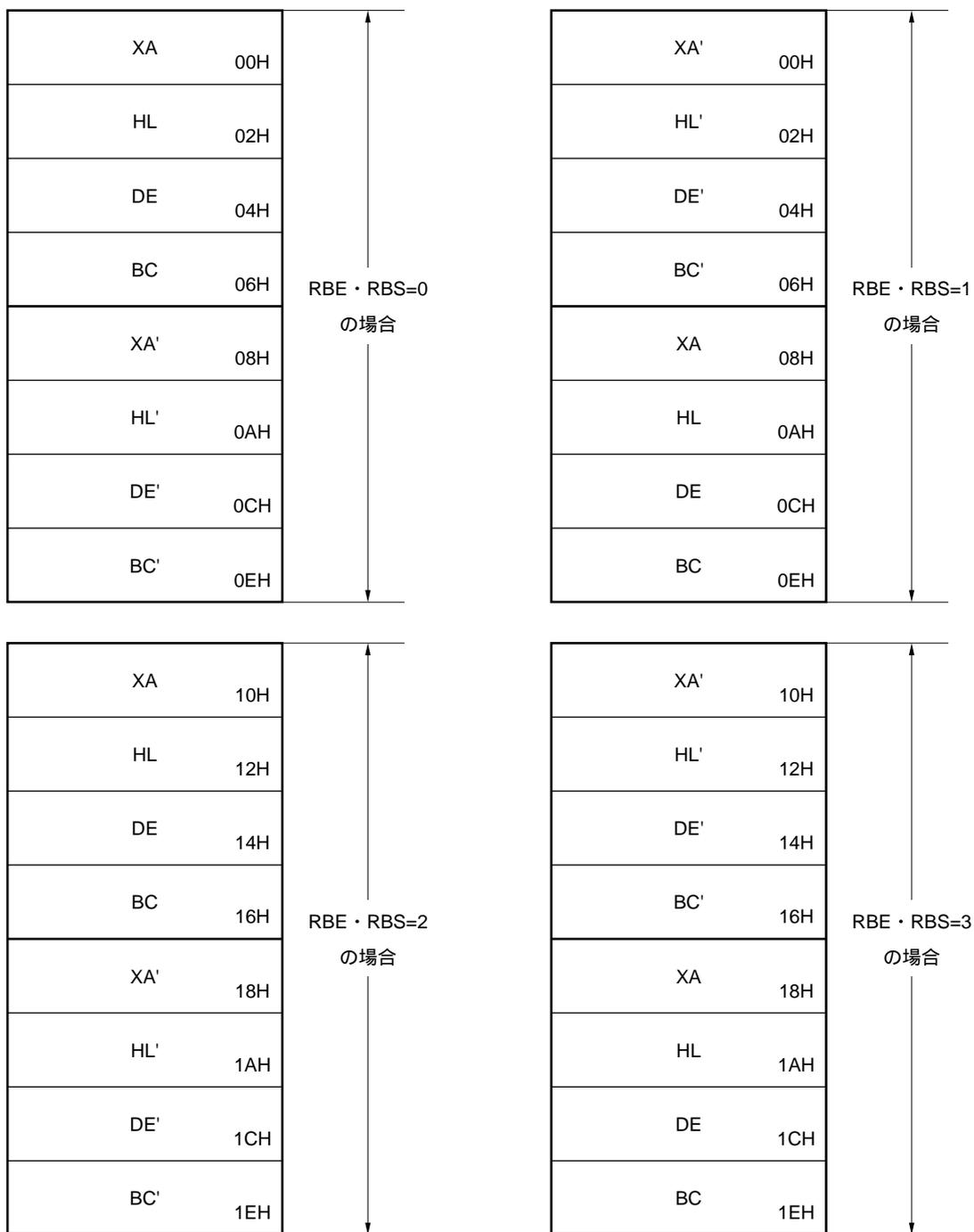
2 . タイマ/イベント・カウンタ0のカウンタ・レジスタ(T0)の値がBC'レジスタ・ペアの値より大きいかどうかをテストし大きくなるまで待つ。

```
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

μPD750008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ニ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	

図3-7にμPD750008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

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 μPD750008 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)				R/W	-		-	mem.bit	
F85H	スタック・バンク選択レジスタ (SBS)				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・モード・レジスタ (BTM)				R	-	-		-	
F8BH	ベーシック・インターバル・タイマ (BT)				W		-	-	mem.bit	
F8BH	WDTM ^{注2}				W		-	-	mem.bit	
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・テスト可能
						-	-		-	

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 μPD750008 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		
FA0H	タイマ/イベント・カウンタ・ モード・レジスタ (TM0)				R/W	(W)	-	(R/W)	mem.bit	ビット3のみ ビット操作W可能
						-	-		-	
FA2H	TOE0 ^{注1}				W		-	-	mem.bit	
FA4H	タイマ/イベント・カウンタ・ カウント・レジスタ (T0)				R	-	-		-	
FA6H	タイマ/イベント・カウンタ・ モジュロ・レジスタ (TMOD0)				R/W	-	-		-	
FA8H	タイマ・カウンタ・モード・ レジスタ (TM1)				R/W	(W)	-	(R/W)	mem.bit	ビット3のみ ビット操作W可能
						-	-		-	
FAAH	TOE1 ^{注2}				W		-	-	mem.bit	
FACH	タイマ・カウンタ・カウント・ レジスタ (T1)				R	-	-		-	
FAEH	タイマ・カウンタ・モジュロ・ レジスタ (TMOD1)				R/W	-	-		-	

注1 . TOE0 : タイマ/イベント・カウンタ出力許可フラグ (W)

2 . TOE1 : タイマ・カウンタ出力許可フラグ (W)

図3 - 7 μPD750008 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)					-	-			
	CY	SK2	SK1	SK0						
FB2H	割り込みプライオリティ選択レジスタ (IPS)				R/W	-	-			注1
FB3H	プロセッサ・クロック・コントロール・レジスタ (PCC)				R/W	-	-			注2
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3, 2, 1は0に固定
FB6H	INT2エッジ検出モード・レジスタ (IM2)				R/W	-	-			ビット3, 2, は 0に固定
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1, 2は 0に固定
FB8H	IE4	IRQ4	IEBT	IRQBT	R/W			-	fmem.bit	
FBAH	/	/	IEW	IRQW	R/W					
FBCH	IET1	IRQT1	IET0	IRQT0	R/W			-		
FBDH	/	/	IECSI	IRQCSI	R/W					
FBEH	IE1	IRQ1	IE0	IRQ0	R/W			-		
FBFH	/	/	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 . ビット3のみEI/DI命令により操作可能。

2 . ビット3, 2はSTOP/HALT命令実行時ビット操作可能。

図3-7 μPD750008 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	-		-	-	
FDCH	ブルアップ抵抗指定レジスタ・ グループA(POGA)				R/W	-	-		-	
FDEH	ブルアップ抵抗指定レジスタ・ グループB(POGB)				R/W	-	-		-	

FE0H	シリアル動作モード・レジスタ(CSIM)				R/W	-	-		-	注
	CSIE	COI	WUP			(R/W)	-		mem.bit	
FE2H	CMDDD	RELD	CMDT	RELT	R/W		-	-	mem.bit	R/Wは,ビットにより異なります。
	SBIコントロール・レジスタ(SBIC)					BSYE	ACKD	ACKE	ACKT	
FE4H	シリアルI/Oシフト・レジスタ(SIO)				R/W	-	-		-	
FE6H	スレーブ・アドレス・レジスタ(SVA)				R/W	-	-		-	
FE8H	PM33	PM32	PM31	PM30	R/W	-	-		-	
	ポート・モード・レジスタ・グループA(PMGA)					PM63	PM62	PM61	PM60	
FECH	-	PM2	-	-	R/W	-	-		-	
	ポート・モード・レジスタ・グループB(PMGB)					PM7	-	PM5	PM4	
FEEH	-	-	-	PM8	R/W	-	-		-	
	ポート・モード・レジスタ・グループC(PMGC)					-	-	-	-	

注 1ビット操作は,ビットによりR/Wが異なります。

図3 - 7 μPD750008 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 (PORT0) SCKP _ _ _ _				R/W	(R)R/W	(R)	-	fmem.bit pmem.@L	注1
FF1H	ポート1 (PORT1)				R					
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 ^{注2}	KR3	KR2	KR1	KR0	R/W					
ポート6 (PORT6)										
FF7H ^{注2}	KR7	KR6	KR5	KR4	R/W					
ポート7 (PORT7)										
FF8H	ポート8 (PORT8)				R/W			-		

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-KR7は1ビット単位での読み出し (R) のみ可能です。4ビット並列入力時はPORT6またはPORT7で指定します。

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

μPD750008サブシリーズの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：μPD75008サブシリーズと上位互換性があります。
ROMが16 Kバイトまでの75XL CPUで使用できます。
- ・Mk モード：μPD75008サブシリーズとの互換性はありません。
ROM16 Kバイト以上の製品も含め、75XLのCPU全部で使用できます。

表4 - 1に、Mk モードとMk モードとの違いを示します。

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	不定動作	正常動作
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、プログラム・メモリが24 Kバイト以上の75Xシリーズや75XLシリーズとのソフトウェア上の互換性を保つためのものです。

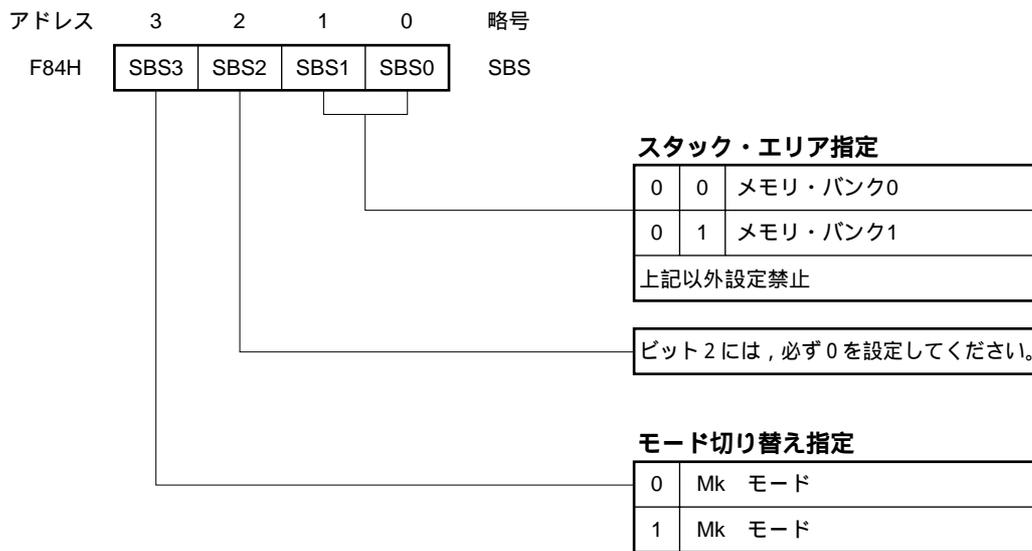
したがって、ROM効率やスピードを重視する場合はMk モードを使用してください。

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10××B[≠]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00××B[≠]にイニシャライズしてください。

図4 - 1 スタック・バンク選択レジスタのフォーマット



注 ××には希望の値を設定してください。

注意 SBS.3は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 (μ PD750004)
 13ビット (μ PD750006, 750008)
 14ビット (μ PD75P0016)

プログラム・メモリのアドレス情報を保持するバイナリ・カウンタです。 μ PD750004では12ビットの構成 (図4 - 2 (a) 参照) で、 μ PD750006, 750008では13ビットの構成 (図4 - 2 (b) 参照) で、 μ PD75P0016では14ビットの構成 (図4 - 2 (c) 参照) です。

図4 - 2 プログラム・カウンタの構成

(a) μ PD750004の場合

PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

(b) μ PD750006, 750008の場合

PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

(c) μ PD75P0016の場合

PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
------	------	------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

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。

$\overline{\text{RESET}}$ 信号発生により、プログラム・カウンタはプログラム・メモリの000H、001H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。

μ PD750004 :

PC₁₁-PC₈ (000H)₃₋₀, PC₇-PC₀ (001H)₇₋₀

μ PD750006, 750008 :

PC₁₂-PC₈ (000H)₄₋₀, PC₇-PC₀ (001H)₇₋₀

μ PD75P0016 :

PC₁₃-PC₈ (000H)₅₋₀, PC₇-PC₀ (001H)₇₋₀

4.3 プログラム・メモリ (ROM)

4096ワード×8ビット (μ PD750004 : マスクROM)

6144ワード×8ビット (μ PD750006 : マスクROM)

8192ワード×8ビット (μ PD750008 : マスクROM)

16384ワード×8ビット

(μ PD75P0016 : ワン・タイムPROM)

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・データ等を格納します。 μ PD750004、 μ PD750006、 μ PD750008ではマスク・プログラマブルROMで、 μ PD75P0016ではワン・タイムPROMです。

図4-3～4-6にプログラム・メモリ・マップを示します。

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なアドレス範囲は、図4-3～4-6に示すとおりです。相対分岐命令(BR \$addr)ではブロックに関係なく、[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 : μ PD750004

0000H-17FFH : μ PD750006

0000H-1FFFH : μ PD750008

0000H-3FFFH : μ PD75P0016

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。

0000-0001H番地

$\overline{\text{RESET}}$ 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。

0002-000DH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。

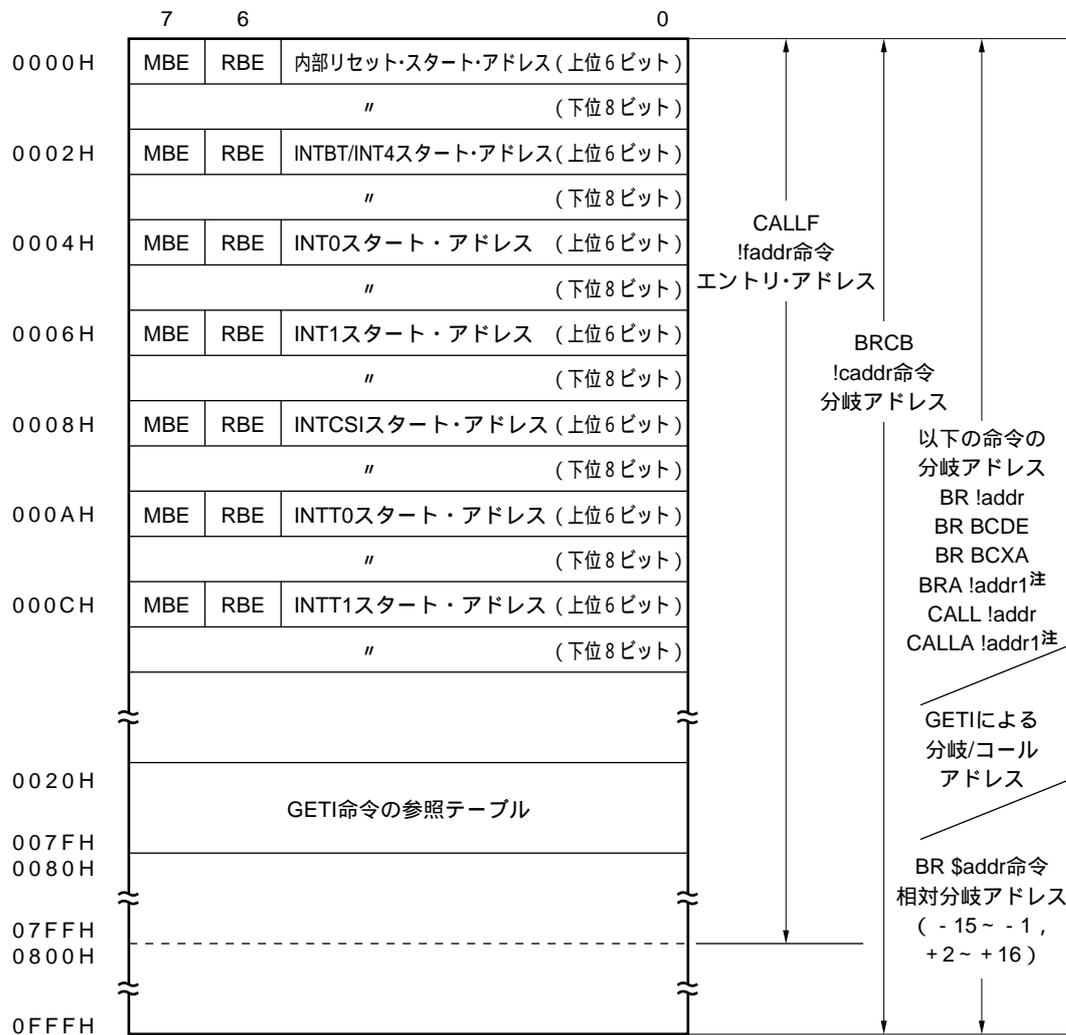
任意の番地から割り込み処理をスタートできます。

0020-007FH番地

GETI命令^注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)。

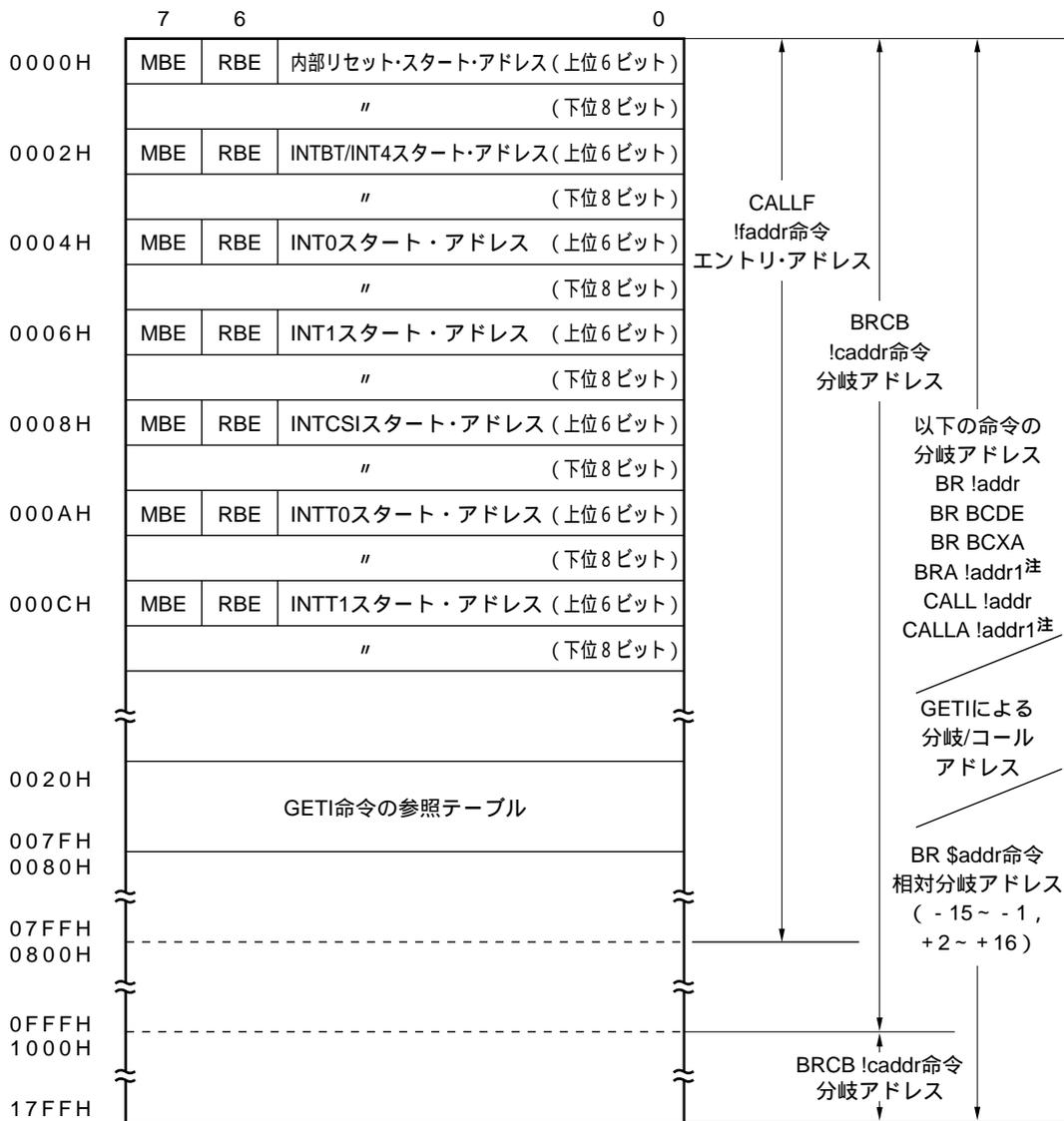
図4 - 3 プログラム・メモリ・マップ (μPD750004)



注 Mk モードでのみ使用できます。

備考 上記以外では、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。

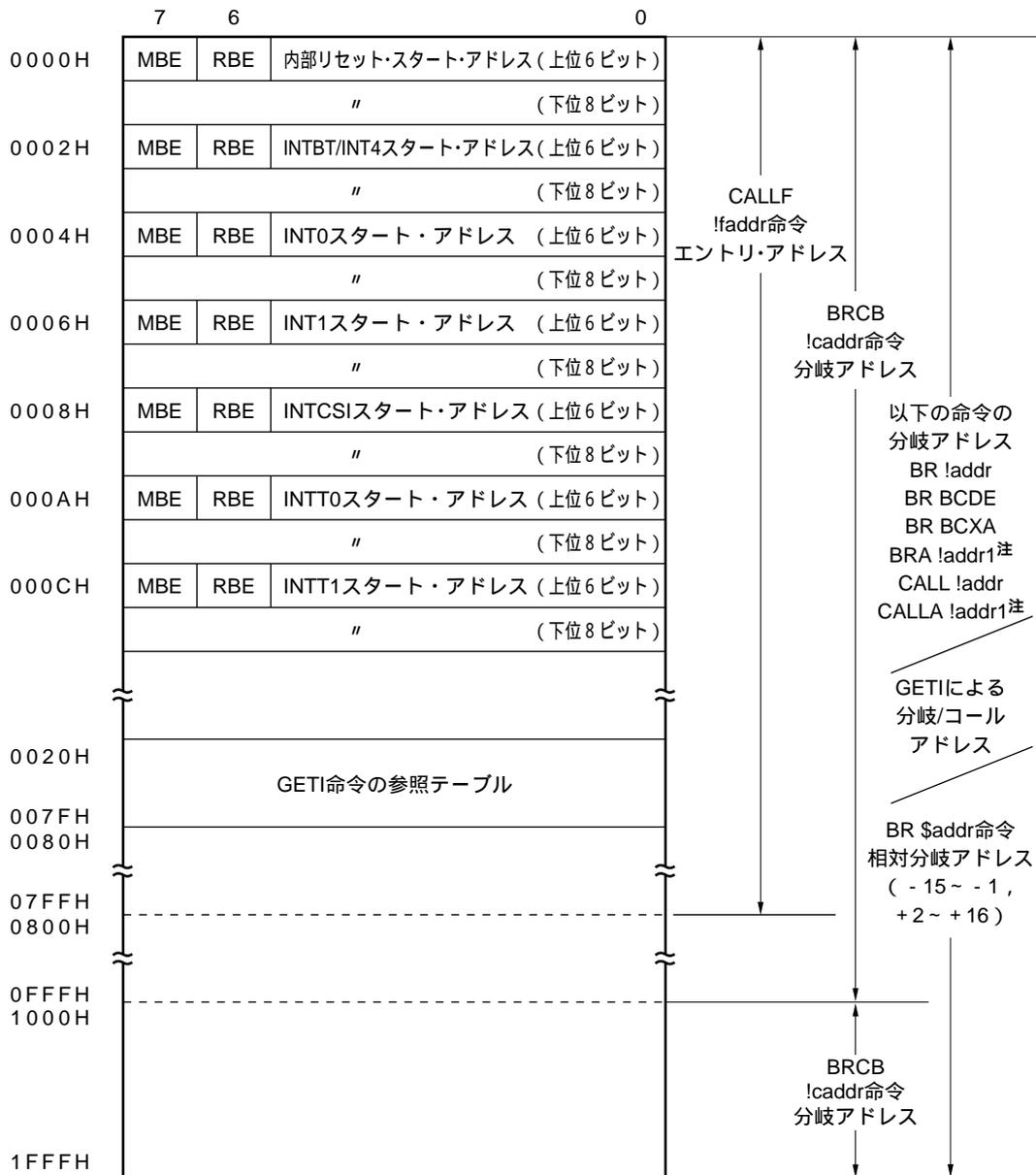
図4-4 プログラム・メモリ・マップ(μPD750006)



注 Mk モードでのみ使用できます。

備考 上記以外では、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。

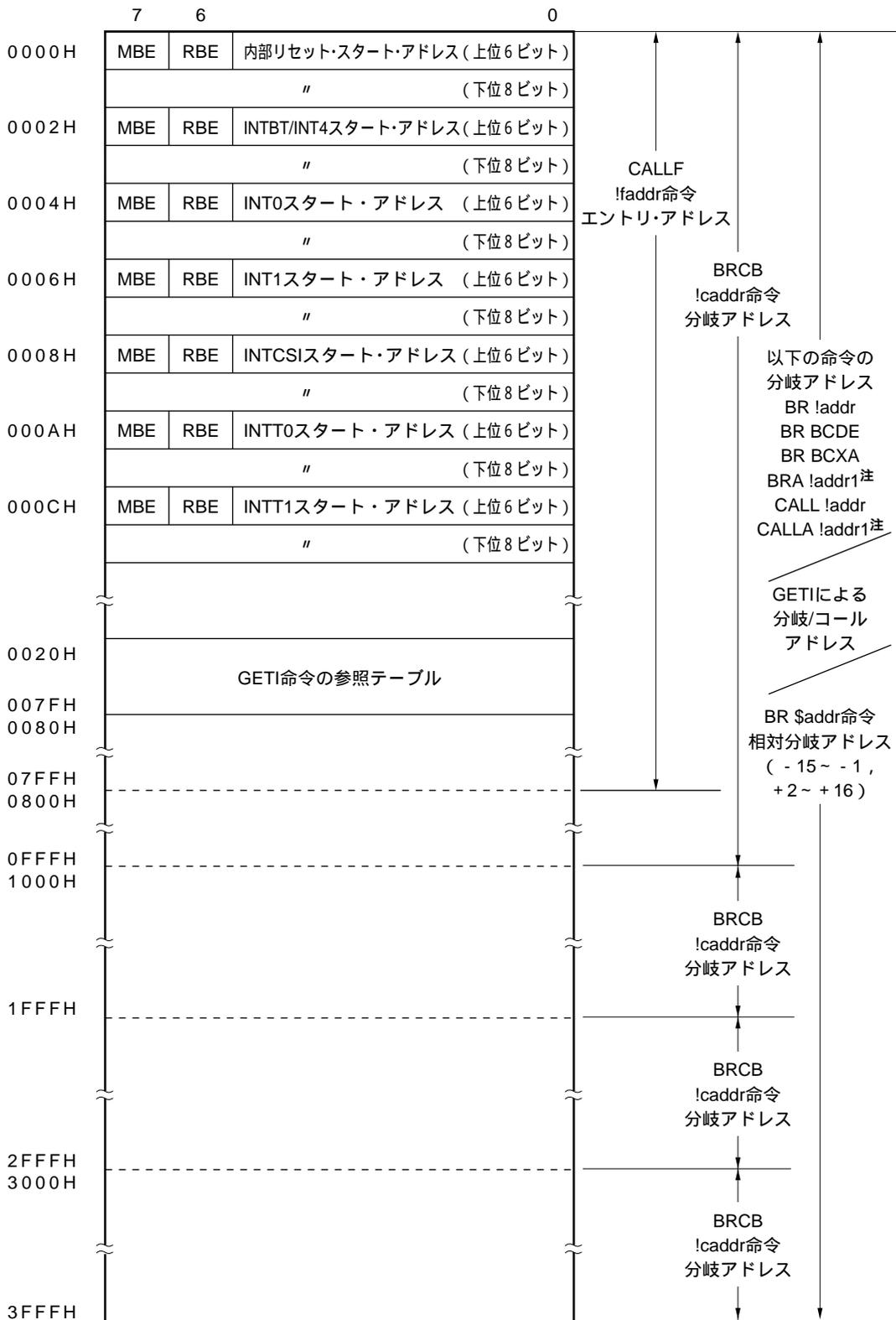
図4 - 5 プログラム・メモリ・マップ (μ PD750008)



注 Mk モードでのみ使用できます。

備考 上記以外では、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4-6 プログラム・メモリ・マップ (μPD75P0016)



注 Mk モードでのみ使用できます。

備考 上記以外では、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 - 7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、プログラムのデータのストアや、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 × 4ビットずつマッピングされています。バンク0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^注 (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で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・エリアは、メモリ・バンク0, 1から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-FFFH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(図3-7 μ PD750008I/Oマップを参照)。

4.4.2 データ・メモリのバンクの指定

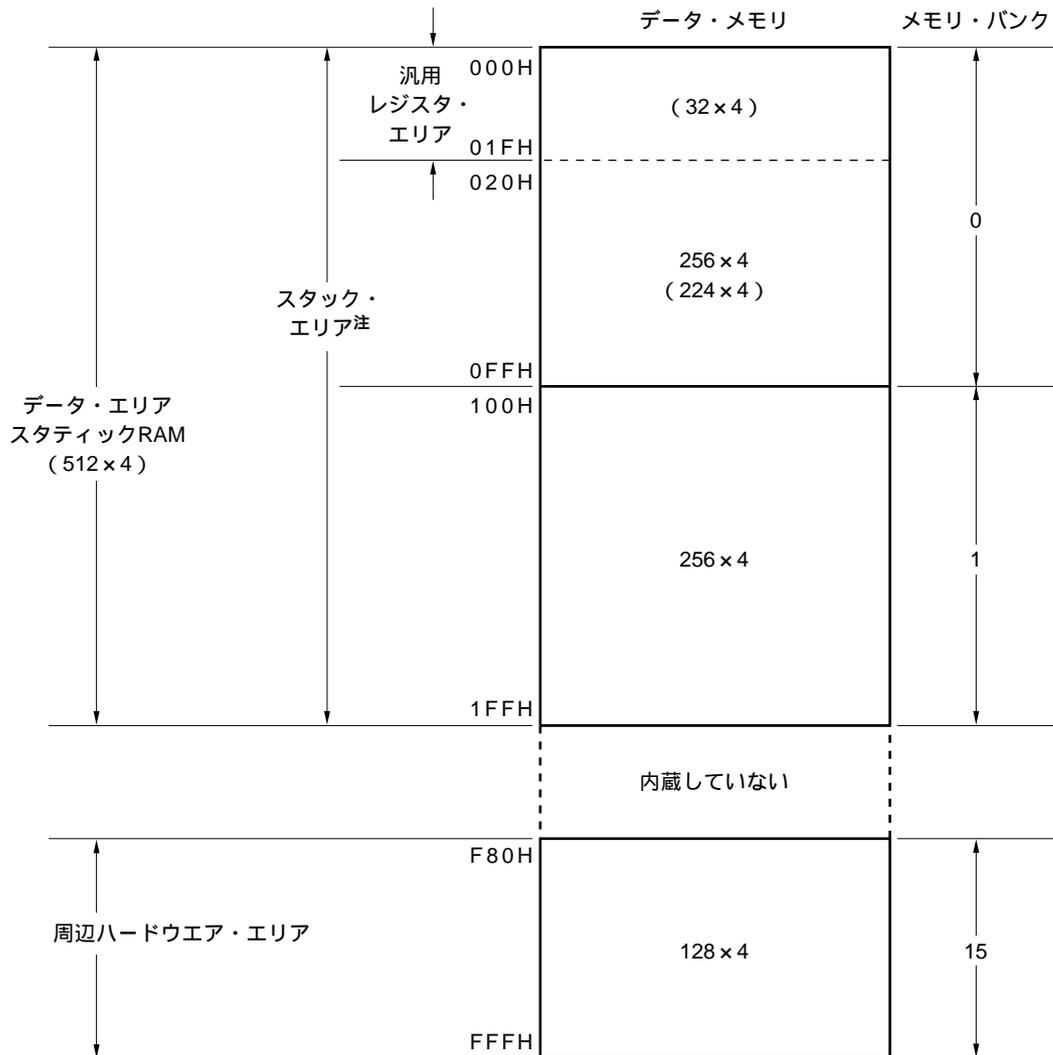
メモリ・バンクは、メモリ・バンク許可フラグ(MBE)によってバンク指定を許可したとき(MBE=1)、4ビット・メモリ・バンク選択レジスタ(MBS)によって指定されます(MBS=0, 1, 15)。バンク指定を禁止したとき(MBE=0)、メモリ・バンクはそのときのアドレッシング・モードによってバンク0または15が自動的に指定されます。また、バンク内のアドレスは、8ビット・イミディエト・データやレジスタ・ペアなどによってアドレスされます。

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・メモリ・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.....	第5章	周辺ハードウェア機能

図4-7 データ・メモリ・マップ



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
```

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・ポインタとして使用することができます。

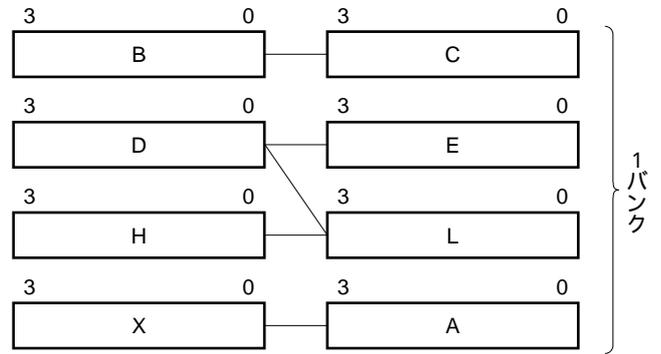
8ビット操作時は、BC, DE, HL, XAのほか、レジスタ・バンク（RB）のビット0を反転したレジスタ・バンク（0 1, 2 3）のレジスタ・ペアをBC', DE', HL', XA'として使用できます（3.2 汎用レジスタのバンク構成参照）。

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 - 8 汎用レジスタの構成



図4 - 9 レジスタ・ペアの構成

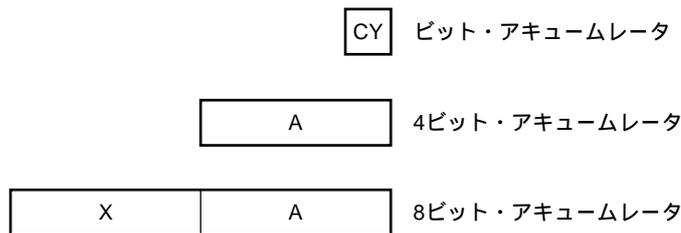


4.6 アキュムレータ

μPD750008では、Aレジスタ、XAレジスタ・ペアがアキュ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・アキュムレータとして機能します。

図4 - 10 アキュムレータ



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)

μPD750008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 - 12 ~ 4 - 15のようになります。

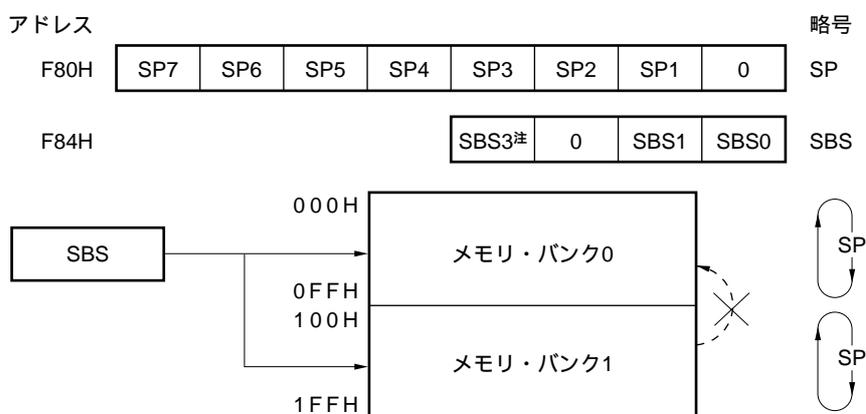
SPは、8ビット・メモリ操作命令、SBSは4ビット・メモリ操作命令によって初期値を設定し、スタック・エリアを決定します。また内容の読み出しもできます。

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。

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。

図4 - 11 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）。

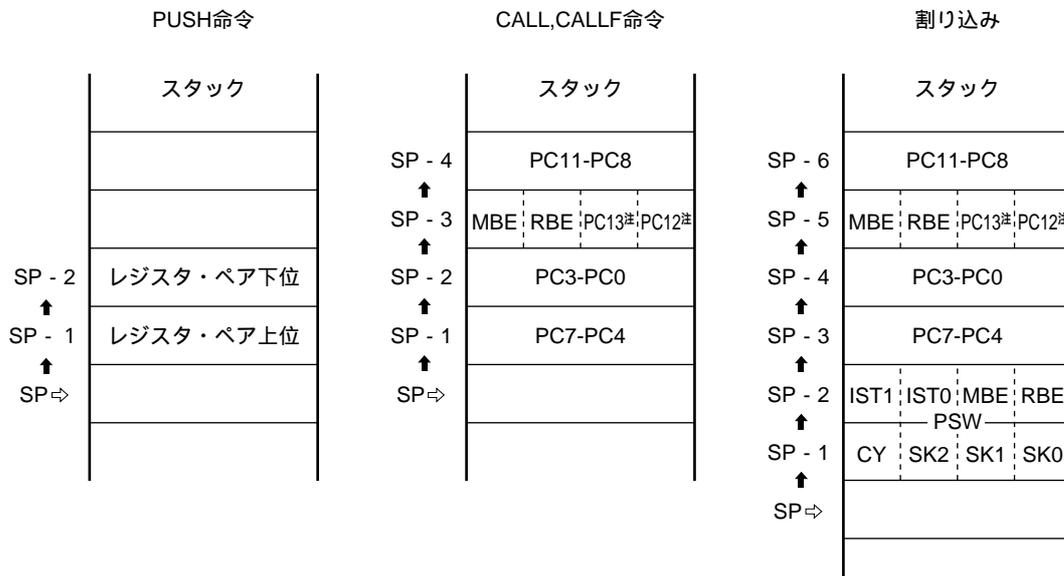
例 SPのイニシャライズ

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

```

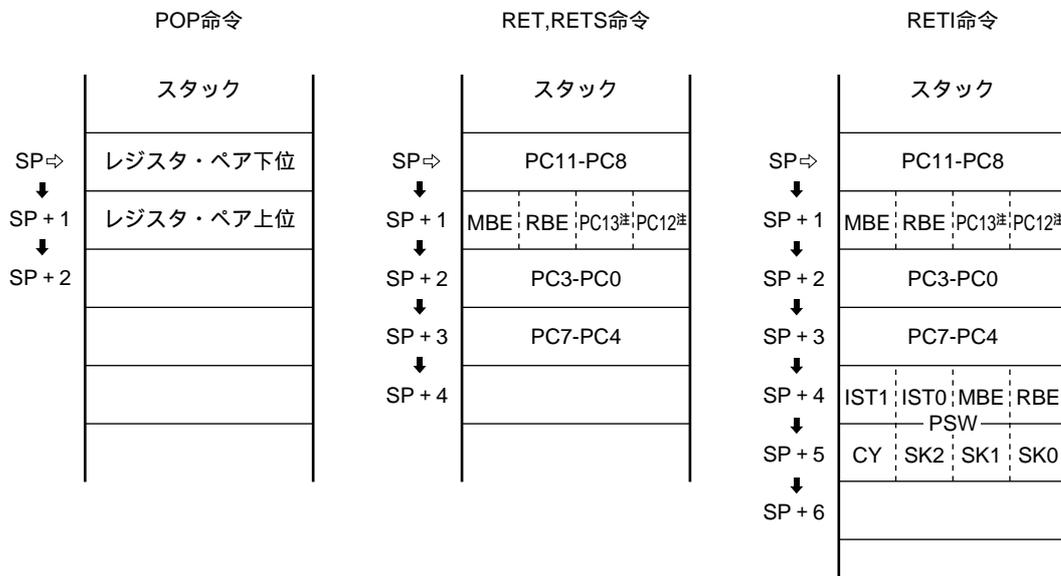
SEL    MB15      ;またはCLR1 MBE
MOV    A, # 1
MOV    SBS, A    ;メモリ・バンク1をスタック・エリア指定
MOV    XA, #00H
MOV    SP, XA    ;SP 00H
    
```

図4 - 12 スタック・メモリへ退避されるデータ (Mk I モード)



注 μ PD750004ではPC12 = 0 , PC13 = 0 です。 μ PD750006 , 750008ではPC13 = 0 です。

図4 - 13 スタック・メモリから復帰されるデータ (Mk I モード)



注 μ PD750004ではPC12 = 0 , PC13 = 0 です。 μ PD750006 , 750008ではPC13 = 0 です。

図4 - 14 スタック・メモリへ退避されるデータ (Mk II モード)

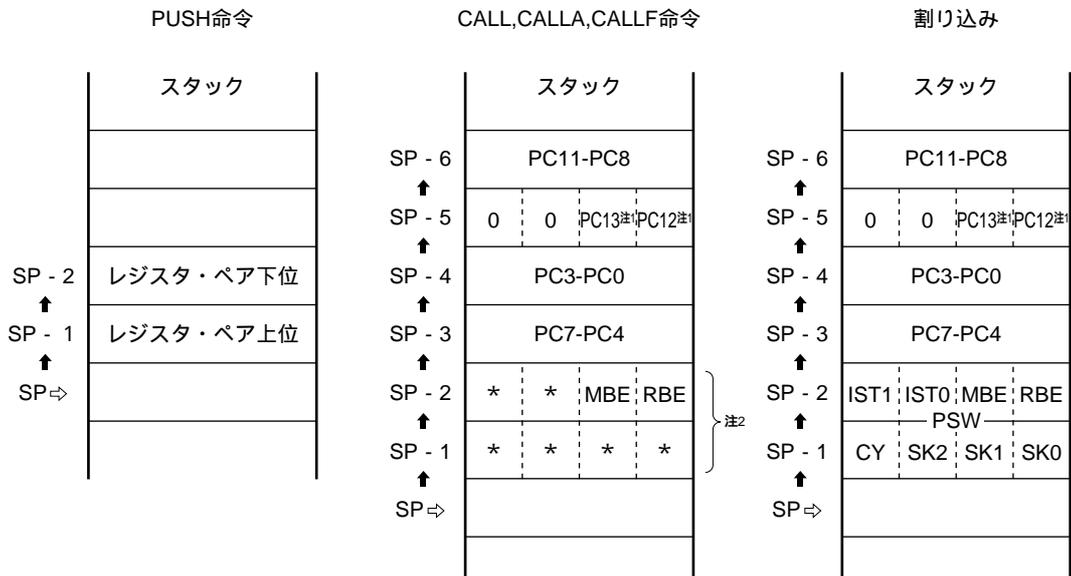
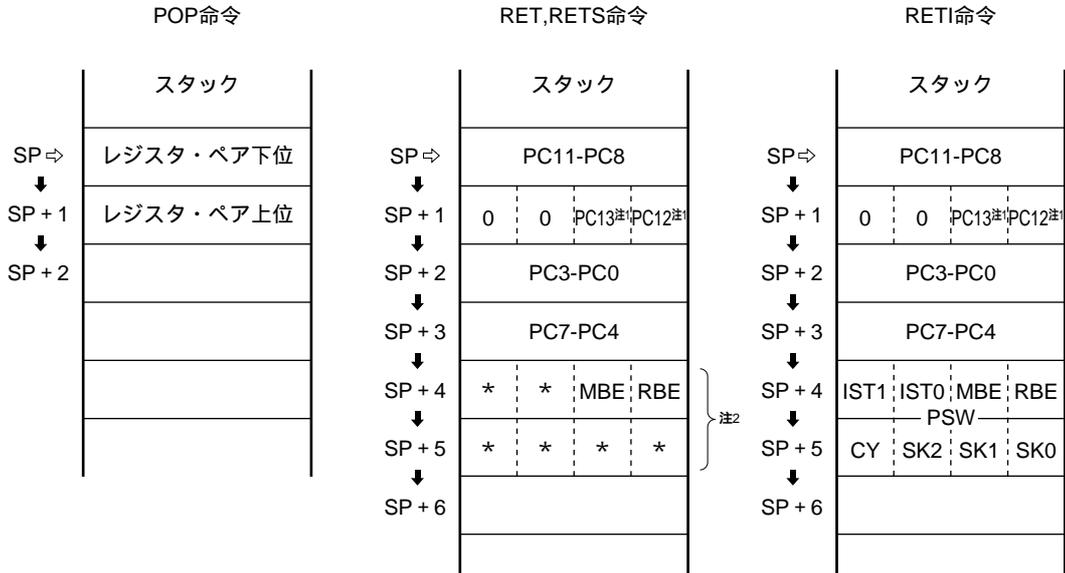


図4 - 15 スタック・メモリから復帰されるデータ (Mk II モード)



注1 . μ PD750004ではPC12 = 0 , PC13 = 0です。 μ PD750006 , 750008ではPC13 = 0です。

2 . MBE , RBE以外のPSW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 - 16 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 (ニモニック)	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	他の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, 他のPSWと並列に復帰

備考 mem*.bitは次の3通り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・アドレスをセット
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。

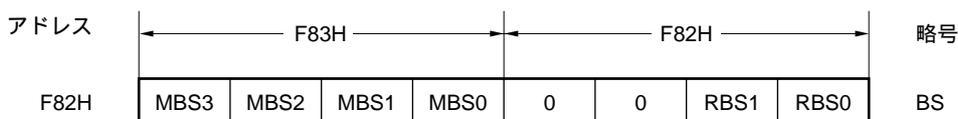
4.9 バンク選択レジスタ (BS)

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 - 17 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、μPD750008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

μPD750008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
FF7H	P73	P72	P71	P70	PORT7
FF8H	-	-	P81	P80	PORT8

備考 - 部はスタティックRAMとして使用可能。

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からPORT7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4-7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(略号)	機能	動作・特徴	備考
PORT0	4ビット入出力	兼用端子の動作モードにかかわらず，常に読み込みまたはテスト可能。	INT4, \overline{SCK} , SO/SB0, SI/SB1と端子を兼用。
PORT1			INT0-2, TIOと端子を兼用。
PORT3 ^{注1}	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。	MD0-MD3と端子を兼用 ^{注2} 。
PORT6			KR0-KR3と端子を兼用。
PORT2			PTO0, PTO1, PCL, BUZと端子を兼用。
PORT7	KR4-7と端子を兼用。		
PORT4 ^{注1}	4ビット入出力 (N-chオープン・ドレイン13V耐圧)	4ビット単位で入力または出力モードに指定可能。 ポート4と5はペアとなって8ビット単位でデータの入出力が可能。	マスク・オプションで，1ビット単位にプルアップ抵抗の内蔵を指定可能 ^{注3} 。
PORT5 ^{注1}			
PORT8	2ビット入出力	2ビット単位で入力または出力モードに設定可能。	

注1．LEDを直接駆動可能。

2． μ PD75P0016のみ兼用しています。

3． μ PD75P0016にはマスク・オプションはなく，プルアップ抵抗は内蔵できません。

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，ポート2～ポート8の出力ラッチはクリアされ，出力バッファはオフし，入力モードになります。

図5 - 2 ポート0, 1の構成

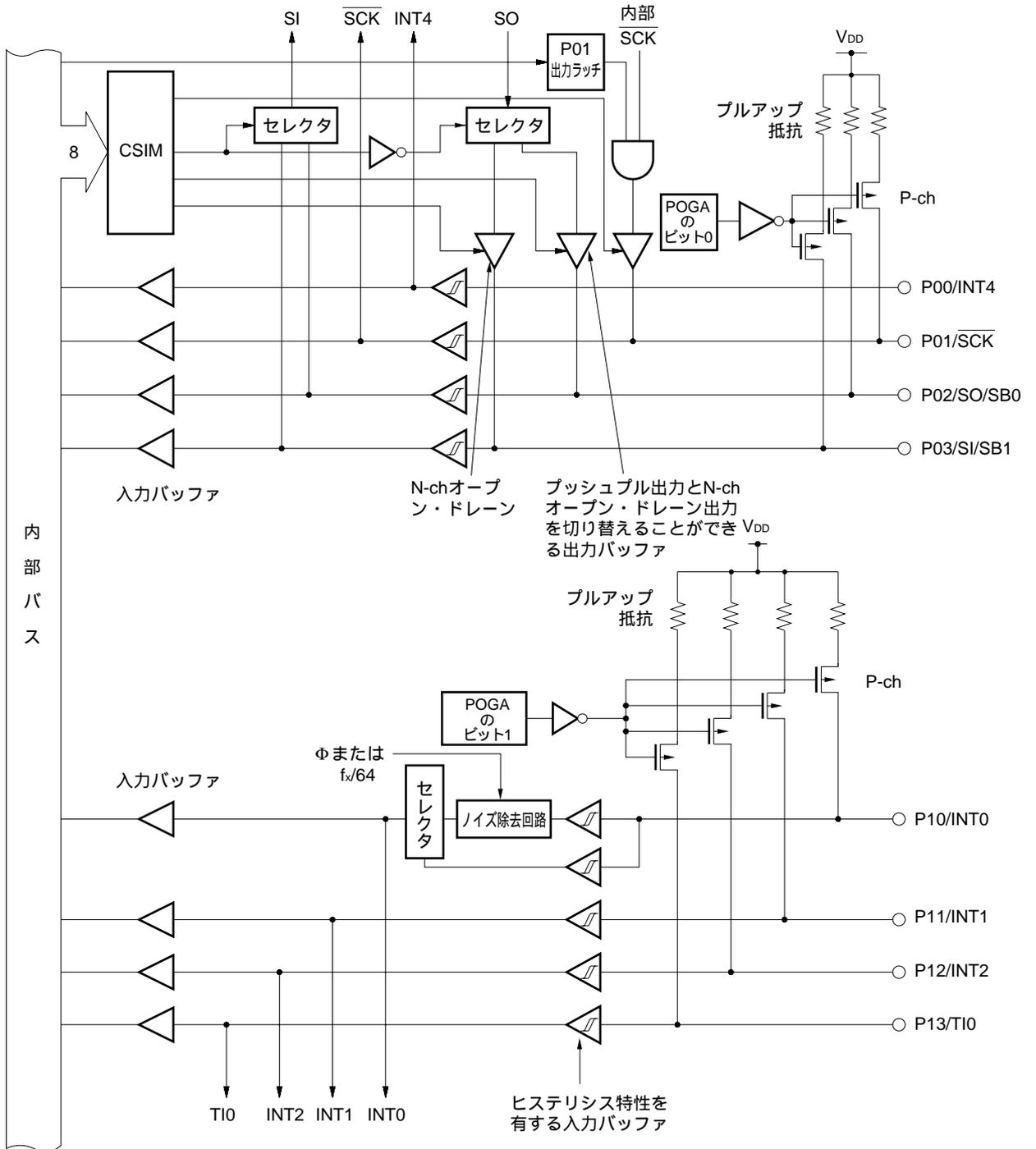
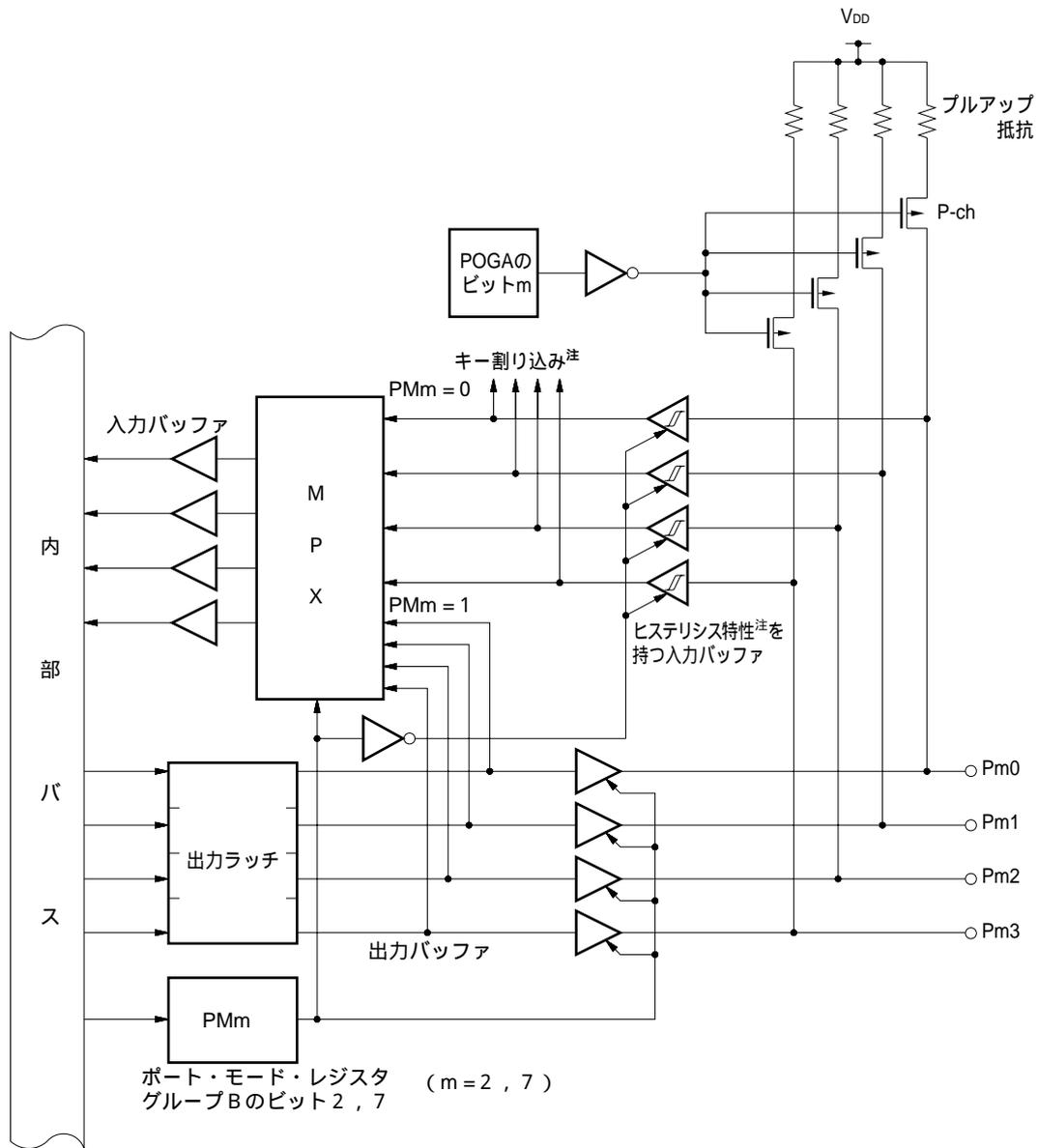
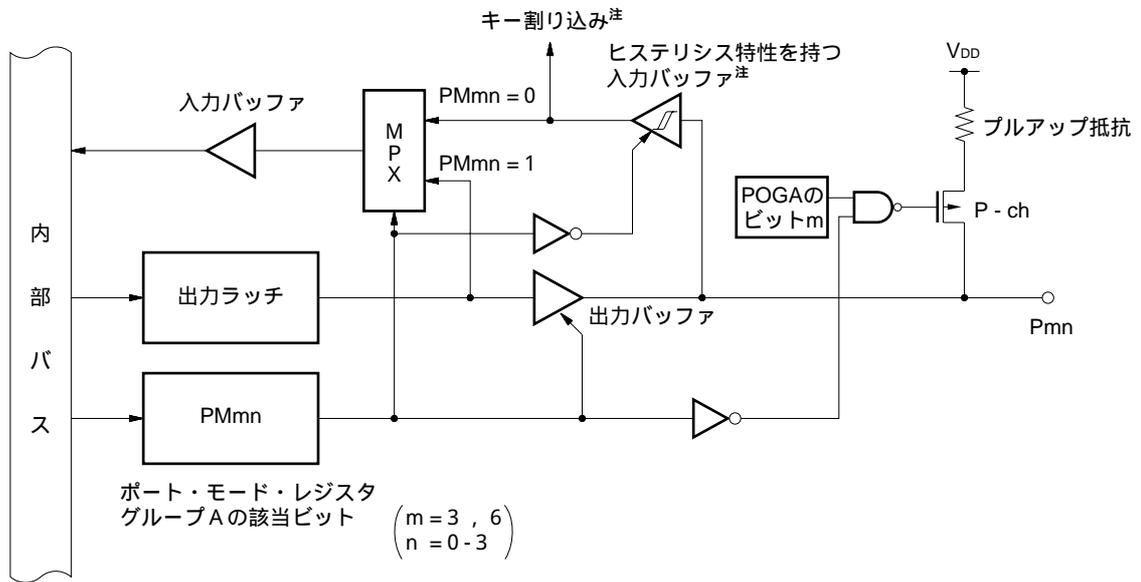


図5 - 3 ポート2, 7の構成



注 ポート7のみ

図5 - 4 ポート3n, ポート6nの構成 (n = 0-3)



注 ポート6nのみ

図5 - 5 ポート4, 5の構成

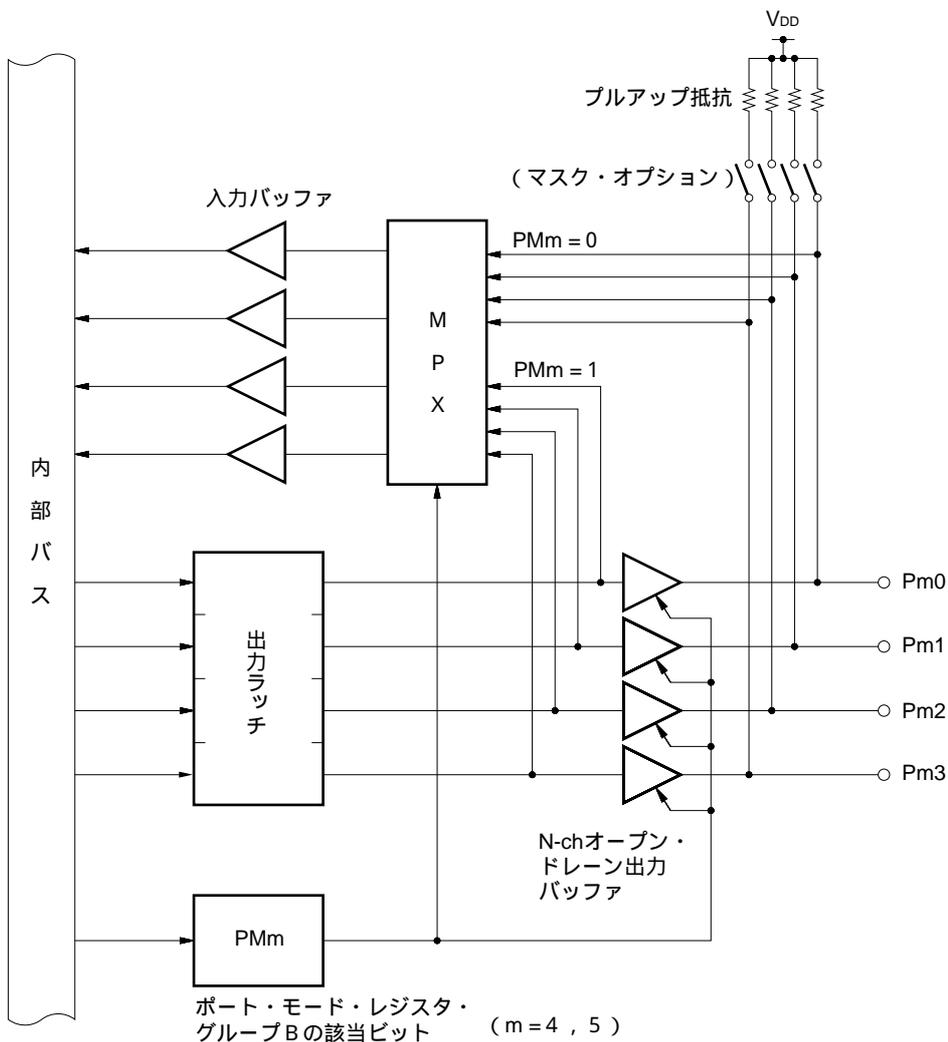
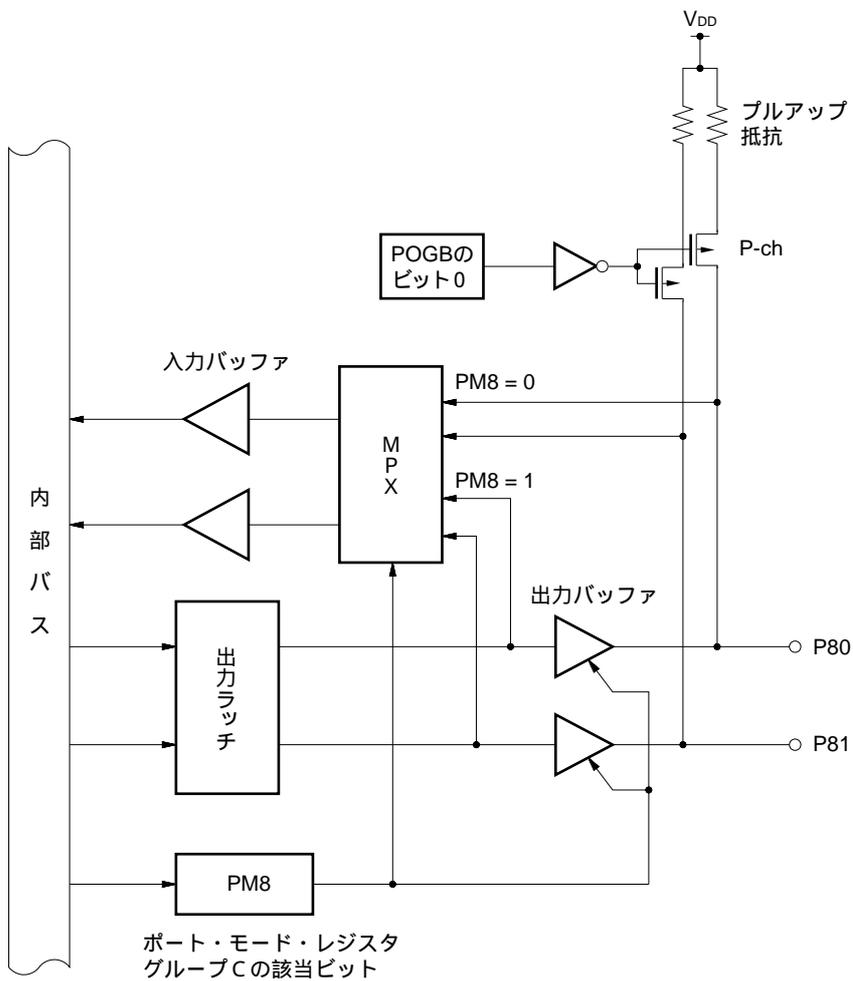


図5 - 6 ポート8の構成



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, 7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。ポート8は、ポート・モード・レジスタ・グループC (PMGC) によって2ビット単位で入出力を指定できます。

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, C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
```

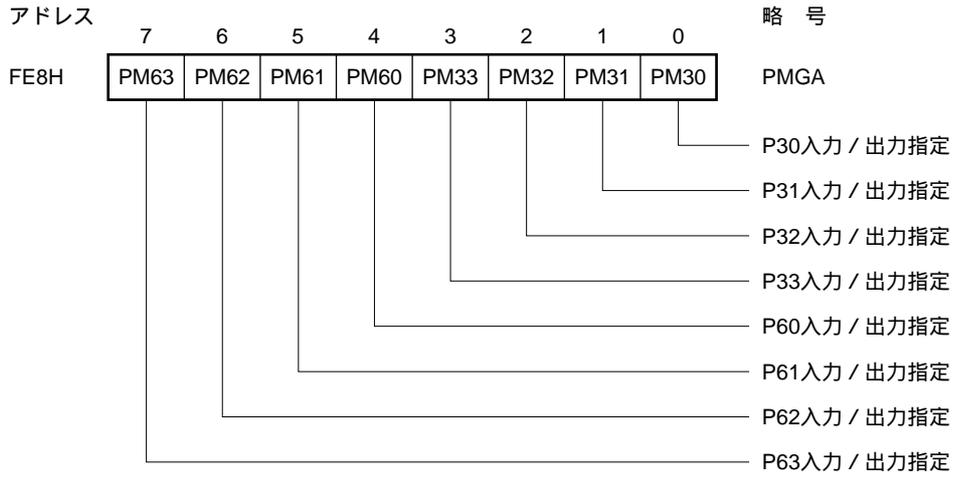
```
MOV   XA, #3CH
```

```
MOV   PMGA, XA
```

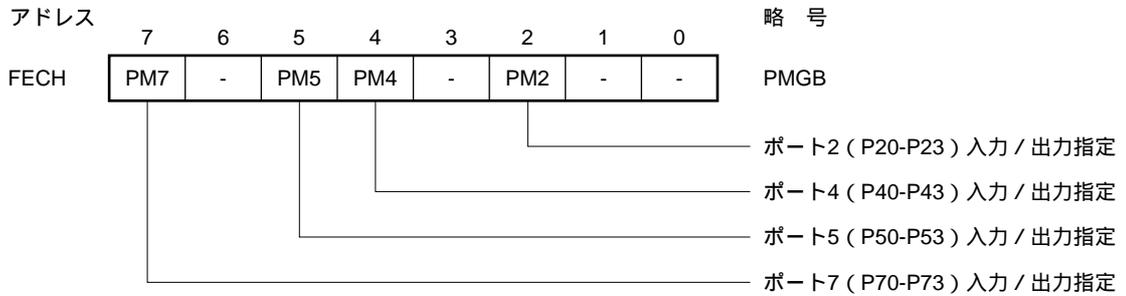
図5 - 7 各ポート・モード・レジスタのフォーマット

	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)

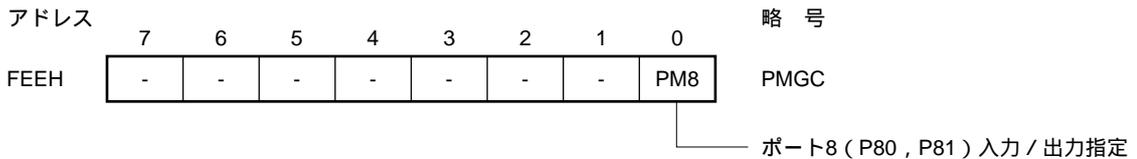
ポート・モード・レジスタ・グループA



ポート・モード・レジスタ・グループB



ポート・モード・レジスタ・グループC



5.1.3 デジタル入出力ポート操作命令

μPD750008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5-3にまとめます。

(1) ビット操作命令

デジタル入出力ポートPORT0-8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。

```

SET1      CY          ; CY   1
AND1      CY, PORT5.0 ; CY   CY P50
OR1       CY, PORT4.1 ; CY   CY P41
SKT       CY
BR        CLRP
SET1      PORT6.1     ; P61  1
:
CLRP : CLR1      PORT6.1 ; P61  0

```

(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。

```
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命令が使用できます。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

命令		PORT		PORT		PORT		PORT		PORT	
		0	1	2	3	4	5	6	7	8	
IN	A, PORTn	注1									
IN	XA, PORTn	注1	-	-							-
OUT	PORTn, A	注1	-								
OUT	PORTn, XA	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									
MOV1	CY, PORTn.bit										
MOV1	CY, PORTn.@L	注2									
MOV1	PORTn.bit, CY		-	-							
MOV1	PORTn.@L, CY	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）。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, OUT, 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/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、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時は、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 ① SKF ①	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, ①	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, ① OR1 CY, ① XOR1 CY, ①	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, @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 @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 ① CLR1 ① MOV1 ①, CY SKTCLR ①	指定したビットの出力ラッチは命令の指示どおり書き換わるが、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したがって変更

① : 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 プルアップ抵抗の内蔵

μPD750008の各ポート端子にはプ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)。プ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。

各ポートのプルアップ抵抗内蔵の指定法を表5 - 4に示します。また、ソフトウェアによる内蔵プ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。

表5 - 4 プルアップ抵抗内蔵の指定方法

ポート(端子名)	プルアップ抵抗内蔵の指定法	POGAのビット	POGBのビット
ポート0 (P01-P03) ^注	3ビット単位でソフトウェアにより接続を指定	ビット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	ビット1	-
ポート2 (P20-P23)		ビット2	-
ポート3 (P30-P33)		ビット3	-
ポート6 (P60-P63)		ビット6	-
ポート7 (P70-P73)		ビット7	-
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8 (P80,P81)	2ビット単位でソフトウェアにより接続を指定	-	ビット0

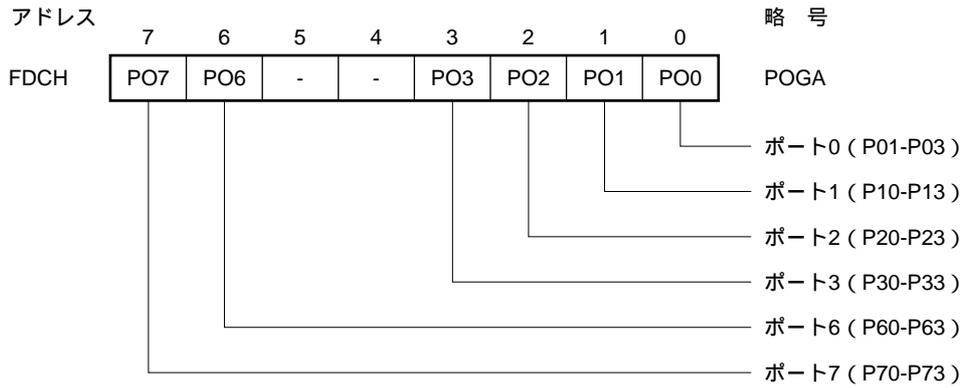
注 P00端子には内蔵プルアップ抵抗の接続を指定できません。

備考 μPD75P0016は、マスク・オプションによるプルアップ抵抗がなく、常にオープンとなっています。

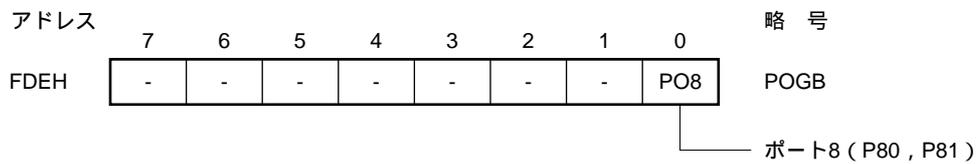
図5 - 8 プルアップ抵抗指定レジスタのフォーマット

	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



プルアップ抵抗指定レジスタ・グループB



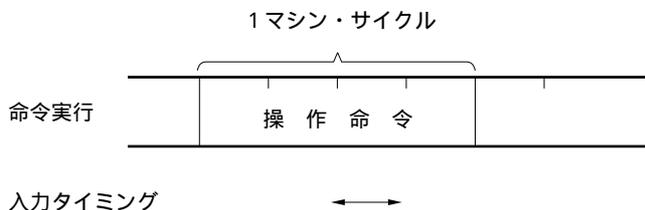
5.1.6 デジタル入出力ポートの入出力タイミング

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。

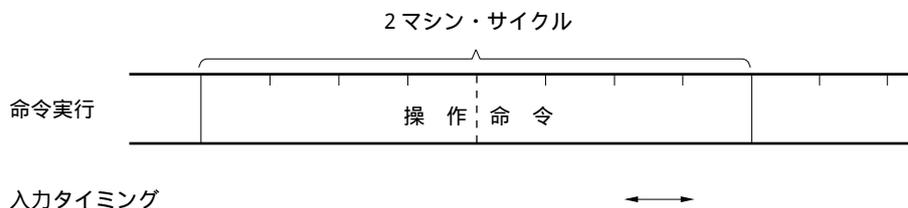
また、ソフトウェアにより内蔵プルアップ抵抗の接続を指定した場合のONタイミングを図5 - 10に示します。

図5 - 9 デジタル入出力ポートの入出力タイミング

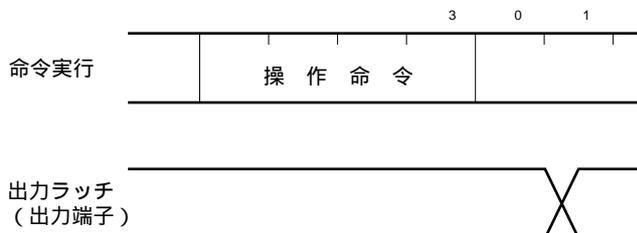
(a) 1マシン・サイクル命令によってデータが取り込まれる場合



(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

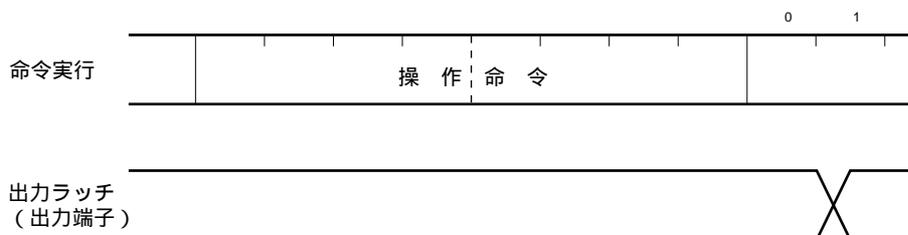
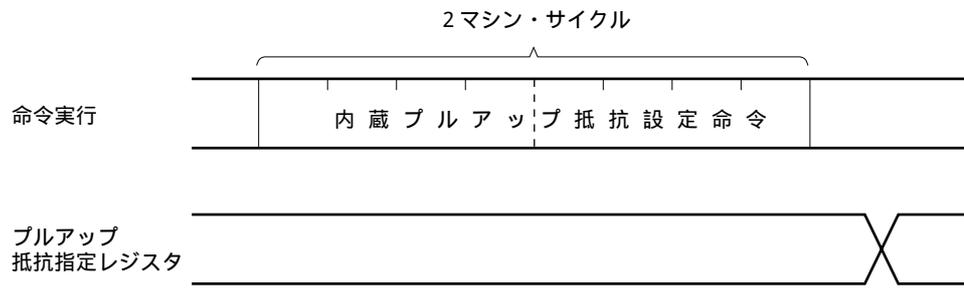


図5 - 10 ソフトウェアによる内蔵プルアップ抵抗のONタイミング



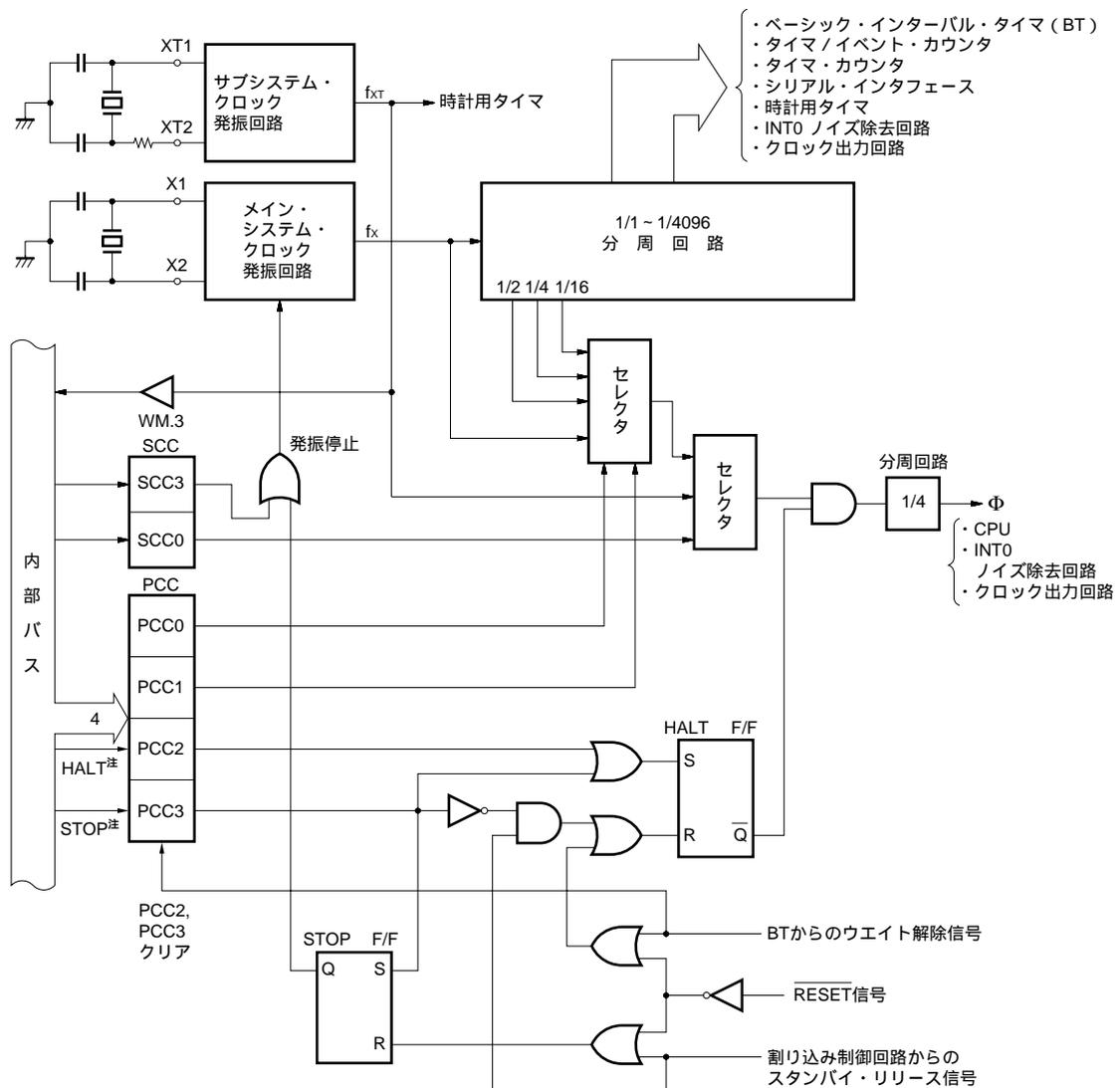
5.2 クロック発生回路

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 .fx : メイン・システム・クロック周波数

2 .fxT : サブシステム・クロック周波数

3 . = 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 . の1クロック・サイクル (tcy) 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・モードなどの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 ($10.7 \mu\text{s} : 6.00 \text{ MHz}$ 動作時)^{※1} 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 ($0.67, 1.33, 2.67, 10.7 \mu\text{s} : 6.00 \text{ MHz}$ 動作時)^{※2}。
- (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 ($122 \mu\text{s} : 32.768 \text{ kHz}$ 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、時計機能とブ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・カウンタおよびタイマ・カウンタは、クロックとして外部クロックを選択した場合、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。

注1 . $f_x = 4.19$ MHz動作時は, $15.3 \mu s$ 。

2 . $f_x = 4.19$ MHz動作時は, $0.95 \mu s$, $1.91 \mu s$, $3.81 \mu s$, $15.3 \mu s$ 。

(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 ($0.67 \mu s$: $f_x = 6.00$ 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。

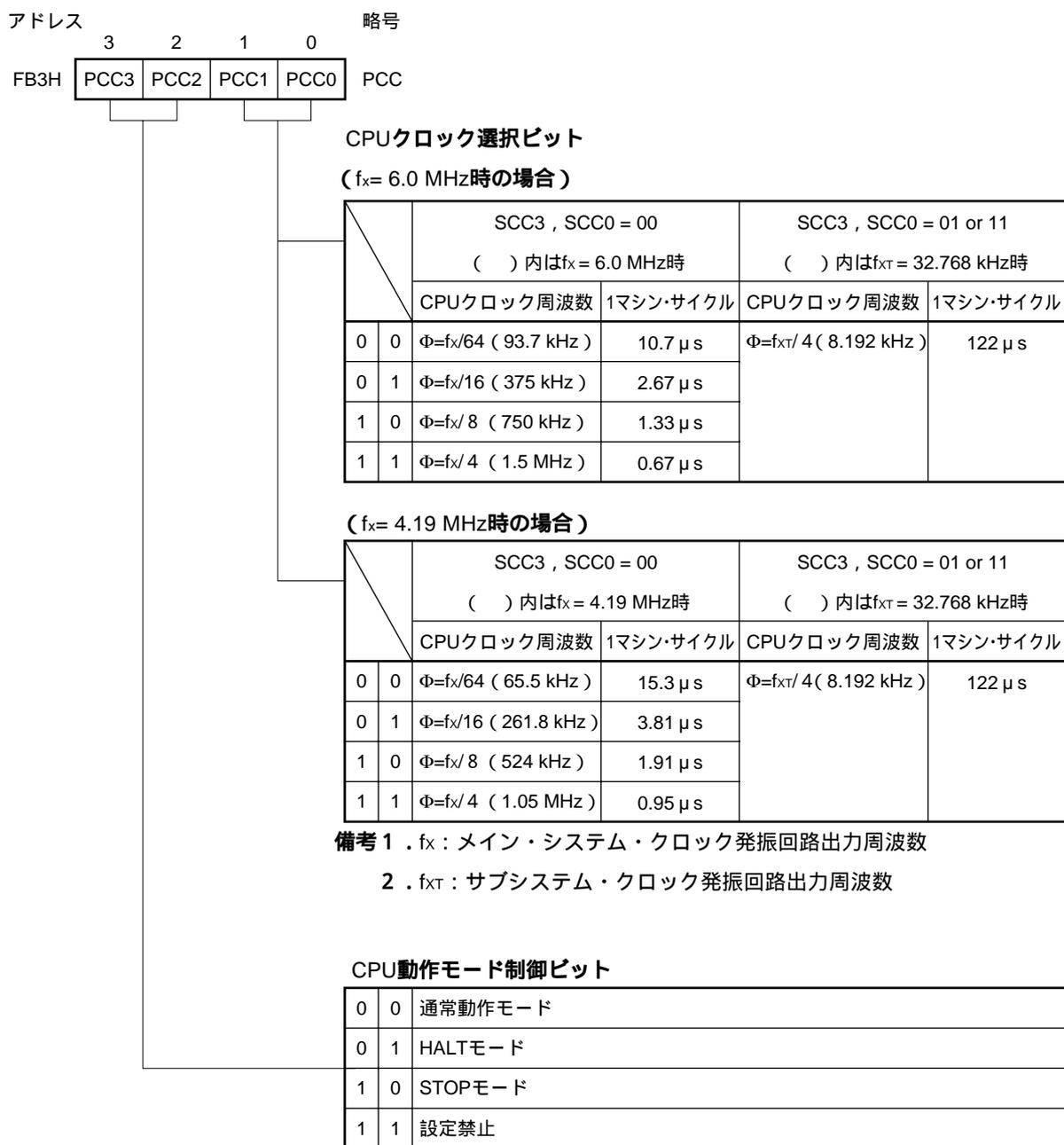
```
SEL  MB15
MOV  A, #0010B
MOV  PCC, A
```

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{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限り許されます。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。

$\overline{\text{RESET}}$ 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 $1/f_{XT}$ 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5-4に示すマシン・サイクル以上経過後, SCCのビット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のビット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. $PCC = 0001B$ ($= f_x/16$ 選択) のときには, SCC.0に“1”をセットしないでください。メイン・システム・クロックからサブシステム・クロックへ切り替える場合は, PCCをほかの設定($PCC = 0001B$)にしてから行ってください。

また, サブシステム・クロックで動作しているときには, $PCC = 0001B$ に設定しないでください。

4. SCC.3に“1”をセットすると, クリスタル発振回路のリークを抑えるために, X1入力が内部で V_{SS} (グランド電位)にショートされます。

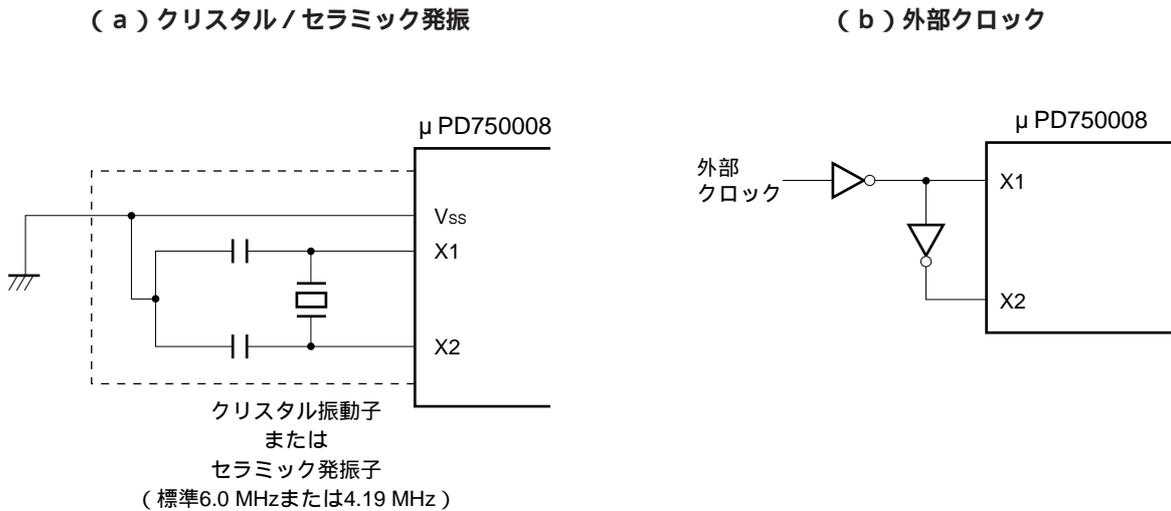
したがって, メイン・システム・クロックに外部クロックを使用する場合は, SCC.3に“1”をセットしないでください。

(3) 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。

また、外部クロックを入力することもできます。その場合、X1端子にクロック信号を入力し、X2端子にはその反転した信号を入力してください。

図5-14 メイン・システム・クロック発振回路の外付け回路

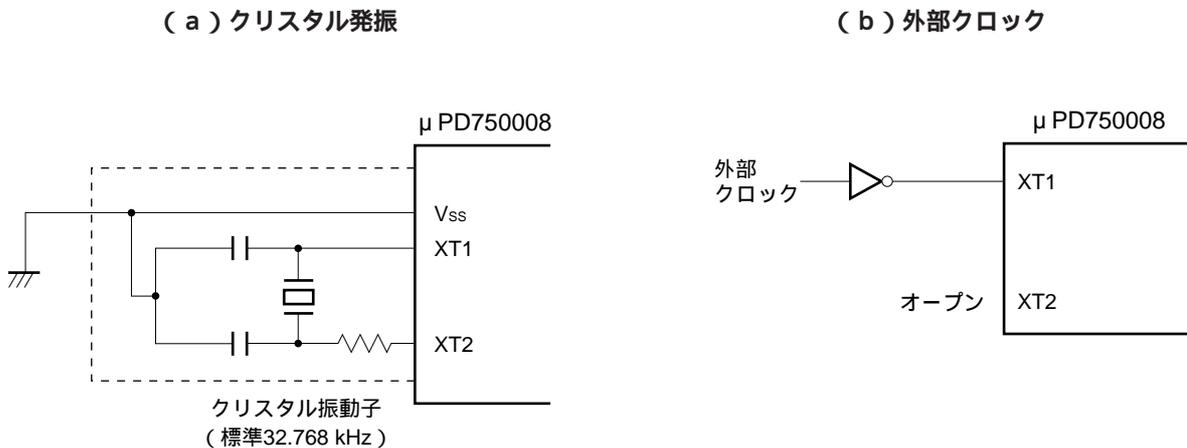


サブシステム・クロック発振回路はXT1, XT2端子に接続されたクリスタル振動子（標準：32.768 kHz）によって発振します。

また、外部クロックを入力することもできます。その場合、XT1端子にクロック信号を入力し、XT2端子はオープンにしてください。

XT1端子の状態は、時計モード・レジスタ（WM）のビット3によってテストすることができます。

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．メイン・システム・クロックまたはサブシステム・クロックとして外部クロックを入力しているとき，STOPモードを設定することはできません。

これはSTOPモードではX1端子がV_{SS}にショートされるためです。

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。また，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。

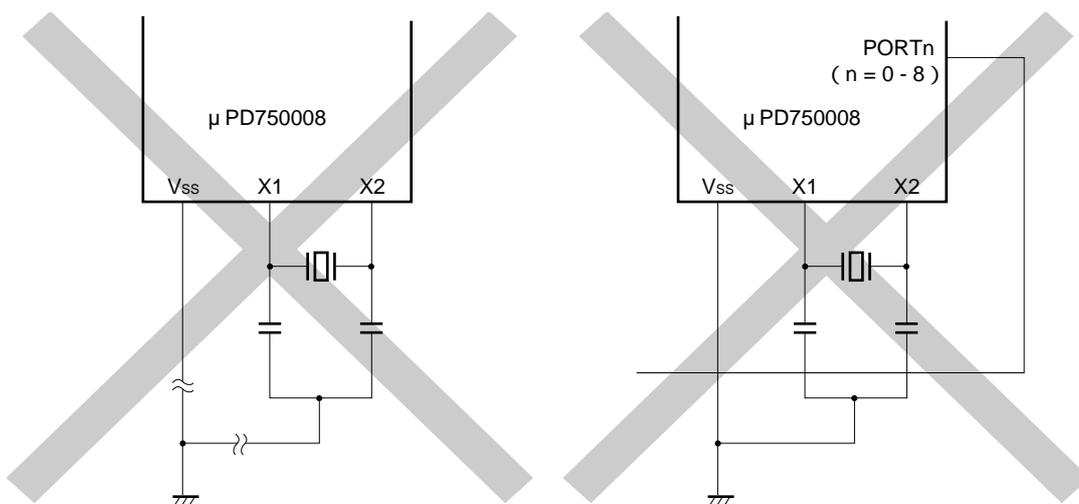
サブシステム・クロック発振回路は低消費電流にするために増幅度の低い回路になっています。このため，メイン・システム・クロック発振回路よりも，ノイズによる誤動作が起こりやすくなっています。したがって，サブシステム・クロックを使用する場合は，配線方法について特にご注意ください。

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

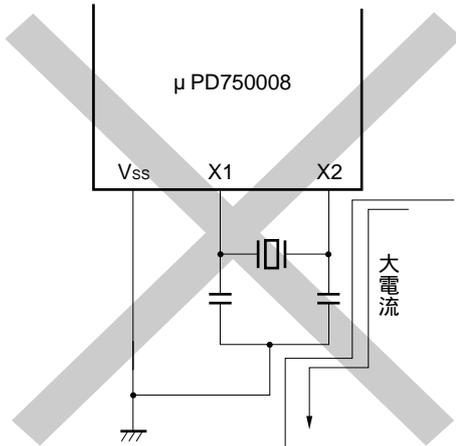
(b) 信号線が交差している



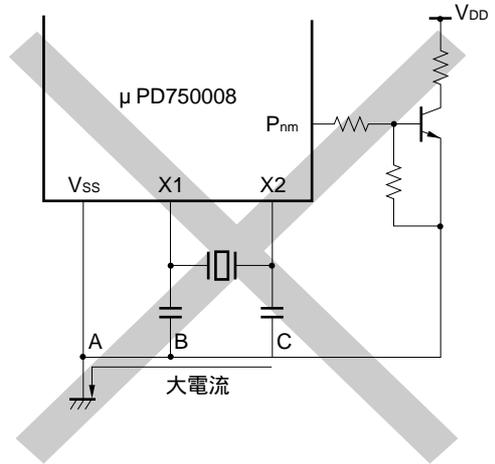
備考 サブシステム・クロックをご使用の場合は，X1，X2をXT1,XT2 と読み換えてください。また，XT2側に直列に抵抗を挿入してください。

図5-16 発振子の接続回路の悪い例(2/2)

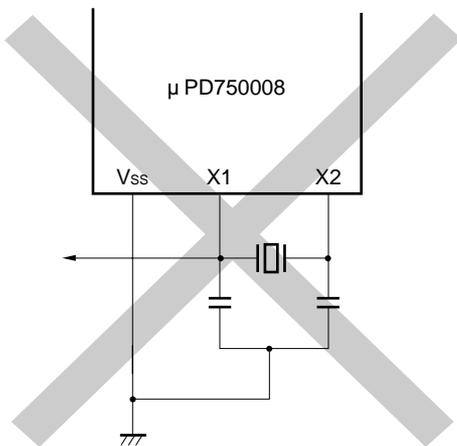
(c) 変化する大電流が信号線に近接している



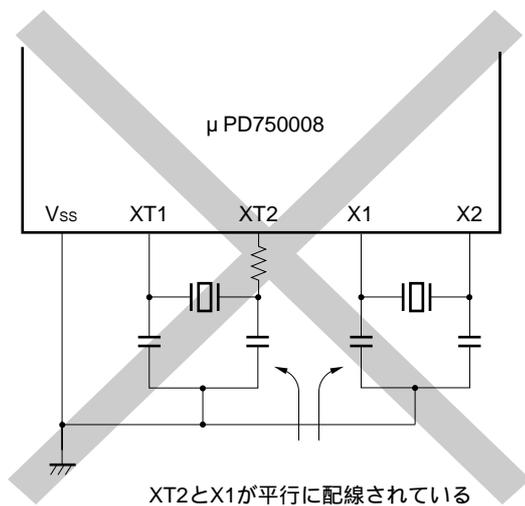
(d) 発振回路部のグランド・ライン上に電流が流れる
(A点, B点, C点の電位が変動する)



(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。

(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 (fx) を分周して、各種クロックを生成します。

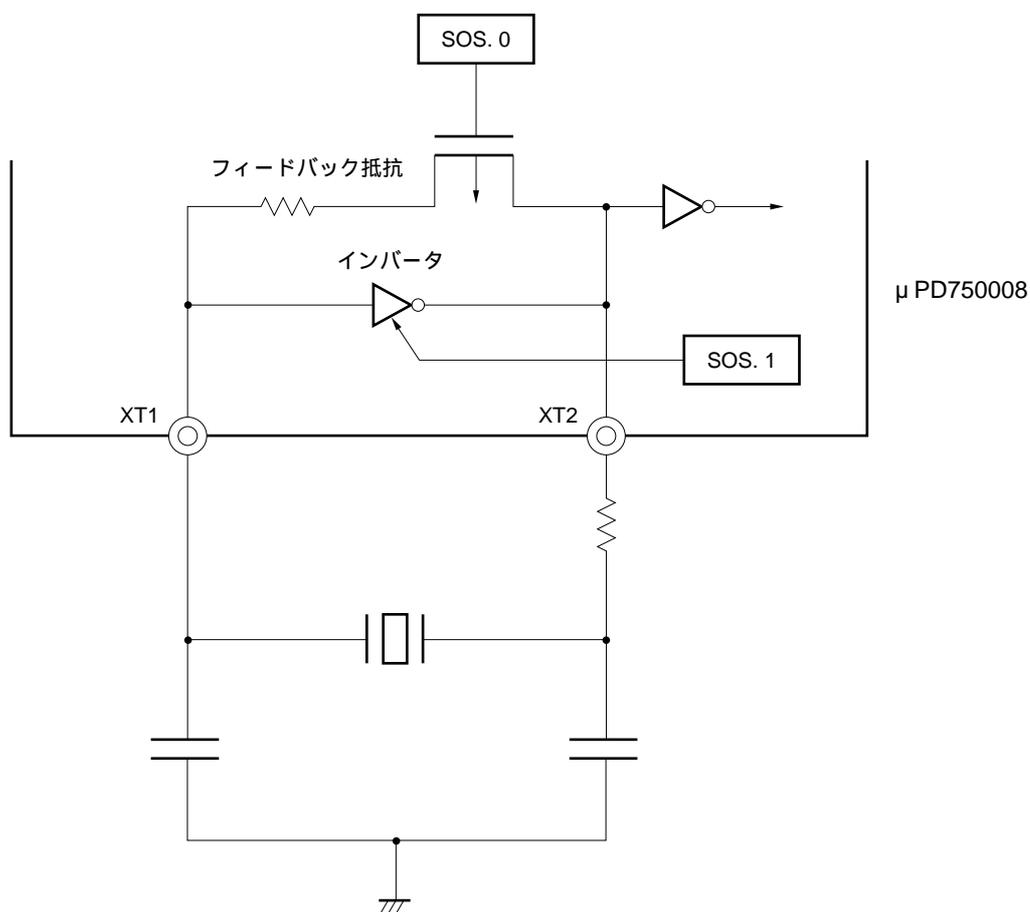
(5) サブシステム・クロック発振回路の制御機能

μPD750008サブシリーズのサブシステム・クロック発振回路には、電源電流を下げるため、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
- ・内蔵インバータのドライブ電流を下げて、電源電圧が高いとき ($V_{DD} = 2.7V$) に消費電流を抑える機能

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を制御するレジスタです (図5 - 18参照)。

$\overline{\text{RESET}}$ 入力により全ビットが0にクリアされます。次にSOSレジスタの各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)

サブシステム・クロックのフィードバック抵抗を使用するには、マスク・オプションの設定とソフトウェアによるSOS.0の切り替えが必要です。SOS.0を“0”にすると、フィードバック回路がオンになります。

発振子を使用しない場合にSOS.0を“1”に設定すると、フィードバック回路をオフにして消費電流を下げることができます。

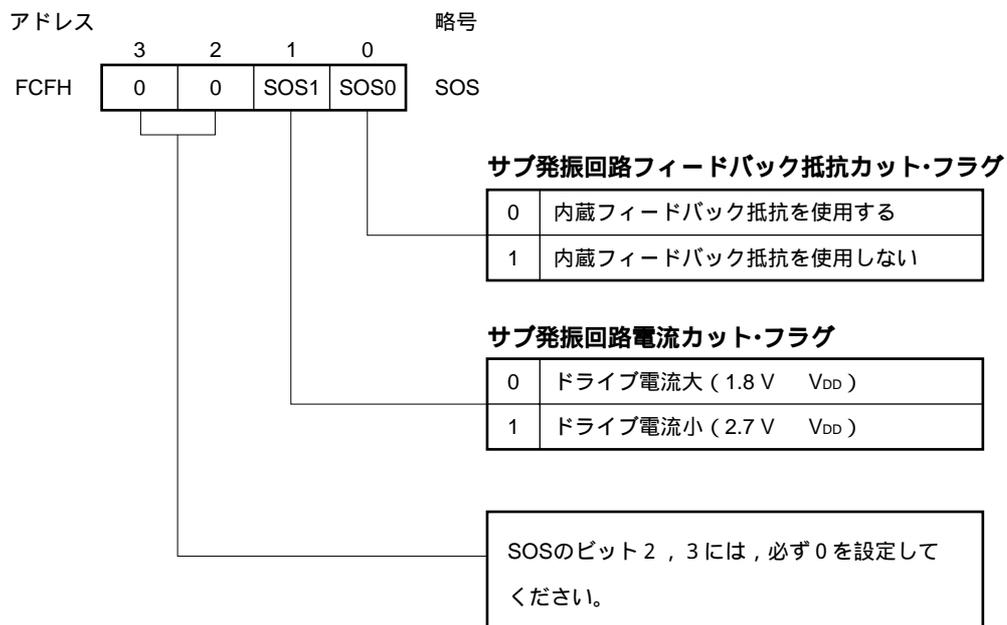
発振子を使用する場合には、必ずマスク・オプションの設定時に「フィードバック抵抗を使用可能」を選択し、SOS.0は“0” (フィードバック回路オン) に設定してください。

(b) SOS.1 (ドライブ能力の切り替えフラグ)

μ PD750008サブシリーズのサブシステム・クロック発振回路の内蔵インバータは、低電源電圧対応 ($V_{DD} = 1.8\text{V}$ 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、電源電流を下げるすることができます。

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, 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 : 0001B

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命令を実行する必要があります。

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			設定禁止		
	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			設定禁止			1マシン・サイクル			1マシン・サイクル			/		

備考1. ()内は、 $f_x = 6.00 \text{ MHz}$, $f_{XT} = 32.768 \text{ kHz}$ の場合。

2. x : don't care

3. CPUクロック は、 $\mu\text{PD750008}$ の内部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。

注意1 . PCC = 0001B (= $f_x/16$ 選択) のときには, SCC.0に “ 1 ” をセットしないでください。メイン・システム・クロックからサブシステム・クロックへ切り替える場合は, PCCをほかの設定 (PCC 0001B) にしてから行ってください。

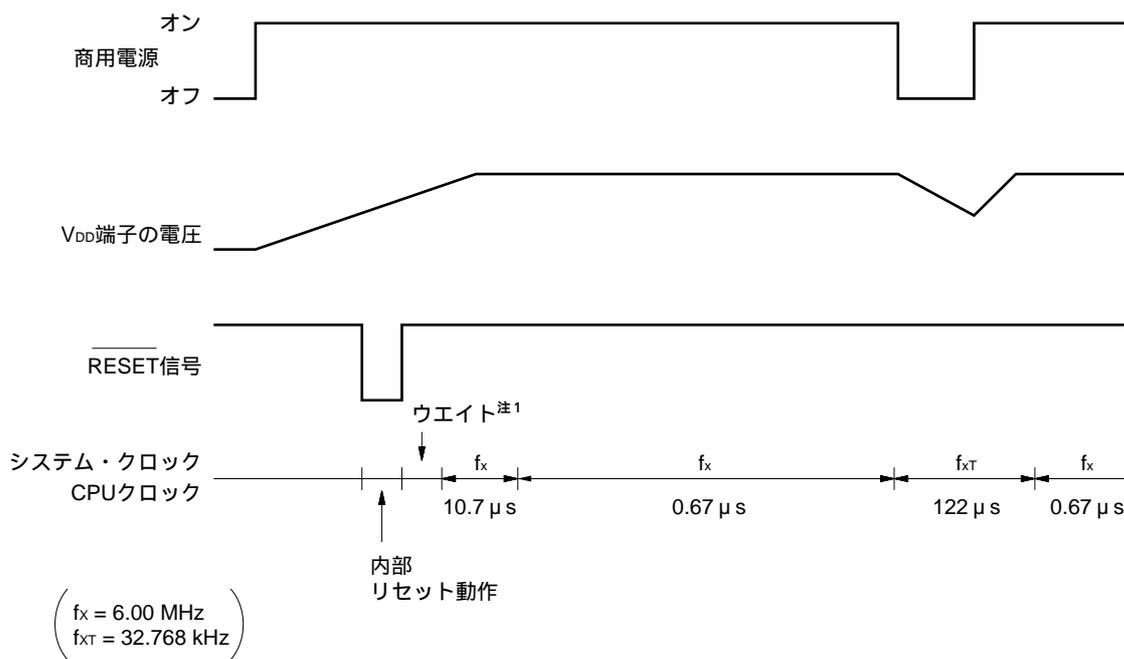
また, サブシステム・クロックで動作しているときには, PCC = 0001Bに設定しないでください。

- 2 . 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 と 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 $f_x/64f_{XT}$, $f_x/8f_{XT}$, $f_x/4f_{XT}$ 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。

(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



- ① $\overline{\text{RESET}}$ 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^{注1}後, CPUはメイン・システム・クロックの最低速 (10.7 μs : 6.00 MHz動作時, 15.3 μs : 4.19 MHz動作時) で動作をスタートします。
- ② V_{DD} 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。

- ③ 商用電流がオフしたことを割り込み入力^{注2}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ることを確認しておく必要があります）。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。
- ④ 商用電流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”にクリアして最高速で動作します。

注1．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。

$2^{15}/f_x$ （21.8 ms：6.00 MHz動作時，31.3 ms：4.19 MHz動作時）

$2^{17}/f_x$ （5.46 ms：6.00 MHz動作時，7.81 ms：4.19 MHz動作時）

ただし、 μ PD75P0016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。

2．INT4を使うと効果的です。

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。

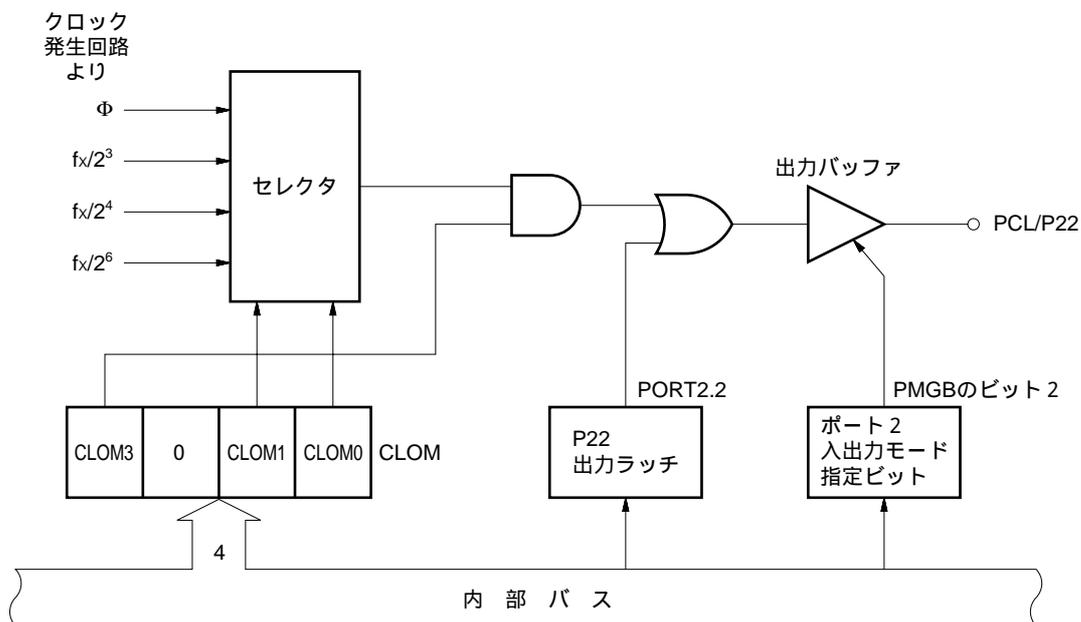
(2) クロック出力回路の機能

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、リモコン出力や周辺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構成



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。読み出しはできません。

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



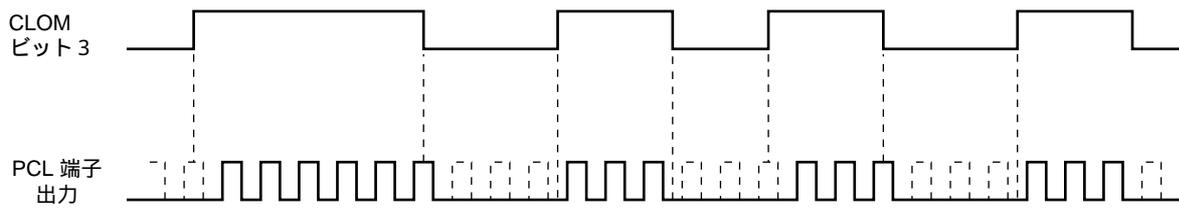
注意 CLOMのビット2には、必ず0を書き込むようにしてください。

(4) リモコン出力への応用例

μPD750008のクロック出力機能は、リモコン出力に応用可能です。リモコン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

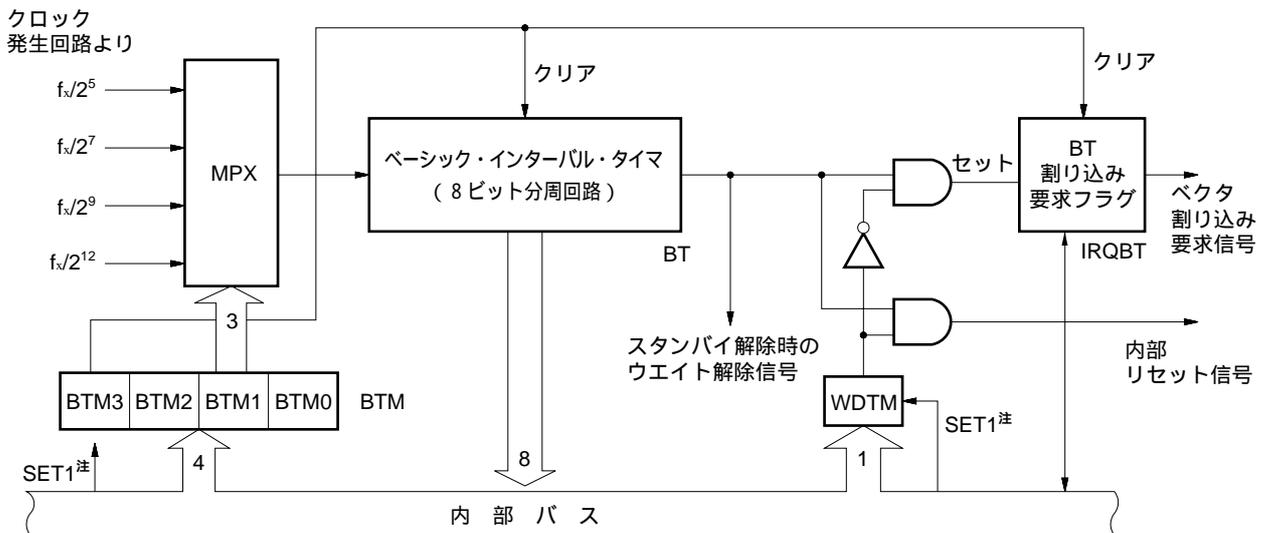
μPD750008は、8ビット・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマを備えており、次のような機能があります。

- (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。

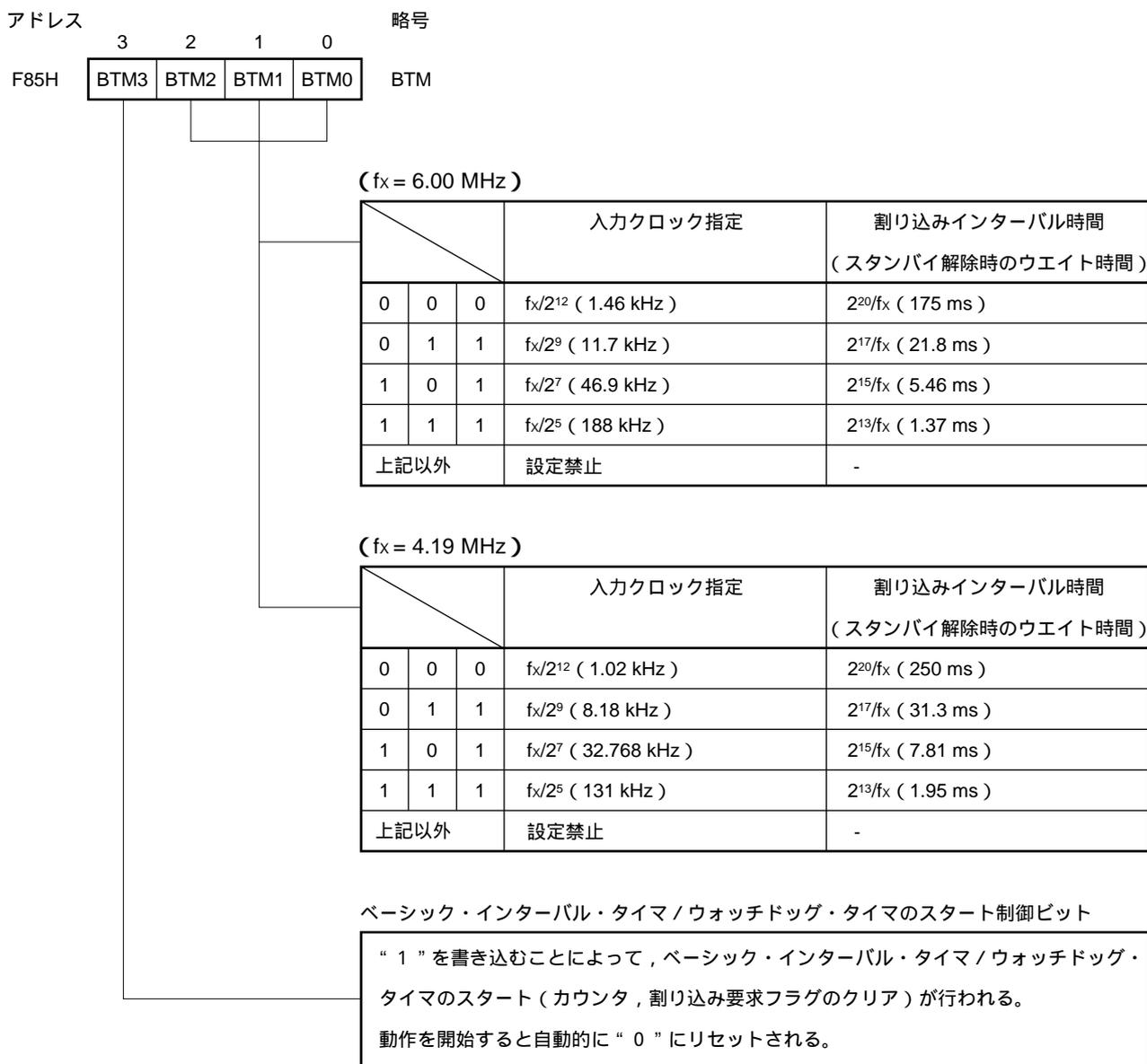
例 割り込み発生インターバルを1.37 ms (6.00 MHz) に設定

```
SEL  MB15          ;またはCLR1 MBE
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

アドレス	F8BH.3	WDTM
	0	BTモード 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IRQBTをセットします。
	1	WTモード 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内部リセット信号を発生します。

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 (①, ②の設定は同時に行ってもかまいません)。

- ① BTMにインターバル時間をセットする。
- ② 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) のオーバ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バフローでは、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のオーバ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はオーバフローし、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 (①, ②の設定は同時に行ってもかまいません)。

- | | |
|--|--------|
| ① BTMにインターバル時間をセットする。 | } 初期設定 |
| ② BTMのビット3に“1”をセットする。 | |
| ③ WDTMに“1”をセットする。 | |
| ④ ① ~ ③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：

```

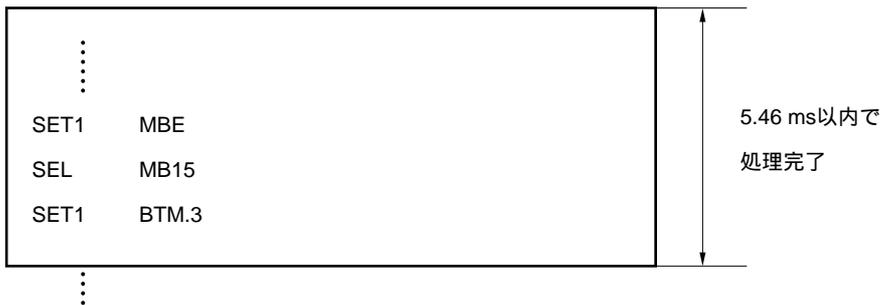
SET1   MBE
SEL    MB15
MOV    A, #1101B
MOV    BTM, A      ; 時間設定とスタート
SET1   WDTM       ; ウォッチドッグ・タイマを許可
      ⋮
    
```

（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：



モジュール2：



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。

- ① 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
- ② 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ていますが、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           ; 時間を設定
STOP                    ; STOPモードを設定
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てしまふを防ぐため、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(MBH=0)

```
LOOP : MOV    XA, BT        ; 1回目の読み出し
        MOV    BC, XA       ; データを格納
        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?
        BA     AA          ; NO
        MOV    XA, BC      ; データをデータ・メモリに格納
        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
        SET1   FLAG       ; データあり。フラグをセット
        RETI
```

5.4 時計用タイマ

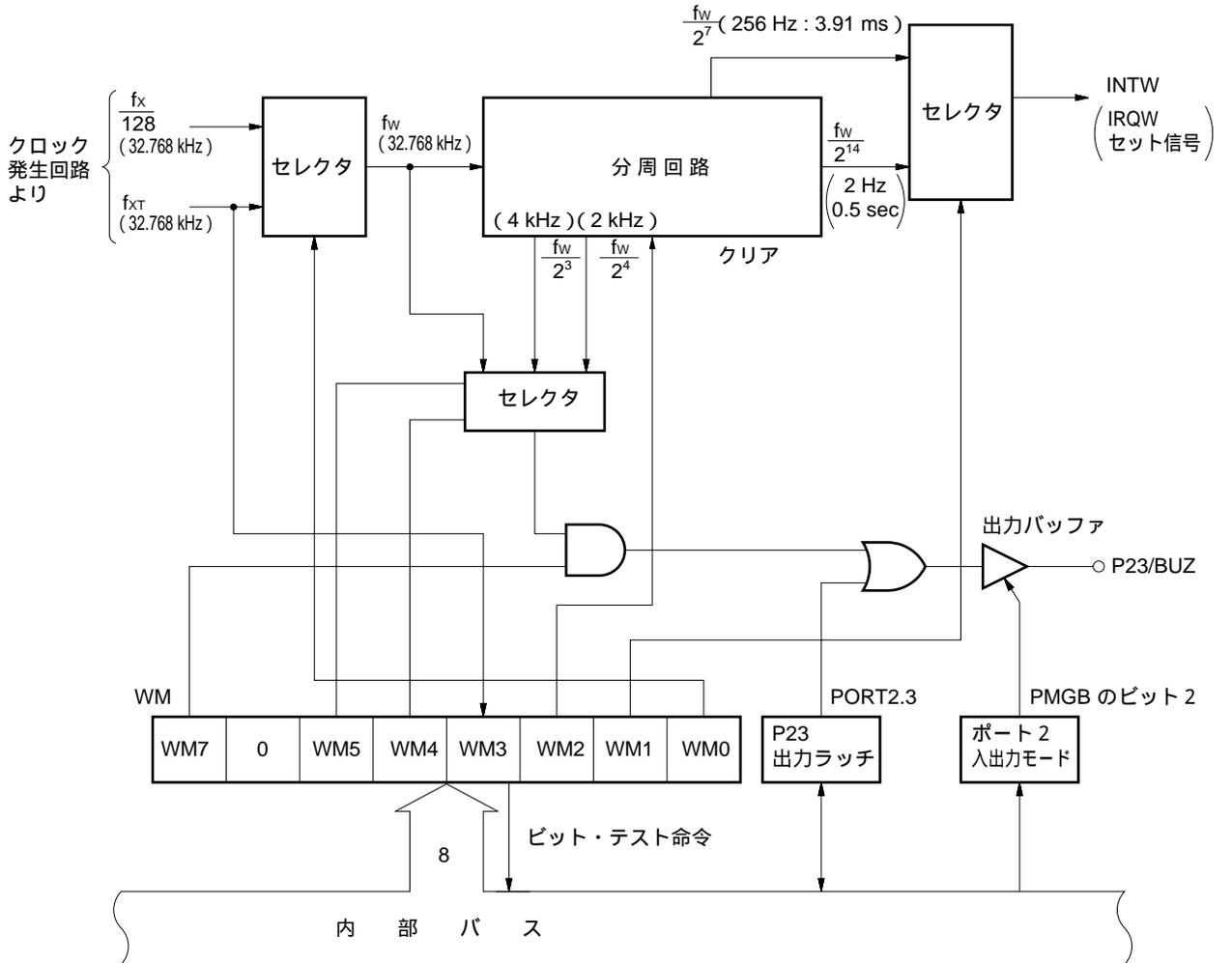
μPD750008は時計用タイマを1チャンネル内蔵しており、次のような機能があります。

- (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・クロックとサブシステム・クロックのいずれでも0.5秒の時間間隔を作ることができます。メイン・システム・クロックは周波数4.194304 MHzを選択してください。
- (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) 分周回路のクリアができますから、時計をゼロ秒スタートできます。

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 $f_x = 4.194304 \text{ MHz}$, $f_{xT} = 32.768 \text{ kHz}$ 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。ブザー出力許可。

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

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

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

μPD750008は、タイマ/イベント・カウンタを1チャンネル(チャンネル0)、タイマ・カウンタを1チャンネル(チャンネル1)内蔵しています。その構成を図5-28、5-29に示します。

この項では、タイマ/イベント・カウンタとタイマ・カウンタをあわせてタイマ/イベント・カウンタと呼びます。チャンネル1の説明としてお読みになる場合は、タイマ/イベント・カウンタをタイマ・カウンタと読み替えてください。

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_n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(チャンネル0のみ)
- (d) TI₀端子入力をN分周してPTO₀端子へ出力(分周回路動作)(チャンネル0のみ)
- (e) シリアル・インタフェース回路へのシリアル・シフト・クロック供給(チャンネル0のみ)
- (f) カウント状態の呼び出し機能

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・カウンタのブロック図(チャンネル0)

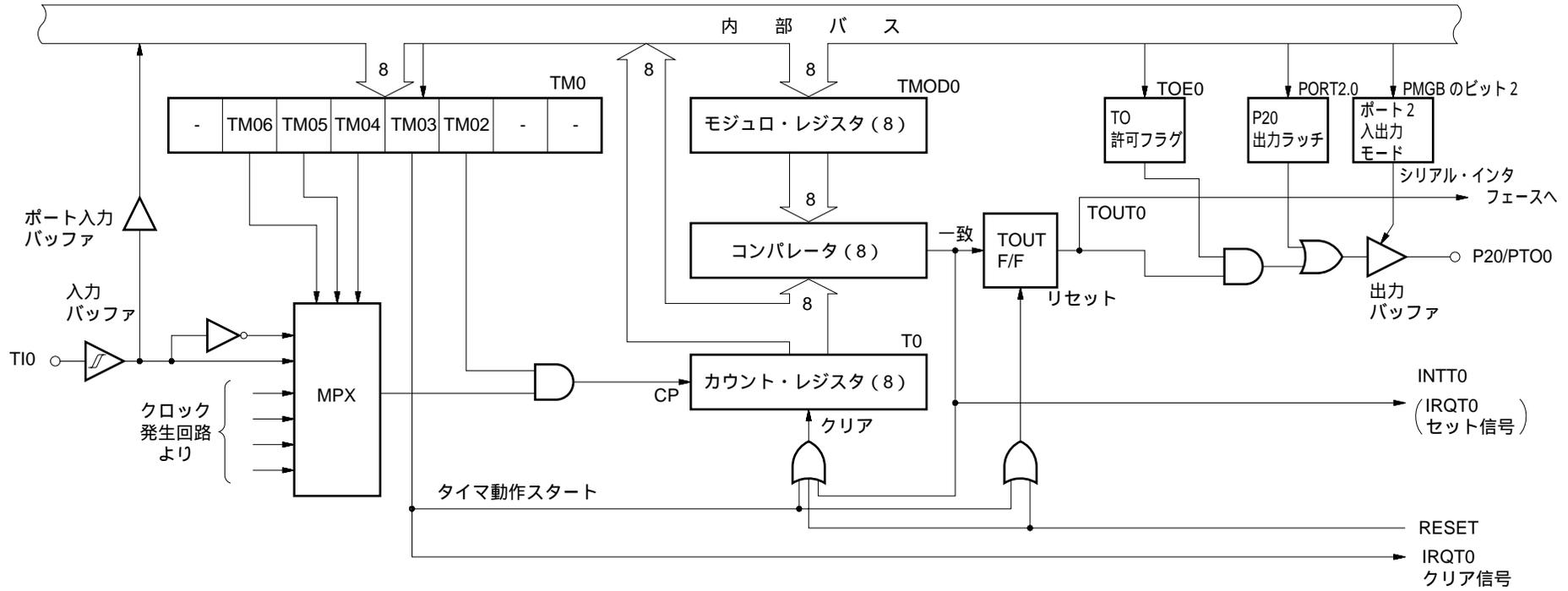
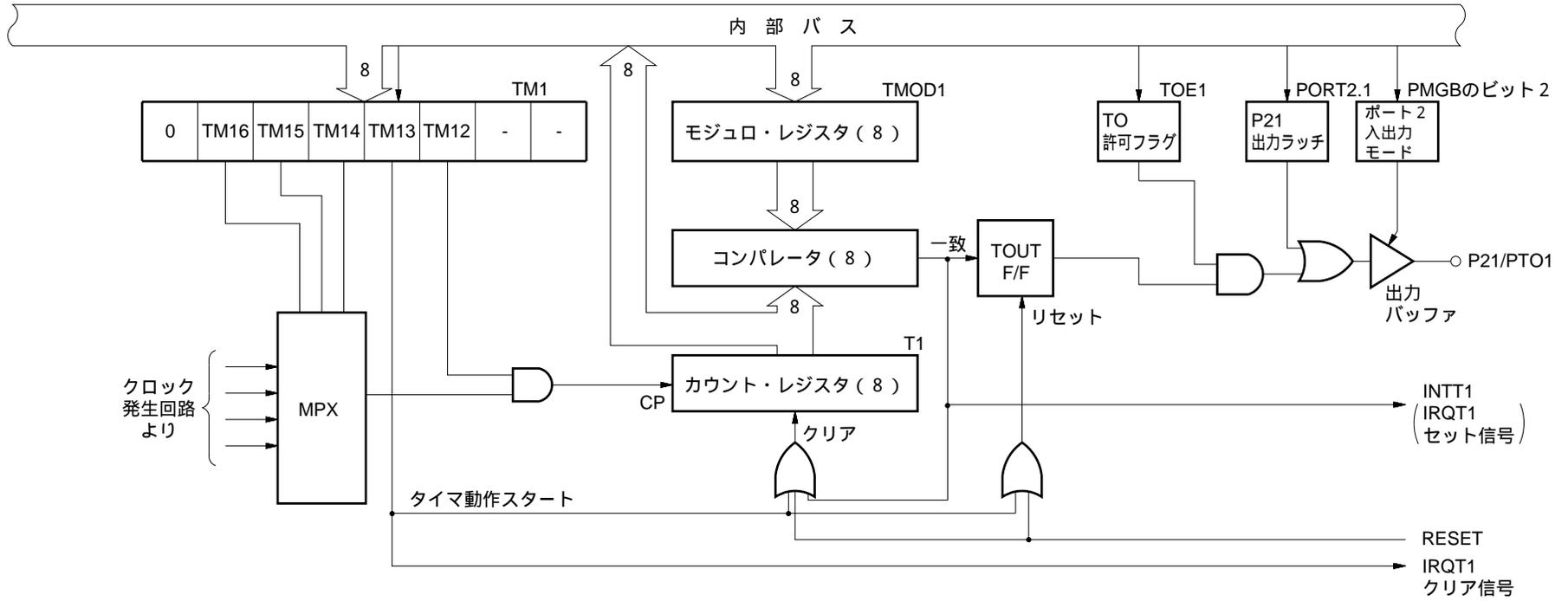


図5 - 29 タイマ・カウンタのブロック図 (チャンネル1)



(1) タイマ/イベント・カウンタ・モード・レジスタ (TM0, TM1)

モード・レジスタ (TMn) 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・モードでタイマをスタート。

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #01001100B
MOV    TMn, XA             ;TMn 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
```

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

アドレス	7	6	5	4	3	2	1	0	略号
FA0H	-	TM06	TM05	TM04	TM03	TM02	-	-	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
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)
上記以外			設定禁止

$f_x = 4.19 \text{ MHz}$ 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
1	0	0	$f_x/2^{10}$ (4.09 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)
上記以外			設定禁止

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5-31 タイマ・カウンタ・モード・レジスタ(チャンネル1)のフォーマット

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

カウント・パルス (CP) 選択ビット

fx = 6.00 MHz時の場合

TM16	TM15	TM14	カウント・パルス (CP)
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)
上記以外			設定禁止

fx = 4.19 MHz時の場合

TM16	TM15	TM14	カウント・パルス (CP)
1	0	0	$f_x/2^{12}$ (1.02 kHz)
1	0	1	$f_x/2^{10}$ (4.09 kHz)
1	1	0	$f_x/2^8$ (16.4 kHz)
1	1	1	$f_x/2^6$ (65.5 kHz)
上記以外			設定禁止

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

注意 TM1のビット7には必ず0を設定してください。

(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（チャンネル0のみ）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。

図5 - 33 タイマ/イベント・カウンタ・モード・レジスタの設定 (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	-	-	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
1	0	0	$f_x/2^{10}$
1	0	1	$f_x/2^8$
1	1	0	$f_x/2^6$
1	1	1	$f_x/2^4$
上記以外			設定禁止

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 - 33 タイマ/イベント・カウンタ・モード・レジスタの設定 (2/2)

(b) タイマ・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	0	TM16	TM15	TM14	TM13	TM12	-	-	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)
1	0	0	$fx/2^{12}$
1	0	1	$fx/2^{10}$
1	1	0	$fx/2^8$
1	1	1	$fx/2^6$
上記以外			設定禁止

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

注意 TM1のビット7には必ず0を設定してください。

(b) タイマ/イベント・カウンタ出力許可フラグ (TOEn)

TOEnは、ビット操作命令により操作します。

TOEnは、内部リセット信号発生により“0”にクリアされます。

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_{CP} (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 (IRQTn) をセットします。

表5 - 6 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 - 6 分解能と最長設定時間

(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			6.00 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
1	0	0	171 μ s	43.7 ms	244 μ s	62.5 ms
1	0	1	42.7 μ s	10.9 ms	61.0 μ s	15.6 ms
1	1	0	10.7 μ s	2.73 ms	15.3 μ s	3.91 ms
1	1	1	2.67 μ s	683 μ s	3.82 μ s	977 μ s

(b) タイマ・カウンタ (チャンネル1) の場合

モード・レジスタ			6.00 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
1	0	0	685 μ s	175 ms	980 μ s	250 ms
1	0	1	171 μ s	43.7 ms	244 μ s	62.5 ms
1	1	0	42.7 μ s	10.9 ms	61.0 μ s	15.6 ms
1	1	1	10.7 μ s	2.73 ms	15.3 μ s	3.91 ms

(3) 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 - 35にタイマ/イベント・カウンタ動作時の構成を示します。

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、カウント・レジスタ (Tn) に入力されます。

Tnとモジュロ・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。

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。

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 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。

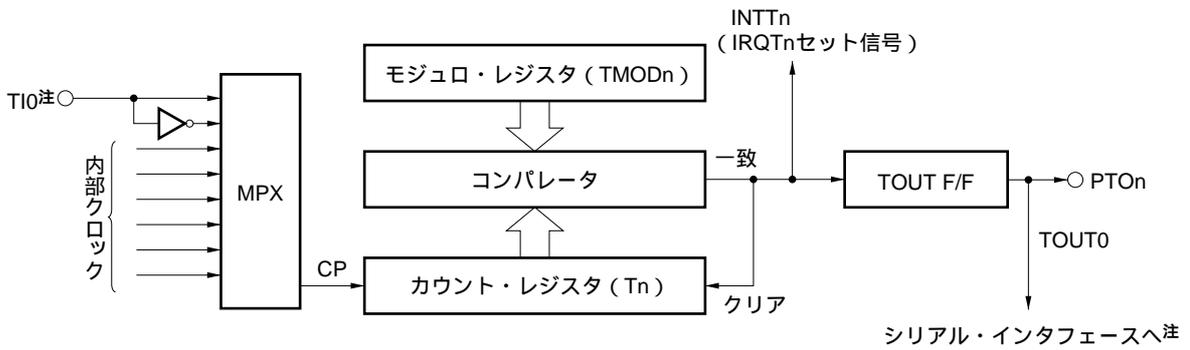
P2nの出力ラッチをクリアする。

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。

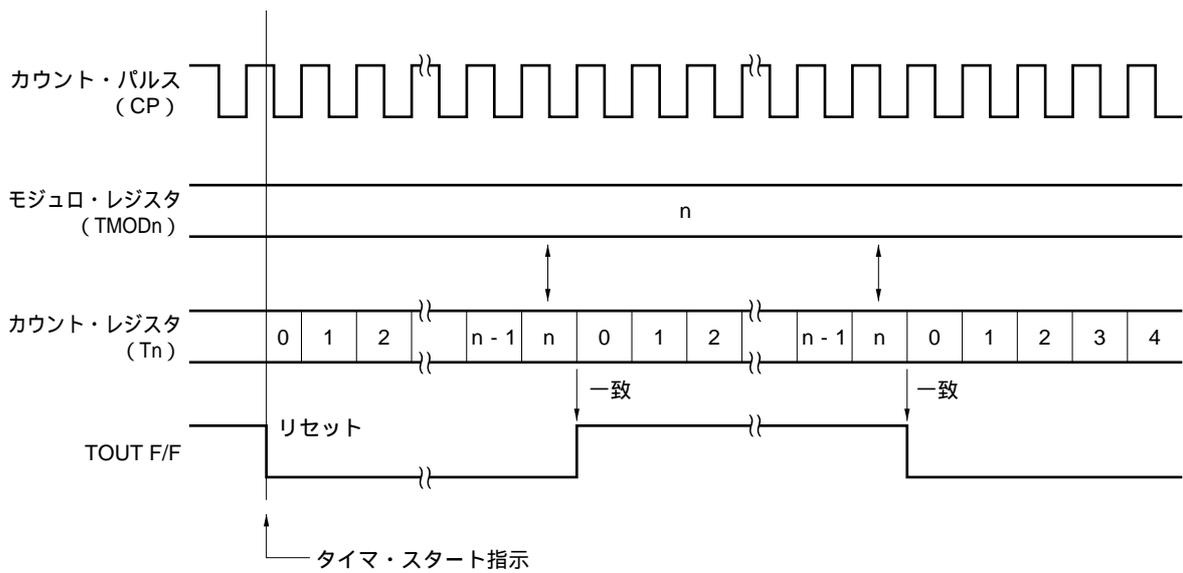
タイマ/イベント・カウンタ出力許可フラグ (TOEn) を1にする。

図5 - 35 タイマ/イベント・カウンタ動作時の構成



注 タイマ/イベント・カウンタのチャンネル0のみです。

図5 - 36 カウント動作時のタイミング



(4) タイマ/イベント・カウンタの応用

(a) 30 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。

モード・レジスタの上位4ビットは0100Bとし、最長設定時間43.7 ms (fx = 6.00 MHz時) を選択します。

モード・レジスタの下位4ビットは1100Bとします。

モジュール・レジスタの設定値は次のとおりです。

$$\frac{30 \text{ ms}}{171 \text{ } \mu\text{s}} = 175.4 \text{ AFH}$$

プログラム例

```
SEL      MB15
MOV      XA, #0AEH
MOV      TMOD0, XA      ; モジュールを設定
MOV      XA, #01001100B
MOV      TM0, XA        ; モードを設定, タイマ・スタート
EI
          ; 割り込みを許可
EI      IET0           ; タイマ割り込みを許可
```

備考 この応用の場合、TIO端子は入力端子として使用できます。

(b) TIO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)。

モード・レジスタの上位4ビットは0000で立ち上がりエッジを選択。

モード・レジスタの下位4ビットは1100B。

モジュール・レジスタは99 = 100 - 1。

プログラム例

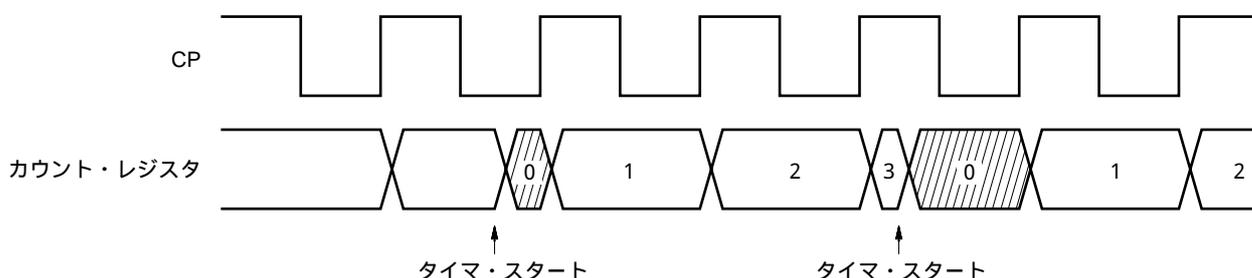
```
SEL      MB15
MOV      XA, #100 - 1
MOV      TMOD0, XA      ; モジュールを設定
MOV      XA, #00001100B
MOV      TM0, XA        ; モードを設定
EI
          ;
EI      IET0           ; INTT0を設定
```

5.5.3 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 (TM0のビット3のセット) 後, 一致信号が発生するまでの時間は5.5.2(2) タイマ/イベント・カウンタの時間設定で計算した値に対し, 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T0のクリアが非同期で行われるためです。

図5-37 カウント・レジスタのクリア・タイミング



(2) タイマ/スタート時の注意

通常タイマ・スタート (TM0のビット3をセット) によってカウント・レジスタT0と割り込みリクエスト・フラグIRQT0はクリアされますが, タイマが動作モードにあり, IRQT0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, IRQT0をクリアできない場合があります。これは, IRQT0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, IRQT0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, IRQT0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, IRQT0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 (TM0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 IRQT0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    TM0, XA    ; タイマ停止
MOV    XA, #4CH
MOV    TM0, XA    ; 再スタート
```

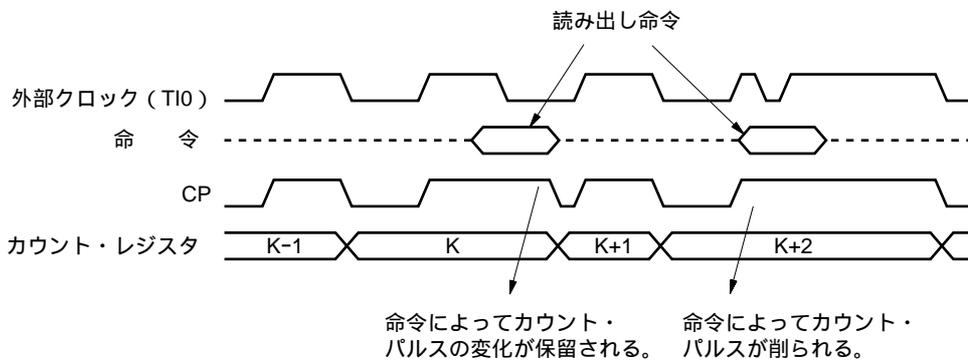
または

```
SEL    MB15
SET1   TM0.3
SET1   TM0.3    ; 再スタート
```

(3) カウント・レジスタ読み出し時の誤差

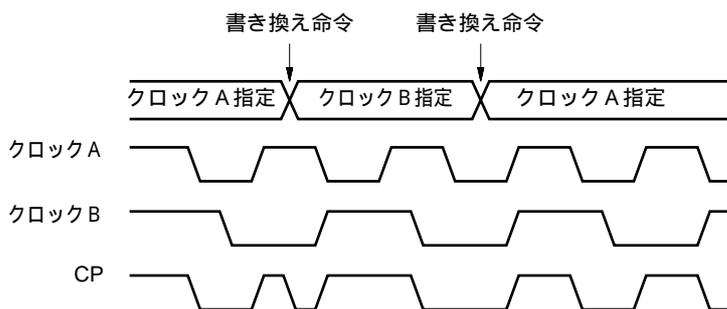
カウント・レジスタの内容は、8ビット・データ・メモリ操作命令によっていつでも読み出し可能ですが、この命令の動作中、カウント・パルスの変化が保留され、カウント・レジスタが変化しないようになっています。このため、カウント・パルスの信号源をTIO入力にした場合、命令実行時間だけカウント・パルスが削られます（内部クロックをカウント・パルスとした場合は、命令と同期するのでこの現象は起こりません）。

したがって、TIO入力をカウント・パルスとして入力し、カウント・レジスタの内容を読み取る場合は、カウント・パルスが削られてもミス・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、TIOに入力するパルスはそれ以上の幅が必要となります。

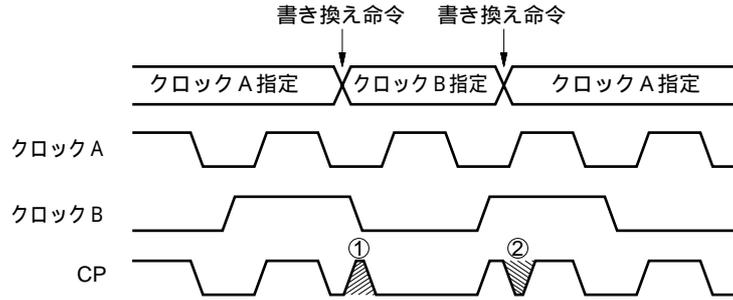


(4) カウント・パルス変更時の注意

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

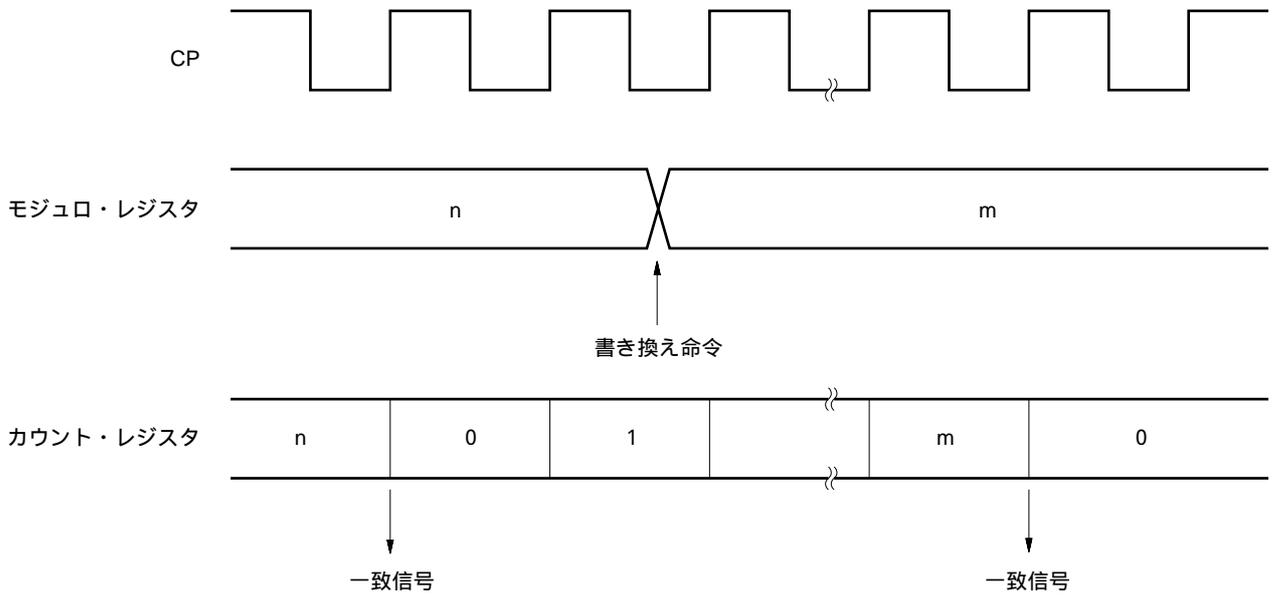


カウント・パルス変更時のクロックの組み合わせにより、次図のようにヒゲ状(①または②)のカウント・パルスが発生することも考えられます。この場合、ミス・カウントあるいはカウント・レジスタの内容が破壊される可能性がありますので、カウント・パルスを変更する場合は必ず、タイマ/イベント・カウンタ・モード・レジスタのビット3を“1”にして、タイマの再スタートも同時に行ってください。

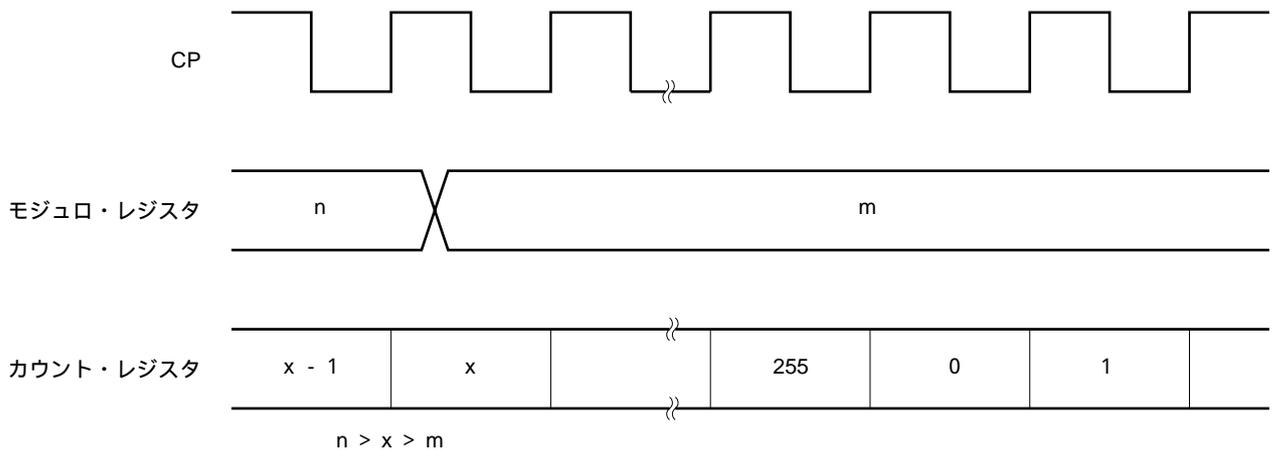


(5) モジユロ・レジスタ変更後の動作

モジユロ・レジスタ変更は、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。



モジユロ・レジスタ変更後の値が、カウント・レジスタの値よりも小さい場合には、カウント・レジスタ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、モジユロ・レジスタの変更後の値(m)が変更前の値(n)より小さい場合には、モジユロ・レジスタを変更後、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

μPD750008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4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{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ビット長のデータの転送を行います。

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{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{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。

(4) SBIモード (シリアル・バス・インタフェース・モード)

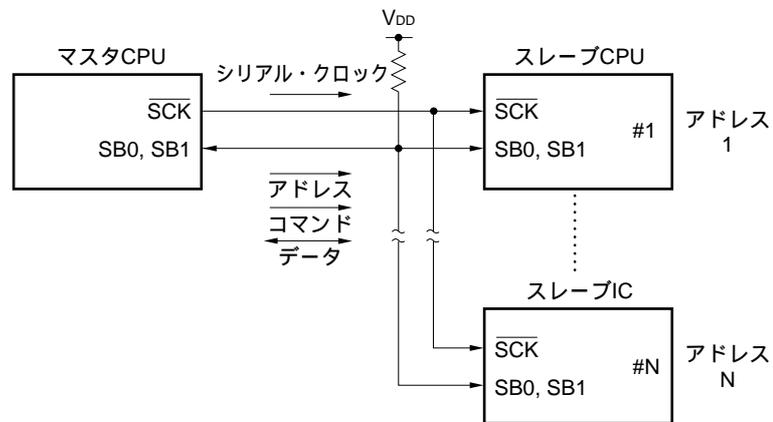
シリアル・クロック (\overline{SCK})、シリアル・データ・バス (SB0、またはSB1) の2本のラインにより複数のデバイスと通信可能なモードです。

NECのシリアル・バス・フォーマットに準拠します。

SBIモードでは、送信側はシリアル・データ・バス上にシリアル通信の対象デバイス選択のための“アドレス”、対象デバイスに対して指令を与える“コマンド”および実際の“データ”を出力することができます。

また、受信側は受信したデータをハードウェアにより、“アドレス”、“コマンド”、“データ”に判別することができます。この機能により、2線式シリアルI/Oモードと同様、入出力ポートの有効活用ができるほか、さらに、応用プログラムのシリアル・インタフェースの制御部分を簡単にすることができます。

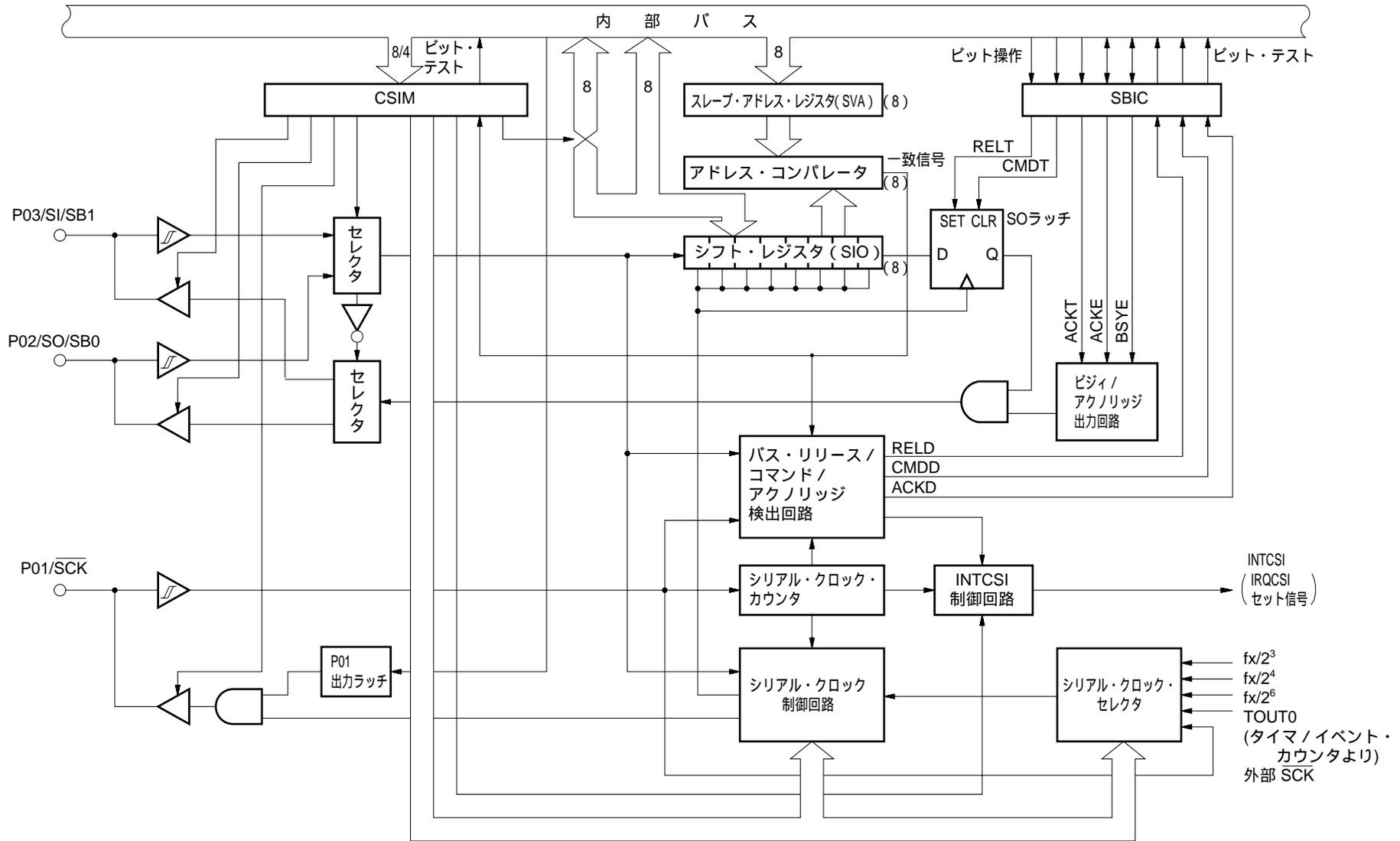
図5 - 38 SBIのシステム構成例



5.6.2 シリアル・インタフェースの構成

図5 - 39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 - 39 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。おもにSBIモードで使用します(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、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。SBIモードでは、 $\overline{\text{SCK}}$ の8発目終了時にセットされます(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) スレーブ・アドレス・レジスタ (SVA), アドレス・コンパレータ**・ SBIモード時**

μ PD750008をスレーブ・デバイスとするときに使用します。スレーブは自分の指定番号(スレーブ・アドレス値)をSVAにセットします。マスタは特定のスレーブを選択するためのスレーブ・アドレスを出力します。

アドレス・コンパレータにより、マスタから受信したスレーブ・アドレスとSVAの値を比較して、一致すると、そのスレーブが選択されたことになります。

・ 2線式シリアルI/Oモード, SBIモード時

μ PD750008をマスタまたはスレーブとして送信する場合に、エラー検出を行います(詳細は、5.6.3(4)スレーブ・アドレス・レジスタを参照)。

(8) INTCSI制御回路

割り込みの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。

- ・ SBIモード時

WUP^注 = “0” の時...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。

WUP = “1” の時 ... アドレス受信後、SVAとSIOの値が一致したときに割り込み要求を発生します。

注 WUP...ウエイク・アップ機能指定ビット (CSIMのビット5)

(9) 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。

(10) ビジィ/アクノリッジ出力回路, バス・リリース/コマンド/アクノリッジ検出回路

SBIモード時に各種制御信号の出力および検出を行います。

3線式および2線式シリアルI/Oモード時には動作しません。

(11) 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 - 40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。

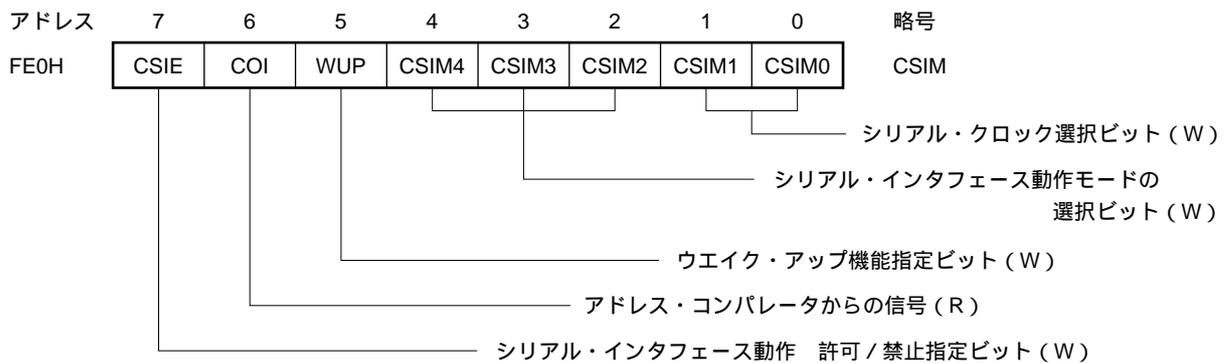
CSIMは、シリアル・インタフェースの動作モード、シリアル・クロック、ウエイク・アップ機能などを指定する8ビット・レジスタです。

操作は、8ビット・メモリ操作命令で行います。上位3ビットについては、1ビット単位の操作もできます。ビット操作の場合は各ビットの名称を用いて操作します。

ビットにより読み出し/書き込み動作の可否は異なります(図5 - 40参照)。ビット6は、ビット・テストのみ可能で、書き込んだデータは無効となります。

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

図5 - 40 シリアル動作モード・レジスタ (CSIM) のフォーマット (1/4)



備考1 . (R) 読み出しのみ可能

2 . (W) 書き込みのみ可能

図5 - 40 シリアル動作モード・レジスタ (CSIM) のフォーマット (2/4)

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

アドレス・コンパレータからの信号 (R)

COI ^注	クリアされる条件 (COI = 0)	セットされる条件 (COI = 1)
	スレーブ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致しなかったとき	スレーブ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致したとき

注 COIの読み出しはシリアル転送開始前、完了後にのみ有効です。転送中には不定の値が読み出されます。

また、8ビット操作命令によって書き込まれたCOIデータは無視されます。

ウエイク・アップ機能指定ビット (W)

WUP	0	各モードでのシリアル転送終了時に毎回IRQCSIをセットする。
	1	SBIモード時のみに使用。バス・リリース後に受信したアドレスがスレーブ・アドレス・レジスタのデータと一致したとき (ウエイク・アップ状態) のみIRQCSIをセットする。SB0, SB1はハイ・インピーダンス。

注意 $\overline{\text{BUSY}}$ 信号出力中にWUP = 1にすると、 $\overline{\text{BUSY}}$ が解除されません。SBIでは $\overline{\text{BUSY}}$ の解除指示後、次のシリアル・クロック ($\overline{\text{SCK}}$) の立ち下がりまで $\overline{\text{BUSY}}$ 信号が出力されます。WUP = 1とするときは、必ず $\overline{\text{BUSY}}$ を解除したのちにSB0 (またはSB1) 端子がハイ・レベルになったことを確認してから、WUP = 1としてください。

図5 - 40 シリアル動作モード・レジスタ (CSIM) のフォーマット (3/4)

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端子機能	SI端子機能
x	0	0	3線式シリアル I/Oモード	SIO ₇₋₀ XA (MSB先頭で転送)	SO/P02 (CMOS出力)	SI/P03 (入力)
		1		SIO ₀₋₇ XA (LSB先頭で転送)		
0	1	0	SBIモード	SIO ₇₋₀ XA (MSB先頭で転送)	SB0/P02 (N-chオープン・ドレイン) 入出力	P03入力
1					P02入力	SB1/P03 (N-chオープン・ドレイン) 入出力
0	1	1	2線式シリアル I/Oモード	SIO ₇₋₀ XA (MSB先頭で転送)	SB0/P02 (N-chオープン・ドレイン) 入出力	P03入力
1					P02入力	SB1/P03 (N-chオープン・ドレイン) 入出力

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
		3線式シリアルI/Oモード	SBIモード	2線式シリアルI/O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・カウンタ出力 (TOUT0)			出力
1	0	f _x /2 ⁴ (375 kHz : 6.00 MHz時, 262 kHz : 4.19 MHz時)		f _x /2 ⁶ (93.8 kHz : 6.00 MHz時, 65.47 kHz : 4.19 MHz時)	
1	1	f _x /2 ³ (750 kHz : 6.00 MHz時, 524 kHz : 4.19 MHz時)			

図5 - 40 シリアル動作モード・レジスタ (CSIM) のフォーマット (4/4)

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	CSIM2	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I/Oモード
1	1	0	SBIモード
1	1	1	2線式シリアルI/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
1	0	0	ハイ・インピーダンス
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 (ハイ・レベル出力)
1	1	0	
1	1	1	

備考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。

- ① 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。
- ② CSIEをクリアする。
- ③ 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 $f_x/2^4$ 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、SB0端子をシリアル・データ・バスとしてSBIモードで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, #10001010B
MOV      CSIM, XA            ;CSIM  10001010B
```

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 - 41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ている8ビット・レジスタで、おもにSBIモードで使用されます。

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できません。

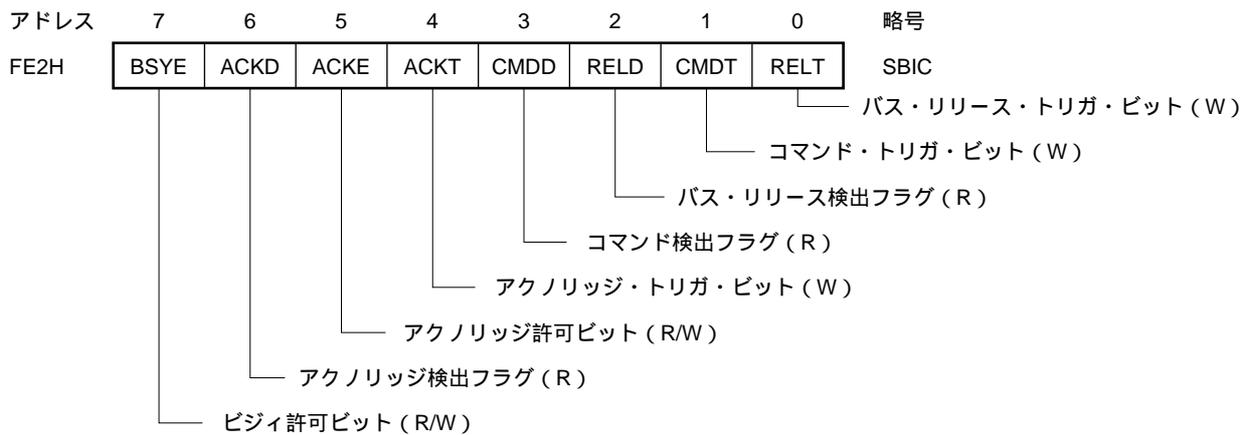
ビットにより読み出し/書き込み動作の可否は異なります(図5 - 41参照)。

RESET 信号発生により全ビット0にクリアされます。

注意 3線式および2線式シリアルI/Oモードでは次のビット以外は使用できません。

- ・バス・リリース・トリガ・ビット (RELT) ...SOラッチのセット
- ・コマンド・トリガ・ビット (CMDT)SOラッチのクリア

図5 - 41 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 (1/3)



- 備考1** . (R) 読み出しのみ可能
- 2 . (W) 書き込みのみ可能
- 3 . (R/W) 読み出し / 書き込みともに可能

図5 - 41 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 (2/3)

ビジー許可ビット (R/W)

BSYE	0	① ビジー信号の自動出力禁止 ② クリア命令実行直後の、 \overline{SCK} の立ち下がりに同期して、ビジー信号の出力を停止
	1	アクノリッジ信号に続き \overline{SCK} の立ち下がりに同期して、ビジー信号を出力

アクノリッジ検出フラグ (R)

ACKD	クリアされる条件 (ACKD = 0)	セットされる条件 (ACKD = 1)
①	転送スタート時	アクノリッジ信号 (\overline{ACK}) 検出時 (\overline{SCK} の立ち上がりに同期します)
②	\overline{RESET} 信号発生時	

アクノリッジ許可ビット (R/W)

ACKE	0	アクノリッジ信号 (\overline{ACK}) の自動出力を禁止します (ACKTによる出力は可能)。	
	1	転送終了前にセットした場合	\overline{SCK} の9クロック目に同期して \overline{ACK} を出力します。
		転送終了後にセットした場合	セット命令実行直後の \overline{SCK} に同期して \overline{ACK} を出力します。

アクノリッジ・トリガ・ビット (W)

ACKT	転送終了後にセットすると、次の \overline{SCK} に同期して \overline{ACK} が出力されます。ACK信号出力後、自動的にクリア (0) されます。
------	--

注意 1 . シリアル転送終了前、および転送中にセット (1) しないでください。

2 . ACKTは、ソフトウェアでクリアすることはできません。

3 . ACKTをセットするときは、ACKE = 0にしてください。

コマンド検出フラグ (R)

CMDD	クリアされる条件 (CMDD = 0)	セットされる条件 (CMDD = 1)
①	転送スタート命令実行時	コマンド信号 (CMD) 検出時
②	バス・リリース信号 (REL) 検出時	
③	\overline{RESET} 信号発生時	
④	CSIE = 0 (図5 - 40参照)	

バス・リリース検出フラグ (R)

RELD	クリアされる条件 (RELD = 0)	セットされる条件 (RELD = 1)
①	転送スタート命令実行時	バス・リリース信号 (REL) 検出時
②	リセット信号発生時	
③	CSIE = 0 (図5 - 40参照)	
④	アドレス受信時にSVAとSIOが一致しなかったとき	

図5 - 41 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 (3/3)

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 SB0 (または、SB1) のクリア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 SB0 (または、SB1) のクリア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。

例1 . コマンド信号を出力する。

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

2 . RELD, CMDDをテストし、受信データの種別を判別して処理を分ける。

なお、この割り込みルーチンはWUP = 1とし、一致アドレスの場合のみ処理されるようにする。

```
SEL    MB15
SKF    RELD          ; RELDをテスト
BR     !ADRS
SKT    CMDD          ; CMDDをテスト
BR     !DATA
BR     !CMD

CMD : ..... ; コマンドを解釈
DATA : ..... ; データを処理
ADRS : ..... ; アドレスをデコード
```

(3) シフト・レジスタ (SIO)

図5 - 42にシフト・レジスタ周辺の構成を示します。SIOはパラレル-シリアルの変換を行い、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。

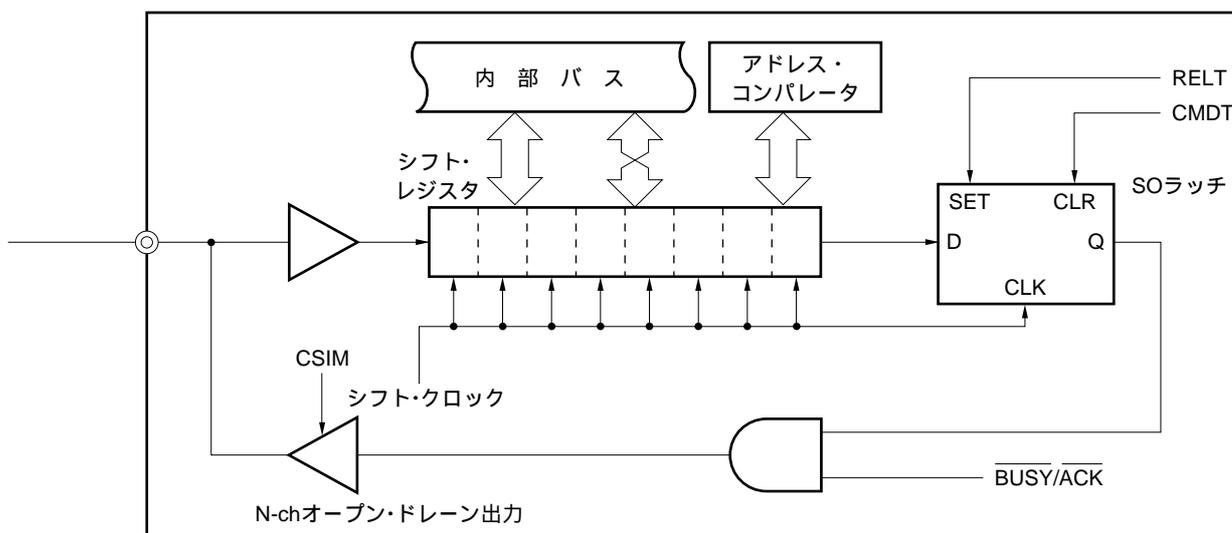
送信時はSIOに書き込まれたデータが、シリアル出力(SO)またはシリアル・データ・バス(SB0またはSB1)に出力されます。受信時はデータが、シリアル入力(SI)またはSB0またはSB1からSIOに読み込まれます。

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

8ビット送信/受信終了後、シフト動作は停止します。

図5 - 42 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、SBI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。

(4) スレーブ・アドレス・レジスタ (SVA)

SVAはスレーブがスレーブ・アドレス値(自分の指定番号)をセットするための8ビット・レジスタです。

操作は、8ビット操作命令で行います。

$\overline{\text{RESET}}$ 信号が発生により、SVAの値は不定になります。ただし、スタンバイ・モード時に $\overline{\text{RESET}}$ 信号が発生した場合は、SVAの値は保持されます。

SVAが持つ2つの機能を次に示します。

(a) スレーブ・アドレスの検出

[SBIモード時]

μ PD750008をスレーブ・デバイスとしてシリアル・バスに接続するとき 사용합니다。SVAはスレーブがスレーブ・アドレス値(自分の指定番号)をセットするための8ビット・レジスタです。マスタは接続されているスレーブに対して、特定のスレーブを選択するためのスレーブ・アドレスを出力します。アドレス・コンパレータによりこれらの2つのデータ(マスタから出力されたスレーブ・アドレスとSVAの値)を比較して、一致すると、そのスレーブが選択されたことになります。

このときシリアル動作モード・レジスタ(CSIM)のビット6(COI)は“1”になります。

アドレス受信時には、一致が検出されなければバス・リリース検出フラグ(RELD)をクリア(0)します。WUP=1のときには一致が検出されたときのみ、IRQCSIをセットします。この割り込み要求により、マスタから μ PD750008に通信要求があったことを知ることができます。

(b) エラーの検出

[2線式シリアルI/Oモード, SBIモード時]

μ PD750008をマスタ・デバイスとしてアドレス、コマンド、データを送信する場合、またスレーブ・デバイスとしてデータ送信する場合に、SVAはエラーの検出を行います(詳細は、5.6.6(6) エラーの検出または5.6.7(8) エラーの検出を参照)。

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。消費電力は低減されます。

このモードでは、シフト・レジスタはシフト動作は行わないので、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/ $\overline{\text{SCK}}$ 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。

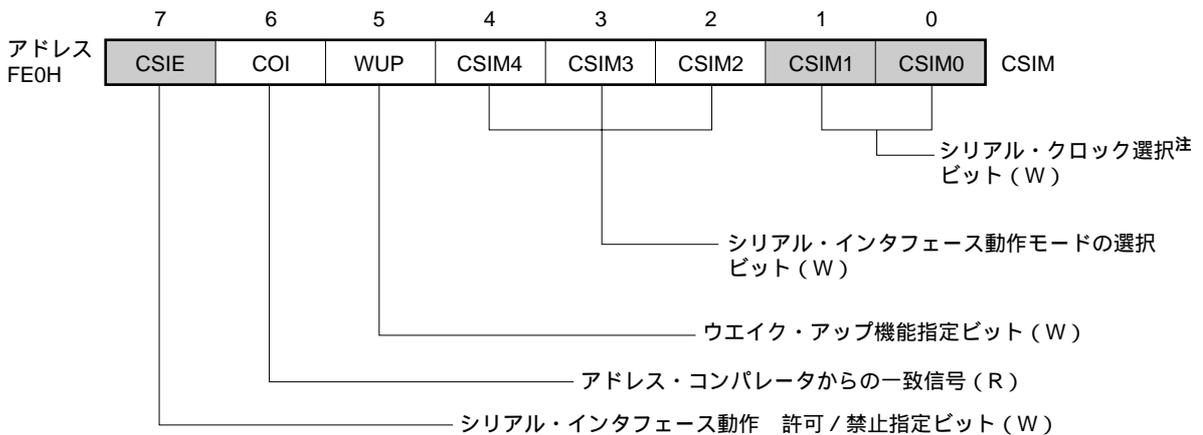
(1) レジスタの設定

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表します。



注 P01/ $\overline{\text{SCK}}$ 端子の状態を選択可能

備考 (R) 読み出しのみ可能

(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/ $\overline{\text{SCK}}$ 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/ $\overline{\text{SCK}}$ 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をクリアする場合は次の手順で行ってください。

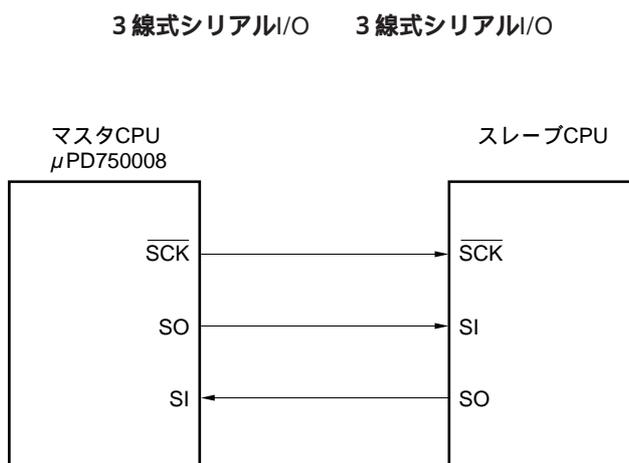
- ① 割り込み許可フラグ (IECSI) をクリアして割り込み禁止状態にする。
- ② CSIEをクリアする。
- ③ 割り込み要求フラグ (IRQCSI) をクリアする。

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、 μ PD7500シリーズ、87ADシリーズ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 - 43 3線式シリアルI/Oのシステム構成例



備考 μ PD750008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モードは、次の2つのレジスタを使用します。

- ・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)

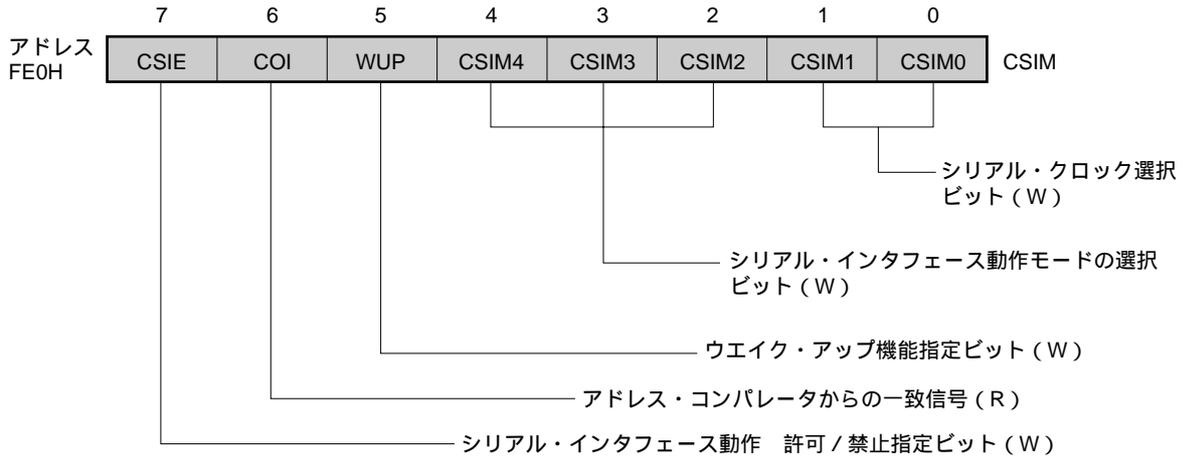
(a) シリアル動作モード・レジスタ (CSIM)

3線式シリアル/Oモードを使用する場合はCSIMを次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, 6, 5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/Oモードにおける使用ビットを表します。



備考 (R) 読み出しのみ可能

(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

アドレス・コンパレータからの信号 (R)

COI ^注	クリアされる条件 (COI = 0)	セットされる条件 (COI = 1)
		スレープ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致しなかったとき

注 COIの読み出しはシリアル転送開始前, 完了後にのみ有効です。転送中には不定の値が読み出されます。

また, 8ビット操作命令によって書き込まれたCOIのデータは無視されます。

ウエイク・アップ機能指定ビット (W)

WUP	0	シリアル転送終了時に毎回IRQCSIをセットする。
-----	---	---------------------------

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 ₇₋₀ XA (MSB先頭で転送)	SO/P02 (CMOS出力)	SI/P03 (入力)
		1	SIO ₀₋₇ 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	$\overline{\text{SCK}}$ 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・カウンタ出力 (TOUT0)	出力
1	0	$f_x/2^4$ (262 kHz) 注	
1	1	$f_x/2^3$ (524 kHz) 注	

注 () 内は $f_x = 4.19 \text{ MHz}$ 時

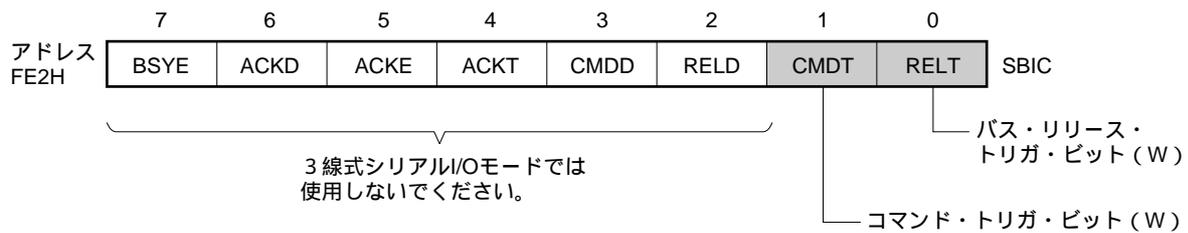
(b) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

3線式シリアルI/Oモードを使用する場合はSBICを次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表します。



備考 (W) 書き込みのみ可能

コマンド・トリガ・ビット

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。
------	--

注意 3線式シリアルI/OモードではRELT, CMDT以外は使用しないでください。

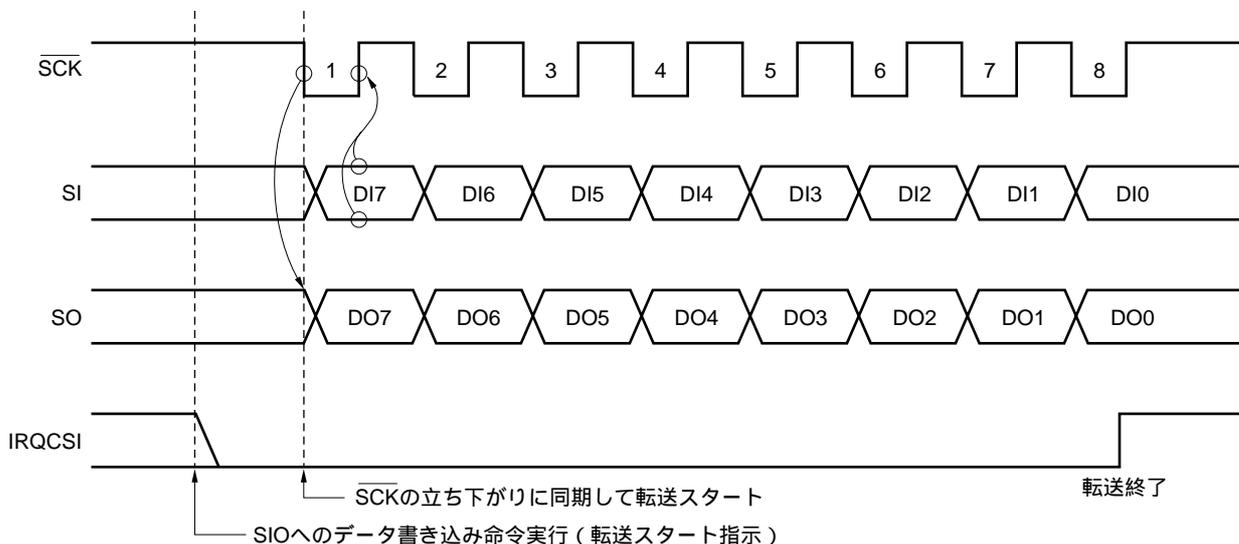
(2) 通信動作

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 ($\overline{\text{SCK}}$) 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、 $\overline{\text{SCK}}$ 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 - 44 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。

$\overline{\text{SCK}}$ 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 (5.6.8 $\overline{\text{SCK}}$ 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。

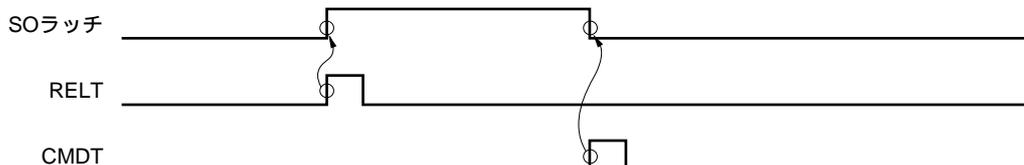
表5 - 7 シリアル・クロックの選択と応用 (3線式シリアルI/Oモード時)

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SCK	8ビット・データ転送終了により, 自動的にマスクする。	① 動作停止モード時 (CSIE = 0) ② 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 ③ SCKがハイ・レベルのとき	スレープ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 - 45にRELT, CMDTの動作を示します。

図5 - 45 RELT, CMDTの動作



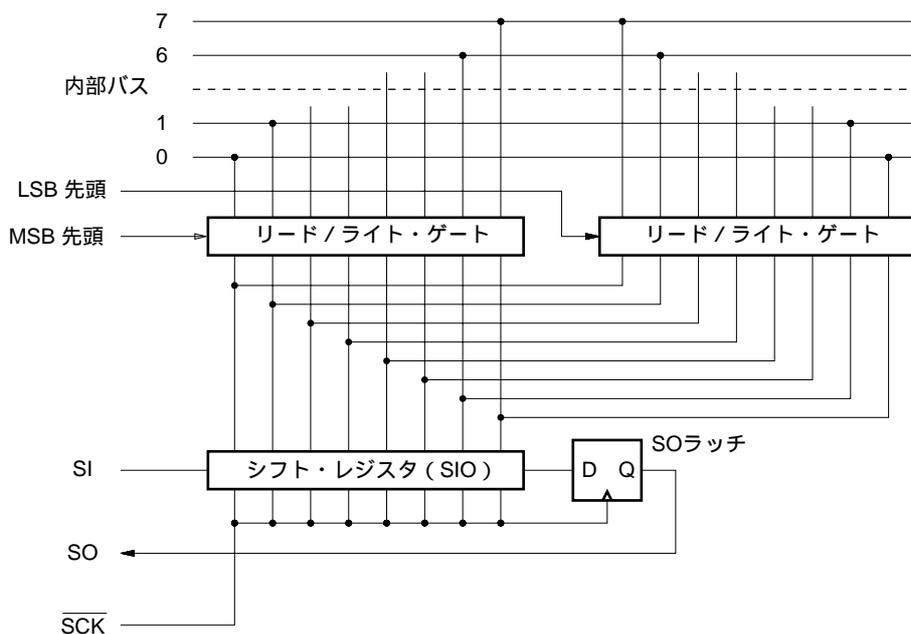
(5) MSB/LSB先頭の切り替え

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5-46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5-46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5-46 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にシフト・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、もしくは $\overline{\text{SCK}}$ 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から転送データを取り出す
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線シリアルI/Oモードの応用

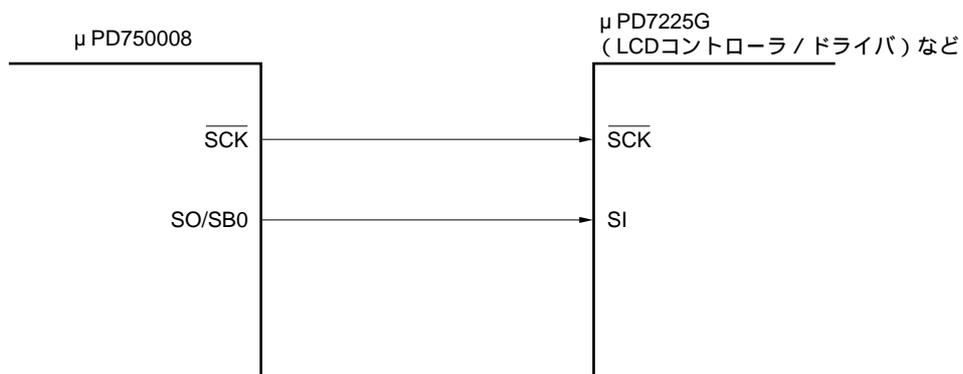
(a) 転送クロック262 kHz (4.19 MHz動作時)で、データをMSB先頭で転送する(マスタ動作)。

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
    
```

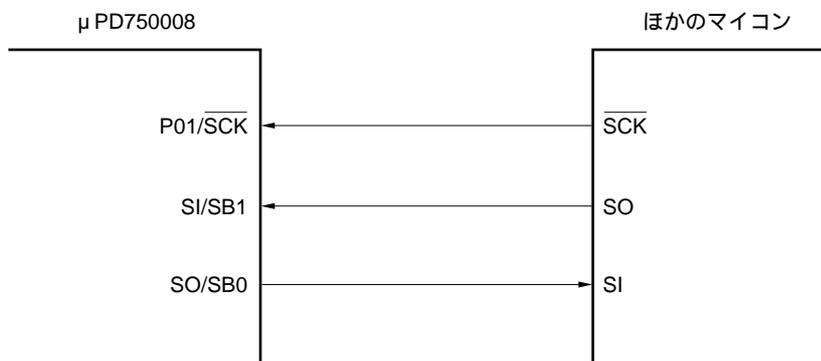
注意 2回目以降はSIOへのデータのセット(MOV SIO, XAまたはXCH XA, SIO)によって転送スタートできます。



この応用の場合、μ PD750008のSI/SB1端子は入力として使用できます。

(b) 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

```

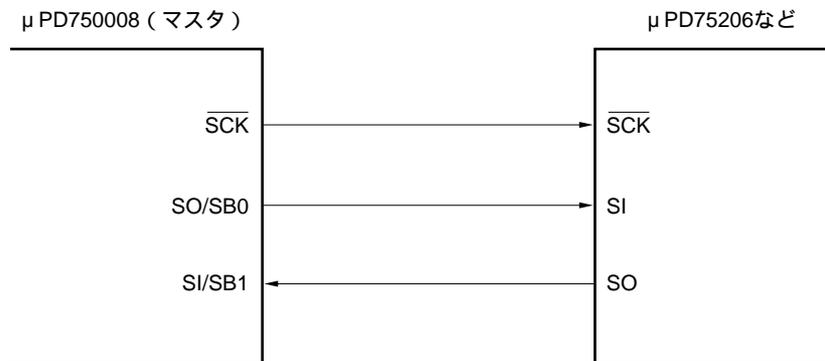
CLR1  MBE
MOV   XA, #84H
MOV   CSIM, XA ;シリアル動作停止, LSB/MSB反転モード, 外部クロック
MOV   XA, TDATA
MOV   SIO, XA ;転送データをセット, 転送スタート
EI    IECSI
EI
  
```

割り込みルーチン (MBE = 0)

```

MOV   XA, TDATA
XCH   XA, SIO ;受信データと送信データを転送スタート
MOV   RDATA, XA ;受信データをセーブ
RETI
  
```

(c) 転送クロックに524kHz (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

```

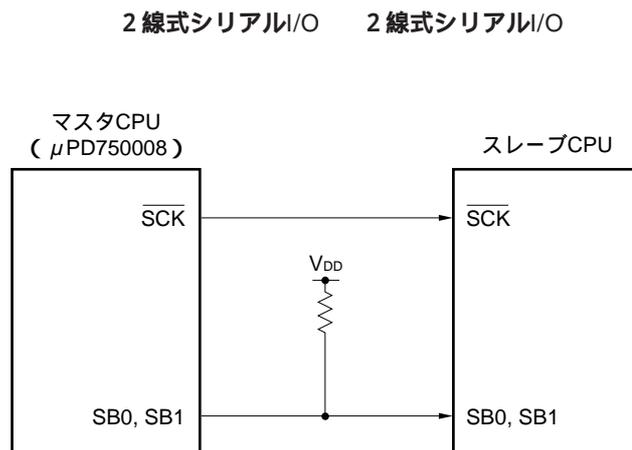
CLR1  MBE
MOV   XA, # 10000011B
MOV   CSIM, XA          ; 転送モードの設定
MOV   XA, TDATA
MOV   SIO, XA          ; 転送データをセット, 転送スタート
      .....
LOOP : SKTCLR  IRQCSI   ; IRQCSIのテスト
      BR      LOOP
MOV   XA, SIO          ; 受信データを取り込み
  
```

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・データ出力/入力 (SB0またはSB1) の2本のラインで通信を行います。

図5 - 47 2線式シリアルI/Oのシステム構成例



備考 μ PD750008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)

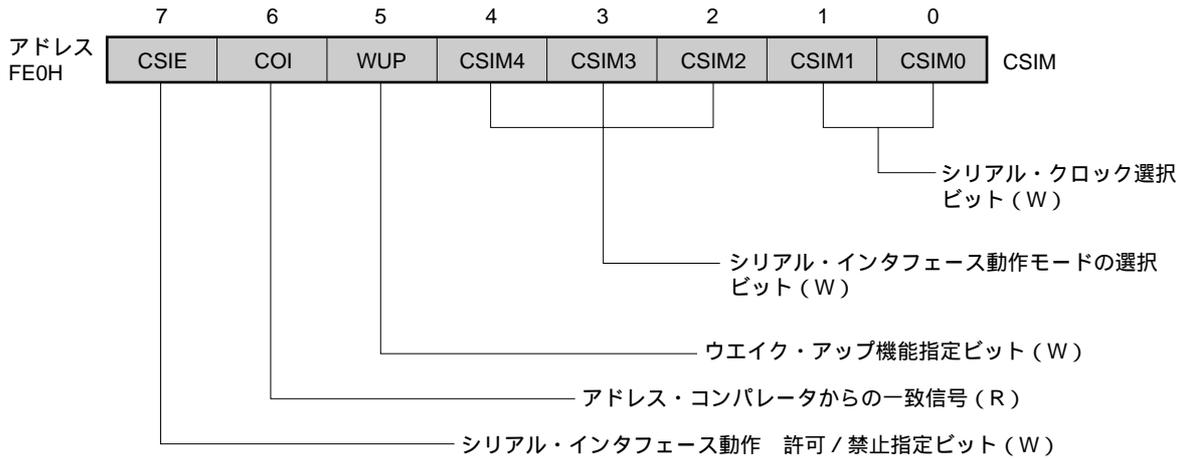
(a) シリアル動作モード・レジスタ (CSIM)

2線式シリアルI/Oモードを使用する場合はCSIMを次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, 6, 5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表します。



備考 (R) 読み出しのみ可能

(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	カウント動作	セット可能	各モードでの機能とポート0兼用

アドレス・コンパレータからの信号 (R)

COI ^注	クリアされる条件 (COI = 0)	セットされる条件 (COI = 1)
	スレーブ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致しなかったとき	スレーブ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致したとき

注 COIの読み出しはシリアル転送開始前, 完了後にのみ有効です。転送中には不定の値が読み出されます。

また, 8ビット操作命令によって書き込まれたCOIのデータは無視されます。

ウェイク・アップ機能指定ビット (W)

WUP	0	シリアル転送終了時に毎回IRQCSIをセットする。
-----	---	---------------------------

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
0	1	1	SIO ₇₋₀ XA (MSB先頭で転送)	SB0/P02 (N-chオープン・ドレイン) 入出力	P03入力
1				P02入力	SB1/P03 (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	$\overline{\text{SCK}}$ 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・カウンタ出力 (TOUT0)	出力
1	0	$f_x/2^6$ (65.5 kHz)	
1	1		

備考 ()内は $f_x = 4.19$ MHz時

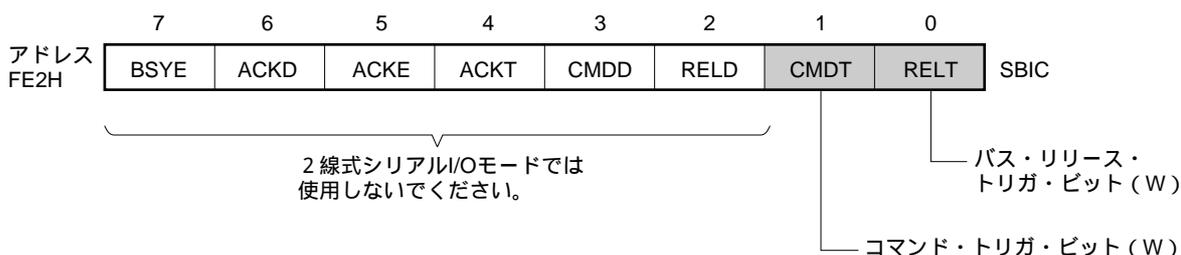
(b) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

2線式シリアル/I/Oモードを使用する場合はSBICを次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表します。



備考 (W) 書き込みのみ可能

コマンド・トリガ・ビット

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。
------	--

注意 2線式シリアル/I/OモードではRELT, CMDT以外は使用しないでください。

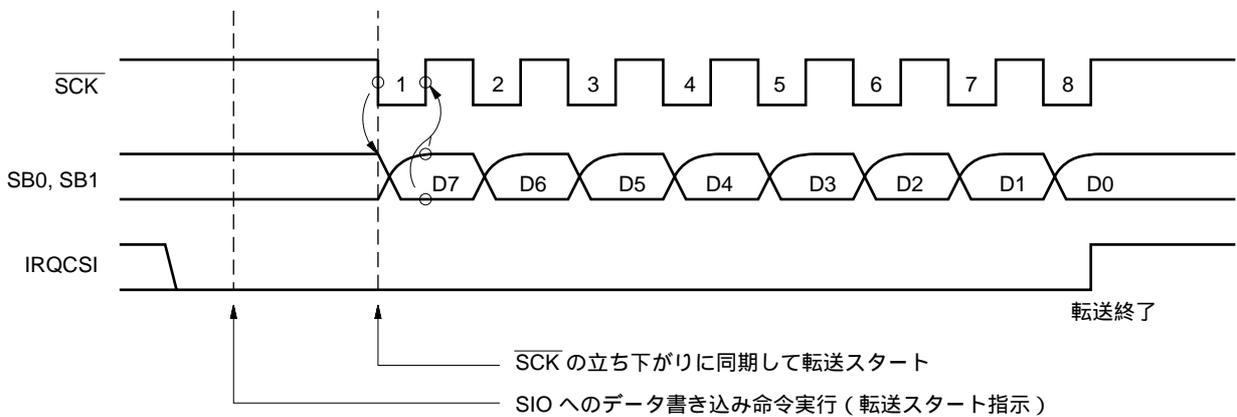
(2) 通信動作

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 - 48 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・バ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。

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 (5.6.8 \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。

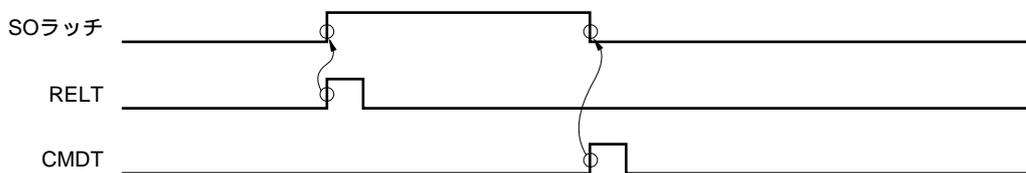
表5 - 8 シリアル・クロックの選択と応用 (2線式シリアルI/Oモード時)

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SCK	8ビット・データ転送終了により, 自動的にマスクする。	① 動作停止モード時 (CSIE = 0)	スレーブCPU
0	1	TOUT F/F		② 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	任意速シリアル転送
1	0	$f_x/2^6$		③ SCKがハイ・レベルのとき	低速シリアル転送
1	1				

(4) 各種信号

図5 - 49にRELT, CMDTの動作を示します。

図5 - 49 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にソフト・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、もしくは $\overline{\text{SCK}}$ 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/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。

(a) 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。

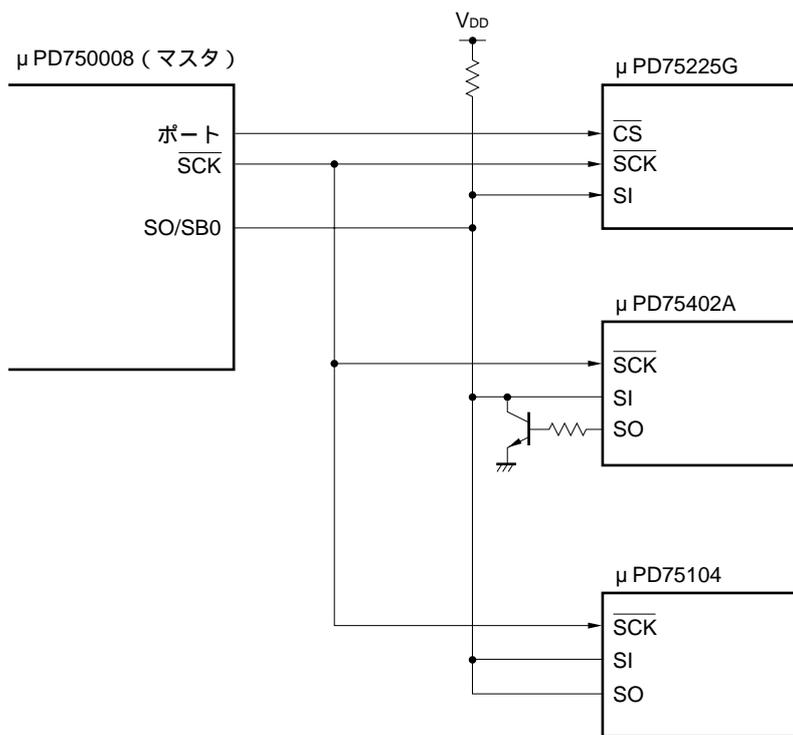
(b) スレーブ・アドレス・レジスタ(SVA)を利用する方法

送信データをSVAにもセットし、送信を行います。送信完了後に、シリアル動作モード・レジスタ(CSIM)のCOIビット(アドレス・コンパレータからの一致信号)をテストし、“1”ならば正常な送信、“0”ならば送信エラーと判断します。

(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 μ PD750008をマスタとし、スレーブに μ PD75104, μ PD75402A, μ PD7225Gを接続してシステム構成する。



この例のようにバス構成する場合、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、あらかじめシフト・レジスタにFFHを書き込んでSO端子にハイを出力し、出力バッファをオフしてバスを開放します。

μ 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、取り決めておきます。

シリアル・クロックはマスタ・マイコンである μ PD750008が出力し、他のスレーブ・マイコンはいずれも外部クロックで動作します。

5.6.7 SBIモードの動作

SBI (シリアル・バス・インタフェース) は、NECシリアル・バス・フォーマット準拠の高速シリアル・インタフェース方式です。

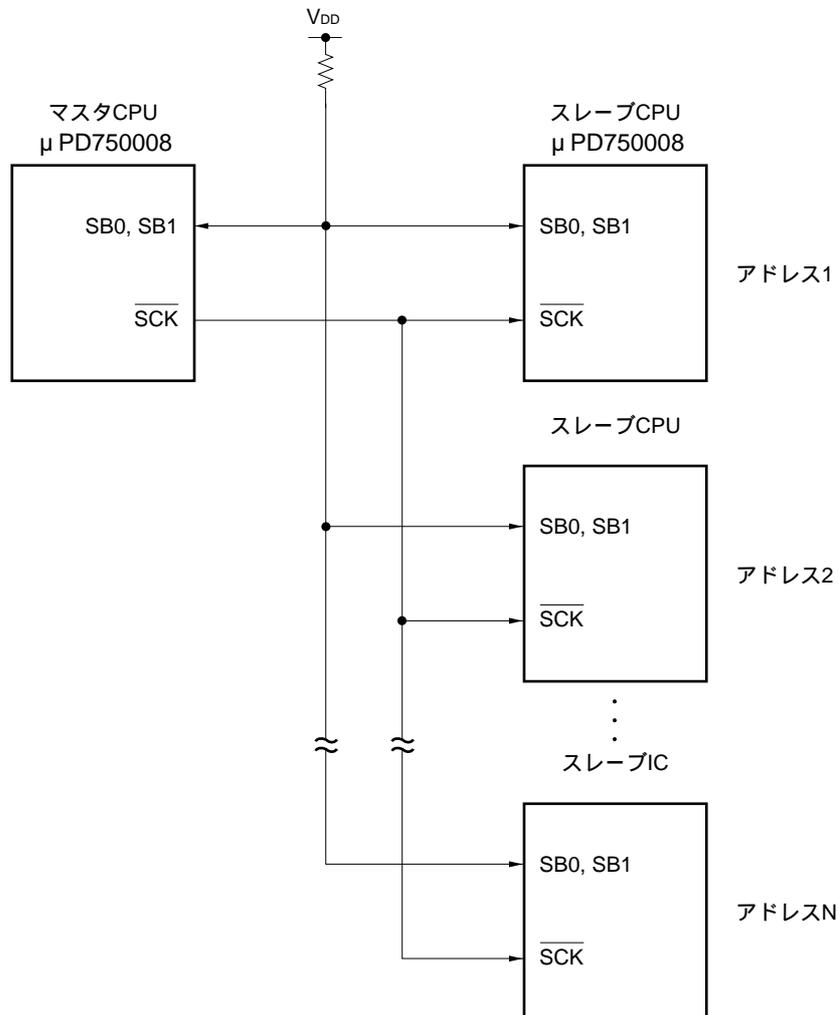
SBI は、シングル・マスタの高速シリアル・バスで、2本の信号線で複数のデバイスとの通信を行うことができるように、クロック同期式のシリアルI/O方式に、バス構成のための機能が追加されたフォーマットになっています。そのため複数のマイコンや周辺ICでシリアル・バスを構成する場合に、使用するポート数や基板上の配線数を削減することができます。

また、マスタはスレーブに対してシリアル・データ・バス上にシリアル通信の対象デバイス選択のための“アドレス”、対象デバイスに対して指令を与える“コマンド”および実際の“データ”を出力することができます。スレーブは受信したデータをハードウェアにより、“アドレス”、“コマンド”、“データ”に判別することができます。この機能により、応用プログラムのシリアル・インタフェースの制御部分を簡単にすることができます。

SBI機能は「75Xシリーズ」,「75XLシリーズ」, 8, 16ビット・シングルチップ・マイクロコンピュータ「78Kシリーズ」の数種のデバイスに内蔵されています。

SBIに準拠するシリアル・インタフェースを有するCPUや、周辺ICを使用した場合のシリアル・バス構成例を図5 - 50に示します。

図5-50 SBIのシステム構成例



- 注意1 . SBIでは、シリアル・データ・バス端子SB0（またはSB1）が、オープン・ドレイン出力になっているため、シリアル・データ・バス・ラインはワイアード・オア状態になります。また、シリアル・データ・バス・ラインには、プルアップ抵抗が必要です。
- 2 . マスタ/スレーブの交換処理を行う場合は、シリアル・クロック・ライン (\overline{SCK}) の入力/出力の切り替えがマスタ、スレーブ間で非同期に行われるため、 \overline{SCK} にもプルアップ抵抗が必要となります。

(1) SBIの機能

従来のシリアル/I/O方式では、データ転送機能しかないために、複数のデバイスを接続してシリアル・バスを構成した場合に、チップ・セレクト信号やコマンド/データの区別、ビジー状態の判断などのため多くのポートや配線が必要となります。また、これらの制御を、ソフトウェアで行おうとすると、ソフトウェアの負担が大きくなってしまいます。

SBIでは、シリアル・クロック $\overline{\text{SCK}}$ と、シリアル・データ・バス (SB0またはSB1) の2本の信号線でシリアル・バスを構成することができます。そのため、マイコンのポート数の削減や基板内の配線や引き回しの減少に有効となります。

SBIの機能について以下に示します。

(a) アドレス/コマンド/データの判断機能

シリアル・データを、アドレス、コマンド、およびデータの3種類に区別します。

(b) アドレスによるチップ・セレクト機能

マスタは、アドレスの送信により、チップ・セレクト(選択)を行います。

(c) ウェイク・アップ機能

スレーブはウェイク・アップ機能(ソフトウェアで設定/解除が可能)によりアドレス受信の判断(チップ・セレクトの判断)を容易に行うことができます。

ウェイク・アップ機能を設定した場合、一致アドレス受信時に割り込み(IRQCSI)が発生します。そのため、複数のデバイスと通信を行う場合も、選択されたスレーブ以外のCPUはシリアル通信に関係なく動作することができます。

(d) アクノリッジ信号 ($\overline{\text{ACK}}$) 制御機能

シリアル・データの受信確認のための、アクノリッジ信号を制御します。

(e) ビジー信号 ($\overline{\text{BUSY}}$) 制御機能

スレーブのビジー状態を知らせるための、ビジー信号を制御します。

(2) SBIの定義

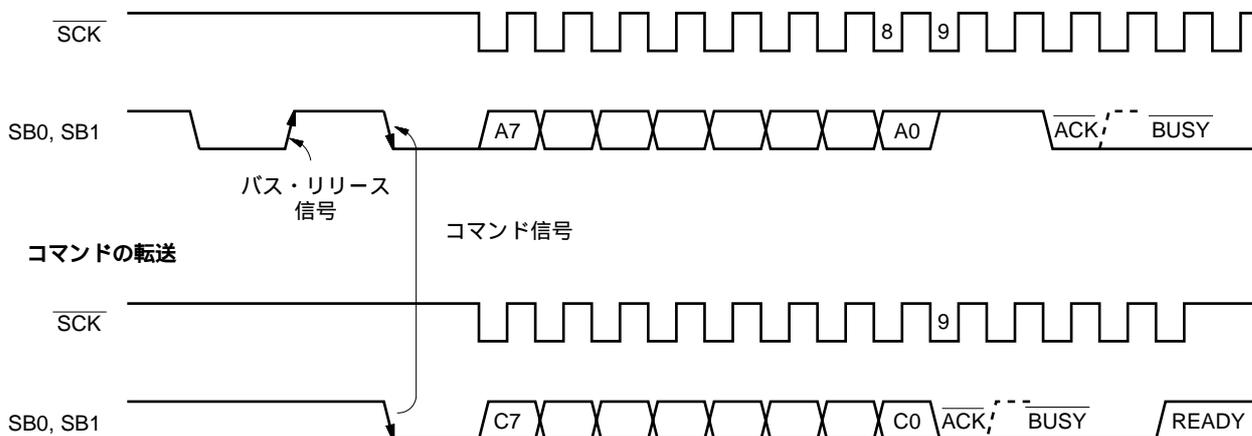
SBIのシリアル・データのフォーマットおよび、使用する信号の意味について説明します。

SBIで転送されるシリアル・データは、「アドレス」、「コマンド」、「データ」の3種類に区別されます。また、シリアル・データは、以下の構成で1フレームを形成します。

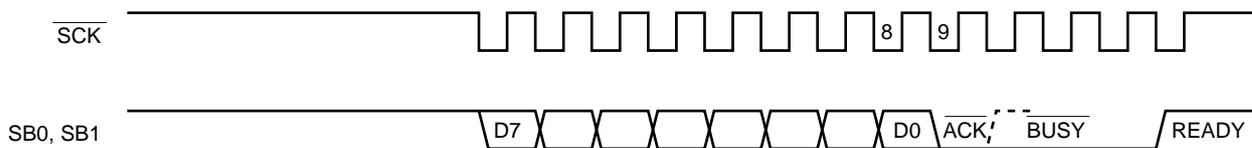
図5-51に、アドレス、コマンド、およびデータの転送タイミングを示します。

図 5 - 51 SBI 転送のタイミング

アドレスの転送



データの転送



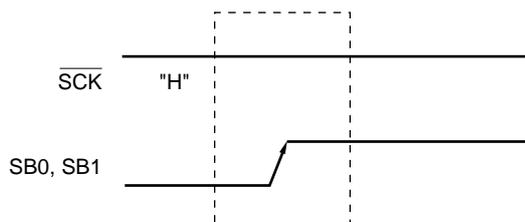
バス・リリース信号およびコマンド信号はマスタが出力します。また、 $\overline{\text{BUSY}}$ はスレーブが出力します。 $\overline{\text{ACK}}$ はマスタ、スレーブのどちらでも出力できます（通常、8ビット・データの受信側が出力します）。

シリアル・クロックは、8ビット・データ転送開始から、 $\overline{\text{BUSY}}$ が解除されるまで、マスタが出力し続けます。

(a) バス・リリース信号 (REL)

バス・リリース信号は、 $\overline{\text{SCK}}$ ラインがハイ・レベルのとき（シリアル・クロックが出力されていない場合）に、SB0（またはSB1）ラインがロウ・レベルからハイ・レベルに変化した信号です。この信号は、マスタが出力します。

図5 - 52 バス・リリース信号

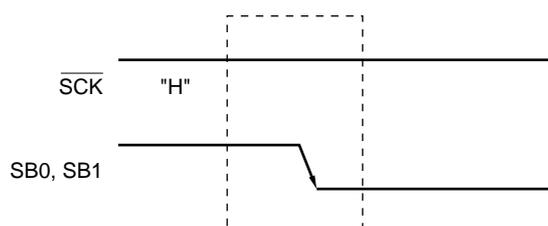


バス・リリース信号は、これからマスタがスレーブに対してアドレスを送信することを示すものです。スレーブは、バス・リリース信号を検出するハードウェアを内蔵しています。

(b) コマンド信号 (CMD)

コマンド信号は、 $\overline{\text{SCK}}$ ラインが、ハイ・レベルのとき（シリアル・クロックが出力されていない場合）に、SB0（またはSB1）ラインがハイ・レベルからロウ・レベルに変化した信号です。この信号は、マスタが出力します。

図5 - 53 コマンド信号

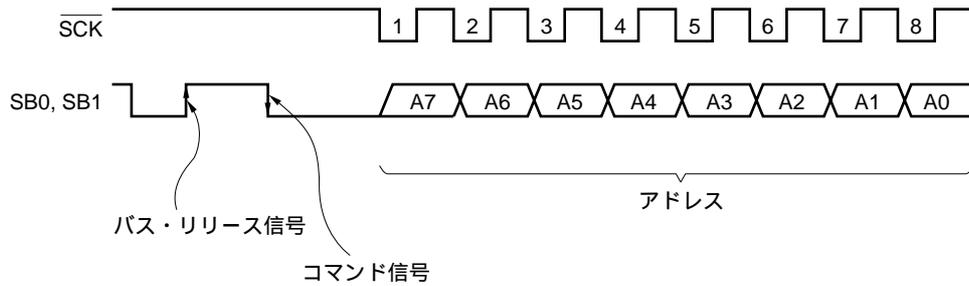


スレーブは、コマンド信号を検出するハードウェアを内蔵しています。

(c) アドレス

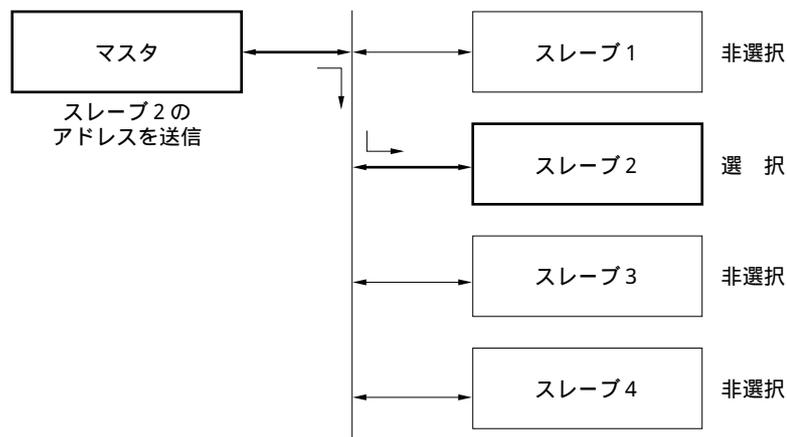
アドレスは、マスタがバス・ラインに接続されているスレーブに対して、特定のスレーブを選択するために出力する8ビット・データです。

図5 - 54 アドレス



バス・リリース信号，コマンド信号に続く8ビット・データはアドレスと定義されています。スレーブでは、ハードウェアでこの条件を検出し、8ビット・データが自分の指定番号（スレーブ・アドレス）と一致しているかをハードウェアでチェックします。このとき、8ビット・データと、スレーブ・アドレスが一致すると、そのスレーブが選択されたことになり、以後、マスタから切り離し指示があるまで、マスタとの通信を行います。

図5 - 55 アドレスによるスレーブの選択



(d) コマンド, データ

アドレスの送信により選択したスレーブに対して、マスタはコマンドの送信や、データの送受信を行います。

図5 - 56 コマンド

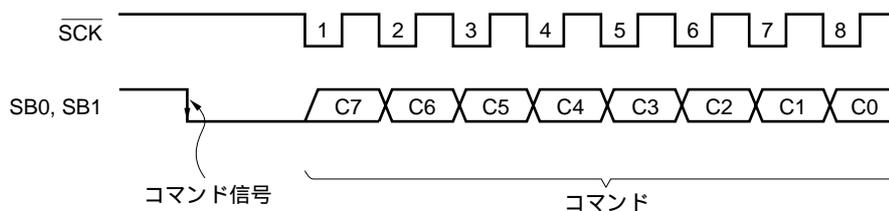
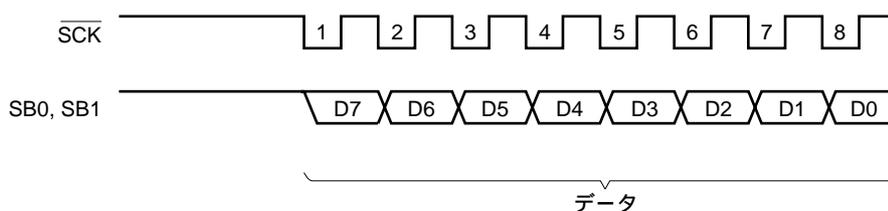


図5 - 57 データ

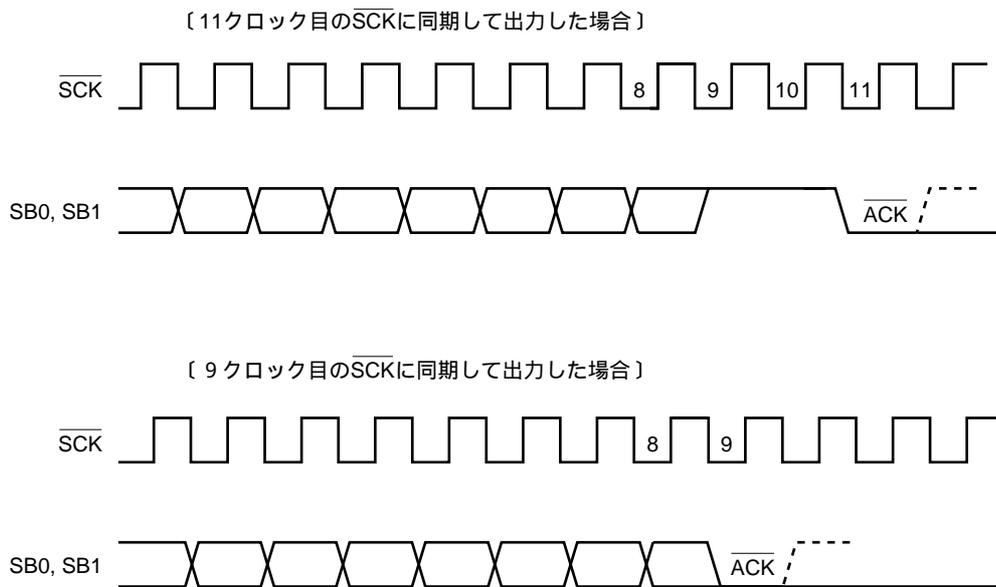


コマンド信号の次の8ビット・データはコマンドと定義されています。コマンド信号なしの8ビット・データはデータと定義されています。コマンド, データの使用方法は、通信の仕様によって任意に決定することができます。

(e) アクノリッジ信号 ($\overline{\text{ACK}}$)

アクノリッジ信号は、送信側と受信側の間における、シリアル・データ受信の確認のための信号です。

図5 - 58 アクノリッジ信号



アクノリッジ信号は、8ビット・データ転送後の $\overline{\text{SCK}}$ の立ち下がりに同期したワンショット・パルスで、その位置は任意で何クロック目の $\overline{\text{SCK}}$ に同期させてもかまいません。

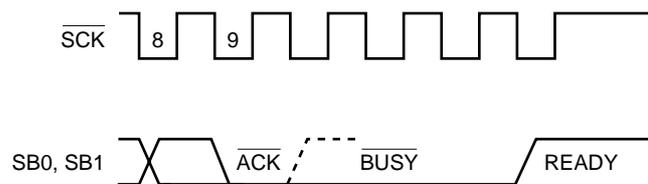
送信側は、8ビット・データ送信後、受信側がアクノリッジ信号を返したかをチェックします。データ送信後、一定時間、アクノリッジ信号が返らない場合は、受信が正しく行われなかったものと判断することができます。

(f) ビジィ信号 ($\overline{\text{BUSY}}$) , レディ信号 (READY)

ビジィ信号は、スレーブがデータの送受信のための準備中であることをマスタに知らせるための信号です。

レディ信号は、スレーブがデータの送受信が可能であることをマスタに知らせるための信号です。

図5 - 59 ビジィ信号, レディ信号



SBIでは、スレーブが、 SB0 (または SB1) ラインをロウ・レベルにすることにより、マスタにビジィ状態を知らせます。

ビジィ信号は、マスタ、またはスレーブの出力したアクノリッジ信号にひき続いて出力されます。ビジィ信号は $\overline{\text{SCK}}$ の立ち下がりに同期して、設定/解除を行います。マスタは、ビジィ信号が解除されると自動的にシリアル・クロック $\overline{\text{SCK}}$ の出力を終了します。

マスタは、ビジィ信号が解除され、レディ信号の状態になると次の転送を開始することができます。

(3) レジスタの設定

SBIモードの動作は、次の2つのレジスタの設定により行います。

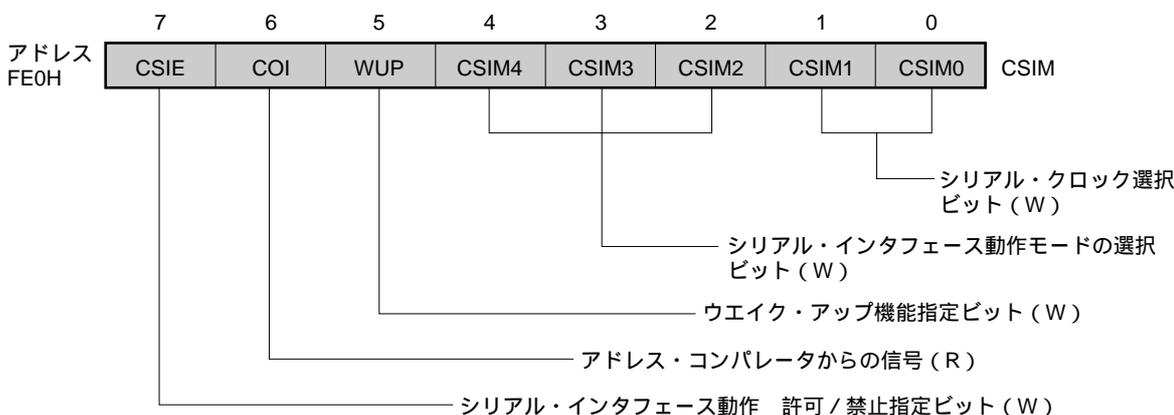
- ・シリアル動作モード・レジスタ (CSIM)
- ・シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

(a) シリアル動作モード・レジスタ (CSIM)

SBIモードを使用する場合はCSIMを次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・レジスタ参照)。

操作は8ビット操作命令により行います。ビット7, 6, 5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SBIモードにおける使用ビットを表します。



備考 (R) 読み出しのみ可能
(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

アドレス・コンパレータからの信号 (R)

COI ^注	クリアされる条件 (COI = 0)	セットされる条件 (COI = 1)
		スレーブ・アドレス・レジスタ (SVA) とシフト・レジスタのデータが一致しなかったとき

注 COIの読み出しはシリアル転送開始前、完了後にのみ有効です。転送中には不定の値が読み出されます。また、8ビット操作命令によって書き込まれたCOIのデータは無視されます。

ウエイク・アップ機能指定ビット (W)

WUP	0	1
	SBIモードのマスク状態でのシリアル転送終了時に毎回IRQCSIをセットする。	SBIモードのスレーブ時のみに使用。バス・リリース後に受信したアドレスがスレーブ・アドレス・レジスタのデータと一致したとき (ウエイク・アップ状態) のみIRQCSIをセットする。SB0, SB1はハイ・インピーダンス。

注意 $\overline{\text{BUSY}}$ 信号出力中にWUP = 1にすると、 $\overline{\text{BUSY}}$ が解除されません。SBIでは、 $\overline{\text{BUSY}}$ の解除指示後、次のシリアル・クロック ($\overline{\text{SCK}}$) の立ち下がりまで $\overline{\text{BUSY}}$ 信号が出力されます。WUP = 1とするときは、必ず $\overline{\text{BUSY}}$ 信号を解除したのちに、SB0 (またはSB1) 端子がハイ・レベルになったことを確認してから、WUP = 1としてください。

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
0	1	0	SIO ₇₋₀ XA (MSB先頭で転送)	SB0/P02 (N-chオープン・ドレイン) 入出力	P03入力
1				P02入力	SB1/P03 (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	$\overline{\text{SCK}}$ 端子モード
0	0	$\overline{\text{SCK}}$ 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・カウンタ出力 (TOUT0)	出力
1	0	$f_x/2^4$ (262 kHz)	
1	1	$f_x/2^3$ (524 kHz)	

備考 () 内は = 4.19 MHz時

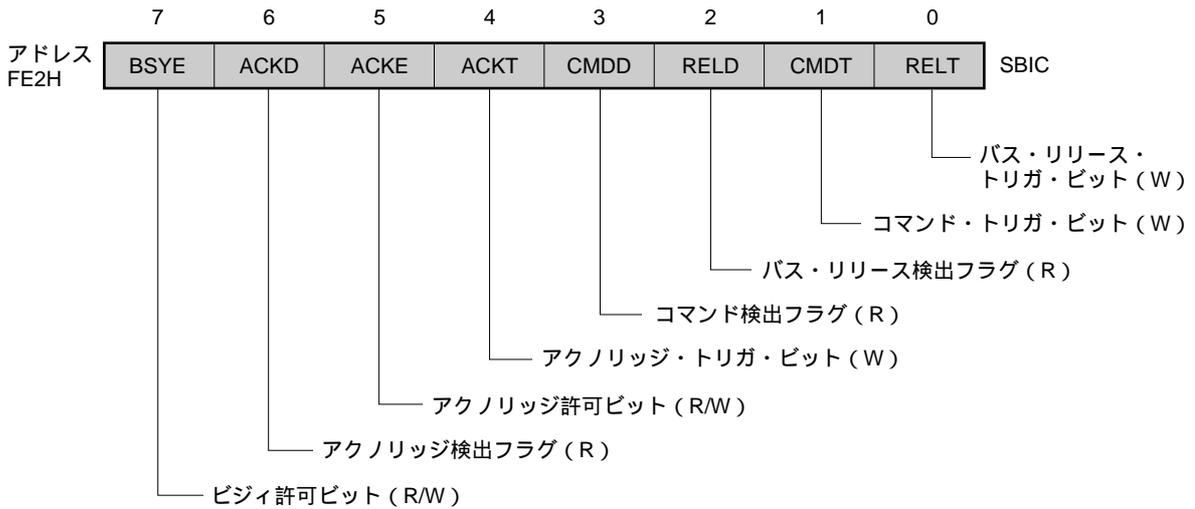
(b) シリアル・バス・インタフェース・コントロール・レジスタ (SBIC)

SBIモードを使用する場合はSBICを次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SBIモードにおける使用ビットを表します。



- 備考** (R) 読み出しのみ可能
 (W) 書き込みのみ可能
 (R/W) 読み出し / 書き込みともに可能

ビジィ許可ビット (R/W)

BSYE	0	① ビジィ信号の自動出力禁止 ② クリア命令実行直後の、 \overline{SCK} の立ち下がりに同期して、ビジィ信号の出力を停止
	1	アクノリッジ信号に続き \overline{SCK} の立ち下がりに同期して、ビジィ信号を出力

アクノリッジ検出フラグ (R)

ACKD	クリアされる条件 (ACKD = 0)		セットされる条件 (ACKD = 1)	
	①	転送スタート時	アクノリッジ信号 (\overline{ACK}) 検出時	
	②	リセット入力時	(\overline{SCK} の立ち上がりに同期します)	

アクノリッジ許可ビット (R/W)

ACKE	0	アクノリッジ信号の自動出力を禁止します (ACKTによる出力は可能)		
	1	転送終了前にセットした場合	\overline{SCK} の9クロック目に同期して \overline{ACK} を出力します。	
		転送終了後にセットした場合	セット命令実行直後の \overline{SCK} に同期して \overline{ACK} を出力します。	

ACKトリガ・ビット (W)

ACKT	転送終了後にセットすると、次のSCKに同期してACKが出力されます。ACK信号出力後、自動的にクリア(0)されます。
------	--

注意 1. シリアル転送終了前、および転送中にセット(1)しないでください。

2. ACKTは、ソフトウェアでクリアすることはできません。

3. ACKTをセットするときは、ACKE = 0にしてください。

コマンド検出フラグ (R)

CMDD	クリアされる条件 (CMDD = 0)	セットされる条件 (CMDD = 1)
	① 転送スタート命令実行時 ② バス・リリース信号 (REL) 検出時 ③ リセット入力時 ④ CSIE = 0 (図5 - 40参照)	コマンド信号 (CMD) 検出時

バス・リリース検出フラグ (R)

RELD	クリアされる条件 (RELD = 0)	セットされる条件 (RELD = 1)
	① 転送スタート命令実行時 ② リセット入力時 ③ CSIE = 0 (図5 - 40参照) ④ アドレス受信時にSVAとSIOが一致しなかったとき	バス・リリース信号 (REL) 検出時

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 SB0 (または、SB1) のセットはシリアル転送中に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 SB0 (または、SB1) のセットはシリアル転送中に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。

(4) 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 (SBI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SCK	8ビット・データ転送終了により, 自動的にマスクする。	① 動作停止モード時 (CSIE = 0) ② 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 ③ SCKがハイ・レベルのとき	スレープCPU
0	1	TOUT F/F			任意速シリアル転送
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

内部システム・クロックを選択した場合, SCKは内部に対して8発で停止しますが, 外部に対しては, スレープがレディ状態になるまでカウントします。

(5) 各種信号

SBIにおける各種の信号と, SBIC上のフラグの動作について図5 - 60 ~ 5 - 65に示します。また, SBIの各種の信号の一覧を表5 - 10に示します。

図5 - 60 RELT, CMDT, RELD, CMDD (マスタ) の動作

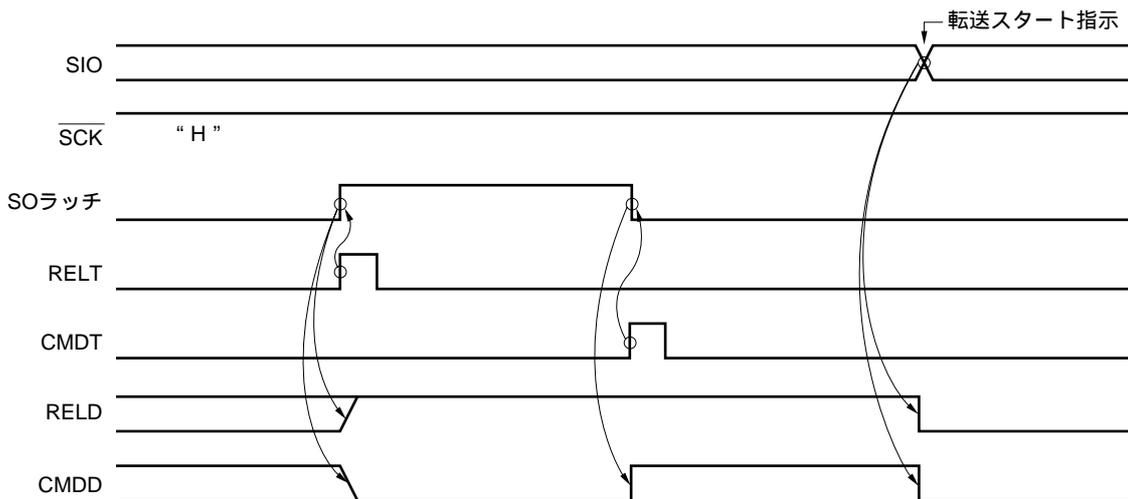


図5 - 61 RELT , CMDT , RELD , CMDD (スレーブ) の動作

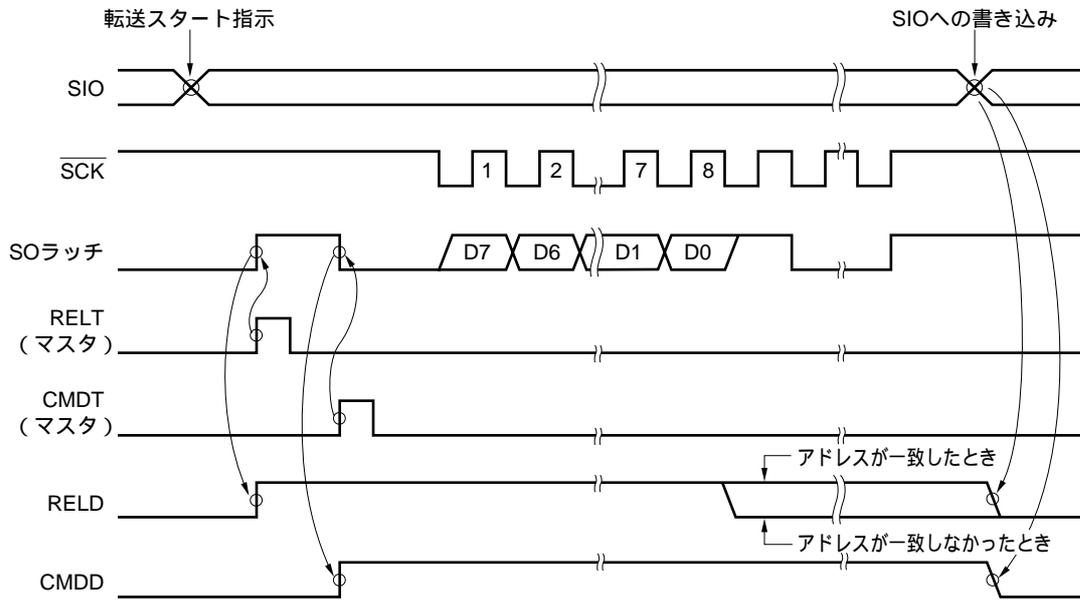
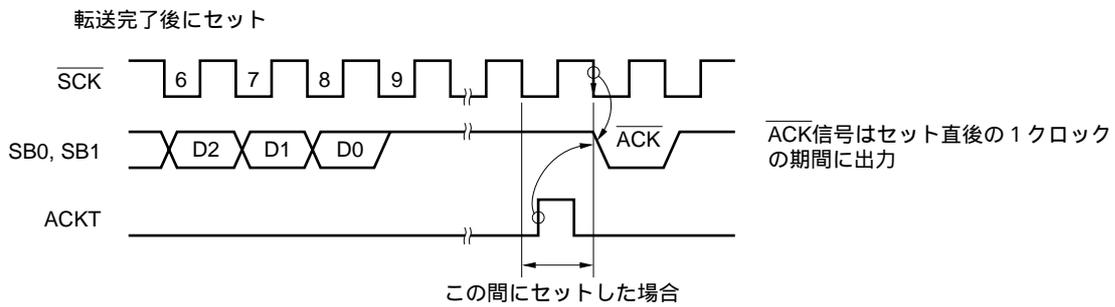


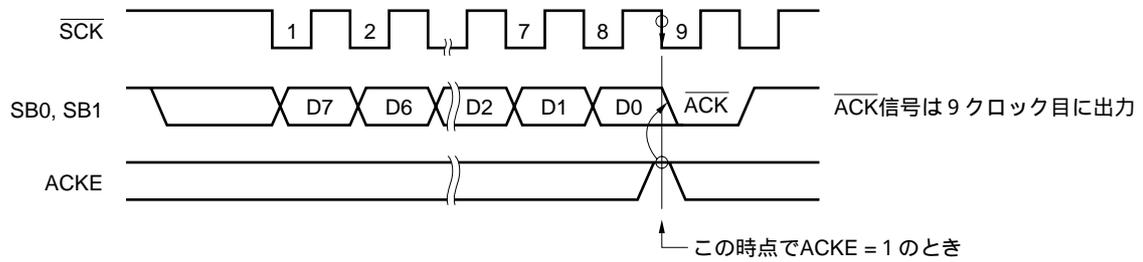
図5 - 62 ACKTの動作



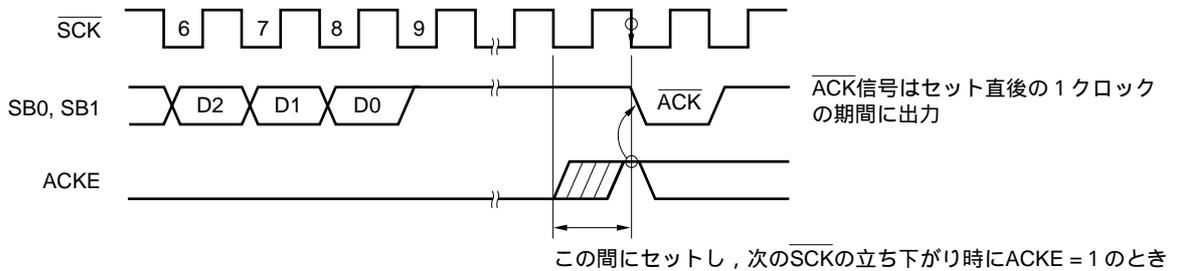
注意 ACKTは転送終了前にはセットしないでください。

図5 - 63 ACKEの動作

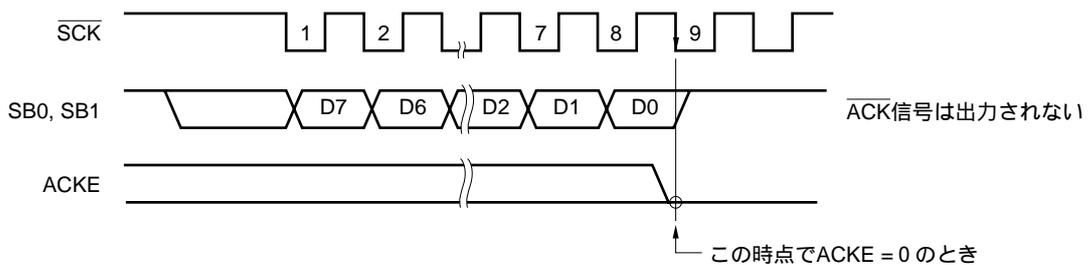
(a) 転送完了時にACKE = 1の場合



(b) 転送完了後にセットした場合



(c) 転送完了時にACKE = 0の場合



(d) ACKE = 1の期間が短い場合

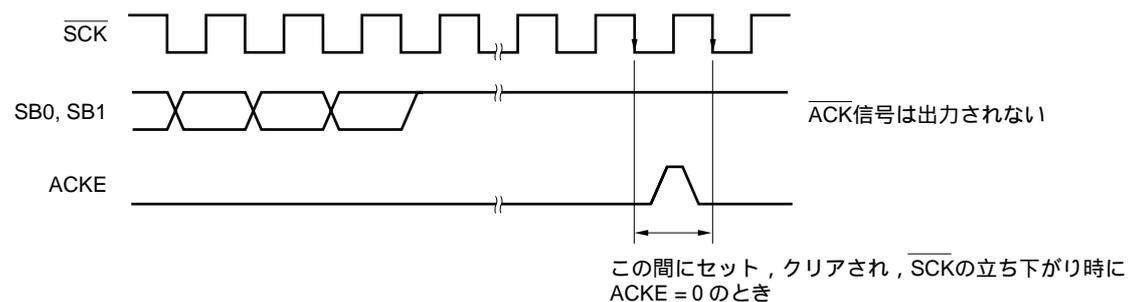
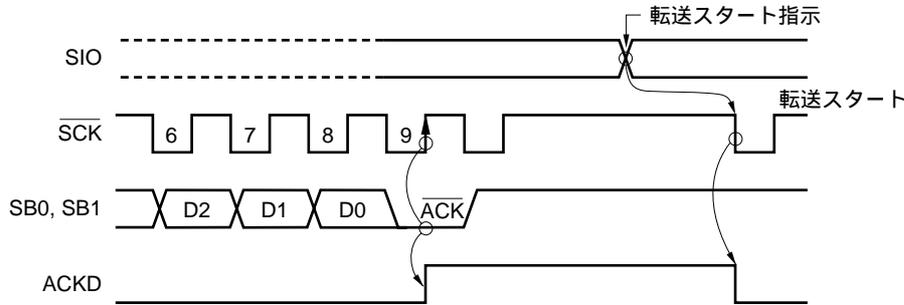
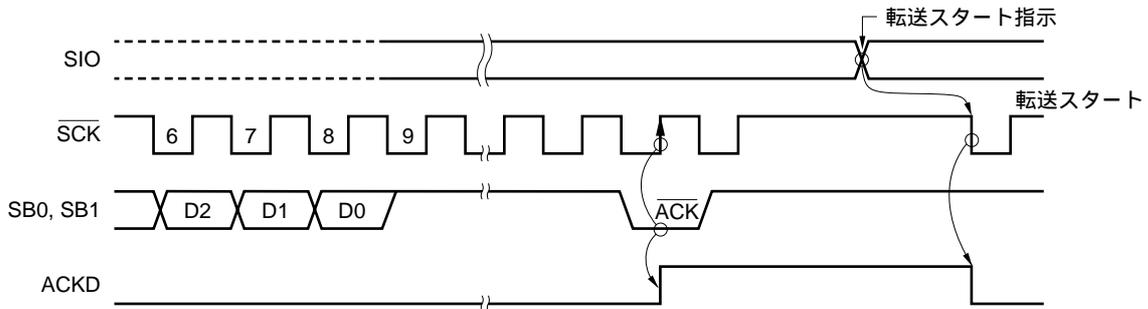


図5 - 64 ACKDの動作

(a) $\overline{\text{SCK}}$ の9クロック目の期間に $\overline{\text{ACK}}$ 信号が出力された場合



(b) $\overline{\text{SCK}}$ の9クロック目以降に $\overline{\text{ACK}}$ 信号が出力された場合



(c) BUSY中に転送スタート指示した場合のクリアのタイミング

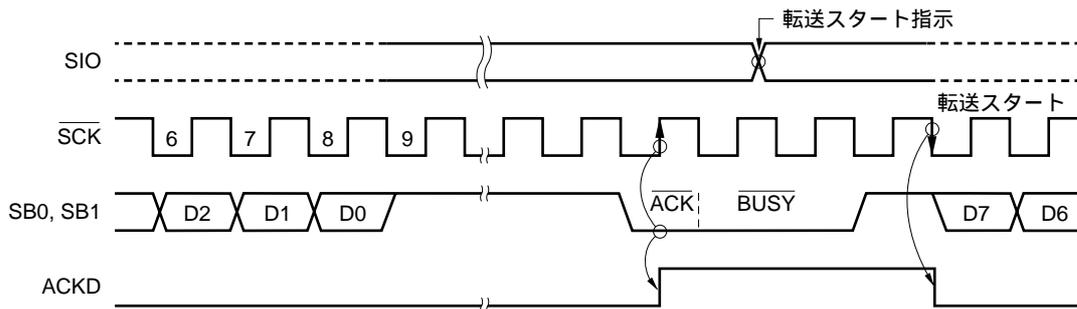


図5 - 65 BSYEの動作

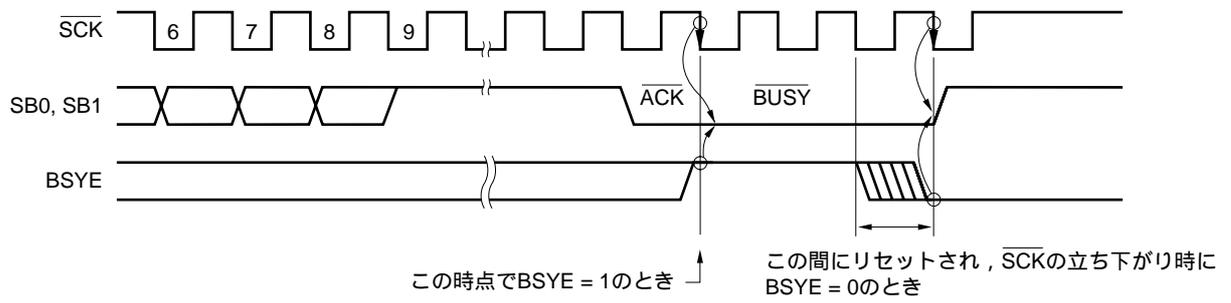


表5 - 10 SBIにおける各種の信号 (1/2)

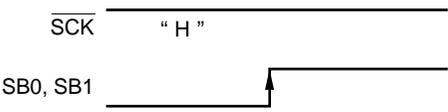
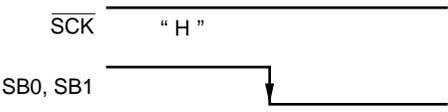
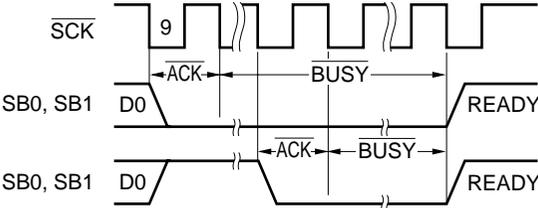
信号名称	出力するデバイス	定義	タイミング・チャート	出力される条件	フラグへの影響	信号の意味
バス・リリース信号 (REL)	マスタ	$\overline{SCK} = 1$ のときの、 SB0, SB1の立ち上がりエッジ		・ RELTのセット	・ RELDをセット ・ CMDDをクリア	続いてCMD信号を出力し、 送信データがアドレスであることを示す。
コマンド信号 (CMD)	マスタ	$\overline{SCK} = 1$ のときの、 SB0, SB1の立ち下がりエッジ		・ CMDTのセット	・ CMDDをセット	i) REL信号出力後送信データはアドレス。 ii) REL信号出力なし送信データはコマンド。
アクノリッジ信号 (ACK)	マスタ/ スレーブ	シリアル受信完了後、 \overline{SCK} の1クロックの期間SB0, SB1に出力されるロウ・レベルの信号	〔同期ビジー出力〕 	① ACKE = 1 ② ACKTのセット	・ ACKDをセット	受信完了。
ビジー信号 (BUSY)	スレーブ	〔同期ビジー信号〕 アクノリッジ信号に続いてSB0, SB1に出力されるロウ・レベルの信号。		・ BSYE = 1	-	処理中のため、シリアル受信不可能状態。
レディ信号 (READY)	スレーブ	シリアル転送開始前、完了後SB0, SB1に出力されるハイ・レベルの信号。		① BSYE = 0 ② SIOへのデータ書き込み命令実行 (転送開始指示)	-	シリアル受信可能状態。

表5 - 10 SBIにおける各種の信号 (2/2)

信号名称	出力するデバイス	定義	タイミング・チャート	出力される条件	フラグへの影響	信号の意味
シリアル・クロック ($\overline{\text{SCK}}$)	マスタ	アドレス/コマンド/データ, $\overline{\text{ACK}}$ 信号, 同期BUSY信号などの出力のための同期クロック。最初の8個でアドレス/コマンド/データを転送する。		CSIE = 1のときの, SIOへのデータ書き込み命令実行 (シリアル転送のスタート指示)注2	IRQCSIをセット (9クロック目の立ち上がり)注1	シリアル・データ・バスへの信号出力のタイミング
アドレス (A7 - A0)	マスタ	REL信号, CMD信号出力後に, $\overline{\text{SCK}}$ に同期して転送される8ビット・データ。				シリアル・バス上のスレーブ・デバイスのアドレス値
コマンド (C7 - C0)	マスタ	REL信号は出力されず, CMD信号のみ出力されたあと, $\overline{\text{SCK}}$ に同期して転送される8ビット・データ。				スレーブ・デバイスへの指示・メッセージ
データ (D7 - D0)	マスタ/ スレーブ	REL信号, CMD信号とも出力されず, $\overline{\text{SCK}}$ に同期して転送される8ビット・データ。				スレーブ, またはマスタ・デバイスが処理する数値

注1 . WUP = 0のとき, 常に9クロック目の $\overline{\text{SCK}}$ の立ち上がりでIRQCSIをセットする。

WUP = 1のとき, アドレスを受信し, そのアドレスがスレーブ・アドレス・レジスタ (SVA) の値と一致したときのみ, $\overline{\text{SCK}}$ の9クロック目の立ち上がりでIRQCSIをセットする。

2 . $\overline{\text{BUSY}}$ 状態のときは, READY状態になったあと, 転送スタートする。

(6) 端子構成

シリアル・クロック端子 ($\overline{\text{SCK}}$) と、シリアル・データ・バス端子 (SB0またはSB1) の構成は、次のようになっています。

(a) $\overline{\text{SCK}}$ シリアル・クロックを入出力するための端子

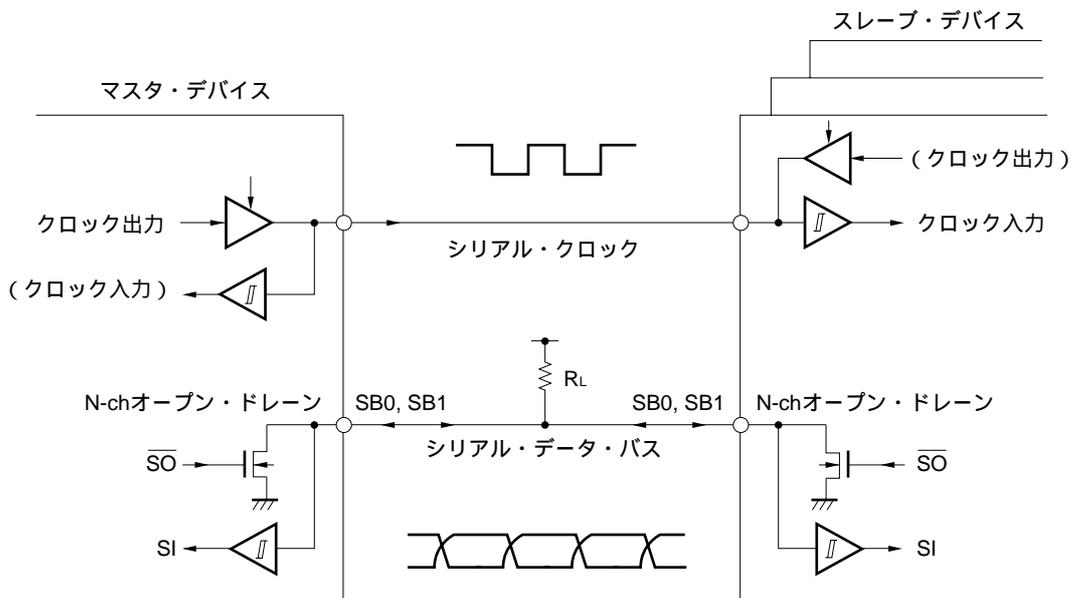
- ① マスタ CMOS, プッシュプル出力
- ② スレーブ シュミット入力

(b) SB0, SB1 シリアル・データの入出力兼用端子

マスタ, スレーブともに出力はN-chオープン・ドレイン, 入力はシュミット入力

シリアル・データ・バス・ラインは、出力がN-chオープン・ドレインのため、外部にプルアップ抵抗が必要となります。

図5 - 66 端子構成図



注意 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。転送中は常にオフさせることができます。ただし、ウエイク・アップ機能指定ビット (WUP) = 1の場合は、N-chトランジスタは常にオフとなりますので、受信前に、SIOにFFHを書き込む必要はありません。

(7) アドレスの一致検出方法

SBIモードでは、マスタのアドレス通信より、特定のスレーブ・デバイスを選択し、通信がスタートされます。

アドレス一致の検出は、ハードウェアで行います。スレーブ・アドレス・レジスタ (SVA) を備え、ウェイク・アップ状態 (WUP = 1) では、マスタから送信されたアドレスとSVAに設定された値が一致したときのみ、IRQCSIをセットします。

注意 1 . スレーブの選択、非選択状態の検出は、バス・リリース (RELD = 1の状態) のあとに受信したスレーブ・アドレスの一致検出により行います。

この一致検出は、通常WUP = 1の状態が発生するアドレスの一致割り込み (IRQCSI) を使用します。したがって、スレーブ・アドレスによる選択、非選択の検出は、WUP = 1の状態で行ってください。

2 . WUP = 0で、割り込みを使用せずに選択、非選択を検出する場合には、アドレスの一致検出による方法を使用せず、あらかじめプログラムで設定したコマンドの送受信で行ってください。

(8) エラーの検出

SBI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。

(a) 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。

(b) スレーブ・アドレス・レジスタ (SVA) を利用する方法

送信データをSIOとSVAにもセットし、送信を行います。送信完了後に、シリアル動作モード・レジスタ (CSIM) のCOIビット (アドレス・コンパレータからの一致信号) をテストし、“1” ならば正常な送信、“0” ならば送信エラーと判断します。

(9) 通信動作

SBIでは、マスタがシリアル・バス上に「アドレス」を出力することで複数のデバイスのうち、通信対象となるスレーブ・デバイスを通常一つ選択します。

通信対象デバイスを決定したのちに、マスタ・デバイスとスレーブ・デバイスの間で、コマンド、データの送受信を行い、シリアル通信を実現します。

各データ通信のタイミング・チャートを図5 - 67 ~ 5 - 70に示します。

SBIモードでは、シリアル・クロック ($\overline{\text{SCK}}$) の立ち下がりに同期してシフト・レジスタのシフト動作が行われ、送信データがSOラッチに、SB0/P02端子または、SB1/P03端子からMSBを先頭にして出力されます。また、 $\overline{\text{SCK}}$ の立ち上がりでSB0 (またはSB1) 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。

図5-67 マスタ・デバイスからスレーブ・デバイス (WUP = 1) へのアドレス送信動作

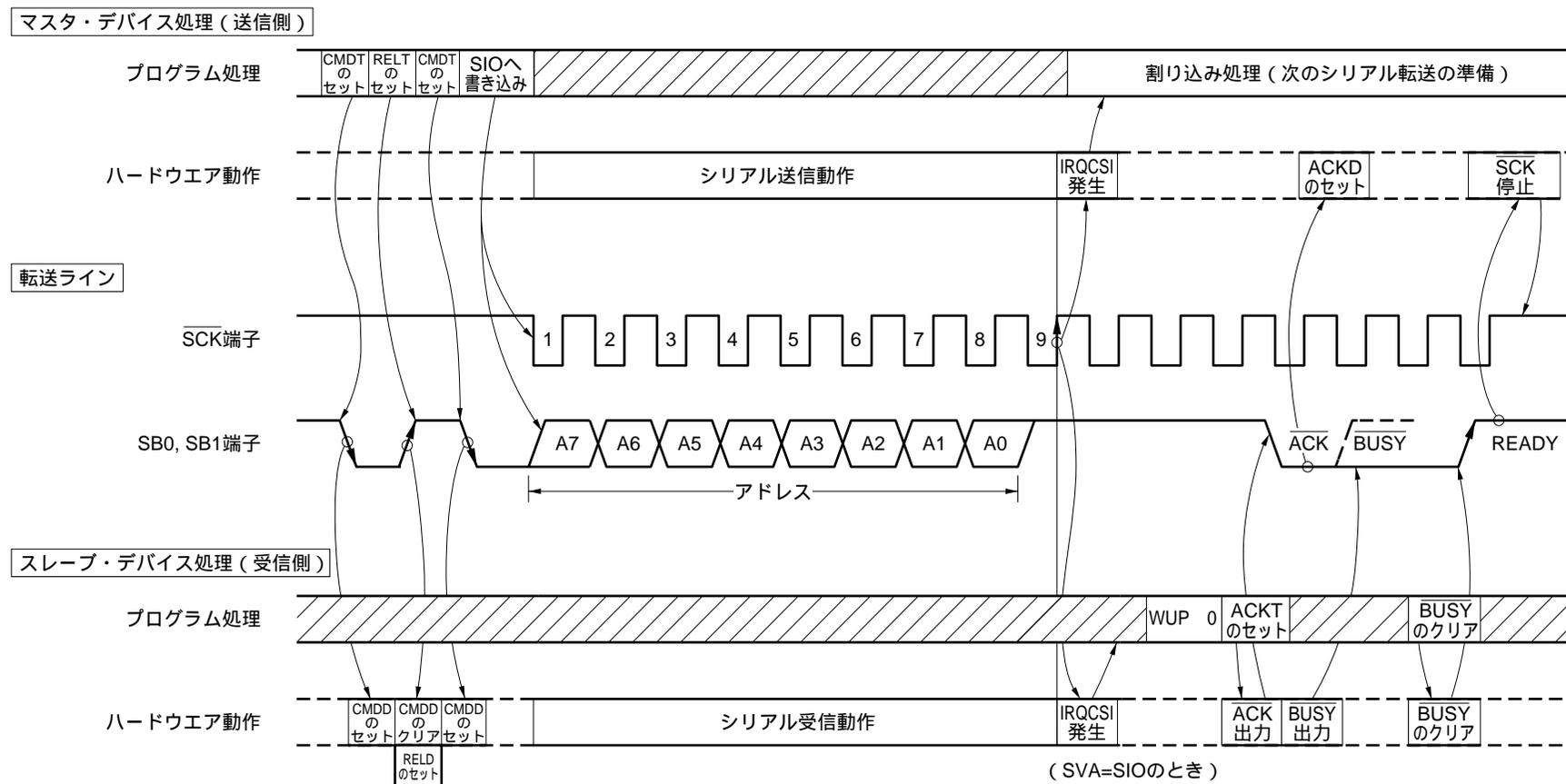


図5 - 68 マスタ・デバイスからスレーブ・デバイスへのコマンド送信動作

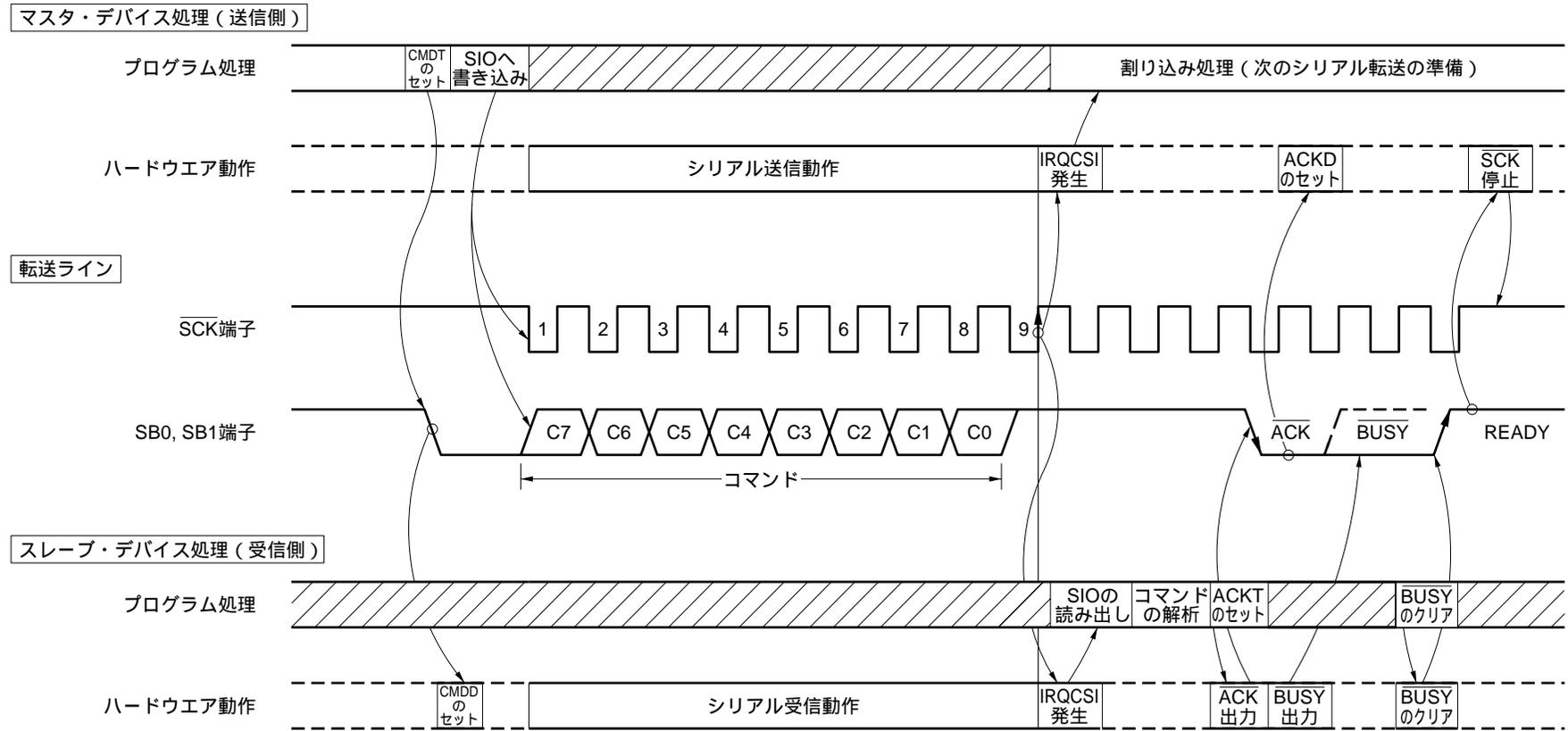


図5 - 69 マスタ・デバイスからスレーブ・デバイスへのデータ送信動作

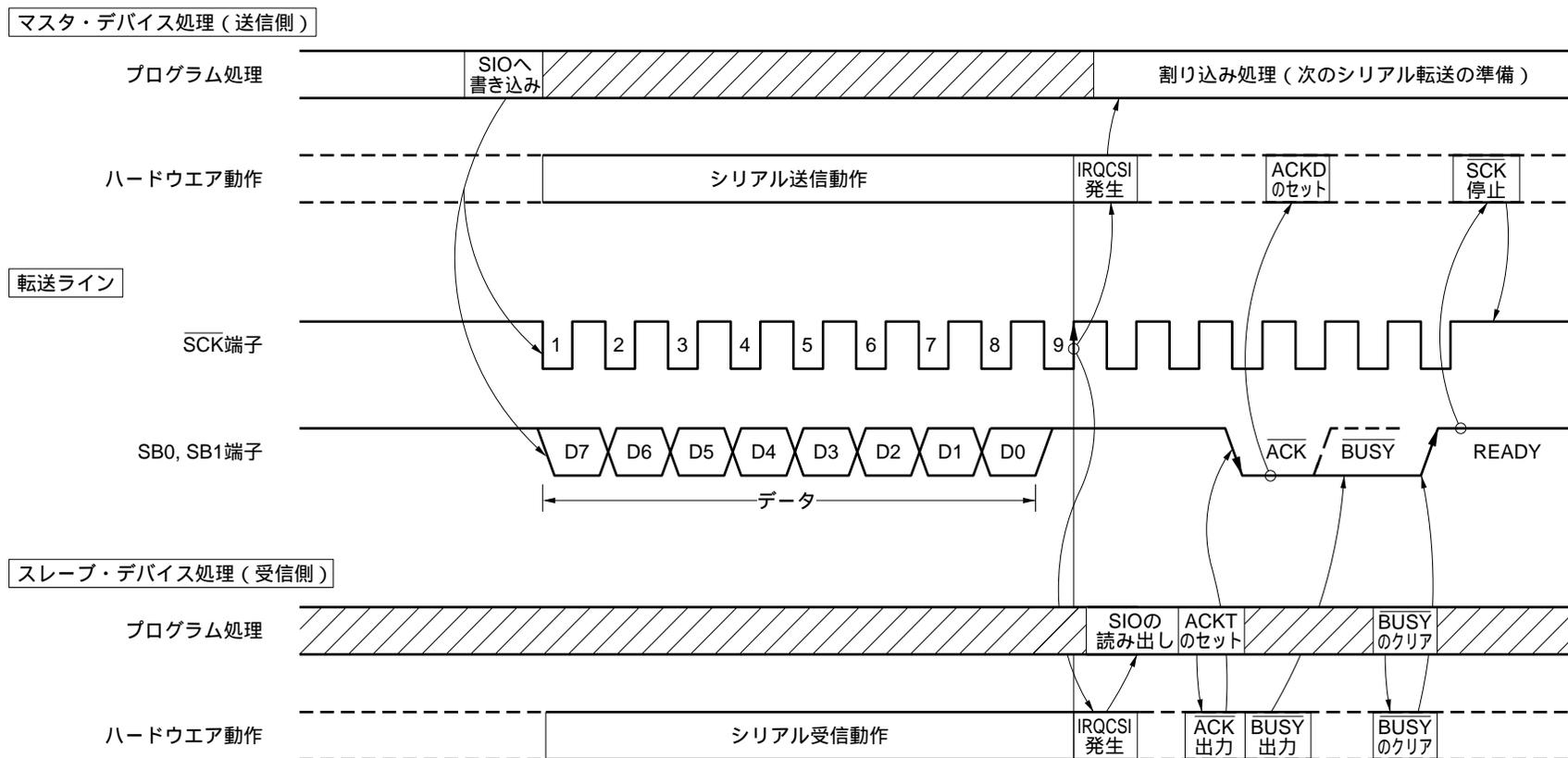
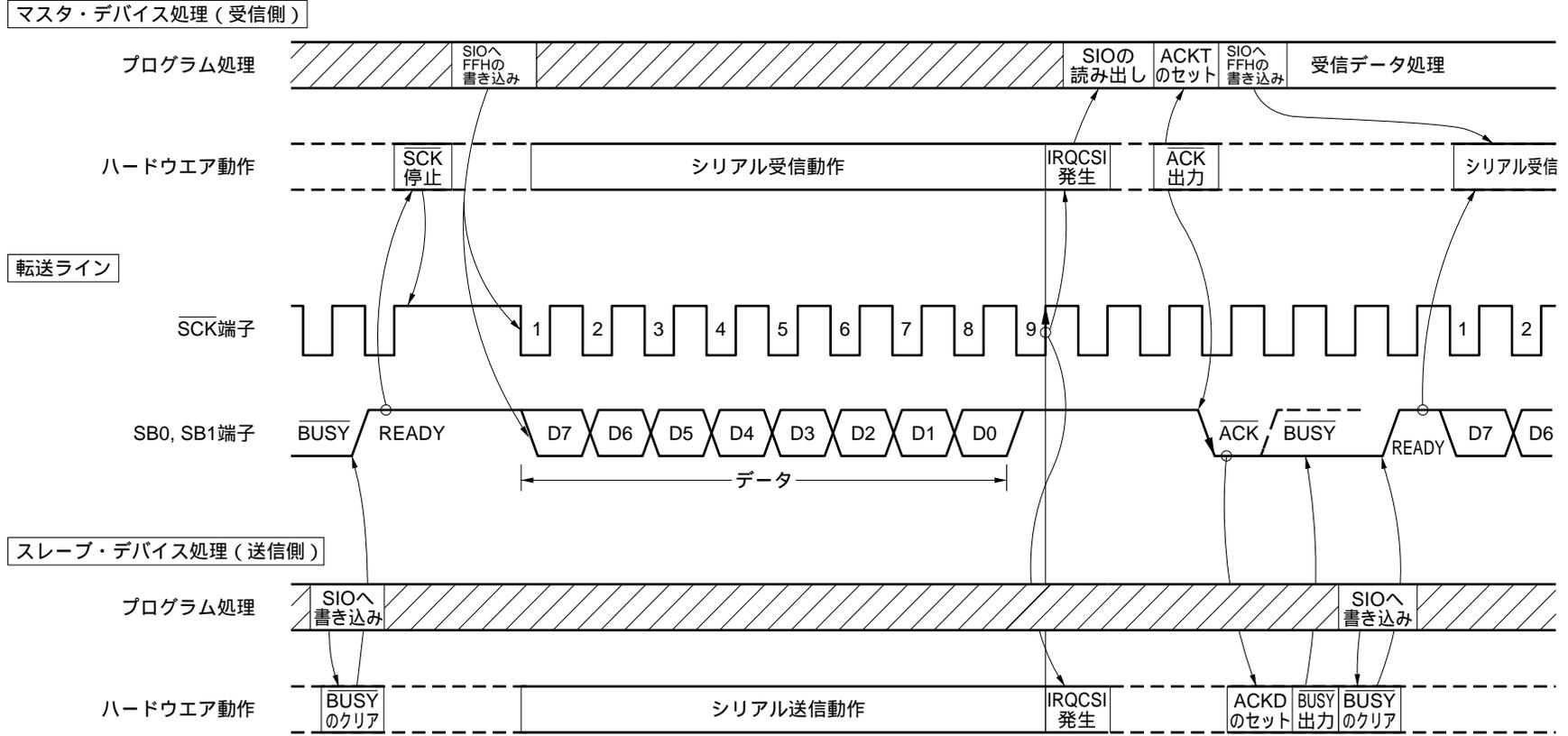


図5-70 スレーブ・デバイスからマスタ・デバイスへのデータ送信動作



(10) 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にシフト・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、もしくは $\overline{\text{SCK}}$ 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。

ただし、ウエイク・アップ機能指定ビット (WUP) = 1の場合は、N-chトランジスタは常にオフとなりますので、受信前に、SIOにFFHを書き込む必要はありません。

3 . スレーブがビジィ状態のときに、SIOにデータを書き込んだ場合、そのデータは失われません。

ビジィ状態が解除されて、SB0 (またはSB1) 入力が高レベル (レディ) 状態になったときに転送はスタートします。

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。
```

(11) SBIモードの注意事項

(a) スレーブの選択、非選択状態の検出は、バス・リリース (RELD = 1の状態) のあとに受信したスレーブ・アドレスの一致検出により行います。

この一致検出は、通常WUP = 1の状態が発生するアドレスの一致割り込み (IRQCSI) を使用します。したがって、スレーブ・アドレスによる選択、非選択の検出は、WUP = 1の状態で行ってください。

(b) $WUP = 0$ で、割り込みを使用せずに選択、非選択を検出する場合には、アドレスの一致検出による方法を使用せず、あらかじめプログラムで設定したコマンドの送受信で行ってください。

(c) \overline{BUSY} 信号出力中に $WUP = 1$ とすると、 \overline{BUSY} が解除されません。SBIでは、 \overline{BUSY} の解除指示後、次のシリアル・クロック (\overline{SCK}) の立ち下がりまで \overline{BUSY} 信号が出力されます。 $WUP = 1$ とするときは、必ず \overline{BUSY} を解除したのちに、SB0 (またはSB1) 端子がハイ・レベルになったことを確認してから、 $WUP = 1$ としてください。

(12) SBIモードの応用

この項では、SBIモードでシリアル・データ通信を行う応用例を紹介します。この応用例では、 μ PD750008をシリアル・バス上のマスタCPU、スレーブCPUのどちらでも動作できるようになっています。

また、コマンドにより、マスタの変更もできるようになっています。

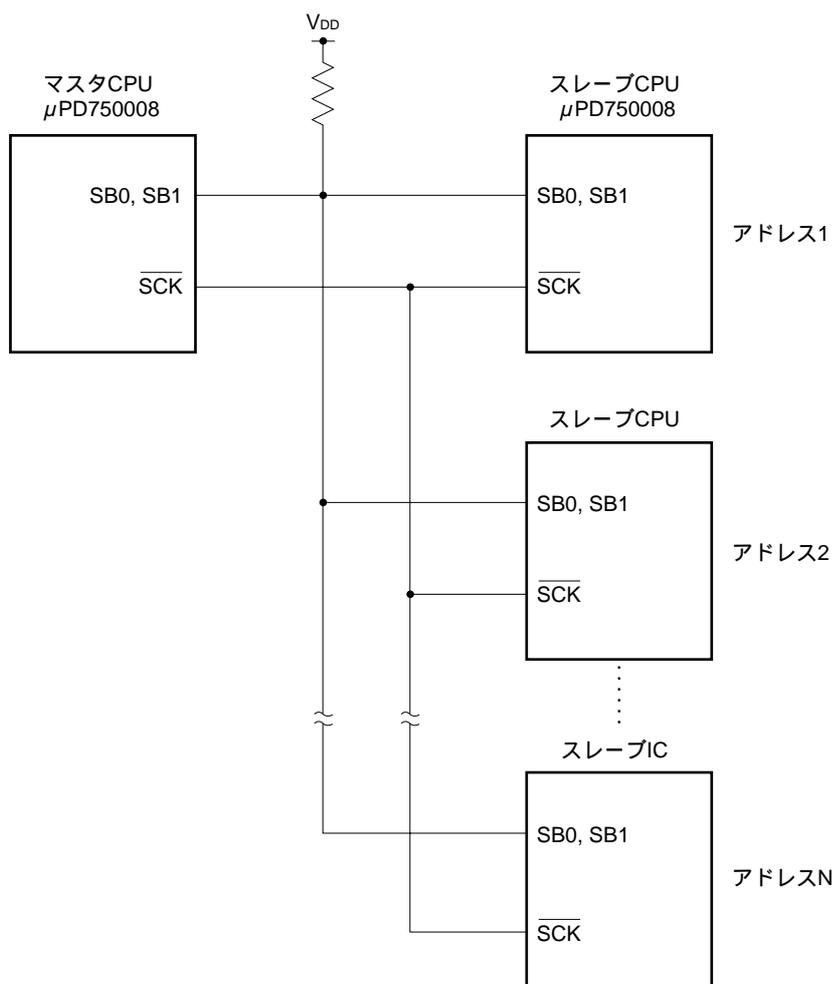
(a) シリアル・バス構成

この項に記載されている応用例でのシリアル・バスの構成は、 μ PD750008がシリアル・バス内のデバイスの1つとしてバス・ラインに接続されているものとします。

μ PD750008の使用する端子は、シリアル・データ・バスSB0（またはSB1）とシリアル・クロック $\overline{\text{SCK}}$ （P01）の2本です。

図5 - 71にシリアル・バスの構成例を示します。

図5 - 71 シリアル・バス構成例



(b) コマンド説明**(i) コマンドの種類**

この応用例では次に示すコマンドを設定します。

- ① READコマンド : スレーブからマスタへのデータ転送を行います。
- ② WRITEコマンド : マスタからスレーブへのデータ転送を行います。
- ③ ENDコマンド : WRITEコマンドの完了をスレーブに示します。
- ④ STOPコマンド : WRITEコマンドの中断をスレーブに示します。
- ⑤ STATUSコマンド : スレーブ側の状態を読み出します。
- ⑥ RESETコマンド : 現在選択されているスレーブを非選択にします。
- ⑦ CHGMSTコマンド : スレーブ側にマスタの権利を受け渡します。

(ii) 通信手順

マスタ、スレーブ間の通信手順は次のように行います。

- ① マスタが通信しようとするスレーブのアドレスを送信して、スレーブの選択を行う（チップ・セレクト）ことにより通信を開始します。

アドレスを受信したスレーブは、 $\overline{\text{ACK}}$ を返してマスタとの通信を行います（非選択状態より選択状態になります）。

- ② ①の処理で選択されたスレーブとマスタとの間で、コマンドおよびデータの転送より通信を行います。

ただし、コマンド、データの転送はマスタ、スレーブの1対1の転送なので、ほかのスレーブは非選択状態である必要があります。

- ③ 通信はスレーブが非選択状態になることにより終了します。スレーブが非選択状態になるのは次の場合です。

- ・RESETコマンドをマスタより送信すると、選択されていたスレーブは非選択状態になります。
- ・CHGMSTコマンドによりマスタに変更した場合、マスタからスレーブになったデバイスは非選択状態になります。

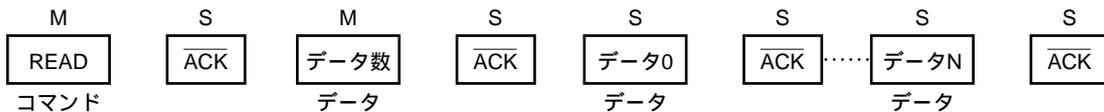
(iii) コマンドのフォーマット

各コマンドの転送フォーマットを以下に示します。

① READコマンド

スレーブからの読み出しを行うコマンドです。読み出すデータ数は1-256バイトの可変とし、マスタからパラメータとしてデータ数を指定します。データ数として00Hが指定された場合は、256バイトのデータ転送が指定されたことになります。

図5 - 72 READコマンドの転送フォーマット



備考 M : マスタが出力
S : スレーブが出力

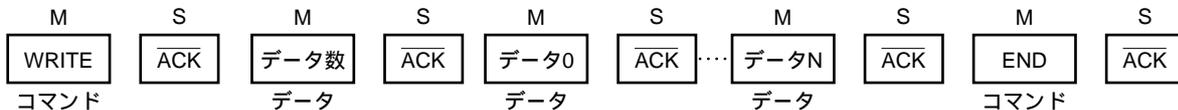
スレーブはデータ数を受信後、送信可能なデータがそのデータ数以上あれば、 $\overline{\text{ACK}}$ を返します。データが不足すれば、 $\overline{\text{ACK}}$ を返さずにエラーとなります。

マスタはデータ受信時には1バイト受信することに $\overline{\text{ACK}}$ をスレーブに送ります。

② WRITEコマンド, ENDコマンド, STOPコマンド

スレーブへのデータ書き込みを行うコマンドです。書き込みデータ数は、1-256バイトの可変とし、マスタがパラメータとしてデータ数を指定します。データ数として00Hが指定された場合は、256バイトのデータ転送が指定されます。

図5 - 73 WRITE, ENDコマンドの転送フォーマット



備考 M : マスタが出力
S : スレーブが出力

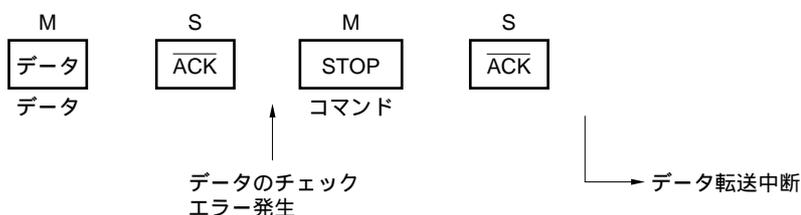
スレーブはデータ数を受信後、受信データを格納するエリアがそのデータ数以上あれば、 $\overline{\text{ACK}}$ を返します。格納するエリアが不足すれば、 $\overline{\text{ACK}}$ を返さずエラーとなります。

マスタは、すべてのデータの転送が終了するとENDコマンドを送信します。ENDコマンドにより、すべてのデータが正しく転送されたことをスレーブに知らせます。

また、スレーブは、すべてのデータ受信前でもENDコマンドを受けます。この場合ENDコマンドを受信する直前のデータまでを有効にします。

マスタはデータ送信時には、送信前と送信後のSIOの内容を比較することにより、データが正しくバス上に出力されたことを確かめます。送信前と送信後のSIOの内容が変化していたならば、STOPコマンドを送信してデータ転送を中断します。

図5 - 74 STOPコマンドの転送フォーマット



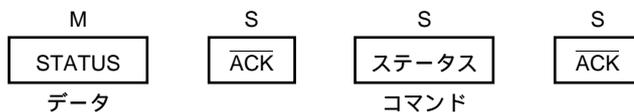
備考 M : マスタが出力
S : スレーブが出力

スレーブはSTOPコマンドを受信すると、その直前に受信した1バイトのデータを無効とします。

③ STATUSコマンド

現在選択されているスレーブ状態（ステータス）を読み出すためのコマンドです。

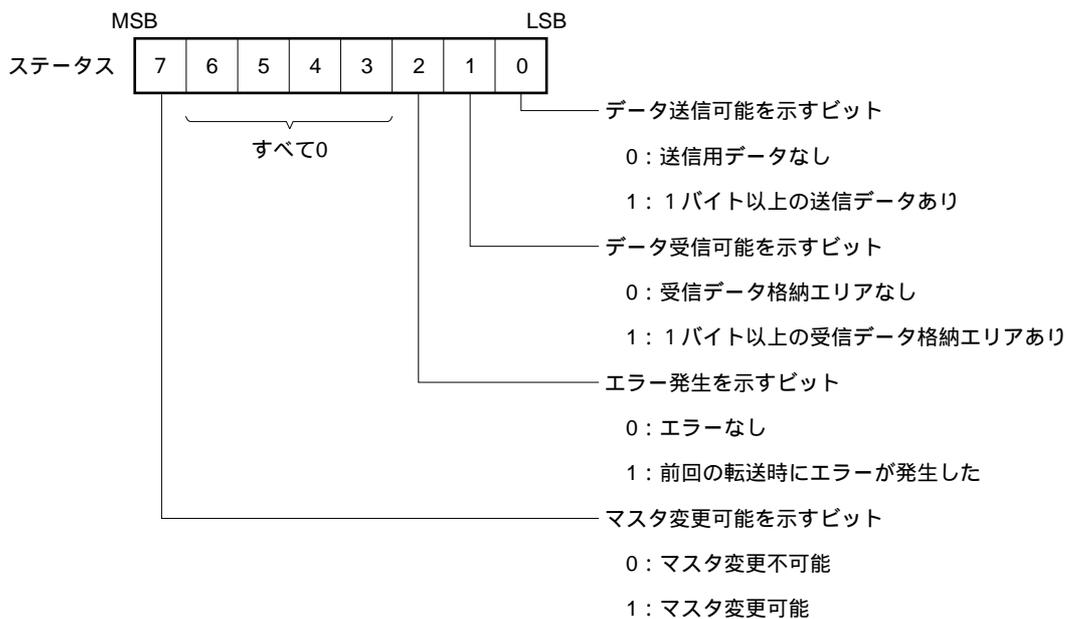
図 5 - 75 STATUSコマンドの転送フォーマット



備考 M : マスタが出力
S : スレーブが出力

スレーブが返すステータスのフォーマットは次のようになります。

図 5 - 76 STATUSコマンドのステータス・フォーマット

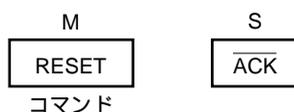


マスタはステータスのデータを受信すると、 $\overline{\text{ACK}}$ をスレーブへ返します。

④ RESETコマンド

現在選択されているスレーブを非選択状態にするためのコマンドです。RESETコマンドを送信することにより、すべてのスレーブを非選択状態とすることができます。

図5 - 77 RESETコマンドの転送フォーマット

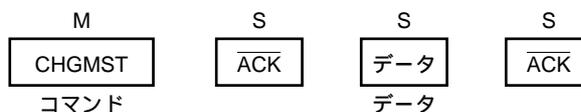


備考 M : マスタが出力
S : スレーブが出力

⑤ CHGMSTコマンド

現在選択されているスレーブにマスタの権利を渡すコマンドです。

図5 - 78 CHGMSTコマンドの転送フォーマット



備考 M : マスタが出力
S : スレーブが出力

スレーブはCHGMSTコマンドを受信すると、マスタの権利を受け取れるかを判断し、マスタにデータを返します。データは、以下のとおりです。

- ・ 0FFH : マスタ変更可能
- ・ 00H : マスタ変更不可能

スレーブはデータの転送時に、転送前と転送後のSIOの内容を比較し、一致しなければACKを返さずエラーとなります。

マスタはデータを受信するとACKをスレーブに返し、受信データが0FFHであれば、以後スレーブとして動作します。スレーブは0FFHを送信し、マスタからACKが返ってきたあとにマスタとして動作します。

(iv) エラーの発生

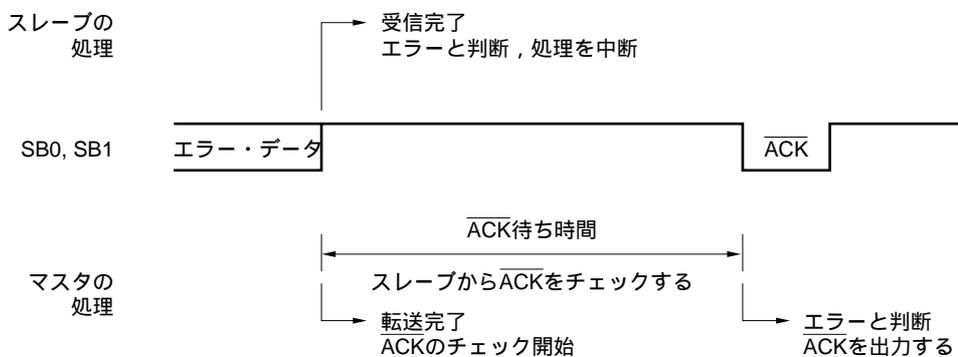
通信に対してのエラーの発生時の動作について説明します。

スレーブは、マスタに対しての $\overline{\text{ACK}}$ を返さないことにより、エラーの発生を知らせます。スレーブが受信をする場合のみエラーが発生すると、ステータスのエラー発生を示すビットをセットし、実行中のコマンドの処理はすべてキャンセルします。

マスタは、1バイトの送信の完了後に、スレーブから $\overline{\text{ACK}}$ をチェックします。

送信の完了後、一定時間の間にスレーブから $\overline{\text{ACK}}$ が返されなければ、エラーが発生したものと判断し、マスタが $\overline{\text{ACK}}$ 信号(ダミーとして)を出力します。

図5 - 79 エラー発生時のマスタ, スレーブの動作



エラーには次に示すものがあります。

・スレーブ側で発生するエラー

- ① コマンドの転送フォーマットが違っている場合
 - ② 未定義なコマンドを受信した場合
 - ③ READコマンド時に、転送するデータ数が不足する場合
 - ④ WRITEコマンド時に、データを格納するエリアが不足する場合
 - ⑤ READ, STATUS, CHGMSTコマンドのデータ送信時に、データが変化する場合
- 以上、①-⑤が発生すると $\overline{\text{ACK}}$ を返しません。

・マスタ側で発生するエラー

WRITEコマンドのデータ送信時にデータが変化した場合、STOPコマンドをスレーブに送信します。

5.6.8 $\overline{\text{SCK}}$ 端子出力の操作

$\overline{\text{SCK}}$ /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、 $\overline{\text{SCK}}$ 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/P02, SI/SB1/P03端子の制御はSBICのRELT, CMDTビットによって行います）。

次に、 $\overline{\text{SCK}}$ /P01端子出力の操作方法を示します。

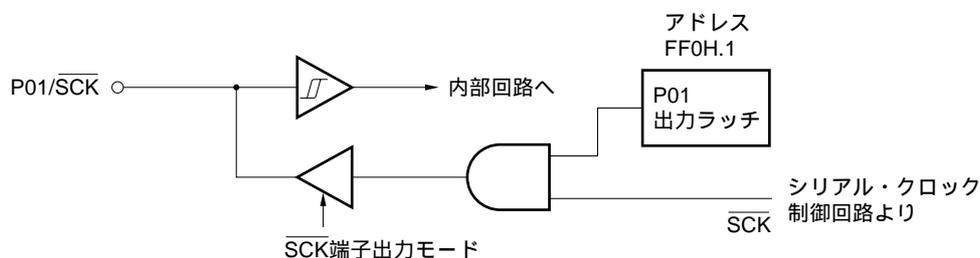
- ① シリアル動作モード・レジスタ（CSIM）を設定します（ $\overline{\text{SCK}}$ 端子：出力モード）。シリアル転送停止中では、シリアル・クロック制御回路からの $\overline{\text{SCK}}$ が1となっています。
- ② P01出力ラッチを、ビット操作命令により操作します。

例 ソフトウェアにより $\overline{\text{SCK}}$ /P01端子に1クロック出力する。

```

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

図5 - 80 $\overline{\text{SCK}}$ /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。 $\overline{\text{RESET}}$ 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。

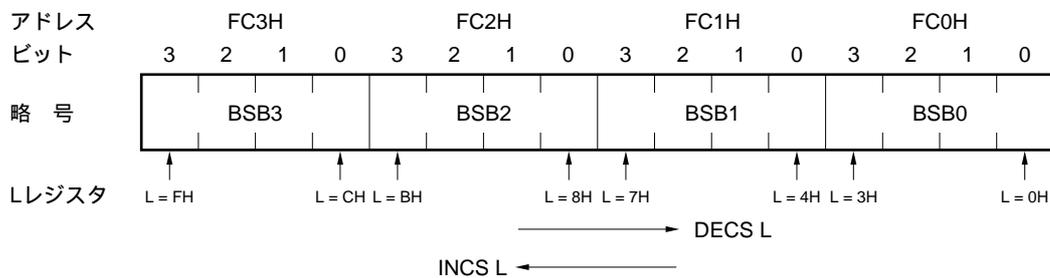
CLR1	PORT0.1	} 使用不可
SET1	PORT0.1	
CLR1	OFF0H.1	} 使用可
SET1	OFF0H.1	

5.7 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アドレスが可能で、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 - 81 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

μPD750008には7つのベクタ割り込みソースと、2つのテスト入力があり多彩な応用が可能です。

また、μPD750008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 x x x) 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 x x x) 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)

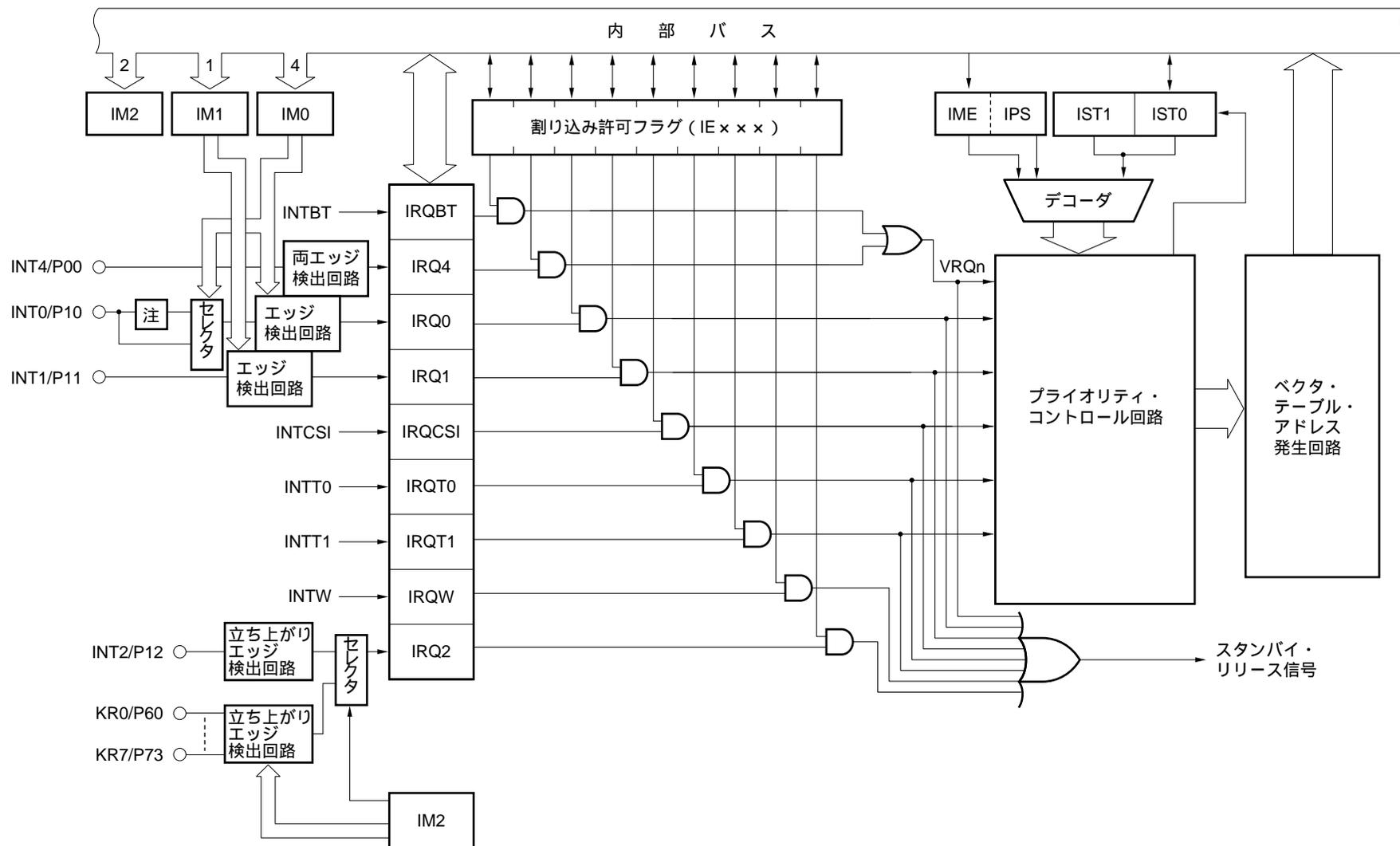
(2) テスト機能

- (a) ソフトウェアでテスト要求フラグ (IRQ x x x) 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

割り込みソースの種類と割り込みベクタ・テーブルは、表6-1、図6-2のようになっています。

表6-1 割り込みソースの種類

割り込みソース	内/外	割り込み順位 ^注	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がりエッジ検出の両方 が有効)	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・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

アドレス	MBE	RBE	割り込みベクタ	ビット幅
0000H			内部リセット・スタート・アドレス	(上位6ビット)
	"			(下位8ビット)
0002H			INTBT/INT4スタート・アドレス	(上位6ビット)
	"			(下位8ビット)
0004H			INT0スタート・アドレス	(上位6ビット)
	"			(下位8ビット)
0006H			INT1スタート・アドレス	(上位6ビット)
	"			(下位8ビット)
0008H			INTCSIスタート・アドレス	(上位6ビット)
	"			(下位8ビット)
000AH			INTT0スタート・アドレス	(上位6ビット)
	"			(下位8ビット)
000CH			INTT1スタート・アドレス	(上位6ビット)
	"			(下位8ビット)

表6 - 1の中で、割り込み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) によって行います。

例 INTBT/INT4のベクタ・テーブルの設定。

<u>VENT1</u>	<u>MBE = 0, RBE = 0,</u>	<u>GOTOBT</u>
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・RBEの設定値	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) で指定されるベクタ・テーブル・アドレスは2n番地となります。

例 INTBT/INT4とINTT0のベクタ・テーブルの設定。

VENT1	MBE = 0, RBE = 0, GOTOBT
VENT5	MBE = 0, RBE = 1, GOTOTO

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 x x x) 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)	(IRQCSI)
INT4割り込み要求フラグ (IRQ4)	タイマ/イベント・カウンタ割り込み要求フラグ (IRQT0)
BT割り込み要求フラグ (IRQBT)	(IRQT0)
	タイマ・カウンタ割り込み要求フラグ (IRQT1)

割り込み要求フラグは、割り込み要求発生でセット (1) され、割り込み処理が実行されると自動的にクリア (0) 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 x x x) 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・カウンタ割り込み許可フラグ (IET0)
INT4割り込み許可フラグ (IE4)	タイマ・カウンタ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) 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 x x x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・タイマ/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・カウンタからの一致信号でセット。	IET0
IRQT1	タイマ・カウンタ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否かを指定する、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。
```

$\overline{\text{RESET}}$ 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、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[※]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6-4(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。

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 : $2t_{CY}$
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。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。

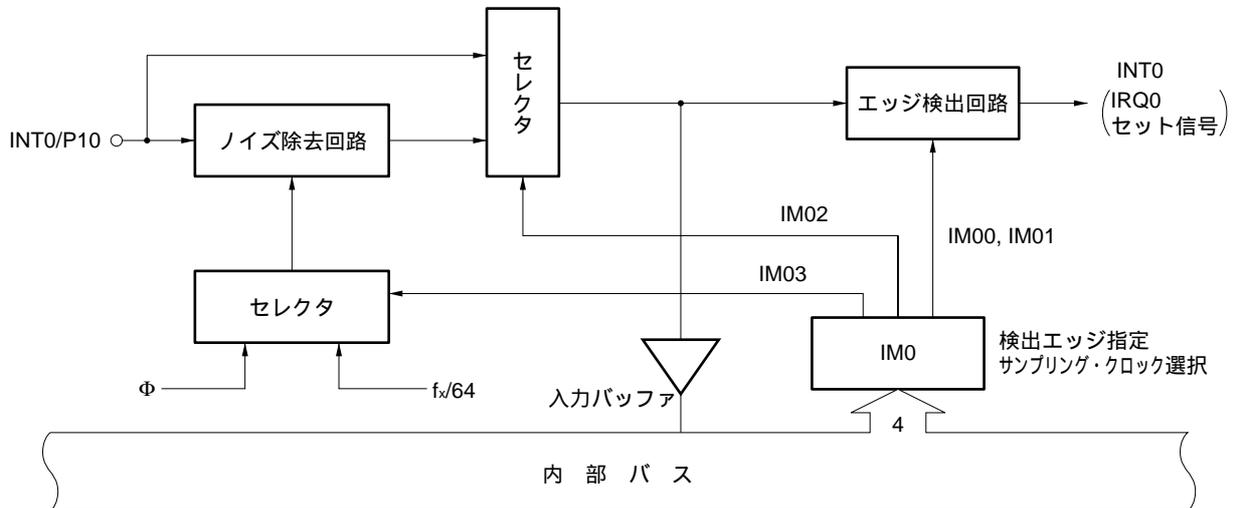
検出エッジの選択は、INT1エッジ検出モード・レジスタ(IM1)によって行います。

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。

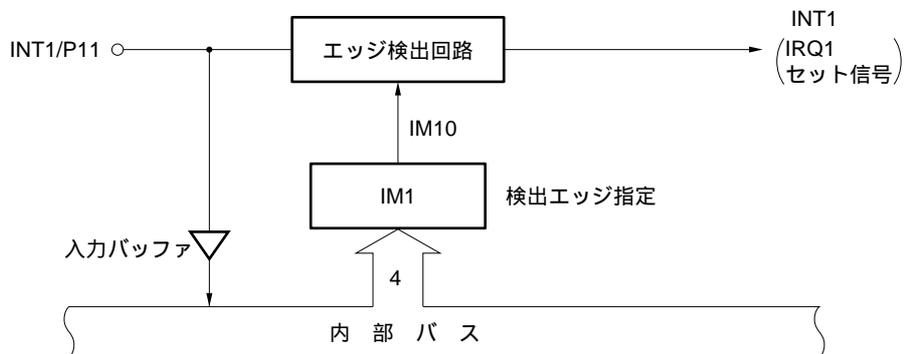
(c) INT4の構成を図6-4(c)に示します。立ち上がり、立ち下がりの両エッジを検出できる外部割り込み入力となっています。

図6-4 INT0, INT1, INT4の構成

(a) INT0のハードウェア



(b) INT1のハードウェア



(c) INT4のハードウェア

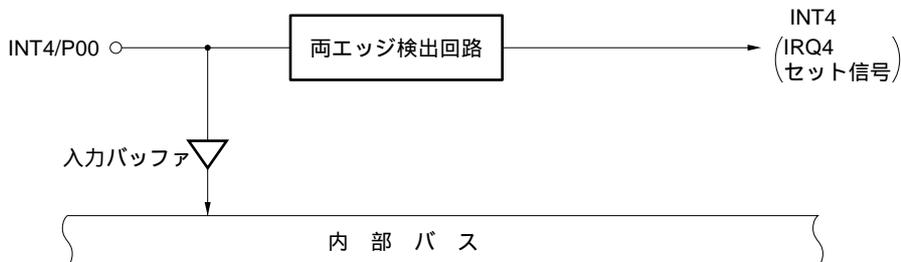
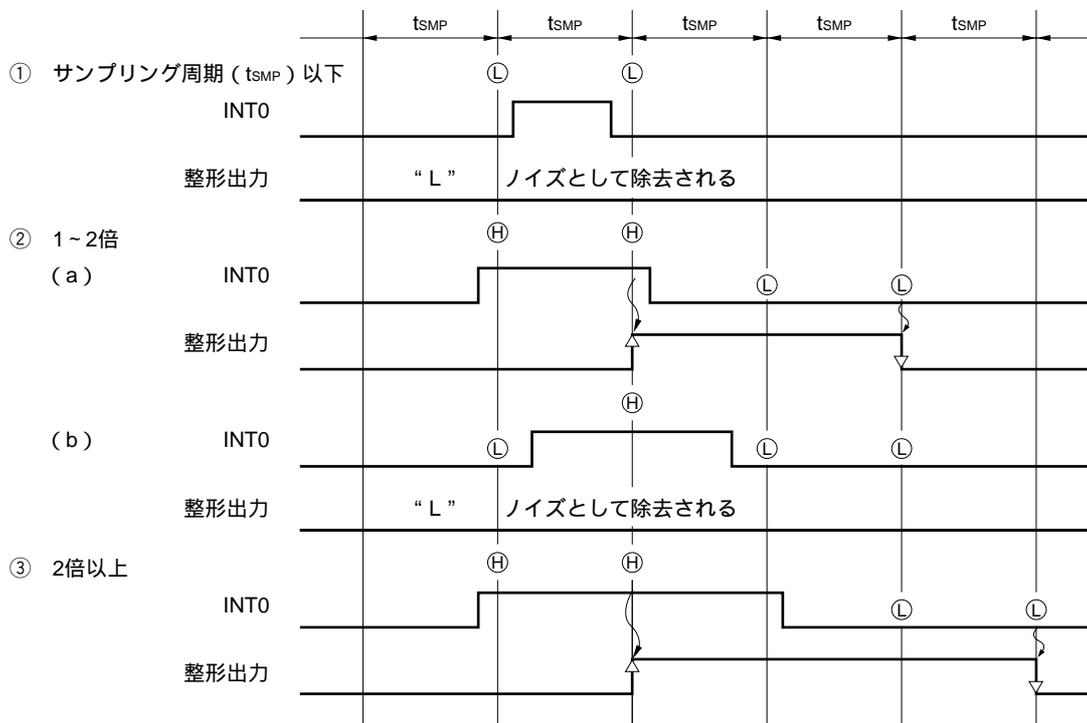


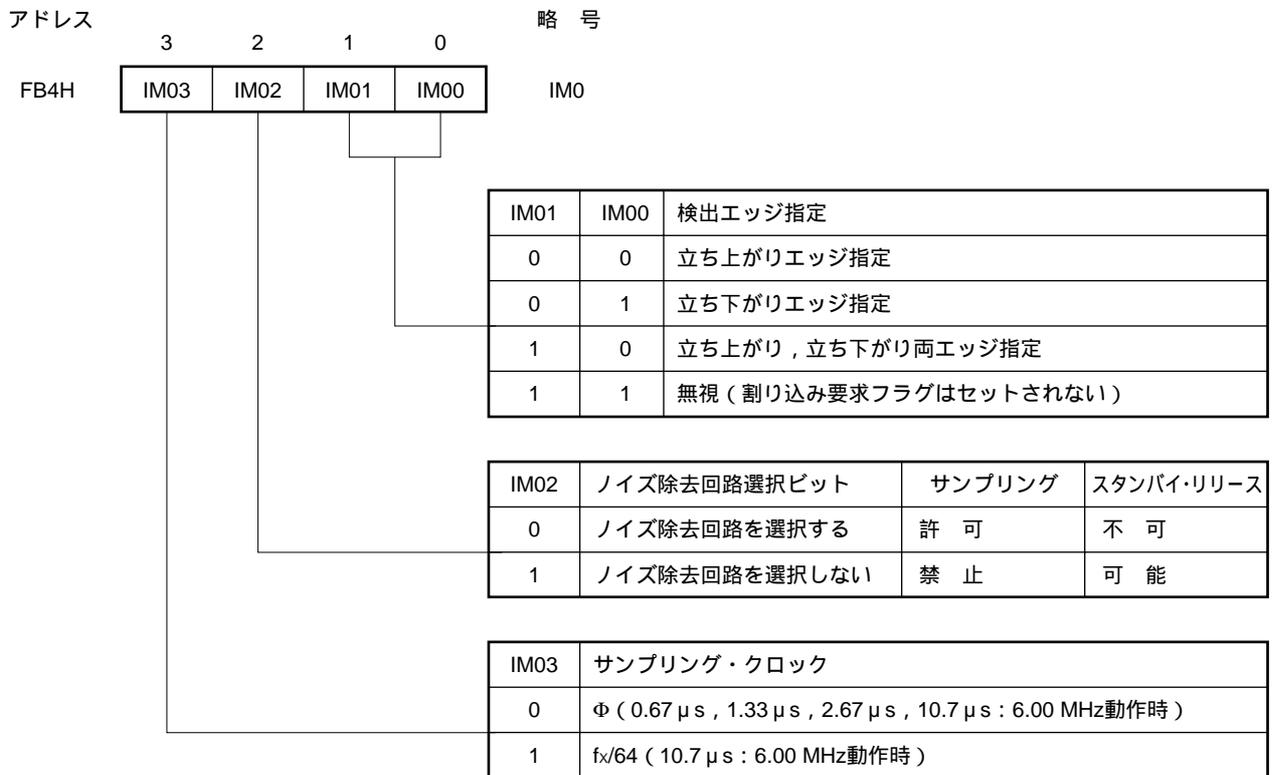
図6-5 ノイズ除去回路の入出力タイミング



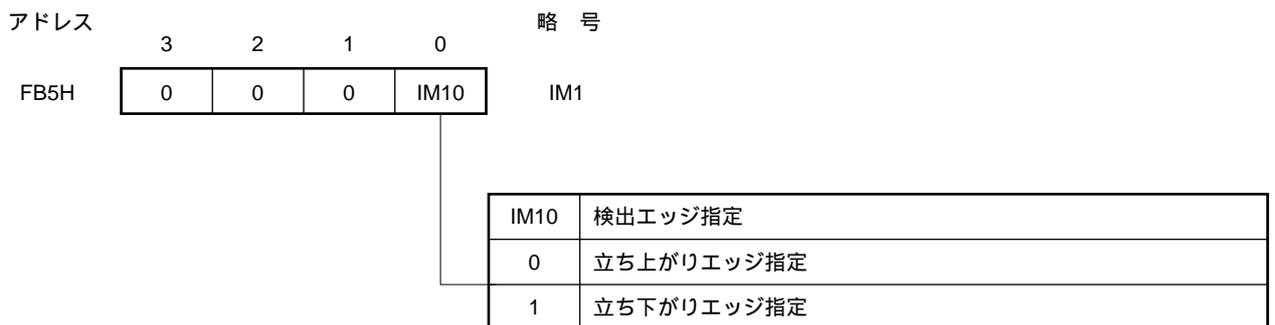
備考 $t_{SMP} = t_{CY}$ または $64/f_x$

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命令をして割り込み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ほかの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。

$\overline{\text{RESET}}$ 入力によりフラグの内容はクリア (0) されます。

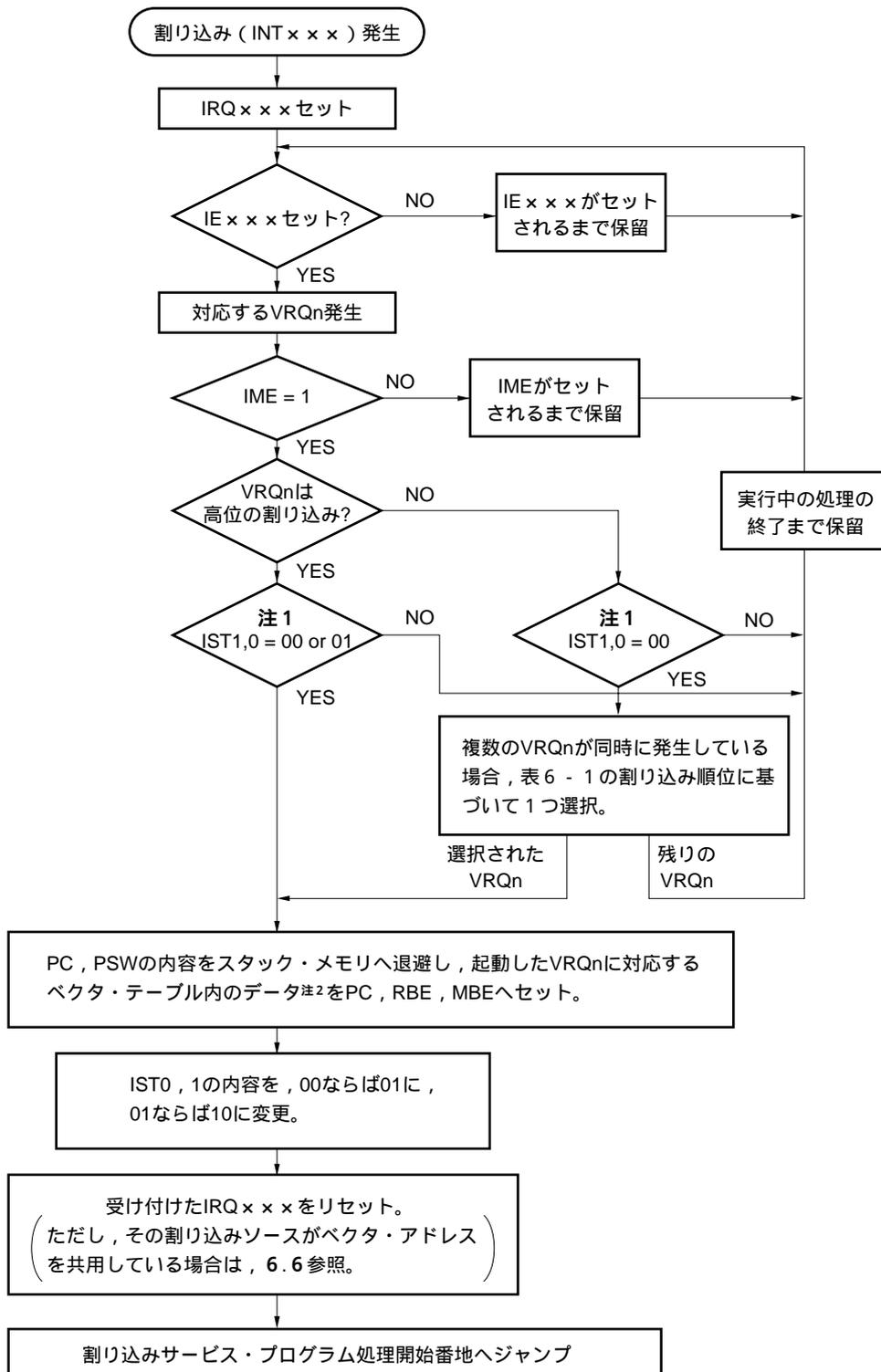
表6-3 IST1, IST0と割り込み処理状態

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

μPD750008は、次の2つの方法により多重割り込みができます。

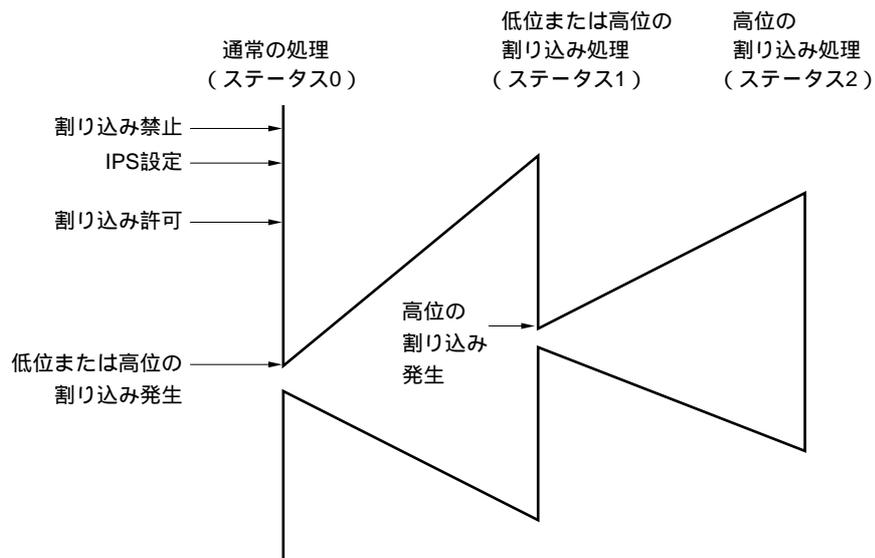
(1) 高位の割り込みを指定する多重割り込み

この方法は、μPD750008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



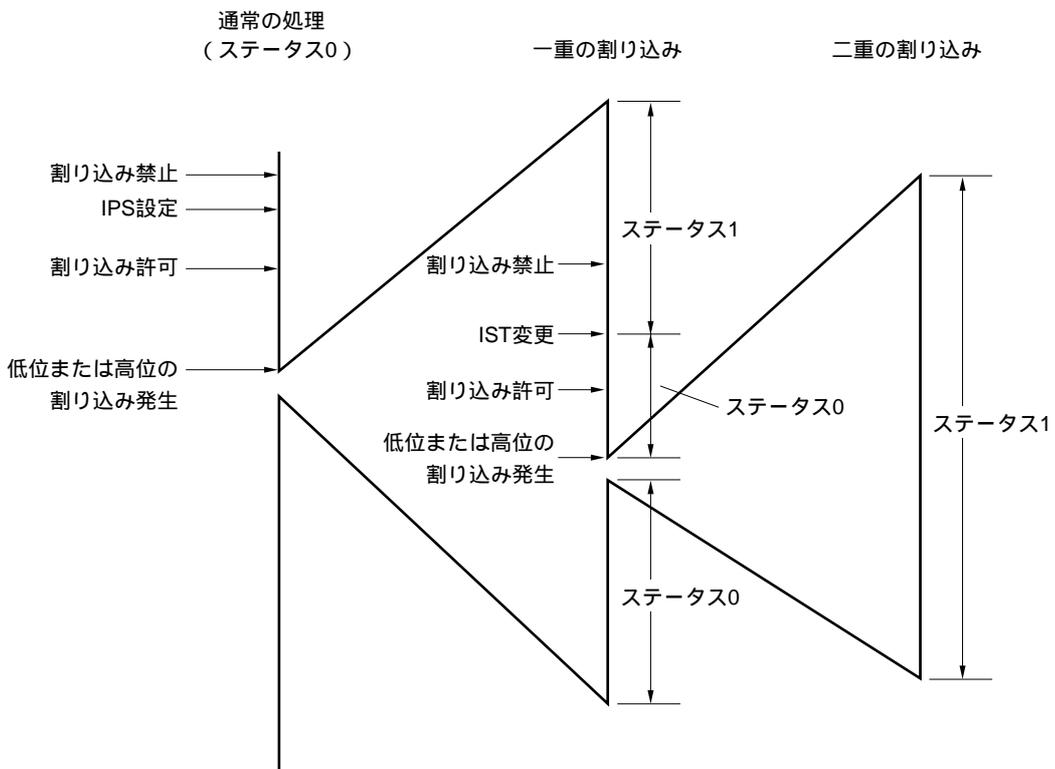
(2) 割り込みステータス・フラグを変更する多重割り込み

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、以下に示すようにして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。

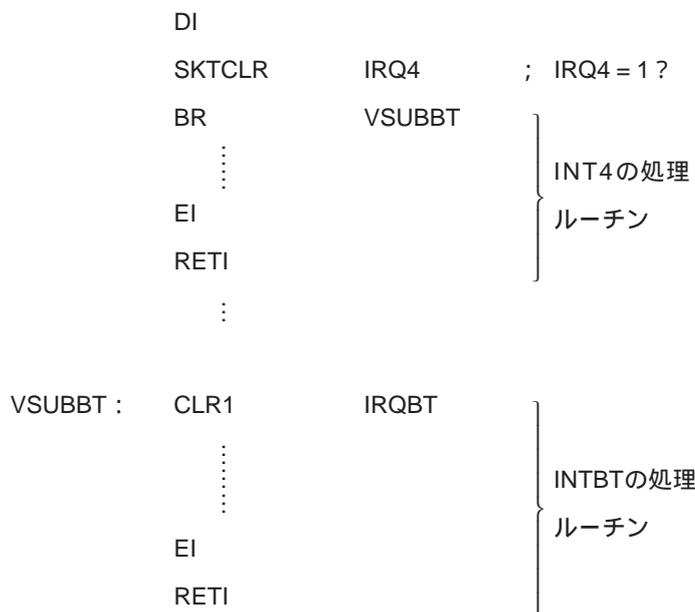
この割り込み要求フラグのテスト&クリアを行ったとき、両方の要求フラグがセットされていると、一方の要求フラグがクリアされても割り込み要求は残ります。この割り込みが「高位の割り込み」として選択されていた場合には、残った割り込み要求により二重割り込み処理に入ります。

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。

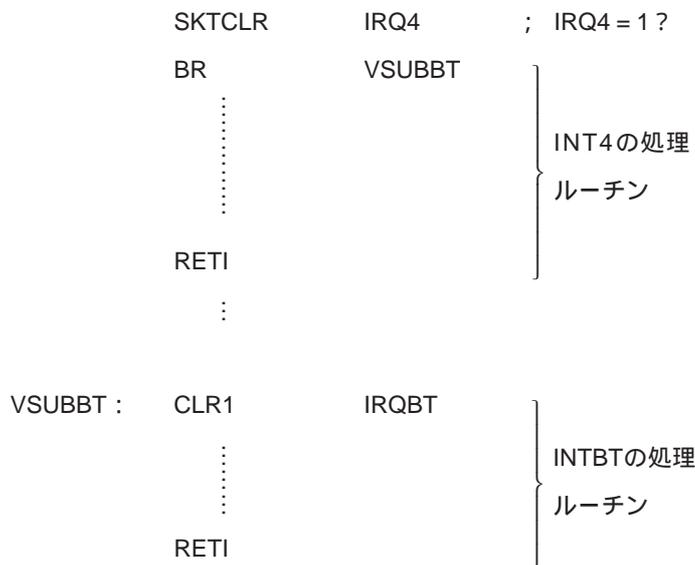
表6-4 共用割り込みの判別

「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。



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。

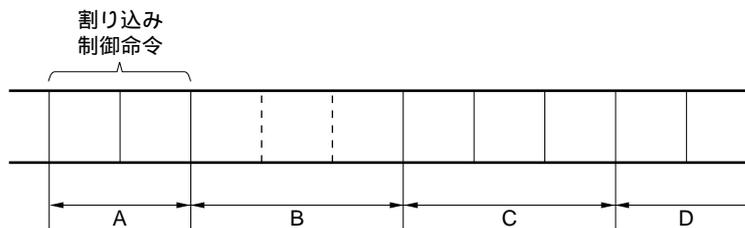


6.7 割り込み処理までのマシン・サイクル

μPD750008では、割り込みリクエスト・フラグ（IRQn）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IRQnがセットされた場合

割り込み制御命令実行中にIRQn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 : IRQn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

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、割り込みを受け付け時に操作されるスタックの操作時間などです。

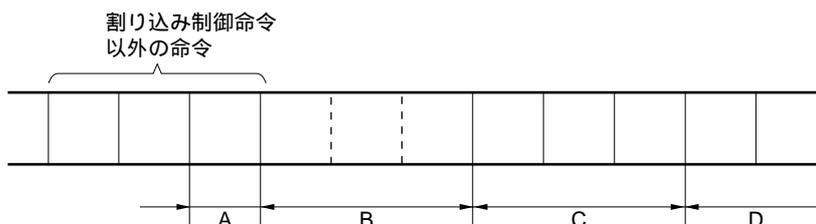
注意1．割り込み制御命令が連続している場合には、連続した制御命令を最後まで実行します。最後に実行した割り込み制御命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。

2．IRQn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IRQn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せん。

(2) (1) 以外の命令実行中にIRQn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IRQn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。

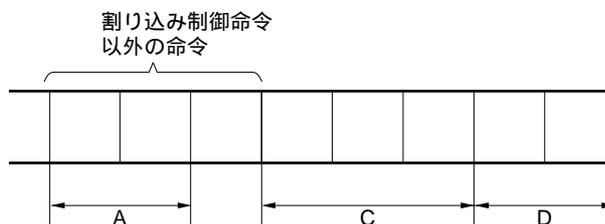


- A : IRQnセット
- B : 続く1命令を実行 (1-3マシン・サイクル, 命令により異なる)
- C : 割り込み処理 (3マシン・サイクル)
- D : 割り込みルーチンの実行

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IRQn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IRQnの割り込み要求は保留されます。

(b) 実行中の命令の最終マシン・サイクル以前でIRQn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では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

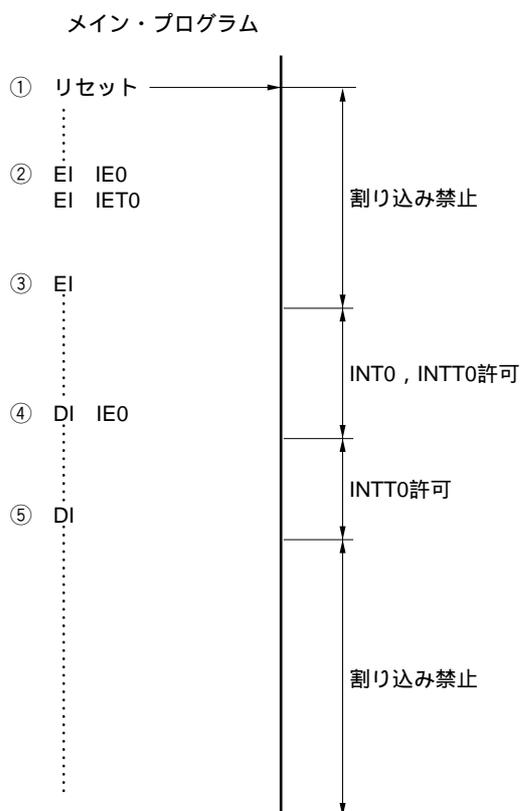
割り込み機能を使用する場合はまず、メイン・プログラムで次のように設定します。

- (a) 使用する割り込み許可フラグをセット (EI IE x x x 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 (IME) 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。

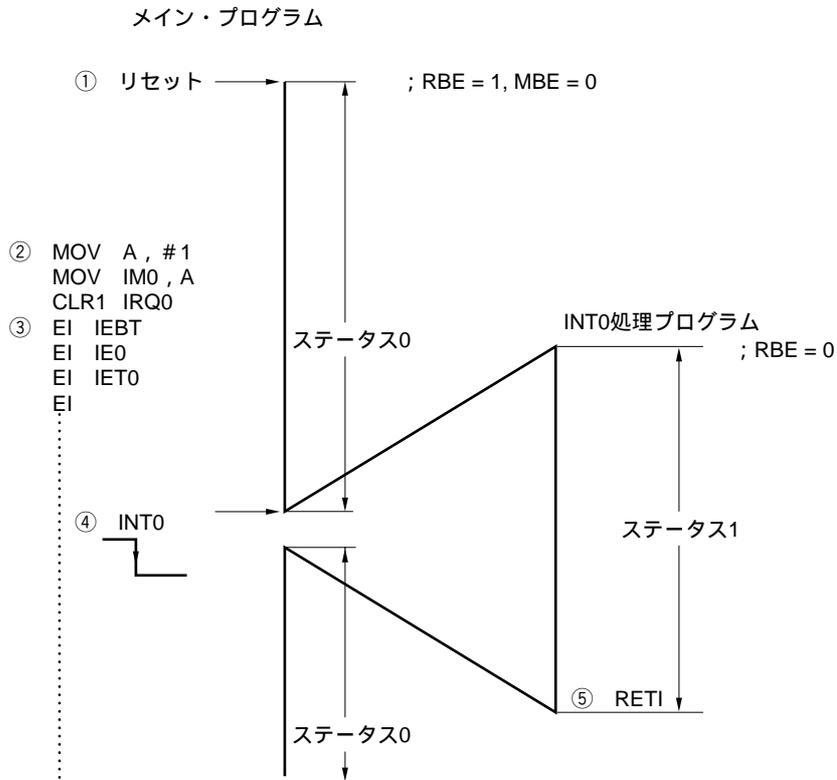
割り込み処理プログラムからの復帰はRETI命令を使用します。

(1) 割り込み許可・禁止



- ① $\overline{\text{RESET}}$ 信号により、すべての割り込みは禁止。
- ② EI IE × × 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。
- ③ 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, INTT0が許可される。
- ④ DI IE × × 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。
- ⑤ DI命令によりすべての割り込みが禁止。

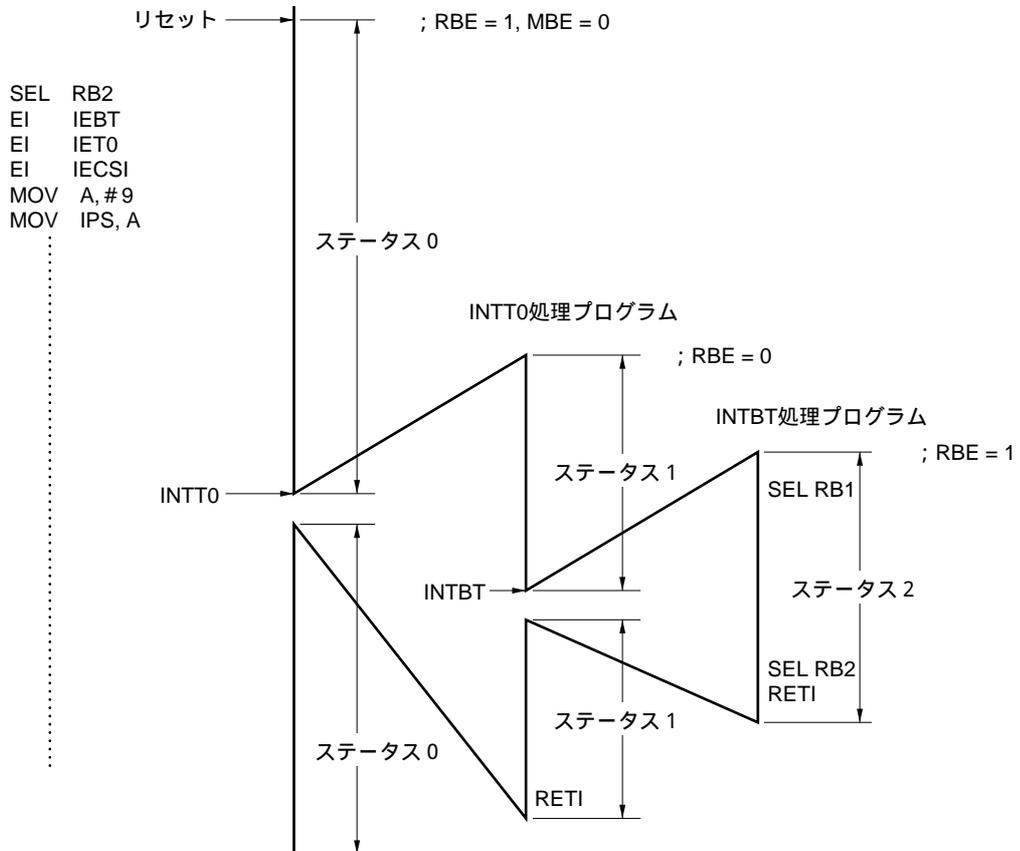
(2) INTBT, INTO (立ち下がりエッジ・アクティブ), INTT0を使用する例, 多重割り込みはしない。



- ① $\overline{\text{RESET}}$ 信号により, すべての割り込みが禁止され, ステータス0になる。
- ② INTOを立ち下がりエッジ・アクティブに設定。
- ③ EI, EI IE x x x 命令により割り込み許可。
- ④ INTOの立ち下がりにより, INTO割り込み処理プログラムをスタート。ステータスは1に変更され, すべての割り込みは禁止される。
- ⑤ RETI命令により割り込みから復帰。ステータスは0に戻り割り込みは許可される。

備考 この例のようにすべての割り込みを「低位の割り込み」として使用する場合は, メイン・プログラムではRBE = 1, RBS = 2としてレジスタ・バンク 2, 3を, 割り込み処理プログラムではRBE = 0としてレジスタ・バンク 0, 1を使えば, レジスタの退避/復帰はまったく不要となります。

(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。

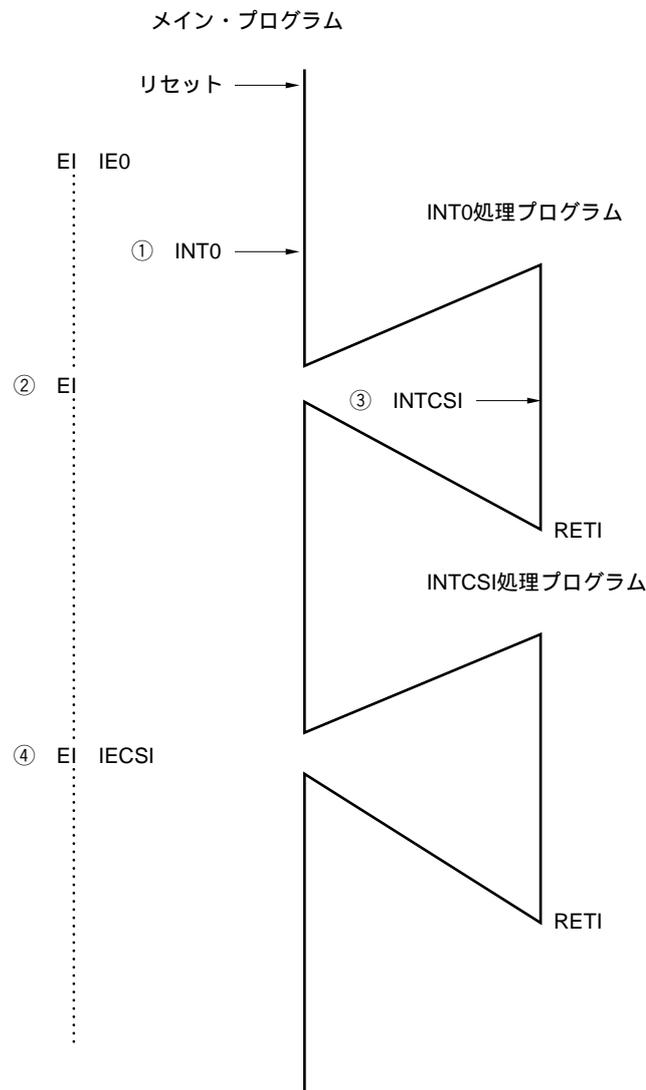
低位の割り込みINTT0発生により, INTT0処理プログラム開始, ステータス 1 となり低位の割り込みは禁止。RBE = 0 としてレジスタ・バンク 0 を使用。

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命令で回避してもよい)。

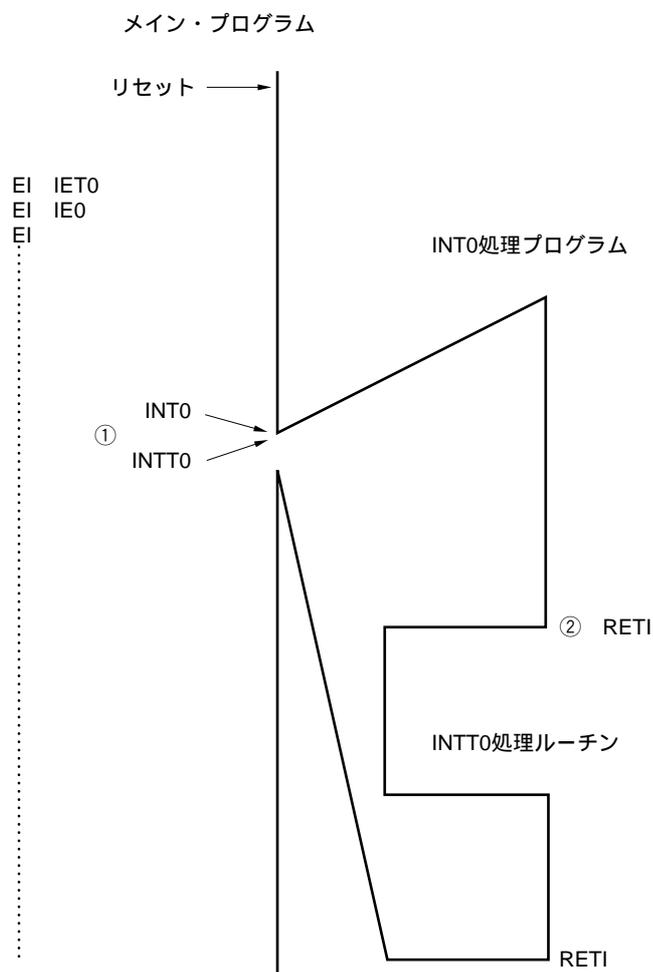
RBSを 2 に戻し, リターン。ステータスは 1 に戻る。

(4) 保留割り込みの実行 - 割り込み禁止中の割り込み入力 -



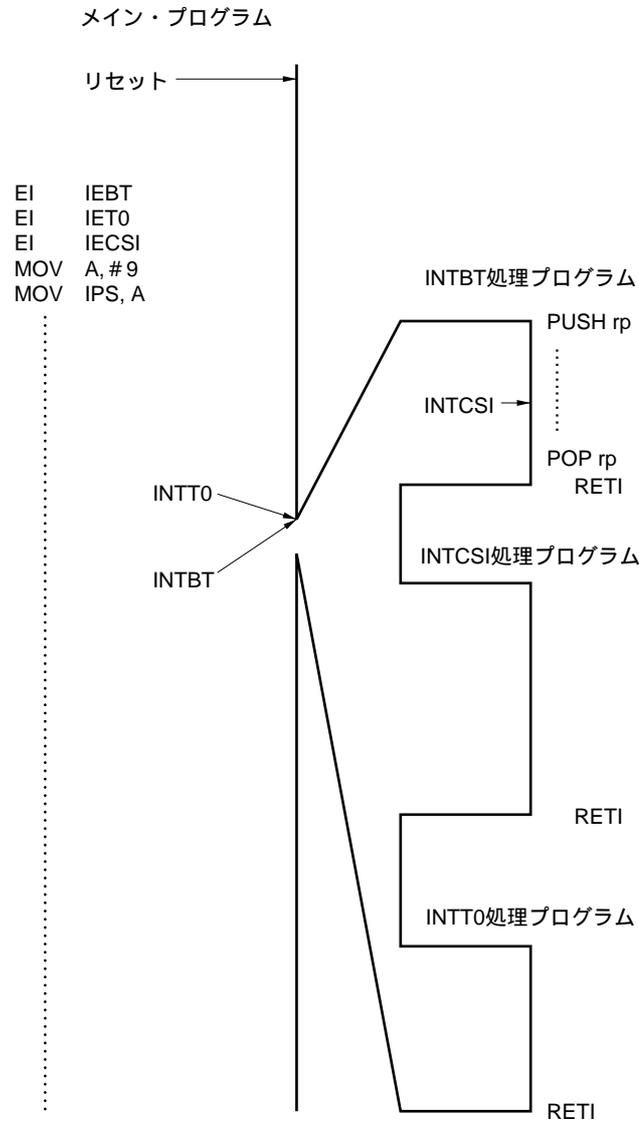
- ① 割り込み禁止中にINT0がセットされても要求フラグは保留される。
- ② EI命令で割り込みが許可された時点で、INT0処理プログラム・スタート。
- ③ ①と同様。
- ④ 保留されていたINTCSIが許可された時点で、INTCSI処理プログラム・スタート。

(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 -



- ① 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）。
- ② 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)。

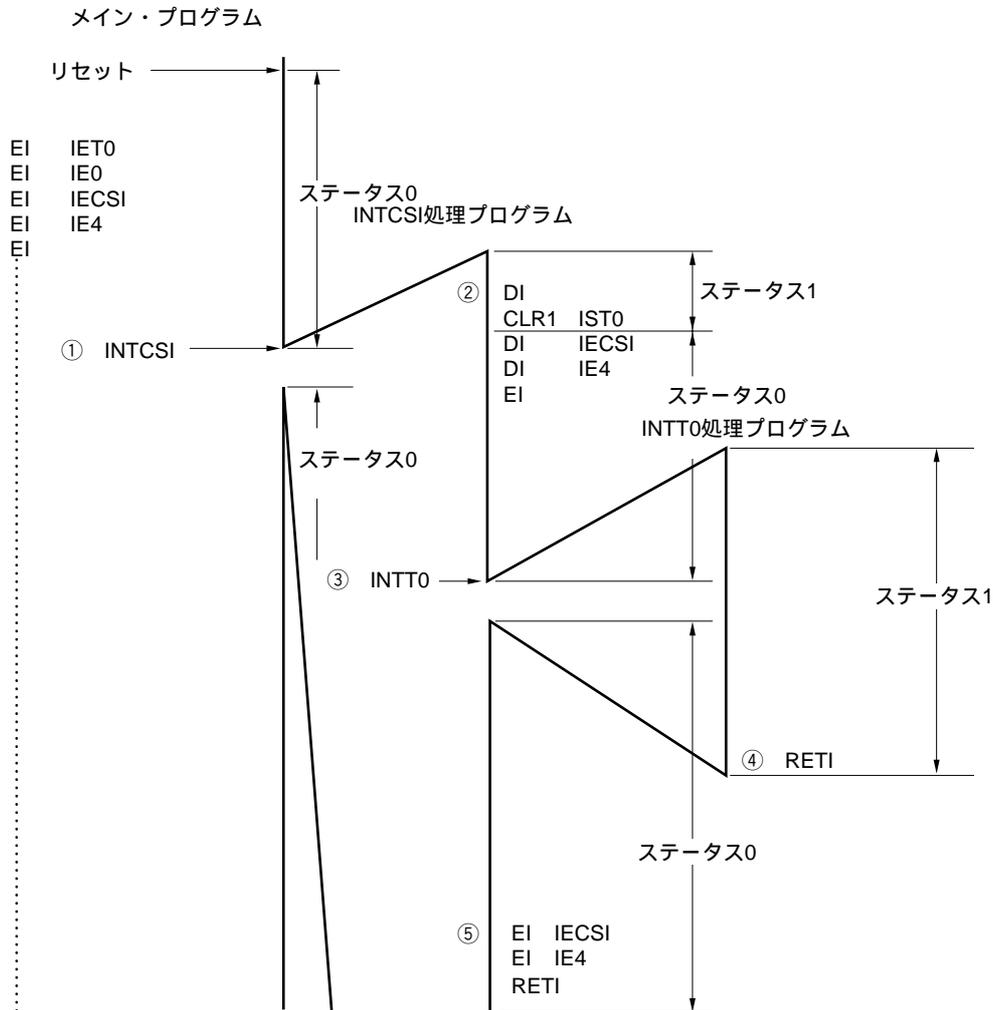
高位の割り込み実行中に低位の割り込みが発生すると、割り込みは保留される。

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 -



- ① 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。
- ② 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。
- ③ 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。
- ④ INTT0処理終了により、ステータスは0に戻る。
- ⑤ 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

μPD750008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ابل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-KR7への入力のいずれかの立ち下がりエッジ検出)	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 テスト要求信号のセット信号

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-KR7/P73端子への入力のいずれか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-KR7) のハードウェア

INT2, KR0-KR7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-KR7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-KR7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-KR7のブロック図

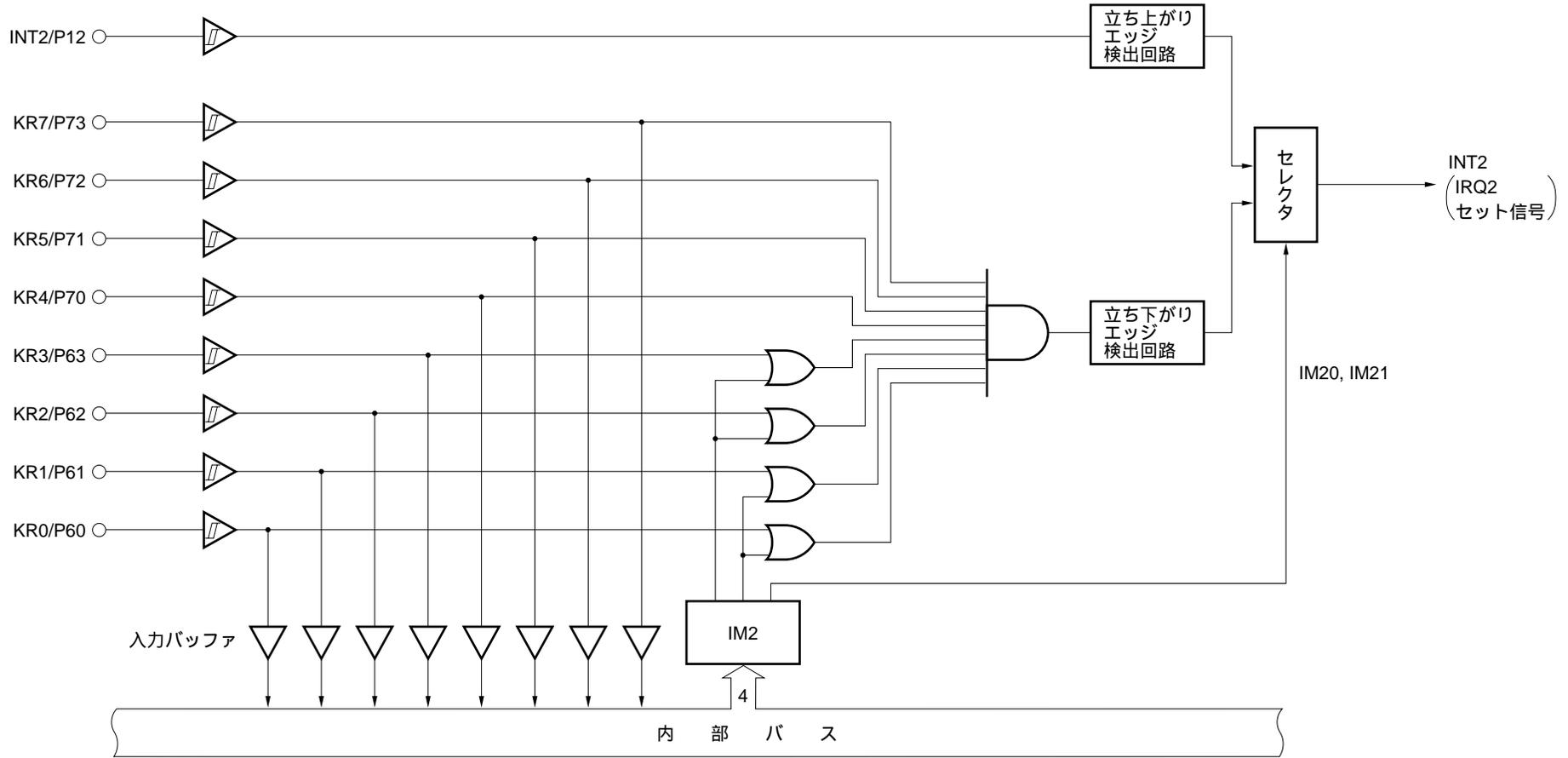


図6-11 INT2エッジ検出モード・レジスタ(IM2)のフォーマット



- 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

μPD750008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

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。

μPD750008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エ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。

使用上の注意を次頁に示します。

- 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・クロックのどちらかの動作状態でも使用できません。
- 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

項目	モード	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・クロックのみ発振停止	CPUクロックのみ停止（発振継続）
	ベーシック・インターバル・タイム / ウォッチドッグ・タイム	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の端子入力を指定した場合のみ、動作可能	カウント・クロックにTI0の端子入力を指定した場合、またはメイン・システム・クロック発振時のみ動作
	タイマ・カウンタ	動作停止	動作可能 ^{注1}
	時計用タイマ	カウント・クロックにf _{xr} を選択した場合動作可能	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 ^{注2}	
	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RESET信号発生	

注1．メイン・システム・クロック発振時のみ動作可能。

- 2．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。

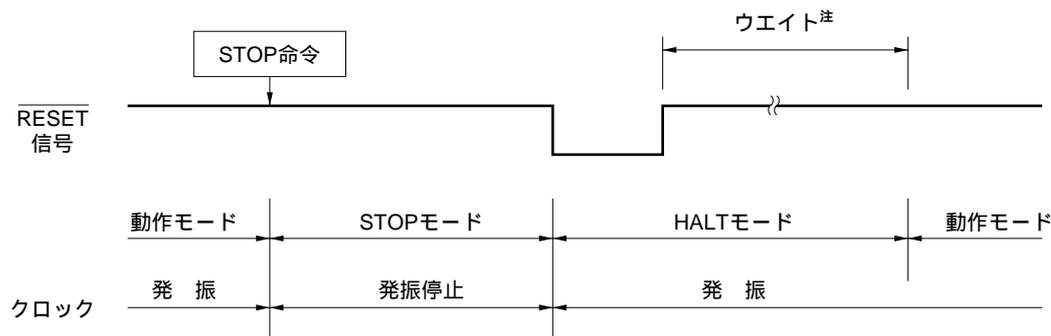
- 注意1 . STOPモードに設定されると、クリスタル発振回路部のリークを抑えるためにX1入力が入部でV_{SS}（GND電位）にショートされます。したがって、外部クロックを使用するシステムでは、STOPモードは使用しないでください。
- 2 . スタンバイ・モードを設定する前に、あらかじめすべての割り込み要求フラグをリセットしてください。割り込み要求フラグと割り込み許可フラグの両方がセットされている割り込みソースがある場合には、スタンバイ・モードに入ってもただちに解除されます（図6 - 1 割り込み制御回路ブロック図を参照）。ただし、STOPモードを設定した場合は、STOP命令実行後すぐHALT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

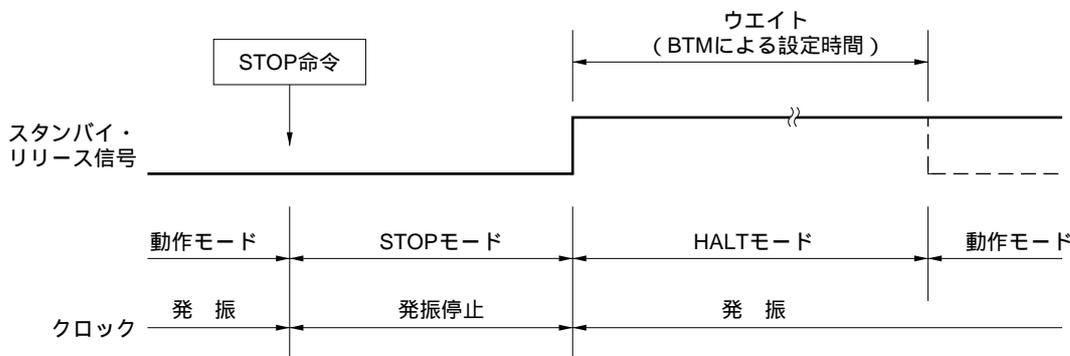
STOPモード，HALTモードいずれも割り込み許可フラグで許可されている割り込み要求信号の発生，および $\overline{\text{RESET}}$ 信号発生により解除されます。各モードの解除動作は図7-1のようになります。

図7-1 スタンバイ・モードの解除動作(1/2)

(a) STOPモードの $\overline{\text{RESET}}$ 入力による解除



(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 : 6.00 MHz動作時, 31.3 ms : 4.19 MHz動作時)

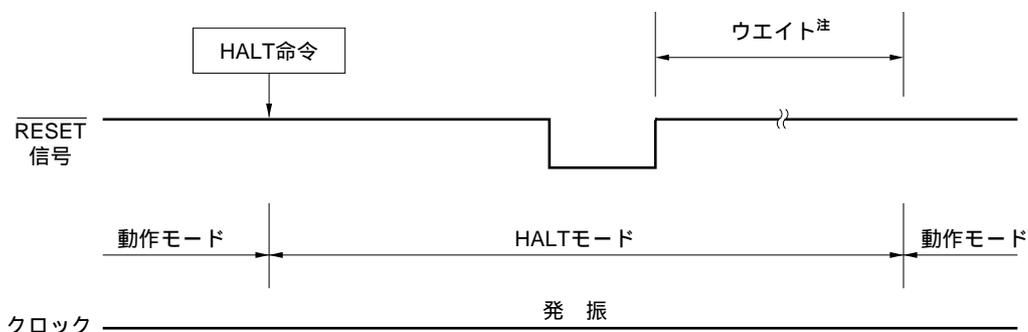
$2^{15}/f_x$ (5.46 ms : 6.00 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 μ PD75P0016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。

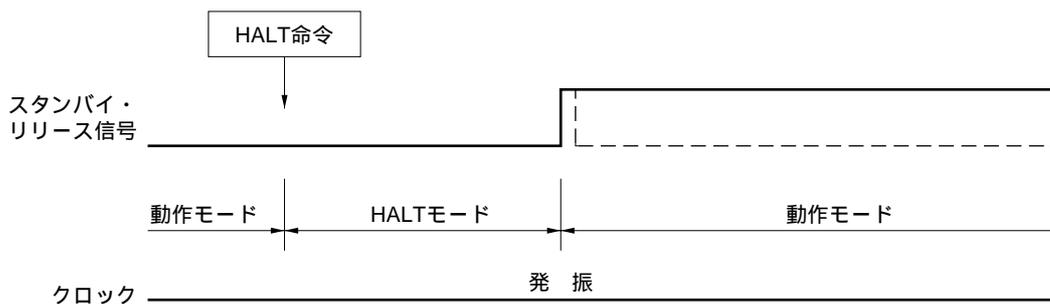
備考 破線は、スタンバイを解除した割り込み要求が受け付けられた場合です。

図7-1 スタンバイ・モードの解除動作 (2/2)

(c) HALTモードの $\overline{\text{RESET}}$ 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 : 6.00 MHz動作時, 31.3 ms : 4.19 MHz動作時)

$2^{15}/f_x$ (5.46 ms : 6.00 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 μ PD75P0016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。

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、ウェイト時間は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ェイト時間を選択し、STOPモードを設定する前にBTMをセットします。

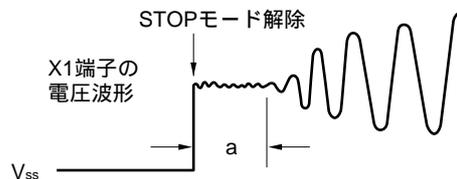
表7 - 2 BTMによるウェイト時間の選択

BTM3	BTM2	BTM1	BTM0	ウェイト時間 ^注	
				()内は $f_x = 6.00$ MHz時	()内は $f_x = 4.19$ 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

μ PD750008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。

- ① $2^{17}/f_x$ (21.8 ms : 6.00 MHz動作時, 31.3 ms : 4.19 MHz動作時)
- ② $2^{15}/f_x$ (5.46 ms : 6.00 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 μ PD75P0016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。

- ① 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）。
- ② 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）。
- ③ 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）。
- ④ 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）。
- ⑤ 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・モードの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）。
- ⑥ 解除時のウェイト時間の選択。
- ⑦ 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 = 4.19 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。

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。

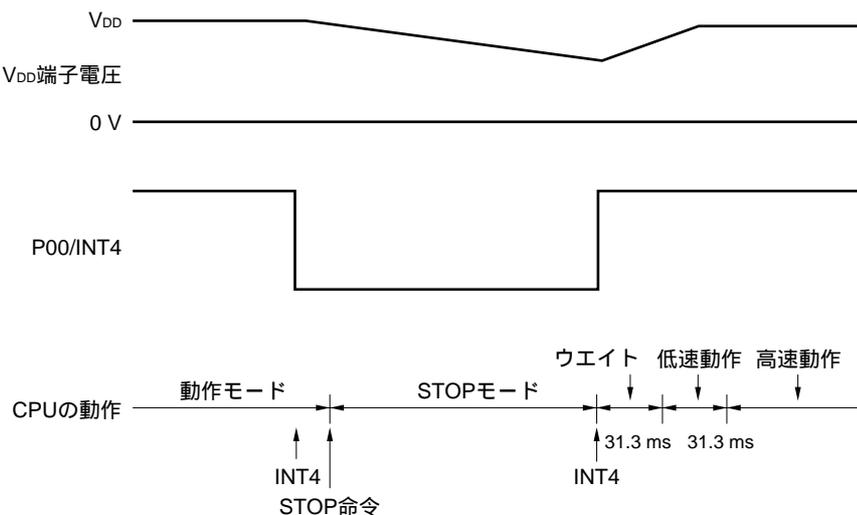
解除後も割り込み許可する。

解除後は、最低速のCPUクロックで動作スタートし、31.3 ms後に高速に切り替える。

解除時のウェイト時間は約31.3 msとする。

解除後、電源の安定のためにさらに31.3ms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)

```

VSUB4 :   SKT    PORT0.0      ; P00=1 ?
          BR     PDOWN       ; パワーダウン
          SET1   BTM.3        ; パワーオン
WAIT :    SKT    IRQBT        ; 31.3 msウエイト
          BR     WAIT
          SKT    PORT0.0      ; チャタリングをチェック
          BR     PDOWN
          MOV    A, #0011B
          MOV    PCC, A       ; 高速モードを設定
          ( 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       ; ウエイト時間 31.3 ms
          STOP   ; STOPモードをセット
          NOP
          RETI

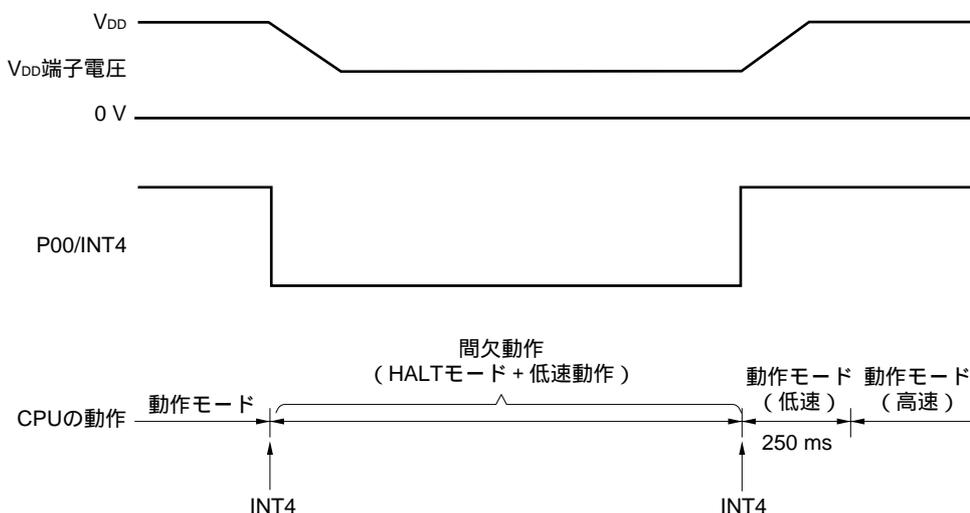
```

(2) HALTモードの応用 (fx = 4.19 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、サブシステム・クロックに切り替える。
- メイン・システム・クロックの発振を停止し、HALTモードに設定する。
- スタンバイ・モード中は0.5 sec間隔で、間欠動作する。
- INT4の立ち上がりで、メイン・システム・クロックに再び切り替える。
- INTBTは使用しない。

< 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, #05
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          ; 0.5 sec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, #8
          MOV    BTM, A
WAIT1 :  SKT    IRQBT        ; 250 msウエイト
          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, #6
WAIT2 :  INCS   A            ; 32マシン・サイクル待機
          BR     WAIT2
          SET1   SCC.3       ; メイン・システム・クロック発振停止
          RETI

```

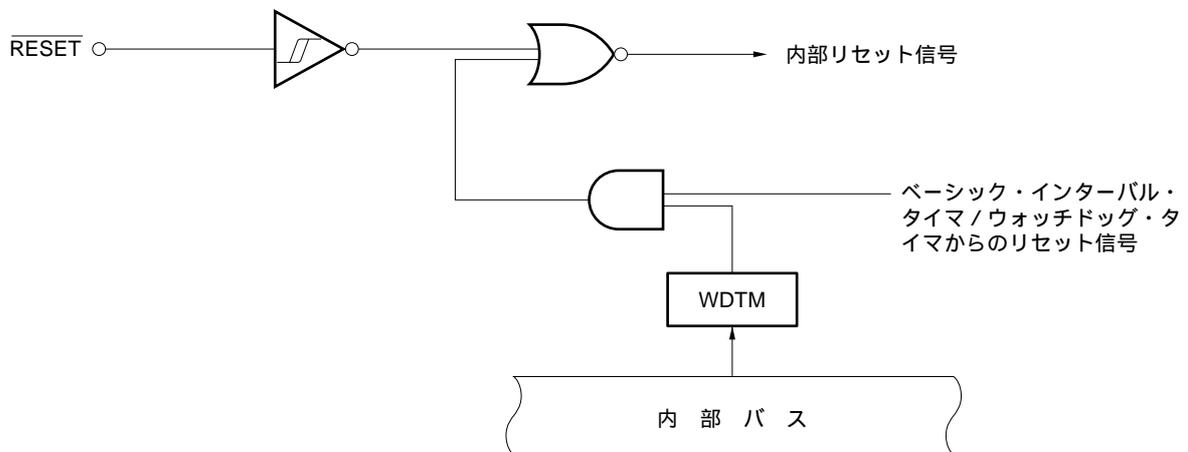
注意 メイン・システム・クロックからサブシステム・クロックに切り替える場合、サブシステム・クロックの発振安定を待ってから切り替えてください。

(メモ)

第8章 リセット機能

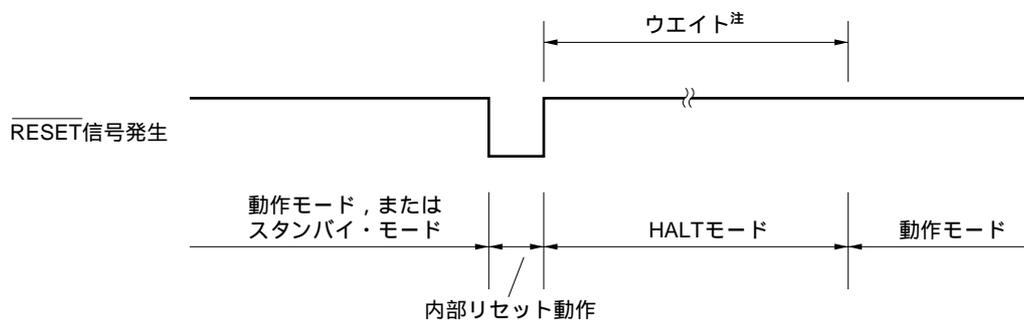
リセット入力には、外部リセット信号 ($\overline{\text{RESET}}$) とベーシック・インターバル・タイマ/ウォッチドッグ・タイマからのリセット信号の2種類があります。どちらか一方のリセット信号が入力されると、内部リセット信号が発生します。図8 - 1 にそれぞれの構成を示します。

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 : 6.00 MHz動作時, 31.3 ms : 4.19 MHz動作時)

$2^{15}/f_x$ (5.46 ms : 6.00 MHz動作時, 7.81 ms : 4.19 MHz動作時)

ただし、 μ PD75P0016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。

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 PD750004	プログラム・メモリの0000H番地の下位4ビットをPC11-8に、0001H番地の内容をPC7-0にセット	左に同じ
	μ PD750006,750008	プログラム・メモリの0000H番地の下位5ビットをPC12-8に、0001H番地の内容をPC7-0にセット	
	μ PD75P0016	プログラム・メモリの0000H番地の下位6ビットをPC13-8に、0001H番地の内容をPC7-0にセット	
PSW	キャリー・フラグ (CY)	保持	不定
	スキップ・フラグ (SK0-2)	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 ^注	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	カウンタ (BT)	不定	不定
バル・タイマ/ウォッチ ドッグ・タイマ	モード・レジスタ (BTM)	0	0
	ウォッチドッグ・タイマ許可フラグ (WDTM)	0	0
タイマ/イベント・ カウンタ	カウンタ (T0)	0	0
	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・カウンタ	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
	TOE1, TOUT F/F	0, 0	0, 0

注 データ・メモリの0F8H-0FDH番地のデータは、RESET信号発生により不定となります。

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
時計用タイマ	モード・レジスタ (WM)	0	0
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
	スレーブ・アドレス・レジスタ (SVA)	保持	不定
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
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
	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, 1, 2)	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, B, C)	0	0
	プルアップ抵抗指定レジスタ (POGA, B)	0	0
ビット・シーケンシャル・バッファ (BSB0-3)		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

μPD75P0016に内蔵されているプログラム・メモリは16384 × 8ビットのワン・タイムPROMです。このワン・タイムPROMの書き込み/ベリファイのために次の表に示すような端子を使用します。なお、アドレス入力ではなく、代わりにX1端子からのクロック入力により、アドレスを更新する方法をとっています。

端子名	機能
V _{PP}	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 _{DD} 電位)。
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。X2端子にはX1端子の逆相信号を入力します。
MD0 - MD3	プログラム・メモリ書き込み/ベリファイ時の動作モード選択端子。
P40 - P43 (下位4) P50 - P53 (下位4)	プログラム・メモリ書き込み/ベリファイ時の8ビット・データ入出力端子。
V _{DD}	電源電圧印加端子。 通常動作時は2.2~5.5V, プログラム・メモリ書き込み/ベリファイ時は+6Vを印加。

注意 1. μPD75P0016CU/GBは、消去用窓を備えておらず、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...プルダウン抵抗を介してV_{SS}に接続
- XT2端子.....オープン

9.1 プログラム・メモリ書き込み/ベリファイ時の動作モード

μPD75P0016は、V_{DD}端子に+6V, V_{PP}端子に+12.5Vを印加すると、プログラム・メモリ書き込み/ベリファイ・モードになります。このモードは、MD0 - MD3端子の設定により次のような動作モードとなります。

動作モードの指定						動作モード
V _{PP}	V _{DD}	MD0	MD1	MD2	MD3	
+12.5V	+6V	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

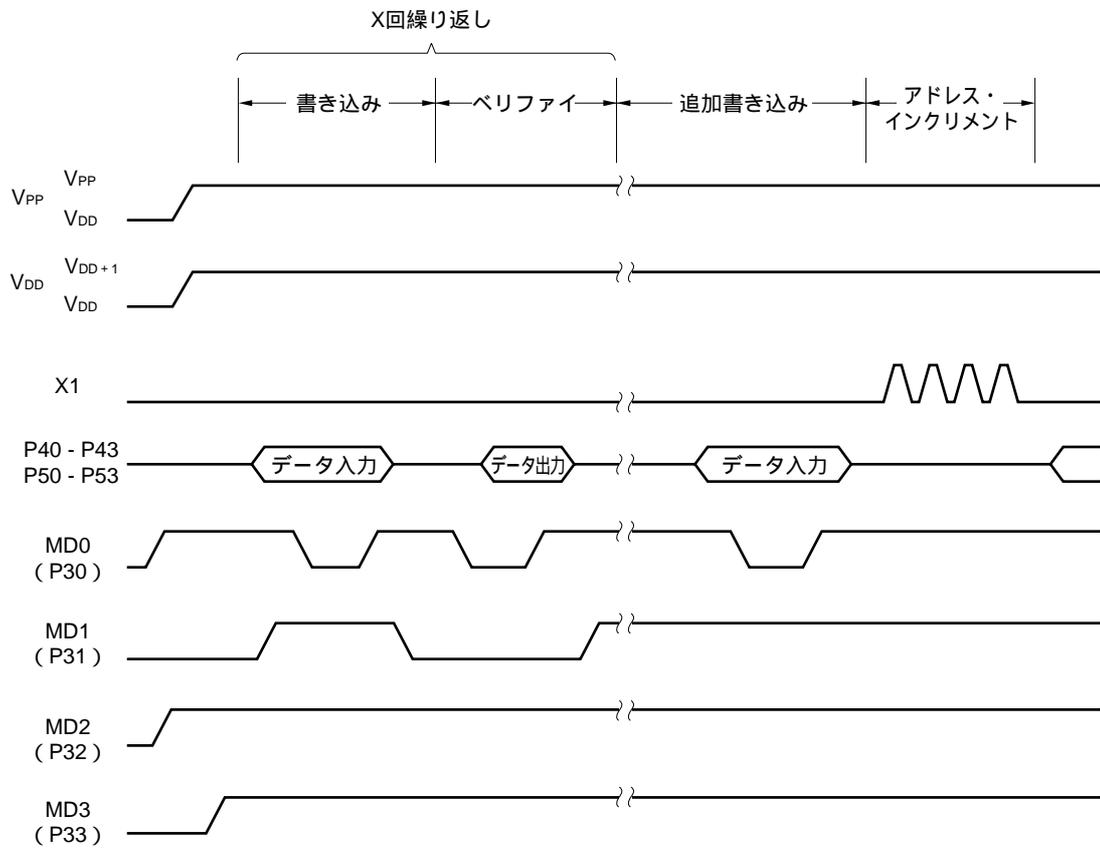
備考 x : LまたはH

9.2 プログラム・メモリ書き込みの手順

プログラム・メモリ書き込みの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_{SS}にプルダウン。X1端子はロウ・レベル。
- (2) V_{DD}, V_{PP}端子に5 Vを供給。
- (3) 10 μ 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_{DD}に6 V, V_{PP}に12.5 Vを供給。
- (6) プログラム・インヒビット・モード。
- (7) 1 msの書き込みモードでデータを書き込む。
- (8) プログラム・インヒビット・モード。
- (9) ペリファイ・モード。書き込めていれば(10)へ、書き込めていなければ(7)~(9)を繰り返す。
- (10) ((7)~(9)で書き込んだ回数: X) \times 1 msの追加書き込み。
- (11) プログラム・インヒビット・モード。
- (12) 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)。
- (13) (7)~(12)を最終アドレスまで繰り返す。
- (14) プログラム・メモリ・アドレスの0クリア・モード。
- (15) V_{DD}, V_{PP}端子の電圧を5 Vに変更。
- (16) 電源オフ。

(2) ~ (12) の手順を下图に示します。

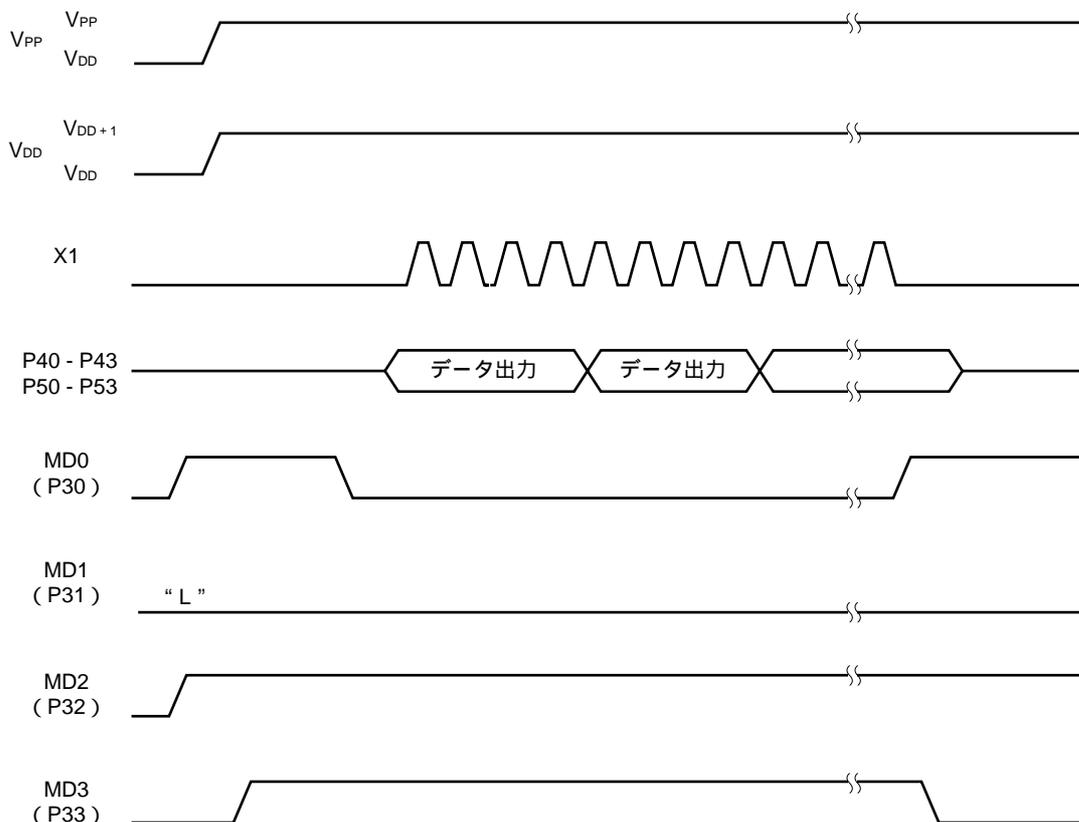


9.3 プログラム・メモリ読み出しの手順

μPD75P0016は、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_{SS}にプルダウン。X1端子はロウ・レベル。
- (2) V_{DD}, V_{PP}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_{DD}に6 V, V_{PP}に12.5 Vを供給。
- (6) プログラム・インヒビット・モード。
- (7) 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。
- (8) プログラム・インヒビット・モード。
- (9) プログラム・メモリ・アドレスの0クリア・モード。
- (10) V_{DD}, V_{PP}の電圧を5 Vに変更。
- (11) 電源オフ。

(2) ~ (9) 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

μPD750008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	ビット単位でプルアップ抵抗の内蔵を指定可能
P50-P53	

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。

μPD75P0016にはマスク・オプションはなく、常にオープンになっています。

10.2 スタンバイ機能のマスク・オプション

μPD750008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。ウェイト時間とは、 $\overline{\text{RESET}}$ 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： $f_x = 6.00$ MHz動作時，31.3 ms： $f_x = 4.19$ MHz動作時）

$2^{15}/f_x$ （5.46 ms： $f_x = 6.00$ MHz動作時，7.81 ms： $f_x = 4.19$ MHz動作時）

μPD75P0016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

μPD750008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。

μPD75P0016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

μPD750008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、μPD750008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) 全空間のサブルーチン・コール命令
- (b) 全空間への分岐命令
- (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

μ PD750008のビット操作は次のような多彩な命令により行うことができます。

(a) ビットのセット	: SET1	mem. bit
	SET1	mem. bit*
(b) ビットのクリア	: CLR1	mem. bit
	CLR1	mem. bit*
(c) ビットのテスト	: SKT	mem. bit
	SKT	mem. bit*
(d) ビットのテスト	: SKF	mem. bit
	SKF	mem. bit*
(e) ビットのテスト&クリア	: SKTCLR	mem. bit*
(f) プーリアン演算	: AND1	CY, mem. bit*
	OR1	CY, mem. bit*
	XOR1	CY, mem. bit*

mem. bit* は、ビット操作アドレッシング (fmem. bit, pmem. @L, @H + mem. bit) によって示されるビット・アドレスです。

特に、入出力ポートについては、上記のすべてのビット操作命令が常に適用できるため、入出力ポートの操作を非常に効率的に行うことができます。

11.1.3 たてづみ命令

μ PD750008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μPD750008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。

(a) 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
    ⋮
```

(b) 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、このときこの命令を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

μPD750008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・サイクル数は次のようになります。

- (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。

イミディエト・データの場合は、適当な数値またはレーベルを記述します。

レーベルとして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・フラグの略号を記述できます（ただし、fmem, pmemは記述できるレー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 μ PD750008 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 ^注
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,	0000H - 0FFFHイミディエト・データまたはレーベル (μ PD750004)
addr1(Mk	0000H - 17FFHイミディエト・データまたはレーベル (μ PD750006)
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 PD750008)
	0000H - 3FFFHイミディエト・データまたはレーベル (μ PD75P0016)
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 - PORT8
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
PORTn	: ポートn (n=0-8)
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 x x x	: 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・エリアの欄の記号説明

* 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 (00H - 7FH) MB = 15 (F80H - FFFH) MBE = 1 : MB = MBS (MBS = 0 , 1 , 15)		
* 4	MB = 15 , fmem = FB0H - FBFH , FF0H - FFFH		
* 5	MB = 15 , pmem = FC0H - FFFH		
* 6	μ PD750004	addr, addr1 = 0000H - 0FFFH	プログラム・メモリ アドレッシング
	μ PD750006	addr, addr1 = 0000H - 17FFH	
	μ PD750008	addr, addr1 = 0000H - 1FFFH	
	μ PD75P0016	addr, addr1 = 0000H - 3FFFH	
* 7	addr, addr1 = (Current PC) - 15 ~ (Current PC) - 1 (Current PC) + 2 ~ (Current PC) + 16		
* 8	μ PD750004	caddr = 0000H - 0FFFH	
	μ PD750006	caddr = 0000H - 0FFFH (PC ₁₂ = 0) or 1000H - 17FFH (PC ₁₂ = 1)	
	μ PD750008	caddr = 0000H - 0FFFH (PC ₁₂ = 0) or 1000H - 1FFFH (PC ₁₂ = 1)	
	μ PD75P0016	caddr = 0000H - 0FFFH (PC ₁₃ , PC ₁₂ = 00B) or 1000H - 1FFFH (PC ₁₃ , PC ₁₂ = 01B) or 2000H - 2FFFH (PC ₁₃ , PC ₁₂ = 10B) or 3000H - 3FFFH (PC ₁₃ , PC ₁₂ = 11B)	
* 9	faddr = 0000H - 07FFH		
* 10	taddr = 0020H - 007FH		
* 11	Mk モード時のみ addr1 = 0000H - 0FFFH (μ PD750004) 0000H - 17FFH (μ PD750006) 0000H - 1FFFH (μ PD750008) 0000H - 3FFFH (μ PD75P0016)		

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 ~ * 10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^注のとき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 の1サイクル分 (=tcy) 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	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
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 PD750004 XA (PC ₁₁₋₈ + DE) _{ROM}		
					• μ PD750006, 750008 XA (PC ₁₂₋₈ + DE) _{ROM}		
					• μ PD75P0016 XA (PC ₁₃₋₈ + DE) _{ROM}		
		XA, @PCXA	1	3	• μ PD750004 XA (PC ₁₁₋₈ + XA) _{ROM}		
					• μ PD750006, 750008 XA (PC ₁₂₋₈ + XA) _{ROM}		
					• μ PD75P0016 XA (PC ₁₃₋₈ + XA) _{ROM}		
	XA, @BCDE	1	3	XA (BCDE) _{ROM} ^注	* 6		
	XA, @BCXA	1	3	XA (BCXA) _{ROM} ^注	* 6		
ビット転送	MOV1	CY, fmem.bit	2	2	CY (fmem.bit)	* 4	
		CY, pmem.@L	2	2	CY (pmem ₇₋₂ + L ₃₋₂ .bi(L ₁₋₀))	* 5	
		CY, @H + mem.bit	2	2	CY (H + mem ₃₋₀ .bit)	* 1	
		fmem.bit, CY	2	2	(fmem.bit) CY	* 4	
		pmem.@L, CY	2	2	(pmem ₇₋₂ + L ₃₋₂ .bi(L ₁₋₀)) CY	* 5	
		@H + mem.bit, CY	2	2	(H + mem ₃₋₀ .bit) 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
	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		

注 μ PD750004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 PD750006, 750008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 PD75P0016を使用する場合は、下位2ビットのみ有効です。

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
演算	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 ⊕ 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		
ビット操作	RORC	A	1	1	CY A ₀ , A ₃ CY, A _{n-1} A _n		
	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'
キャリー・フラグ操作	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 ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀)) 1	* 5	
		@ H + mem.bit	2	2	(H + mem ₃₋₀ .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 ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀)) 0	* 5	
		@ H + mem.bit	2	2	(H + mem ₃₋₀ .bit) 0	* 1	
	SKT	mem.bit	2	2 + S	Skip if(mem.bit)= 1	* 3	(mem.bit)= 1
		fmem.bit	2	2 + S	Skip if(fmem.bit)= 1	* 4	(fmem.bit)= 1
		pmem.@ L	2	2 + S	Skip if(pmem ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))= 1	* 5	(pmem.@ L)= 1
		@ H + mem.bit	2	2 + S	Skip if(H + mem ₃₋₀ .bit)= 1	* 1	(@ H + mem.bit)= 1
	SKF	mem.bit	2	2 + S	Skip if(mem.bit)= 0	* 3	(mem.bit)= 0
		fmem.bit	2	2 + S	Skip if(fmem.bit)= 0	* 4	(fmem.bit)= 0
		pmem.@ L	2	2 + S	Skip if(pmem ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))= 0	* 5	(pmem.@ L)= 0
		@ H + mem.bit	2	2 + S	Skip if(H + mem ₃₋₀ .bit)= 0	* 1	(@ H + mem.bit)= 0
	SKTCLR	fmem.bit	2	2 + S	Skip if(fmem.bit)= 1 and clear	* 4	(fmem.bit)= 1
		pmem.@ L	2	2 + S	Skip if(pmem ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))= 1 and clear	* 5	(pmem.@ L)= 1
		@ H + mem.bit	2	2 + S	Skip if(H + mem ₃₋₀ .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 ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))	* 5	
		CY , @ H + mem.bit	2	2	CY CY ∧ (H + mem ₃₋₀ .bit)	* 1	
	OR1	CY , fmem.bit	2	2	CY CY ∨ (fmem.bit)	* 4	
		CY , pmem.@ L	2	2	CY CY ∨ (pmem ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))	* 5	
		CY , @ H + mem.bit	2	2	CY CY ∨ (H + mem ₃₋₀ .bit)	* 1	
	XOR1	CY , fmem.bit	2	2	CY CY ⊖ (fmem.bit)	* 4	
		CY , pmem.@ L	2	2	CY CY ⊖ (pmem ₇₋₂ + L ₃₋₂ .bit(L ₁₋₀))	* 5	
		CY , @ H + mem.bit	2	2	CY CY ⊖ (H + mem ₃₋₀ .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	-	-	<ul style="list-style-type: none"> ・ μPD750004 <p>PC₁₁₋₀ 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"> ・ BR !addr ・ BR \$addr ・ BRCB !caddr 	* 6	
					<ul style="list-style-type: none"> ・ μPD750006, 750008 <p>PC₁₂₋₀ 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"> ・ BR !addr ・ BRCB !caddr ・ BR \$addr 		
		addr1 ^注	-	-	<ul style="list-style-type: none"> ・ μPD750004 <p>PC₁₁₋₀ addr1</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"> ・ BRA !addr1 ・ BR !addr ・ BRCB !caddr ・ BR \$addr1 	* 11	
					<ul style="list-style-type: none"> ・ μPD750006, 750008 <p>PC₁₂₋₀ addr1</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"> ・ BRA !addr1 ・ BR !addr ・ BRCB !caddr ・ BR \$addr1 		

注 示す部分はMk モード時のみ対応可能です。

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
分岐	BR	addr1 ^注	-	-	<ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ 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。 <ul style="list-style-type: none"> • BRA !addr1 • BR !addr • BRCB !caddr • BR \$addr1 	* 11	
		laddr	3	3	<ul style="list-style-type: none"> • μPD750004 PC ₁₁₋₀ addr <ul style="list-style-type: none"> • μPD750006, 750008 PC ₁₂₋₀ addr <ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ addr	* 6	
		\$addr	1	2	<ul style="list-style-type: none"> • μPD750004 PC ₁₁₋₀ addr <ul style="list-style-type: none"> • μPD750006, 750008 PC ₁₂₋₀ addr <ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ addr	* 7	
		\$addr1	1	2	<ul style="list-style-type: none"> • μPD750004 PC ₁₁₋₀ addr1 <ul style="list-style-type: none"> • μPD750006, 750008 PC ₁₂₋₀ addr1 <ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ addr1	* 7	
		PCDE	2	3	<ul style="list-style-type: none"> • μPD750004 PC ₁₁₋₀ PC ₁₁₋₈ + DE <ul style="list-style-type: none"> • μPD750006, 750008 PC ₁₂₋₀ PC ₁₂₋₈ + DE <ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ PC ₁₃₋₈ + DE		
		PCXA	2	3	<ul style="list-style-type: none"> • μPD750004 PC ₁₁₋₀ PC ₁₁₋₈ + XA <ul style="list-style-type: none"> • μPD750006, 750008 PC ₁₂₋₀ PC ₁₂₋₈ + XA <ul style="list-style-type: none"> • μPD75P0016 PC ₁₃₋₀ PC ₁₃₋₈ + XA		

注 で示す部分はMk モード時のみ対応可能です。

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
分岐	BR	BCDE	2	3	・ μ PD750004 PC ₁₁₋₀ BCDE ^{注1}	* 11	
					・ μ PD750006, 750008 PC ₁₂₋₀ BCDE ^{注2}		
					・ μ PD75P0016 PC ₁₃₋₀ BCDE ^{注3}		
		BCXA	2	3	・ μ PD750004 PC ₁₁₋₀ BCXA ^{注1}		
					・ μ PD750006, 750008 PC ₁₂₋₀ BCXA ^{注2}		
					・ μ PD75P0016 PC ₁₃₋₀ BCXA ^{注3}		
	BRA ^{注4}	!addr1	3	3	・ μ PD750004 PC ₁₁₋₀ addr	* 11	
					・ μ PD750006, 750008 PC ₁₂₋₀ addr		
					・ μ PD75P0016 PC ₁₃₋₀ addr1		
	BRCB	!caddr	2	2	・ μ PD750004 PC ₁₁₋₀ caddr ₁₁₋₀	* 8	
・ μ PD750006, 750008 PC ₁₂₋₀ PC ₁₂ + caddr ₁₁₋₀							
・ μ PD75P0016 PC ₁₃₋₀ PC _{13, 12} + caddr ₁₁₋₀							

注1．Bレジスタには必ず“0”を設定してください。

2．Bレジスタは、下位1ビットのみ有効です。

3．Bレジスタは、下位2ビットのみ有効です。

4．で示す部分はMkモード時のみ対応可能です。

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	CALLA ^注	laddr1	3	3	<ul style="list-style-type: none"> • μPD750004 (SP - 2) x , x , MBE, RBE (SP - 6 X SP - 3 X SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , 0 PC₁₁₋₀ addr, SP SP - 6 	* 11	
				<ul style="list-style-type: none"> • μPD750006, 750008 (SP - 2) x , x , MBE, RBE (SP - 6 X SP - 3 X SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , PC₁₂ PC₁₂₋₀ addr, SP SP - 6 			
	CALL ^注	laddr	3	3	<ul style="list-style-type: none"> • μPD750004 (SP - 3) MBE, RBE, 0 , 0 (SP - 4 X SP - 1 X SP - 2) PC₁₁₋₀ PC₁₁₋₀ addr, SP SP - 4 	* 6	
				<ul style="list-style-type: none"> • μPD750006, 750008 (SP - 3) MBE, RBE, 0 , PC₁₂ (SP - 4 X SP - 1 X SP - 2) PC₁₁₋₀ PC₁₂₋₀ addr, SP SP - 4 			
			4	<ul style="list-style-type: none"> • μPD75P0016 (SP - 3) MBE, RBE, PC₁₃ , PC₁₂ (SP - 4 X SP - 1 X SP - 2) PC₁₁₋₀ PC₁₃₋₀ addr1, SP SP - 4 			
				4	<ul style="list-style-type: none"> • μPD750004 (SP - 2) x , x , MBE, RBE (SP - 6 X SP - 3 X SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , 0 PC₁₁₋₀ 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 モード時のみに対応しています。

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	CALL ^注	laddr	3	4	<ul style="list-style-type: none"> • μPD750006, 750008 (SP - 2) x , x , MBE, RBE (SP - 6 \bar{x} SP - 3 \bar{x} SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , PC₁₂ PC₁₂₋₀ addr, SP SP - 6 <hr/> <ul style="list-style-type: none"> • μPD75P0016 (SP - 2) x , x , MBE, RBE (SP - 6 \bar{x} SP - 3 \bar{x} SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , PC₁₃ , PC₁₂ PC₁₃₋₀ addr, SP SP - 6 	* 6	
	CALLF ^注	lfaddr	2	2	<ul style="list-style-type: none"> • μPD750004 (SP - 3) MBE, RBE, 0 , 0 (SP - 4 \bar{x} SP - 1 \bar{x} SP - 2) PC₁₁₋₀ PC₁₁₋₀ 0 + faddr, SP SP - 4 <hr/> <ul style="list-style-type: none"> • μPD750006, 750008 (SP - 3) MBE, RBE, 0 , PC₁₂ (SP - 4 \bar{x} SP - 1 \bar{x} SP - 2) PC₁₁₋₀ PC₁₂₋₀ 00 + faddr, SP SP - 4 <hr/> <ul style="list-style-type: none"> • μPD75P0016 (SP - 3) MBE, RBE, PC₁₃ , PC₁₂ (SP - 4 \bar{x} SP - 1 \bar{x} SP - 2) PC₁₁₋₀ PC₁₃₋₀ 000 + faddr, SP SP - 4 	* 9	
				3	<ul style="list-style-type: none"> • μPD750004 (SP - 2) x , x , MBE, RBE (SP - 6 \bar{x} SP - 3 \bar{x} SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , 0 PC₁₁₋₀ 0 + faddr, SP SP - 6 <hr/> <ul style="list-style-type: none"> • μPD750006, 750008 (SP - 2) x , x , MBE, RBE (SP - 6 \bar{x} SP - 3 \bar{x} SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , 0 , PC₁₂ PC₁₂₋₀ 00 + faddr, SP SP - 6 		

注 以示す部分はMk モード時のみ対応可能です。そのほかはMk モード時のみに対応しています。

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 ^注	!faddr	2	3	<ul style="list-style-type: none"> • μPD75P0016 (SP - 2) x , x , MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC₁₁₋₀ (SP - 5) 0 , 0 , PC₁₃ , PC₁₂ PC₁₃₋₀ 000 + faddr, SP SP - 6 	* 9	
	RET ^注		1	3	<ul style="list-style-type: none"> μPD750004 PC₁₁₋₀ (SP) (SP + 3) (SP + 2) MBE, RBE, 0, 0 (SP + 1) , SP SP + 4 <hr/> <ul style="list-style-type: none"> • μPD750006, 750008 PC₁₁₋₀ (SP) (SP + 3) (SP + 2) MBE, RBE, 0, PC₁₂ (SP + 1) SP SP + 4 <hr/> <ul style="list-style-type: none"> • μPD75P0016 PC₁₁₋₀ (SP) (SP + 3) (SP + 2) MBE, RBE, PC₁₃ , PC₁₂ (SP + 1) SP SP + 4 <hr/> <ul style="list-style-type: none"> 3 • μPD750004 x , x , MBE, RBE (SP + 4) 0 , 0 , 0 , 0 (SP + 1) PC₁₁₋₀ (SP) (SP + 3) (SP + 2) SP SP + 6 <hr/> <ul style="list-style-type: none"> • μPD750006, 750008 x , x , MBE, RBE (SP + 4) MBE, 0 , 0 , PC₁₂ (SP + 1) PC₁₁₋₀ (SP) (SP + 3) (SP + 2) SP SP + 6 <hr/> <ul style="list-style-type: none"> • μPD75P0016 x , x , MBE, RBE (SP + 4) 0 , 0 , PC₁₃ , PC₁₂ (SP + 1) PC₁₁₋₀ (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 モード時のみに対応しています。

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 ^注		1	3 + S	<ul style="list-style-type: none"> • μPD750004 MBE, RBE, 0, 0 (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) SP SP + 4 then skip unconditionally 		無条件
					<ul style="list-style-type: none"> • μPD750006, 750008 MBE, 0, 0, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) SP SP + 4 then skip unconditionally 		
					<ul style="list-style-type: none"> • μPD75P0016 MBE, RBE, PC₁₃, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) SP SP + 4 then skip unconditionally 		
				3 + S	<ul style="list-style-type: none"> • μPD750004 0, 0, 0, 0 (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) x, x, MBE, RBE (SP + 4) SP SP + 6 then skip unconditionally 		
					<ul style="list-style-type: none"> • μPD750006, 750008 0, 0, 0, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) x, x, MBE, RBE (SP + 4) SP SP + 6 then skip unconditionally 		
					<ul style="list-style-type: none"> • μPD75P0016 0, 0, PC₁₃, PC₁₂ (SP + 1) PC₁₁₋₀ (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 モード時のみに対応しています。

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	RETI ^注		1	3	<ul style="list-style-type: none"> • μPD750004 MBE, RBE, 0, 0 (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) PSW (SP + 4 ∇ SP + 5), SP SP + 6 		無条件						
					<ul style="list-style-type: none"> • μPD750006, 750008 MBE, RBE, 0, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) PSW (SP + 4 ∇ SP + 5), SP SP + 6 								
					<ul style="list-style-type: none"> • μPD75P0016 MBE, RBE, PC₁₃, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) PSW (SP + 4 ∇ SP + 5), SP SP + 6 								
					<ul style="list-style-type: none"> • μPD750004 0, 0, 0, 0 (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) PSW (SP + 4 ∇ SP + 5), SP SP + 6 								
					<ul style="list-style-type: none"> • μPD750006, 750008 0, 0, 0, PC₁₂ (SP + 1) PC₁₁₋₀ (SP ∇ SP + 3 ∇ SP + 2) PSW (SP + 4 ∇ SP + 5), SP SP + 6 								
					<ul style="list-style-type: none"> • μPD75P0016 0, 0, PC₁₃, PC₁₂ (SP + 1) PC₁₁₋₀ (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 モード時のみに対応しています。

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	アドレッシング・エリア	スキップ条件
入出力	IN ^注	A, PORTn	2	2	A PORTn (n=0-8)		
		XA, PORTn	2	2	XA PORTn+1, PORTn(n=4, 6)		
	OUT ^注	PORTn, A	2	2	PORTn A (n=2-8)		
		PORTn, XA	2	2	PORTn+1, PORTn XA(n=4, 6)		
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		

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	アドレッシング・エリア	スキップ条件
特	SEL	RbN	2	2	RBS n (n=0-3)		
		MbN	2	2	MBS n (n=0, 1, 15)		
殊	GETI ^注	taddr	1	3	<ul style="list-style-type: none"> • μPD750004 TBR命令のとき $PC_{11-0} (taddr)_{8-0} + (taddr + 1)$	* 10	
					TCALL命令のとき $(SP - 4 \ \& \ SP - 1 \ \& \ SP - 2) \ PC_{11-0}$ $(SP - 3) \ MBE, RBE, 0, 0$ $PC_{11-0} (taddr)_{8-0} + (taddr + 1)$ $SP \ SP - 4$		
					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
					<ul style="list-style-type: none"> • μPD750006, 750008 TBR命令のとき $PC_{12-0} (taddr)_{8-0} + (taddr + 1)$		
					TCALL命令のとき $(SP - 4 \ \& \ SP - 1 \ \& \ SP - 2) \ PC_{11-0}$ $(SP - 3) \ MBE, RBE, 0, PC_{12}$ $PC_{12-0} (taddr)_{8-0} + (taddr + 1)$ $SP \ SP - 4$		
					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
					<ul style="list-style-type: none"> • μPD75P0016 TBR命令のとき $PC_{13-0} (taddr)_{8-0} + (taddr + 1)$		
					TCALL命令のとき $(SP - 4 \ \& \ SP - 1 \ \& \ SP - 2) \ PC_{11-0}$ $(SP - 3) \ MBE, RBE, PC_{13}, PC_{12}$ $PC_{13-0} (taddr)_{8-0} + (taddr + 1)$ $SP \ SP - 4$		
					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。

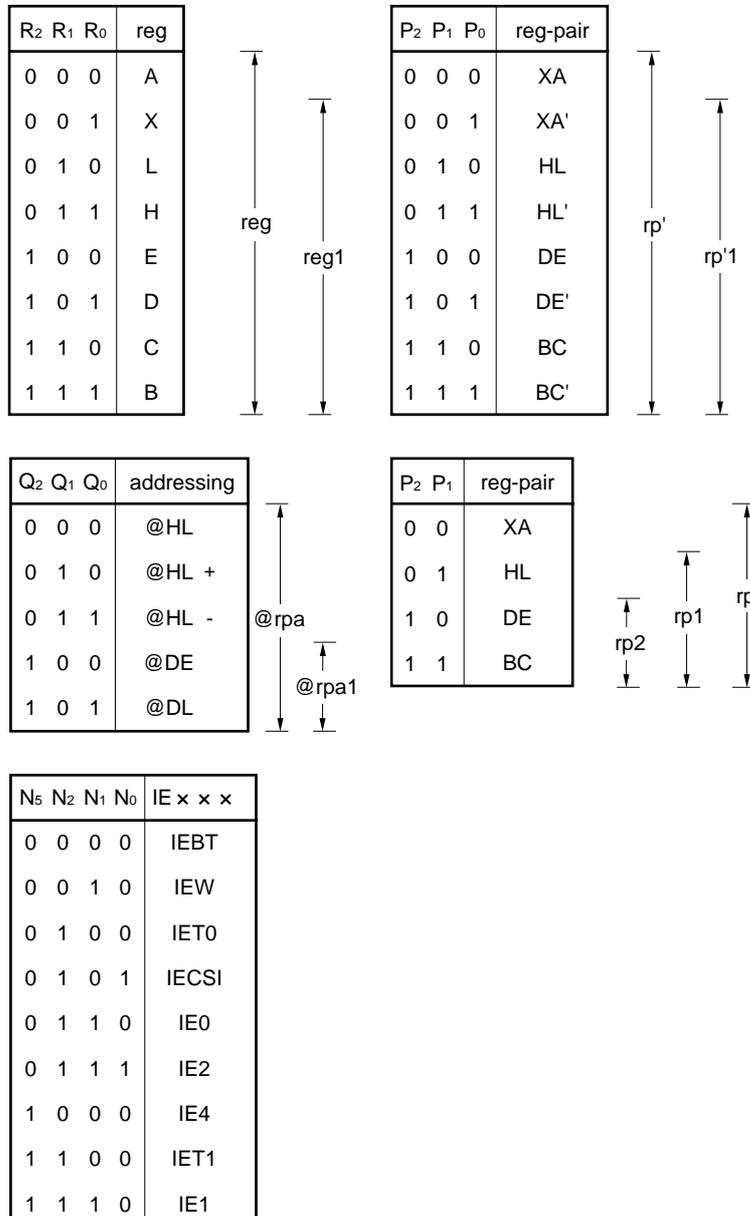
命令群	ニモニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件			
特殊	GETI ^{注1,2}	taddr	1	3	<ul style="list-style-type: none"> • μ PD750004 TBR命令のとき $PC_{11-0} (taddr)_{8-0} + (taddr + 1)$	* 10				
				4	TCALL命令のとき $(SP - 6) \text{ } \text{ } \text{ } (SP - 3) \text{ } \text{ } (SP - 4) \text{ } \text{ } PC_{11-0}$ $(SP - 5) \text{ } 0, 0, 0, 0$ $(SP - 2) \text{ } \times, \times, \text{MBE, RBE}$ $PC_{11-0} (taddr)_{8-0} + (taddr + 1)$ SP SP - 6					
				3	TBR, TCALL命令以外のとき (taddr)と(taddr + 1)の命令実行			参照した命令による		
				3	<ul style="list-style-type: none"> • μ PD750006, 750008 TBR命令のとき $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$					
				4	TCALL命令のとき $(SP - 6) \text{ } \text{ } \text{ } (SP - 3) \text{ } \text{ } (SP - 4) \text{ } \text{ } PC_{11-0}$ $(SP - 5) \text{ } 0, 0, 0, PC_{12}$ $(SP - 2) \text{ } \times, \times, \text{MBE, RBE}$ $PC_{12-0} (taddr)_{4-0} + (taddr + 1)$ SP SP - 6					
				3	TBR, TCALL命令以外のとき (taddr)と(taddr + 1)の命令実行				参照した命令による	
				3	<ul style="list-style-type: none"> • μ PD75P0016 TBR命令のとき $PC_{13-0} (taddr)_{6-0} + (taddr + 1)$					
				4	TCALL命令のとき $(SP - 6) \text{ } \text{ } \text{ } (SP - 3) \text{ } \text{ } (SP - 4) \text{ } \text{ } PC_{11-0}$ $(SP - 5) \text{ } 0, 0, PC_{13}, PC_{12}$ $(SP - 2) \text{ } \times, \times, \text{MBE, RBE}$ $PC_{13-0} (taddr)_{6-0} + (taddr + 1)$ SP SP - 6					
				3	TBR, TCALL命令以外のとき (taddr)と(taddr + 1)の命令実行					参照した命令による

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 モード時のみに対応しています。

2 . 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。

11.3 各命令の命令コード

(1) 命令コードの記号説明



I_n : n4, n8に対するイミディエト・データ

D_n : memに対するイミディエト・データ

B_n : bitに対するイミディエト・データ

N_n : n, IE x x xに対するイミディエト・データ

T_n : taddr x $\frac{1}{2}$ に対するイミディエト・データ

A_n : {分岐先アドレスとの相対アドレス距離(2 - 16)} - 1に対するイミディエト・データ

S_n : {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 ₁ B ₀ F ₃ F ₂ F ₁ F ₀	FB0H - FBFHの操作可能ビット
	1 1 B ₁ B ₀ F ₃ F ₂ F ₁ F ₀	FF0H - FFFHの操作可能ビット
pmem. @L	0 1 0 0 G ₃ G ₂ G ₁ G ₀	FC0H - FFFHの操作可能ビット
@H + mem. bit	0 0 B ₁ B ₀ D ₃ D ₂ D ₁ D ₀	アクセス可能なメモリ・バンクの操作可能ビット

B_n : bitに対するイミューディエト・データ

F_n : fmemに対するイミューディエト・データ

(アドレスの下位4ビットを示す)

G_n : pmemに対するイミューディエト・データ

(アドレスのビット5-2を示す)

D_n : memに対するイミューディエト・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 ₁	B ₂	B ₃
転送	MOV	A, #n4	0 1 1 1 I ₃ I ₂ I ₁ I ₀		
		reg1, #n4	1 0 0 1 1 0 1 0	I ₃ I ₂ I ₁ I ₀ 1 R ₂ R ₁ R ₀	
		rp, #n8	1 0 0 0 1 P ₂ P ₁ 1	I ₇ I ₆ I ₅ I ₄ I ₃ I ₂ I ₁ I ₀	
		A,@rpa1	1 1 1 0 0 Q ₂ Q ₁ Q ₀		
		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 ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		XA, mem	1 0 1 0 0 0 1 0	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ 0	
		mem, A	1 0 0 1 0 0 1 1	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		mem, XA	1 0 0 1 0 0 1 0	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ 0	
		A, reg	1 0 0 1 1 0 0 1	0 1 1 1 1 R ₂ R ₁ R ₀	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 1 1 P ₂ P ₁ P ₀	
		reg1, A	1 0 0 1 1 0 0 1	0 1 1 1 0 R ₂ R ₁ R ₀	
	rp' 1, XA	1 0 1 0 1 0 1 0	0 1 0 1 0 P ₂ P ₁ P ₀		
	XCH	A, @rpa1	1 1 1 0 1 Q ₂ Q ₁ Q ₀		
		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 ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
XA, mem		1 0 1 1 0 0 1 0	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ 0		
A, reg1		1 1 0 1 1 R ₂ R ₁ R ₀			
XA, rp'		1 0 1 0 1 0 1 0	0 1 0 0 0 P ₂ P ₁ P ₀		
テーブル参照	MOV _T	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	MOV ₁	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 ₁	B ₂	B ₃
演算	ADDS	A, #n4	0 1 1 0 I ₃ I ₂ I ₁ I ₀		
		XA, #n8	1 0 1 1 1 0 0 1	I ₇ I ₆ I ₅ I ₄ I ₃ I ₂ I ₁ I ₀	
		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 ₂ P ₁ P ₀	
		rp' 1, XA	1 0 1 0 1 0 1 0	1 1 0 0 0 P ₂ P ₁ P ₀	
	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 ₂ P ₁ P ₀	
		rp' 1, XA	1 0 1 0 1 0 1 0	1 1 0 1 0 P ₂ P ₁ P ₀	
	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 ₂ P ₁ P ₀	
		rp' 1, XA	1 0 1 0 1 0 1 0	1 1 1 0 0 P ₂ P ₁ P ₀	
	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 ₂ P ₁ P ₀	
		rp' 1, XA	1 0 1 0 1 0 1 0	1 1 1 1 0 P ₂ P ₁ P ₀	
	AND	A, #n4	1 0 0 1 1 0 0 1	0 0 1 1 I ₃ I ₂ I ₁ I ₀	
		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 ₂ P ₁ P ₀	
		rp' 1, XA	1 0 1 0 1 0 1 0	1 0 0 1 0 P ₂ P ₁ P ₀	
	OR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 0 I ₃ I ₂ I ₁ I ₀	
		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 ₂ P ₁ P ₀		
rp' 1, XA		1 0 1 0 1 0 1 0	1 0 1 0 0 P ₂ P ₁ P ₀		
XOR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 1 I ₃ I ₂ I ₁ I ₀		
	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 ₂ P ₁ P ₀		
	rp' 1, XA	1 0 1 0 1 0 1 0	1 0 1 1 0 P ₂ P ₁ P ₀		
操作	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 ₁	B ₂	B ₃
増 減	INCS	reg	1 1 0 0 0 R ₂ R ₁ R ₀		
		rp1	1 0 0 0 1 P ₂ P ₁ 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 ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
	DECS	reg	1 1 0 0 1 R ₂ R ₁ R ₀		
		rp'	1 0 1 0 1 0 1 0	0 1 1 0 1 P ₂ P ₁ P ₀	
比 較	SKE	reg, #n4	1 0 0 1 1 0 1 0	I ₃ I ₂ I ₁ I ₀ 0 R ₂ R ₁ R ₀	
		@HL, #n4	1 0 0 1 1 0 0 1	0 1 1 0 I ₃ I ₂ I ₁ I ₀	
		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 ₂ R ₁ R ₀	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 0 1 P ₂ P ₁ P ₀	
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 ₁ B ₀ 0 1 0 1	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		[*1]	1 0 0 1 1 1 0 1	* 2	
	CLR1	mem. bit	1 0 B ₁ B ₀ 0 1 0 0	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		[*1]	1 0 0 1 1 1 0 0	* 2	
	SKT	mem. bit	1 0 B ₁ B ₀ 0 1 1 1	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		[*1]	1 0 1 1 1 1 1 1	* 2	
	SKF	mem. bit	1 0 B ₁ B ₀ 0 1 1 0	D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	
		[*1]	1 0 1 1 1 1 1 0	* 2	
	SKTCLR	[*1]	1 0 0 1 1 1 1 1	* 2	
	AND1	CY, [*1]	1 0 1 0 1 1 0 0	* 2	
OR1	CY, [*1]	1 0 1 0 1 1 1 0	* 2		
XOR1	CY, [*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 ₁	B ₂	B ₃	
分岐	BR	! addr	1 0 1 0 1 0 1 1	0 0 ←————→	addr —————→	
		$\left. \begin{matrix} (+16) \\ \$ \text{addr1} \\ \left. \begin{matrix} (+2) \\ (-1) \\ (-15) \end{matrix} \right\} \end{matrix} \right\}$	0 0 0 0 A ₃ A ₂ A ₁ A ₀			
			1 1 1 1 S ₃ S ₂ S ₁ S ₀			
		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		
	BRA	! addr1	1 0 1 1 1 0 1 0	0 ←————→	addr1 —————→	
	BRCB	! caddr	0 1 0 1 ←————→	caddr —————→		
サブルーチン・スタック制御	CALL	! addr	1 0 1 0 1 0 1 1	0 1 ←————→	addr —————→	
		! addr1	1 0 1 1 1 0 1 1	0 ←————→	addr1 —————→	
	CALLF	! faddr	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 ₂ P ₁ 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 ₂ P ₁ 0			
	BS		1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 0		
入出力	IN	A, PORT _n	1 0 1 0 0 0 1 1	1 1 1 1 N ₃ N ₂ N ₁ N ₀		
		XA, PORT _n	1 0 1 0 0 0 1 0	1 1 1 1 N ₃ N ₂ N ₁ N ₀		
	OUT	PORT _n , A	1 0 0 1 0 0 1 1	1 1 1 1 N ₃ N ₂ N ₁ N ₀		
		PORT _n , XA	1 0 0 1 0 0 1 0	1 1 1 1 N ₃ N ₂ N ₁ N 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 ₅ 1 1 N ₂ N ₁ N ₀		
	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 ₅ 1 1 N ₂ N ₁ N ₀		
CPU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 _n	1 0 0 1 1 0 0 1	0 0 1 0 0 0 N ₁ N ₀		
		MB _n	1 0 0 1 1 0 0 1	0 0 0 1 N ₃ N ₂ N ₁ N ₀		
	GETI	taddr	0 0 T ₅ T ₄ T ₃ T ₂ T ₁ T ₀			

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。μPD750004, 750006, 750008, 75P0016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 PD750004	}	のMk モード時とMk モード時
μ PD750006		
μ PD750008		
μ PD75P0016		

Ⅰ μPD750004, 750006, 750008, 75P0016のMk モード時のみ使用できます。

Ⅱ μPD750004, 750006, 750008, 75P0016のMk モード時のみ使用できます。

Ⅰ/Ⅱ μPD750004, 750006, 750008, 75P0016のMk モード時, 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μPD750006, 750008を例として説明しています。μPD750004はプログラム・カウンタが12ビット、μPD75P0016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 = 13-0 : 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 = 13-0 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 = 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

○ MOV A, @HL +

○ MOV A, @HL -

○ MOV A, @rpa1

機能：A (オペランドで指定されたレジスタ・ペア)

レジスタ・ペアにHL + を指定したとき：skip if L = 0

レジスタ・ペアにHL - を指定したとき：skip if L = FH

指定されたレジスタ・ペア (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。

レジスタ・ペアに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。

レジスタ・ペアに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
- XCH A, @HL +
- XCH A, @HL -
- XCH A, @rpa1

機能 : $A \leftrightarrow$ (オペランドで指定されたレジスタ・ペア)

レジスタ・ペアにHL+を指定したとき : skip if L = 0

レジスタ・ペアにHL-を指定したとき : skip if L = FH

Aレジスタの内容と、指定されたレジスタ・ペア(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。

レジスタ・ペアにオートインクリメント(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し、その結果が0になると続く1命令をスキップします。

レジスタ・ペアに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し、その結果が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×)
      XCH A, @DL    ; A ↔ (2×)
      XCH A, @HL +  ; A ↔ (3×)
BR   LOOP

```

- XCH XA, @HL

機能 : $A \leftrightarrow (HL), X \leftrightarrow (HL + 1)$

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 : $A \leftrightarrow (\text{mem})$ $\text{mem} = D7-0 : 00H - FEH$

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ます。

○ XCH XA, mem

機能 : $A \leftrightarrow (\text{mem}), X \leftrightarrow (\text{mem} + 1)$ $\text{mem} = D7-0 : 00H - FEH$

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 : $A \leftrightarrow \text{reg1}$

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 : $XA \leftrightarrow \text{rp}'$

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

○ MOVT XA, @PCDE

機能： μ PD750006, 750008の場合

XA ROM (PC₁₂₋₈ + DE)

プログラム・カウンタ (PC) の下位 8 ビット (PC₇₋₀) 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。

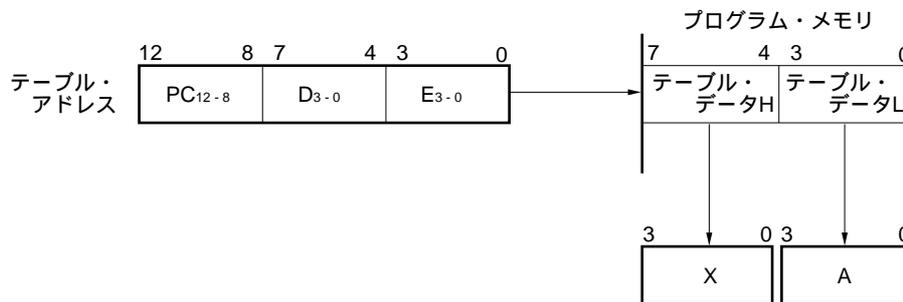
テーブル・アドレスはこの命令実行時のプログラム・カウンタ (PC) の内容によって決定されます。

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。

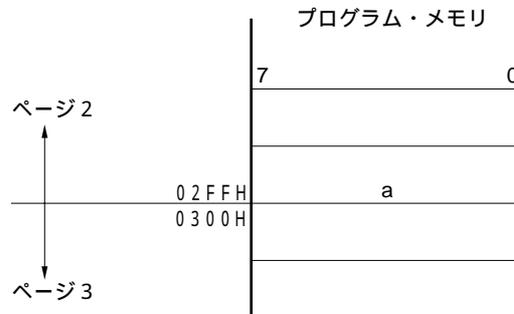
例 μ PD750006, 750008の場合



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 μ PD750006, 750008を例として説明しています。 μ PD750004はプログラム・カウンタが12ビット、 μ PD75P0016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 x F0H
DB   x x H, x x H, ..... ; テーブル・データ
    
```

○ MOV_T XA, @PCXA

機能 : μ PD750006, 750008の場合
XA ROM (PC₁₂₋₈ + XA)

プログラム・カウンタ (PC) の下位 8 ビット (PC₇₋₀) 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_T XA, @PCDEと同様に, 命令が $x \times \text{FFH}$ 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 μ PD750006, 750008を例として説明しています。 μ PD750004はプログラム・カウンタが12ビット, μ PD75P0016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。

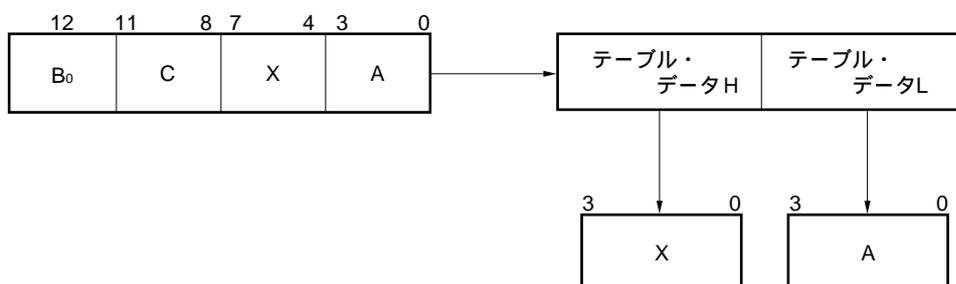
○ MOVT XA, @BCXA

機能：μPD750006, 750008の場合

XA (BCXA) ROM

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。



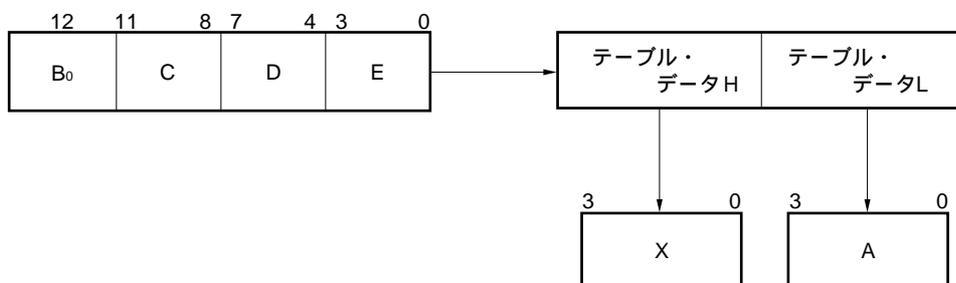
○ MOVT XA, @BCDE

機能：μPD750006, 750008の場合

XA (BCDE) ROM

レジスタBの下位3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。



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μPD750006, 750008を例として説明しています。μPD750004はプログラム・カウンタが12ビット, μPD75P0016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 = 13-0 : 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 特徴的な命令を参照)。

○ ADDS XA, #n8

機能 : XA XA + n8 ; Skip if carry. n8 = 17-0 : 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 : $rp'1 \leftarrow rp'1 + XA$; 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 : $A, CY \leftarrow A + (HL) + CY$

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 特徴的な命令を参照)。

○ ADDC XA, rp'

機能 : $XA, CY \leftarrow XA + rp' + CY$

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。

○ 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。

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 特徴的な命令を参照)。

○ 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 AND n4 n4 = 13-0 : 0 - FH

Aレジスタの内容と4ビット・イミディエイト・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 AND (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 AND 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 AND 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 OR n4 n4 = 13-0 : 0-FH

Aレジスタの内容と4ビット・イミディエイト・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 OR (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 OR 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 OR 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: $A \leftarrow A \nabla n4$ $n4 = 13-0:0-FH$

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: $A \leftarrow A \nabla (HL)$

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: $XA \leftarrow XA \nabla rp'$

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: $rp'1 \leftarrow rp'1 \nabla XA$

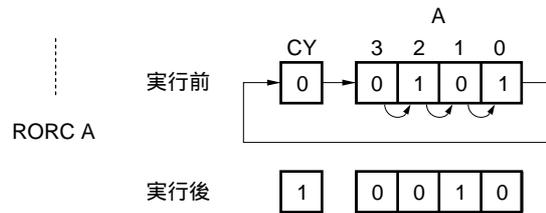
レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容とレジスタ・ペアXAの内容との排他的論理和をとり、結果をレジスタ・ペアrp'1にセットします。

11.4.5 アキュムレータ操作命令

○ RORC A

機能 : CY A₀, A_{n-1} A_n, A₃ 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・フラグも含めて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 = l3-0 : 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・イミューディエト・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 ∇ (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

I BR addr

機能 : μ PD750008の場合 PC₁₂₋₀ addr
addr = 0000H - 1FFFH

イミディエト・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。

II BR addr1

機能 : μ PD750008の場合 PC₁₂₋₀ addr1
addr1 = 0000H - 1FFFH

イミディエト・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。

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 PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット、addr = 0000H-0FFFH、 μ PD750006はプログラム・カウンタが13ビット、addr = 0000H-17FFH、 μ PD75P0016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。

Ⅱ BRA ! addr1

機能：μPD750008の場合 PC₁₂₋₀ addr1

○ BR ! addr

機能：μPD750008の場合 PC₁₂₋₀ addr
addr = 0000H - 1FFFFH

イミディエイト・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。

Ⅰ BR \$ addr

機能：μPD750008の場合 PC₁₂₋₀ 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。

Ⅱ BR \$ addr1

機能：μPD750008の場合 PC₁₂₋₀ addr1
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。

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μPD750008を例として説明してあります。μPD750004はプログラム・カウンタが12ビット、addr = 0000H-0FFFFH、μPD750006はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、μPD75P0016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。

○ BRCB ! caddr

機能： μ PD750008の場合 PC_{12-0} $PC_{12} + caddr_{11-0}$
 $caddr = n000H-nFFFH$
 $n = PC_{12} = 0, 1$

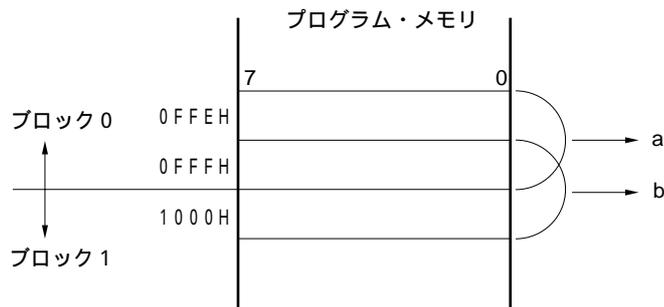
プログラム・カウンタの下位12ビット (PC_{11-0}) を12ビット・イミューディエト・データ $caddr$ (A_{11-0}) で置き換えた番地に分岐します。

μ PD750004はプログラム・カウンタが11ビットの構成なので、この命令で全空間に分岐できます。

μ PD750006, 750008は PC_{12} を, μ PD75P0016は $PC_{12,13}$ を変えることはできず、ブロック内の分岐となります。

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_H番地、または0FFF_H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 μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット, $addr = 0000H-0FFFH$, μ PD750006はプログラム・カウンタが13ビット, $addr = 0000H-17FFH$, μ PD75P0016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。

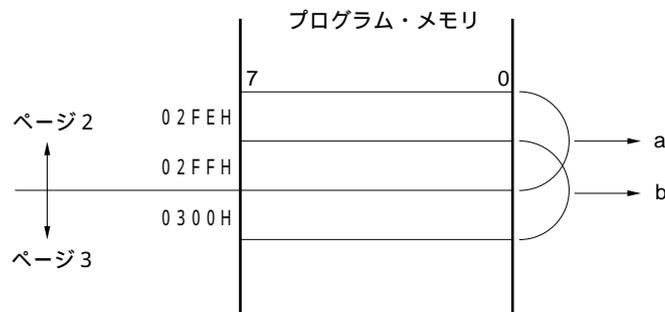
○ BR PCDE

機能: μ PD750008の場合 PC₁₂₋₀ PC₁₂₋₈ + DE
PC₇₋₄ D, PC₃₋₀ E

プログラム・カウンタの下位8ビット(PC₇₋₀)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 $\times \times$ FE番地、または $\times \times$ FFH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: μ PD750008の場合 PC₁₂₋₀ PC₁₂₋₈ + XA
PC₇₋₄ X, PC₃₋₀ A

プログラム・カウンタの下位8ビット(PC₇₋₀)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

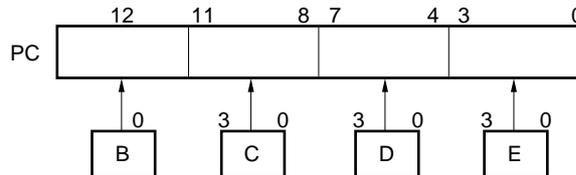
BR PCDE命令と同様に、1バイト目が $\times \times$ FEH、または $\times \times$ FFH番地にある場合は同一ページ内には分岐せず、次のページに分岐します。

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 μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット、addr = 0000H-0FFFH、 μ PD750006はプログラム・カウンタが13ビット、addr = 0000H-17FFH、 μ PD75P0016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：μPD750008の場合 PC₁₂₋₀ BCDE

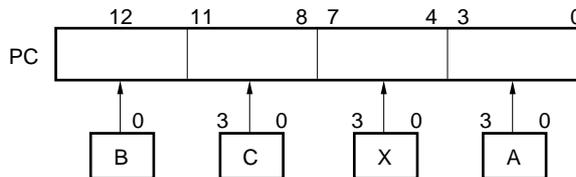
プログラム・カウンタをレジスタB₀, C, D, Eの内容で置き換えた番地に分岐します。



○ BR BCXA

機能：μPD750008の場合 PC₁₂₋₀ BCXA

プログラム・カウンタをレジスタB₀, 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μPD750008を例として説明してあります。μPD750004はプログラム・カウンタが12ビット、addr = 0000H-0FFFH、μPD750006はプログラム・カウンタが13ビット、addr = 0000H-17FFH、μPD75P0016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

② CALLA !addr1

機能: μ PD750008の場合(SP - 2) x, x, MBE, RBE, (SP - 3) PC₇₋₄(SP - 4) PC₃₋₀, (SP - 5) 0, 0, 0, PC₁₂(SP - 6) PC₁₁₋₈PC₁₂₋₀ addr1, SP SP - 6

①/② CALL !addr

機能: μ PD750008の場合

[Mk モード]

(SP - 1) PC₇₋₄, (SP - 2) PC₃₋₀(SP - 3) MBE, RBE, 0, PC₁₂(SP - 4) PC₁₁₋₈, PC₁₂₋₀ addr, SP SP - 4

addr = 0000H - 1FFFH

[Mk モード]

(SP - 2) x, x, MBE, RBE

(SP - 3) PC₇₋₄, (SP - 4) PC₃₋₀(SP - 5) 0, 0, 0, PC₁₂, (SP - 6) PC₁₁₋₈PC₁₂₋₀ addr, SP SP - 6

addr = 0000H - 1FFFH

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・イメージ・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-1FFFHの μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット, addr = 0000H-0FFFH, μ PD750006はプログラム・カウンタが13ビット, addr = 0000H-17FFH, μ PD75P0016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 CALLF ! faddr

機能：μPD750008の場合

[Mk モード]

(SP - 1) PC₇₋₄, (SP - 2) PC₃₋₀

(SP - 3) MBE, RBE, 0, PC₁₂

(SP - 4) PC₁₁₋₈, SP SP - 4

PC₁₂₋₀ 00 + faddr

faddr = 0000H - 07FFH

[Mk モード]

(SP - 2) x, x, MBE, RBE

(SP - 3) PC₇₋₄, (SP - 4) PC₃₋₀

(SP - 5) 0, 0, 0, PC₁₂, (SP - 6) PC₁₁₋₈

SP SP - 6

PC₁₂₋₀ 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・イミューディエト・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 μPD750008 を例として説明してあります。μPD750004 はプログラム・カウンタが 12 ビット, addr = 0000H-0FFFH, μPD750006 はプログラム・カウンタが 13 ビット, addr = 0000H-17FFFH, μPD75P0016 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。

Ⅰ/Ⅱ RET

機能: μ PD750008の場合

```
[ Mk モード ] PC11-8 (SP)
                MBE, RBE, 0, PC12 (SP + 1)
                PC3-0 (SP + 2)
                PC7-4 (SP + 3), SP SP + 4

[ Mk モード ] PC11-8 (SP), x, x, x, PC12 (SP + 1)
                PC3-0 (SP + 2), PC7-4 (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 以外に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, $\text{addr} = 0000\text{H} - 1\text{FFFH}$ の μ PD750008 を例として説明してあります。 μ PD750004 はプログラム・カウンタが12ビット, $\text{addr} = 0000\text{H} - 0\text{FFFH}$, μ PD750006 はプログラム・カウンタが13ビット, $\text{addr} = 0000\text{H} - 17\text{FFH}$, μ PD75P0016 はプログラム・カウンタが14ビット, $\text{addr} = 0000\text{H} - 3\text{FFFH}$ ですので, それぞれ読み替えてお使いください。

I/II RETS

機能： μ PD750008の場合

```
[ Mk モード ] PC11-8 ( SP )
                MBE , 0 , 0 , PC12 ( SP + 1 )
                PC3-0 ( SP + 2 ) , PC7-4 ( SP + 3 ) , SP SP + 4
                Then skip unconditionally

[ Mk モード ] PC11-8 ( SP ) , 0 , 0 , 0 , PC12 ( SP + 1 )
                PC3-0 ( SP + 2 ) , PC7-4 ( 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, $\text{addr} = 0000\text{H}-1\text{FFFH}$ の μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット, $\text{addr} = 0000\text{H}-0\text{FFFH}$, μ PD750006はプログラム・カウンタが13ビット, $\text{addr} = 0000\text{H}-17\text{FFFH}$, μ PD75P0016はプログラム・カウンタが14ビット, $\text{addr} = 0000\text{H}-3\text{FFFH}$ ですので, それぞれ読み替えてお使いください。

Ⓜ RETI

機能： μ PD750008の場合

[Mk モード] PC₁₁₋₈ (SP), MBE, RBE, 0, PC₁₂ (SP + 1)
 PC₃₋₀ (SP + 2)
 PC₇₋₄ (SP + 3)
 PSW_L (SP + 4), PSW_H (SP + 5)
 SP SP + 6

[Mk モード] PC₁₁₋₈ (SP), 0, 0, 0, PC₁₂ (SP + 1)
 PC₃₋₀ (SP + 2)
 PC₇₋₄ (SP + 3)
 PSW_L (SP + 4), PSW_H (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 μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット, addr = 0000H-0FFFH, μ PD750006はプログラム・カウンタが13ビット, addr = 0000H-17FFH, μ PD75P0016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) r_{pH}, (SP - 2) r_{pL}, 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 (r_{pH}: X, H, D, B) は (SP - 1) で, 下位側 (r_{pL}: 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: rpL (SP), rpH (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 (rpH: A, L, E, C) には (SP) の内容が, 上位側 (rpH: 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 × × × = N₅, N₂₋₀

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 × × × = N₅, N₂₋₀

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 $n = N_{3-0} : 0-8$

PORTn ($n = 0-8$) 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0-8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 PORT $n+1$ $n = N_{3-0} : 4, 6$

PORTn ($n = 4, 6$) 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, 6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 $n = N_{3-0} : 2-8$

Aレジスタの内容をPORTn ($n = 2-8$) 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2-8のみ指定できます。

○ OUT PORT_n, XA

機能：PORT_n A, PORT_{n+1} X n = N₃₋₀: 4, 6

Aレジスタの内容をPORT_n (n = 4, 6) 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, 6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 RBn

機能: RBS n n = N₁₋₀: 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 MBn

機能: MBS n n = N₃₋₀: 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。
nとしては, 0, 1, 15のみ指定できます。

I/II GETI taddr

機能 : taddr = T₅₋₀ , 0 : 20H - 7FH

μ PD750008の場合

[Mk モード]

TBR命令で定義したテーブルを参照した場合

PC₁₂₋₀ (taddr)₄₋₀ + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合

(SP - 1) PC₇₋₄ , (SP - 2) PC₃₋₀

(SP - 3) MBE , RBE , 0 , PC₁₂

(SP - 4) PC₁₁₋₈

PC₁₂₋₀ (taddr)₄₋₀ + (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

[Mk モード]

TBR命令で定義したテーブルを参照した場合

PC₁₂₋₀ (taddr)₄₋₀ + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合

(SP - 2) x , x , MBE , RBE

(SP - 3) PC₇₋₄ , (SP - 4) PC₃₋₀

(SP - 5) 0 , 0 , 0 , PC₁₂ , (SP - 6) PC₁₁₋₈

PC₁₂₋₀ (taddr)₄₋₀ + (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

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 μ PD750008を例として説明してあります。 μ PD750004はプログラム・カウンタが12ビット、addr = 0000H-0FFFFH、 μ PD750006はプログラム・カウンタが13ビット、addr = 0000H-17FFFH、 μ PD75P0016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。

(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。

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	{ INCS L DECS L INCS H DECS H INCS HL
MOV A, @DE XCH A, @DE	{ INCS E DECS E INCS D DECS D INCS DE
MOV A, @DL XCH A, @DL	{ 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有する場合、GETI 命令はほかの 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有する場合、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有する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有する命令の場合、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有する命令の場合、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	}	をGETI命令に置き換える。
	MOV XA, #FFH		
	CALL SUB1		
	BR SUB2		

```

                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
    
```

付録A μ PD75008 , 750008 , 75P0016の機能一覧表

(1/2)

項 目		μ PD75008	μ PD750008	μ PD75P0016
プログラム・メモリ		マスクROM 0000H - 1F7FH (8064 × 8 ビット)	マスクROM 0000H - 1FFFFH (8192 × 8 ビット)	ワン・タイムPROM 0000H - 3FFFFH (16384 × 8 ビット)
データ・メモリ		000H - 1FFFH (512 × 4 ビット)		
CPU		75X Standard CPU	75XL CPU (75X High End CPUに相当)	
発振安定時間		31.3 ms	$2^{15}/f_x$, $2^{17}/f_x$ (マスク・オプションで選択可能)	$2^{15}/f_x$ に固定
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 μ s (4.19 MHz動作時)	<ul style="list-style-type: none"> 0.95, 1.91, 3.81, 15.3 μs (4.19 MHz動作時) 0.67, 1.33, 2.67, 10.7 μs (6.0 MHz動作時) 	
	サブシステム・クロック選択時	122 μ s (32.768 kHz動作時)		
端子 接 続	20ピン (CU)	NC	IC	V_{PP}
	38ピン (GB)			
	24ピン (CU)	P21	P21/PTO1	
	42ピン (GB)			
	6-9ピン (CU)	P33 - P30		P33/MD3 - P30/MD0
23-26ピン (GB)				
スタ ック	SBSレジスタ	なし	あり	<ul style="list-style-type: none"> SBS.3 = 1 : Mk モード選択 SBS.3 = 0 : Mk モード選択
	スタック・エリア	000H - 0FFFH	n00H - nFFFH (n=0, 1)	
	サブルーチン・コール命令 のスタック動作	2 バイト・スタック	<ul style="list-style-type: none"> Mk モード時: 2 バイト・スタック Mk モード時: 3 バイト・スタック 	
命 令	BRA !addr1	使用不可	Mk モード時: 使用不可	
	CALLA !addr1		Mk モード時: 使用可能	
	MOVT XA, @BCDE		使用可能	
	MOVT XA, @BCXA			
	BR BCDE			
	BR BCXA			
CALL !addr	3 マシン・サイクル	Mk モード時: 3 マシン・サイクル, Mk モード時: 4 マシン・サイクル		
CALLF !addr	2 マシン・サイクル	Mk モード時: 2 マシン・サイクル, Mk モード時: 3 マシン・サイクル		

(2/2)

項 目	μPD75008	μPD750008	μPD75P0016
タイマ	3チャンネル ・ベーシック・インター バル・タイマ：1チャンネル ・タイマ/イベント・カウ ンタ：1チャンネル ・時計用タイマ：1チャンネル	4チャンネル ・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 ・タイマ/イベント・カウン タ：1チャンネル ・タイマ・カウンタ：1チャンネル ・時計用タイマ：1チャンネル	
クロック出力 (PCL)	Φ, 524, 262, 65.5 kHz (メイン・システム・クロック ：4.19 MHz動作時)	・Φ, 524, 262, 65.5 kHz (メイン・システム・クロック：4.19 MHz動作時) ・Φ, 750, 375, 93.7 kHz (メイン・システム・クロック：6.0 MHz動作時)	
BUZ出力 (BUZ)	2 kHz	・2, 4, 32 kHz (メイン・システム・クロック：4.19 MHz動作時) ・2.86, 5.72, 45.8 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/Oモード...MSB/LSB先頭切り替え可能 ・2線式シリアル/Oモード ・SBIモード		
S O S レ ジ ス タ	フィードバック抵抗カット・フラグ (SOS.0)	マスク・オプションでフィ ードバック抵抗内蔵可能	内蔵
	サブ発振器電流カット・フラグ (SOS.1)	なし	内蔵
レジスタ・バンク選択レジスタ (RBS)	なし	あり	
INT0によるスタンバイ解除	不可	可能	
ベクタ割り込み	外部：3本, 内部：3本	外部：3本, 内部：4本	
プロセッサ・クロック・コントロール・レジスタ	PCC = 0, 2, 3 使用可能	PCC = 0 ~ 3 使用可能	
電源電圧	V _{DD} = 2.7 ~ 6.0 V	V _{DD} = 2.2 ~ 5.5V	
動作周囲温度	T _A = - 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)		

付録 B 開発ツール

μ PD750008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。

RA75X リロケータブル・アセンブラ	ホスト・マシン	OS	供給媒体	オーダ名称(品名)
		PC-9800シリーズ	MS-DOS (Ver.3.30 } Ver.6.2 ^注)	
IBM PC/AT TM 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	

デバイス・ファイル	ホスト・マシン	OS	供給媒体	オーダ名称(品名)
		PC-9800シリーズ	MS-DOS (Ver.3.30 } Ver.6.2 ^注)	
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 S7B13DF750008	

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・アローンまたは、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-75P008CU	μ PD75P008CU/GBと兼用のμ PD75P0016CU/GB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	OS	供給媒体	オーダ名称（品名）
		PC-9800シリーズ	MS-DOS { Ver.3.30 } Ver.6.2 ^注	3.5インチ2HD	
		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

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

μ PD750008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 ^{注1}	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。μ PD750008サブシリーズを開発する場合、IE-75000-Rと別売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て使用します。ホスト・マシン、PROMプログラマ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。別売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・プローブと組み合わせて使用します。ホスト・マシン、PROMプログラマ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 ^{注2}	μ PD750008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と組み合わせて使用します。		
	EP-75008GB-R EV-9200G-44	μ PD75008GBと兼用のμ PD750008GB用エミュレーション・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。ターゲット・システムとの接続を容易にする44ピン変換ソケットEV-9200G-44を添付しています。		
	EP-75008CU-R	μ PD75008CUと兼用のμ PD750008CU用エミュレーション・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。		
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	OS	供給媒体	オーダ名称 (品名)
	PC-9800シリーズ	MS-DOS (Ver.3.30) Ver.6.2 ^{注3}	3.5インチ2HD	μ S5A13IE75X
	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 S7B13IE75X

注1 . 保守品です。

2 . 別売りです。

3 . 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。

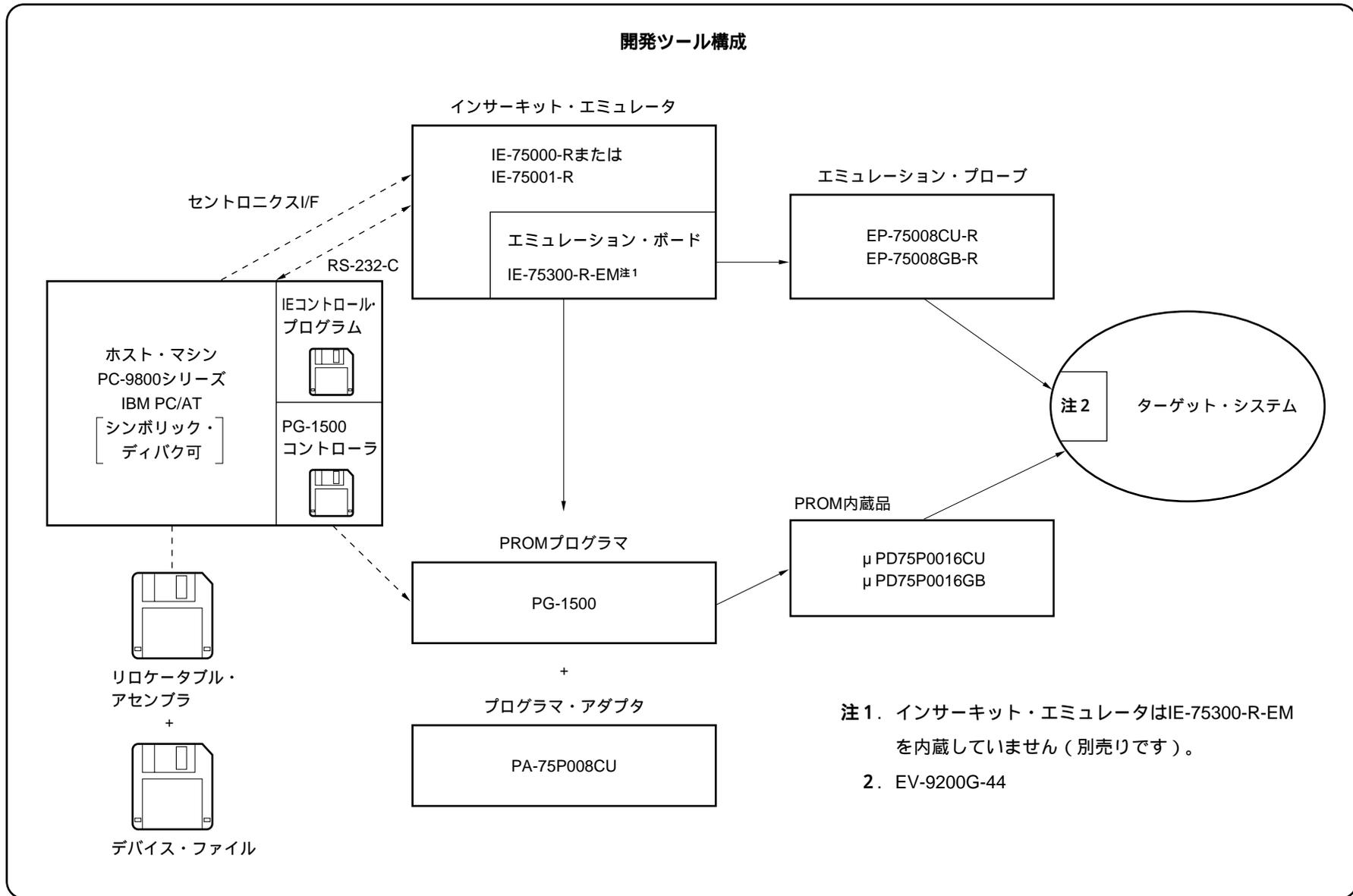
IBM PC用のOSについて

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.3.1 ~ Ver.6.3 J6.1/V ^注 ~ J6.3/V ^注
MS-DOS	Ver.5.0 ~ Ver.6.22 5.0/V ^注 ~ 6.2/V ^注
IBM DOS™	J5.02/V ^注

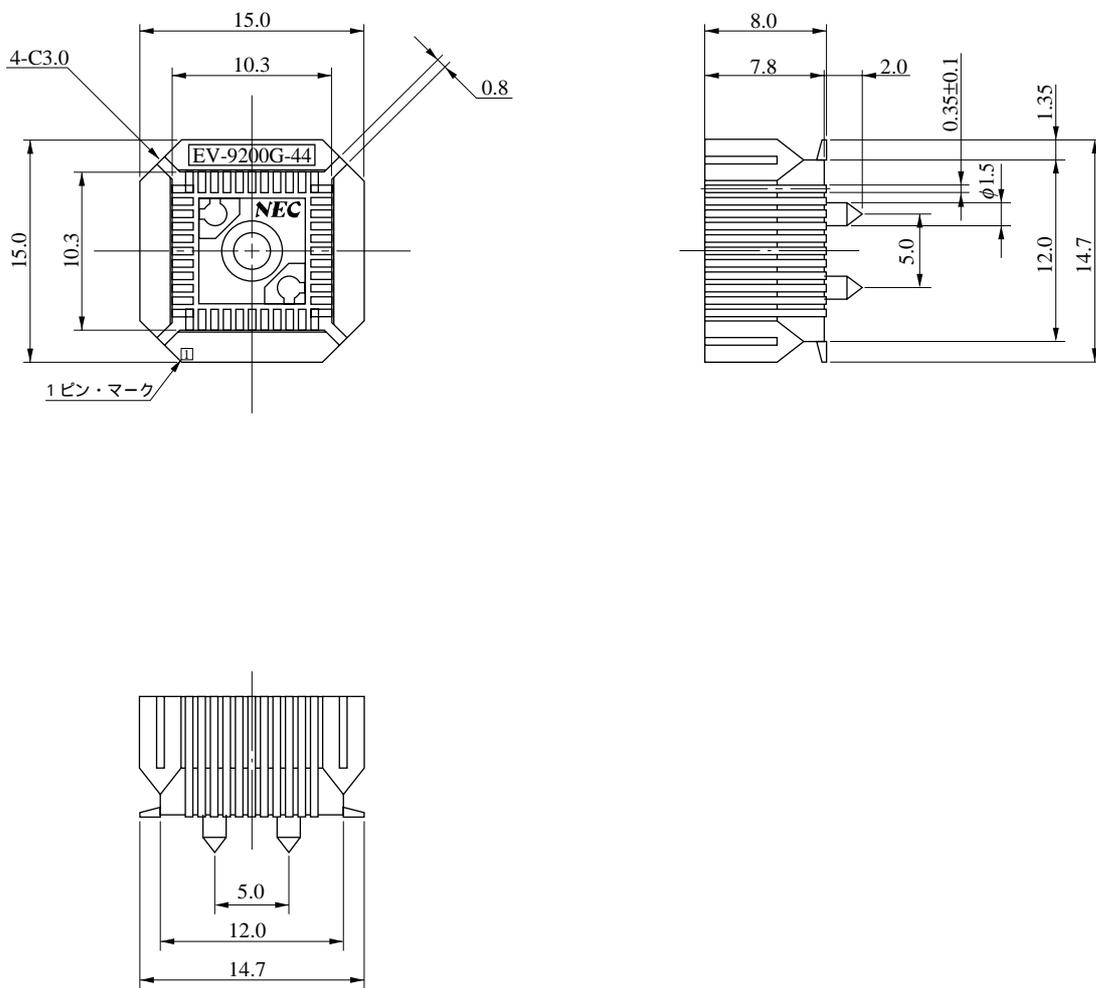
注 英語モードのみサポートしています。

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。



変換ソケット (EV-9200G-44) の外形図と基板取り付け推奨パターン

図B - 1 EV-9200G-44 外形図 (参考) (単位 : mm)

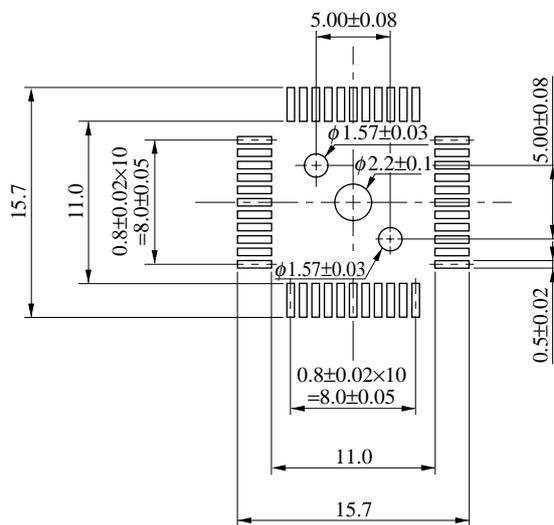


EV-9200G-44-G0

図B - 2 EV-9200G-44 基板取り付け推奨パターン（参考）（単位：mm）

参考図：EV-9200G-44

(2) パッド図（単位：mm）



EV-9200G-44-P1

注意 EV-9200用のマウント・パッド寸法と、対象製品のマウント・パッド寸法（QFP用）は、その一部が異なる場合があります。QFP用の推奨マウント・パッド寸法は、「半導体デバイス 実装マニュアル, C10535J」をご参照ください。

(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発注用の媒体は、UV-EPROMまたは3.5インチもしくは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）です。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（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、提出してください）。

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 IEM-834]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 ... 281, 303
MOV reg1, #n4 ... 281, 303
MOV XA, #n8 ... 281, 303
MOV HL, #n8 ... 281, 304
MOV rp2, #n8 ... 281, 304
MOV A, @HL ... 281, 304
MOV A, @HL + ... 281, 304
MOV A, @HL - ... 281, 304
MOV A, @rpa1 ... 281, 304
MOV XA, @HL ... 281, 305
MOV @HL, A ... 281, 305
MOV @HL, XA ... 281, 305
MOV A, mem ... 281, 305
MOV XA, mem ... 281, 306
MOV mem, A ... 281, 306
MOV mem, XA ... 281, 306
MOV A, reg ... 281, 306
MOV XA, rp' ... 281, 307
MOV reg1, A ... 281, 307
MOV rp'1 XA ... 281, 307
XCH A, @HL ... 281, 308
XCH A, @HL + ... 281, 308
XCH A, @HL - ... 281, 308
XCH A, @rpa1 ... 281, 308
XCH XA, @HL ... 281, 308
XCH A, mem ... 281, 309
XCH XA, mem ... 281, 309
XCH A, reg1 ... 281, 309
XCH XA, rp' ... 281, 309

[テーブル参照命令]

MOVT XA, @PCDE ... 282, 310
MOVT XA, @PCXA ... 282, 312
MOVT XA, @BCDE ... 282, 313
MOVT XA, @BCXA ... 282, 313

[ビット転送命令]

MOV1 CY, fmem.bit ... 282, 314
MOV1 CY, pmem.@L ... 282, 314
MOV1 CY, @H + mem.bit ... 282, 314
MOV1 fmem.bit, CY ... 282, 314
MOV1 pmem.@L, CY ... 282, 314
MOV1 @H + mem.bit, CY ... 282, 314

[演算命令]

ADDS A, #n4 ... 282, 315
ADDS XA, #n8 ... 282, 315
ADDS A, @HL ... 282, 315
ADDS XA, rp' ... 282, 315
ADDS rp'1, XA ... 282, 316
ADDC A, @HL ... 282, 316
ADDC XA, rp' ... 282, 316
ADDC rp'1, XA ... 282, 317
SUBS A, @HL ... 282, 317
SUBS XA, rp' ... 282, 317
SUBS rp'1, XA ... 282, 318
SUBC A, @HL ... 282, 318
SUBC XA, rp' ... 282, 318
SUBC rp'1, XA ... 282, 318
AND A, #n4 ... 283, 319
AND A, @HL ... 283, 319

AND XA , rp' ... 283, 319
 AND rp'1 , XA ... 283, 319
 OR A , #n4 ... 283, 320
 OR A , @HL ... 283, 320
 OR XA , rp' ... 283, 320
 OR rp'1 , XA ... 283, 320
 XOR A , #n4 ... 283, 321
 XOR A , @HL ... 283, 321
 XOR XA , rp' ... 283, 321
 XOR rp'1 , XA ... 283, 321

[アキュムレータ操作命令]

RORC A ... 283, 322
 NOT A ... 283, 322

[増減命令]

INCS reg ... 283, 322
 INCS rp1 ... 283, 322
 INCS @HL ... 283, 323
 INCS mem ... 283, 323
 DECS reg ... 283, 323
 DECS rp' ... 283, 323

[比較命令]

SKE reg , #n4 ... 283, 323
 SKE @HL , #n4 ... 283, 324
 SKE A , @HL ... 283, 324
 SKE XA , @HL ... 283, 324
 SKE A , reg ... 283, 324
 SKE XA , rp' ... 283, 324

[キャリー・フラグ操作命令]

SET1 CY ... 283, 325
 CLR1 CY ... 283, 325
 SKT CY ... 283, 325
 NOT1 CY ... 283, 325

[メモリ・ビット操作命令]

SET1 mem.bit ... 284, 325
 SET1 fmem.bit ... 284, 326
 SET1 pmem.@L ... 284, 326
 SET1 @H + mem.bit ... 284, 326
 CLR1 mem.bit ... 284, 326
 CLR1 fmem.bit ... 284, 326
 CLR1 pmem.@L ... 284, 326
 CLR1 @H + mem.bit ... 284, 326
 SKT mem.bit ... 284, 326
 SKT fmem.bit ... 284, 327
 SKT pmem.@L ... 284, 327
 SKT @H + mem.bit ... 284, 327
 SKF mem.bit ... 284, 327
 SKF fmem.bit ... 284, 327
 SKF pmem.@L ... 284, 327
 SKF @H + mem.bit ... 284, 327
 SKTCLR fmem.bit ... 284, 328
 SKTCLR pmem.@L ... 284, 328
 SKTCLR @H + mem.bit ... 284, 328
 AND1 CY , fmem.bit ... 284, 328
 AND1 CY , pmem.@L ... 284, 328
 AND1 CY , @H + mem.bit ... 284, 328
 OR1 CY , fmem.bit ... 284, 328
 OR1 CY , pmem.@L ... 284, 328
 OR1 CY , @H + mem.bit ... 284, 328
 XOR1 CY , fmem.bit ... 284, 329
 XOR1 CY , pmem.@L ... 284, 329
 XOR1 CY , @H + mem.bit ... 284, 329

[分岐命令]

BR addr ... 285, 329
 BR addr1 ... 285, 329
 BR ! addr ... 286, 330
 BR \$ addr ... 286, 330
 BR \$ addr1 ... 286, 330
 BR PCDE ... 286, 332

BR PCXA ... 286, 332
BR BCDE ... 287, 333
BR BCXA ... 287, 333
BRA ! addr1 ... 287, 330
BRCB ! caddr ... 287, 331
TBR addr ... 294, 333

[特殊命令]

SEL RBn ... 294, 343
SEL MBn ... 294, 343
GETI taddr ... 294, 344

[サブルーチン・スタック制御命令]

CALLA ! addr1 ... 288, 334
CALL ! addr ... 288, 334
CALLF ! faddr ... 289, 335
TCALL ! addr ... 294, 335
RET ... 290, 336
RETS ... 291, 337
RETI ... 292, 338
PUSH rp ... 292, 338
PUSH BS ... 292, 339
POP rp ... 292, 339
POP BS ... 292, 339

[割り込み制御命令]

EI ... 292, 340
EI IE x x x ... 292, 340
DI ... 292, 340
DI IE x x x ... 292, 340

[入出力命令]

IN A, PORTn ... 293, 341
IN XA, PORTn ... 293, 341
OUT PORTn, A ... 293, 341
OUT PORTn, XA ... 293, 342

[CPU制御命令]

HALT ... 293, 342
STOP ... 293, 342
NOP ... 293, 342

D.2 命令索引 (アルファベット順)

[A]

ADDC A , @HL ... 282, 316
 ADDC rp'1 , XA ... 282, 317
 ADDC XA , rp' ... 282, 316
 ADDS A , #n4 ... 282, 315
 ADDS A , @HL ... 282, 315
 ADDS rp'1 , XA ... 282, 316
 ADDS XA , rp' ... 282, 315
 ADDS XA , #n8 ... 282, 315
 AND A , #n4 ... 283, 319
 AND A , @HL ... 283, 319
 AND rp'1 , XA ... 283, 319
 AND XA , rp' ... 283, 319
 AND1 CY , fmem.bit ... 284, 328
 AND1 CY , pmem.@L ... 284, 328
 AND1 CY , @H + mem.bit ... 284, 328

[B]

BR addr ... 285, 329
 BR addr1 ... 285, 329
 BR BCDE ... 287, 333
 BR BCXA ... 287, 333
 BR PCDE ... 286, 332
 BR PCXA ... 286, 332
 BR !addr ... 286, 330
 BR \$addr ... 286, 330
 BR \$addr1 ... 286, 330
 BRA !addr1 ... 287, 330
 BRCB !caddr ... 287, 331

[C]

CALL !addr ... 288, 334
 CALLA !addr1 ... 288, 334
 CALLF !faddr ... 289, 335
 CLR1 CY ... 283, 325

CLR1 fmem.bit ... 284, 326
 CLR1 mem.bit ... 284, 326
 CLR1 pmem.@L ... 284, 326
 CLR1 @H + mem.bit ... 284, 326

[D]

DECS reg ... 283, 323
 DECS rp' ... 283, 323
 DI ... 292, 340
 DI IE x x x ... 292, 340

[E]

EI ... 292, 340
 EI IE x x x ... 292, 340

[G]

GETI taddr ... 294, 344

[H]

HALT ... 293, 342

[I]

IN A , PORTn ... 293, 341
 IN XA , PORTn ... 293, 341
 INCS mem ... 283, 323
 INCS reg ... 283, 322
 INCS rp1 ... 283, 322
 INCS @HL ... 283, 323

[M]

MOV A , mem ... 281, 305
 MOV A , reg ... 281, 306
 MOV A , #n4 ... 281, 303
 MOV A , @HL ... 281, 304
 MOV A , @HL + ... 281, 304

SKT mem.bit ... 284, 326
SKT pmem.@L ... 284, 327
SKT @H + mem.bit ... 284, 327
SKTCLR fmem.bit ... 284, 328
SKTCLR pmem.@L ... 284, 328
SKTCLR @H + mem.bit ... 284, 328
STOP ... 293, 342
SUBC A , @HL ... 282, 318
SUBC rp'1 , XA ... 282, 318
SUBC XA , rp' ... 282, 318
SUBS A , @HL ... 282, 317
SUBS rp'1 , XA ... 282, 318
SUBS XA , rp' ... 282, 317

【T】

TBR addr ... 294, 333
TCALL !addr ... 294, 335

【X】

XCH A , mem ... 281, 309
XCH A , reg1 ... 281, 309
XCH A , @HL ... 281, 308
XCH A , @HL + ... 281, 308
XCH A , @HL - ... 281, 308
XCH A , @rpa1 ... 281, 308
XCH XA , mem ... 281, 309
XCH XA , rp' ... 281, 309
XCH XA , @HL ... 281, 308
XOR A , #n4 ... 283, 321
XOR A , @HL ... 283, 321
XOR rp'1 , XA ... 283, 321
XOR XA , rp' ... 283, 321
XOR1 CY , fmem.bit ... 284, 329
XOR1 CY , pmem.@L ... 284, 329
XOR1 CY , @H + mem.bit ... 284, 329

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]

アクノリッジ許可ビット ...	149
アクノリッジ検出フラグ ...	149
アクノリッジ・トリガ・ビット ...	149
アドレス・コンパレータからの信号 ...	144
INT0エッジ検出モード・レジスタ ...	224
INT0割り込み許可フラグ ...	217
INT0割り込み要求フラグ ...	217
INT1エッジ検出モード・レジスタ ...	224
INT1割り込み許可フラグ ...	217
INT1割り込み要求フラグ ...	217
INT2エッジ検出モード・レジスタ ...	245
INT2割り込み許可フラグ ...	242
INT2割り込み要求フラグ ...	242
INT4割り込み許可フラグ ...	217
INT4割り込み要求フラグ ...	217
ウエイク・アップ機能指定ビット ...	144
ウォッチドッグ・タイマ許可フラグ ...	110
エッジ検出モード・レジスタ ...	224, 245

[か行]

キー割り込み入力 ...	243
キャリー・フラグ ...	66
クロック出力モード・レジスタ ...	105
コマンド検出フラグ ...	149
コマンド・トリガ・ビット ...	150

[さ行]

サブ発振回路コントロール・レジスタ ...	99
システム・クロック・コントロール・レジスタ ...	94
シフト・レジスタ ...	151

シリアル・インタフェース動作許可 / 禁止指定ビット ...	144
シリアル・インタフェース割り込み許可フラグ ...	217
シリアル・インタフェース割り込み要求フラグ ...	217
シリアル動作モード・レジスタ ...	143
シリアル・バス・インタフェース・コントロール・レジスタ ...	148
スキップ・フラグ ...	67
スタック・バンク選択レジスタ ...	48, 62
スタック・ポインタ ...	62
スレーブ・アドレス・レジスタ ...	152

[た行]

タイマ/イベント・カウンタ0カウント・レジスタ ...	121
タイマ/イベント・カウンタ0出力許可フラグ ...	126
タイマ/イベント・カウンタ0モード・レジスタ ...	123
タイマ/イベント・カウンタ0モジュロ・レジスタ ...	121
タイマ/イベント・カウンタ0割り込み許可フラグ ...	217
タイマ/イベント・カウンタ0割り込み要求フラグ ...	217
タイマ・カウンタ1カウント・レジスタ ...	122
タイマ・カウンタ1出力許可フラグ ...	126
タイマ・カウンタ1モード・レジスタ ...	123
タイマ・カウンタ1モジュロ・レジスタ ...	122
タイマ・カウンタ1割り込み許可フラグ ...	217
タイマ・カウンタ1割り込み要求フラグ ...	217
時計モード・レジスタ ...	118
時計用タイマ割り込み許可フラグ ...	242
時計用タイマ割り込み要求フラグ ...	242

[は行]

バス・リリース検出フラグ ...	149
バス・リリース・トリガ・ビット ...	150
バンク選択レジスタ ...	69

BT割り込み許可フラグ ...	217
BT割り込み要求フラグ ...	217
ビジィ許可ビット ...	149
ビット・シーケンシャル・バッファ ...	210
プルアップ抵抗指定レジスタ・グループA ...	86
プルアップ抵抗指定レジスタ・グループB ...	86
プログラム・カウンタ ...	49
プログラム・ステータス・ワード ...	66
プロセッサ・クロック・コントロール・レジスタ ...	92
ベーシック・インターバル・タイマ ...	108
ベーシック・インターバル・タイマ・モード・レジスタ ...	108
ポート0-8 ...	72
ポート・モード・レジスタ・グループA ...	79
ポート・モード・レジスタ・グループB ...	79
ポート・モード・レジスタ・グループC ...	79

[ま行]

メモリ・バンク許可フラグ ...	21, 68
メモリ・バンク選択レジスタ ...	21, 69

[ら行]

レジスタ・バンク許可フラグ ...	36, 69
レジスタ・バンク選択レジスタ ...	36, 70

[わ行]

割り込みステータス・フラグ ...	67, 225
割り込みマスタ許可フラグ ...	220

E.2 ハードウェア索引 (アルファベット順)

- [A]
- ACKD ... 149
 - ACKE ... 149
 - ACKT ... 149
- [B]
- BS ... 69
 - BSB0 - BSB3 ... 210
 - BSYE ... 149
 - BT ... 108
 - BTM ... 108
- [C]
- CLOM ... 105
 - CMDD ... 149
 - CMDT ... 150
 - COI ... 144
 - CSIE ... 144
 - CSIM ... 143
 - CY ... 66
- [I]
- IE0 ... 217
 - IE1 ... 217
 - IE2 ... 242
 - IE4 ... 217
 - IEBT ... 217
 - IECSI ... 217
 - IET0 ... 217
 - IET1 ... 217
 - IEW ... 242
 - IM0, IM1 ... 224
 - IM2 ... 245
 - IME ... 220
 - IRQ0 ... 217
 - IRQ1 ... 217
 - IRQ2 ... 242
 - IRQ4 ... 217
 - IRQBT ... 217
 - IRQCSI ... 217
 - IRQT0 ... 217
 - IRQT1 ... 217
 - IRQW ... 242
 - IST0 ... 67, 225
 - IST1 ... 67, 225
- [K]
- KR0 - KR7 ... 243
- [M]
- MBE ... 21, 68
 - MBS ... 21, 69
- [P]
- PC ... 49
 - PCC ... 93
 - PMGA ... 79
 - PMGB ... 79
 - PMGC ... 79
 - POGA ... 86
 - POGB ... 86
 - PORT0 - PORT8 ... 72
 - PSW ... 66
- [R]
- RBE ... 36, 69
 - RBS ... 36, 70
 - RELD ... 149
 - RELT ... 150

[S]

SBS ... 48, 62
SBIC ... 148
SCC ... 94
SIO ... 151
SK0-SK2 ... 67
SOS ... 100
SP ... 62
SVA ... 152

[T]

T0 ... 121
T1 ... 122
TOE0 ... 126
TOE1 ... 126
TM0 ... 124
TM1 ... 125
TMOD0 ... 121
TMOD1 ... 122

[W]

WDTM ... 110
WM ... 118
WUP ... 144

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	前版からの改版内容	適用箇所
第 2 版	44ピン・プラスチックQFPのパッケージを μ PD750008GB- $x \times x$ -3B4から μ PD750008GB- $x \times x$ -3BS-MTXに変更	全般
	μ PD75P0016を開発中から開発済みに変更	
	ポート 4, 5 のN-chオープン・ドレイン時, 入力耐圧を12 Vから13 Vに変更	
	TM1のビット 7 を “ 0 ” 固定に変更	
	関連資料に英文版の資料番号を追加	はじめに
	サブシリーズ製品間の比較の表のフォーマットを変更	第 1 章 概 説
	Mk モードとMk モードの違いの表に, Mk モードを使用する場合の注意を追加	第 4 章 内部CPU機能
	サブ発振回路コントロール・レジスタの項にフィードバック抵抗のマスク・オプションの説明を追加	第 5 章 周辺ハードウェア機能
	割り込み機能を制御する各種ハードウェアの項に, 割り込み許可フラグについての説明を追加	第 6 章 割り込み機能とテスト機能
	ベクタ・アドレス共用割り込みの項に, 共用割り込みの判別の表を追加	
	ワン・タイムPROMのスクリーニングについての説明を追加	第 9 章 PROM (プログラム・メモリ) の書き込みとベリファイ
	マスク・オプションの説明を追加	第10章 マスク・オプション
	命令セットのオペランドの@rpaを@rpa1に変更	第11章 命令セット
	命令セットのオペランドの表現形式と記述方法の表に@rpa1を追加	
	11.3 各命令の命令コードの命令一覧表(オペランド: BCDE)を訂正	
	命令機能と応用の見出しを, 命令セットとそのオペレーションに合わせて修正	
サポートするOSのバージョンをアップ	付録 B 開発ツール	

— お問い合わせ先 —

【技術的なお問い合わせ先】

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

電話 : 044-548-8899
FAX : 044-548-7900
E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東京 (03)3798-6106, 6107,
6108

名古屋 (052)222-2375

大阪 (06)6945-3178, 3200,
3208, 3212

仙台 (022)267-8740

郡山 (024)923-5591

千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,
6112

立川 (042)526-5981, 6167

松本 (0263)35-1662

静岡 (054)254-4794

金沢 (076)232-7303

松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156

水戸 (029)226-1702

広島 (082)242-5504

高崎 (027)326-1303

鳥取 (0857)27-5313

太田 (0276)46-4014

名古屋 (052)222-2170, 2190

福岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【インターネット電子デバイス・ニュース】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] μPD750008 ユーザーズ・マニュアル

(U10740JJ2V1UM00 (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,
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