

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、誤りが無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。

# $\mu$ PD754202

## 4ビット・シングルチップ・マイクロコンピュータ

第1章	概説	1
第2章	端子機能	2
第3章	アーキテクチャの特徴とメモリ・マップ	3
第4章	内部CPU機能	4
第5章	周辺ハードウェア機能	5
第6章	割り込み機能とテスト機能	6
第7章	スタンバイ機能	7
第8章	リセット機能	8
第9章	マスク・オプション	9
第10章	命令セット	10
付録A	$\mu$ PD754202, 75F4264の機能一覧表	付A
付録B	開発ツール	付B
付録C	マスクROM発注の手順	付C
付録D	命令索引	付D
付録E	ハードウェア索引	付E
付録F	改版履歴	付F

## 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。

EEPROMは、日本電気株式会社の商標です。

MS-DOSは、米国マイクロソフト社の商標です。

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戦略物資等（または役務）に該当するか否かは、ユーザー（仕様を決定した者）が判定してください。

**本資料の内容は、後日変更する場合があります。**

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所有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。

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。

この製品は耐放射線設計をしておりません。

## 本版で改訂された主な箇所

箇所	内容
p.273	付録A $\mu$ PD754202, 75F4264の機能一覧表を追加
p.275	付録B 開発ツールのデバイス・ファイル名を変更
p.291	付録F 改版履歴を追加

本文欄外の★印は、本版で改訂された主な箇所を示しています。

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

# はじめに

**ご利用対象者** このマニュアルは、 $\mu$ PD754202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 $\mu$ PD754202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。

- ・ニモニックが分かっているときの命令機能を調べるとき

付録D 命令索引を利用してください。

- ・特定の内蔵回路などの機能を調べたいとき

付録E ハードウェア索引を利用してください。

- ・一通り $\mu$ PD754202の機能を理解しようとするとき

目次に従って読んでください。



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

- ★ 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
μPD754202 ユーザーズ・マニュアル	U11132J (このマニュアル)	U11132E
μPD754202 データ・シート	U12181J	U12181E
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-754144GS-R ユーザーズ・マニュアル	U10695J	U10695E	
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	EEU-731	EEU-1346
		言語編	EEU-730	EEU-1363

#### その他のドキュメント

資料名	資料番号	
	和文	英文
IC PACKAGE MANUAL	C10943X	
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) 試験について	MEM-539	-
半導体デバイスの品質保証ガイド	C11893J	MEI-1202
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 ...	2
1.3 ブロック図 ...	3
1.4 端子接続図 (Top View) ...	4
<b>第2章 端子機能</b> ...	5
2.1 端子機能一覧 ...	5
2.2 端子機能の説明 ...	7
2.2.1 P30-P33 (PORT3) , P60-P63 (PORT6) , P80 (PORT8) ...	7
2.2.2 P70-P73 (PORT7) ...	7
2.2.3 PTO0-PTO2 ...	7
2.2.4 INT0 ...	8
2.2.5 KR4-KR7 ...	8
2.2.6 KRREN ...	8
2.2.7 X1, X2 ...	8
2.2.8 $\overline{\text{RESET}}$ ...	9
2.2.9 IC ...	9
2.2.10 $V_{DD}$ ...	9
2.2.11 $V_{SS}$ ...	9
2.3 端子の入出力回路 ...	10
2.4 未使用端子の処理について ...	11
<b>第3章 アーキテクチャの特徴とメモリ・マップ</b> ...	13
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 ...	13
3.1.1 データ・メモリのバンク構成 ...	13
3.1.2 データ・メモリのアドレッシング・モード ...	15
3.2 汎用レジスタのバンク構成 ...	27
3.3 メモリ・マップトI/O ...	32
<b>第4章 内部CPU機能</b> ...	43
4.1 Mk モードとMk モードの切り替え機能 ...	43
4.1.1 Mk モードとMk モードの違い ...	43
4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法 ...	44
4.2 プログラム・カウンタ (PC) ...	45
4.3 プログラム・メモリ (ROM) ...	46
4.4 データ・メモリ (RAM) ...	48
4.4.1 データ・メモリの構成 ...	48
4.4.2 データ・メモリのバンクの指定 ...	49
4.5 汎用レジスタ ...	52

- 4.6 アキュムレータ ... 53
- 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) ... 53
- 4.8 プログラム・ステータス・ワード (PSW) ... 57
- 4.9 バンク選択レジスタ (BS) ... 61

## 第5章 周辺ハードウェア機能 ...63

- 5.1 デジタル入出力ポート ... 63
  - 5.1.1 デジタル入出力ポートの種類, 特徴, 構成 ... 64
  - 5.1.2 入出力モードの設定 ... 68
  - 5.1.3 デジタル入出力ポート操作命令 ... 70
  - 5.1.4 デジタル入出力ポートの動作 ... 72
  - 5.1.5 プルアップ抵抗の内蔵 ... 74
  - 5.1.6 デジタル入出力ポートの入出力タイミング ... 75
- 5.2 クロック発生回路 ... 77
  - 5.2.1 クロック発生回路の構成 ... 77
  - 5.2.2 クロック発生回路の機能と動作 ... 78
  - 5.2.3 CPUクロックの設定 ... 84
- 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) ... 89
  - 5.3.4 ベーシック・インターバル・タイマの動作 ... 90
  - 5.3.5 ウォッチドッグ・タイマの動作 ... 91
  - 5.3.6 その他の機能 ... 93
- 5.4 タイマ・カウンタ ... 95
  - 5.4.1 タイマ・カウンタの構成 ... 95
  - 5.4.2 8ビット・タイマ・カウンタ・モードの動作 ... 107
  - 5.4.3 PWMパルス・ジェネレータ・モード (PWMモード) の動作 ... 116
  - 5.4.4 16ビット・タイマ・カウンタ・モードの動作 ... 122
  - 5.4.5 キャリア・ジェネレータ・モード (CGモード) の動作 ... 130
  - 5.4.6 タイマ・カウンタ応用上の注意 ... 144
- 5.5 ビット・シーケンシャル・バッファ ... 151

## 第6章 割り込み機能とテスト機能 ... 153

- 6.1 割り込み制御回路の構成 ... 153
- 6.2 割り込みソースの種類とベクタ・テーブル ... 155
- 6.3 割り込み機能を制御する各種ハードウェア ... 157
- 6.4 割り込みシーケンス ... 164
- 6.5 多重割り込み処理制御 ... 165
- 6.6 ベクタ・アドレス共用割り込み処理 ... 167
- 6.7 割り込み処理までのマシン・サイクル ... 169
- 6.8 割り込みの効果的な使い方 ... 171
- 6.9 割り込みの応用 ... 171

6.10	テスト機能	...	179
6.10.1	テスト・ソースの種類	...	179
6.10.2	テスト機能を制御する各種ハードウェア	...	179
<b>第7章 スタンバイ機能</b> ... 183			
7.1	スタンバイ・モードの設定および動作状態	...	185
7.2	スタンバイ・モードの解除	...	187
7.3	スタンバイ・モード解除後の動作	...	191
7.4	スタンバイ・モードの応用	...	191
<b>第8章 リセット機能</b> ... 197			
8.1	リセット機能の構成および動作	...	197
8.2	ウォッチドッグ・フラグ(WDF), キー・リターン・フラグ(KRF)	...	201
<b>第9章 マスク・オプション</b> ... 203			
9.1	端子のマスク・オプション	...	203
9.1.1	P70/KR4-P73/KR7のマスク・オプション	...	203
9.1.2	RESET端子のマスク・オプション	...	203
9.2	発振安定ウエイト時間のマスク・オプション	...	203
<b>第10章 命令セット</b> ... 205			
10.1	特徴的な命令	...	205
10.1.1	GETI命令	...	205
10.1.2	ビット操作命令	...	206
10.1.3	たてづみ命令	...	206
10.1.4	進数補正命令	...	207
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	...	208
10.2	命令セットとそのオペレーション	...	208
10.3	各命令の命令コード	...	218
10.4	命令機能と応用	...	224
10.4.1	転送命令	...	225
10.4.2	テーブル参照命令	...	233
10.4.3	ビット転送命令	...	237
10.4.4	演算命令	...	238
10.4.5	アキュムレータ操作命令	...	246
10.4.6	増減命令	...	247
10.4.7	比較命令	...	249
10.4.8	キャリー・フラグ操作命令	...	251
10.4.9	メモリ・ビット操作命令	...	252
10.4.10	分岐命令	...	256
10.4.11	サブルーチン・スタック制御命令	...	260
10.4.12	割り込み制御命令	...	265
10.4.13	入出力命令	...	266
10.4.14	CPU制御命令	...	267
10.4.15	特殊命令	...	268

★	付録 A	μPD754202, 75F4264の機能一覧表	...	273
	付録 B	開発ツール	...	275
	付録 C	マスクROM発注の手順	...	279
	付録 D	命令索引	...	281
		D.1 命令索引(機能別)	...	281
		D.2 命令索引(アルファベット順)	...	284
	付録 E	ハードウェア索引	...	287
		E.1 ハードウェア索引(50音順)	...	287
		E.2 ハードウェア索引(アルファベット順)	...	290
★	付録 F	改版履歴	...	291

## 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 ... 14
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 ... 16
3 - 3	スタティックRAMのアドレス更新方法 ... 21
3 - 4	レジスタ・バンクの使い分け例 ... 28
3 - 5	汎用レジスタの構成 (4ビット処理の場合) ... 30
3 - 6	汎用レジスタの構成 (8ビット処理の場合) ... 31
3 - 7	I/Oマップ ... 34
4 - 1	スタック・バンク選択レジスタのフォーマット ... 44
4 - 2	プログラム・カウンタの構成 ... 45
4 - 3	プログラム・メモリ・マップ ... 47
4 - 4	データ・メモリ・マップ ... 50
4 - 5	汎用レジスタの構成 ... 52
4 - 6	レジスタ・ペアの構成 ... 52
4 - 7	アキュムレータ ... 53
4 - 8	スタック・ポインタおよびスタック・バンク選択レジスタの構成 ... 54
4 - 9	スタック・メモリへ退避されるデータ (Mk モード) ... 55
4 - 10	スタック・メモリから復帰されるデータ (Mk モード) ... 55
4 - 11	スタック・メモリへ退避されるデータ (Mk モード) ... 56
4 - 12	スタック・メモリから復帰されるデータ (Mk モード) ... 56
4 - 13	プログラム・ステータス・ワードの構成 ... 57
4 - 14	バンク選択レジスタの構成 ... 61
5 - 1	デジタル・ポートのデータ・メモリ・アドレス ... 63
5 - 2	P3nの構成 (n = 0-2) ... 65
5 - 3	P33の構成 ... 65
5 - 4	P6nの構成 (n = 0, 2, 3) ... 66
5 - 5	P61の構成 ... 66
5 - 6	P7nの構成 (n = 0-3) ... 67
5 - 7	P80の構成 ... 67
5 - 8	各ポート・モード・レジスタのフォーマット ... 69
5 - 9	プルアップ抵抗指定レジスタのフォーマット ... 74
5 - 10	デジタル入出力ポートの入出力タイミング ... 75
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 ... 76



## 図の目次 (2/3)

図番号	タイトル, ページ
5 - 12	クロック発生回路のブロック図 ... 77
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 ... 80
5 - 14	クリスタル/セラミック発振の外付け回路 ... 81
5 - 15	発振子の接続回路の悪い例 ... 82
5 - 16	CPUクロックの切り替え ... 85
5 - 17	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 - 18	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 ... 88
5 - 19	ウォッチドッグ・タイマ許可フラグ (WDTM) のフォーマット ... 89
5 - 20	タイマ・カウンタ (チャンネル0) のブロック図 ... 96
5 - 21	タイマ・カウンタ (チャンネル1) のブロック図 ... 97
5 - 22	タイマ・カウンタ (チャンネル2) のブロック図 ... 98
5 - 23	タイマ・カウンタ・モード・レジスタ (チャンネル0) のフォーマット ... 100
5 - 24	タイマ・カウンタ・モード・レジスタ (チャンネル1) のフォーマット ... 101
5 - 25	タイマ・カウンタ・モード・レジスタ (チャンネル2) のフォーマット ... 103
5 - 26	タイマ・カウンタ出力許可フラグのフォーマット ... 105
5 - 27	タイマ・カウンタ・コントロール・レジスタのフォーマット ... 106
5 - 28	タイマ・カウンタ・モード・レジスタの設定 ... 108
5 - 29	タイマ・カウンタ・コントロール・レジスタの設定 ... 111
5 - 30	タイマ・カウンタ出力許可フラグの設定 ... 111
5 - 31	タイマ・カウンタ動作時の構成 ... 114
5 - 32	カウンタ動作時のタイミング ... 114
5 - 33	タイマ・カウンタ・モード・レジスタの設定 ... 117
5 - 34	タイマ・カウンタ・コントロール・レジスタの設定 ... 118
5 - 35	PWMパルス・ジェネレータ動作時の構成 ... 120
5 - 36	PWMパルス・ジェネレータ動作時のタイミング ... 120
5 - 37	タイマ・カウンタ・モード・レジスタの設定 ... 123
5 - 38	タイマ・カウンタ・コントロール・レジスタの設定 ... 124
5 - 39	タイマ・カウンタ動作時の構成 ... 127
5 - 40	カウンタ動作時のタイミング ... 128
5 - 41	タイマ・カウンタ・モード・レジスタの設定 ... 131
5 - 42	タイマ・カウンタ出力許可フラグの設定 ... 132
5 - 43	タイマ・カウンタ・コントロール・レジスタの設定 ... 132
5 - 44	キャリア・ジェネレータ動作時の構成 ... 135
5 - 45	キャリア・ジェネレータ動作時のタイミング ... 136
5 - 46	ビット・シーケンシャル・バッファのフォーマット ... 151

## 図の目次 (3/3)

図番号	タイトル, ページ
6 - 1	割り込み制御回路ブロック図 ... 154
6 - 2	割り込みベクタ・テーブル ... 156
6 - 3	割り込みプライオリティ選択レジスタ ... 159
6 - 4	INT0の構成 ... 161
6 - 5	ノイズ除去回路の入出力タイミング ... 161
6 - 6	INT0エッジ検出モード・レジスタ (IM0) のフォーマット ... 162
6 - 7	割り込み処理シーケンス ... 164
6 - 8	高位の割り込みによる多重割り込み ... 165
6 - 9	割り込みステータス・フラグの変更による多重割り込み ... 166
6 - 10	KR4-KR7のブロック図 ... 180
6 - 11	INT2エッジ検出モード・レジスタ (IM2) のフォーマット ... 181
7 - 1	スタンバイ・モードの解除動作 ... 187
7 - 2	STOPモード解除時のウェイト時間 ... 189
7 - 3	キー・リターン・リセットまたは $\overline{\text{RESET}}$ 入力によるSTOPモードの解除 ... 190
8 - 1	リセット機能の構成 ... 197
8 - 2	$\overline{\text{RESET}}$ 信号発生によるリセット動作 ... 198
8 - 3	各信号発生時におけるWDFの動作 ... 201
8 - 4	各信号発生時におけるKRFの動作 ... 202

## 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 ... 5
2 - 2	ポート端子以外の端子機能一覧 ... 6
2 - 3	未使用端子の処理一覧 ... 11
3 - 1	アドレッシング・モード ... 17
3 - 2	RBE, RBSと選択されるレジスタ・バンク ... 27
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 ... 27
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 ... 32
4 - 1	Mk モードとMk モードの違い ... 43
4 - 2	SBSで選択されるスタック・エリア ... 53
4 - 3	スタック動作時に退避 / 復帰されるPSWのフラグ ... 57
4 - 4	キャリー・フラグ操作命令 ... 58
4 - 5	割り込みステータス・フラグの指示内容 ... 59
4 - 6	MBE, MBSと選択されるメモリ・バンク ... 61
4 - 7	RBE, RBSと選択されるレジスタ・バンク ... 62
5 - 1	デジタル・ポートの種類と特徴 ... 64
5 - 2	入出力端子操作命令一覧表 ... 71
5 - 3	入出力ポート操作時の動作 ... 73
5 - 4	プルアップ抵抗内蔵の指定法 ... 74
5 - 5	CPUクロックの切り替えに要する最大時間 ... 84
5 - 6	モード一覧 ... 95
5 - 7	分解能と最長設定時間 (8ビット・タイマ・カウンタ・モードの場合) ... 112
5 - 8	分解能と最長設定時間 (16ビット・タイマ・カウンタ・モードの場合) ... 125
6 - 1	割り込みソースの種類 ... 155
6 - 2	割り込み要求フラグのセット信号 ... 158
6 - 3	IST1, IST0と割り込み処理状態 ... 163
6 - 4	共用割り込みの判別 ... 167
6 - 5	テスト・ソースの種類 ... 179
6 - 6	テスト要求フラグのセット信号 ... 179
6 - 7	KR4-KR7端子とKRREN端子およびテスト機能 ... 181

## 表の目次 (2/2)

表番号	タイトル, ページ
7 - 1	スタンバイ・モード時の各動作状態 ... 185
7 - 2	BTMによるウェイト時間の選択 ... 189
8 - 1	各ハードウェアのリセット後の状態 ... 199
8 - 2	各信号に対するWDF, KRFの内容 ... 201
9 - 1	マスク・オプションの選択 ... 203
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 ... 206

〔メ モ〕

μPD754202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。

μPD754202は、75Xシリーズに比べて、CPU機能を拡張し、低電圧1.8Vでの高速動作を可能にしています。小型のプラスチック・シュリンクSOP（300 mil, 0.65 mmピッチ）を用意しています。

次に特徴を示します。

- ・低電圧動作可能： $V_{DD}=1.8\sim 6.0\text{ V}$
- ・高速動作とパワー・セーブに有利な命令実行時間可変機能
  - 0.95, 1.91, 3.81, 15.3  $\mu\text{s}$ （4.19 MHz動作時）
  - 0.67, 1.33, 2.67, 10.7  $\mu\text{s}$ （6.0 MHz動作時）
- ・タイマを4チャンネル内蔵
- ・キーレス・エントリ用のキー・リターン・リセット機能を内蔵
- ・小型パッケージ（20ピン・プラスチック・シュリンクSOP（300 mil, 0.65 mmピッチ））を用意

## 応用分野

- ・キーレス・エントリなどの自動車電装

## 1.1 機能概要

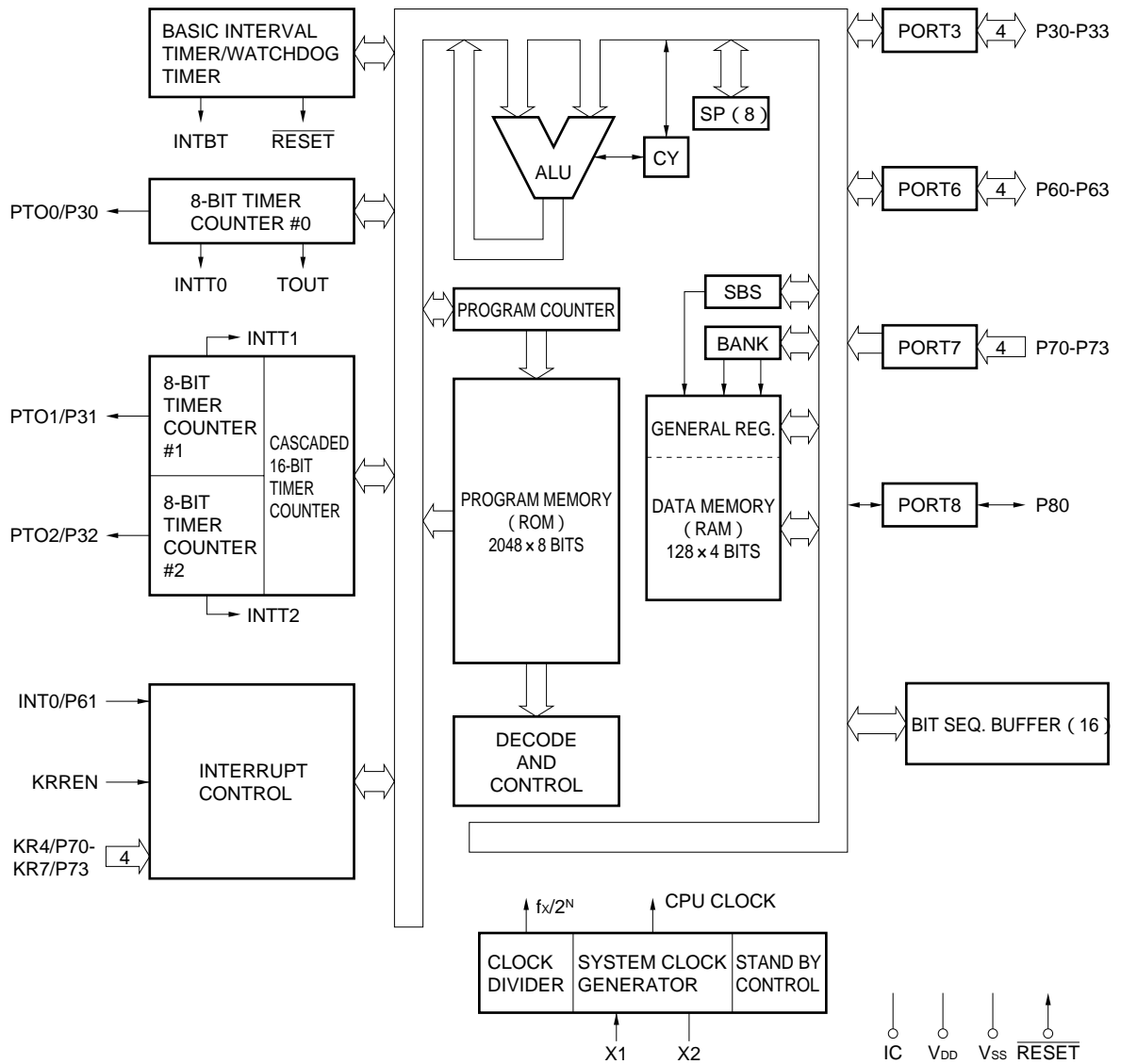
項 目		機 能	
命令実行時間		<ul style="list-style-type: none"> <li>・ 0.95, 1.91, 3.81, 15.3 <math>\mu</math>s (4.19 MHz動作時)</li> <li>・ 0.67, 1.33, 2.67, 10.7 <math>\mu</math>s (6.0 MHz動作時)</li> </ul>	
内蔵メモリ	マスクROM	2048 × 8 ビット (0000H-07FFH)	
	RAM	128 × 4 ビット (000H-07FH)	
システム・クロック発振回路		クリスタル/セラミック発振回路	
汎用レジスタ		<ul style="list-style-type: none"> <li>・ 4 ビット操作時: 8 個 × 4 バンク</li> <li>・ 8 ビット操作時: 4 個 × 4 バンク</li> </ul>	
入出力ポート	CMOS入力	4 本	マスク・オプションによるプルアップ抵抗内蔵可能
	CMOS入出力	9 本	ソフトウェアによる内蔵プルアップ抵抗の接続指定可能
	合計	13本	
タイマ		4チャンネル <ul style="list-style-type: none"> <li>・ 8ビット・タイマ・カウンタ: 3チャンネル (16ビット・タイマ・カウンタとして使用可能)</li> <li>・ ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ: 1チャンネル</li> </ul>	
ビット・シーケンシャル・バッファ		16ビット	
ベクタ割り込み		外部: 1本, 内部: 4本	
テスト入力		外部: 1本 (キー・リターン・リセット機能あり)	
スタンバイ機能		STOPモード/HALTモード	
動作周囲温度		$T_A = -40 \sim +85$	
電源電圧		$V_{DD} = 1.8 \sim 6.0$ V	
パッケージ		<ul style="list-style-type: none"> <li>・ 20ピン・プラスチックSOP (300 mil, 1.27 mmピッチ)</li> <li>・ 20ピン・プラスチック・シュリンクSOP (300 mil, 0.65 mmピッチ)</li> </ul>	

## 1.2 オーダ情報

オーダ名称	パッケージ
$\mu$ PD754202GS-x x x-BA5	20ピン・プラスチックSOP (300 mil, 1.27 mmピッチ)
$\mu$ PD754202GS-x x x-GJG	20ピン・プラスチック・シュリンクSOP (300 mil, 0.65 mmピッチ)

**備考** x x xはROMコード番号です。

### 1.3 ブロック図





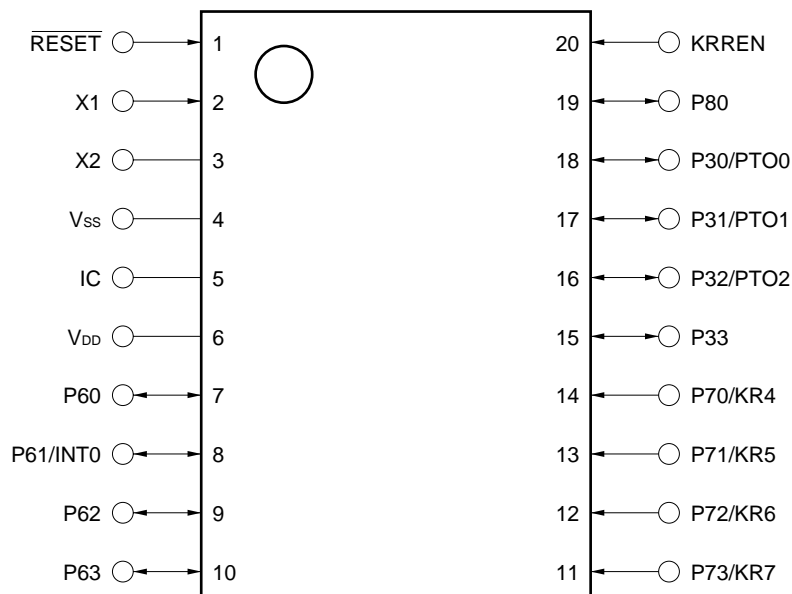
## 1.4 端子接続図 (Top View)

- ・ 20ピン・プラスチックSOP (300 mil, 1.27 mm ピッチ)

μ PD754202GS-XXX-BA5

- ・ 20ピン・プラスチック・シュリンクSOP (300 mil, 0.65 mm ピッチ)

μ PD754202GS-XXX-GJG



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

### 端子名称

P30-P33	: Port 3
P60-P63	: Port 6
P70-P73	: Port 7
P80	: Port 8
KR4-KR7	: Key Return 4-7
INT0	: External Vectored Interrupt0
PTO0-PTO2	: Programmable Timer Output0-2
KRREN	: Key Return Reset Enable
X1, X2	: System Clock ( Crystal/Ceramic )
IC	: Internally Connected
$\overline{\text{RESET}}$	: Reset
V <sub>ss</sub>	: Ground
V <sub>DD</sub>	: Positive Power Supply

## 第2章 端子機能

### 2.1 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</sup>
P30	入出力	PTO0	プログラマブル4ビット入出力ポート (PORT3)。 ビット単位で入力/出力指定可能。 4ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。	×	入力	E-B
P31		PTO1				
P32		PTO2				
P33		-				
P60	入出力	-	プログラマブル4ビット入出力ポート (PORT6)。 ビット単位で入力/出力指定可能。 4ビット単位で、ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。 P61/INT0はノイズ除去回路を選択可能。	×	入力	Ⓕ-A
P61		INT0				
P62		-				
P63		-				
★ P70	入力	KR4	4ビット入力ポート (PORT7)。 ビット単位でプルアップ抵抗を内蔵可能 (マスク・オプション)。	×	入力	Ⓑ-A
P71		KR5				
P72		KR6				
P73		KR7				
★ P80	入出力	-	1ビット入出力ポート (PORT8)。 ソフトウェアにより内蔵プルアップ抵抗の接続を指定可能。	×	入力	Ⓕ-A

注 印はシュミット・トリガ入力を示します。

表2-2 ポート端子以外の端子機能一覧

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
PTO0	出 力	P30	タイマ・カウンタ出力。		入 力	E-B
PTO1		P31				
PTO2		P32				
★ INT0	入 力	P61	エッジ検出ベクタ割り込み入力 (検出エッジ選択可能)。 ノイズ除去回路を選択可能。	ノイズ除去回路付き/ 非同期選択可	入 力	Ⓕ-A
KR4-KR7	入 力	P70-P73	立ち下がりエッジ検出テストブル入力。		入 力	Ⓖ-A
★ KRREN	入 力	-	キー・リターン・リセット・イネーブル。 STOPモードでKRREN = ハイ・レベルのとき, KRnの立ち 下がりエッジでリセット信号が発生。		入 力	Ⓖ
★ X1	入 力	-	システム・クロック発振用クリスタル/セラミック接続端 子。外部クロックの場合, X1へ入力しX2にその逆相を入力。		-	-
X2	-					
$\overline{\text{RESET}}$	入 力	-	システム・リセット入力(ロウ・レベル・アクティブ)。 ブ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 印はシュミット・トリガ入力を示します。

## 2.2 端子機能の説明

### 2.2.1 P30-P33 (PORT3) ...PTO0-PTO2兼用入出力 P60-P63 (PORT6) ...INT0兼用入出力 P80 (PORT8) ...入出力

出力ラッチ付き4ビットの入出力ポート(ポート3, 6), および出力ラッチ付き1ビットの入出力ポート(ポート8)です。

ポート3, 6は入出力ポートの機能のほかに, 次のような機能を兼用しています。

ポート3: タイマ・カウンタ出力(PTO0-PTO2)

ポート6: ベクタ割り込み入力(INT0)

ポート3, 6の入出力のモード選択はポート・モード・レジスタ・グループA (PMGA), ポート8の入出力のモード選択はポート・モード・レジスタ・グループC (PMGC)により設定します。ポート3, 6は, ビット単位で設定できます。

また, ポート3, 6, 8はソフトウェアにより内蔵プルアップ抵抗の接続を指定することができます。プルアップ抵抗指定レジスタ(POGA, POGB)を操作して行います。ポート3, 6は4ビット単位で指定します。ポート8は1ビット単位で指定します。

ポート3, 6は4ビット単位またはビット単位での入出力が可能です。8ビット単位での操作はできません。

RESET信号の発生により, 入力モードになります。

### 2.2.2 P70-P73 (PORT7) ...KR4-KR7兼用入力

4ビットの入力ポートです。

入力ポートの機能のほかにキー割り込み入力(KR4-KR7)の機能を兼用しています。

各端子は兼用端子の動作にかかわらず, 常に入力となっています。また, ノイズによる誤動作を防ぐため, 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, ビット単位でプルアップ抵抗の内蔵を指定できます。

### 2.2.3 PTO0-PTO2...ポート3兼用出力

タイマ・カウンタ0-2の出力端子です。方形波パルスを出力します。タイマ・カウンタの信号を出力する場合は, 出力ラッチを“0”にクリアし, ポート・モード・レジスタ・グループA (PMGA)のポート3のビットを出力モード“1”にセットします。

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

詳細は5.4.2(3)タイマ・カウンタ動作(8ビット時)を参照してください。

## 2.2.4 INTO...ポート6兼用入力

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

### (1) INTO (IM0のビット0, 1)

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

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

$\overline{\text{RESET}}$ 信号発生により、IM0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せん。

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

INT0は、シュミット・トリガ入力になっています。

## 2.2.5 KR4-KR7...ポート7兼用入力

キー割り込み入力端子です。KR4-KR7は、並列立ち下がりエッジ検出割り込み入力端子です。

エッジ検出モード・レジスタ (IM2) に従って割り込みソースを“KR4-KR7”に指定できます。

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

## 2.2.6 KRREN

キー・リターン・リセット機能の選択端子です。常に入力となっています。

KRREN端子がハイ・レベルでSTOPモードのときは、KR4/P70-KR7/P73端子の立ち下がり入力により、システム・リセットが発生します。また、このときSTOPモードを解除します。

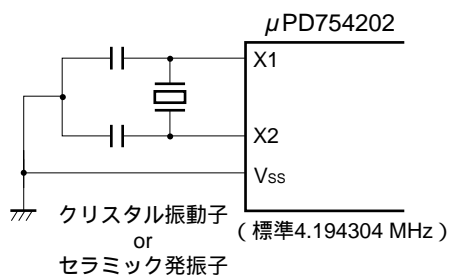
KRREN端子がロウ・レベルのときは、KR4/P70-KR7/P73端子は通常の入力端子またはスタンバイの解除となります。

## 2.2.7 X1, X2

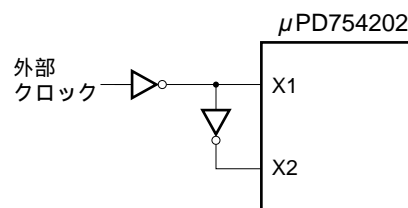
システム・クロック発振用クリスタル/セラミック接続端子です。

外部クロックを入力することも可能です。

(a) クリスタル/セラミック発振



(b) 外部クロック



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

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します。

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

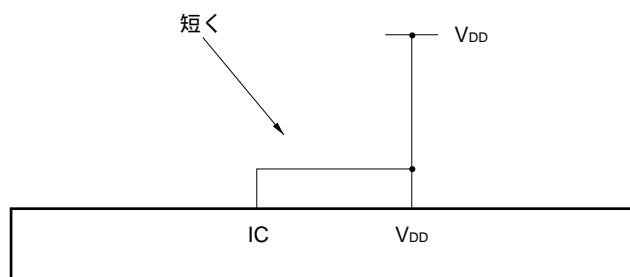
- ★ マスク・オプションにより、プルアップ抵抗の内蔵を指定できます。

## 2.2.9 IC

IC (Internally Connected) 端子は、当社出荷時に  $\mu\text{PD754202}$  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  $V_{\text{DD}}$  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  $V_{\text{DD}}$  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  $V_{\text{DD}}$  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  $V_{\text{DD}}$  に直接接続してください



## 2.2.10 $V_{\text{DD}}$

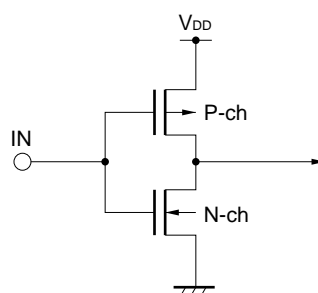
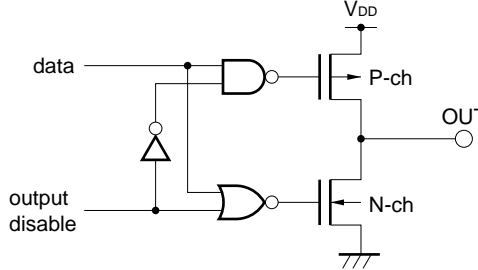
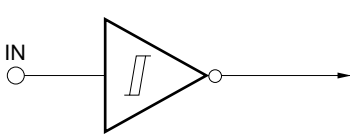
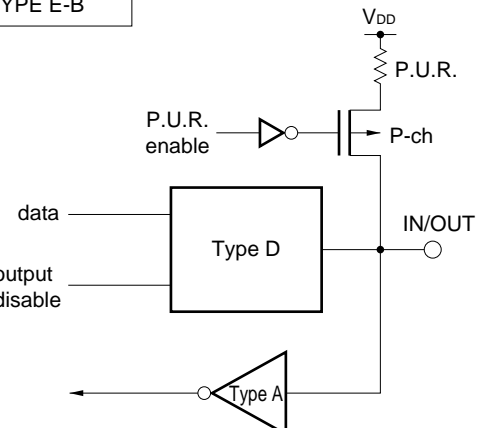
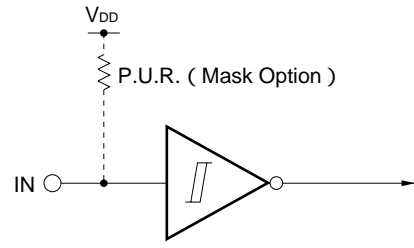
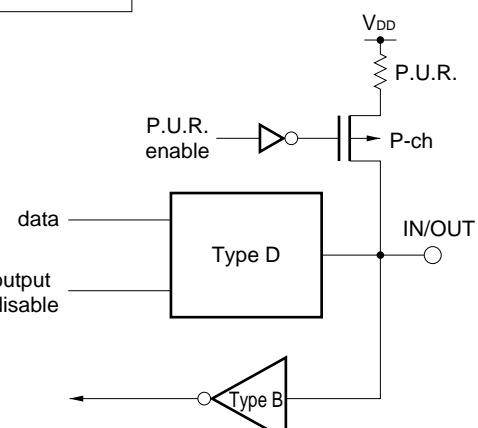
正電源供給端子です。

## 2.2.11 $V_{\text{SS}}$

GND電位です。

## 2.3 端子の入出力回路

μPD754202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。

<p>TYPE A</p>  <p>CMOS規格の入力バッファになっています。</p>	<p>TYPE D</p>  <p>出力ハイ・インピーダンス（P-ch, N-chともにオフ）とすることができるプッシュプル出力となっています。</p>
<p>TYPE B</p>  <p>ヒステリシス特性を持つシュミット・トリガ入力となっています。</p>	<p>TYPE E-B</p>  <p>P.U.R. : Pull-Up Resistor</p>
<p>TYPE B-A</p>  <p>P.U.R. : Pull-Up Resistor</p>	<p>TYPE F-A</p>  <p>P.U.R. : Pull-Up Resistor</p>

## 2.4 未使用端子の処理について

表2-3 未使用端子の処理一覧

端 子	推奨接続方法
P30/PTO0	入力状態：個別に抵抗を介して、V <sub>SS</sub> またはV <sub>DD</sub> に接続 出力状態：オープン
P31/PTO1	
P32/PTO2	
P33	
P60	
P61/INT0	
P62	
P63	
P70/KR4	個別に抵抗を介して、V <sub>DD</sub> に接続
P71/KR5	
P72/KR6	
P73/KR7	
P80	入力状態：個別に抵抗を介して、V <sub>SS</sub> またはV <sub>DD</sub> に接続 出力状態：オープン
KRREN	V <sub>DD</sub> に接続すると、STOPモード時にKR <sub>n</sub> 端子の立ち下がりエッジで内部リセット信号を発生します。 V <sub>SS</sub> に接続すると、STOPモード時にKR <sub>n</sub> 端子の立ち下がりエッジを検出しても内部リセット信号は発生しません。
IC	V <sub>DD</sub> に直接接続



〔メ モ〕

## 第3章 アーキテクチャの特徴とメモリ・マップ

μPD754202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

μPD754202ではデータ・メモリ空間のメモリ・バンク0の000H-07FH番地にスタティックRAMが内蔵されています。また、周辺ハードウェア(入出力ポートやタイマなど)がメモリ・バンク15のF80H-FFFH番地に割り付けられています。

μPD754202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。

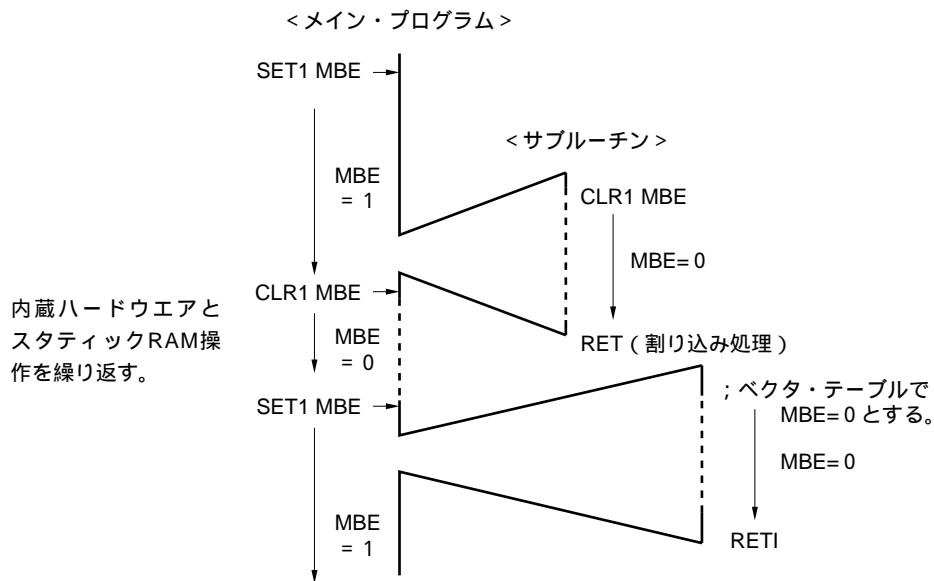
- ・メモリ・バンク許可フラグ(MBE)
- ・メモリ・バンク選択レジスタ(MBS)

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, 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	通常のプロダラム処理	

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15を選択する。

```
SET1      MBE      ; MBE  1
```

```
SEL       MB15     ; MBS  15
```

### 3.1.2 データ・メモリのアドレッシング・モード

μPD754202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にまとめたとように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-FFFH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 057H.2 ; 57H番地ビット2
```

```
FLAG3 EQU 077H.0 ; 77H番地ビット0
```

```
SET1 MBE ; MBE 1
```

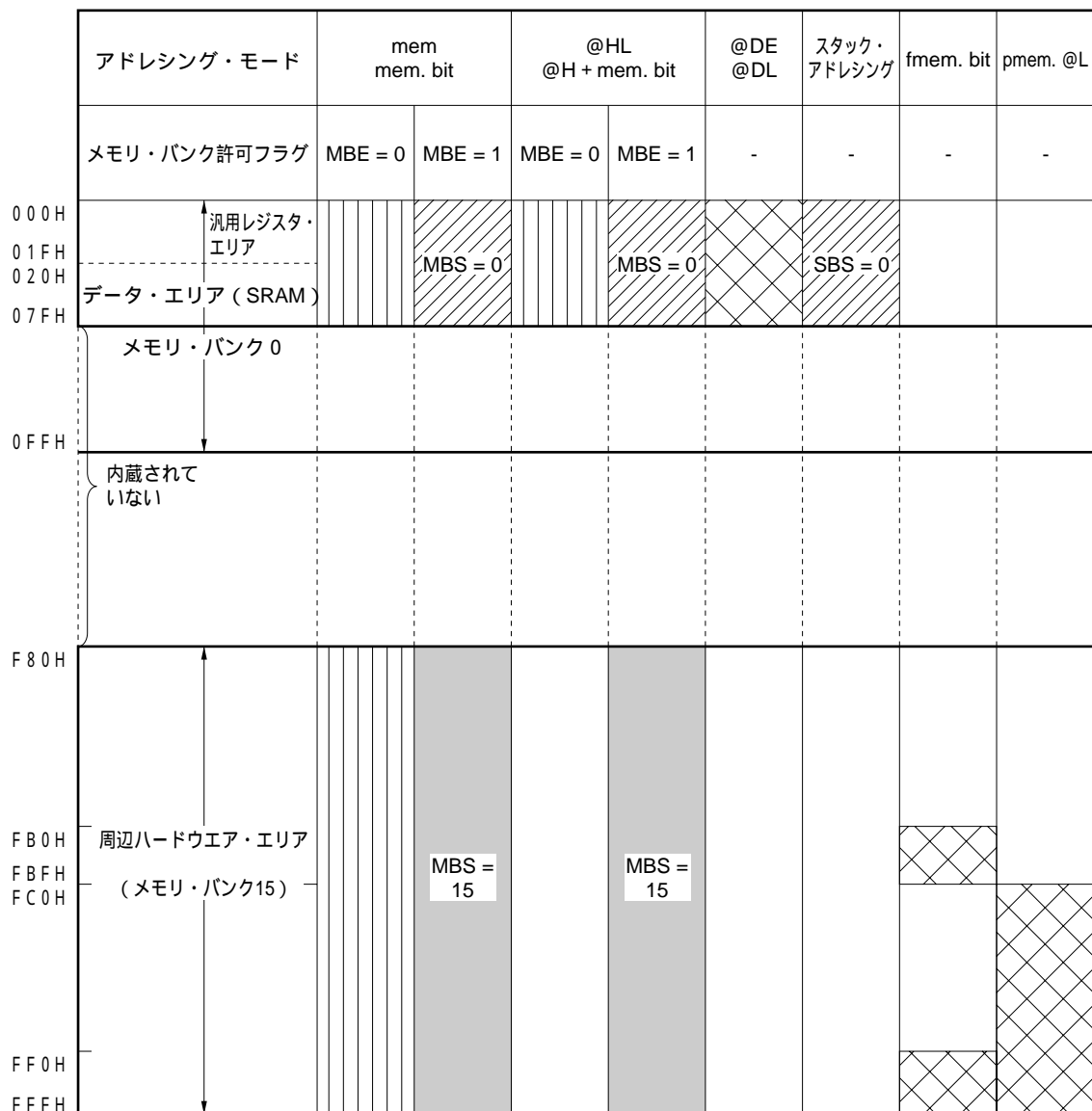
```
SEL MB0 ; MBS 0
```

```
SET1 FLAG1 ; FLAG1 1
```

```
CLR1 FLAG2 ; FLAG2 0
```

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

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



備考 - : don't care

表3 - 1 アドレッシング・モード

アドレッシング・モード	表現形式	指定されるアドレス
1ビット 直接アドレッシング	mem.bit	MBとmemで示されるアドレスの, bitで示されるビット。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB=MBS
4ビット 直接アドレッシング	mem	MBとmemで示されるアドレス。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB = MBS
8ビット 直接アドレッシング		MBとmemで示されるアドレス (memは偶数アドレス)。 ・ MBE = 0のとき mem = 00H-7FHのとき : MB = 0 mem = 80H-FFHのとき : MB = 15 ・ MBE = 1のとき : MB = MBS
4ビット・レジスタ 間接アドレッシング	@HL	MBとHLで示されるアドレス。 ただし, MB = MBE・MBS
	@HL + @HL -	MBとHLで示されるアドレス。ただし, MB = MBE・MBS HL + は, アドレッシングのあとLレジスタを自動的にインクリメント。 HL - は, アドレッシングのあとLレジスタを自動的にデクリメント。
	@DE	メモリ・バンク0の, DEで示されるアドレス。
	@DL	メモリ・バンク0の, DLで示されるアドレス。
8ビット・レジスタ 間接アドレッシング	@HL	MBとHLで示されるアドレス (Lレジスタの内容は偶数)。 ただし, MB = MBE・MBS
ビット操作 アドレッシング	fmem.bit	fmemで示されるアドレスの, bitで示されるビット。 fmem = FB0H-FBFH ( 割り込み関係のハードウェア ) FF0H-FFFH ( I/Oポート )
	pmem.@L	pmemの上位10ビットとLレジスタの上位2ビットで示されるアドレスの, Lレジスタの下位2ビットで示されるビット。 ただし, pmem = FC0H-FFFH
	@H + mem.bit	MBとHそしてmemの下位4ビットで示されるアドレスの, bitで示される ビット。 ただし, MB = MBE・MBS
スタック・アドレッシング	-	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-FFFFH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
SEL MB0
MOV D, #DATA1 SHR 4
MOV HL, #DATA2 AND 0FFH ; HL 17H
LOOP: MOV A, @DL ; A (DL)
XCH A, @HL ; A (HL)
DECS L ; L L-1
BR LOOP

```

データ・ポインタとしてHLレジスタ・ペアを使用するアドレッシング・モードは、データの転送、演算、比較、入出力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。

```

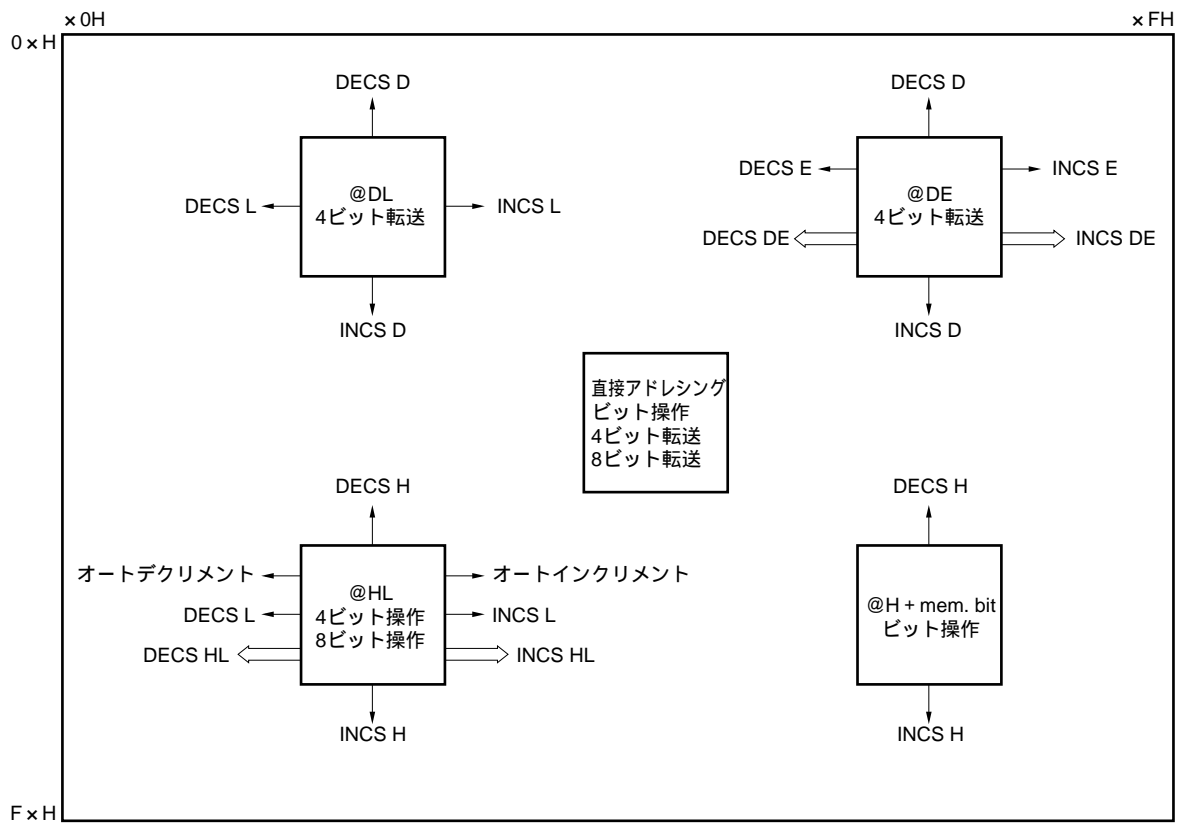
DATA1 EQU 57H
DATA2 EQU 67H
      SET1 MBE
      SEL MB0
      MOV D, #DATA1 SHR 4
      MOV HL, #DATA2 AND 0FFH
LOOP : MOV A, @DL
      SKE A, @HL ; A = (HL) ?
      BR NO ; NO
      DECS L ; YES, L L - 1
      BR LOOP
    
```

2 . 004H-07FHのデータ・メモリを0クリアする。

```

CLR1 RBE
CLR1 MBE
MOV XA, #00H
MOV HL, #04H
LOOP : MOV @HL, A ; (HL) A
      INCS L ; L L + 1
      BR LOOP
      INCS H ; H H + 1
      NOP
      SKE H, #08H
      BR LOOP
    
```

図3 - 3 スタティックRAMのアドレス更新方法



## (5) 8ビット・レジスタ間接アドレッシング (@HL)

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

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

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

- ★ 例1 . タイマ・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 . 004H-07FH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 L
      INCS L
      BR LOOP
      INCS H
      NOP
      SKE H, #08H
      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、ビット転送など）を行うためのアドレッシング・モードです。

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

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

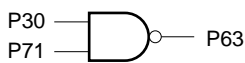
(a) 特定アドレス・ビット直接アドレッシング (fmem.bit)

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

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

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

2．P30とP71の端子がともに1であれば、P63をリセットする。



```
( i )      SET1  CY          ; CY  1
           AND1  CY , PORT3.0 ; CY  P30
           AND1  CY , PORT7.1 ; CY  P71
           SKT   CY          ; CY=1?
           BR    SETP
           CLR1  PORT6.3      ; P63  0
           :
           SETP : SET1  PORT6.3 ; P63  1
```

```

        ⋮
(ii)   SKT   PORT3.0      ; P30=1?
        BR    SETP
        SKT   PORT7.1      ; P71=1?
        BR    SETP
        CLR1  PORT6.3      ; P63  0
        ⋮
        SETP : SET1  PORT6.3      ; P63  1
    
```

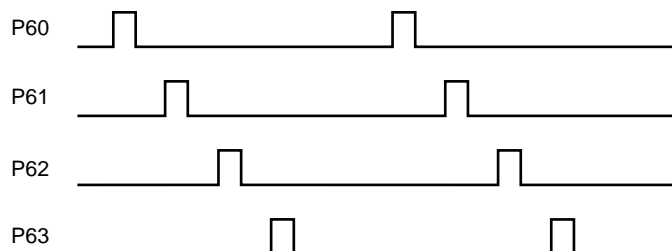
**(b) 特定アドレス・ビット・レジスタ間接アドレッシング (pmem.@L)**

このアドレッシング・モードは周辺ハードウェアのうち入出力ポート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6の各ビットに順にパルス出力する。



```

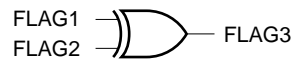
LOOP2 : MOV   L, #0
LOOP1 : SET1  PORT6.@L ; ポート6のビット(L1-0)  1
        CLR1  PORT6.@L ; ポート6のビット(L1-0)  0
        INCS  L
        SKE   L, #4H
        BR    LOOP1
        BR    LOOP2
    
```

(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
CLR1 CY ; CY 0
OR1 CY, @H + FLAG1 ; CY CY FLAG1
XOR1 CY, @H + FLAG2 ; CY CY ∨ FLAG2
SET1 @H + FLAG3 ; FLAG3 1
SKT CY ; CY=1?
CLR1 @H + FLAG3 ; FLAG3 0
    
```

**(7) スタック・アドレッシング**

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

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

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

**例1** . サブルーチン処理でレジスタを退避 / 復帰する。

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

**2** . HLレジスタ・ペアの内容をDEレジスタ・ペアに転送する。

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

**3** . [XABC] のレジスタで示されるアドレスへ分岐する。

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

### 3.2 汎用レジスタのバンク構成

μPD754202は、X, A, B, C, D, E, H, Lの8つの汎用レジスタを1バンクとして、4つのレジスタ・バンクを内蔵しています。この汎用レジスタ・エリアはデータ・メモリのメモリ・バンク0の00H-1FH番地にマッピングされています（図3-5 汎用レジスタの構成（4ビット処理の場合）参照）。この汎用レジスタのバンクを指定するためにレジスタ・バンク許可フラグ（RBE）とレジスタ・バンク選択レジスタ（RBS）が内蔵されています。RBSはレジスタ・バンクを選択するためのレジスタで、RBEはRBSで選択されたレジスタ・バンクを有効とするか否かを決定するフラグです。命令実行の際に有効となるレジスタ・バンク（RB）は、次のようになります。

$$RB = RBE \cdot RBS$$

表3-2 RBE, RBSと選択されるレジスタ・バンク

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

└─ 0に固定

備考 x : don't care

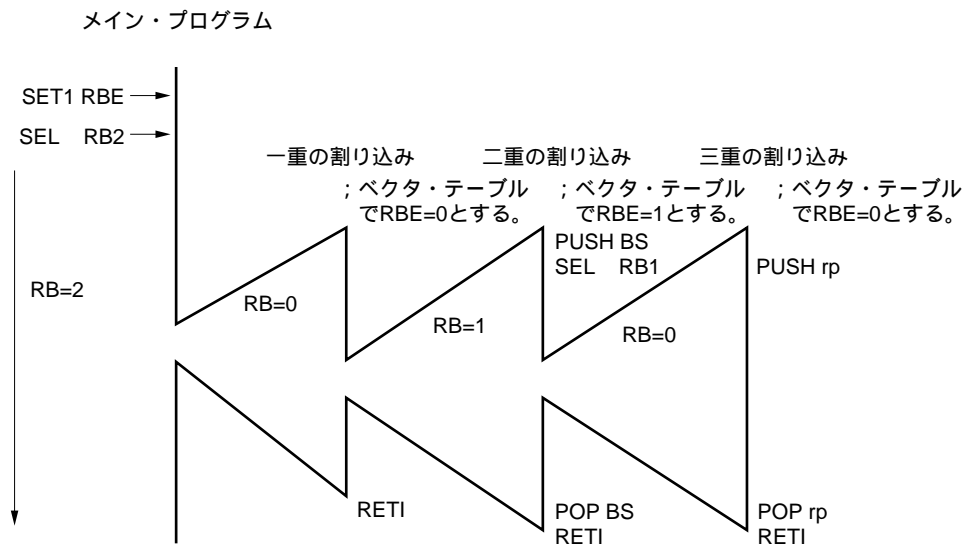
RBEは、サブルーチン処理時に自動的に退避／復帰されますので、サブルーチン処理中に自由に設定することができます。また、割り込み処理時は、自動的に退避／復帰されるとともに、割り込みベクタ・テーブルの設定によって、割り込み処理開始と同時に、割り込み処理中のRBEを設定することができます。したがって、表3-3に示すように、通常処理と割り込み処理で、レジスタ・バンクを使い分ければ、一重割り込みでは、汎用レジスタの退避／復帰は不要、二重割り込みでは、RBSの退避／復帰のみとなり、割り込み処理の高速化が図れます。

表3-3 通常ルーチンと割り込みルーチンでのレジスタ・バンクの使い分けの例

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

μPD754202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 (PC10-8 + DE)ROM, 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のカウント・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。

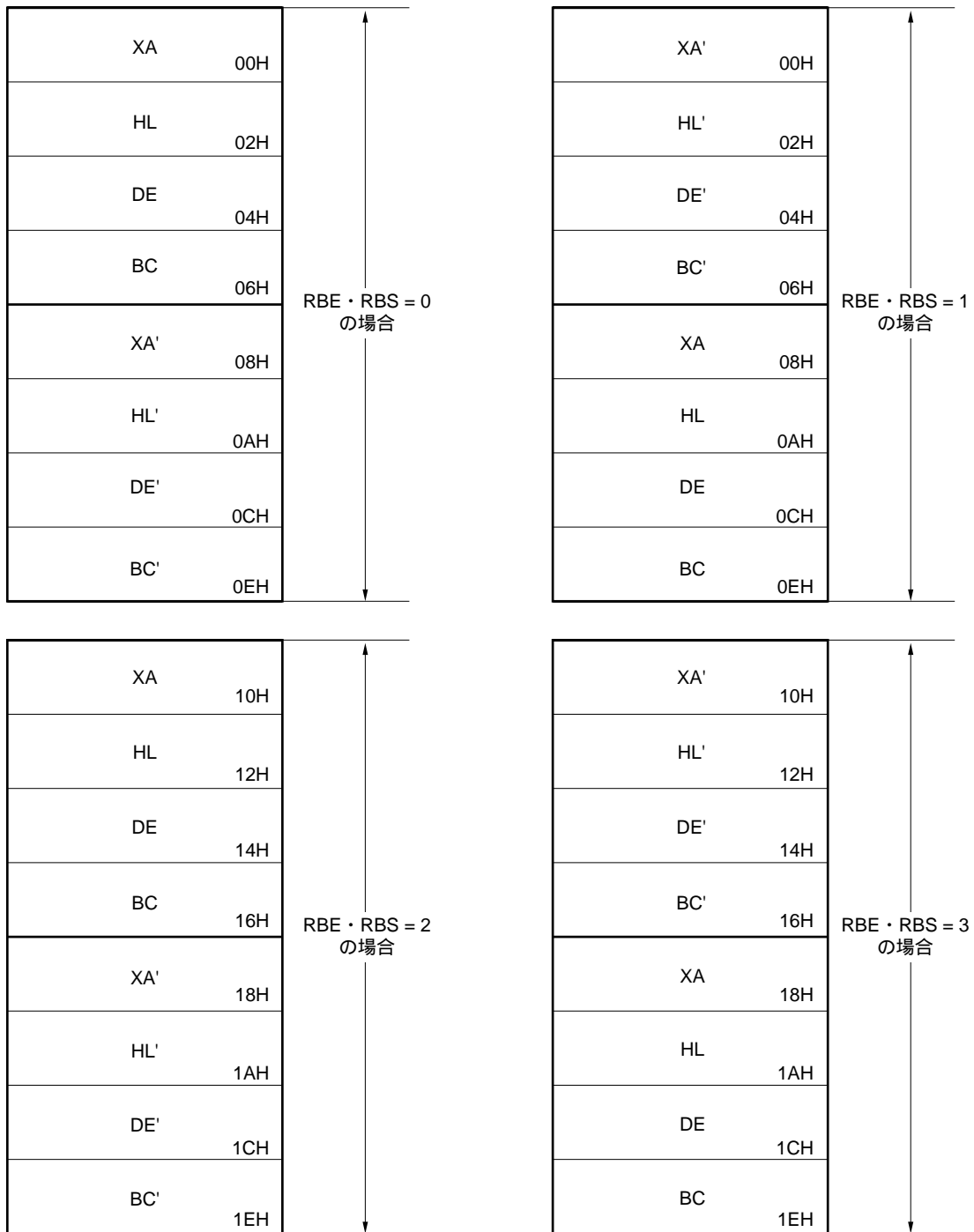
```

      CLR1  MBE
NO :   MOV  XA, T0      ; カウント・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

μPD754202は、図3 - 2に示したようにデータ・メモリ空間上のF80H-FFFH番地に入出力ポートやタイマなどの周辺ハードウェアをマッピングしたメモリ・マップトI/Oを採用しています。このため、周辺ハードウェアを制御する特別な命令はなく、すべてメモリ操作命令により制御します（一部、プログラムをわかりやすくするためのハードウェア制御の二モニックが用意してあります）。

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

表3 - 4 周辺ハードウェア操作時に適用可能なアドレッシング・モード

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

```

例 CLR1    MBE        ; MBE=0
    SET1    TM0.3     ; タイマ0のスタート
    EI      IE0       ; INT0の許可
    DI      IET1      ; INTT1の禁止
    SKTCLR  IRQ2      ; INT2要求フラグのテスト・アンド・クリア
    SET1    PORT3.@L ; ポート3のセット
    IN      A, PORT6  ; A ポート6
    
```

図3 - 7に $\mu$ PD754202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 I/Oマップ (1/8)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
F80H	スタック・ポインタ (SP)				R/W	-	-		-	ビット0は0に固定
★ F82H	レジスタ・バンク選択レジスタ (RBS)				R	-			-	注1
F83H	バンク選択レジスタ (BS) 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	-	-		-	
F88H	タイマ・カウンタ2ハイ・レベル期間設定用				R/W	-	-		-	
	モジュロ・レジスタ (TMOD2H)									
F8AH	非実装									
F8BH	WDTM <sup>注2</sup>	-	-	-	W		-	-	mem. bit	
F8CH } F8FH	非実装									

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 (n = 0, 15), SEL RBn (n = 0-3) 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) ; 一度セットすると命令ではクリアできません。

図3 - 7 I/Oマップ(2/8)

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		
F90H	タイマ・カウンタ2モード・レジスタ				R/W	(W)	-		-	ビット3のみ ビット操作可能
	(TM2)					-	-		-	
F92H	TOE2	REMC	NRZB	NRZ	R/W				-	ビット3は Wのみ可能
	タイマ・カウンタ2コントロール・レジスタ(TC2)					-	-		-	ビット3には 0のみ記述可能
F94H	タイマ・カウンタ2カウント・レジスタ				R	-	-		-	
	(T2)									
F96H	タイマ・カウンタ2モジュロ・レジスタ				R/W	-	-		-	
	(TMOD2)									
F98H ┆ F9FH	非実装									



図3 - 7 I/Oマップ (3/8)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
FA0H	タイマ・カウンタ0モード・レジスタ (TM0)				R/W	(W)	-		mem. bit	ビット3のみ ビット操作可能
						-	-		-	
FA2H	TOE0 <sup>注1</sup>	-	-	-	W		-	-	mem. bit	
FA3H	非実装									
FA4H	タイマ・カウンタ0カウント・レジスタ (T0)				R	-	-		-	
FA6H	タイマ・カウンタ0モジュロ・レジスタ (TMOD0)				R/W	-	-		-	
FA8H	タイマ・カウンタ1モード・レジスタ (TM1)				R/W	(W)	-		mem. bit	ビット3のみ ビット操作可能
						-	-		-	
FAAH	TOE1 <sup>注2</sup>	-	-	-	W		-	-	mem. bit	
FABH	非実装									
FACH	タイマ・カウンタ1カウント・レジスタ (T1)				R	-	-		-	
FAEH	タイマ・カウンタ1モジュロ・レジスタ (TMOD1)				R/W	-	-		-	

注1 . TOE0 : タイマ・カウンタ出力許可フラグ (チャンネル0) (W)

2 . TOE1 : タイマ・カウンタ出力許可フラグ (チャンネル1) (W)

図3 - 7 I/Oマップ (4/8)

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	(RW)	(RW)	(R)	fmem. bit	8ビット操作 はRのみ可能	
	プログラム・ステータス・ワード (PSW)					注2	-				
	CY <sup>注1</sup>	SK2 <sup>注1</sup>	SK1 <sup>注1</sup>	SK0 <sup>注1</sup>							
FB2H	割り込みプライオリティ選択レジスタ (IPS)				R/W	-		-		注3	
FB3H	プロセッサ・クロック・コントロール・レジスタ (PCC)				R/W	-		-		注4	
FB4H	INT0エッジ検出モード・レジスタ (IM0)				R/W	-		-	-		
FB5H	非実装										
FB6H	INT2エッジ検出モード・レジスタ (IM2) <sup>注5</sup>				R/W	-		-	-		
FB7H	非実装										
FB8H	INTAレジスタ (INTA)				R/W			-	fmem. bit	ビット操作 は予約語の みで可能	
	-	-	IEBT	IRQBT							
FB9H } FBBH	非実装										
FBCH	INTEレジスタ (INTE)				R/W			-	fmem. bit	ビット操作 は予約語の みで可能	
	IET1	IRQT1	IET0	IRQT0							
FBDH	INTFレジスタ (INTF)				R/W			-			
	IET2	IRQT2	-	-							
FBEH	INTGレジスタ (INTG)				R/W			-	fmem. bit	ビット操作 は予約語の みで可能	
	-	-	IE0	IRQ0							
FBFH	INTHレジスタ (INTH)				R/W			-	fmem. bit		ビット操作 は予約語の みで可能
	-	-	IE2	IRQ2							

備考1 . IE x x x は割り込み許可フラグ

2 . IRQ x x x は割り込み要求フラグ

注1 . 予約語としては登録されていません。

2 . CYへの書き込みは、CY操作命令で行ってください。

3 . IME (ビット3) は、EI/DI命令でのみ操作可能。

4 . PCC3 (ビット3) , PCC2 (ビット2) は、STOP/HALT命令でビット操作可能。

5 . KRn端子の立ち下がリエッジを、割り込み要求フラグ (IRQ2) のセット信号として選択するレジスタです。リセット時は00Hとなっていますので、KRn 端子の立ち下がリエッジをIRQ2にするには01Hを書き込んでください。

図3 - 7 I/Oマップ (5/8)

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		
FC0H	ビット・シーケンシャル・バッファ 0 (BSB0)				R/W				mem. bit pmem. @L	
FC1H	ビット・シーケンシャル・バッファ 1 (BSB1)				R/W					
FC2H	ビット・シーケンシャル・バッファ 2 (BSB2)				R/W					
FC3H	ビット・シーケンシャル・バッファ 3 (BSB3)				R/W					
FC4H	非実装									
FC5H										
FC6H	リセット検出フラグ・レジスタ (RDF)				R/W			-	mem. bit	ビット 2, 3 のみ操作可能
	KRF	WDF	-	-						
FC7H } FCFH	非実装									

図3 - 7 I/Oマップ (6/8)

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 ┆ FDBH	非実装									
★ FDCH	PO3 <sup>注</sup>	-	-	-	R/W	-	-		-	非実装領域は、書き込みは無効となり、読み出し値は不定です。
	ブルアップ抵抗指定レジスタ・グループA (POGA)									
	-	PO6 <sup>注</sup>	-	-						
★ FDEH	-	-	-	PO8 <sup>注</sup>	R/W	-	-		-	
	ブルアップ抵抗指定レジスタ・グループB (POGB)									
	-	-	-	-						

注 予約語としては登録されていません。

図3 - 7 I/Oマップ (7/8)

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		
FE0H ┆ FE7H	非実装									
FE8H	PM33	PM32	PM31	PM30	R/W		-		-	
	ポート・モード・レジスタ・グループA (PMGA)									
	PM63 <sup>注</sup>	PM62 <sup>注</sup>	PM61 <sup>注</sup>	PM60 <sup>注</sup>						
FEAH ┆ FEDH	非実装									
★ FEEH	-	-	-	PM8 <sup>注</sup>	R/W	-	-		-	非実装領域は、書き込みは無効となり、読み出し値は不定です。
	ポート・モード・レジスタ・グループC (PMGC)									
	-	-	-	-						

注 予約語としては登録されていません。ただし、次のようにするとビット操作が可能です。

0FE9.0-0FE9.3

図3 - 7 I/Oマップ (8/8)

アドレス	ハードウェア名称 (略号)				R/W	操作可能なビット数			ビット操作 アドレッシング	備 考
	b3	b2	b1	b0		1ビット	4ビット	8ビット		
FF0H ┆ FF2H	非実装									
FF3H	ポート3 (PORT3)				R/W			-	fmem. bit pmem. @L	
FF4H ┆ FF5H	非実装									
FF6H	ポート6 (PORT6)				R/W			-	fmem. bit pmem. @L	
FF7H <sup>注1</sup>	ポート7 (PORT7) KR7      KR6      KR5      KR4				R			-		
FF8H	ポート8 (PORT8) -      -      -      P80 <sup>注2</sup>				R/W			-		
FF9H ┆ FFFH	非実装									非実装領域は、書き込みは無効となり、読み出し値は不定です。

注1 . KR4-KR7は1ビット単位での読み出し (R) のみ可能です。4ビット並列入力時はPORT7で指定します。

2 . 予約語としては登録されていません。

〔メ モ〕

## 第4章 内部CPU機能

### 4.1 Mk モードとMk モードの切り替え機能

#### 4.1.1 Mk モードとMk モードの違い

μPD754202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、16K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、16K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・サイクル分長くかかります。したがって、ソフトウェアの互換性よりRAMの使用効率や処理能力を重視する場合は、Mk モードを使用してください。

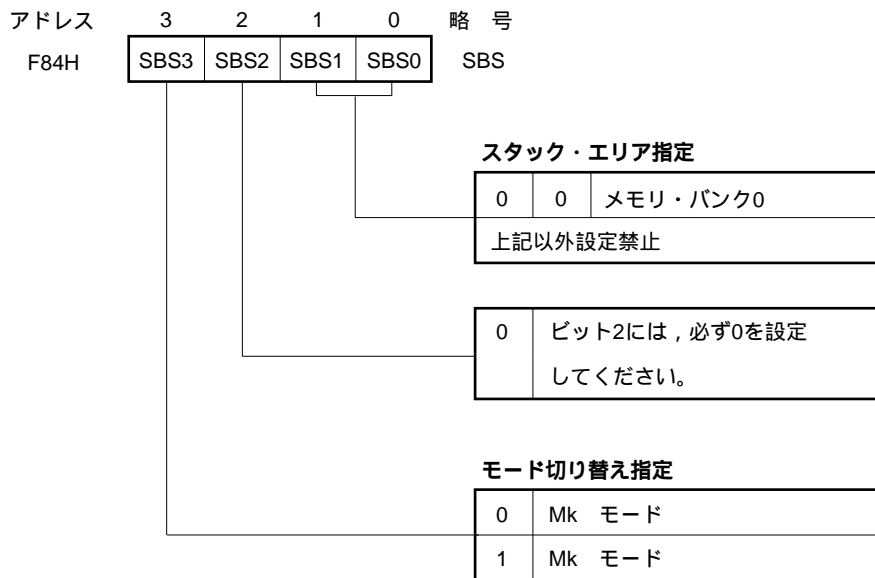


### 4.1.2 スタック・バンク選択レジスタ (SBS) の設定方法

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

スタック・バンク選択レジスタは、4ビット・メモリ操作命令により設定します。Mk モードを使用する場合は、プログラムの初期で必ずスタック・バンク選択レジスタを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) .....11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

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。

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

PC<sub>10-8</sub> (0000H)<sub>2-0</sub>, PC<sub>7-0</sub> (0001H)<sub>7-0</sub>

### 4.3 プログラム・メモリ (ROM) .....2048 × 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 \$addr1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-07FFH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

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

0002H-000DH番地

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

0020H-007FH番地

GETI命令<sup>注</sup>で参照するテーブル領域です。

**注** GETI命令は、任意の2バイト/3バイト命令、または1バイト命令2つを1バイトで実現するための命令で、プログラム・ステップ数を縮小するために使用します (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) .....128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 (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128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-07FH) として使用できます。

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

#### ・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。

#### ・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。

#### (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

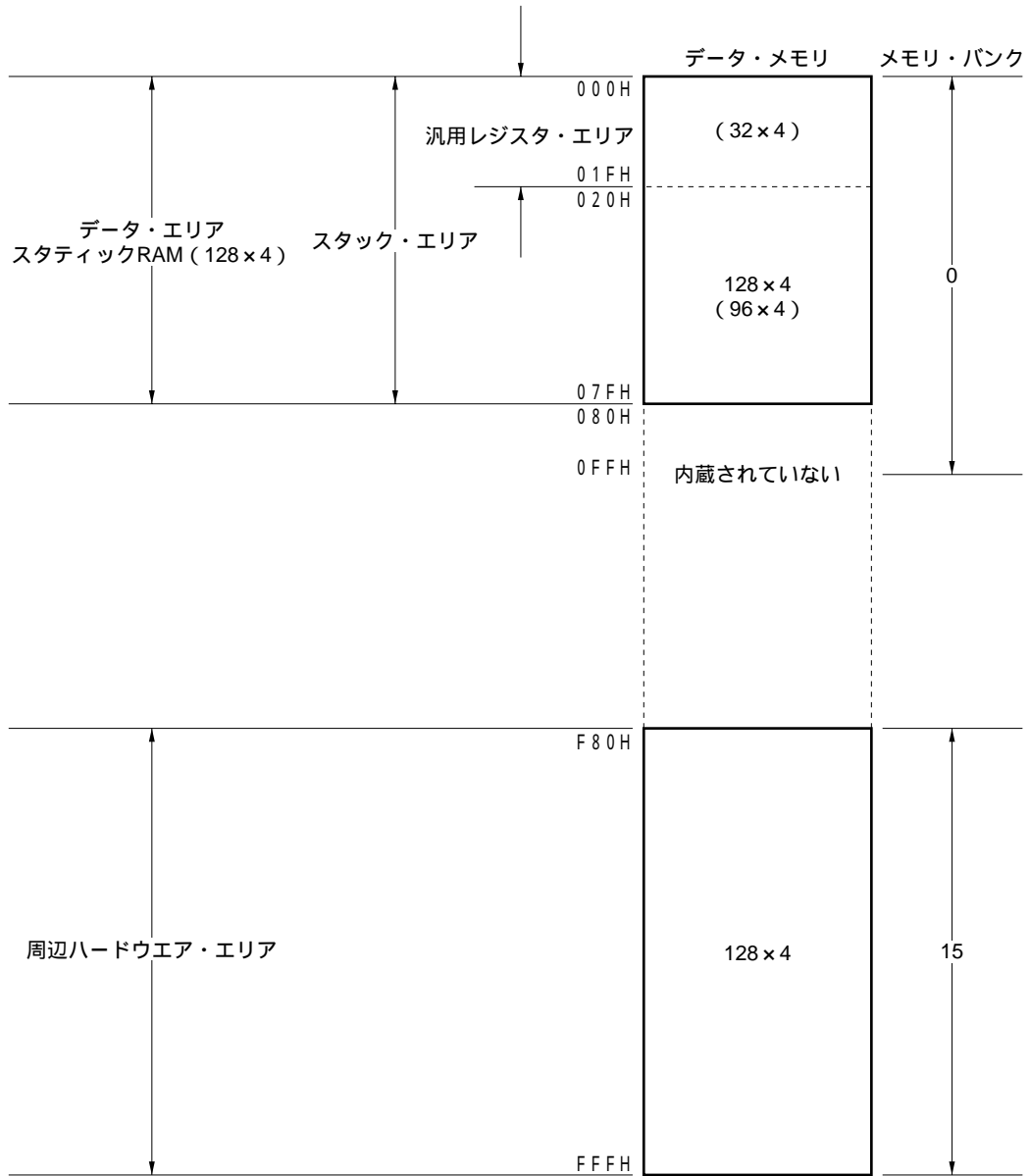
メモリ・バンクは、メモリ・バンク許可フラグ（MBE）によってバンク指定を許可したとき（MBE=1）、4ビット・メモリ・バンク選択レジスタ（MBS）によって指定されます（MBS=0, 15）。バンク指定を禁止したとき（MBS=0）、メモリ・バンクはそのときのアドレッシング・モードによってバンク0またはバンク15が自動的に指定されます。また、バンク内のアドレスは、8ビット・イミディエト・データやレジスタ・ペアなどによってアドレスされます。

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）
- ・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



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-07FH番地のRAMをクリアする。

```
SET1    MBE
SEL     MB0
MOV     XA, #00H
MOV     HL, #04H
RAMC0:  MOV    @HL, A      ; 004H-07FHをクリア注
        INCS   L          ; L  L + 1
        BR     RAMC0
        INCS   H          ; H  H + 1
        NOP
        SKE   H, #08H
        BR     RAMC0
```

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



## 4.5 汎用レジスタ..... 8 × 4 ビット × 4 バンク

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

命令実行時に有効となるレジスタ・バンク（RB）は次の式により決まります。

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

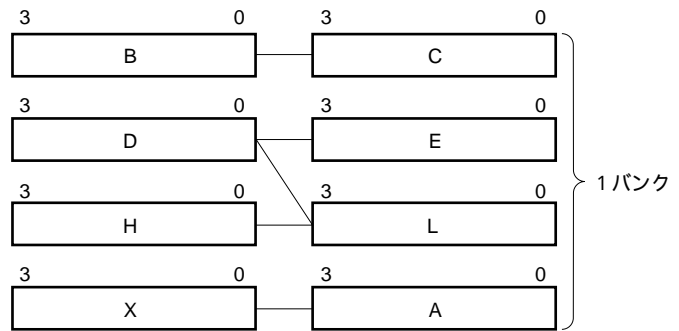
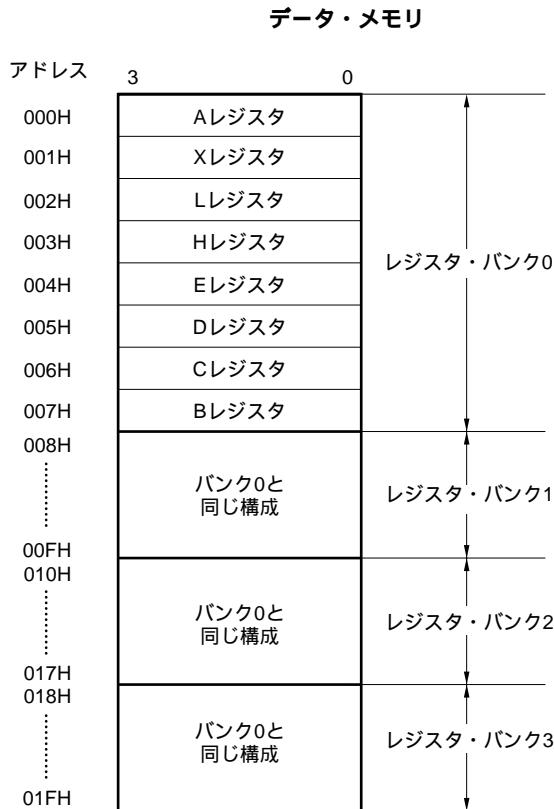
汎用レジスタは、それぞれ4ビット単位で操作されるほか、BC, DE, HL, XAがレジスタ・ペアを形成し、8ビット操作に使用されます。また、DE, HLのほかDLもレジスタ・ペアを形成し、この3組はデータ・ポインタとして使用することができます。

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 - 5 汎用レジスタの構成

図4 - 6 レジスタ・ペアの構成

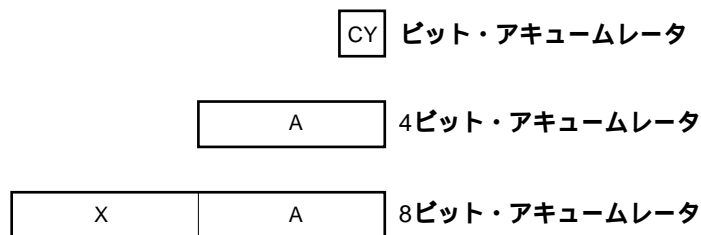


## 4.6 アキュムレータ

μPD754202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4-7 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)

μPD754202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-07FH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4-9 ~ 4-12のようになります。

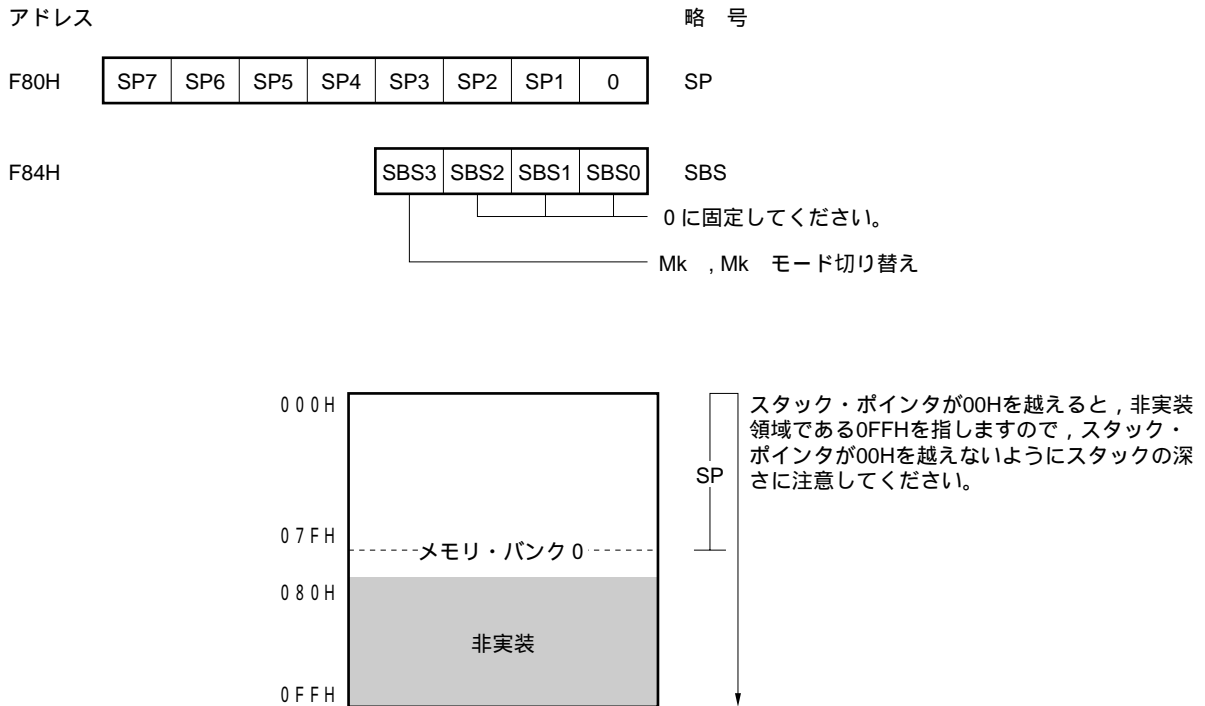
SPは、8ビット・メモリ操作命令、SBSは4ビット・メモリ操作命令によって初期値を設定し、スタック・エリアを決定します。また内容の読み出しもできます。

SPの初期値を00Hに設定すると、SBSで指定したメモリ・バンク0の最上位アドレス (07FH) 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に限られます。000H番地からさらにスタック動作を行うと、非実装領域である0FFHを指しますので、スタック・ポインタは000Hを越えないよう注意してください。

- ★  $\overline{\text{RESET}}$ 信号発生により、SPの内容は不定、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 - 8 スタック・ポインタおよびスタック・バンク選択レジスタの構成



SPのイニシャライズ例

スタック・エリアをメモリ・バンク0とし、07FH番地からスタック動作させる。

```

SEL  MB15      ;またはCLR1 MBE
MOV  A, #0
MOV  SBS, A    ;メモリ・バンク0をスタック・エリアに指定
MOV  XA, #80H
MOV  SP, XA    ;SP 80H (7FHからスタック動作)
    
```

図4-9 スタック・メモリへ退避されるデータ (Mk モード)

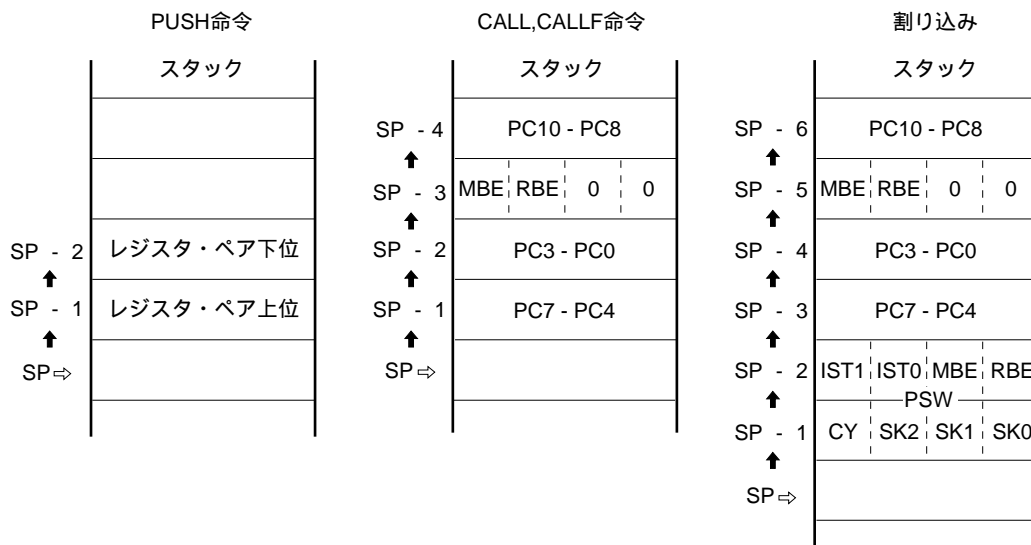


図4-10 スタック・メモリから復帰されるデータ (Mk モード)

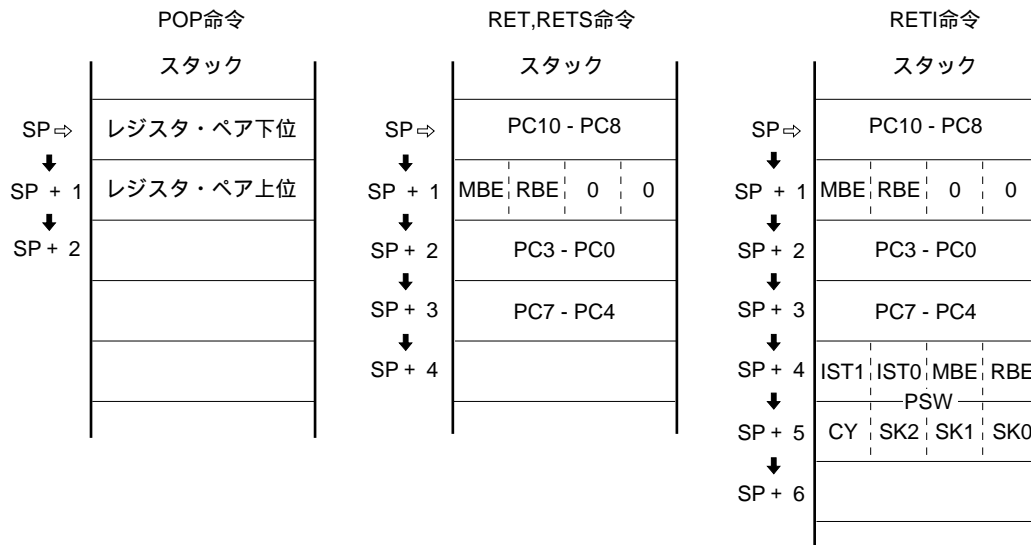


図4 - 11 スタック・メモリへ退避されるデータ (Mk モード)

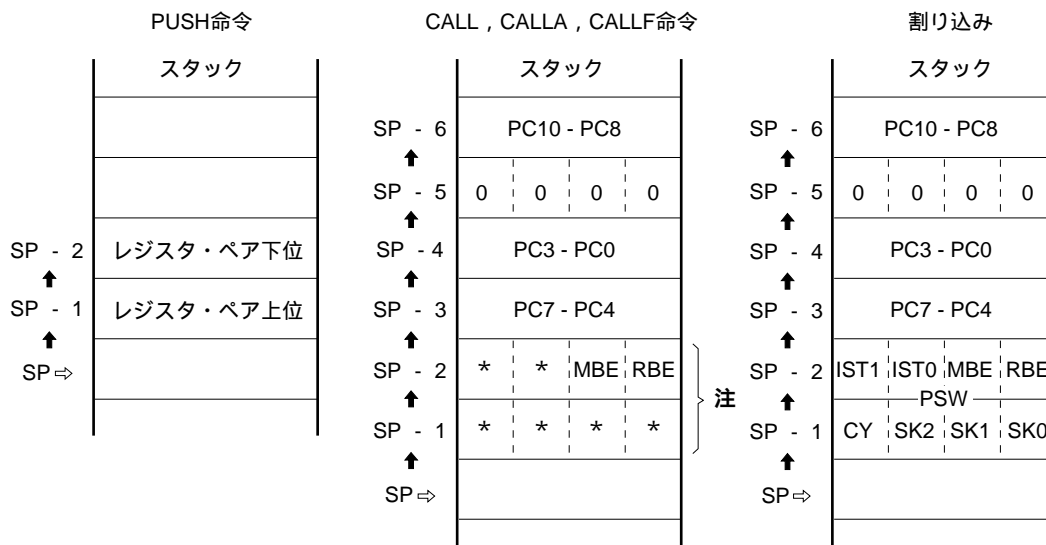
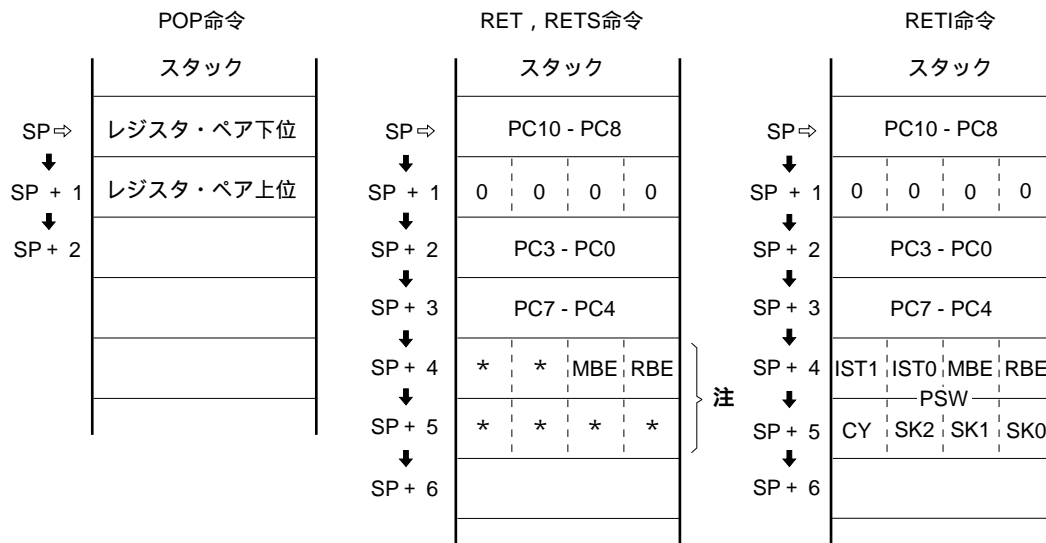


図4 - 12 スタック・メモリから復帰されるデータ (Mk モード)



注 MBE, RBE以外のPSWは退避 / 復帰しません。

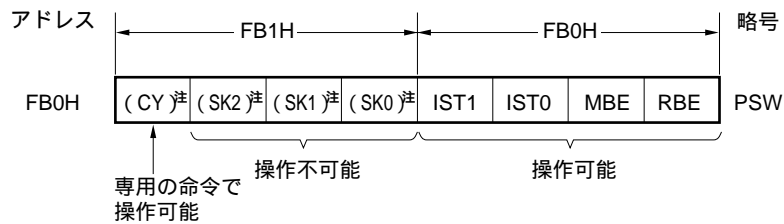
備考 \* は不定。

## 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 - 13 プログラム・ステータス・ワードの構成



注 予約語としては登録されていません。

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。

$\overline{\text{RESET}}$ 信号発生により、キャリー・フラグは不定となります。

表4-4 キャリー・フラグ操作命令

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

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

fmem.bit

pmem.@L

@H + mem.bit

**例** 3FH番地のビット3とP33のANDをとり, P60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   PORT6.0, CY    ; P60 CY
```

## (2) スキップ・フラグ (SK2, SK1, SK0)

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

ユーザがオペランドとして直接操作することはできません。

**(3) 割り込みステータス・フラグ (IST1, IST0)**

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

表4-5 割り込みステータス・フラグの指示内容

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

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

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

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

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

**(4) メモリ・バンク許可フラグ (MBE)**

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

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

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

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

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

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

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



### (5) レジスタ・バンク許可フラグ (RBE)

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

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

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

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

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

ベクタ割り込み発生時は、該当するベクタ・アドレス・テーブルのビット6の内容がセットされて、割り込みサービス時の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。

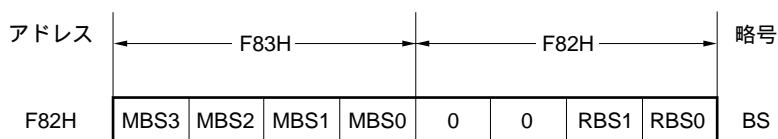
## 4.9 バンク選択レジスタ (BS)

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 - 14 バンク選択レジスタの構成



### (1) メモリ・バンク選択レジスタ (MBS)

メモリ・バンク選択レジスタは、データ・メモリ・アドレス (12ビット) 上位4ビット・アドレス情報を記憶する4ビット・レジスタで、このレジスタの内容によりアクセスするメモリ・バンクを指定します。ただし、μPD754202では、バンク0, 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, 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
	1	1	1	1	メモリ・バンク15 "
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

## 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

μPD754202ではメモリ・マップトI/Oが採用されており、すべての入出力ポートはデータ・メモリ空間上にマッピングされています。

図5 - 1 デジタル・ポートのデータ・メモリ・アドレス

アドレス	3	2	1	0	
FF0H	_____				
FF1H	_____				
FF2H	_____				
FF3H	P33	P32	P31	P30	PORT3
FF4H	_____				
FF5H	_____				
FF6H	P63	P62	P61	P60	PORT6
FF7H	P73	P72	P71	P70	PORT7
FF8H				P80	PORT8

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およびPORT6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 . P73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3に出力。

SKT PORT7.3 ; ポート7のビット3が1ならスキップ。

MOV XA, #8H ; XA 8H  
MOV XA, #4H ; XA 4H } たてづみ

SEL MB15 ; または、CLR1 MBE

OUT PORT3, A ; ポート3 A

2 . SET1 PORT6.@L ; ポート6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	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。	PTO0-PTO2と端子を兼用。
PORT6			INT0と端子を兼用。
PORT7	4ビット入力	4ビット入力専用ポート。 マスク・オプションで，1ビット単位にプルアップ抵抗の内蔵を指定可能。	KR4-KR7と端子を兼用。
PORT8	1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。	-

P61は，外部ベクタ割り込み入力端子と兼用になっており，ノイズ除去回路を選択可能です（詳細は6.3 割り込み機能を制御する各種ハードウェアを参照）。

$\overline{\text{RESET}}$ 信号発生により，ポート3，6，8の出力ラッチは0にクリアされ，出力バッファはオフし，入力モードになります。

図5 - 2 P3nの構成 (n = 0-2)

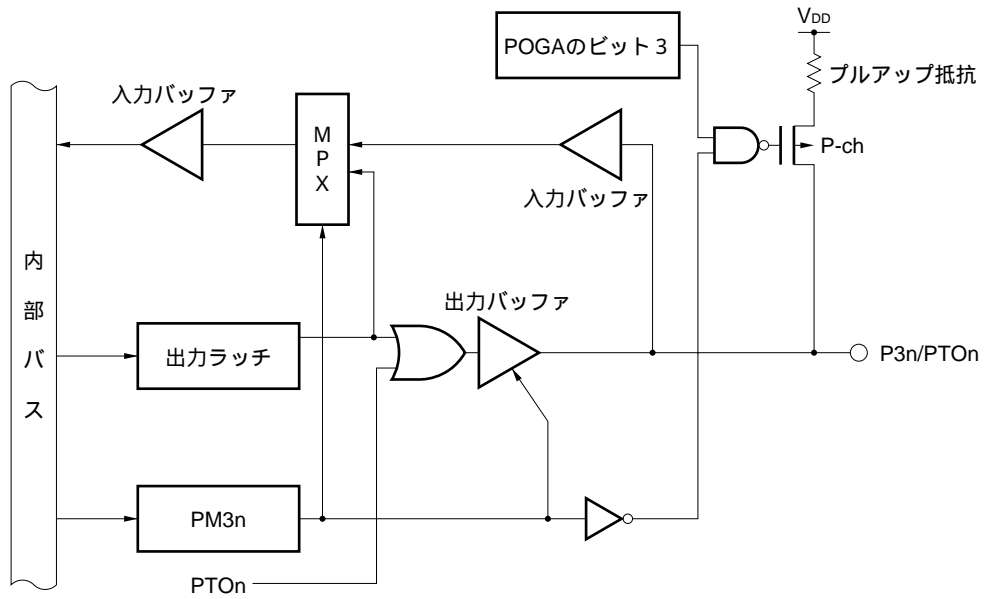


図5 - 3 P33の構成

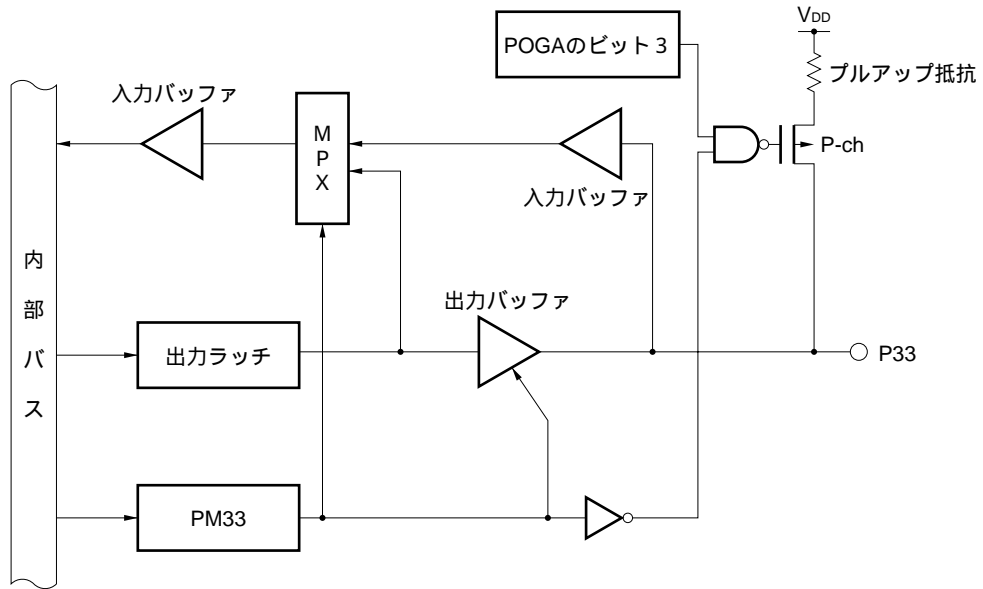


図5 - 4 P6nの構成 (n = 0, 2, 3)

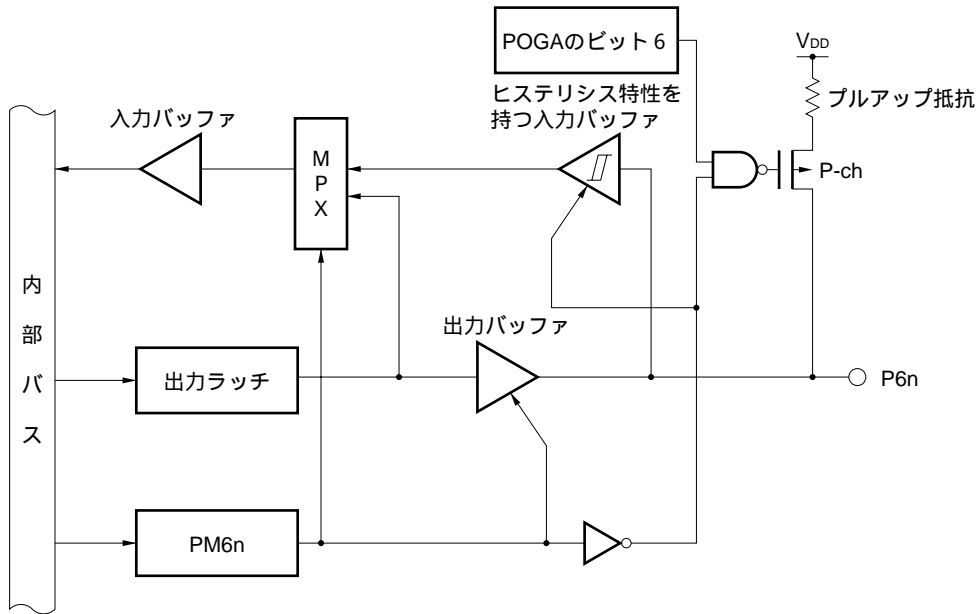


図5 - 5 P61の構成

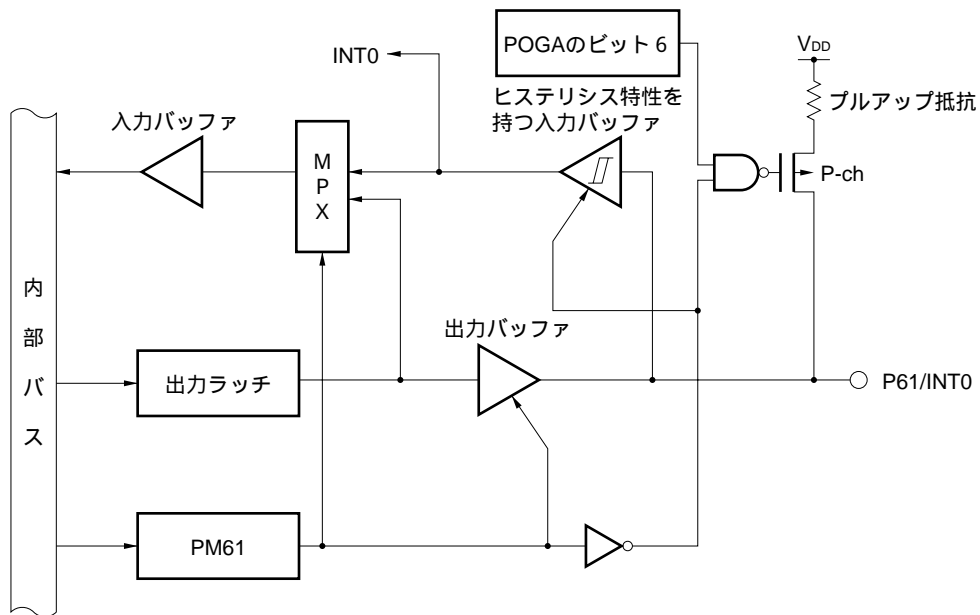
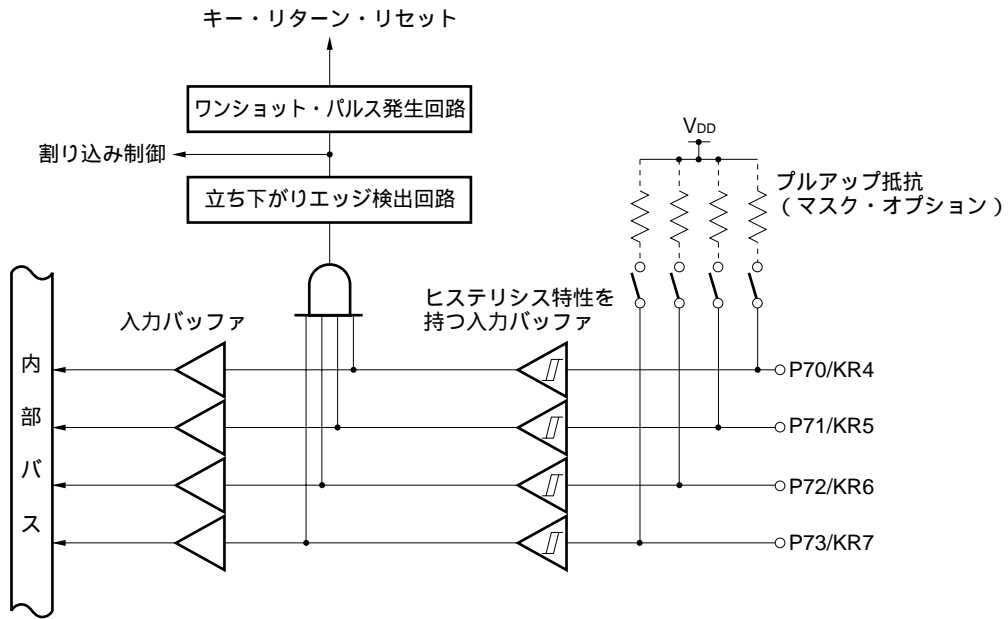
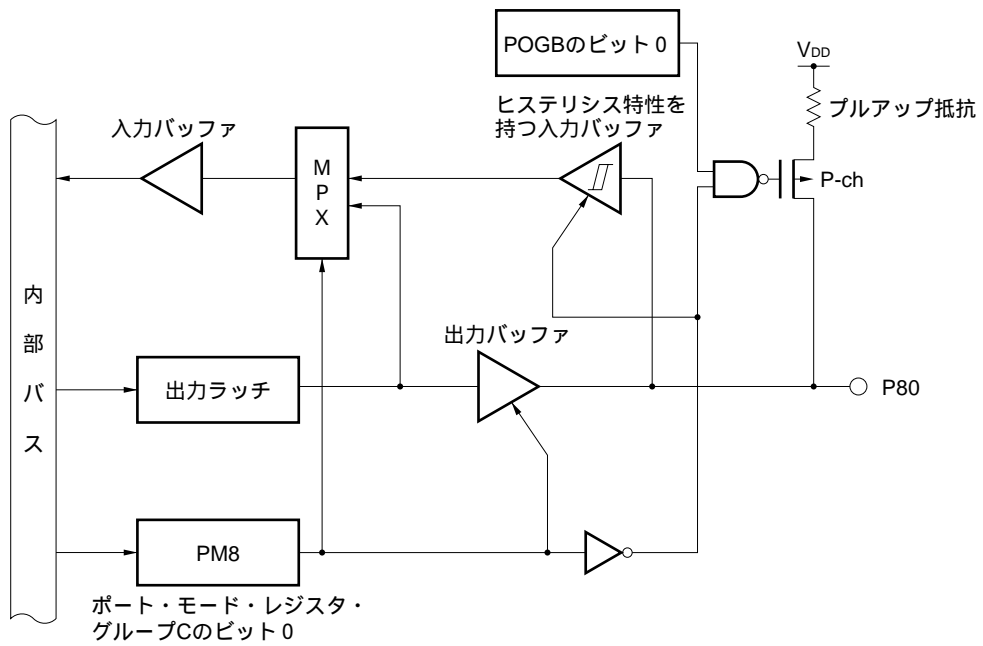


図5-6 P7nの構成 (n=0-3)



★

図5-7 P80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, 6はポート・モード・レジスタ・グループA (PMGA) によって、ビットごとに入出力を指定できます。ポート8はポート・モード・レジスタ・グループC (PMGC) によって、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は、それぞれ8ビット・メモリ操作命令で設定します。

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

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

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

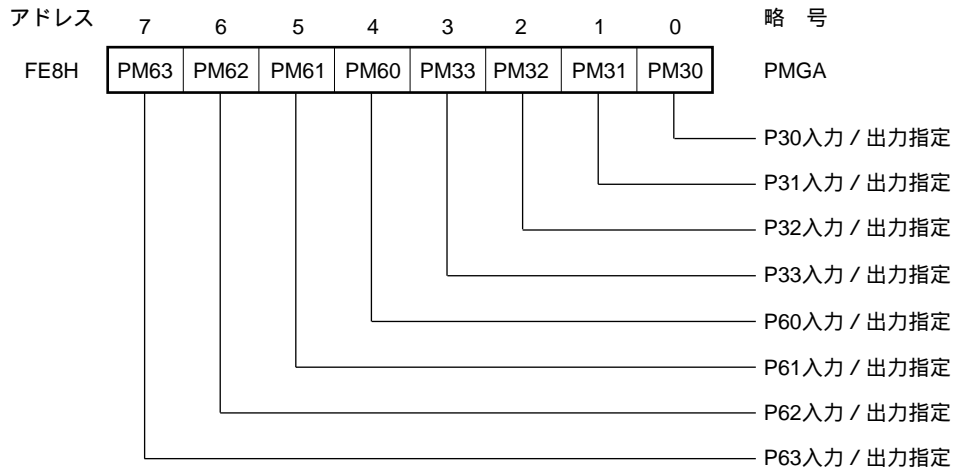
```
MOV     XA, #3CH
```

```
MOV     PMGA, XA
```

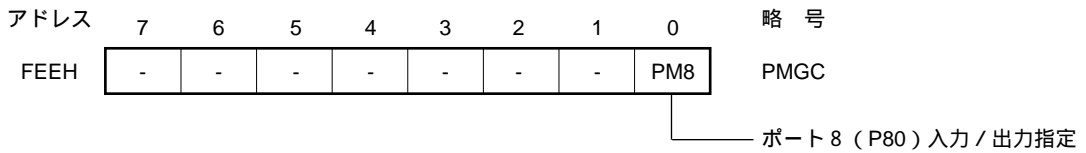
図5 - 8 各ポート・モード・レジスタのフォーマット

	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



### 5.1.3 デジタル入出力ポート操作命令

μ PD754202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, 6, 8は、特定アドレス・ビット直接アドレッシング (fmem. bit) と特定アドレス・ビット・レジスタ間接アドレッシング (pmem. @L) が適用できますから、MBE, MBSの指定にかかわらずいつでも、自由に、ポートのビット操作が可能です。

例 P30とP61のORをとってP80に出力する。

```
MOV1    CY, PORT3.0    ; CY P30
OR1     CY, PORT6.1    ; CY CY P61
MOV1    PORT8.0, CY    ; P80 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6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, #PORT6
ADDS   A, @HL         ; A A + PORT6
NOP
MOV     @HL, A        ; PORT6 A
```

3 . ポート3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, #PORT3
SUBS   A, @HL         ; A < PORT3
BR     NO             ; NO
                       ; YES
```

★

表5 - 2 入出力端子操作命令一覧表

命令 \ PORT	PORT3	PORT6	PORT7	PORT8
IN A, PORTn <sup>注1</sup>				
IN XA, PORTn <sup>注1</sup>		-		
OUT PORTn, A <sup>注1</sup>			-	
OUT PORTn, XA <sup>注1</sup>		-		
MOV A, PORTn <sup>注1</sup>				
MOV XA, PORTn <sup>注1</sup>		-		
MOV PORTn, A <sup>注1</sup>				
MOV PORTn, XA <sup>注1</sup>		-		
XCH A, PORTn <sup>注1</sup>				
XCH XA, PORTn <sup>注1</sup>		-		
MOV1 CY, PORTn. bit				
MOV1 CY, PORTn. @L <sup>注2</sup>				
MOV1 PORTn. bit, CY			-	
MOV1 PORTn. @L, CY <sup>注2</sup>			-	
INCS PORTn <sup>注1</sup>				
SET1 PORTn. bit				
SET1 PORTn. @L <sup>注2</sup>				
CLR1 PORTn. bit				
CLR1 PORTn. @L <sup>注2</sup>				
SKT PORTn. bit				
SKT PORTn. @L <sup>注2</sup>				
SKF PORTn. bit				
SKTCLR PORTn. bit				
SKTCLR PORTn. @L <sup>注2</sup>				
SKF PORTn. @L <sup>注2</sup>				
AND1 CY, PORTn. bit				
AND1 CY, PORTn. @L <sup>注2</sup>				
OR1 CY, PORTn. bit				
OR1 CY, PORTn. @L <sup>注2</sup>				
XOR1 CY, PORTn. bit				
XOR1 CY, PORTn. @L <sup>注2</sup>				

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="checkbox"/> SKF <input type="checkbox"/>	端子のデータをテスト	出力ラッチのデータをテスト
MOV1 CY, <input type="checkbox"/>	端子のデータをCYへ転送	出力ラッチのデータをCYへ転送
AND1 CY, <input type="checkbox"/> OR1 CY, <input type="checkbox"/> XOR1 CY, <input type="checkbox"/>	端子のデータとCYの間で演算	出力ラッチのデータとCYの間で演算
IN A, PORTn 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にラッチ	出力ラッチの内容を + 1
SET1 <input type="checkbox"/> CLR1 <input type="checkbox"/> MOV1 <input type="checkbox"/> , CY SKTCLR <input type="checkbox"/>	指定したビットの出力ラッチは命令の指示通り書き替わるが、その他のビットの出力ラッチは不定	出力端子の状態を命令に従って変更

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

### 5.1.5 プルアップ抵抗の内蔵

μPD754202の各ポート端子にはプルアップ抵抗を内蔵することができます。プルアップ抵抗の内蔵の指定はソフトウェアによる端子と、マスク・オプションによる端子があります。

各ポート端子の指定法を表5-4に示します。また、ソフトウェアによる指定は、図5-9に示したフォーマットに基づいて行います。

ポート3, 6, 8に対するプルアップ抵抗の内蔵指定は、入力モードに指定されている端子にのみ有効です。出力モードに指定された端子は、POGA, POGBの設定にかかわらず、プルアップ抵抗の内蔵を指定できません。

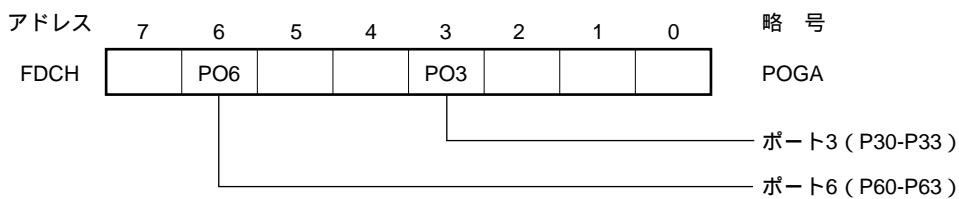
表5-4 プルアップ抵抗内蔵の指定法

ポート (端子名)	プルアップ抵抗内蔵の指定法	指定ビット
ポート3 (P30-P33)	4ビット単位でソフトウェアにより内蔵プルアップ抵抗の接続を指定	POGA.3
ポート6 (P60-P63)		POGA.6
ポート7 (P70-P73)	1ビット単位でマスク・オプションによりプルアップ抵抗の内蔵を指定	-
ポート8 (P80)	1ビット単位でソフトウェアにより内蔵プルアップ抵抗の接続を指定	POGB.0

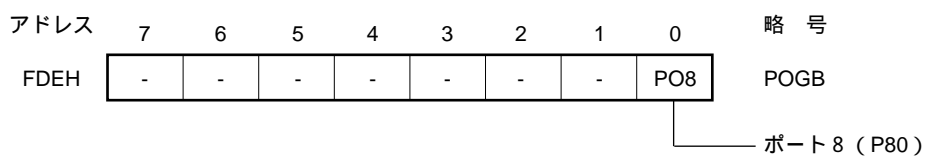
図5-9 プルアップ抵抗指定レジスタのフォーマット

	指定内容
0	プルアップ抵抗を内蔵しない
1	プルアップ抵抗を内蔵する

#### プルアップ抵抗指定レジスタ・グループA



#### プルアップ抵抗指定レジスタ・グループB



### 5.1.6 デジタル入出力ポートの入出力タイミング

出力ラッチにデータが出力されるタイミングおよび端子のデータまたは、出力ラッチのデータが内部バスに取り込まれるタイミングは図5 - 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

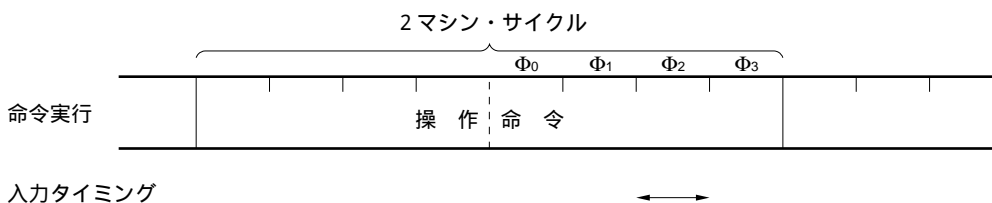
★

(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

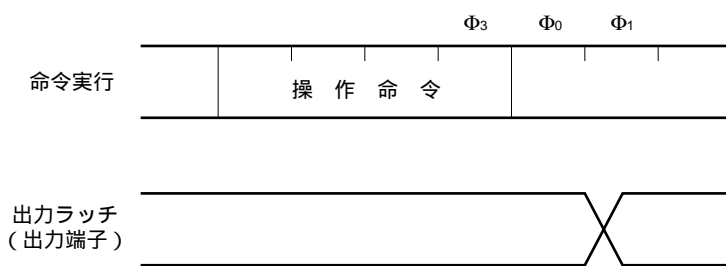


★

(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



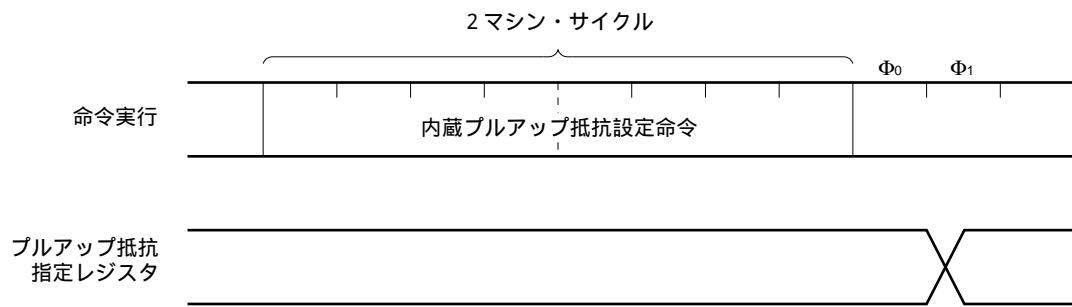
(d) 2マシン・サイクル命令によってデータがラッチされる場合





★

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



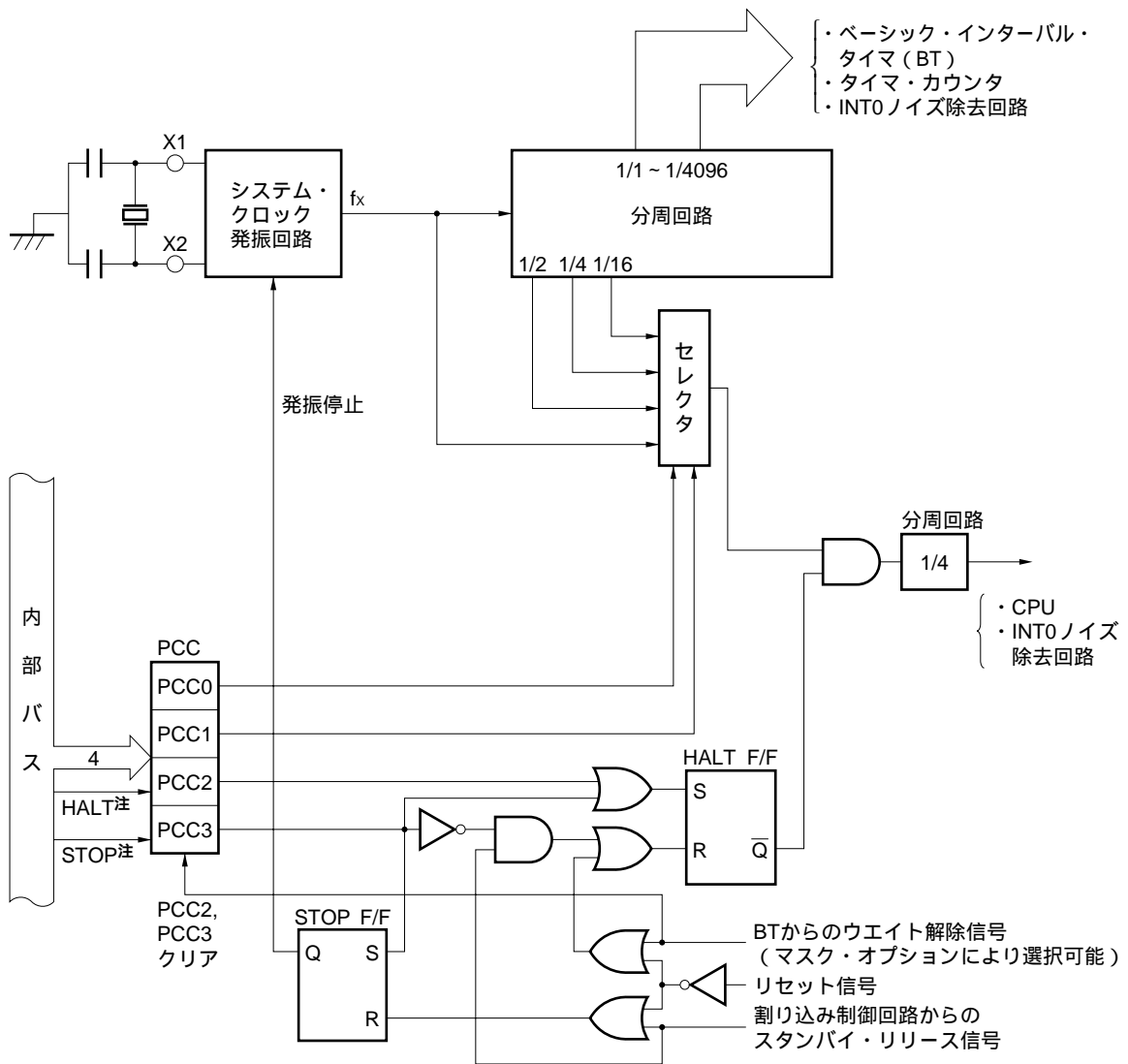
## 5.2 クロック発生回路

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 .  $f_x$  = システム・クロック周波数

2 . = CPUクロック

3 . PCC : プロセッサ・クロック・コントロール・レジスタ

4 . の1クロック・サイクル ( $t_{CY}$ ) は命令の1マシン・サイクル

### 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。

システム・クロック  $f_x$   
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) により決定され、次のような機能、動作となります。

- ( a ) RESET信号発生によりシステム・クロックの最低速モード<sup>注1</sup>が選択されます ( PCC = 0 )。
- ( b ) 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<sup>注2</sup>。
- ( c ) 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 ) 周辺ハードウェアへのクロックはシステム・クロックを分周して供給されます。

注1 . 4.19 MHz動作時で15.3  $\mu$ s。また, 6.0 MHz動作時で10.7  $\mu$ s。

2 . 4.19 MHz動作時で0.95, 1.91, 3.81, 15.3  $\mu$ s。また, 6.0 MHz動作時で0.67, 1.33, 2.67, 10.7  $\mu$ s。

**(1) プロセッサ・クロック・コントロール・レジスタ (PCC)**

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

ビット3またはビット2の片方が“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。

**例1** . マシン・サイクルを最高速モード<sup>注1</sup>にする。

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

**2** . マシン・サイクルを1.33  $\mu$ s ( $f_x = 6.0$  MHz動作時)<sup>注2</sup>にする。

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

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

```
STOP
NOP
```

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

**注1** . 0.67  $\mu$ s ( $f_x = 6.0$  MHz動作時)、または0.95  $\mu$ s ( $f_x = 4.19$  MHz動作時)。

**2** .  $f_x = 4.19$  MHz動作時では、1.91  $\mu$ s。



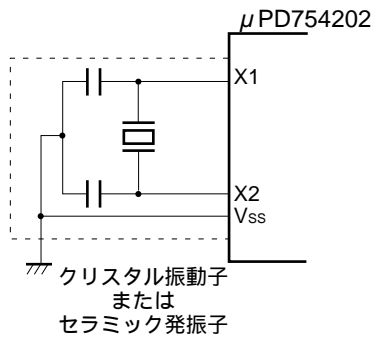
(2) システム・クロック発振回路

システム・クロック発振回路はX1, X2端子に接続されたクリスタル振動子またはセラミック発振子によって発振します(標準: 6.0 MHz または4.19 MHz)。

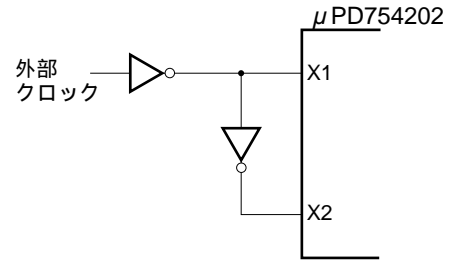
また, 外部クロックを入力することもできます。

図5-14 クリスタル/セラミック発振の外付け回路

(i) クリスタル/セラミック発振



(ii) 外部クロック



★

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

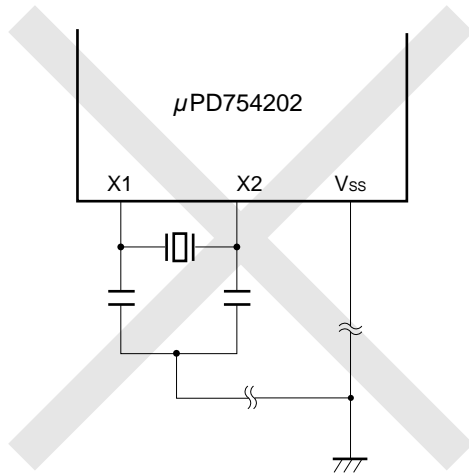
2 . システム・クロック発振回路を使用する場合は, 配線容量などの影響を避けるために, 図5-14の破線の部分を次のように配線してください。

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

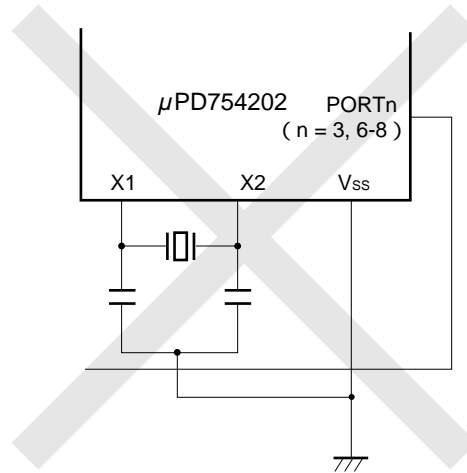
図5-15に発振子の接続回路の悪い例を示します。

図5-15 発振子の接続回路の悪い例(1/2)

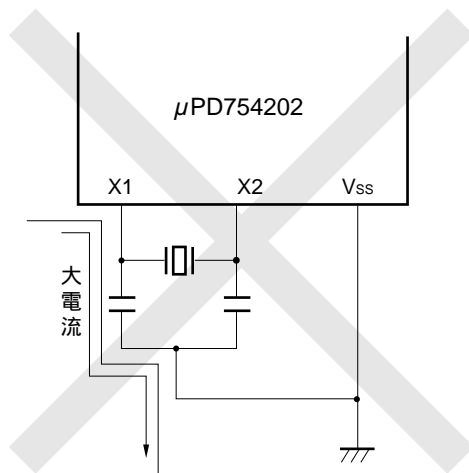
(a) 接続回路の配線が長い



(b) 信号線が交差している



(c) 変化する大電流が信号線に近接している



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

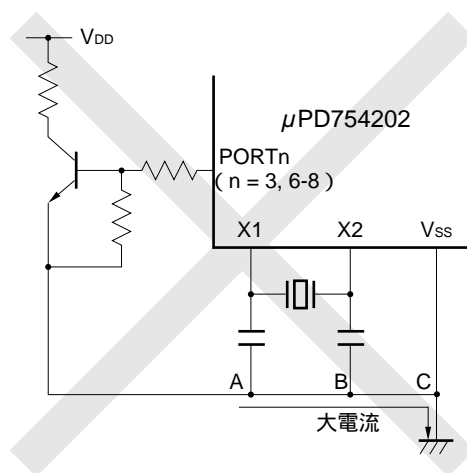
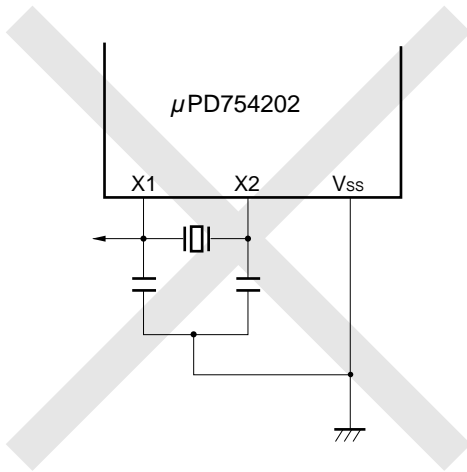


図5 - 15 発振子の接続回路の悪い例 (2/2)

(e) 信号を取り出している



### (3) 分周回路

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



### 5.2.3 CPUクロックの設定

#### (1) CPUクロックの切り替えに要する時間

CPUクロックは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、システム・クロックの発振を停止させる場合には、この切り替え時間経過後、STOP命令の実行を行う必要があります。

表5 - 5 CPUクロックの切り替えに要する最大時間

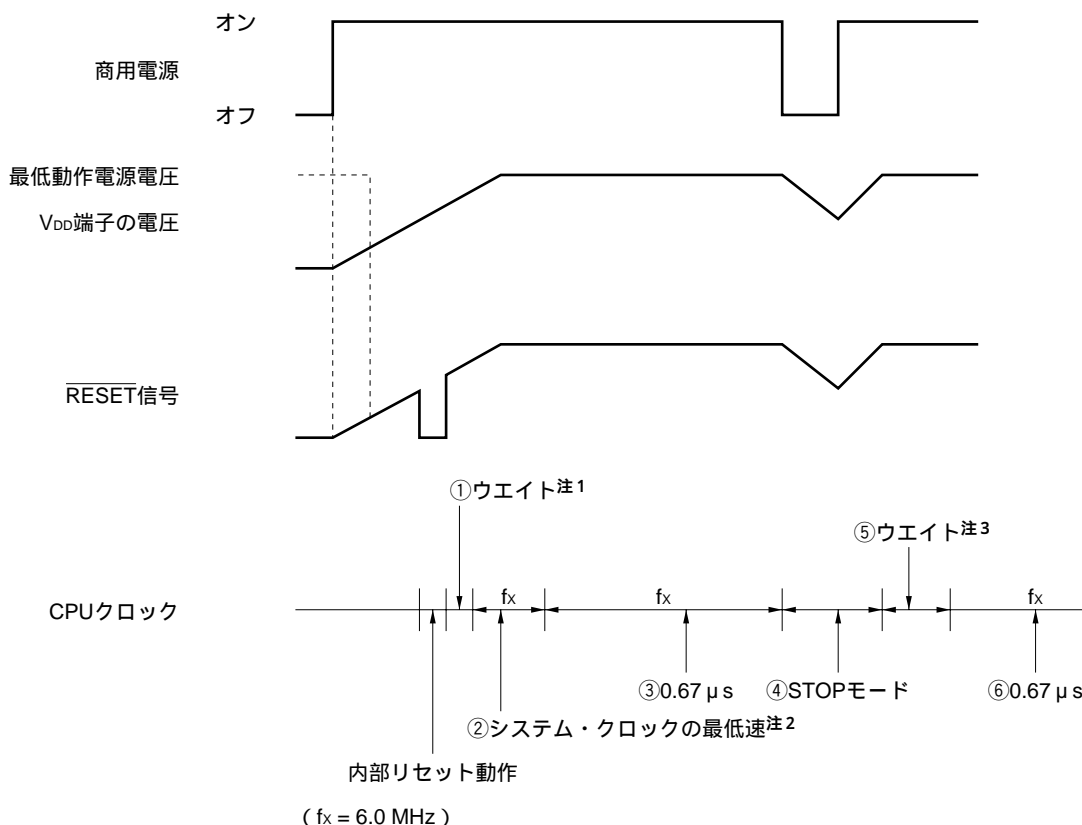
切り替え前の設定値		切り替え後の設定値							
PCC1	PCC0	PCC1	PCC0	PCC1	PCC0	PCC1	PCC0	PCC1	PCC0
		0	0	0	1	1	0	1	1
0	0	1マシン・サイクル		1マシン・サイクル		1マシン・サイクル		1マシン・サイクル	
0	1			4マシン・サイクル		4マシン・サイクル		4マシン・サイクル	
1	0	8マシン・サイクル		8マシン・サイクル		8マシン・サイクル		8マシン・サイクル	
1	1	16マシン・サイクル		16マシン・サイクル		16マシン・サイクル		16マシン・サイクル	

**注意** 発振子の周囲温度、負荷容量の性能のばらつきなどの条件により、 $f_x$ の値は変化します。特に $f_x$ が公称値より高い場合には、表中のマシン・サイクルは、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、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には、 $f_x$ の公称値で求めたマシン・サイクルよりも長くしてください。

★ (2) CPUクロックの切り替え手順

CPUクロックの切り替えを図5 - 16に従って説明します。

図5 - 16 CPUクロックの切り替え例



RESET信号発生による発振安定時間確保のためのウエイト時間<sup>注1</sup>。

CPUはシステム・クロックの最低速<sup>注2</sup>で動作をスタートします。

V<sub>DD</sub>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、最高速で動作させます。

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、STOPモードにします。

商用電源が復帰したことを割り込みで検出したあとの、STOPモードから復帰、発振安定時間確保のためのウエイト時間<sup>注3</sup>。

通常動作します。

注1．ウエイト時間はマスク・オプションで選択することができます。

4.19 MHz動作時で $2^{15}/f_x = 7.81 \text{ ms}$ と $2^{17}/f_x = 31.3 \text{ ms}$ の2つから、また6.0 MHz動作時で $2^{15}/f_x = 5.46 \text{ ms}$ と $2^{17}/f_x = 21.8 \text{ ms}$ の2つから選択できます。

2．4.19MHz動作時で15.3  $\mu\text{s}$ 。また、6.0MHz動作時で10.7  $\mu\text{s}$ 。

3．BTMにより次の4つの時間から選択することができます。

$2^{20}/f_x, 2^{17}/f_x, 2^{15}/f_x, 2^{13}/f_x$

## 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

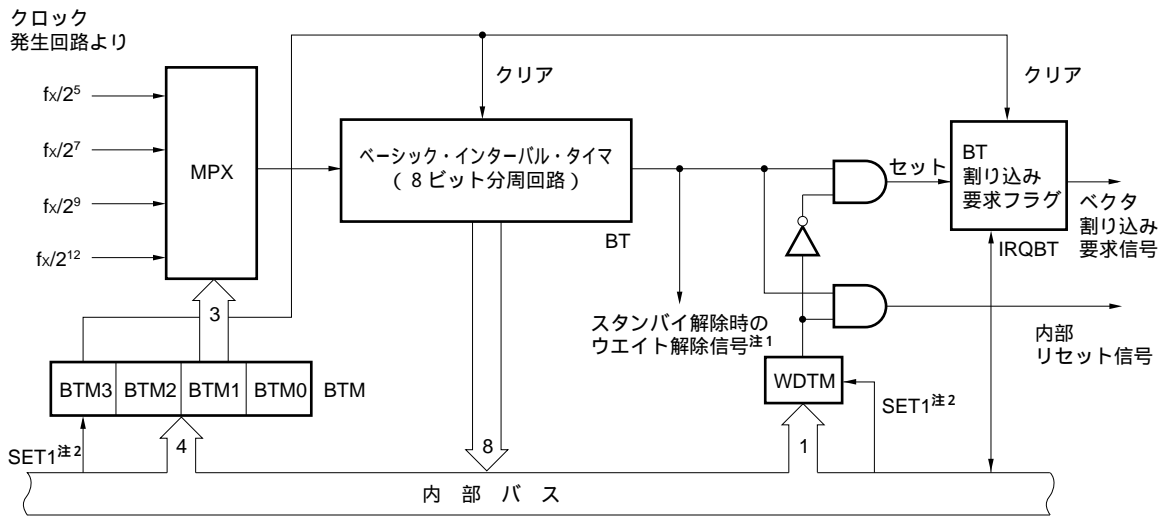
μPD754202は、8ビット・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマを備えており、次のような機能があります。

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

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

ベーシック・インターバル・タイマ/ウォッチドッグ・タイマは、図5-17のように構成されています。

図5-17 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1．スタンバイ・モード解除時にウエイト時間を選択することができます。

詳細は、第7章 スタンバイ機能、および、第8章 リセット機能を参照してください。

2．命令の実行です。

### 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**  $\mu$ PD754202の割り込み発生インターバルを1.37 ms (6.0 MHz動作時)<sup>注</sup>に設定

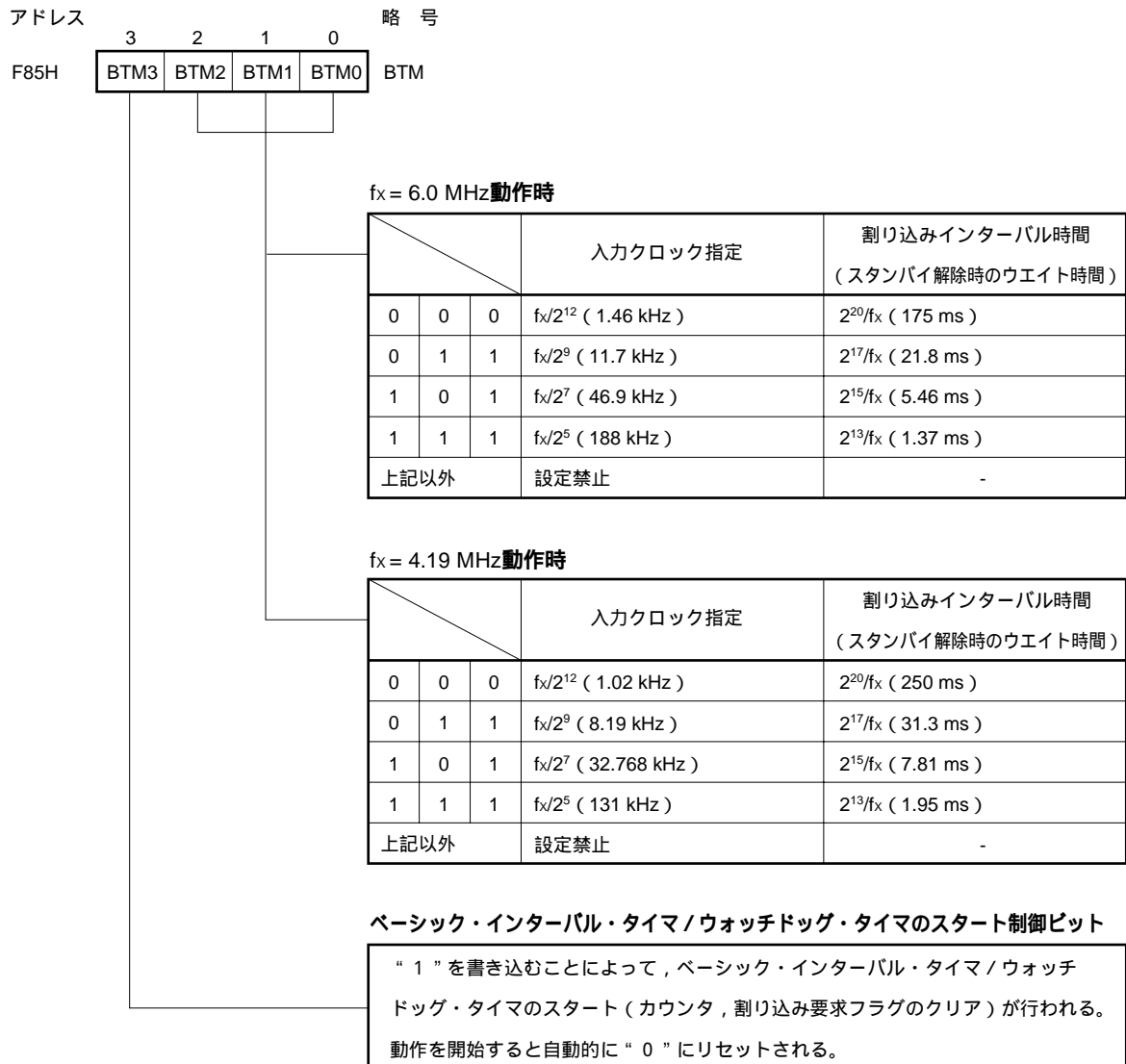
```
SEL      MB15          ;またはCLR1 MBE
CLR1     WDTM
MOV      A, #1111B
MOV      BTM, A        ;BTM 1111B
```

**注** 4.19 MHz動作時では、1.95 msになります。

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

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

図5 - 18 ベーシック・インターバル・タイマ・モード・レジスタのフォーマット



### 5.3.3 ウォッチドッグ・タイマ許可フラグ (WDTM)

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

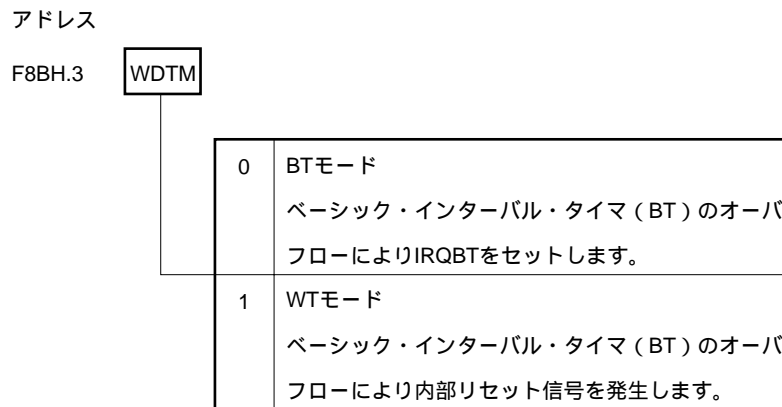
WDTMは、ビット操作命令により設定します。1度セットすると、命令ではクリアできません。

例 ウォッチドッグ・タイマ機能の設定

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

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

図5 - 19 ウォッチドッグ・タイマ許可フラグ (WDTM) のフォーマット



### 5.3.4 ベーシック・インターバル・タイマの動作

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

割り込み発生インターバルは、BTMの設定により4通りの時間が選択できます(図5-18参照)。

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

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

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

BTMにインターバル時間をセットする。

BTMのビット3に“1”をセットする。

**例** 1.37 ms (6.0 MHz動作時)<sup>注</sup>ごとに割り込みを発生させる。

```
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
```

**注** 4.19 MHz 動作では、1.95 msになります。

### 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5-18参照）。これらの中からユーザ・システムに応じて暴走検出に必要な時間を決めてください。インターバル時間を設定しておいて、プログラムをその時間内に実行できる単位に分割し、それぞれの単位の最後でBTをクリアする命令を実行させるようにします。そうすると、設定時間内にこのBTクリアを実行する命令にたどりつかなければ（順調にプログラムの実行が進んでいなければ＝暴走）BTはオーパフローし、内部リセット信号が発生してプログラムを強制終了させてしまいます。この結果、内部リセットがかかったということはプログラムの暴走が起きたことを示し、その検出ができたことになります。

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

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

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



例 5.46 msのウォッチドッグ・タイマとして使用する（6.0 MHz動作時）<sup>注</sup>。

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

初期設定：

```

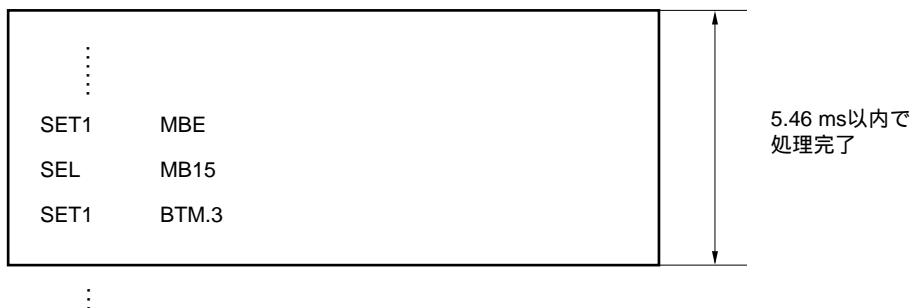
SET1    MBE
SEL     MB15
MOV     A, #1101B
MOV     BTM, A    ; 時間設定とスタート
SET1    WDTM     ; ウォッチドッグ・タイマを許可
      ⋮
    
```

（以後、5.46 msごとにBTMのビット3に“1”をセットする。）

モジュール1：



モジュール2：



注 4.19 MHz動作時では、7.81 msになります。

### 5.3.6 その他の機能

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

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

#### (1) STOPモード解除後のウエイト時間の選択とカウント

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

$\overline{\text{RESET}}$ 信号発生後のウエイト時間は、マスク・オプションにより固定されますが、割り込み発生によるSTOPモード解除時は、BTMの設定によりウエイト時間を選択できます。その場合のウエイト時間は図5-18のインターバル時間と同じになります。BTMへの設定は、STOPモードに設定する前に行います (詳しくは第7章 スタンバイ機能を参照)。

**例** STOPモードを割り込みで解除したときのウエイト時間を5.46 msに設定する (6.0 MHz動作時) 注。

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

**注** 4.19 MHz動作時では、7.81 msになります。

#### (2) カウント動作の読み出し

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

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

例 BTのカウンタ内容の読み出し。

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

## 5.4 タイマ・カウンタ

μPD754202は、タイマ・カウンタを3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 ) PTO0-PTO2端子への、任意の方形波出力
- ( c ) カウント値の読み出し機能

また、タイマ・カウンタの動作は、モード・レジスタの設定によって、次に示す4種類のモードでの動作が可能です。

表5 - 6 モード一覧

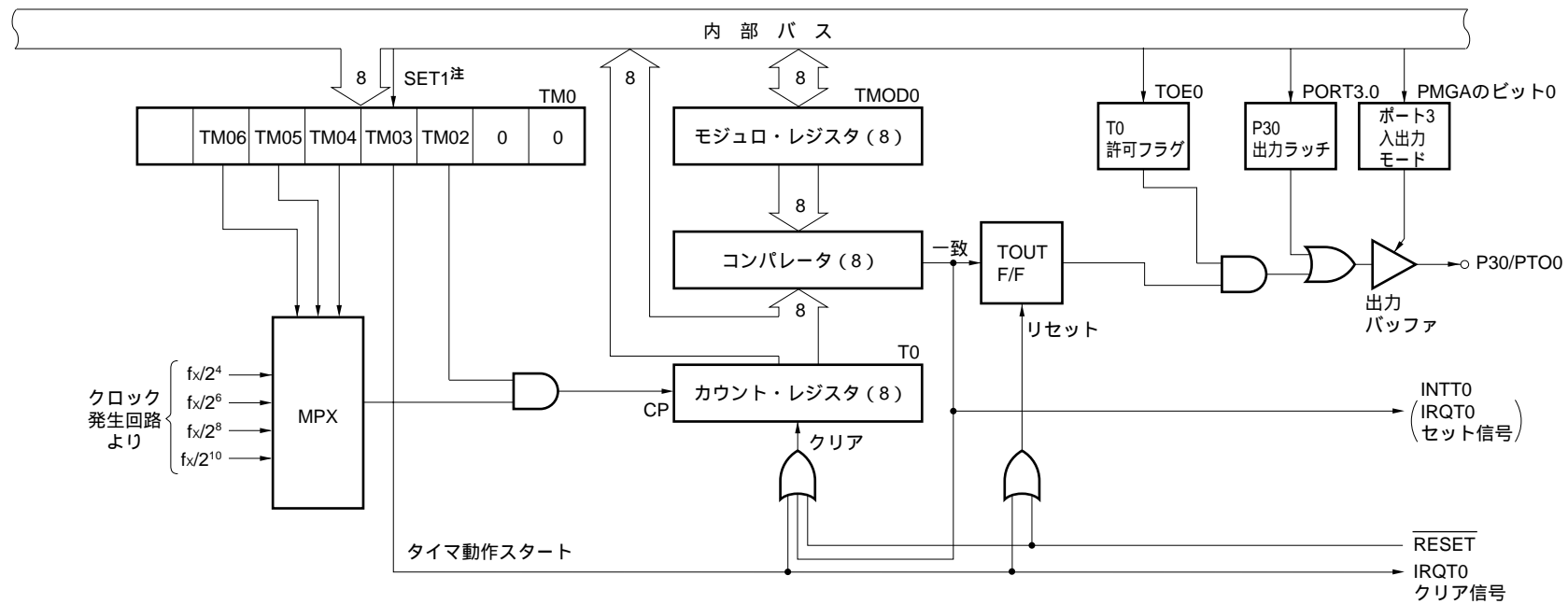
モード	チャンネル			TM11	TM10	TM21	TM20	参照項
	チャンネル0	チャンネル1	チャンネル2					
8ビット・タイマ・カウンタ・モード				0	0	0	0	5.4.2
PWMパルス・ジェネレータ・モード	×	×		0	0	0	1	5.4.3
16ビット・タイマ・カウンタ・モード	×			1	0	1	0	5.4.4
キャリア・ジェネレータ・モード	×			0	0	1	1	5.4.5

備考 ×：該当する機能がありません。

### 5.4.1 タイマ・カウンタの構成

タイマ・カウンタは図5 - 20～5 - 22のように構成されています。

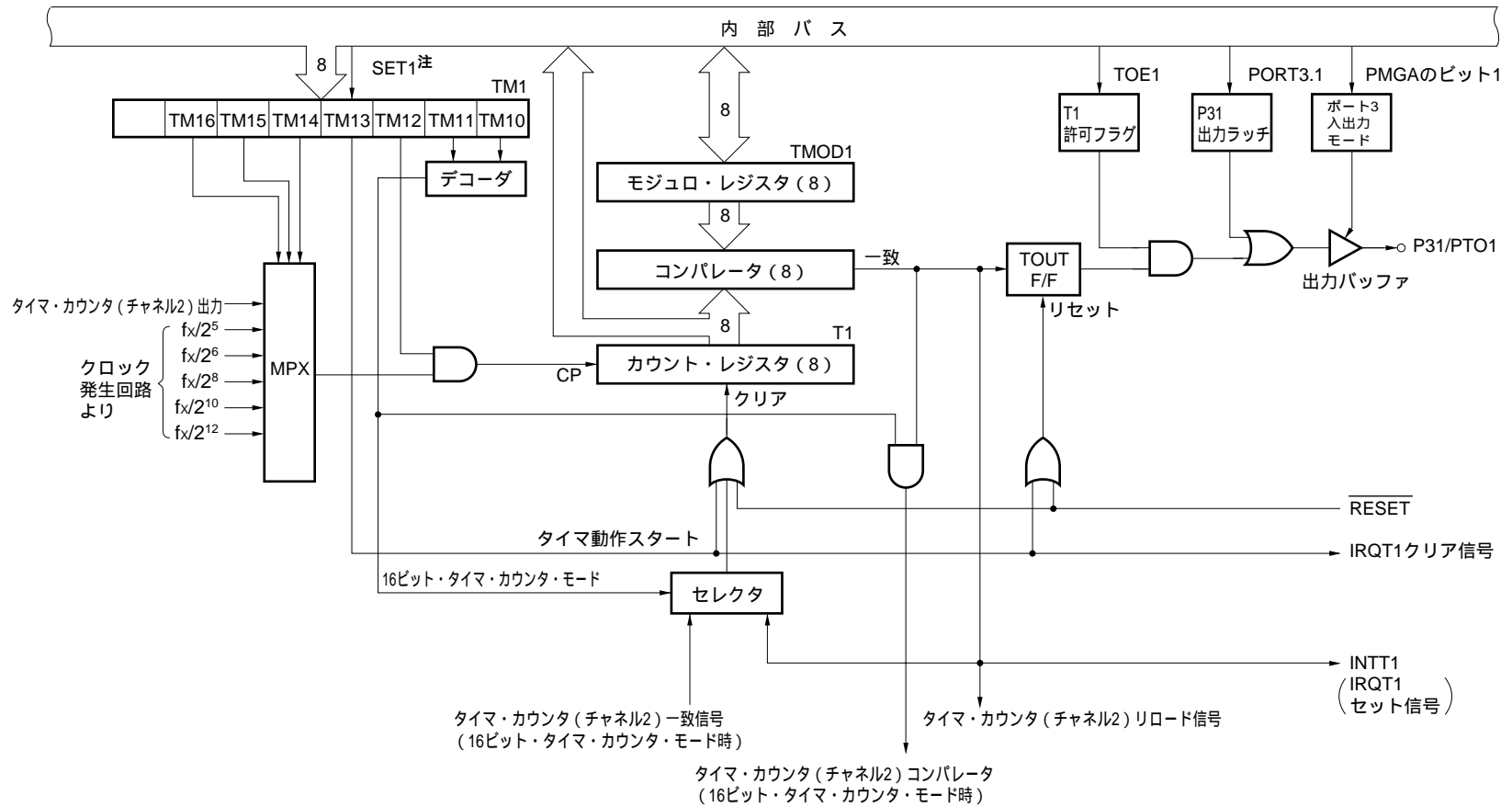
★図5 - 20 タイマ・カウンタ (チャンネル0) のブロック図



注 命令の実行

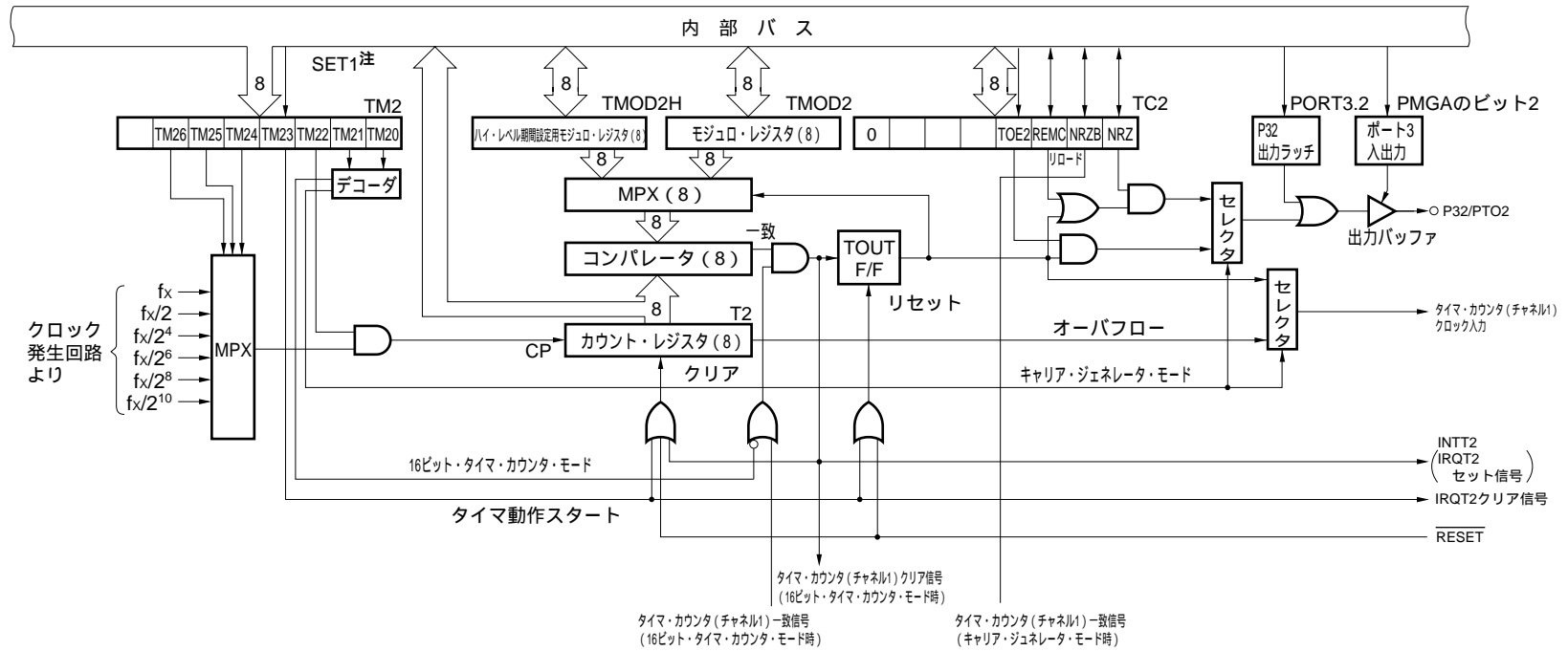
注意 TM0にデータを設定するときは、ビット0, 1を必ず0に設定してください。

図5-21 タイマ・カウンタ(チャンネル1)のブロック図



注 命令の実行

図5-22 タイマ・カウンタ(チャンネル2)のブロック図



注 命令の実行

注意 TC2にデータを設定するときは、ビット7を必ず0に設定してください。

## (1) タイマ・カウンタ・モード・レジスタ (TM0, TM1, TM2)

タイマ・カウンタ・モード・レジスタ (TMn) 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 - 23 ~ 5 - 25 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・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 = 5.86 kHz (6.0 MHz動作時)<sup>注</sup>のインターバル・タイマ・モードでタイマをスタート。

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

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

```
SEL    MB15                ;またはCLR1 MBE
SET1   TMn.3              ;TMn. bit3 1
```

**注** 4.19 MHz動作時の場合は、CP = 4.10 kHzになります。

**備考** n = 0-2



図5 - 23 タイマ・カウンタ・モード・レジスタ (チャンネル0) のフォーマット

アドレス	7	6	5	4	3	2	1	0	略号
FA0H	-	TM06	TM05	TM04	TM03	TM02	0 <sup>注</sup>	0 <sup>注</sup>	TM0

**カウント・パルス (CP) 選択ビット**

$f_x = 6.0 \text{ MHz}$ 動作時

TM06	TM05	TM04	カウント・パルス (CP)
1	0	0	$f_x/2^{10}$ (5.86 kHz)
1	0	1	$f_x/2^8$ (23.4 kHz)
1	1	0	$f_x/2^6$ (93.8 kHz)
1	1	1	$f_x/2^4$ (375 kHz)
上記以外			設定禁止

$f_x = 4.19 \text{ MHz}$ 動作時

TM06	TM05	TM04	カウント・パルス (CP)
1	0	0	$f_x/2^{10}$ (4.10 kHz)
1	0	1	$f_x/2^8$ (16.4 kHz)
1	1	0	$f_x/2^6$ (65.5 kHz)
1	1	1	$f_x/2^4$ (262 kHz)
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 TM0にデータを設定するときは、ビット0、1を必ず0に設定してください。

注意 リセットすると、TM0は全ビット“ 0 ” になりますので、タイマを動作させる場合は、最初  
にカウント・パルスの値を設定する必要があります。また、CPに上記以外の値を記述した  
場合、設定されるカウント・パルスは0となり、タイマとして機能しません。

図5 - 24 タイマ・カウンタ・モード・レジスタ (チャンネル1) のフォーマット (1/2)

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

**カウント・パルス (CP) 選択ビット**

$f_x = 6.0 \text{ MHz}$ 動作時

TM16	TM15	TM14	カウント・パルス (CP)
0	1	0	タイマ・カウンタ (チャンネル2) のオーバフロー
0	1	1	$f_x/2^5$ (188 kHz)
1	0	0	$f_x/2^{12}$ (1.46 kHz)
1	0	1	$f_x/2^{10}$ (5.86 kHz)
1	1	0	$f_x/2^8$ (23.4 kHz)
1	1	1	$f_x/2^6$ (93.8 kHz)
上記以外			設定禁止

$f_x = 4.19 \text{ MHz}$ 動作時

TM16	TM15	TM14	カウント・パルス (CP)
0	1	0	タイマ・カウンタ (チャンネル2) のオーバフロー
0	1	1	$f_x/2^5$ (131 kHz)
1	0	0	$f_x/2^{12}$ (1.02 kHz)
1	0	1	$f_x/2^{10}$ (4.10 kHz)
1	1	0	$f_x/2^8$ (16.4 kHz)
1	1	1	$f_x/2^6$ (65.5 kHz)
上記以外			設定禁止

**タイマ・スタート指示ビット**

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

**動作モード**

TM12	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

図5 - 24 タイマ・カウンタ・モード・レジスタ（チャンネル1）のフォーマット（2/2）

## 動作モード選択ビット

TM11	TM10	モード
0	0	8ビット・タイマ・カウンタ・モード <sup>注</sup>
1	0	16ビット・タイマ・カウンタ・モード
上記以外		設定禁止

**注** タイマ・カウンタ・モード・レジスタ（チャンネル2）のTM20, TM21（= 11）と組み合わせて使用する場合は、キャリア・ジェネレータ・モードとなります。

**注意** リセット時には、TM1は全ビット“0”になりますので、タイマを動作させる場合は、最初にカウント・パルスの値を設定する必要があります。また設定禁止の値を設定した場合、カウント・パルスは0になり、タイマとして機能しません。

図5 - 25 タイマ・カウンタ・モード・レジスタ (チャンネル2) のフォーマット (1/2)

アドレス	7	6	5	4	3	2	1	0	略号
F90H	-	TM26	TM25	TM24	TM23	TM22	TM21	TM20	TM2

**カウント・パルス (CP) 選択ビット**

fx = 6.0 MHz動作時

TM26	TM25	TM24	カウント・パルス (CP)
0	1	0	fx/2 (3.0 MHz)
0	1	1	fx (6.0 MHz)
1	0	0	fx/2 <sup>10</sup> (5.86 kHz)
1	0	1	fx/2 <sup>8</sup> (23.4 kHz)
1	1	0	fx/2 <sup>6</sup> (93.8 kHz)
1	1	1	fx/2 <sup>4</sup> (375 kHz)
上記以外			設定禁止

fx = 4.19 MHz動作時

TM26	TM25	TM24	カウント・パルス (CP)
0	1	0	fx/2 (2.10 MHz)
0	1	1	fx (4.19 MHz)
1	0	0	fx/2 <sup>10</sup> (4.10 kHz)
1	0	1	fx/2 <sup>8</sup> (16.4 kHz)
1	1	0	fx/2 <sup>6</sup> (65.5 kHz)
1	1	1	fx/2 <sup>4</sup> (262 kHz)
上記以外			設定禁止

**タイマ・スタート指示ビット**

TM23	“ 1 ” を書き込むことによって、カウンタ、IRQT2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**

TM22	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

図5 - 25 タイマ・カウンタ・モード・レジスタ（チャンネル2）のフォーマット（2/2）

## 動作モード選択ビット

TM21	TM20	モード
0	0	8ビット・タイマ・カウンタ・モード
0	1	PWMパルス・ジェネレータ・モード
1	0	16ビット・タイマ・カウンタ・モード
1	1	キャリア・ジェネレータ・モード

**注意** リセット時には、TM2は全ビット“0”になりますので、タイマを動作させる場合は、最初にカウント・パルスの値を設定する必要があります。また設定禁止の値を設定した場合、カウント・パルスは0になり、タイマとして機能しません。

(2) タイマ・カウンタ出力許可フラグ (TOE0, TOE1)

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

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

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

図5 - 26 タイマ・カウンタ出力許可フラグのフォーマット



(3) タイマ・カウンタ・コントロール・レジスタ (TC2)

タイマ・カウンタ・コントロール・レジスタ (TC2) は、タイマ・カウンタ (チャンネル2) を制御する 8 ビット・レジスタです。図 5 - 27 にそのフォーマットを示します。

タイマ・カウンタ (チャンネル1) と組み合わせて使用するキャリア・ジェネレータ・モードのタイマ出力許可の制御を行います。

TC2は、8 ビット、4 ビット操作命令、ビット操作命令により設定します。

内部リセット信号発生により、TC2は全ビット 0 にクリアされます。

図 5 - 27 タイマ・カウンタ・コントロール・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
F92H	0 <sup>注</sup>	-	-	-	TOE2	REMC	NRZB	NRZ	TC2

タイマ・カウンタ出力許可フラグ

TOE2	タイマ出力
0	禁止 (ロウ・レベルを出力する)
1	許可

リモコン出力制御フラグ

REMC	リモコン出力
0	NRZ = 1 のとき、PTO2端子にキャリア・パルスを出力する
1	NRZ = 1 のとき、PTO2端子にハイ・レベルを出力する

ノー・リターン・ゼロ・バッファ・フラグ

NRZB	次に出力するノー・リターン・ゼロ・データを格納するエリア。 タイマ・カウンタ (チャンネル1) の割り込み発生時、NRZへ転送される。
------	--

ノー・リターン・ゼロ・フラグ

NRZ	ノー・リターン・ゼロ・データ
0	PTO2端子にロウ・レベルを出力する
1	PTO2端子にキャリア・パルスを出力する

注 TC2にデータを設定するときは、ビット7を必ず0に設定してください。

## 5.4.2 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・タイマ動作およびカウント動作を行います。

### (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4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 (TMn)

タイマ・カウンタ・コントロール・レジスタ (TC2)<sup>注</sup>

タイマ・カウンタ・カウント・レジスタ (Tn)

タイマ・カウンタ・モジュロ・レジスタ (TMODn)

**注** タイマ・カウンタのチャンネル0とチャンネル1は、タイマ・カウンタ出力許可フラグ (TOE0, TOE1) を使用します。

### (a) タイマ・カウンタ・モード・レジスタ (TMn)

8ビット・タイマ・カウンタ・モードを使用する場合はTMnを図5 - 28のように設定します (TMnのフォーマットについては図5 - 23 ~ 5 - 25を参照してください)。

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

TMnは、内部リセット信号発生時に00Hにクリアされます。

**備考** n = 0-2



図5 - 28 タイマ・カウンタ・モード・レジスタの設定 (1/3)

(a) タイマ・カウンタ (チャンネル0) の場合

アドレス	7	6	5	4	3	2	1	0	略号
FA0H		TM06	TM05	TM04	TM03	TM02	0 <sup>注</sup>	0 <sup>注</sup>	TM0

カウント・パルス (CP) 選択ビット

TM06	TM05	TM04	カウント・パルス (CP)
1	0	0	$f_x/2^{10}$
1	0	1	$f_x/2^8$
1	1	0	$f_x/2^6$
1	1	1	$f_x/2^4$
上記以外			設定禁止

タイマ・スタート指示ビット

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

動作モード

TM02	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

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

図5 - 28 タイマ・カウンタ・モード・レジスタの設定 (2/3)

(b) タイマ・カウンタ (チャンネル1) の場合

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

カウント・パルス (CP) 選択ビット

TM16	TM15	TM14	カウント・パルス (CP)
0	1	0	タイマ・カウンタ (チャンネル2) のオーバフロー
0	1	1	$f_x/2^5$
1	0	0	$f_x/2^{12}$
1	0	1	$f_x/2^{10}$
1	1	0	$f_x/2^8$
1	1	1	$f_x/2^6$
上記以外			設定禁止

タイマ・スタート指示ビット

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

動作モード

TM12	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

動作モード選択ビット

TM11	TM10	モード
0	0	8ビット・タイマ・カウンタ・モード

図5 - 28 タイマ・カウンタ・モード・レジスタの設定 (3/3)

(c) タイマ・カウンタ (チャンネル2) の場合

アドレス	7	6	5	4	3	2	1	0	略号
F90H		TM26	TM25	TM24	TM23	TM22	TM21	TM20	TM2

カウント・パルス (CP) 選択ビット

TM26	TM25	TM24	カウント・パルス (CP)
0	1	0	$f_x/2$
0	1	1	$f_x$
1	0	0	$f_x/2^{10}$
1	0	1	$f_x/2^8$
1	1	0	$f_x/2^6$
1	1	1	$f_x/2^4$
上記以外			設定禁止

タイマ・スタート指示ビット

TM23	“ 1 ” を書き込むことによって、カウンタ、IRQT2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

TM22	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

動作モード選択ビット

TM21	TM20	モード
0	0	8ビット・タイマ・カウンタ・モード

(b) タイマ・カウンタ・コントロール・レジスタ (TC2)

8ビット・タイマ・カウンタ・モードを使用する場合はTC2を図5-29のように設定します (TC2のフォーマットについては図5-27 タイマ・カウンタ・コントロール・レジスタのフォーマットを参照してください)。

TC2は、8ビット、4ビット操作命令、ビット操作命令により操作します。

TC2は、内部リセット信号発生により00Hにクリアされます。

実線で示されるフラグは、8ビット・タイマ・カウンタ・モードにおける使用フラグを表します。

点線で示されるフラグは、8ビット・タイマ・カウンタ・モードでは使用しないでください (0を設定してください)。

図5-29 タイマ・カウンタ・コントロール・レジスタの設定

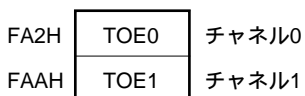


タイマ・カウンタ出力許可フラグ

TOE2	タイマ出力
0	禁止 (ロウ・レベルを出力する)
1	許可

図5-30 タイマ・カウンタ出力許可フラグの設定

アドレス



タイマ・カウンタ出力許可フラグ (W)

0	禁止 (ロウ・レベルを出力する)
1	許可

(2) タイマ・カウンタの時間設定

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

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

T (sec) : タイマ設定時間 (秒)  
 f<sub>CP</sub> (Hz) : CP周波数 (Hz)  
 n : モジユロ・レジスタの内容 (n = 0)

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

表5 - 7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 (モジユロ・レジスタにFFHをセットしたときの時間) を示します。

表5 - 7 分解能と最長設定時間 (8ビット・タイマ・カウンタ・モードの場合) (1/2)  
 (TM10 = 0, TM11 = 0, TM20 = 0, TM21 = 0)

(a) 8ビット・タイマ・カウンタ (チャンネル0) の場合

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM06	TM05	TM04	分解能	最長設定時間	分解能	最長設定時間
1	0	0	171 μs	43.7 ms	244 μs	62.5 ms
1	0	1	42.7 μs	10.9 ms	61.0 μs	15.6 ms
1	1	0	10.7 μs	2.73 ms	15.3 μs	3.91 ms
1	1	1	2.67 μs	683 μs	3.81 μs	977 μs

(b) 8ビット・タイマ・カウンタ (チャンネル1) の場合

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM16	TM15	TM14	分解能	最長設定時間	分解能	最長設定時間
0	1	1	5.33 μs	1.37 ms	7.63 μs	1.95 ms
1	0	0	683 μs	175 ms	977 μs	250 ms
1	0	1	171 μs	43.7 ms	244 μs	62.5 ms
1	1	0	42.7 μs	10.9 ms	61.0 μs	15.6 ms
1	1	1	10.7 μs	2.73 ms	15.3 μs	3.91 ms

表5 - 7 分解能と最長設定時間（8ビット・タイマ・カウンタ・モードの場合）（2/2）

（TM10 = 0, TM11 = 0, TM20 = 0, TM21 = 0）

（c）8ビット・タイマ・カウンタ（チャンネル2）の場合

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM26	TM25	TM24	分解能	最長設定時間	分解能	最長設定時間
0	1	0	333 ns	85.3 μs	477 ns	122 μs
0	1	1	167 ns	42.7 μs	238 ns	61.0 μs
1	0	0	171 μs	43.7 ms	244 μs	62.5 ms
1	0	1	42.7 μs	10.9 ms	61.0 μs	15.6 ms
1	1	0	10.7 μs	2.73 ms	15.3 μs	3.91 ms
1	1	1	2.67 μs	683 μs	3.81 μs	977 μs

（3）タイマ・カウンタ動作（8ビット時）

タイマ・カウンタ動作は、次のようになります。

図5 - 31にタイマ・カウンタ動作時の構成を示します。

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

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

タイマ・カウンタ動作のタイミングは、図5 - 32のようになります。

タイマ・カウンタ動作は、通常、次の手順により開始します。

TMODnにカウント数を設定します。

TMnに動作モード，カウント・パルス，スタート指示を設定します。

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

タイマ・カウンタ出力端子（PTOn）を使用する場合は、兼用端子のP3nを次のように設定してください。

P3nの出力ラッチをクリアする。

ポート3を出力モードに設定する。

ポート3の内蔵プルアップ抵抗を接続しない状態にする。

タイマ・カウンタ出力許可フラグ（TOEn）を1にする。

**備考** n = 0-2

図5 - 31 タイマ・カウント動作時の構成

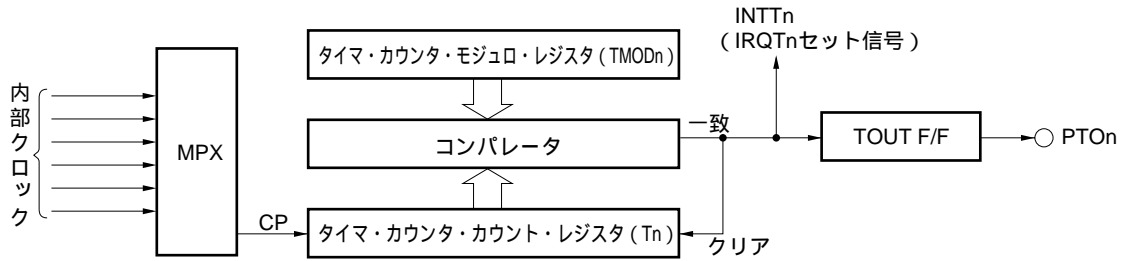
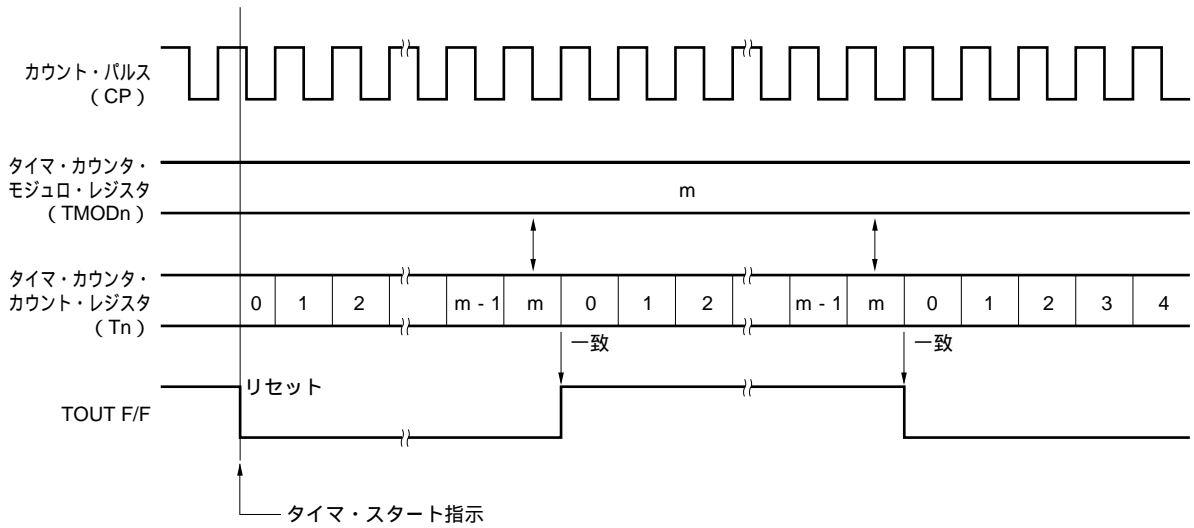


図5 - 32 カウント動作時のタイミング



備考 m : タイマ・カウンタ・モジュール・レジスタの設定値

n = 0-2

**(4) 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。

タイマ・カウンタ・モード・レジスタ (TMn) の上位4ビットは0100Bとし、最長設定時間62.5 ms<sup>注</sup>を選択します。

TMnの下位4ビットは1100Bとします。

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

$$\frac{50 \text{ ms}}{244 \mu\text{s}} = 205 \quad \text{CDH}$$

**プログラム例**

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

**注** この例は $f_x = 4.19 \text{ MHz}$ 動作時の場合です。 $f_x = 6.0 \text{ MHz}$ 動作時では、同じ設定でも最長設定時間やインターバル時間が異なります。

**備考** n = 0-2



### 5.4.3 PWMパルス・ジェネレータ・モード (PWMモード) の動作

タイマ・カウンタ (チャンネル2) をPWMパルス・ジェネレータとして使用するモードです。8ビット長でのPWMパルス・ジェネレータ動作を行います。

タイマ・カウンタ (チャンネル2) をPWMパルス・ジェネレータ・モードで使用しているとき、タイマ・カウンタ (チャンネル0, チャンネル1) はそれぞれ8ビット・タイマ・カウンタとして使用できます。

#### (1) レジスタの設定

PWMモードは、次の5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 (TM2)

タイマ・カウンタ・コントロール・レジスタ (TC2)

タイマ・カウンタ・カウント・レジスタ (T2)

タイマ・カウンタ・ハイ・レベル期間設定用モジュロ・レジスタ (TMOD2H)

タイマ・カウンタ・モジュロ・レジスタ (TMOD2)

#### (a) タイマ・カウンタ・モード・レジスタ (TM2)

PWMモードを使用する場合はTM2を図5 - 33のように設定します (TM2のフォーマットについては、図5 - 25 タイマ・カウンタ・モード・レジスタ (チャンネル2) のフォーマットを参照してください)。

TM2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にクリアされます。

TM2は、内部リセット信号発生時に00Hにクリアされます。

図5 - 33 タイマ・カウンタ・モード・レジスタの設定

アドレス	7	6	5	4	3	2	1	0	略号
F90H		TM26	TM25	TM24	TM23	TM22	TM21	TM20	TM2

**カウント・パルス (CP) 選択ビット**

TM26	TM25	TM24	カウント・パルス (CP)
0	1	0	$f_x/2$
0	1	1	$f_x$
1	0	0	$f_x/2^{10}$
1	0	1	$f_x/2^8$
1	1	0	$f_x/2^6$
1	1	1	$f_x/2^4$
上記以外			設定禁止

**タイマ・スタート指示ビット**

TM23	“ 1 ” を書き込むことによって、カウンタ、IRQT2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**

TM22	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

**動作モード選択ビット**

TM21	TM20	モード
0	1	PWMパルス・ジェネレータ・モード

**備考** タイマ・カウンタ (チャンネル2) をPWMパルス・ジェネレータ・モードで使用するときは、タイマ・カウンタ (チャンネル1) の動作モード選択ビットTM10, TM11に “ 0 ” を設定してください。

(b) タイマ・カウンタ・コントロール・レジスタ (TC2)

PWMモードを使用する場合はTC2を図5 - 34のように設定します (TC2のフォーマットについては図5 - 27 タイマ・カウンタ・コントロール・レジスタのフォーマットを参照してください)。

TC2は、8ビット、4ビット操作命令、ビット操作命令により操作します。

TC2は、内部リセット信号発生により00Hにクリアされます。

実線で示されるフラグは、PWMモードにおける使用フラグを表します。

点線で示されるフラグは、PWMモードでは使用しないでください (0を設定してください)。

図5 - 34 タイマ・カウンタ・コントロール・レジスタの設定



タイマ・カウンタ出力許可フラグ

TOE2	タイマ出力
0	禁止 (ロウ・レベルを出力する)
1	許可

## (2) PWMパルス・ジェネレータ動作

PWMパルス・ジェネレータ・モード時のタイマ・カウンタ(チャンネル2)は、ハイ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2H)とロウ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2)の2つを持ちます。図5-35にPWMパルス・ジェネレータの構成を示します。

各モジュール・レジスタは、それぞれに設定した時間が経過すると、信号を反転します。このためPTO0端子から出力されるパルスは、各モジュール・レジスタごとに任意に設定できます。

PWMパルス・ジェネレータは、次のように動作します。動作を停止するまで と を繰り返し、パルスを生成します。

タイマ・カウンタ・モード・レジスタ(TM2)の設定によりカウント・パルス(CP)が選択され、タイマ・カウンタ・カウント・レジスタ(T2)に入力されます。

T2とハイ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2H)の内容を比較し、一致すると一致信号を発生し、タイマ・アウト・フリップフロップ(TOUT F/F)は反転します。

また、カウント比較用モジュール・レジスタをロウ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2)に切り替えます。

T2とタイマ・カウンタ・モジュール・レジスタ(TMOD2)の内容を比較し、一致すると一致信号を発生し、割り込み要求フラグ(IRQT2)をセットします。同時にTOUT F/Fを反転し、また、カウント比較用モジュール・レジスタをハイ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2H)に切り替えます。

と の動作を交互に繰り返し、パルス波形を生成します。

PWMパルス・ジェネレータ動作のタイミングは、図5-36のようになります。

PWMパルス・ジェネレータ動作は、通常、次の手順により開始します。

TMOD2Hにハイ・レベル幅のカウント数を設定します。

TMOD2にはロウ・レベル幅のカウント数を設定します。

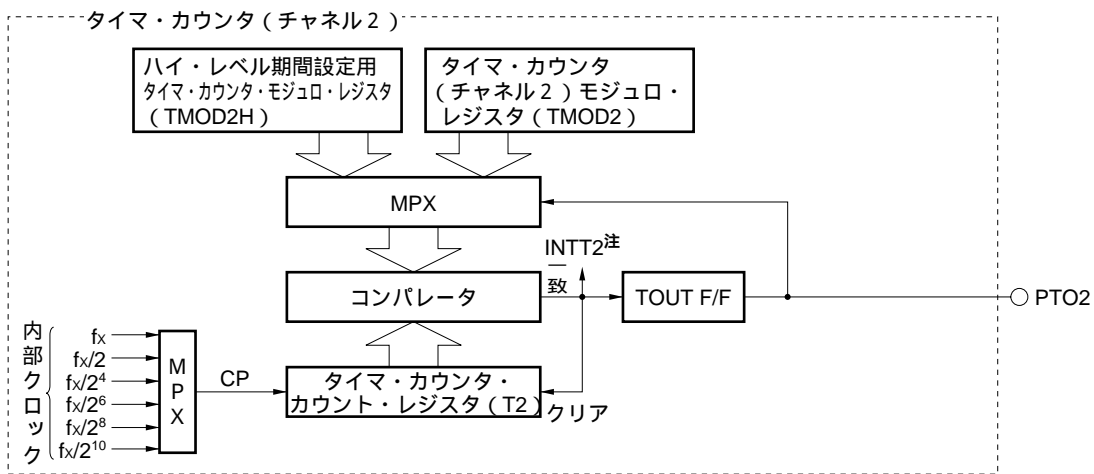
TM2に動作モード、カウント・パルス、スタート指示を設定します。

**注意** タイマ・カウンタ・モジュール・レジスタ(TMOD2)とハイ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ(TMOD2H)には00H以外の値を設定してください。

タイマ・カウンタ出力端子 (PTO2) を使用する場合は、兼用端子のP32を次のように設定してください。

- P32の出力ラッチをクリアする。
- ポート3を出力モードに設定する。
- ポート3の内蔵プルアップ抵抗を接続しない状態にする。
- タイマ・カウンタ出力許可フラグ (TOE2) を1にする。

図5 - 35 PWMパルス・ジェネレータ動作時の構成

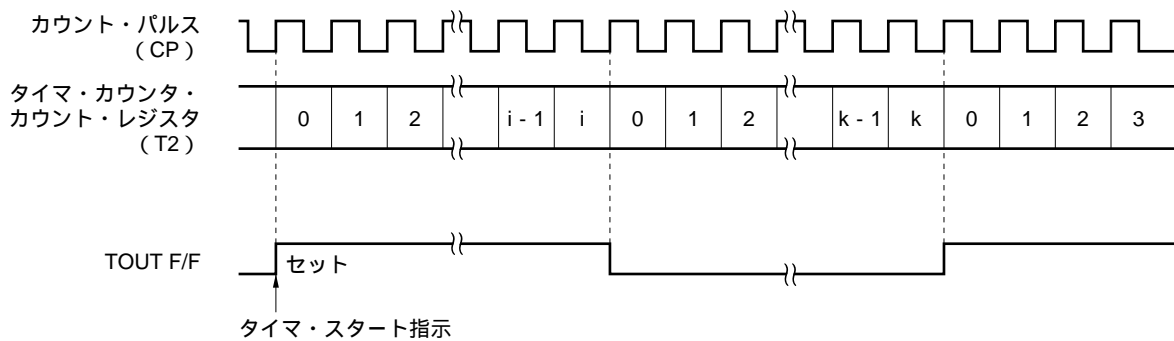


注 IRQT2セット信号です。TMOD2とT2が一致したときのみセットされます。

図5 - 36 PWMパルス・ジェネレータ動作時のタイミング

タイマ・カウンタ (チャンネル2) の動作とキャリア・クロック

(モジュロ・レジスタH (TMOD2H) =  $i$ , モジュロ・レジスタ (TMOD2) =  $k$ )



**(3) PWMモードの応用**

周波数が38.0 kHz (周期が26.3 μs) , デューティ比が1/3のパルスをPTO2端子に出力します。

タイマ・カウンタ・モード・レジスタ (TM2) の上位4ビットは0011Bとし, 最長設定時間61.0 μs<sup>※</sup>を選択します。

TM2の下位4ビットは1101Bとし, PWMモード, カウント動作を選択, タイマ・スタートを指示します。

タイマ・カウンタ出力許可フラグ (TOE2) は “ 1 ” とし, タイマ出力を許可します。

ハイ・レベル期間設定用タイマ・カウンタ・モジュロ・レジスタ (TMOD2H) の設定値は, 次のとおりです。

$$\frac{1}{3} \cdot \frac{26.3 \mu s}{238 ns} - 1 = 36.8 - 1 \quad 36 = 24H$$

タイマ・カウンタ・モジュロ・レジスタ (TMOD2) の設定値は, 次のとおりです。

$$\frac{2}{3} \cdot \frac{26.3 \mu s}{238 ns} - 1 = 73.7 - 1 \quad 73 = 49H$$

**<プログラム例>**

```
SEL    MB15                ;またはCLR1 MBE
SET1   TOE2                ;タイマの出力を許可
MOV    XA, #024H
MOV    TMOD2H, XA          ;モジュロを設定 (ハイ・レベル期間)
MOV    XA, #49H
MOV    TMOD2, XA           ;モジュロを設定 (ロウ・レベル期間)
MOV    XA, #00111101B
MOV    TM2, XA             ;モードを設定, タイマ・スタート
```

**注** この例は,  $f_x = 4.19$  MHz動作時の場合です。  $f_x = 6.0$  MHz動作時では, 同じ設定でも周期が異なります。

#### 5.4.4 16ビット・タイマ・カウンタ・モードの動作

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

16ビット・タイマ・カウンタ・モードを使用する場合は、タイマ・カウンタのチャンネル1とチャンネル2を組み合わせて使用します。

##### (1) レジスタの設定

16ビット・タイマ・カウンタ・モードは、次の7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 (TM1, TM2)

タイマ・カウンタ・コントロール・レジスタ (TC2)<sup>注</sup>

タイマ・カウンタ・カウント・レジスタ (T1, T2)

タイマ・カウンタ・モジュロ・レジスタ (TMOD1, TMOD2)

**注** タイマ・カウンタのチャンネル1は、タイマ・カウンタ出力許可フラグ (TOE1) を使用しません。

##### (a) タイマ・カウンタ・モード・レジスタ (TM1, TM2)

16ビット・タイマ・カウンタ・モードを使用する場合はTM1, TM2を図5 - 37のように設定します (TM1, TM2のフォーマットについては図5 - 24 **タイマ・カウンタ・モード・レジスタ (チャンネル1)** のフォーマット, 図5 - 25 **タイマ・カウンタ・モード・レジスタ (チャンネル2)** のフォーマットを参照してください)。

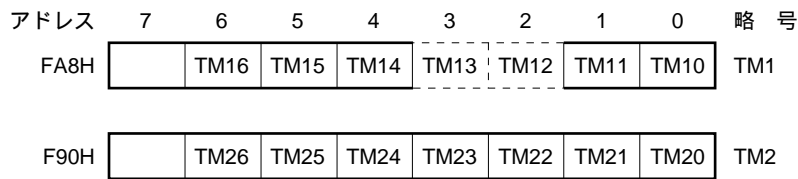
TM1, TM2は、8ビット操作命令により操作します。ビット3は、タイマ・スタート指示ビットで、ビット単位の操作もでき、タイマが動作を開始すると自動的に0クリアされます。

TM1, TM2は、内部リセット信号発生時に00Hにクリアされます。

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

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

図5 - 37 タイマ・カウンタ・モード・レジスタの設定



**カウント・パルス (CP) 選択ビット**

TMn6	TMn5	TMn4	TM1	TM2
0	1	0	カウント・レジスタ (T2) のオーバフロー	$f_x/2$
0	1	1	$f_x/2^5$	$f_x$
1	0	0	$f_x/2^{12}$	$f_x/2^{10}$
1	0	1	$f_x/2^{10}$	$f_x/2^8$
1	1	0	$f_x/2^8$	$f_x/2^6$
1	1	1	$f_x/2^6$	$f_x/2^4$
上記以外			設定禁止	

**タイマ・スタート指示ビット**

TM23	“ 1 ” を書き込むことによって、カウンタ、IRQTn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。
------	---

備考 n = 1, 2

**動作モード**

TM22	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

**動作モード選択ビット**

TM21	TM20	TM11	TM10	モード
1	0	1	0	16ビット・タイマ・カウンタ・モード



(b) タイマ・カウンタ・コントロール・レジスタ (TC2)

16ビット・タイマ・カウンタ・モードを使用する場合はTC2を図5 - 38のように設定します (TC2のフォーマットについては図5 - 27 タイマ・カウンタ・コントロール・レジスタのフォーマットを参照してください)。

TC2は、8ビット、4ビット操作命令、ビット操作命令により操作します。

TC2は、内部リセット信号発生により00Hにクリアされます。

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

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

図5 - 38 タイマ・カウンタ・コントロール・レジスタの設定



タイマ・カウンタ出力許可フラグ

TOE2	タイマ出力
0	禁止 (ロウ・レベルを出力する)
1	許可

(2) タイマ・カウンタの時間設定

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

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

T (sec) : タイマ設定時間 (秒)

f<sub>CP</sub> (Hz) : CP周波数 (Hz)

n : モジユロ・レジスタの内容 (n = 0)

タイマは一度設定すれば, その間隔で毎回割り込み要求フラグ (IRQT2) をセットします。

表5 - 8 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 (モジユロ・レジスタ 1, 2 に各々FFHをセットしたときの時間) を示します。

表5 - 8 分解能と最長設定時間 (16ビット・タイマ・カウンタ・モードの場合)

(TM10 = 0, TM11 = 1, TM20 = 0, TM21 = 1)

モード・レジスタ			6.0 MHz動作時		4.19 MHz動作時	
TM26	TM25	TM24	分解能	最長設定時間	分解能	最長設定時間
0	1	0	333 ns	21.8 ms	477 ns	31.3 ms
0	1	1	167 ns	10.9 ms	238 ns	15.6 ms
1	0	0	171 μs	11.2 s	244 μs	16.0 s
1	0	1	42.7 μs	2.80 s	61.0 μs	4.0 s
1	1	0	10.7 μs	699 ms	15.3 μs	1.0 s
1	1	1	2.67 μs	175 ms	3.81 μs	250 ms

### (3) タイマ・カウンタ動作 (16ビット時)

タイマ・カウンタ動作は、次のようになります。

図5 - 39にタイマ・カウンタ動作時の構成を示します。

タイマ・カウンタ・モード・レジスタ (TM1, TM2) の設定によりカウント・パルス (CP) が選択され、タイマ・カウンタ・カウント・レジスタ (T2) に入力されます。タイマ・カウンタ・カウント・レジスタ (T1) には、T2のオーバフローが入力されます。

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

T2とタイマ・カウンタ・モジュロ・レジスタ (TMOD2) の内容を比較し、一致すると一致信号を発生します。

と の一致信号が重複すると、割り込み要求フラグ (IRQT2) をセットします。同時にタイマ・アウト・フリップフロップ (TOOUT F/F) は反転します。

タイマ・カウンタ動作のタイミングは、図5 - 40のようになります。

タイマ・カウンタ動作は、通常、次の手順により開始します。

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

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

TM1 にカウント・パルスを設定します。

TM2 に動作モード、カウント・パルス、スタート指示を設定します。

**注意** タイマ・カウンタ・モジュロ・レジスタ (TMOD2) には00H以外の値を設定してください。  
また、IET1は“0”としてください。

タイマ・カウンタ出力端子 (PTO2) を使用する場合は、兼用端子のP32を次のように設定してください。

P32の出力ラッチをクリアする。

ポート3を出力モードに設定する。

ポート3の内蔵プルアップ抵抗を接続しない状態にする。

タイマ・カウンタ出力許可フラグ (TOE2) を1にする。

図5 - 39 タイマ・カウント動作時の構成

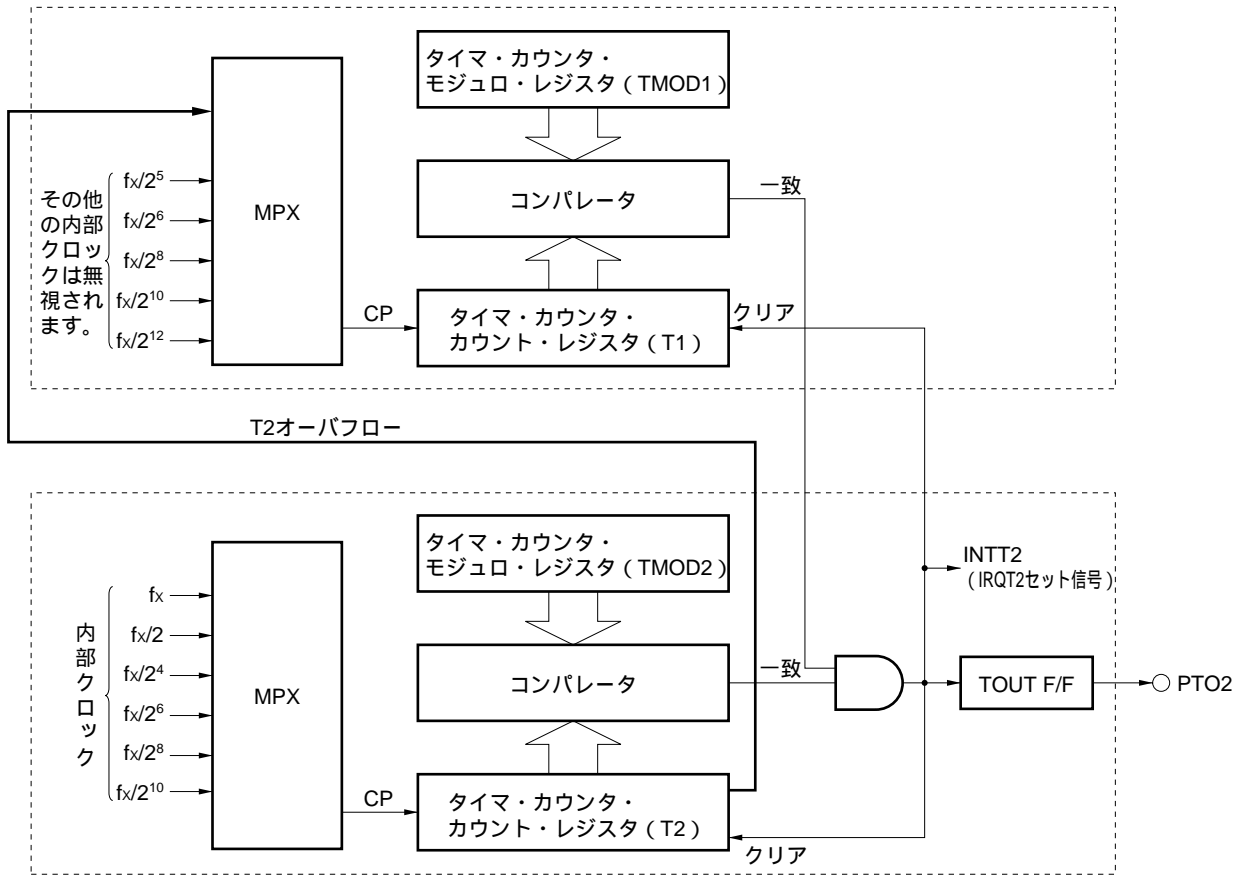
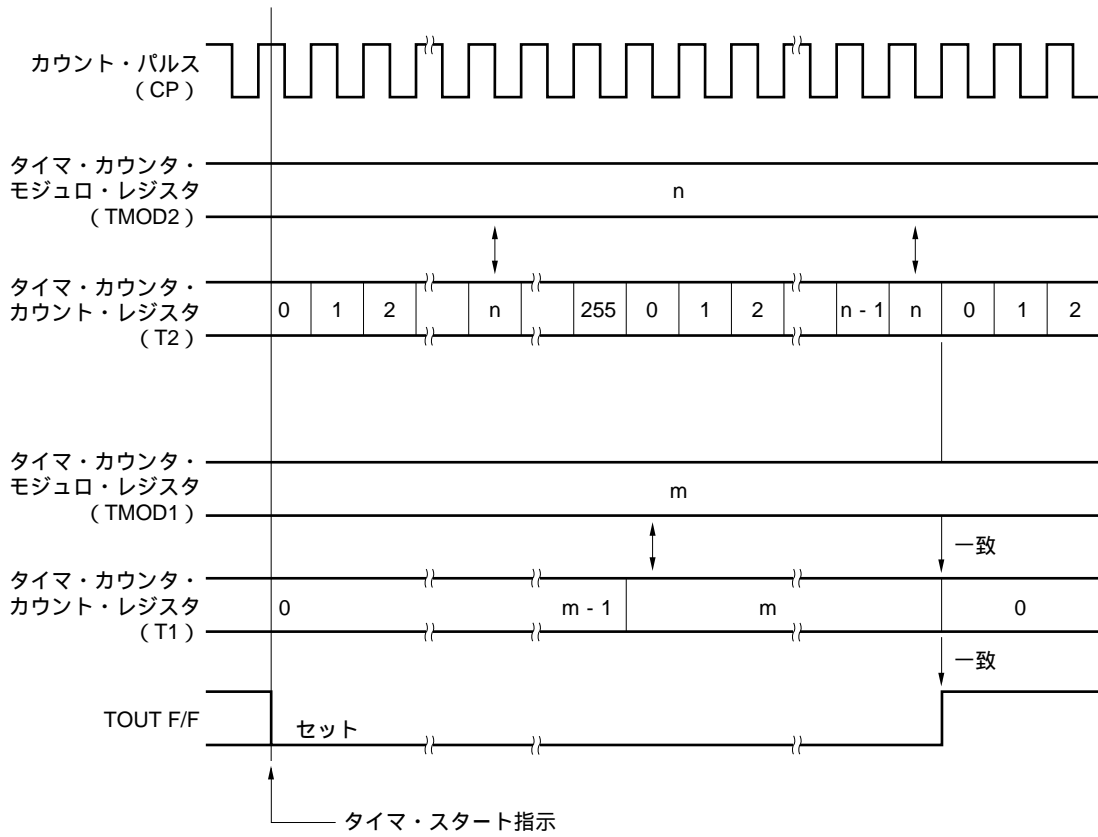


図5 - 40 カウント動作時のタイミング



**備考** m : タイマ・カウンタ・モジュロ・レジスタ (TMOD1) の設定値

n : タイマ・カウンタ・モジュロ・レジスタ (TMOD2) の設定値

## (4) 16ビット・タイマ・カウンタ・モードの応用

5 sec間隔で割り込みを発生するインターバル・タイマとして応用します。

タイマ・カウンタ・モード・レジスタ (TM1) の上位4ビットは0010Bとし、タイマ・カウンタ・カウンタ・レジスタ (T2) のオーバフローを選択します。

TM2の上位4ビットは0100Bとし、最長設定時間16.0 sec<sup>注</sup>を選択します。

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

TM2の下部4ビットは1110Bとし、16ビット・タイマ・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ・カウンタ・モジュロ・レジスタ (TMOD1, TMOD2) の設定値は、次のとおりです。

$$\frac{5 \text{ sec}}{244 \mu\text{s}} = 20491.8 \approx 1 \quad 500\text{BH}$$

## &lt;プログラム例&gt;

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #050H
MOV    TMOD1, XA          ;モジュロを設定(上位8ビット分)
MOV    XA, #00BH
MOV    TMOD2, XA          ;モジュロを設定(下部8ビット分)
MOV    XA, #00100010B
MOV    TM1, XA            ;モードを設定
MOV    XA, #01001110B
MOV    TM2, XA            ;モードを設定, タイマ・スタート
DI     IET1                ;タイマ(チャンネル1)の割り込みを禁止
EI
EI     IET2                ;タイマ(チャンネル2)割り込みを許可
```

**注** この例は、 $f_x = 4.19 \text{ MHz}$ 動作時の場合です。 $f_x = 6.0 \text{ MHz}$ 動作時では、同じ設定でも最長設定時間やインターバル時間が異なります。

### 5.4.5 キャリア・ジェネレータ・モード（CGモード）の動作

キャリア・ジェネレータとして使用するモードです。8ビット長でのキャリア・ジェネレータ動作を行います。

CGモードを使用する場合は、タイマ・カウンタのチャンネル1とチャンネル2を組み合わせで使用します。

タイマ・カウンタ（チャンネル1）は、リモコン信号を生成します。

タイマ・カウンタ（チャンネル2）は、キャリア・クロックを生成します。

#### （1）レジスタの設定

CGモードは、次の8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（TM1, TM2）

タイマ・カウンタ・コントロール・レジスタ（TC2）<sup>注</sup>

タイマ・カウンタ・レジスタ（T1, T2）

タイマ・カウンタ・モジュロ・レジスタ（TMOD1, TMOD2）

タイマ・カウンタ・ハイ・レベル期間設定用モジュロ・レジスタ（TMOD2H）

**注** タイマ・カウンタのチャンネル1は、タイマ・カウンタ出力許可フラグ（TOE1）を使用します。

#### （a）タイマ・カウンタ・モード・レジスタ（TM1, TM2）

CGモードを使用する場合はTM1, TM2を図5 - 41のように設定します（TM1, TM2のフォーマットについては図5 - 24 タイマ・カウンタ・モード・レジスタ（チャンネル1）のフォーマット、図5 - 25 タイマ・カウンタ・モード・レジスタ（チャンネル2）のフォーマットを参照してください）。

TM1, TM2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。

TM1, TM2は、内部リセット信号発生時に00Hにクリアされます。

図5 - 41 タイマ・カウンタ・モード・レジスタの設定

アドレス	7	6	5	4	3	2	1	0	略号
FA8H		TM16	TM15	TM14	TM13	TM12	TM11	TM10	TM1
F90H		TM26	TM25	TM24	TM23	TM22	TM21	TM20	TM2

**カウント・パルス (CP) 選択ビット**

TMn6	TMn5	TMn4	TM1	TM2
0	1	0	キャリア・クロック入力	$f_x/2$
0	1	1	$f_x/2^5$	$f_x$
1	0	0	$f_x/2^{12}$	$f_x/2^{10}$
1	0	1	$f_x/2^{10}$	$f_x/2^8$
1	1	0	$f_x/2^8$	$f_x/2^6$
1	1	1	$f_x/2^6$	$f_x/2^4$
上記以外			設定禁止	

**タイマ・スタート指示ビット**

TMn3	“ 1 ” を書き込むことによって、カウンタ、IRQTn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**

TMn2	カウント動作
0	停止 (カウント内容保持)
1	カウント動作

**動作モード選択ビット**

TM21	TM20	TM11	TM10	モード
1	1	0	0	キャリア・ジェネレータ・モード

備考 n = 1, 2



(b) タイマ・カウンタ・コントロール・レジスタ (TC2)

CGモードを使用する場合は、タイマ・カウンタ出力許可フラグ (TOE1) とTC2を図5 - 42, 図5 - 43のように設定します (TC2のフォーマットについては図5 - 27 タイマ・カウンタ・コントロール・レジスタのフォーマットを参照してください)。

TOE1は、ビット操作命令により操作します。またTC2は、8ビット、4ビット操作命令、ビット操作命令により操作します。

TOE1, TC2は、内部リセット信号発生により00Hにクリアされます。

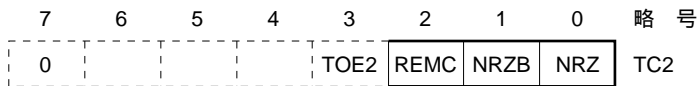
実線で示されるフラグは、CGモードにおける使用フラグを表します。

点線で示されるフラグは、CGモードでは使用しないでください (0を設定してください)。

図5 - 42 タイマ・カウンタ出力許可フラグの設定



図5 - 43 タイマ・カウンタ・コントロール・レジスタの設定



リモコン出力制御フラグ

REMC	リモコン出力
0	NRZ = 1のとき, PTO2端子にキャリア・パルスを出力する
1	NRZ = 1のとき, PTO2端子にハイ・レベルを出力する

ノー・リターン・ゼロ・バッファ・フラグ

NRZB	次に出力するノー・リターン・ゼロ・データを格納するエリア。 タイマ・カウンタ (チャンネル1) の割り込み発生時, NRZへ転送される。
------	---

ノー・リターン・ゼロ・フラグ

NRZ	ノー・リターン・ゼロ・データ
0	PTO2端子にロウ・レベルを出力する (キャリア・クロックの停止)
1	PTO2端子にキャリア・パルスを出力する

## (2) キャリア・ジェネレータ動作

キャリア・ジェネレータ動作は、次のようになります。図5 - 44にその構成を示します。

### (a) タイマ・カウンタ(チャンネル1)の動作

キャリア・ジェネレータ・モード時のタイマ・カウンタ(チャンネル1)は、タイマ・カウンタ(チャンネル2)で生成されるキャリア・クロックをPTO2端子に出力する時間、およびその出力を停止する時間を決定します。

また、タイマ・カウンタ(チャンネル1)のオーパフロー時間はタイマ・カウンタ(チャンネル2)のノー・リターン・ゼロ・バッファ・フラグ(NRZB)からノー・リターン・ゼロ・フラグ(NRZ)へのリロード間隔を決定します。

タイマ・カウンタ・モード・レジスタ(TM1)の設定によりカウント・パルス(CP)が設定され、タイマ・カウンタ・カウント・レジスタ(T1)に入力されます。

T1とタイマ・カウンタ・モジュロ・レジスタ(TMOD1)の内容を比較し、一致すると一致信号を発生し、割り込み要求フラグ(IRQT1)をセットします。同時にタイマ・アウト・フリップフロップ(TOUT F/F)は反転し、NRZBからNRZへのリロード信号を発生します。

### (b) タイマ・カウンタ(チャンネル2)の動作

キャリア・ジェネレータ・モード時のタイマ・カウンタ(チャンネル2)は、PTO2端子に出力するキャリア・クロックを生成します。

また、タイマ・カウンタ(チャンネル1)のオーパフロー信号により、ノー・リターン・ゼロ・バッファ・フラグ(NRZB)からノー・リターン・ゼロ・フラグ(NRZ)へのリロードを行います。

生成したキャリア・クロックをPTO2端子に出力するかどうかは、NRZで決定します。

タイマ・カウンタ(チャンネル2)の動作は、次の手順で行われます。動作を停止させるまでとを繰り返し、キャリア波形を生成します。

タイマ・カウンタ・モード・レジスタ(TM2)の設定によりカウント・パルス(CP)が選択され、タイマ・カウンタ・カウント・レジスタ(T2)に入力されます。

T2とハイ・レベル期間設定用タイマ・カウンタ・モジュロ・レジスタ(TMOD2H)の内容を比較し、一致すると一致信号を発生し、タイマ・アウト・フリップフロップ(TOUT F/F)は反転します。

同時に、カウント比較用モジュロ・レジスタをロウ・レベル期間設定用タイマ・カウンタ・モジュロ・レジスタ(TMOD2)に切り替えます。

T2とタイマ・カウンタ・モジュロ・レジスタ(TMOD2)の内容を比較し、一致すると一致信号を発生し、割り込み要求フラグ(IRQT2)をセットします。同時に、TOUT F/Fを反転し、カウント比較用モジュロ・レジスタをハイ・レベル期間設定用タイマ・カウンタ・モジュロ・レジスタ(TMOD2H)に切り替えます。

と の動作を繰り返します。

ノー・リターン・ゼロ・データは、タイマ・カウンタ（チャンネル1）に割り込みが発生すると、NRZBからNRZへリロードされます。

リモコン出力制御フラグ（REMC）の設定によりNRZ = 1のとき、キャリア・クロックまたはハイ・レベルを出力します。また、NRZ = 0のときロウ・レベルを出力します。

キャリア・ジェネレータ動作のタイミングは、図5 - 45のようになります。

キャリア・ジェネレータ動作は、通常、次の手順により開始します。

TMOD2Hにキャリア・クロックのハイ・レベル幅のカウント数を設定します。

TMOD2にキャリア・クロックのロウ・レベル幅のカウント数を設定します。

REMCに出力波形の形態を設定します。

TM2に動作モード、カウント・パルス、スタート指示を設定します。

TMOD1にNRZの切り替えタイミングのカウント数を設定します。

TM1に動作モード、カウント・パルス、スタート指示を設定します。

タイマ・カウンタ（チャンネル1）に割り込みが発生する前に随時、次に出力するノー・リターン・ゼロ・データをNRZBに設定します。

**注意** タイマ・カウンタ・モジュロ・レジスタ（TMOD1, TMOD2, TMOD2H）には00H以外の値を設定してください。

タイマ・カウンタ出力端子（PTO1）を使用する場合は、兼用端子のP31を次のように設定してください。

P31の出力ラッチをクリアする。

ポート3を出力モードに設定する。

ポート3の内蔵プルアップ抵抗を接続しない状態にする。

タイマ・カウンタ出力許可フラグ（TOE1）を1にする。

図5 - 44 キャリア・ジェネレータ動作時の構成

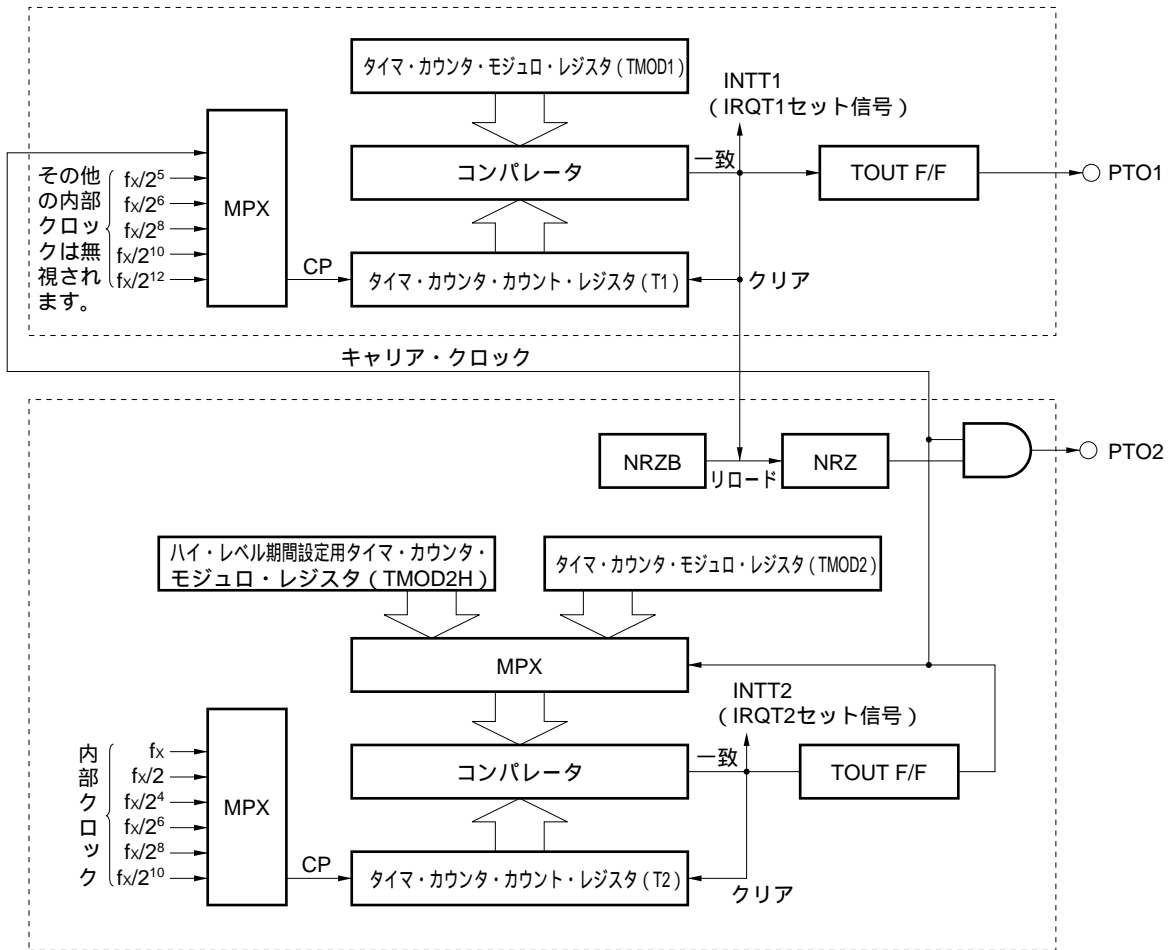
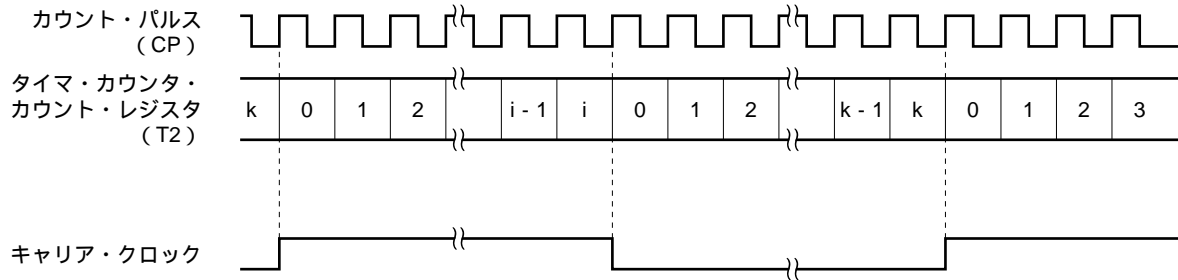


図5 - 45 キャリア・ジェネレータ動作時のタイミング

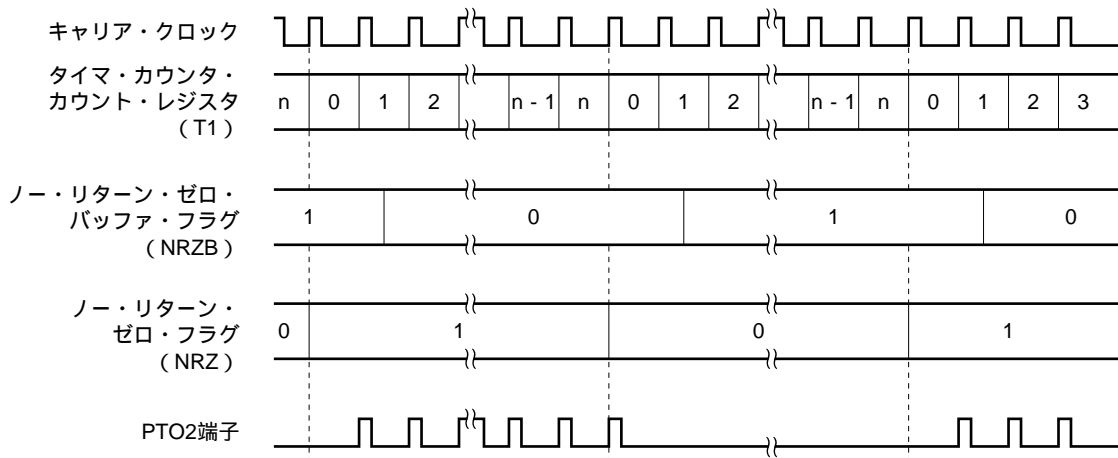
タイマ (チャンネル2) の動作とキャリア・クロック

(モジュロ・レジスタH (TMOD2H) =  $i$ , モジュロ・レジスタ (TMOD2) =  $k$ )



キャリア・クロック, タイマ (チャンネル1), NRZB, NRZとPTO2端子

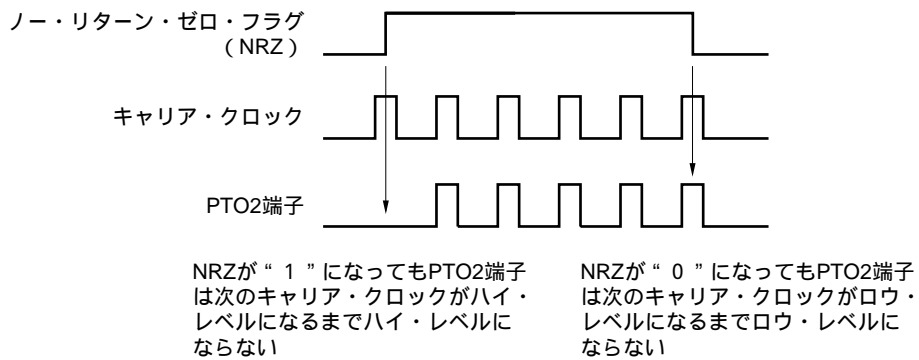
(モジュロ・レジスタ (TMOD1) =  $n$ , タイマ (チャンネル1) のカウント・パルス = キャリア・クロック)



**備考** PTO2端子がロウ・レベルでキャリア・クロックがハイ・レベルの状態 (NRZ = 0, キャリア・クロック = ハイ・レベル) で、タイマ(チャンネル1)の割り込みが発生した場合、キャリア・クロックの次にパルスからPTO2端子にキャリアが出力されます。

PTO2端子がハイ・レベルでキャリア・クロックがハイ・レベルの状態 (NRZ = 1, キャリア・クロック = ハイ・レベル) で、タイマ(チャンネル1)の割り込みが発生した場合、出力中のキャリア・クロックが終了するまでPTO2端子はロウ・レベルになりません。

この処理は、PTO2端子から出力するハイ・レベル・パルス幅を、NRZの切り替わりタイミングにかかわらず一定に保つために行われています(下図参照)。



**(3) CGモードの応用**

リモコン送信用のキャリア・ジェネレータとして応用します。

この項で紹介する応用例は、すべて $f_x = 4.19$  MHz動作時の場合です。 $f_x = 6.0$  MHz動作時では、同じ設定でも周期や信号出力期間が異なります。

周波数が38.0 kHz (周期が26.3  $\mu$ s) , デューティ比が1/3のキャリア・クロックを発生します。

タイマ・カウンタ・モード・レジスタ (TM2) の上位4ビットは0011Bとし、最大設定時間を61.0  $\mu$ sを選択します。

TM2の下部4ビットは1111Bとし、CGモード、カウント動作を選択、タイマ・スタートを指示します。

タイマ・カウンタ出力許可フラグ (TOE2) は“1”とし、タイマ出力を許可します。

ハイ・レベル期間設定用タイマ・カウンタ・モジュロ・レジスタ (TMOD2H) の設定値は、次のとおりです。

$$\frac{1}{3} \cdot \frac{26.3 \mu\text{s}}{238 \text{ ns}} - 1 = 36.8 - 1 \quad 36 = 24\text{H}$$

タイマ・カウンタ・モジュロ・レジスタ (TMOD2) の設定値は、次のとおりです。

$$\frac{2}{3} \cdot \frac{26.3 \mu\text{s}}{238 \text{ ns}} - 1 = 73.7 - 1 \quad 73 = 49\text{H}$$

**<プログラム例>**

```
SEL    MB15                ;またはCLR1 MBE
MOV    XA, #024H
MOV    TMOD2H, XA         ;モジュロを設定 (ハイ・レベル期間)
MOV    XA, #49H
MOV    TMOD2, XA          ;モジュロを設定 (ロウ・レベル期間)
MOV    XA, #00111111B
MOV    TM2, XA            ;モードを設定, タイマ・スタート
```

キャリア・クロックを出力する期間が9 ms、ロウ・レベルを出力する期間が4.5 msのリーダ・コードを出力します（下図参照）。

タイマ・カウンタ・モード・レジスタ（TM1）の上位4ビットは0110Bとし、最大設定時間15.6 msを選択します。

TM1の下位4ビットは1100Bとし、8ビット・タイマ・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ・カウンタ・モジュロ・レジスタ（TMOD1）の初回の設定値は、次のとおりです。

$$\frac{9 \text{ ms}}{61 \mu\text{s}} - 1 = 147.5 - 1 \quad 147 = 93\text{H}$$

TMOD1の書き換え時の設定値は、次のとおりです。

$$\frac{4.5 \text{ ms}}{61 \mu\text{s}} - 1 = 73.8 - 1 \quad 73 = 49\text{H}$$

TC2の上位4ビットは0000Bとします。

TC2の下位4ビットは0000Bとし、ノー・リターン・ゼロ・データが“1”のときキャリア・クロックを出力し、次に出力するノー・リターン・ゼロ・データを“0”にします。

<プログラム例>

```

SEL    MB15                ;またはCLR1 MBE
MOV    XA, #093H
MOV    TMOD1, XA           ;モジュロを設定（キャリア・クロック出力期間）
MOV    XA, #00000000B
MOV    TC2, XA
SET1   NRZ                 ;ノー・リターン・ゼロ・データを“1”に設定
MOV    XA, #01101100B
MOV    TM1, XA             ;モードを設定、タイマ・スタート
EI                      ;割り込みを許可
EI     IET1                ;タイマ・カウンタ（チャンネル1）の割り込みを許可
    
```

; <サブルーチン>

```

MOV    XA, #049H
MOV    TMOD1, XA          ;モジュロを書き換える（ロウ・レベル出力期間）
RETI
    
```





データが“1”のときキャリア・クロックを出力する期間が0.56 ms、ロウ・レベルを出力する期間が1.69 ms、データが“0”のときキャリア・クロックを出力する期間が0.56 ms、ロウ・レベルを出力する期間が0.56 msのカスタム・コードを出力します（下図参照）。

タイマ・カウンタ・モード・レジスタ（TM1）の上位4ビットは0011Bとし、最大設定時間1.95 msを選択します。

TM1の下部4ビットは1100Bとし、8ビット・タイマ・カウンタ・モード、カウント動作を選択、タイマ・スタートを指示します。

タイマ・カウンタ・モジュロ・レジスタ（TMOD1）の初回の設定値は、次のとおりです。

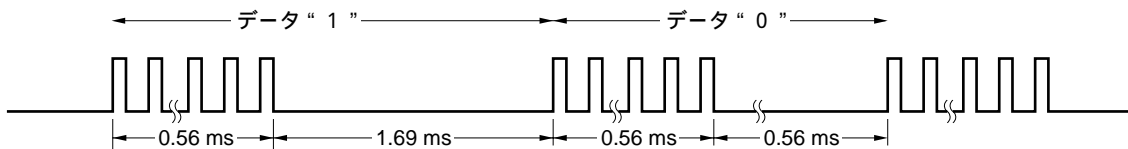
$$\frac{0.56 \text{ ms}}{7.64 \mu\text{s}} - 1 = 73.3 - 1 = 72 = 48\text{H}$$

TMOD1のキャリア出力を行わない期間は、データ“0”のときは出力期間と同じ長さ、データ“1”のときは出力期間の3倍の長さ（ソフトウェア上は、データ“0”のときのキャリア出力を行わない期間を3回繰り返す）で処理をします。

TC2の上位4ビットは0000Bとします。

TC2の下部4ビットは0000Bとし、ノー・リターン・ゼロ・データが“1”のときキャリア・クロックを出力し、次に出力するノー・リターン・ゼロ・データを“0”にします。

送信データ（“0”または“1”）をビット・シーケンシャル・バッファに設定しておきます。



### プログラム例

PTO2端子の出力ラッチは“0”に設定され、かつ端子の入出力モードは出力モードに設定された状態を想定しています。また、キャリア・クロックの生成は、前例のプログラムで発生中の状態を想定しています。

```

; SEND_CARRIER_DATA_PRO
        SEL    MB15          ; または CLR1 MBE
        MOV    HL, #00H     ; LにBSB (ビット・シーケンシャル・バッ
                             ; ファ)のポインタを設定
                             ; HIは、BSBのビット・データ一時退避エリア
                             ; として使用

; CG_Init & Send_1st_Data
        MOV    XA, #48H
        MOV    TMOD1, XA    ; モジュロ・レジスタ (キャリア・クロック出
                             ; 力期間)を設定
        MOV    XA, #00000000B ; キャリア・クロックの出力可能状態, NRZB,
                             ; NRZを0に初期化
        MOV    TC2, XA
        SET1   NRZ          ; ノー・リターン・ゼロ・フラグに“1”を設定
        MOV    XA, #01101100B ; カウント・パルスを選択, 8ビット・タイ
                             ; マ・カウンタ・モード
        MOV    TM1, XA     ; タイマ・カウンタ動作許可, タイマ・スター
                             ; トを指示

; Send_1st_Data
        CALL   !GET_DAT    ; BSBからデータをゲット
        CALL   !SEND_D_0   ; データが0および1のキャリア出力と1回分
                             ; のロウ・レベル出力期間設定処理
        SKE   H, #1H      ; ビット0が1なら, ロウ・レベル出力期間を
                             ; 2回分追加処理へ
        BR    SEND_1_F    ; ビット0が0なら, ロウ・レベル出力の状態
                             ; で, 次のデータのサーチへ処理を受け渡す
        CALL   !SEND_D_1   ; ロウ・レベル出力期間を2回分追加処理
                             ; PTO2端子からロウ・レベル出力の状態で,
                             ; BSBのビット0-Fのデータ送信処理へ処理を
                             ; 移す
SEND_1_F :
        SET1  NRZB        ; BSBのビット0-Fのデータ送信処理
                             ; 前回のデータのロウ・レベル出力期間中に,
                             ; 次に発生するIRQ1で, 次に送信するデー
                             ; タのキャリアが出力されるようにNRZB=1する
    
```

```

INCS    L                ; 送信中のデータをカウントし、Lが0FH 0H
                        ; でデータ送信終了

BR      LOOP_C_0
BR      SEND_END
LOOP_C_0: SKTCLR IRQT1    ; 前データのロウ・レベル出力ウエイト（前
                        ; データの終了確認）

BR      LOOP_C_0
                        ; キャリア出力を開始

CLR1    NRZB             ; 次に発生するIRQT1で、1回目のロウ・レベ
                        ; ル出力が行われるように、あらかじめNRZB
                        ; を0にクリアする

CALL    ! GET_DAT
CALL    ! SEND_D_0
SKE     H, #1H           ; ゲットしたデータが1なら、ロウ・レベル出
                        ; 力期間を2回追加処理（SEND_D_1）へ

BR      SEND_1_F         ; データが0なら、PTO2端子からロウ・レベ
                        ; ル出力の状態で次のデータの送信処理へ

CALL    ! SEND_D_1
BR      SEND_1_F

SEND_END:                ; 16ビット分のデータ送信終了

; サブルーチン
GET_DAT:                 ; @Lで示されるBSBのデータをサーチ。Hレ
                        ; ジスタに値を設定

SKT     BSB0. @L
MOV     A, #0
MOV     A, #1
MOV     H, A
RET

SEND_D_0:                ; データが0および1のキャリア出力と1回分
                        ; のロウ・レベル出力設定処理

LOOP_1ST: SKTCLR IRQT1
BR      LOOP_1ST        ; キャリア出力をウエイト
RET                                           ; 1回目のロウ・レベルを出力開始

SEND_D_1:
CLR1    NRZB             ; データが1であったとき、2回目のロウ・レ
                        ; ベル出力設定

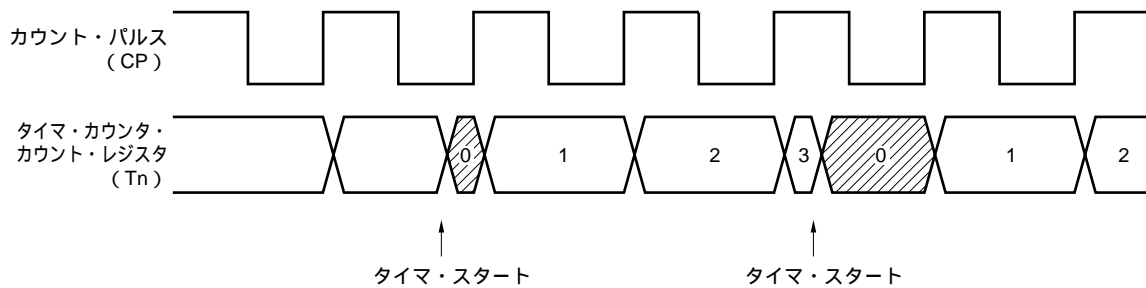
```

```
LOOP_2ND : SKTCLR IRQT1
           BR      LOOP_2ND      ; 1回目のロウ・レベル出力をウエイト
                                   ; 2回目のロウ・レベル出力を開始
           CLR1   NRZB          ; 3回目のロウ・レベル出力を設定
LOOP_3RD : SKTCLR IRQT1
           BR      LOOP_3RD      ; 2回目のロウ・レベル出力をウエイト
                                   ; 3回目のロウ・レベル出力を開始
RET
```

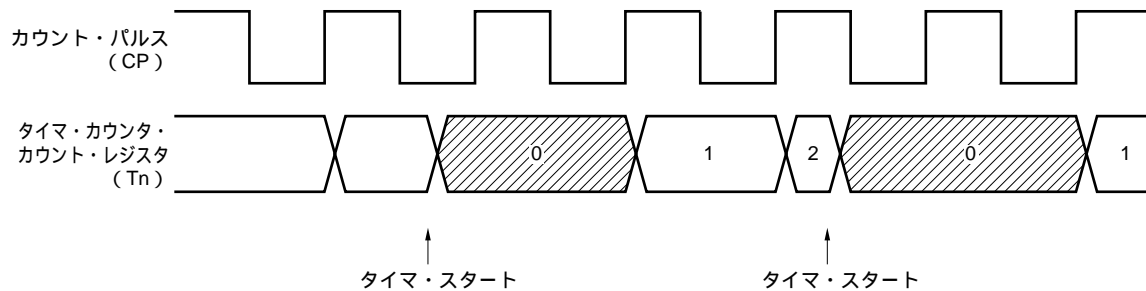
### 5.4.6 タイマ・カウンタ応用上の注意

#### (1) タイマ・スタート時の誤差

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



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



備考 n = 0-2

## (2) タイマ・スタート時の注意

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

## 例 IRQTnが立つ可能性のあるタイミングでのタイマ・スタート

```

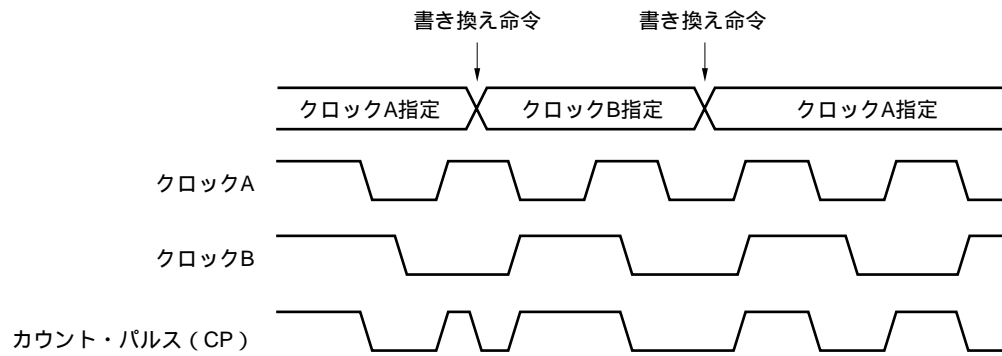
SEL      MB15
MOV      XA, #0
MOV      TMn, XA          ; タイマを停止
MOV      XA, #4CH
MOV      TMn, XA          ; 再スタート
または
SEL      MB15
SET1     TMn.3
SET1     TMn.3          ; 再スタート

```

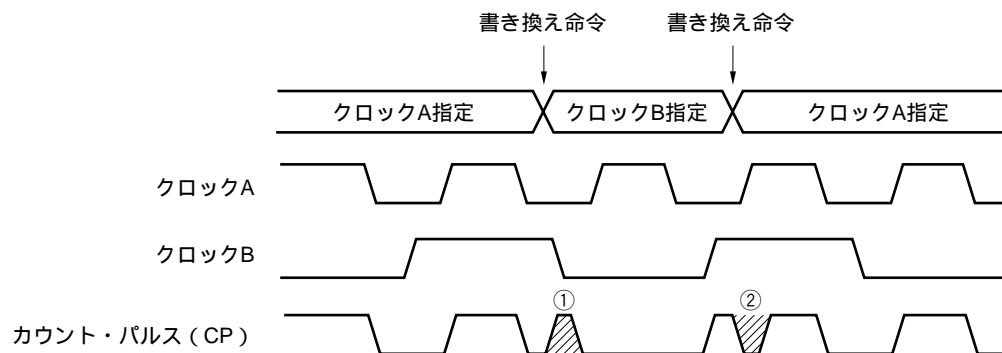
備考 n = 0-2

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



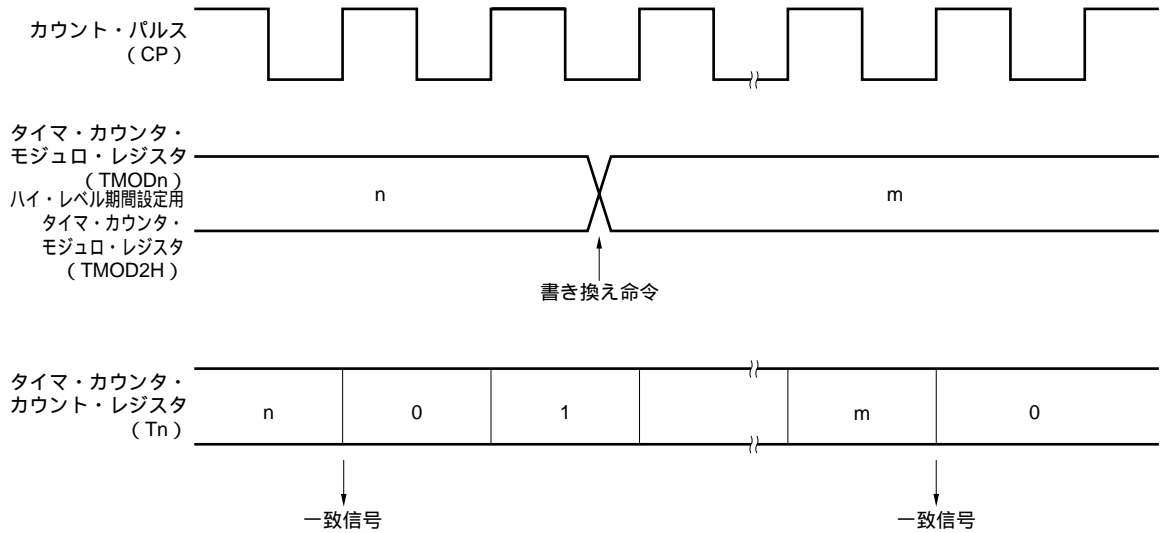
CP変更時のクロックの組み合わせにより、次図のようにヒゲ状 (または ) のCPが発生することもあります。この場合、ミス・カウントあるいはカウンタ・レジスタ (Tn) の内容が破壊される可能性がありますので、CPを変更する場合は必ず、TMnのビット3を“1”にして、タイマの再スタートも同時に行ってください。



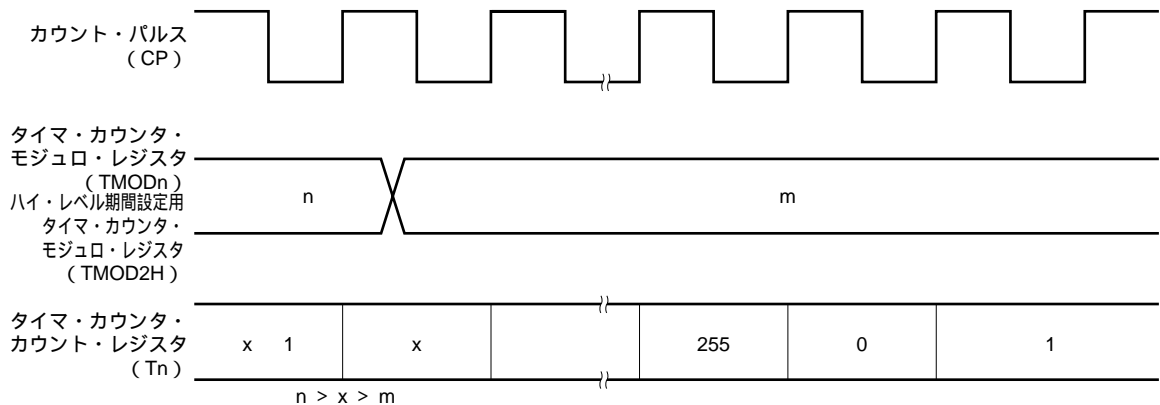
備考 n = 0-2

(4) モジュール・レジスタ変更後の動作

タイマ・カウンタ・モジュール・レジスタ (TMODn) とハイ・レベル期間設定用タイマ・カウンタ・モジュール・レジスタ (TMOD2H) の変更は、8ビット・データ・メモリ操作命令を実行した時点で行われます。



TMODnの変更後の値が、タイマ・カウンタ・カウンタ・レジスタ (Tn) の値よりも小さい場合には、Tnはカウントを継続しオーバーフローして0から再カウントします。したがって、TMODn, TMOD2Hの変更後の値 (m) が変更前の (n) より小さい場合には、TMODn, TMOD2Hを変更後、タイマを再スタートさせる必要があります。

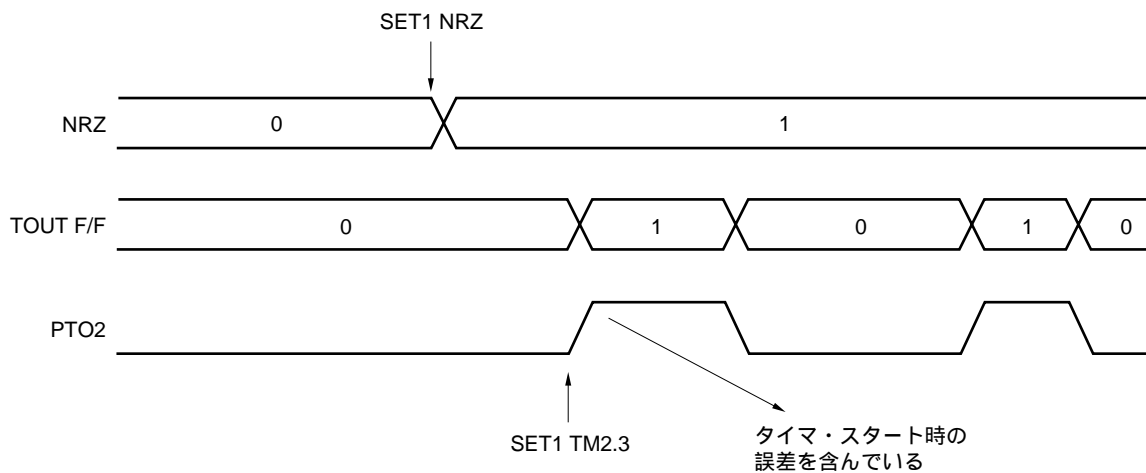




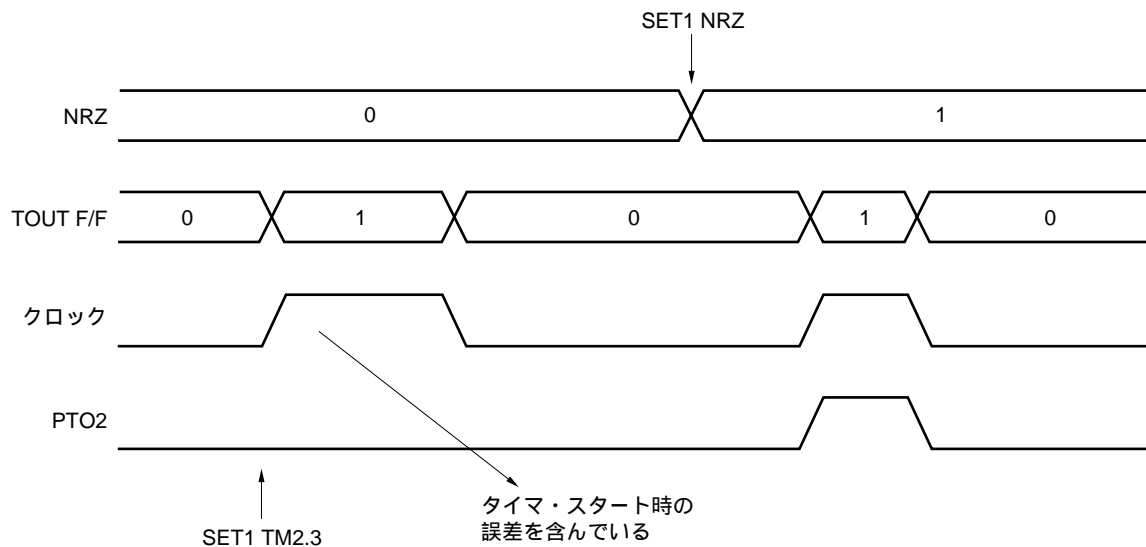
(5) キャリア・ジェネレータ応用上の注意(スタート時)

キャリア・クロックを発生させるとき、タイマ・スタート(TM2のビット3を“1”にセット)後、初回のキャリア・クロックのハイ・レベル期間は、(モジュロ・レジスタの内容 + 1) × 分解能の式で求めた値に対し、最大でカウント・パルス(CP)1クロック分(CPの周波数が1マシン・サイクル以上の場合は、最大でCP2クロック分)の誤差が生じます(詳細は、(1)タイマ・スタート時の誤差を参照してください)。

初回のコードとしてキャリアを出力する場合、ノー・リターン・ゼロ・フラグ(NRZ)を“1”にセットしたあとにタイマ・スタート(TM2のビット3を“1”にセット)を行うと、初回のキャリア・クロックのハイ・レベル期間はタイマ・スタート時の誤差が含まれます。



したがって、初回のコードとしてキャリアを出力する場合、タイマ・スタート(TM2のビット3を“1”にセット)したあとに、NRZを“1”にセットするようにしてください。

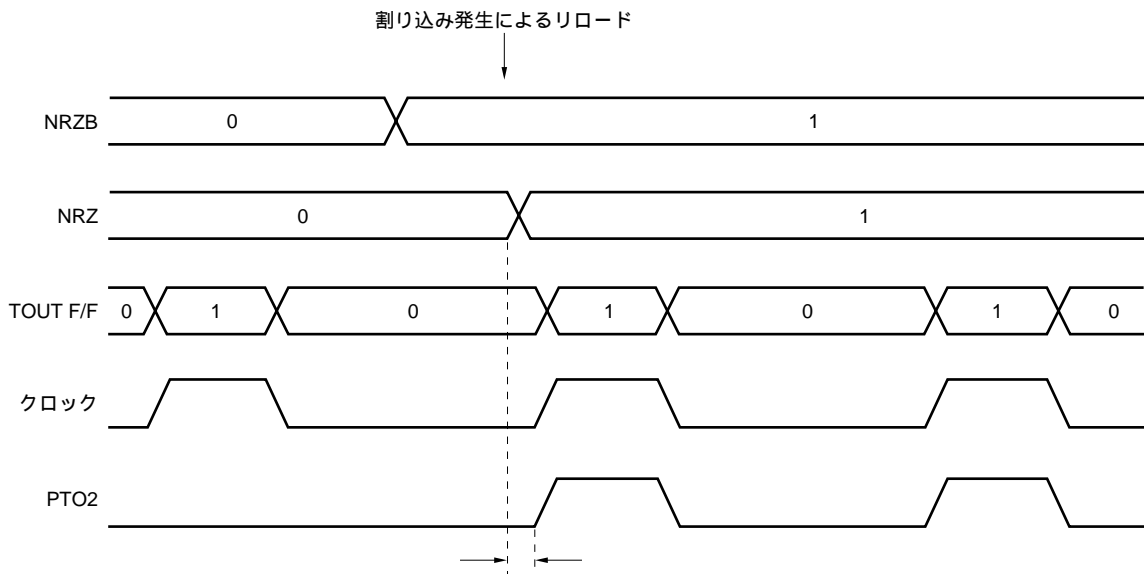


(6) キャリア・ジェネレータ応用上の注意(リロード時)

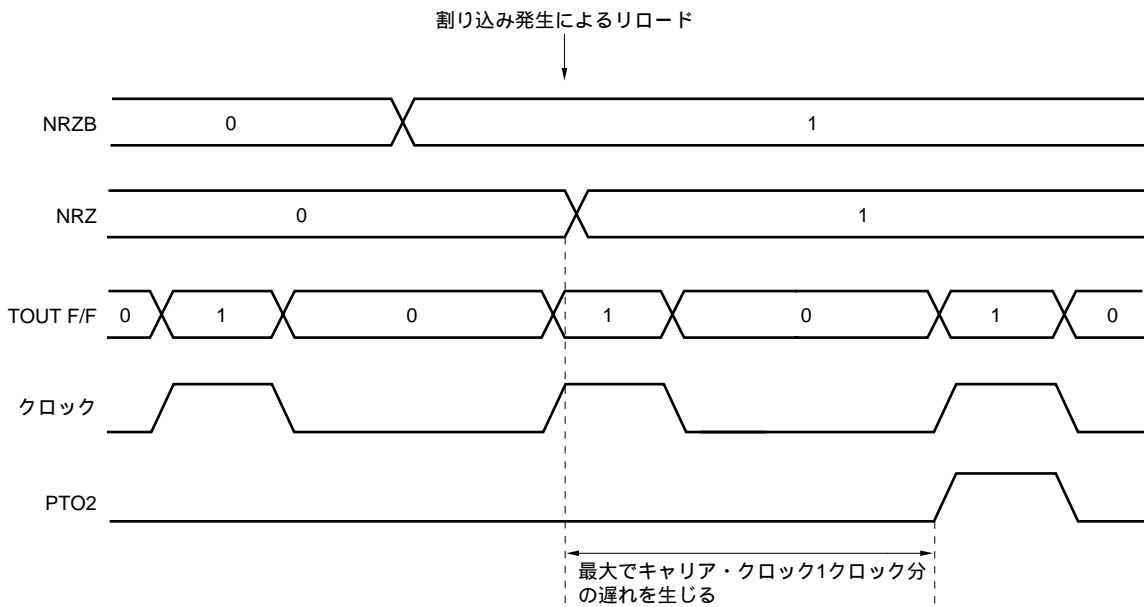
PTO2端子にキャリアを出力するとき、リロード(タイマ・カウンタ(チャンネル1)の割り込み発生により、ノー・リターン・ゼロ・バッファ・フラグ(NRZB)の内容がノー・リターン・ゼロ・フラグ(NRZ)へ転送され、NRZの内容が、“1”に更新)後、初回のキャリアが発生するまでの時間は、最大でキャリア・クロック1クロック分の遅れを生じます。

これは、次の図に示すように、キャリア・クロックに対してリロードが非同期で行われるためであり、またキャリアのハイ・レベル期間を一定に保つためでもあります。

<リロード後の遅れが最小の場合>

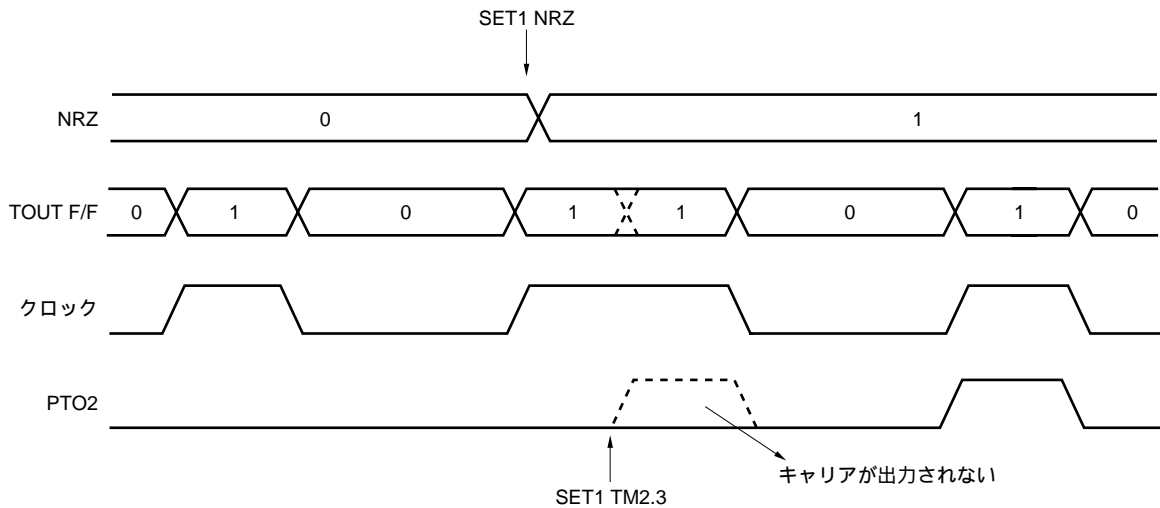


<リロード後の遅れが最大の場合>

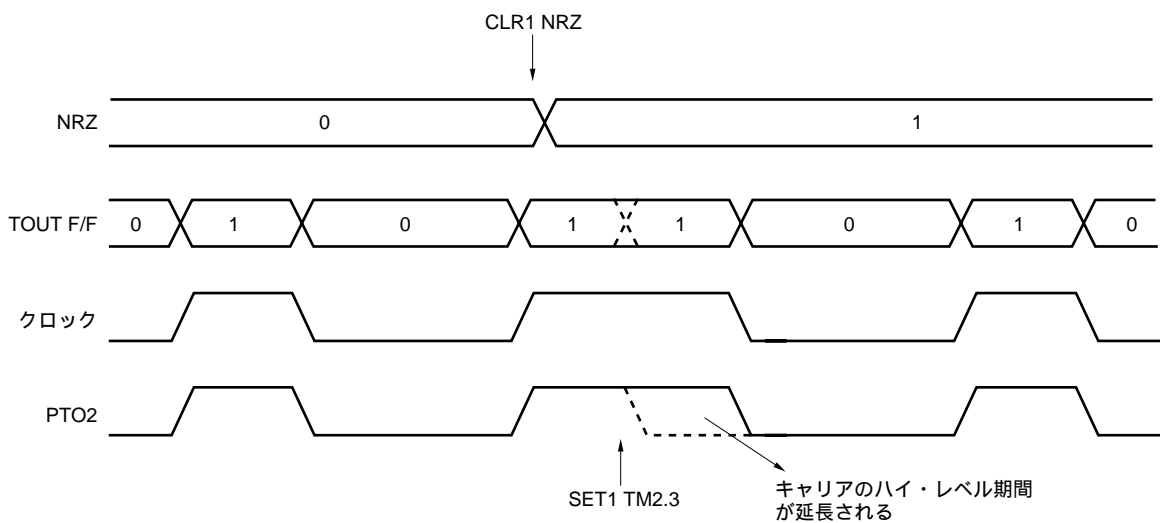


(7) キャリア・ジェネレータ応用上の注意(再スタート時)

キャリア・クロックがハイ・レベル(TOUT F/Fが“1”を保持)のとき、ノー・リターン・ゼロ・フラグ(NRZ)の内容を直接書き換えることにより強制的にリロードを行い、さらにタイマ再スタート(TM2のビット3を“1”にセット)を行う場合、次の図に示すようにPTO2端子にキャリアが出力されないことがあります。



同様にキャリア・クロックがハイ・レベル(TOUT F/Fが“1”を保持)のとき、NRZの内容を直接書き換えることにより強制的にリロードを行い、さらにタイマ再スタート(TM2のビット3を“1”にセット)を行う場合、次の図に示すようにPTO2端子に出力されるキャリアのハイ・レベル期間が長くなる場合があります。

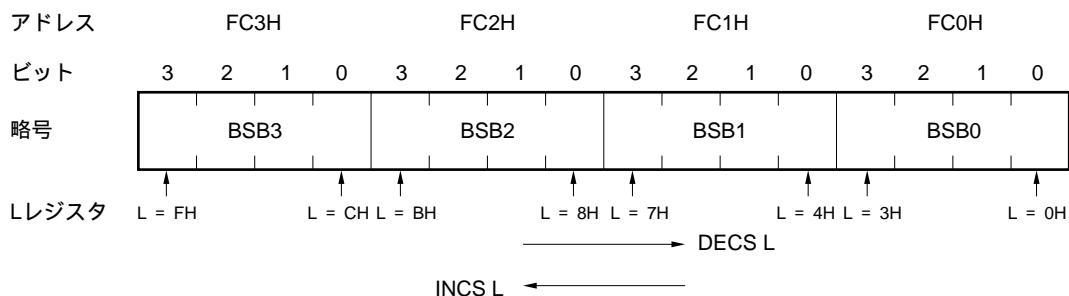


## 5.5 ビット・シーケンシャル・バッファ.....16ビット

ビット・シーケンシャル・バッファ (BSB) は、ビット操作の特殊データ・メモリで、特にアドレスおよびビット指定を順次変更してビット操作が容易にできるので、ビット長の長いデータをビット単位で処理するときに便利です。

このデータ・メモリは16ビット構成になっており、ビット操作命令のpmem. @Lアドレッシングが可能で、Lレジスタにより間接ビット指定ができます。この場合、Lレジスタをプログラム・ループ内でインクリメント/デクリメントするだけで順次指定ビットを移動させながら処理を進めることができます。

図5 - 46 ビット・シーケンシャル・バッファのフォーマット



備考1 . pmem. @Lアドレッシングでは、Lレジスタに対応して指定ビットが移動します。

2 . pmem. @Lアドレッシングでは、MBE, MBSの指定にかかわらずに、いつでもBSBを操作できます。

直接アドレッシングでもデータ操作が可能です。1ビット、4ビット、8ビットの直接アドレッシングとpmem. @Lアドレッシングとを合わせて、1ビット・データの連続入力や連続出力などに応用できます。なお、8ビット操作の場合は、BSB0, BSB2を指定して上位、下位8ビットずつを操作します。

例 BUFF1, 2の16ビット・データをポート3のビット0からシリアルに出力する。

```
CLR1  MBE
MOV   XA, BUFF1
MOV   BSB0, XA           ; BSB0, 1をセット
MOV   XA, BUFF2
MOV   BSB2, XA           ; BSB2, 3をセット
MOV   L, #0
LOOP0: SKT   BSB0, @L     ; BSBの指定ビットをテスト
BR    LOOP1
NOP                                ; ダミー (タイミング調整)
SET1  PORT3. 0           ; ポート3のビット0をセット
BR    LOOP2
LOOP1: CLR1  PORT3. 0     ; ポート3のビット0をクリア
NOP                                ; ダミー (タイミング調整)
NOP
LOOP2: INCS  L            ; L L + 1
BR    LOOP0
RET
```

## 第6章 割り込み機能とテスト機能

μPD754202には、5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、μPD754202の割り込み制御回路には次のような特色があり、非常に高速な割り込み処理が可能となります。

### (1) 割り込み機能

- (a) 割り込み許可フラグ (IE × × ×) と、割り込みマスタ許可フラグ (IME) により受け付け可否を制御できるハードウェア制御のベクタ割り込み機能。
- (b) 割り込みスタート・アドレスを任意に設定可能。
- (c) 割り込みプライオリティ選択レジスタ (IPS) により優先順位が指定できる多重割り込み機能。
- (d) 割り込み要求フラグ (IRQ × × ×) のテスト機能 (ソフトウェアで割り込み発生の確認可能)。
- (e) スタンバイ・モードの解除 (割り込み許可フラグにより、解除する割り込みの選択可能)。

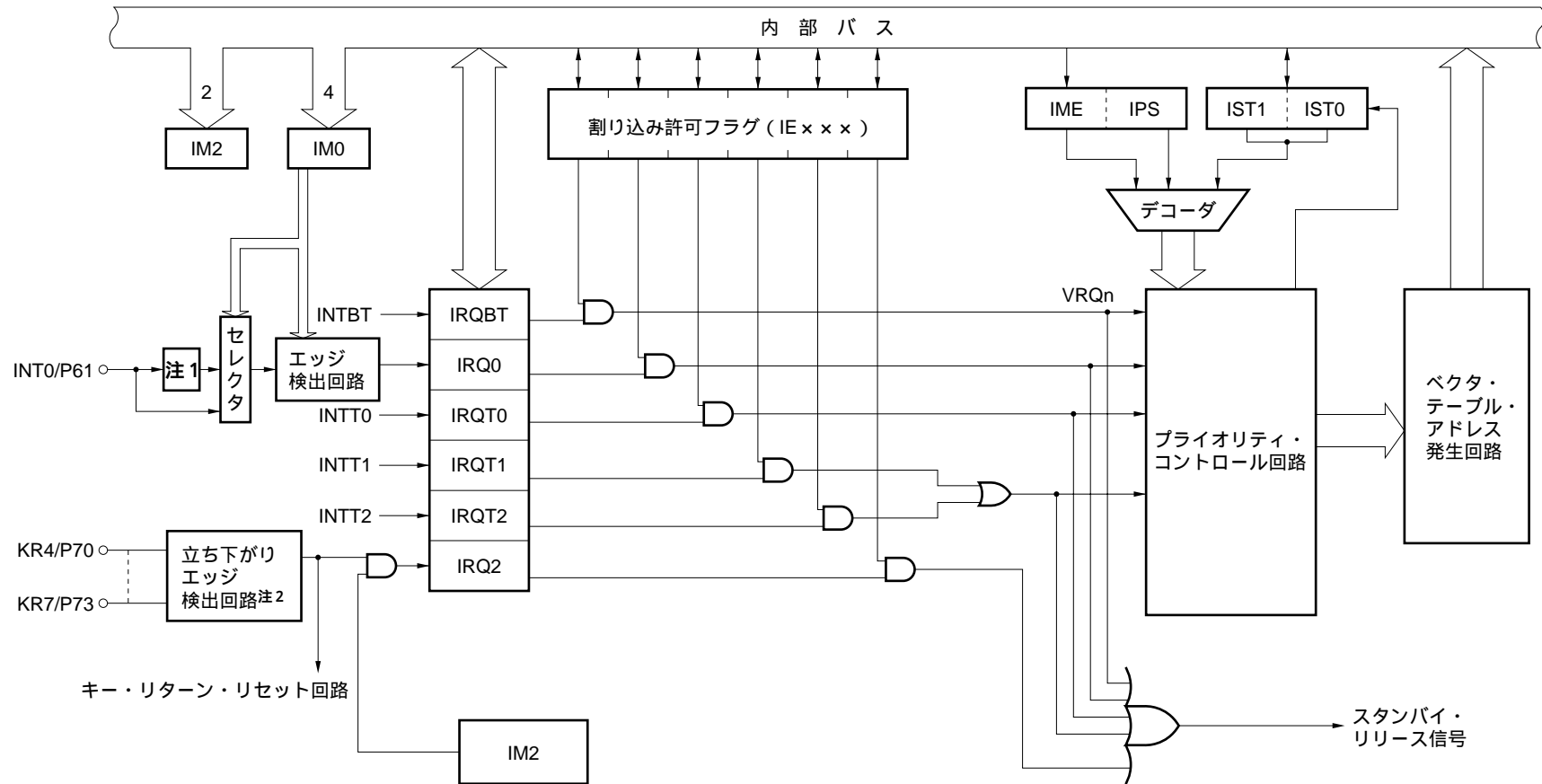
### (2) テスト機能

- (a) ソフトウェアでテスト要求フラグ (IRQ2) 発生の確認可能。
- (b) スタンバイ・モードの解除 (テスト許可フラグにより、解除するテスト・ソースの選択可能)。

## 6.1 割り込み制御回路の構成

割り込み制御回路は図6-1のように構成されており、各ハードウェアはデータ・メモリ空間にマッピングされています。

図 6 - 1 割り込み制御回路ブロック図



注 1 . ノイズ除去回路 (ノイズ除去回路選択時はスタンバイ・リリース不可)

2 . INT2端子はありません。IM20, IM21 = 1, 0のときKRn端子の立ち下がりエッジで割り込み要求フラグ (IRQ2) をセットします。

## 6.2 割り込みソースの種類とベクタ・テーブル

μPD754202には、次の5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)
INT0 (立ち上がり, 立ち下がりエッジ選択)	外	2	VRQ2 (0004H)
★ 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)
INTT1 (タイマ・カウンタ1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	4	VRQ6 (000CH)
INTT2 (タイマ・カウンタ2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		

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



図6-2 割り込みベクタ・テーブル

0002H	MBE	RBE	0	0	0	INTBT	スタート・アドレス	(上位3ビット)
								"
0004H	MBE	RBE	0	0	0	INT0	スタート・アドレス	(上位3ビット)
								"
0006H	(この行は図中の線によって削除されている)							
0008H	(この行は図中の線によって削除されている)							
000AH	MBE	RBE	0	0	0	INTT0	スタート・アドレス	(上位3ビット)
								"
000CH	MBE	RBE	0	0	0	INTT1, INTT2	スタート・アドレス	(上位3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, 2, 5, 6) 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, 2, 5, 6) 命令のオペランドに記述した内容 (MBE, RBE, 開始アドレス) は、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 $\times\times\times$ ) には、割り込みソースに対応して次に示す5つがあります。

INT0割り込み要求フラグ (IRQ0)	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)	タイマ・カウンタ1 割り込み要求フラグ (IRQT1)
	タイマ・カウンタ2 割り込み要求フラグ (IRQT2)

割り込み要求フラグは、割り込み要求発生で“1”にセットされ、割り込み処理が実行されると自動的に“0”にクリアされます。ただし、IRQT1とIRQT2はベクタ・アドレスを共用しているのでクリア動作が異なります(6.6 ベクタ・アドレス共用割り込み処理参照)。

割り込み許可フラグ (IE $\times\times\times$ ) には、割り込み要求フラグに対応して次に示す5つがあります。

INT0割り込み許可フラグ (IE0)	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)	タイマ・カウンタ1 割り込み許可フラグ (IET1)
	タイマ・カウンタ2 割り込み許可フラグ (IET2)

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 (VRQ $n$ :  $n=1, 2, 5, 6$ ) が発生します。この信号はスタンバイ・モードの解除にも使用されます。

割り込み要求フラグと、割り込み許可フラグは、ビット操作命令、および4ビット・メモリ操作命令により操作します。ビット操作命令の場合は、MBEの設定にかかわらず常に直接操作できます。また、割り込み許可フラグはEI IE $\times\times\times$ 命令、DI IE $\times\times\times$ 命令で操作します。割り込み要求フラグのテストには通常SKTCLR命令を使用します。

**例** EI            IE0            ; INT0許可  
 DI            IET1           ; INTT1禁止  
 SKTCLR IRQBT       ; IRQBT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
IRQ0	INT0/P61端子入力信号のエッジ検出によりセット。検出エッジはINT0エッジ検出モード・レジスタ(IM0)により選択。	IE0
IRQT0	タイマ・カウンタ0からの一致信号でセット。	IET0
IRQT1	タイマ・カウンタ1からの一致信号でセット。	IET1
IRQT2	タイマ・カウンタ2からの一致信号でセット。	IET2

### (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。

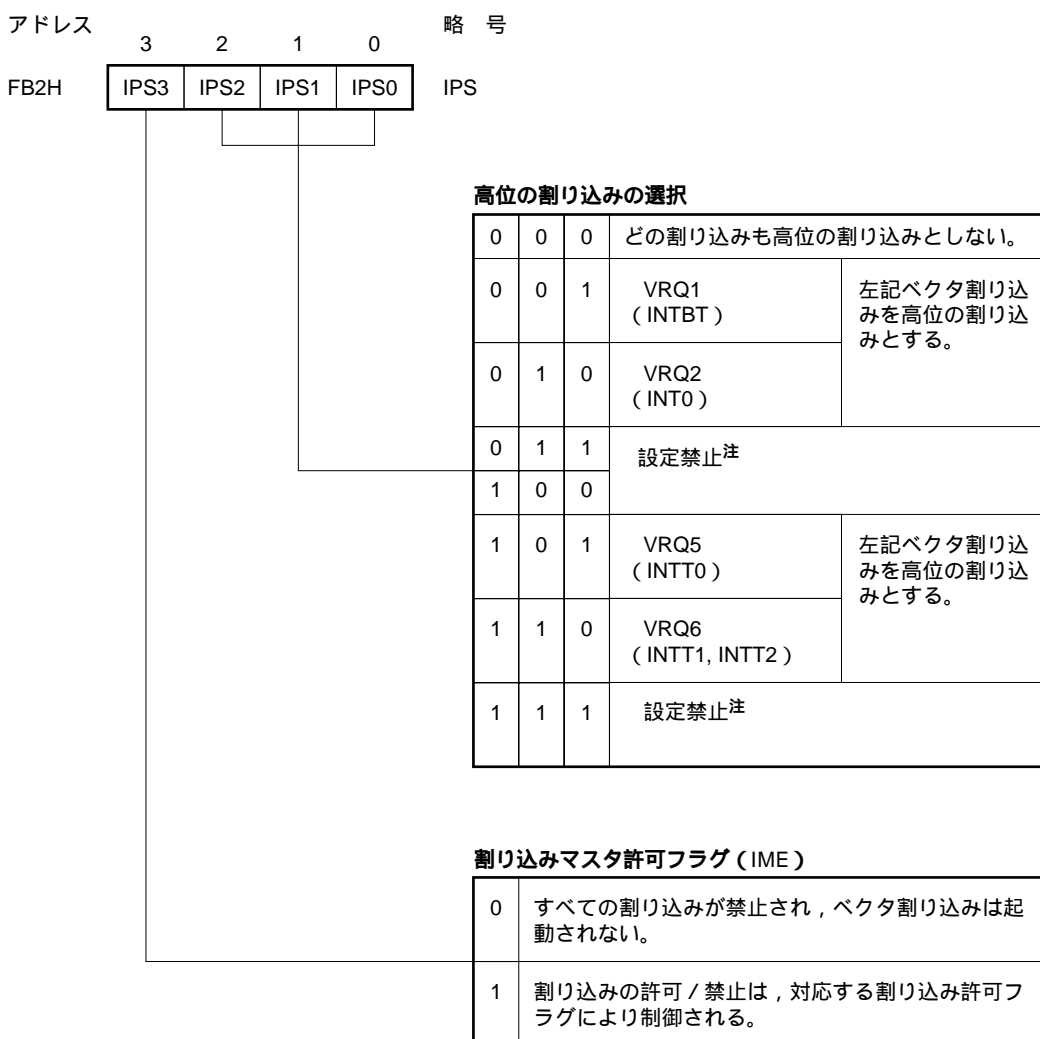
```

例  DI          ; 割り込み禁止
    CLR1  MBE
    MOV   A, #1001
    MOV   IPS, A          ; INTBTを高位の割り込みにし、割り込みを許可する

```

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

図6 - 3 割り込みプライオリティ選択レジスタ



注 IPSにこの値を設定すると、IPS = x000B (どの割り込みも高位の割り込みとしない) と設定した場合と同じになります。

## (3) INT0のハードウェア

(a) INT0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。

また、INT0は、サンプリング・クロックによるノイズ除去機能を持っています(図6-5 ノイズ除去回路の入出力タイミング参照)。ノイズ除去回路によって、サンプリング・クロックの2周期分<sup>注</sup>より幅の狭いパルスはノイズとして除去されます。ただし、サンプリング・クロックの1周期分以上のパルスは、サンプリングを行うタイミングによっては、割り込み信号として受け付けられる場合もあります(図6-5 (a)参照)。サンプリング・クロックの2周期分以上のパルスは確実に割り込み信号として受け付けることができます。

INT0は、サンプリング・クロックとして  $f_x/64$  の2つを持っており、どちらかを選択して使用できるようになっています。その選択はINT0エッジ検出モード・レジスタ(IM0)のビット3(IM03)で行います(図6-6参照)。

また、検出エッジの選択は、INT0エッジ検出モード・レジスタ(IM0)のビット0,1(IM00, IM01)で行います。

IM0のフォーマットを図6-6に示します。IM0は4ビット操作命令によって設定します。リセット信号の発生により全ビット“0”にクリアされ、立ち上がりエッジ指定となります。

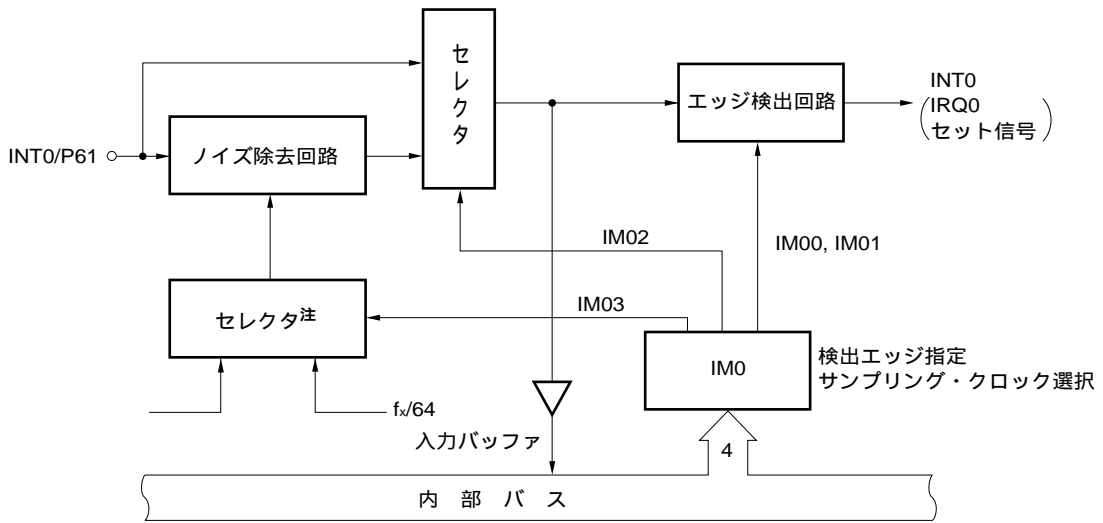
注 サンプリング・クロックが のとき :  $2t_{cy}$   
サンプリング・クロックが  $f_x/64$  のとき :  $128/f_x$

注意1. INT0/P61端子は、ポートとして入力する場合にも、ノイズ除去回路を通して入力されますので、サンプリング・クロックの2周期分以上のパルスを入力してください。

2. ノイズ除去回路を選択する(IM02 = 0に設定する)と、INT0はクロックによるサンプリングを行うため、スタンバイ・モード時は動作しません(ノイズ除去回路はCPUクロックΦが供給されていないときは動作しません)。したがって、スタンバイ・モード時にINT0によるスタンバイ・リリースを行う必要がある場合は、ノイズ除去回路を選択しないでください(IM02 = 1に設定してください)。

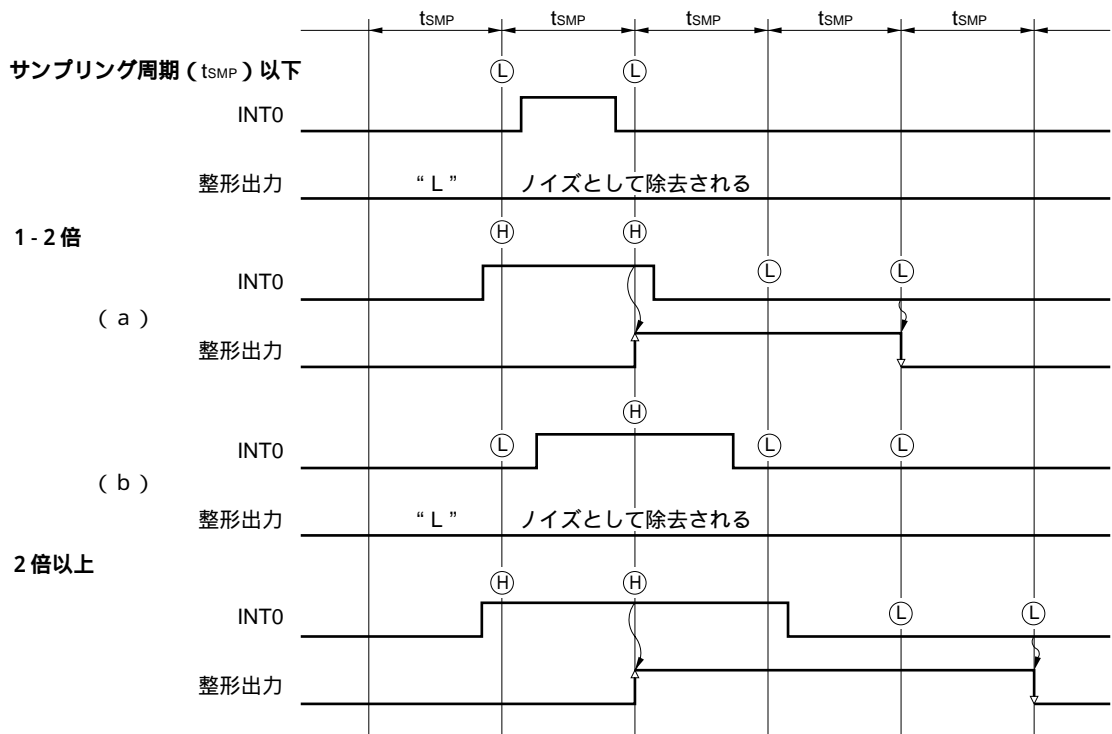
★

図6-4 INTOの構成



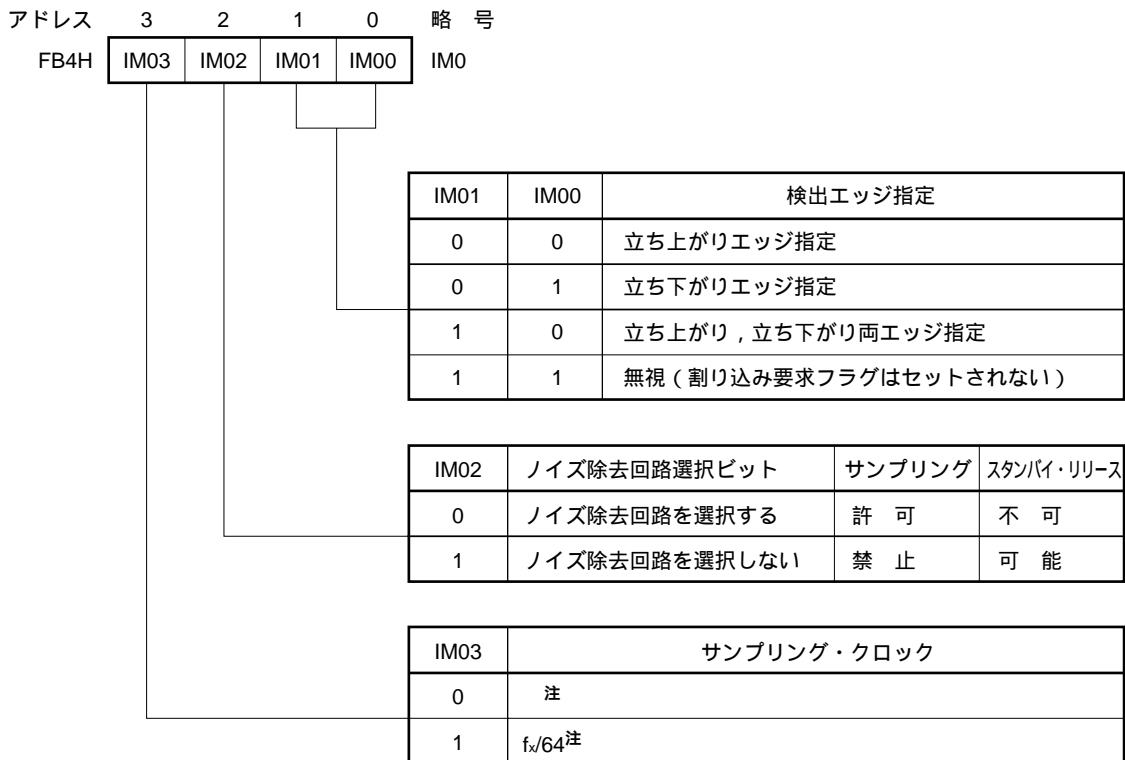
★ 注  $f_x/64$ を選択しても、INT0によるHALTモードの解除はできません。

図6-5 ノイズ除去回路の入出力タイミング



備考  $t_{SMP} = t_{CY}$ または $64/f_x$

図6-6 INT0エッジ検出モード・レジスタ (IM0) のフォーマット



注 この値は、システム・クロック周波数 ( $f_x$ ) の値により異なります。

**注意** エッジ検出モード・レジスタを変更すると、割り込み要求フラグがセットされる場合がありますので、あらかじめ割り込みを禁止してモード・レジスタを変更し、CLR1命令によって割り込み要求フラグをクリアしてから、割り込みを許可してください。なお、IM0の変更で、サンプリング・クロックとして $f_x/64$ を選択した場合には、モード・レジスタ変更後16マシン・サイクル経過してから、割り込み要求フラグをクリアする必要があります。

(4) 割り込みステータス・フラグ

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

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

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

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

IST1, IST0は、割り込み受け付け時に他のPSWとともにスタック・メモリに退避されたのち、自動的に1つ上位のステータスに変更されます。RETI命令が実行されると、もとのIST1, IST0の値が復帰します。

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

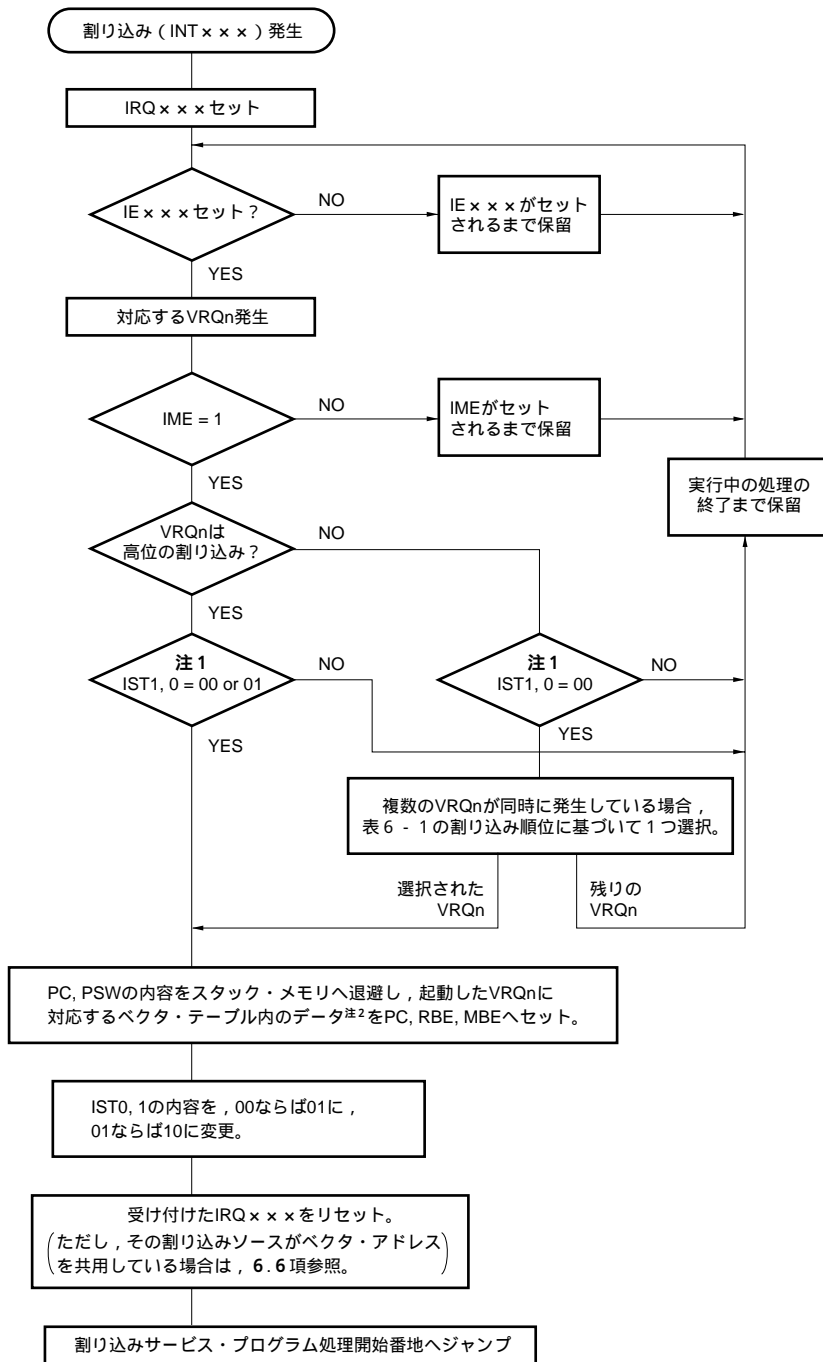
IST1	IST0	実行中の処理のステータス	CPUの処理内容	受け付け可能な割り込み要求	割り込み受け付け後	
					IST1	IST0
0	0	ステータス0	通常プログラム処理中	すべての割り込みを受け付け可能	0	1
0	1	ステータス1	低位の割り込み、または高位の割り込み処理中	高位の割り込みのみ受け付け可能	1	0
1	0	ステータス2	高位の割り込み処理中	すべての割り込みの受け付け禁止	-	-
1	1	設定禁止				



## 6.4 割り込みシーケンス

割り込みが発生すると、以下の手順で処理されます。

図6-7 割り込み処理シーケンス



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

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

## 6.5 多重割り込み処理制御

μPD754202は、次の2つの方法により多重割り込みができます。

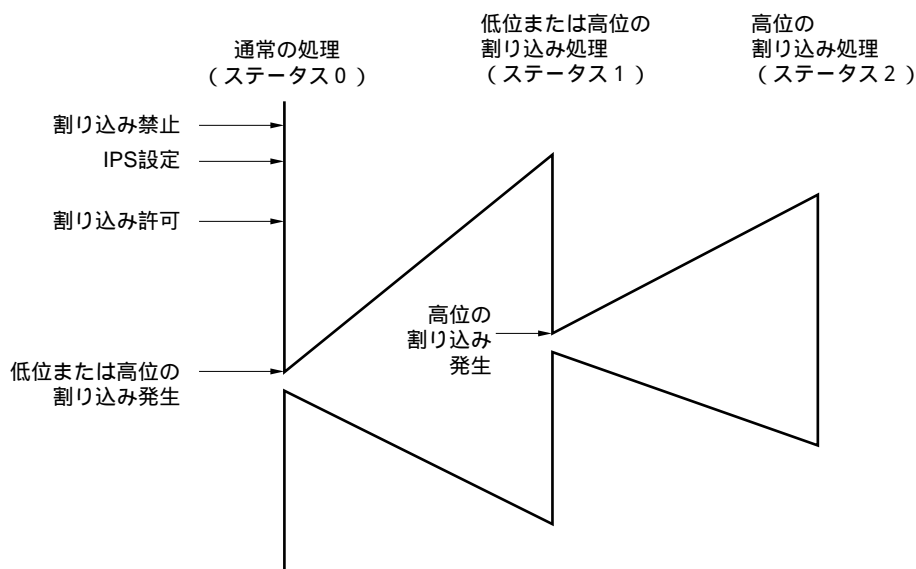
### (1) 高位の割り込みを指定する多重割り込み

この方法は、μPD754202の標準的な多重割り込み方法で、割り込みソースのうち1つを選び、その割り込みの多重割り込み（二重割り込み）を可能にするものです。

すなわち、割り込みプライオリティ選択レジスタ（IPS）によって指定された高位の割り込みは、実行中の処理のステータスが0および1のときに割り込み可能で、それ以外の割り込み（低位の割り込み）はステータス0のときのみ割り込み可能です（図6-8、表6-3参照）。

したがって、使用する割り込みのうち1つだけを多重割り込みできるようにしたい場合には、この方法を使えば、割り込み処理中での割り込み許可、禁止などの操作を行わなくても、すなわち割り込みステータス・フラグを変更しなくても、二重割り込みを実現し、かつ、ネスティング・レベルを2レベルに抑えることができます。

図6-8 高位の割り込みによる多重割り込み



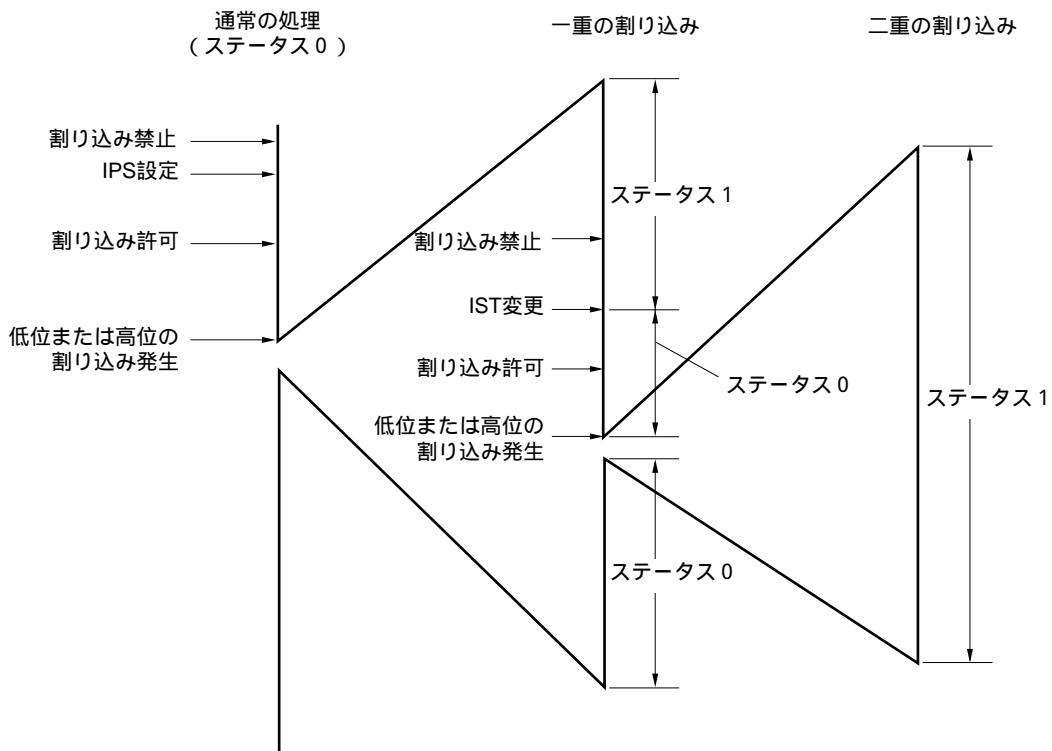
(2) 割り込みステータス・フラグを変更する多重割り込み

割り込みステータス・フラグをプログラムにより変更すれば、多重割り込み可能になります。すなわち、割り込み処理プログラムでIST1, IST0を“0, 0”に変更し、ステータス0にすれば多重割り込みできます。

この方法は2つ以上の割り込みの多重割り込みを許可したい場合、あるいは三重以上の多重割り込みを行いたい場合に使用します。

IST1, IST0の変更は、あらかじめDI命令により割り込みを禁止した状態で行います。

図6-9 割り込みステータス・フラグの変更による多重割り込み



## 6.6 ベクタ・アドレス共用割り込み処理

INTT1とINTT2の各割り込みソースはベクタ・テーブルを共用しているため、以下に示すように割り込みソースの選択を行います。

### (1) 一方の割り込みのみ使用する場合

ベクタ・テーブルを共用する2種類の割り込みソースのうち、必要とする割り込みソースの割り込み許可フラグを“1”にセット、もう一方の割り込み許可フラグを“0”にクリアします。この場合、許可した( $IE \times x = 1$ )方の割り込みソースにより割り込み要求を発生し、受け付けられるとその割り込み要求フラグはリセットされます。

### (2) 両方の割り込みを使用する場合

2種類の割り込みソースに対応する割り込み許可フラグを共に“1”にセットします。この場合、2種類の割り込みソースの割り込み要求フラグの論理和が割り込み要求となります。

この場合、一方あるいは両方の割り込み要求フラグのセットによる割り込み要求が受け付けられてもどちらの割り込み要求フラグもリセットされません。これは、割り込み処理内でどちらの割り込みが発生したかを確認するためです。

したがって、この場合は割り込みサービス・ルーチンでどちらの割り込みソースによる割り込みかを判断する必要があります。それは、割り込みサービス・ルーチンの最初にSKTCLR命令で割り込み要求フラグをチェックすることにより行います。

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

すなわち、テストしない方の割り込み要求が先に処理されることになります。一方、「低位の割り込み」の場合には、残った割り込みは保留されますから、テストした方の割り込み要求が先に処理されます。したがって、表6-4のように、「高位の割り込み」であるか否かによって共用割り込み判別方法は異なります。

表6-4 共用割り込みの判別

「高位の割り込み」の場合	割り込みを禁止し、優先したい割り込み要求フラグをテストする。
「低位の割り込み」の場合	優先したい割り込みソースの割り込み要求フラグをテストする。

例1 . INTT1とINTT2の両方を「高位の割り込み」として使用し、INTT2を優先する場合。

```

DI
SKTCLR    IRQT2    ;  IRQT2 = 1 ?
BR        VSUBBT
           ⋮
EI
RETI
           ⋮

VSUBBT :  CLR1     IRQT1
           ⋮
           ⋮
EI
RETI
    
```

} INTT2の処理  
ルーチン

} INTT1の処理  
ルーチン

2 . INTT1とINTT2の両方を「低位の割り込み」として使用しINTT2を優先する場合。

```

SKTCLR    IRQT2    ;  IRQT2 = 1 ?
BR        VSUBBT
           ⋮
           ⋮
           ⋮
RETI
           ⋮

VSUBBT :  CLR1     IRQT1
           ⋮
           ⋮
           ⋮
RETI
    
```

} INTT2の処理  
ルーチン

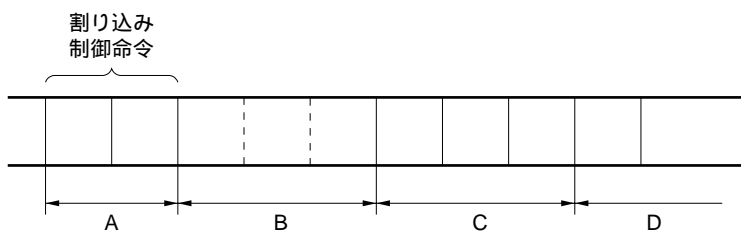
} INTT1の処理  
ルーチン

## 6.7 割り込み処理までのマシン・サイクル

μPD754202では、割り込み要求フラグ（IRQ×××）がセットされてから割り込みルーチンのプログラムを実行するまでのマシン・サイクルは、次のようになります。

### （1）割り込み制御命令実行中にIRQ×××がセットされた場合

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



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

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

**2** . IRQ×××がセットされるタイミング（上図のA）または、そのあとのタイミングでDI命令を実行すると、セットされたIRQ×××の割り込み要求は次にEI命令を実行するまで保留されます。

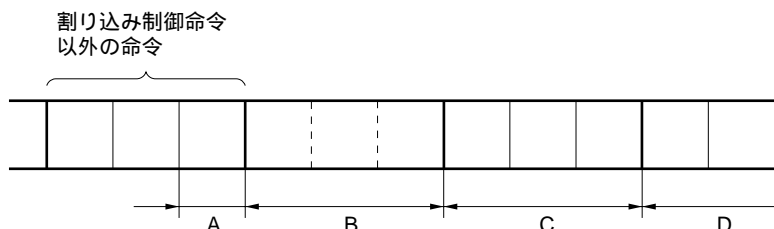
**備考 1** . 割り込み制御命令とは、割り込み関係のハードウェア（データ・メモリのFB×H番地）を操作する命令です。この命令には、DI命令、EI命令があります。

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

(2)(1) 以外の命令実行中にIRQ×××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。

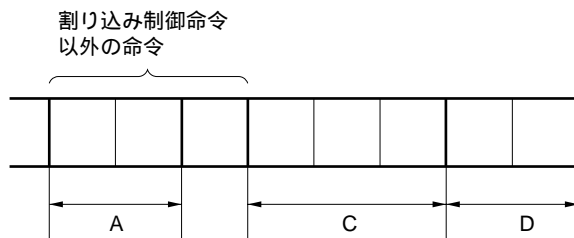


- A : IRQ×××セット
- B : 続く1命令を実行 (1-3マシン・サイクル; 命令により異なる)
- C : 割り込み処理 (3マシン・サイクル)
- D : 割り込みルーチンの実行

**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

## 6.8 割り込みの効果的な使い方

割り込み機能は、次のように使用すると効果的です。

### (1) 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。

## 6.9 割り込みの応用

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

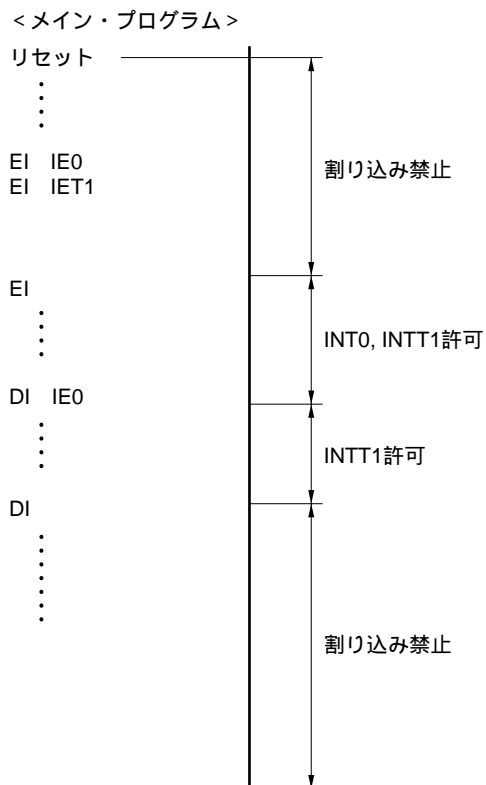
- (a) 使用する割り込み許可フラグをセット (EI IE × × × 命令)
- (b) INT0を使用する場合は、アクティブ・エッジの選択 (IM0セット)。
- (c) 二重割り込み (高位の割り込みによる) を使用する場合は、IPSを設定 (同時にIMEセットもできる)。
- (d) 割り込みマスタ許可フラグをセット (EI命令)。

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

割り込み処理プログラムからの復帰はRETI命令を使用します。



(1) 割り込みの許可・禁止



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

EI IE x x x 命令により，割り込み許可フラグをセット。

この段階ではまだすべての割り込みは禁止

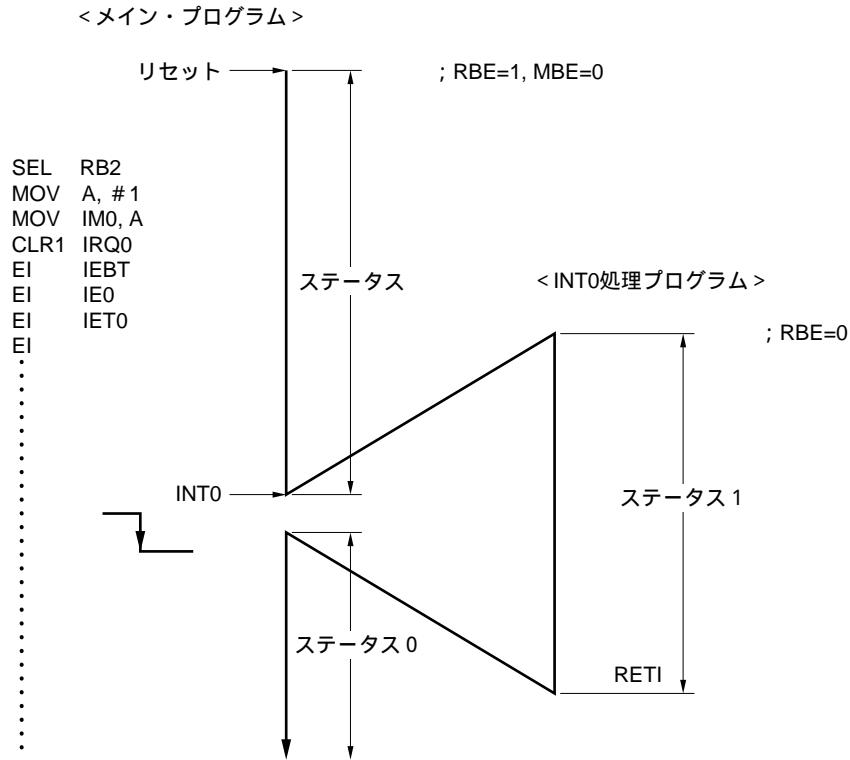
EI命令により，割り込みマスタ許可フラグをセット。

この段階で，INT0, INTT1が許可される。

DI IE x x x 命令により，割り込み許可フラグをクリア，INT0が禁止される。

DI命令によりすべての割り込みが禁止。

( 2 ) INTBT, INT0 ( 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 )



$\overline{\text{RESET}}$ 信号により、すべての割り込みが禁止され、ステータス 0 になる。

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。

INT0 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

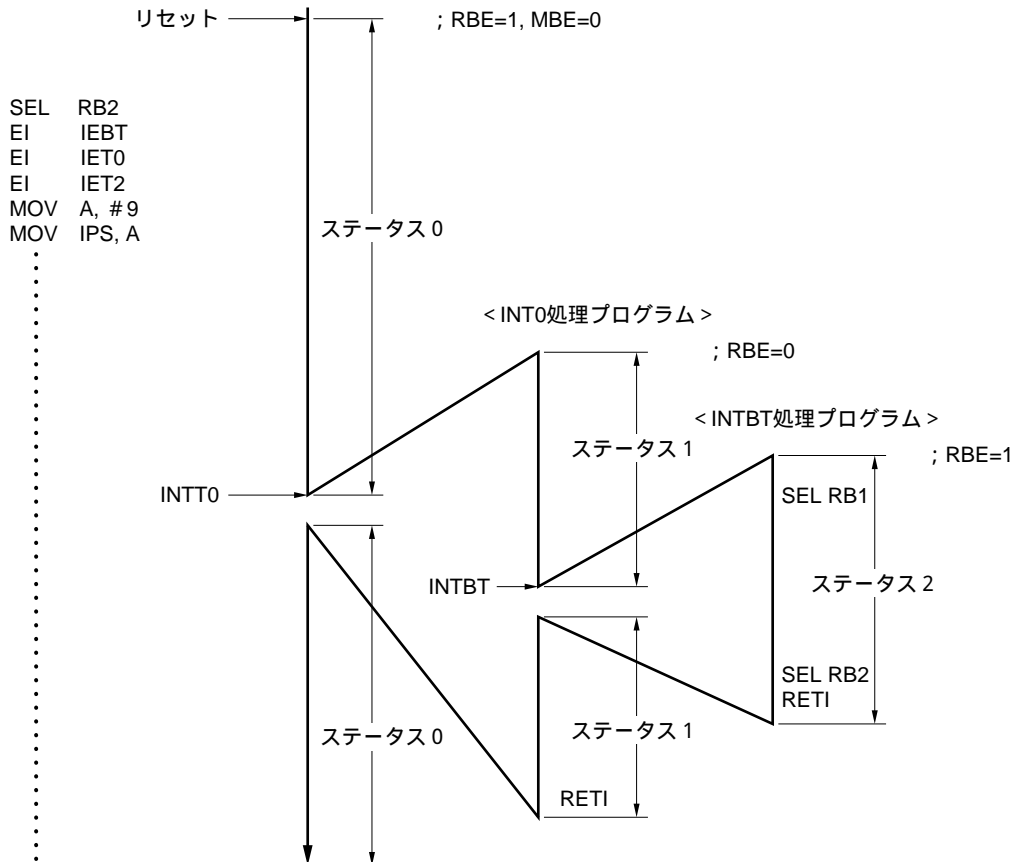
INT0 の立ち下がりにより、INT0 割り込み処理プログラムをスタート。ステータスは 1 に変更され、すべての割り込みは禁止される。

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, INTT2を低位の割り込み)



IPSの設定により, INTBTを「高位の割り込み」とし, 同時に割り込み許可。

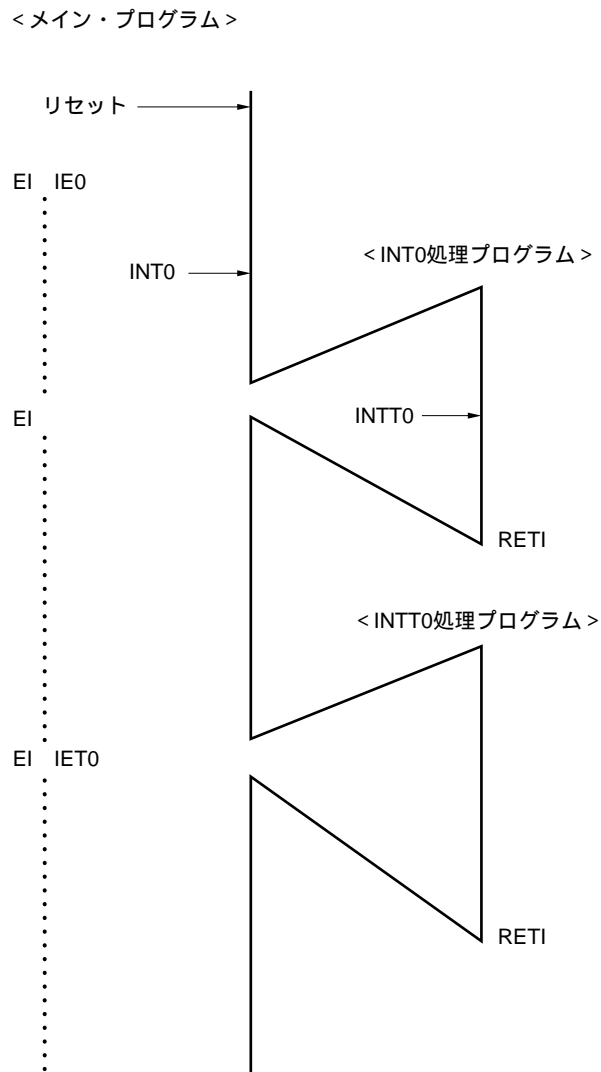
低位の割り込みINTT0発生により, INTT0処理プログラム開始, ステータス 1 となり低位の割り込みは禁止。RBE = 0としてレジスタ・バンク 0 を使用。

高位の割り込みINTBT発生により, 二重割り込み実行。ステータスとなりすべての割り込み禁止。

RBE = 1, RBS = 1としてレジスタ・バンク 1 を使用 (使用するレジスタだけをPUSH命令で退避してもよい)。

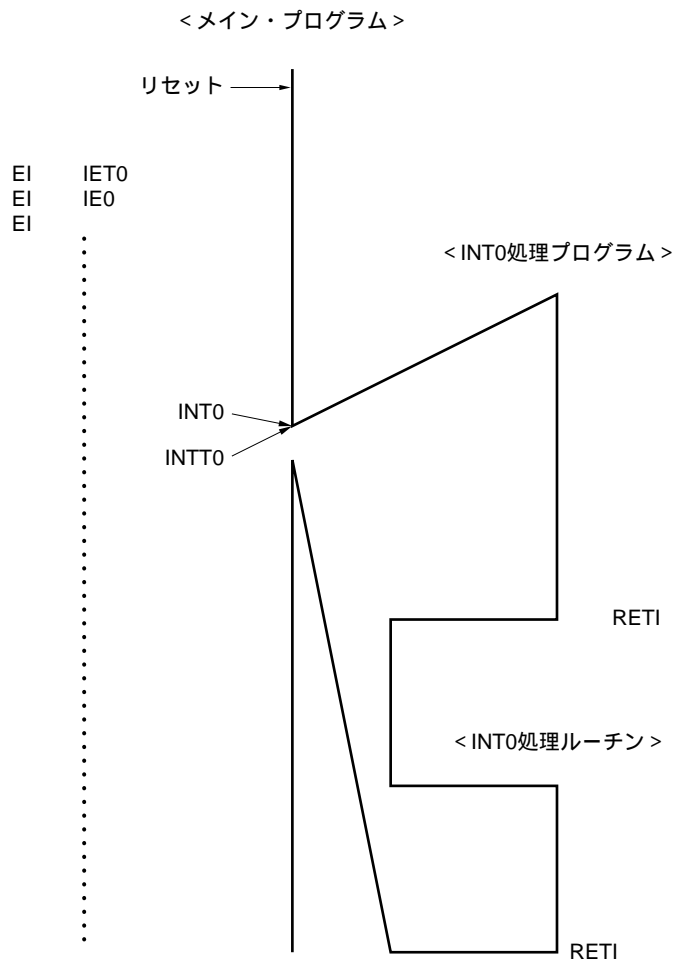
RBSを 2 に戻し, リターン。ステータスは 1 に戻る。

(4) 保留割り込みの実行 割り込み禁止中の割り込み入力



割り込み禁止中にINT0がセットされても要求フラグは保留される。  
 EI命令で割り込みが許可された時点で、INT0処理プログラム・スタート。  
 と同様。  
 保留されていたINTT0が許可された時点で、INTT0処理プログラム・スタート。

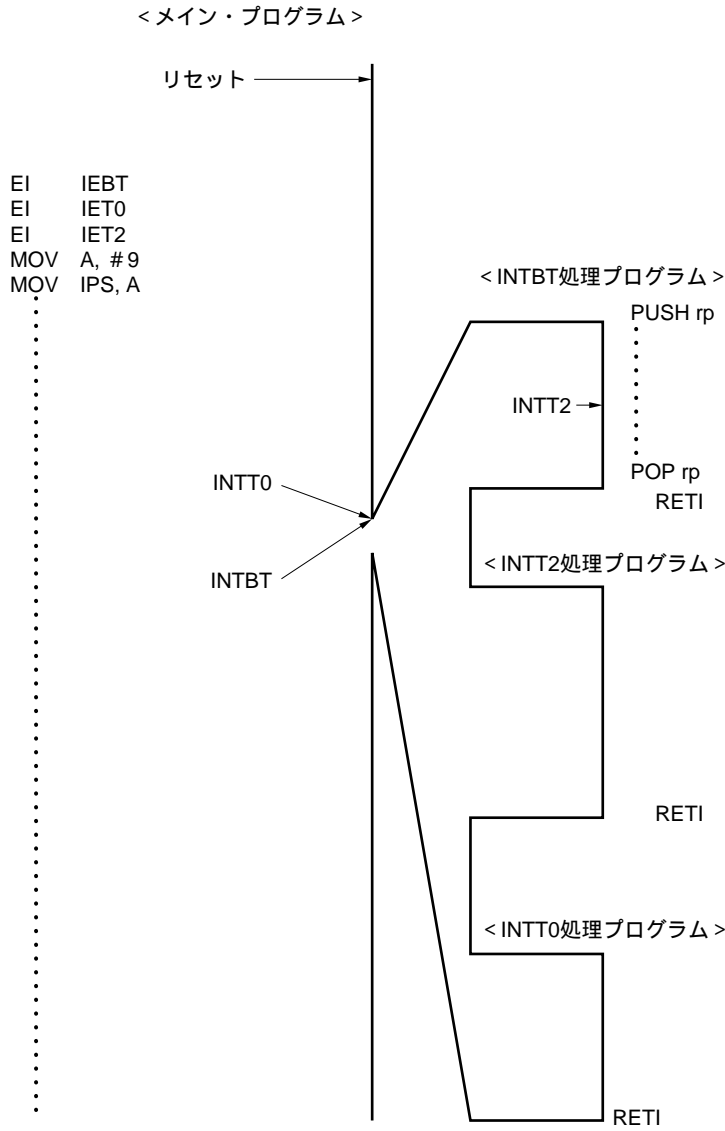
(5) 保留割り込みの実行 低位の割り込みが同時に2つ発生



「低位の割り込み」INT0とINTT0が同時（同一命令実行中）に発生した場合は、割り込み順位の高いINT0を先に実行（INTT0は保留される）。

RETI命令により、INT0処理プログラムが終わると保留されていたINTT0処理プログラムをスタート。

(6) 保留割り込みの実行 割り込み処理中の割り込み発生 (INTBTを高位, INTT0, INTT2を低位の割り込み)



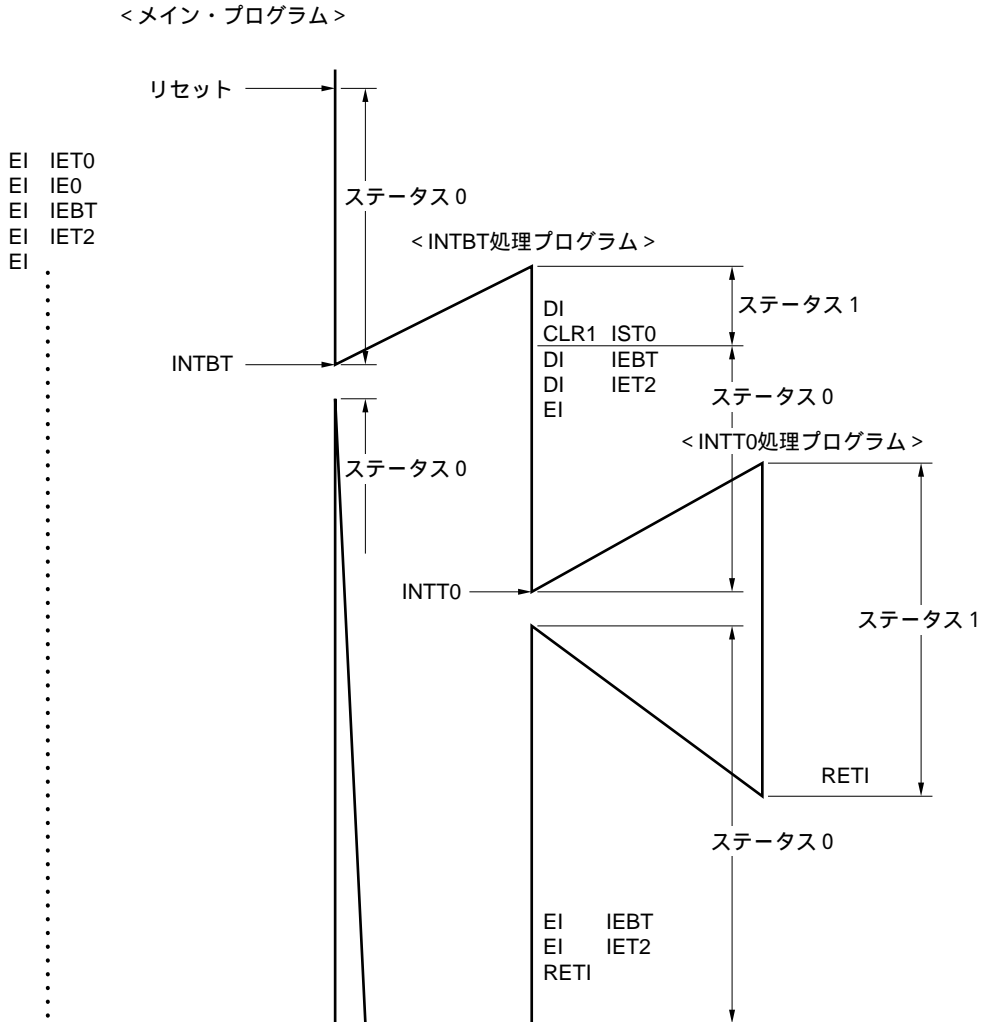
高位の割り込みINTBTと低位の割り込みINTT0が同時に発生すると、高位の割り込み処理を開始 (高位の割り込み処理中に、高位の割り込みが発生するおそれがないことが確実ならDI IE × ×は不要)。

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INTT2処理が実行される。

INTT2処理が終了すると保留されていたINTT0が処理される。

(7) 二重割り込みを2つ許可する。 INTT0とINT0は二重割り込みを許可。 INTBT, INTT2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とINTT2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, INTT2を許可して復帰。

## 6.10 テスト機能

### 6.10.1 テスト・ソースの種類

μPD754202には、テスト・ソースINT2があります。INT2は、エッジ検出テストブル入力です。

表6-5 テスト・ソースの種類

テスト・ソース	内/外
INT2 (KR4-KR7端子への入力のいずれかの立ち下がりエッジ検出)	外

### 6.10.2 テスト機能を制御する各種ハードウェア

#### (1) テスト要求フラグ, テスト許可フラグ

テスト要求フラグ (IRQ2) は、テスト要求発生 (INT2) 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 (IE2) は、テスト要求フラグに対応しており、内容が “ 1 ” のときスタンバイ・リリース信号を許可し、“ 0 ” のとき禁止します。

テスト要求フラグ、テスト許可フラグがともに “ 1 ” にセットされている場合、スタンバイ・リリース信号を発生します。

表6-6 にテスト要求フラグのセット信号を示します。

表6-6 テスト要求フラグのセット信号

テスト要求フラグ	テスト要求フラグのセット信号	テスト許可フラグ
IRQ2	KR4/P70-KR7/P73端子への入力のいずれかの立ち下がりエッジ検出によりセット。検出エッジはINT2エッジ検出モード・レジスタ (IM2) により選択。	IE2

#### (2) キー割り込み (KR4-KR7) のハードウェア

KR4-KR7の構成を図6-10に示します。

キー割り込みでのエッジ検出により、IRQ2セット信号が出力されます。いずれの端子の立ち下がり入力を選択するかは、INT2エッジ検出モード・レジスタ (IM2) で行います。

IM2のフォーマットを図6-11に示します。IM2は4ビット操作命令によって設定します。リセット信号の発生により全ビット “ 0 ” にクリアされます。



図6-10 KR4-KR7のブロック図

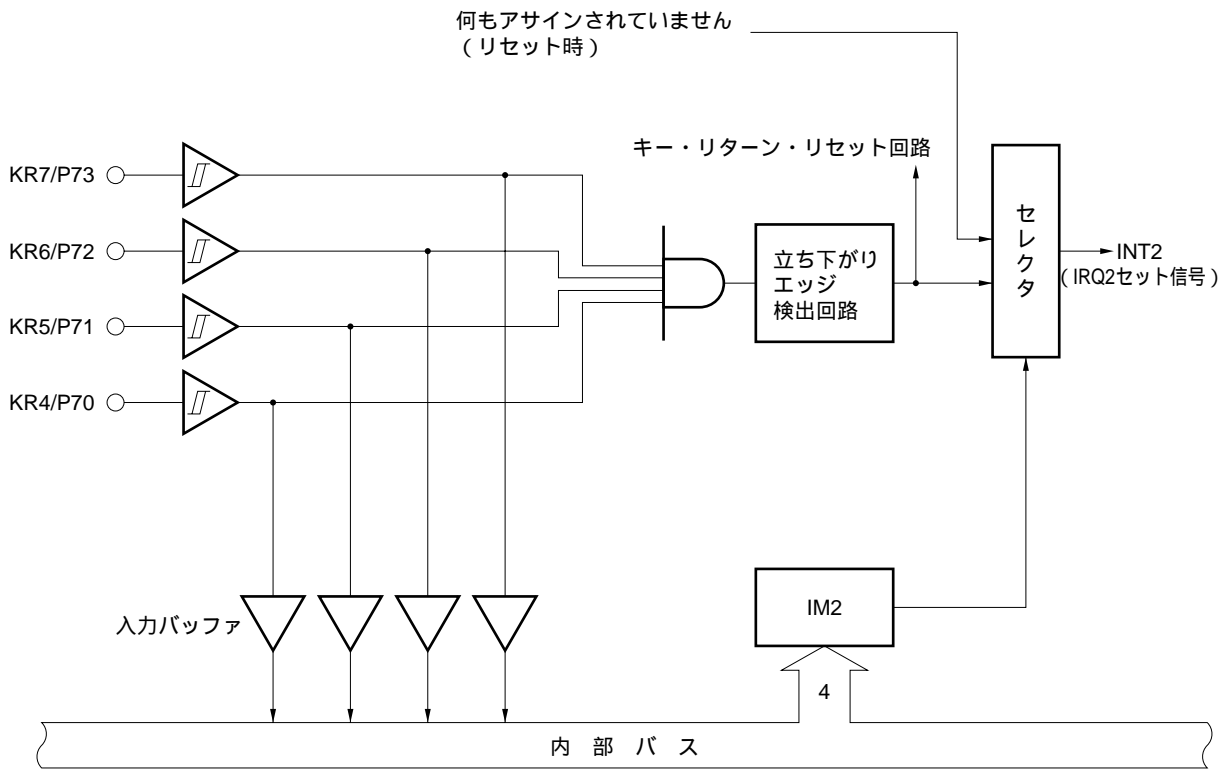
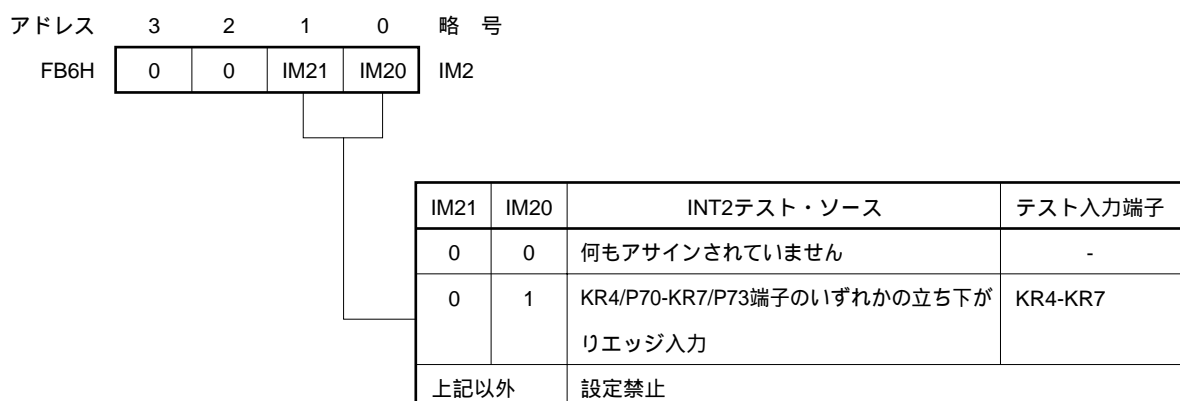


図6-11 INT2エッジ検出モード・レジスタ(IM2)のフォーマット



- 注意1. エッジ検出モード・レジスタを変更すると、テスト要求フラグがセットされる場合がありますので、あらかじめテスト入力を禁止してモード・レジスタを変更し、CLR1命令によってテスト要求フラグをクリアしてから、テスト入力を許可してください。
2. KR×端子のうち1本でもロウ・レベルが入力されていると、ほかの端子に立ち下がりエッジが入力されても、IRQ2はセットされません。
  3. リセット時はIM2は全ビット0となっています。このため、INT2のテスト・ソースとしては何もアサインされていません。KR4-KR7端子のいずれかの立ち下がりエッジ入力による割り込みを使用する場合は、必ずIM2を0001Bに設定してください。

(3) KRREN端子の機能

STOPモード時で、KRREN端子がハイ・レベルのとき、KR4-KR7端子のいずれかの立ち下がりエッジ入力が発生すると、システム・リセットがかかります。

表6-7 KR4-KR7端子とKRREN端子およびテスト機能

KR4-KR7端子	動作モード	KRREN端子	テスト機能
立ち下がりエッジ 信号発生	通常動作および	ロウ・レベル	IRQ2をセット
	HALTモード	ハイ・レベル	
	STOPモード	ロウ・レベル	
		ハイ・レベル	システム・リセット信号発生のため無効

またSTOPモードは、割り込み許可フラグを変化させることなく、KRREN端子がハイ・レベルのとき、KRn (n=4-7) 端子の立ち下がりエッジ入力(キー・リターン・リセット)によって解除することができます。

〔メ モ〕

## 第7章 スタンバイ機能

μPD754202は、システムの消費電力を低減させることのできるスタンバイ機能を持っています。スタンバイ機能には次のような2つのモードがあります。

STOPモード

HALTモード

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

### (1) STOPモード

システム・クロック発振回路を停止させ、システム全体が停止するモードです。CPUの消費電流は、かなり低減されます。

また、データ・メモリの低電圧 ( $V_{DD} = 1.8V$ 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。

μPD754202のSTOPモードは、割り込み要求によって解除できるため、間欠動作も可能です。しかし、STOPモードの解除時に発振安定時間確保のためのウェイト時間がとられるため、割り込み要求によって、すぐ処理を開始しなければならない場合にはHALTモードを選択してください。

### (2) HALTモード

CPUの動作クロックを停止させるモードです。システム・クロック発振回路の発振は続きます。このモードでは、STOPモードほどの消費電流の低減はできませんが、割り込み要求により、すぐに処理を再開したい場合や、時計動作のような間欠動作をさせたい場合に有効です。

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

使用上の注意を次ページに示します。

- 注意 1 . スタンバイ・モードとCPU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 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。
- 2 . スタンバイ・モードを使用する場合には、入出力ポートは消費電流が最小となるよう処理してください。
- 特に、入力ポートはオープンにしないでください。必ずロウ・レベルかハイ・レベルを入力してください。

## 7.1 スタンバイ・モードの設定および動作状態

表7-1 スタンバイ・モード時の各動作状態

		STOPモード	HALTモード
設定命令		STOP命令	HALT命令
★ 動作状態	クロック発生回路	動作停止	CPUクロックのみ停止（発振継続）
	ベーシック・インターバル・タイム/ウォッチドッグ・タイム	動作停止	動作可能 （BTモード：基準時間間隔でIRQBTをセット） （WTモード：BTのオーバーフローによりリセット信号発生）
	タイマ・カウンタ	動作停止	動作可能
	外部割り込み	INT0は動作不可能 <sup>注</sup> INT2は、KRnの立ち下がり時のみ動作可能	
★	CPU	動作停止	
解除信号		<ul style="list-style-type: none"> <li>・リセット信号</li> <li>・割り込みが許可されているハードウェアからの割り込み要求信号</li> <li>・KRREN端子が1のとき、KRnの立ち下がりによって発生するシステム・リセット信号（キー・リターン・リセット）</li> </ul>	<ul style="list-style-type: none"> <li>・リセット信号</li> <li>・割り込みが許可されているハードウェアからの割り込み要求信号</li> </ul>

注 エッジ検出モード・レジスタ（IM0）のビット2により、ノイズ除去回路を選択しない場合（IM02 = 1のとき）のみ動作可能。

STOPモードにはSTOP命令で、HALTモードにはHALT命令で設定します（STOP命令、HALT命令は、それぞれPCCのビット3、ビット2をセットする命令です）。

STOP命令、HALT命令の次には必ずNOP命令を書くようにしてください。

PCCの下位2ビットによってCPU動作クロックを変更する場合、表5-5 CPUクロックの切り替えに要する最大時間に示すようにPCCを書き換えてから、CPUクロックが変更されるまでには、時間的なずれを生じる場合があります。したがって、スタンバイ・モード前の動作クロックとスタンバイ・モード解除後のCPUクロックを変更するような場合には、PCCを書き換えてからCPUクロックの変更に必要なマシン・サイクル経過後にスタンバイ・モードを設定します。

スタンバイ・モードでは、汎用レジスタ、フラグ類、モード・レジスタ類、出力ラッチなど、スタンバイ・モード中動作を停止しているすべてのレジスタとデータ・メモリのデータが保持されます。

- ★ **注意1** . STOPモードに設定すると、X2端子が内部で $V_{DD}$ に50 k (typ.)の抵抗でプルアップされます。
- 2** . スタンバイ・モードを設定する前に、あらかじめすべての割り込み要求フラグをリセットしてください。
- 割り込み要求フラグと割り込み許可フラグの両方がセットされている割り込みソースがある場合には、スタンバイ・モードに入ってもただちに解除されます（図6-1 割り込み制御回路ブロック図を参照）。
- ただし、STOPモードを設定した場合は、STOP命令実行後すぐHALTモードに入り、BTMレジスタによる設定時間だけウエイトしたあと動作モードに戻ります。

## 7.2 スタンバイ・モードの解除

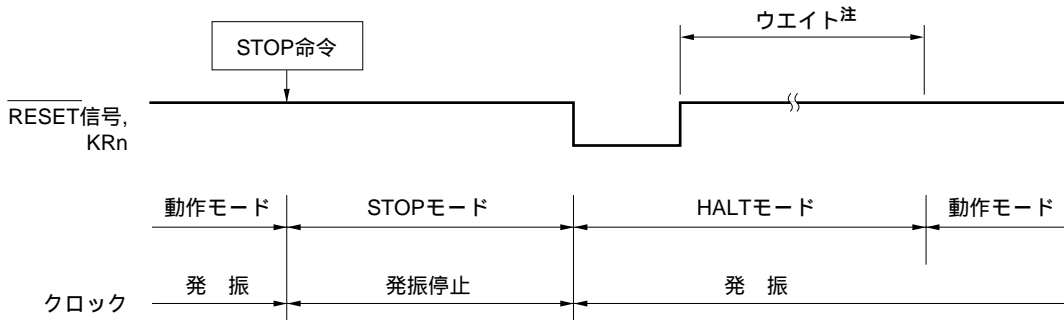
STOPモード，HALTモードいずれも割り込み許可フラグで許可されている割り込み要求信号の発生，および $\overline{\text{RESET}}$ 信号発生により解除されます。

またSTOPモードは，割り込み許可フラグを変化させることなく，KRREN端子がハイ・レベルのとき，KRn端子の立ち下がりエッジ入力（キー・リターン・リセット）によって解除することができます。

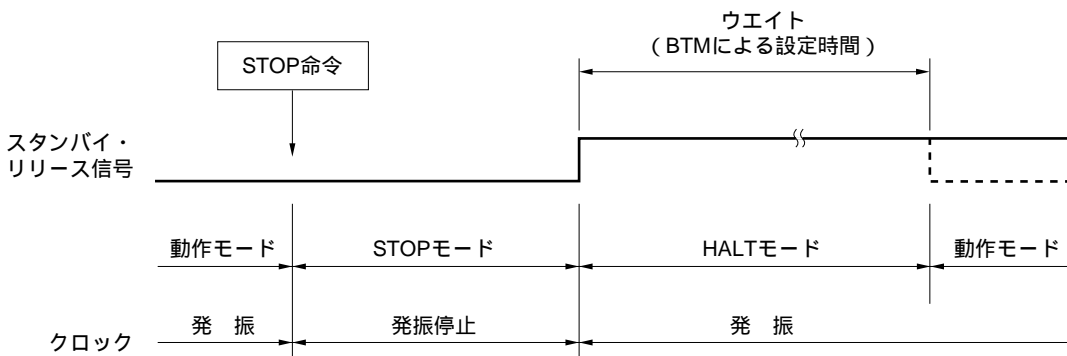
各モードの解除動作は図7 - 1のようになります。

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，またはキー・リターン・リセットによる解除



(b) STOPモードの割り込み発生による解除（ただし，キー・リターン・リセットによる解除を除く）



注 マスク・オプションにより次の2つの時間を選択することができます。

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

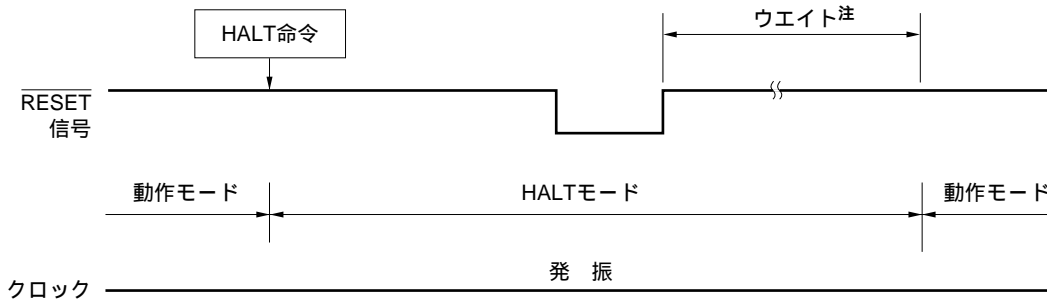
$2^{15}/f_x$  (5.46 ms : 6.0 MHz動作時，7.81 ms : 4.19 MHz動作時)

備考 破線は，スタンバイを解除した割り込み要求が受け付けられた場合です。

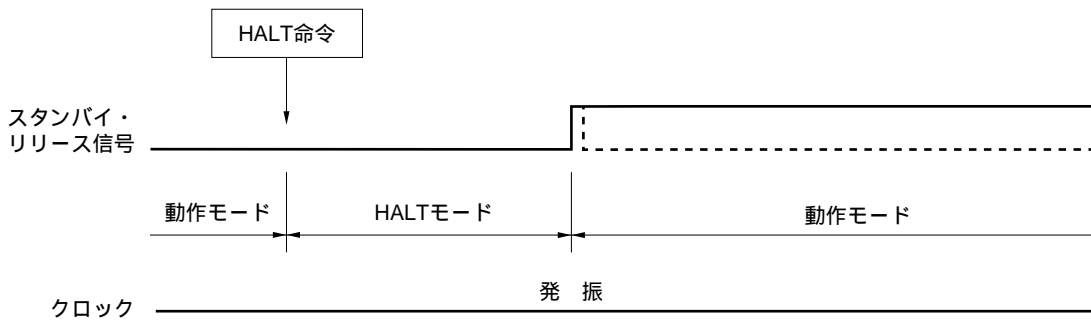


図7-1 スタンバイ・モードの解除動作(2/2)

(c) HALTモードのRESET信号発生による解除



(d) HALTモードの割り込み発生による解除



注 マスク・オプションにより次の2つの時間を選択することができます。

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

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

備考 破線は、スタンバイを解除した割り込み要求が受け付けられた場合です。

STOPモードの解除に使用する割り込みは、該当する割り込み許可フラグ(IE)をセット(1)することにより選択します。STOPモード中に選択された割り込み要求フラグ(IRQ)がセット(1)されると、STOPモードが解除されます。

このとき、該当する割り込みのIEをセット(1)する前に、必ず、すべてのIEとIRQをクリア(0)しておきます。これは、選択以外の割り込みによってSTOPモードが解除されないためです。

次に、割り込み発生によりSTOPモードを解除する手順を示します。

すべてのIEとIRQをクリアします。

STOPモードの解除に使用する割り込みのIEをセット。

あらためてSTOPモードの解除に使用するIRQをクリアし、STOPモードに入ります。

このSTOPモード中に、選択した割り込みのIRQがセットされ、HALTモードに入ります。このあと、ウェイト時間をおいてから、通常動作モードに戻ります。

STOPモードを割り込み発生により解除した場合、ウェイト時間はBTMの設定により決まります（表7 - 2参照）。

発振が安定するまでの時間は、使用する発振子の種類と、STOPモード解除時の電源電圧によって変わってきます。したがって使用状況に応じてウェイト時間を選択し、STOPモードを設定する前に、BTMをセットします。

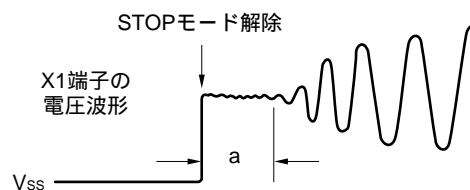
表7 - 2 BTMによるウェイト時間の選択

BTM3	BTM2	BTM1	BTM0	ウェイト時間 <sup>注</sup>	
				$f_x = 6.0 \text{ MHz}$ 動作時	$f_x = 4.19 \text{ MHz}$ 動作時
-	0	0	0	約 $2^{20}/f_x$ (約175 ms)	約 $2^{20}/f_x$ (約250 ms)
-	0	1	1	約 $2^{17}/f_x$ (約21.8 ms)	約 $2^{17}/f_x$ (約31.3 ms)
-	1	0	1	約 $2^{15}/f_x$ (約5.46 ms)	約 $2^{15}/f_x$ (約7.81 ms)
-	1	1	1	約 $2^{13}/f_x$ (約1.37 ms)	約 $2^{13}/f_x$ (約1.95 ms)
上記以外				設定禁止	

注 この時間は、STOPモード解除後、発振を開始するまでの時間は含みません。

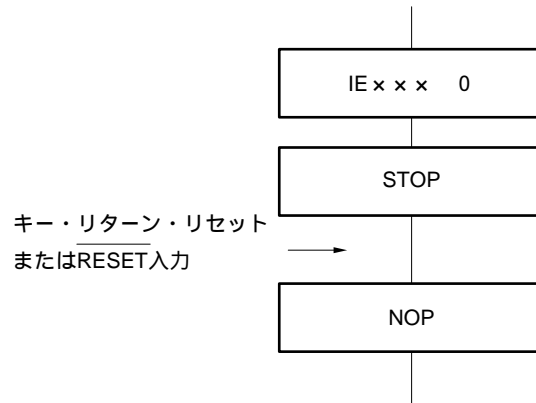
注意 STOPモード解除時のウェイト時間は、 $\overline{\text{RESET}}$ 信号発生による場合も、割り込み発生による場合もSTOPモード解除後クロック発振を開始するまでの時間（図7 - 2のa）は含みません。

図7 - 2 STOPモード解除時のウェイト時間



また、割り込み入力ではなく、キー・リターン・リセットまたは $\overline{\text{RESET}}$ 入力によってSTOPモードを解除する前には、必ず、すべての割り込み許可フラグ（IE2を含む）を図7 - 3に示すようにクリアしてください。

図7-3 キー・リターン・リセットまたは $\overline{\text{RESET}}$ 入力によるSTOPモードの解除



キー・リターン・リセットによる解除と $\overline{\text{RESET}}$ 入力による解除の違いは次のとおりです。

	$\overline{\text{RESET}}$ 入力	キー・リターン・リセット
キー・リターン・フラグ (KRF)	0	1
ウォッチドッグ・フラグ (WDF)	0	保持

## 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、INT2 (テストブル入力) によって解除された場合はベクタ割り込みは発生しませんから、(a)と同様の処理を行います。

## 7.4 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。

この項で紹介する応用例は、すべて $f_x = 4.19$  MHz動作時の場合です。 $f_x = 6.0$  MHz動作時では、同じ設定でもCPUクロックやウェイト時間が異なります。

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・セットまでに必要なマシン・サイクル経過するようにする)。

解除時のウェイト時間の選択。

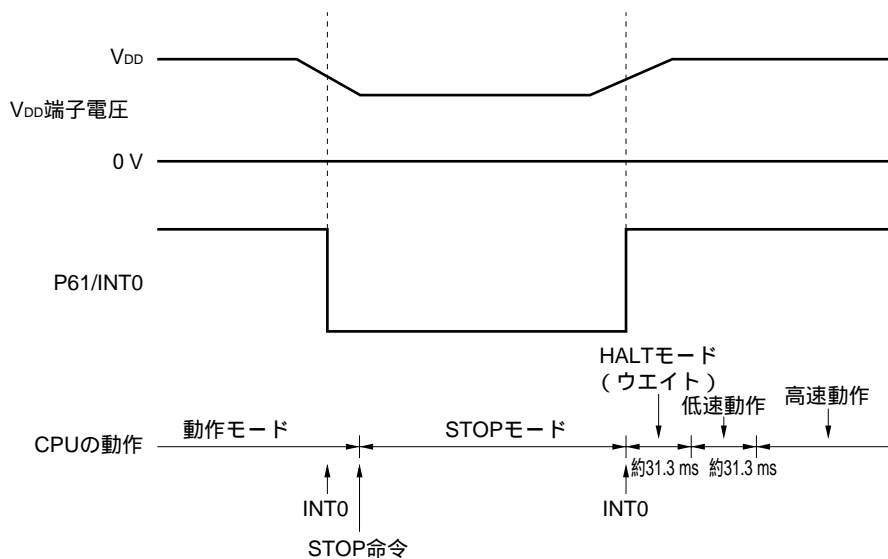
スタンバイ・モード設定 (STOP, HALT命令)。

(1) STOPモードの応用例 (fx = 4.19 MHz動作時)

次の条件でSTOPモードを使用する場合

INT0の立ち下がリエッジ入力によりSTOPモードを設定し、立ち上がりエッジ入力で解除する。  
 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約31.3 msとする。  
 解除後、電源の安定のためにさらに31.3 msウェイトする。またP61/INT0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

(INT0処理プログラム, MBE = 0)

```

VSUB0 : SKT      PORT6.1    ; P61 = 1?
        BR       PDOWN     ; パワーダウン
        SET1     BTM.3      ; パワーオン
WAIT :   SKT      IRQBT     ; 31.3 msのウエイト
        BR       WAIT
        SKT      PORT6.1    ; チャタリングをチェック
        BR       PDOWN
        MOV      A, #0011B
        MOV      PCC, A     ; 高速モード設定
        [ MOV     XA, #x xH ] ; ポート・モード・レジスタをセット
        [ MOV     PMGm, XA ]
        EI       IEBT
        EI       IET0
        RETI
PDOWN : MOV      A, #0       ; 最低速モード
        MOV      PCC, A
        MOV      XA, #00H
        MOV      PMGA, XA   ; 入出力ポート・ハイ・インピーダンス
        DI       IEBT      ; INTBT, INTT0禁止
        DI       IET0
        MOV      A, #1011B
        MOV      BTM, A     ; ウエイト時間 31.3 ms
        NOP
        STOP                      ; STOPモードをセット
        NOP
        RETI

```

(2) HALTモードの応用例 (fx = 4.19 MHz動作時)

次の条件で間欠動作する場合

INT0の立ち下がりで、スタンバイ・モードを設定し、立ち上がりエッジ入力で解除する。

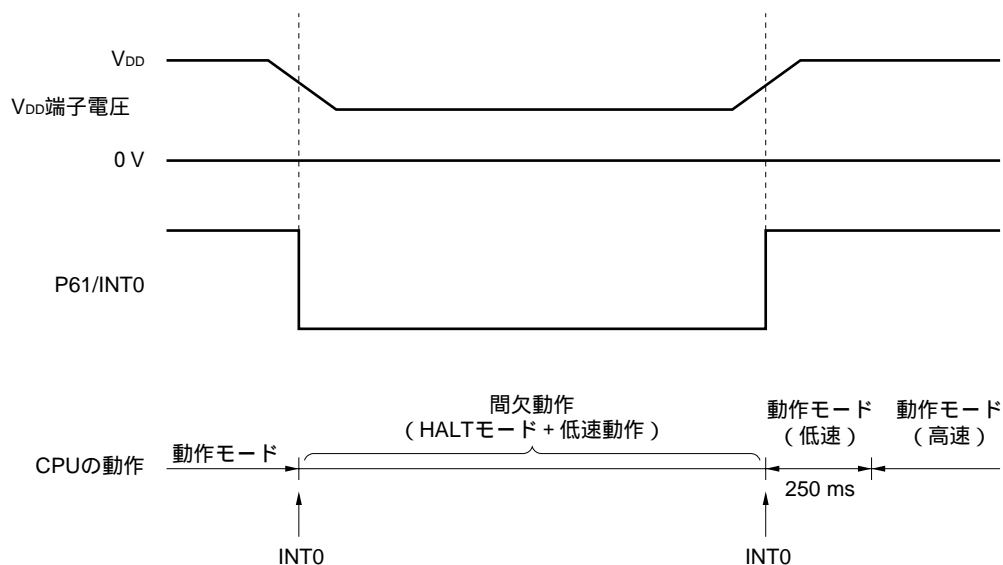
スタンバイ・モード中は250 ms間隔で、間欠動作する (INTBT)。

INT0, INTBTは低位の割り込みとする。

スタンバイ・モード中のCPUクロックは最低速とする。

★

タイミング・チャート



プログラム例

```

BTAND4 : SKTCLR      IRQ0          ; INT0 = 1?
          BR         VSUBBT       ; NO
          SKT        PORT6.1      ; P61 = 1?
          BR         PDOWN        ; パワー・ダウン
          SET1       BTM.3        ; BTスタート
WAIT :    SKT        IRQBT        ; 250 msのウエイト
          BR         WAIT
          SKT        PORT6.1
          BR         PDOWN
          MOV        A, #0011B    ; 高速モード
          MOV        PCC, A
          [ EI      IEn ]        ; IEn 1
          RETI
PDOWN :  MOV        A, #0         ; 最低速モード
          MOV        PCC, A
          [ DI      IEn ]        ; IEn 0
                                     }
                                     32マシン・サイクル確保する。
SETHLT : HALT                ; HALTモード
          NOP
          RETI
VSUBBT : CLR1      IRQBT
          .....
          BR         SETHLT
    
```

間欠動作時の処理



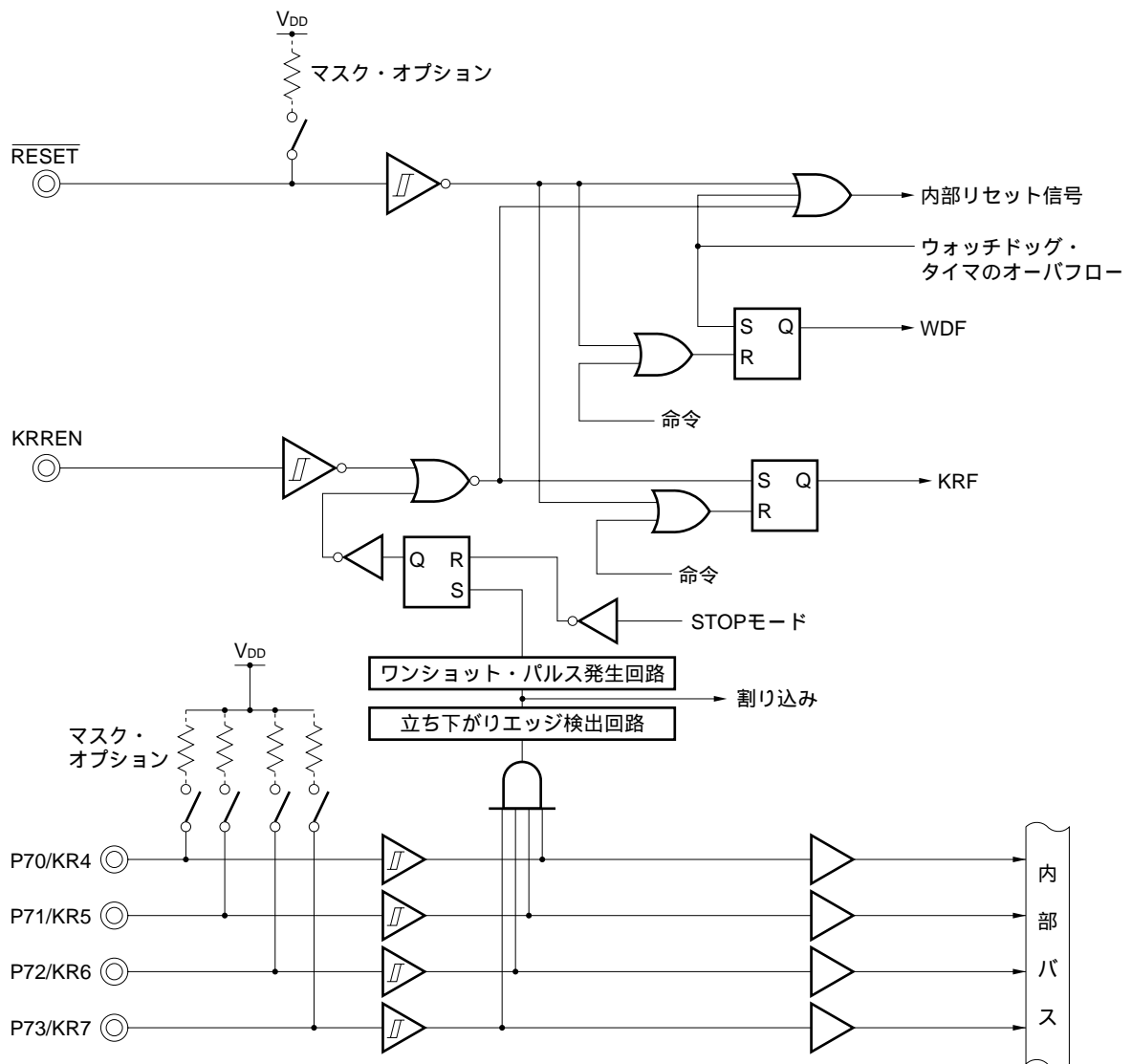
〔メ モ〕

# 第8章 リセット機能

## 8.1 リセット機能の構成および動作

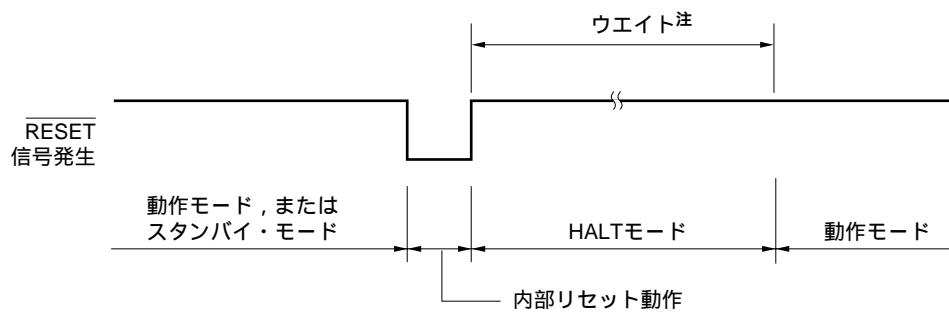
リセット入力には、外部リセット信号 ( $\overline{\text{RESET}}$ ) とベーシック・インターバル・タイマ/ウォッチドッグ・タイマからのリセット信号、キー・リターン・リセットの3種類があります。いずれかのリセット信号が入力されると、内部リセット信号が発生します。図8-1にそれぞれの構成を示します。

図8-1 リセット機能の構成



$\overline{\text{RESET}}$ 信号発生によって、各ハードウェアは表8-1に示すようにイニシャライズされます。リセット動作のタイミングを図8-2に示します。

図8 - 2  $\overline{\text{RESET}}$ 信号発生によるリセット動作



注 マスク・オプションにより次の2つの時間を選択することができます。

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

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

表8 - 1 各ハードウェアのリセット後の状態 (1/3)

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 の下位3ビットをPC10-PC8に, 0001H番地の内容をPC7-PC0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-SK2)	0	0
	割り込みステータス・フラグ (IST0, IST1)	0	0
	バンク許可フラグ (MBE, RBE)	プログラム・メモリの0000H番地の のビット6をRBE, ビット7を MBEにセット	左に同じ
スタック・ポインタ (SP)		不定	不定
スタック・バンク選択レジスタ (SBS)		1000B	1000B
データ・メモリ (RAM)		保持	不定
汎用レジスタ (X, A, H, L, D, E, B, C)		保持	不定
バンク選択レジスタ (MBS, RBS)		0, 0	0, 0
★ ベーシック・インターバル・タイマ/ウォッチ ドッグ・タイマ	カウンタ (BT)	不定	不定
	モード・レジスタ (BTM)	0	0
タイマ・カウンタ (T0)	ウォッチドッグ・タイマ許可 フラグ (WDTM)	0	0
	カウンタ (T0)	0	0
	モジュロ・レジスタ (TMOD0)	FFH	FFH
	モード・レジスタ (TM0)	0	0
		TOE0, TOUT F/F	0, 0
タイマ・カウンタ (T1)	カウンタ (T1)	0	0
	モジュロ・レジスタ (TMOD1)	FFH	FFH
	モード・レジスタ (TM1)	0	0
			TOE1, TOUT F/F

表8 - 1 各ハードウェアのリセット後の状態 (2/3)

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
タイマ・カウンタ (T2)	カウンタ (T2)	0	0
	モジュロ・レジスタ (TMOD2)	FFH	FFH
	ハイ・レベル期間設定用 モジュロ・レジスタ (TMOD2H)	FFH	FFH
	モード・レジスタ (TM2)	0	0
	TOE2, TOUT F/F	0, 0	0, 0
	REMC, NRZ, NRZB	0, 0, 0	0, 0, 0
クロック発生 回路	プロセッサ・クロック・ コントロール・レジスタ (PCC)	0	0
割り込み機能	割り込み要求フラグ (IRQ $\times\times\times$ )	リセット (0)	リセット (0)
	割り込み許可フラグ (IE $\times\times\times$ )	0	0
	割り込みマスタ許可フラグ (IME)	0	0
	割り込みプライオリティ 選択レジスタ (IPS)	0	0
	INT0, 2モード・レジスタ (IM0, IM2)	0, 0	0, 0
デジタル・ ポート	出力バッファ	オフ	オフ
	出力ラッチ	クリア (0)	クリア (0)
	入出力モード・レジスタ (PMGA, PMGC)	0	0
	プルアップ抵抗指定レジスタ (POGA, POGB)	0	0
ビット・シーケンシャル・バッファ (BSB0-BSB3)		保持	不定

表8 - 1 各ハードウェアのリセット後の状態 (3/3)

ハードウェア	キー・リターン・リセッ トによる $\overline{\text{RESET}}$ 信号発生	スタンバイ・モード 中の $\overline{\text{RESET}}$ 信号発生	動作中のWDTによる $\overline{\text{RESET}}$ 信号発生	動作中の $\overline{\text{RESET}}$ 信号 発生
ウォッチドッグ・フラグ (WDF)	それ以前の状態を保 持する	0	1	0
キー・リターン・フラグ (KRF)	1	0	それ以前の状態を保 持する	0

## 8.2 ウォッチドッグ・フラグ (WDF) , キー・リターン・フラグ (KRF)

WDFはFC6H番地のビット2に、KRFはFC6H番地のビット3にマッピングされています。

WDF, KRFは初期状態では不定となっておりますが、外部RESET信号の発生によりクリアされます。

WDFはウォッチドッグ・タイマのオーバーフロー信号の発生により、KRFはKRnによるリセット信号の発生によりセットされます。このため、WDFおよびKRFの内容によりリセット信号の種類の確認ができます。

WDF, KRFは、外部信号および命令によってのみクリアすることができます。このため、一度セットされたあとは、外部信号またはクリア命令が実行されるまで内容がクリアされることはありません。リセット・スタート後にSKTCLR命令などによってWDFおよびKRFの内容を確認し、クリアしてください。

表8-2に各信号に対するWDF, KRFの内容を、図8-3に各信号発生時におけるWDFの動作、図8-4に各信号発生時におけるKRFの動作を示します。

表8-2 各信号に対するWDF, KRFの内容

ハードウェア	外部RESET信号の発生	ウォッチドッグ・タイマのオーバーフローによるリセット信号の発生	KRnによるリセット信号の発生	WDFクリア命令の実行	KRFクリア命令の実行
ウォッチドッグ・フラグ (WDF)	0	1	保持	0	保持
キー・リターン・フラグ (KRF)	0	保持	1	保持	0

図8-3 各信号発生時におけるWDFの動作

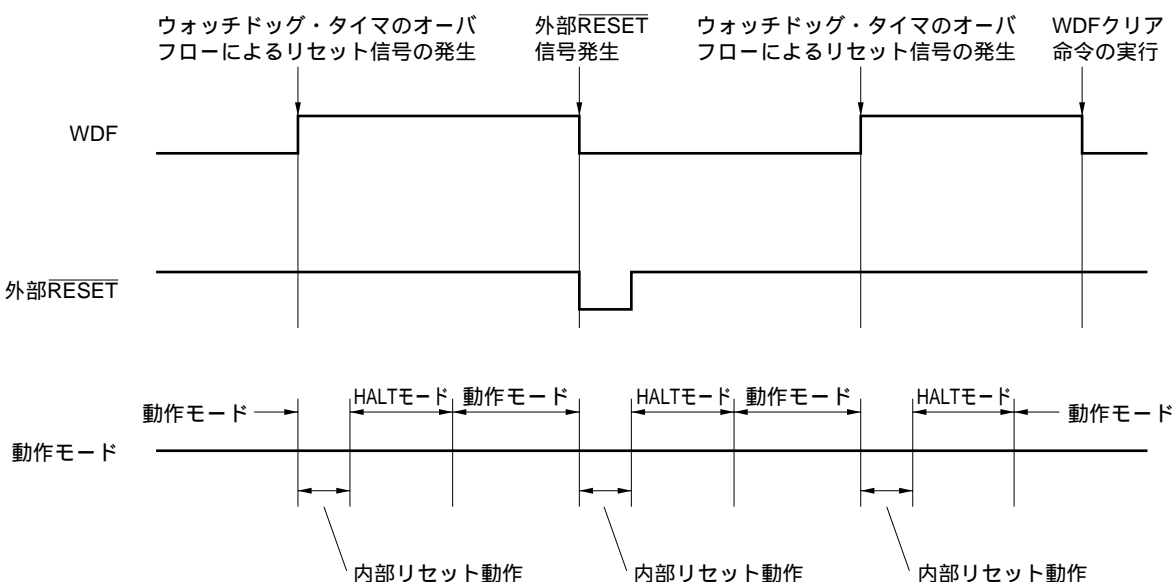
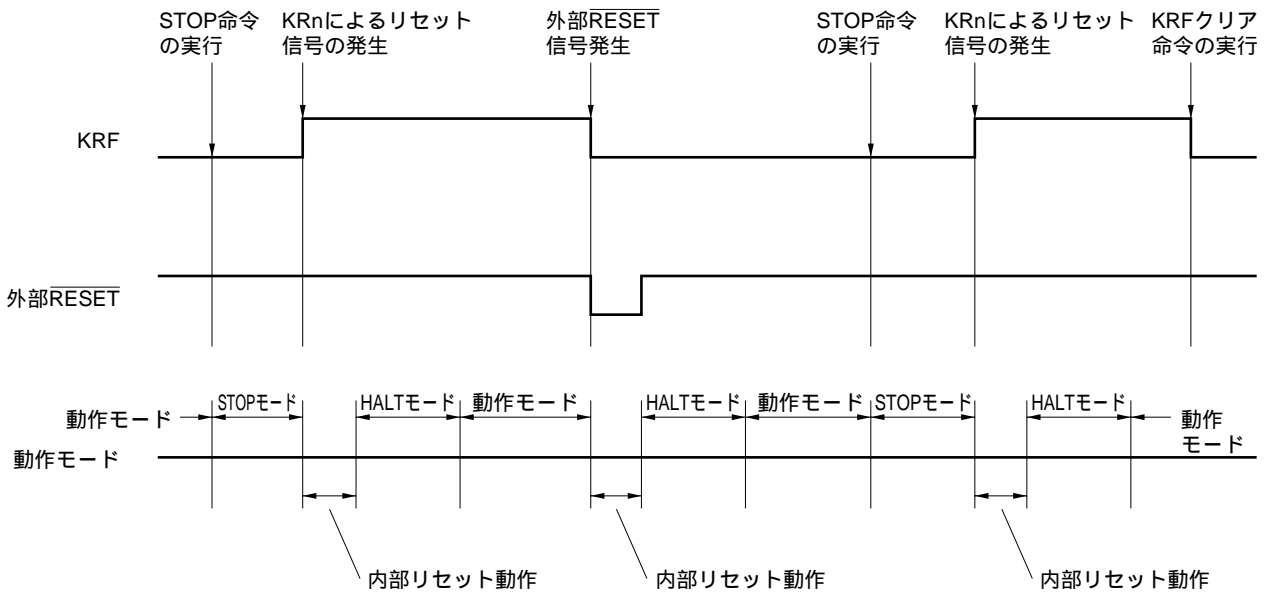


図8-4 各信号発生時におけるKRFの動作



## 第9章 マスク・オプション

μPD754202には、次のようなマスク・オプションがあります。

表9 - 1 マスク・オプションの選択

項目	マスク・オプション
P70/KR4-P73/KR7	マスク・オプションにより、ビット単位でプルアップ抵抗の内蔵を指定可能
RESET端子	マスク・オプションにより、プルアップ抵抗の内蔵を指定可能
発振安定ウエイト時間	$2^{17}/f_x$ , $2^{15}/f_x$ の2つから選択可能

### 9.1 端子のマスク・オプション

#### 9.1.1 P70/KR4-P73/KR7のマスク・オプション

P70/KR4-P73/KR7 (ポート7) は、マスク・オプションによりプルアップ抵抗の内蔵を指定することができます。マスク・オプションはビット単位で指定することができます。

マスク・オプションは、次の3通りから選択できます。

プルアップ抵抗を内蔵しない

30 k (typ.) のプルアップ抵抗を内蔵する

100 k (typ.) のプルアップ抵抗を内蔵する

ポート7のリセット後の状態は、マスク・オプションによるプルアップ抵抗の内蔵を指定した場合、ハイ・レベルになります。内蔵を指定しない場合は、ハイ・インピーダンスになります。

#### 9.1.2 $\overline{\text{RESET}}$ 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100 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 発振安定ウエイト時間のマスク・オプション

発振安定ウエイト時間は、マスク・オプションにより選択することができます。これは、 $\overline{\text{RESET}}$ 信号によってスタンバイ機能を解除したあと、通常動作モードに戻るまでの時間です (詳細は7.2 スタンバイ・モードの解除参照)。

ウエイト時間は次の2つの時間から選択することができます。

- ①  $2^{17}/f_x$  (21.8 ms :  $f_x = 6.0$  MHz動作時, 31.3 ms :  $f_x = 4.19$  MHz動作時)
- ②  $2^{15}/f_x$  (5.46 ms :  $f_x = 6.0$  MHz動作時, 7.81 ms :  $f_x = 4.19$  MHz動作時)



〔メ モ〕

## 第10章 命令セット

μPD754202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，μPD754202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 ) 2 Kバイト空間(0000H-07FFH)のサブルーチン・コール命令
- ( b ) 2 Kバイト空間(0000H-07FFH)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

μ PD754202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, IE x x x , IRQ x x x	FB0H-FBFH
	PORT3, 6, 7, 8	FF0H-FFFH
pmem. @L	BSB0-3, PORT8	FC0H-FFFH
@H + mem. bit	ビット操作可能なすべての周辺ハードウェア	MBで指定されるメモリ・バンクのビット操作可能なすべてのビット

備考 1 . x x x : 0, 2, BT, T0, T1, T2

2 . MB = MBE · MBS

### 10.1.3 たてづみ命令

μ PD754202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μPD754202の命令セットでは、4ビット・データの加算あるいは減算の結果を、任意の進数に補正するための進数補正命令が用意されています。

#### (1) 加算時の進数補正

補正したい進数値をmとすると

```

ADDS  A, #16 - m
ADDC  A, @HL      ; A, CY  A + (HL) + CY
ADDS  A, #m

```

の組み合わせによって、アキュムレータとメモリ（HL）を加算し、加算結果をm進補正します。オーバーフローはキャリー・フラグに残ります。

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

例 アキュムレータと、メモリを10進加算する。

```

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

```

#### (2) 減算時の進数補正

補正したい進数値をmとすると

```

SUBC  A, @HL
ADDS  A, #m

```

の組み合わせにより、アキュムレータからメモリ（HL）を減算し、結果をm進補正します。

アンダフローはキャリー・フラグに残ります。

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

### 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

μPD754202の命令セットでは、スキップによって条件判断をして、プログラムを構成するようになっていきます。

スキップ命令を実行したとき、スキップ条件が満足されると、続く1命令を飛ばして(スキップして)1命令あとの命令が実行されます。

スキップが発生したとき、スキップに要するマシン・サイクル数は

(a) スキップ命令に続く命令(スキップされる命令)が、3バイト命令(BR !addr命令, BRA !addr1命令, CALL !addr命令, CALLA !addr1命令)の場合: 2マシン・サイクル

(b) (a)以外の命令の場合: 1マシン・サイクル

## 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**(EEU-730)を参照してください)。記述方法の中で、複数個あるものは、それらの要素の1つを選択します。大文字で書かれた英字および+, -記号はキー・ワードであり、そのまま記述します。

イミューディエト・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・フラグの略語を記述できます(ただし、fmem, pmemは記述できるラベルに制限があります。詳しくは、表3-1 **アドレッシング・モード**, 図3-7 **I/Oマップ**を参照してください)。

表現形式	記述方法
reg	X, A, B, C, D, E, H, L
reg1	X, B, C, D, E, H, L
rp	XA, BC, DE, HL
rp1	BC, DE, HL
rp2	BC, DE
rp'	XA, BC, DE, HL, XA', BC', DE', HL'
rp'1	BC, DE, HL, XA', BC', DE', HL'
rpa	HL, HL + , HL - , DE, DL
rpa1	DE, DL
n4	4ビット・イミディエト・データまたはレーベル
n8	8ビット・イミディエト・データまたはレーベル
mem	8ビット・イミディエト・データまたはレーベル <sup>注</sup>
bit	2ビット・イミディエト・データまたはレーベル
fmem	FB0H-FBFH, FF0H-FFFHイミディエト・データまたはレーベル
pmem	FC0H-FFFHイミディエト・データまたはレーベル
addr	0000H-07FFHイミディエト・データまたはレーベル
addr1	0000H-07FFHイミディエト・データまたはレーベル
caddr	12ビット・イミディエト・データまたはレーベル
faddr	11ビット・イミディエト・データまたはレーベル
taddr	20H-7FHイミディエト・データ(ただしbit0 = 0)またはレーベル
PORTn	PORT3, 6, 7, 8
IE x x x	IEBT, IET0-IET2, IE0, IE2
RBn	RB0-RB3
MBn	MB0, MB15

注 memは、8ビット・データ処理の場合は偶数アドレスのみ記述できます。

## (2) オペレーション説明上の凡例

A	: Aレジスタ; 4ビット・アキュムレータ
B	: Bレジスタ
C	: Cレジスタ
D	: Dレジスタ
E	: Eレジスタ
H	: Hレジスタ
L	: Lレジスタ
X	: Xレジスタ
XA	: レジスタ・ペア (XA); 8ビット・アキュムレータ
BC	: レジスタ・ペア (BC)
DE	: レジスタ・ペア (DE)
HL	: レジスタ・ペア (HL)
XA'	: 拡張レジスタ・ペア (XA')
BC'	: 拡張レジスタ・ペア (BC')
DE'	: 拡張レジスタ・ペア (DE')
HL'	: 拡張レジスタ・ペア (HL')
PC	: プログラム・カウンタ
SP	: スタック・ポインタ
CY	: キャリー・フラグ; ビット・アキュムレータ
PSW	: プログラム・ステータス・ワード
MBE	: メモリ・バンク許可フラグ
RBE	: レジスタ・バンク許可フラグ
PORT <sub>n</sub>	: ポート <sub>n</sub> (n=3, 6, 7, 8)
IME	: 割り込みマスタ許可フラグ
IPS	: 割り込みプライオリティ選択レジスタ
IE <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・エリア欄の記号説明

* 1	MB = MBE · MBS ( MBS = 0, 15 )	データ・メモリ・ アドレッシング
* 2	MB = 0	
* 3	MBE = 0 : MB = 0 ( 000H-07FH ) MB = 15 ( F80H-FFFFH ) MBE = 1 : MB = MBS ( MBS = 0, 15 )	
* 4	MB = 15, fmem = FB0H-FBFH, FF0H-FFFFH	
* 5	MB = 15, pmem = FC0H-FFFFH	
* 6	addr = 0000H-07FFH	プログラム・メモリ・ アドレッシング
* 7	addr, addr1 = ( Current PC ) - 15 ~ ( Current PC ) - 1 ( Current PC ) + 2 ~ ( Current PC ) + 16	
* 8	caddr = 0000H-07FFH	
* 9	faddr = 0000H-07FFH	
* 10	taddr = 0020H-007FH	
* 11	addr1 = 0000H-07FFH	

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<sub>cy</sub>) 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	アドレッシング・エリア	スキップ条件
テーブル参照	MOV <sub>T</sub>	XA, @PCDE	1	3	XA (PC <sub>10-8</sub> + DE) <sub>ROM</sub>		
		XA, @PCXA	1	3	XA (PC <sub>10-8</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	MOV <sub>1</sub>	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、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	アドレッシング・エリア	スキップ条件
操作 アキュムレータ	RORC	A	1	1	CY A <sub>0</sub> , A <sub>3</sub> CY, A <sub>n-1</sub> A <sub>n</sub>		
	NOT	A	2	2	A $\bar{A}$		
増減	INCS	reg	1	1 + S	reg reg + 1		reg = 0
		rp1	1	1 + S	rp1 rp1 + 1		rp1 = 00H
		@HL	2	2 + S	(HL) (HL) + 1	*1	(HL) = 0
		mem	2	2 + S	(mem) (mem) + 1	*3	(mem) = 0
	DECS	reg	1	1 + S	reg reg - 1		reg = FFH
		rp'	2	2 + S	rp' rp' - 1		rp' = FFH
比較	SKE	reg, #n4	2	2 + S	Skip if reg = n4		reg = n4
		@HL, #n4	2	2 + S	Skip if (HL) = n4	*1	(HL) = n4
		A, @HL	1	1 + S	Skip if A = (HL)	*1	A = (HL)
		XA, @HL	2	2 + S	Skip if XA = (HL)	*1	XA = (HL)
		A, reg	2	2 + S	Skip if A = reg		A = reg
		XA, rp'	2	2 + S	Skip if XA = rp'		XA = rp'
操作 キャリア・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 $\overline{CY}$		
メモリ・ビット操作	SET1	mem.bit	2	2	(mem.bit) 1	*3	
		fmem.bit	2	2	(fmem.bit) 1	*4	
		pmem.@L	2	2	(pmem <sub>7-2</sub> + L <sub>3-2</sub> .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	アドレッシング・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 (pmem7-2 + L3-2.bit (L1-0))	*5		
		CY, @H + mem.bit	2	2	CY CY (H + mem3-0.bit)	*1		
	OR1	CY, fmem.bit	2	2	CY CY (fmem.bit)	*4		
		CY, pmem.@L	2	2	CY CY (pmem7-2 + L3-2.bit (L1-0))	*5		
		CY, @H + mem.bit	2	2	CY CY (H + mem3-0.bit)	*1		
	XOR1	CY, fmem.bit	2	2	CY CY $\nabla$ (fmem.bit)	*4		
		CY, pmem.@L	2	2	CY CY $\nabla$ (pmem7-2 + L3-2.bit (L1-0))	*5		
		CY, @H + mem.bit	2	2	CY CY $\nabla$ (H + mem3-0.bit)	*1		
分岐	BR <sup>注1</sup>	addr	-	-	PC <sub>10-0</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>10-0</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>10-0</sub> addr	*6		
		\$addr	1	2	PC <sub>10-0</sub> addr	*7		
		\$addr1	1	2	PC <sub>10-0</sub> addr1	*7		
		PCDE	2	3	PC <sub>10-0</sub> PC <sub>10-8</sub> + DE			
		PCXA	2	3	PC <sub>10-0</sub> PC <sub>10-8</sub> + XA			
		BCDE	2	3	PC <sub>10-0</sub> BCDE <sup>注2</sup>	*6		
		BCXA	2	3	PC <sub>10-0</sub> BCXA <sup>注2</sup>	*6		
		BRA <sup>注1</sup>	!addr1	3	3	PC <sub>10-0</sub> addr1	*11	
		BRCB	!caddr	2	2	PC <sub>10-0</sub> caddr <sub>10-0</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、0を設定してください。

命令群	二モニク	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件
サブルーチン・スタック制御	CALLA <sup>注</sup>	laddr1	3	3	(SP - 6) (SP - 3) (SP - 4) 0, PC <sub>10-0</sub> (SP - 5) 0, 0, 0, 0 (SP - 2) x, x, MBE, RBE PC <sub>10-0</sub> addr1, SP SP - 6	* 11	
	CALL <sup>注</sup>	laddr	3	3	(SP - 4) (SP - 1) (SP - 2) 0, PC <sub>10-0</sub> (SP - 3) MBE, RBE, 0, 0 PC <sub>10-0</sub> addr, SP SP - 4	* 6	
				4	(SP - 6) (SP - 3) (SP - 4) 0, PC <sub>10-0</sub> (SP - 5) 0, 0, 0, 0 (SP - 2) x, x, MBE, RBE PC <sub>10-0</sub> addr, SP SP - 6		
	CALLF <sup>注</sup>	!faddr	2	2	(SP - 4) (SP - 1) (SP - 2) 0, PC <sub>10-0</sub> (SP - 3) MBE, RBE, 0, 0 PC <sub>10-0</sub> 0 + faddr, SP SP - 4	* 9	
				3	(SP - 6) (SP - 3) (SP - 4) 0, PC <sub>10-0</sub> (SP - 5) 0, 0, 0, 0 (SP - 2) x, x, MBE, RBE PC <sub>10-0</sub> 0 + faddr, SP SP - 6		
	RET <sup>注</sup>		1	3	MBE, RBE, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) SP SP + 4		
x, x, MBE, RBE (SP + 4) 0, 0, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) SP SP + 6							
RETS <sup>注</sup>		1	3 + S	MBE, RBE, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) SP SP + 4 then skip unconditionally		無条件	
				x, x, MBE, RBE (SP + 4) 0, 0, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) SP SP + 6 then skip unconditionally			
RETI <sup>注</sup>		1	3	MBE, RBE, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6			
				0, 0, 0, 0 (SP + 1) PC <sub>10-0</sub> (SP) <sub>2-0</sub> (SP + 3) (SP + 2) PSW (SP + 4) (SP + 5), SP SP + 6			

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

命令群	二モニック	オペランド	バイト数	マシン・サイクル	オペレーション	アドレッシング・エリア	スキップ条件	
スタック制御	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1</sup>	A, PORTn	2	2	A PORTn (n = 3, 6, 7, 8)			
	OUT <sup>注1</sup>	PORTn, A	2	2	PORTn A (n = 3, 6, 8)			
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, 15)			
	GETI <sup>注2, 3</sup>	taddr	1	3	・ TBR命令のとき PC <sub>10-0</sub> (taddr) <sub>2-0</sub> + (taddr + 1)	* 10		
					・ TCALL命令のとき (SP - 4) (SP - 1) (SP - 2) 0, PC <sub>10-0</sub> (SP - 3) MBE, RBE, 0, 0 PC <sub>10-0</sub> (taddr) <sub>2-0</sub> + (taddr + 1) SP SP - 4			
					・ TBR, TCALL命令以外のとき (taddr) (taddr + 1)の命令実行			参照した命令による
					1			3
4	・ TCALL命令のとき (SP - 6) (SP - 3) (SP - 4) 0, PC <sub>10-0</sub> (SP - 5) 0, 0, 0, 0 (SP - 2) x, x, MBE, RBE PC <sub>10-0</sub> (taddr) <sub>2-0</sub> + (taddr + 1) SP SP - 6							
	3	・ TBR, TCALL命令以外のとき (taddr) (taddr + 1)の命令実行	参照した命令による					

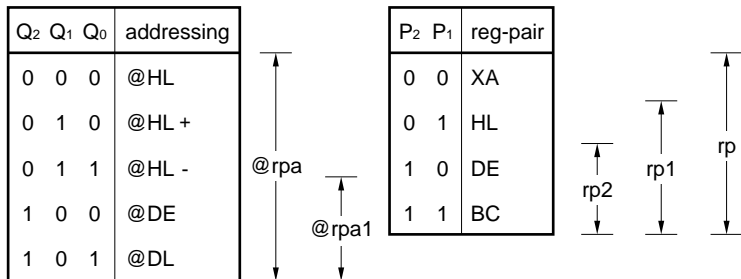
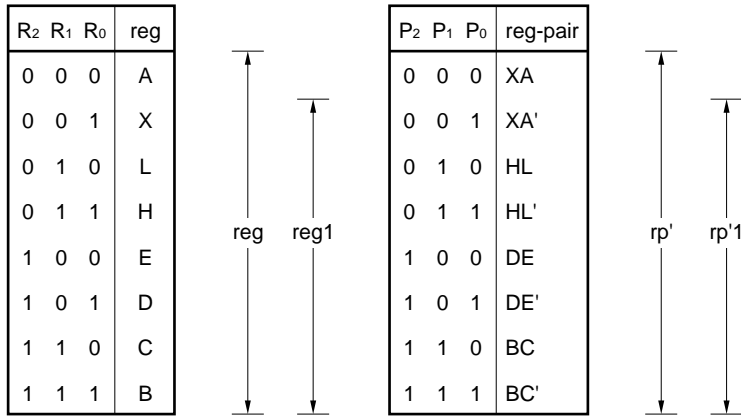
注1 . IN/OUT命令実行時には, MBE = 0またはMBE = 1, MBS = 15としておく必要があります。

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。

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。

### 10.3 各命令の命令コード

(1) 命令コードの記号説明



N <sub>5</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	IE x x x
0	0	0	0	IEBT
0	1	0	0	IET0
0	1	1	0	IE0
0	1	1	1	IE2
1	1	0	0	IET1
1	1	0	1	IET2

I<sub>n</sub> : n4, n8に対するイミーディエト・データ

D<sub>n</sub> : memに対するイミーディエト・データ

B<sub>n</sub> : bitに対するイミーディエト・データ

N<sub>n</sub> : n, IE x x xに対するイミーディエト・データ

T<sub>n</sub> : taddr ×  $\frac{1}{2}$  に対するイミーディエト・データ

A<sub>n</sub> : [ 分岐先アドレスとの相対アドレス距離 ( 2 ~ 16 ) ] - 1 に対するイミーディエト・データ

S<sub>n</sub> : [ 分岐先アドレスとの相対アドレス距離 ( 15 ~ 1 ) ] の 1 の補数に対するイミーディエト・データ

(2) ビット操作アドレッシングの命令コード

オペランドの欄の \* 1 は、以下の3種類があることを示します。

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

上記アドレッシングに対応する命令コードの2バイト目 \* 2 は、下表のようになります。

* 1	命令コード・2バイト目								アクセス可能なビット
fmem. bit	1	0	B <sub>1</sub>	B <sub>0</sub>	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	FB0H-FBFHの操作可能ビット
	1	1	B <sub>1</sub>	B <sub>0</sub>	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	FF0H-FFFHの操作可能ビット
pmem. @L	0	1	0	0	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>	FC0H-FFFHの操作可能ビット
@H + mem. bit	0	0	B <sub>1</sub>	B <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	アクセス可能なメモリ・バンクの操作可能ビット

B<sub>n</sub> : bitに対するイミディエト・データ

F<sub>n</sub> : fmemに対するイミディエト・データ

(アドレスの下位4ビットを示す)

G<sub>n</sub> : pmemに対するイミディエト・データ

(アドレスのビット5-2を示す)

D<sub>n</sub> : memに対するイミディエト・データ

(アドレスの下位4ビットを示す)



命令群	二モニック	オペランド	命 令 コ ー ド		
			B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>
転送	MOV	A, #n4	0 1 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>		
		reg1, #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> 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	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, @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, @BCXA	1 1 0 1 0 0 0 1		
		XA, @BCDE	1 1 0 1 0 1 0 1		
転送ビット	MOV <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	laddr	1 0 1 0 1 0 1 1	0 0 ←	addr →
		\$addr1 <sup>(+16)</sup>	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	0 0 0 0 0 1 0 1			
BCXA	1 0 0 1 1 0 0 1	0 0 0 0 0 0 0 1			
分岐	BRA	laddr1	1 0 1 1 1 0 1 0	0 ←	addr1 →
	BRCB	lcaddr	0 1 0 1 ←	caddr →	
サブリーチン・スタック制御	CALLA	laddr1	1 0 1 1 1 0 1 1	0 ←	addr1 →
		CALL	laddr	1 0 1 0 1 0 1 1	0 1 ←
	CALLF	lfaddr	0 1 0 0 0 ←	faddr →	
	RET		1 1 1 0 1 1 1 0		
	RETS		1 1 1 0 0 0 0 0		
	RETI		1 1 1 0 1 1 1 1		
	PUSH	rp	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 1		
		BS	1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 1	
POP	rp	0 1 0 0 1 P <sub>2</sub> P <sub>1</sub> 0			
	BS	1 0 0 1 1 0 0 1	0 0 0 0 0 1 1 0		
割り込み制御	EI		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 0	
		IE x x x	1 0 0 1 1 1 0 1	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
	DI		1 0 0 1 1 1 0 0	1 0 1 1 0 0 1 0	
		IE x x x	1 0 0 1 1 1 0 0	1 0 N <sub>5</sub> 1 1 N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
入出力	IN	A, PORT <sub>n</sub>	1 0 1 0 0 0 1 1	1 1 1 1 N <sub>3</sub> N <sub>2</sub> N <sub>1</sub> N <sub>0</sub>	
	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>	
CPU制御	HALT		1 0 0 1 1 1 0 1	1 0 1 0 0 0 1 1	
	STOP		1 0 0 1 1 1 0 1	1 0 1 1 0 0 1 1	
	NOP		0 1 1 0 0 0 0 0		
特殊	SEL	RB <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。μPD754202のMk モード、Mk モードでは、使用できる命令、または命令の機能が異なりますので、次の読み方に従って読んでください。

### 読み方

- Mk モード時とMk モード時に共通して使用できます。
- Mk モード時のみ使用できます。
- Mk モード時のみ使用できます。
- / 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，MOV A, #n4またはMOV XA, #n8命令が続けて置かれると実行した命令以降のたてづみ命令はNOPとして処理されます。

## 応用例

（1）アキュムレータに0BHをセットする。

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

（2）ポート3に出力するデータを0-2から選ぶ。

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

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

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

```
OUT PORT3, A
```

## ○ MOV reg1, #n4

機能：reg1 n4 n4 = l3-0 0-FH

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

## ○ MOV XA, #n8

機能：XA n8 n8 = l7-0 : 00H-FFH

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

## ○ MOV HL, #n8

機能: HL 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命令として処理されます。

## ○ MOV rp2, #n8

機能: rp2 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rp2 (BC, DE) に転送します。

## ○ MOV A, @HL

機能: A (HL)

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。

## ○ MOV A, @HL +

機能: A (HL), L L + 1  
skip if L = 0H

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。そのあとLレジスタの内容を自動的に+ 1し、その結果が0Hになると、続く1命令をスキップします。

## ○ MOV A, @HL -

機能: A (HL), L L - 1  
skip if L = FH

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送します。そのあとLレジスタの内容を自動的に- 1し、その結果がFHになると、続く1命令をスキップします。

## ○ MOV A, @rpa1

機能：A (rpa)

rpa = HL + のとき：skip if L = 0

rpa = HL - のとき：skip if L = FH

レジスタ・ペアrpa (HL, HL + , HL - , DE, DL) でアドレスされるデータ・メモリの内容をAレジスタに転送します。

rpaとしてオートインクリメント (HL + ) を指定した場合は、データの転送後、Lレジスタの内容を自動的に + 1 し、その結果が0になると続く1命令をスキップします。

rpaとしてオートデクリメント (HL - ) を指定した場合は、データの転送後、Lレジスタの内容を自動的に - 1 し、その結果がFHになると続く1命令をスキップします。

## ○ MOV XA, @HL

機能：A (HL), X (HL + 1)

レジスタ・ペアHLでアドレスされるデータ・メモリの内容をAレジスタに転送し、メモリの次の番地の内容をXレジスタに転送します。

ただし、Lレジスタの内容が奇数の場合、最下位ビットが無視されたアドレスが転送されます。

### 応用例

3EH, 3FH番地のデータをレジスタ・ペアXAに転送する。

```
MOV HL, #3EH
```

```
MOV XA, @HL
```

## ○ MOV @HL, A

機能：(HL) A

Aレジスタの内容をレジスタ・ペアHLでアドレスされるデータ・メモリへ転送します。



## ○ MOV @HL, XA

機能：(HL) ← A, (HL + 1) ← X

Aレジスタの内容をレジスタ・ペアHLでアドレスされるデータ・メモリに転送し、Xレジスタの内容をメモリの次の番地に転送します。

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

## ○ MOV A, mem

機能：A ← (mem) mem = D7-0 : 00H-FFH

8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容をAレジスタに転送します。

## ○ MOV XA, mem

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

8ビット・イミディエト・データmemでアドレスされるデータ・メモリの内容をAレジスタに、その次の番地の内容をXレジスタに転送します。

memで指定できるアドレスは偶数アドレスです。

## 応用例

40H, 41H番地のデータをレジスタ・ペアXAに転送する。

```
MOV XA, 40H
```

## ○ MOV mem, A

機能：(mem) ← A mem = D7-0 : 00H-FFH

Aレジスタの内容を8ビット・イミディエト・データmemでアドレスされるデータ・メモリに転送します。

## ○ MOV mem, XA

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

Aレジスタの内容を8ビット・イミディエト・データmemでアドレスされるデータ・メモリに転送し、Xレジスタの内容をメモリの次の番地に転送します。

memで指定できるアドレスは偶数アドレスです。

## ○ MOV A, reg

機能：A reg

レジスタreg ( X, A, H, L, D, E, B, C ) の内容をAレジスタに転送します。

## ○ MOV XA, rp'

機能：XA rp'

レジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をレジスタ・ペアXAへ転送します。

## 応用例

レジスタ・ペアXA'のデータをレジスタ・ペアXAに転送する。

```
MOV XA, XA'
```

## ○ MOV reg1, A

機能：reg1 A

Aレジスタの内容をレジスタreg1 ( X, H, L, D, E, B, C ) に転送します。

## ○ MOV rp'1, XA

機能：rp'1 XA

レジスタ・ペアXAの内容をレジスタ・ペアrp'1 ( HL, DE, BC, XA', HL', DE', BC' ) に転送します。

○ XCH A, @HL

機能: A (HL)

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

○ XCH A, @HL +

機能: A (HL), L L + 1  
skip if L = 0H

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換します。その後Lレジスタの内容を自動的に + 1 し、その結果が0Hになると、続く1命令をスキップします。

○ XCH A, @HL -

機能: A (HL), L L - 1  
skip if L = FH

Aレジスタの内容とレジスタ・ペアHLでアドレスされるデータ・メモリの内容を交換します。その後Lレジスタの内容を自動的に - 1 し、その結果がFHになると、続く1命令をスキップします。

## ○ XCH A, @rpa1

**機能** : A ( rpa )

rpa = HL + のとき : skip if L = 0

rpa = HL - のとき : skip if L = FH

Aレジスタの内容とレジスタ・ペアrpa ( HL, HL + , HL - , DE, DL ) でアドレスされるデータ・メモリの内容を交換します。rpaとしてオートインクリメント ( HL + ) / オートデクリメント ( HL - ) を指定した場合は、データの交換後、Lレジスタの内容を自動的に + 1 / - 1 し、その結果がHL+の場合は0に、HL-の場合はFHになると続く1命令をスキップします。

### 応用例

データ・メモリの20H-2FH番地のデータと30H-3FH番地のデータを交換する。

```
SEL MB0
MOV D, #2
MOV HL, #30H
LOOP : XCH A, @HL ; A ( 3 × )
      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・イミディエイト・データmemでアドレスされるデータ・メモリの内容を交換します。

## ○ XCH XA, mem

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

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

memで指定できるアドレスは偶数アドレスです。

## ○ XCH A, reg1

機能: A reg1

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

## ○ XCH XA, rp'

機能: XA rp'

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

## 10.4.2 テーブル参照命令

## ○ MOVT XA, @PCDE

機能: XA ROM (PC<sub>10-8</sub> + DE)

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

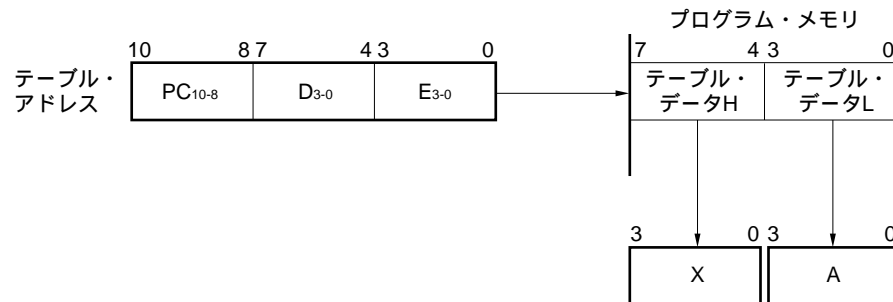
テーブル・アドレスはこの命令実行時のプログラム・カウンタ (PC) の内容によって決定されます。

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

プログラム・カウンタはこの命令の実行により影響を受けません。

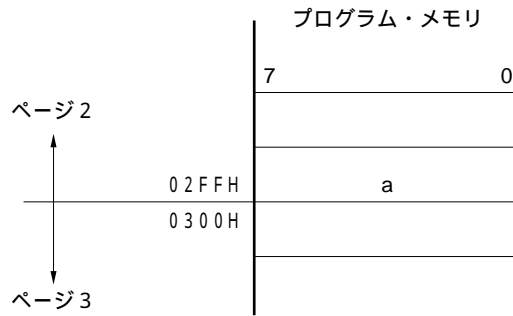
この命令は、テーブル・データを連続して参照する場合に有効です。

## 例



**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  $0 \times F0H$ - $0 \times FFH$  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  0xF0H
      DB   x x H, x x H, ..... ; テーブル・データ
    
```

## ○ MOVT XA, @PCXA

**機能:** XA ROM (PC<sub>10-8</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x × 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。

## ○ MOV<sub>T</sub> XA, @BCDE

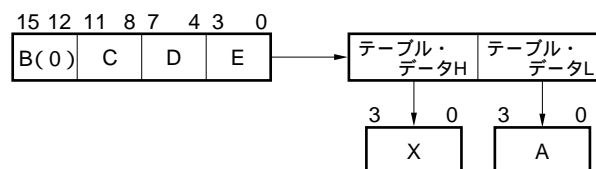
**機能:** XA ROM (BCDE)

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, μPD754202では, レジスタBは無効です。レジスタBには必ず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。

### 例





# MOV<sub>T</sub> XA, @BCXA

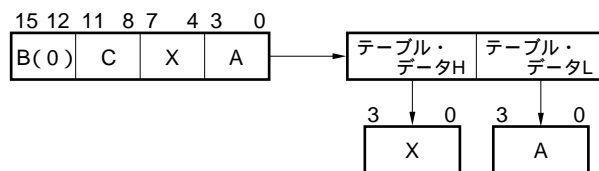
機能: XA ROM (BCXA)

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,  $\mu$ PD754202では, レジスタBは無効です。レジスタBには必ず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。

例



## 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 SHR 6 ; 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'    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    rp'1 - XA ; Skip if borrow

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

キャリー・フラグには影響を与えません。

## ○ SUBC A, @HL

**機能** : A, CY    A - (HL) - CY

Aレジスタの内容からレジスタ・ペアHLでアドレスされるデータ・メモリの内容をキャリー・フラグを含めて減算し、結果をAレジスタにセットします。減算の結果ポローが出るとき、キャリー・フラグがセットされ、ポローが出ないときキャリー・フラグはリセットされます。

この命令の次にADDS A, #n4命令が置かれた場合、この命令にポローが出なければADDS A, #n4命令をスキップします。ポローが出るとADDS A, #n4命令を実行し、ADDS A, #n4命令のスキップ機能を禁止するという機能が生じます。したがって、これらの命令の組み合わせを進数補正に利用できます ( 10.1.4 進数補正命令を参照 ) 。

## ○ SUBC XA, rp'

**機能** : XA, CY    XA - rp' - CY

レジスタ・ペアXAの内容からレジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をキャリー・フラグを含めて減算し、結果をレジスタ・ペアXAにセットします。減算の結果ポローが出るとキャリー・フラグがセットされ、ポローが出ないときはキャリー・フラグはリセットされます。

## ○ SUBC rp'1, XA

機能: rp'1, CY rp'1 - XA - CY

レジスタ・ペアrp'1 (HL, DE, BC, XA', HL', DE', BC') の内容からレジスタ・ペアXAの内容をキャリー・フラグを含めて減算し、結果を指定されたレジスタ・ペアrp'1にセットされます。減算の結果ポローが出るとキャリー・フラグがセットされ、ポローが出ないときはキャリー・フラグはリセットされます。

## ○ AND A, #n4

機能: A A n4 n4 = l3-0 : 0-FH

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

## 応用例

アキュムレータの上位2ビットを0にする。

```
AND A, #0011B
```

## ○ AND A, @HL

機能: A A (HL)

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

## ○ AND XA, rp'

機能: XA XA rp'

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

## ○ AND rp'1, XA

機能: rp'1 rp'1 XA

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

## ○ OR A, #n4

機能: A A n4 n4 = l3-0 : 0-FH

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

## 応用例

アキュムレータの下位3ビットを1にする。

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

## ○ OR A, @HL

機能: A A (HL)

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

## ○ OR XA, rp'

機能: XA XA rp'

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



## ○ OR rp'1, XA

機能 : rp'1 rp'1 XA

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

## ○ XOR A, #n4

機能 : A A  $\forall$  n4 n4 = l3-0 : 0-FH

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

## 応用例

アキュムレータの最上位4ビットを反転する。

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

## ○ XOR A, @HL

機能 : A A  $\forall$  (HL)

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

## ○ XOR XA, rp'

機能 : XA XA  $\forall$  rp'

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

 XOR rp'1, XA

**機能** : rp'1 ← rp'1 ∨ XA

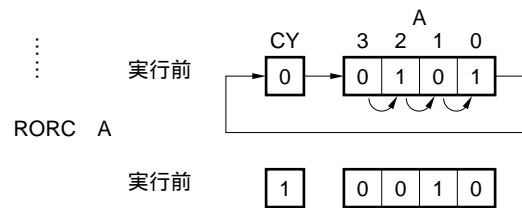
レジスタ・ペアrp'1 ( HL, DE, BC, XA', HL', DE', BC' ) の内容とレジスタ・ペアXAの内容との排他的論理和をとり，結果をレジスタ・ペアrp'1にセットします。

## 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・フラグを含めて1ビットずつ右回転します。



## ○ NOT A

機能: A  $\bar{A}$

Aレジスタ(4ビット・アキュムレータ)の1の補数をとります(各ビットを反転させます)。

## 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 = FH

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

## ○ DECS rp'

機能 : rp'    rp' - 1 ; Skip if rp' = FFH

レジスタ・ペアrp' ( XA, HL, DE, BC, XA', HL', DE', BC' ) の内容をデクリメントします。デクリメントの結果rp' = FFHになると、続く1命令をスキップします。

## 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 ← CY  $\nabla$  (オペランドで指定されるビット)

キャリー・フラグの内容と、ビット操作アドレッシング ( fmem.bit, pmem.@L, @H + mem.bit ) で指定されるデータ・メモリのビットの内容との排他的論理和をとり、結果をキャリー・フラグにセットします。

## 10.4.10 分岐命令

## ○ BR addr

機能：PC<sub>10-0</sub> addr  
addr = 0000H-07FFH

イミューディエト・データaddrでアドレスされるアドレスに分岐します。

この命令はアセンブラの疑似命令であり、アセンブル時、アセンブラよりBR !addr命令、BR CB !caddr命令、BR \$addr命令の中から最適な命令に自動的に置き換えられます。

## ○ BR addr1

機能：PC<sub>10-0</sub> addr1  
addr1 = 0000H-07FFH

イミューディエト・データaddr1でアドレスされるアドレスに分岐します。

この命令はアセンブラの疑似命令であり、アセンブル時、アセンブラよりBRA !addr1命令、BR !addr命令、BR CB !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>10-0</sub> addr1

## ○ BR !addr

機能：PC<sub>10-0</sub> addr  
addr = 0000H-07FFH

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

## ○ BR \$addr

機能：PC<sub>10-0</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>10-0</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>10-0</sub> caddr<sub>10-0</sub>

caddr = 0000H-07FFH

プログラム・カウンタ (PC<sub>10-0</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07FEH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>10-0</sub> PC<sub>10-8</sub> + DE  
PC<sub>7-4</sub> D, PC<sub>3-0</sub> E

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

### 注 意

BR PCDE命令は、通常、その命令のあるページ内に分岐しますが、命令コードの1バイト目がx xFE番地、またはx xFFH番地にある場合は、そのページ内には分岐せず次のページに分岐します。



たとえば、上図のように a または b の位置に BR PCDE 命令がある場合は、ページ2ではなくページ3の、レジスタ・ペアDEの内容により指定される下位8ビット・アドレスへ分岐します。

## ○ BR PCXA

機能：PC<sub>10-0</sub> PC<sub>10-8</sub> + XA  
PC<sub>7-4</sub> X, PC<sub>3-0</sub> A

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

### 注 意

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

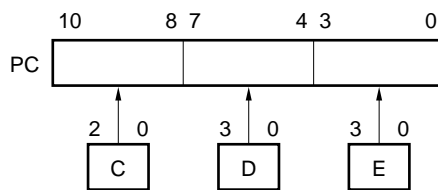
## ○ BR BCDE

機能：PC<sub>10-0</sub> BCDE

## 例

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、μPD754202のPCは11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。



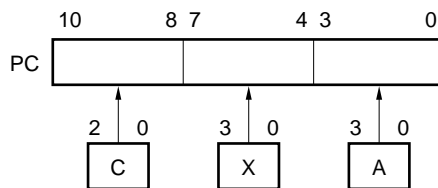
## ○ BR BCXA

機能：PC<sub>10-0</sub> BCXA

## 例

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、μPD754202のPCは11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 (EEU-730)**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) 0, PC<sub>10-8</sub>  
 PC<sub>10-0</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) 0, PC<sub>10-8</sub>, PC<sub>10-0</sub> addr, SP SP - 4

addr = 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) 0, PC<sub>10-8</sub>  
 PC<sub>10-0</sub> addr, SP SP - 6

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・イミーディエト・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) 0, PC<sub>10-8</sub>, SP SP - 4

PC<sub>10-0</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) 0, PC<sub>10-8</sub>

SP SP - 6

PC<sub>10-0</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・イミディエイト・データ faddr でアドレスされる番地へ分岐します。

## ○ TCALL !addr

機能：

GETI 命令のテーブル定義用アセンブラ疑似命令です。3 バイトの CALL !addr 命令を, 1 バイトの GETI 命令で置き換える場合に使用します。addr には 12 ビット・アドレス・データを記述します。詳しくは, RA75X **アセンブラ・パッケージ ユーザーズ・マニュアル 言語編** (EEU-730) を参照してください。

## ① RET

**機能:** 【Mk モード】 PC<sub>10-8</sub> (SP)<sub>2-0</sub>, 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>10-8</sub> (SP)<sub>2-0</sub>, 0, 0, 0, 0 (SP + 1)  
 PC<sub>3-0</sub> (SP + 2), PC<sub>7-4</sub> (SP + 3)  
 ×, ×, MBE, RBE (SP + 4), SP SP + 6

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

### 注意

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

## ① RETS

**機能:** 【Mk モード】 PC<sub>10-8</sub> (SP)<sub>2-0</sub>, 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>10-8</sub> (SP)<sub>2-0</sub>, 0, 0, 0, 0 (SP + 1)  
 PC<sub>3-0</sub> (SP + 2), PC<sub>7-4</sub> (SP + 3)  
 ×, ×, 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。

## ① RETI

**機能：【Mk モード】** PC<sub>10-8</sub> (SP)<sub>2-0</sub>, 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>10-8</sub> (SP)<sub>2-0</sub>, 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, T1, T2, 0, 2 )

○ 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, T1, T2, 0, 2 )

## 10.4.13 入出力命令

 IN A, PORTn

機能 : A PORTn n = N<sub>3-0</sub> : 3, 6, 7, 8

PORTn (n = 3, 6, 7, 8) で指定されるポートの内容をAレジスタに転送します。

**注 意**

この命令実行時には、MBE = 0とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。nには3, 6, 7, 8のみ指定できます。

入出力モードの指定により、出力ラッチのデータ (出力モード) または、端子のデータ (入力モード) が取り込まれます。

 OUT PORTn, A

機能 : PORTn A n = N<sub>3-0</sub> : 3, 6, 8

Aレジスタの内容をPORTn (n = 3, 6, 8) で指定されるポートの出力ラッチへ転送します。

**注 意**

この命令実行時には、MBE = 0とするか、もしくは (MBE = 1, MBS = 15) としておく必要があります。

nには3, 6, 8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<sub>1-0</sub>: 0-3

2ビット・イミディエト・データ n をレジスタ・バンク選択レジスタ (RBS) にセットします。

## ○ SEL MBn

機能: MBS n n = N<sub>3-0</sub>: 0, 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<sub>5-0</sub>, 0 : 20H-7FH

【Mk モード】

TBR命令で定義したテーブルを参照した場合

PC<sub>10-0</sub> (taddr)<sub>2-0</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) 0, PC<sub>10-8</sub>PC<sub>10-0</sub> (taddr)<sub>2-0</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

PC<sub>10-0</sub> (taddr)<sub>2-0</sub> + (taddr + 1)

## TCALL命令で定義したテーブルを参照した場合

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

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

(SP - 5) 0, 0, 0, 0, (SP - 6) 0, PC<sub>10-8</sub>

PC<sub>10-0</sub> (taddr)<sub>2-0</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

(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	{ INCS L DECS L INCS H DECS H INCS HL
MOV @HL, A	
XCH A, @HL	
MOV A, @DE	{ INCS E DECS E INCS D DECS D INCS DE
XCH A, @DE	
MOV A, @DL	{ INCS L DECS L INCS D DECS D
XCH A, @DL	

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

GETI命令の前の命令がスキップ機能を持つ場合、GETI命令は他の1バイト命令と同様にスキップされます。また、GETI命令で参照した命令がスキップ機能を持つ場合、GETI命令の次に続く命令はスキップされます。

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

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

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

## 応用例

```

{
MOV  HL, #00H
MOV  XA, #FFH
CALL SUB1
BR   SUB2
} をGETI命令に置き換える。

      ORG    20H
HL00: MOV    HL, #00H
XAFF: MOV    XA, #FFH
CSUB1: TCALL SUB1
BSUB2: TBR   SUB2
      ⋮
      GETI   HL00      ; MOV  HL, #00H
      ⋮
      GETI   BSUB2     ; BR   SUB2
      ⋮
      GETI   CSUB1     ; CALL SUB1
      ⋮
      GETI   XAFF      ; MOV  XA, #FFH
```

〔メ モ〕

★ 付録A  $\mu$ PD754202, 75F4264の機能一覧表

項目		$\mu$ PD754202	$\mu$ PD75F4264 <sup>注</sup>
プログラム・メモリ		マスクROM 0000H-07FFH (2048×8ビット)	フラッシュ・メモリ 0000H-0FFFH (4096×8ビット)
データ・メモリ	スタティックRAM	000H-07FH (128×4ビット)	
	EEPROM <sup>TM</sup>	なし	400H-43FH (32×8ビット)
CPU		75XL CPU	
汎用レジスタ		(4ビット×8または8ビット×4)×4バンク	
命令実行時間		<ul style="list-style-type: none"> <li>・0.67, 1.33, 2.67, 10.7 <math>\mu</math>s (fx = 6.0 MHz動作時)</li> <li>・0.95, 1.91, 3.81, 15.3 <math>\mu</math>s (fx = 4.19 MHz動作時)</li> </ul>	
入出力ポート	CMOS入力	4本(マスク・オプションによるプルアップ抵抗内蔵可能)	
	CMOS入出力	9本(ソフトウェアによる内蔵プルアップ抵抗の接続を指定可能)	
	合計	13本	
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	
リセット後起動時間		$2^{17}/f_x, 2^{15}/f_x$ (マスク・オプションで選択)	$2^{15}/f_x$
タイマ		4チャンネル <ul style="list-style-type: none"> <li>・8ビット・タイマ・カウンタ: 3チャンネル (16ビット・タイマ・カウンタとして使用可能)</li> <li>・ベーシック・インターバル・タイマ/ウォッチドッグ・タイマ: 1チャンネル</li> </ul>	
A/Dコンバータ		なし	<ul style="list-style-type: none"> <li>・8ビット分解能×2チャンネル(逐次比較型, ハードウェア制御)</li> <li>・V<sub>DD</sub> = 1.8Vから動作可能</li> </ul>
プログラマブル・スレッシュホールド・ポート		なし	2チャンネル
ベクタ割り込み		外部: 1本, 内部: 4本	外部: 1本, 内部: 5本
テスト入力		外部: 1本(キー・リターン・リセット機能あり)	
電源電圧		V <sub>DD</sub> = 1.8 ~ 6.0V	
動作周囲温度		T <sub>A</sub> = -40 ~ +85	
パッケージ		<ul style="list-style-type: none"> <li>・20ピン・プラスチックSOP (300 mil, 1.27 mmピッチ)</li> <li>・20ピン・プラスチック・シュリンクSOP (300 mil, 0.65 mmピッチ)</li> </ul>	<ul style="list-style-type: none"> <li>・20ピン・プラスチックSOP (300 mil, 1.27 mmピッチ)</li> </ul>

注 開発中

付A

〔メ モ〕

## 付録 B 開発ツール

μ PD754202を使用するシステム開発のために次のような開発ツールを用意しております。75XLシリーズでは、シリーズ共通のリロケータブル・アセンブラを品種ごとのデバイス・ファイルと組み合わせて使用します。

### 言語プロセッサ

RA75X リロケータブル・ アセンブラ	ホスト・マシン			オーダ名称(品名)	
		OS	供給媒体		
	PC-9800シリーズ	MS-DOS™	3.5インチ2HD	μ S5A13RA75X	
		<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Ver.3.30</td></tr> <tr><td>}</td></tr> <tr><td>Ver.6.2<sup>注</sup></td></tr> </table>	Ver.3.30	}	Ver.6.2 <sup>注</sup>
Ver.3.30					
}					
Ver.6.2 <sup>注</sup>					
	IBM PC/AT™ およびその互換機	「IBM PC用のOSにつ いて」参照	3.5インチ2HC	μ S7B13RA75X	
			5インチ2HC	μ S7B10RA75X	

★ デバイス・ファイル	ホスト・マシン			オーダ名称(品名)	
		OS	供給媒体		
	PC-9800シリーズ	MS-DOS	3.5インチ2HD	μ S5A13DF754202	
		<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Ver. 3.30</td></tr> <tr><td>}</td></tr> <tr><td>Ver.6.2<sup>注</sup></td></tr> </table>	Ver. 3.30	}	Ver.6.2 <sup>注</sup>
Ver. 3.30					
}					
Ver.6.2 <sup>注</sup>					
	IBM PC/AT およびその互換機	「IBM PC用のOSにつ いて」参照	3.5インチ2HC	μ S7B13DF754202	
			5インチ2HC	μ S7B10DF754202	

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

備考 アセンブラ、デバイス・ファイルの動作は、上記のホスト・マシンとOS上でのみ保証されます。



ディバグ用ツール

μ PD754202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 PD754202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-754144GS-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 PD754202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-754144GS-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。			
	IE-75300-R-EM	μ PD754202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-754144GS-R	μ PD754202用エミュレーション・プローブです。			
	EV-9500GS-20 EV-9501GS-20	IE-75000-RまたはIE-75001-Rと、IE-75300-R-EMに接続して使用します。 ターゲット・システムとの接続を容易にするフレキシブル基板EV-9500GS-20 (20ピン・プラスチック・シュリンクSOPに対応)、およびEV-9501GS-20 (20ピン・プラスチックSOPに対応) を添付しています。			
ソフトウェア	IEコントロール・プログラム	IE-75000-RまたはIE-75001-Rとホスト・マシンをRS-232-C、およびセントロニクスI/Fで接続し、ホスト・マシン上でIE-75000-RまたはIE-75001-Rを制御します。			
		ホスト・マシン	オーダ名称 (品名)		
			OS	供給媒体	
		PC-9800シリーズ	MS-DOS [ Ver.3.30 ] } [ Ver.6.2 <sup>注2</sup> ]	3.5インチ2HD 5インチ2HD	μ S5A13IE75X μ S5A10IE75X
		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 5インチ2HC	μ S7B13IE75X μ S7B10IE75X

注1 . 保守品です。

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

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

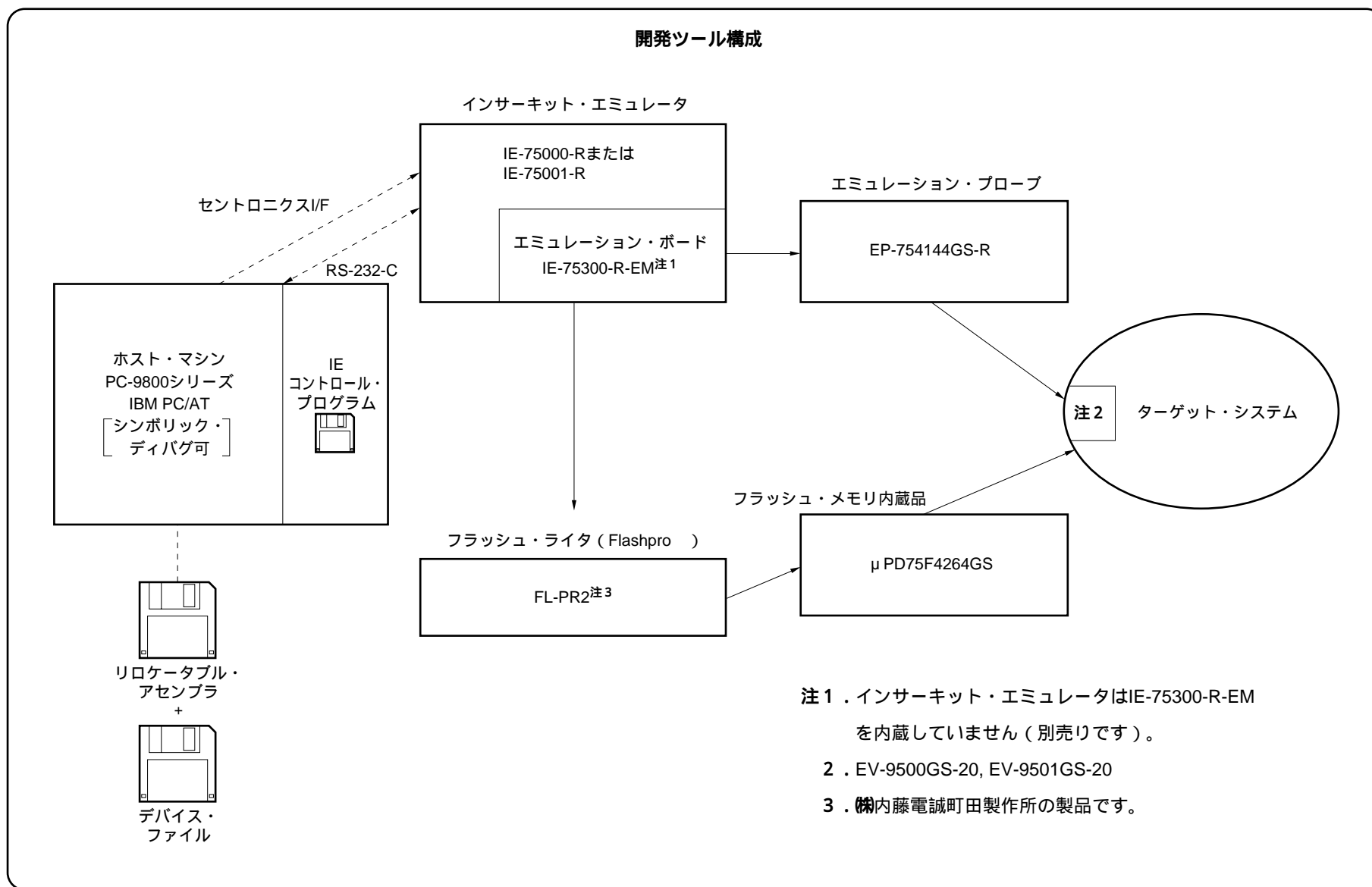
## IBM PC用のOSについて

IBM PC用のOSとして、次のものがサポートされています。

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

注 英語モードのみサポートしています。

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



## 付録C マスクROM発注の手順

プログラム開発が完了して、マスクROMを発注する場合には、次のような手順となります。

### マスクROM発注の予約

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

### 発注用媒体の作成

マスクROM発注用の媒体には、次の3つがあります。

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

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

### 必要書類の作成

マスクROM発注にあたって、下記の書類に記入してください。

- ・ マスク式ROM発注書
- ・ マスク式ROM発注チェック・シート
- ・ マスク・オプション情報書

### 発 注

で作成した媒体と で記入した書類をまとめて、発注予約日までに特約店あるいは販売部門に提出していただきます。

注意 詳しくはインフォメーション資料「ROMコードの発注方法」（資料番号 C10302J）をご覧ください。

〔メ モ〕

## 付録D 命令索引

### D.1 命令索引 (機能別)

#### [ 転送命令 ]

MOV A, #n4 ... 212, 225  
MOV reg1, #n4 ... 212, 225  
MOV XA, #n8 ... 212, 225  
MOV HL, #n8 ... 212, 226  
MOV rp2, #n8 ... 212, 226  
MOV A, @HL ... 212, 226  
MOV A, @HL + ... 212, 226  
MOV A, @HL - ... 212, 226  
MOV A, @rpa1 ... 212, 227  
MOV XA, @HL ... 212, 227  
MOV @HL, A ... 212, 227  
MOV @HL, XA ... 212, 228  
MOV A, mem ... 212, 228  
MOV XA, mem ... 212, 228  
MOV mem, A ... 212, 228  
MOV mem, XA ... 212, 229  
MOV A, reg ... 212, 229  
MOV XA, rp' ... 212, 229  
MOV reg1, A ... 212, 229  
MOV rp'1, XA ... 212, 229  
XCH A, @HL ... 212, 230  
XCH A, @HL + ... 212, 230  
XCH A, @HL - ... 212, 230  
XCH A, @rpa1 ... 212, 231  
XCH XA, @HL ... 212, 231  
XCH A, mem ... 212, 231  
XCH XA, mem ... 212, 232  
XCH A, reg1 ... 212, 232  
XCH XA, rp' ... 212, 232

#### [ テーブル参照命令 ]

MOVT XA, @PCDE ... 213, 233  
MOVT XA, @PCXA ... 213, 235  
MOVT XA, @BCDE ... 213, 235  
MOVT XA, @BCXA ... 213, 236

#### [ ビット転送命令 ]

MOV1 CY, fmem.bit ... 213, 237  
MOV1 CY, pmem.@L ... 213, 237  
MOV1 CY, @H + mem.bit ... 213, 237  
MOV1 fmem.bit, CY ... 213, 237  
MOV1 pmem.@L, CY ... 213, 237  
MOV1 @H + mem.bit, CY ... 213, 237

#### [ 演算命令 ]

ADDS A, #n4 ... 213, 238  
ADDS XA, #n8 ... 213, 238  
ADDS A, @HL ... 213, 238  
ADDS XA, rp' ... 213, 238  
ADDS rp'1, XA ... 213, 239  
ADDC A, @HL ... 213, 239  
ADDC XA, rp' ... 213, 239  
ADDC rp'1, XA ... 213, 240  
SUBS A, @HL ... 213, 240  
SUBS XA, rp' ... 213, 240  
SUBS rp'1, XA ... 213, 241  
SUBC A, @HL ... 213, 241  
SUBC XA, rp' ... 213, 241  
SUBC rp'1, XA ... 213, 242  
AND A, #n4 ... 213, 242

AND A, @HL ... 213, 242  
 AND XA, rp' ... 213, 242  
 AND rp'1, XA ... 213, 243  
 OR A, #n4 ... 213, 243  
 OR A, @HL ... 213, 243  
 OR XA, rp' ... 213, 243  
 OR rp'1, XA ... 213, 244  
 XOR A, #n4 ... 213, 244  
 XOR A, @HL ... 213, 244  
 XOR XA, rp' ... 213, 244  
 XOR rp'1, XA ... 213, 245

**[ アキュムレータ操作命令 ]**

RORC A ... 214, 246  
 NOT A ... 214, 246

**[ 増減命令 ]**

INCS reg ... 214, 247  
 INCS rp1 ... 214, 247  
 INCS @HL ... 214, 247  
 INCS mem ... 214, 247  
 DECS reg ... 214, 248  
 DECS rp' ... 214, 248

**[ 比較命令 ]**

SKE reg, #n4 ... 214, 249  
 SKE @HL, #n4 ... 214, 249  
 SKE A, @HL ... 214, 249  
 SKE XA, @HL ... 214, 249  
 SKE A, reg ... 214, 250  
 SKE XA, rp' ... 214, 250

**[ キャリー・フラグ操作命令 ]**

SET1 CY ... 214, 251  
 CLR1 CY ... 214, 251  
 SKT CY ... 214, 251  
 NOT1 CY ... 214, 251

**[ メモリ・ビット操作命令 ]**

SET1 mem.bit ... 214, 252  
 SET1 fmem.bit ... 214, 252  
 SET1 pmem.@L ... 214, 252  
 SET1 @H+ mem.bit ... 214, 252  
 CLR1 mem.bit ... 214, 252  
 CLR1 fmem.bit ... 214, 252  
 CLR1 pmem.@L ... 214, 252  
 CLR1 @H+ mem.bit ... 214, 252  
 SKT mem.bit ... 214, 253  
 SKT fmem.bit ... 214, 253  
 SKT pmem.@L ... 214, 253  
 SKT @H+ mem.bit ... 214, 253  
 SKF mem.bit ... 214, 253  
 SKF fmem.bit ... 214, 253  
 SKF pmem.@L ... 214, 253  
 SKF @H+ mem.bit ... 214, 253  
 SKTCLR fmem.bit ... 214, 254  
 SKTCLR pmem.@L ... 214, 254  
 SKTCLR @H+ mem.bit ... 214, 254  
 AND1 CY, fmem.bit ... 215, 254  
 AND1 CY, pmem.@L ... 215, 254  
 AND1 CY, @H+ mem.bit ... 215, 254  
 OR1 CY, fmem. bit ... 215, 254  
 OR1 CY, pmem.@L ... 215, 254  
 OR1 CY, @H+ mem.bit ... 215, 254  
 XOR1 CY, fmem.bit ... 215, 255  
 XOR1 CY, pmem.@L ... 215, 255  
 XOR1 CY, @H+ mem.bit ... 215, 255

**[ 分岐命令 ]**

BR addr ... 215, 256  
 BR addr1 ... 215, 256  
 BR !addr ... 215, 256  
 BR \$addr ... 215, 256  
 BR \$addr1 ... 215, 257  
 BR PCDE ... 215, 258

BR PCXA ... 215, 258  
BR BCDE ... 215, 259  
BR BCXA ... 215, 259  
BRA !addr1 ... 215, 256  
BRCB !caddr ... 215, 257  
TBR addr ... 215, 259  
SEL MBn ... 217, 268  
GETI taddr ... 217, 268

**[ サブルーチン・スタック制御命令 ]**

CALLA !addr1 ... 216, 260  
CALL !addr ... 216, 260  
CALLF !faddr ... 216, 261  
TCALL !addr ... 217, 261  
RET ... 216, 262  
RETS ... 216, 262  
RETI ... 216, 263  
PUSH rp ... 217, 264  
PUSH BS ... 217, 264  
POP rp ... 217, 264  
POP BS ... 217, 264

**[ 割り込み制御命令 ]**

EI ... 217, 265  
EI IE x x x ... 217, 265  
DI ... 217, 265  
DI IE x x x ... 217, 265

**[ 入出力命令 ]**

IN A, PORTn ... 217, 266  
OUT PORTn, A ... 217, 266

**[ CPU制御命令 ]**

HALT ... 217, 267  
STOP ... 217, 267  
NOP ... 217, 267

**[ 特殊命令 ]**

SEL RBn ... 217, 268



## D.2 命令索引 (アルファベット順)

## [ A ]

ADDC A, @HL ... 213, 239  
 ADDC rp'1, XA ... 213, 240  
 ADDC XA, rp' ... 213, 239  
 ADDS A, #n4 ... 213, 238  
 ADDS A, @HL ... 213, 238  
 ADDS rp'1, XA ... 213, 239  
 ADDS XA, rp' ... 213, 238  
 ADDS XA, #n8 ... 213, 238  
 AND A, #n4 ... 213, 242  
 AND A, @HL ... 213, 242  
 AND rp'1, XA ... 213, 243  
 AND XA, rp' ... 213, 242  
 AND1 CY, fmem.bit ... 215, 254  
 AND1 CY, pmem.@L ... 215, 254  
 AND1 CY, @H+ mem.bit ... 215, 254

## [ B ]

BR addr ... 215, 256  
 BR addr1 ... 215, 256  
 BR BCDE ... 215, 259  
 BR BCXA ... 215, 259  
 BR PCDE ... 215, 258  
 BR PCXA ... 215, 258  
 BR !addr ... 215, 256  
 BR \$addr ... 215, 256  
 BR \$addr1 ... 215, 257  
 BRA !addr1 ... 215, 256  
 BRCB !caddr ... 215, 257

## [ C ]

CALL !addr ... 216, 260  
 CALLA !addr1 ... 216, 260  
 CALLF !faddr ... 216, 261  
 CLR1 CY ... 214, 251

CLR1 fmem.bit ... 214, 252  
 CLR1 mem.bit ... 214, 252  
 CLR1 pmem.@L ... 214, 252  
 CLR1 @H+ mem.bit ... 214, 252

## [ D ]

DECS reg ... 214, 248  
 DECS rp' ... 214, 248  
 DI ... 217, 265  
 DI IE x x x ... 217, 265

## [ E ]

EI ... 217, 265  
 EI IE x x x ... 217, 265

## [ G ]

GETI taddr ... 217, 268

## [ H ]

HALT ... 217, 267

## [ I ]

IN A, PORTn ... 217, 266  
 INCS mem ... 214, 247  
 INCS reg ... 214, 247  
 INCS rp1 ... 214, 247  
 INCS @HL ... 214, 247

## [ M ]

MOV A, mem ... 212, 228  
 MOV A, reg ... 212, 229  
 MOV A, #n4 ... 212, 225  
 MOV A, @HL ... 212, 226  
 MOV A, @HL+ ... 212, 226  
 MOV A, @HL- ... 212, 226



SKT @H+mem.bit ... 214, 253  
SKTCLR fmem.bit ... 214, 254  
SKTCLR pmem.@L ... 214, 254  
SKTCLR @H+mem.bit ... 214, 254  
STOP ... 217, 267  
SUBC A, @HL ... 213, 241  
SUBC rp'1, XA ... 213, 242  
SUBC XA, rp' ... 213, 241  
SUBS A, @HL ... 213, 240  
SUBS rp'1, XA ... 213, 241  
SUBS XA, rp' ... 213, 240

**[ T ]**

TBR addr ... 217, 259  
TCALL !addr ... 217, 261

**[ X ]**

XCH A, mem ... 212, 231  
XCH A, reg1 ... 212, 232  
XCH A, @HL ... 212, 230  
XCH A, @HL+ ... 212, 230  
XCH A, @HL- ... 212, 230  
XCH A, @rpa1 ... 212, 231  
XCH XA, mem ... 212, 232  
XCH XA, rp' ... 212, 232  
XCH XA, @HL ... 212, 231  
XOR A, #n4 ... 213, 244  
XOR A, @HL ... 213, 244  
XOR rp'1, XA ... 213, 245  
XOR XA, rp' ... 213, 244  
XOR1 CY, fmem.bit ... 215, 255  
XOR1 CY, pmem.@L ... 215, 255  
XOR1 CY, @H+mem.bit ... 215, 255

## 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
INTEレジスタ (INTE)	...	37
INTFレジスタ (INTF)	...	37
INTGレジスタ (INTG)	...	37
INTHレジスタ (INTH)	...	37
INT0エッジ検出モード・レジスタ (IM0)	...	162
INT0割り込み許可フラグ (IE0)	...	157
INT0割り込み要求フラグ (IRQ0)	...	157
INT2エッジ検出モード・レジスタ (IM2)	...	181
INT2割り込み許可フラグ (IE2)	...	179
INT2割り込み要求フラグ (IRQ2)	...	179
ウォッチドッグ・タイマ許可フラグ (WDTM)	...	89
ウォッチドッグ・フラグ (WDF)	...	201

#### [か行]

キー・リターン・リセット (KRREN)	...	181
キー・リターン・フラグ (KRF)	...	201
キー割り込み4-7 (KR4-KR7)	...	179
キャリー・フラグ (CY)	...	57

#### [さ行]

スキップ・フラグ0-2 (SK0-SK2)	...	58
スタック・バンク選択レジスタ (SBS)	...	44, 53
スタック・ポインタ (SP)	...	53

#### [た行]

タイマ・カウンタ・コントロール・レジスタ (TC2)	...	106
タイマ・カウンタ0カウント・レジスタ (T0)	...	36
タイマ・カウンタ0出力許可フラグ (TOE0)	...	105
タイマ・カウンタ0モード・レジスタ (TM0)	...	100

タイマ・カウンタ0 モジュロ・レジスタ (TMOD0) ...	36
タイマ・カウンタ0 割り込み許可フラグ (IET0) ...	157
タイマ・カウンタ0 割り込み要求フラグ (IRQT0) ...	157
タイマ・カウンタ1 カウント・レジスタ (T1) ...	36
タイマ・カウンタ1 出力許可フラグ (TOE1) ...	105
タイマ・カウンタ1 モード・レジスタ (TM1) ...	101
タイマ・カウンタ1 モジュロ・レジスタ (TMOD1) ...	36
タイマ・カウンタ1 割り込み許可フラグ (IET1) ...	157
タイマ・カウンタ1 割り込み要求フラグ (IRQT1) ...	157
タイマ・カウンタ2 カウント・レジスタ (T2) ...	35
タイマ・カウンタ2 出力許可フラグ (TOE2) ...	106
タイマ・カウンタ2 ハイ・レベル期間設定用モジュロ・レジスタ (TMOD2H) ...	34
タイマ・カウンタ2 モード・レジスタ (TM2) ...	103
タイマ・カウンタ2 モジュロ・レジスタ (TMOD2) ...	35
タイマ・カウンタ2 割り込み許可フラグ (IET2) ...	157
タイマ・カウンタ2 割り込み要求フラグ (IRQT2) ...	157
テスト許可フラグ (IE2) ...	179
テスト要求フラグ (IRQ2) ...	179

**[ な行 ]**

ノー・リターン・ゼロ・バッファ・フラグ (NRZB) ...	106
ノー・リターン・ゼロ・フラグ (NRZ) ...	106

**[ は行 ]**

バンク選択レジスタ (BS) ...	61
BT割り込み許可フラグ (IEBT) ...	157
BT割り込み要求フラグ (IRQBT) ...	157
ビット・シーケンシャル・バッファ0-3 (BSB0-BSB3) ...	151
プルアップ抵抗指定レジスタ・グループA, B (POGA, POGB) ...	74
プログラム・カウンタ (PC) ...	45
プログラム・ステータス・ワード (PSW) ...	57
プロセッサ・クロック・コントロール・レジスタ (PCC) ...	79
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, 6, 7, 8) ...	63
ポート・モード・レジスタ・グループA, C (PMGA, PMGC) ...	69

**[ま行]**

- メモリ・バンク許可フラグ (MBE) ... 59
- メモリ・バンク選択レジスタ (MBS) ... 61

**[ら行]**

- レジスタ・バンク許可フラグ (RBE) ... 60
- レジスタ・バンク選択レジスタ (RBS) ... 62
- リモコン出力制御フラグ (REMC) ... 106

**[わ行]**

- 割り込みステータス・フラグ 0, 1 (IST0, 1) ... 59, 163
- 割り込みプライオリティ選択レジスタ (IPS) ... 158
- 割り込みマスタ許可フラグ (IME) ... 159

## E.2 ハードウェア索引 (アルファベット順)

### [ B ]

BS ... 61  
BSB0-BSB3 ... 151  
BT ... 87  
BTM ... 87

### [ C ]

CY ... 57

### [ I ]

IE0 ... 157  
IE2 ... 179  
IEBT ... 157  
IET0 ... 157  
IET1 ... 157  
IET2 ... 157  
IM0 ... 162  
IM2 ... 181  
IME ... 159  
INTA ... 37  
INTE ... 37  
INTF ... 37  
INTG ... 37  
INTH ... 37  
IPS ... 158  
IRQ0 ... 157  
IRQ2 ... 179  
IRQBT ... 157  
IRQT0 ... 157  
IRQT1 ... 157  
IRQT2 ... 157  
IST0, IST1 ... 59, 163

### [ K ]

KR4-KR7 ... 179  
KRF ... 201  
KRREN ... 181

### [ M ]

MBE ... 59  
MBS ... 61

### [ N ]

NRZ ... 106  
NRZB ... 106

### [ P ]

PC ... 45  
PCC ... 79  
PMGA, PMGC ... 69  
POGA, POGB ... 74  
PORT3, 6, 7, 8 ... 63  
PSW ... 57

### [ R ]

RBE ... 60  
RBS ... 62  
REMC ... 106

### [ S ]

SBS ... 44, 53  
SK0-SK2 ... 58  
SP ... 53

### [ T ]

T0, T1, T2 ... 35, 36  
TC2 ... 106  
TM0, TM1, TM2 ... 99  
TMOD0, TMOD1, TMOD2 ... 35, 36  
TMOD2H ... 34  
TOE0, TOE1, TOE2 ... 105, 106

### [ W ]

WDF ... 201  
WDTM ... 89

★

## 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	前版からの主な改版内容	適用箇所
第 2 版	μ PD754202, 75F4264の機能一覧表を追加	付録 A μ PD754202, 75F4264の機能一覧表
	デバイス・ファイル名を変更	付録 B 開発ツール



〔メ モ〕

## 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] μPD754202 ユーザーズ・マニュアル

(U11132JJ2V0UM00 (第2版))

[お名前など] (さしつかえのない範囲で)

御社名(学校名, その他) ( )  
ご住所 ( )  
お電話番号 ( )  
お仕事の内容 ( )  
お名前 ( )

1. ご評価(各欄に をご記入ください)

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

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

4. ご意見, ご要望

5. このドキュメントをお届けしたのは  
NEC販売員, 特約店販売員, NEC半導体ソリューション技術本部員,  
その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡ししてください。

NEC半導体インフォメーションセンター

FAX: (044) 548-7900

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3454-1111 (大代表)
中部支社 半導体第一販売部 半導体第二販売部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2170 名古屋 (052)222-2190
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3178 大阪 (06) 945-3200 大阪 (06) 945-3208
北海道支社 東北支社 岩手支店 山形支店 郡山支店 いわき支店 長岡支店 土浦支店 水戸支店 神奈川支社 群馬支店	札幌 (011)231-0161 仙台 (022)267-8740 盛岡 (019)651-4344 山形 (0236)23-5511 郡山 (0249)23-5511 いわき (0246)21-5511 長岡 (0258)36-2155 土浦 (0298)23-6161 水戸 (029)226-1717 横浜 (045)324-5524 高崎 (0273)26-1255	太田支店 (0276)46-4011 宇都宮支店 (028)621-2281 小山支店 (0285)24-5011 長野支社 (0263)35-1662 甲府支店 (0552)24-4141 埼玉支社 (048)641-1411 立川支社 (0425)26-5981 千葉支社 (043)238-8116 静岡支社 (054)255-2211 北陸支社 (0762)23-1621 福井支店 (0776)22-1866
富山支店 三重支店 京都支社 神戸支社 中国支社 鳥取支店 岡山支店 四国支社 新居浜支店 松山支店 九州支社	富山 (0764)31-8461 津 (0592)25-7341 京都 (075)344-7824 神戸 (078)333-3854 広島 (082)242-5504 鳥取 (0857)27-5311 岡山 (086)225-4455 高松 (0878)36-1200 新居浜 (0897)32-5001 松山 (089)945-4149 福岡 (092)271-7700	

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎 (044)548-7923	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお願い致します)
半導体販売技術本部 東日本販売技術部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3798-9619	
半導体販売技術本部 中部販売技術部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2125	
半導体販売技術本部 西日本販売技術部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3383	