

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

μPD753304

4ビット・シングルチップ・マイクロコンピュータ

---

μPD753304

資料番号 U12020JJ2V1UM00 (第2版)  
発行年月 June 2000 N CP(K)

© NEC Corporation 1996

[メモ]

## 目次要約

第1章	概 説	...	1
第2章	端子機能	...	7
第3章	アーキテクチャの特徴とメモリ・マップ	...	15
第4章	内部CPU機能	...	39
第5章	周辺ハードウェア機能	...	61
第6章	割り込み機能とテスト機能	...	133
第7章	スタンバイ機能	...	157
第8章	リセット機能	...	171
第9章	マスク・オプション	...	175
第10章	命令セット	...	177
付録A	$\mu$ PD75308B , 753108 , 753304の機能一覧表	...	247
付録B	開発ツール	...	251
付録C	マスクROM発注の手順	...	255
付録D	命令索引	...	257
付録E	ハードウェア索引	...	263
付録F	改版履歴	...	267

## CMOSデバイスの一般的注意事項

### 静電気対策（MOS全般）

**注意** MOSデバイス取り扱いの際は静電気防止を心がけてください。

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

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

### 未使用入力の処理（CMOS特有）

**注意** CMOSデバイスの入力レベルは固定してください。

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

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

### 初期化以前の状態（MOS全般）

**注意** 電源投入時、MOSデバイスの初期状態は不定です。

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

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

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（仕様を決定した者）が判定してください。

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

## 本版で改訂された主な箇所

箇所	内容
全般	μ PD753304を開発中から開発済みに変更
	量産品の出荷形態をペレット/ウエハに変更
	メイン・システム・クロックを4.0 MHzから3.6 MHz (typ. )に , サブシステム・クロックを32.768 kHzから47 kHz (typ. )に変更
p. 7	表2-2 ポート3のマスク・オプションによるリセット時の状態を追加
p.18	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を訂正
p.48	図4-5 表示データ・メモリの構成の中の説明を訂正
p.111	5.5.6 表示データ・メモリの本文のビット操作の記述を訂正
p.117	5.6.8 LCD駆動用電源 $V_{LC0}$ , $V_{LC1}$ , $V_{LC2}$ の供給 LCD表示モードのマスク・オプションを追加 LCD駆動用分割抵抗の10 k (typ. )を削除
p.118	図5-39 LCD駆動用電源の接続例を変更
p.157	第7章 スタンバイ機能 サブシステム・クロックのSTOPモード解除信号に割り込み要求信号を追加
p.180	10.1.6 表示データ・メモリ操作可能命令を追加
p.197	10.3 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.267	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、 $\mu$ PD753304の機能を理解し、それを用いたアプリケーション・システムを設計するユーザのエンジニアを対象としています。

**目的** このマニュアルは、次の構成に示す $\mu$ PD753304の持つハードウェア機能をユーザに理解していただくことを目的としています。

**構成** このマニュアルは、大きく分けて次の内容で構成しております。

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

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

**$\mu$ PD75308B, 753108をすでに経験しているユーザ**

付録A  $\mu$ PD75308B, 753108, 753304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 $\mu$ PD753304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	データの表記の重み	: 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..... $\text{xxx}$ または $\text{xxx}$ B
		10進数..... $\text{xxx}$
		16進数..... $\text{xxx}$ H

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

#### デバイス関連資料一覧

資料名	資料番号	
	和文	英文
μPD753304 データ・シート	U11874J	U11874E
μPD753304 ユーザーズ・マニュアル	U12020J (このマニュアル)	U12020E
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-753304DU-R ユーザーズ・マニュアル	U12173J	U12173E	
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

#### その他のドキュメント

資料名	資料番号	
	和文	英文
SEMICONDUCTOR 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	U11416J	-

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

(メモ)

# 目 次

<b>第1章 概 説</b> ...	1
1.1 機能概要 ...	2
1.2 オーダ情報 ...	3
1.3 ブロック図 ...	3
1.4 端子配置図 ...	4
<b>第2章 端子機能</b> ...	7
2.1 $\mu$ PD753304の端子機能一覧 ...	7
2.2 端子機能の説明 ...	9
2.2.1 P30-P33 ( PORT3 ) ,P80-P83 ( PORT8 ) P100-P103 ( PORT10 ) ...	9
2.2.2 PCL ...	9
2.2.3 BUZ ...	9
2.2.4 INT1 ...	10
2.2.5 S0-S23 ...	10
2.2.6 COM0-COM3 ...	10
2.2.7 CL1, CL2 ...	10
2.2.8 RESET ...	11
2.2.9 IC ...	11
2.2.10 VDD ...	11
2.2.11 Vss ...	11
2.3 端子の入出力回路 ...	12
2.4 未使用端子の処理について ...	14
<b>第3章 アーキテクチャの特徴とメモリ・マップ</b> ...	15
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 ...	15
3.1.1 データ・メモリのバンク構成 ...	15
3.1.2 データ・メモリのアドレッシング・モード ...	17
3.2 汎用レジスタのバンク構成 ...	29
3.3 メモリ・マップトI/O ...	34
<b>第4章 内部CPU機能</b> ...	39
4.1 Mk モードとMk モードの切り替え機能 ...	39
4.1.1 Mk モードとMk モードの違い ...	39
4.1.2 スタック・バンク選択レジスタ ( SBS ) の設定方法 ...	40
4.2 プログラム・カウンタ ( PC ) ...	41
4.3 プログラム・メモリ ( ROM ) ...	42
4.4 データ・メモリ ( RAM ) ...	44
4.4.1 データ・メモリの構成 ...	44
4.4.2 データ・メモリのバンクの指定 ...	45

- 4.5 汎用レジスタ ... 49
- 4.6 アキュームレータ ... 50
- 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) ... 50
- 4.8 プログラム・ステータス・ワード (PSW) ... 54
- 4.9 バンク選択レジスタ (BS) ... 58

## 第5章 周辺ハードウェア機能 ... 61

- 5.1 デジタル入出力ポート ... 61
  - 5.1.1 デジタル入出力ポートの種類, 特徴, 構成 ... 62
  - 5.1.2 入出力モードの設定 ... 65
  - 5.1.3 デジタル入出力ポート操作命令 ... 67
  - 5.1.4 デジタル入出力ポートの動作 ... 69
  - 5.1.5 プルアップ抵抗の内蔵 ... 71
  - 5.1.6 デジタル入出力ポートの入出力タイミング ... 72
- 5.2 クロック発生回路 ... 73
  - 5.2.1 クロック発生回路の構成 ... 73
  - 5.2.2 クロック発生回路の機能と動作 ... 75
  - 5.2.3 システム・クロックとCPUクロックの設定 ... 81
  - 5.2.4 クロック出力回路 ... 83
- 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 ... 86
  - 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 ... 86
  - 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) ... 87
  - 5.3.3 ウォッチドッグ・タイマ許可フラグ (WDTM) ... 88
  - 5.3.4 ベーシック・インターバル・タイマの動作 ... 89
  - 5.3.5 ウォッチドッグ・タイマの動作 ... 90
  - 5.3.6 その他の機能 ... 92
- 5.4 時計用タイマ ... 94
  - 5.4.1 時計用タイマの構成 ... 94
  - 5.4.2 時計モード・レジスタ ... 95
- 5.5 タイマ・カウンタ ... 97
  - 5.5.1 タイマ・カウンタの構成 ... 97
  - 5.5.2 8ビット・タイマ・カウンタ・モードの動作 ... 100
  - 5.5.3 タイマ・カウンタ応用上の注意 ... 102
- 5.6 LCDコントローラ/ドライバ ... 106
  - 5.6.1 LCDコントローラ/ドライバの構成 ... 106
  - 5.6.2 LCDコントローラ/ドライバの機能 ... 107
  - 5.6.3 表示モード・レジスタ (LCDM) ... 107
  - 5.6.4 表示制御レジスタ (LCDC) ... 109
  - 5.6.5 LCD/ポート選択レジスタ (LPS) ... 110
  - 5.6.6 表示データ・メモリ ... 111
  - 5.6.7 コモン信号とセグメント信号 ... 113
  - 5.6.8 LCD駆動用電源V<sub>LC0</sub>, V<sub>LC1</sub>, V<sub>LC2</sub>の供給 ... 117
  - 5.6.9 表示モード ... 119

<b>第6章</b>	<b>割り込み機能とテスト機能</b>	...	133
6.1	割り込み制御回路の構成	...	133
6.2	割り込みソースの種類とベクタ・テーブル	...	135
6.3	割り込み機能を制御する各種ハードウェア	...	137
6.4	割り込みシーケンス	...	142
6.5	多重割り込み処理制御	...	143
6.6	割り込み処理までのマシン・サイクル	...	145
6.7	割り込みの効果的な使い方	...	147
6.8	割り込みの応用	...	147
6.9	テスト機能	...	155
6.9.1	テスト・ソースの種類	...	155
6.9.2	テスト機能を制御する各種ハードウェア	...	155
<b>第7章</b>	<b>スタンバイ機能</b>	...	157
7.1	スタンバイ・モードの設定および動作状態	...	159
7.2	スタンバイ・モードの解除	...	161
7.3	スタンバイ・モード解除後の動作	...	163
7.4	マスク・オプションの選択	...	163
7.5	スタンバイ・モードの応用	...	164
<b>第8章</b>	<b>リセット機能</b>	...	171
<b>第9章</b>	<b>マスク・オプション</b>	...	175
9.1	RESET端子のマスク・オプション	...	175
★ 9.2	LCD表示モードのマスク・オプション	...	175
9.3	スタンバイ機能のマスク・オプション	...	175
★ 9.4	ポート3のマスク・オプション	...	175
<b>第10章</b>	<b>命令セット</b>	...	177
10.1	特徴的な命令	...	177
10.1.1	GETI命令	...	177
10.1.2	ビット操作命令	...	178
10.1.3	たてづみ命令	...	178
10.1.4	進数補正命令	...	179
10.1.5	スキップ命令とスキップに要するマシン・サイクル数	...	180
10.1.6	表示データ・メモリ操作可能命令	...	180
10.2	命令セットとそのオペレーション	...	181
10.3	各命令の命令コード	...	192
10.4	命令機能と応用	...	198
10.4.1	転送命令	...	199
10.4.2	テーブル参照命令	...	206
10.4.3	ビット転送命令	...	210
10.4.4	演算命令	...	211
10.4.5	アキュムレータ操作命令	...	219
10.4.6	増減命令	...	220

10.4.7	比較命令	...	222
10.4.8	キャリー・フラグ操作命令	...	224
10.4.9	メモリ・ビット操作命令	...	225
10.4.10	分岐命令	...	229
10.4.11	サブルーチン・スタック制御命令	...	234
10.4.12	割り込み制御命令	...	239
10.4.13	入出力命令	...	240
10.4.14	CPU制御命令	...	241
10.4.15	特殊命令	...	242

<b>付録A</b>	<b>μPD75308B , 753108 , 753304の機能一覧表</b>	...	247
------------	--	-----	-----

<b>付録B</b>	<b>開発ツール</b>	...	251
------------	--------------	-----	-----

<b>付録C</b>	<b>マスクROM発注の手順</b>	...	255
------------	--------------------	-----	-----

<b>付録D</b>	<b>命令索引</b>	...	257
------------	-------------	-----	-----

D.1	命令索引(機能別)	...	257
D.2	命令索引(アルファベット順)	...	260

<b>付録E</b>	<b>ハードウェア索引</b>	...	263
------------	-----------------	-----	-----

E.1	ハードウェア索引(50音順)	...	263
E.2	ハードウェア索引(アルファベット順)	...	265

★ <b>付録F</b>	<b>改版履歴</b>	...	267
--------------	-------------	-----	-----



## 図の目次 (1/3)

図番号	タイトル, ページ
3 - 1	MBE = 0 のモードと MBE = 1 のモードの使い分け ... 16
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 ... 18
3 - 3	スタティックRAMのアドレス更新方法 ... 23
3 - 4	レジスタ・バンクの使い分け例 ... 30
3 - 5	汎用レジスタの構成 (4ビット処理の場合) ... 32
3 - 6	汎用レジスタの構成 (8ビット処理の場合) ... 33
3 - 7	μPD753304 I/Oマップ ... 36
4 - 1	スタック・バンク選択レジスタのフォーマット ... 40
4 - 2	プログラム・カウンタの構成 ... 41
4 - 3	プログラム・メモリ・マップ ... 43
4 - 4	データ・メモリ・マップ ... 46
4 - 5	表示データ・メモリの構成 ... 48
4 - 6	汎用レジスタの構成 ... 49
4 - 7	レジスタ・ペアの構成 ... 49
4 - 8	アキュムレータ ... 50
4 - 9	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 ... 51
4 - 10	スタック・メモリへ退避されるデータ (Mk モード) ... 52
4 - 11	スタック・メモリから復帰されるデータ (Mk モード) ... 52
4 - 12	スタック・メモリへ退避されるデータ (Mk モード) ... 53
4 - 13	スタック・メモリから復帰されるデータ (Mk モード) ... 53
4 - 14	プログラム・ステータス・ワードの構成 ... 54
4 - 15	バンク選択レジスタの構成 ... 58
5 - 1	デジタル・ポートのデータ・メモリ・アドレス ... 61
5 - 2	P30の構成 ... 62
5 - 3	P31の構成 ... 62
5 - 4	P3nの構成 (n = 2, 3) ... 63
5 - 5	ポート8の構成 ... 63
5 - 6	P10nの構成 (n = 0-2) ... 64
5 - 7	P103の構成 ... 64
5 - 8	各ポート・モード・レジスタのフォーマット ... 66
5 - 9	プルアップ抵抗指定レジスタのフォーマット ... 71
5 - 10	デジタル入出力ポートの入出力タイミング ... 72

## 図の目次 (2/3)

図番号	タイトル, ページ
5 - 11	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 ... 73
5 - 12	クロック発生回路のブロック図 ... 74
5 - 13	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 ... 77
5 - 14	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 ... 78
5 - 15	メイン・システム・クロック発振回路の外付け回路 ... 78
5 - 16	接続回路の悪い例 ... 79
5 - 17	サブ発振回路コントロール・レジスタ (SOS) のフォーマット ... 80
5 - 18	システム・クロックとCPUクロックの切り替え ... 82
5 - 19	クロック出力回路のブロック図 ... 83
5 - 20	クロック出力モード・レジスタのフォーマット ... 84
5 - 21	リモコン波形出力応用例 ... 85
5 - 22	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 ... 86
5 - 23	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 ... 87
5 - 24	ウォッチドッグ・タイマ許可フラグ (WDTM) のフォーマット ... 88
5 - 25	時計用タイマのブロック図 ... 95
5 - 26	時計モード・レジスタのフォーマット ... 96
5 - 27	タイマ・カウンタのブロック図 ... 97
5 - 28	タイマ・カウンタ・モード・レジスタのフォーマット ... 99
5 - 29	LCDコントローラ/ドライバのブロック図 ... 106
5 - 30	表示モード・レジスタのフォーマット ... 108
5 - 31	表示制御レジスタのフォーマット ... 109
5 - 32	LCD/ポート選択レジスタのフォーマット ... 110
5 - 33	データ・メモリ・マップ ... 111
5 - 34	表示データ・メモリとコモン・セグメント対応 ... 112
5 - 35	コモン信号波形 (スタティック) ... 115
5 - 36	コモン信号波形 (1/2バイアス法) ... 115
5 - 37	コモン信号波形 (1/3バイアス法) ... 115
5 - 38	コモン信号とセグメント信号の電位と位相 ... 116
5 - 39	LCD駆動用電源の接続例 ... 118
5 - 40	スタティック方式LCDの表示パターンと電極結線 ... 119
5 - 41	スタティックLCDパネルの結線例 ... 120
5 - 42	スタティックLCD駆動波形例 ... 121
5 - 43	2時分割方式LCDの表示パターンと電極結線 ... 122
5 - 44	2時分割LCDパネルの結線例 ... 123

## 図の目次 (3/3)

図番号	タイトル, ページ
5 - 45	2時分割LCD駆動波形例 (1/2バイアス法) ... 124
5 - 46	3時分割方式LCDの表示パターンと電極結線 ... 125
5 - 47	3時分割LCDパネルの結線例 ... 126
5 - 48	3時分割LCD駆動波形例 (1/2バイアス法) ... 127
5 - 49	3時分割LCD駆動波形例 (1/3バイアス法) ... 128
5 - 50	4時分割方式LCDの表示パターンと電極結線 ... 129
5 - 51	4時分割LCDパネルの結線例 ... 130
5 - 52	4時分割LCD駆動波形例 (1/3バイアス法) ... 131
6 - 1	割り込み制御回路ブロック図 ... 134
6 - 2	割り込みベクタ・テーブル ... 136
6 - 3	割り込みプライオリティ選択レジスタ ... 139
6 - 4	INT1の構成 ... 140
6 - 5	INT1エッジ検出モード・レジスタのフォーマット ... 140
6 - 6	割り込み処理シーケンス ... 142
6 - 7	高位の割り込みによる多重割り込み ... 143
6 - 8	割り込みステータス・フラグの変更による多重割り込み ... 144
7 - 1	スタンバイ・モードの解除動作 ... 161
8 - 1	リセット機能の構成 ... 171
8 - 2	$\overline{\text{RESET}}$ 信号発生によるリセット動作 ... 171

## 表の目次 (1/2)

表番号	タイトル, ページ
2 - 1	デジタル入出力ポートの端子機能一覧 ... 7
2 - 2	ポート3のマスク・オプションによるリセット時の状態 ... 7
2 - 3	ポート端子以外の端子機能一覧 ... 8
2 - 4	未使用端子の処理一覧 ... 14
3 - 1	アドレッシング・モード ... 19
3 - 2	RBE, RBSと選択されるレジスタ・バンク ... 29
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 ... 30
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 ... 34
4 - 1	Mk モードとMk モードの違い ... 39
4 - 2	SBSで選択されるスタック・エリア ... 50
4 - 3	スタック動作時に退避/復帰されるPSWのフラグ ... 54
4 - 4	キャリー・フラグ操作命令 ... 55
4 - 5	割り込みステータス・フラグの指示内容 ... 56
4 - 6	MBE, MBSと選択されるメモリ・バンク ... 58
4 - 7	RBE, RBSと選択されるレジスタ・バンク ... 59
5 - 1	デジタル・ポートの種類と特徴 ... 62
5 - 2	入出力端子操作命令一覧表 ... 68
5 - 3	入出力ポート操作時の動作 ... 70
5 - 4	プルアップ抵抗内蔵の指定法 ... 71
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 ... 81
5 - 6	分解能と最長設定時間 ... 101
5 - 7	最大表示画素 ... 107
5 - 8	COM信号 ... 113
5 - 9	LCD駆動電圧 (スタティック) ... 114
5 - 10	LCD駆動電圧 (1/2バイアス法) ... 114
5 - 11	LCD駆動電圧 (1/3 バイアス法) ... 114
5 - 12	LCD駆動用電源の供給値 ... 117
5 - 13	S16-S23端子の選択, 非選択電圧 (スタティック表示) ... 119
5 - 14	S12-S15端子の選択, 非選択電圧 (2時分割表示例) ... 122
5 - 15	S6-S8端子の選択, 非選択電圧 (3時分割表示例) ... 125
5 - 16	S12, S13端子の選択, 非選択電圧 (4時分割表示例) ... 129

## 表の目次 (2/2)

表番号	タイトル, ページ
6 - 1	割り込みソースの種類 ... 135
6 - 2	割り込み要求フラグのセット信号 ... 138
6 - 3	IST1, IST0と割り込み処理状態 ... 141
6 - 4	テスト・ソースの種類 ... 155
6 - 5	テスト要求フラグのセット信号 ... 155
7 - 1	スタンバイ・モード時の各動作状態 ... 159
8 - 1	各ハードウェアのリセット後の状態 ... 172
9 - 1	ポート3のマスク・オプションによるリセット時の状態 ... 175
10 - 1	ビット操作アドレッシングの種類と指定可能な範囲 ... 178

(メモ)

# 第1章 概 説

μPD753304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。

従来の75XシリーズのLCDコントローラ/ドライバ内蔵製品は80ピン・パッケージを採用していましたが、μPD753304は、LCD表示を持った携帯機器などに組み込めるようにペレット/ウエハで販売します。

次に特徴を示します。

RC発振回路を内蔵

- ★ ・メイン・システム・クロック :  $f_{CC} = 3.6 \text{ MHz}$   
(外付け抵抗6.8 k 接続時のtyp.値, 10 pF (typ.) のコンデンサを内蔵)
  - ・サブシステム・クロック :  $f_{CT} = 47 \text{ kHz (typ.)}$   
(抵抗, コンデンサとも内蔵)
- スタンバイ・モード解除後ただちに処理を開始することが可能  
STOPモード時, サブシステム・クロックを発振停止可能
- ★ LCD表示を持った携帯機器などに組み込めるようにペレット/ウエハで販売
- 電源電圧 :  $V_{DD} = 2.5 \sim 5.5 \text{ V}$   
プログラマブルなLCDコントローラ/ドライバを内蔵

応用分野

- ・小型LCD表示装置 など

## 1.1 機能概要

項 目		機 能	
★	命令実行時間	<ul style="list-style-type: none"> <li>・ 1.1, 2.2, 4.4, 17.8 <math>\mu</math>s (メイン・システム・クロック : 3.6 MHz動作時)</li> <li>・ 85.1 <math>\mu</math>s (サブシステム・クロック : 47 kHz動作時)</li> </ul>	
	内蔵メモリ	ROM	4096 $\times$ 8 ビット
		RAM	256 $\times$ 4 ビット
	汎用レジスタ	<ul style="list-style-type: none"> <li>・ 4 ビット操作時 : 8 個 <math>\times</math> 4 バンク</li> <li>・ 8 ビット操作時 : 4 個 <math>\times</math> 4 バンク</li> </ul>	
入出力 ポート	CMOS入出力	12本	ソフトウェアにより、内蔵プルアップ抵抗の接続を指定可能 : 4 本 セグメント端子と兼用 : 4 本
	LCDコントローラ/ドライバ	<ul style="list-style-type: none"> <li>・ セグメント数選択 : 20/24セグメント (4 本単位でCMOS入出力ポートに切り替え可能, 最大 4 本)</li> <li>・ 表示モード選択 : スタティック, 1/2デューティ (1/2バイアス), 1/3デューティ (1/2バイアス), 1/3デューティ (1/3バイアス), 1/4デューティ (1/3バイアス)</li> </ul>	
★		<ul style="list-style-type: none"> <li>・ LCD表示モードをマスク・オプションで選択可能</li> </ul>	
	タイマ	3 チャンネル <ul style="list-style-type: none"> <li>・ 8 ビット・タイマ・カウンタ : 1 チャンネル (サブクロック・ソース入力機能あり)</li> <li>・ ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ : 1 チャンネル</li> <li>・ 時計用タイマ : 1 チャンネル</li> </ul>	
★	クロック出力 (PCL)	<ul style="list-style-type: none"> <li>・ , 3.6 MHz, 450 kHz, 225 kHz (メイン・システム・クロック : 3.6 MHz動作時)</li> </ul>	
★	ブザー出力 (BUZ)	<ul style="list-style-type: none"> <li>・ 2.94, 5.88, 47 kHz (サブシステム・クロック : 47 kHz動作時)</li> <li>・ 1.76, 3.52, 28.13 kHz (メイン・システム・クロック : 3.6 MHz動作時)</li> </ul>	
	ベクタ割り込み	外部 : 1 本, 内部 : 2 本	
	テスト入力	内部 : 1 本	
	システム・クロック発振回路	<ul style="list-style-type: none"> <li>・ メイン・システム・クロック発振用RC発振回路 (抵抗は外付け, 10 pF (typ.) のコンデンサを内蔵)</li> <li>・ サブシステム・クロック発振用RC発振回路 (抵抗, コンデンサとも内蔵)</li> </ul>	
	スタンバイ機能	STOPモード / HALTモード	
★	動作周囲温度	$T_A = -10 \sim +60$	
	電源電圧	$V_{DD} = 2.5 \sim 5.5$ V	
★	パッケージ	<ul style="list-style-type: none"> <li>・ 量産品 : ペレット / ウエハ</li> <li>・ ES品 (評価用) : 42ピン・セラミック・シュリンクDIP (600 mil)</li> </ul>	



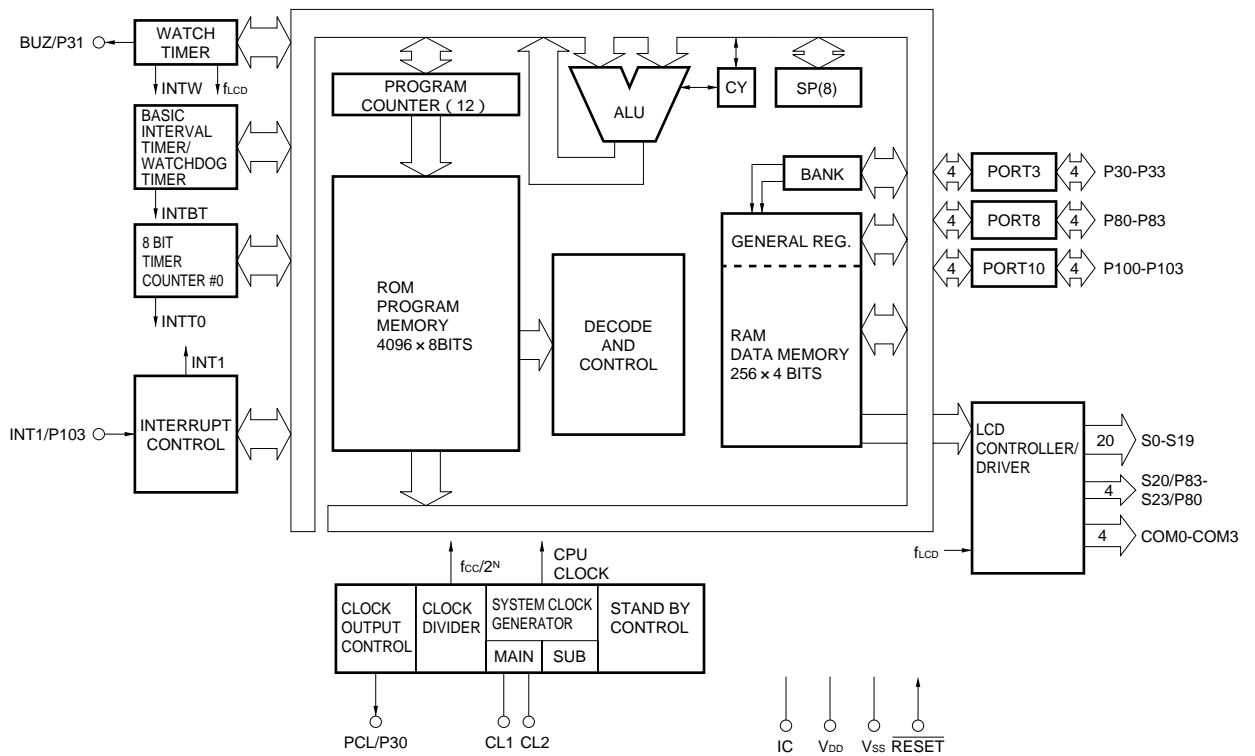
## 1.2 オーダ情報

	オーダ名称	出荷形態
★	μPD753304P- x x x	ペレット
★	μPD753304W- x x x	ウエハ

**注意** μPD753304はペレット/ウエハで販売します。ただしES品として42ピン・セラミック・シュリンクDIPを用意しています。

**備考** x x xはROMコード番号です。

## 1.3 ブロック図



## 1.4 端子配置図

### 量産品の端子配置図（パッド配置）

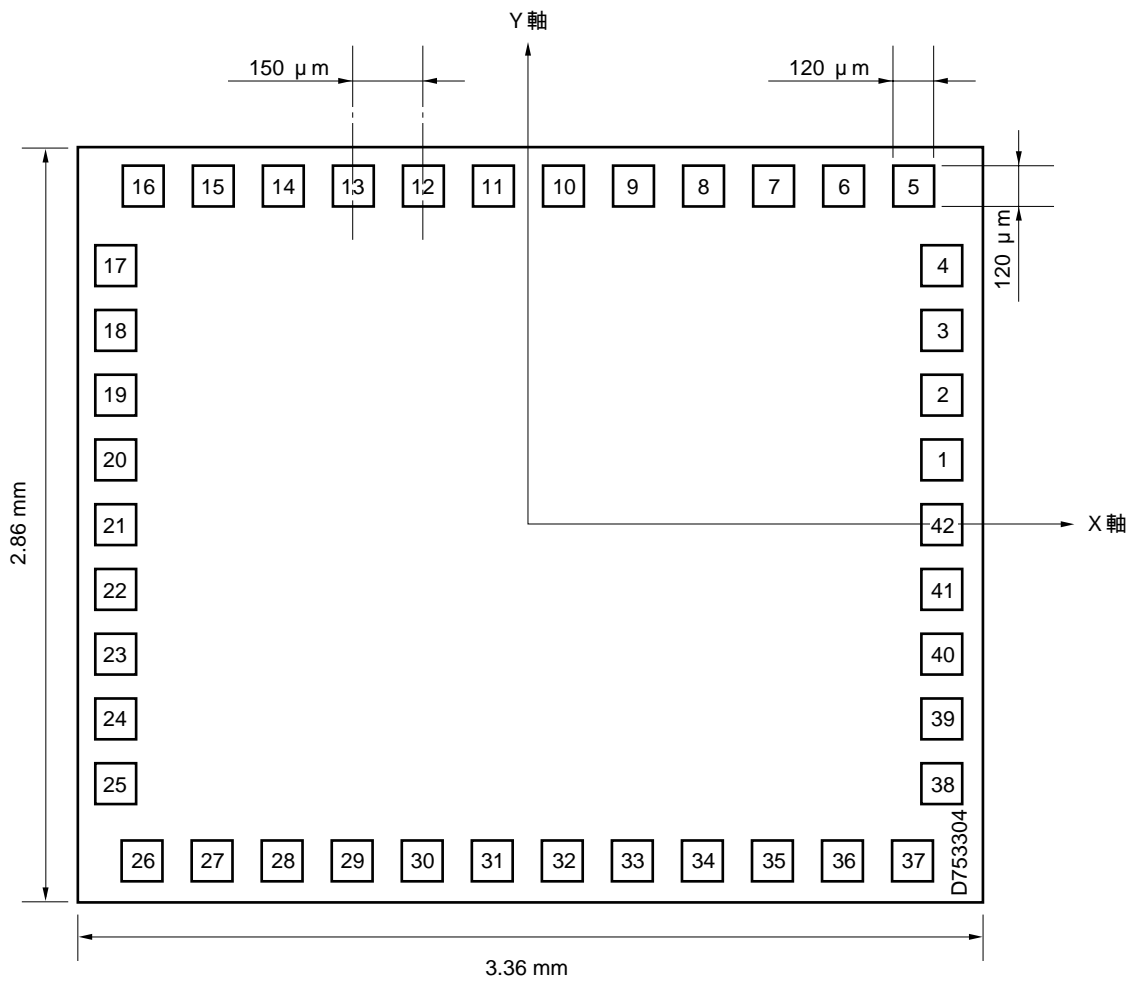
・ペレット

μ PD753304P-×××

チップ・サイズ : 3.36×2.86 mm<sup>2</sup>

パッド間隔 : 150 μm

パッド・サイズ : 120×120 μm



パッド座標 (単位:  $\mu\text{m}$ : パッド中心座標)

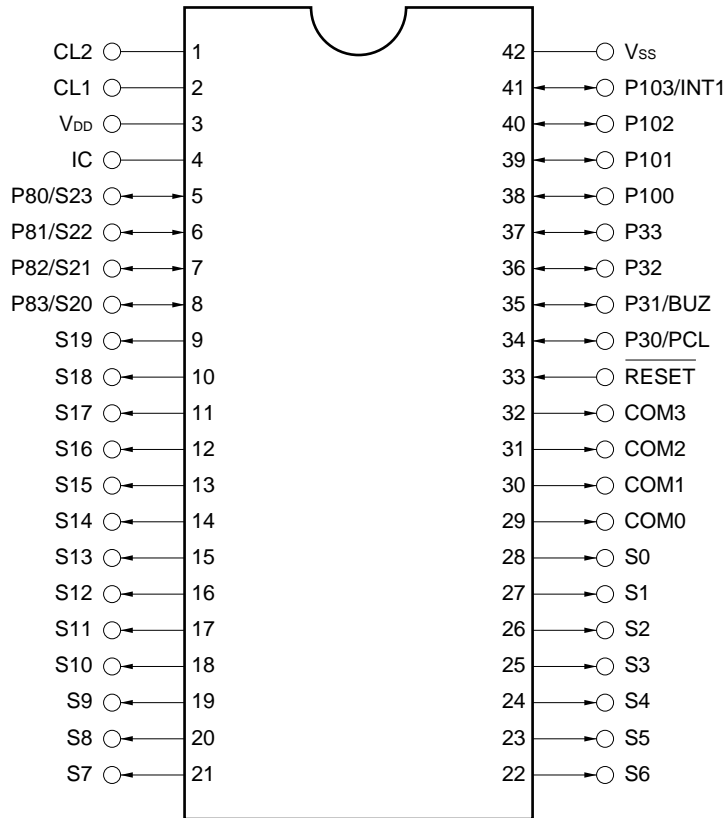
No.	端子名	X軸	Y軸
1	CL2	1549	311
2	CL1	1549	540
3	V <sub>DD</sub>	1549	769
4	IC	1549	998
5	P80/S23	1422.5	1299
6	P81/S22	1169.5	1299
7	P82/S21	916.5	1299
8	P83/S20	663.5	1299
9	S19	410.5	1299
10	S18	157.5	1299
11	S17	- 216.5	1299
12	S16	- 469.5	1299
13	S15	- 715.5	1299
14	S14	- 961.5	1299
15	S13	- 1207.5	1299
16	S12	- 1453.5	1299
17	S11	- 1549	992.5
18	S10	- 1549	746.5
19	S9	- 1549	500.5
20	S8	- 1549	254.5
21	S7	- 1549	- 105.5

No.	端子名	X軸	Y軸
22	S6	- 1549	- 351.5
23	S5	- 1549	- 597.5
24	S4	- 1549	- 843.5
25	S3	- 1549	- 1089.5
26	S2	- 1301	- 1299
27	S1	- 1055	- 1299
28	S0	- 809	- 1299
29	COM0	- 563	- 1299
30	COM1	- 317	- 1299
31	COM2	- 71	- 1299
32	COM3	289	- 1299
33	$\overline{\text{RESET}}$	518	- 1299
34	P30/PCL	747	- 1299
35	P31/BUZ	976	- 1299
36	P32	1205	- 1299
37	P33	1434	- 1299
38	P100	1549	- 997
39	P101	1549	- 768
40	P102	1549	- 539
41	P103/INT1	1549	- 310
42	V <sub>SS</sub>	1549	0.5

注意 ペレット背面はGNDに接続してください。

ES品の端子接続図 (Top View)

・42ピン・セラミック・シュリンクDIP (600 mil)



IC : Internally Connected (V<sub>DD</sub>に直接接続してください)

**注意**  $\mu$ PD753304はベレット/ウエハで販売します。上記の端子接続図はES品の場合を示しています。

端子名称

BUZ	: Buzzer Clock	P100-P103	: Port10
CL1, CL2	: RC Oscillator	PCL	: Programmable Clock
COM0-COM3	: Common Output0-3	RESET	: Reset
IC	: Internally Connected	S0-S23	: Segment Output0-23
INT1	: External Vectored Interrupt1	V <sub>DD</sub>	: Positive Power Supply
P30-P33	: Port3	V <sub>SS</sub>	: Ground
P80-P83	: Port8		

## 第2章 端子機能

### 2.1 μPD753304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

端子名称	入出力	兼用端子	機能	8ビットI/O	リセット時	入出力回路TYPE <sup>注1</sup>
★ P30	入出力	PCL	プログラマブル4ビット入出力ポート (PORT3)。 ビット単位で入力/出力指定可能。 リセット時の入力/出力モード指定可能 (マスク・オプション) <sup>注2</sup> 。	×	入力 <sup>注2</sup>	E
P31		BUZ				
P32		-				
P33		-				
P80	入出力	S23	4ビット入出力ポート (PORT8)。	×	入力	H
P81		S22				
P82		S21				
P83		S20				
P100	入出力	-	プログラマブル4ビット入出力ポート (PORT10)。 ビット単位で入力/出力指定可能。 4ビット単位で、ソフトウェアにより内蔵ブルアップ抵抗の接続を指定可能。	×	ブルアップ抵抗付き入力	E-B
P101		-				
P102		-				
P103		INT1				(F)-A

注1. 印はシュミット・トリガ入力を示します。

- ★ 2. マスク・オプションにより、リセット時の入力/出力モードの指定が可能です。詳細は表2-2を参照してください。

★ 表2-2 ポート3のマスク・オプションによるリセット時の状態

端子名称	リセット時の状態		
	マスク・オプション	マスク・オプション	マスク・オプション
P30/PCL	入力	ロウ・レベル出力	ロウ・レベル出力
P31/BUZ			ロウ・レベル出力
P32			
P33			ハイ・レベル出力

表2-3 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
★ PCL	出力	P30	クロック出力。		入力注2	E
BUZ		P31	任意の周波数出力（ブザー用またはシステム・クロックのトリミング用）。			
INT1	入力	P103	エッジ検出ベクタ割り込み入力（検出エッジ選択可能）。	非同期	プルアップ抵抗付き入力	Ⓕ-A
★ S0-S19	出力	-	セグメント信号出力。		ハイ・インピーダンス	G-B
S20-S23	出力	P83-P80	セグメント信号出力。		入力	H
★ COM0-COM3	出力	-	コモン信号出力。		ハイ・インピーダンス	G-B
CL1	-	-	メイン・システム・クロック発振用抵抗（R）接続端子。外部クロックを入力することはできません。		-	-
CL2	-	-				
$\overline{\text{RESET}}$	入力	-	システム・リセット入力（ロウ・レベル・アクティブ）。プルアップ抵抗を内蔵可能（マスク・オプション）。		-	Ⓖ-A
IC	-	-	Internally Connected. V <sub>DD</sub> に直接接続してください。		-	-
V <sub>DD</sub>	-	-	正電源。		-	-
V <sub>SS</sub>	-	-	グランド電位。		-	-

注1 . 印はシュミット・トリガ入力を示します。

- ★ 2 . マスク・オプションにより，リセット時の入力／出力モードの指定が可能です。  
詳細は表2-2を参照してください。

## 2.2 端子機能の説明

### 2.2.1 P30-P33 (PORT3) ...PCL, BUZ兼用入出力 P80-P83 (PORT8) ...S23-S20兼用入出力 P100-P103 (PORT10) ...INT1兼用入出力

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

ポート3：クロック出力 (PCL)

任意の周波数出力 (BUZ)

ポート8：セグメント信号出力 (S23-S20)

ポート10：ベクタ割り込み入力 (INT1)

ポートの入出力モードの選択はポート・モード・レジスタにより設定します。ポート3, 10はビット単位で、ポート8は4ビット単位で指定できます。

また、ポート10はソフトウェアにより内蔵プルアップ抵抗の接続を4ビット単位で指定可能で、プルアップ抵抗指定レジスタ・グループB (POGB) を操作して行います。

$\overline{\text{RESET}}$ 信号発生により、各ポートは次のようになります。

- ★ ポート3 (P30-P33)：マスク・オプションにより、リセット時の入力/出力モードの指定が可能です。詳細は表2-2を参照してください。

ポート10 (P100-P103)：入力モード (プルアップ抵抗付き)

### 2.2.2 PCL...ポート3兼用出力

プログラマブル・クロック出力端子です。周辺LSI (スレーブのマイクロコンピュータなど) にクロックを供給するときに使用し、P30端子と兼用となっています。 $\overline{\text{RESET}}$ 信号発生により、クロック出力モード・レジスタ (CLOM) が“0”にクリアされクロックは出力禁止状態になり、通常のポートとしての動作モードになります。

詳細は5.2.4 クロック出力回路を参照してください。

### 2.2.3 BUZ...ポート3兼用出力

周波数出力端子です。任意の周波数 (2.94 kHz, 5.88 kHz, 47 kHz: サブシステム・クロック47 kHz動作時) を出力することにより、ブザー音発生やシステム・クロック発振周波数のトリミングに使用します。P31端子と兼用となっており、時計モード・レジスタ (WM) のビット7 (WM7) が“1”にセットされているときのみ有効となります。

$\overline{\text{RESET}}$ 信号発生により、WM7は“0”にクリアされ、通常のポートとしての動作モードになります。  
詳細は5.4.2 時計モード・レジスタを参照してください。

## 2.2.4 INT1...ポート10兼用入力

エッジ検出ベクタ割り込み入力端子で、エッジ検出モード・レジスタ(IM1)により、検出エッジの選択ができます。

### ・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

INT1は非同期入力であり、CPUの動作クロックに関係なく一定のハイ・レベル幅を持つ入力があれば受け付けられ、P103端子と兼用となっています。

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

★ STOPモード、HALTモードの解除に利用できます。

また、INT1はシュミット・トリガ入力になっています。

## 2.2.5 S0-S23...出力

LCDのセグメント端子(前面電極)を直接駆動可能なセグメント信号出力端子でスタティック、1/2バイアス法の2または3時分割、1/3バイアス法の3または4時分割駆動を行います。

S20-S23はポート8と兼用しており、表示モード・レジスタ(LCDM)により切り替えて使用します。

## 2.2.6 COM0-COM3...出力

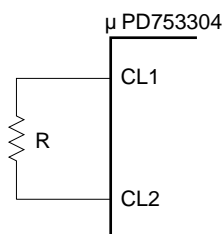
LCDのコモン端子(背面電極)を直接駆動可能なコモン信号出力端子です。スタティック(COM0, 1, 2, 3出力)、1/2バイアス法の2時分割駆動時(COM0, 1出力)または3時分割駆動時(COM0, 1, 2出力)、1/3バイアス法の3時分割駆動時(COM0, 1, 2出力)または4時分割駆動時(COM0, 1, 2, 3出力)にコモン信号をそれぞれ出力します。

## 2.2.7 CL1, CL2

メイン・システム・クロック発振用の抵抗(R)接続端子です。10 pF (typ.) のコンデンサ(C)を内蔵しています。

外部クロックを入力することはできません。

### RC発振





### 2.2.8 $\overline{\text{RESET}}$

ロウ・レベル・アクティブのリセット入力端子です。

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

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

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

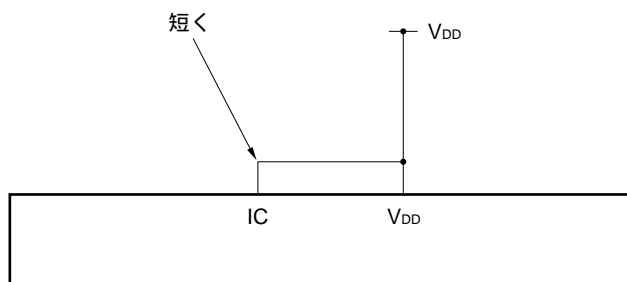
マスク・オプションにより60 k (typ.) のプルアップ抵抗の内蔵を指定できます。

### 2.2.9 IC

IC (Internally Connected) 端子は、当社出荷時に  $\mu$ PD753304を検査するためのテスト・モードに設定するための端子です。通常動作時には、IC端子は $V_{DD}$ 端子に直接接続し、その配線長を極力短くしてください。

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

IC端子は $V_{DD}$ 端子に直接接続してください



### 2.2.10 $V_{DD}$

正電源供給端子です。

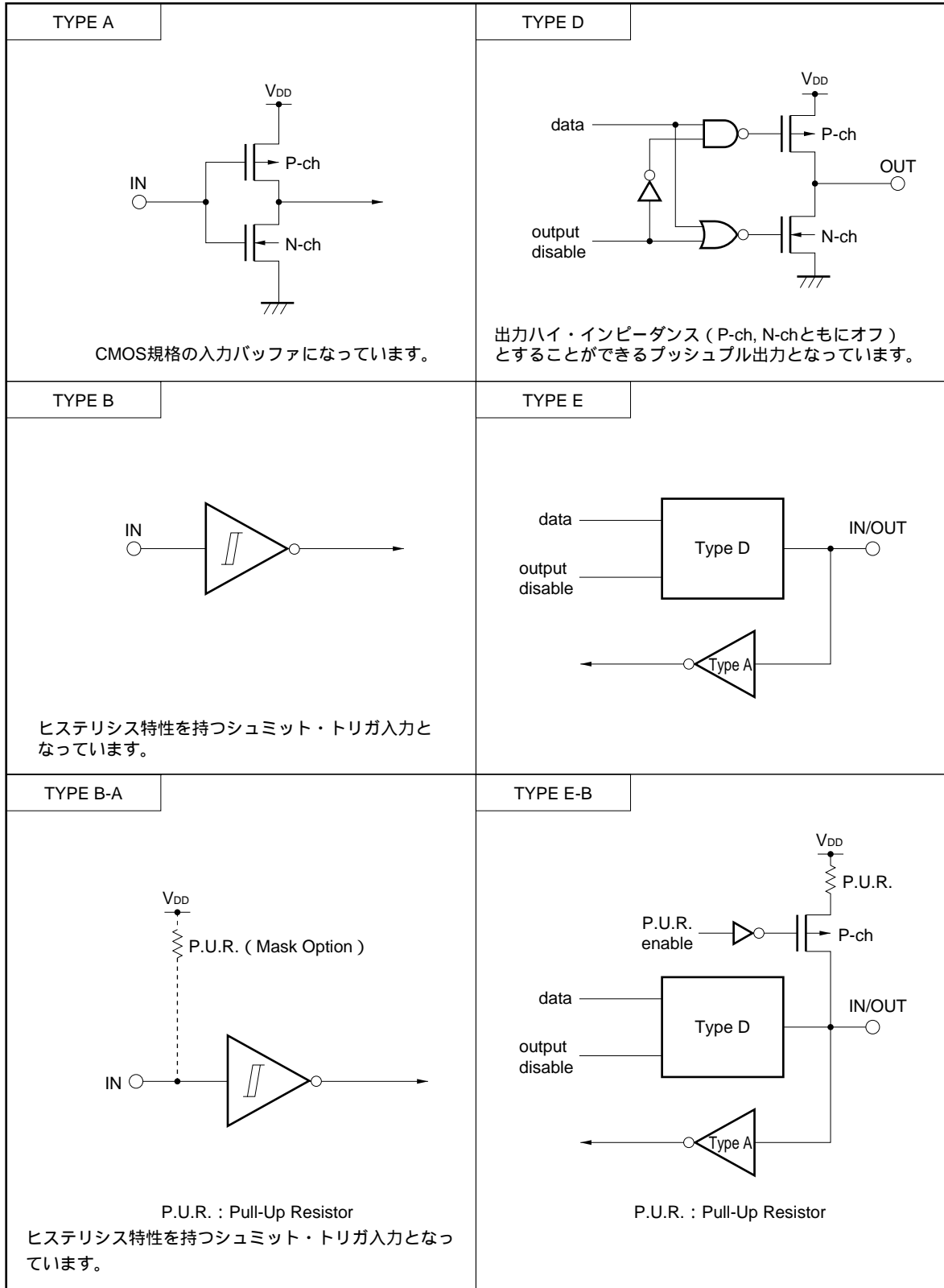
### 2.2.11 $V_{SS}$

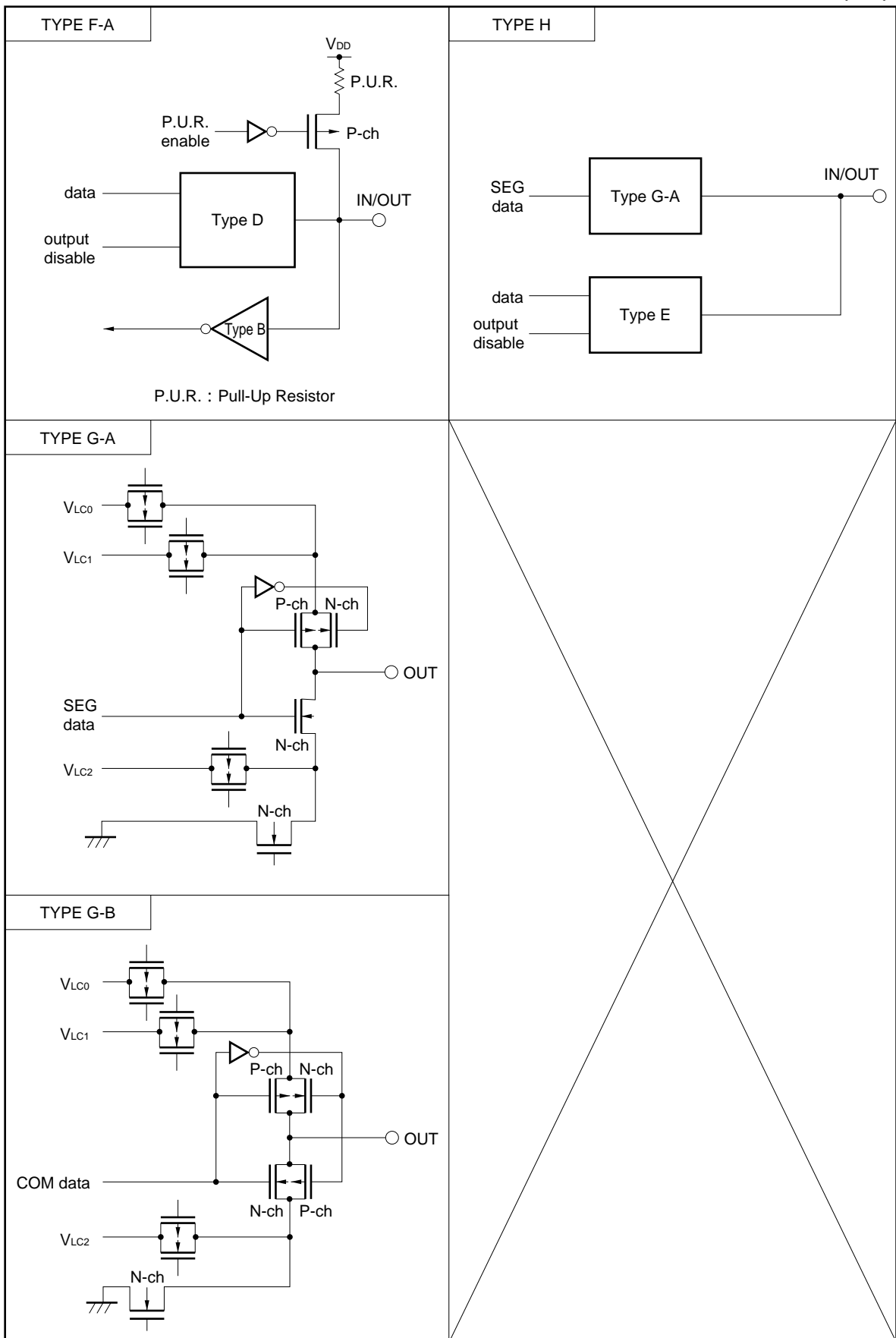
GND電位です。

## 2.3 端子の入出力回路

μ PD753304の各端子の入出力回路を一部簡略した形式を用いて示します。

( 1/2 )





## 2.4 未使用端子の処理について

★

表2 - 4 未使用端子の処理一覧

端 子	推奨接続方法
P30/PCL	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
P31/BUZ	
P32	
P33	
P100	
P101	
P102	
P103/INT1	
S0-S19	
COM0-COM3	
S20/P83-S23/P80	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
IC	$V_{DD}$ に直接接続

## 第3章 アーキテクチャの特徴とメモリ・マップ

μPD753304に採用されている75XLのアーキテクチャは以下の特徴を持っています。

内蔵RAM最大4Kワード×4ビット(12ビット・アドレス)

周辺ハードウェアの拡張性

この優れた特徴を実現するため、以下の方式を採用しています。

(1) データ・メモリのバンク構成

(2) 汎用レジスタのバンク構成

(3) メモリ・マップI/O

本章では、これらのおのおのについて解説します。

### 3.1 データ・メモリのバンク構成とアドレッシング・モード

#### 3.1.1 データ・メモリのバンク構成

μPD753304ではデータ・メモリ空間のメモリ・バンク0の000H-0FFH番地にスタティックRAMが内蔵されています。また、書き込みのみ可能な表示データ・メモリ(24×4ビット)がメモリ・バンク1の1E0H-1F7H番地に、周辺ハードウェア(入出力ポートやタイマなど)がメモリ・バンク15のF80H-FFFH番地に割り付けられています。

μPD753304は、12ビット・アドレス(4Kワード×4ビット)のデータ・メモリ空間をアドレッシングするために、下位8ビット・アドレスは命令によって直接あるいは間接に指定し、上位4ビット・アドレスをメモリ・バンクによって指定するメモリ・バンク構成となっています。

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

メモリ・バンク許可フラグ(MBE)

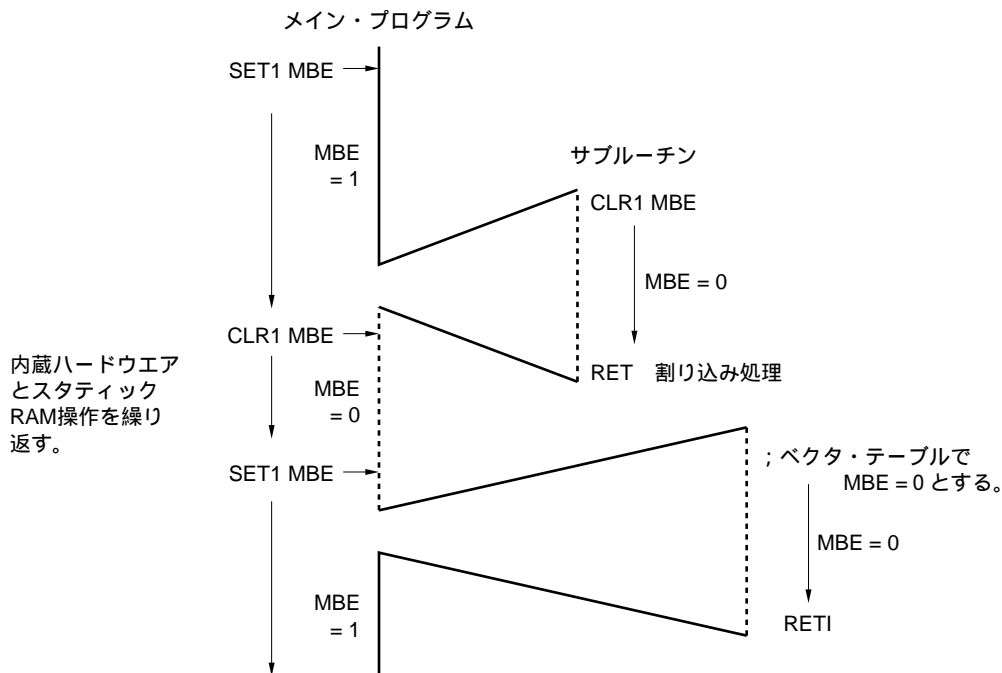
メモリ・バンク選択レジスタ(MBS)

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

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

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

図3 - 1 MBE = 0 のモードとMBE = 1 のモードの使い分け



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

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

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

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

例1 . MBEをクリアし、メモリ・バンクを固定する。

```
CLR1 MBE ; MBE 0
```

2 . メモリ・バンク1を選択する。

```
SET1 MBE ; MBE 1
```

```
SEL MB1 ; MBS 1
```

### 3.1.2 データ・メモリのアドレッシング・モード

μPD753304に採用されている75XLシリーズのアーキテクチャでは、データ・メモリ空間を、処理するデータのビット長ごとに効率よくアドレッシングするために、表3 - 1にまとめたように7種類のアドレッシング・モードが用意されており、効率のよいプログラミングが可能となります。

#### (1) 1ビット直接アドレッシング (mem. bit)

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

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

MBE = 1 のモードでは、MB = MBS となります。

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 @HL	スタック アドレッシング	fmem.bit	pmem.@L
	MBE =0	MBE =1	MBE =0	MBE =1	MBE =0	MBE =1	MBE =0	MBE =1				
000H	メモリ・バンク許可フラグ	汎用 レジスタ・エリア	MBS =0	MBS =0	MBS =0	MBS =0	MBS =0	MBS =0	SBS =0			
01FH												
020H												
07FH	データ・エリア スタティックRAM (メモリ・バンク0)		MBS =0	MBS =0	MBS =0	MBS =0	MBS =0	MBS =0				
080H												
0FFH												
100H	内蔵されていない											
1DFH												
1E0H												
1F7H	表示データ・メモリ (書き込みのみ可能)		MBS =1				MBS =1					
1F8H												
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)		MBS =15	MBS =15			MBS =15	MBS =15				
FB0H												
FBFH												
FC0H												
FF0H												
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。 ただし, MB = MBE · MBS
	@HL + @HL -	MBとHLで示されるアドレス。ただし, MB = MBE · MBS HL + は, アドレッシングのあとLレジスタを自動的にインクリメント。 HL - は, アドレッシングのあとLレジスタを自動的にデクリメント。
	@DE	メモリ・バンク0の, DEで示されるアドレス。
	@DL	メモリ・バンク0の, DLで示されるアドレス。
8ビット・レジスタ 間接アドレッシング	@HL	MBとHLで示されるアドレス (Lレジスタの内容は偶数)。 ただし, MB = MBE · MBS
ビット操作 アドレッシング	fmem.bit	fmemで示されるアドレスの, bitで示されるビット。 fmem = FB0H-FBFH (割り込み関係のハードウェア) FF0H-FFFH (I/Oポート)
	pmem. @L	pmemの上位10ビットとLレジスタの上位2ビットで示されるアドレスの, Lレジスタの下位2ビットで示されるビット。 ただし, pmem = FC0H-FFFH
	@H + mem.bit	MBとHそしてmemの下位4ビットで示されるアドレスの, bitで示される ビット。 ただし, MB = MBE · MBS
スタック・アドレッシング	-	メモリ・バンク0の, SPで示されるアドレス。

**(2) 4ビット直接アドレッシング (mem)**

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

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

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。

**(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。

## (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60H-67Hに転送する。

```

DATA1      EQU      57H
DATA2      EQU      67H
           SET1     MBE                ; MBE  1
           SEL     MB0                ; MBS  0
           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、入出力など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60H-67Hのデータを比較する。

```

DATA1 EQU 57H
DATA2 EQU 67H
      SET1 MBE
      SEL MB0
      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

```

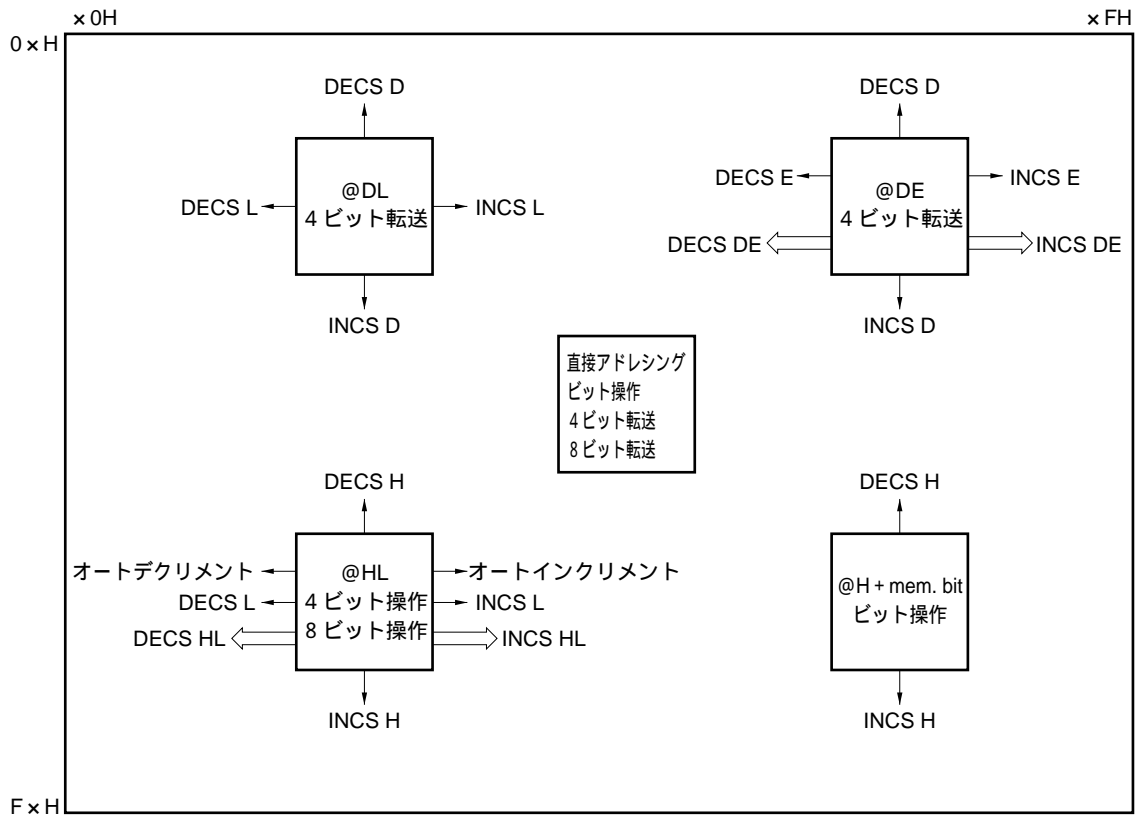
2 . 00H-FFHのデータ・メモリを0クリアする。

```

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・カウンタ0のカウンタ・レジスタ(T0)値が30H, 31H番地のデータと等しいかどうか比較する。

```

DATA EQU 30H
      CLR1 MBE
      MOV HL, #DATA
      MOV XA, T0 ; XA カウンタ・レジスタ0
      SKE 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、SKTCLR命令によるテスト&リセットなど、多彩なビット操作が可能になります。

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

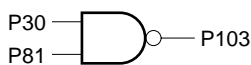
(a) 特定アドレス・ビット直接アドレッシング (fmem. bit)

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

例1．タイマ0 割り込み要求フラグ (IRQT0) をテストし、セットされていたら要求フラグをクリアし、P100をリセットする。

```
SKTCLR  IRQT0          ; IRQT0 = 1 ?
BR      NO             ; NO
CLR1    PORT10.0      ; YES
```

2．P30とP81の端子がともに1であれば、P103をリセットする。



```
MOV1    CY, PORT 3.0  ; CY  P30
AND1    CY, PORT8. 1  ; CY  P81
NOT1    CY             ; CY   $\overline{CY}$ 
MOV1    PORT 10.3, CY ; P103  CY
```

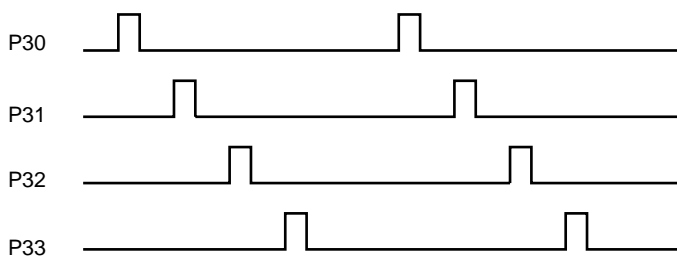
(b) 特定アドレス・ビット・レジスタ間接アドレッシング (pmem. @L)

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など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。

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

例 ポート3の各ビットに順にパルス出力する。



```

LOOP2 : MOV     L, #0
LOOP1 : SET1    PORT3.@L    ; ポート3のビット(L1-0)  1
        CLR1    PORT3.@L    ; ポート3のビット(L1-0)  0
        INCS    L
        SKE     L, #4H
        BR     LOOP1
        BR     LOOP2
    
```



(c) 特殊1ビット直接アドレッシング (@H+mem. bit)

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

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

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



```

FLAG1 EQU 30H. 3
FLAG2 EQU 31H. 0
FLAG3 EQU 32H. 2
SEL MB0
MOV H, #FLAG1 SHR 6
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** . サブルーチン処理時でレジスタを退避 / 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

μPD753304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

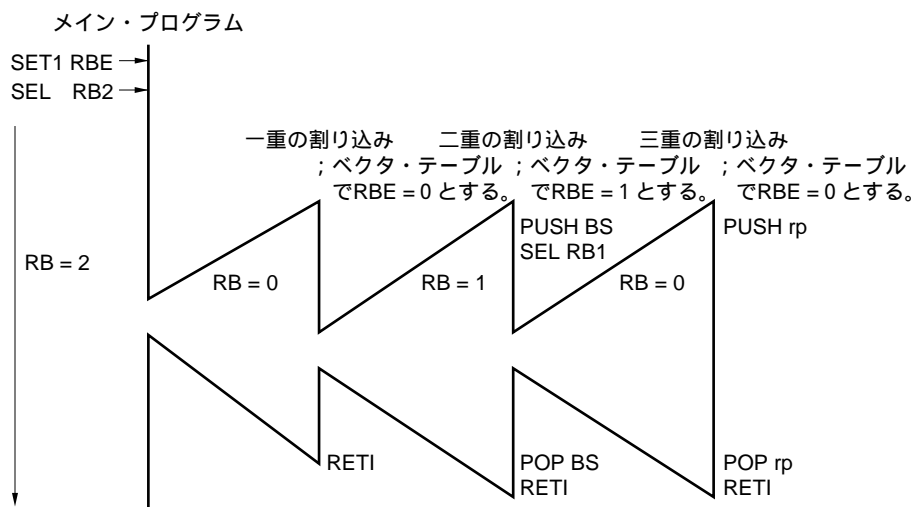
備考 x = don't care

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。
一重割り込み処理	RBE = 0 とし、レジスタ・バンク 0 を使用する。
二重割り込み処理	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)
三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

μ PD753304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、 $RB = RBE \cdot RBS$ 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 (PC <sub>11-8</sub> + DE) <sub>ROM</sub> , テーブル参照
	SKE	XA, BC	; XA = BCならスキップ

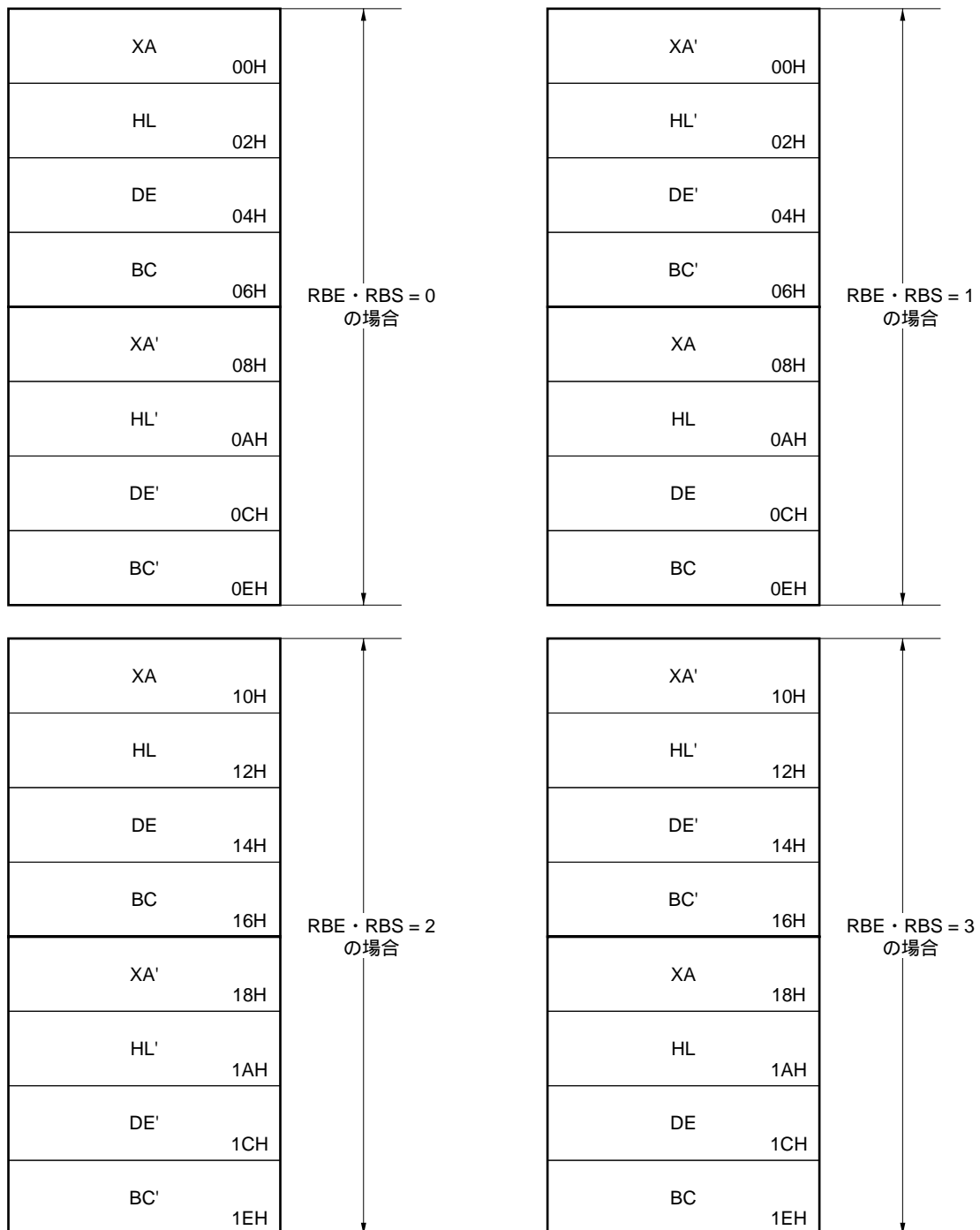
2. タイマ・カウンタ0のカウンタ・レジスタ (T0) の値がBC'レジスタ・ペアの値より大きいかどうかをテストし、大きくなるまで待つ。

	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

μPD753304は、図3 - 2に示したようにデータ・メモリ空間上のF80H-FFFH番地に入出力ポートやタイマなどの周辺ハードウェアをマッピングしたメモリ・マップトI/Oを採用しています。このため、周辺ハードウェアを制御する特別な命令はなく、すべてメモリ操作命令により制御します（一部、プログラムをわかりやすくするためのハードウェア制御の二モニックが用意してあります）。

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

なお、1E0H-1F7H番地にマッピングしてある表示データ・メモリ（書き込みのみ可能）はメモリ・バンク1を指定して操作します。

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	PORTn. x
4ビット操作	MBE = 0 , または ( MBE = 1 , MBS = 15 ) として 直接アドレッシングmemで指定	4ビット操作可能なすべてのハードウェア
	( MBE = 1 , MBS = 15 ) として レジスタ間接アドレッシング@HLで指定	
8ビット操作	MBE = 0 , または ( MBE = 1 , MBS = 15 ) として直接アドレッシングmemで指定。ただし、memは偶数アドレス	8ビット操作可能なすべてのハードウェア
	MBE = 1 , MBS = 15としてレジスタ間接アドレッシング@HLで指定。ただし、Lレジスタの内容は偶数	

例 CLR1      MBE            ; MBE = 0  
 SET1      TM0. 3        ; タイマ0のスタート  
 EI        IEBT        ; INTBTの許可  
 DI        IE1          ; INT1の禁止  
 SKTCLR    IRQT0        ; INTT0要求フラグのテスト・アンド・クリア  
 SET1      PORT8, @L    ; ポート8のセット

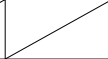


図3 - 7 に  $\mu$ PD753304のI/Oマップをまとめます。

同図における項目の意味は次のようになっています。

- ・略号.....内蔵されたハードウェアのアドレスを示すネームです。  
命令のオペランド欄に記述できます。
- ・R/W.....該当するハードウェアが、読み出し / 書き込み可能かどうかを示します。
  - R/W : 読み出し (Read) / 書き込み (Write) 可能
  - R : 読み出し (Read) のみ
  - W : 書き込み (Write) のみ
- ・操作可能なビット数.....該当するハードウェアを操作する場合に、適用可能な処理ビット数を示します。
  - : その欄で使用している単位 (1/4/8 ビット) でのビットの操作が可能。
  - ◐ : 一部のビットのみ操作が可能。操作の可能なビットについては備考欄を参照してください。
  - : その欄で使用している単位 (1/4/8 ビット) でのビットの操作が不可能。
- ・ビット操作アドレッシング.....該当するハードウェアをビット操作する場合に、適用可能なビット操作アドレッシングを示します。

図3-7 μPD753304 I/Oマップ (1/3)

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
	バンク選択レジスタ (BS)									
F83H	メモリ・バンク選択レジスタ (MBS)									
F84H	スタック・バンク選択レジスタ (SBS)				R/W	-	-	-	-	
F85H	ベーシック・インターバル・タイム・モード・レジスタ (BTM)				W			-	mem. bit	ビット3のみ ビット操作可能
F86H	ベーシック・インターバル・タイム (BT)				R	-	-		-	
F8BH	WDTM <sup>注2</sup>				W		-	-	mem. bit	ビット3のみ ビット操作可能
F8CH	表示モード・レジスタ (LCDM)				R/W	(W)	-		mem. bit	ビット3のみ ビット操作可能
						-	-			
F8EH	表示制御レジスタ (LCDC)				R/W	-	-	-	-	
F8FH	LCD/ポート選択レジスタ (LPS)				R/W	-	-	-	-	
F98H	時計モード・レジスタ (WM)				R/W	-	-		-	
FA0H	タイマ・カウンタ0				R/W	(W)	-		mem. bit	ビット3のみ ビット操作可能
	モード・レジスタ (TM0)					-	-		-	
FA4H	タイマ・カウンタ0				R	-	-		-	
	カウント・レジスタ (T0)									
FA6H	タイマ・カウンタ0				R/W	-	-		-	
	モジュロ・レジスタ (TMOD0)									

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。

BSは、PUSH BS, POP BS命令で行ってください。

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

図3 - 7 μPD753304 I/Oマップ (2/3)

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) 注1 (SK2) 注1 (SK1) 注1 (SK0) 注1					-	-			
FB2H	割り込みプライオリティ選択レジスタ (IPS)				R/W	-		-		注2
	IME	-	-	-						
FB3H	プロセッサ・クロック・コントロール・レジスタ (PCC)				R/W	-		-		注3
FB5H	INT1エッジ検出モード・レジスタ (IM1)				R/W	-		-	-	
FB7H	システム・クロック・コントロール・レジスタ (SCC)				R	(R/W)		-	-	ビット0, 3のみ ビット操作可能
FB8H	INTAレジスタ (INTA)				R/W			-	fmem. bit	注4
	-	-	IEBT	IRQBT						
FBAH	INTCレジスタ (INTC)				R/W			-	fmem. bit	注4
	-	-	IEW	IRQW						
FBCH	INTEレジスタ (INTE)				R/W			-	fmem. bit	注4
	-	-	IET0	IRQT0						
FBEH	INTGレジスタ (INTG)				R/W			-	fmem. bit	注4
	IE1	IRQ1	-	-						
FCFH	サブ発振回路コントロール・レジスタ (SOS)				R/W	-		-	-	

注1 . 予約語としては登録されていません。

- 2 . ビット3のみEI/DI命令により操作可能。
- 3 . ビット , 3 , 2はSTOP/HALT命令実行時ビット操作可能。
- 4 . 1ビット操作は , 1ビット操作用予約語でのみ可能。

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

図3 - 7 μPD753304 I/Oマップ (3/3)

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	-		-	-	
FDEH	-	-	-	(PO10)注	R/W	-	-		-	
	プルアップ抵抗指定レジスタ・グループB (POGB)									
	-	-	-	-						

FE8H	(PM33)注	(PM32)注	(PM31)注	(PM30)注	R/W	-	-		-	
	ポート・モード・レジスタ・グループA (PMGA)									
	-	-	-	-						
FEAH	(PM103)注	(PM102)注	(PM101)注	(PM100)注	R/W	-	-		-	
	ポート・モード・レジスタ・グループD (PMGD)									
	-	-	-	-						
FEEH	-	-	-	(PM8)注	R/W	-	-		-	
	ポート・モード・レジスタ・グループC (PMGC)									
	-	-	-	-						

FF3H	ポート3 (PORT3)				R/W			-	-	
FF8H	ポート8 (PORT8)				R/W			-		
FFAH	ポート10 (PORT10)				R/W			-		

注 予約語としては登録されていません。

## 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

μPD753304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：75Xシリーズと上位互換性があります。  
ROM容量が16 Kバイトまでの75XL CPUで使用できます。
- ・Mk モード：75Xシリーズとの互換性はありません。  
ROM容量が16 Kバイト以上の製品も含め、75XL CPU全部で使用できます。

表4-1 Mk モードとMk モードの違い

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

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

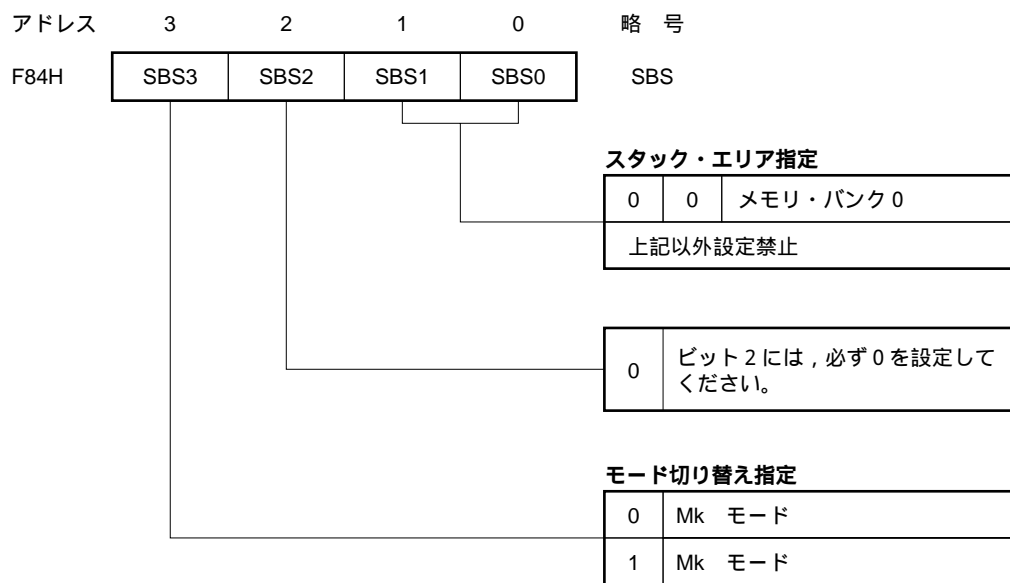
なお、Mk モードを選択すると、サブルーチン・コール命令実行時のスタック・バイト数(使用エリア)がMk モードに比べ1スタックごとに1バイト分増えます。また、CALL !addr, CALLF !faddr命令使用時は、マシン・サイクルがそれぞれ1マシン・サイクル分長くなります。したがって、ソフトウェアの互換性よりRAMの使用効率や処理能力を重視する場合は、Mk モードを使用してください。

### 4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法

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

スタック・バンク選択レジスタは、4ビット・メモリ操作命令により設定します。Mk モードを使用する場合は、プログラムの初期で必ずスタック・バンク選択レジスタを1000B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0000B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

プログラム・メモリのアドレスを保持するバイナリ・カウンタです。

図4 - 2 プログラム・カウンタの構成

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つ実行することによってその命令のバイト数に応じて自動的にインクリメントされます。

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

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

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

RESET信号発生により、プログラム・カウンタ (PC) の内容はプログラム・メモリの0000H, 0001H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。

PC<sub>11-8</sub> (0000H)<sub>3-0</sub>, PC<sub>7-0</sub> (0001H)<sub>7-0</sub>

## 4.3 プログラム・メモリ (ROM) .....4096 × 8 ビット

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

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

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

プログラム・メモリのアドレスは0000H-0FFFHで、そのうち次に示すアドレスには特別な機能を割り付けています。また、0000H, 0001Hを除くすべての領域は、通常のプログラム・メモリとして使用できます。

- ・ 0000H, 0001H番地

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

- ・ 0002H-000BH番地

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

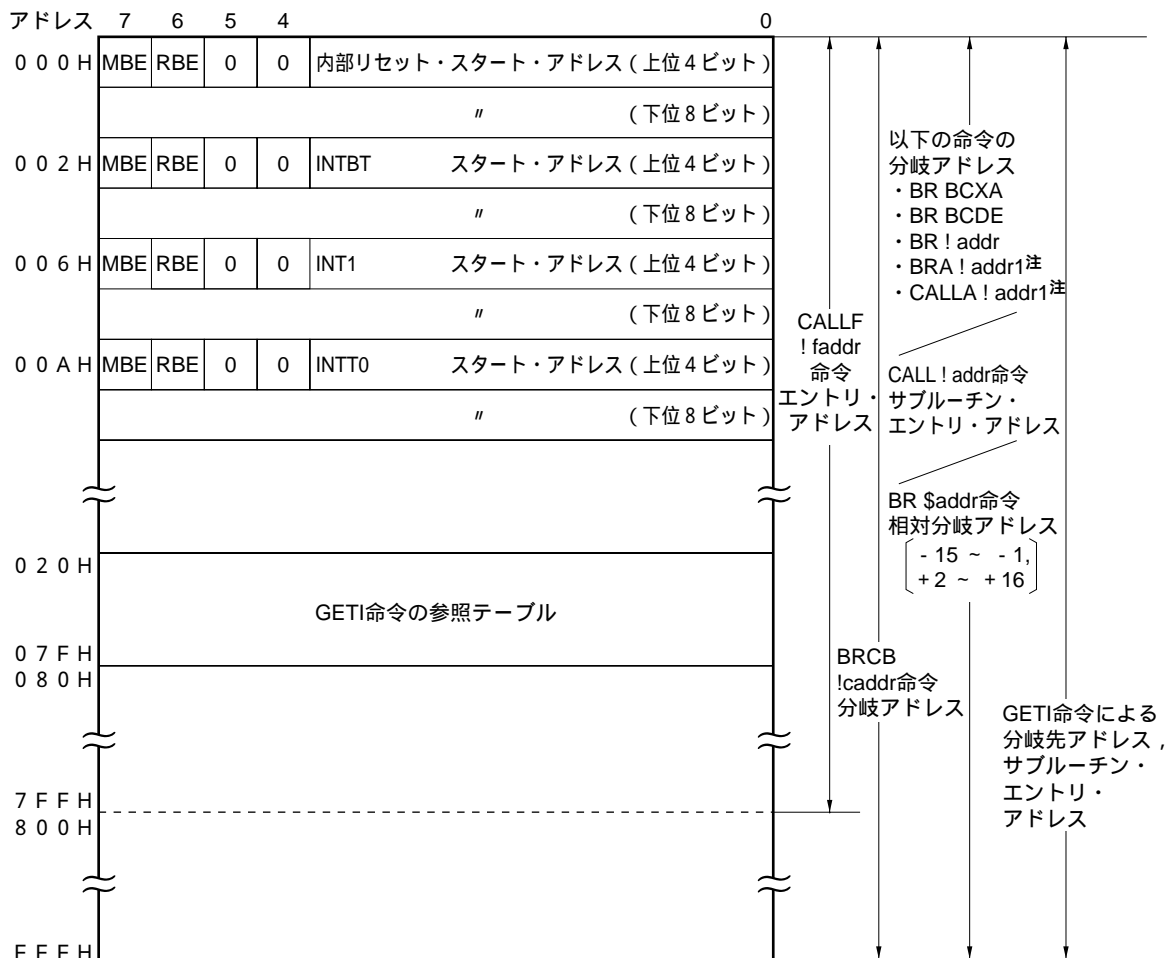
- ・ 0020H-007FH番地

GETI命令<sup>注</sup>で参照するテーブル領域です。

**注** GETI命令は、任意の2バイト/3バイト命令、または1バイト命令2つを1バイトで実現するための命令で、プログラム・ステップ数を縮小するために使用します (10.1.1 GETI命令参照)。



図4-3 プログラム・メモリ・マップ



注 Mk モード時のみ使用できます。

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

## 4.4 データ・メモリ (RAM) .....256ワード×4ビット

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

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

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

### 4.4.1 データ・メモリの構成

#### (1) データ・エリア

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

スタティックRAMは、メモリ・バンク0に256ワード×4ビットのみ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 (000H-0FFH) として使用できます。また、バンク1の1E0H-1F7H番地は表示データ・メモリ<sup>注1</sup>として使用します。

スタティックRAMは、1アドレスが4ビット構成です。しかし、8ビット・メモリ操作命令により8ビット単位で操作することも、ビット操作命令によりビット単位に操作することもできます<sup>注2</sup>。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1 . 書き込みのみ可能です。

2 . 表示データ・メモリは、8ビット操作はできません。

#### ・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)。

・表示データ・メモリ（書き込みのみ可能）

LCDの表示データを書き込むエリアです。LCD駆動時には、この表示データ・メモリに書き込まれたデータが、ハードウェアにより自動的に読み出され、表示されます。表示に使用していない部分をデータ・エリアとして使用することはできません。

（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。

#### 4.4.2 データ・メモリのバンクの指定

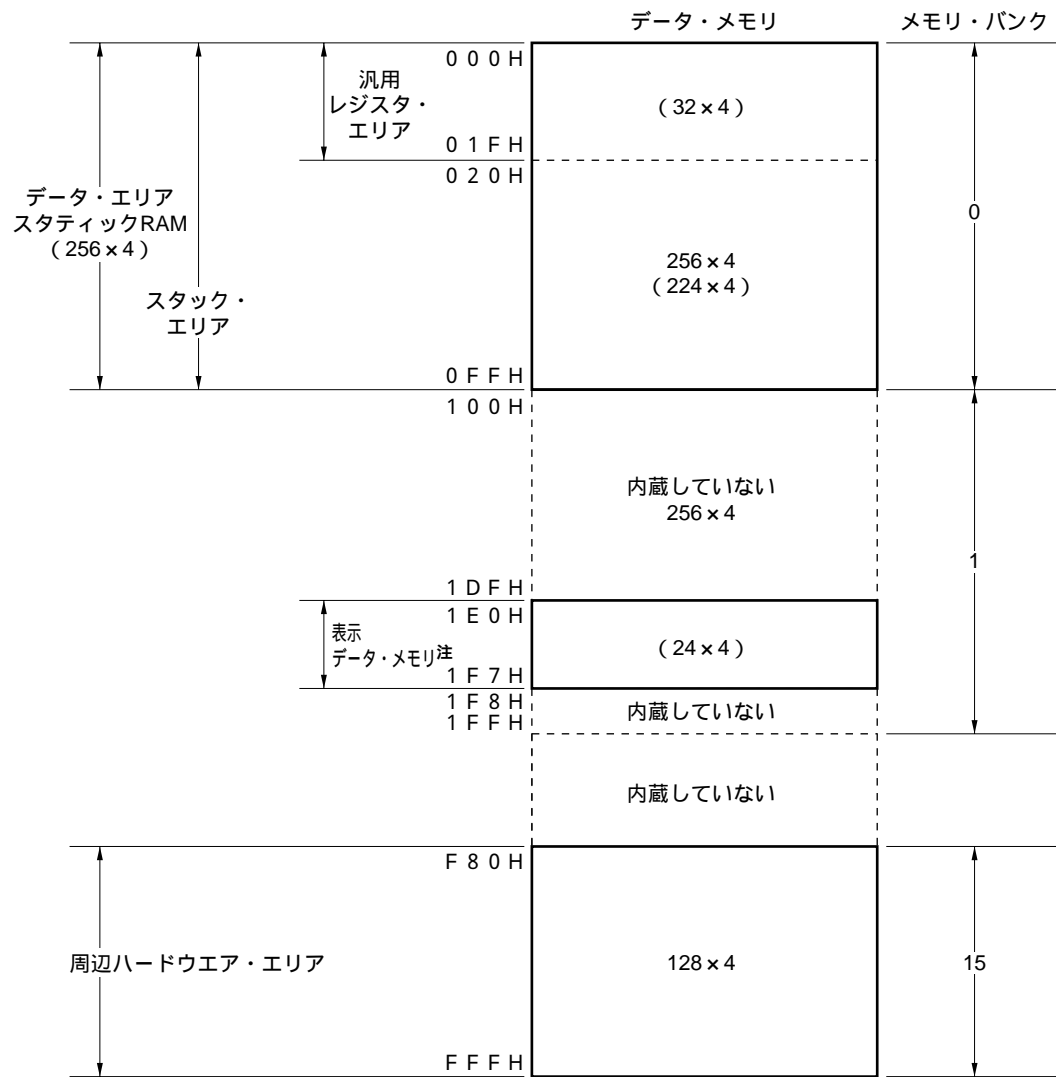
メモリ・バンクは、メモリ・バンク許可フラグ（MBE）によってバンク指定を許可したとき（MBE = 1）、4ビット・メモリ・バンク選択レジスタ（MBS）によって指定されます（MBS = 0, 1, 15）。バンク指定を禁止したとき（MBE = 0）、メモリ・バンクはそのときのアドレッシング・モードによってバンク0またはバンク15が自動的に指定されます。また、バンク内のアドレスは、8ビット・イミディエト・データやレジスタ・ペアなどによってアドレスされます。

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

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

- ・汎用レジスタ・エリア ..... 4.5 汎用レジスタ
- ・スタック・エリア ..... 4.7 スタック・ポインタ（SP）、スタック・バンク選択レジスタ（SBS）
- ・表示データ・メモリ ..... 5.6.6 表示データ・メモリ
- ・周辺ハードウェア・エリア ..... 第5章 周辺ハードウェア機能

図4-4 データ・メモリ・マップ



注 書き込みのみ可能です。

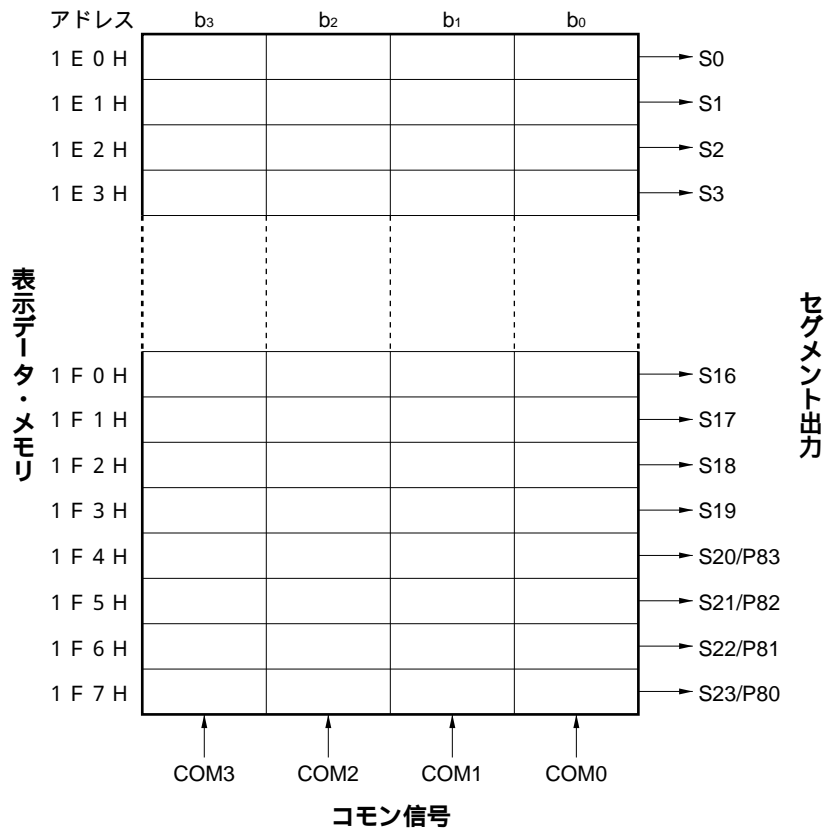
データ・メモリは、リセット時には不定となっています。このため、通常プログラムの初期でゼロにイニシャライズします（RAMクリア）。これを行わないと思わぬバグ発生の原因となりますので必ず行ってください。

**例** 000H-0FFH, 1E0H-1F7H番地のRAMをクリアする。

```
SET1  MBE
SEL    MB0
MOV    A, #0H
MOV    HL, #04H
RAMC0 : MOV  @HL, A      ; 04H-FFHをクリア注
        INCS HL          ; HL HL + 1
        BR   RAMC0
SEL    MB1
MOV    HL, #0E0H
RAMC1 : MOV  @HL, A      ; 1E0H-1F7Hをクリア
        INCS HL          ; HL HL + 1
        SKE  H, #0EH
        SKE  L, #8H
        BR   RAMC1
```

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

図4 - 5 表示データ・メモリの構成



- ★ 表示データ・メモリは4ビット単位で操作します。
- ★ **注意** 表示データ・メモリは、1ビット操作、8ビット操作はできません。また、書き込みのみ可能です。

例 1E0H-1F7Hの表示データ・メモリをクリアする。

```

SET1   MBE
SEL    MB1
MOV    HL, #0E0H
MOV    A, #0H
LOOP:  MOV  @HL, A      ; 4ビット単位で表示データ・メモリを一度に0クリアする。
       INCS HL
       SKE  H, #0EH
       SKE  L, #8H
       BR  LOOP
    
```

## 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のほかに、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 - 6 汎用レジスタの構成

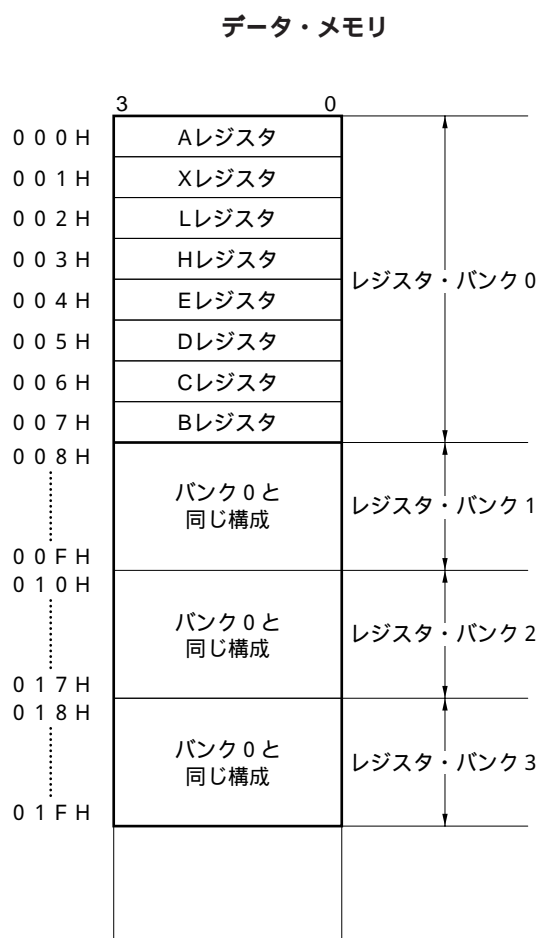
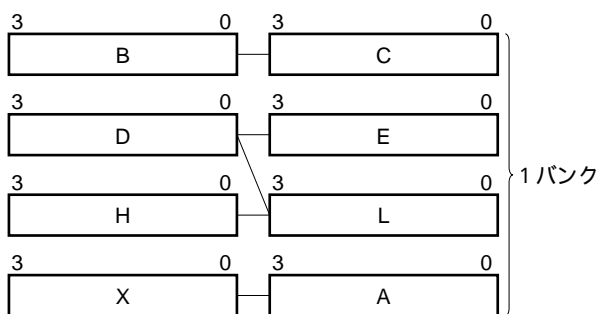


図4 - 7 レジスタ・ペアの構成

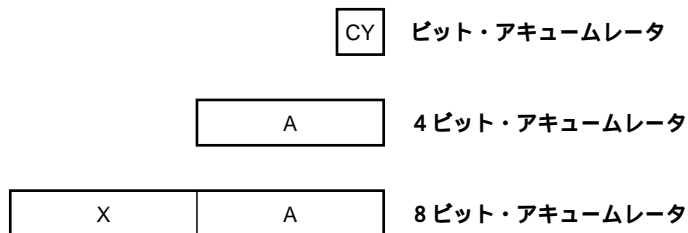


## 4.6 アキュムレータ

μPD753304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 - 8 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)

μPD753304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の000H-0FFH番地です。2ビットのSBSによってメモリ・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
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4-10～4-13のようになります。

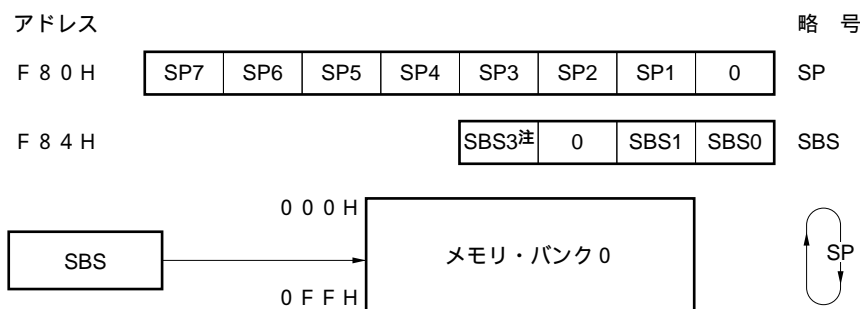
SPは、8ビット・メモリ操作命令、SBSは4ビット・メモリ操作命令によって初期値を設定し、スタック・エリアを決定します。また内容の読み出しもできます。

SPの初期値を00Hに設定すると、SBSで設定したメモリ・バンク0の最上位アドレス(0FFH)からスタックされます。

スタック・エリアはメモリ・バンク0に限られ、000H番地からさらにスタック動作を行うと、0FFHへ戻ります。

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4-9 スタック・ポインタおよびスタック・バンク選択レジスタの構成



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

例 SPのイニシャライズ

スタック・エリアをメモリ・バンク0とし、0FFH番地からスタック動作させる場合

```
SEL    MB15      ;またはCLR1 MBE
MOV    A, #8H
MOV    SBS, A    ;メモリ・バンク0をスタック・エリアに指定(Mkモード)
MOV    XA, #00H
MOV    SP, XA    ;SP 00H(0FFHからスタック動作)
```

図4 - 10 スタック・メモリへ退避されるデータ (Mk モード)

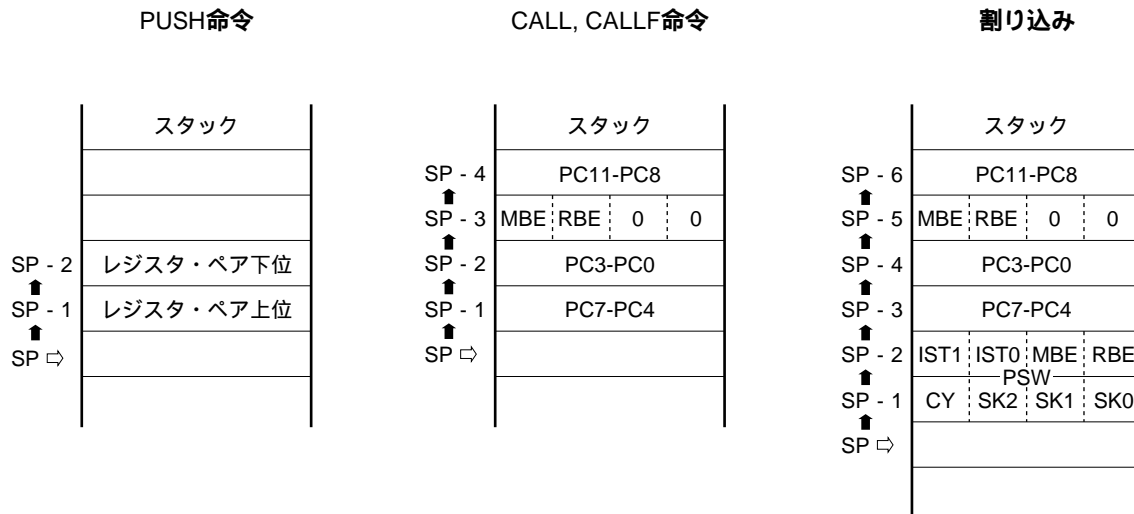


図4 - 11 スタック・メモリから復帰されるデータ (Mk モード)

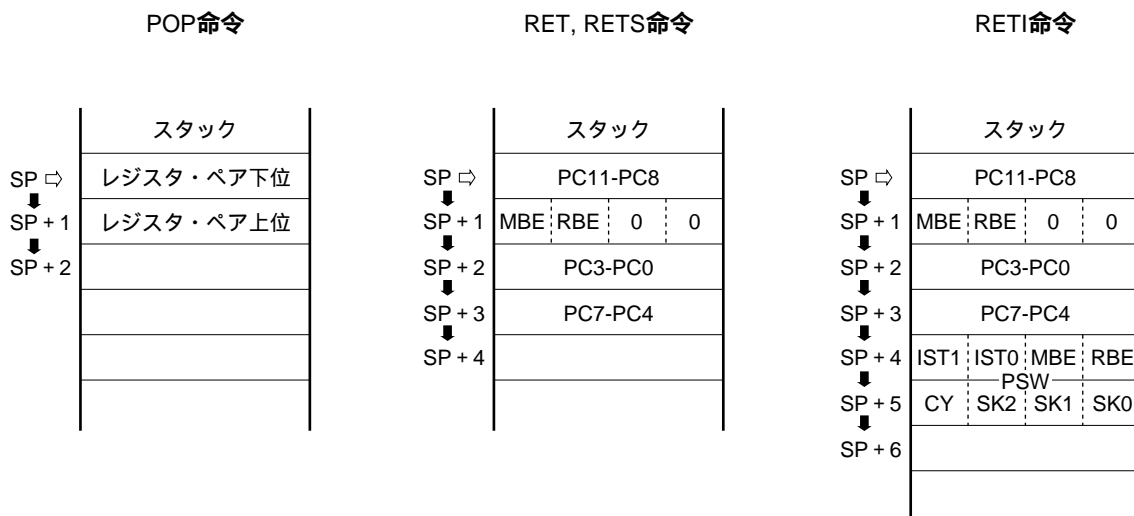


図4 - 12 スタック・メモリへ退避されるデータ (Mk モード)

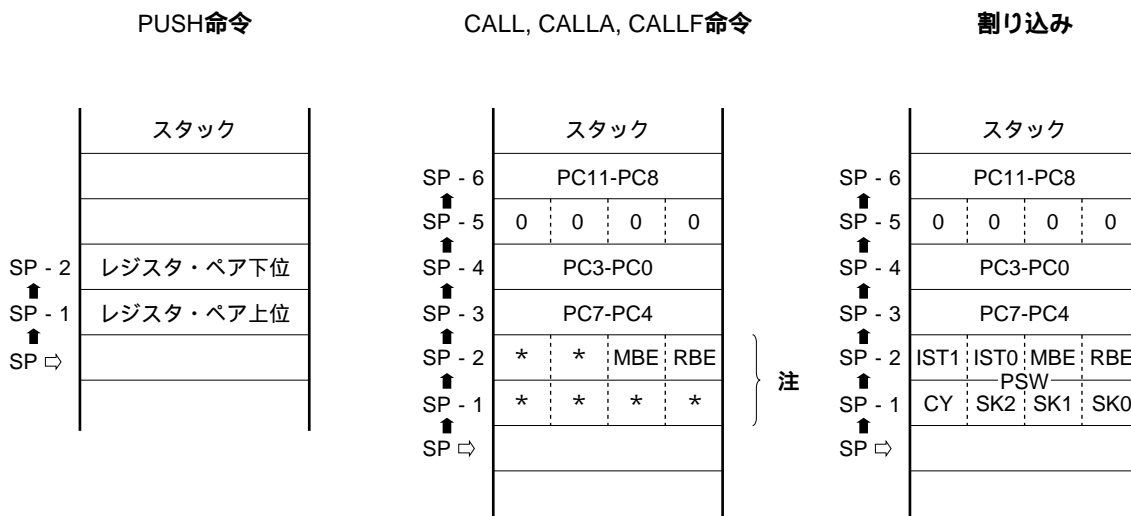
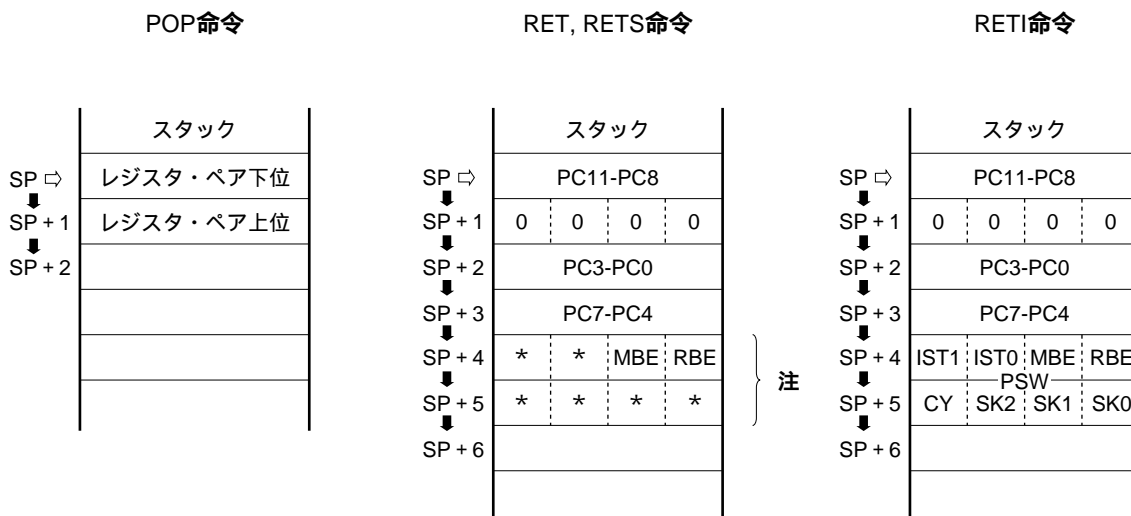


図4 - 13 スタック・メモリから復帰されるデータ (Mk モード)



注 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 - 14 プログラム・ステータス・ワードの構成

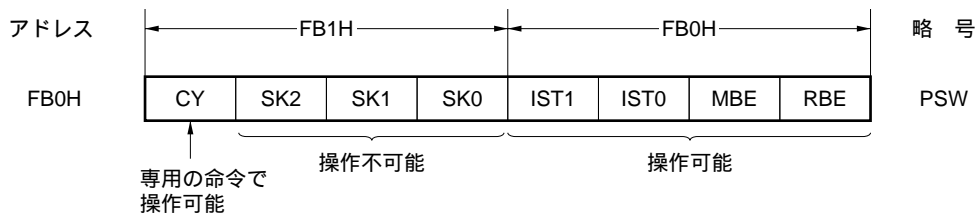


表4 - 3 スタック動作時に退避 / 復帰されるPSWのフラグ

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

### (1) キャリー・フラグ (CY)

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

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

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

RESET信号発生により、キャリー・フラグは不定となります。

表4 - 4 キャリー・フラグ操作命令

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

備考 mem\*. bitは次の3つのビット操作アドレッシングを示します。

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

例 3FH番地のビット3とP33のANDをとり, P80に出力する。

```
MOV    H, #3H          ;   Hレジスタに上位4ビット・アドレス・セット
MOV1   CY, @H+0FH.3   ;   CY 3FHのビット3
AND1   CY, PORT3.3    ;   CY CY P33
MOV1   PORT8.0, CY    ;   P80 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”にセットすると、データ・メモリ・アドレス空間はMBSの設定によります。

“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、各アドレッシング・モードのアドレッシング範囲参照)。

$\overline{\text{RESET}}$ 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、割り込みサービス時の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。

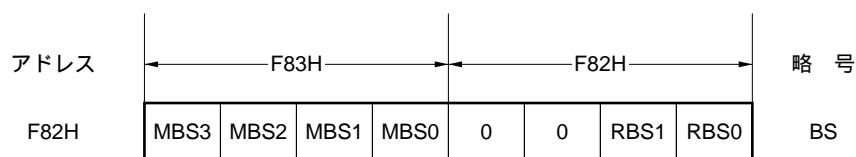
## 4.9 バンク選択レジスタ (BS)

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 - 15 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、μPD753304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。

RESET信号発生により、MBSは“0”にイニシャライズされます。

表4 - 6 MBE, MBSと選択されるメモリ・バンク

MBE	MBS				メモリ・バンク
	3	2	1	0	
0	x	x	x	x	メモリ・バンク0に固定
1	0	0	0	0	メモリ・バンク0を選択
	0	0	0	1	メモリ・バンク1を選択
	1	1	1	1	メモリ・バンク15を選択
上記以外					設定禁止

x : don't care



(2) レジスタ・バンク選択レジスタ (RBS)

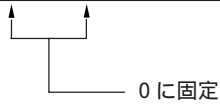
レジスタ・バンク選択レジスタは、汎用レジスタとして使用するレジスタ・バンクを指定するレジスタで、バンク0からバンク3まで設定できます。

RBSは、SEL RBn命令により設定します (n=0-3)。

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 - 7 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 "



x : don't care

(メモ)

## 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

μPD753304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	
FF3H	P33	P32	P31	P30	PORT3
FF8H	P83	P82	P81	P80	PORT8
FFAH	P103	P102	P101	P100	PORT10

入出力ポートの操作命令は表5 - 2のようになっており、PORT3, PORT10については4ビット入出力のほかビット操作が可能です。

**例1** . P33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8に出力。

```
SKT  PORT3.3    ; ポート3のビット3が1ならスキップ。
MOV  A, #8H     ; A 8H
MOV  A, #4H     ; A 4H } たてづみ
SEL  MB15       ; または, CLR1 MBE
OUT  PORT8, A   ; ポート8 A
```

2 . SET1 PORT10.@L ; ポート10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5-2～5-7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

ポート名	機能	動作，特徴	備考
PORT3	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。	PCL, BUZと端子を兼用。
PORT8		4ビット単位で入力または出力モードに設定可能。	S20-S23と端子を兼用。
PORT10		1ビット単位で入力または出力モードに設定可能。	INT1と端子を兼用。

図5-2 P30の構成

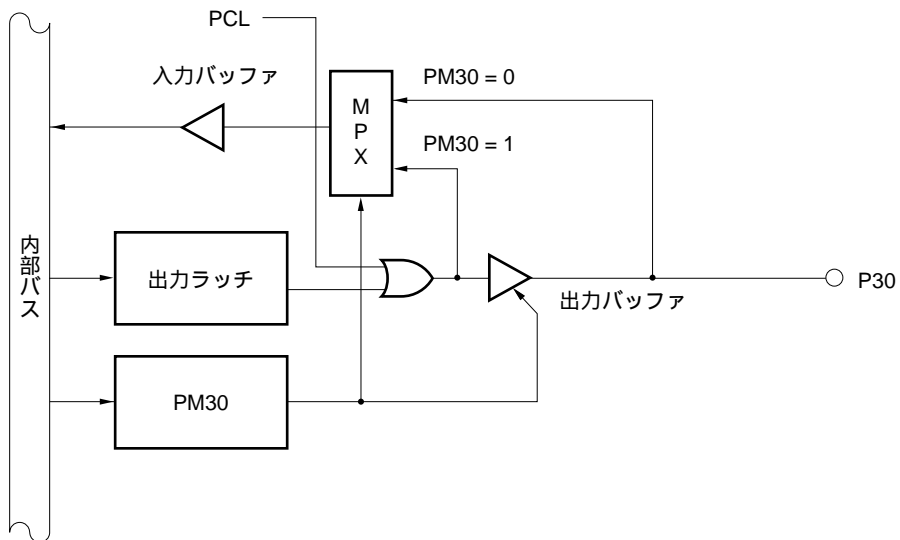


図5-3 P31の構成

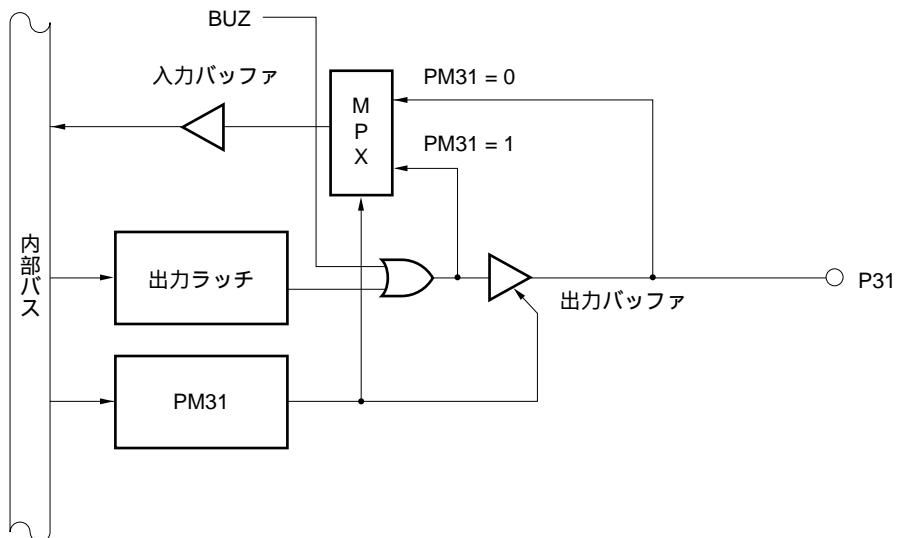


図5-4 P3nの構成 (n = 2, 3)

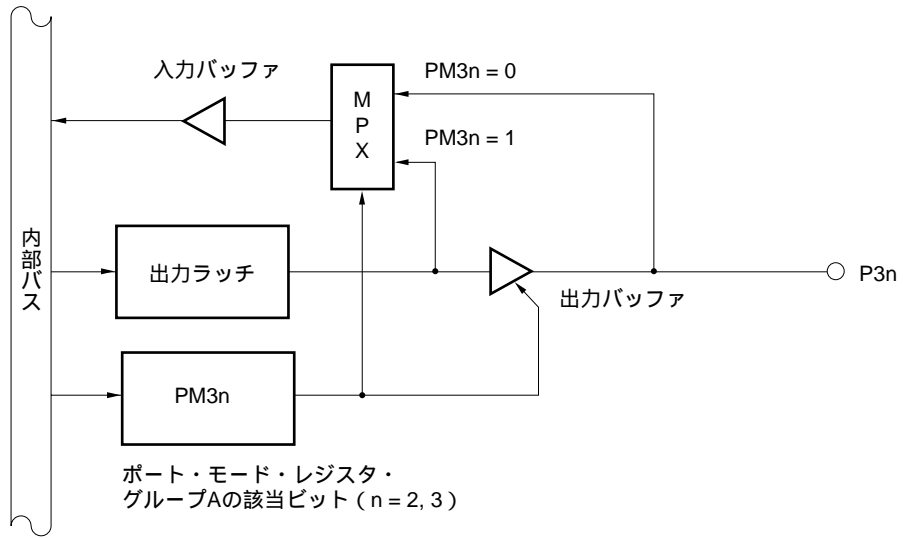


図5-5 ポート8の構成

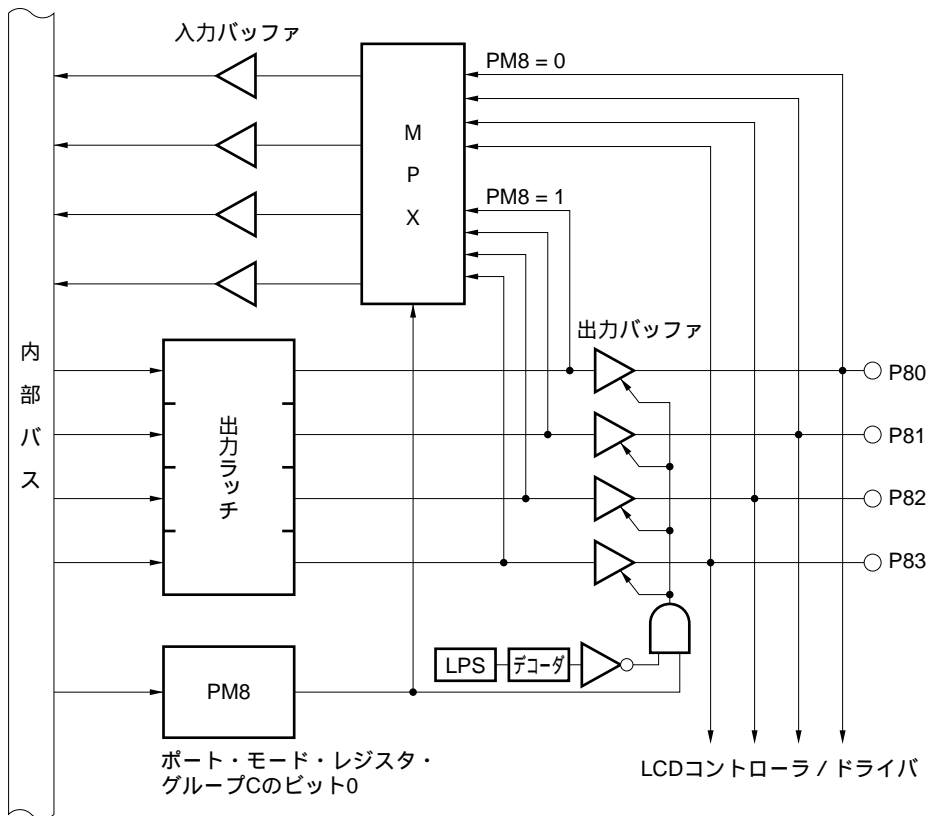


図5 - 6 P10nの構成 (n = 0-2)

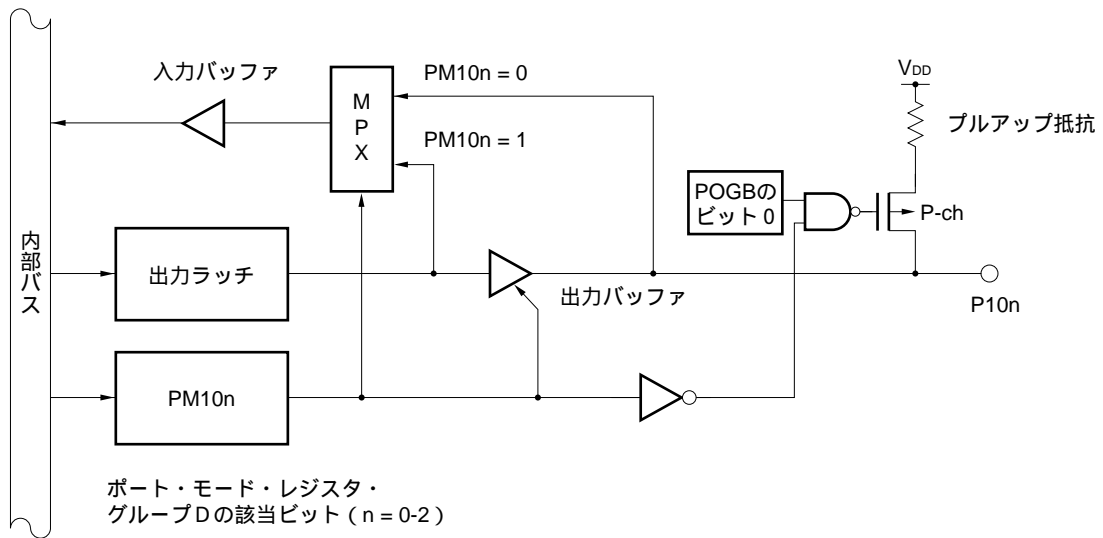
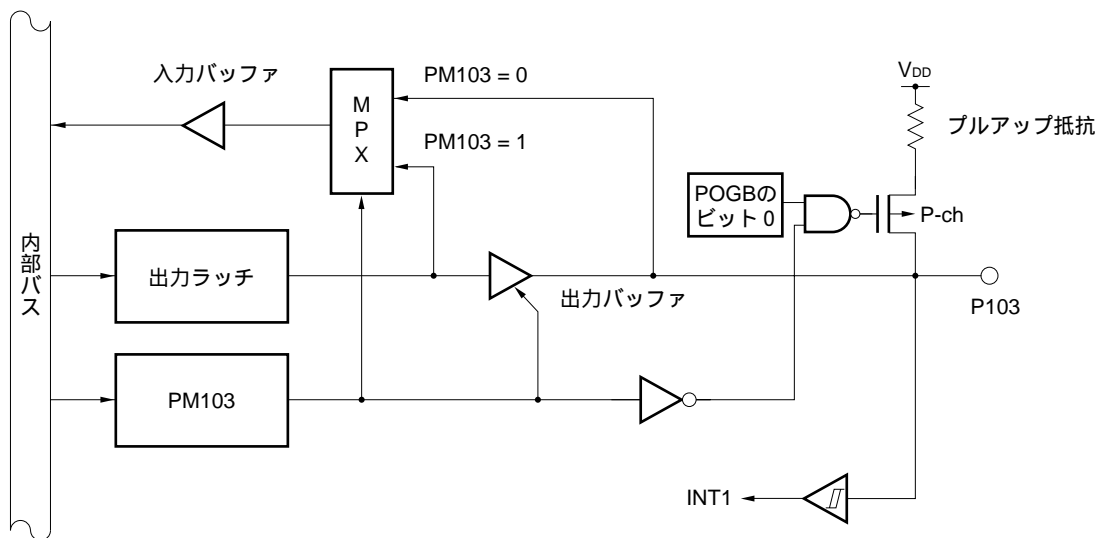


図5 - 7 P103の構成



## 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 - 8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はポート・モード・レジスタ・グループA (PMGA) によって、ポート10はポート・モード・レジスタ・グループD (PMGD) によって、ビットごとに入出力を指定できます。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、4ビット単位で入出力を指定します。

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

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

ポート・モード・レジスタ・グループA, C, D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・レジスタと各ポートは次のようになります。

ポート・モード・レジスタ・グループA (PMGA) : 0FH

ポート・モード・レジスタ・グループC (PMGC) : 00H

ポート・モード・レジスタ・グループD (PMGD) : 00H

★ ポート3 (P30-P33) : マスク・オプションにより、リセット時の入力/出力モードの指定が可能です。詳細は、表2 - 2を参照してください。

ポート8 (P80-P83) : 入力モード

ポート10 (P100-P103) : 入力モード (プルアップ抵抗付き)

**例** P30, 31を入力, P32, 33を出力端子として使う。

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

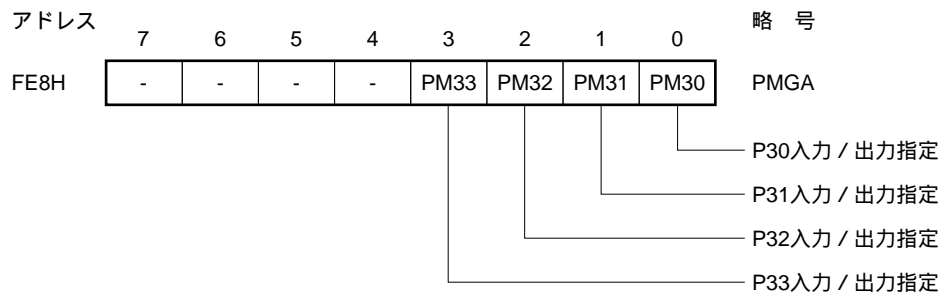
```
MOV   XA, #0CH
```

```
MOV   PMGA, XA
```

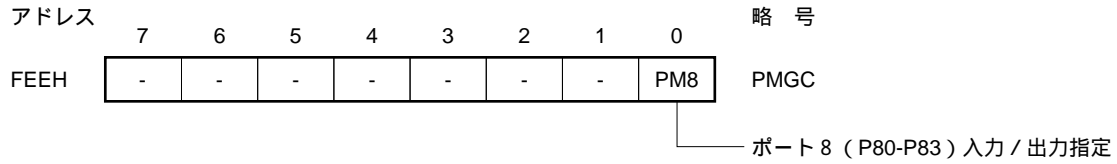
図5 - 8 各ポート・モード・レジスタのフォーマット

	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)

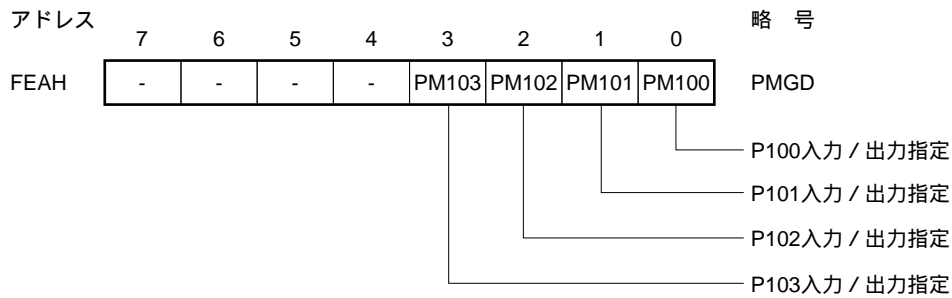
ポート・モード・レジスタ・グループA



ポート・モード・レジスタ・グループC



ポート・モード・レジスタ・グループD





### 5.1.3 デジタル入出力ポート操作命令

μPD753304に内蔵されている入出力ポートはすべてデータ・メモリ空間にマッピングされていますから、データ・メモリ操作命令はすべて適用することができます。データ・メモリ操作命令のうち、特に入出力端子の操作に有効と思われる命令とその適用範囲を表5 - 2にまとめます。

#### (1) ビット操作命令

デジタル入出力ポートPORT3, 8, 10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** P30とP31のORをとってP81に出力する。

```
MOV1    CY, PORT3.0        ; CY P30
OR1     CY, PORT3.1        ; CY CY P31
MOV1    PORT8.1, CY        ; P81 CY
```

#### (2) 4ビット操作命令

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

**例1** . ポート3に、アキュムレータの内容を出力する。

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

**2** . ポート8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, #PORT8
ADDS   A, @HL                ; A A + PORT8
NOP
MOV     @HL, A                ; PORT8 A
```

**3** . ポート10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, #PORT10
SUBS   A, @HL                ; A < PORT10
BR     NO                    ; NO
                                ; YES
```

表5 - 2 入出力端子操作命令一覧表

命令	PORT	PORT 3	PORT 8	PORT 10
IN	A, PORTn 注1			
IN	XA, PORTn		-	
OUT	PORTn, A 注1			
OUT	PORTn, XA		-	
MOV	A, PORTn 注1			
MOV	PORTn, A 注1			
XCH	A, PORTn 注1			
MOV1	CY, PORTn. bit			
MOV1	CY, PORTn. @L 注2			
MOV1	PORTn. bit, CY			
MOV1	PORTn. @L, CY 注2			
INCS	PORTn 注1			
SET1	PORTn. bit			
SET1	PORTn. @L 注2			
CLR1	PORTn. bit			
CLR1	PORTn. @L 注2			
SKT	PORTn. bit			
SKT	PORTn. @L 注2			
SKF	PORTn. bit			
SKF	PORTn. @L 注2			
SKTCLR	PORTn. bit			
SKTCLR	PORTn. @L 注2			
AND1	CY, PORTn. bit			
AND1	CY, PORTn. @L 注2			
OR1	CY, PORTn. bit			
OR1	CY, PORTn. @L 注2			
XOR1	CY, PORTn. bit			
XOR1	CY, PORTn. @L 注2			

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

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

### 5.1.4 デジタル入出力ポートの動作

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

#### (1) 入力モードに設定されているときの動作

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

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

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

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

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

#### (2) 出力モードに設定されているときの動作

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

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

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

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

表5 - 3 入出力ポート操作時の動作

実行する命令	ポートおよび各端子の動作	
	入力モード	出力モード
SKT <input type="text"/> SKF <input type="text"/>	端子のデータをテスト	出力ラッチのデータをテスト
MOV1 CY, <input type="text"/>	端子のデータをCYへ転送	出力ラッチのデータをCYへ転送
AND1 CY, <input type="text"/> OR1 CY, <input type="text"/> XOR1 CY, <input type="text"/>	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 MOV A, 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 MOV PORTn, A 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 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	出力ラッチの内容を + 1
SET1 <input type="text"/> CLR1 <input type="text"/> MOV1 <input type="text"/> , CY SKTCLR <input type="text"/>	にラッチ 指定したビットの出力ラッチは命令の指示通り書き替わるが、その他のビットの出力ラッチは不定	出力端子の状態を命令に従って変更

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

### 5.1.5 プルアップ抵抗の内蔵

μPD753304のポート10はプルアップ抵抗を内蔵することができます。プルアップ抵抗の内蔵の指定はソフトウェアにより行い、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5-9に示したフォーマットに基づいて行います。

内蔵プルアップ抵抗の接続指定は、入力モードに指定されている端子にのみ有効です。出力モードに指定された端子は、POGBの設定にかかわらず、内蔵プルアップ抵抗の接続を指定できません。

RESET信号発生により、POGBは01Hにイニシャライズされます。

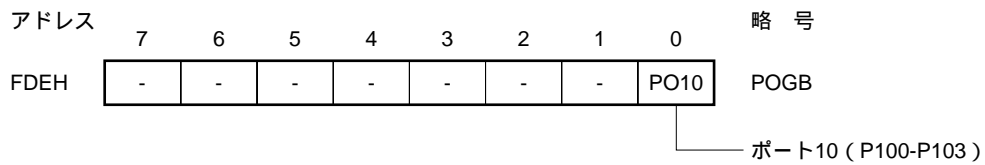
表5-4 プルアップ抵抗内蔵の指定法

ポート (端子名)	プルアップ抵抗内蔵の指定方法	指定ビット
ポート10 (P100-P103)	4ビット単位でソフトウェアにより接続を指定	POGB.0

図5-9 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



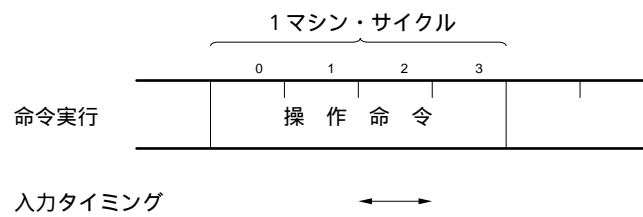
### 5.1.6 デジタル入出力ポートの入出力タイミング

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 - 10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 - 11に示します。

図5 - 10 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

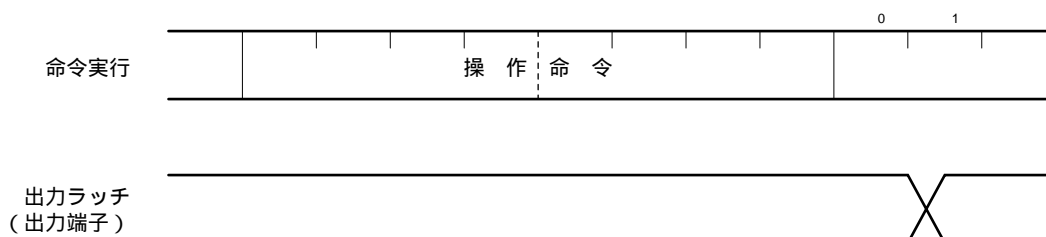
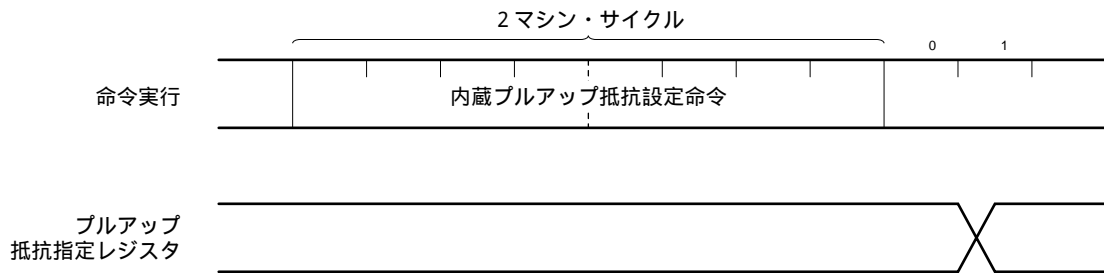


図5 - 11 ソフトウェアによる内蔵プルアップ抵抗のONタイミング



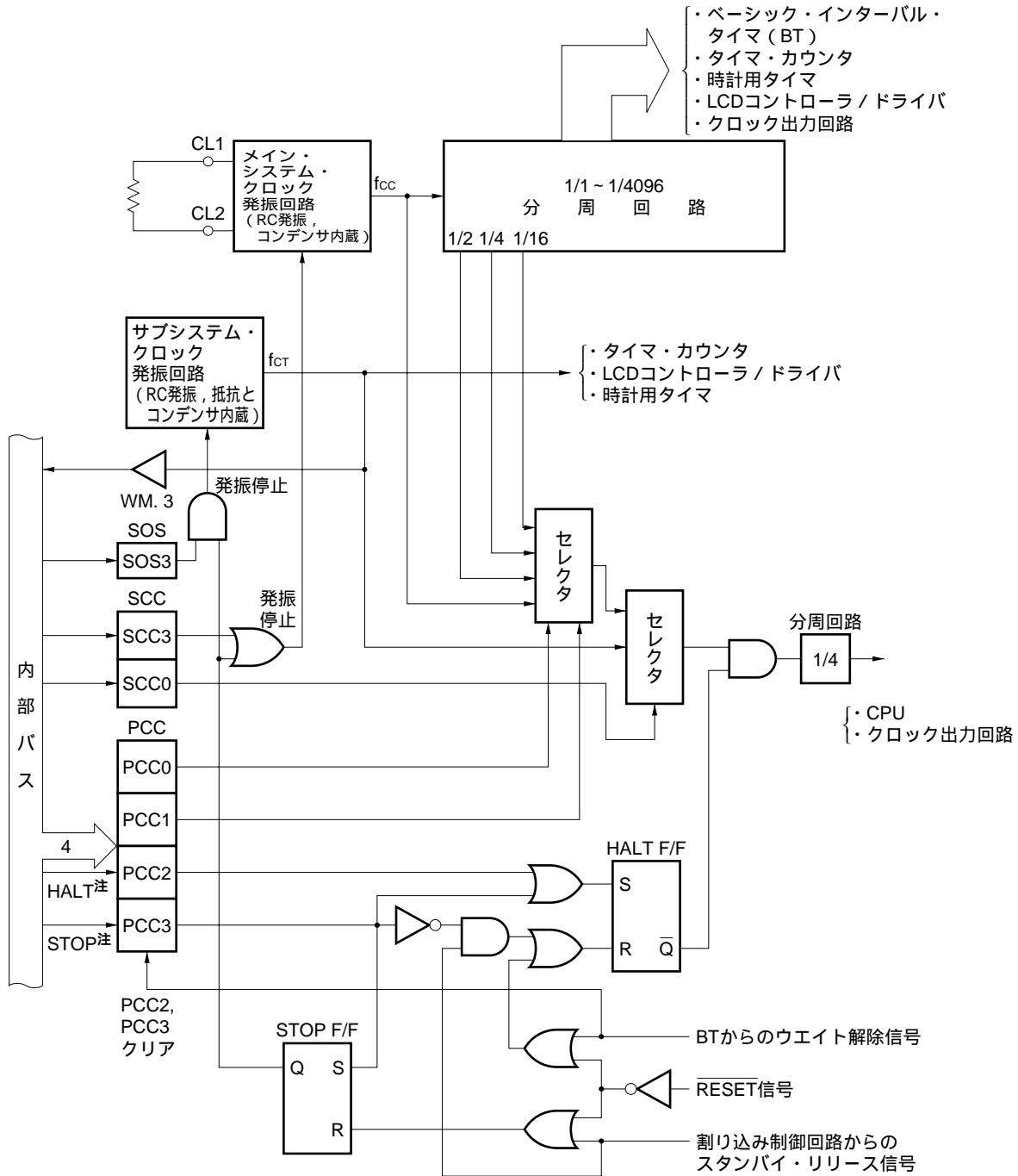
## 5.2 クロック発生回路

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 - 12のように構成されています。

図5-12 クロック発生回路のブロック図



注 命令の実行

備考1 . fcc = メイン・システム・クロック周波数

2 . fct = 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 fcc
- ・サブシステム・クロック fcr
- ・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)、システム・クロック・コントロール・レジスタ (SCC) およびサブ発振回路コントロール・レジスタ (SOS) により決定され、以下のような機能、動作となります。

- ( a ) 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 ( 17.8  $\mu$ s : 3.6 MHz動作時 ) が選択されます ( PCC = 0, SCC = 0 )。
- ( b ) メイン・システム・クロックを選択した状態でPCCの設定により4段階のCPUクロックを選択することができます ( 1.1, 2.2, 4.4, 17.8  $\mu$ s : 3.6 MHz動作時 )。
- ( c ) 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 ( 85.1  $\mu$ s : 47 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モードとSTOPモードのいずれも使用することもできます。SOSのビット3を“1”にセットしたあとに、STOP命令を実行するとサブシステム・クロックの発振を停止することができます。
- ( 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、メイン・システム・クロックの発振を停止した状態でも、時計機能と時計用タイマからのクロックで動作するタイマ・カウンタ、LCDコントローラ/ドライバおよび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タイマ・カウンタ, LCDコントローラ/ドライバおよび時計用タイマは正常動作を継続することができます。しかし, ほかのハードウェアはメイン・システム・クロックによって動作しますので, メイン・システム・クロックを停止させたときは使用することはできません。

#### (1) プロセッサ・クロック・コントロール・レジスタ (PCC)

PCCは, 下位2ビットでCPUクロックの選択, 上位2ビットでCPU動作モードの制御を行う4ビット・レジスタです(図5-13参照)。

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<sub>CT/4</sub>に固定されます。また, STOP命令はメイン・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実行可能です。

**例1** . マシン・サイクルを最高速モード(1.1 μs : 3.6 MHz動作時)にする。

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

**2** . マシン・サイクルを2.2 μs (3.6 MHz動作時)にする。

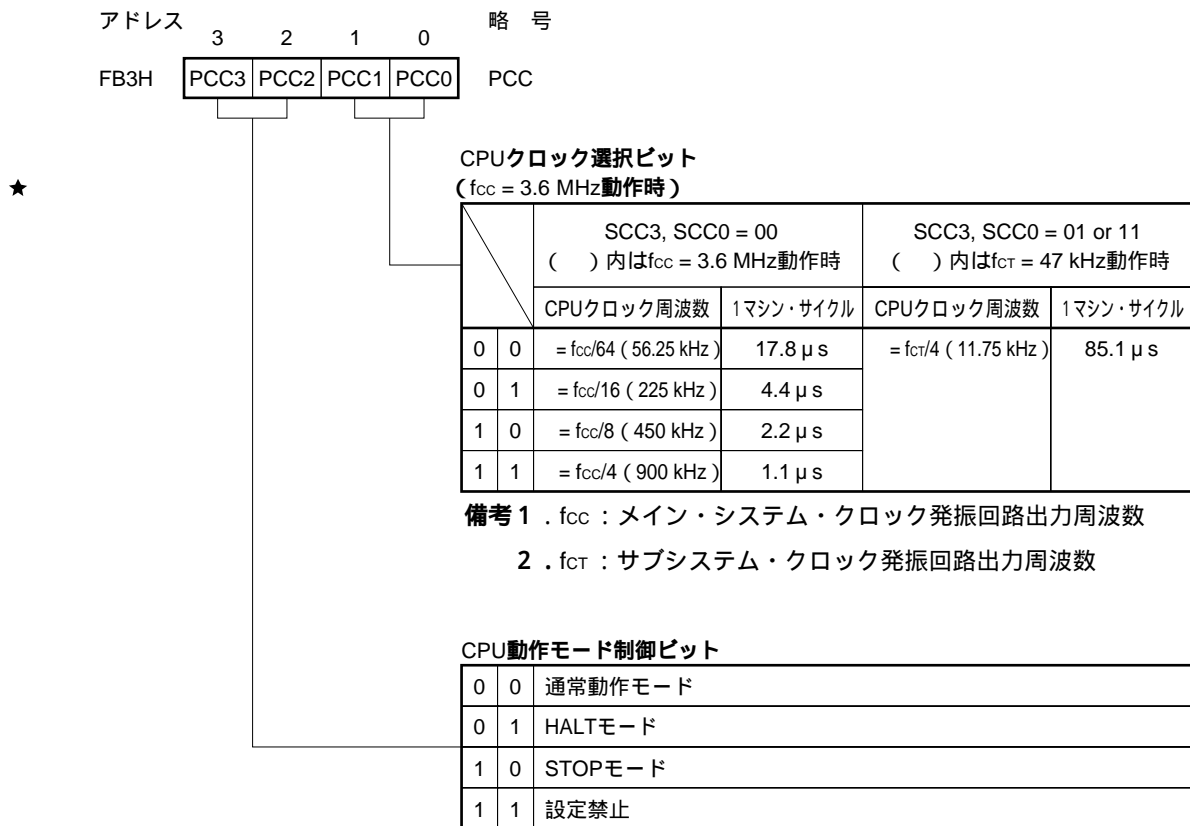
```
SEL  MB15
MOV  A, #0010B
MOV  PCC, A
```

**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
```

RESET信号発生によりPCCは“0”にクリアされます。

図5 - 13 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 - 14参照)。

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がぎり許されます。メイン・システム・クロックで動作しているときの発振停止はSTOP命令により行います。

RESET信号発生によりSCCは“0”にクリアされます。

図5 - 14 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 $1/f_{CT}$ の時間が必要です。したがって、メイン・システム・クロックの発振を停止させる場合は、サブシステム・クロック変更後、表5 - 5に示すマシン・サイクル以上経過後、SCC. 3を1に設定してください。
- 2 . メイン・システム・クロックで動作しているときにSCC. 3をセットして発振を停止させても正常なSTOPモードには入りません。

(3) 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CL1, CL2端子に図5 - 15のように抵抗 ( R ) を接続することによって発振します。外部クロックを入力することはできません。

メイン・システム・クロック発振回路出力周波数 (  $f_{CC}$  ) と抵抗値 ( R ) , コンデンサ容量値 ( C ) 注の関係式は次のようになります。

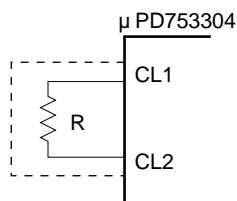
注 10 pF ( typ. ) のコンデンサ ( C ) を内蔵しています。

$$f_{CC} = \frac{1}{2RC}$$

注意  $f_{CC}$  は電源電圧や温度等の変動により周波数偏差を持つ場合があります。

図5 - 15 メイン・システム・クロック発振回路の外付け回路

RC発振



サブシステム・クロック発振回路は、外部端子を持っていないため、抵抗（R）、コンデンサ（C）を接続することはできません（抵抗、コンデンサとも内蔵しています）。また、外部クロックを入力することもできません。

**注意** メイン・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5-15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。

ほかの信号線と交差させない。

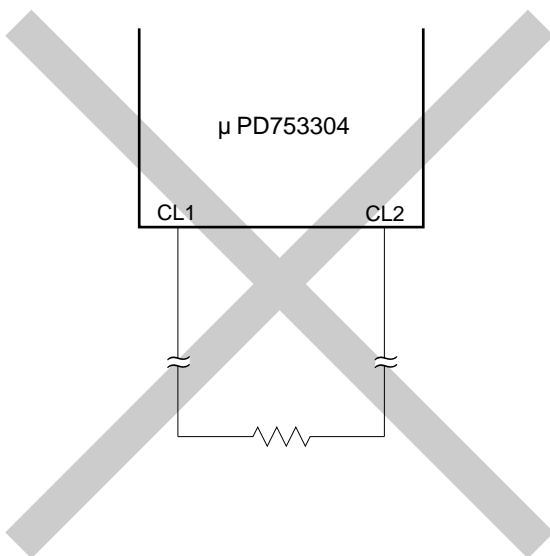
変化する大電流が流れる線と接近させない。

発振回路から信号を取り出さない。

図5-16に接続回路の悪い例を示します。

図5-16 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

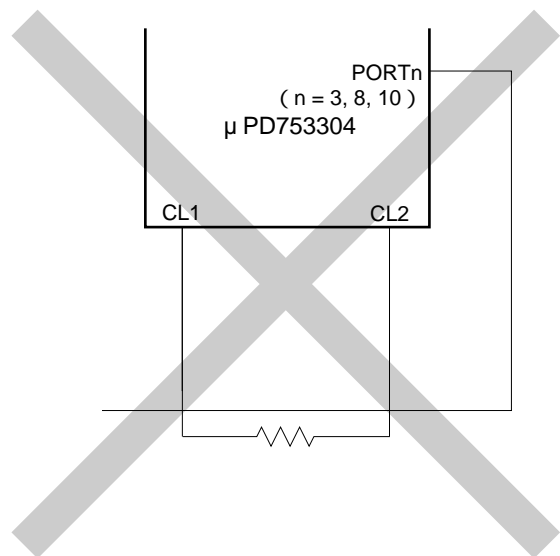
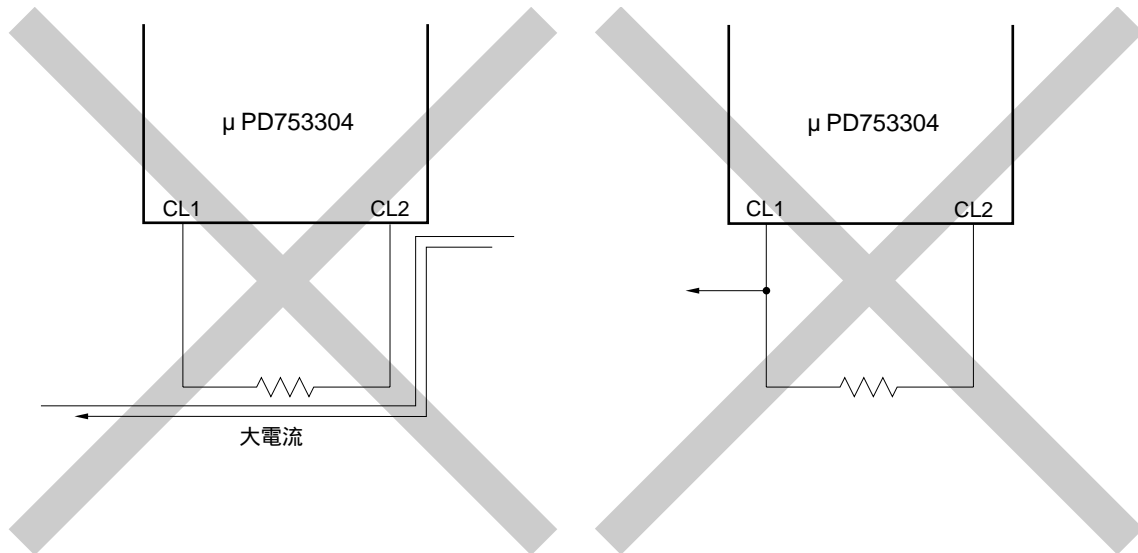


図5-16 接続回路の悪い例(2/2)

(c) 変化する大電流が信号線に  
近接している

(d) 信号を取り出している



(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 (fcc) を分周して、各種クロックを生成します。

(5) サブ発振回路コントロール・レジスタ (SOS)

SOSは最上位ビットでサブシステム・クロックの発振停止制御を行う4ビットのレジスタです(図5-17参照)。SOSのビット3を“1”にセットしたあとに、STOP命令を実行するとサブシステム・クロックは発振停止します。メイン・システム・クロック、サブシステム・クロックのいずれでもSTOP命令を実行可能です。

RESET信号発生によりSOSは“0”にクリアされます。

図5-17 サブ発振回路コントロール・レジスタ (SOS) のフォーマット



### 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			$\frac{f_{cc}}{64f_{ct}}$ マシン・サイクル (2マシン・サイクル)		
	0	1				4マシン・サイクル			4マシン・サイクル			4マシン・サイクル			$\frac{f_{cc}}{16f_{ct}}$ マシン・サイクル <sup>注</sup> (5マシン・サイクル)		
	1	0				8マシン・サイクル			8マシン・サイクル			8マシン・サイクル			$\frac{f_{cc}}{8f_{ct}}$ マシン・サイクル (10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			$\frac{f_{cc}}{4f_{ct}}$ マシン・サイクル (20マシン・サイクル)		
1	x	x	1マシン・サイクル			1マシン・サイクル <sup>注</sup>			1マシン・サイクル			1マシン・サイクル					

注 ツールではエミュレーションできません。

**注意** 発振子の周囲温度，負荷容量の性能のばらつきなどの条件により， $f_{cc}$ と $f_{ct}$ の値は変化します。特に $f_{cc}$ が公称値より高い場合や， $f_{ct}$ が公称値より低い場合には，表中の $\frac{f_{cc}}{64f_{ct}}$ ， $\frac{f_{cc}}{16f_{ct}}$ ， $\frac{f_{cc}}{8f_{ct}}$ ， $\frac{f_{cc}}{4f_{ct}}$ の式で求められるマシン・サイクルは， $f_{cc}$ ， $f_{ct}$ 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， $f_{cc}$ ， $f_{ct}$ の公称値で求めたマシン・サイクルよりも長くしてください。

備考1．( )内は， $f_{cc} = 3.6 \text{ MHz}$ ， $f_{ct} = 47 \text{ kHz}$ の場合。

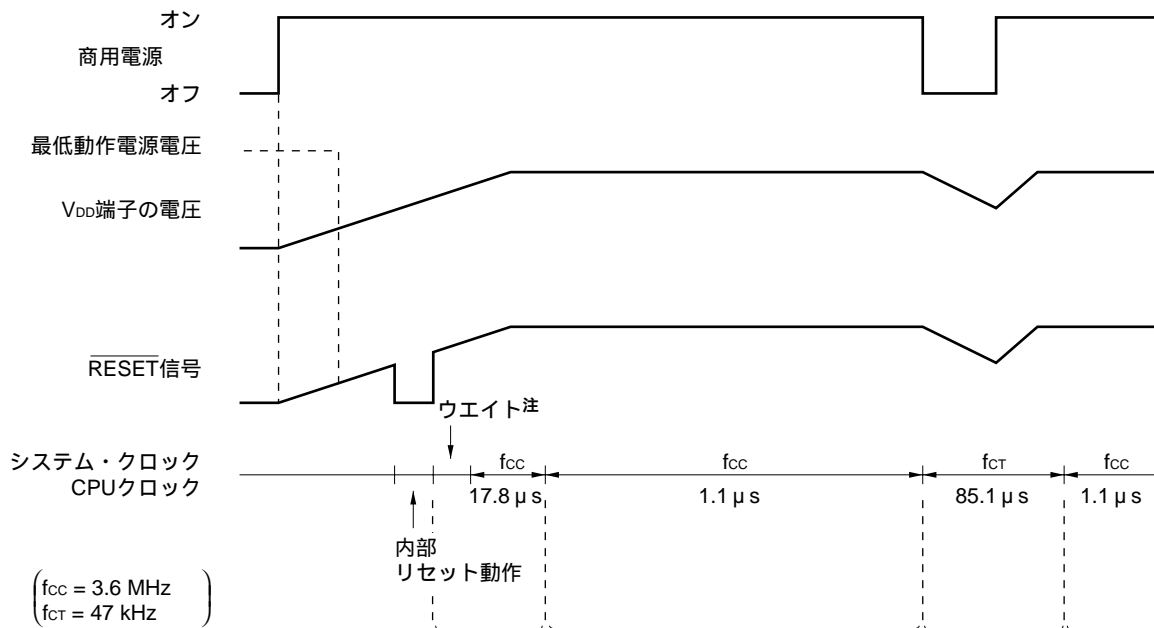
2．x : don't care

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

(2) システム・クロックとCPUクロックの切り替え手順

システム・クロックとCPUクロックの切り替えを図5 - 18に従って説明します。

図5 - 18 システム・クロックとCPUクロックの切り替え



RESET信号発生により、発振安定時間確保のためのウエイト時間<sup>注</sup>後、CPUはメイン・システム・クロックの最低速（17.8 μs : 3.6 MHz動作時）で動作をスタートします。

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

商用電源がオフしたことを割り込み入力などにより検出し、SCCのビット0を“1”にしサブシステム・クロックで動作します（このとき、あらかじめ、サブシステム・クロックの発振が開始していなければなりません）。そしてサブシステム・クロックに切り替わるのに必要な時間（20マシン・サイクル）経過後、SCCのビット3を“1”にセットしてメイン・システム・クロックの発振を停止します。

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

注 56/fcc（15.6 μs : 3.6 MHz動作時）に固定されています。



## 5.2.4 クロック出力回路

### (1) クロック出力回路の構成

クロック出力回路は、図5-19のように構成されています。

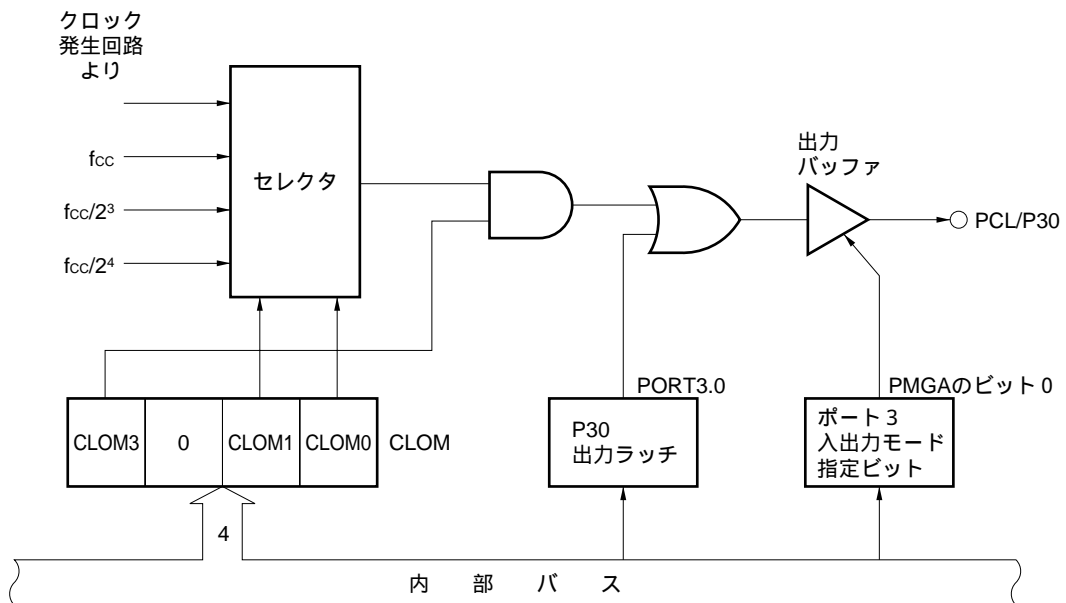
### (2) クロック出力回路の機能

クロック出力回路は、PCL/P30端子からクロック・パルスを出力するための回路で、リモコン波形出力への応用や周辺LSIにクロック・パルスを供給する場合などに利用します。

クロック・パルスを出力するときは、次の手順で行います。

- (a) クロック出力周波数を選択する。クロックの出力は禁止。
- (b) P30の出力ラッチに0を書き込む。
- (c) ポート3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5-19 クロック出力回路のブロック図



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

(3) クロック出力モード・レジスタ (CLOM)

CLOMは、クロック出力を制御する4ビット・レジスタです。

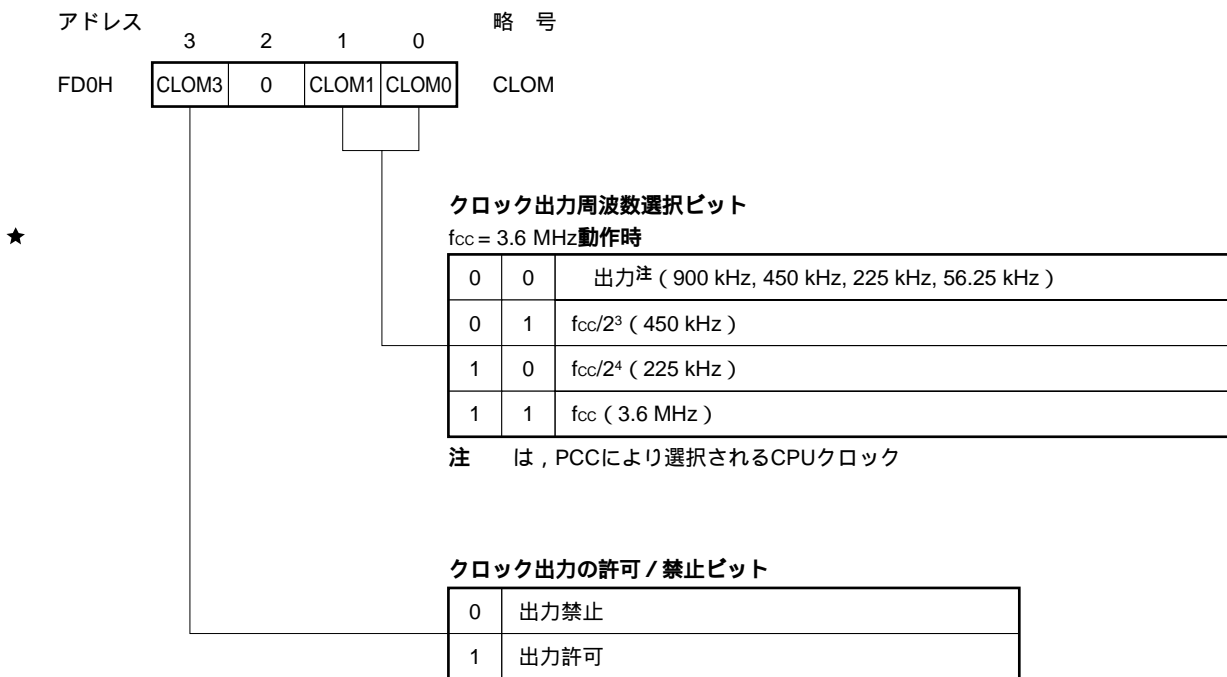
CLOMは、4ビット・メモリ操作命令で設定します。

```

例 CPUクロック をPCL/P30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 - 20 クロック出力モード・レジスタのフォーマット

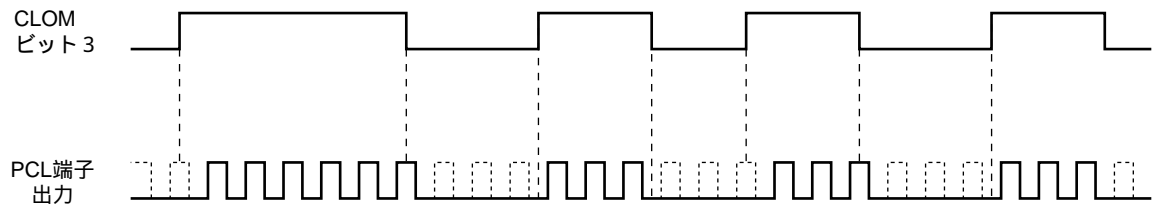


注意 CLOMのビット2には、必ず0を設定してください。

(4) リモコン波形出力への応用例

μPD753304のクロック出力機能は、リモコン波形出力に応用可能です。リモコン波形出力のキャリア周波数は、クロック出力モード・レジスタのクロック周波数選択ビットにより選択します。パルスの出力の許可/禁止はクロック出力の許可/禁止ビットをソフトウェアで制御することにより行います。クロック出力の許可/禁止切り替え時に、幅の狭いパルスは出力されないよう考慮されています。

図5 - 21 リモコン波形出力応用例



## 5.3 ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ

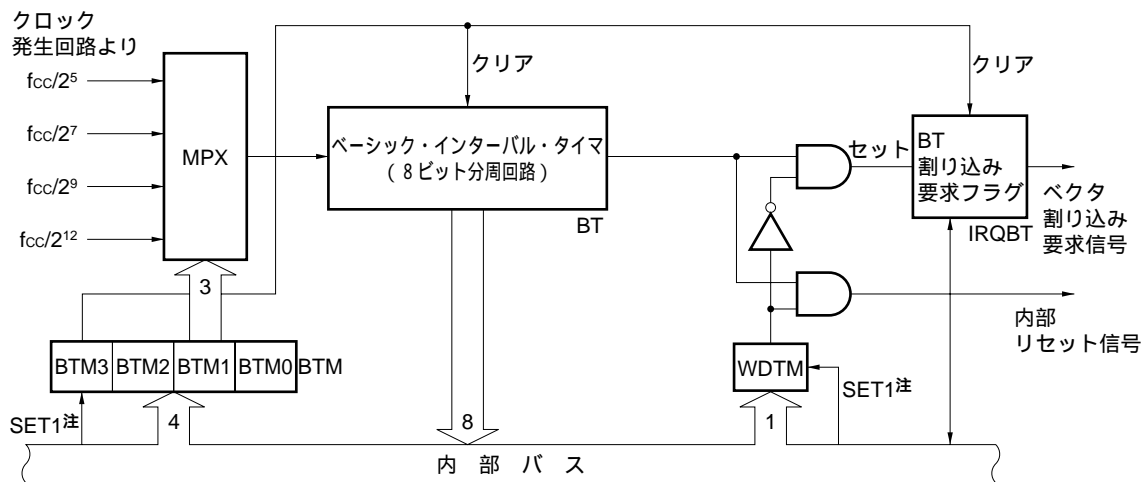
$\mu$ PD753304は8ビット・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマを備えており、次のような機能があります。

- (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) 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5-22のように構成されています。

図5-22 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。

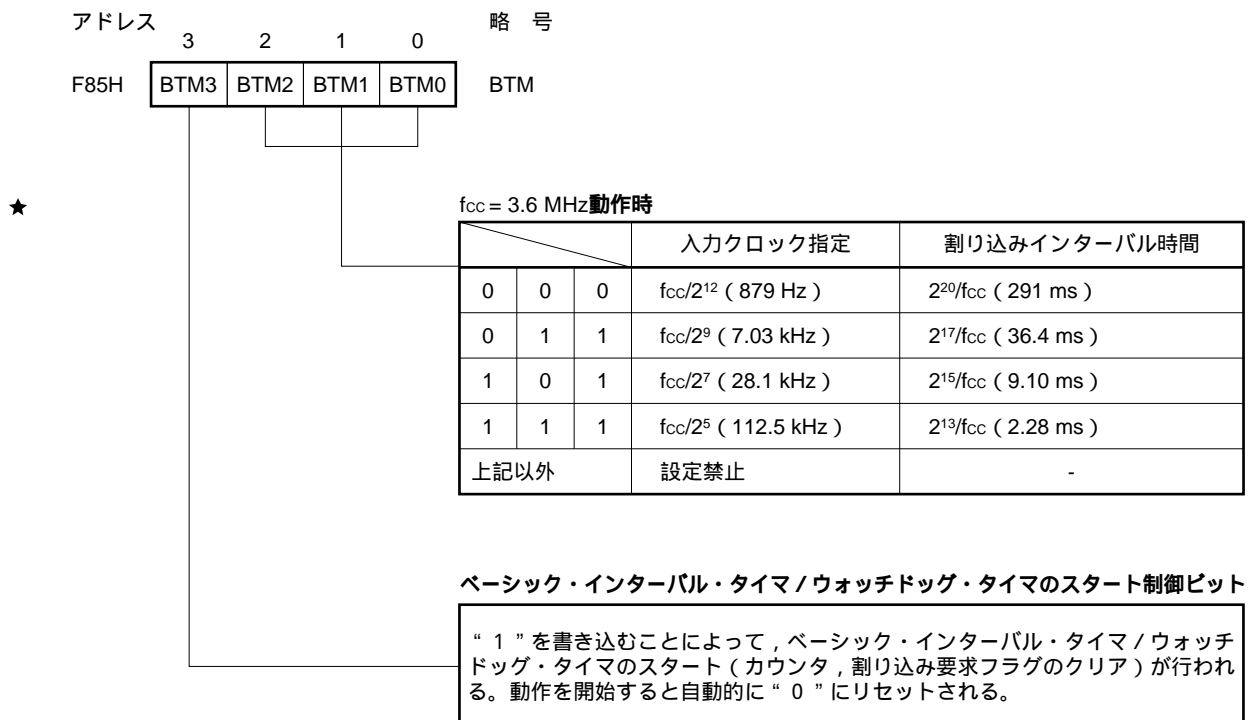
**例** 割り込み発生インターバルを2.28 ms (3.6 MHz動作時) に設定

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

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

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 - 23 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の発生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。一度セットすると、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
```

RESET信号発生により内容は0にクリアされます。

図5 - 24 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5-23参照）。

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

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

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

BTMにインターバル時間をセットする。

BTMのビット3に“1”をセットする。

**例** 2.28 ms（3.6 MHz動作時）ごとに割り込みを発生する。

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

### 5.3.5 ウォッチドッグ・タイマの動作

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

ウォッチドッグ・タイマ・モードでは、BTのオーパフローするインターバル時間を利用して、プログラムの暴走を検出します。このインターバル時間は、BTMのビット2-0の設定により4通りの時間が選択できます (図5-23参照)。これらの中からユーザ・システムに応じて暴走検出に必要な時間を決めてください。インターバル時間を設定しておいて、プログラムをその時間内に実行できる単位に分割し、それぞれの単位の最後でBTをクリアする命令を実行させるようにします。そうすると、設定時間内にこのBTクリアを実行する命令にたどりつかなければ (順調にプログラムの実行が進んでいなければ = 暴走) BTはオーパフローし、内部リセット信号が発生してプログラムを強制終了させてしまいます。この結果、内部リセットがかかったということはプログラムの暴走が起きたことを示し、その検出ができたことになります。

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

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

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



例 9.10 msのウォッチドッグ・タイマとして使用する（3.6 MHz動作時）。

プログラムをBTMの設定時間（9.10 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、9.10 msごとにBTMのビット3に“1”をセットする。）

モジュール1：

```

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

9.10 ms以内で  
処理完了

モジュール2：

```

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

9.10 ms以内で  
処理完了

### 5.3.6 その他の機能

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

#### 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のを防ぐため、読み出し命令は2回実行してください。そして、読み出した2つの内容を比較して、妥当な値であればあとの方を読み出し結果とし、まったく異なれば初めからやり直すようにしてください。

例1 . BTのカウント内容の読み出し。

```

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

```

2 . INT1割り込み (検出エッジ選択可能) へ入力されるパルスの周期を設定する (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9.10 ms以上 (3.6 MHz動作時) とする)。INT1割り込みルーチンに入るまえに、あらかじめINT1の検出エッジを指定してください。

INT1割り込みルーチン (MBE = 0)

```

LOOP : MOV  XA, BT    ; 1回目の読み出し
MOV    BC, XA        ; データを格納
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    PORT10.3     ; P103 = 1 ?
BR     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
DI     IE1
CLR1   IM1.0
CLR1   IRQ1
EI     IE1
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

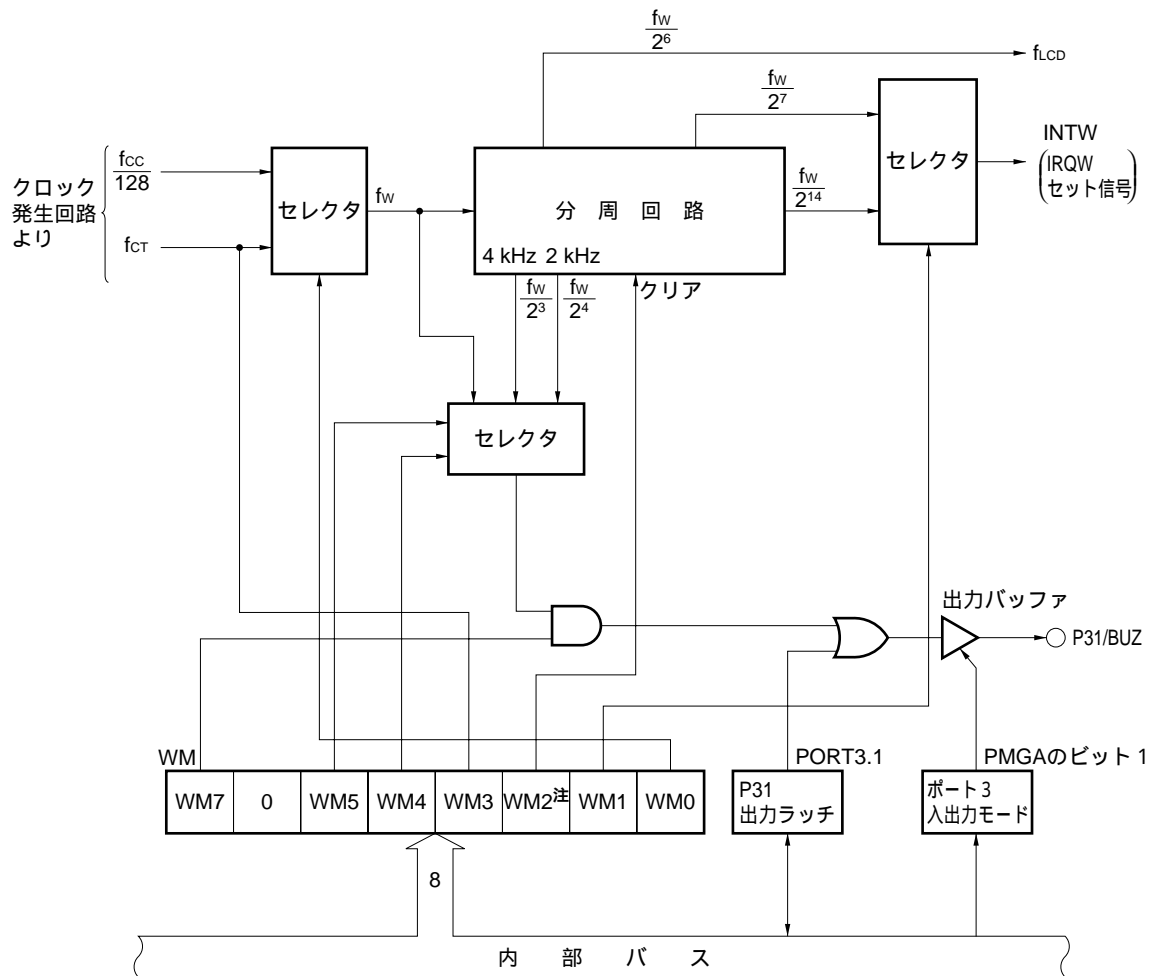
μPD753304は時計用タイマを1チャンネル内蔵しています。時計用タイマには、次のような機能があります。

- ( a )  $f_w/2^{14}$ 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 ) 早送りモードにより128倍 ( $f_w/2^7$ ) の時間間隔となり、プログラムのデバッグや検査に便利です。
- ( c ) 任意の周波数 ( $f_w/2^4$ ,  $f_w/2^3$ ,  $f_w$ ) をP31/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。
- ( d ) 分周回路のクリアをすることによって、時計をゼロ秒スタートできます。

### 5.4.1 時計用タイマの構成

時計用タイマは図5 - 25のように構成されています。

図5 - 25 時計用タイマのブロック図



注 LCDコントローラ/ドライバを用いるときは、1にセットしてください。

### 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 - 26にそのフォーマットを示します。

時計モード・レジスタは、8ビット操作命令により設定されます。

RESET信号発生により、全ビットは“0”にクリアされます。

図5-26 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 <sup>注</sup>	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	$\frac{f_w}{2^4}$
0	1	$\frac{f_w}{2^3}$
1	0	設定禁止
1	1	$f_w$

WM3	バイト・リード時は不定です。
-----	----------------

時計動作 許可/禁止ビット

WM2 <sup>注</sup>	0	時計動作停止（分周回路クリア，LCDクロック供給不可）
	1	時計動作可能（LCDクロック供給可能）

注 LCDコントローラ/ドライバを用いるときは1にセットしてください。

動作モード選択ビット

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

カウント・クロック（ $f_w$ ）の選択ビット

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

## 5.5 タイマ・カウンタ

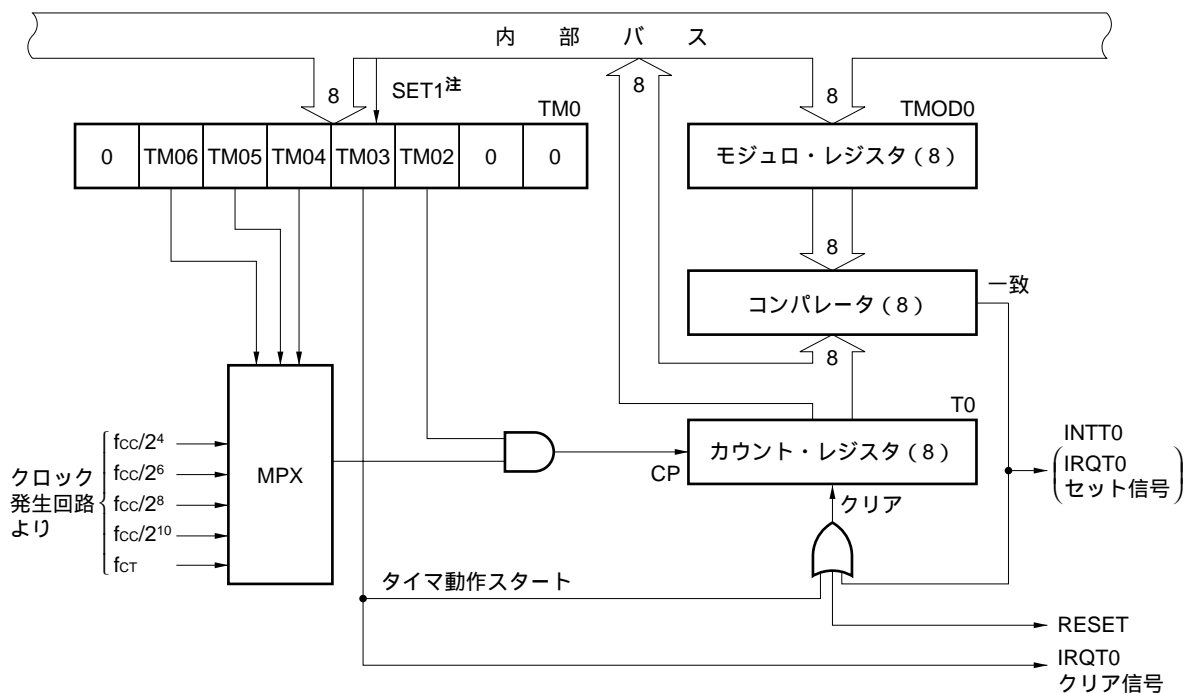
μPD753304は、タイマ・カウンタを1チャンネル内蔵しています。タイマ・カウンタには、次のような機能があります。

- ( a ) プログラマブル・インターバル・タイマ動作
- ( b ) カウント値の読み出し機能

### 5.5.1 タイマ・カウンタの構成

タイマ・カウンタは図5-27のように構成されています。

図5-27 タイマ・カウンタのブロック図



注 命令の実行

注意 TM0にデータを設定するときは、ビット0, 1, 7を必ず0に設定してください。

**タイマ・カウンタ・モード・レジスタ (TM0)**

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

図 5 - 28にそのフォーマットを示します。

タイマ・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・スタート指示ビットで、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・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 = 3.52 kHz (3.6 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    TM0, XA              ;TM0 4CH
```

**2** . タイマ・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   TM0.3               ;TM0.bit3 1
```



図5 - 28 タイマ・カウンタ・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
FA0H	0	TM06	TM05	TM04	TM03	TM02	0	0	TM0

カウント・パルス (CP) 選択ビット

★

f<sub>cc</sub> = 3.6 MHz動作時

TM06	TM05	TM04	カウント・パルス (CP)
0	1	0	f <sub>CT</sub> (47 kHz)
1	0	0	f <sub>cc</sub> /2 <sup>10</sup> (3.52 kHz)
1	0	1	f <sub>cc</sub> /2 <sup>8</sup> (14.1 kHz)
1	1	0	f <sub>cc</sub> /2 <sup>6</sup> (56.25 kHz)
1	1	1	f <sub>cc</sub> /2 <sup>4</sup> (225 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

注意 TM0にデータを設定するときは、ビット0, 1, 7を必ず0に設定してください。

### 5.5.2 8ビット・タイマ・カウンタ・モードの動作

8ビット・タイマ・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・タイマ動作を行います。

#### (1) レジスタの設定

8ビット・タイマ・カウンタ・モードは、次の3つのレジスタを使用します。

- ・タイマ・カウンタ・モード・レジスタ (TM0)
- ・タイマ・カウンタ・カウント・レジスタ (T0)
- ・タイマ・カウンタ・モジュロ・レジスタ (TMOD0)

#### (a) タイマ・カウンタ・モード・レジスタ (TM0)

8ビット・タイマ・カウンタ・モードを使用する場合はTM0を次のように設定します (TM0のフォーマットについては図5-28を参照してください)。

TM0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。

TM0は、内部リセット信号発生時に00Hにクリアされます。

#### (2) タイマ・カウンタの時間設定

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

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

T (sec) : タイマ設定時間 (秒)

f<sub>CP</sub> (Hz) : CP周波数 (Hz)

n : モジュロ・レジスタの内容 (n = 0)

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

表5-6にタイマ・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 (モジュロ・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

モード・レジスタ			f <sub>CC</sub> = 3.6 MHz動作時	
TM06	TM05	TM04	分解能	最長設定時間
0	1	0	21.3 μs <sup>注</sup>	5.45 ms <sup>注</sup>
1	0	0	284 μs	72.8 ms
1	0	1	71.1 μs	18.2 ms
1	1	0	17.8 μs	4.55 ms
1	1	1	4.4 μs	1.14 ms

注 f<sub>CT</sub> = 47 kHz動作時

(3) 8ビット・タイマ・カウンタ・モードの応用

50 ms間隔で割り込みを発生するインターバル・タイマとして応用します (f<sub>CC</sub> = 3.6 MHz動作時)。

- ・モード・レジスタ (TM0) の上位4ビットは0100Bとし、最長設定時間72.8 msを選択します。
- ・TM0の下位4ビットは1100Bとします。
- ・モジュロ・レジスタ (TMOD0) の設定値は次のとおりです。

$$\frac{50 \text{ ms}}{284 \mu\text{s}} = 176, 176 - 1 = \text{AFH}$$

プログラム例

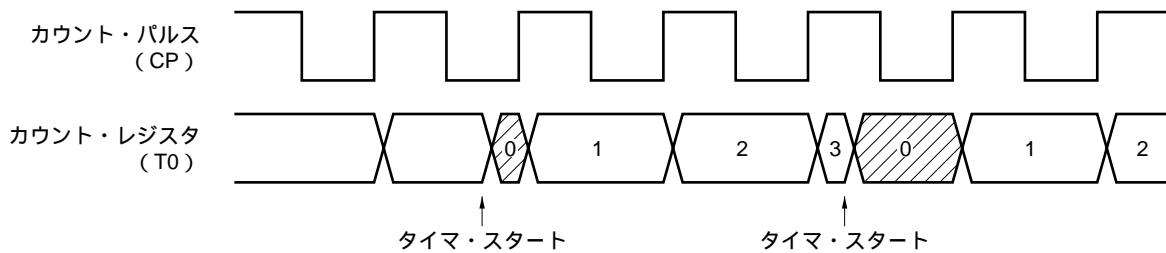
```

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

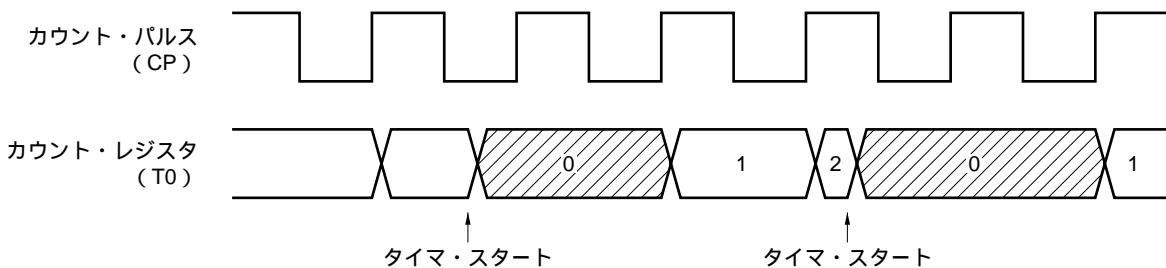
### 5.5.3 タイマ・カウンタ応用上の注意

#### (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を“1”にセット) 後、一致信号が発生するまでの時間は (モジュール・レジスタの内容 + 1) × 分解能の式で求めた値に対し、最大でカウント・パルス (CP) 1 クロック分の誤差が生じます。これは次の図に示すように、CPに対してカウント・レジスタT0のクリアが非同期で行われるためです。



またCPの周波数が1マシン・サイクル以上の場合、タイマ・スタート (TM0のビット3を“1”にセット) 後、一致信号が発生するまでの時間は、(モジュール・レジスタの内容 + 1) × 分解能の式で求めた値に対し、最大でCP2クロック分の誤差が生じます。これは、次の図に示すようにCPに対してT0のクリアがCPUクロックを基に非同期で行われるためです。



## (2) タイマ・スタート時の注意

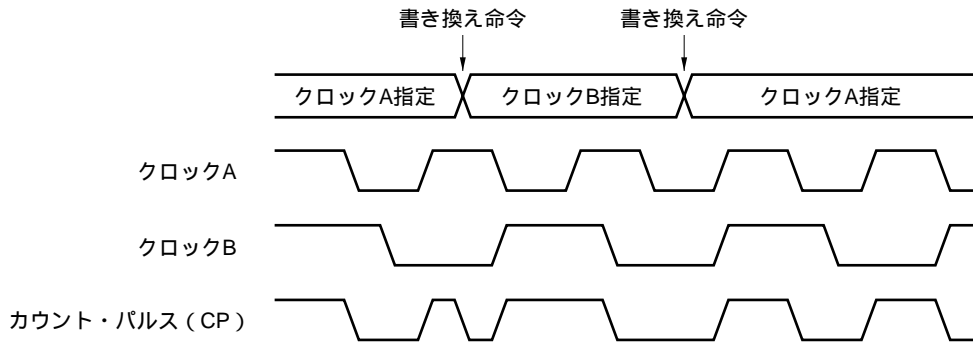
通常タイマ・スタート(TM0のビット3を“1”にセット)によってカウント・レジスタT0と割り込み要求フラグIRQT0はクリアされますが、タイマが動作モードにあり、IRQT0のセットと、タイマ・スタートが同一タイミングで発生した場合には、IRQT0をクリアできない場合が生じます。これは、IRQT0をベクタ割り込みとして使用しているときは問題ありませんが、IRQT0をテストする応用の場合には、タイマ・スタートを行ったにもかかわらず、IRQT0がセットされていることになり不都合を生じます。したがって、IRQT0が立つ可能性のあるタイミングでタイマ・スタートする場合にはいったん、タイマを停止(TM0のビット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

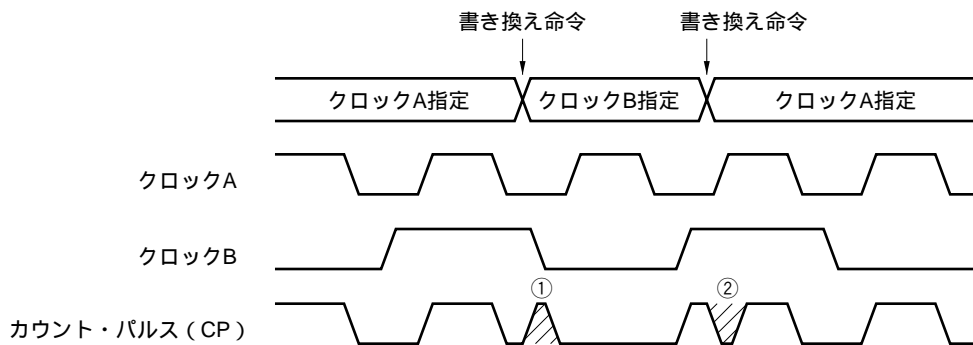
```
SEL    MB15
MOV    XA, #0
MOV    TM0, XA    ; タイマを停止
MOV    XA, #4CH
MOV    TM0, XA    ; 再スタート
または
SEL    MB15
SET1   TM0.3
SET1   TM0.3    ; 再スタート
```

(3) カウント・パルス変更時の注意

タイマ・カウンタ・モード・レジスタ (TM0) を書き換えてカウント・パルス (CP) を変更した場合、その指定は命令実行直後から有効になります。

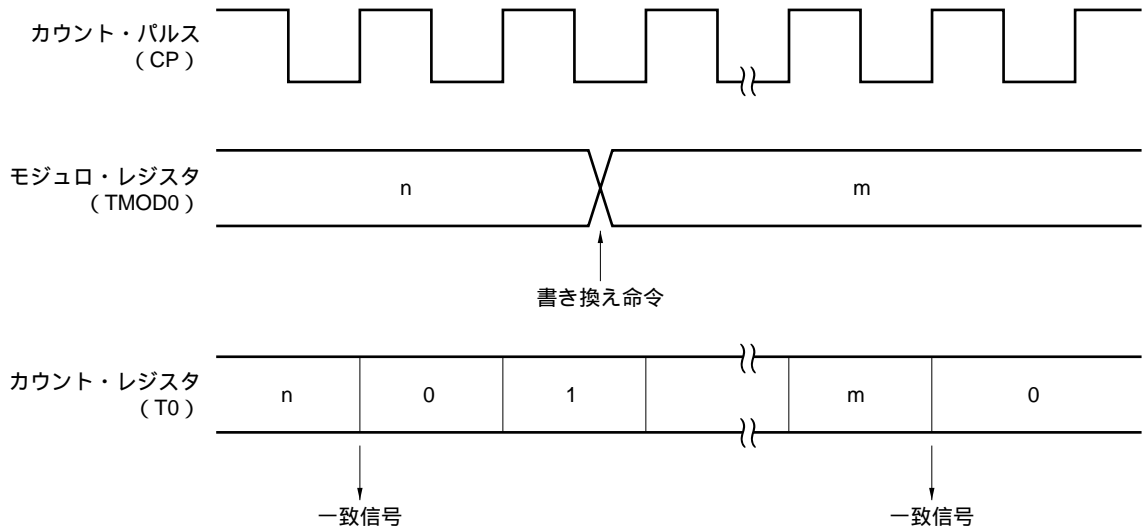


CP変更時のクロックの組み合わせにより、次図のようにヒゲ状 (または ) のCPが発生することもあります。この場合、ミス・カウントあるいはカウンタ・レジスタ (T0) の内容が破壊される可能性がありますので、CPを変更する場合は必ず、TM0のビット3を“1”にして、タイマの再スタートも同時に行ってください。

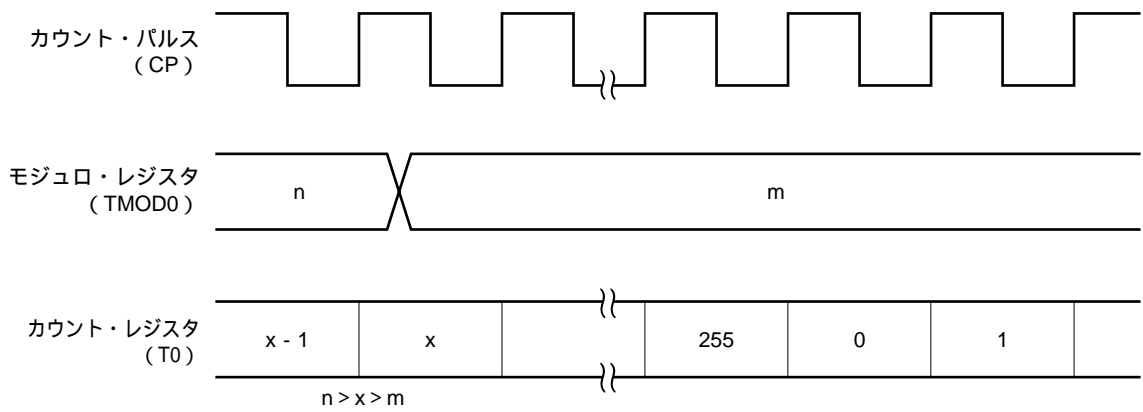


(4) モジュロ・レジスタ変更後の動作

モジュロ・レジスタ (TMOD0) の変更は、8ビット・データ・メモリ操作命令を実行した時点で行われます。



TMOD0の変更後の値が、カウント・レジスタ (T0) の値よりも小さい場合には、T0はカウントを継続しオーバフローして0から再カウントします。したがって、TMOD0の変更後の値 ( $m$ ) が変更前の値 ( $n$ ) より小さい場合には、TMOD0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 LCDコントローラ/ドライバ

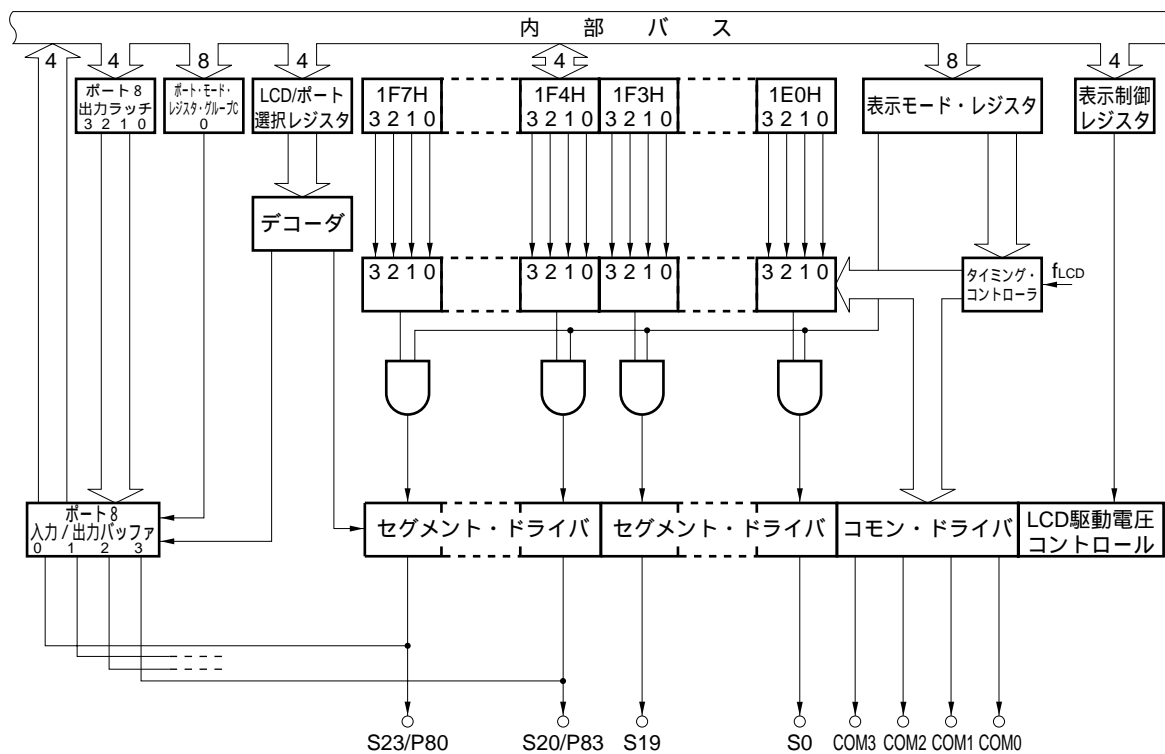
### 5.6.1 LCDコントローラ/ドライバの構成

μPD753304は、表示データ・メモリのデータに従い、セグメント信号とコモン信号を発生する表示コントローラと、LCDパネルを直接駆動可能なセグメント・ドライバ、コモン・ドライバを内蔵しています。

LCDコントローラ/ドライバの構成は図5-29のようになっています。

**注意** LCDコントローラ/ドライバを用いるときは、時計モード・レジスタWMMのビット2を“1”にセットしてください。

図5-29 LCDコントローラ/ドライバのブロック図





### 5.6.2 LCDコントローラ/ドライバの機能

μPD753304に内蔵されているLCDコントローラ/ドライバの機能は次のようなものです。

- ( a ) DMA動作により自動的に表示データ・メモリを読み出してセグメント信号とコモン信号を発生。
- ( b ) 5種類の表示モードが選択可能。
  - スタティック
  - 1/2デューティ ( 2 時分割 ) , 1/2バイアス
  - 1/3デューティ ( 3 時分割 ) , 1/2バイアス
  - 1/3デューティ ( 3 時分割 ) , 1/3バイアス
  - 1/4デューティ ( 4 時分割 ) , 1/3バイアス
- ( c ) 各表示モードにおいて、4種類のフレーム周波数が選択可能。
- ( d ) セグメント信号出力は最大24本 ( S0-S23 ) , コモン出力は4本 ( COM0-COM3 ) 。
- ( e ) セグメント信号出力 ( S20-S23 ) は、入出力ポート ( PORT8 ) に切り替え可能。
- ( f ) LCD駆動用電源供給用の分割抵抗を内蔵。
  - ・各種バイアス法, LCD駆動電圧に対応可能。
  - ・表示オフ時には、分割抵抗へ流れる電流をカット。
- ( g ) サブシステム・クロックによる動作も可能。
- ★ ( h ) LCD表示モードを選択可能 ( マスク・オプション ) 。

各表示モードにおける表示可能な最大画素数を表5 - 7 に示します。

表5 - 7 最大表示画素

バイアス法	時分割	使用COM信号	最大画素
-	スタティック	COM0 ( COM1, 2, 3 )	24 (セグメント24×コモン1) 注1
1/2	2	COM0, 1	48 (セグメント24×コモン2) 注2
	3	COM0, 1, 2	72 (セグメント24×コモン3) 注3
1/3	3		
	4	COM0, 1, 2, 3	96 (セグメント24×コモン4) 注4

注1 .  $\frac{5}{8}$  . 形のLCDパネルで8セグメント信号/桁のもの3桁。

2 .  $\frac{5}{8}$  . 形のLCDパネルで4セグメント信号/桁のもの6桁。

3 .  $\frac{5}{8}$  . 形のLCDパネルで3セグメント信号/桁のもの8桁。

4 .  $\frac{5}{8}$  . 形のLCDパネルで2セグメント信号/桁のもの12桁。

### 5.6.3 表示モード・レジスタ (LCDM)

表示モード・レジスタ (LCDM) は、表示モード、LCDクロック、フレーム周波数、および表示出力のオン、オフ制御を指定する8ビット・レジスタです。

LCDMは、8ビット・メモリ操作命令により設定します。またビット3 (LCDM3) のみビット操作命令によりセット、クリア可能です。

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

図5 - 30 表示モード・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
F8CH	0	0	LCDM5	LCDM4	LCDM3	LCDM2	LCDM1	LCDM0	LCDM

★ LCDクロックの選択

LCDM5	LCDM4	LCDCL <sup>注</sup>
0	0	$f_w/2^9$ ( 92 Hz )
0	1	$f_w/2^8$ ( 184 Hz )
1	0	$f_w/2^7$ ( 367 Hz )
1	1	$f_w/2^6$ ( 734 Hz )

注 LCDCLは時計用タイマが動作しているときのみ供給されます。LCDコントローラ/ドライバを用いるときは時計モード・レジスタWMのビット2を“1”にセットしてください。

表示モードの選択

LCDM3	LCDM2	LCDM1	LCDM0	時分割数	バイアス法
0	x	x	x	表示オフ <sup>注</sup>	
1	0	0	0	4	1/3
1	0	0	1	3	1/3
1	0	1	0	2	1/2
1	0	1	1	3	1/2
1	1	0	0	スタティック	
上記以外				設定禁止	

注 全セグメント信号は非選択レベル。

★ フレーム周波数 (Hz)

LCDCL	$f_w/2^9$ ( 92 Hz )	$f_w/2^8$ ( 184 Hz )	$f_w/2^7$ ( 367 Hz )	$f_w/2^6$ ( 734 Hz )
表示デューティ				
スタティック	92	184	367	734
1/2	46	92	184	367
1/3	31	61	122	245
1/4	23	46	92	184

$f_w = 47$  kHzの場合

$f_w$  : 時計用タイマへの入力クロック (  $f_{cc}/128$ または $f_{ct}$  )

### 5.6.4 表示制御レジスタ (LCDC)

表示制御レジスタは、次に示すようなLCD駆動の制御を行います。

- ・ コモンおよびセグメント出力の許可 / 禁止。
- ・ LCD駆動用電源供給用の分割抵抗へ流れる電流のカット。

LCDCは4ビット・メモリ操作命令により設定します。

RESET信号発生により表示制御レジスタはすべて“0”にクリアされます。

図5-31 表示制御レジスタのフォーマット

アドレス	3	2	1	0	略号
F8EH	0	0	0	LCDC0	LCDC

LCDC0とLCDCM3による表示出力の状態

LCDC0	0	1	
LCDCM3	×	0	1
COM0-COM3	“L”を出力(表示オフ)	表示モードに対応したコモン信号を出力	表示モードに対応したコモン信号を出力
S0-S19	“L”を出力(表示オフ)	表示モードに対応したセグメント信号を出力 (非選択レベル出力, 表示オフ)	表示モードに対応したセグメント信号を出力 (表示オン)
S20-S23 のセグメント指定端子			
S20-S23 のポート指定端子	入出力ポート (入力ポートか出力ポートかは, ポート・モード・レジスタ・グループC (PMGC) の指定によります)		
分割抵抗への電源供給	オフ	オン	オン

### 5.6.5 LCD/ポート選択レジスタ (LPS)

LCD/ポート選択レジスタ (LPS) は、セグメント信号出力 (S20-S23) を入出力ポートに切り替えるレジスタです。

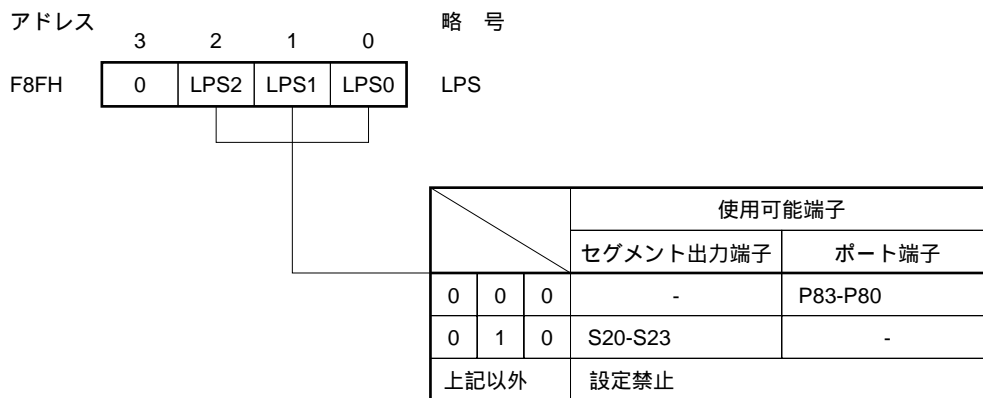
セグメント信号出力のうち、S20-S23はPORT8と兼用になっており、4本単位で切り替えられます。

また、LPSの値を“0000”にセットすることにより、S20-S23は入出力ポート (PORT8) に切り替えることができます。

LPSは4ビット・メモリ操作命令により設定します。

RESET信号発生によりLPSはすべて“0”にクリアされます。

図5 - 32 LCD/ポート選択レジスタのフォーマット



**注意 1** . リセット時、LPSはすべて“0”にクリアされるため、LCDを使用することはできません。したがって、LCDを使用する場合には、必ずLPSの値を0010にセットしてください。

**2** . LPSのビット3には必ず“0”をセットしてください。

**3** . LPSによりセグメント信号出力に指定したポートは、ポート・モード・レジスタ・グループCにより入力モードを指定しても、フローティングにはなりません。

セグメント信号出力に指定したポートに対して入力命令を実行した場合は、出力ラッチの内容が入力されます。

### 5.6.6 表示データ・メモリ

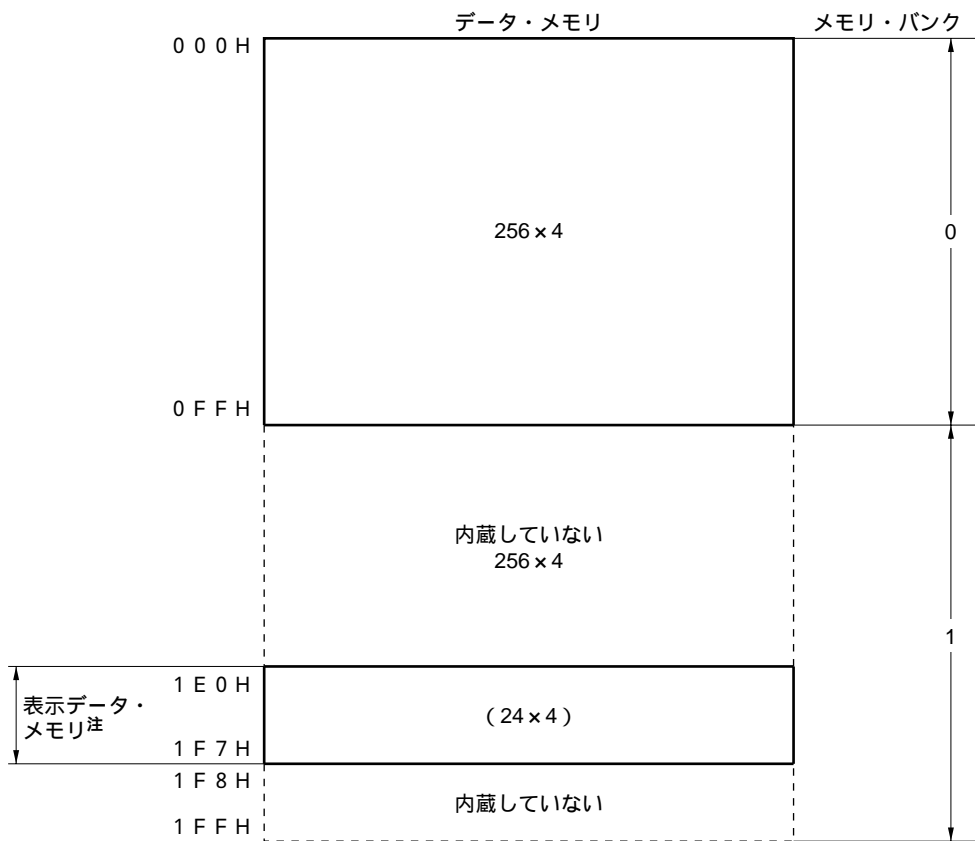
表示データ・メモリは、1E0H-1F7Hにマッピングされ、書き込みのみ可能です。

表示データ・メモリは、CPUの動作とは無関係にLCDコントローラ/ドライバがDMA動作によって読み出すメモリ・エリアです。LCDコントローラは表示データ・メモリのデータに従ってセグメント信号を制御します。

LCD表示やポートとして使用しない領域を通常のデータ・メモリとして使用することはできません。

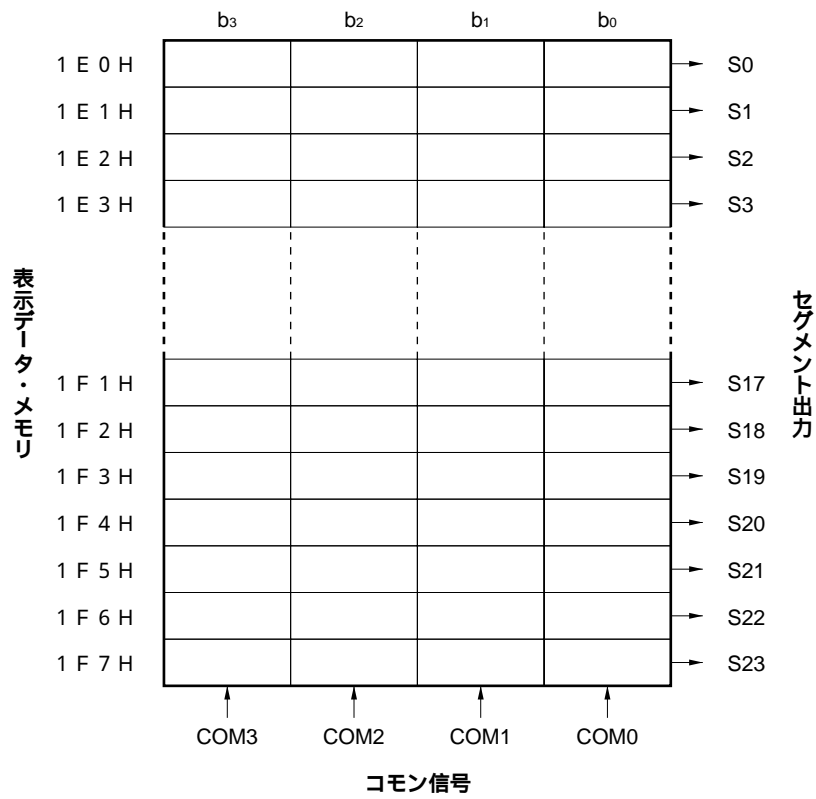
- ★ 表示データ・メモリは4ビット単位で操作します。1ビット単位、8ビット単位の操作はできません。表示データ・メモリの各ビットとセグメント出力の対応を図5-34に示します。

図5-33 データ・メモリ・マップ



注 書き込みのみ可能です。

図5 - 34 表示データ・メモリとコモン・セグメント対応



### 5.6.7 コモン信号とセグメント信号














LCDパネルの各画素は、それに対応するコモン信号とセグメント信号の電位差が一定電圧（LCD駆動電圧  $V_{LCD}$ ）以上になると点灯します。 $V_{LCD}$ 以下の電位差または電位差が0Vになると消灯します。

LCDパネルは、コモン信号とセグメント信号にDC電圧が加えられると劣化するため、AC電圧によって駆動されます。

#### (1) コモン信号

コモン信号，設定される時分割数に応じて表5-8に示す順で選択タイミングとなり，それらを一周期として繰り返し動作を行います。スタティック・モードの場合はCOM0-COM3は同一信号が出力されます。なお，2時分割の場合のCOM2，COM3端子および3時分割の場合のCOM3端子はオープンにして使用してください。

表5-8 COM信号

COM信号 時分割数	COM0	COM1	COM2	COM3
スタティック				
2			オープン	オープン
3				オープン
4				

#### (2) セグメント信号

セグメント信号は，データ・メモリの表示データ・メモリ（1E0H-1F7H）の24ロケーションに対応して24本あり，各ロケーションのビット0はCOM0，ビット1はCOM1，ビット2はCOM2，ビット3はCOM3の選択タイミングに同期して自動的に読み出され，各ビットの内容が1ならば選択電圧に変換され，0ならば非選択電圧に変換されてセグメント端子（S0-S23）より出力されます。

以上のことから表示データ・メモリには使用するLCDパネルの前面電極（セグメント信号に対応）と背面電極（コモン信号に対応）がどのような組み合わせで表示パターンを形成するかを確認のうえ，表示したいパターンに1対1に対応するビット・データを書き込むようにします。

なお，スタティック方式の場合の表示データ・メモリのビット1/2/3，2時分割方式の場合のビット2/3，3時分割方式の場合のビット3はアクセスされませんが，表示以外の目的に使用することはできません。

(3) コモン信号とセグメント信号の出力波形

コモン信号とセグメント信号には表5-9～5-11に示すようなレベルの電圧が出力されます。両者がともを選択電圧となったときのみ +V<sub>LCD</sub> / -V<sub>LCD</sub> の点灯電圧となり、それ以外の組み合わせでは消灯電圧となります。

表5-9 LCD駆動電圧(スタティック)

セグメント信号Sn		選 択	非 選 択
		コモン信号COM0	V <sub>Lc0</sub> /V <sub>ss</sub>
V <sub>ss</sub> /V <sub>Lc0</sub>		+ V <sub>LCD</sub> / - V <sub>LCD</sub>	0 V/0 V

表5-10 LCD駆動電圧(1/2バイアス法)

セグメント信号Sn		選 択	非 選 択
		コモン信号COMm	V <sub>Lc0</sub> /V <sub>ss</sub>
選 択	V <sub>ss</sub> /V <sub>Lc0</sub>	+ V <sub>LCD</sub> / - V <sub>LCD</sub>	0 V/0 V
非選択	V <sub>Lc1</sub> = V <sub>Lc2</sub>	+ $\frac{1}{2}$ V <sub>LCD</sub> / - $\frac{1}{2}$ V <sub>LCD</sub>	- $\frac{1}{2}$ V <sub>LCD</sub> / + $\frac{1}{2}$ V <sub>LCD</sub>

表5-11 LCD駆動電圧(1/3バイアス法)

セグメント信号Sn		選 択	非 選 択
		コモン信号COMm	V <sub>Lc0</sub> /V <sub>ss</sub>
選 択	V <sub>ss</sub> /V <sub>Lc0</sub>	+ V <sub>LCD</sub> / - V <sub>LCD</sub>	+ $\frac{1}{3}$ V <sub>LCD</sub> / - $\frac{1}{3}$ V <sub>LCD</sub>
非選択	V <sub>Lc1</sub> /V <sub>Lc2</sub>	+ $\frac{1}{3}$ V <sub>LCD</sub> / - $\frac{1}{3}$ V <sub>LCD</sub>	+ $\frac{1}{3}$ V <sub>LCD</sub> / - $\frac{1}{3}$ V <sub>LCD</sub>



図5 - 35 ~ 5 - 37にコモン信号波形を，図5 - 38にコモン信号とセグメント信号の電位と位相を示します。

図5 - 35 コモン信号波形 (スタティック)

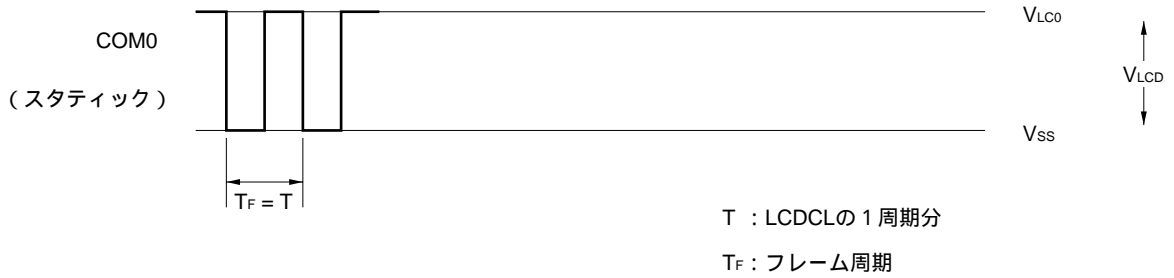


図5 - 36 コモン信号波形 (1/2バイアス法)

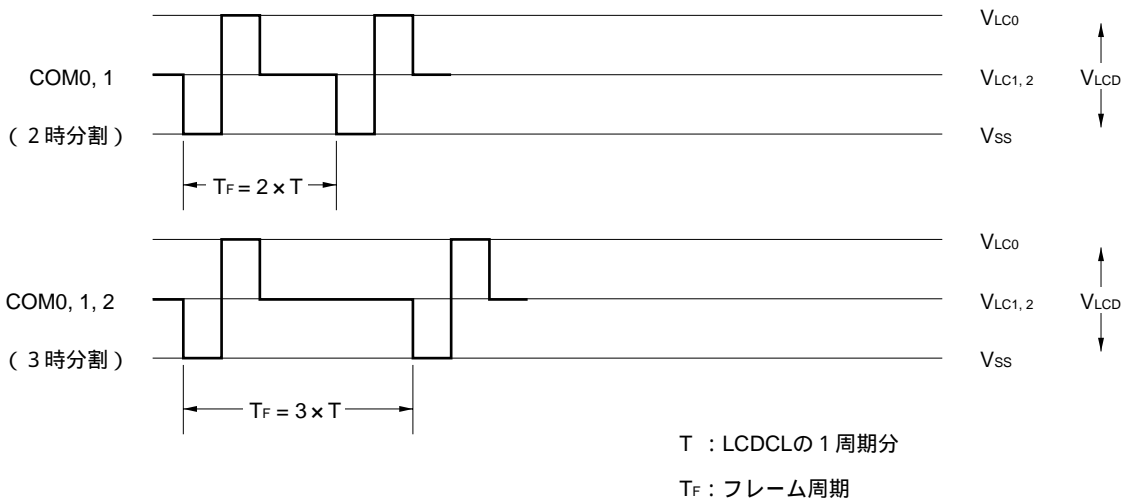


図5 - 37 コモン信号波形 (1/3バイアス法)

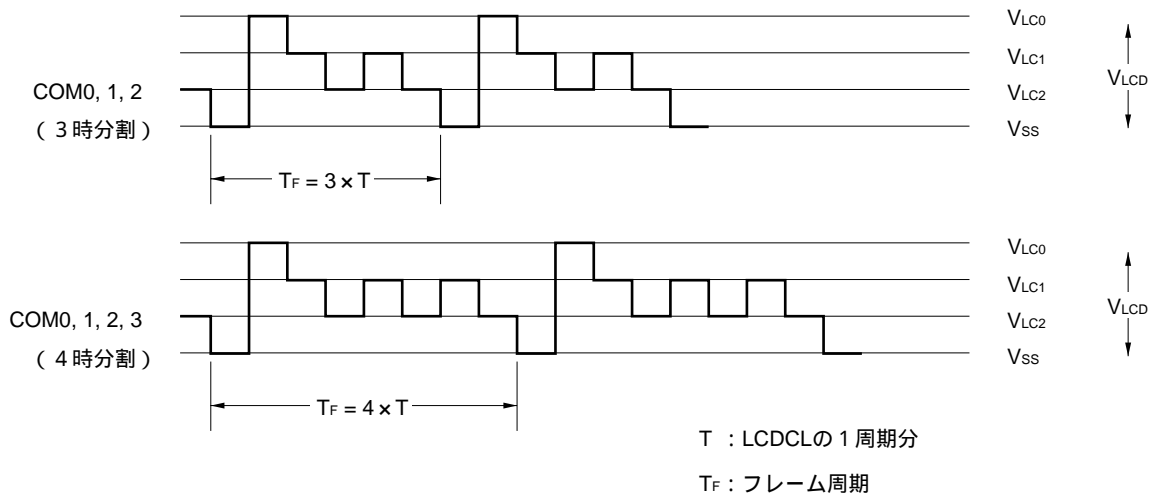
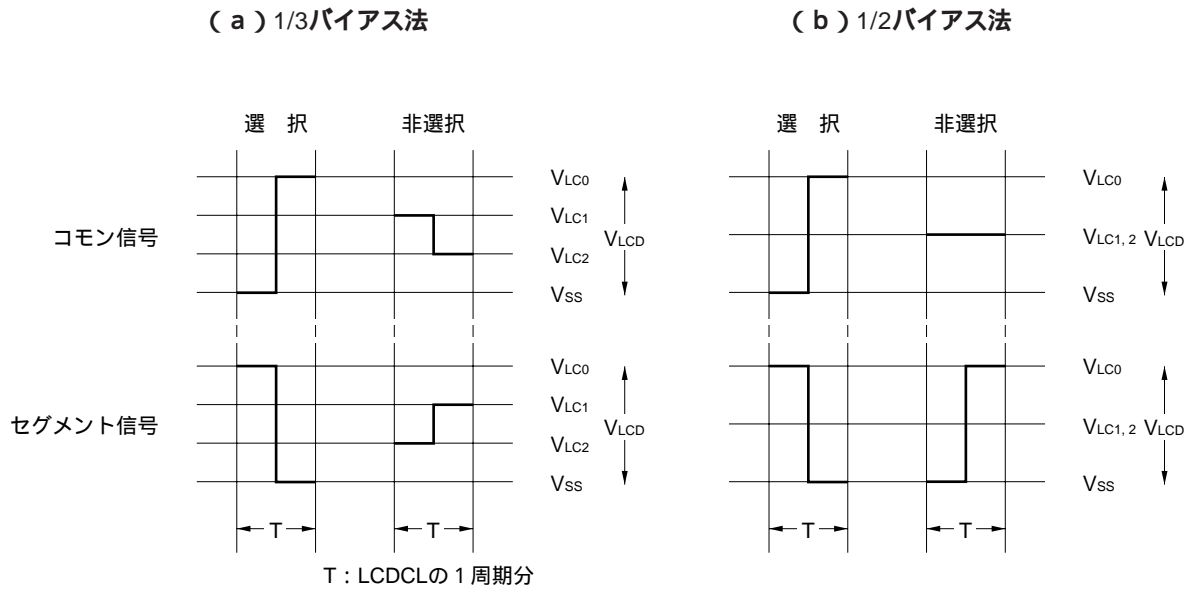
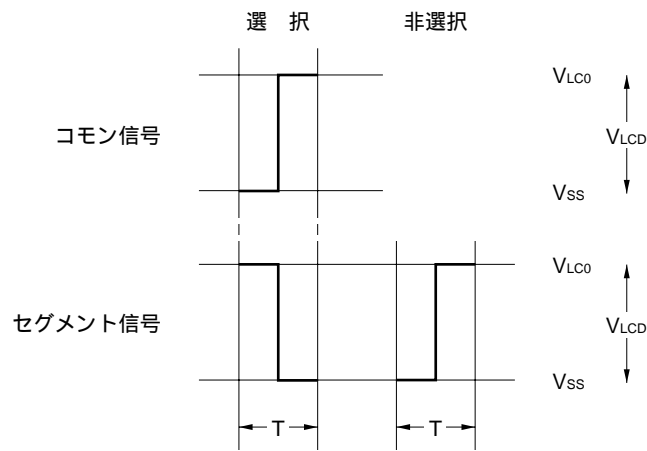


図5 - 38 コモン信号とセグメント信号の電位と位相



(c) スタティック表示モード



### 5.6.8 LCD駆動用電源 $V_{LC0}$ , $V_{LC1}$ , $V_{LC2}$ の供給

$\mu$ PD753304はLCD駆動用電源供給用に分割抵抗が内蔵されていますが、外部端子として $V_{LC0}$ - $V_{LC2}$ 端子（LCD駆動用電源）とBIAS端子（外付け分割抵抗カット用端子）を持っていません。そこで各表示モードに対応するために、次の3つのマスク・オプションから選択することができます。

- ★ スタティック・モード（BIAS- $V_{LC0}$ 間ショート,  $V_{LC0}$ - $V_{LC1}$ 間オープン）
- 1/2 バイアス・モード（BIAS- $V_{LC0}$ 間ショート,  $V_{LC1}$ - $V_{LC2}$ 間ショート）
- 1/3 バイアス・モード（BIAS- $V_{LC0}$ 間ショート）

スタティック, 1/2 および 1/3 バイアス法に基づくLCD駆動用電源として、次の表に示す値が供給されます。

表5 - 12 LCD駆動用電源の供給値

バイアス法 LCD 駆動用電源	バイアスなし ( スタティック ・モード )	1/2	1/3
$V_{LC0}$	$V_{LCD} (= V_{DD})$	$V_{LCD}$	$V_{LCD}$
$V_{LC1}$	0 V	$1/2 V_{LCD}$	$2/3 V_{LCD}$
$V_{LC2}$			$1/3 V_{LCD}$
$V_{SS}$		0 V	0 V

表5 - 12に従ったLCD駆動用電源の供給例を図5 - 39 ( a ) - ( c ) に示します。

また、表示制御レジスタのビット0 (LCDC0) を“0”にクリアすることにより分割抵抗を流れる電流をカットすることができます。

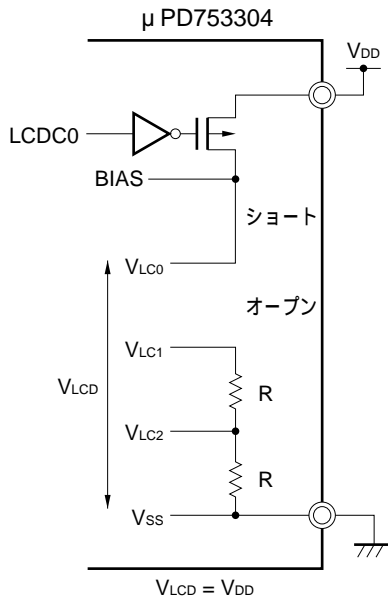
このLCD電源のオン, オフ制御は、時計用タイマがメイン・システム・クロックによって動作しているときSTOP命令によってLCDクロック（システム・クロック系を選択している場合）が停止し、LCDにDC電圧が加わってしまうのを防ぐためにも有効です。すなわち、STOP命令を実行する直前に表示制御レジスタのビット0 (LCDC0) を“0”にクリアし、LCD駆動用電源のすべてを同電位 $V_{SS}$ にすることによって、LCDクロックが停止してもLCDの電極間には電位差が生じないようにすることができます。時計用タイマがサブシステム・クロックによって動作しているときにはLCD表示は継続可能です。

★

図5 - 39 LCD駆動用電源の接続例

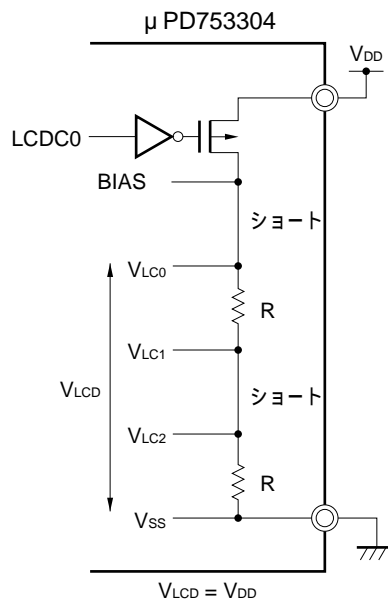
( a ) スタティック表示モード

(  $V_{DD} = 5V$ ,  $V_{LCD} = 5V$  の例 )



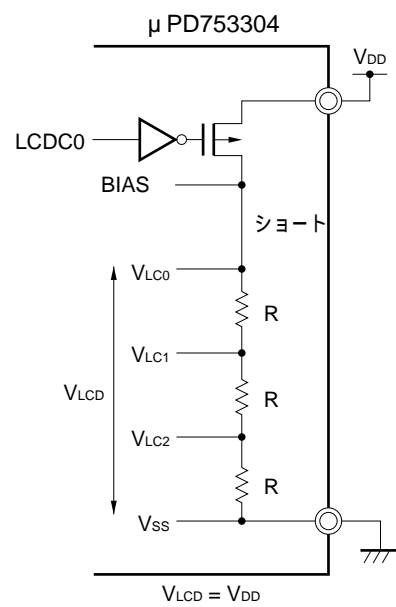
( b ) 1/2バイアス法

(  $V_{DD} = 5V$ ,  $V_{LCD} = 5V$  の例 )



( c ) 1/3バイアス法

(  $V_{DD} = 5V$ ,  $V_{LCD} = 5V$  の例 )



### 5.6.9 表示モード

#### (1) スタティック表示例

図5 - 41は、図5 - 40の表示パターンを持つスタティック方式の3桁LCDパネルとμPD753304のセグメント信号(S0-S23)およびコモン信号(COM0)との接続を示します。表示例は1 2.3で、表示データ・メモリ(1E0H-1F7H番地)の内容はこれに対応しています。

ここでは2桁目の2.(点)を例にとって説明します。図5 - 40の表示パターンに従って、COM0のコモン信号のタイミングで表5 - 13に示すような選択、非選択電圧をS8-S15端子に出力する必要があります。

表5 - 13 S16-S23端子の選択、非選択電圧(スタティック表示)

セグメント コモン	S8	S9	S10	S11	S12	S13	S14	S15
COM0	選	非	選	選	非	選	選	選

これにより、S8-S15に対応する表示データ・メモリ(1E8H-1EFH番地)のビット0には、10110111を用意すればよいことが分かります。

S11, S12とCOM0とのLCD駆動波形を図5 - 42に示します。COM0との選択タイミングでS11が選択電圧になるときに、LCD点灯レベルである $+V_{LCD}/-V_{LCD}$ の交流矩形波が発生されることが分かります。

COM1, 2, 3にはCOM0と同じ波形が出力されますので、COM0, 1, 2, 3を接続してドライブ能力を上げることができます。

図5 - 40 スタティック方式LCDの表示パターンと電極結線

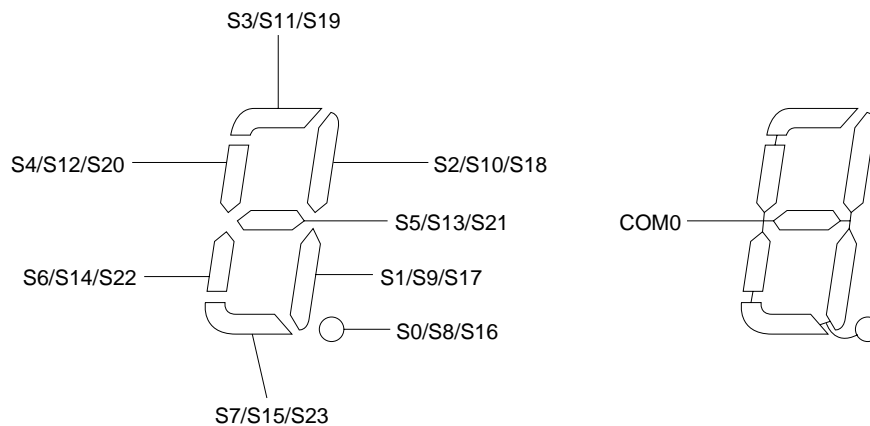


図 5 - 41 スタティックLCDパネルの結線例

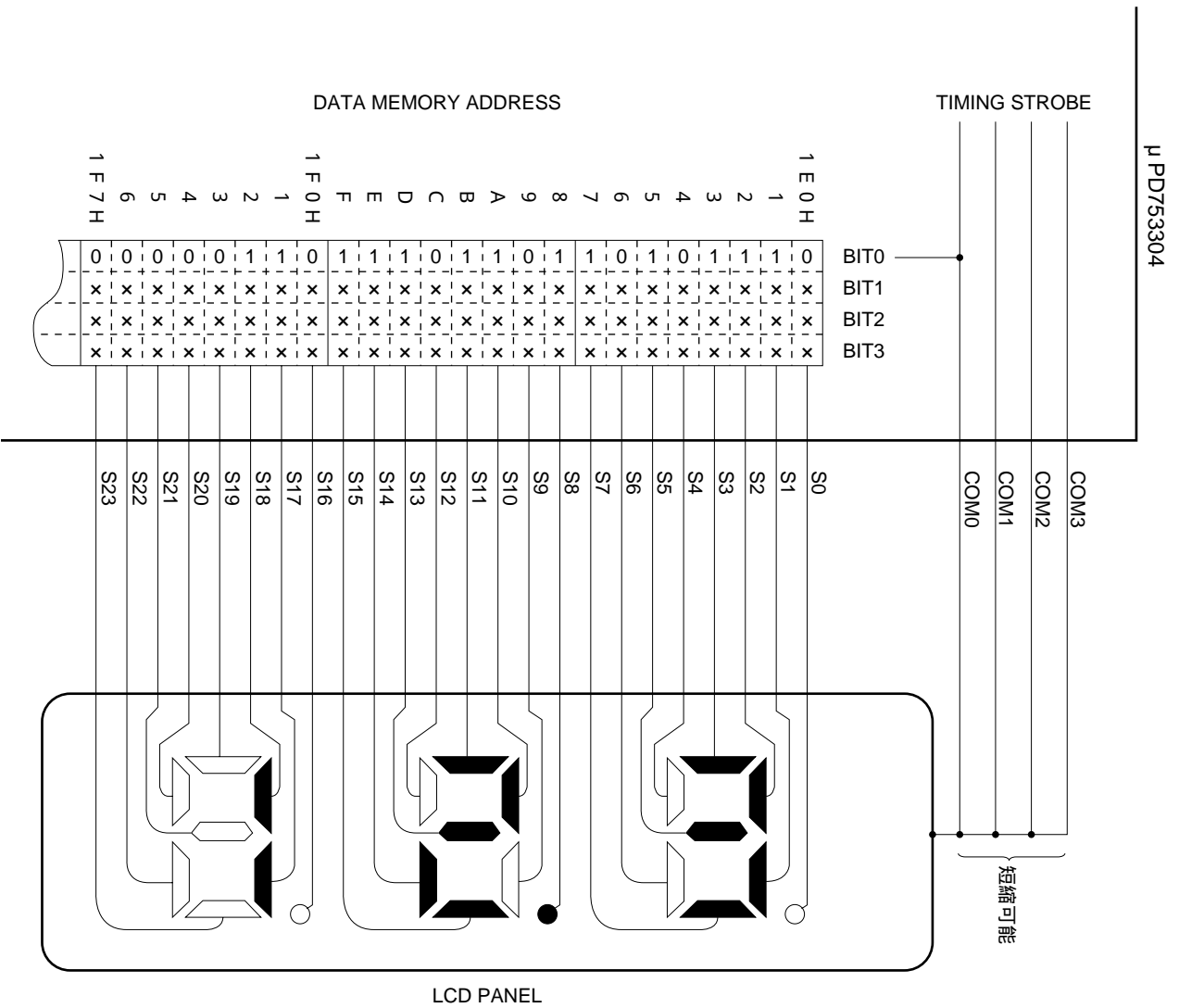
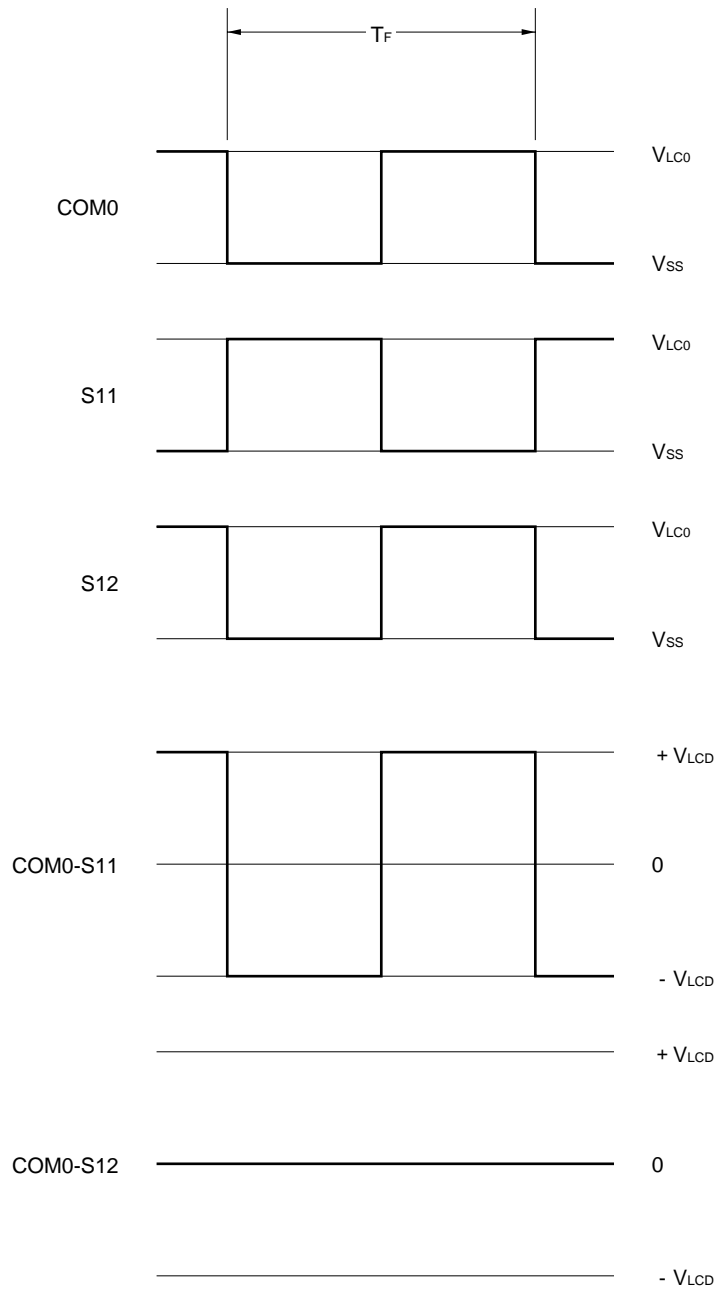


図5 - 42 スタティックLCD駆動波形例



(2) 2時分割表示例

図5 - 44は、図5 - 43の表示パターンを持つ2時分割方式の6桁LCDパネルとμPD753304のセグメント信号(S0-S23)およびコモン信号(COM0, COM1)との接続を示します。表示例は1234.56で、表示データ・メモリ(1E0H-1F7H番地)の内容はそれらに対応しています。

ここでは4桁目の3.(3)を例にとって説明します。図5 - 43の表示パターンに従って、COM0, 1の各コモン信号のタイミングで表5 - 14に示すような選択、非選択電圧をS12-S15端子に出力する必要があります。

表5 - 14 S12-S15端子の選択、非選択電圧(2時分割表示例)

セグメント コモン	S12	S13	S14	S15
COM0	選	選	非	非
COM1	非	選	選	選

これにより、たとえばS15に対応する表示データ・メモリ(1EFH番地)には、 $\times \times 10$ を用意すればよいことが分かります。

S15と各コモン信号間のLCD駆動波形例を図5 - 45に示します。COM1の選択タイミングでS15が選択電圧のときに、LCD点灯レベルである $+V_{LCD}/-V_{LCD}$ の交流矩形波が発生されることが分かります。

図5 - 43 2時分割方式LCDの表示パターンと電極結線

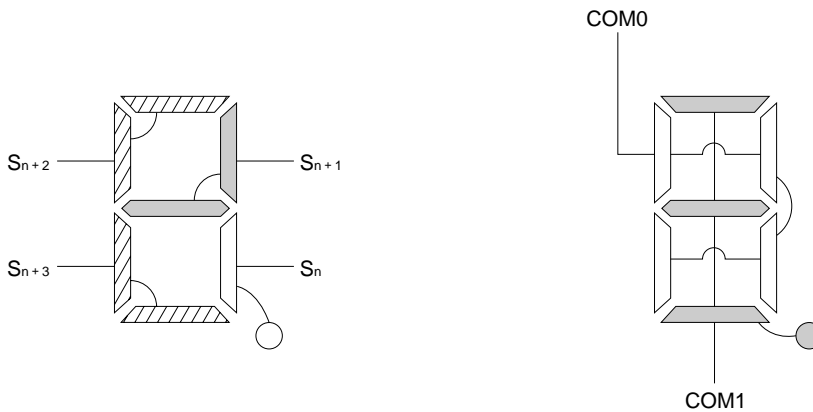
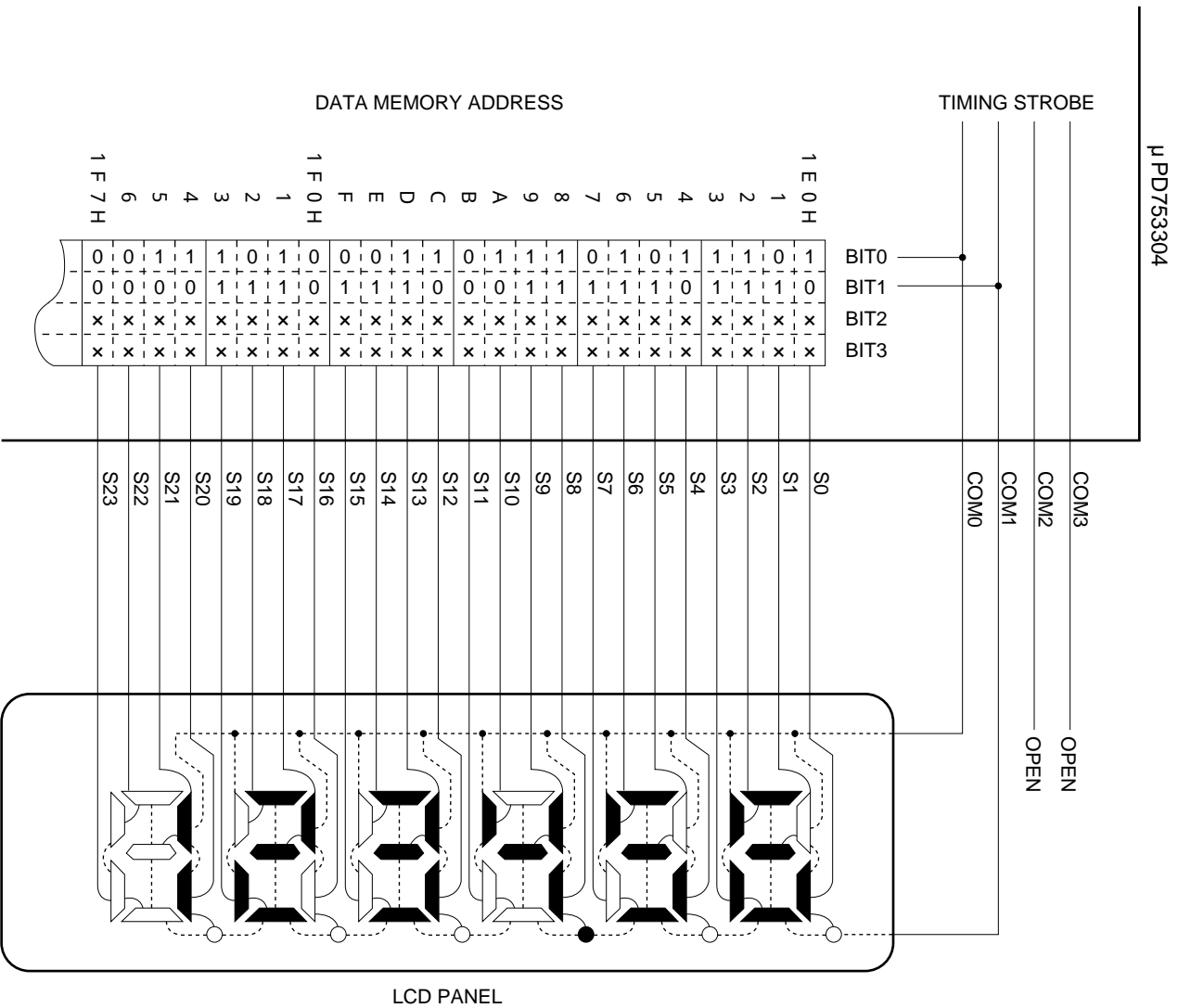


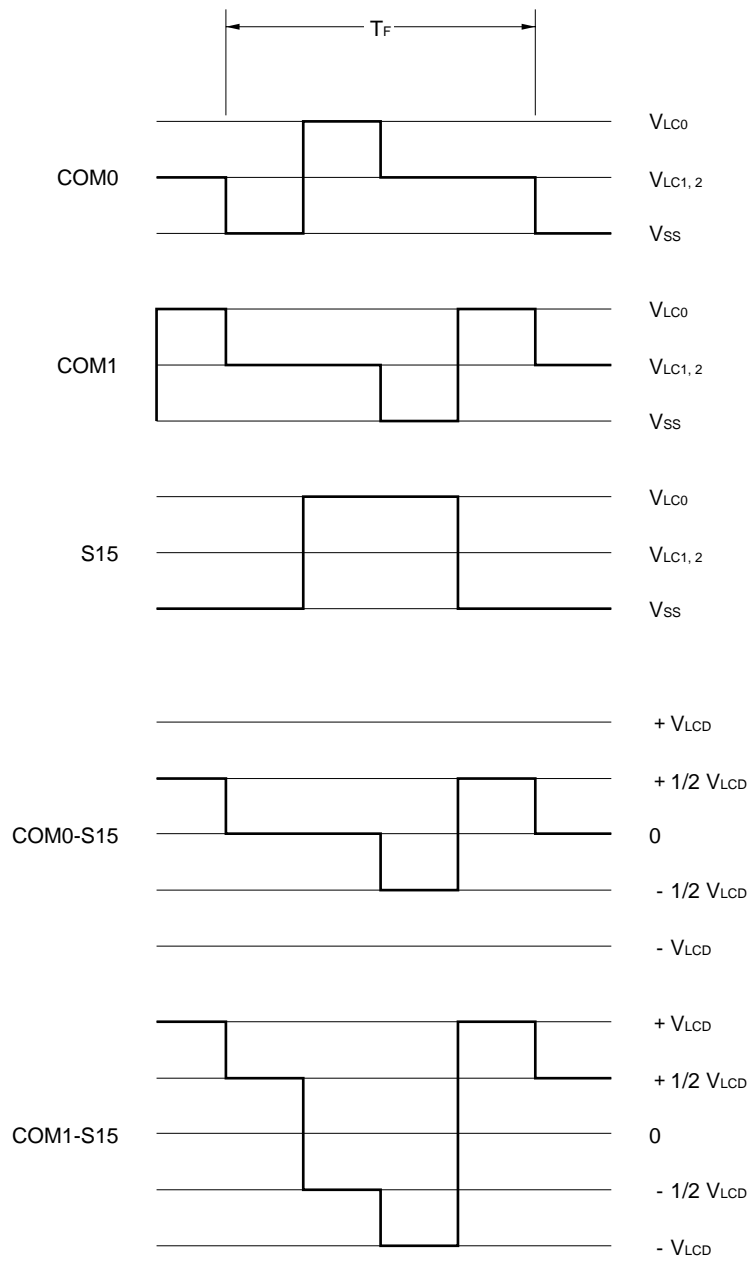


図 5 - 44 2 時分割 LCD パネルの結線例



x : 2 時分割表示のため、常に任意データをストア可能です。

図5 - 45 2時分割LCD駆動波形例(1/2バイアス法)



(3) 3時分割表示

図5 - 47は、図5 - 46の表示パターンを持つ3時分割方式の8桁LCDパネルとμPD753304のセグメント信号(S0-S23)およびコモン信号(COM0-COM2)との接続を示します。表示例は123456.78で、表示データ・メモリ(1E0H-1F7H番地)の内容はこれに対応しています。

ここでは3桁目の6.( $\bar{6}$ .)を例にとって説明します。図5 - 46の表示パターンに従って、COM0-COM2の各コモン信号のタイミングで表5 - 15に示すような選択、非選択電圧をS6-S8端子に出力する必要があります。

表5 - 15 S6-S8端子の選択、非選択電圧(3時分割表示例)

セグメント コモン	S6	S7	S8
COM0	非	選	選
COM1	選	選	選
COM2	選	選	-

これによりS6に対応する表示データ・メモリ(1E6H番地)には、×110を用意すればよいことが分かります。

S6と各コモン信号間のLCD駆動波形を図5 - 48(1/2バイアス法)、図5 - 49(1/3バイアス法)に示します。COM1の選択タイミングでS6が選択電圧のとき、およびCOM2の選択タイミングでS6が選択電圧のときに、LCD点灯レベルである $+V_{LCD}/-V_{LCD}$ の交流矩形波が発生されることが分かります。

図5 - 46 3時分割方式LCDの表示パターンと電極結線

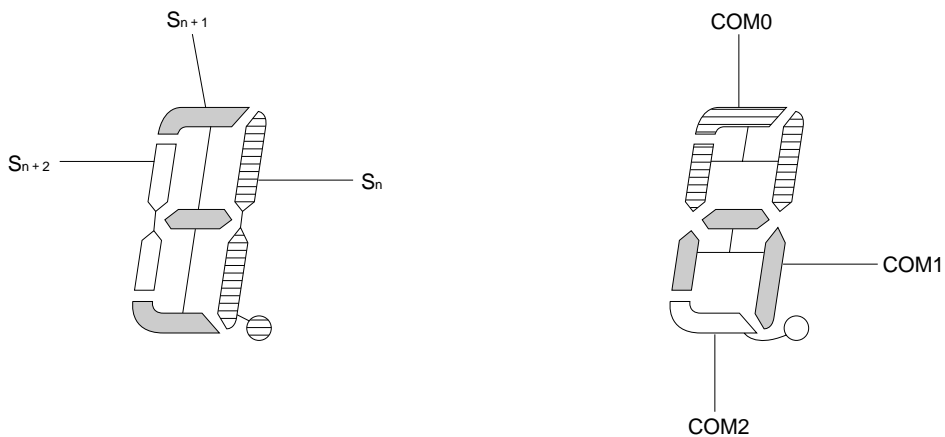
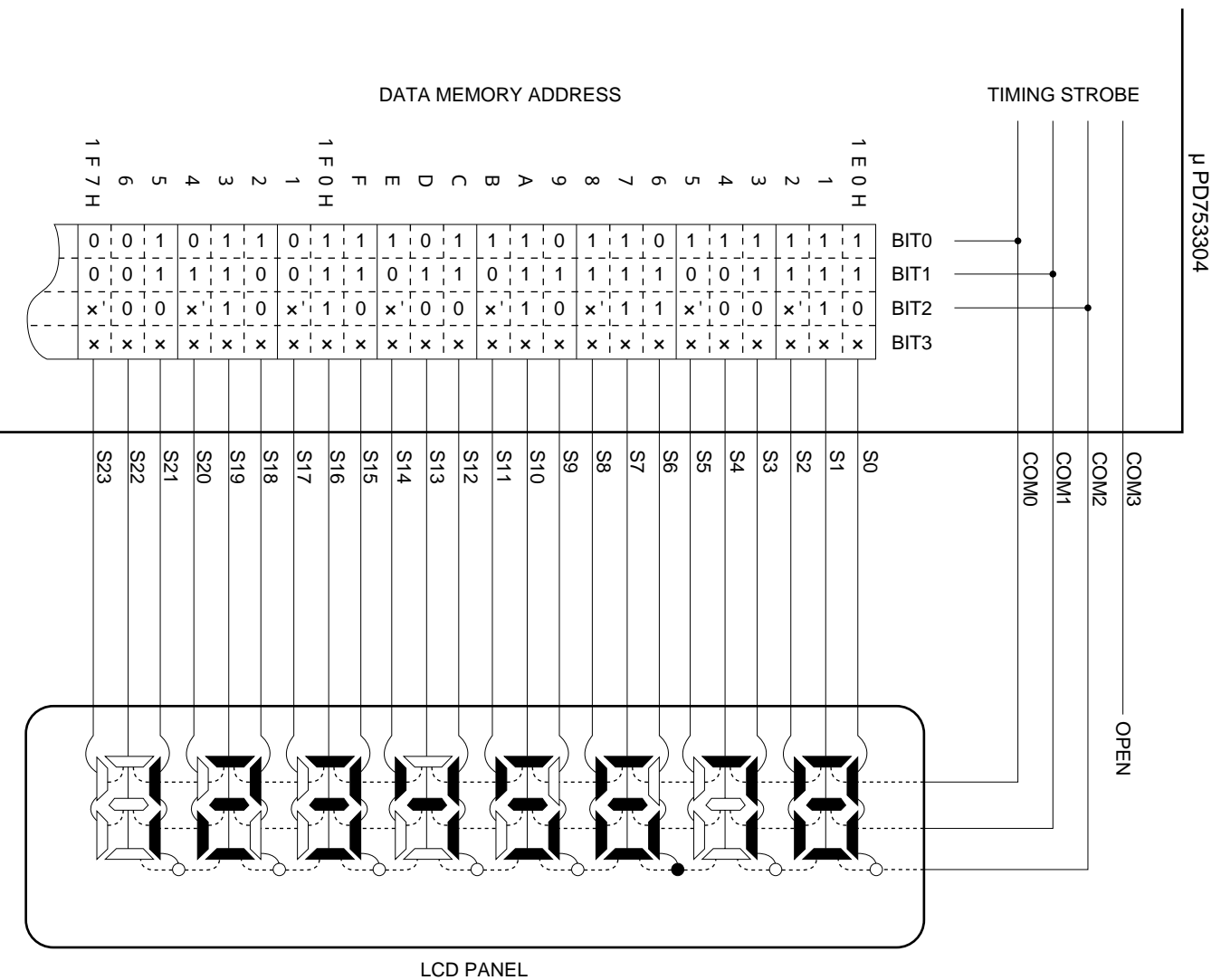


図5 - 47 3時分割LCDパネルの結線例



×': LCDパネルに対応セグメントがないため任意データをストア可能です。

× : 3時分割表示のため常に任意データをストア可能です。

図5 - 48 3時分割LCD駆動波形例（1/2バイアス法）

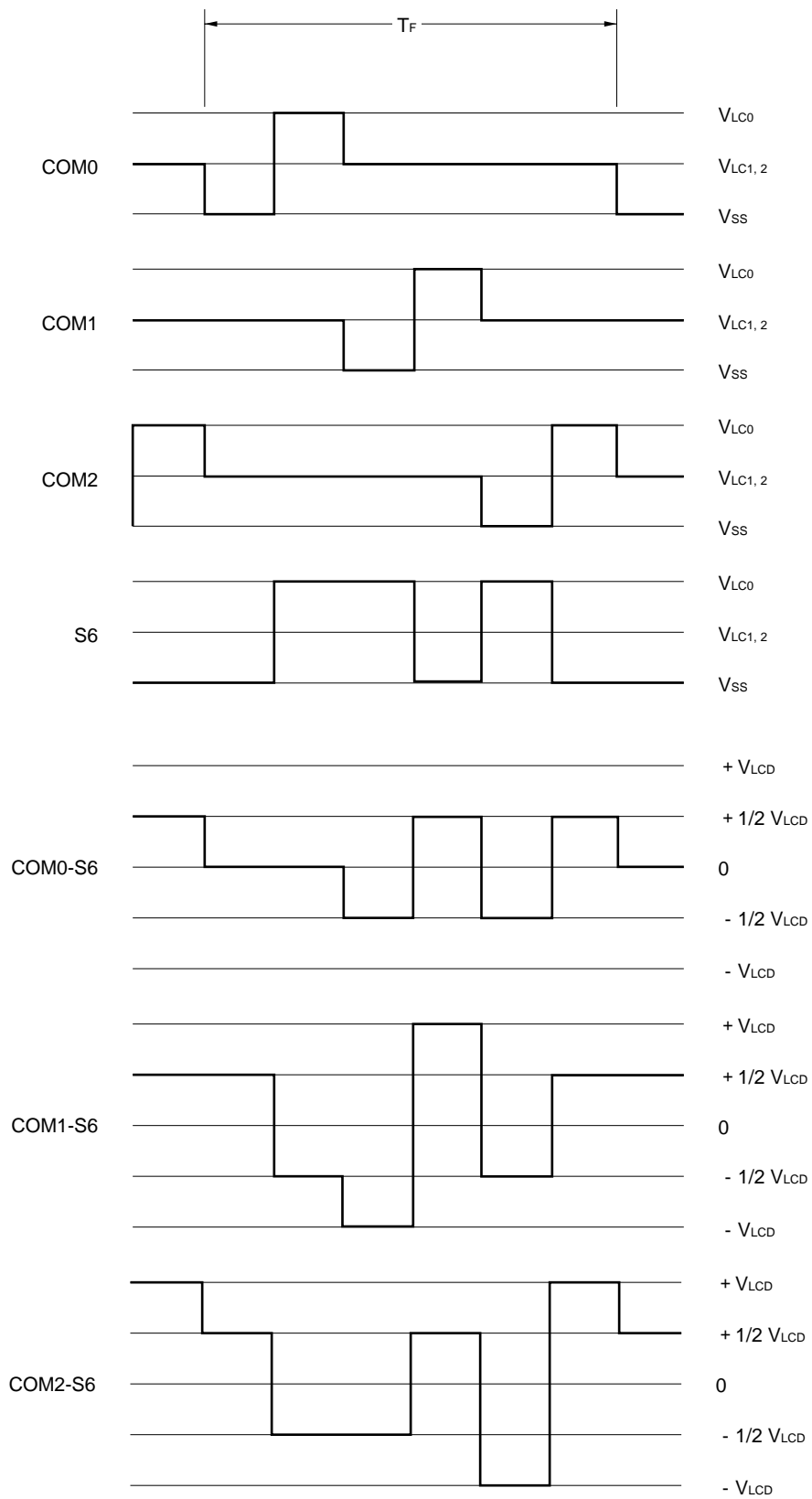
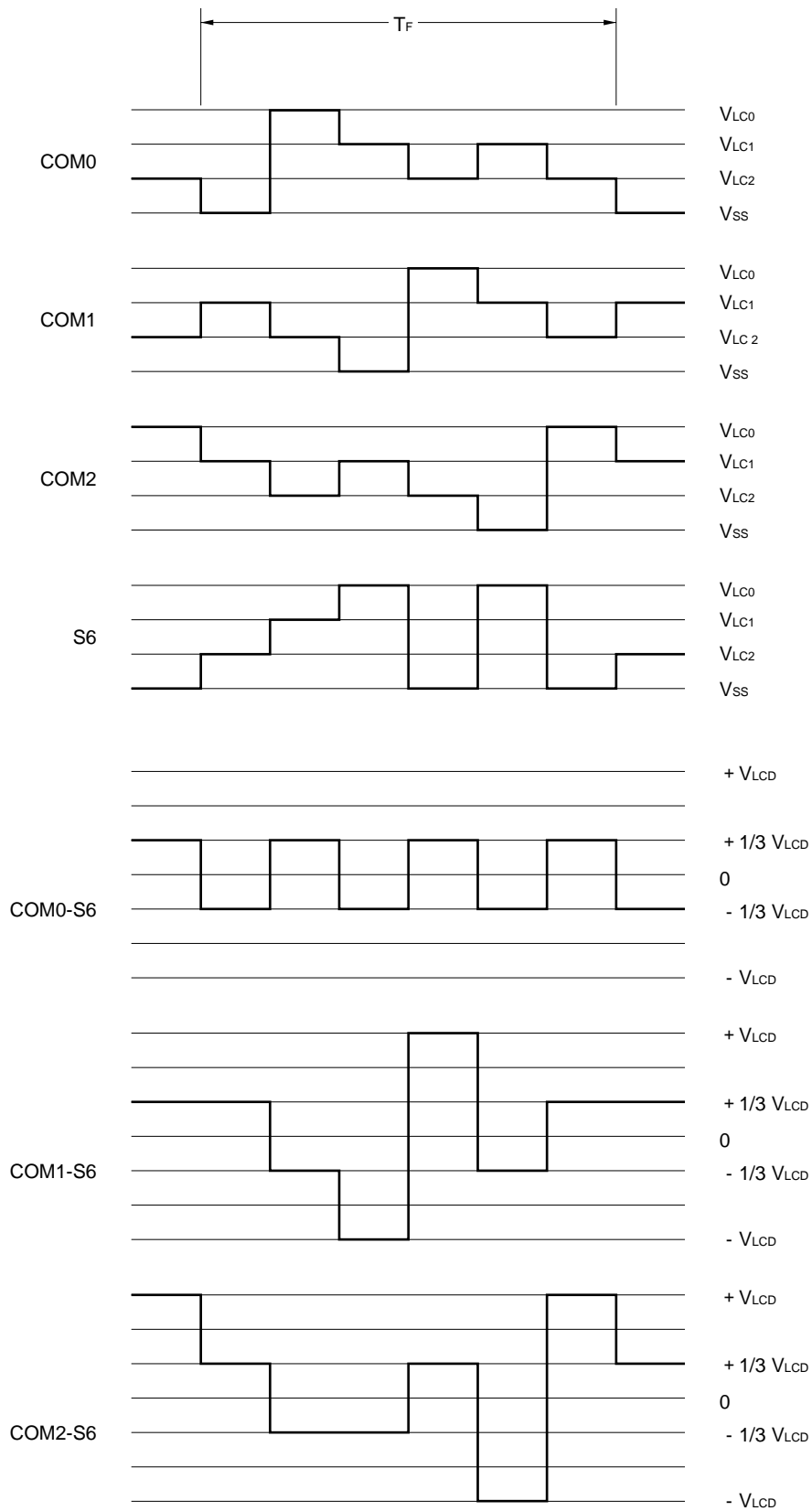


図5 - 49 3時分割LCD駆動波形例(1/3バイアス法)



(4) 4時分割表示例

図5 - 51は、図5 - 50の表示パターンを持つ4時分割方式の12桁LCDパネルとμPD753304のセグメント信号(S0-S23)およびコモン信号(COM0-COM3)との接続を示します。表示例は123456.789012で、表示データ・メモリ(1E0H-1F7H番地)の内容はこれに対応しています。

ここでは7桁目の6.(6.)を例にとって説明します。図5 - 50の表示パターンに従って、COM0-COM3の各コモン信号のタイミングで表5 - 16に示すような選択、非選択電圧をS12, S13端子に出力する必要があります。

表5 - 16 S12, S13端子の選択, 非選択電圧(4時分割表示例)

セグメント コモン	S12	S13
COM0	選	選
COM1	非	選
COM2	選	選
COM3	選	選

これによりS12に対応する表示データ・メモリ(1ECH番地)には、1101を用意すればよいことが分かります。

S12とCOM0, COM1信号間とのLCD駆動波形を図5 - 52に示します(図面の都合でCOM2, COM3に対する波形は省略)。COM0の選択タイミングでS12が選択電圧になるときに、LCD点灯レベルである $+V_{LCD} / -V_{LCD}$ の交流矩形波が発生されることが分かります。

図5 - 50 4時分割方式LCDの表示パターンと電極結線

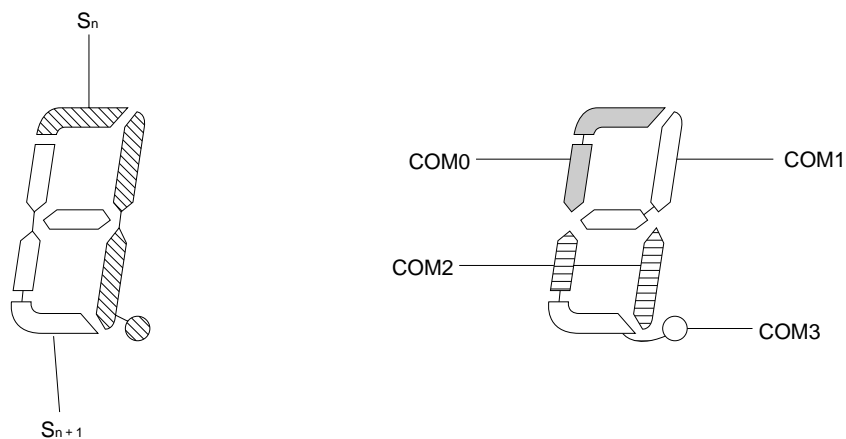


図 5 - 51 4 時分割 LCD パネルの結線例

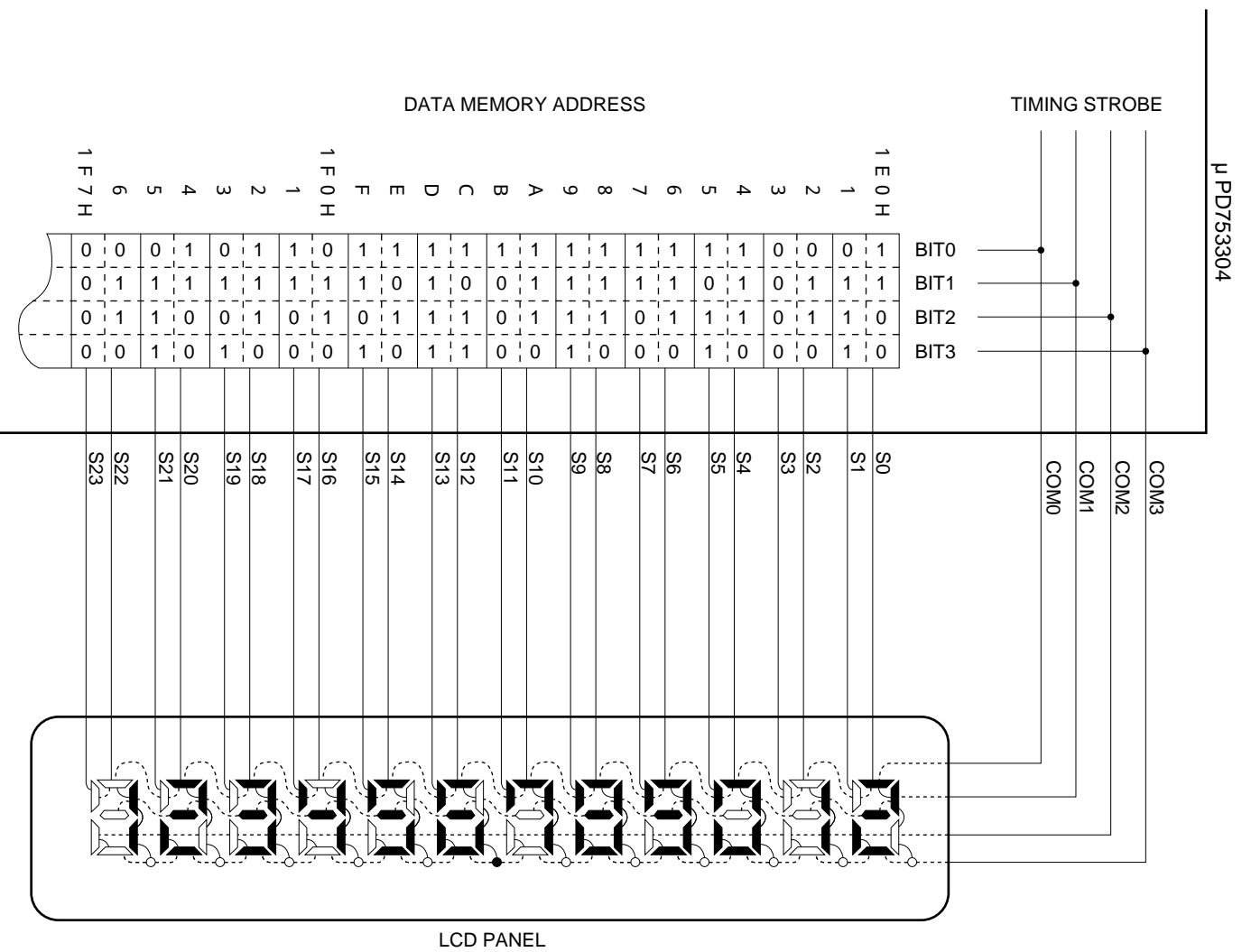
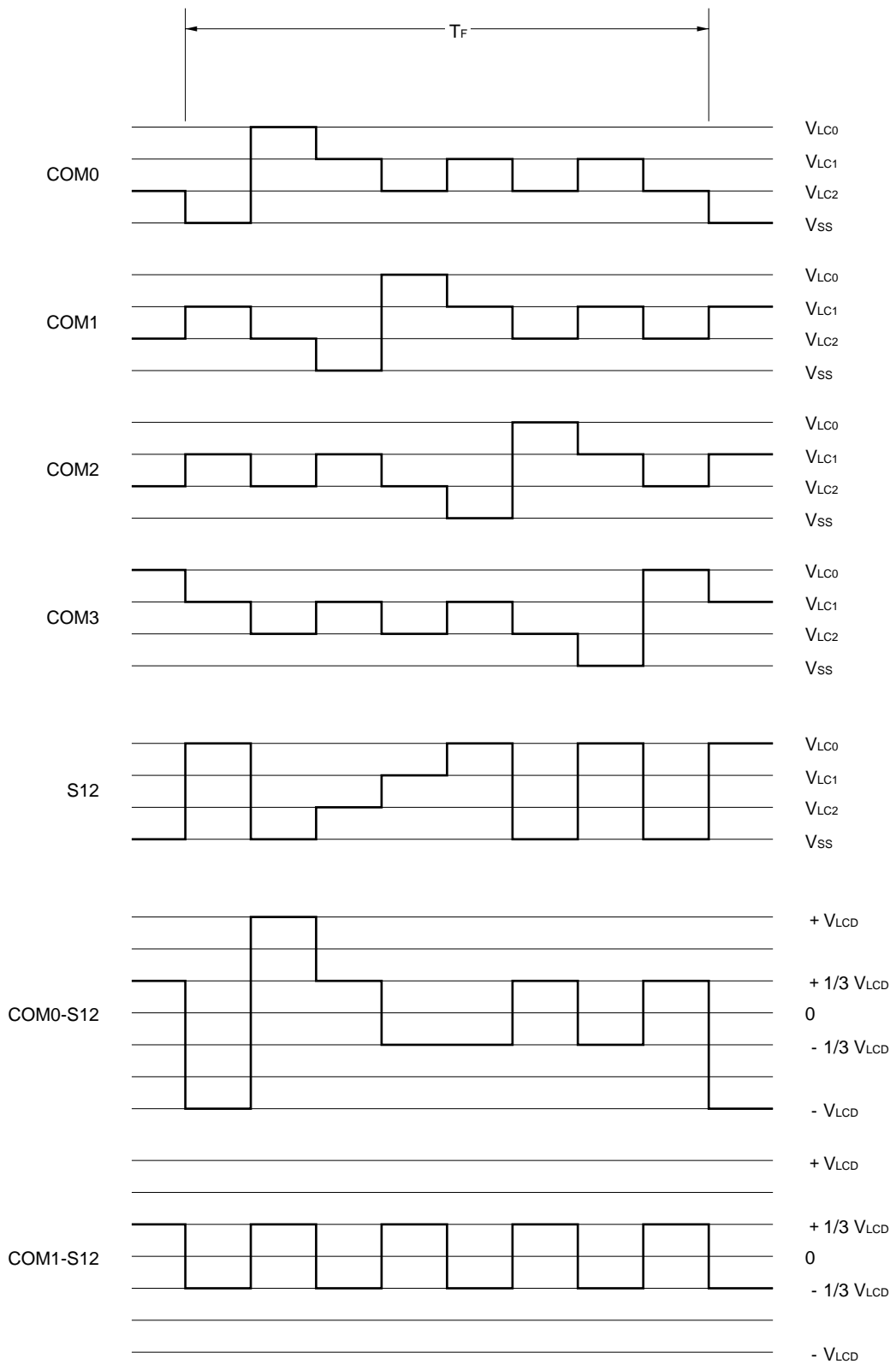




図5 - 52 4時分割LCD駆動波形例(1/3バイアス法)



(メモ)

## 第6章 割り込み機能とテスト機能

μPD753304には、3種類のベクタ割り込みソースと、1種類のテスト入力があり多彩な応用が可能です。また、μPD753304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)。

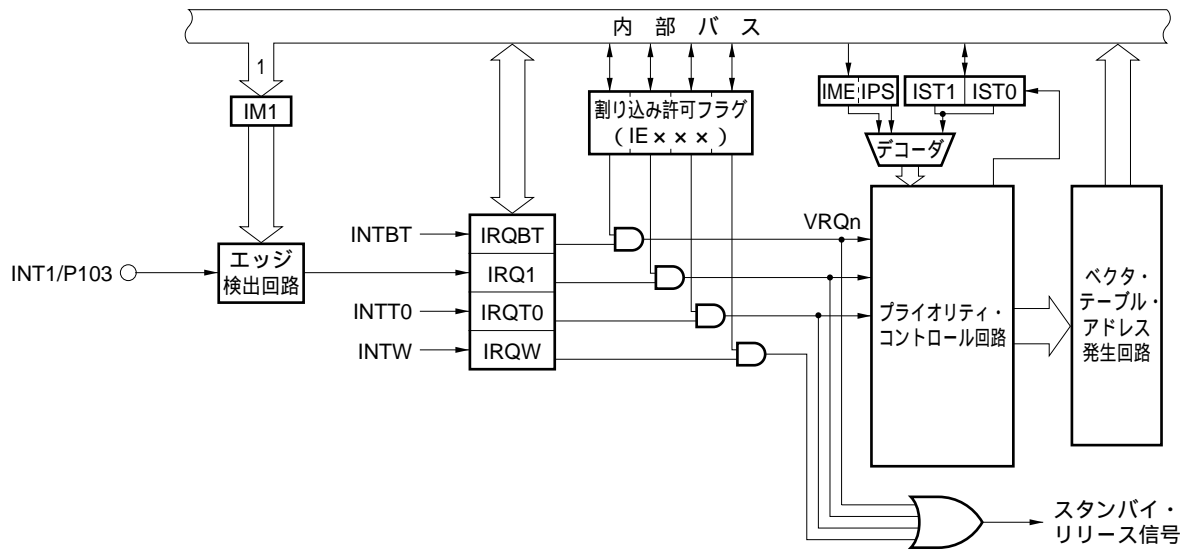
### (2) テスト機能

- (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



## 6.2 割り込みソースの種類とベクタ・テーブル

μPD753304には、次の3種類の割り込みソースがあり、ソフトウェア制御による多重割り込みが可能です。

表6 - 1 割り込みソースの種類

割り込みソース	内/外	割り込み 優先順位 <sup>注</sup>	ベクタ割り込み要求信号 (ベクタ・テーブル・アドレス)
INTBT (ベーシック・インターバル・タイマ/ウォッチドッグ・タイマからの基準時間間隔信号)	内	1	VRQ1 (0002H)
INT1 (立ち上がり, 立ち下がりエッジ選択)	外	2	VRQ3 (0006H)
INTT0 (タイマ・カウンタ0のカウント・レジスタとモジュール・レジスタとの一致信号)	内	3	VRQ5 (000AH)

注 割り込み優先順位は、複数の割り込み要求が同時に発生した場合に優先される順位です。

図6-2 割り込みベクタ・テーブル

アドレス				
0002H	MBE	RBE	INTBT	スタート・アドレス(上位6ビット)
				" (下位8ビット)
0006H	MBE	RBE	INT1	スタート・アドレス(上位6ビット)
				" (下位8ビット)
000AH	MBE	RBE	INTT0	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、割り込み優先順位と書かれているのは、複数の割り込み要求が同時に発生した場合や、複数の割り込み要求が保留されていた場合に、割り込みが実行される順位を示します。

ベクタ・テーブルには割り込み処理の開始アドレスと、割り込み処理中のMBE, RBEの設定値を書き込みます。ベクタ・テーブルの設定は、アセンブラ疑似命令 (VENTn : n = 1, 3, 5) によって行います。

例 INTBTのベクタ・テーブルの設定

VENT1     MBE = 0, RBE = 0, GOTOBT

- アドレス0002のベクタ・テーブル
- 割り込み処理ルーチンでのMBEの設定
- 割り込み処理ルーチンでのRBEの設定
- 割り込み処理ルーチンの開始アドレスを示すシンボル

**注意** VENTn (n = 1, 3, 5) 命令のオペランドに記述した内容 (MBE, RBE, 開始アドレス) は、2n番地のベクタ・テーブル・アドレスに格納されます。

例 INTBTとINTT0のベクタ・テーブルの設定

VENT1     MBE = 0, RBE = 0, GOTOBT ; INTBTスタート・アドレス  
 VENT5     MBE = 0, RBE = 1, GOTOT0 ; INTT0スタート・アドレス

## 6.3 割り込み機能を制御する各種ハードウェア

### (1) 割り込み要求フラグ, 割り込み許可フラグ

割り込み要求フラグ (IRQ x x x) には, 割り込みソースに対応して次に示す3つがあります。

INT1割り込み要求フラグ (IRQ1)  
 タイマ・カウンタ0 割り込み要求フラグ (IRQT0)  
 BT割り込み要求フラグ (IRQBT)

割り込み要求フラグは, 割り込み要求発生で “ 1 ” にセットされ, 割り込み処理が実行されると自動的に “ 0 ” にクリアされます。

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3つがあります。

INT1割り込み許可フラグ (IE1)  
 タイマ・カウンタ0 割り込み許可フラグ (IET0)  
 BT割り込み許可フラグ (IEBT)

割り込み許可フラグは, 内容が “ 1 ” のとき割り込みを許可し, “ 0 ” のとき禁止します。

割り込み要求フラグがセットされ, 割り込み許可フラグが割り込みを許可している場合, ベクタ割り込み要求 (VRQn : n = 1, 3, 5) 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 x x x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            IEBT            ; INTBTを許可  
 DI            IE1             ; INT1を禁止  
 SKTCLR    IRQT0           ; IRQT0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。

$\overline{\text{RESET}}$ 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
IRQ1	INT1/P103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
IRQT0	タイマ・カウンタ0からの一致信号でセット。	IET0

## (2) 割り込みプライオリティ選択レジスタ (IPS)

割り込みプライオリティ選択レジスタは、多重割り込みが可能な高位の割り込みを選択するレジスタで、下位3ビットにより指定します。

ビット3は、すべての割り込みを禁止するかどうかを指定する、割り込みマスタ許可フラグ(IME)となっています。

IPSは4ビット・メモリ操作命令で設定されますが、ビット3はEI/DI命令により、セット/リセットされます。

IPSの下位3ビットの内容を変更する場合は、割り込みが禁止されている(IME=0)状態で行う必要があります。

```

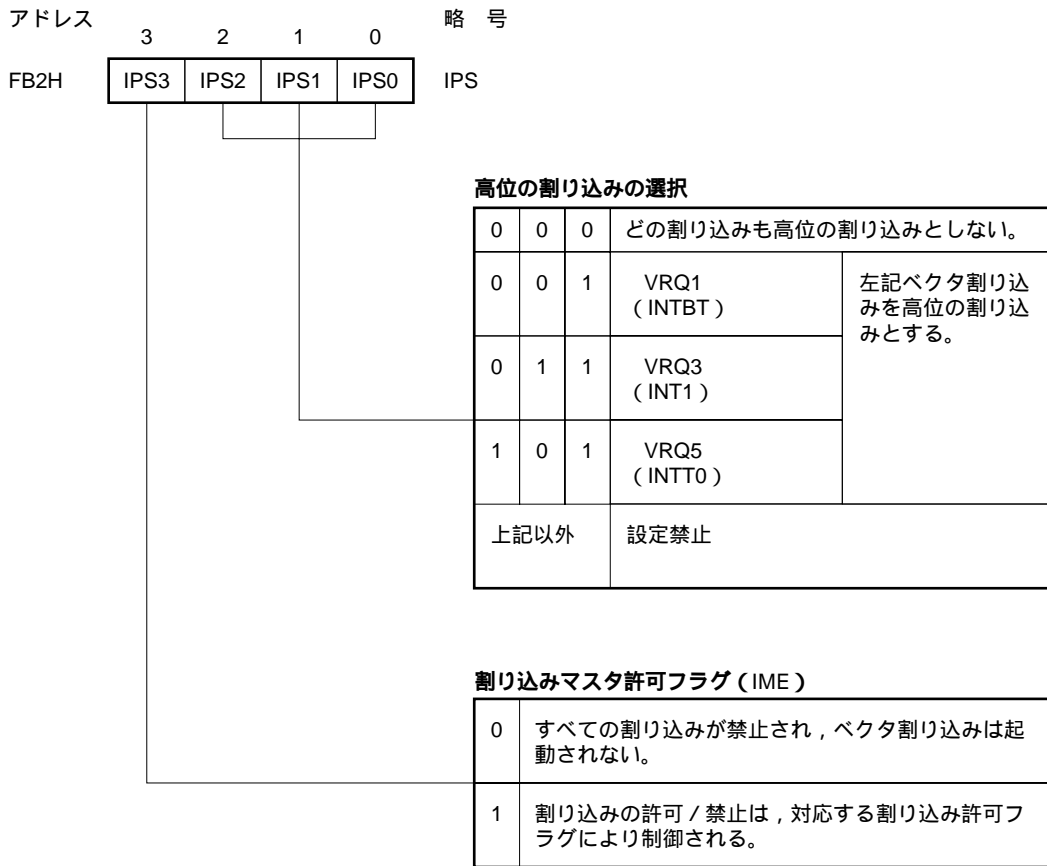
例  DI          ; 割り込み禁止
    CLR1  MBE
    MOV   A, #1011
    MOV   IPS, A      ; INT1を高位の割り込みにし、割り込みを許可する

```

RESET信号発生により全ビットが“0”にクリアされます。



図6-3 割り込みプライオリティ選択レジスタ



(3) INT1のハードウェア

INT1の構成を図6 - 4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 - 5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。

図6 - 4 INT1の構成

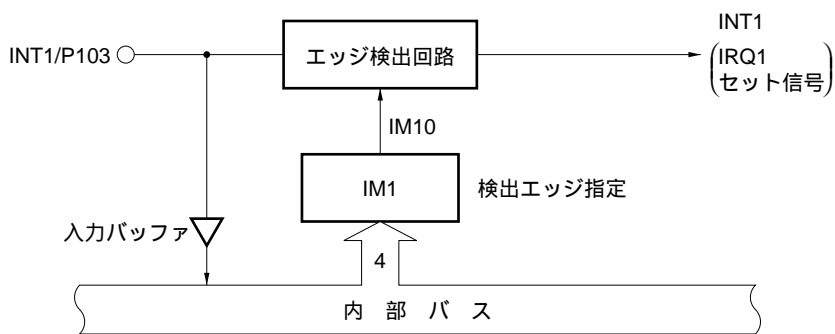
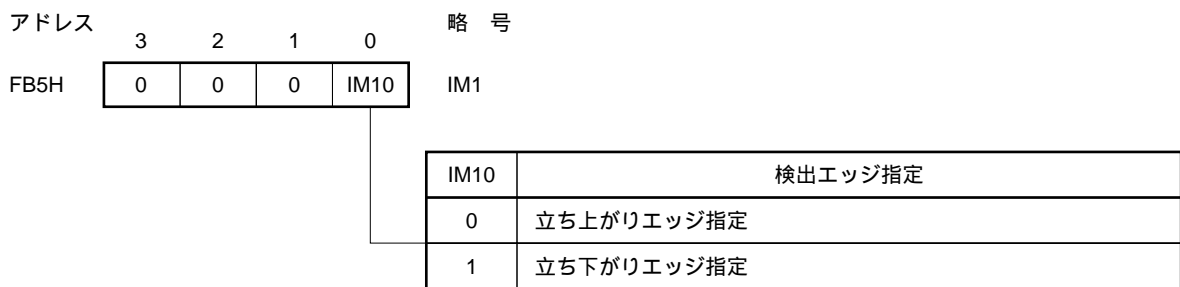


図6 - 5 INT1エッジ検出モード・レジスタのフォーマット



**注意** エッジ検出モード・レジスタを変更すると、割り込み要求フラグがセットされる場合がありますので、あらかじめ割り込みを禁止してモード・レジスタを変更し、CLR1命令によって割り込み要求フラグをクリアしてから、割り込みを許可してください。

**(4) 割り込みステータス・フラグ**

割り込みステータス・フラグ (IST0, IST1) はCPUが現在実行中の処理のステータスを示すフラグでPSWに含まれています。

割り込みプライオリティ・コントロール回路は、このフラグの内容によって表6-3に示されるように、多重割り込みの制御を行います。

IST0, IST1は4ビット操作命令またはビット操作命令により変更することができるため、実行中のステータスを変えて多重割り込みを行うこともできます。IST0, IST1をビット単位で操作する場合は、MBEの設定に関係なくいつでも行うことができます。

なお、IST0, IST1を操作する場合は、操作前に必ずDI命令を実行して割り込みを禁止し、操作後にEI命令を実行して割り込みを許可するようにしてください。

IST1, IST0は、割り込み受け付け時にほかの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。

RESET信号の発生によりフラグの内容は“0”にクリアされます。

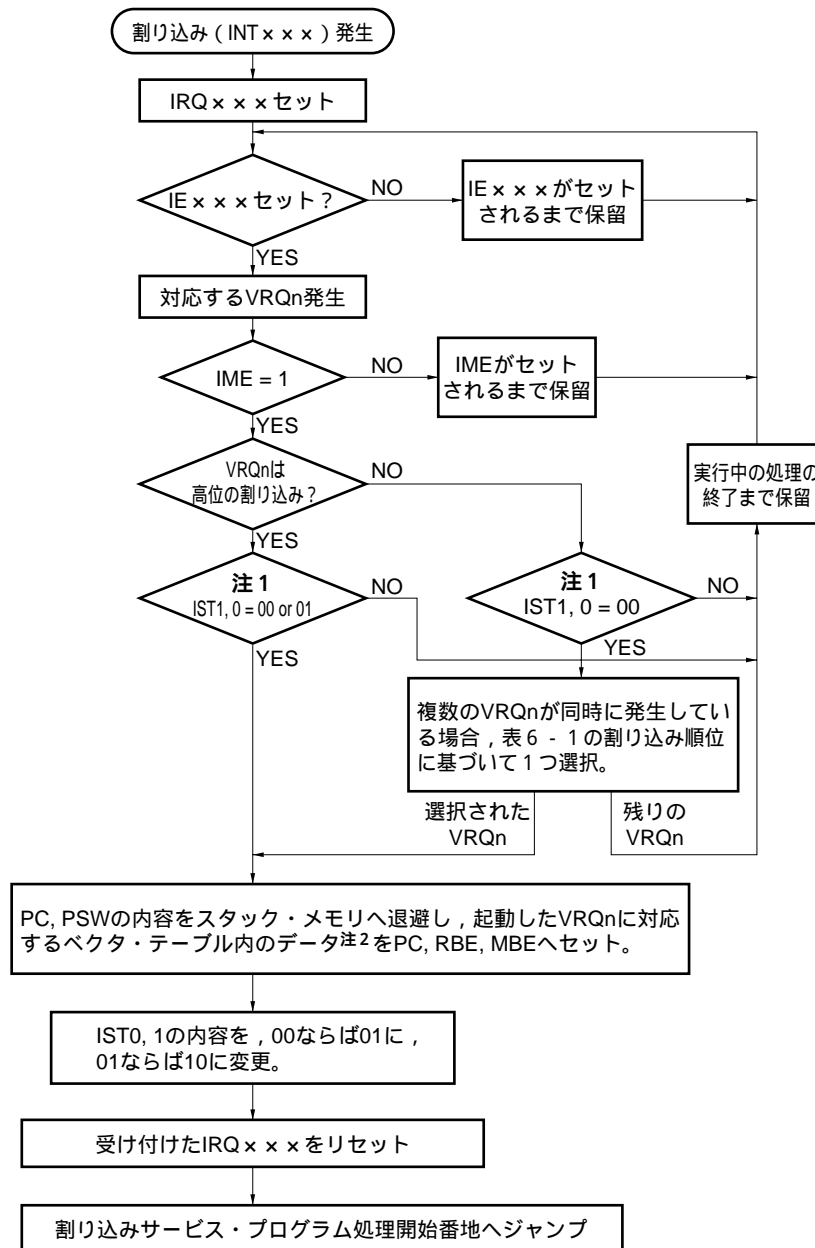
**表6-3 IST1, IST0と割り込み処理状態**

IST1	IST0	実行中の処理のステータス	CPUの処理内容	受け付け可能な割り込み要求	割り込み受け付け後	
					IST1	IST0
0	0	ステータス0	通常のプログラム処理中	すべての割り込みを受け付け可能	0	1
0	1	ステータス1	低位の割り込み、または高位の割り込み処理中	高位の割り込みのみ受け付け可能	1	0
1	0	ステータス2	高位の割り込み処理中	すべての割り込みの受け付け禁止	-	-
1	1	設定禁止				

## 6.4 割り込みシーケンス

割り込みが発生すると、以下の手順で処理されます。

図6-6 割り込み処理シーケンス



注1 . IST1, 0 : 割り込みステータス・フラグ (PSWのビット3, 2 ; 表6-3 参照)

2 . 各ベクタ・テーブルには、割り込みサービス・プログラムの開始アドレスと、割り込み開始時の MBE, RBE の設定値をストアしておきます。

## 6.5 多重割り込み処理制御

μPD753304は、次の2つの方法により多重割り込みができます。

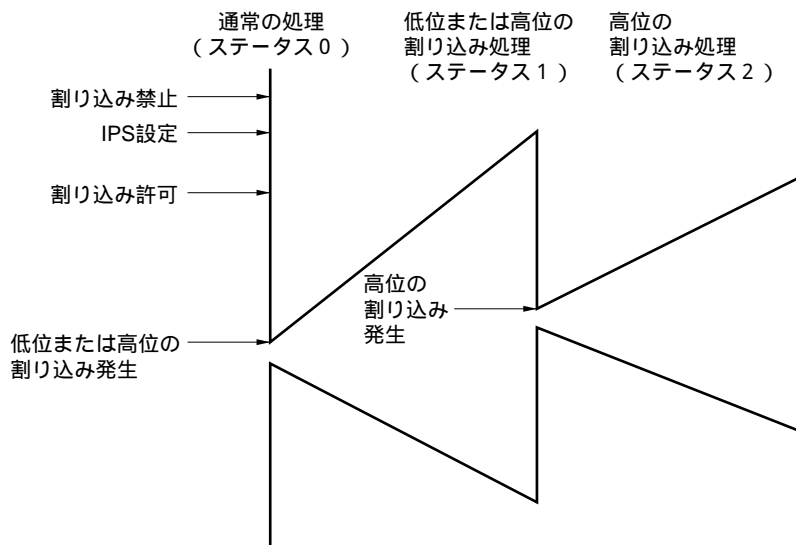
### (1) 高位の割り込みを指定する多重割り込み

この方法は、μPD753304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6-7、表6-3参照）。

したがって、使用する割り込みのうち1つだけを二重割り込みできるようにしたい場合には、この方法を使えば、割り込み処理中の割り込み許可、禁止などの動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6-7 高位の割り込みによる多重割り込み



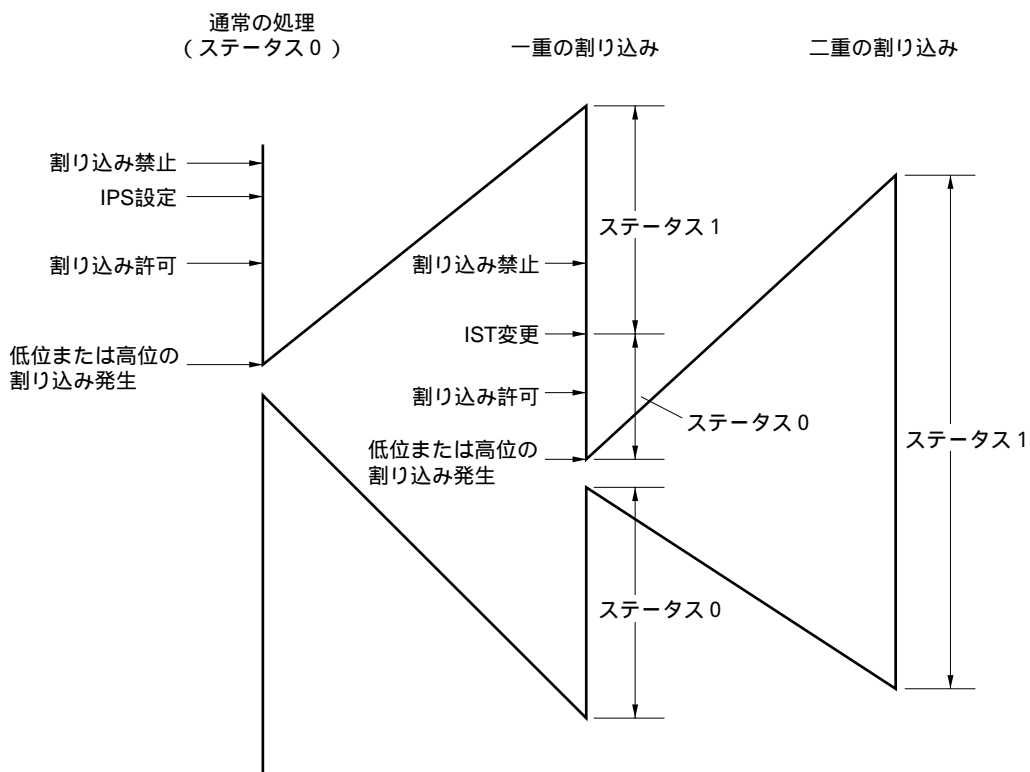
(2) 割り込みステータス・フラグを変更する多重割り込み

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6-8 割り込みステータス・フラグの変更による多重割り込み

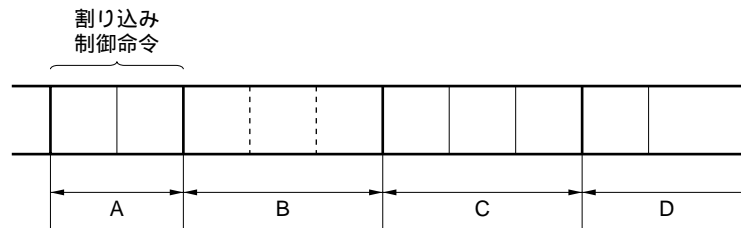


## 6.6 割り込み処理までのマシン・サイクル

μPD753304では、割り込み要求フラグ（IRQ×××）がセットされてから割り込みルーチンのプログラムを実行するまでのマシン・サイクルは、次のようになります。

### （1）割り込み制御命令実行中にIRQ×××がセットされた場合

割り込み制御命令実行中にIRQ×××がセットされた場合には、続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。



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

**注意 1**．割り込み制御命令が連続している場合には、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**．IRQ×××がセットされるタイミング（上図のA）または、そのあとのタイミングでDI命令を実行すると、セットされたIRQ×××の割り込み要求は次にEI命令を実行するまで保留されます。

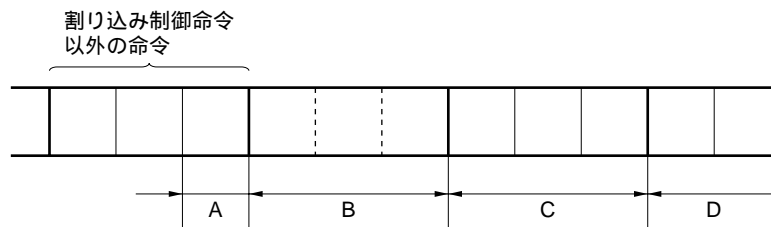
**備考 1**．割り込み制御命令とは、割り込み関係のハードウェア（データ・メモリのFB×H番地）を操作する命令です。この命令には、DI命令、EI命令があります。

**2**．割り込み処理の3マシン・サイクルは、割り込み受け付け時に操作されるスタックの操作時間などです。

(2)(1) 以外の命令実行中にIRQ×××がセットされた場合

(a) 実行中の命令の最終マシン・サイクルでIRQ×××がセットされた場合

この場合は、実行中の命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。

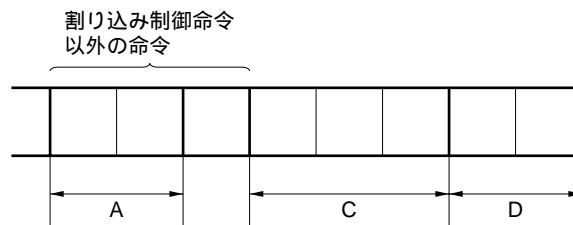


- A : IRQ×××セット
- B : 続く1命令を実行 (1-3マシン・サイクル; 命令により異なる)
- C : 割り込み処理 (3マシン・サイクル)
- D : 割り込みルーチンの実行

**注意** 続く命令が割り込み制御命令の場合には、最後に実行した割り込み制御命令に続く1命令を実行したのち、3マシン・サイクルの割り込み処理を行ったあと割り込みルーチンのプログラムを実行します。またIRQ×××がセットされたあとに実行される割り込み制御命令がDI命令の場合には、セットされたIRQ×××の割り込み要求は保留されません。

(b) 実行中の命令の最終マシン・サイクル以前でIRQ×××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 : IRQ×××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



## 6.7 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) 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。

### (2) 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。

## 6.8 割り込みの応用

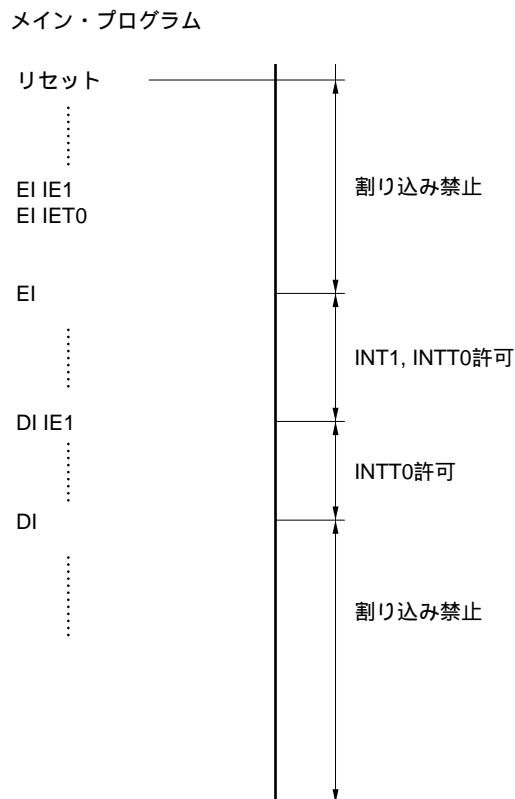
割り込み機能を使用する場合は、まず、メイン・プログラムで次のように設定します。

- (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) INT1を使用する場合は、アクティブ・エッジの選択 (IM1セット)。
- (c) 二重割り込み (高位の割り込みによる) を使用する場合は、IPSを設定 (同時にIMEセットもできる)。
- (d) 割り込みマスタ許可フラグをセット (EI 命令)。

割り込み処理プログラムでは、ベクタ・テーブルによってMBE, RBEが設定されます。ただし、「高位の割り込み」に指定された割り込みでは、レジスタ・バンクの退避と設定が必要です。

割り込み処理プログラムからの復帰はRETI命令を使用します。

(1) 割り込みの許可・禁止



$\overline{\text{RESET}}$ 信号により、すべての割り込みは禁止。

EI IE × × × 命令により、割り込み許可フラグをセット。

この段階ではまだすべての割り込みは禁止。

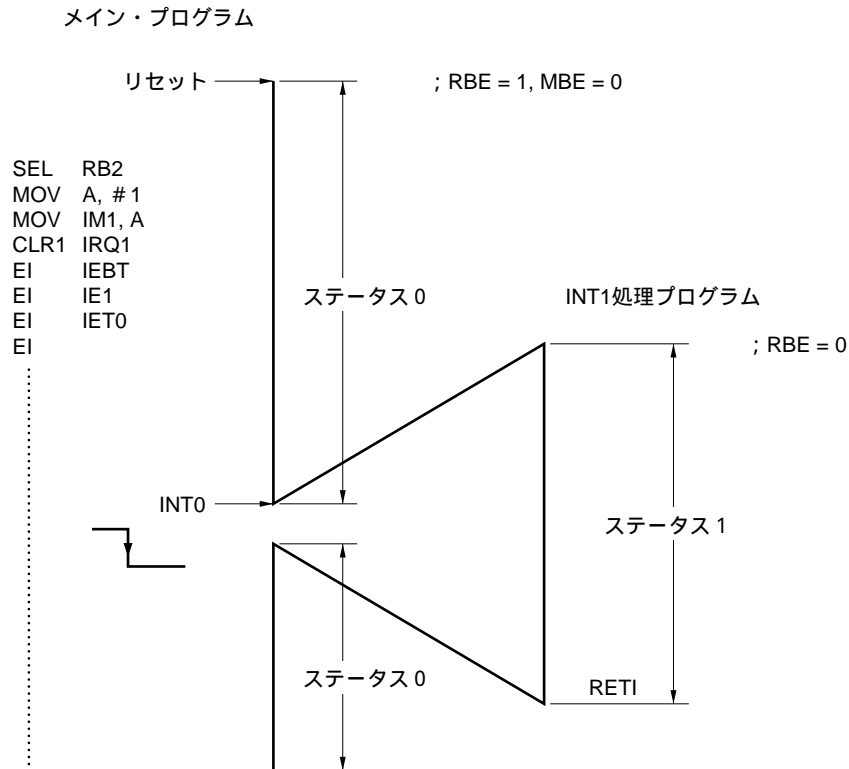
EI 命令により、割り込みマスタ許可フラグをセット。

この段階で、INT1, INTT0が許可される。

DI IE × × × 命令により、割り込み許可フラグをクリア、INT1が禁止される。

DI 命令によりすべての割り込みが禁止。

(2) INTBT, INT1 (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  
(すべて低位の割り込み)



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。

RBE = 1 をリセット・ベクタ・テーブルで指定し, SEL RB2命令でレジスタ・バンク 2, 3 を使用。

INT1を立ち下がりエッジ・アクティブに設定。

EI, EI IE × × × 命令により割り込み許可。

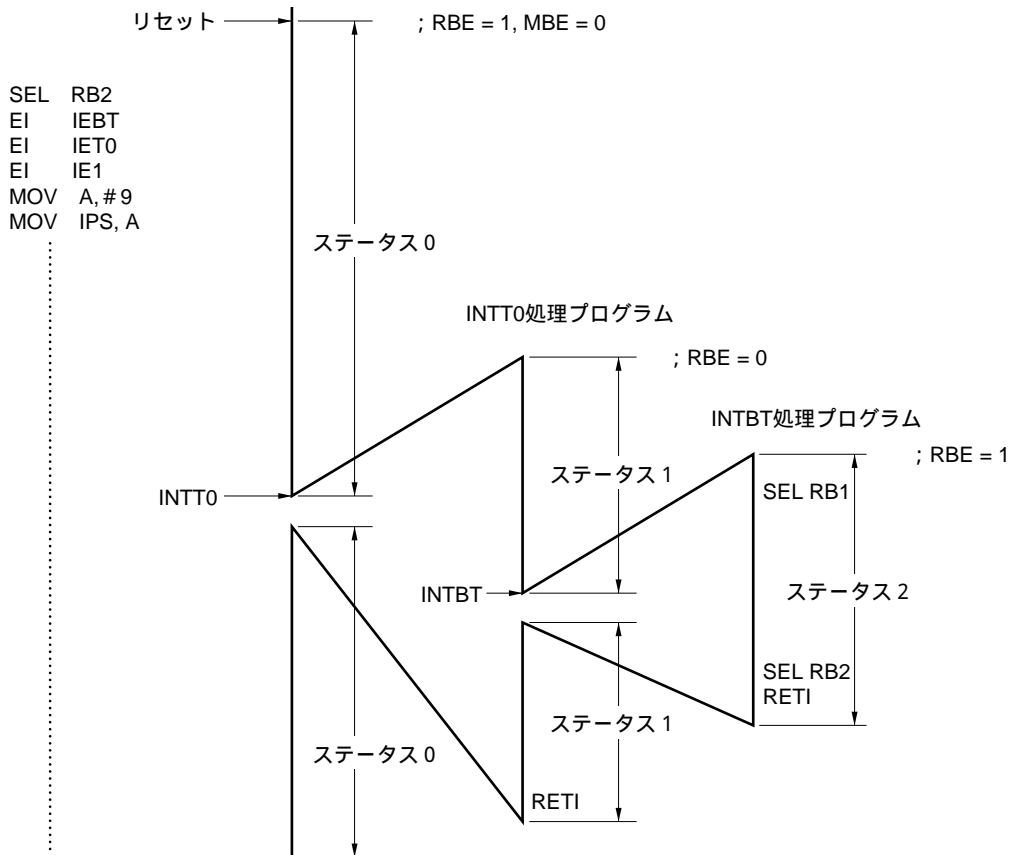
INT1の立ち下がりにより, INT1割り込み処理プログラムをスタート。ステータスは 1 に変更され, すべての割り込みは禁止される。

RBE = 0 となり, レジスタ・バンク 0, 1 を使用。

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, INT1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。

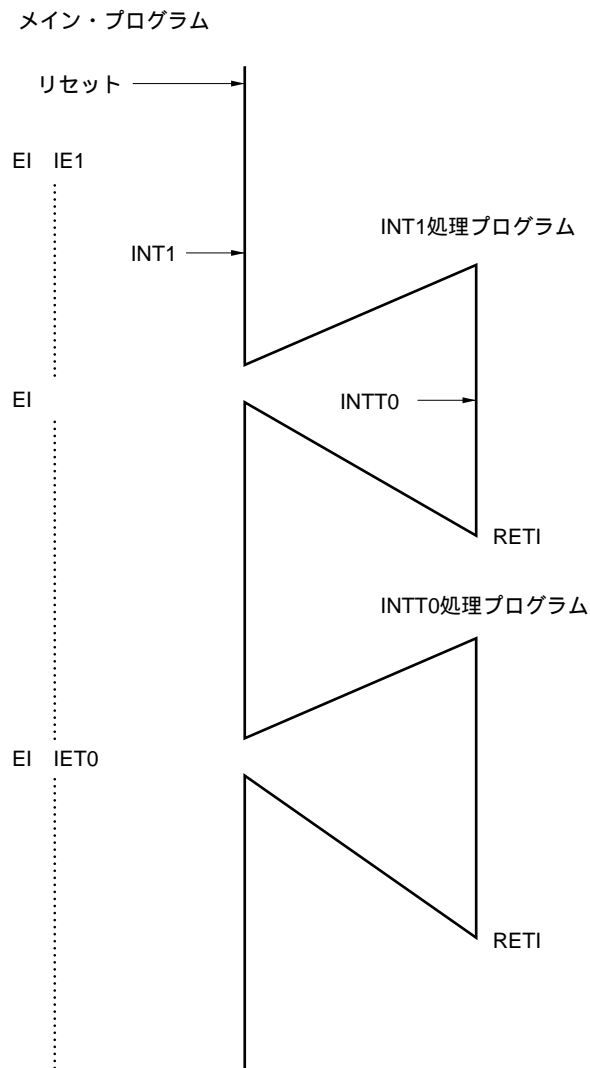
低位の割り込みINTT0発生により, INTT0処理プログラム開始, ステータス 1 となり低位の割り込みは禁止。RBE = 0 としてレジスタ・バンク 0 を使用。

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)。

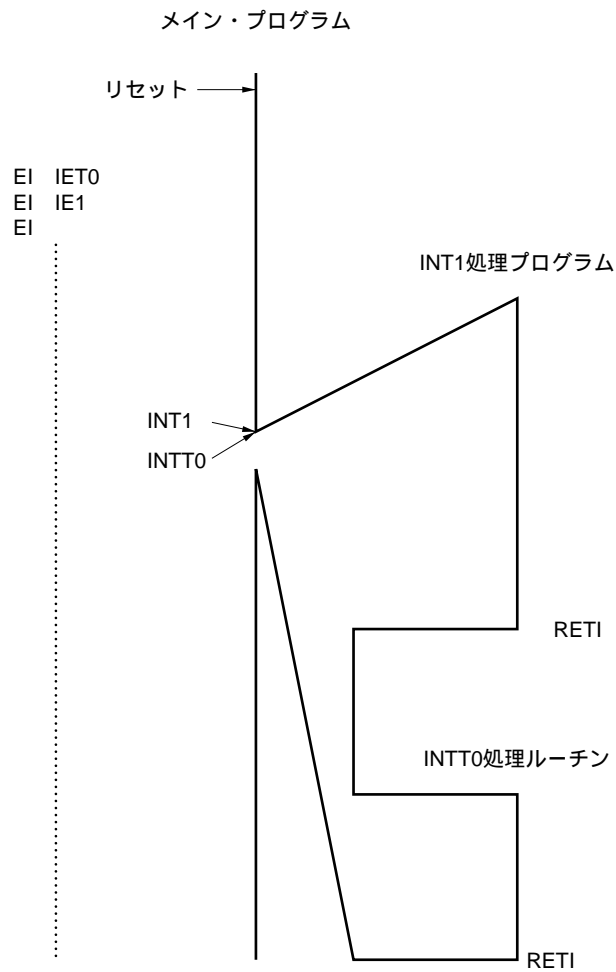
RBSを 2 に戻し, リターン。ステータスは 1 に戻る。

(4) 保留割り込みの実行——割り込み禁止中の割り込み入力——



割り込み禁止中にINT1がセットされても要求フラグは保留される。  
 EI命令で割り込みが許可された時点で、INT1処理プログラム・スタート。  
 と同様  
 保留されていたINTT0が許可された時点で、INTT0処理プログラム・スタート。

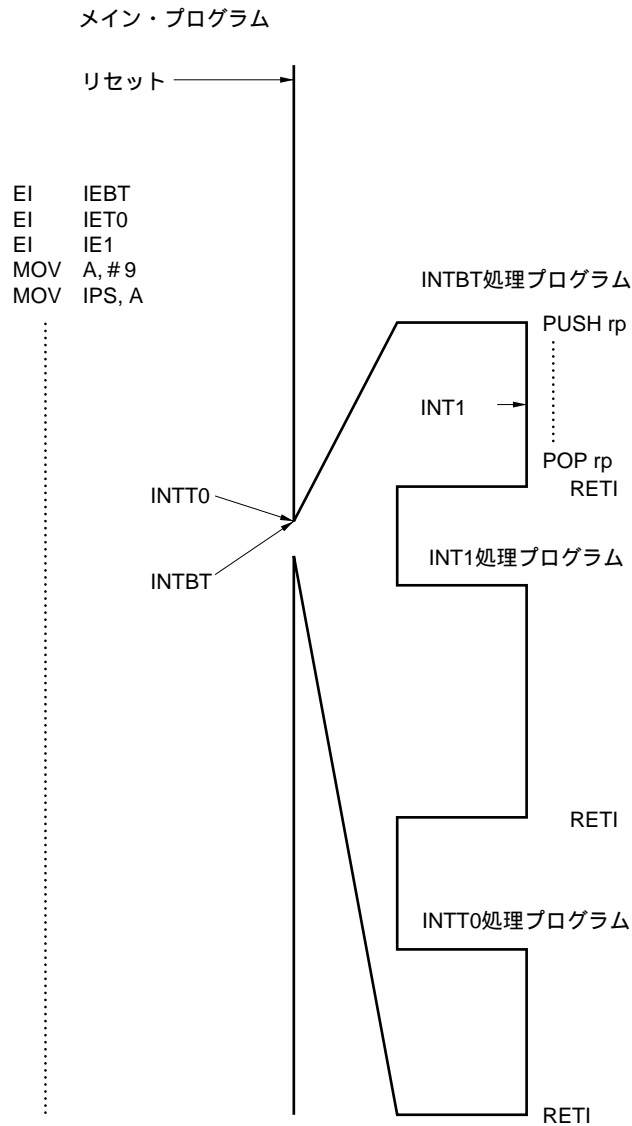
(5) 保留割り込みの実行——低位の割り込みが同時に2つ発生——



「低位の割り込み」INT1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INT1を先に実行（INTT0は保留される）。

RETI命令により、INT1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, INT1を低位の割り込み)——



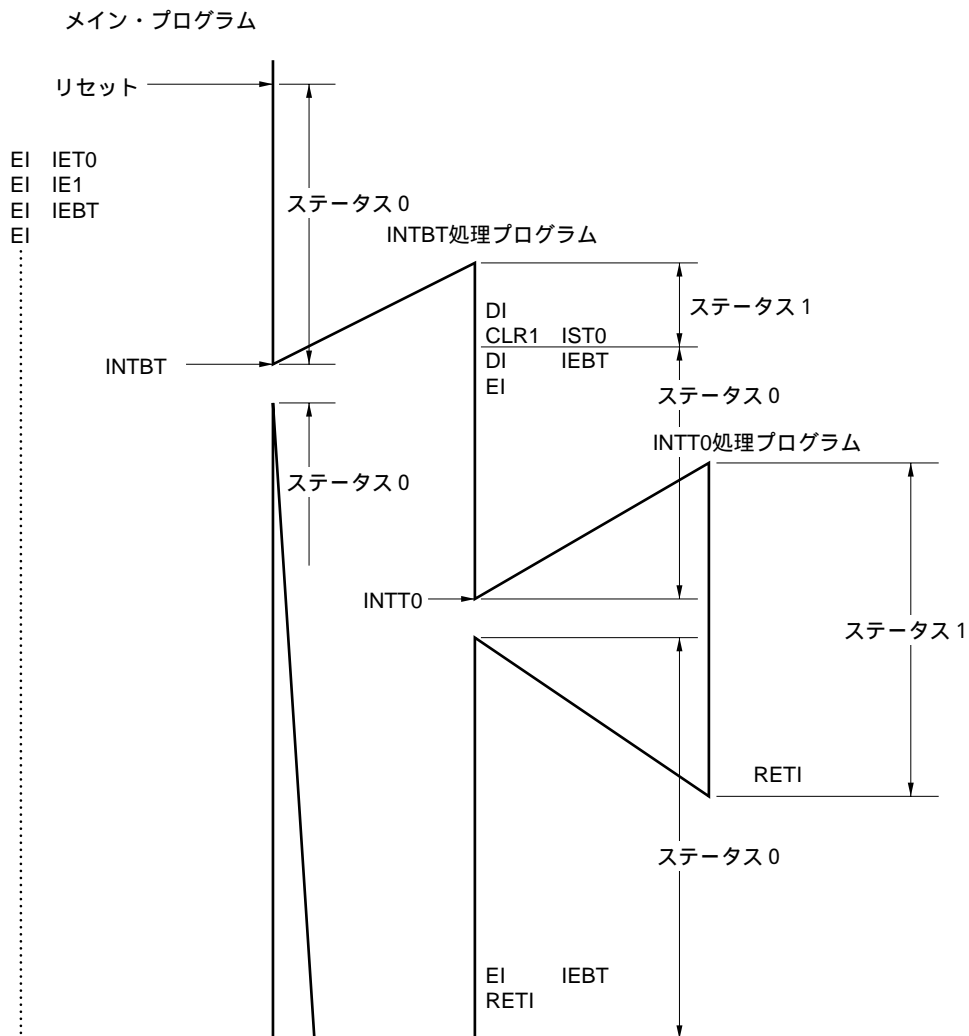
高位の割り込みINTBTと低位の割り込みINTT0が同時に発生すると、高位の割り込み処理を開始 (高位の割り込み処理中に、高位の割り込みが発生するおそれがないことが確実ならDI IE × ×は不要)。

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INT1処理が実行される。

INT1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INT1は二重割り込みを許可。INTBTは一重割り込み——



二重割り込みを許可しない割り込みINTBTの発生によりINTBT処理プログラムをスタート。ステータスは1となる。

IST0のクリアにより、ステータスを0にする。二重割り込みを許可しないINTBTを禁止。

二重割り込みを許可するINTT0発生により、二重割り込み実行、ステータス1となり、すべての割り込みは禁止される。

INTT0処理終了により、ステータスは0に戻る。

禁止していたINTBTを許可して復帰。



## 6.9 テスト機能

### 6.9.1 テスト・ソースの種類

μPD753304には、1種類のテスト・ソースがあります。

表6-4 テスト・ソースの種類

テスト・ソース	内/外
INTW (時計用タイマからの信号)	内

### 6.9.2 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 (IRQW) は、テスト要求発生 (INTW) 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 (IEW) は、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6-5にテスト要求フラグのセット信号を示します。

表6-5 テスト要求フラグのセット信号

テスト要求フラグ	テスト要求フラグのセット信号	テスト許可フラグ
IRQW	時計用タイマからの信号でセット。	IEW

〔メモ〕

## 第7章 スタンバイ機能

μPD753304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。

- ★ いずれのモードでも、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。

STOPモード

HALTモード

STOPモード、HALTモードはそれぞれ以下に示すような機能を持っています。

### (1) STOPモード

メイン・システム・クロック発振回路、サブシステム・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 ( $V_{DD} = 2.5 \text{ V}$ 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。

割り込み要求によるSTOPモード解除時のウェイト時間は、マスク・オプションにより512/fccまたはウェイトなしを選択できます。したがって、割り込み要求によってすぐに処理を開始しなければならない場合には、ウェイトなしを選択してください。

なお、512/fccを選択し、割り込み要求によってすぐに処理を開始しなければならない場合は、HALTモードを選択してください。

SOSのビット3を“1”にセットしたあとに、STOP命令を実行するとサブシステム・クロックの発振を停止させることができます。

### (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継続します。このモードでは、STOPモードほどの消費電流の低減はできませんが、割り込み要求により、すぐに処理を再開したい場合や、時計動作のような間欠動作をさせたい場合に有効です。

いずれのモードでも、スタンバイ・モードに設定される直前のレジスタ、フラグ、データ・メモリの内容はすべて保持されます。また、入出力ポートの出力ラッチ、出力バッファの状態も保持されますから、入出力ポートの状態を、あらかじめ、システム全体の消費電流が最も小さくなるように処理しておきます。

**使用上の注意を次ページに示します。**

注意1 . STOPモードとHALTモードは、メイン・システム・クロック、サブシステム・クロックのどちらの動作状態でも使用できます。

★ いずれのモードでも、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。

2 . LCDコントローラ/ドライバ、時計用タイマとタイマ・カウンタの動作クロックがメイン・システム・クロック $f_{cc}$ のとき、メイン・システム・クロックを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 $f_{ct}$ 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・クロックとサブシステム・クロックのいずれでも設定可。	
動作状態	クロック発生回路	メイン・システム・クロック発振停止。サブ発振回路停止許可フラグ(SOS.3)を1に設定するとサブシステム・クロックも発振停止。	CPUクロックのみ停止(発振継続)。
	ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ	動作停止。	メイン・システム・クロック発振時のみ動作。 BTモード : 基準時間間隔でIRQBTをセット WTモード : BTのオーバフローによりリセット発生
	タイマ・カウンタ	SOS.3を0に設定し、カウント・クロックにf <sub>CT</sub> を選択した場合のみ動作可能。	メイン・システム・クロック停止時に、カウント・クロックにメイン・システム・クロックの分周を選択した場合のみ動作不可能。
	LCDコントローラ/ドライバ	SOS.3を0に設定し、LCDCLにf <sub>CT</sub> を選択した場合のみ動作可能。	動作可能。
	時計用タイマ	SOS.3を0に設定し、カウント・クロックにf <sub>CT</sub> を選択した場合のみ動作可能。	動作可能。
	外部割り込み	SOS.3を0に設定した場合のみ動作可能。	
	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。	

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** 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、ウエイト<sup>注</sup>したあと動作モードに戻ります。

**注** マスク・オプションにより次のどちらかを選択することができます。

512/fcc (142  $\mu$ s : 3.6 MHz動作時)

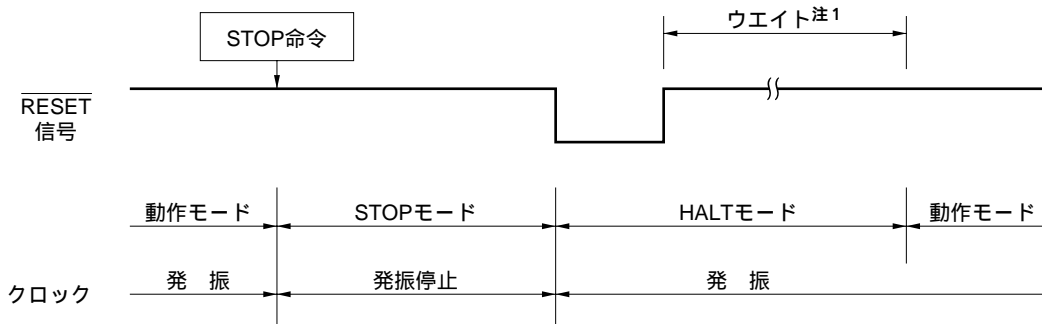
ウエイトなし

## 7.2 スタンバイ・モードの解除

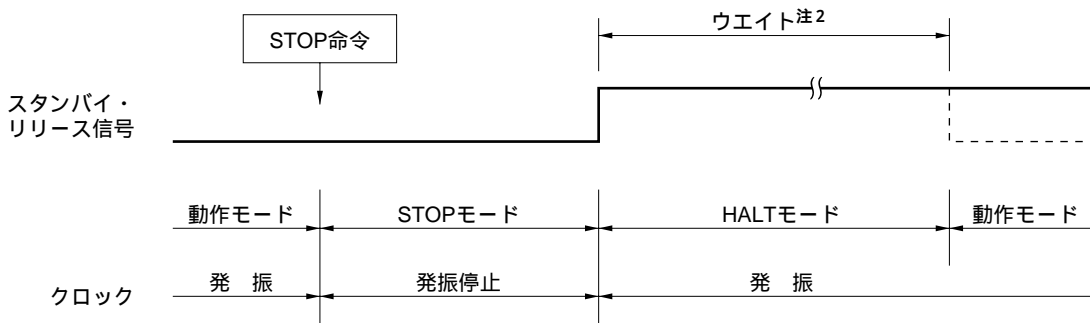
STOPモード、HALTモードいずれも割り込み許可フラグで許可されている割り込み要求信号の発生、および $\overline{\text{RESET}}$ 信号発生により解除されます。各モードの解除動作は図7-1のようになります。

図7-1 スタンバイ・モードの解除動作(1/2)

### (a) STOPモードの $\overline{\text{RESET}}$ 信号発生による解除



### (b) STOPモードの割り込み発生による解除



注1 .  $56/f_{cc}$  (  $15.6 \mu\text{s}$  :  $3.6 \text{ MHz}$ 動作時 )

2 . マスク・オプションにより次のどちらかを選択することができます。

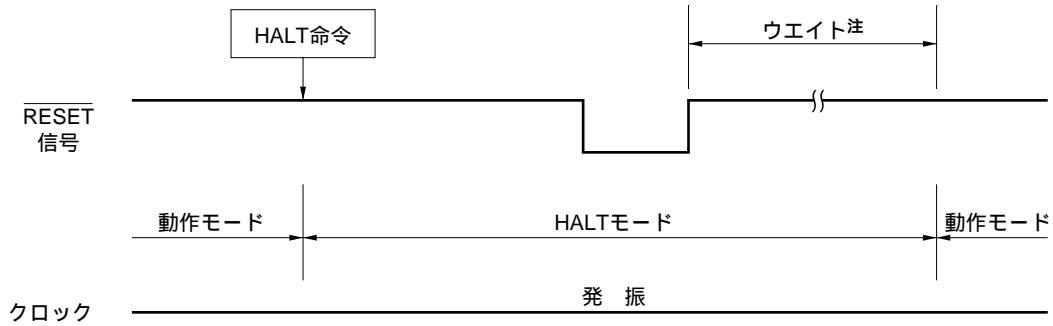
$512/f_{cc}$  (  $142 \mu\text{s}$  :  $3.6 \text{ MHz}$ 動作時 )

ウエイトなし

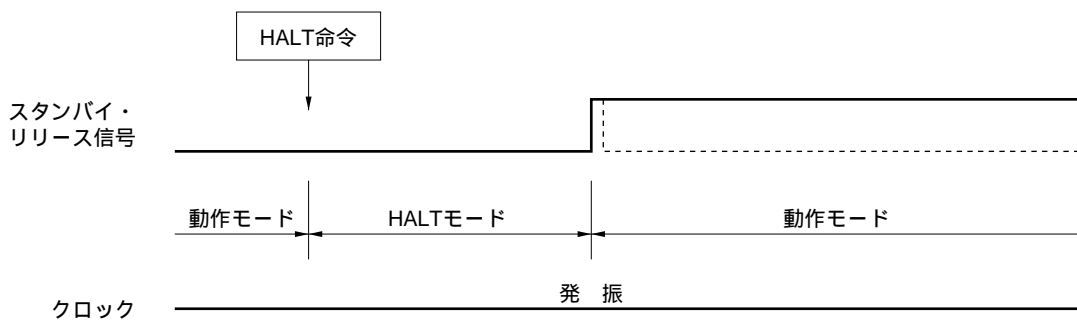
備考 破線は、スタンバイを解除した割り込み要求が受け付けられた場合です。

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 56/fcc (15.6  $\mu$ s : 3.6 MHz動作時)

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。



## 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によって解除された場合はベクタ割り込みは発生しませんから、(a)と同様の処理を行います。

## 7.4 マスク・オプションの選択

$\mu$ PD753304のスタンバイ機能では、割り込み発生によるSTOPモード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。

512 / fcc (142  $\mu$ s : 3.6 MHz動作時)

ウェイトなし

## 7.5 スタンバイ・モードの応用

スタンバイ・モードを使用する場合には次のような手順で行います。

割り込み入力あるいはポート入力による電源断などスタンバイ・モード設定要因の検出。

入出力ポートの処理（消費電流が最小となるよう処理する）。

特に，入力ポートはオープンにしないでください。必ずロウ・レベルかハイ・レベルを入力してください。

スタンバイ・モードを解除する割り込みの指定。

解除後の動作指定（割り込み処理をするかしないかによりIMEを操作）。

解除後のCPUクロックの指定（切り替える場合は，スタンバイ・モード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）。

解除時のウェイト時間の選択（マスク・オプションで指定してください）。

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 ( $f_{cc} = 3.6 \text{ MHz}$ 動作時)

## 次の条件でSTOPモードを使用する場合

INT1の立ち下がりエッジ入力によりSTOPモードを設定し、立ち上がりエッジ入力で解除する。入出力ポートは、すべてハイ・インピーダンスとする（ハイ・インピーダンスにおいて消費電流が少なくなるように、外部で端子が処理されている場合）。

プログラムで使用する割り込みはINTBT、INTT0。ただし、これらをSTOP解除には使用しない。

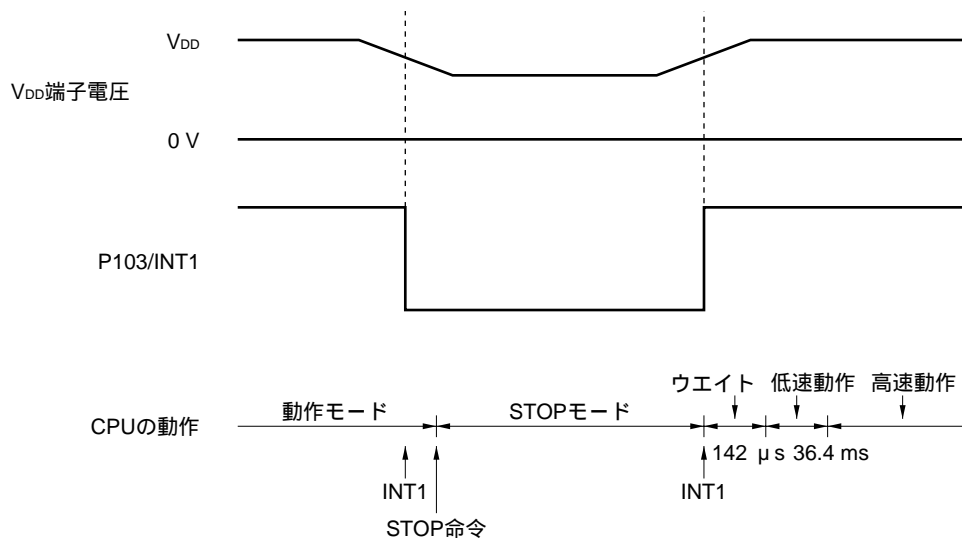
解除後も割り込み許可する。

解除後は、最低速のCPUクロックで動作スタートする。

解除時のウェイト時間はマスク・オプションで $142 \mu\text{s}$ を選択する。

解除後、電源の安定のためさらに $36.4 \text{ ms}$ ウェイトする。またP103/INT1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

( INT1処理プログラム , MBE = 0 )

```

VSUB1 :   SKT    PORT10.3      ; P103 = 1 ?
          BR     PDOWN        ; パワーダウン
          SET1   BTM. 3       ; パワーオン
WAIT :    SKT    IRQBT        ; 36.4 msウエイト
          BR     WAIT
          SKT    PORT10.3     ; チャタリングをチェック
          BR     PDOWN
          MOV    A,  0011B
          MOV    PCC, A       ; 高速モード設定
          ( MOV   XA,  x x H ) ; ポート・モード・レジスタをセット
            MOV   PMGm, XA
          SET1   IM1.0
          EI     IEBT
          EI     IET0
          RETI
PDOWN :   MOV    A,  0        ; 最低速モード
          MOV    PCC, A
          MOV    XA,  00H
          MOV    LCDM, XA     ; LCD表示オフ
          MOV    LCDC, A
          MOV    PMGD, XA     ; 入出力ポート・ハイ・インピーダンス
          DI     IEBT        ; INTBT, INTT0禁止
          DI     IET0
          MOV    A,  1011B
          MOV    BTM, A       ; ウエイト時間 36.4 ms
          CLR1   IM1.0
          NOP
          STOP                    ; STOPモードをセット
          NOP
          RETI
    
```

(2) HALTモードの応用 (fcc = 3.6 MHz動作時)

次の条件で間欠動作する場合

INT1の立ち下がり、サブシステム・クロック (f<sub>CT</sub> = 47 kHz) 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0.35 sec間隔で、間欠動作する。

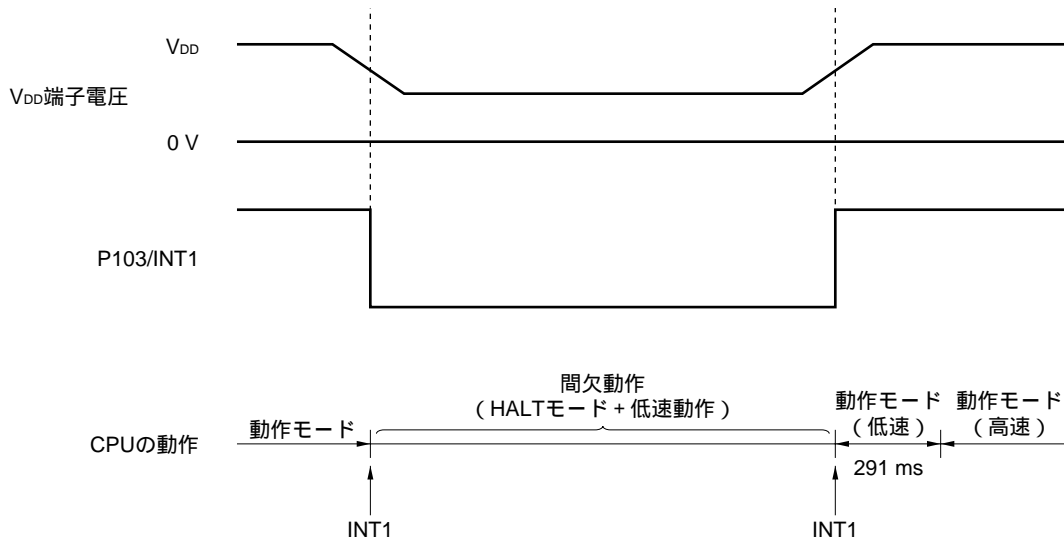
INT1の立ち上がり、メイン・システム・クロックに再び切り替える。

INTBTは使用しない。

解除後、電源の安全のためにさらに291 msウェイトする。

また、P103/INT1端子を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

(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     IE1
EI     IEW
EI                      ; 割り込み許可

```

(メイン・ルーチン)

```

SKT    PORT10.3   ; 電源OK?
HALT                   ; パワーダウン・モード
NOP                      ; 電源OK?
SKTCLR IRQW         ; 0.35 secフラグあり?
BR     MAIN         ; NO
CALL   WATCH       ; 時計サブルーチン

```

```

MAIN :  ⋮
        ⋮
        ⋮

```

(INT1処理ルーチン)

```

VINT1 : SKT    PORT10.3   ; 電源OK?, MBE = 0
        BR     PDOWN
        CLR1   SCC. 3    ; メイン・システム・クロック発振開始
        MOV    A,  #1000B
        MOV    BTM, A
WAIT1 : SKT    IRQBT     ; 291 msウエイト
        BR     WAIT1
        SKT    PORT10.3   ; 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
        SET1   IM1.0
        RETI
PDOWN : MOV    XA,  #00H
        MOV    LCDM, XA   ; LCD表示オフ
        MOV    LCDC, A
        SET1   SCC.0     ; サブシステム・クロックに切り替え

```

```
MOV      A, #6
WAIT2 :  INCS   A           ; 20マシン・サイクル以上待機注
        BR    WAIT2
        CLR1  IM1.0
        SET1  SCC.3       ; メイン・システム・クロック発振停止
        RETI
```

**注** システム・クロックとCPUクロックの切り替えについては、5.2.3 システム・クロックとCPUクロックの設定を参照してください。

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

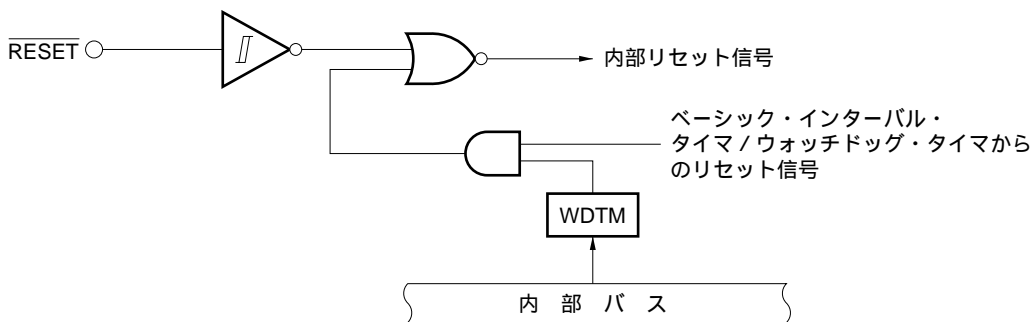
(メモ)



## 第8章 リセット機能

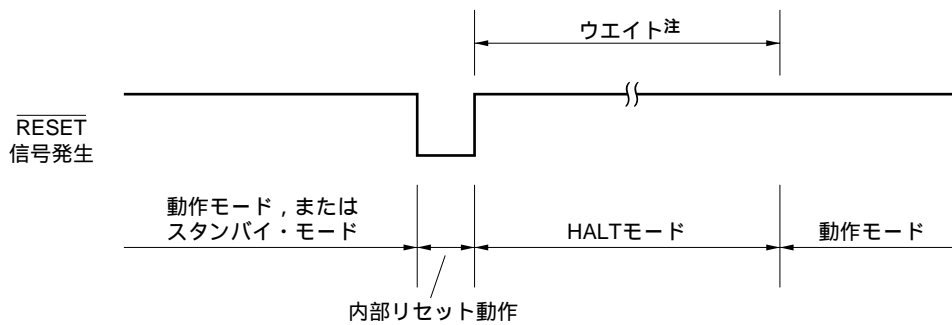
リセット入力には、外部リセット信号 ( $\overline{\text{RESET}}$ ) とベーシック・インターバル・タイマ/ウォッチドッグ・タイマからのリセット信号の2種類があります。どちらか一方のリセット信号が入力されると、内部リセット信号が発生します。図8 - 1 にそれぞれの構成を示します。

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 56/f<sub>cc</sub> (15.6  $\mu$ s : 3.6 MHz動作時)

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)		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
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)	0	0
	バンク許可フラグ(MBE, RBE)	プログラム・メモリの0000H番 地のビット6をRBE, ビット7 をMBEにセット	左に同じ
スタック・ポインタ(SP)		不 定	不 定
スタック・バンク選択レジスタ(SBS)		1000B	1000B
データ・メモリ(RAM)		保 持	不 定
汎用レジスタ(X, A, H, L, D, E, B, C)		保 持	不 定
バンク選択レジスタ(MBS, RBS)		0, 0	0, 0
ベーシック・イ	カウンタ(BT)	不 定	不 定
ンターバル・タ	モード・レジスタ(BTM)	0	0
イマ/ウォッチ	ウォッチドッグ・タイマ許可フラグ	0	0
ドッグ・タイマ	(WDTM)		
タイマ・ カウンタ(T0)	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
時計用タイマ	モード・レジスタ(WM)	0	0
クロック発生回路, クロック出力回路	プロセッサ・クロック・コントロー ル・レジスタ(PCC)	0	0
	システム・クロック・コントロー ル・レジスタ(SCC)	0	0
	クロック出力モード・レジスタ (CLOM)	0	0
サブ発振回路コントロール・レジスタ(SOS)		0	0
LCDコントロー ラ/ドライバ	表示モード・レジスタ(LCDM)	0	0
	表示制御レジスタ(LCDC)	0	0
	LCD/ポート選択レジスタ(LPS)	0	0
割り込み機能	割り込み要求フラグ (IRQxxx)	リセット(0)	リセット(0)
	割り込み許可フラグ (IExxx)	0	0
	プライオリティ選択レジスタ(IPS)	0	0
	INT1モード・レジスタ(IM1)	0	0

表8 - 1 各ハードウェアのリセット後の状態 (2/2)

ハードウェア		スタンバイ・モード中の $\overline{\text{RESET}}$ 信号発生	動作中の $\overline{\text{RESET}}$ 信号発生
デジタル・ ポート	出力バッファ (P30-P33)	オン	オン
	出力バッファ (P80-P83, P100-P103)	オフ	オフ
	出力ラッチ (P30-P32, P80-P83, P100-P103)	クリア (0)	クリア (0)
	出力ラッチ (P33)	セット (1)	セット (1)
	入出力モード・レジスタ (PMGA)	0FH	0FH
	入出力モード・レジスタ (PMGC, D)	00H	00H
	プルアップ抵抗指定レジスタ (POGB)	01H	01H

(メモ)

## 第9章 マスク・オプション

### 9.1 $\overline{\text{RESET}}$ 端子のマスク・オプション

$\overline{\text{RESET}}$ 端子は、マスク・オプションにより60 k (typ.)のプルアップ抵抗の内蔵を指定することができます。

### ★ 9.2 LCD表示モードのマスク・オプション

$\mu$ PD753304はLCD駆動用電源供給用に分割抵抗が内蔵されていますが、外部端子として $V_{LC0}$ - $V_{LC2}$ 端子（LCD駆動用電源）とBIAS端子（外付け分割抵抗カット用端子）を持っていません。そこで各表示モードに対応するために、次の3つのマスク・オプションから選択することができます。

スタティック・モード（BIAS- $V_{LC0}$ 間ショート， $V_{LC0}$ - $V_{LC1}$ 間オープン）

1/2 バイアス・モード（BIAS- $V_{LC0}$ 間ショート， $V_{LC1}$ - $V_{LC2}$ 間ショート）

1/3 バイアス・モード（BIAS- $V_{LC0}$ 間ショート）

### 9.3 スタンバイ機能のマスク・オプション

$\mu$ PD753304のスタンバイ機能では、マスク・オプションによりウエイト時間を選択することができます。ウエイト時間とは、割り込み発生によってSTOPモードを解除したあと、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・モードの解除を参照してください）。

ウエイト時間は次のどちらかを選択することができます。

512/fcc (142  $\mu$ s : 3.6MHz動作時)

ウエイトなし

### ★ 9.4 ポート3のマスク・オプション

ポート3はマスク・オプションにより、リセット時の入力/出力モードの設定が可能です。

表9-1に示す3種類から選択することができます。

表9-1 ポート3のマスク・オプションによるリセット時の状態

端子名称	リセット時の状態		
	マスク・オプション	マスク・オプション	マスク・オプション
P30/PCL	入力	ロウ・レベル出力	ロウ・レベル出力
P31/BUZ			
P32			
P33			ハイ・レベル出力

〔メモ〕

## 第10章 命令セット

μPD753304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。

### 10.1 特徴的な命令

ここでは，μPD753304の命令セットのうち，特徴的な命令の概要を説明します。

#### 10.1.1 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 ) 4 Kバイト空間 ( 0000H-0FFFH ) のサブルーチン・コール命令
- ( b ) 4 Kバイト空間 ( 0000H-0FFFH ) 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。

### 10.1.2 ビット操作命令

μPD753304は、通常のビット操作命令（セット、クリア命令）に加え、ビット・テスト命令、ビット転送命令、およびビット・ブーリアン命令（AND, OR, XOR）が強化されています。

操作ビットの指定は、ビット操作アドレッシングにより行います。ビット操作アドレッシングには、3種類あり、各アドレッシングにより操作されるビットを表10 - 1 に示します。

表10 - 1 ビット操作アドレッシングの種類と指定可能な範囲

アドレッシング	操作可能な周辺ハードウェア	操作可能なビットのアドレッシング範囲
fmem. bit	RBE, MBE, IST1, IST0, SCC, IE x x x ,	FB0H-FBFH
	IRQ x x x	
	PORT3, 8, 10	FF0H-FFFH
pmem. @L	PORT3, 8, 10	FC0H-FFFH
@H + mem. bit	ビット操作可能なすべての周辺ハードウェア	MBで指定されるメモリ・バンクのビット操作可能なすべてのビット

備考 1 . x x x : 1, BT, T0, W

2 . MB = MBE · MBS

### 10.1.3 たてづみ命令

μPD753304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。



### 10.1.4 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μPD753304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。オーバーフローはキャリー・フラグに残ります。

ADDC A, @HL命令の実行の結果、キャリーが出ると続くADDS A, #n4命令をスキップします。またキャリーが出なければADDS A, #n4命令が実行されますが、このときこの命令のスキップ機能は禁止され、加算の結果キャリーが出ても続く命令をスキップしません。したがってADDS A, #n4命令に続けてプログラムを書くことができます。

例 アキュムレータと、メモリを10進加算する。

```
ADDS A, 6
ADDC A, @HL ; A, CY A + (HL) + CY
ADDS A, #10
    :
```

#### (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。

### 10.1.5 スキップ命令とスキップに要するマシン・サイクル数

μPD753304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

(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, CALL !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

### 10.1.6 表示データ・メモリ操作可能命令

表示データ・メモリの操作については、次の命令のみ使用できます。

```
MOV @HL, A
```

```
MOV mem, A
```

GETIによる上記2命令

## 10.2 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** (U12385J) 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、適当な数値またはラベルを記述します。

ラベルとしてmem, fmem, pmem, bitなどの代わりに、図3-7に書かれた各種レジスタ・フラグの略号を記述できます（ただし、fmem, pmemは記述できるラベルに制限があります。詳しくは、表3-1 **アドレッシング・モード**、図3-7 **μPD753304 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・データまたはラベル
n8	8ビット・イミディエト・データまたはラベル
mem	8ビット・イミディエト・データまたはラベル <sup>注</sup>
bit	2ビット・イミディエト・データまたはラベル
fmem	FB0H-FBFH, FF0H-FFFHイミディエト・データまたはラベル
pmem	FC0H-FFFHイミディエト・データまたはラベル
addr, addr1	000H-FFFHイミディエト・データまたはラベル
caddr	12ビット・イミディエト・データまたはラベル
faddr	11ビット・イミディエト・データまたはラベル
taddr	20H-7FHイミディエト・データ（ただしbit0=0）またはラベル
PORTn	PORT3, PORT8, PORT10
IE x x x	IEBT, IET0, IE1, IEW
RBn	RB0-RB3
MBn	MB0, MB1, MB15

注 mem1は、8ビット・データ処理の場合は偶数アドレスのみ記述できます。

## (2) オペレーション説明上の凡例

A	: Aレジスタ; 4ビット・アキュムレータ
B	: Bレジスタ
C	: Cレジスタ
D	: Dレジスタ
E	: Eレジスタ
H	: Hレジスタ
L	: Lレジスタ
X	: Xレジスタ
XA	: レジスタ・ペア (XA); 8ビット・アキュムレータ
BC	: レジスタ・ペア (BC)
DE	: レジスタ・ペア (DE)
HL	: レジスタ・ペア (HL)
XA'	: 拡張レジスタ・ペア (XA')
BC'	: 拡張レジスタ・ペア (BC')
DE'	: 拡張レジスタ・ペア (DE')
HL'	: 拡張レジスタ・ペア (HL')
PC	: プログラム・カウンタ
SP	: スタック・ポインタ
CY	: キャリー・フラグ; ビット・アキュムレータ
PSW	: プログラム・ステータス・ワード
MBE	: メモリ・バンク許可フラグ
RBE	: レジスタ・バンク許可フラグ
PORT <sub>n</sub>	: ポート <sub>n</sub> (n = 3, 8, 10)
IME	: 割り込みマスタ許可フラグ
IPS	: 割り込みプライオリティ選択レジスタ
IE <sub>xxx</sub>	: 割り込み許可フラグ
RBS	: レジスタ・バンク選択レジスタ
MBS	: メモリ・バンク選択レジスタ
PCC	: プロセッサ・クロック・コントロール・レジスタ
.	: アドレス, ビット区切り
(xx)	: xxでアドレスされる内容
xxH	: 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 ( 000H-07FH ) 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	addr = 000H-FFFH	プログラム・メモリ・アドレッシング
* 7	addr = ( Current PC ) - 15 ~ ( Current PC ) - 1 ( Current PC ) + 2 ~ ( Current PC ) + 16 addr1 = ( Current PC ) - 15 ~ ( Current PC ) - 1 ( Current PC ) + 2 ~ ( Current PC ) + 16	
* 8	caddr = 000H-FFFH	
* 9	faddr = 0000H-07FFH	
* 10	taddr = 0020H-007FH	
* 11	addr1 = 000H-FFFH	

備考 1 . MBはアクセス可能なメモリ・バンクを示します。

- 2 . \* 2 ではMBE, MBSに関係なくMB = 0です。
- 3 . \* 4 , \* 5 ではMBE, MBSに関係なくMB = 15です。
- 4 . \* 6 ~ \* 11は, それぞれアドレッシング可能な領域を示します。

**(4) マシン・サイクルの欄の説明**

Sは、スキップ付き命令がスキップ動作をするときに要するマシン・サイクル数を示します。Sの値は次のように変わります。

- ・スキップしないとき.....S = 0
- ・スキップされる命令が、1バイト命令、または2バイト命令のとき.....S = 1
- ・スキップされる命令が、3バイト命令<sup>注</sup>のとき..... S = 2

**注** 3バイト命令：BR !addr, BRA !addr1, CALL !addr, CALLA !addr1命令

**注意** GETI命令は1マシン・サイクルでスキップされます。

1マシン・サイクルはCPUクロック の1サイクル分 (=  $t_{CY}$ ) に等しく、PCCの設定により4通りの時間が選択できます(図5 - 13 プロセッサ・クロック・コントロール・レジスタのフォーマット参照)。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
転送	MOV	A, #n4	1	1	A n4		たてづみA
		reg1, #n4	2	2	reg1 n4		
		XA, #n8	2	2	XA n8		たてづみA
		HL, #n8	2	2	HL n8		たてづみB
		rp2, #n8	2	2	rp2 n8		
		A, @HL	1	1	A (HL)	* 1	
		A, @HL +	1	2 + S	A (HL), then L L + 1	* 1	L = 0
		A, @HL -	1	2 + S	A (HL), then L L - 1	* 1	L = FH
		A, @rpa1	1	1	A (rpa1)	* 2	
		XA, @HL	2	2	XA (HL)	* 1	
		@HL, A	1	1	(HL) A	* 1	
		@HL, XA	2	2	(HL) XA	* 1	
		A, mem	2	2	A (mem)	* 3	
		XA, mem	2	2	XA (mem)	* 3	
		mem, A	2	2	(mem) A	* 3	
		mem, XA	2	2	(mem) XA	* 3	
		A, reg	2	2	A reg		
		XA, rp'	2	2	XA rp'		
	reg1, A	2	2	reg1 A			
	rp' 1, XA	2	2	rp' 1 XA			
	XCH	A, @HL	1	1	A (HL)	* 1	
		A, @HL +	1	2 + S	A (HL), then L L + 1	* 1	L = 0
		A, @HL -	1	2 + S	A (HL), then L L - 1	* 1	L = FH
		A, @rpa1	1	1	A (rpa1)	* 2	
		XA, @HL	2	2	XA (HL)	* 1	
		A, mem	2	2	A (mem)	* 3	
		XA, mem	2	2	XA (mem)	* 3	
		A, reg1	1	1	A reg1		
XA, rp'	2	2	XA rp'				

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
テーブル参照	MOVT	XA, @PCDE	1	3	XA (PC <sub>11-8</sub> + DE) <sub>ROM</sub>		
		XA, @PCXA	1	3	XA (PC <sub>11-8</sub> + XA) <sub>ROM</sub>		
		XA, @BCDE	1	3	XA (BCDE) <sub>ROM</sub> <sup>注</sup>	* 6	
		XA, @BCXA	1	3	XA (BCXA) <sub>ROM</sub> <sup>注</sup>	* 6	
ビット転送	MOV1	CY, fmem. bit	2	2	CY (fmem. bit)	* 4	
		CY, pmem. @L	2	2	CY (pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> ))	* 5	
		CY, @H + mem. bit	2	2	CY (H + mem <sub>3-0</sub> . bit)	* 1	
		fmem. bit, CY	2	2	(fmem. bit) CY	* 4	
		pmem. @L, CY	2	2	(pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) CY	* 5	
		@H + mem. bit, CY	2	2	(H + mem <sub>3-0</sub> . 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		
	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			

注 Bレジスタには必ず“0”を設定してください。



命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドッシング・エリア	スキップ条件
操作	RORC	A	1	1	CY A <sub>0</sub> , A <sub>3</sub> CY, A <sub>n-1</sub> A <sub>n</sub>		
	NOT	A	2	2	A $\bar{A}$		
増減	INCS	reg	1	1 + S	reg reg + 1		reg = 0
		rp1	1	1 + S	rp1 rp1 + 1		rp1 = 00H
		@HL	2	2 + S	(HL) (HL) + 1	* 1	(HL) = 0
		mem	2	2 + S	(mem) (mem) + 1	* 3	(mem) = 0
	DECS	reg	1	1 + S	reg reg - 1		reg = FH
		rp'	2	2 + S	rp' rp' - 1		rp' = FFH
比較	SKE	reg, #n4	2	2 + S	Skip if reg = n4		reg = n4
		@HL, #n4	2	2 + S	Skip if (HL) = n4	* 1	(HL) = n4
		A, @HL	1	1 + S	Skip if A = (HL)	* 1	A = (HL)
		XA, @HL	2	2 + S	Skip if XA = (HL)	* 1	XA = (HL)
		A, reg	2	2 + S	Skip if A = reg		A = reg
		XA, rp'	2	2 + S	Skip if XA = rp'		XA = rp'
操作 キー 作リ ・ フラ グ	SET1	CY	1	1	CY 1		
	CLR1	CY	1	1	CY 0		
	SKT	CY	1	1 + S	Skip if CY = 1		CY = 1
	NOT1	CY	1	1	CY $\bar{CY}$		
メモ リ ・ ピ ツ ト 操 作	SET1	mem. bit	2	2	(mem. bit) 1	* 3	
		fmem. bit	2	2	(fmem. bit) 1	* 4	
		pmem. @L	2	2	(pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) 1	* 5	
		@H + mem. bit	2	2	(H + mem <sub>3-0</sub> . bit) 1	* 1	
	CLR1	mem. bit	2	2	(mem. bit) 0	* 3	
		fmem. bit	2	2	(fmem. bit) 0	* 4	
		pmem. @L	2	2	(pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) 0	* 5	
		@H + mem. bit	2	2	(H + mem <sub>3-0</sub> . bit) 0	* 1	
	SKT	mem. bit	2	2 + S	Skip 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 <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) = 1	* 5	(pmem. @L) = 1
		@H + mem. bit	2	2 + S	Skip if (H + mem <sub>3-0</sub> . 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 <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) = 0	* 5	(pmem. @L) = 0
		@H + mem. bit	2	2 + S	Skip if (H + mem <sub>3-0</sub> . 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 <sub>7-2</sub> + L <sub>3-2</sub> . bit (L <sub>1-0</sub> )) = 1 and clear	* 5	(pmem. @L) = 1
		@H + mem. bit	2	2 + S	Skip if (H + mem <sub>3-0</sub> . bit) = 1 and clear	* 1	(@H + mem. 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	アドッシング・エリア	スキップ条件	
メモリ・ビット操作	AND1	CY, fmem. bit	2	2	CY CY ( fmem. bit )	* 4		
		CY, pmem. @L	2	2	CY CY ( pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit ( L <sub>1-0</sub> ) )	* 5		
		CY, @H + mem. bit	2	2	CY CY ( H + mem <sub>3-0</sub> . bit )	* 1		
	OR1	CY, fmem. bit	2	2	CY CY ( fmem. bit )	* 4		
		CY, pmem. @L	2	2	CY CY ( pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit ( L <sub>1-0</sub> ) )	* 5		
		CY, @H + mem. bit	2	2	CY CY ( H + mem <sub>3-0</sub> . bit )	* 1		
	XOR1	CY, fmem. bit	2	2	CY CY <sup>-</sup> ( fmem. bit )	* 4		
		CY, pmem. @L	2	2	CY CY <sup>-</sup> ( pmem <sub>7-2</sub> + L <sub>3-2</sub> . bit ( L <sub>1-0</sub> ) )	* 5		
		CY, @H + mem. bit	2	2	CY CY <sup>-</sup> ( H + mem <sub>3-0</sub> . bit )	* 1		
分岐	BR <sup>注1</sup>	addr	-	-	PC <sub>11-0</sub> addr <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">                     アセンブラにより、次のうち最適な命令を選択します。                      ・ BR ! addr                      ・ BRCB ! caddr                      ・ BR \$addr                 </div>	* 6		
		addr1	-	-	PC <sub>11-0</sub> addr1 <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">                     アセンブラにより、次のうち最適な命令を選択します。                      ・ BR ! addr                      ・ BRA ! addr1                      ・ BRCB ! caddr                      ・ BR \$addr1                 </div>	* 11		
		! addr	3	3	PC <sub>11-0</sub> addr	* 6		
		\$addr	1	2	PC <sub>11-0</sub> addr	* 7		
		\$addr1	1	2	PC <sub>11-0</sub> addr1	* 7		
		PCDE	2	3	PC <sub>11-0</sub> PC <sub>11-8</sub> + DE			
		PCXA	2	3	PC <sub>11-0</sub> PC <sub>11-8</sub> + XA			
		BCDE	2	3	PC <sub>11-0</sub> BCDE <sup>注2</sup>	* 6		
		BCXA	2	3	PC <sub>11-0</sub> BCXA <sup>注2</sup>	* 6		
		BRA <sup>注1</sup>	! addr1	3	3	PC <sub>11-0</sub> addr1	* 11	
		BRCB	! caddr	2	2	PC <sub>11-0</sub> caddr <sub>11-0</sub>	* 8	

注1 .  で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

2 . Bレジスタには必ず“ 0 ”をいれてください。

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 <sup>注</sup>	! addr1	3	3	(SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> addr1, SP SP - 6	* 11	
	CALL <sup>注</sup>	! addr	3	3	(SP - 3) MBE, RBE, 0, 0 (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>11-0</sub> addr, SP SP - 4	* 6	
				4	(SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> addr, SP SP - 6		
	CALLF <sup>注</sup>	! faddr	2	2	(SP - 3) MBE, RBE, 0, 0 (SP - 4) (SP - 1) (SP - 2) PC <sub>11-0</sub> PC <sub>11-0</sub> 0 + faddr, SP SP - 4	* 9	
				3	(SP - 2) x, x, MBE, RBE (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 PC <sub>11-0</sub> 0 + faddr, SP SP - 6		
RET <sup>注</sup>		1	3	PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) MBE, RBE, 0, 0 (SP + 1), SP SP + 4 x, x, MBE, RBE (SP + 4) 0, 0, 0, 0 (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2), SP SP + 6			
RETS <sup>注</sup>		1	3 + S	MBE, RBE, 0, 0 (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) SP SP + 4 then skip unconditionally 0, 0, 0, 0 (SP + 1) PC <sub>11-0</sub> (SP) (SP + 3) (SP + 2) x, x, MBE, RBE (SP + 4) SP SP + 6 then skip unconditionally		無条件	

注 で示す部分はMk モード時のみ対応可能です。そのほかはMk モード時にのみ対応可能です。

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
サブルーチン・スタック制御	RETI <sup>注1</sup>		1	3	MBE, RBE, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) PSW (SP+4) (SP+5), SP SP+6 0, 0, 0, 0 (SP+1) PC <sub>11-0</sub> (SP) (SP+3) (SP+2) PSW (SP+4) (SP+5), SP SP+6		
	PUSH	rp	1	1	(SP-1) (SP-2) rp, SP SP-2		
		BS	2	2	(SP-1) MBS, (SP-2) RBS, SP SP-2		
	POP	rp	1	1	rp (SP+1) (SP), SP SP+2		
BS		2	2	MBS (SP+1), RBS (SP), SP SP+2			
割り込み制御	EI		2	2	IME (IPS.3) 1		
		IE x x x	2	2	IE x x x 1		
	DI		2	2	IME (IPS.3) 0		
		IE x x x	2	2	IE x x x 0		
入出力	IN <sup>注2</sup>	A, PORTn	2	2	A PORTn (n=3, 8, 10)		
	OUT <sup>注2</sup>	PORTn, A	2	2	PORTn A (n=3, 8, 10)		
CPU制御	HALT		2	2	Set HALT Mode (PCC.2 1)		
	STOP		2	2	Set STOP Mode (PCC.3 1)		
	NOP		1	1	No Operation		
特殊	SEL	RBn	2	2	RBS n (n=0-3)		
		MBn	2	2	MBS n (n=0, 1, 15)		
	GETI <sup>注3</sup>	taddr	1	3	・TBR命令のとき PC <sub>11-0</sub> (taddr) <sub>3-0</sub> + (taddr+1) ----- ・TCALL命令のとき (SP-4) (SP-1) (SP-2) PC <sub>11-0</sub> (SP-3) MBE, RBE, 0, 0 PC <sub>11-0</sub> (taddr) <sub>3-0</sub> + (taddr+1) SP SP-4 ----- ・TBR, TCALL命令以外のとき (taddr) (taddr+1)の命令実行	* 10	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 モード時にのみ対応可能です。

2 . 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。

3 . 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。

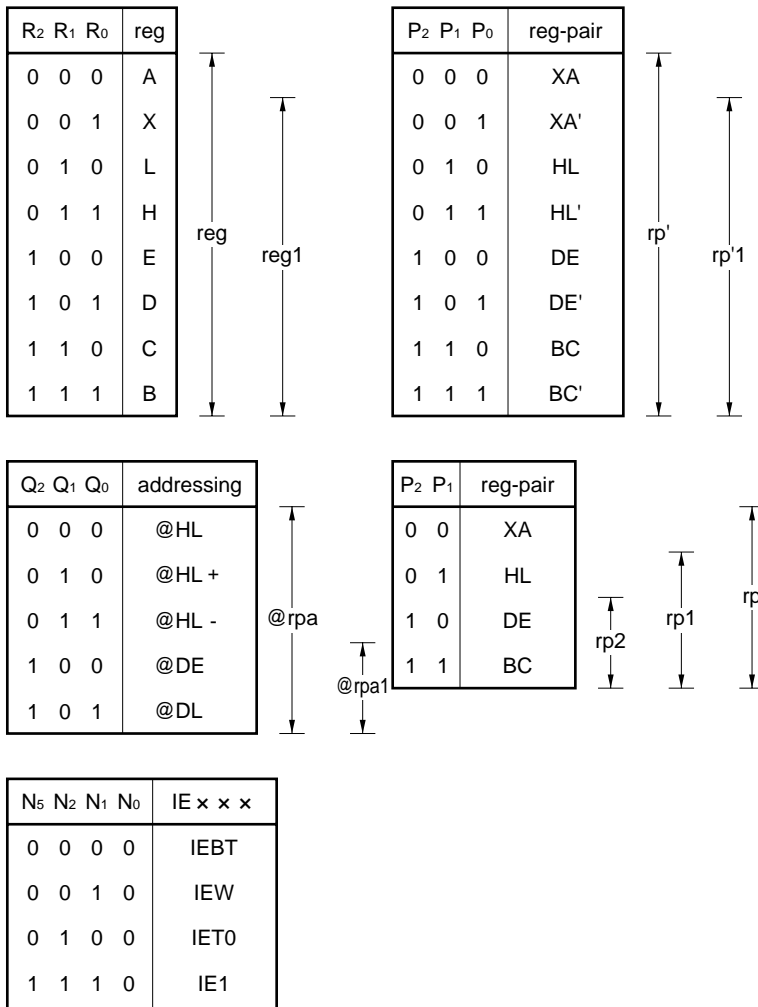
命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
特殊	GETI <sup>注1,2</sup>	taddr	1	3	・ TBR命令のとき PC <sub>11-0</sub> (taddr) <sub>3-0</sub> + (taddr + 1)	* 10	
				4	・ TCALL命令のとき (SP - 6) (SP - 3) (SP - 4) PC <sub>11-0</sub> (SP - 5) 0, 0, 0, 0 (SP - 2) x, x, MBE, RBE PC <sub>11-0</sub> (taddr) <sub>3-0</sub> + (taddr + 1) SP SP - 6		
				3	・ TBR, TCALL命令以外のとき (taddr) (taddr + 1) の命令実行		

注1 . TBR, TCALL命令はGETI命令のテーブル定義用アセンブラ疑似命令です。

2 . で示す部分はMk モード時のみ対応可能です。そのほかは、Mk モード時にのみ対応可能です。

### 10.3 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

$T_n$  :  $taddr \times \frac{1}{2}$ 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 <sub>1</sub> B <sub>0</sub> F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	FB0H-FBFHの操作可能ビット
	1 1 B <sub>1</sub> B <sub>0</sub> F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	FF0H-FFFHの操作可能ビット
pmem. @L	0 1 0 0 G <sub>3</sub> G <sub>2</sub> G <sub>1</sub> G <sub>0</sub>	FC0H-FFFHの操作可能ビット
@H + mem. bit	0 0 B <sub>1</sub> B <sub>0</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	アクセス可能なメモリ・バンクの操作可能ビット

B<sub>n</sub> : bitに対するイミューディエト・データ

F<sub>n</sub> : 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<sub>n</sub> : 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<sub>n</sub> : 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 <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
転送	MOV	A, #n4	0 1 1 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>		
		reg1, #n4	1 0 0 1 1 0 1 0	I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		rp, #n8	1 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 1	I <sub>7</sub> I <sub>6</sub> I <sub>5</sub> I <sub>4</sub> I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	
		A, @rpa1	1 1 1 0 0 Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 1 0 0 0	
		@HL, A	1 1 1 0 1 0 0 0		
		@HL, XA	1 0 1 0 1 0 1 0	0 0 0 1 0 0 0 0	
		A, mem	1 0 1 0 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		XA, mem	1 0 1 0 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		mem, A	1 0 0 1 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		mem, XA	1 0 0 1 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		A, reg	1 0 0 1 1 0 0 1	0 1 1 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		reg1, A	1 0 0 1 1 0 0 1	0 1 1 1 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	0 1 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	XCH	A, @rpa1	1 1 1 0 1 Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 0 0 0 1	
		A, mem	1 0 1 1 0 0 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		XA, mem	1 0 1 1 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> 0	
		A, reg1	1 1 0 1 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
XA, rp'		1 0 1 0 1 0 1 0	0 1 0 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
テーブル参照	MOV <sub>T</sub>	XA, @PCDE	1 1 0 1 0 1 0 0		
		XA, @PCXA	1 1 0 1 0 0 0 0		
		XA, @BCDE	1 1 0 1 0 1 0 1		
		XA, @BCXA	1 1 0 1 0 0 0 1		
転ビット	MOV <sub>1</sub>	CY, *1	1 0 1 1 1 1 0 1	* 2	
		*1, CY	1 0 0 1 1 0 1 1	* 2	



命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
演算	ADDS	A, #n4	0 1 1 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>		
		XA, #n8	1 0 1 1 1 0 0 1	l <sub>7</sub> l <sub>6</sub> l <sub>5</sub> l <sub>4</sub> l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 1 0 1 0 0 1 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 0 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 0 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	ADDC	A, @HL	1 0 1 0 1 0 0 1		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	SUBS	A, @HL	1 0 1 0 1 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 1 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	SUBC	A, @HL	1 0 1 1 1 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 1 1 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 1 1 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	AND	A, #n4	1 0 0 1 1 0 0 1	0 0 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 0 1 0 0 0 0		
		XA, rp'	1 0 1 0 1 0 1 0	1 0 0 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
		rp'1, XA	1 0 1 0 1 0 1 0	1 0 0 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	OR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 1 0 0 0 0 0		
XA, rp'		1 0 1 0 1 0 1 0	1 0 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
rp'1, XA		1 0 1 0 1 0 1 0	1 0 1 0 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
XOR	A, #n4	1 0 0 1 1 0 0 1	0 1 0 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>		
	A, @HL	1 0 1 1 0 0 0 0			
	XA, rp'	1 0 1 0 1 0 1 0	1 0 1 1 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
	rp'1, XA	1 0 1 0 1 0 1 0	1 0 1 1 0 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
アキュムレータ操作	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 <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
増減	INCS	reg	1 1 0 0 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
		rp1	1 0 0 0 1 P <sub>2</sub> P <sub>1</sub> 0		
		@HL	1 0 0 1 1 0 0 1	0 0 0 0 0 0 1 0	
		mem	1 0 0 0 0 0 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
	DECS	reg	1 1 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>		
		rp'	1 0 1 0 1 0 1 0	0 1 1 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
比較	SKE	reg, #n4	1 0 0 1 1 0 1 0	l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub> 0 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		@HL, #n4	1 0 0 1 1 0 0 1	0 1 1 0 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	
		A, @HL	1 0 0 0 0 0 0 0		
		XA, @HL	1 0 1 0 1 0 1 0	0 0 0 1 1 0 0 1	
		A, reg	1 0 0 1 1 0 0 1	0 0 0 0 1 R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	
		XA, rp'	1 0 1 0 1 0 1 0	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
キャリー・フラグ操作	SET1	CY	1 1 1 0 0 1 1 1		
	CLR1	CY	1 1 1 0 0 1 1 0		
	SKT	CY	1 1 0 1 0 1 1 1		
	NOT1	CY	1 1 0 1 0 1 1 0		
メモリ・ビット操作	SET1	mem. bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 0 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		*1	1 0 0 1 1 1 0 1	* 2	
	CLR1	mem. bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 0 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		*1	1 0 0 1 1 1 0 0	* 2	
	SKT	mem. bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 1 1	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		*1	1 0 1 1 1 1 1 1	* 2	
	SKF	mem. bit	1 0 B <sub>1</sub> B <sub>0</sub> 0 1 1 0	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
		*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 <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	
★ 分岐	BR	! addr	1 0 1 0 1 0 1 1	0 0 _____	addr _____	
		\$ addr	( + <sub>16</sub> )	0 0 0 0 A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>		
			{ + <sub>2</sub> } { - <sub>1</sub> }	1 1 1 1 S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>		
			( - <sub>15</sub> )			
		PCDE	1 0 0 1 1 0 0 1	0 0 0 0 0 1 0 0		
		PCXA	1 0 0 1 1 0 0 1	0 0 0 0 0 0 0 0		
	BCDE	1 0 0 1 1 0 0 1	0 0 0 0 0 1 0 1			
BCXA	1 0 0 1 1 0 0 1	0 0 0 0 0 0 0 1				
	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	CALLA	! addr1	1 0 1 1 1 0 1 1	0 _____	addr _____	
	CALL	! addr	1 0 1 0 1 0 1 1	0 1 _____	addr _____	
	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 <sub>2</sub> P <sub>1</sub> 1		
		BS		1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 1	
POP	rp		0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 0			
	BS		1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 0		
入出力	IN	A, PORT <sub>n</sub>	1 0 1 0 0 0 1 1	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>		
	OUT	PORT <sub>n</sub> , A	1 0 0 1 0 0 1 1	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>		
割 り 込 み 制 御	EI		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 0		
		IE x x x	1 0 0 1 1 1 0 1	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>		
	DI		1 0 0 1 1 1 0 0	1 0 1 1 0 0 1 0		
		IE x x x	1 0 0 1 1 1 0 0	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>		
C P U 制 御	HALT		1 0 0 1 1 1 0 1	1 0 1 0 0 0 1 1		
	STOP		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 1		
	NOP		0 1 1 0 0 0 0 0			
特 殊	SEL	RB <sub>n</sub>	1 0 0 1 1 0 0 1	0 0 1 0 0 0 N <sub>1</sub> N <sub>0</sub>		
		MB <sub>n</sub>	1 0 0 1 1 0 0 1	0 0 0 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>		
	GETI	taddr	0 0 T <sub>5</sub> T <sub>4</sub> T <sub>3</sub> T <sub>2</sub> T <sub>1</sub> T <sub>0</sub>			

## 10.4 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。μPD753304のMk モードと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。  
μPD753304のMk モード時とMk モード時
- μPD753304のMk モード時のみ使用できます。
- μPD753304のMk モード時のみ使用できます。
- / μPD753304のMk モード時とMk モード時に共通して使用することができますが、Mk モードとMk モードでは、機能面で異なる箇所があります。  
Mk モードの場合は【Mk モード】、Mk モードの場合は【Mk モード】に示す説明書きを、それぞれの場合に応じて、読み分けてください。

### 10.4.1 転送命令

#### ○ MOV A, #n4

機能 : A n4 n4 = l3-0 : 0-FH

4ビット・イミディエト・データn4をAレジスタ(4ビット・アキュムレータ)に転送します。たてづみ効果(グループA)があり、同じ命令か、またはMOV XA, #n8命令が続けて置かれると実行した命令以降のたてづみ命令はNOPとして処理されます。

#### 応用例

(1) アキュムレータに0BHをセットする。

```
MOV A, #0BH
```

(2) ポート3に出力するデータを0-2から選ぶ。

```
A0 : MOV A, #0
```

```
A1 : MOV A, #1
```

```
A2 : MOV A, #2
```

```
OUT PORT3, A
```

#### ○ MOV reg1, #n4

機能 : reg1 n4 n4 = l3-0 0-FH

4ビット・イミディエト・データn4をAレジスタreg1(X, H, L, D, E, B, C)に転送します。

#### ○ MOV XA, #n8

機能 : XA n8 n8 = l7-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアXAに転送します。たてづみ効果があり、同じ命令か、またはMOV A, #n4命令が続けて置かれると実行した命令以降のたてづみ命令は、NOPとして処理されます。

## ○ MOV HL, #n8

機能 : HL n8 n8 = 17-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアHLに転送します。たてづみ効果があり、同じ命令が続けて置かれると実行した命令以降のたてづみ命令は、NOPとして処理されます。

## ○ MOV rp2, #n8

機能 : rp2 n8 n8 = 17-0 : 00H-FFH

8ビット・イミディエト・データn8をレジスタ・ペアrp2 (BC, DE) に転送します。

## ○ MOV A, @HL

## ○ 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 (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 (HL), X (HL+1)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換し、Xレジスタの内容とメモリの次の番地の内容を交換します。

ただし、Lレジスタの内容が奇数の場合は、最下位ビットを無視したアドレスが指定されます。

## ○ XCH A, mem

**機能** : A (mem) mem = D7-0 : 00H-FEH

Aレジスタの内容と8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容を交換します。

## ○ XCH XA, mem

**機能** : A (mem), X (mem + 1) mem = D7-0 : 00H-FEH

Aレジスタの内容と8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容を交換し、Xレジスタの内容とメモリの次の番地の内容を交換します。

memで指定できるアドレスは偶数アドレスです。

## ○ XCH A, reg1

**機能** : A reg1

Aレジスタの内容とレジスタreg1 (X, H, L, D, E, B, C) の内容を交換します。

## ○ XCH XA, rp'

**機能** : XA rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容を交換します。

### 10.4.2 テーブル参照命令

## ○ MOVT XA, @PCDE

機能：XA ROM (PC<sub>11-8</sub> + DE)

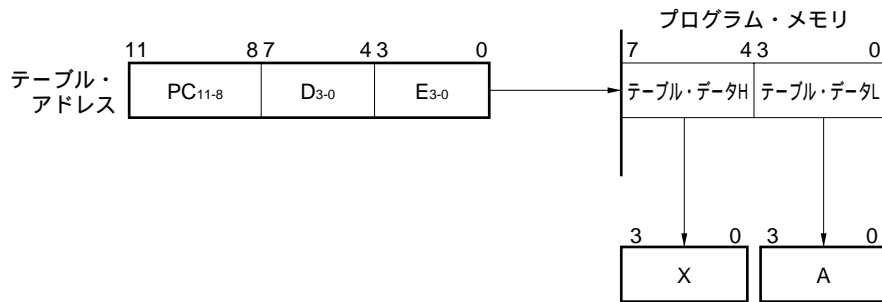
プログラム・カウンタ (PC) の下位 8 ビット (PC<sub>7-0</sub>) をレジスタ・ペア DE の内容で置き換えたときにアドレスされるプログラム・メモリ内のテーブル・データの下位 4 ビットを A レジスタに、上位 4 ビットを X レジスタに転送します。

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

テーブル領域には、必要なデータをあらかじめアセンブラ疑似命令 (DB 命令) によりプログラムしておく必要があります。

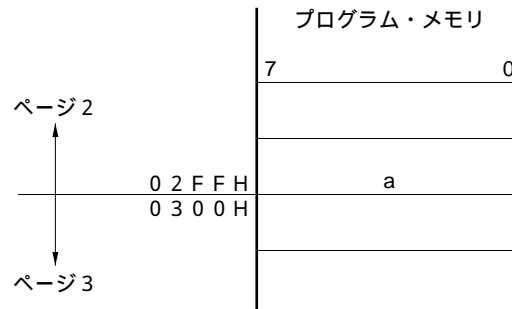
プログラム・カウンタはこの命令の実行により影響を受けません。

この命令は、テーブル・データを連続して参照する場合に有効です。



**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
    
```

## ○ MOVT XA, @PCXA

**機 能** : XA ROM (PC<sub>11-8</sub> + XA)

プログラム・カウンタ (PC) の下位 8 ビット (PC<sub>7-0</sub>) をレジスタ・ペアXAの内容で置き換えたときアドレスされるプログラム・メモリ内のテーブル・データの下位 4 ビットをAレジスタに, 上位 4 ビットをXレジスタに転送します。

テーブル・アドレスはこの命令実行時のPCの内容によって決まります。

テーブル領域には, 必要なデータをあらかじめアセンブラ疑似命令 (DB命令) によりプログラムしておく必要があります。

PCは, この命令の実行により影響を受けません。

### 注 意

MOV<sub>T</sub> XA, @PCDEと同様に, 命令が  $\times \times$  FFH番地にある場合は, 次のページのテーブル・データが転送されます。

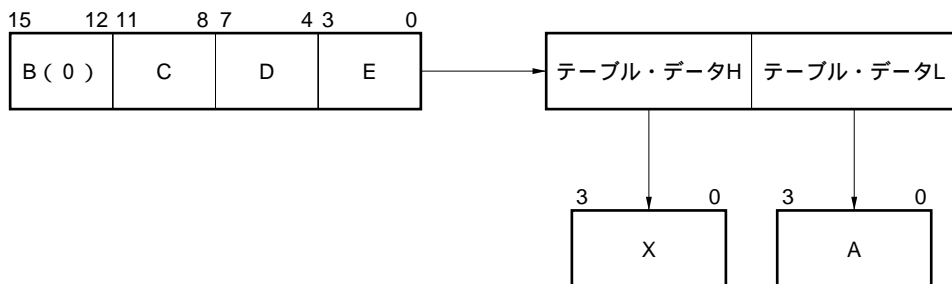
## ○ MOVT XA, @BCDE

**機能:** XA ROM (BCDE)

レジスタB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。

ただし,  $\mu$ PD753304では, レジスタBは無効です。レジスタBには0000Bを設定してください。

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。



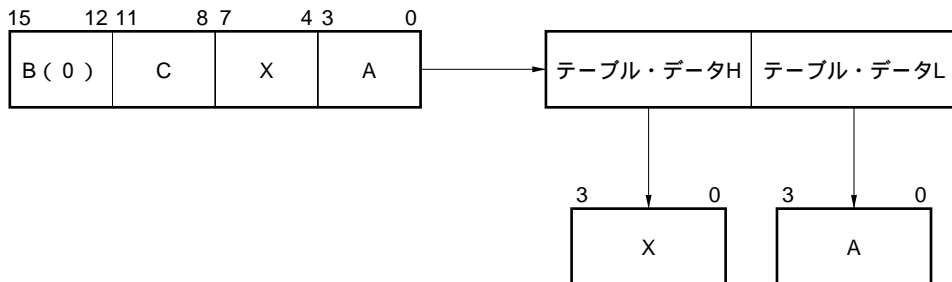
## ○ MOVT XA, @BCXA

**機能:** XA ROM (BCXA)

レジスタB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。

ただし,  $\mu$ PD753304では, レジスタBは無効です。レジスタBには0000Bを設定してください。

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。



## 10.4.3 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
```



## 10.4.4 演算命令

## ○ ADDS A, #n4

**機能** : A ← A + n4 ; Skip if carry. n4 = l3-0 : 0-FH

Aレジスタの内容に4ビット・イミディエト・データn4を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

ADDC A, @HL命令およびSUBC A, @HL命令と組み合わせると、進数補正命令となります(10.1.4 進数補正命令を参照)。

## ○ ADDS XA, #n8

**機能** : XA ← XA + n8 ; Skip if carry. n8 = l7-0 : 00H-FFH

レジスタ・ペアXAの内容に8ビット・イミディエト・データn8を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS A, @HL

**機能** : A ← A + (HL) ; Skip if carry.

Aレジスタの内容にレジスタ・ペアHLでアドレスされるデータ・メモリの内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS XA, rp'

**機能** : XA ← XA + rp' ; Skip if carry.

レジスタ・ペアXAの内容にレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC')の内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

## ○ ADDS rp'1, XA

**機能** :  $rp'1$   $rp'1 + XA$  ; Skip if carry.

レジスタ・ペア $rp'1$  (HL, DE, BC, XA', HL', DE', BC') の内容にレジスタ・ペアXAの内容を2進加算し、加算の結果キャリーが出たときは続く1命令をスキップします。キャリー・フラグは影響を受けません。

### 応用例

レジスタ・ペアを左シフトする。

```
MOV  XA, rp'1
ADDS rp'1, XA
NOP
```

## ○ ADDC A, @HL

**機能** : A, CY  $A + (HL) + CY$

Aレジスタの内容にレジスタ・ペアHLでアドレスされるデータ・メモリの内容をキャリー・フラグも含めて2進加算します。加算の結果キャリーが出たらキャリー・フラグがセットされ、キャリーが出ないときはキャリー・フラグはリセットされます。

この命令の次にADDS A, #n4命令が置かれた場合、この命令にキャリーが発生するとADDS A, #n4命令をスキップします。キャリーが発生しないときはADDS A, #n4命令を実行し、ADDS A, #n4命令のスキップ機能を禁止するという機能が生じます。したがって、これらの命令の組み合わせを進数補正に利用できます (10.1.4 進数補正命令を参照)。

## ○ ADDC XA, rp'

**機能** : XA, CY  $XA + rp' + CY$

レジスタ・ペアXAの内容にレジスタ・ペア $rp'$  (XA, HL, DE, BC, XA', HL', DE', BC') の内容をキャリー・フラグも含めて2進加算します。加算の結果キャリーが出たとき、キャリー・フラグはセットされ、キャリーが出ないとき、キャリー・フラグはリセットされます。

## ○ ADDC rp'1, XA

**機能** : rp'1, CY  $rp'1 + XA + CY$

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容にレジスタ・ペアXAの内容をキャリー・フラグを含めて2進加算します。加算の結果キャリーが出たらキャリー・フラグがセットされます。キャリーが出ないときはキャリー・フラグはリセットされます。

## ○ SUBS A, @HL

**機能** : A  $A - (HL)$  ; Skip if borrow

Aレジスタの内容からレジスタ・ペアHLでアドレスされるデータ・メモリの内容を減算し、結果をAレジスタにセットします。減算の結果ボローが出ると続く1命令をスキップします。

キャリー・フラグには影響を与えません。

## ○ SUBS XA, rp'

**機能** : XA  $XA - rp'$  ; Skip if borrow

レジスタ・ペアXAの内容からレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容を減算し、結果をレジスタ・ペアXAにセットします。減算の結果ボローが出ると続く1命令をスキップします。

キャリー・フラグには影響を与えません。

## 応用例

データ・メモリとレジスタ・ペアの大小比較をする。

```
MOV  XA, mem
SUBS XA, rp'
      ; (mem) > rp'
      ; (mem) < rp'
```

## ○ SUBS rp'1, XA

機能 :  $rp'1 \leftarrow rp'1 + XA$  ; 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 :  $A, CY \leftarrow A - (HL) - CY$

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 (10.1.4 進数補正命令を参照)。

## ○ SUBC XA, rp'

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

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せん。

## ○ AND A, #n4

**機能** : A A n4 n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミューディエイト・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 (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容との論理積をとり、結果をAレジスタにセットします。

## ○ AND XA, rp'

**機能** : XA XA rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' (XA, HL, DE, BC, XA', HL', DE', BC') の内容との論理積をとり、結果をレジスタ・ペアXAにセットします。

## ○ AND rp'1, XA

機能 : rp'1 rp'1 XA

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容とレジスタ・ペアXAの内容との論理積をとり、結果を指定されたレジスタ・ペアにセットします。

## ○ OR A, #n4

機能 : A A n4 n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミディエト・データn4との論理和をとり、結果をAレジスタにセットします。

### 応用例

アキュムレータの下位3ビットを1にする。

```
OR A, #0111B
```

## ○ OR A, @HL

機能 : A A (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容と論理和をとり、結果をAレジスタにセットします。

## ○ OR XA, rp'

機能 : XA XA rp'

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

## ○ OR rp'1, XA

機能: rp'1 rp'1 XA

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

## ○ XOR A, #n4

機能: A A ← n4 n4 = l3-0 : 0-FH

Aレジスタの内容と4ビット・イミューディエト・データn4との排他的論理和をとり、結果をAレジスタにセットします。

## 応用例

アキュムレータの最上位4ビットを反転する。

```
XOR A, #1000B
```

## ○ XOR A, @HL

機能: A A ← (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 XA ← rp'

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

 XOR rp'1, XA

機能 : rp'1 ← rp'1 ⊕ XA

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

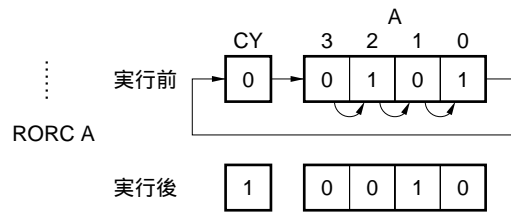


## 10.4.5 アキュムレータ操作命令

## ○ RORC A

機能: CY A<sub>0</sub>, A<sub>n-1</sub> A<sub>n</sub>, A<sub>3</sub> CY (n = 1-3)

Aレジスタ(4ビット・アキュムレータ)の内容をキャリー・フラグも含めて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)。

### 10.4.6 増減命令

#### ○ INCS reg

機能 : reg reg + 1 ; Skip if reg = 0

レジスタreg (X, A, H, L, D, E, B, C)の内容をインクリメントします。インクリメントの結果reg = 0になると続く1命令をスキップします。

#### ○ INCS rp1

機能 : rp1 rp1 + 1 ; Skip if rp1 = 00H

レジスタ・ペアrp1 (HL, DE, BC)の内容をインクリメントします。インクリメントの結果rp1 = 00Hになると続く1命令をスキップします。

#### ○ INCS @HL

機能 : (HL) (HL) + 1 ; Skip if (HL) = 0

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をインクリメントします。インクリメントの結果そのデータ・メモリの内容が0になると続く1命令をスキップします。

#### ○ INCS mem

機能 : (mem) (mem) + 1 ; Skip if (mem) = 0, mem = D7-0 : 00H-FFH

8ビット・イミディエイト・データmemでアドレスされるデータ・メモリの内容をインクリメントします。インクリメントの結果、そのデータ・メモリの内容が0になると続く1命令をスキップします。

## ○ DECS reg

**機能** : reg = reg - 1 ; Skip if reg = FFH

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 = FFH 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。

## 10.4.7 比較命令

## ○ SKE reg, #n4

機能 : Skip if reg = n4    n4 = I3-0 : 0-FH

レジスタreg (X, A, H, L, D, E, B, C) の内容と4ビット・イミディエト・データn4が等しければ, 続く1命令をスキップします。

## ○ SKE @HL, #n4

機能 : Skip if (HL) = n4    n4 = I3-0 : 0-FH

レジスタ・ペアHLでアドレスされるデータ・メモリの内容と4ビット・イミディエト・データn4が等しければ, 続く1命令をスキップします。

## ○ SKE A, @HL

機能 : Skip if A = (HL)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容が等しければ, 続く1命令をスキップします。

## ○ SKE XA, @HL

機能 : Skip if A = (HL) and X = (HL + 1)

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容が等しく, かつXレジスタの内容とメモリの次の番地の内容が等しければ, 続く1命令をスキップします。

ただし, Lレジスタの内容が奇数の場合, 最下位ビットを無視したアドレスが指定されます。

## ○ SKE A, reg

**機能** : Skip if A = reg

Aレジスタの内容とレジスタreg ( X, A, H, L, D, E, B, C ) の内容が等しければ, 続く 1 命令をスキップします。

## ○ SKE XA, rp'

**機能** : Skip if XA = rp'

レジスタ・ペアXAの内容とレジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容が等しければ, 続く 1 命令をスキップします。

## 10.4.8 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。

## 10.4.9 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 ← (オペランドで指定されるビット)

キャリア・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、結果をキャリア・フラグにセットします。

## 10.4.10 分岐命令

## ○ BR addr

機能 : PC<sub>11-0</sub> addr  
addr = 0000H-0FFFH

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 laddr命令、BRCB lcaddr命令、BR \$ addr命令の中から最適な命令に自動的に置き換えられます。

## ○ BR addr1

機能 : PC<sub>11-0</sub> addr1  
addr1 = 0000H-0FFFH

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。

## ○ BRA !addr1

機能 : PC<sub>11-0</sub> addr1

## ○ BR !addr

機能 : PC<sub>11-0</sub> addr  
addr = 0000H-0FFFH

イミディエト・データaddrがプログラム・カウンタ (PC) へ転送され、PCでアドレスされる番地に分岐します。

 BR \$ addr

機能: PC<sub>11-0</sub> addr

addr = (PC - 15) ~ (PC - 1), (PC + 2) ~ (PC + 16)

現在のアドレスから (-15 ~ -1), (+2 ~ +16) の分岐範囲を持つ相対分岐命令です。ページの境界や、ブロックの境界には影響を受けません。

 BR \$ addr1

機能: PC<sub>11-0</sub> addr1

addr1 = (PC - 15) ~ (PC - 1), (PC + 2) ~ (PC + 16)

現在のアドレスから (-15 ~ -1), (+2 ~ +16) の分岐範囲を持つ相対分岐命令です。ページの境界や、ブロックの境界には影響を受けません。

## ○ BRCB ! caddr

機能: PC<sub>11-0</sub> caddr<sub>11-0</sub>  
caddr = 0000H-0FFFH

プログラム・カウンタの下位12ビット (PC<sub>11-0</sub>) を12ビット・イミディエト・データcaddrで置き換えた番地に分岐します。

### 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、0FFEH番地、または0FFFH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 (非実装) に分岐します。0FFE H番地ではBRCB ! caddr命令を使用しないでください。

## ○ BR PCDE

機能：PC<sub>11-0</sub> PC<sub>11-8</sub> + DE  
PC<sub>7-4</sub> D, PC<sub>3-0</sub> E

プログラム・カウンタの下位8ビット(PC<sub>7-0</sub>)をレジスタ・ペアDEの内容で置き換えた番地に分岐します。プログラム・カウンタの上位のビットには影響を与えません。

### 注 意

BR PCDE命令は、通常、その命令のあるページ内に分岐しますが、命令コードの1バイト目がx×FE番地、またはx×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：PC<sub>11-0</sub> PC<sub>11-8</sub> + XA  
PC<sub>7-4</sub> X, PC<sub>3-0</sub> A

プログラム・カウンタの下位8ビット(PC<sub>7-0</sub>)をレジスタ・ペアXAの内容で置き換えた番地に分岐します。プログラム・カウンタの上位のビットには影響を与えません。

### 注 意

BR PCDE命令と同様に、1バイト目が0×FEH、または0×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。

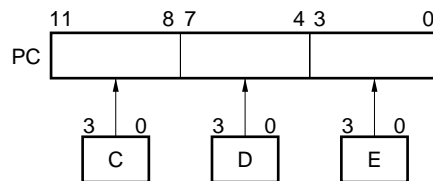
## ○ BR BCDE

機能：PC<sub>11-0</sub> BCDE

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

ただし，μPD753304のPCは12ビットです。PCには，レジスタC, D, Eの内容が置き換えられます。

レジスタBには，必ず0000Bを設定してください。



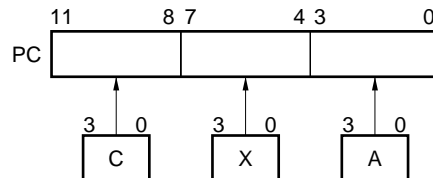
## ○ BR BCXA

機能：PC<sub>11-0</sub> BCXA

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

ただし，μPD753304のPCは12ビットです。PCには，レジスタC, X, Aの内容が置き換えられます。

レジスタBには，必ず0000Bを設定してください。



## ○ TBR addr

機能：

GETI命令のテーブル定義用アセンブラ疑似命令です。3バイトのBR !addr命令を，1バイトのGETI命令で置き換える場合に使用します。addrには12ビット・アドレス・データを記述します。詳しくは，RA75X **アセンブラ・パッケージ ユーザーズ・マニュアル 言語編** (U12385J)を参照してください。

## 10.4.11 サブルーチン・スタック制御命令

## ○ CALLA ! addr1

機能：(SP - 2) x, x, MBE, RBE, (SP - 3) PC<sub>7-4</sub>  
 (SP - 4) PC<sub>3-0</sub>, (SP - 5) 0, 0, 0, 0  
 (SP - 6) PC<sub>11-8</sub>  
 PC<sub>11-0</sub> addr1, SP SP - 6

## / ○ CALL ! addr

機能：【Mk モード】  
 (SP - 1) PC<sub>7-4</sub>, (SP - 2) PC<sub>3-0</sub>  
 (SP - 3) MBE, RBE, 0, 0  
 (SP - 4) PC<sub>11-8</sub>, PC<sub>11-0</sub> addr, SP SP - 4

addr = 0000H-0FFFH

【Mk モード】

(SP - 2) x, x, MBE, RBE  
 (SP - 3) PC<sub>7-4</sub>, (SP - 4) PC<sub>3-0</sub>  
 (SP - 5) 0, 0, 0, 0, (SP - 6) PC<sub>11-8</sub>  
 PC<sub>11-0</sub> addr, SP SP - 6

addr = 0000H-0FFFH

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, 12ビット・イミューディアト・データaddrでアドレスされる番地へ分岐します。



## ○ / CALLF ! faddr

### 機能：【Mk モード】

( SP - 1 ) PC<sub>7-4</sub>, ( SP - 2 ) PC<sub>3-0</sub>

( SP - 3 ) MBE, RBE, 0, 0

( SP - 4 ) PC<sub>11-8</sub>, SP SP - 4

PC<sub>11-0</sub> 0 + faddr

faddr = 0000H-07FFH

### 【Mk モード】

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

( SP - 3 ) PC<sub>7-4</sub>, ( SP - 4 ) PC<sub>3-0</sub>

( SP - 5 ) 0, 0, 0, 0, ( SP - 6 ) PC<sub>11-8</sub>

SP SP - 6

PC<sub>11-0</sub> 0 + faddr

faddr = 0000H-07FFH

プログラム・カウンタ (PC; リターン・アドレス) および MBE, RBE の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, SP をデクリメントしたのち, 11 ビット・イミディエト・データ faddr でアドレスされる番地へ分岐します。コールできる範囲は 0000H-07FFH (0-2047) 番地に限られます。

## ○ TCALL ! addr

### 機能

GETI 命令のテーブ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 (U12385J)** を参照してください。

## / RET

**機能:** 【Mk モード】 PC<sub>11-8</sub> (SP), MBE, RBE, 0, 0 (SP+1)  
 PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3), SP SP+4  
 【Mk モード】 PC<sub>11-8</sub> (SP), 0, 0, 0, 0 (SP+1)  
 PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3)  
 x, x, MBE, RBE (SP+4)  
 SP SP+6

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をプログラム・カウンタ (PC), メモリ・バンク許可フラグ (MBE), およびレジスタ・バンク許可フラグ (RBE) にリストアし, そのあとSPの内容をインクリメントします。

**注 意**

プログラム・ステータス・ワード (PSW) は, MBE, RBE以外リストアされません。

## / RETS

**機能:** 【Mk モード】 PC<sub>11-8</sub> (SP), MBE, RBE, 0, 0 (SP+1)  
 PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3), SP SP+4  
 Then skip unconditionally  
 【Mk モード】 PC<sub>11-8</sub> (SP), 0, 0, 0, 0 (SP+1)  
 PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3)  
 x, x, MBE, RBE (SP+4), SP SP+6  
 Then skip unconditionally

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をプログラム・カウンタ (PC), メモリ・バンク許可フラグ (MBE), およびレジスタ・バンク許可フラグ (RBE) にリストアし, SPの内容をインクリメント後, 無条件にスキップします。

**注 意**

プログラム・ステータス・ワード (PSW) は, MBE, RBE以外リストアされません。

## / RETI

**機能:** 【Mk モード】 PC<sub>11-8</sub> (SP), MBE, RBE, 0, 0 (SP+1),  
PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3)  
PSW<sub>L</sub> (SP+4), PSW<sub>H</sub> (SP+5)  
SP SP+6

【Mk モード】 PC<sub>11-8</sub> (SP), 0, 0, 0, 0 (SP+1),  
PC<sub>3-0</sub> (SP+2), PC<sub>7-4</sub> (SP+3)  
PSW<sub>L</sub> (SP+4), PSW<sub>H</sub> (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。

## ○ PUSH rp

**機能:** (SP - 1) r<sub>pH</sub>, (SP - 2) r<sub>pL</sub>, SP SP - 2

レジスタ・ペアrp (XA, HL, DE, BC) の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, そのあとSPをデクリメントします。

レジスタ・ペアの上位側 (r<sub>pH</sub>: X, H, D, B) は (SP - 1) で, 下位側 (r<sub>pL</sub>: A, L, E, C) は (SP - 2) でアドレスされるスタックに退避されます。

## ○ PUSH BS

**機能:** (SP - 1) MBS, (SP - 2) RBS, SP SP - 2

メモリ・バンク選択レジスタ (MBS), およびレジスタ・バンク選択レジスタ (RBS) の内容をスタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) に退避し, そのあとSPをデクリメントします。

## ○ POP rp

**機能:** r<sub>pL</sub> (SP), r<sub>pH</sub> (SP + 1), SP SP + 2

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をレジスタ・ペアrp (XA, HL, DE, BC) へ復帰し, そのあとSPをインクリメントします。

レジスタ・ペアの下位側 (r<sub>pL</sub>: A, L, E, C) には (SP) の内容が, 上位側 (r<sub>pH</sub>: X, H, D, B) には (SP + 1) の内容がリストアされます。

## ○ POP BS

**機能:** RBS (SP), MBS (SP + 1), SP SP + 2

スタック・ポインタ (SP) でアドレスされるデータ・メモリ (スタック) の内容をレジスタ・バンク選択レジスタ (RBS), メモリ・バンク選択レジスタ (MBS) へ復帰し, そのあとSPをインクリメントします。

## 10.4.12 割り込み制御命令

EI

機能：IME (IPS.3) 1

割り込みマスタ許可フラグ (割り込みプライオリティ選択レジスタのビット3) を, “ 1 ” にセットし, 割り込みを許可します。割り込み受け付けの可否は, 各割り込み許可フラグにより制御されます。

EI IE × × ×

機能：IE × × × 1     × × × = N<sub>5</sub>, N<sub>2-0</sub>

割り込み許可フラグ (IE × × ×) を “ 1 ” にセットし, 割り込みを受け付け可能状態にします。  
( × × × = BT, T0, W, 1 )

DI

機能：IME (IPS.3) 0

割り込みマスタ許可フラグ (割り込みプライオリティ選択レジスタのビット3) を, “ 0 ” にリセットし, 各割り込み許可フラグの内容に関係なく, すべての割り込みを禁止します。

DI IE × × ×

機能：IE × × × 0     × × × = N<sub>5</sub>, N<sub>2-0</sub>

割り込み許可フラグ (IE × × ×) を “ 0 ” にリセットし, 割り込みの受け付けを禁止します。  
( × × × = BT, T0, W, 1 )

### 10.4.13 入出力命令

#### ○ IN A, PORTn

機能 : A PORTn  $n = N_{3-0} : 3, 8, 10$

PORTn ( $n = 3, 8, 10$ ) 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3, 8, 10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。

#### ○ OUT PORTn, A

機能 : PORTn A  $n = N_{3-0} : 3, 8, 10$

Aレジスタの内容をPORTn ( $n = 3, 8, 10$ ) 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3, 8, 10のみ指定できます。

## 10.4.14 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。

## 10.4.15 特殊命令

## ○ SEL RBn

機能：RBS n  $n = N_{1-0} : 0-3$

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  $n = N_{3-0} : 0, 1, 15$

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。

## / GETI taddr

機能：taddr =  $T_{5-0}$ , 0 : 20H-7FH

## 【Mk モード】

TBR命令で定義したテーブルを参照した場合

PC<sub>11-0</sub> (taddr)<sub>3-0</sub> + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合

(SP - 1) PC<sub>7-4</sub>, (SP - 2) PC<sub>3-0</sub>

(SP - 3) MBE, RBE, 0, 0

(SP - 4) PC<sub>11-8</sub>

PC<sub>11-0</sub> (taddr)<sub>3-0</sub> + (taddr + 1)

SP SP - 4

TBR, TCALL命令以外で定義したテーブルを参照した場合

(taddr) (taddr + 1) を命令コードとする命令を実行



## 【Mk モード】

TBR命令で定義したテーブルを参照した場合<sup>注</sup>

PC<sub>11-0</sub> (taddr)<sub>3-0</sub> + (taddr + 1)

TCALL命令で定義したテーブルを参照した場合<sup>注</sup>

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

(SP - 3) PC<sub>7-4</sub>, (SP - 4) PC<sub>3-0</sub>

(SP - 5) 0, 0, 0, 0, (SP - 6) PC<sub>11-8</sub>

PC<sub>11-0</sub> (taddr)<sub>3-0</sub> + (taddr + 1)

SP SP - 6

TBR, TCALL命令以外で定義したテーブルを参照した場合

(taddr) (taddr + 1) を命令コードとする命令を実行

**注** TBR, TCALL命令で指定されるアドレスは0000H-0FFFHに限ります。

(taddr), (taddr + 1) で指定されるプログラム・メモリ・アドレスにある2バイトのデータを参照し、命令として実行します。

参照テーブルの領域は0020H-007FHで、あらかじめデータを書き込んでおきます。データの書き込みは、1バイト命令、2バイト命令の場合は、ニ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	<table style="border: none;"> <tr><td rowspan="3" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS L</td></tr> <tr><td>DECS L</td></tr> <tr><td>INCS HL</td></tr> <tr><td rowspan="3" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS H</td></tr> <tr><td>DECS H</td></tr> <tr><td>INCS HL</td></tr> </table>	{	INCS L	DECS L	INCS HL	{	INCS H	DECS H	INCS HL
{	INCS L								
	DECS L								
	INCS HL								
{	INCS H								
	DECS H								
	INCS HL								
MOV A, @DE XCH A, @DE	<table style="border: none;"> <tr><td rowspan="2" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS E</td></tr> <tr><td>DECS E</td></tr> <tr><td rowspan="3" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS D</td></tr> <tr><td>DECS D</td></tr> <tr><td>INCS DE</td></tr> </table>	{	INCS E	DECS E	{	INCS D	DECS D	INCS DE	
{	INCS E								
	DECS E								
{	INCS D								
	DECS D								
	INCS DE								
MOV A, @DL XCH A, @DL	<table style="border: none;"> <tr><td rowspan="2" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS L</td></tr> <tr><td>DECS L</td></tr> <tr><td rowspan="2" style="font-size: 3em; vertical-align: middle;">{</td><td>INCS D</td></tr> <tr><td>DECS D</td></tr> </table>	{	INCS L	DECS L	{	INCS D	DECS D		
{	INCS L								
	DECS L								
{	INCS D								
	DECS D								

PCは、GETI命令実行中はインクリメントは行いませんので、参照命令実行後、GETI命令の次の番地から処理を続けます。

GETI命令の前の命令がスキップ機能を持つ場合、GETI命令はほかの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持つ場合、GETI命令の次に続く命令はスキップされます。

GETI命令でたてづみ効果を持つ命令を参照した場合は次のように実行されます。

GETI命令の前の命令も同じグループのたてづみ効果を持つ命令の場合、GETI命令を実行するとたてづみ効果は消え、参照した命令はスキップされません。

GETI命令の次の命令も同じグループのたてづみ効果を持つ命令の場合、参照した命令により生ずるたてづみ効果は有効で、次に続く命令はスキップされます。

## 応用例

```
{ MOV HL, #00H }  
{ MOV XA, #FFH } をGETI命令に置き換える。  
{ 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
```

(メモ)

# 付録A $\mu$ PD75308B, 753108, 753304の機能一覧表

( 1 / 3 )

項 目		$\mu$ PD75308B	$\mu$ PD753108	$\mu$ PD753304
プログラム・メモリ		マスクROM 0000H-1F7FH ( 8064 × 8 ビット )	マスクROM 0000H-1FFFH ( 8192 × 8 ビット )	マスクROM 0000H-0FFFH ( 4096 × 8 ビット )
データ・メモリ		000H-1FFFH ( 512 × 4 ビット )		000H-0FFFH ( 256 × 4 ビット )
CPU		75X Standard	75XL CPU	
メイン・システム・クロック発振回路		クリスタル / セラミック発振回路		RC発振回路
サブシステム・クロック発振回路		クリスタル発振回路		RC発振回路
RESET信号により解除したときのウエイト時間		$2^{17}/f_x$	$2^{17}/f_x, 2^{15}/f_x$ ( マスク・オプションで選択 )	56/f <sub>cc</sub>
割り込み発生によるSTOPモード解除時のウエイト時間		$2^{20}/f_x, 2^{17}/f_x, 2^{15}/f_x, 2^{13}/f_x$ ( BTMの設定により選択 )		512/f <sub>cc</sub> , ウエイトなし ( マスク・オプションより選択 )
STOP命令を実行可能なクロック発振回路		メイン・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 とサブシステム・クロック発振回路
★ 命令実行時間	メイン・システム・クロック選択時	0.95, 1.91, 15.3 $\mu$ s ( 4.19 MHz動作時 )	<ul style="list-style-type: none"> <li>• 0.95, 1.91, 3.81, 15.3 <math>\mu</math>s ( 4.19 MHz動作時 )</li> <li>• 0.67, 1.33, 2.67, 10.7 <math>\mu</math>s ( 6.0 MHz動作時 )</li> </ul>	1.1, 2.2, 4.4, 17.8 $\mu$ s ( 3.6 MHz動作時 )
	サブシステム・クロック選択時	122 $\mu$ s ( 32.768 kHz動作時 )		85.1 $\mu$ s ( 47 kHz動作時 )
★ スタック	SBSレジスタ	なし	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	000H-1FFFH	000H-0FFFH
	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	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 !faddr	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 / 3 )

項 目		μPD75308B	μPD753108	μPD753304
入出力ポート	CMOS入力	8本	8本	0本
	CMOS入出力	16本	20本	12本
	ビット・ポート出力	8本	0本	0本
	N-chオープン・ドレイン入出力	8本	4本	0本
	合計	40本	32本	12本
LCDコントローラ / ドライバ		セグメント数選択 : 24/28/32 セグメント ( 4本単位でCMOS入出力 ポートに切り替え可能, 最 大 8本 )	セグメント数選択 : 16/20/24 セグメント ( 4本単位でCMOS入出力 ポートに切り替え可能, 最大 8本 )	セグメント数選択 : 20/24 セグメント ( 4本単位でCMOS入出力 ポートに切り替え可能, 最大 4本 )
		表示モード選択 : スタティック, 1/2デューティ ( 1/2バイアス ), 1/3デューティ ( 1/2 バイアス ), 1/3デューティ ( 1/3バイアス ), 1/4デューティ ( 1/3バ イアス )		
		LCD駆動用分割抵抗をマスク・オプションで内蔵可能		LCD駆動用分割抵抗を内蔵
		LCD駆動電圧を選択不可		
★ ★	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 ・ 8ビット・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	5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 ・ 8ビット・タイマ/イベ ント・カウンタ : 3チャ ネル( 16ビット・タイマ/イ ベント・カウンタとして使 用可能 ) ・ 時計用タイマ : 1チャンネル	3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 ネル ・ 8ビット・タイマ・カウ ンタ : 1チャンネル ( サブ クロック・ソース入力機 能あり ) ・ 時計用タイマ : 1チャンネル
★	クロック出力 ( PCL )	・ , 524, 262, 65.5 kHz (メイン・システム・クロッ ク : 4.19 MHz動作時)	・ , 524, 262, 65.5 kHz (メイン・システム・ク ロック : 4.19 MHz動作 時) ・ , 750, 375, 93.8 kHz (メイン・システム・ク ロック : 6.0 MHz動作時)	・ , 3.6 MHz, 450 kHz, 225 kHz (メイン・システム・ク ロック : 3.6 MHz動作時)

項 目		μPD75308B	μPD753108	μPD753304
★	BUZ出力 (BUZ)	<ul style="list-style-type: none"> <li>・ 2 kHz (メイン・システム・クロック : 4.19 MHz 動作時)</li> </ul>	<ul style="list-style-type: none"> <li>・ 2, 4, 32 kHz (メイン・システム・クロック : 4.19 MHz動作時 またはサブシステム・クロック : 32.768 kHz動作時)</li> <li>・ 2.93, 5.86, 46.9 kHz (メイン・システム・クロック : 6.0 MHz動作時)</li> </ul>	<ul style="list-style-type: none"> <li>・ 2.94, 5.88, 47 kHz (サブシステム・クロック : 47 kHz動作時)</li> <li>・ 1.76, 3.52, 28.13 kHz (メイン・システム・クロック : 36 MHz動作時)</li> </ul>
	シリアル・インタフェース	3種のモードに対応可能 <ul style="list-style-type: none"> <li>・ 3線式シリアル/Oモード...MSB/LSB先頭切り替え可能</li> <li>・ 2線式シリアル/Oモード</li> <li>・ SBIモード</li> </ul>		なし
S O S レ ジ ス タ	フィードバック抵抗カット・フラグ (SOS.0)	なし	内蔵	なし
	サブ発振回路電流カット・フラグ (SOS.1)	なし	内蔵	なし
	サブ発振回路停止許可フラグ (SOS.3)	なし		あり
	レジスタ・バンク選択レジスタ (RBS)	なし	あり	
	ベクタ割り込み	外部 : 3本, 内部 : 3本	外部 : 3本, 内部 : 5本	外部 : 1本, 内部 : 2本
	電源電圧	V <sub>DD</sub> = 2.0 ~ 6.0 V	V <sub>DD</sub> = 1.8 ~ 5.5 V	V <sub>DD</sub> = 2.5 ~ 5.5 V
★	動作周囲温度	T <sub>A</sub> = - 40 ~ + 85		T <sub>A</sub> = - 10 ~ + 60
★	パッケージ	<ul style="list-style-type: none"> <li>・ 80ピン・プラスチック QFP ( 14 × 20 mm )</li> <li>・ 80ピン・プラスチック QFP ( 14 mm )</li> <li>・ 80ピン・プラスチック TQFP (ファインピッチ) ( 12 mm )</li> </ul>	<ul style="list-style-type: none"> <li>・ 64ピン・プラスチック QFP ( 14 mm )</li> <li>・ 64ピン・プラスチック QFP ( 12 mm )</li> </ul>	<ul style="list-style-type: none"> <li>・ 量産品 : ペレット / ウエハ</li> <li>・ ES品 : 42ピン・セラミック・シュリンク DIP ( 600 mil ) , ( 評価用 )</li> </ul>

〔メモ〕



## 付録B 開発ツール

μPD753304を使用するシステム開発のために次のような開発ツールを用意しております。75XLシリーズでは、シリーズ共通のリロケータブル・アセンブラを品種ごとのデバイス・ファイルと組み合わせて使用します。

### 言語プロセッサ

RA75X リロケータブル・アセン ブラ	ホスト・マシン			オーダ名称 (品名)
		OS	供給媒体	
	PC-9800シリーズ	MS-DOS ( Ver.3.30 、 Ver.6.2 <sup>注</sup> )	3.5インチ2HD	μ S5A13RA75X
	IBM PC/AT <sup>TM</sup> 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			オーダ名称 (品名)
		OS	供給媒体	
	PC-9800シリーズ	MS-DOS ( Ver.3.30 、 Ver.6.2 <sup>注</sup> )	3.5インチ2HD	μ S5A13DF753304
	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 S7B13DF753304

**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** アセンブラおよびデバイス・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。

ディバグ用ツール

μ PD753304のプログラム・ディバグ用ツールとしてインサーキット・エミュレータ (IE-75000-R, IE-75001-R) を用意しています。

それぞれのシステム構成を次に示します。

ハードウェア	IE-75000-R <sup>注1</sup>	IE-75000-Rは、75Xシリーズ、75XLシリーズを使用する応用システムを開発する際に、ハードウェア、ソフトウェアのディバグを行うためのインサーキット・エミュレータです。μ PD753304を開発する場合、IE-75000-Rと別売りのエミュレーション・ボードIE-75300-R-EM、およびエミュレーション・プローブEP-753304DU-Rを組み合わせて使用します。 ホスト・マシン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。μ PD753304を開発する場合、IE-75001-Rと別売りのエミュレーション・ボードIE-75300-R-EM、およびエミュレーション・プローブEP-753304DU-Rを組み合わせて使用します。 ホスト・マシン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	μ PD753304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-753304DU-R	ES品用のエミュレーション・プローブです。 IE-75000-RまたはIE-75001-Rと、IE-75300-R-EMに接続して使用します。			
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・インタフェースで接続し、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 <sup>注2</sup> )	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。

- Ver.5.00以降にはタスク・スワップ機能がありますが、このソフトウェアではタスク・スワップ機能は使用できません。

備考 IEコントロール・プログラムの動作は、上記のホスト・マシンとOS上でのみ保証されます。

## IBM PC用のOSについて

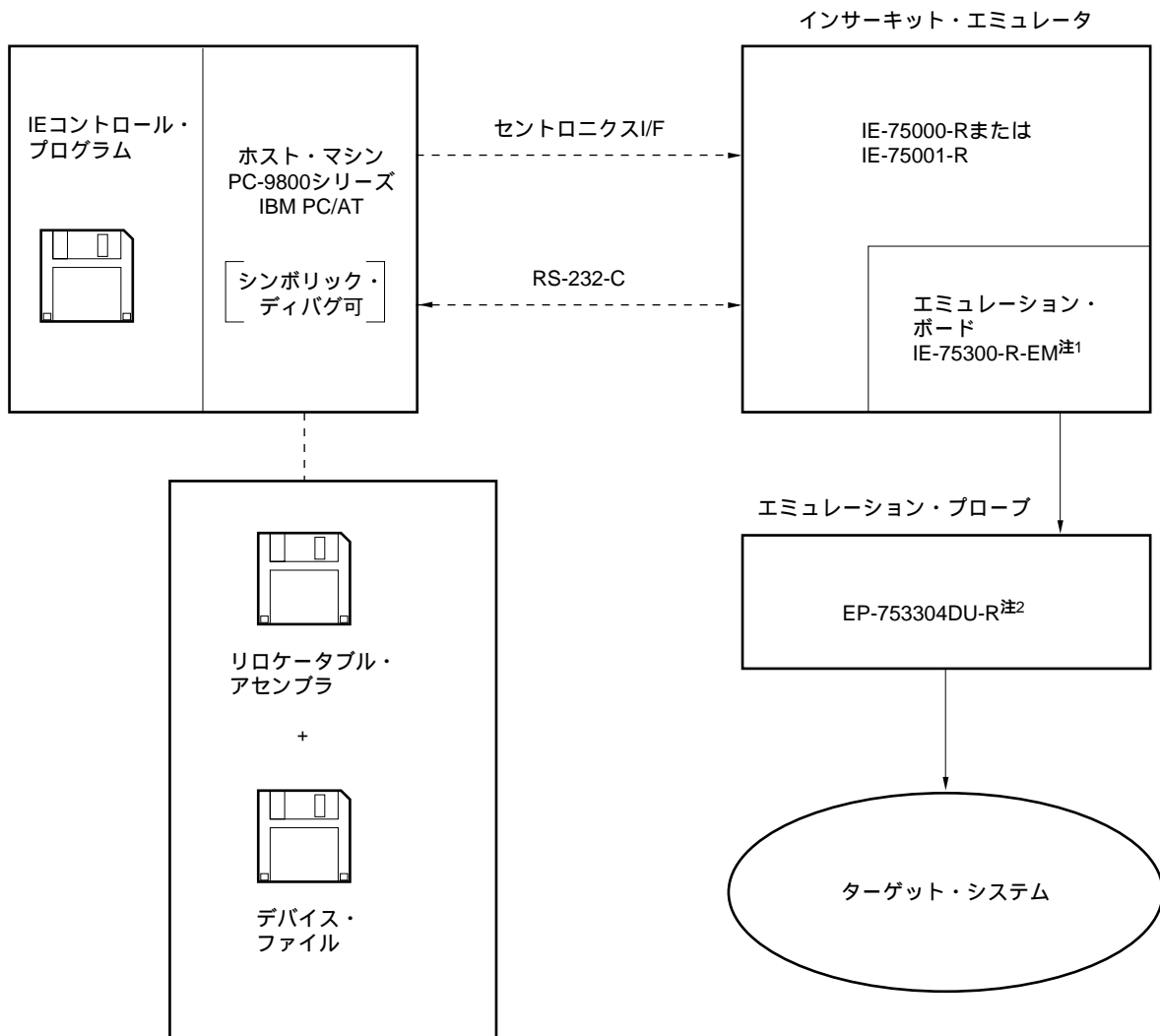
IBM PC用のOSとして、次のものがサポートされています。

OS	バージョン
PC DOS	Ver.5.02 ~ Ver.6.3 J6.1/V <sup>注</sup> ~ J6.3/V <sup>注</sup>
MS-DOS	Ver.5.0 ~ Ver.6.22 5.0/V <sup>注</sup> ~ 6.2/V <sup>注</sup>
IBM DOS™	J5.02/V <sup>注</sup>

注 英語モードのみサポートしています。

**注意** Ver.5.0以降にはタスク・スワップ機能がありますが、このソフトウェアでは、タスク・スワップ機能は使用できません。

開発ツール構成



注1 . インサーキット・エミュレータはIE-75300R-EMを内蔵していません（別売りです）。

2 . ES品用のエミュレーション・プローブです。

## 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、次の手順で行ってください。

### マスクROM発注の予約

特約店あるいは、当社販売部門に、マスクROM発注の予定を連絡してください（あらかじめご連絡をいただかないと処理が遅れる場合があります）。

### 発注用媒体の作成

マスクROM発注用の媒体には、次の3つがあります。

- ・ UV-EPROM<sup>注</sup>
- ・ 3.5インチのIBMフォーマットのフロッピー・ディスク（海外のみ）
- ・ 5インチのIBMフォーマットのフロッピー・ディスク（海外のみ）

**注** UV-EPROMで発注する場合には、同じ内容のUV-EPROMを3個作成してください。  
また、マスク・オプション・データはマスク・オプション情報書に記入してください。

### 必要書類の作成

マスクROMの発注にあたっては、下記の書類に記入してください。

- A．マスク式ROM発注書
- B．マスク式ROM発注チェック・シート
- C．マスク・オプション情報書（マスク・オプションがある製品の場合必要）

### 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当社販売部門に提出してください。

**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」（C10302J）をご覧ください。

(メモ)

## 付録D 命令索引

### D.1 命令索引 (機能別)

#### [ 転送命令 ]

MOV A, #n4 ... 185, 199  
MOV reg1, #n4 ... 185, 199  
MOV XA, #n8 ... 185, 199  
MOV HL, #n8 ... 185, 200  
MOV rp2, #n8 ... 185, 200  
MOV A, @HL ... 185, 200  
MOV A, @HL + ... 185, 200  
MOV A, @HL - ... 185, 200  
MOV A, @rpa1 ... 185, 200  
MOV XA, @HL ... 185, 201  
MOV @HL, A ... 185, 201  
MOV @HL, XA ... 185, 201  
MOV A, mem ... 185, 201  
MOV XA, mem ... 185, 202  
MOV mem, A ... 185, 202  
MOV mem, XA ... 185, 202  
MOV A, reg ... 185, 202  
MOV XA, rp' ... 185, 203  
MOV reg1, A ... 185, 203  
MOV rp' 1, XA ... 185, 203  
XCH A, @HL ... 185, 204  
XCH A, @HL + ... 185, 204  
XCH A, @HL - ... 185, 204  
XCH A, @rpa1 ... 185, 204  
XCH XA, @HL ... 185, 204  
XCH A, mem ... 185, 205  
XCH XA, mem ... 185, 205  
XCH A, reg1 ... 185, 205  
XCH XA, rp' ... 185, 205

#### [ テーブル参照命令 ]

MOVT XA, @PCDE ... 186, 206  
MOVT XA, @PCXA ... 186, 208  
MOVT XA, @BCDE ... 186, 209  
MOVT XA, @BCXA ... 186, 209

#### [ ビット転送命令 ]

MOV1 CY, fmem. bit ... 186, 210  
MOV1 CY, pmem. @L ... 186, 210  
MOV1 CY, @H + mem. bit ... 186, 210  
MOV1 fmem. bit, CY ... 186, 210  
MOV1 pmem. @L, CY ... 186, 210  
MOV1 @H + mem. bit, CY ... 186, 210

#### [ 演算命令 ]

ADDS A, #n4 ... 186, 211  
ADDS XA, #n8 ... 186, 211  
ADDS A, @HL ... 186, 211  
ADDS XA, rp' ... 186, 211  
ADDS rp' 1, XA ... 186, 212  
ADDC A, @HL ... 186, 212  
ADDC XA, rp' ... 186, 212  
ADDC rp' 1, XA ... 186, 213  
SUBS A, @HL ... 186, 213  
SUBS XA, rp' ... 186, 213  
SUBS rp' 1, XA ... 186, 214  
SUBC A, @HL ... 186, 214  
SUBC XA, rp' ... 186, 214  
SUBC rp' 1, XA ... 186, 215  
AND A, #n4 ... 186, 215  
AND A, @HL ... 186, 215

AND XA, rp' ... 186, 215  
 AND rp' 1, XA ... 186, 216  
 OR A, #n4 ... 186, 216  
 OR A, @HL ... 186, 216  
 OR XA, rp' ... 186, 216  
 OR rp' 1, XA ... 186, 217  
 XOR A, #n4 ... 186, 217  
 XOR A, @HL ... 186, 217  
 XOR XA, rp' ... 186, 217  
 XOR rp' 1, XA ... 186, 218

#### [ アキュムレータ操作命令 ]

RORC A ... 187, 219  
 NOT A ... 187, 219

#### [ 増減命令 ]

INCS reg ... 187, 220  
 INCS rp1 ... 187, 220  
 INCS @HL ... 187, 220  
 INCS mem ... 187, 220  
 DECS reg ... 187, 221  
 DECS rp' ... 187, 221

#### [ 比較命令 ]

SKE reg, #n4 ... 187, 222  
 SKE @HL, #n4 ... 187, 222  
 SKE A, @HL ... 187, 222  
 SKE XA, @HL ... 187, 222  
 SKE A, reg ... 187, 223  
 SKE XA, rp' ... 187, 223

#### [ キャリー・フラグ操作命令 ]

SET1 CY ... 187, 224  
 CLR1 CY ... 187, 224  
 SKT CY ... 187, 224  
 NOT1 CY ... 187, 224

#### [ メモリ・ビット操作命令 ]

SET1 mem. bit ... 187, 225  
 SET1 fmem. bit ... 187, 225  
 SET1 pmem. @L ... 187, 225  
 SET1 @H + mem. bit ... 187, 225  
 CLR1 mem. bit ... 187, 225  
 CLR1 fmem. bit ... 187, 225  
 CLR1 pmem. @L ... 187, 225  
 CLR1 @H + mem. bit ... 187, 225  
 SKT mem. bit ... 187, 226  
 SKT fmem. bit ... 187, 226  
 SKT pmem. @L ... 187, 226  
 SKT @H + mem. bit ... 187, 226  
 SKF mem. bit ... 187, 226  
 SKF fmem. bit ... 187, 226  
 SKF pmem. @L ... 187, 226  
 SKF @H + mem. bit ... 187, 226  
 SKTCLR fmem. bit ... 187, 227  
 SKTCLR pmem. @L ... 187, 227  
 SKTCLR @H + mem. bit ... 187, 227  
 AND1 CY, fmem. bit ... 188, 227  
 AND1 CY, pmem. @L ... 188, 227  
 AND1 CY, @H + mem. bit ... 188, 227  
 OR1 CY, fmem. bit ... 188, 227  
 OR1 CY, pmem. @L ... 188, 227  
 OR1 CY, @H + mem. bit ... 188, 227  
 XOR1 CY, fmem. bit ... 188, 228  
 XOR1 CY, pmem. @L ... 188, 228  
 XOR1 CY, @H + mem. bit ... 188, 228

#### [ 分岐命令 ]

BR addr ... 188, 229  
 BR addr1 ... 188, 229  
 BRA ! addr1 ... 188, 229  
 BR ! addr ... 188, 229  
 BR \$ addr ... 188, 230  
 BR \$ addr1 ... 188, 230



BRCB ! caddr ... 188, 231  
BR PCDE ... 188, 232  
BR PCXA ... 188, 232  
BR BCDE ... 188, 233  
BR BCXA ... 188, 233  
TBR addr ... 190, 233  
SEL MBn ... 190, 242  
GETI taddr ... 190, 242

**[ サブルーチン・スタック制御命令 ]**

CALLA ! addr1 ... 189, 234  
CALL ! addr ... 189, 234  
CALLF ! faddr ... 189, 235  
TCALL ! addr ... 190, 235  
RET ... 189, 236  
RETS ... 189, 236  
RETI ... 190, 237  
PUSH rp ... 190, 238  
PUSH BS ... 190, 238  
POP rp ... 190, 238  
POP BS ... 190, 238

**[ 割り込み制御命令 ]**

EI ... 190, 239  
EI IE x x x ... 190, 239  
DI ... 190, 239  
DI IE x x x ... 190, 239

**[ 入出力命令 ]**

IN A, PORTn ... 190, 240  
OUT PORTn, A ... 190, 240

**[ CPU制御命令 ]**

HALT ... 190, 241  
STOP ... 190, 241  
NOP ... 190, 241

**[ 特殊命令 ]**

SEL RBn ... 190, 242

## D.2 命令索引 (アルファベット順)

## [ A ]

ADDC A, @HL ... 186, 212  
 ADDC rp' 1, XA ... 186, 213  
 ADDC XA, rp' ... 186, 212  
 ADDS A, #n4 ... 186, 211  
 ADDS A, @HL ... 186, 211  
 ADDS rp' 1, XA ... 186, 212  
 ADDS XA, rp' ... 186, 211  
 ADDS XA, #n8 ... 186, 211  
 AND A, #n4 ... 186, 215  
 AND A, @HL ... 186, 215  
 AND rp' 1, XA ... 186, 216  
 AND XA, rp' ... 186, 215  
 AND1 CY, fmem. bit ... 188, 227  
 AND1 CY, pmem. @L ... 188, 227  
 AND1 CY, @H+mem. bit ... 188, 227

## [ B ]

BR addr ... 188, 229  
 BR addr1 ... 188, 229  
 BR BCDE ... 188, 233  
 BR BCXA ... 188, 233  
 BR PCDE ... 188, 232  
 BR PCXA ... 188, 232  
 BR !addr ... 188, 229  
 BR \$addr ... 188, 230  
 BR \$addr1 ... 188, 230  
 BRA !addr1 ... 188, 229  
 BR CB !caddr ... 188, 231

## [ C ]

CALL !addr ... 189, 234  
 CALLA !addr1 ... 189, 234  
 CALLF !faddr ... 189, 235  
 CLR1 CY ... 187, 224

CLR1 fmem. bit ... 187, 225  
 CLR1 mem. bit ... 187, 225  
 CLR1 pmem. @L ... 187, 225  
 CLR1 @H+mem. bit ... 187, 225

## [ D ]

DECS reg ... 187, 221  
 DECS rp' ... 187, 221  
 DI ... 190, 239  
 DI IE x x x ... 190, 239

## [ E ]

EI ... 190, 239  
 EI IE x x x ... 190, 239

## [ G ]

GETI taddr ... 190, 242

## [ H ]

HALT ... 190, 241

## [ I ]

IN A, PORTn ... 190, 240  
 INCS mem ... 187, 220  
 INCS reg ... 187, 220  
 INCS rp1 ... 187, 220  
 INCS @HL ... 187, 220

## [ M ]

MOV A, mem ... 185, 201  
 MOV A, reg ... 185, 202  
 MOV A, #n4 ... 185, 199  
 MOV A, @HL ... 185, 200  
 MOV A, @HL+ ... 185, 200  
 MOV A, @HL- ... 185, 200



SKT @H + mem. bit ... 187, 226  
SKTCLR fmem. bit ... 187, 227  
SKTCLR pmem. @L ... 187, 227  
SKTCLR @H + mem. bit ... 187, 227  
STOP ... 190, 241  
SUBC A, @HL ... 186, 214  
SUBC rp' 1, XA ... 186, 215  
SUBC XA, rp' ... 186, 214  
SUBS A, @HL ... 186, 213  
SUBS rp' 1, XA ... 186, 214  
SUBS XA, rp' ... 186, 213

## [ T ]

TBR addr ... 190, 233  
TCALL ! addr ... 190, 235

## [ X ]

XCH A, mem ... 185, 205  
XCH A, reg1 ... 185, 205  
XCH A, @HL ... 185, 204  
XCH A, @HL + ... 185, 204  
XCH A, @HL - ... 185, 204  
XCH A, @rpa1 ... 185, 204  
XCH XA, mem ... 185, 205  
XCH XA, rp' ... 185, 205  
XCH XA, @HL ... 185, 204  
XOR A, #n4 ... 186, 217  
XOR A, @HL ... 186, 217  
XOR rp' 1, XA ... 186, 218  
XOR XA, rp' ... 186, 217  
XOR1 CY, fmem. bit ... 188, 228  
XOR1 CY, pmem. @L ... 188, 228  
XOR1 CY, @H + mem. bit ... 188, 228

## 付録E ハードウェア索引

### E.1 ハードウェア索引 (50音順)

#### [あ行]

INTAレジスタ (INTA) ...	37
INTCレジスタ (INTC) ...	37
INTEレジスタ (INTE) ...	37
INTGレジスタ (INTG) ...	37
INT1エッジ検出モード・レジスタ (IM1) ...	140
INT1割り込み許可フラグ (IE1) ...	137
INT1割り込み要求フラグ (IRQ1) ...	137
ウォッチドッグ・タイマ許可フラグ (WDTM) ...	88

#### [か行]

キャリー・フラグ (CY) ...	54
クロック出力モード・レジスタ (CLOM) ...	84

#### [さ行]

サブ発振回路コントロール・レジスタ (SOS) ...	80
システム・クロック・コントロール・レジスタ (SCC) ...	77
スキップ・フラグ0-2 (SK0-SK2) ...	55
スタック・バンク選択レジスタ (SBS) ...	40, 50
スタック・ポインタ (SP) ...	50

#### [た行]

タイマ・カウンタ0カウント・レジスタ (T0) ...	36, 97
タイマ・カウンタ0モード・レジスタ (TM0) ...	98, 99
タイマ・カウンタ0モジュロ・レジスタ (TMOD0) ...	36, 97
タイマ・カウンタ0割り込み許可フラグ (IET0) ...	137
タイマ・カウンタ0割り込み要求フラグ (IRQT0) ...	137
時計モード・レジスタ (WM) ...	95

## 【は行】

- バンク選択レジスタ (BS) ... 58
- BT割り込み許可フラグ (IEBT) ... 137
- BT割り込み要求フラグ (IRQBT) ... 137
- 表示制御レジスタ (LCDC) ... 109
- 表示モード・レジスタ (LCDM) ... 107
- プルアップ抵抗指定レジスタ・グループB (POGB) ... 71
- プログラム・カウンタ (PC) ... 41
- プログラム・ステータス・ワード (PSW) ... 54
- プロセッサ・クロック・コントロール・レジスタ (PCC) ... 76
- ベーシック・インターバル・タイマ (BT) ... 87
- ベーシック・インターバル・タイマ・モード・レジスタ (BTM) ... 87
- ポート (PORT3, 8, 10) ... 62
- ポート・モード・レジスタ・グループA, C, D (PMGA, PMGC, PMGD) ... 65

## 【ま行】

- メモリ・バンク許可フラグ (MBE) ... 15, 57
- メモリ・バンク選択レジスタ (MBS) ... 15, 58

## 【ら行】

- レジスタ・バンク許可フラグ (RBE) ... 29, 57
- レジスタ・バンク選択レジスタ (RBS) ... 29, 59

## 【わ行】

- 割り込みステータス・フラグ0, 1 (IST0, IST1) ... 56, 141
- 割り込みプライオリティ選択レジスタ (IPS) ... 138
- 割り込みマスタ許可フラグ (IME) ... 139

## E.2 ハードウェア索引 (アルファベット順)

### [B]

BS ... 58  
BT ... 87  
BTM ... 87

### [C]

CLOM ... 84  
CY ... 54

### [I]

IE1 ... 137  
IEBT ... 137  
IETO ... 137  
IEW ... 155  
IM1 ... 140  
IME ... 139  
INTA ... 37  
INTC ... 37  
INTE ... 37  
INTG ... 37  
IPS ... 138  
IRQ1 ... 137  
IRQBT ... 137  
IRQT0 ... 137  
IRQW ... 155  
IST0, IST1 ... 56, 141

### [L]

LCDC ... 109  
LCDM ... 107

### [M]

MBE ... 15, 57  
MBS ... 15, 58

### [P]

PC ... 41  
PCC ... 76  
PMGA, PMGC, PMGD ... 65  
POGB ... 71  
PORT3, 8, 10 ... 62  
PSW ... 54

### [R]

RBE ... 29, 57  
RBS ... 29, 59

### [S]

SBS ... 40, 50  
SCC ... 77  
SK0-SK2 ... 55  
SOS ... 80  
SP ... 50

### [T]

T0 ... 36, 97  
TM0 ... 98, 99  
TMOD0 ... 36, 97

### [W]

WDTM ... 88  
WM ... 95

(メモ)



★

## 付録F 改版履歴

これまでの改版履歴を次に示します。なお、適用箇所は各版での章を示します。

版 数	前版からの改版内容	適用箇所
第 2 版	μ PD753304を開発中から開発済みに変更	全 般
	量産品の出荷形態をペレット/ウエハに変更	
	メイン・システム・クロックを4.0 MHzから3.6 MHz (typ.) に、サブシステム・クロックを32.768 kHzから47 kHz (typ.) に変更	
	ポート 3 のマスク・オプションによるリセット時の状態を追加	第 2 章 端子機能
	<b>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</b> を訂正	第 3 章 アーキテクチャの特徴とメモリ・マップ
	<b>図 4 - 5 表示データ・メモリの構成</b> の中の説明を訂正	第 4 章 内部CPU機能
	LCD表示モードのマスク・オプションを追加	第 5 章 周辺ハードウェア機能
	LCD駆動用分割抵抗の10 k (typ.) を削除	
	<b>5.5.6 表示データ・メモリの本文のビット操作の記述</b> を訂正	
	サブシステム・クロックのSTOPモード解除信号に割り込み要求信号を追加	第 7 章 スタンバイ機能
	<b>10.1.6 表示データ・メモリ操作可能命令</b> を追加	第10章 命令セット
<b>10.3 各命令の命令コードの命令一覧表</b> (オペランド: BCDE) を訂正		

---

## — 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] μPD753304 ユーザーズ・マニュアル

(U12020JJ2V1UM00 (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