

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。

PLL周波数シンセサイザ、イメージ・ディスプレイ・コントローラ内蔵
4ビット・シングルチップ・マイクロコントローラ

μ PD17062は、多様な表示が可能なイメージ・ディスプレイ・コントローラや、PLL周波数シンセサイザを1チップに納めたデジタル・チューニング・システム用4ビットCMOSマイクロコントローラです。

CPUは、4ビット並列加算、論理演算、複数のビット・テスト、キャリー・フラグのセット、リセット、強力な割り込み機能、およびタイマ機能を持っています。

オンスクリーン用にユーザ・プログラマブルなイメージ・ディスプレイ・コントローラ (IDC) を内蔵していますので、多様な表示が簡単なプログラムで制御できます。

また、強力な入出力命令で制御される豊富な I/O (Input/Output) ポートやシリアル・インタフェース機能、さらには4ビット A/D コンバータや D/A コンバータ用に6ビット PWM (Pulse Width Modulation) 出力を持っています。

特 徴

- デジタル・チューニング・システム用4ビット・マイクロコントローラ
- PLL周波数シンセサイザ内蔵
プリスケアラ: μ PB595 使用
- 5V \pm 10%
- CMOS低消費電力
- プログラム・メモリ (ROM) :
8Kバイト (16ビット \times 3968ステップ)
- データ・メモリ (RAM) : 4ビット \times 336ワード
- スタック・レベル: 6
- 35種類の分かりやすいインストラクション・セット
- 10進演算可能
- 命令実行時間: 2 μ s (8MHz発振子接続)
- D/Aコンバータ内蔵: 6ビット \times 4 (PWM出力)
- A/Dコンバータ内蔵: 4ビット \times 6
- 水平同期信号カウンタ内蔵
- 商用電源周波数カウンタ内蔵
- 停電検出回路およびパワーオン・リセット回路内蔵
- IDC (Image Display Controller) 内蔵 (ユーザ・プログラマブル)
表示文字数: 一画面最大99文字
表示位置: 14行 \times 19桁
文字の種類: 120種類
文字形式: 10 \times 15ドット (ふちどり可能)
色: 8色
文字サイズ: 縦横独立に4種類設定可能
1H縦揺れ防止回路内蔵
- 8ビット・シリアル・インタフェース内蔵
(1系統2チャンネル: 3線式および2線式)
- リモコン信号用割り込み入力 (ノイズ・キャンセラ付き)
- 豊富な I/O ポート
入出力ポート: 15
入力ポート: 4
出力ポート: 8

本資料の内容は、後日変更する場合があります。

オーダ情報

オーダ名称	パッケージ	品質水準
μPD17062CU-×××	48ピン・プラスチック・シュリンク DIP (600 mil)	標準 (一般電子機器用)
μPD17062GC-×××-3BE	64ピン・プラスチック QFP (□14 mm)	〃

備考 ×××は ROM コード番号です。

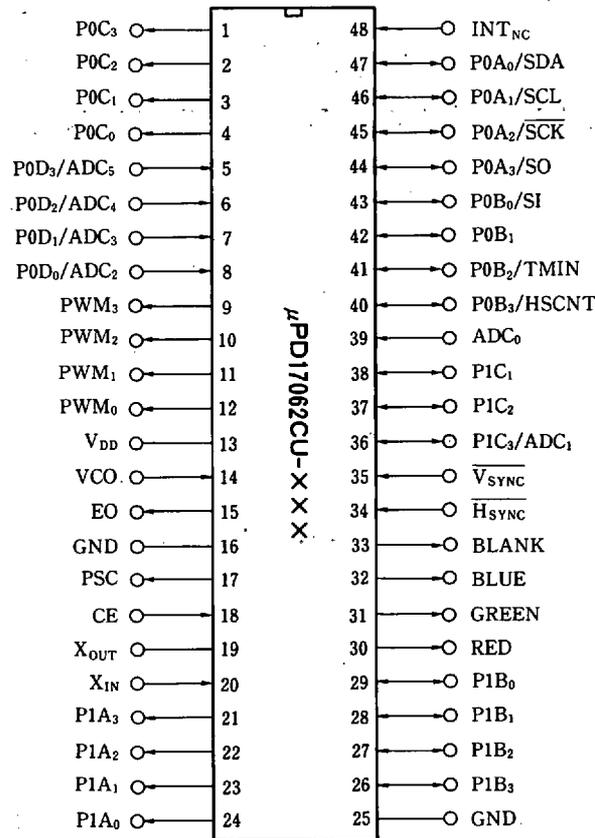
品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(IEI-620)をご覧ください。

機能一覧

項目	機能
ROM (プログラム・メモリ) 容量	3968×16ビット (マスク ROM)
CROM (キャラクタ ROM) 容量	1920×16ビット (ROM と兼用)
RAM (データ・メモリ) 容量	336×4 ビット (VRAM 兼用エリアあり)
VRAM (ビデオ RAM) 容量	224×4 ビット (RAM と兼用)
命令実行時間	2 μs (8 MHz 水晶振動子使用時)
スタック・レベル	6 レベル (スタック操作可)
入出力ポート数	入力ポート : 4 本
	出力ポート : 8 本
	入出力ポート : 15 本
IDC (Image Display Controller)	表示文字数 : 1 画面最大 99 文字
	表示形式 : 10×15 ドット 14 行×19 桁
	文字の種類 : 120 種類 (ユーザ・プログラマブル)
	色 : 8 色
	文字サイズ : 縦 1 倍-4 倍 (行ごとに設定可) : 横 1 倍-4 倍 (文字ごとに設定可)
シリアル・インタフェース	1 系統 { シリアル・インタフェース 0 (2 線式または I ² C バス対応) シリアル・インタフェース 1 (2 線式または 3 線式)
D/A コンバータ	6 ビット×4 本 (PWM 出力, 耐圧 12.5 V MAX.)
A/D コンバータ	4 ビット×6 本 (ソフトウェアによる逐次比較方式)
割り込み	4 チャンネル { 外部割り込み : 2 チャンネル 内部割り込み : 2 チャンネル
タイマ	1 チャンネル (内部クロック/ゼロクロス入力)
PLL 周波数シンセサイザ	分周方式 : パルス・スワロ方式 (VCO 端子 40 MHz MAX.) 専用 2 モジュラス・プリスケアラ外付け (μPB595 など)
	基準周波数 : 6.25, 12.5, 25 kHz
	チャージ・ポンプ : エラー・アウト出力
	位相比較器 : プログラムによりアンロック検出可能
リセット	パワーオン・リセット
	CE 端子によるリセット
	停電検出機能付き
電源電圧	5 V ±10 %

端子接続図 (Top View)

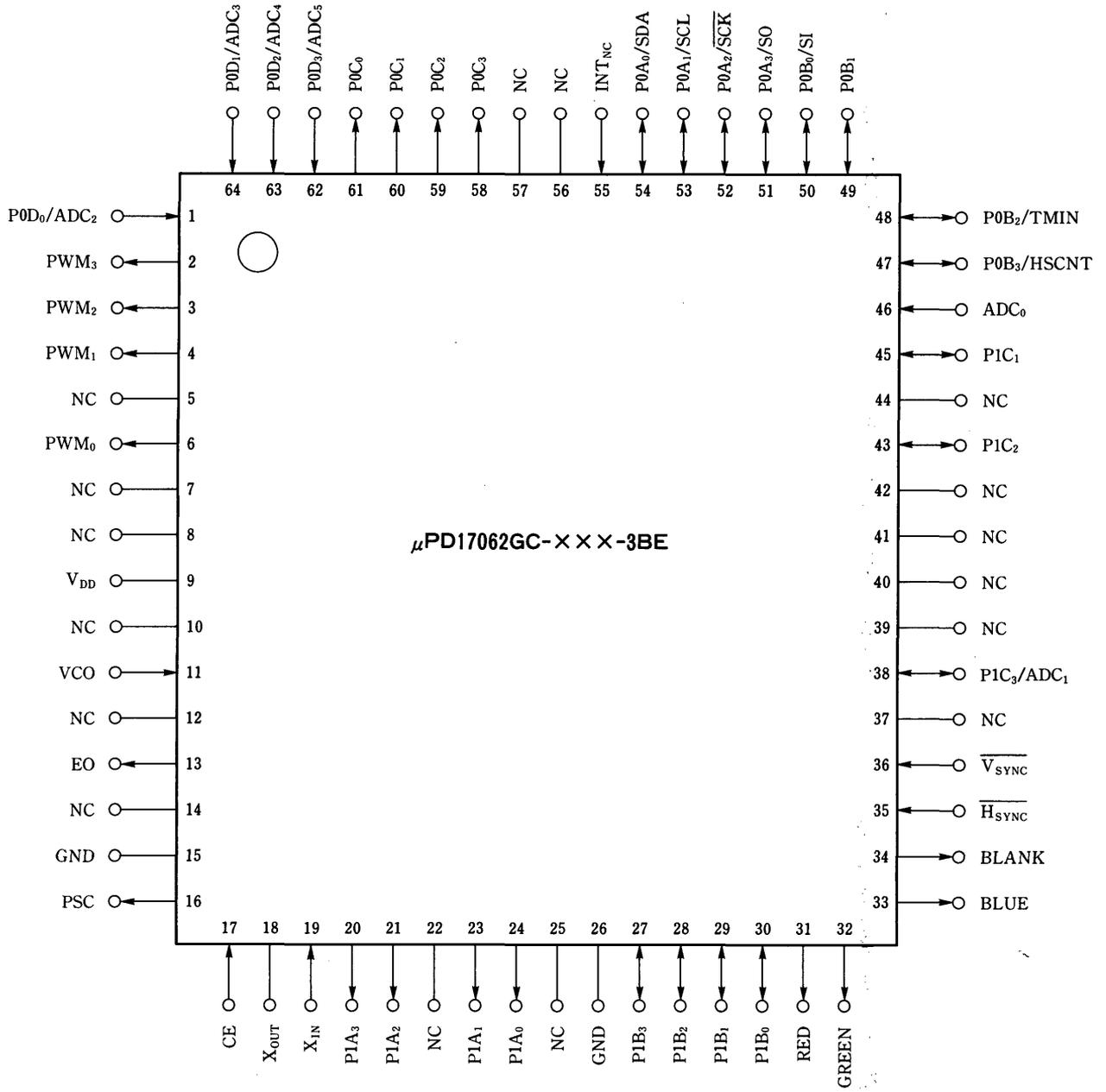
48ピン・プラスチック・シュリンク DIP (600 mil)



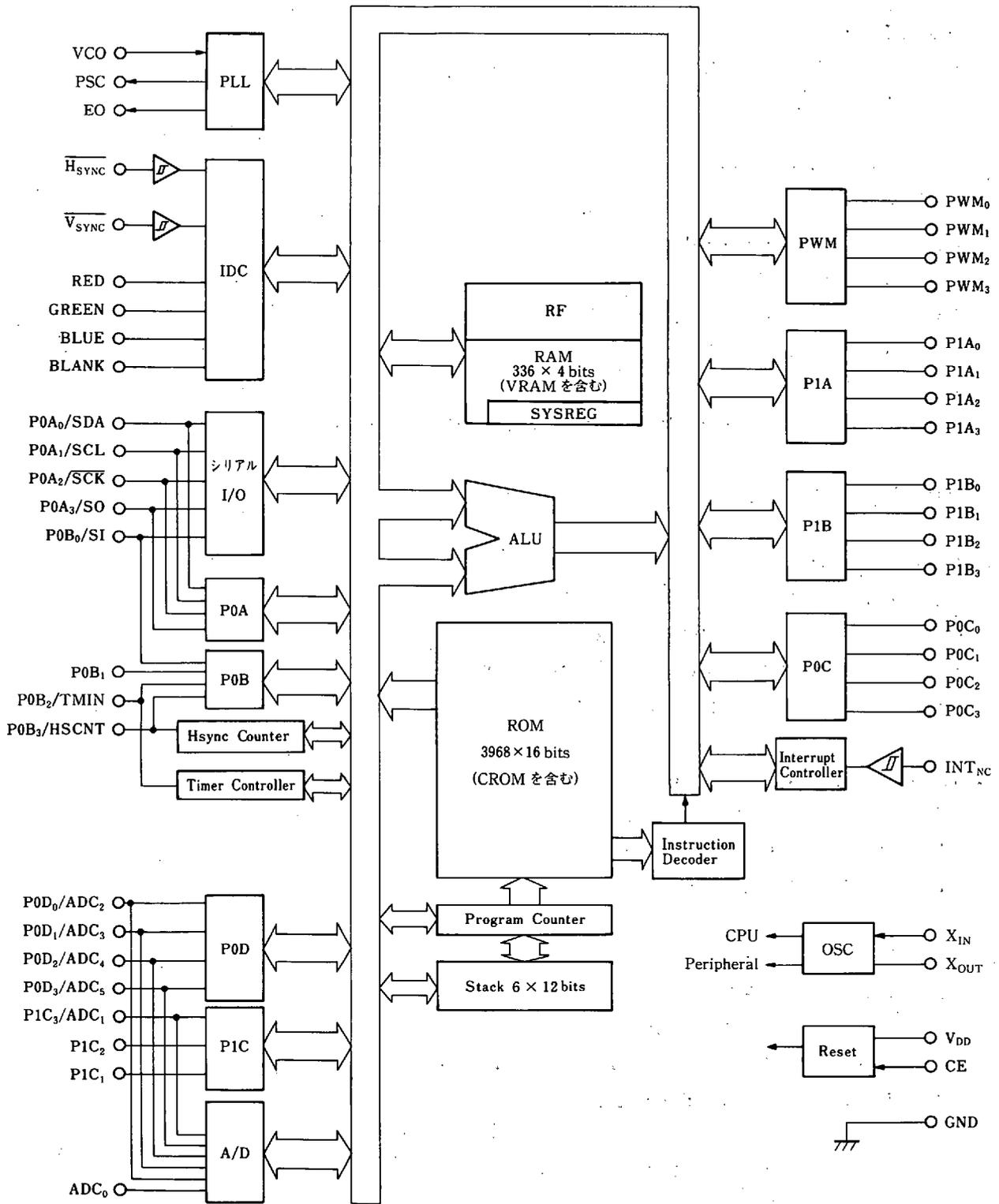
- ADC₀-ADC₅ : A/D コンバータ入力
- BLANK : ブランキング信号出力
- BLUE : 文字信号出力
- CE : チップ・イネーブル
- EO : エラー・アウト
- GND : グランド
- GREEN : 文字信号出力
- HSCNT : 水平同期信号カウンタ入力
- \overline{H}_{SYNC} : 水平同期信号入力
- INT_{NC} : 割り込み信号入力
- NC : ノー・コネクション
- PSC : パルス・スワロ・コントロール出力
- PWM₀-PWM₃ : パルス幅変調出力
- P0A₀-P0A₃ : ポート 0A
- P0B₀-P0B₃ : ポート 0B
- P0C₀-P0C₃ : ポート 0C

- P0D₀-P0D₃ : ポート 0D
- P1A₀-P1A₃ : ポート 1A
- P1B₀-P1B₃ : ポート 1B
- P1C₁-P1C₃ : ポート 1C
- RED : 文字信号出力
- \overline{SCK} : シフト・クロック入出力
- SCL : シフト・クロック入出力
- SDA : シリアル・データ入出力
- SI : シリアル・データ入力
- SO : シリアル・データ出力
- TMIN : タイマ・イベント入力
- VCO : 局部発振入力
- V_{DD} : 正電源
- \overline{V}_{SYNC} : 垂直同期信号入力
- X_{IN} : クロック発振
- X_{OUT} : クロック発振

64ピン・プラスチック QFP (□14 mm)



ブロック図



目 次

1. 端子機能 … 11
 - 1.1 端子機能一覧 … 11
 - 1.2 端子の等価回路 … 14

2. プログラム・メモリ (ROM) … 18
 - 2.1 プログラム・メモリの構成 … 18
 - 2.2 プログラム・メモリの機能 … 19
 - 2.3 プログラムの流れ … 19
 - 2.4 プログラムの分岐 … 20
 - 2.5 サブルーチン … 22
 - 2.6 テーブル参照 … 24
 - 2.7 分岐命令およびサブルーチン・コール命令使用時の注意 … 24

3. プログラム・カウンタ (PC) … 25

4. スタック … 26
 - 4.1 構成 … 26
 - 4.2 スタック・ポインタ (SP) … 26
 - 4.3 アドレス・スタック・レジスタ (ASR) … 27
 - 4.4 割り込みスタック・レジスタ … 27

5. データ・メモリ (RAM) … 29
 - 5.1 データ・メモリの構成 … 29
 - 5.2 データ・メモリの機能 … 34
 - 5.3 データ・メモリ使用時の注意 … 38

6. ジェネラル・レジスタ (GR) … 40
 - 6.1 ジェネラル・レジスタの構成 … 40
 - 6.2 ジェネラル・レジスタの機能 … 40
 - 6.3 各命令におけるジェネラル・レジスタとデータ・メモリのアドレス生成 … 42
 - 6.4 ジェネラル・レジスタ使用時の注意 … 46

7. ALU (Arithmetic Logic Unit) ブロック … 48
 - 7.1 ALUブロック概要 … 48
 - 7.2 各ブロックの構成と機能 … 49
 - 7.3 ALU処理命令一覧 … 49
 - 7.4 ALU使用時の注意 … 53

- 8. システム・レジスタ (SYSREG) ... 54
 - 8.1 アドレス・レジスタ (AR) ... 55
 - 8.2 ウインドウ・レジスタ (WR) ... 55
 - 8.3 バンク・レジスタ (BANK) ... 56
 - 8.4 メモリ・ポインタ・イネーブル・フラグ (MPE) ... 56
 - 8.5 インデクス・レジスタ (IX) とデータ・メモリ・ロウ・アドレス・ポインタ (MP) ... 57
 - 8.6 ジェネラル・レジスタ・ポインタ (RP) ... 66
 - 8.7 プログラム・ステータス・ワード (PSWORD) ... 66

- 9. レジスタ・ファイル (RF) ... 67
 - 9.1 IDCDMAEN (00H, b₁) ... 75
 - 9.2 SP (01H) ... 75
 - 9.3 CE (07H, b₀) ... 76
 - 9.4 シリアル・インタフェース・モード・レジスタ (08H) ... 76
 - 9.5 BTM0MD (09H) ... 77
 - 9.6 INTVSYN (0FH, b₂) ... 77
 - 9.7 INTNC (0FH, b₀) ... 78
 - 9.8 水平同期信号カウンタ制御 (11H, 12H) ... 78
 - 9.9 PLL レファレンス・モード・セレクト・レジスタ (13H) ... 79
 - 9.10 INT_{NC} 端子受け付けパルス幅の設定 (15H) ... 79
 - 9.11 タイマ・キャリー (17H) ... 80
 - 9.12 シリアル・インタフェース・ウエイト・コントロール (18H) ... 80
 - 9.13 IEGNC (1FH) ... 80
 - 9.14 A/D コンバータ制御 (21H) ... 81
 - 9.15 PLL アンロック FF ジャッジ・レジスタ (22H) ... 81
 - 9.16 PORT1C の入出力の設定 (27H) ... 82
 - 9.17 シリアル I/O ステータス・レジスタ (28H) ... 82
 - 9.18 割り込み許可フラグ (2FH) ... 83
 - 9.19 CROM バンク選択 (30H) ... 83
 - 9.20 IDCEN (31H) ... 84
 - 9.21 PLL アンロック FF ディレイ・コントロール・レジスタ (32H) ... 84
 - 9.22 P1BBIO_n (35H) ... 85
 - 9.23 P0BBIO_n (36H) ... 85
 - 9.24 P0ABIO_n (37H) ... 86
 - 9.25 シリアル・インタフェース・モードでの割り込み要求発生タイミングの設定 (38H) ... 86
 - 9.26 シフト・クロック周波数の設定 (39H) ... 87
 - 9.27 IRQNC (3FH) ... 87

- 10. データ・バッファ (DBF) ... 88
 - 10.1 データ・バッファの構成 ... 88
 - 10.2 データ・バッファの機能 ... 90
 - 10.3 データ・バッファとテーブル参照 ... 91
 - 10.4 データ・バッファと周辺ハードウェア ... 93

- 10.5 データ・バッファと各周辺レジスタ … 97
- 10.6 データ・バッファ使用時の注意 … 104
- 11. 割り込み … 106
 - 11.1 割り込みブロックの構成 … 106
 - 11.2 割り込みの機能 … 108
 - 11.3 割り込み受け付け動作 … 111
 - 11.4 割り込み受け付け後の動作 … 116
 - 11.5 割り込み処理ルーチンからの復帰処理 … 116
 - 11.6 割り込み処理ルーチン … 117
 - 11.7 外部割り込み (INT_{NC} 端子, $\overline{V_{SYNC}}$ 端子) … 121
 - 11.8 内部割り込み (タイマ, シリアル・インタフェース) … 123
 - 11.9 多重割り込み … 124
- 12. タイマ機能 … 133
 - 12.1 タイマの構成 … 133
 - 12.2 タイマの機能 … 134
 - 12.3 タイマ・キャリー・フリップフロップ (タイマ・キャリーFF) … 136
 - 12.4 タイマ・キャリーFF使用時の注意 … 141
 - 12.5 タイマ割り込み … 147
 - 12.6 タイマ割り込み使用時の注意 … 151
- 13. スタンバイ機能 … 153
 - 13.1 スタンバイ・ブロックの構成 … 153
 - 13.2 スタンバイ機能 … 154
 - 13.3 CE端子によるデバイス動作モード … 155
 - 13.4 ホールト機能 … 156
 - 13.5 クロック・ストップ機能 … 164
 - 13.6 ホールトおよびクロック・ストップ時のデバイス動作 … 167
- 14. リセット機能 … 171
 - 14.1 リセット・ブロックの構成 … 171
 - 14.2 リセットの機能 … 172
 - 14.3 CEリセット … 173
 - 14.4 パワーオン・リセット … 177
 - 14.5 CEリセットとパワーオン・リセットの関係 … 180
 - 14.6 停電検出 … 184
- 15. 汎用ポート … 189
 - 15.1 汎用ポートの構成と分類 … 189
 - 15.2 汎用ポートの機能概略 … 191
 - 15.3 汎用入出力ポート (P0A, P0B, P1B, P1C) … 194
 - 15.4 汎用入力ポート (P0D) … 198

- 15.5 汎用出力ポート (P0C, P1A) ... 199

- 16. シリアル・インタフェース ... 201
 - 16.1 シリアル・インタフェース・モード・レジスタ ... 201
 - 16.2 クロック・カウンタ ... 206
 - 16.3 ステータス・レジスタ ... 207
 - 16.4 ウェイト・レジスタ ... 209
 - 16.5 プリセッタブル・シフト・レジスタ (PSR) ... 214
 - 16.6 シリアル・インタフェース割り込みソース・レジスタ (SIO0IMD) ... 215
 - 16.7 シフト・クロック周波数レジスタ (SIO0CK) ... 216

- 17. D/A コンバータ ... 217
 - 17.1 PWM 端子 ... 217

- 18. PLL 周波数シンセサイザ ... 219
 - 18.1 PLL 周波数シンセサイザの構成 ... 219
 - 18.2 PLL 周波数シンセサイザの各ブロックの機能概略 ... 220
 - 18.3 プログラマブル・ディバイダ (PD) と PLL モード・セレクト・レジスタ ... 221
 - 18.4 基準周波数発生器 (RFG) ... 223
 - 18.5 位相比較器 (ϕ -DET), チャージ・ポンプおよびアンロック検出ブロック ... 225
 - 18.6 PLL デイスエーブル状態 ... 231
 - 18.7 PLL 周波数シンセサイザのデータ設定方法 ... 232

- 19. A/D コンバータ ... 233
 - 19.1 動作原理 ... 233
 - 19.2 D/A コンバータの構造 ... 234
 - 19.3 比較電圧設定レジスタ (ADCR) ... 235
 - 19.4 比較判定レジスタ (ADCCMP) ... 235
 - 19.5 ADC 端子選択レジスタ (ADCCHn) ... 236
 - 19.6 A/D 変換のプログラム例 ... 237

- 20. IDC (Image Display Controller) ... 240
 - 20.1 仕様概要および制限事項 ... 240
 - 20.2 DMA ... 243
 - 20.3 IDC イネーブル・フラグ ... 245
 - 20.4 VRAM ... 246
 - 20.5 CROM (キャラクタ ROM) ... 255
 - 20.6 BLANK, R, G, B 端子 ... 263
 - 20.7 表示開始位置の設定 ... 264
 - 20.8 プログラム例 ... 268

- 21. 水平同期信号カウンタ ... 274
 - 21.1 水平同期信号カウンタの構成 ... 274

- 21.2 ゲート・コントロール・レジスタ (HSCGT) … 275
- 21.3 HSYNC カウンタ (HSC) … 276
- 21.4 水平同期信号カウンタの使用例 … 276

- 22. μPD17062 命令 … 277
 - 22.1 命令セット概要 … 277
 - 22.2 凡例 … 278
 - 22.3 命令セット一覧 … 279
 - 22.4 組み込みマクロ命令 … 281

- 23. アセンブラの予約シンボル … 282
 - 23.1 システム・レジスタ … 282
 - 23.2 データ・バッファ … 282
 - 23.3 ポート・レジスタ … 283
 - 23.4 レジスタ・ファイル … 284
 - 23.5 周辺ハードウェア・レジスタ … 286
 - 23.6 その他 … 286

- 24. 電気的特性 … 287

- 25. 外形図 … 289

- 26. 半田付け推奨条件 … 291

- 付録 開発ツール … 292

1. 端子機能

1.1 端子機能一覧

端子番号		記号	説明	出力形式	パワーオン・リセット時
DIP	QFP (GC)				
1	58	P0C ₃	4ビットの出力ポートです。	CMOS プッシュプル	不定
4	61	P0C ₀			
5	62	P0D ₃ /ADC ₅	ポート 0D と A/D コンバータの入力です。 ●P0D ₃ -P0D ₀ 4ビットの入力ポートです。 プルダウン抵抗を内蔵しています。 ●ADC ₅ -ADC ₂ 4ビットの A/D コンバータの入力です。 プログラムによる逐次比較型で、基準電圧は V _{DD} です。	—	入力
8	1	P0D ₀ /ADC ₂			
9	2	PWM ₃	6ビットの D/A コンバータ出力です。 出力形式は PWM で 15.625 kHz の周波数で出力します。 1ビットの出力ポートとしても使用できます。	N-ch オープン・ドレイン	不定
12	6	PWM ₀			
13	9	V _{DD} V _{DD1} V _{DD0}	デバイスの電源端子です。 すべての機能を動作させるには 5 V ± 10 % の電圧を供給します。 CPU のみ動作時は 4 V、クロック・ストップ時には 3.5 V まで動作させることができます。 電源電圧が 0 V から 4 V まで立ち上がるとデバイスにパワーオン・リセットが働き 0 番地からプログラムがスタートします。 すべての端子に同電位の電圧を加えてください。	—	—
14	11	VCO	局部発振出力を専用プリスケアラで分周した信号を入力します。	—	入力
15	13	EO	PLL のエラー信号出力です。 外部 LPF を介して局部発振回路へ入力します。	CMOS 3ステート	Hi-z
16	15	GND GND ₂ GND ₁ GND ₀	デバイスのグラウンドです。 すべての端子をグラウンドに接続してください。	—	—
17	16	PSC	専用プリスケアラの分周比切り替え信号出力です。	CMOS プッシュプル	不定
18	17	CE	デバイスの選択信号入力です。 PLL および IDC を機能させる場合はハイ・レベルを入力します。 ロウ・レベル入力時はストップ命令を実行することにより、低消費電流でのバックアップも可能です。 入力レベルがロウ・レベルからハイ・レベルに立ち上がるとデバイスにリセットがかかり、プログラムは 0 番地からスタートします。	—	入力
19	18	X _{OUT}	水晶振動子の接続端子です。	—	—
20	19	X _{IN}	8 MHz の振動子を使用します。	—	入力

端子番号		記号	説明	出力形式	パワーオン・リセット時
DIP	QFP (GC)				
21 24	20 24	P1A ₃ P1A ₀	4ビットの出力ポートです。 中耐圧 N-ch オープン・ドレイン出力です。	N-ch オープン・ドレイン	不定
26 29	27 30	P1B ₃ P1B ₀	4ビットの入出力ポートです。 1ビット単位の入出力指定ができます。	CMOS プッシュプル	入力
30 31 32	31 32 33	RED GREEN BLUE	IDC 表示の R, G, B に対応する文字データ出力です。 アクティブ・ハイで出力します。	CMOS プッシュプル	ロウ・レベル
33	34	BLANK	IDC 表示の映像信号カット用のブランキング信号出力です。 アクティブ・ハイで出力します。	CMOS プッシュプル	ロウ・レベル
34	35	\overline{H}_{SYNC}	IDC 表示の水平同期信号入力端子です。 アクティブ・ロウで入力してください。	—	入力
35	36	\overline{V}_{SYNC}	IDC 表示の垂直同期信号入力端子です。 アクティブ・ロウで入力してください。 入力信号で割り込みをかけることができます。	—	入力
36 38	38 45	P1C ₃ /ADC ₁ P1C ₂ P1C ₁	ポート 1C と A/D コンバータの入力です。 ●P1C ₃ -P1C ₁ 3ビットの入出力ポートです。 ●ADC ₁ 4ビットの A/D コンバータの入力です。	CMOS プッシュプル	入力
39	46	ADC ₀	4ビットの A/D コンバータの入力です。	—	入力
40 43 44 47	47 50 51 54	P0B ₃ /HSCNT P0B ₂ /TMIN P0B ₁ P0B ₀ /SI P0A ₃ /SO P0A ₂ / \overline{SCK} P0A ₁ /SCL P0A ₀ /SDA	ポート 0B, ポート 0A, 水平同期信号カウンタ入力, タイマ入力およびシリアル・インタフェースです。 ●P0A ₃ -P0A ₀ 4ビットの入出力ポートです。 1ビット単位で入出力を指定できます。 ●P0B ₃ -P0B ₀ 4ビットの入出力ポートです。 1ビット単位で入出力を指定できます。 ●HSCNT 水平同期信号のカウンタ入力です。 自己バイアスされます。 ●TMIN タイマ入力です。 商用電源などを入力して、時計等に使用します。 ●SI, SO, \overline{SCK} 3線式シリアル・インタフェースの入出力です。 ・SI: シリアル・データ入力 ・SO: シリアル・データ出力 ・ \overline{SCK} : シフト・クロック入出力 ●SDA, SCL 2線式シリアル・インタフェースの入出力です。 ・SCL: シリアル・クロック入出力 ・SDA: シリアル・データ入出力	N-ch オープン・ドレイン (P0A ₁ , P0A ₀) CMOS プッシュプル (P0A ₁ , P0A ₀ 以外)	入力

端子番号		記号	説明	出力形式	パワーオン・リセット時
DIP	QFP (GC)				
48	55	INT _{NC}	<p>割り込み入力です。 ノイズ・キャンセラを内蔵しています。 入力信号の立ち上がりまたは立ち下がりで割り込みを かけられます。</p>	—	入力
—	5 7 8 10 12 14 22 25 37 39 40 41 42 44 56 57	NC	<p>ノー・コネクションです。 デバイス内部とは接続されていませんので自由に接続 してください。</p>		

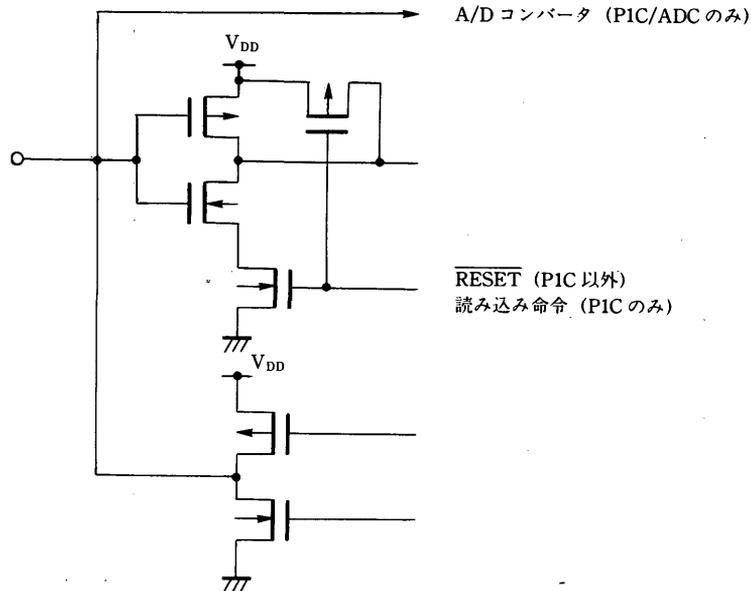
1.2 端子の等価回路

P0A (P0A₃/SO, P0A₂/SCK)

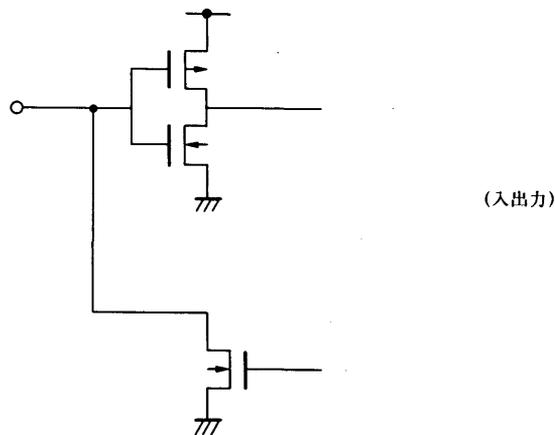
P0B (P0B₁, P0B₀/SI)

P1B (P1B₃, P1B₂, P1B₁, P1B₀)

P1C (P1C₃/ADC₁, P1C₂, P1C₁)

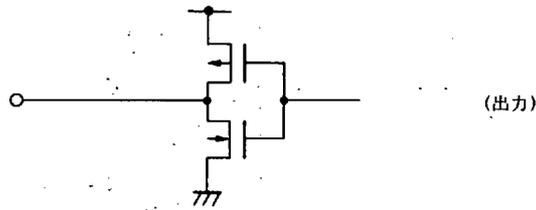


P0A (P0A₁/SCL, P0A₀/SDA)



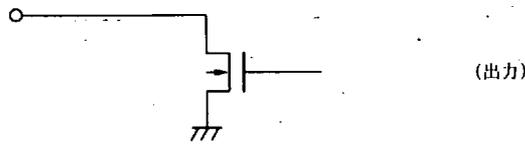
P0C (P0C₃, P0C₂, P0C₁, P0C₀)

RED, GREEN, BLUE, BLANK, PSC

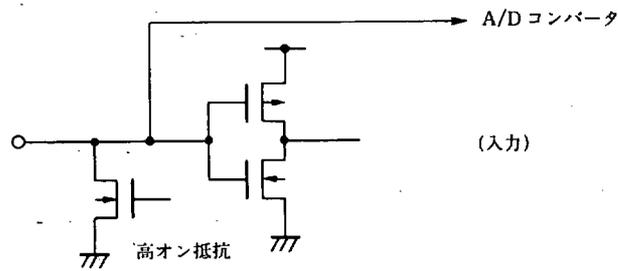


PWM (PWM₃, PWM₂, PWM₁, PWM₀)

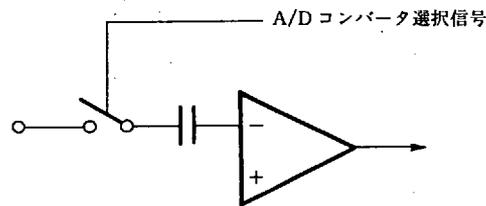
P1A (P1A₃, P1A₂, P1A₁, P1A₀)



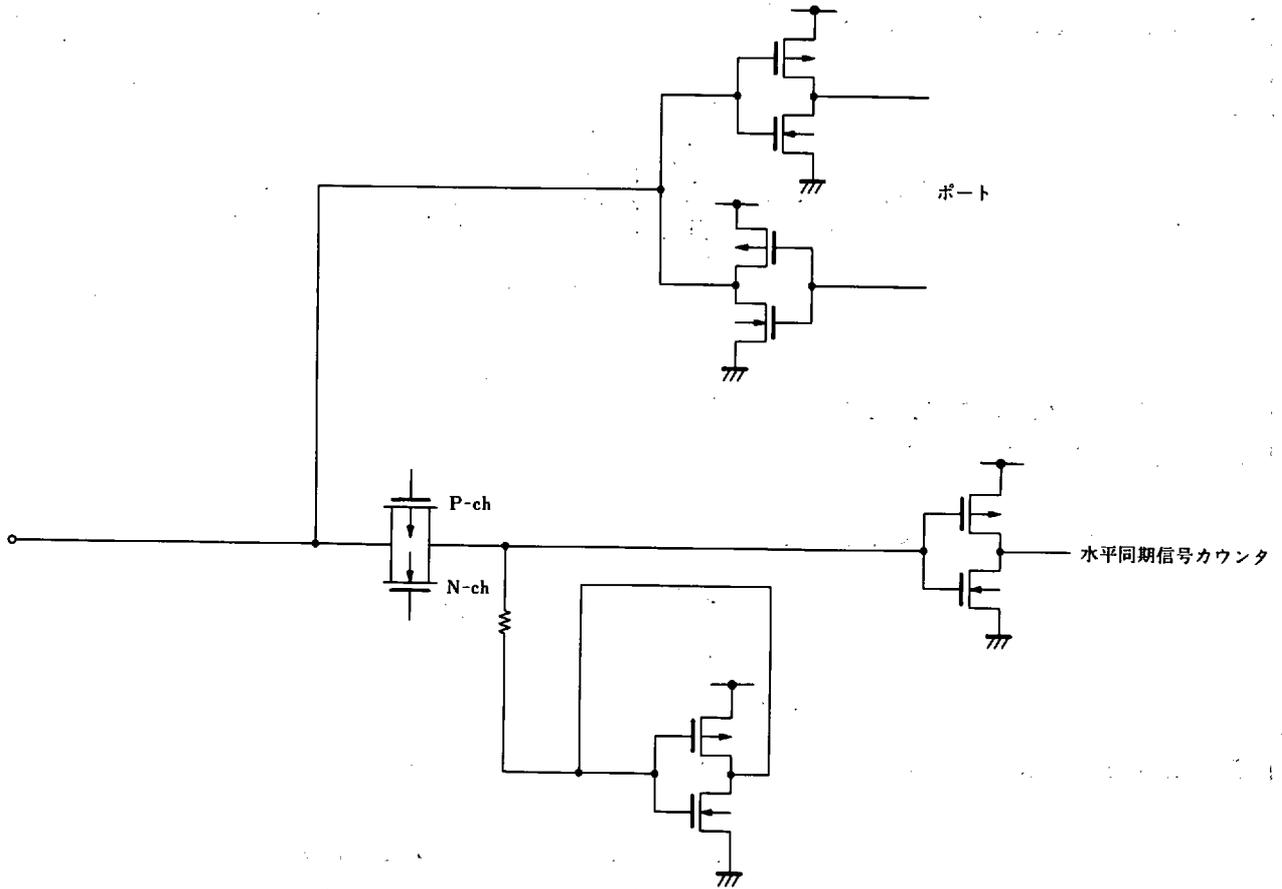
P0D (P0D₃/ADC₅, P0D₂/ADC₄, P0D₁/ADC₃, P0D₀/ADC₂)



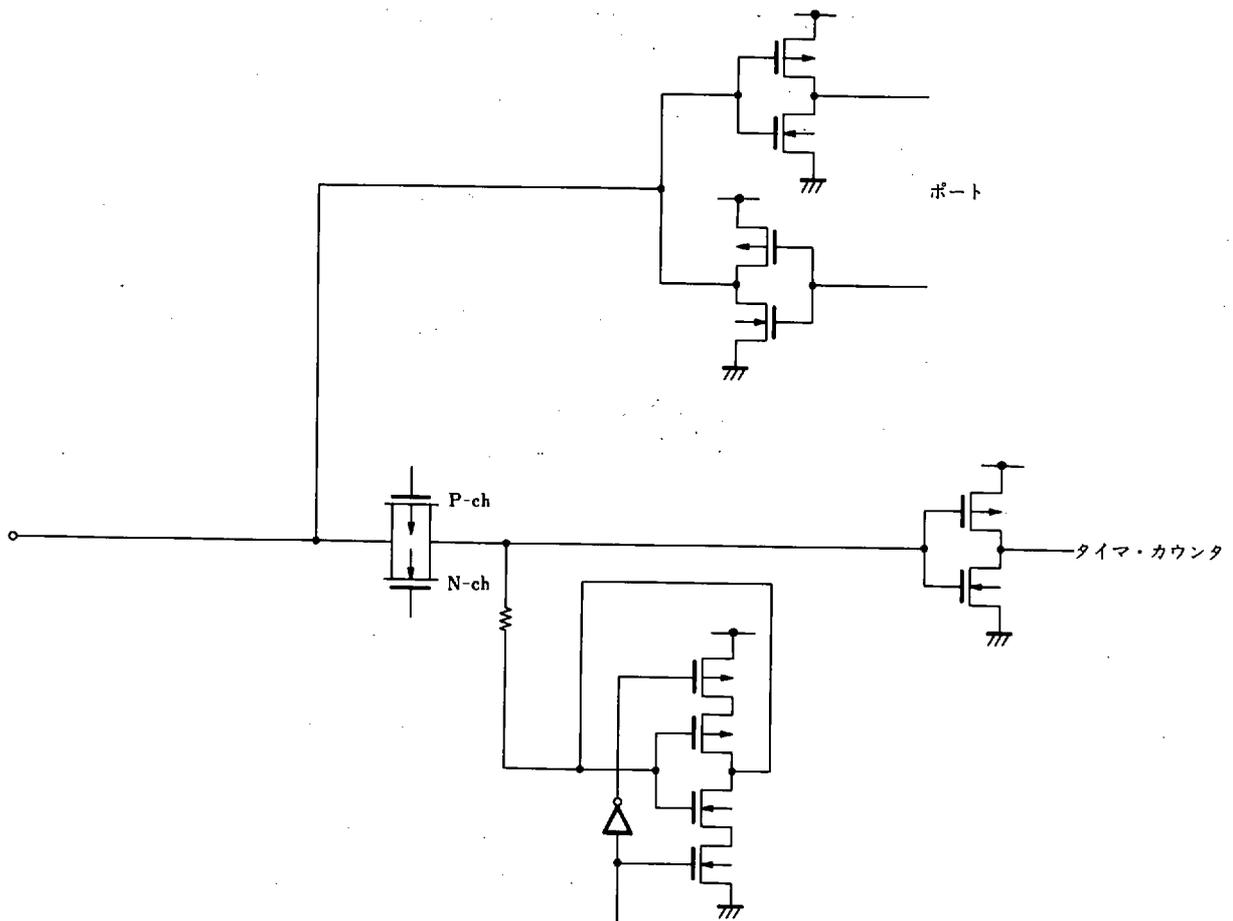
ADC₀



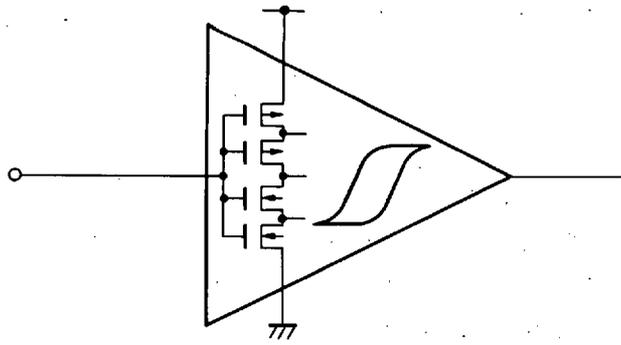
P0B₃/HSCNT



P0B₂/TMIN

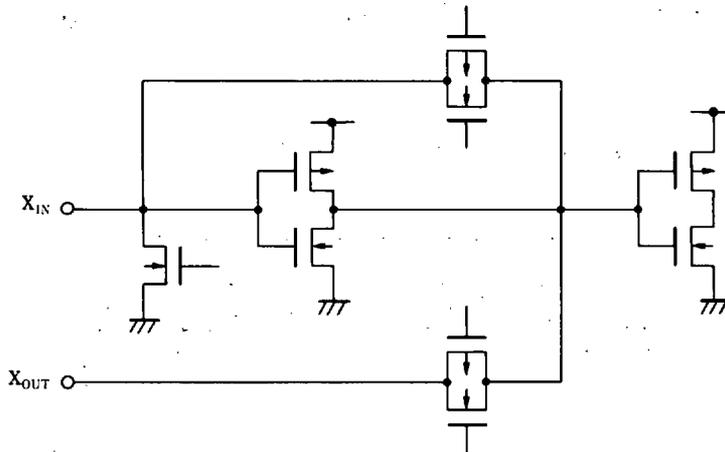


\overline{H}_{SYNC} , \overline{V}_{SYNC} , INT_{NC} , CE

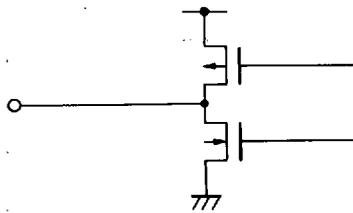


(ヒステリシス入力)

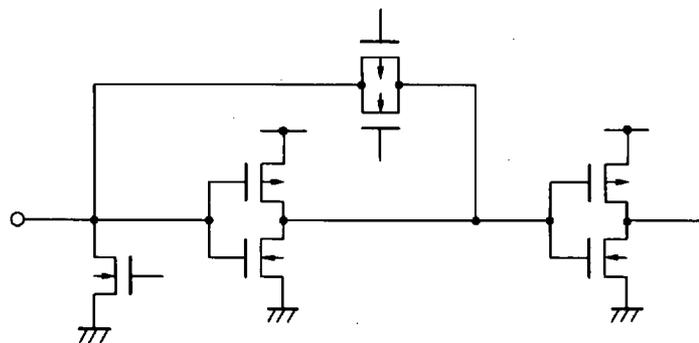
X_{OUT} , X_{IN}



EO



VCO



(入力)

2. プログラム・メモリ (ROM)

プログラム・メモリはCPUが実行する“プログラム”およびあらかじめ決まっている“定数データ”を格納しておきます。

2.1 プログラム・メモリの構成

図2-1にプログラム・メモリの構成を示します。

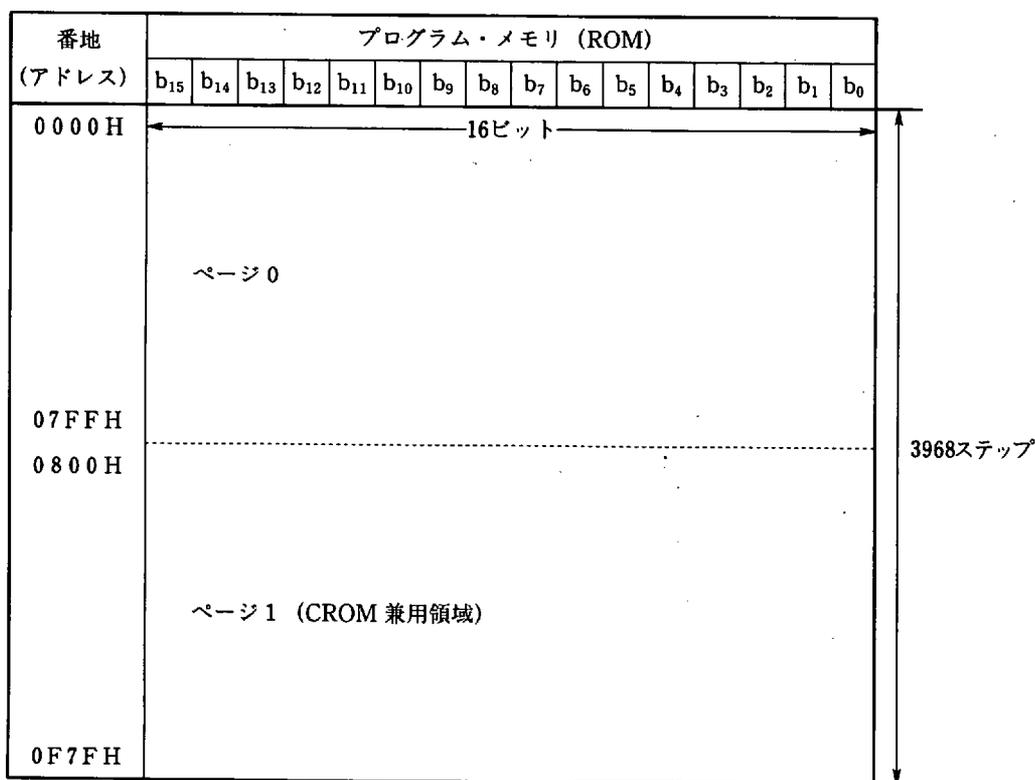
図2-1に示すようにプログラム・メモリは8Kバイト(3968×16ビット)で構成されています。

プログラム・メモリは16ビット単位で“番地(アドレス)”がつけられており、0000H番地から0F7FH番地まであります。また、“ページ”という概念があり、0000H-07FFH番地を“ページ0”、0800H-0F7FH番地を“ページ1”と呼びます。

0800H-0F7FHはIDC用の表示パターンをストアするためのCROM(Charactor ROM)領域としても使用します。CROMとして使用しない場合はプログラム領域として使用可能です。

0000H-00FFHはテーブル参照用エリアで、JMP @AR, CALL @AR, MOVT, PUSH, POP命令で使用します。

図2-1 プログラム・メモリの構成



2.2 プログラム・メモリの機能

プログラム・メモリは大別して次の2つの機能があります。

- (1) プログラムを格納しておく
- (2) 定数データを格納しておく

プログラムとはCPU(Central Processing Unit:実際にマイクロコントローラを制御する)を動作させる“命令”の集まりであり、プログラムに書かれた“命令”に従い順次CPUが処理を実行していきます。すなわち、CPUはプログラム・メモリに格納されているプログラムから順次“命令”を読み出していき、各“命令”に従って処理を実行します。

“命令”はすべて16ビット長の“1語命令”であるため、プログラム・メモリの1つの番地に1つの命令を格納することができます。

定数データとは、たとえば表示用パターンのようにあらかじめ決まっているようなデータです。定数データは、専用命令であるMOV命令を使用することによりプログラム・メモリからデータ・メモリ(RAM)上のデータ・バッファ(DBF)に読み込まれます。このようにプログラム・メモリ上の定数データを読み込むことを“テーブル参照”と呼びます。

プログラム・メモリは読み出し専用のメモリであるため“命令”により書き換えることはできません。したがってプログラム・メモリとROM(Read Only Memory)は同一の意味として使用します。

2.3 プログラムの流れ

プログラム・メモリに格納されているプログラムは、通常0000H番地から1番地ごとに実行されます。ところが、たとえばある条件により異なるプログラムを実行させるような場合はプログラムの流れを分岐する必要があります。このような場合は、分岐命令(BR)を使用します。

また、同一のプログラムを何度も実行させたい場合は、そのたびに同一プログラムを用いているとプログラム・メモリの効率が低下します。このような場合は一箇所にプログラムをまとめておき、専用命令であるCALL命令によりこのプログラムを呼び出すことにより何度も同一のプログラムを実行させることができます。このプログラムのことを“サブルーチン”と呼びます。サブルーチンに対して通常実行しているプログラムを“メイン・ルーチン”と呼びます。

また、プログラムの流れとはまったく関係なく、ある条件が成立した場合に実行させたいプログラムがあるときは、割り込み機能を使用します。割り込み機能は条件が成立すると現在のプログラムの流れとは関係なくあらかじめ決められた番地(ベクタ・アドレスと呼ぶ)へ分岐することができます。

前述したようなプログラムの流れは、プログラム・メモリの番地を指定するプログラム・カウンタ(PC)によって制御されます。

2.4 プログラムの分岐

プログラムの分岐は分岐命令 (BR) により行います。

図 2-2 に分岐命令の動作を示します。

分岐命令 (BR) には、命令のオペランドで指定されたプログラム・メモリのアドレス (addr) に直接分岐する直接分岐命令 (BR addr) と、後述するアドレス・レジスタ (AR) の内容で指定されたプログラム・メモリのアドレスへ分岐する間接分岐命令 (BR @AR) の 2 種類があります。

“3. プログラム・カウンタ (PC)” の項も参照してください。

2.4.1 直接分岐

直接分岐命令では命令のオペレーション・コードの最下位 1 ビットと命令のオペランド 11 ビットの合計 12 ビットで分岐先のプログラム・メモリ・アドレスを指定します。したがって直接分岐命令における分岐先アドレスはプログラム・メモリの全アドレスである 0000H-0F7FH 番地になります。

2.4.2 間接分岐

間接分岐命令ではアドレス・レジスタの 8 ビット・データにより分岐先のアドレスを指定します。したがって間接分岐命令の分岐先アドレスは 0000H-00FFH 番地に限定されます。

“8.1 アドレス・レジスタ (AR)” の項を参照してください。

図 2-2 分岐命令の動作とマシン・コード

(a) 直接分岐 (BR addr)

(b) 間接分岐 (BR @AR)

アドレス	プログラム・メモリ
0000H	ラベル:命令 (マシン・コード)
	BR AAA (0C500)
	BR BBB (0D100)
0500H	AAA:
07FFH	ページ 0
0800H	BR AAA (0C500)
0900H	BBB:
	BR BBB (0D100)
0F7FH	ページ 1

アドレス	プログラム・メモリ
0000H	ラベル:命令 (マシン・コード)
0010H	
0085H	
	MOV AR0, #5H
	MOV AR1, #8H
	BR @AR
0500H	
07FFH	ページ 0
0800H	MOV AR0, #0H
	MOV AR1, #1H
	BR @AR
0F7FH	ページ 1

備考 17K シリーズのマシン・コード (16ビット) は、1ビット、4ビット、3ビット、4ビット、4ビットで構成されます。したがって、このドキュメントではマシン・コードを分かりやすくするため、各構成ビットごとに桁を分けて表現しています。

例 マシン・コード 0C500 → $\begin{matrix} 0 & 1100 & 101 & 0000 & 0000 \\ 1 & 4 & 3 & 4 & 4 \end{matrix}$

2.4.3 デバッグ時の注意

図 2-2 に示すように、ページ 0 内 (0000H-07FFH 番地) への直接分岐命令とページ 1 内 (0800H-0F7FH 番地) への直接分岐命令のオペレーション・コードは異なるオペレーション・コードになっています。

ページ 0 内への直接分岐命令のオペレーション・コードは“0CH”でありページ 1 内へのオペレーション・コードは“0DH”です。

これは、直接分岐命令のオペランド“addr”が11ビットしかないためオペレーション・コードの最下位ビットを分岐先のアドレスとして使用しているためです。

このオペレーション・コードはアセンブルの際 17K シリーズのアセンブラ (AS17K) を使用すればアセンブラがラベルで指定されたジャンプ先を参照して自動的に変換します。

デバッグ時にパッチ修正を行う場合は、プログラマ自身がページ 0 またはページ 1 への分岐を判断し、オペレーション・コードの“0CH”と“0DH”を変換する必要があります。

たとえば、図 2-2 の (a) で BBB のアドレス 0900H 番地をアドレス 0910H にパッチ修正する場合は、“BR BBB”命令のマシン・コードを“0D110”と入力します。

2.5 サブルーチン

サブルーチンは専用命令であるサブルーチン・コール命令(CALL)とサブルーチン・リターン命令(RET, RETSK)で使用します。

図 2-3 にサブルーチン・コールの動作を示します。

サブルーチン・コール命令には、命令のオペランドで指定されたプログラム・メモリ・アドレス (addr) を直接コールする直接サブルーチン・コール命令 (CALL addr) とアドレス・レジスタの内容で指定されたプログラム・メモリ・アドレスをコールする間接サブルーチン・コール命令 (CALL @AR) があります。

サブルーチンからの復帰命令には RET 命令と RETSK 命令を使用します。RET および RETSK 命令を実行することによりサブルーチン・コール命令 (CALL) を実行した次のプログラム・メモリ・アドレスへ復帰します。このとき RETSK 命令は復帰した最初の命令をノー・オペレーション命令 (NOP) として実行します。

“3. プログラム・カウンタ (PC)” の項も参照してください。

2.5.1 直接サブルーチン・コール

直接サブルーチン・コール命令は、命令のオペランド11ビットでコール先のプログラム・メモリ・アドレスを指定します。したがって、直接サブルーチン・コール命令を使用する場合は、その呼び出し番地つまりサブルーチンの先頭番地はページ0内 (0000H-07FFH 番地) に置く必要があります。ページ1 (0800H-0F7FH 番地) に先頭番地のあるサブルーチンは呼び出すことができません。

ただし、サブルーチン・リターン命令 (RET, RETSK) はページ1内にあってもかまいません。また CALL 命令はページ0, ページ1のどちらにあってもかまいません。

例 1. サブルーチンからのリターン命令がページ0内にある場合

図 2-4 に示すように、サブルーチンの先頭番地がページ0内であれば、戻り番地、リターン命令とともにページ0およびページ1内のいずれにあってもかまいません。

サブルーチンの先頭番地がページ0にある限り、CALL 命令はページ0の概念なく使用することができます。しかしプログラムの作成上、サブルーチンの先頭番地をページ0内に置くことができない場合は、例 2 に示す手法が有効となります。

2. サブルーチンの先頭番地がページ1内にある場合

図 2-4 に示すように、ページ0内に分岐命令 (BR) を設け、この BR 命令を介して、実際のサブルーチン (SUB1) を呼び出す手法です。

2.5.2 間接サブルーチン・コール

間接サブルーチン・コール命令 (CALL @AR) は、アドレス・レジスタ (AR) の8ビット・データによりサブルーチン・コール先のアドレスを指定します。したがって、サブルーチン・コールできるプログラム・メモリ・アドレスは 0000H-00FFH 番地になります。

“8.1 アドレス・レジスタ (AR)” の項を参照してください。

2.6 テーブル参照

テーブル参照は、プログラム・メモリ内の定数データを参照するときに使用します。MOVT DBF, @AR 命令を実行すると、アドレス・レジスタで指定されるプログラム・メモリ・アドレスの内容がデータ・バッファ (DBF) に格納されます。

プログラム・メモリの内容は16ビットで構成されているので、MOVT 命令でデータ・バッファに格納される定数データは16ビット (4ワード) になります。アドレス・レジスタは8ビットで構成されているので、MOVT 命令により参照できるプログラム・メモリ・アドレスは 0000H-00FFH 番地になります。

テーブル参照を行うときは、スタックが1レベル使用されます。

“8.1 アドレス・レジスタ” および “10.3 データ・バッファとテーブル参照” の項を参照してください。

2.7 分岐命令およびサブルーチン・コール命令使用時の注意

17K シリーズのアセンブラ (AS17K) を使用するうえでは例1に示すように分岐命令 (BR) やサブルーチン・コール命令 (CALL) のオペランドに直接プログラム・メモリ・アドレス (数値によるアドレス) を指定すると、エラーが発生します。

これは、プログラム修正時などにおけるバグ要因を低減する目的でアセンブラに組み込まれています。

例1. エラーとなる場合

```

;①
BR      0005H    ;アセンブラでエラーになる。
;②
CALL   00F0H    ;

```

2. エラーとならない場合

```

;③
LOOP1:                ;プログラムの中にレーベルを用い、そのレーベルに対して BR 命令や
BR      LOOP1        ; CALL 命令を行う。
;④
SUB1:                 ;
CALL   SUB1          ;
;⑤
LOOP2 LAB 0005H      ; 0005H をレーベル型として LOOP2 に割り当てる。
BR      LOOP2        ;
;⑥
BR. LD. 0005H        ; オペランドの数値をレーベル型に変換する。
;                  ;   ただし、この方法はバグ要因を低減するためにはあまり使用するべ
;                  ;   きではない。

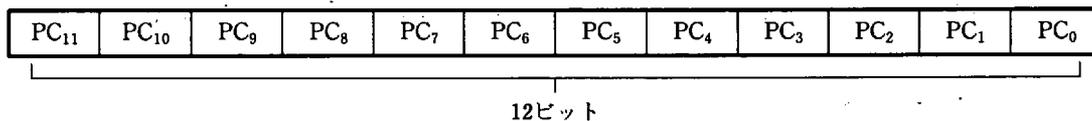
```

詳しくは “AS17K ユーザーズ・マニュアル” を参照してください。

3. プログラム・カウンタ (PC)

プログラム・カウンタはプログラム・メモリすなわちプログラムをアドレスするカウンタで、12ビットのバイナリ・カウンタで構成されています。

図 3-1 プログラム・カウンタ



通常、命令を1つ実行するごとに1つずつインクリメントされますが、分岐命令、サブルーチン・コール命令実行時にはオペランド部で指定されたアドレスがロードされます。スキップ命令を実行したときはスキップ条件の内容にかかわらず、スキップ命令の次の命令のアドレスを指定します。そのときスキップすべき条件であれば、スキップ命令の次の命令はNOP命令とみなされます。つまりNOP命令を実行して、さらに次の命令のアドレスを指定することになります。

また、割り込み要求受け付け時には無条件に1-4番地(割り込み要因により異なります)をPCにロードします。パワーオン・リセット時およびCEリセット時には、プログラム・カウンタは0番地にリセットされます。

表 3-1 割り込み時のベクタ・アドレス

優先順位	割り込み要因	ベクタ・アドレス
1	INT _{NC} 端子	4H
2	内部タイマ	3H
3	\overline{V}_{SYNC} 端子	2H
4	シリアル・インタフェース	1H

4. スタック

スタックとはサブルーチン・コール時や割り込み受け付け時にプログラムの戻り番地や後述するシステム・レジスタの内容を退避するためのレジスタです。

4.1 構成

スタックは4ビットのバイナリ・カウンタであるスタック・ポインタ (SP) を1個と13ビットのアドレス・スタック・レジスタ (ASR) を6個と3ビットの割り込みスタック・レジスタ2個より構成されています。

4.2 スタック・ポインタ (SP)

スタック・ポインタはレジスタ・ファイル上の01H番地にあり、アドレス・スタック・レジスタを指定します。PUSH操作 (CALL, MOVT, PUSH命令, 割り込み受け付け)により-1し, POP操作 (RET, RETSK, RETI, MOVT, POP命令)により+1します。

スタック・ポインタの上位1ビットは、常に“0”です。スタック・ポインタの取り得る値は、0H-7Hの8通りですが、6H, 7Hに対応するスタックはありません。

図4-1. スタック・ポインタの構成

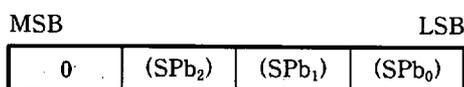


表4-1 スタック・ポインタの動作

命 令	スタック・ポインタの値
CALL addr CALL @AR MOVT DBF, @AR PUSH AR 割り込み受け付け	SP-1
RET RETSK MOVT DBF, @AR POP AR RETI	SP+1

4.3 アドレス・スタック・レジスタ (ASR)

6 × 13ビットで構成されるレジスタで、サブルーチン・コール命令実行時あるいは割り込み要求受け付け時に、プログラム・カウンタの内容に+1した値、すなわち戻り番地を格納します。スタック・レジスタの内容は、リターン命令実行でプログラム・カウンタにロードされ、もとのプログラムの流れに復帰します。

スタック・レジスタはサブルーチン・コールと割り込みの両方に使用されますので、割り込みに2レベル使用すると、サブルーチン・コールに使用できるスタック・レジスタは残りの4レベルとなります。

MOVT 命令を実行すると一時的にスタック・レジスタを使用します。

図 4-2 アドレス・スタック・レジスタの構成

スタック・ポインタの値

0H	ASR ₀
1H	ASR ₁
2H	ASR ₂
3H	ASR ₃
4H	ASR ₄
5H	ASR ₅

4.4 割り込みスタック・レジスタ

割り込みスタック・レジスタは図 4-3 に示すように 2 × 3ビットで構成されるレジスタです。

割り込みが受け付けられると後述するシステム・レジスタ (SYSREG) 中のバンク・レジスタ (BANK) の2ビットとインデックス・イネーブル・フラグ (IXE) の1ビットを退避します。次に割り込みリターン命令 (RETI) が実行されると割り込みスタック・レジスタの内容をシステム・レジスタのバンク・レジスタとインデックス・イネーブル・フラグに復帰させます。

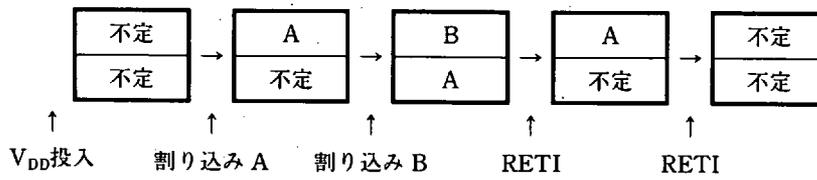
割り込みスタック・レジスタはアドレス・スタック・レジスタのようにスタック・ポインタで指定されるアドレスがなく、図 4-4 に示すように割り込みが受け付けられるごとにデータを退避していき、割り込みリターン命令が実行されるごとにデータを復帰します。また、2レベルを越える割り込みが受け付けられると最初のデータははき出されてしまうため、プログラムにて退避する必要があります。

パワーオン・リセット時、割り込みスタック・レジスタの内容は不定です。また CE リセット時およびクロック・ストップ命令実行時は以前の状態を保持します。

図 4-3 割り込みスタック・レジスタの構成

	MSB	LSB
0H	BANKSK0	IXESK0
1H	BANKSK1	IXESK1

図 4-4 割り込みスタック・レジスタの動作



5. データ・メモリ (RAM)

データ・メモリは、演算・制御等のデータを記憶するメモリです。命令により常時データの書き込みや読み出しが行えます。

5.1 データ・メモリの構成

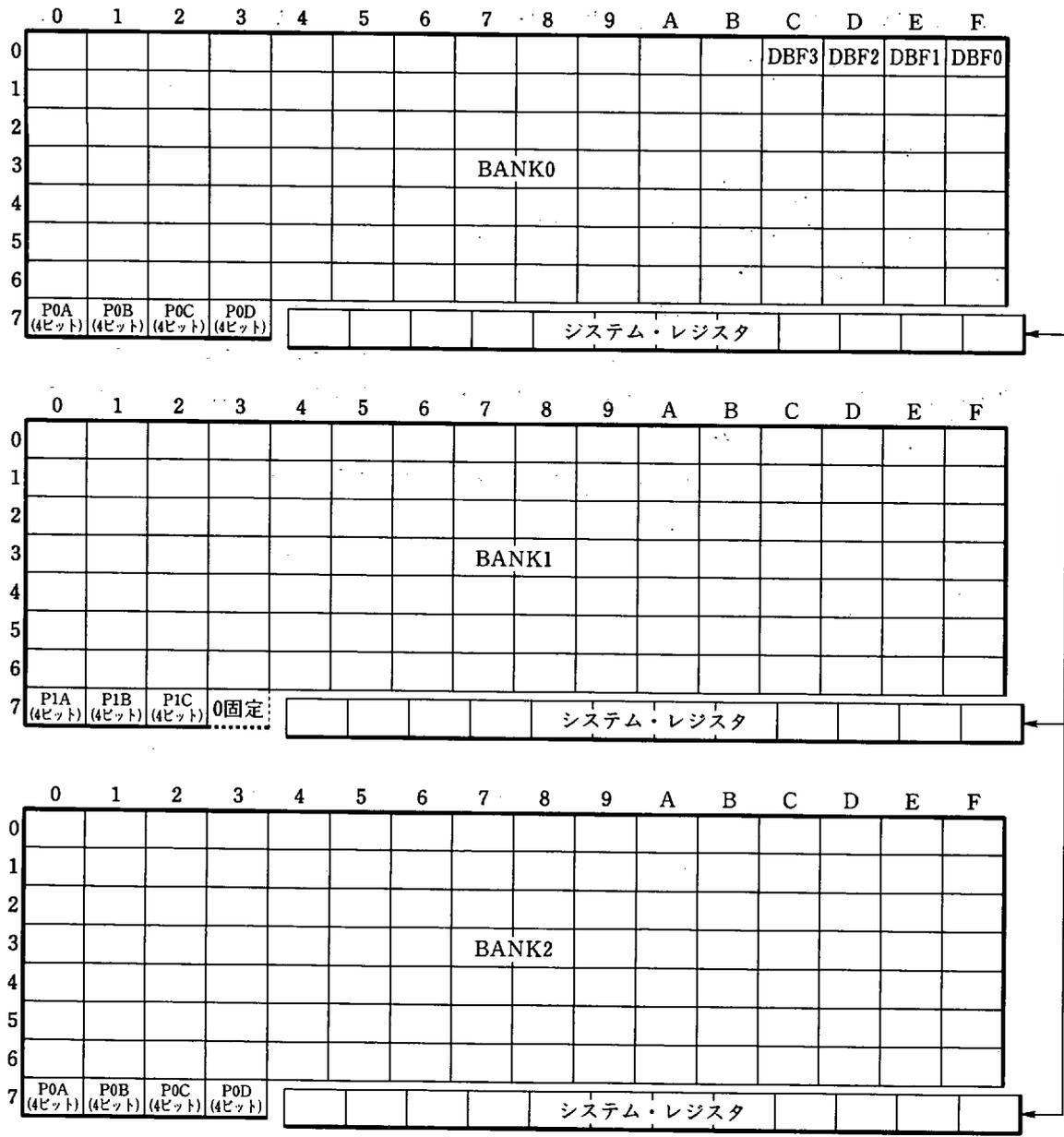
図 5-1 にデータ・メモリの構成を示します。

図 5-1 に示すように、データ・メモリは“バンク”と呼ぶ単位で3つに分割されています。3つのバンクをそれぞれ BANK0, BANK1 および BANK2 と呼びます。

各バンクは4ビット単位のデータごとに番地 (アドレス) が割り付けられており、上位3ビットを“ロウ・アドレス”, 下位4ビットを“カラム・アドレス”と呼びます。たとえば、ロウ・アドレスが 1H でカラム・アドレスが AH のデータ・メモリはアドレス 1AH のデータ・メモリと呼びます。また1つのアドレスは4ビットのメモリで構成されており、これを“1ニブル”と呼びます。

また、データ・メモリは機能別に以下の 5.1.1-5.1.5 に示すブロックに分けられます。

図5-1 データ・メモリの構成



5.1.1 システム・レジスタ (SYSREG) の構成

データ・メモリのアドレス 74H-7FH に割り当てられた12ニブルで構成されています。システム・レジスタはバンクに無関係に割り当てられています。すなわち、どのバンクであってもアドレス 74H-7FH には同一のシステム・レジスタが存在します。

図 5-2 に構成を示します。

図 5-2 システム・レジスタの構成

システム・レジスタ (SYSREG)												
アドレス	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
名称 (記号)	アドレス・レジスタ (AR)				ウインド ウ・レジ スタ (WR)	バンク・ レジスタ (BANK)	インデクス・レジスタ (IX) データ・メモリ・ロウ・ アドレス・ポインタ (MP)			ジェネラル・ レジスタ・ ポインタ (RP)		プログラム・ ステータス・ ワード (PSWORD)

5.1.2 データ・バッファ (DBF) の構成

データ・メモリの BANK0 のアドレス 0CH-0FH に割り当てられた 4 ニブルで構成されています。

図 5-3 に構成を示します。

図 5-3 データ・バッファの構成

データ・バッファ (DBF)				
アドレス	0CH	0DH	0EH	0FH
記号	DBF3	DBF2	DBF1	DBF0

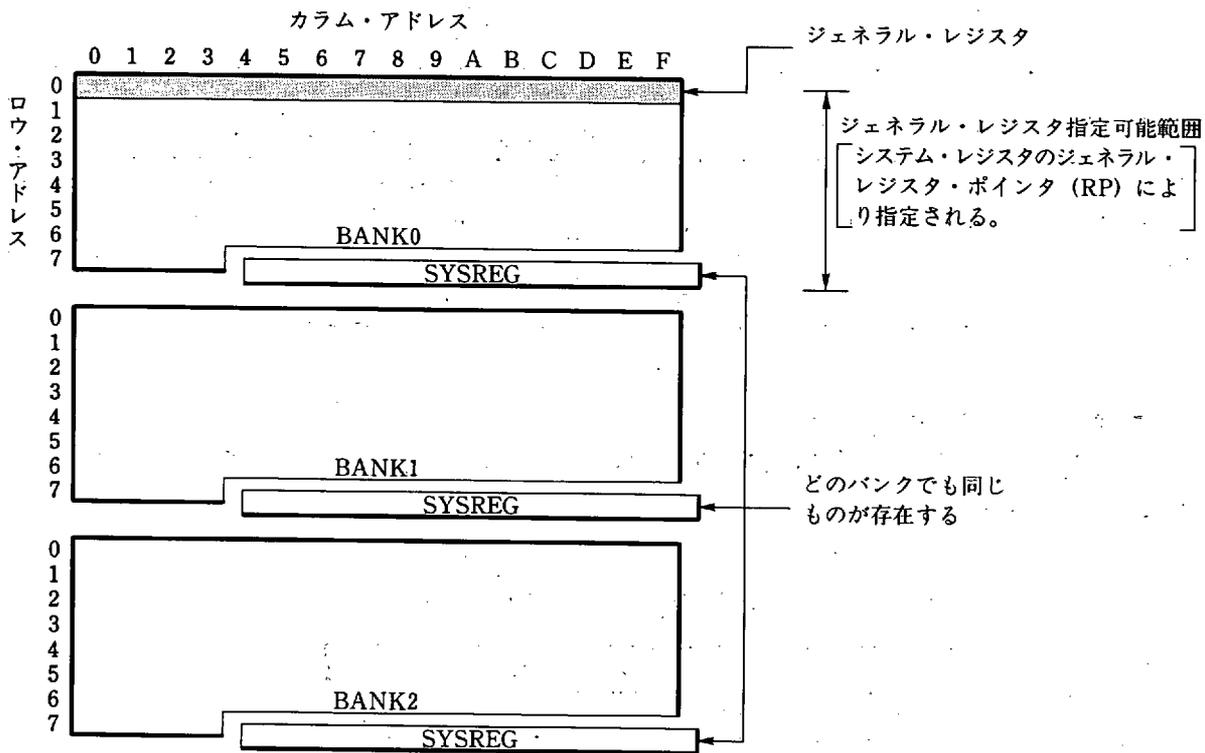
5.1.3 ジェネラル・レジスタ (GR) の構成

データ・メモリの任意のロウ・アドレスで指定される12ニプルで構成されています。

任意のロウ・アドレスとはシステム・レジスタの中のジェネラル・レジスタ・ポインタにより指定されます。

図5-4に構成を示します。

図5-4 ジェネラル・レジスタ (GR) の構成



5.1.4 ポート・データ・レジスタ（ポート・レジスタ）の構成

データ・メモリの各バンクのアドレス 70H-73H に割り当てられた12ニブルで構成されます。

図 5-5 に構成を示します。

図 5-5 に示すように BANK0 と BANK2 は同じポート・レジスタが割り当てられているので実際は 8 ニブルで構成されます。

図 5-5 ポート・レジスタの構成

ポート・レジスタ					
アドレス		70H	71H	72H	73H
記号	BANK0 BANK2	P0A	P0B	P0C	P0D
	BANK1	P1A	P1B	P1C	0 固定

5.1.5 汎用データ・メモリの構成

データ・メモリからシステム・レジスタおよびポート・レジスタを除いた部分です。

BANK0-BANK2 の各112ワードの計336ワードから構成されます。

5.1.6 実装されていないデータ・メモリ

図 5-6 に示したようにポート・レジスタのアドレス BANK1 の 72H 番地のビット 0 は何も割り当てられていません。これらのアドレスに対しては“5.3.2 実装されていないデータ・メモリに対する注意”の項を参照してください。

5.2 データ・メモリの機能

データ・メモリは表 5-1 に示すデータ・メモリ操作命令を実行することにより、データ・メモリ上のデータとイミディエト・データ（任意のデータ）間で4ビットの演算、比較、判断および転送を1命令で行うことができます。

また、ジェネラル・レジスタを使用することにより、データ・メモリと、ジェネラル・レジスタ間で4ビットの演算、比較および転送を1命令で行うことができます。

以下に例を示します。なお詳細については“6. ジェネラル・レジスタ (GR)” および “7. ALU (Arithmetic Logic Unit) ブロック” の項を参照してください。

例1. データ・メモリの演算

;①

MOV 35H, #0001B ; そのとき選択されているバンクのアドレス 35H のデータ・メモリの内容に
; イミディエト・データである 0001B を転送する (書き込む)。

;②

ADD 76H, #0001B ; そのとき選択されているバンクのアドレス 76H のデータ・メモリの内容に
; イミディエト・データである 0001B を加算する。

①、②ともそのとき選択されているバンクとはシステム・レジスタの中のバンク・レジスタにより指定されます。バンク・レジスタについては“8. システム・レジスタ (SYSREG)” の項を参照してください。

②はアドレス 76H のデータ・メモリに対する加算命令ですが、アドレス 76H はシステム・レジスタでもあります。システム・レジスタは BANK に無関係に存在しているため、結局この命令はバンクには無関係にシステム・レジスタの 76H に 0001B を加算したことになります。

備考 命令の記述方法については、5.3.1 データ・メモリのアドレス指定を参照。

2. データ・メモリとジェネラル・レジスタの演算

ジェネラル・レジスタが BANK0 のロウ・アドレス 1H にあるとき

;①

ADD 7H, 36H ; カラム・アドレスが 7H であるジェネラル・レジスタすなわち BANK0 の
; 17H 番地の内容に、そのとき選択されているバンクのアドレス 36H のデー
; タ・メモリの内容を加算する。

;②

LD 7H, 36H ; カラム・アドレスが 7H であるジェネラル・レジスタにデータ・メモリのア
; ドレス 36H の内容を転送する。
; このとき、ジェネラル・レジスタは、BANK0 のアドレス 17H になる。

システム・レジスタ、データ・バッファ、ジェネラル・レジスタおよびポート・レジスタはすべてデータ・メモリ操作命令によりデータ・メモリとして操作することが可能です。

また 5.2.1-5.2.4 に各機能を説明します。

5.2.1 システム・レジスタ (SYSREG) の機能

システム・レジスタはCPUの制御を行うためのレジスタです。

たとえば図5-2に示すバンク・レジスタは、データ・メモリのバンク指定を行い、ジェネラル・レジスタ・ポインタはジェネラル・レジスタのロウ・アドレスを指定します。

詳しくは“8. システム・レジスタ (SYSREG)”の項を参照してください。

5.2.2 ジェネラル・レジスタ (GR) の機能

ジェネラル・レジスタはデータ・メモリとの間で、演算やデータの転送を行うことができるレジスタです。

ジェネラル・レジスタのバンクとロウ・アドレスはシステム・レジスタ上のジェネラル・レジスタ・ポインタで指定します。ただし、μPD17062のジェネラル・レジスタ・ポインタのBANK指定はBANK0に固定です。

たとえばジェネラル・レジスタ・ポインタを0に設定すると、BANK0のロウ・アドレス0の16ニブル、つまりBANK0の00H-0FH番地がジェネラル・レジスタとして指定されます。

ジェネラル・レジスタを使用するときは、ジェネラル・レジスタとイミューディエト・データとの転送命令や演算命令はできないという点に注意が必要です。すなわちジェネラル・レジスタとイミューディエト・データとの転送や演算を行うときは、そのジェネラル・レジスタをデータ・メモリとして扱う必要があります。

たとえば、ジェネラル・レジスタがBANK0のロウ・アドレス0Hにあるとき(ジェネラル・レジスタ・ポインタが0)に、現在選択されているバンクがBANK0のとき(バンク・レジスタが0)は、“ADD 00H, #1”を実行すると、ジェネラル・レジスタに指定されているBANK0の00H番地の内容を+1することになりますが、現在選択されているバンクがBANK1のとき(バンク・レジスタが1)に、この命令を実行するとBANK1の00H番地の内容が+1されます。

詳しくは、“6. ジェネラル・レジスタ (GR)”の項を参照してください。

5.2.3 データ・バッファ (DBF)

データ・バッファは、周辺回路に転送するデータ、たとえばA/Dコンバータの比較電圧設定データ等や、周辺回路から送られてくるデータ、たとえばシリアル・インタフェースの入力データ等を格納するためのバッファです。

詳しくは“10. データ・バッファ (DBF)”の項を参照してください。

5.2.4 汎用ポート・データ・レジスタ (ポート・レジスタ)

ポート・レジスタは各汎用入出力ポートの出力データの設定や入力データの読み込みを行います。出力ポートに設定されている端子に対応するポート・レジスタにデータを設定することにより各端子の出力を設定します。また、入力ポートに設定されている端子に対応するポート・レジスタを読み込むことにより各端子の入力状態を検出できます。ポート・レジスタと各ポート(各端子)の関係を図5-6に示します。

詳しくは“15. 汎用ポート”の項を参照してください。

表 5-1 データ・メモリ操作命令一覧

機 能		命 令
演算	加算	ADD ADDC
	減算	SUB SUBC
	論理	AND OR XOR
比 較		SKE SKGE SKLT SKNE
転 送		MOV LD ST
判 断		SKT SKF

図5-6 ポート・レジスタと各ポート（端子）の関係

汎用ポート・データ・レジスタ				対象ポート	端 子		
バンク	アドレス	記号	ビット記号		記 号	入出力	
BANK0 BANK2	70H	P0A	b ₃	P0A3	Port0A	P0A ₃	入出力 (ビット I/O)
			b ₂	P0A2		P0A ₂	
			b ₁	P0A1		P0A ₁	
			b ₀	P0A0		P0A ₀	
	71H	P0B	b ₃	P0B3	Port0B	P0B ₃	入出力 (ビット I/O)
			b ₂	P0B2		P0B ₂	
			b ₁	P0B1		P0B ₁	
			b ₀	P0B0		P0B ₀	
	72H	P0C	b ₃	P0C3	Port0C	P0C ₃	出力
			b ₂	P0C2		P0C ₂	
			b ₁	P0C1		P0C ₁	
			b ₀	P0C0		P0C ₀	
73H	P0D	b ₃	P0D3	Port0D	P0D ₃	入力	
		b ₂	P0D2		P0D ₂		
		b ₁	P0D1		P0D ₁		
		b ₀	P0D0		P0D ₀		
BANK1	70H	P1A	b ₃	P1A3	Port1A	P1A ₃	出力
			b ₂	P1A2		P1A ₂	
			b ₁	P1A1		P1A ₁	
			b ₀	P1A0		P1A ₀	
	71H	P1B	b ₃	P1B3	Port1B	P1B ₃	入出力 (ビット I/O)
			b ₂	P1B2		P1B ₂	
			b ₁	P1B1		P1B ₁	
			b ₀	P1B0		P1B ₀	
	72H	PIC	b ₃	PIC3	Port1C	PIC ₃	入出力 (グループ I/O)
			b ₂	PIC2		PIC ₂	
			b ₁	PIC1		PIC ₁	
			b ₀	PIC0		—	
	73H	0 固定					

5.3 データ・メモリ使用時の注意

5.3.1 データ・メモリのアドレス指定

17Kシリーズのアセンブラを使用する場合は、例1に示すようにデータ・メモリ操作命令のオペランドに直接データ・メモリ・アドレスを数値で記述するとエラーが発生します。

これは、プログラムのメンテナンス性を考慮し、プログラム修正時などにバグ要因を低減するためです。ただし、このデータ・シート上では、プログラム例は理解しやすいように実アドレス表記にしてあります。実際のプログラミングの際は、アセンブラの取扱説明書をご覧ください。

例1.

エラーになる場合

; ①

MOV 2FH, #0001B ; アドレス 2FH を直接指定する。

; ②

MOV 0.2FH, #0001B ; BANK0のアドレス 2FH を直接指定する。

エラーにならない場合

; ③

M02F MEM 0.2FH ; BANK0のアドレス 2FH をメモリ型として M02F にシンボル定義す

MOV M02F; #0001B ; る。

; ④

MOV .MD.2FH, #0001B ; アドレス 2FH を .MD. によりメモリ型に変換する。ただし、バグ要因を低減するためにはこのような方法は避けるべきである。

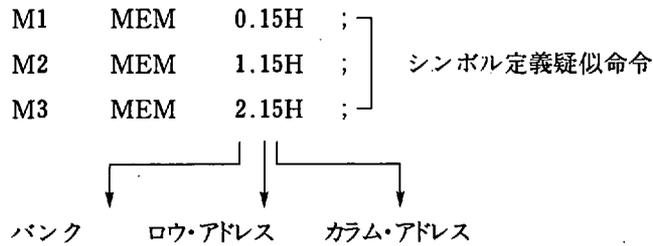
したがってアセンブラ疑似命令である MEM 命令 (シンボル定義疑似命令) を用いてあらかじめ、データ・メモリ・アドレスをシンボル定義しておく必要があります。

データ・メモリのシンボル定義には、例2に示すようにデータ・メモリのバンクも定義する必要があります。

これは、アセンブラ上で自動的にデータ・メモリ・マップを作成するときに使用されます。

ただし、このときたとえば例2に示すように BANK2 にシンボル定義されているデータ・メモリを、プログラム上 BANK1 の範囲で使用すると、BANK1 のデータ・メモリが操作されます。

例 2 .



```

BANK1                ; アセンブラ組み込みマクロ命令  BANK ← 1
MOV  M1,             #0000B ; M1, M2, M3 は①で別のバンクにシンボル定義されているが、
MOV  M2,             #0000B ; プログラム上は BANK1 となっているため、これらの3つの命令は
MOV  M3,             #0000B ; すべて BANK1 のアドレス 15H のデータ・メモリに 0 を書き込みます。
    
```

5.3.2 実装されていないデータ・メモリに対する注意

図 5-6 に示したようにポート・レジスタのアドレス BANK1 の 72H 番地のビット 0 (LSB) には、実際には何も割り当てられていません。

このとき、これらのアドレスに対してデータ・メモリ操作命令を実行すると、以下に示す動作を行います。

(1) デバイス動作

読み込み命令を行うと、“0”が読み込まれます。

書き込み命令を行っても何も変化しません。

(2) アセンブラ動作

通常にアセンブルされます。

“エラー”は発生しません。

(3) エミュレータ (IE-17K) 動作

読み込み命令を行うと“0”が読み込まれます。

書き込み命令を行っても何も変化しません。

“エラー”は発生しません。

6. ジェネラル・レジスタ (GR)

ジェネラル・レジスタはデータ・メモリ空間上に配置されるレジスタで、データ・メモリとの直接演算や、転送を行います。

6.1 ジェネラル・レジスタの構成

ジェネラル・レジスタの構成を図 6-1 に示します。

図 6-1 に示すように、データ・メモリ空間上で同一ロウ・アドレスである16ワード (16ワード×4ビット) をジェネラル・レジスタとして使用できます。

どのロウ・アドレスをジェネラル・レジスタとして使用するかは、システム・レジスタのジェネラル・レジスタ・ポインタによって設定できます。ジェネラル・レジスタ・ポインタは7ビットですが、上位4ビットは0固定のため、ジェネラル・レジスタとして使用できるデータ・メモリ空間は BANK0 のロウ・アドレス 0H-7H になります。

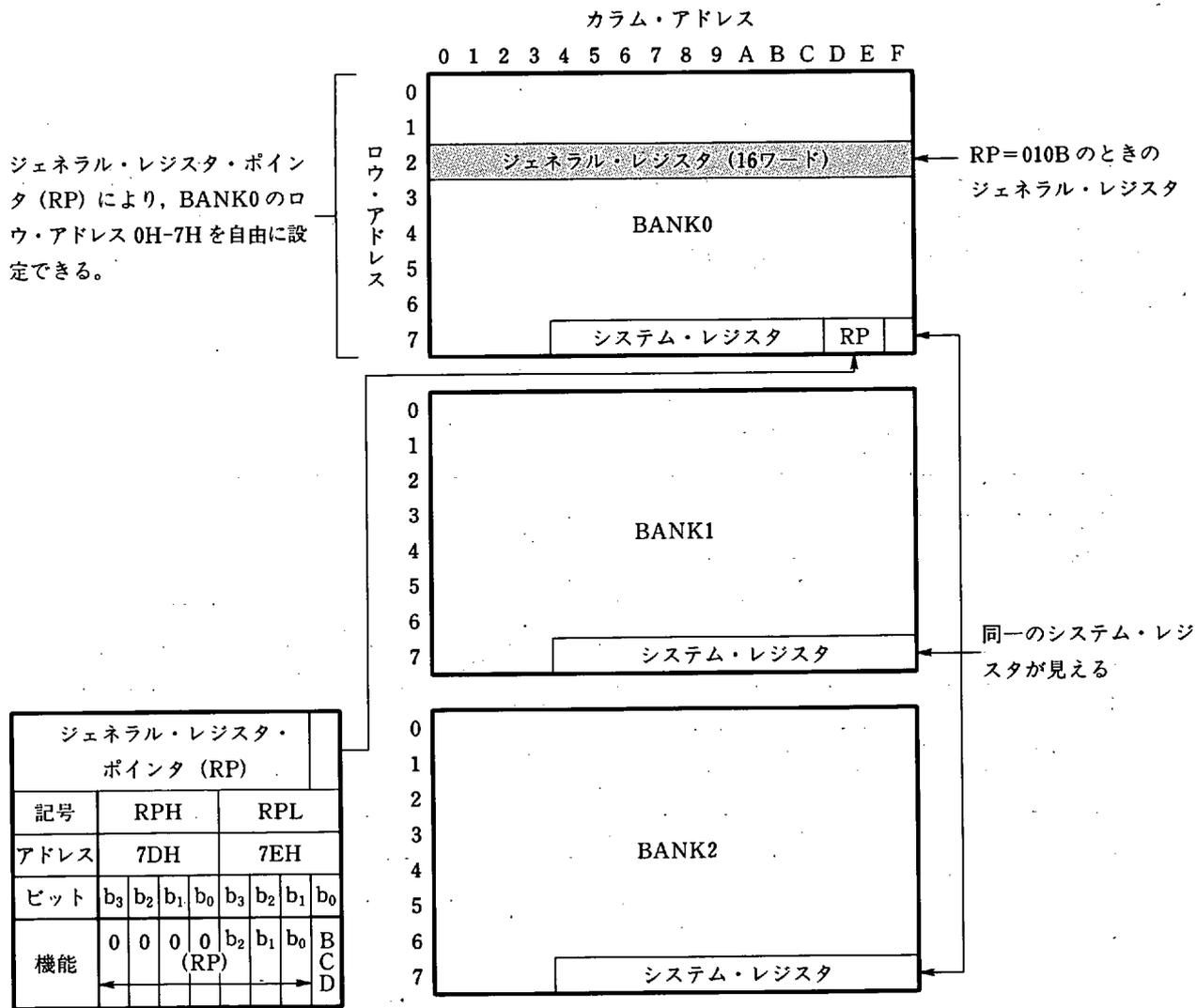
“8.6 ジェネラル・レジスタ・ポインタ (RP)” の項を参照してください。

6.2 ジェネラル・レジスタの機能

ジェネラル・レジスタは、データ・メモリとの間で演算や転送を1命令で行うことが可能です。このジェネラル・レジスタはデータ・メモリ空間にあります。したがって、1命令でデータ・メモリ同士の演算や転送が可能になります。

また、ジェネラル・レジスタは、他のデータ・メモリと同様にデータ・メモリ操作命令で制御することができます。

図6-1 ジェネラル・レジスタの構成



6.3 各命令におけるジェネラル・レジスタとデータ・メモリのアドレス生成

ジェネラル・レジスタとデータ・メモリとの演算および転送命令を表 6-1 に示します。

いま、たとえば

ADD r, m ((r) ← (r) + (m))

を例にとると、表 6-2 に示すようにこの命令で指定されるジェネラル・レジスタのアドレスは、ジェネラル・レジスタ・ポインタと r で指定された値により生成され、生成されたジェネラル・レジスタ・アドレスにより指定されるジェネラル・レジスタの内容と、m で指定されたデータ・メモリの内容を加算し、結果をジェネラル・レジスタに格納します。

上述したジェネラル・レジスタのアドレス生成は表 6-1 に示す他の命令でも同様になります。

表 6-1 ジェネラル・レジスタとデータ・メモリ間の操作命令

命令群	命 令		オペレーション
加 算	ADD	r, m	(r) ← (r) + (m)
	ADDC	r, m	(r) ← (r) + (m) + CY
減 算	SUB	r, m	(r) ← (r) - (m)
	SUBC	r, m	(r) ← (r) - (m) - CY
論理演算	AND	r, m	(r) ← (r) ∧ (m)
	OR	r, m	(r) ← (r) ∨ (m)
	XOR	r, m	(r) ← (r) ⊕ (m)
転 送	LD	r, m	(r) ← (m)
	ST	m, r	(m) ← (r)
	MOV	@r, m	if MPE=1: (MP, (r)) ← (m) if MPE=0: (BANK, m _R , (r)) ← (m)
	MOV	m, @r	if MPE=1: (m) ← (MP, (r)) if MPE=0: (m) ← (BANK, m _R , (r))
シフト	RORC	r	CYを含めた右シフト

表 6-2 ジェネラル・レジスタとデータ・メモリのアドレス生成

命 令	アドレス内容	生 成 ア ド レ ス		
		バ ン ク	ロウ・アドレス	カラム・アドレス
ADD r, m	r で指定されるジェネラル・レジスタのアドレス	(0000B)	(RP)	r
	m で指定されるデータ・メモリ・アドレス	(BANK) (00××B)		m

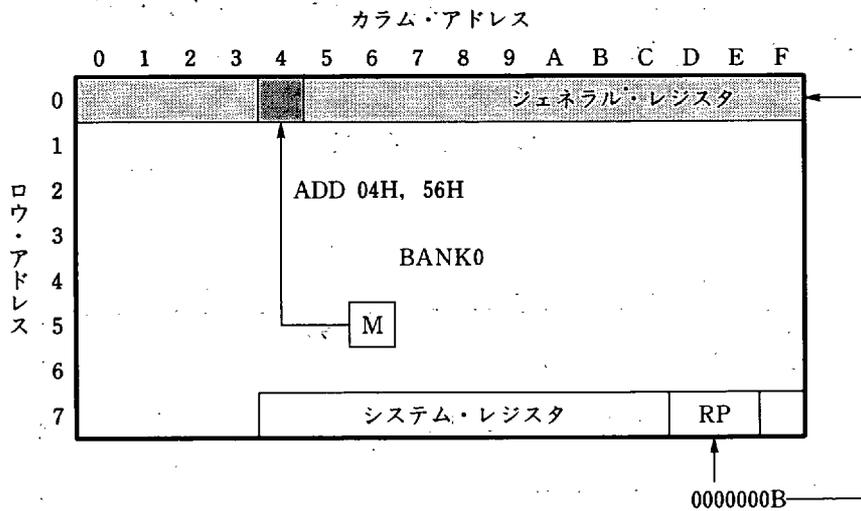
例1. BANK0 のとき

```

AND  RPL,  #0001B ; RP ← 0000000B ; ジェネラル・レジスタを
                        ; BANK0 のロウ・アドレス 0H に設定
ADD  04H,  56H ;
    
```

上の命令を実行すると、図 6-2 に示すようにジェネラル・レジスタである BANK0 の 04H 番地の内容とデータ・メモリの 56H 番地の内容を加算して結果をジェネラル・レジスタの 04H にストアします。

図 6-2 例1の実行



例2. BANK0, MPE=0のとき

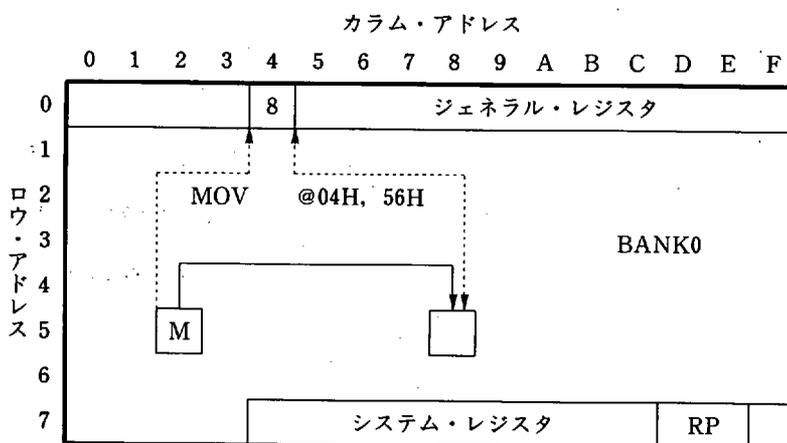
```

MOV  04H,  #8      ; 04H ← 8
AND  RPL,  #0001B ; RP ← 0000000B ; ジェネラル・レジスタを
                          ; BANK0 のロウ・アドレス 0H に設定
MOV  @04H, 52H
    
```

上の命令を実行すると、データ・メモリ 52H 番地の内容を 58H 番地に転送します。すなわち、“MOV @r, m” 命令はジェネラル・レジスタ間接転送と呼ばれ、r で指定されるジェネラル・レジスタの内容（上記では 8）をカラム・アドレスとし、m で指定されるロウ・アドレス（上記では 5）をロウ・アドレスとするデータ・メモリ、すなわちアドレス 58H のデータ・メモリとなります（図 6-3 参照）。

ジェネラル・レジスタ間接転送については“8.5 インデクス・レジスタ (IX) とデータ・メモリ・ロウ・アドレス・ポインタ (MP)” の項も参照してください。

図 6-3 例2の実行



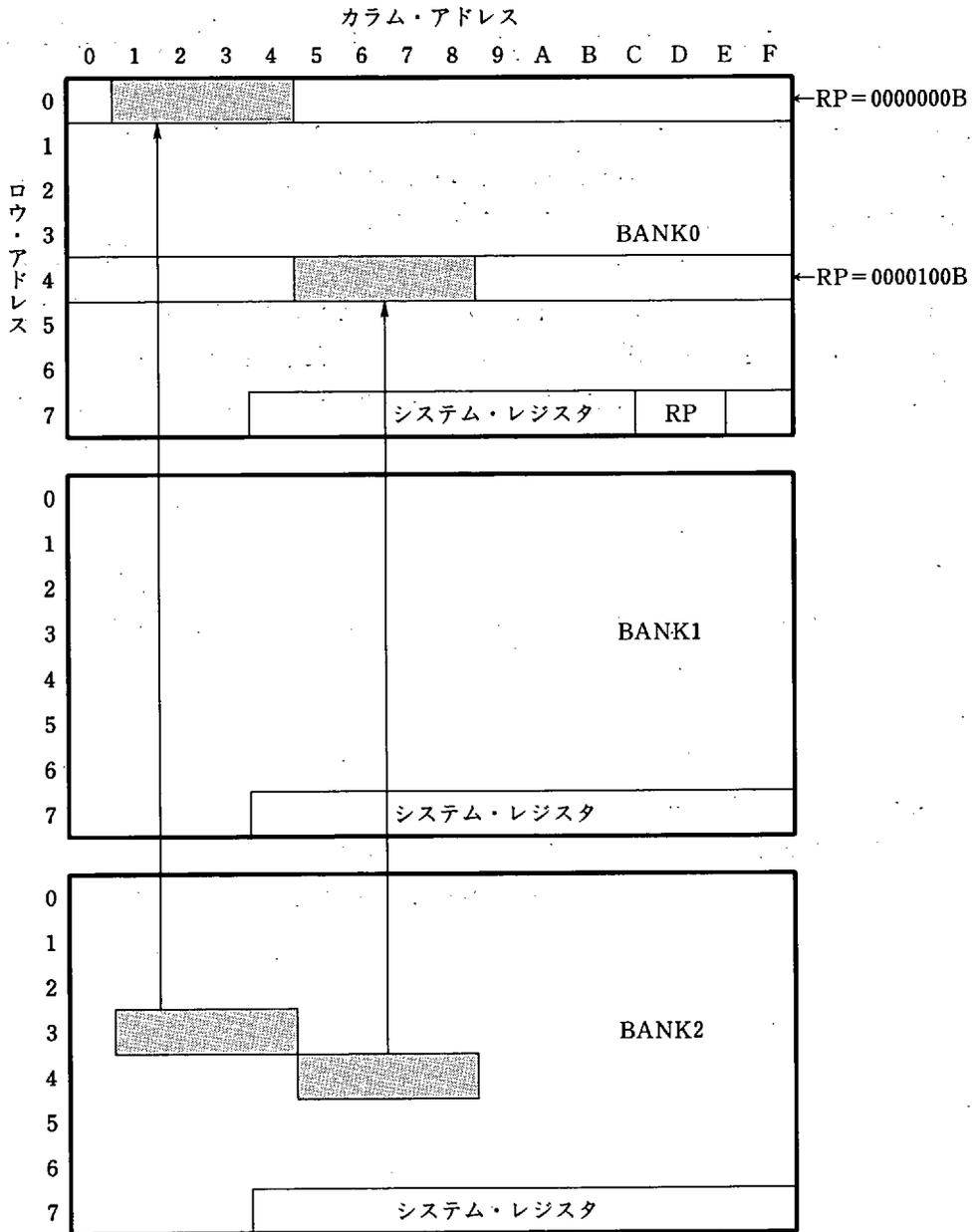
例3.

```

AND  RPL,  #0000B ; RP ← 0000000B ; ジェネラル・レジスタを
                          ; BANK0 のロウ・アドレス 0 に設定
MOV  BANK, #0010B ; BANK2
LD   01H, 31H
LD   02H, 32H
LD   03H, 33H
LD   04H, 34H
OR   RPL,  #1000B ; RP ← 0000100B ; ジェネラル・レジスタを
                          ; BANK0 のロウ・アドレス 4H に設定
LD   05H, 45H
LD   06H, 46H
LD   07H, 47H
LD   08H, 48H
    
```

例3は図6-4に示すように BANK2にある8ワードのデータを BANK0のデータ・メモリに4ワードずつ転送するプログラムです。このとき、ジェネラル・レジスタが固定で、たとえば BANK0のロウ・アドレス0にしか存在しないような場合、一度8ワードすべてをレジスタに転送したあとに再度データ・メモリに格納する命令が必要となります。しかし、この例のようにジェネラル・レジスタ・ポインタでジェネラル・レジスタのロウ・アドレスを替えてやれば格納命令だけで動作が終了できることとなります。

図6-4 例3の実行



6.4 ジェネラル・レジスタ使用時の注意

ジェネラル・レジスタ使用時の注意を以下の例を基に説明します。

例

```
AND RPL, #0001B ; RP ← 0000010B
OR RPL, #0100B ;
MOV BANK, #0000B ; BANK0
LD 04H, 32H
```

上の命令を実行すると、ジェネラル・レジスタの BANK0 の 24H 番地にデータ・メモリの BANK0 の 32H の内容をロードします。

すなわち、LD r, m 命令において、上記の場合、ジェネラル・レジスタが BANK0 のロウ・アドレス 2H であるため、r で指定されるジェネラル・レジスタのアドレスは BANK0 の 24H 番地となり、m で指定されるデータ・メモリのアドレスは BANK0 の 32H 番地となるためです (図 6-5 参照)。

ここで注意する点はアセンブラを使用するとき r で指定する値を実際のデータ・メモリ・アドレスたとえば 24H と記述したときです。この場合でも r は下位 4 ビットしか必要でないためロウ・アドレスである 2H はアセンブラでは無視されます。したがって LD 24H, 32H 命令を実行しても上記と同様の結果が得られます。

また、アセンブラを使用するうえでは、下記に示すように命令のオペランドにジェネラル・レジスタ・アドレスを直接指定するとエラーが発生します。

エラーとなる場合

```
LD 04H, 32H ; ジェネラル・レジスタ・アドレスを 04H で記述する。
```

通常用いる方法

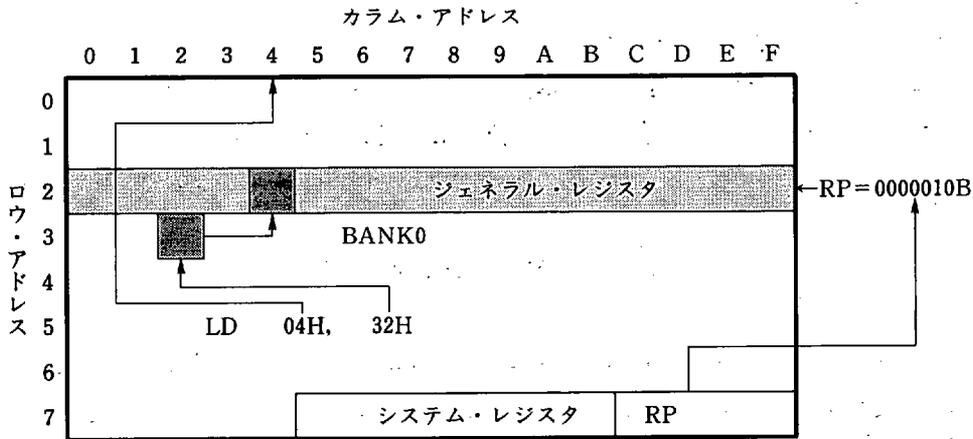
```
R1 MEM 0.04H ;
M1 MEM 0.32H ; ① R1, M1 をメモリ型として
LD R1, M1 ; それぞれ BANK0 の 04H と 32H を割り当てる。
```

このとき、

```
R2 MEM 0.34H
M1 MEM 0.32H
LD R2, M1
```

を実行しても、R1 と R2 は同一カラム・アドレスであるため ① と同様の動作になります。

図 6-5 例の実行



また、ジェネラル・レジスタを使用するうえでもう1つ注意する点は、ジェネラル・レジスタとイミューディエト・データとの演算命令はないことです。すなわち、ジェネラル・レジスタに指定されているデータ・メモリとイミューディエト・データとの演算命令を行うためには、そのデータ・メモリをジェネラル・レジスタではなく、データ・メモリとして扱う必要があります。

7. ALU (Arithmetic Logic Unit) ブロック

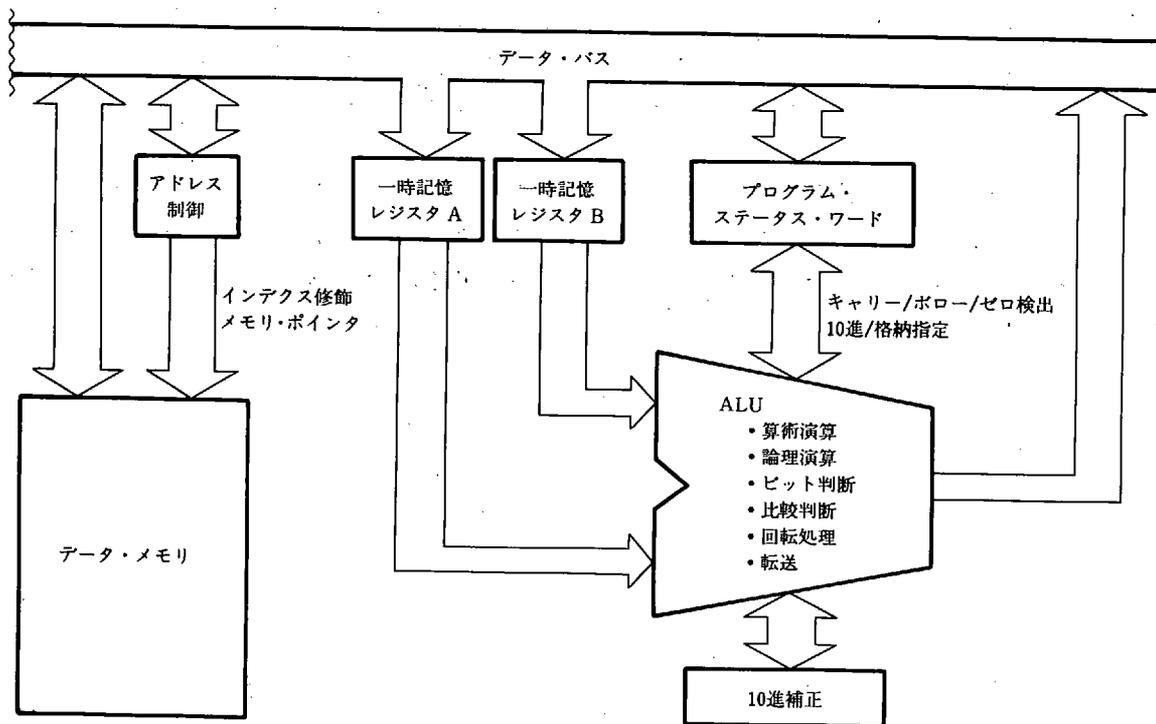
7.1 ALU ブロック概要

図 7-1 に、ALU ブロックの概略を示します。

図 7-1 に示すように、ALU ブロックは、ALU 本体、一時記憶レジスタ A, B, プログラム・ステータス・ワード、10進補正回路およびデータ・メモリ・アドレス制御回路から構成されています。

ALU は、データ・メモリ上の 4 ビット・データの演算、判断、比較、回転および転送を行います。

図 7-1 ALU ブロックの概略



7.2 各ブロックの構成と機能

7.2.1 ALU 本体

ALU 本体は、プログラムにより指定された命令により、4ビットの算術演算、論理演算、ビット判断、比較判断、回転処理および転送を行います。

7.2.2 一時記憶レジスタ A および B

一時記憶レジスタ A および B は、4ビットのデータを一時的に蓄わえておくレジスタです。
このレジスタは、命令実行時に自動的に使用され、プログラムで制御することはできません。

7.2.3 プログラム・ステータス・ワード

プログラム・ステータス・ワードは、ALU の動作制御、および状態を格納します。

プログラム・ステータス・ワードについては、“8.7 プログラム・ステータス・ワード (PSWORD)” を参照してください。

7.2.4 10進補正回路

算術演算時に、プログラム・ステータス・ワードの BCD フラグが“1”にセットされていると、10進補正回路により算術演算結果を10進数に変換します。

7.2.5 アドレス制御回路

アドレス制御回路は、データ・メモリのアドレスを指定します。

このとき、インデックス・レジスタやデータ・メモリ・ロウ・アドレス・ポインタによるアドレスの修飾も制御します。

7.3 ALU 処理命令一覧

表 7-1 に、各命令実行時の ALU 動作一覧を示します。

表 7-2 に、インデックス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタによる、データ・メモリ・アドレスの修飾を示します。

表 7-3 に、10進演算時の10進補正データを示します。

表 7-1 ALU 処理命令動作一覧

ALU機能	命 令		プログラム・ステータス・ワード (PSWORD) による動作の違い				アドレス修飾										
			BCD フラグの値	CMP フラグの値	演算動作	CY フラグの動作	Z フラグの動作	インデックス	メモリ・ポインタ								
加算	ADD	r, m	0	0	2進演算 結果を格納する	キャリーまたはボローでセット 発生しなければリセット	演算の結果 0000B でセット 0000B 以外はりセット										
		m, #n4					0	1	演算の結果 0000B で状態保持 0000B 以外はりセット								
減算	SUB	r, m	1	0	10進演算 結果を格納する	キャリーまたはボローでセット 発生しなければリセット			演算の結果 0000B でセット 0000B 以外はりセット								
		m, #n4					1	1	演算の結果 0000B で状態保持 0000B 以外はりセット								
論理演算	OR	r, m	任意 (保持)	任意 (保持)	変わらない	以前の状態を保持			以前の状態を保持								
		m, #n4					任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持						
		r, m									任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持		
AND	r, m	任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持											
	m, #n4					任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持							
XOR	r, m	任意 (保持)	任意 (保持)	変わらない	以前の状態を保持					以前の状態を保持							
	m, #n4					任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持							
判断	SKT	m, #n	任意 (保持)	任意 (リセット)	変わらない					以前の状態を保持	以前の状態を保持						
	SKF	m, #n				任意 (保持)	任意 (保持)	変わらない	以前の状態を保持		以前の状態を保持						
比較	SKE	m, #n4	任意 (保持)	任意 (保持)	変わらない					以前の状態を保持	以前の状態を保持						
	SKNE	m, #n4				任意 (保持)	任意 (保持)	変わらない	以前の状態を保持		以前の状態を保持						
	SKGE	m, #n4									任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持		
	SKLT	m, #n4													任意 (保持)	任意 (保持)	変わらない
転送	LD	r, m	任意 (保持)	任意 (保持)	変わらない					以前の状態を保持							
		m, #n4				任意 (保持)	任意 (保持)	変わらない	以前の状態を保持								
	ST	m, r									任意 (保持)	任意 (保持)	変わらない	以前の状態を保持			
		m, @r													任意 (保持)	任意 (保持)	変わらない
MOV	@r, m	任意 (保持)	任意 (保持)	変わらない	以前の状態を保持					以前の状態を保持							
	m, @r					任意 (保持)	任意 (保持)	変わらない	以前の状態を保持	以前の状態を保持							
回 転	RORC									r	任意 (保持)	任意 (保持)	変わらない	ジェネラルレジスタの b ₀ の値			

表7-2 インデクス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタによるデータ・メモリ・アドレスの修飾および間接転送アドレスの修飾

IXE	MPE	rで指定されるジェネラル・レジスタ・アドレス						mで指定されるデータ・メモリ・アドレス						@rで指定される間接転送アドレス											
		バンク				ロウ・アドレス		バンク				ロウ・アドレス		バンク		ロウ・アドレス		バンク		ロウ・アドレス					
		b ₃	b ₂	b ₁	b ₀	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₂	b ₁	b ₀	b ₃	b ₂
0	0	← RP				r		← BANK				m		← BANK				m _R		(r)					
0	1					同上						同上		← MP				(r)							
1	0					同上		← BANK				m		← BANK				m _R		(r)					
								Logical IX				OR		Logical IXH, IXM				OR		(r)					
1	1					同上						同上		← MP				(r)							

- BANK : バンク・レジスタ
- IX : インデクス・レジスタ
- IXE : インデクス・イネーブル・フラグ
- IXH : インデクス・レジスタのビット10-ビット8
- IXM : インデクス・レジスタのビット7-ビット4
- IXL : インデクス・レジスタのビット3-ビット0
- m : m_R, m_Cで示されるデータ・メモリ・アドレス
- m_R : データ・メモリ・ロウ・アドレス (上位)
- m_C : データ・メモリ・カラム・アドレス (下位)
- MP : データ・メモリ・ロウ・アドレス・ポインタ
- MPE : メモリ・ポインタ・イネーブル・フラグ
- r : ジェネラル・レジスタ・カラム・アドレス
- RP : ジェネラル・レジスタ・ポインタ
- (×) : ×でアドレスされる内容

表 7-3 10進補正データ

演算結果	16進加算		10進加算	
	CY	演算結果	CY	演算結果
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	0000B
11	0	1011B	1	0001B
12	0	1100B	1	0010B
13	0	1101B	1	0011B
14	0	1110B	1	0100B
15	0	1111B	1	0101B
16	1	0000B	1	0110B
17	1	0001B	1	0111B
18	1	0010B	1	1000B
19	1	0011B	1	1001B
20	1	0100B	1	1110B
21	1	0101B	1	1111B
22	1	0110B	1	1100B
23	1	0111B	1	1101B
24	1	1000B	1	1110B
25	1	1001B	1	1111B
26	1	1010B	1	1100B
27	1	1011B	1	1101B
28	1	1100B	1	1010B
29	1	1101B	1	1011B
30	1	1110B	1	1100B
31	1	1111B	1	1101B

演算結果	16進減算		10進減算	
	CY	演算結果	CY	演算結果
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	1100B
11	0	1011B	1	1101B
12	0	1100B	1	1110B
13	0	1101B	1	1111B
14	0	1110B	1	1100B
15	0	1111B	1	1101B
-16	1	0000B	1	1110B
-15	1	0001B	1	1111B
-14	1	0010B	1	1100B
-13	1	0011B	1	1101B
-12	1	0100B	1	1110B
-11	1	0101B	1	1111B
-10	1	0110B	1	0000B
-9	1	0111B	1	0001B
-8	1	1000B	1	0010B
-7	1	1001B	1	0011B
-6	1	1010B	1	0100B
-5	1	1011B	1	0101B
-4	1	1100B	1	0110B
-3	1	1101B	1	0111B
-2	1	1110B	1	1000B
-1	1	1111B	1	1001B

備考 の部分は、10進補正が正しく行われません。

7.4 ALU 使用時の注意

7.4.1 プログラム・ステータス・ワードへの演算使用時の注意

プログラム・ステータス・ワードに対して算術演算を行うと、プログラム・ステータス・ワードには算術演算の結果が格納されます。

プログラム・ステータス・ワードの中のCYフラグおよびZフラグは、通常、算術演算の結果によりセットまたはリセットされますが、プログラム・ステータス・ワード自身に算術演算が行われると、算術演算結果が格納されてしまい、キャリー、ポローおよびゼロの判定ができなくなります。

ただし、CMPフラグがセットされているときは、算術演算の結果が格納されないため、CYフラグおよびZフラグは通常通りセットまたはリセットされます。

7.4.2 10進演算使用時の注意

10進演算は、演算結果が次の範囲になる場合にかぎり実行できます。

- (1) 加算の結果が、10進で0~19であること
- (2) 減算の結果が、10進で0~9または-10~-1であること

この範囲以外で10進演算を行うと、CYフラグがセットされ、演算結果は1010B (0AH) 以上の値になります。

8.1 アドレス・レジスタ (AR)

プログラム・メモリのアドレスを指定するためのレジスタで、74H-77H に配置されています。アドレス・レジスタを使用する命令には、間接分岐命令 (BR @AR, CALL @AR), テーブル参照命令 (MOVT) およびスタック操作命令 (PUSH, POP) があります。

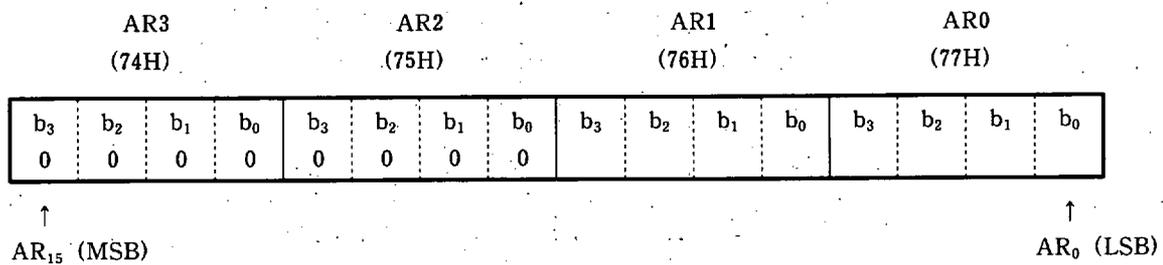
間接分岐はアドレス・レジスタの内容で指定されるプログラム・メモリの番地へ分岐するもので“BR @AR”と“CALL @AR”があります。

テーブル参照は、“MOVT”命令で行い、アドレス・レジスタで指定されたプログラム・メモリの番地の内容をデータ・メモリの DBF (BANK0 の 0DH-0FH) に転送します。

スタック操作は、“PUSH”, “POP”命令によって行います。“PUSH”命令は、アドレス・レジスタの内容を、そのときのスタック・ポインタによって指定されるスタックへ格納し、そしてスタック・ポインタの内容を-1します。“POP”命令は、スタック・ポインタの内容を+1し、そのときのスタック・ポインタによって指定されるスタックの内容をアドレス・レジスタへロードします。

μPD17062 は AR3, AR2 は 0 に固定されています。したがってアドレス・レジスタによって指定できるプログラム・アドレスは 0000H-00FFH の 256 ステップです。

図 8-2 アドレス・レジスタの構成



8.2 ウィンドウ・レジスタ (WR)

ウィンドウ・レジスタは、システム・レジスタの 78H 番地にマッピングされた 4 ビットのレジスタで、後述するレジスタ・ファイル (RF) とのデータ転送に使用します。レジスタ・ファイル上の各レジスタのデータ操作はすべてウィンドウ・レジスタを介して行います。

ウィンドウ・レジスタとレジスタ・ファイルとのデータ転送は専用命令である“PEEK WR, rf”, “POKE rf, WR”命令により行います。

8.3 バンク・レジスタ (BANK)

バンク・レジスタは、データ・メモリのバンク指定を行います。

バンク・レジスタは、リセット時には BANK0 になります。また 79H 番地の上位 2 ビットは常に “0” です。

データ・メモリはバンク・レジスタにより 3 つのバンクに分けられており、データ・メモリ操作命令を実行すると、バンク・レジスタで指定されるバンク内のデータ・メモリを操作します。

たとえば現在 BANK0 であるときに BANK1 のデータ・メモリを操作するためにはバンク・レジスタによってバンクを BANK1 に切り替えておく必要があります。

ただし、データ・メモリの 74H-7FH に配置されているシステム・レジスタにはバンクの概念がなく、すべてのバンクにおいて 74H-7FH 番地には同一のシステム・レジスタが存在しています。これは BANK1 で “MOV 78H, #0” という命令を行っても、BANK2 で “MOV 78H, #0” という命令を行っても結果としてはシステム・レジスタである 78H 番地に 0 を書き込むことになります。つまりシステム・レジスタを操作するかぎりにおいては、バンクの概念は必要ありません。

割り込みが受け付けられると BANK はセーブされます。

表 8-1 データ・メモリのバンク指定

バンク・レジスタ (BANK)				データ・メモリの バンク
b ₃	b ₂	b ₁	b ₀	
0	0	0	0	BANK0
0	0	0	1	BANK1
0	0	1	0	BANK2
0	0	1	1	設定禁止

8.4 メモリ・ポインタ・イネーブル・フラグ (MPE)

“MOV @r, M”, “MOV M, @r” 命令を実行するときのロウ・アドレスを MPL で指定するか、同一ロウ・アドレスで実行するかを指定します。MPE がセットされるとロウ・アドレスは、MPL で指定され、リセットすると同一ロウ・アドレスとなります。

ただし MPL で指定するのは、現在指定されている同一バンクのロウ・アドレスです。

8.5 インデクス・レジスタ (IX) とデータ・メモリ・ロウ・アドレス・ポインタ (MP)

8.5.1 インデクス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタの構成

図8-1に示すようにインデクス・レジスタはシステム・レジスタの7AH (IXH) の下位3ビットと7BH, 7CH (IXM, IXL) の計11ビットからなり、データ・メモリのアドレスを間接指定するときに使用するレジスタです。

またデータ・メモリ・ロウ・アドレス・ポインタは7AH (MPH) の下位3ビットと7BH (MPL) の計7ビットで構成されています。

したがってインデクス・レジスタの上位7ビットとデータ・メモリ・ロウ・アドレス・ポインタは共用していることとなります。

また μ PD17062ではインデクス・レジスタの上位4ビット、つまりデータ・メモリ・ロウ・アドレス・ポインタの上位4ビット (7AHの b_2 - b_0 , 7BHの b_3) は“0”に固定されています。

8.5.2 インデクス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタの機能

インデクス・レジスタは、インデクス・イネーブル・フラグ (IXE) がセット(1)されているときにデータ・メモリ操作命令を実行すると、命令で指定されたデータ・メモリのバンクとアドレスをインデクス・レジスタの内容とOR演算し、その演算結果(実アドレスという)で指定されるアドレスのデータ・メモリに対して命令を実行します。

データ・メモリ・ロウ・アドレス・ポインタはメモリ・ポインタ・イネーブル・フラグがセット(1)されているときにジェネラル・レジスタ間接転送命令("MOV @r, m" および "MOV m, @r")を実行すると、ジェネラル・レジスタで指定される間接アドレスのバンクとロウ・アドレスをデータ・メモリ・ロウ・アドレス・ポインタの値として、命令を実行します。

表8-2にインデクス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタによるデータ・メモリおよび間接アドレスの修飾を示します。

インデクス・レジスタおよびデータ・メモリ・ロウ・アドレス・ポインタは、すべてのデータ・メモリを修飾の対象にできます。

また、次の命令はインデクス・レジスタの修飾対象となりません。

```

INC    AR
INC    IX
MOV    DBF, @AR
PUSH   AR
POP    AR
PEEK   WR, rf
POKE   rf, WR
GET    DBF, p
PUT    p, DBF
BR     addr
BR     @AR
RORC   r
CALL   addr
CALL   @AR
RET
RETSK
RETI
EI
DI
STOP   0
HALT   h
NOP

```

表 8-2 インデクス・レジスタとデータ・メモリ・ロウ・アドレス・ポインタによるデータ・メモリ・アドレスの修飾

IXE	MPE	r で指定されるジェネラル・レジスタ・アドレス						m で指定されるデータ・メモリ・アドレス						@r で指定される間接転送アドレス																	
		バンク		ロウ・アドレス		カラム・アドレス		バンク		ロウ・アドレス		カラム・アドレス		バンク		ロウ・アドレス		カラム・アドレス													
		b ₃	b ₂	b ₁	b ₀	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	b ₂	b ₁	b ₀	b ₃	b ₂	b ₁	b ₀	
0	0	(RP)						r						(BANK)		m						(BANK)		m _R		(R)					
0	1	同上						同上						(MP)								(R)									
1	0	同上						(BANK)		m						(BANK)		m _R						(R)							
								Logical OR						(IX)		Logical OR						(IXH)		(IXM)							
1	1	同上						同上						(MP)								(R)									
アドレス修飾される命令群																															
加 減 算	ADD							r						m																	
	ADDC							r						m																	
	SUB													m, #n4																	
	SUBC													m, #n4																	
比 較	AND													m																	
	OR							r						m																	
	XOR													m, #n4																	
論 理	SKE																														
	SKGE																														
	SKLT													m, #n4																	
	SKNE																														
判 断	SKT													m, #n																	
	SKF													m, #n																	
転 送	LD													m																	
	ST							r						m																	
	MOV													m, #n4																	
														@r						m						間接転送アドレス					

M ; データ・メモリ・アドレス
 (M) ; データ・メモリ・アドレスの内容
 m ; バンクを除くデータ・メモリ・アドレス
 m_R ; データ・メモリ・ロウ・アドレス
 R ; ジェネラル・レジスタ・アドレス
 (R) ; ジェネラル・レジスタ・アドレスの内容
 r ; ジェネラル・レジスタ・カラム・アドレス
 RP ; ジェネラル・レジスタ・ポインタ
 (RP) ; ジェネラル・レジスタ・ポインタの内容

BANK ; バンク・レジスタ
 (BANK) ; バンク・レジスタの内容
 IX ; インデクス・レジスタ
 (IX) ; インデクス・レジスタの内容
 IXH ; インデクス・レジスタのビット b₁₀-b₈
 IXM ; インデクス・レジスタのビット b₇-b₄
 IXL ; インデクス・レジスタのビット b₃-b₀
 MP ; データ・メモリ・ロウ・アドレス・ポインタ
 (MP) ; データ・メモリ・ロウ・アドレス・ポインタの内容

8.5.3 MPE=0, IXE=0 のとき (データ・メモリ修飾なし)

表 8-2 に示すようにデータ・メモリ・アドレスはインデックス・レジスタと、データ・メモリ・ロウ・アドレス・ポインタの影響を受けません。

例 1. BANK0 でジェネラル・レジスタがロウ・アドレス 0 にあるとき

```
ADD 03H, 11H
```

上の命令を実行すると、ジェネラル・レジスタ 03H とデータ・メモリ 11H の内容を加算し、結果をジェネラル・レジスタ 03H に格納します (図 8-3 例 1 参照)。

2. BANK0 でジェネラル・レジスタがロウ・アドレス 0 にあるとき

```
MOV 05H, #8 ; 05H ← 8
MOV @05H, 34H ; レジスタ間接転送
```

上の命令を実行するとデータ・メモリ 34H 番地の内容が 38H 番地へ転送されます。すなわち MOV @r, m 命令は、データ・メモリ m の内容を m と同一ロウ・アドレス (上記では 3) でカラム・アドレスがジェネラル・レジスタ r の内容 (上記では 8) で指定されるアドレス (上記では 38H となる) に転送します (図 8-3 例 2 参照)。

3. BANK0 でジェネラル・レジスタがロウ・アドレス 0 にあるとき

```
MOV 0BH, #0EH ; 0BH ← 0EH
MOV 34H @0BH ; レジスタ間接転送
```

上の命令を実行すると、データ・メモリ 34H 番地へ、3EH 番地の内容が転送されます。すなわち、MOV m, @r 命令はデータ・メモリ m と同一ロウ・アドレス (上記では 3) でカラム・アドレスがジェネラル・レジスタ r の内容 (上記は 0EH) で指定されるアドレス (上記では 3EH となる) の内容を m に転送します (図 8-3 例 3 参照)。
これは、例 2 と比べると転送するデータ・メモリ・アドレスのソース (転送元) とデスティネーション (転送先) が、入れ替わったことになります。

図 8-3 MPE=0, IXE=0のときのジェネラル・レジスタ間接転送

		コラム・アドレス																		
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
ロウ・アドレス	0	例 1. ADD03H, 11H								8	← デスティネーションのコラム・アドレスを指定						E	← ソースのコラム・アドレスを指定		← ジェネラル・レジスタ
	1																			
	2					例 2. MOV @05H, 34H														
	3																			
	4																			
	5																			
	6																			
	7																			

例 2 のアドレス生成

MOV @r, m
 ↓ ↓
 05H 34H

	バンク	ロウ・アドレス	コラム・アドレス
R	0	0	5
M	0	3	4
(@r)	0	3	8
		← M と同一 →	
			← R の内容 →

8.5.4 MPE=1, IXE=0のとき (ななめ間接転送)

表 8-2 に示すようにジェネラル・レジスタ間接転送命令を行ったときのみ、ジェネラル・レジスタで指定される間接側のデータ・メモリ・アドレスのバンクとロウ・アドレスがデータ・メモリ・ロウ・アドレス・ポインタの値になります。

例 1. BANK0 でジェネラル・レジスタがロウ・アドレス 0 のとき

```
MOV  MPL,  #0101B  ; MP ← 00101B
MOV  MPH,  #1000B  ; MPE ← 1
MOV  05H,  #8      ; 05H ← 8
MOV  @05H, 34H    ; レジスタ間接転送
```

上の命令を実行するとデータ・メモリ 34H 番地の内容が、データ・メモリ 58H 番地に転送されます。すなわち、MPE=1 のときの MOV @r, m 命令は、データ・メモリ m の内容をバンクとロウ・アドレスがデータ・メモリ・ロウ・アドレス・ポインタの値 (上記では BANK0, ロウ・アドレス 5) で、カラム・アドレスがジェネラル・レジスタ r の内容 (上記では 8) で指定される (上記では BANK0 の 58H となる) データ・メモリに転送します (図 8-4 例 1 参照)。

これは MPE=0 のとき (8.5.3 例 2) と比べると、ジェネラル・レジスタで指定される間接側のデータ・メモリ・アドレスのバンクとロウ・アドレスをデータ・メモリ・ロウ・アドレス・ポインタで指定できることとなります (8.5.3 例 2 では間接側のバンクとロウ・アドレスは m と同じになる)。

したがって、MPE=1 とすることによりジェネラル・レジスタ間接転送をななめに行うことが可能になります。

また、MOV m, @r 命令も同様に以下の例 2 のようになります。

2. BANK0 でジェネラル・レジスタがロウ・アドレス 0 のとき

```
MOV  MPL,  #0101B  ; MP ← 00101B
MOV  MPH,  #1000B  ; MPE ← 1
MOV  0BH,  #0EH    ; 0BH ← 0EH
MOV  3AH,  @05H
```

(図 8-4 例 2 参照)

図 8-4 MPE = 1, IXE = 0 のときのジェネラル・レジスタ間接転送

ジェネラル・レジスタ	コラム・アドレス															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
→ 0						8						E				
1																
2																
3																
4																
5																
6																
7																

例 1. MOV @05H, 34H
 アステリッシュの位置のカラム・アドレスを指定
 ソースのカラム・アドレスを指定

例 2. MOV 3AH, @0BH
 ソースのカラム・アドレスを指定
 アステリッシュの位置のカラム・アドレスを指定

バンクとロウ・アドレスはデータ・メモリ・ロウ・アドレス・ポインタの値 000101B となる

例 1 のアドレス生成

MOV @r, m
 ↓ ↓
 05H 34H
 MP = 00101B

	バンク	ロウ・アドレス	コラム・アドレス
R	0	0	5
M	0	3	4
(@r)	0 0 0 0	1 0 1	8
	← MP の値 →		← R の内容 →

8.5.5 MPE=0, IXE=1 のとき (インデックス修飾)

表 8-2 に示すようにデータ・メモリ操作命令を行うと、命令で直接指定されたデータ・メモリのバンクとアドレスが、インデックス・レジスタと OR 演算され、その演算結果 (実アドレスという) で指定されるデータ・メモリ・アドレスに対して命令が実行されます。

例 1. BANK0 でジェネラル・レジスタがロウ・アドレス 0 のとき

```
MOV  IXL,  #0010B ; IX ← 000000010B
MOV  IXM,  #0000B ; MPE ← 0
MOV  IXH,  #0000B ;
OR   PSW,  #0001B ; IXE ← 1
ADD  03H,  11H
```

上の命令を実行するとデータ・メモリ 13H 番地の内容とジェネラル・レジスタ 03H 番地の内容を加算し、結果をジェネラル・レジスタの 03H へ格納します。

すなわち、ADD r, m 命令を実行すると m で指定されたアドレス (上記では BANK0 の 11H) と、インデックス・レジスタの値 (上記では 000000010B) を OR 演算し、その結果を実アドレス (上記では BANK 0 の 13H) として、実アドレスに対して命令を実行します (図 8-5 参照)。

これは IXE=0 のとき (8.5.3 の例 1) と比べると命令で直接指定されるデータ・メモリのアドレスがインデックス・レジスタにより修飾 (OR 演算) されたことになります。

2. すべてのバンクのデータ・メモリを 0 にクリアする

```
MOV  IXL,  #0 ;
MOV  IXM,  #0 ; IX ← 0
MOV  IXH,  #0 ;
LOOP:
OR   PSW,  #0001B ; IXE ← 1
MOV  00H,  #0 ; IX で指定されたデータ・メモリを 0 に
INC  IX ; IX ← IX + 1
AND  PSW,  #1110B ; IXE ← 0 : IXE は 7FH 番地のため IX で修飾されない
SKT  IXM,  #0111B ; ロウ・アドレス 7 になったか
BR   LOOP ; 7 でなければ LOOP
ADD  IXM,  #1 ; ロウ・アドレス 7 をクリアしないで次のバンクを指定
ADDC IXH,  #0 ;
SKF  IXM,  #1000B ; BANK2 までクリアしたか
SKT  IXH,  #0001B ;
BR   LOOP ; クリアしなければ LOOP
```

図 8-5 IXE=1のときのデータ・メモリ・アドレスの修飾

		カラム・アドレス							
		0	1	2	3	4	5	6	
ロウ・アドレス	0				R				← ジェネラル・レジスタ
	1		M	IXにより 指定		ADD r, m			
	2								
	3								
	4								

8.6 ジェネラル・レジスタ・ポインタ (RP)

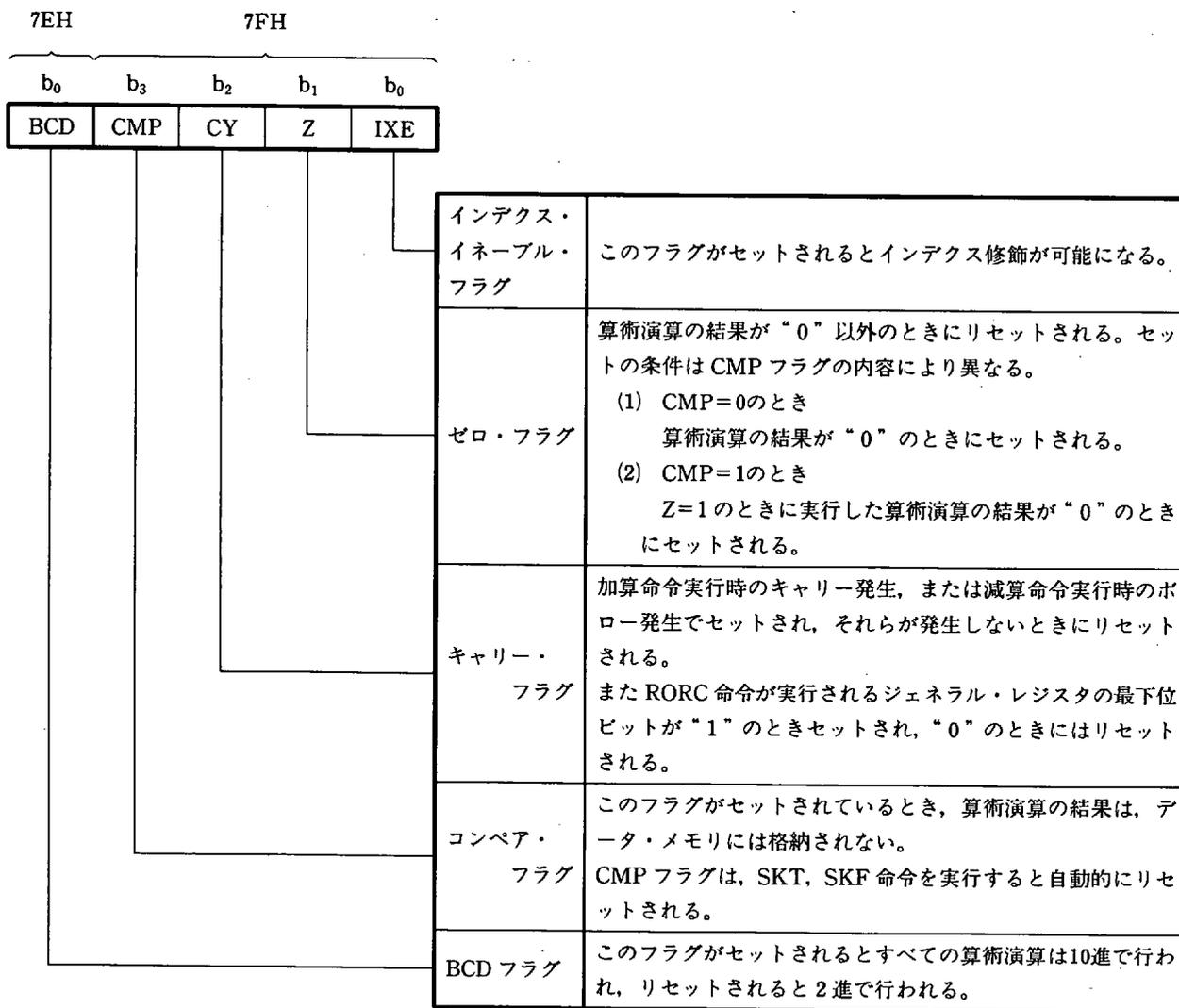
ジェネラル・レジスタのバンク，ロウ・アドレスを指定するポインタです。

ただし μPD17062 では RPH は 0 に固定しているので指定できるのは RPL (3ビット) のみです。つまりレジスタ・ポインタとしては 0-7 が指定可能です。したがって μPD17062 ではジェネラル・レジスタのロウ・アドレスは BANK0 内で自由に指定できます。

8.7 プログラム・ステータス・ワード (PSWORD)

プログラム・ステータス・ワードは CPU 内の ALU による演算結果の状態を示すフラグおよび ALU の機能を修飾する 5 ビットのフラグで構成されます。PSWORD には、BCD(Binary Coded Decimal) フラグ、CMP(Compare) フラグ、CY (Carry) フラグ、Z (Zero) フラグ、IXE (Index Enable) フラグがあります。これらのフラグの機能を図 8-6 に示します。

図 8-6 PSWORDの構成



9. レジスタ・ファイル (RF)

レジスタ・ファイルは、主としてCPUの周辺回路の制御を行うためのレジスタの集合です。128ワード×4ビットの容量を持っています。ただし、上位64ニブル(00H-3FH)には実際に周辺回路アドレスが割り当てられています(コントロール・レジスタ・ファイル)が、下位64ニブル(40H-7FH)は、データ・メモリの現在選択されているバンクの40H-7FH番地が割り当てられています。

すなわちデータ・メモリの各バンクの40H-7FHは、データ・メモリ・アドレス空間とレジスタ・ファイル・アドレス空間の両方に属していることになります。

アセンブラ上ではコントロール・レジスタ・ファイルは80H-BFHに割り当てられます。

図9-1 コントロール・レジスタの構成 (1/2)

カラム・アドレス ロウ・アドレス	項目	0	1	2	3	4	5	6	7
0 (8)注	名称	IDCDMA イネーブル・ レジスタ	スタック・ポインタ (SP)						CE端子 レベル・ジャッジ・ レジスタ
	記号	I D C D M A E N 0 0 0 0	(S P 2) (S P 1) (S P 0)						C E 0 0 0
	Read/ Write	R/W	R/W						R
1 (9)注	名称		H _{SYNC} カウンタ・ ゲート・コントロール・ レジスタ	H _{SYNC} カウンタ・ ゲート・ジャッジ・ レジスタ	PLLレファレンス・ クロック・セレクト・ レジスタ		INT _{NC} モード・ セレクト・レジスタ		ベータ・タイム0 キャリー・FF ジャッジ・レジスタ
	記号		H S C G T 1 0	H S C G O S T T 0 0 0	P L L R F C K 3 P L L R F C K 2 P L L R F C K 1 P L L R F C K 0		I N T N C M D 2 I N T N C M D 1 I N T N C M D 0	B T M O C Y 0 0 0	
	Read/ Write		R/W	R	R/W		R/W		R
2 (A)注	名称		A/Dコンバータ・ コントロール・ レジスタ	PLLアンロック FF ジャッジ・レジスタ					ポートIC グループ I/O セレクト・レジスタ
	記号		A D C C H 2 A D C C H 1 A D C C H 0 A D C C M P 0 0 0 P L L U L						P I C G I O 0 0 0
	Read/ Write		R/W	R					R/W
3 (B)注	名称	IDC CROMバンク・ レジスタ	IDC イネーブル・ レジスタ	PLLアンロックFF センシビリティ・ セレクト・レジスタ			ポート1B ビットI/O セレクト・レジスタ	ポート0B ビットI/O セレクト・レジスタ	ポート0A ビットI/O セレクト・レジスタ
	記号	C R O M B N K 0 0 0	I D C E N 0 0 0	P L L U L S E N 1 P L L U L S E N 0			P 1 B I O 3 P 1 B I O 2 P 1 B I O 1 P 1 B I O 0	P 0 B I O 3 P 0 B I O 2 P 0 B I O 1 P 0 B I O 0	P 0 A B I O 3 P 0 A B I O 2 P 0 A B I O 1 P 0 A B I O 0
	Read/ Write	R/W	R/W	R/W			R/W	R/W	R/W

注 () 内は、アセンブラ (AS17K) を使用する際の番地です。

図9-1 コントロール・レジスタの構成 (2/2)

8		9				A	B	C	D	E	F							
シリアル I/O0 モード・セレクト・ レジスタ		タイマ0 クロック・セレクト・ レジスタ										割り込みレベル・ ジャッジ・レジスタ						
SIO0CH	SBIO0MS	SIO0TX	SIO0TX	BTM0ZX	BTM0CK2	BTM0CK1	BTM0CK0								INTVSYN0	INTNC		
R/W		R/W										R						
シリアル I/O0 ウェイト・コントロール・ レジスタ												割り込みエッジ 選択レジスタ						
SBACK	SIO0NT	SIO0WRQ1	SIO0WRQ0												IEGVSYN0	IEGNC		
R/W												R/W						
シリアル I/O0 ステータス・ ジャッジ・レジスタ												割り込み許可 レジスタ						
SIO0SF8	SIO0SF9	SBSTT	SBBSY												IPPSIO0	IPVSYN0	IPBTM0	IPNC
R												R/W						
シリアル I/O0 割り込みモード・ レジスタ		シリアル I/O0 クロック・セレクト・ レジスタ										割り込み要求 レジスタ						
0	0	SIO0IMD1	SIO0IMD0	0	0	SIO0CK1	SIO0CK0								IRQSI00	IRQVSYN0	IRQBTM0	IRQNC
R/W		R/W										R						

表 9-1 コントロール・レジスタの周辺ハードウェア制御機能一覧 (1/5)

周辺ハードウェア	コントロール・レジスタ				周辺ハードウェア制御機能				リセット時		
	名称	番地	Read/Write	記号 b3 b2 b1 b0	機能概略	設定値		パ ワ ー オ ン	S T O P	C E	
						0	1				
スタック	スタック・ポインタ (SP)	01H	R/W	0	0 固定						
				(SP2)	スタック・ポインタ (3ビット有効)						
				(SP1)							
				(SP0)							
タイマ	タイマ0 クロック・セレクト・レジスタ	09H	R/W	BTM0ZX	ゼロクロス回路の ON/OFF	動作しない	動作する				
				BTM0CK2	ベーシック・タイマ0 のベース・クロックの 設定 (内部/外部)	タイマ・キャリー FF セット用パルス 0: 10 Hz (100 ms, 内部) 1: 200 Hz (5 ms, 内部) 2: 10 Hz (100 ms, 内部) 3: 200 Hz (5 ms, 内部) 4: $f_{TMIN}/5$ Hz (外部) 5: 200 Hz (5 ms, 内部) 6: $f_{TMIN}/6$ Hz (外部) 7: 200 Hz (5 ms, 内部)					
				BTM0CK1		タイマ割り込み用パルス 0: 200 Hz (5 ms, 内部) 1: 10 Hz (100 ms, 内部) 2: 50 Hz (20 ms, 内部) 3: 50 Hz (20 ms, 内部) 4: 200 Hz (5 ms, 内部) 5: $f_{TMIN}/5$ Hz (外部) 6: 200 Hz (5 ms, 内部) 7: $f_{TMIN}/6$ Hz (外部)		0	0	保	
				BTM0CK0							
ベーシック・タイマ0 キャリー FF ジャッジ・レジスタ	17H	R	0	0 固定							
			0								
			0	BTM0CY	キャリー FF の状態を検出	リセット	セット			0	1
割り込み レベル・ジャッジ・レジスタ	0FH	R	0	0 固定							
			INTVSYN	V_{SYNC} 端子の状態を検出	ロウ・レベル	ハイ・レベル			0	0	0
			0	0 固定							
				INTNC	INT _{NC} 端子の状態を検出	ロウ・レベル	ハイ・レベル				
INT _{NC} モード・セレクト・レジスタ	15H	R/W	0	0 固定							
			INTNCMD2	INT _{NC} 端子の割り込み受け付けパルス幅の選択	0: エッジで受け付ける						
			INTNCMD1		1: 200 μs 2: 400 μs 3: 2 ms						
			INTNCMD0		4: 4 ms						

備考 保: 以前の状態を保持 不: 不定

表 9-1 コントロール・レジスタの周辺ハードウェア制御機能一覧 (2/5)

周辺ハードウェア	コントロール・レジスタ				周辺ハードウェア制御機能				リセット時				
	名称	番地	Read/Write	記号 b3 b2 b1 b0	機能概略	設定値		パ ワ ー オ ン	S T O P	C E			
						0	1						
割 り 込 み	割り込みエッジ 選択 レジスタ	1FH	R/W	0	0 固定								
				IEGVSYN	割り込み発行エッジの設定 (V _{SYN})	立ち上がりエッジ	立ち下がりエッジ	0	0	0			
				0	0 固定								
				IEGNC	割り込み発行エッジの設定 (INT _{NC})	立ち上がりエッジ	立ち下がりエッジ						
割 り 込 み 許 可	割り込み許可 レジスタ	2FH	R/W	IPSIO0	シリアル・インタフェース 0 } の割り 込み 許可を 設定	割り込み禁止	割り込み許可	0	1	1			
				IPVSYN							V _{SYN} 信号		
				IPBTM0							ベーシック・タイマ 0		
				IPNC							INT _{NC} 端子		
割 り 込 み 要 求	割り込み要求 レジスタ	3FH	R	IRQSIO0	シリアル・インタフェース 0 } の割り 込み 要求を 検出	割り込み 要求なし /処理中	割り込み 要求あり	0	0	0			
				IRQVSYN							V _{SYN} 信号		
				IRQBTM0							ベーシック・タイマ 0		
				IRQNC							INT _{NC} 端子		
端 子	CE 端子 レベル・ ジャッジ・ レジスタ	07H	R	0	0 固定			0	-	-			
				0									
				0									
				CE							CE 端子の状態を検出	ロウ・レベル	ハイ・レベル
P L L 周 波 数 シ ン セ イ ザ	PLL レファレンス・ クロック・ セレクト・ レジスタ	13H	R/W	PLLRFCK3	1 固定	2 : 6.25 kHz 3 : 12.5 kHz 6 : 25 kHz F : 動作停止 (ディスエーブル状態) 0, 1, 4, 5, 7 ~ E : 設定禁止		F	F	保			
				PLLRFCK2									
				PLLRFCK1									
				PLLRFCK0									
P L L 周 波 数 シ ン セ イ ザ	PLL アンロック FF ジャッジ・ レジスタ	22H	R	0	0 固定			0	保	保			
				0									
				0									
				PLLUL							アンロック FF の状態を検出	ロック状態	アンロック状態
P L L 周 波 数 シ ン セ イ ザ	PLL アンロック FF センシビリティ・ セレクト・ レジスタ	32H	R/W	0	0 固定			0	0	保			
				0									
				PLULSEN1							アンロック FF のセット遅延時 間を設定	0 0 1 1	1.25 3.5 0.25 ディス エーブ ル状態
				PLULSEN0								1.5 μS 3.75 μS 0.5 μS	0 1 0 1

備考 保：以前の状態を保持 不：不定

表 9-1 コントロール・レジスタの周辺ハードウェア制御機能一覧 (3/5)

周辺ハードウェア	コントロール・レジスタ				周辺ハードウェア制御機能			リセット時				
	名称	番地	Read/Write	記号 b3 b2 b1 b0	機能概略	設定値		パ ワ ー オ ン	S T O P	C E		
						0	1					
A/D コンバータ	A/D コンバータ・ コントロール・ レジスタ	21H	R/W	ADCCH2	A/D コンバータとして使用する端子を選択	0 : AD0	1 : AD1	0	0	0		
				ADCCH1		2 : AD2	3 : AD3					
				ADCCH0		4 : AD4	5 : AD5					
				ADCCMP		比較結果の検出					$V_{IN} < V_{REF}$	$V_{IN} > V_{REF}$
汎 用 ポ ー ト	ポート 1C グループ I/O セレクト・ レジスタ	27H	R/W	0	0 固定			0	0	0		
				0								
				0								
				PICGIO		ポート 1C の入出力設定 (グループ I/O)	入力				出力	
汎 用 ポ ー ト	ポート 1B ビット I/O セレクト・ レジスタ	35H	R/W	P1BBIO3	P1B ₃ 端子 P1B ₂ 端子 P1B ₁ 端子 P1B ₀ 端子	入力	出力	0	0	0		
				P1BBIO2								
				P1BBIO1								
				P1BBIO0								
汎 用 ポ ー ト	ポート 0B ビット I/O セレクト・ レジスタ	36H	R/W	P0BBIO3	P0B ₃ 端子 P0B ₂ 端子 P0B ₁ 端子 P0B ₀ 端子	入力	出力	0	0	0		
				P0BBIO2								
				P0BBIO1								
				P0BBIO0								
汎 用 ポ ー ト	ポート 0A ビット I/O セレクト・ レジスタ	37H	R/W	P0ABIO3	P0A ₃ 端子 P0A ₂ 端子 P0A ₁ 端子 P0A ₀ 端子	入力	出力	0	0	0		
				P0ABIO2								
				P0ABIO1								
				P0ABIO0								
シ リ ア ル ・ イ ン タ フ ェ ー ス	シリアル I/O モード・ セレクト・ レジスタ	08H	R/W	SIO0CH	通信線数を設定	2 線式	3 線式	0	0	0		
				SB	通信方式を設定	シリアル I/O 方式	I ² C バス方式 (2 線式のみ)					
				SIO0MS	マスタ/スレーブを設定	マスタ動作	スレーブ動作					
				SIO0TX	転送方向を設定	受信	送信					
シ リ ア ル ・ イ ン タ フ ェ ー ス	シリアル I/O ウェイト・ コントロール・ レジスタ	18H	R/W	SBACK	アクリッジの設定, 検出 (I ² C バス方式)	0, 1 の設定および検出		0	0	0		
				SIO0NWT	ウェイト許可を設定	許可	解除					
				SIO0WRQ1	ウェイト・モードを設定	0	0				1	1
				SIO0WRQ0		ノ・ ウエイ ト	テ・ ウエイ ト				ア・ クリ ッジ	ア ドレ ス ・ ウエイ ト

備考 保：以前の状態を保持 不：不定

表 9-1 コントロール・レジスタの周辺ハードウェア制御機能一覧 (4/5)

周辺ハードウェア	コントロール・レジスタ				周辺ハードウェア制御機能				リセット時					
	名称	番地	Read/Write	記号 b3 b2 b1 b0	機能概略	設定値		パ ワ ー オ ン	S T O P	C E				
						0	1							
シリアル・インターフェース	シリアル I/O0 ステータス・ジャッジ・レジスタ	28H	R	SIO0SF8	クロック・カウンタの内容を検出	クロック・カウンタの内容が0または1でリセット	クロック・カウンタの内容が8になるとセット	0	0	0				
				SIO0SF9		クロック・カウンタの内容が0または1でリセット	クロック・カウンタの内容が9になるとセット							
				SBSTT		クロック数を検出 (I ² Cバス方式)	スタート条件-9クロック目までセット							
				SBBSY		スタート条件を検出 (I ² Cバス方式)	スタート条件-ストップ条件までセット							
シリアル I/O0 割り込みモード・レジスタ	38H	R/W	0	0 固定	シリアル・インタフェース0の割り込み条件の設定	0	1	不	保	保				
			0			7クロック目	8クロック目				スタート条件後の7クロック目	ストップ条件		
			SIO0IMD1			0	1				1	1		
			SIO0IMD0			0	1				0	1		
シリアル I/O0 クロック・セレクト・レジスタ	39H	R/W	0	0 固定	シリアル・インタフェース0の内部クロックの設定	0	1	不	保	保				
			0			100 kHz	200 kHz				500 kHz	1 MHz		
			SIO0CK1			0	1				0	1		
			SIO0CK0			0	1				0	1		
水平同期信号カウンタ	H _{SYNC} カウンタ・ゲート・コントロール・レジスタ	11H	R/W	0	0 固定	H _{SYNC} カウンタのゲートを制御	0	1	0	0	0			
				0			ゲート・クローズ	ゲート・オープン				1.69 ms	ゲート・オープン	設定禁止
				HSCGT1			0	1				0	1	
				HSCGT0			0	1				0	1	
H _{SYNC} カウンタ・ゲート・ジャッジ・レジスタ	12H	R	HSCGOSTT	0 固定	H _{SYNC} カウンタのゲート開閉を検出	ゲート・クローズ	ゲート・オープン	0	-	-				
			0											
			0											
			0											

備考 保：以前の状態を保持 不：不定

表 9-1 コントロール・レジスタの周辺ハードウェア制御機能一覧 (5/5)

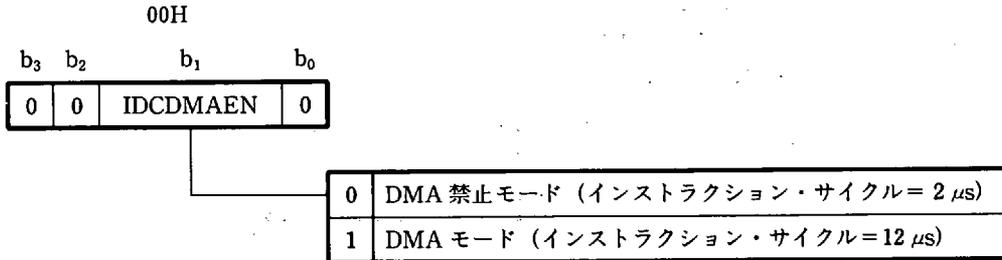
周辺ハードウェア	コントロール・レジスタ				周辺ハードウェア制御機能				リセット時		
	名称	番地	Read/Write	記号 b3 b2 b1 b0	機能概略	設定値		パ ワ ー オ ン	S T O P	C E	
						0	1				
I D C	IDC DMA イネーブル・ レジスタ	00H	R/W	0	0 固定						
				IDCDMAEN	DMA モードの許可設定	許可しない	許可する	0	0	0	
				0	0 固定						
	IDC CROM バンク・ レジスタ	30H	R/W	0	0 固定						
				0	0 固定						
				CROMBNK	CROM のバンクを選択	BANK0 (0800H-0BFFH)	BANK1 (0C00H-0F7FH)	0	0	0	
	IDC イネーブル・ レジスタ	31H	R/W	0	0 固定						
0				0 固定							
IDCEN				IDC の表示のオン/オフを設定	表示オン	表示オフ	0	0	0		

備考 保：以前の状態を保持 不：不定

9.1 IDCDMAEN (00H, b₁)

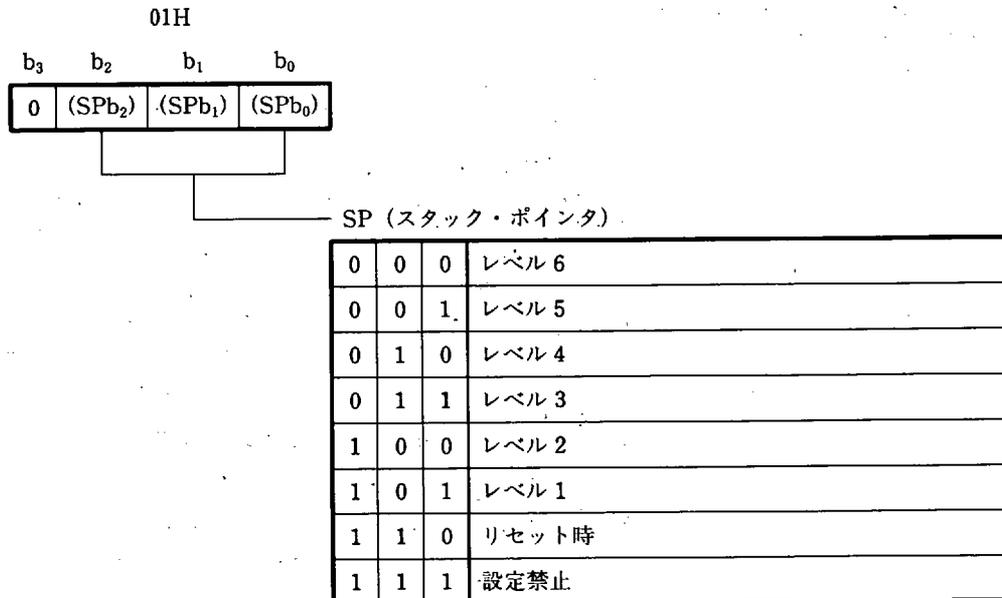
IDCを動作させるときにこのフラグをセットします。

IDCDMAEN フラグがセットされると DMA モードとなり IDC の使用が可能になります。DMA モードでは見かけ上のインストラクション・サイクルは 12 μs になります。詳しくは、“20. IDC” の項をご覧ください。



9.2 SP (01H)

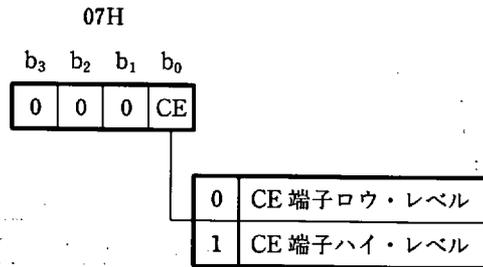
スタック・レジスタをアドレスするポインタです。



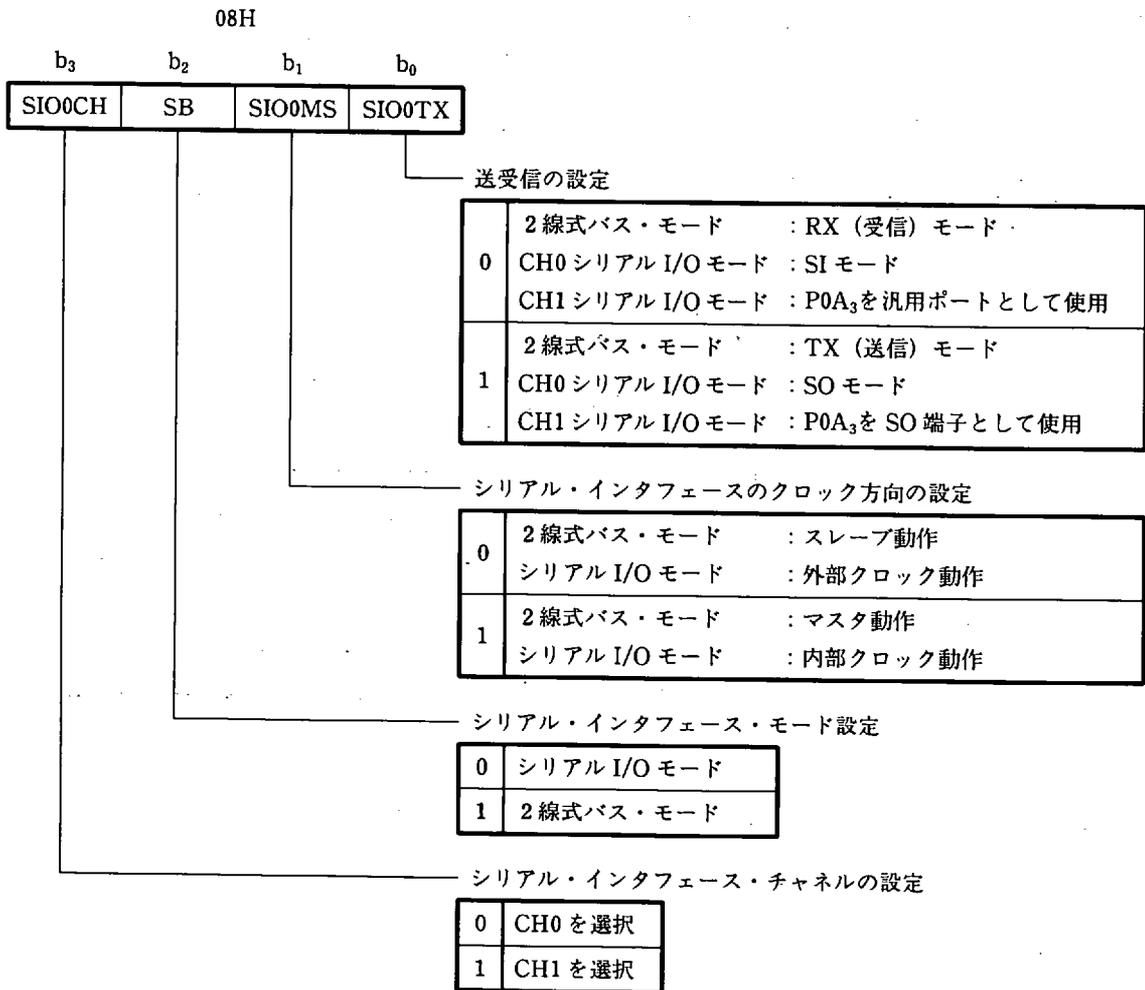
9.3 CE (07H, b₀)

CE 端子のレベルを読むためのフラグです。

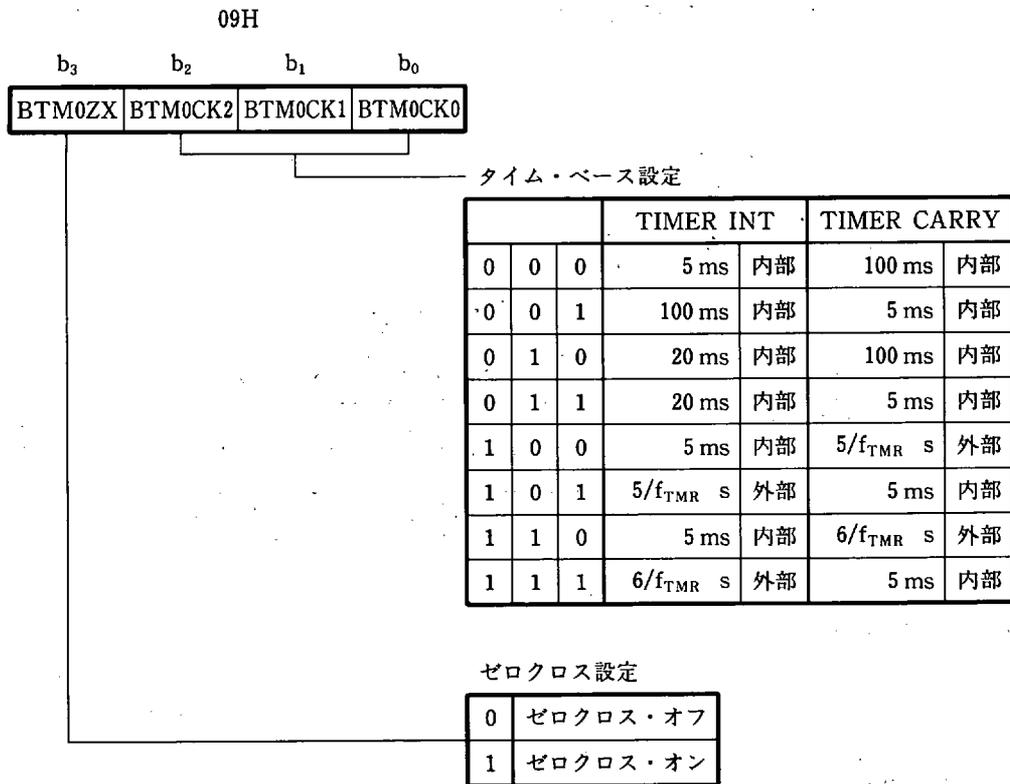
CE 端子にハイ・レベルが入力されているときは“1”，ロウ・レベルが入力されているときは“0”となります。



9.4 シリアル・インタフェース・モード・レジスタ (08H)



9.5 BTM0MD (09H)



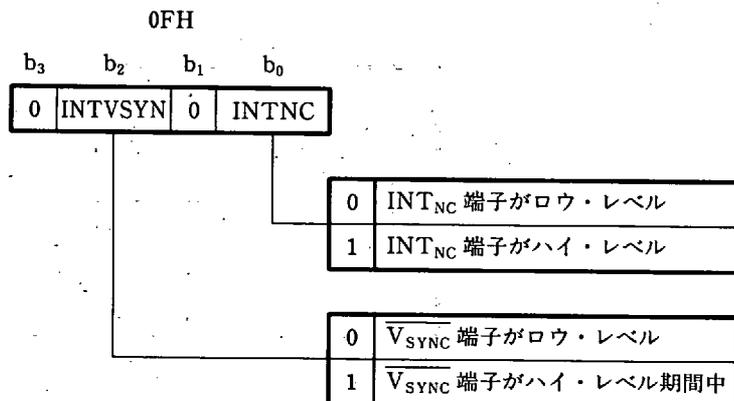
9.6 INTVSYN (0FH, b₂)

垂直同期信号のレベルを読むためのフラグです。V_{SYNC}端子に入力されている信号がハイ・レベルのときセット(1)され、ロウ・レベルのときリセット(0)されます。

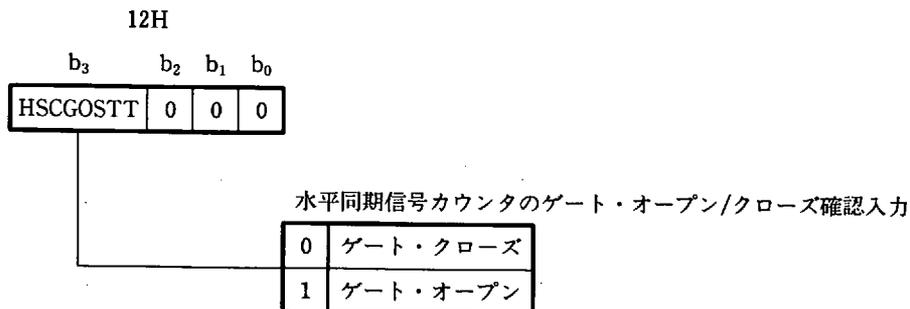
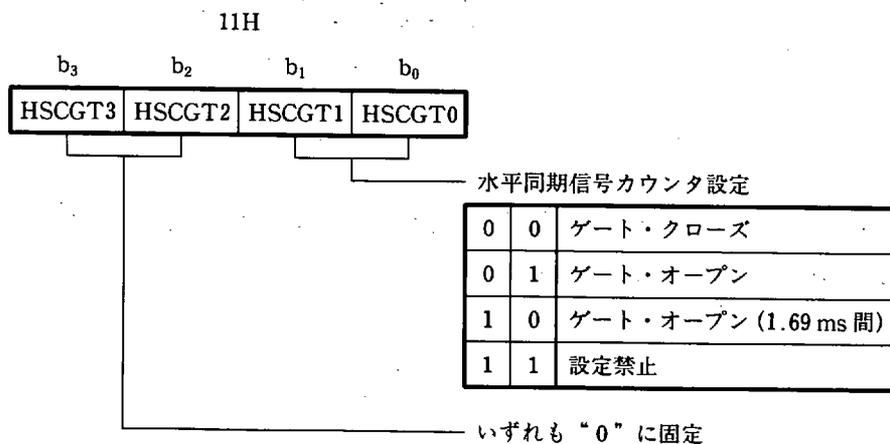
9.7 INTNC (0FH, b₀)

INT_{NC} 端子の状態を読むためのフラグです。

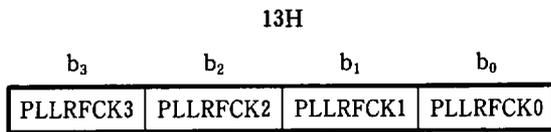
INT_{NC} 端子にハイ・レベルが入力されているときは“1”，ロウ・レベルが入力されているときは“0”となります。



9.8 水平同期信号カウンタ制御 (11H, 12H)



9.9 PLL レファレンス・モード・セレクト・レジスタ (13H)

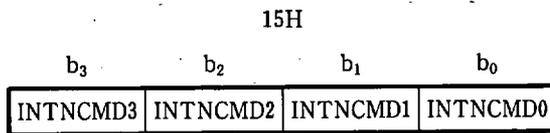


基準周波数 f_r の設定

0	0	1	0	6.25 kHz
0	0	1	1	12.5 kHz
0	1	1	0	25 kHz
1	1	1	1	PLL ディスエーブル
0	1	1	1	設定禁止
1	0	1	0	
1	0	1	1	
1	1	1	0	

“1” に固定

9.10 INT_{NC} 端子受け付けパルス幅の設定 (15H)

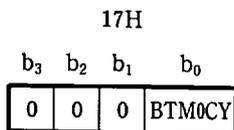


INT_{NC} 端子受け付けパルス幅の設定

0	0	0	エッジ (ノイズ・キャンセラなし)
0	0	1	200 μs
0	1	0	400 μs
0	1	1	2 ms
1	0	0	4 ms

“0” に固定

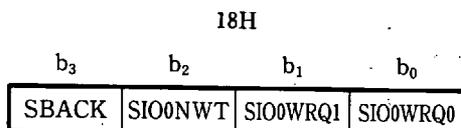
9.11 タイマ・キャリー (17H)



タイム・キャリー用読み出し専用フラグ

選択されたタイム・ベースによりセットされ、読み出すことによりリセットされます。

9.12 シリアル・インタフェース・ウェイト・コントロール (18H)



ウェイトするタイミングの設定

		2線式バス・モード	シリアル I/O モード
0	0	ウェイトしない	ウェイトしない
0	1	クロック・カウンタの内容が“8”のときのクロックの立ち下がりでウェイトする	クロック・カウンタの内容が“8”になればウェイトする
1	0	クロック・カウンタの内容が“9”のときのクロックの立ち下がりでウェイトする	クロック・カウンタの内容が“9”になればウェイトする
1	1	スタート・コンディション検出後クロック・カウンタの内容が“8”になったときのクロックの立ち下がりでウェイトする	設定禁止

ウェイトの設定

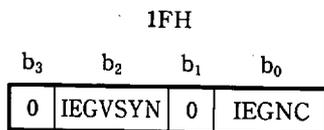
0	強制ウェイト
1	ウェイト解除

2線式バス・モード時のアクノリッジ

9.13 IEGNC (1FH)

INT_{NC} 端子と $\overline{V_{SYNC}}$ 端子の割り込み検出エッジを選択するためのフラグです。

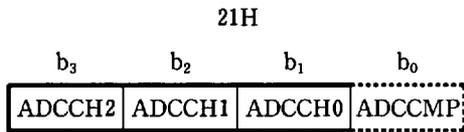
“0”にすると立ち上がりエッジで割り込みがかかり、“1”にすると立ち下がりエッジで割り込みがかかります。



0	INT _{NC} 端子の立ち上がりエッジで割り込み
1	INT _{NC} 端子の立ち下がりエッジで割り込み

0	$\overline{V_{SYNC}}$ 端子の立ち上がりエッジで割り込み
1	$\overline{V_{SYNC}}$ 端子の立ち下がりエッジで割り込み

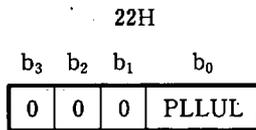
9.14 A/D コンバータ制御 (21H)



A/D コンバータ入力チャネル・セレクト

0	0	0	ADC ₀ セレクト
0	0	1	ADC ₁ セレクト PIC ₃ と兼用
0	1	0	ADC ₂ セレクト P0D ₀ と兼用
0	1	1	ADC ₃ セレクト PID ₁ と兼用
1	0	0	ADC ₄ セレクト P0D ₂ と兼用
1	0	1	ADC ₅ セレクト P0D ₃ と兼用
1	1	0	該当する channel なし
1	1	1	(設定禁止)

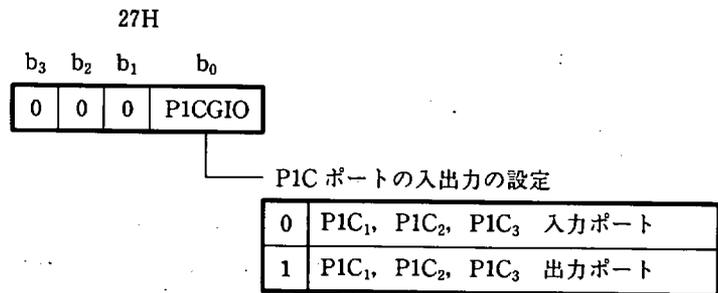
9.15 PLL アンロック FF ジャッジ・レジスタ (22H)



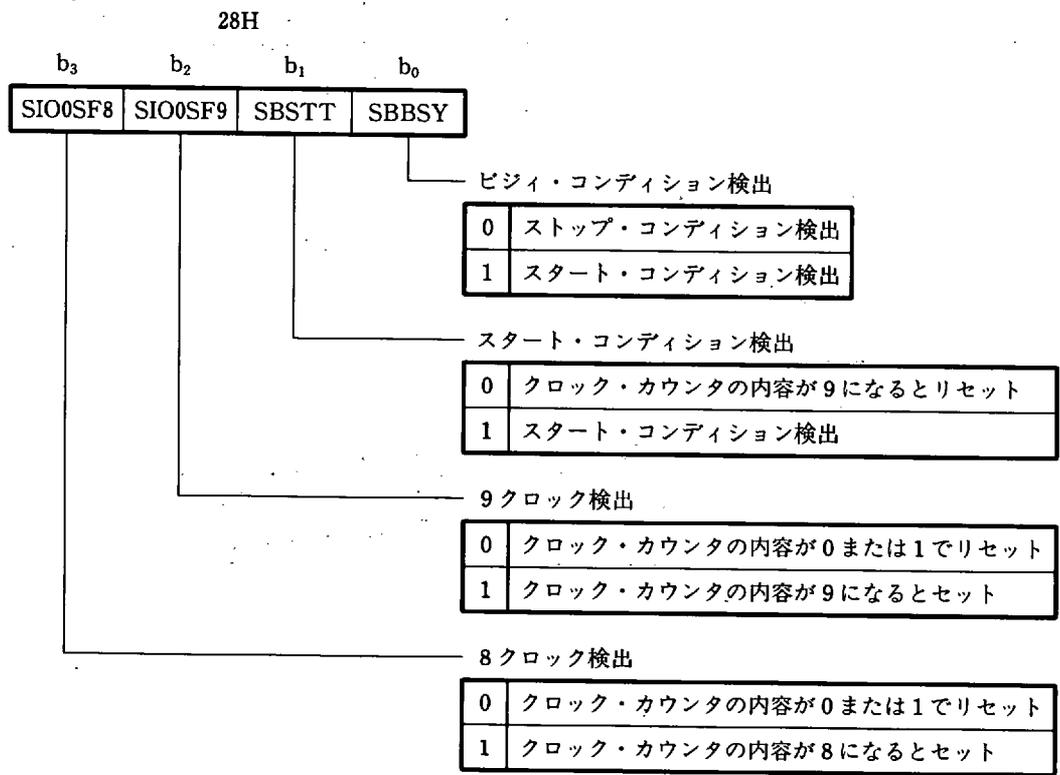
アンロック FF の状態を検出

0	アンロック FF = 0 : PLL ロック状態
0	アンロック FF = 1 : PLL アンロック状態

9.16 PORT1C の入出力の設定 (27H)

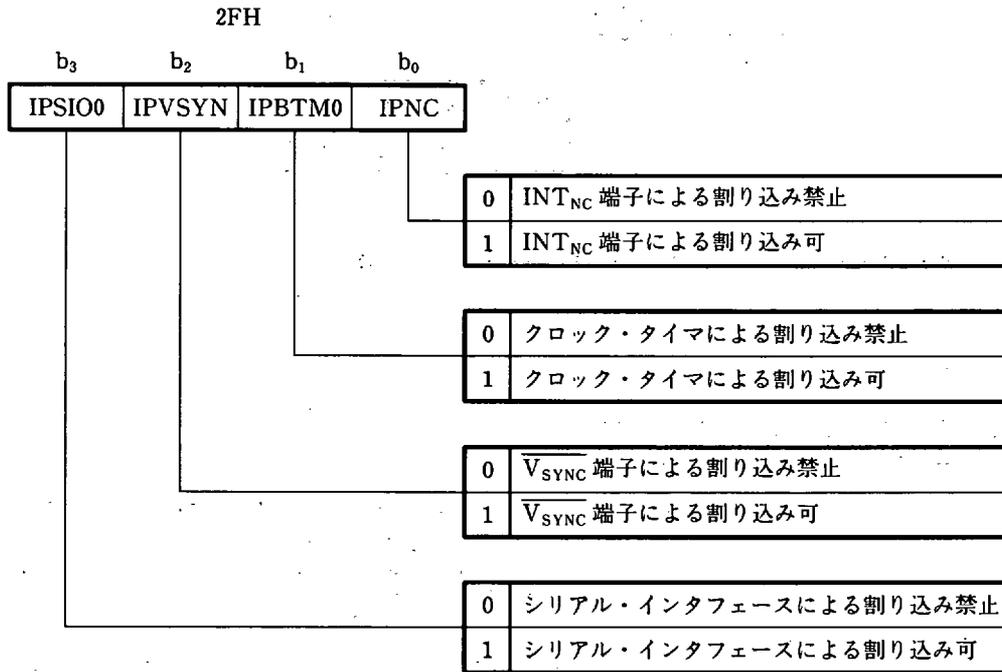


9.17 シリアル I/O ステータス・レジスタ (28H)



9.18 割り込み許可フラグ (2FH)

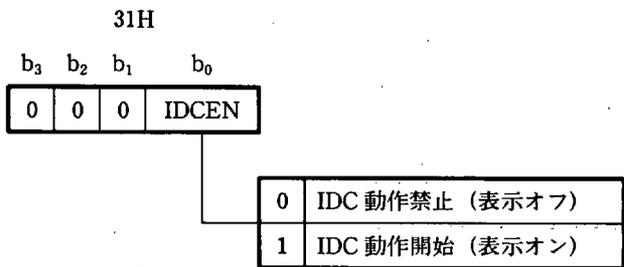
各割り込み要因ごとの割り込みを許可するためのフラグです。“1”にすると割り込みが可能です。“0”にすると割り込みは、禁止されます。



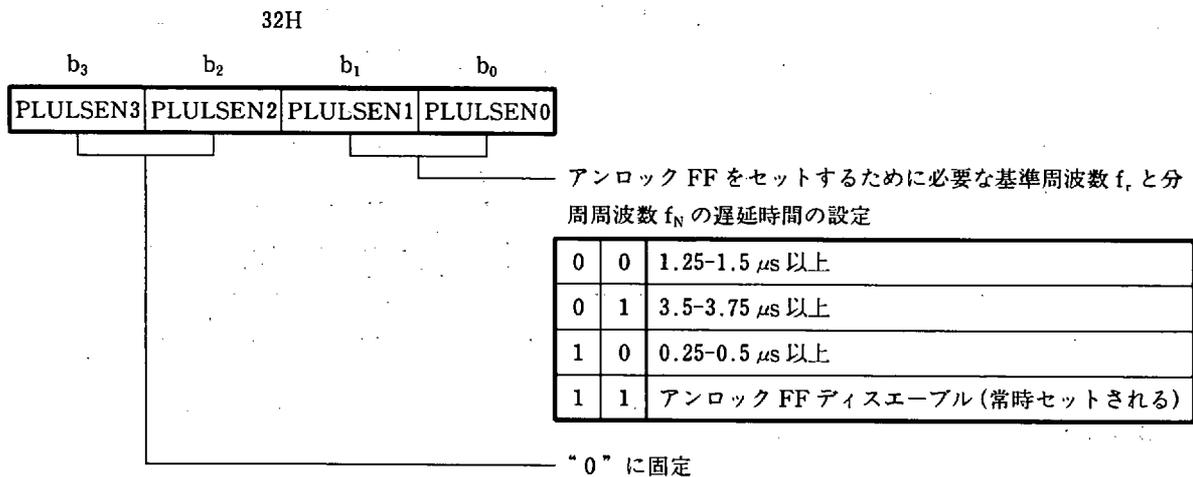
9.19 CROM バンク選択 (30H)



9.20 IDCEN (31H)

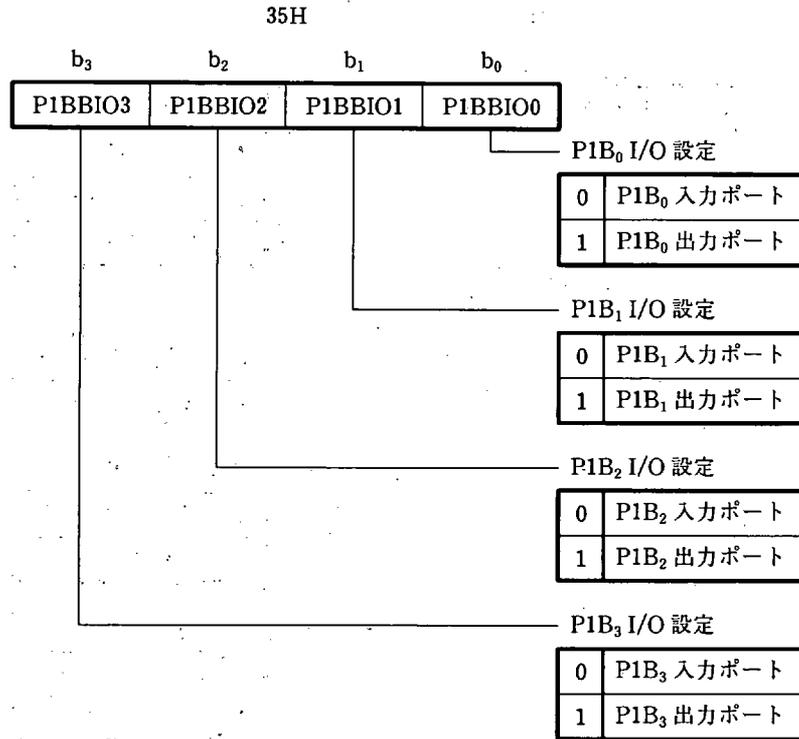


9.21 PLL アンロック FF ディレイ・コントロール・レジスタ (32H)



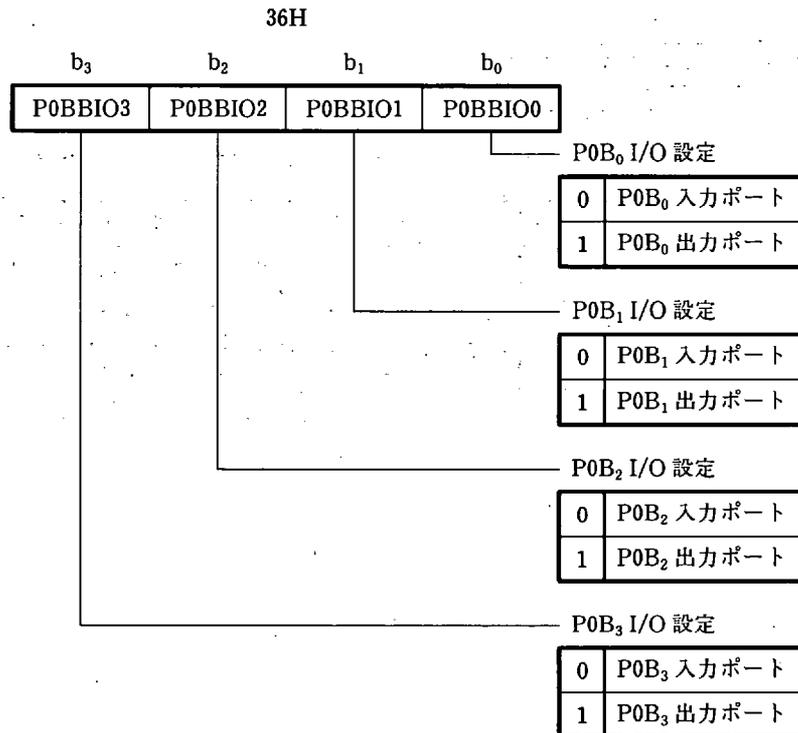
9.22 PIBBIO_n (35H)

PORT1Bの入出力を指定します。“0”にすると入力になり“1”にすると出力になります。



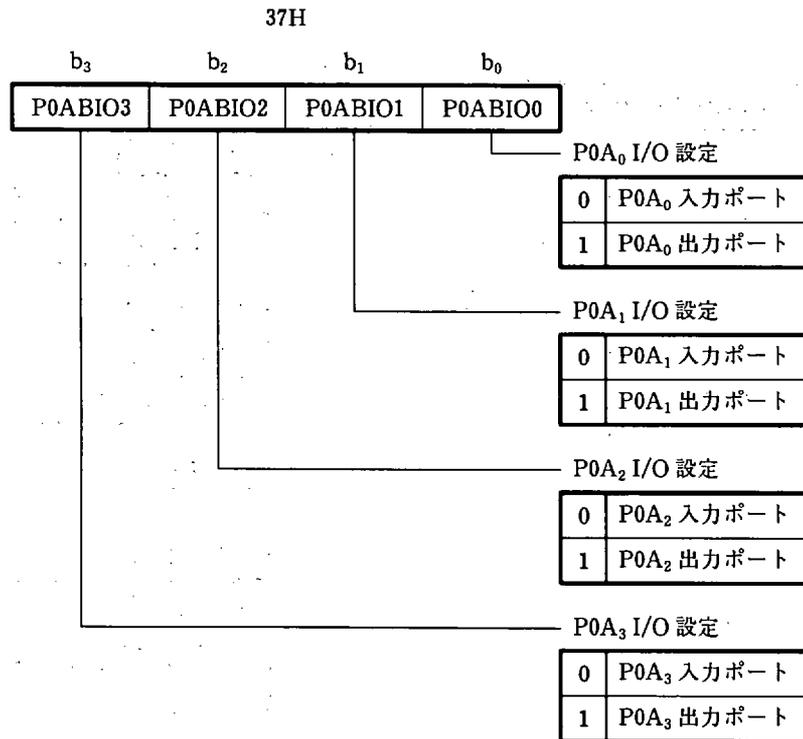
9.23 P0BBIO_n (36H)

PORT0Bの入出力を指定します。“0”にすると入力になり“1”にすると出力になります。

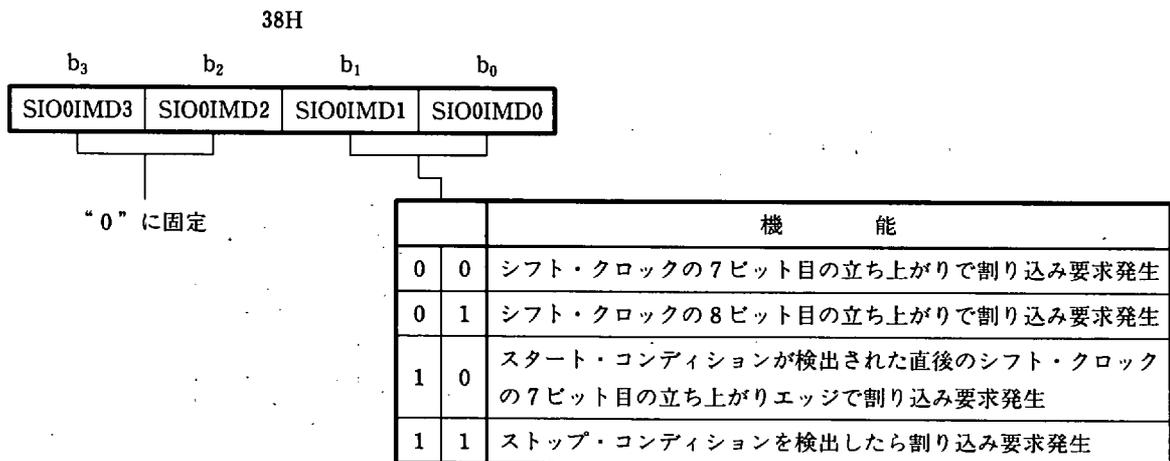


9.24 P0ABIO_n (37H)

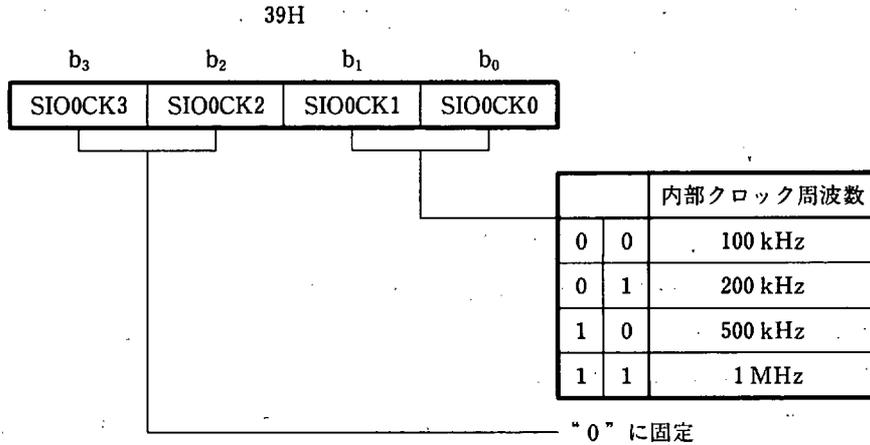
PORT0A の入出力を指定します。“0”にすると入力になり“1”にすると出力になります。



9.25 シリアル・インタフェース・モードでの割り込み要求発生タイミングの設定 (38H)



9.26 シフト・クロック周波数の設定 (39H)



9.27 IRQNC (3FH)

割り込み要求状態を示す割り込み要求フラグです。

割り込み要求が発生すると“1”にセットされます。そして割り込みが受け付けられる（割り込みがかかる）と、割り込み要求フラグは、“0”にリセットされます。

割り込み要求フラグは、プログラムで読み書きができます。したがって“1”を書き込むと、ソフトウェアによる割り込みを発生させることができ、“0”を書き込むことにより、その割り込み保留状態を解除することができます。またIRQNCフラグは、リセット時は“0”になります。

フラグ名	ビット位置	割り込み要因
IRQNC	b ₀	INT _{NC} 端子
IRQBTM0	b ₁	クロック・タイマ
IRQVSYN	b ₂	$\overline{V}_{\text{SYNC}}$ 端子
IRQSIO0	b ₃	シリアル・インタフェース

10. データ・バッファ (DBF)

データ・バッファは周辺ハードウェアとのデータ転送時およびテーブル参照のデータ読み込み時に使用するバッファです。

10.1 データ・バッファの構成

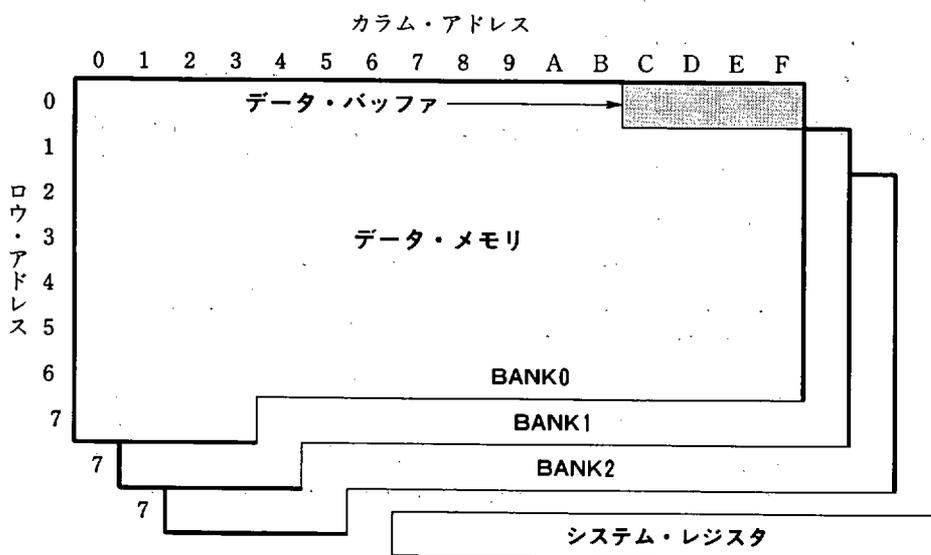
10.1.1 データ・バッファのデータ・メモリ上の配置

図 10-1 にデータ・バッファのデータ・メモリ上の配置を示します。

図 10-1 に示すように、データ・バッファは、データ・メモリ上の BANK0 のアドレス 0CH-0FH に割り当てられており、4ワード×4ビットの計16ビットから構成されています。

データ・バッファは、データ・メモリ上に配置されているため、すべてのデータ・メモリ操作命令で操作できます。

図 10-1 データ・バッファの配置



10.2 データ・バッファの機能

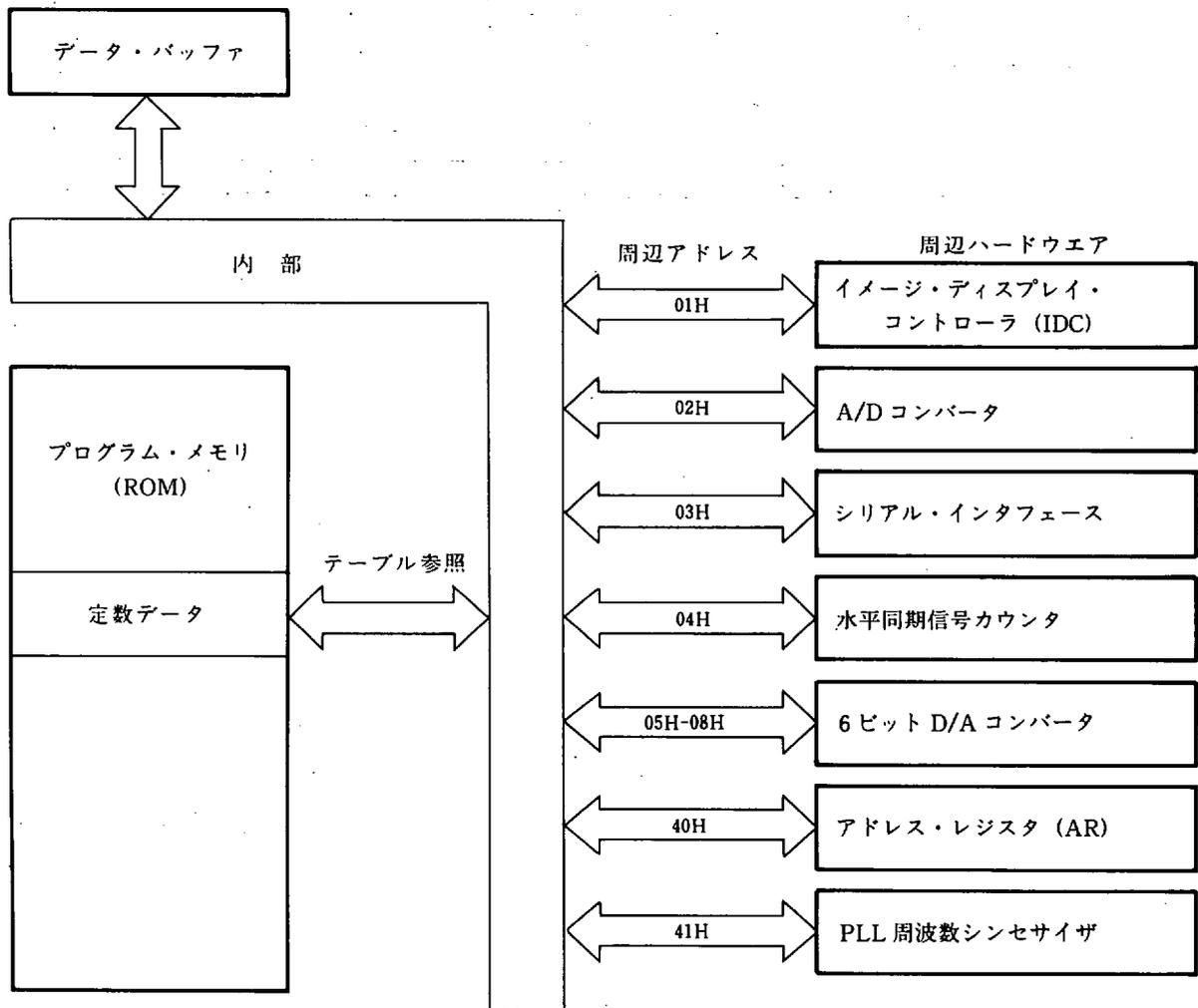
データ・バッファには、以下の(1)および(2)に示す2つの機能があります。

- (1) プログラム・メモリ上の定数データの読み込み（テーブル参照）機能
- (2) 周辺ハードウェアとのデータ転送機能

図10-3にデータ・バッファと周辺ハードウェアおよびテーブル参照の関係を示します。

また、10.3項にテーブル参照について、10.4-10.6項に周辺ハードウェアについて説明します。

図10-3 データ・バッファと周辺ハードウェアおよびテーブル参照との関係



10.3 データ・バッファとテーブル参照

10.3.1 テーブル参照動作

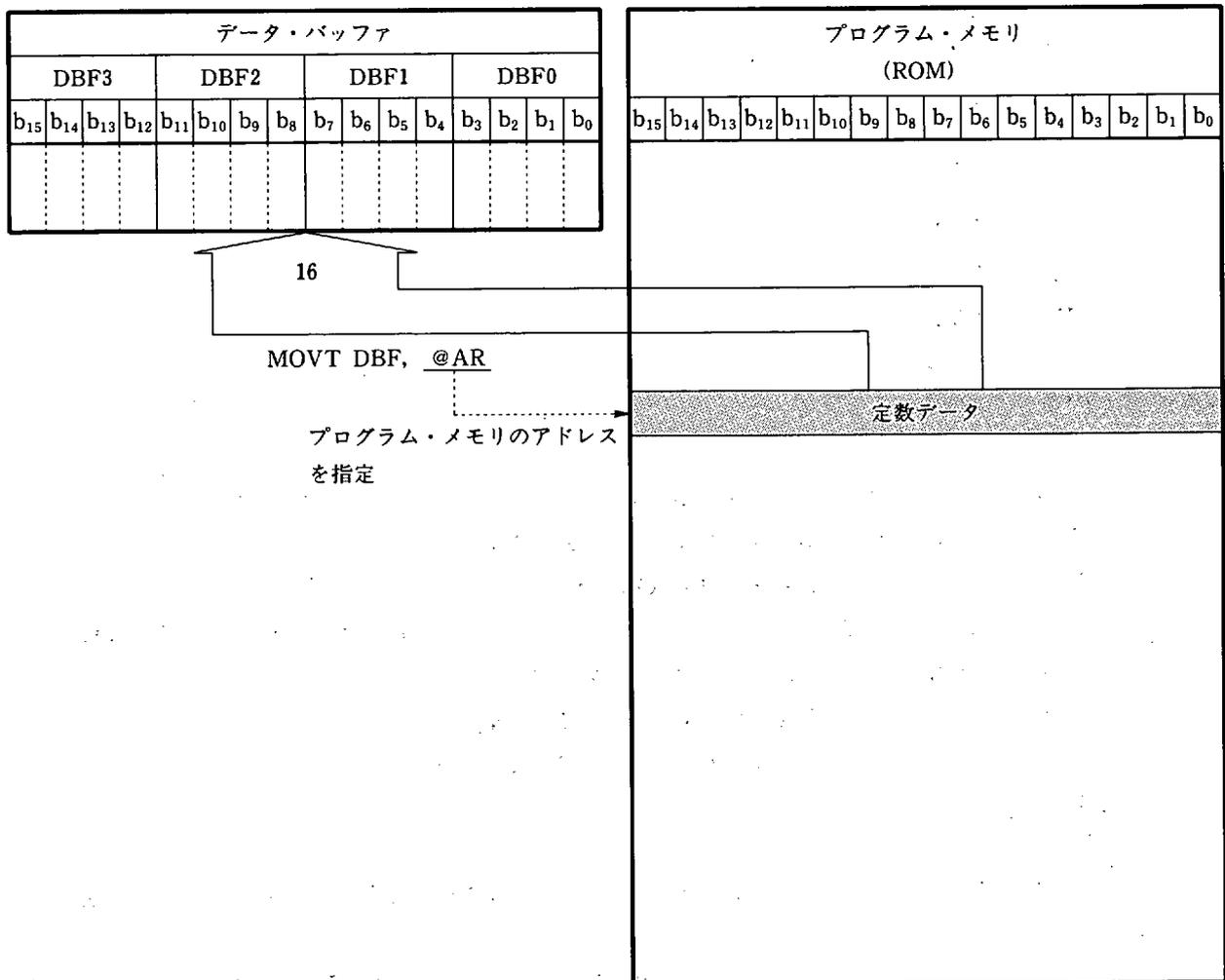
テーブル参照は“MOVT DBF, @AR”命令を用いることにより、プログラム・メモリ上の定数データを、データ・バッファ上に読み込むことができます。

したがって、プログラム・メモリ上にたとえば表示データ等の定数データをあらかじめ書き込んでおき、必要なときにテーブル参照すれば、複雑なデータ変換プログラムを作成する必要がなくなります。

以下に“MOVT”命令について説明します。

また、10.3.2にプログラム例を示します。

MOVT DBF, @AR ; 下図に示すようにアドレス・レジスタの内容によってアドレス指定されるプログラム・メモリの内容を、データ・バッファに読み出します。



テーブル参照命令実行時はスタックが1レベル使用されます。

また、アドレス・レジスタ (AR) の有効ビットは8ビットのため、テーブル参照が可能なプログラム・メモリ・アドレスは 0000H-00FFH 番地の256ステップになります。

“4. スタック” および “8.1 アドレス・レジスタ (AR)” の項も参照してください。

10.3.2 テーブル参照プログラム例

例にテーブル参照のプログラム例を示します。

例

```

P0A    MEM    0.70H    ;
P0B    MEM    0.71H    ;
P0C    MEM    0.72H    ;
ORG    0000H
START :
BR     MAIN
DATA :
DW     0001H    ; 定数データ
DW     0002H    ;
DW     0004H    ;
DW     0008H    ;
DW     0010H    ;
DW     0020H    ;
DW     0040H    ;
DW     0080H    ;
DW     0100H    ;
DW     0200H    ;
DW     0400H    ;
DW     0800H    ;
MAIN :
BANK0          ; 組み込みマクロ
SET4 P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
SET4 P0BBIO3, P0BBIO2, P0BBIO1, P0BBIO0
MOV  RPL,    #1110B    ; ジェネラル・レジスタを BANK0 のロウ・アドレス 7H に設定
MOV  AR1,    # (.DL.DATA SHR 4 AND 0FH)
MOV  AR0,    # (.DL.DATA SHR 0 AND 0FH)
                ; アドレス・レジスタを 0001H に設定
LOOP :
; ①
MOV  DBF,    @AR    ; データ・バッファに AR の内容で指定される ROM の値を転送
; ②
LD   P0A,    DBF2,    ; データ・バッファの値を Port0A (70H), Port0B (71H)
LD   P0B,    DBF1    ; および Port0C (72H) の各ポート・データ・レジスタへ転送
LD   P0C,    DBF0
ADD  AR0,    #1      ; アドレス・レジスタの内容を 1 だけインクリメント
ADDC AR1,    #0
SKNE AR0,    #0CH    ; AR0 の値が 0CH になったとき AR0 に 0 を書き込む
MOV  AR0,    #0
BR   LOOP

```

このプログラムを実行するとプログラム・メモリのアドレス 0001H-000CH 番地に格納されている定数データを①で順次データ・バッファに読み込み、②で Port0A, Port0B, Port0C に出力します。

このとき定数データは1ビットずつ左にシフトしている値が格納されているため、結果として Port0A, Port0B および Port0C の各端子に順次ハイ・レベルを出力していきます。

10.4 データ・バッファと周辺ハードウェア

10.4.1 周辺ハードウェアの制御方法

以下に、データ・バッファを介してデータ転送を行う周辺ハードウェアを示します。

- イメージ・ディスプレイ・コントローラ
- A/D コンバータ
- シリアル・インタフェース
- 水平同期信号カウンタ
- 6ビット D/A コンバータ
- アドレス・レジスタ
- PLL 周波数シンセサイザ

データ・バッファを介してこれらの周辺ハードウェアにデータを設定したり、データを読み込むことにより周辺ハードウェアを制御します。

周辺ハードウェアはそれぞれデータ転送用のレジスタ（周辺レジスタと呼ぶ）があり、この周辺レジスタにはそれぞれアドレス（周辺アドレスと呼ぶ）が割り付けられています。

この周辺レジスタに対して、専用命令である“GET”および“PUT”命令を行うことにより、データ・バッファと各周辺ハードウェアとのデータ転送を行うことができます。

“GET”および“PUT”命令について以下に説明します。また表 10-1 に周辺ハードウェアとデータ・バッファの機能一覧を示します。

GET DBF, p ; データ・バッファに p でアドレスされる周辺レジスタのデータを読み込む。
 PUT p, DBF ; p でアドレスされる周辺レジスタにデータ・バッファのデータを設定する。

周辺レジスタには書き込み読み出し可能 (PUT/GET)、書き込み専用 (PUT) および読み出し専用 (GET) のレジスタがあります。

このとき、書き込み専用 (PUT のみ) や、読み出し専用 (GET のみ) の周辺レジスタに対してそれぞれ“GET”および“PUT”命令を行うと、デバイス上は以下に示すようになります。

- 書き込み専用 (PUT のみ) 周辺レジスタに対して読み出し (GET) 命令実行時
不定の値が読み出されます。
- 読み出し専用 (GET のみ) 周辺レジスタに対して書き込み (PUT) 命令実行時
何ら影響を与えません。

ただし、17K シリーズのアセンブラやエミュレータ使用時に注意が必要です。詳しくは“10.6 データ・バッファ使用時の注意”の項を参照してください。

表 10-1 周辺ハードウェアとデータ・バッファの機能一覧

周辺ハードウェア		データ・バッファとデータ転送を行う周辺レジスタ				機 能		
		名 称	記 号	周辺 アドレス	PUT 命令/ GET 命令 の可否	データ・バッファ 入出力ビット数	実用 ビット数	概 要
イメージ・ディスプレイ・コントローラ		IDC スタート位置設定レジスタ	IDCORG	01H	PUT/GET	8	7	イメージ・ディスプレイ・コントローラの表示開始位置を設定
A/D コンバータ		A/D コンバータ V _{REF} データ・レジスタ	ADCR	02H	PUT/GET	8	4	A/D コンバータの比較電圧 V _{REF} データを設定 $V_{REF} = \frac{x - 0.5}{16} \times V_{DD} (V),$ 1 ≤ x ≤ 15
シリアル・インタフェース		プリセットアップ・シフト・レジスタ	SIO0SFR	03H	PUT/GET	8	8	シリアル・アウト・データの設定およびシリアル・イン・データの読み込み
水平同期信号カウンタ		HSYNC カウンタ・データ・レジスタ	HSC	04H	GET	8	6	水平同期信号カウンタの値を読み込み
6ビット D/A コンバータ (PWM 出力)	PWM ₀ 端子	PWM データ・レジスタ 0	PWMR0	05H	PUT/GET	8	7	D/A コンバータの出力信号のデューティを設定 デューティ $D = \frac{x + 0.75}{64} (\%)$, 0 ≤ x ≤ 63 周波数 f = 15.625 kHz
	PWM ₁ 端子	PWM データ・レジスタ 1	PWMR1	06H				
	PWM ₂ 端子	PWM データ・レジスタ 2	PWMR2	07H				
	PWM ₃ 端子	PWM データ・レジスタ 3	PWMR2	08H				
アドレス・レジスタ		アドレス・レジスタ	AR	40H	PUT/GET	16	16	アドレス・レジスタとのデータ転送
PLL 周波数シンセサイザ		PLL データ・レジスタ	PLL R	41H	PUT/GET	16	16	PLL 周波数シンセサイザの分周比を設定

10.4.2 周辺レジスタとのデータ転送時の注意

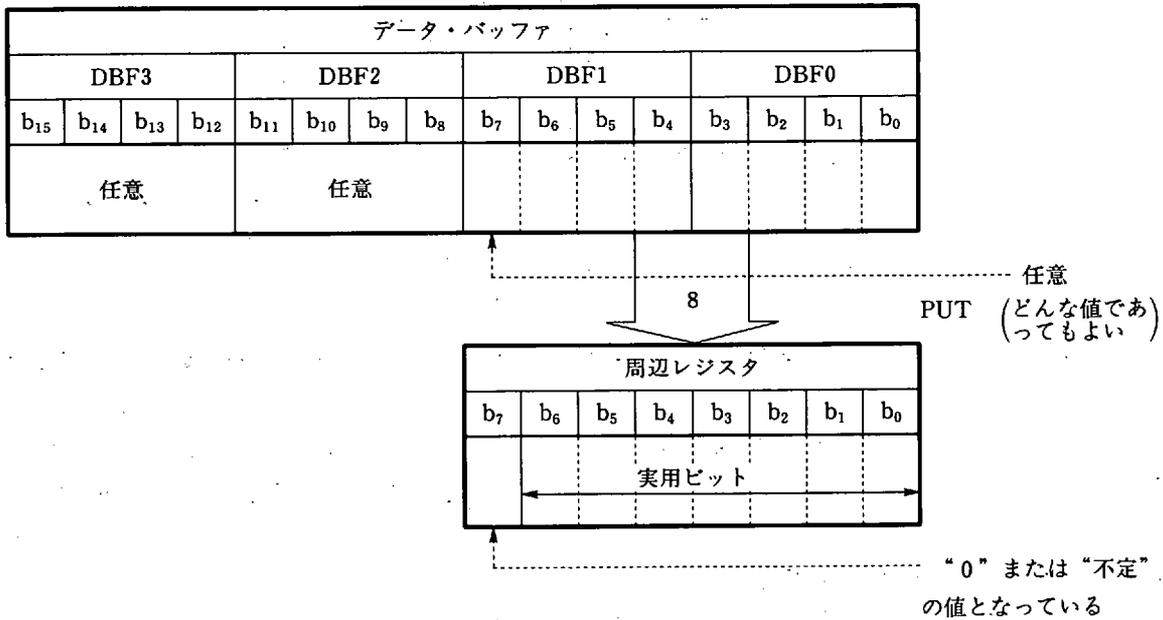
データ・バッファと各周辺レジスタとのデータ転送時は8ビットまたは16ビット単位で行います。

このとき、“PUT”および“GET”命令は、データ・ビットが16ビットであっても1命令実行時間（2μs）で実行できます。

また、周辺レジスタの実行データ・ビット長がたとえば7ビットであるときに8ビットのデータ転送を行うと1ビットは余分なデータとなります。

この余分なデータ・ビット分はデータ読み出し時およびデータ書き込み時に例1、例2に示すように書き込み時は“任意（どんな値でもよい）”，読み出し時には“不定な値”になります。

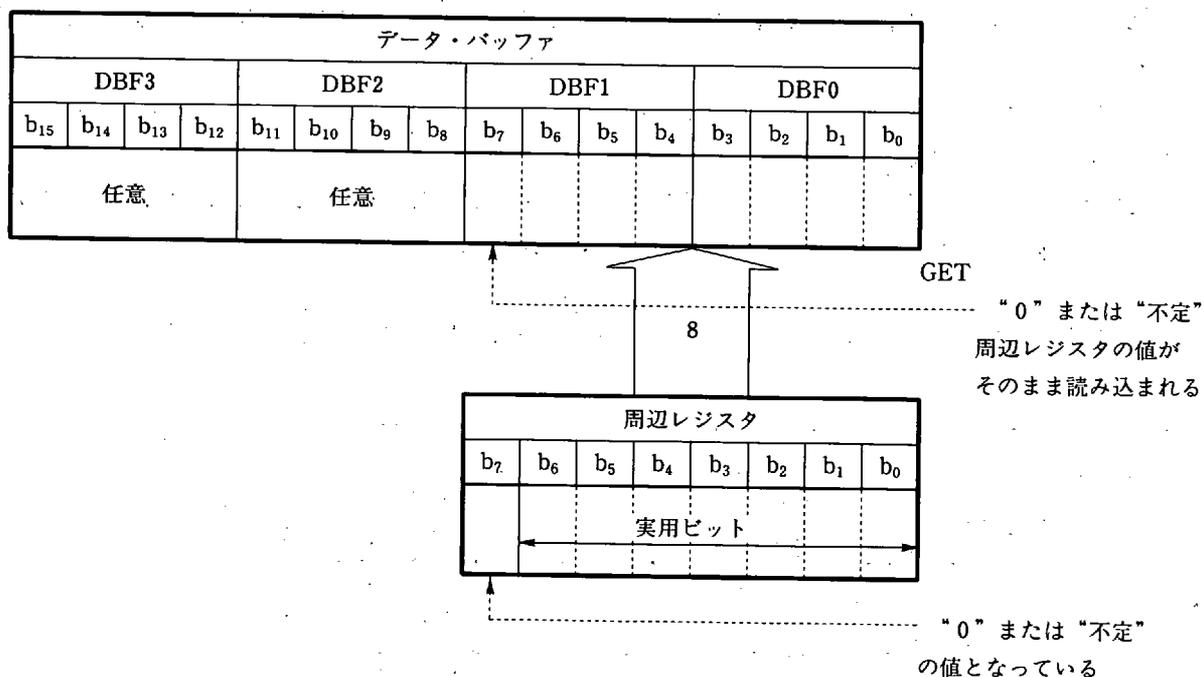
例1. “PUT”命令時（周辺レジスタの実用ビットがビットb₀-ビットb₆の7ビットであるとき）



周辺レジスタに8ビット・データを書き込むときはデータ・バッファの上位8ビット（DBF3およびDBF2の内容）は“任意（どんな値であってもよい）”となります。

データ・バッファの8ビット・データのうち、周辺レジスタの実用ビットに対応しない各ビットは“任意”となります。

例2. "GET" 命令時



周辺レジスタの8ビット・データを読み込むときにはデータ・バッファの上位8ビット(DBF3およびDBF2)の値は変化しません。

データ・バッファの8ビット・データのうち、周辺レジスタの实用ビットでない各ビットは“0”または“不定”となります。“0”となるか、“不定”となるかは、各周辺レジスタによりあらかじめ決定されています。

10.4.3 周辺レジスタのリセット時の状態

各周辺レジスタの实用ビットは、リセット時に以下のように設定されます。

リセット	实用ビットの状態
パワーオン	不定
クロック・ストップ	以前の状態保持
CE	以前の状態保持

10.5 データ・バッファと各周辺レジスタ

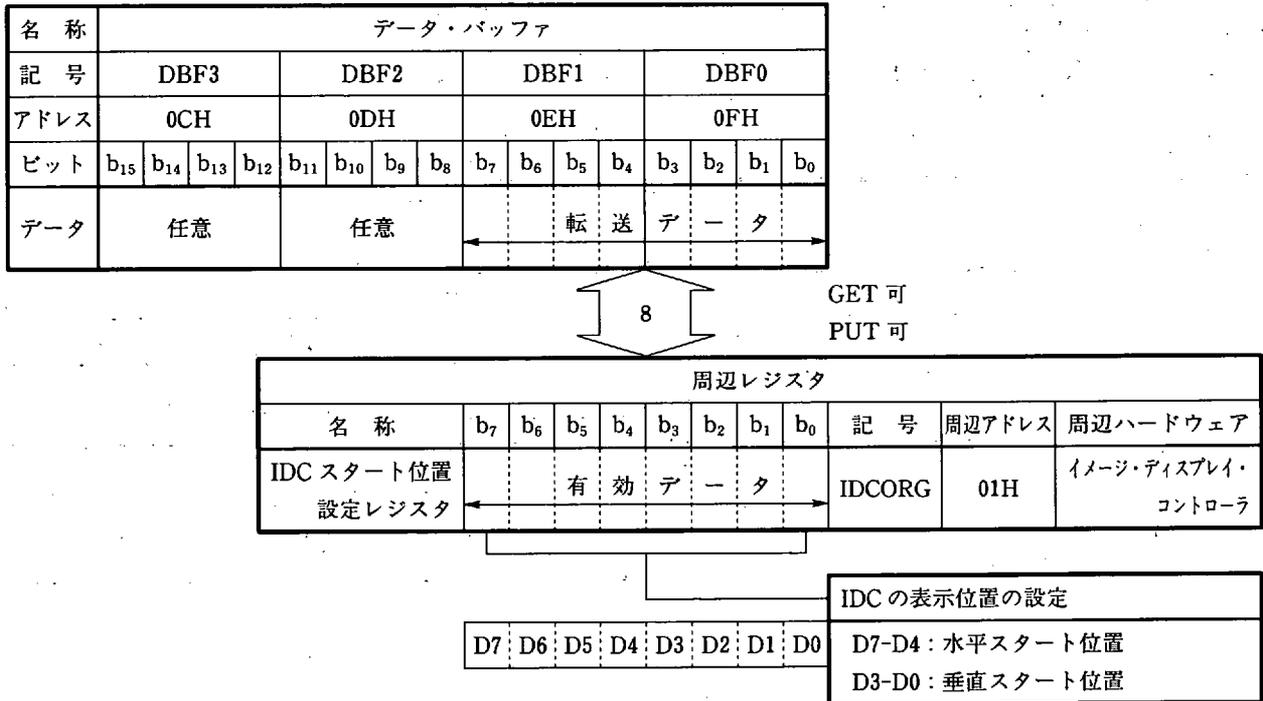
10.5.1-10.5.7 にデータ・バッファと各周辺レジスタについて説明します。

10.5.1 IDC スタート位置設定レジスタ

図 10-4 に、IDC スタート位置設定レジスタの機能を示します。

IDC スタート位置設定レジスタは IDC の表示開始位置を設定します。

図 10-4 IDC スタート位置設定レジスタの機能



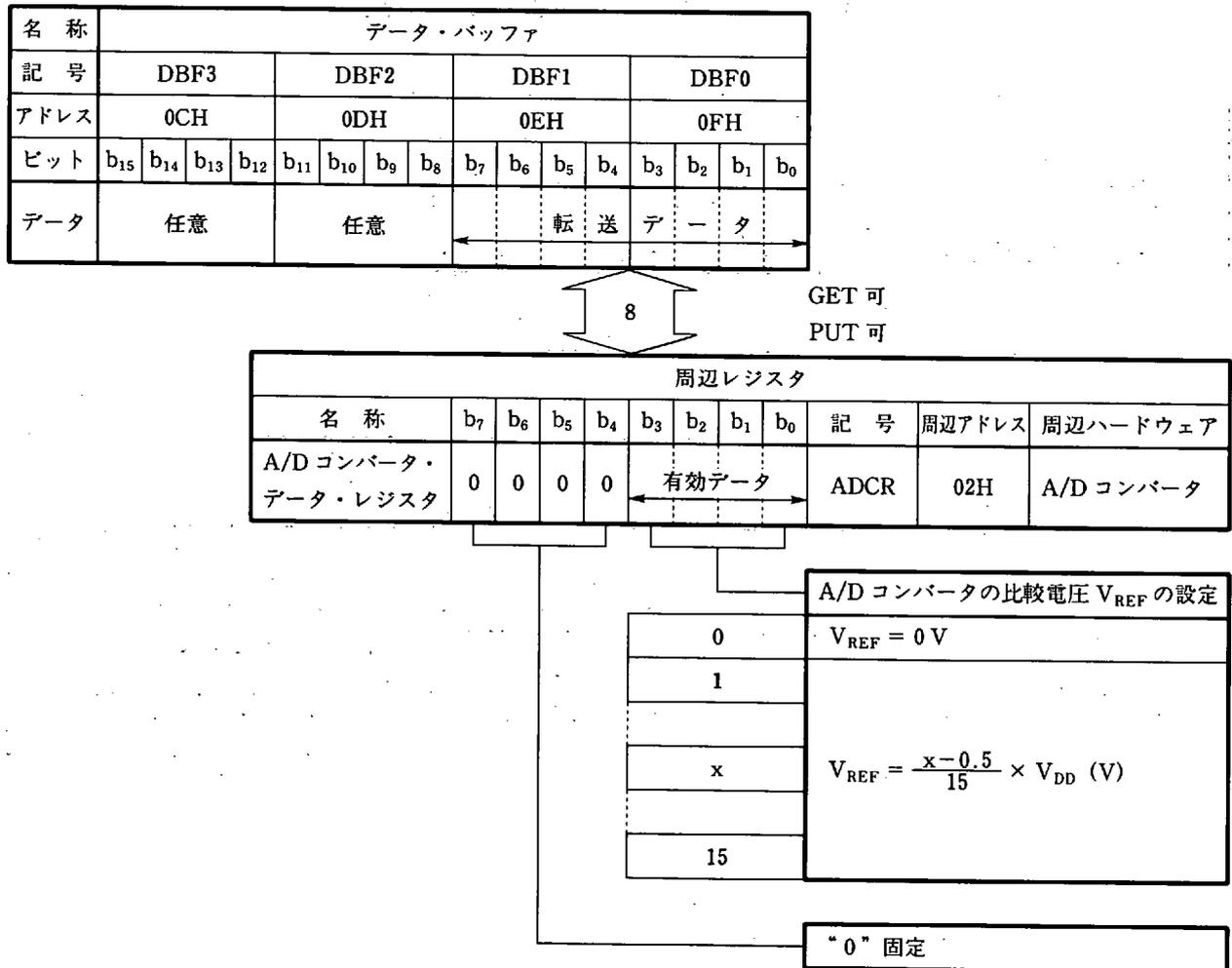
10.5.2 A/Dコンバータ・データ・レジスタ

図10-5に、A/Dコンバータ・データ・レジスタの機能を示します。

A/Dコンバータ・データ・レジスタはA/Dコンバータの比較電圧を設定します。

A/Dコンバータは4ビットであるため、A/Dコンバータ・データ・レジスタの下位4ビットが有効となります。

図10-5 A/Dコンバータ・データ・レジスタの機能



10.5.4 HSYNC カウンタ・データ・レジスタ

図 10-7 に HSYNC カウンタ・データ・レジスタの機能を示します。

HSYNC カウンタ・データ・レジスタは水平同期信号カウンタの計数値を読み込みます。

HSYNC カウンタ・データ・レジスタは 3FH まで計数されると次の入力では 00H になります。

図 10-7 HSYNC データ・レジスタの機能

名 称	データ・バッファ															
記 号	DBF3				DBF2				DBF1				DBF0			
アドレス	0CH				0DH				0EH				0FH			
ビット	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
データ	任意				任意				転 送 デ ー タ							



周辺レジスタ												
名 称	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	記 号	周辺アドレス	周辺ハードウェア	
HSYNC カウンタ・データ・レジスタ	0	0	有 効 デ ー タ						HSC	04H	水平同期信号カウンタ	

水平同期信号カウンタの計数値

10.5.5 PWM データ・レジスタ

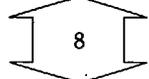
図 10-8 に PWM データ・レジスタの機能を示します。

PWM データ・レジスタは6ビット D/A コンバータ (PWM 出力) 出力信号のデューティを設定します。

6ビット D/A コンバータは PWM₃, PWM₂, PWM₁, PWM₀ 端子の 4 チャンネルあり、それぞれ独立にデューティを設定できるため、PWM データ・レジスタも独立して 4 系統あります。

図 10-8 PWM データ・レジスタの機能

名 称	データ・バッファ															
記 号	DBF3				DBF2				DBF1				DBF0			
アドレス	0CH				0DH				0EH				0FH			
ビット	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
データ	任意				任意				← 転 送 デ ー タ →							



GET 可
PUT 可

周辺レジスタ												
名 称	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	記 号	周辺アドレス	周辺ハードウェア	
PWM0 データ・レジスタ	0		有 効	デ	ー	タ			PWMR0	05H	PWM ₀ 端子	
PWM1 データ・レジスタ	0								PWMR1	06H	PWM ₁ 端子	
PWM2 データ・レジスタ	0								PWMR2	07H	PWM ₂ 端子	
PWM3 データ・レジスタ	0								PWMR3	08H	PWM ₃ 端子	

各端子の PWM 出力デューティを設定

0	$\text{デューティ } D = \frac{x+0.75}{64} (\%)$ 周波数 $f = 15.625 \text{ kHz}$ (f は, PWM の出力くり返し周波数)
x	
63	

0	PWM 端子を D/A コンバータとして使用
0	PWM 端子を 1 ビットの出力端子として使用 b ₅ の内容を出力

10.5.6 アドレス・レジスタ

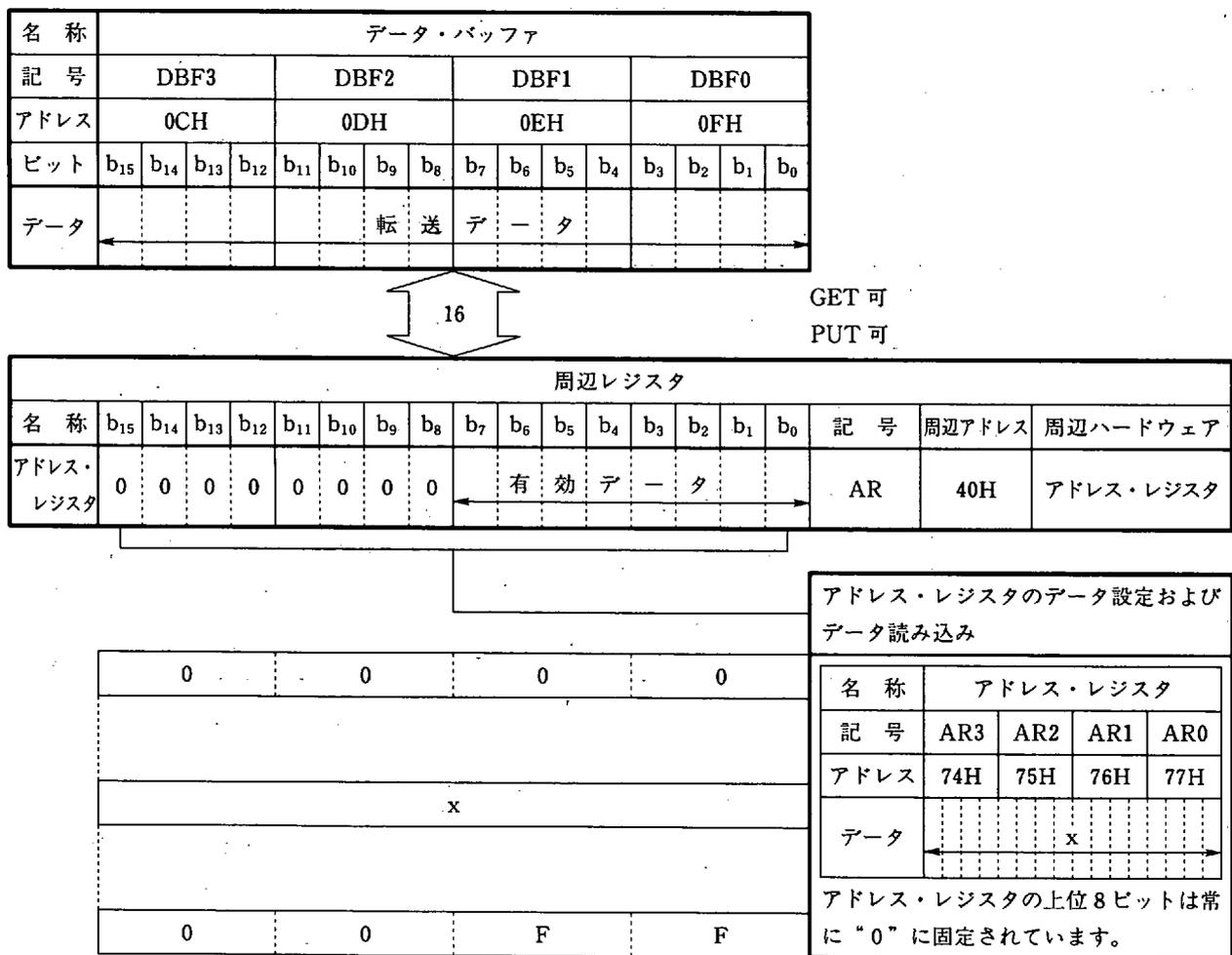
アドレス・レジスタはシステム・レジスタ(データ・メモリ上の74H-7FH番地)のアドレス74H-77Hに配置されたレジスタで、プログラム・メモリのアドレスを操作するためのレジスタです。“8.システム・レジスタ(SYSREG)”の項を参照してください。

アドレス・レジスタは、データ・メモリ操作命令により直接データの操作が可能で、周辺ハードウェアの一部として、データ・バッファを介してのデータ転送も可能です。

すなわち、アドレス・レジスタはデータ・メモリ操作命令の他に、“PUT”および“GET”命令を用いることによりデータ・バッファを介してデータの読み込みおよびデータの書き込みが可能になります。

図10-9に、アドレス・レジスタとデータ・バッファの関係を示します。

図10-9 アドレス・レジスタとデータ・バッファの関係



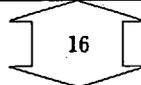
10.5.7 PLL データ・レジスタ

図 10-10 に PLL データ・レジスタの機能を示します。

PLL データ・レジスタは, PLL 周波数シンセサイザの分周比を設定します。パルス・スワロ方式は16ビットすべて有効であり, 上位12ビットがプログラム・カウンタに設定され, 下位4ビットがスワロ・カウンタに設定されます。

図 10-10 PLL データ・レジスタ

名称	データ・バッファ															
記号	DBF3				DBF2				DBF1				DBF0			
アドレス	0CH				0DH				0EH				0FH			
ビット	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
データ	← 転送データ →															



GET 可
PUT 可

周辺レジスタ																			
名称	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	記号	周辺アドレス	周辺ハードウェア
PLL データレジスタ					有	効	テ	ー	タ								PLLR	41H	PLL 周波数 シンセサイザ

PLL 周波数シンセサイザの分周比	
0 (0000H)	設定禁止
256 (0100H)	分周比 N : N=x
x	
2 ¹⁶ -1 (0FFFFH)	

10.6 データ・バッファ使用時の注意

10.6.1 書き込み専用、読み出し専用および未使用アドレスのデータ・バッファ操作時の注意

データ・バッファを介して周辺ハードウェアとデータ転送を行うとき、未使用周辺アドレスや書き込み専用周辺レジスタ (PUT のみ) および読み出し専用周辺レジスタ (GET のみ) に対して以下に示すようにデバイスの動作上、17K シリーズのアセンブラおよびエミュレータ使用時に注意が必要です。

(1) デバイス動作

書き込み専用周辺レジスタを読み出すと“不定な値”が読み出されます。

読み出し専用周辺レジスタに書き込みを行っても何も変化しません。

未使用アドレスを読み出すと“不定な値”が読み出され、書き込んでも何も変化しません。

(2) アセンブラ使用時

書き込み専用周辺レジスタを読み出す命令に“エラー”が発生します。

読み出し専用周辺レジスタに書き込む命令に“エラー”が発生します。

未使用アドレスを読み出す命令、書き込む命令に“エラー”が発生します。

(3) エミュレータ使用時 (パッチ処理等で命令を実行したとき)

書き込み専用周辺レジスタを読み出すと“不定な値”が読み出されます。“エラー”は発生しません。

読み出し専用周辺レジスタに書き込みを行っても何も変化しません。“エラー”は発生しません。

未使用アドレスを読み出すと“不定な値”が読み出され、書き込んでも何も変化しません。“エラー”は発生しません。

10.6.2 周辺レジスタのアドレスと予約語

17Kシリーズのアセンブラを使用するうえでは例1に示すように“PUT p, DBF”命令および“GET DBF, p”命令で指定する周辺アドレス“p”を直接（数値で）指定しても“エラー”は発生しません。

ただし、この方法はプログラムのバグ低減のためには好ましくありません。

したがってアセンブラ疑似命令であるシンボル定義命令を用いて例2に示すように周辺アドレスをシンボル定義しておく必要があります。

ここで、シンボル定義を簡略化するため、アセンブラ内にあらかじめ“予約語”として周辺アドレスが定義されています。

したがって予約語を用いれば、例3に示すようにシンボル定義を行うことなくプログラムを作成できます。

周辺レジスタの予約語は表10-1に示した“記号”および図10-4～10-10に示す“記号”の項に示してあります。

例1.

```
PUT      02H,      DBF ;周辺アドレスを 02H や 03H で直接指定してもアセンブラはエラー
GET      DBF,      03H ;とならないが、プログラムのバグ低減のためには好ましくない。
```

2.

```
SIO0DATA DAT      03H ;シンボル定義命令により SIO0DATA を 03H に割り当てる。
PUT      SIO0DATA, DBF ;
```

3.

```
PUT      SIO0SFR      ;予約語である“SIO0SFR”を用いればシンボル定義を行う
                          ;必要はない。
```

11. 割り込み

割り込みは、周辺ハードウェア (INT_{NC} 端子、タイマ、 $\overline{V_{SYNC}}$ 端子およびシリアル・インタフェース) からの要求により、現在実行しているプログラムを一時中断し、あらかじめ決められた番地 (ベクタ・アドレスと呼ぶ) へプログラムの流れを移します。

11.1 割り込みブロックの構成

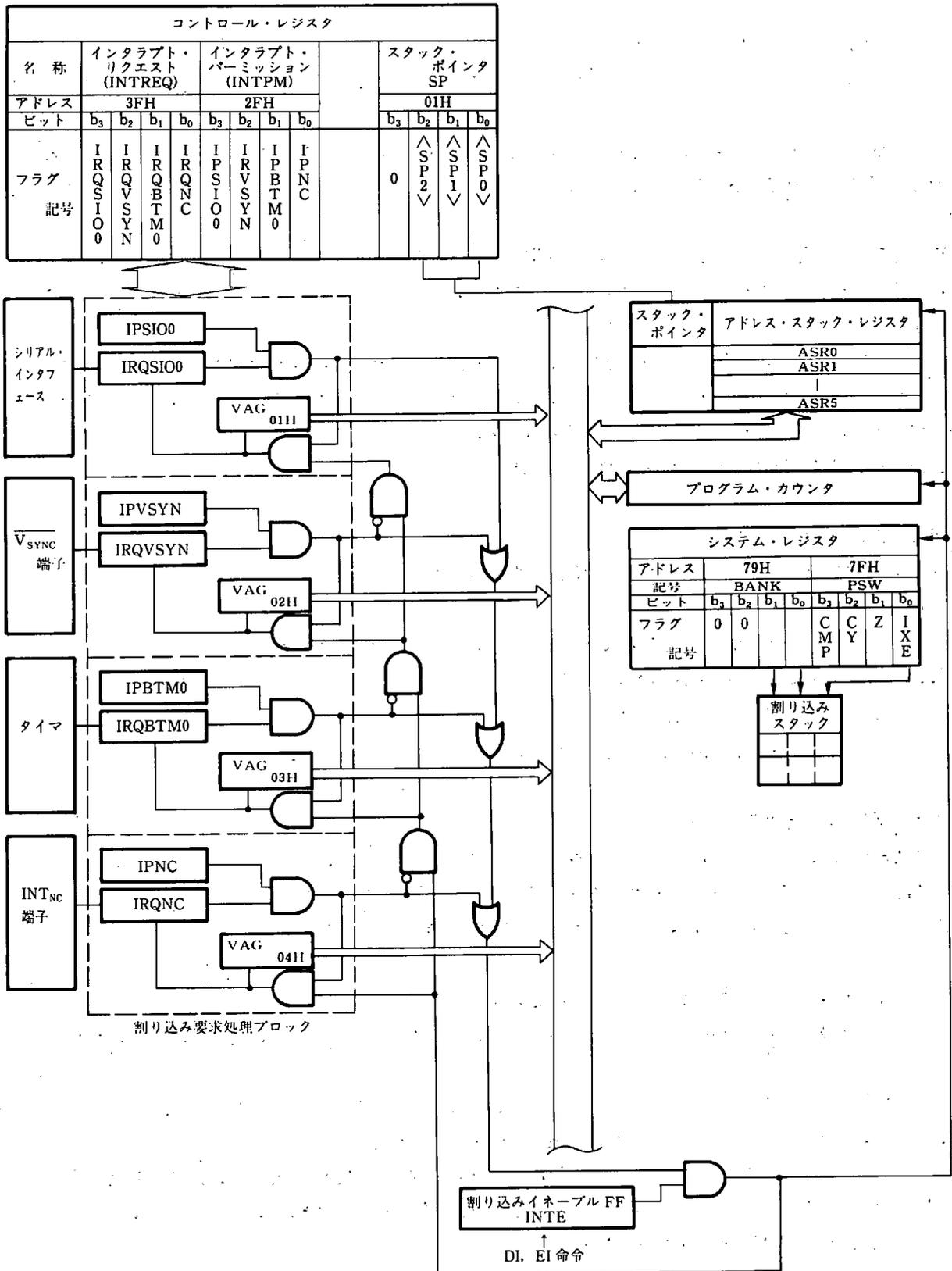
図 11-1 に割り込みブロックの構成を示します。

図 11-1 に示すように、割り込みブロックは INT_{NC} 端子、タイマ、 $\overline{V_{SYNC}}$ 端子およびシリアル・インタフェースの各周辺ハードウェアから出力された割り込み要求を制御する各“割り込み要求制御ブロック”と、すべての割り込みの許可を設定する“割り込みイネーブル・フリップフロップ (INTE)”と、割り込みが受け付けられたときに制御される“スタック・ポインタ”、“アドレス・スタック・レジスタ”、“プログラム・カウンタ”および“割り込みスタック”から構成されています。

各周辺ハードウェアの“割り込み要求処理ブロック”は、“各割り込み要求を検出するフリップフロップ (IRQ ×××)”と、“各割り込みの許可を設定するフリップフロップ (IP ×××)”および、割り込み受け付け時のベクタ・アドレスを指定する“ベクタ・アドレス・ジェネレータ (VAG)”から構成されています。

IRQ ××× フリップフロップおよび IP ××× フリップフロップは、コントロール・レジスタのインタラプト・リクエストおよびインタラプト・パーミッションの各フラグと 1 対 1 に対応しています。

図 11-1 割り込みブロックの構成



11.2 割り込みの機能

割り込み機能を使用できる周辺ハードウェアは、INT_{NC} 端子、タイマ、 $\overline{V}_{\text{SYNC}}$ 端子およびシリアル・インタフェースです。

割り込み機能とは、これらの周辺ハードウェアが、ある条件を満たしたとき（たとえばINT_{NC} 端子に立ち下がり信号が加えられる）に、そのとき実行していたプログラムを一時中断し、専用の処理プログラムを実行させるための機能です。

このとき、周辺ハードウェアからの割り込み信号を“割り込み要求”と呼び、割り込み信号が出力されることを“割り込み要求が発行される”と言います。また、割り込み専用の処理プログラムを“割り込み処理ルーチン”と呼びます。

割り込みが受け付けられると、各割り込み要因別に決められたプログラム・メモリ・アドレス（ベクタ・アドレス）へ分岐します。したがって、このベクタ・アドレスから、各割り込み処理ルーチンを開始すればよいことになります。

割り込み機能は割り込みを受け付けるまでの処理と割り込みを受け付けたあとの処理に分けられます。すなわち、各周辺ハードウェアからの割り込み要求に対してその割り込みを受け付けるまでの機能と、割り込みを受け付けてからベクタ・アドレスへ分岐し、さらに割り込み前のプログラムに復帰する機能に分けられます。

11.2.1 - 11.2.8 に図 11-1 に示した各ブロックの機能を示します。

11.2.1 周辺ハードウェア

割り込み機能を持つ周辺ハードウェアはINT_{NC} 端子、タイマ、 $\overline{V}_{\text{SYNC}}$ 端子およびシリアル・インタフェースの4種類です。

各周辺ハードウェアには、割り込み要求を発行するための条件を設定できます。

たとえばINT_{NC} 端子は、INT_{NC} 端子に加えられた信号の立ち上がりエッジにより要求を発行するか、立ち下がりエッジにより要求を発行するかを選択できるようになっています。

各周辺ハードウェアの割り込み要求発行条件の詳細については 11.3 - 11.7 を参照してください。

11.2.2 割り込み要求処理ブロック

割り込み要求処理ブロックは各周辺ハードウェアごとに設けられており、それぞれの割り込み要求の有無、割り込みの許可および割り込み受け付け時のベクタ・アドレスの生成を行いません。

以下の 11.2.3 - 11.2.8 に割り込み要求処理ブロックの各フラグについて説明します。

11.2.3 インタラプト・リクエスト・フラグ (IRQ ×××)

各周辺ハードウェアから割り込み要求が発行されるとセット (1) され、割り込みが受け付けられるとリセット (0) されます。

コントロール・レジスタのインタラプト・リクエスト・レジスタの各フラグに1対1に対応しているため、ウィンドウ・レジスタを介して読み出しおよび書き込みを行うことが可能です。

ウィンドウ・レジスタを介して“1”を書き込んだ場合も、割り込み要求が発行されたことと同等になります。

一度このフラグがセットされると、対応する割り込みが受け付けられるかまたはウィンドウ・レジスタを介して“0”が書き込まれるまでリセットされません。

複数の割り込み要求が同時に発行された場合でも、受け付けられなかった割り込みに対するインタラプト・リクエスト・フラグはリセットされません。

パワーオン・リセット時、クロック・ストップ命令実行時およびCEリセット時はこれらのフラグはリセット (0) されます。

11.2.4 インタラプト・パーミッション・フラグ (IP×××)

各周辺ハードウェアごとの割り込みの許可を設定します。

これらのフラグがセット (1) されており、かつ対応するインタラプト・リクエスト・フラグがセットされると割り込み要求を出力します。

コントロール・レジスタのインタラプト・パーミッション・レジスタの各フラグに1対1に対応しているため、ウインドウ・レジスタを介して読み出しおよび書き込みを行います。

パワーオン・リセット時、クロック・ストップ時およびCEリセット時はこれらのフラグはリセット (0) されます。

11.2.5 ベクタ・アドレス・ジェネレータ (VAG)

各周辺ハードウェアの割り込みが受け付けられたときに、受け付けられた割り込み要因に対するプログラム・メモリの分岐アドレス (ベクタ・アドレス) を生成します。

各割り込み要因に対するベクタ・アドレスを表 11-1 に示します。

表 11-1 割り込みベクタ・アドレス

割り込み要因	ベクタ・アドレス
INT _{NC} 端子	04H
タイマ	03H
$\overline{V}_{\text{SYNC}}$ 端子	02H
シリアル・インタフェース	01H

11.2.6 割り込みイネーブル・フリップフロップ (INTE)

割り込みイネーブル・フリップフロップは、4種類の割り込みのすべての割り込みの許可を設定します。

このフリップフロップがセット(1)されているときに各割り込み要求処理ブロックから“1”が出力されると、このフリップフロップから“1”が出力され、割り込みが受け付けられます。

このフリップフロップがリセット(0)されているときは各割り込み要求処理ブロックから“1”が出力されても割り込みは受け付けられません。

このフリップフロップのセットおよびリセットは専用命令である“EI”命令(セット)と“DI”命令(リセット)を使用します。

“EI”命令が実行されると、“EI”命令に続いて実行された命令が完了した時点でこのフリップフロップがセットされ、“DI”命令が実行されると“DI”命令実行サイクル中にリセットされます。

割り込みイネーブル・フリップフロップがセットされている状態(EI状態)で割り込みが受け付けられるとその時点でこのフリップフロップはリセットされます(DI状態)。

“DI状態中”に“DI”命令を実行しても、または“EI状態中”に“EI”命令を実行しても何ら影響は与えません。

パワーオン・リセット時、クロック・ストップ時およびCEリセット時はこのフラグはリセット(DI状態)されます。

11.2.7 スタック・ポインタ、アドレス・スタック・レジスタとプログラム・カウンタ

アドレス・スタック・レジスタは、割り込み処理ルーチンからの復帰時の戻り番地を退避します。

スタック・ポインタは6個のアドレス・スタック・レジスタ(ASR0-ASR5)のうちどのレジスタを使用するかを指定します。

すなわち、割り込みが受け付けられるとスタック・ポインタの値を-1し、そのときのプログラム・カウンタの値をスタック・ポインタで指定されるアドレス・スタック・レジスタに退避します。次に、割り込み処理ルーチンの処理を実行したあと、専用復帰命令である“RETI”命令が実行されると、スタック・ポインタで指定されるアドレス・スタック・レジスタの内容をプログラム・カウンタに復帰し、スタック・ポインタの値を+1します。

“4. スタック”の項も参照してください。

11.2.8 割り込みスタック

割り込みスタックは割り込み受け付け時に、システム・レジスタの内のバンク・レジスタとインデックス・イネーブル・フラグの内容を退避します。

割り込みが受け付けられ、バンク・レジスタとインデックス・イネーブル・フラグが退避されると、システム・レジスタ上のバンク・レジスタとインデックス・イネーブル・フラグはリセット(0)されます。

割り込みスタックはバンク・レジスタとインデックス・イネーブル・フラグの内容を2レベルまで退避することができます。したがって、ある割り込み処理ルーチンの中で別の割り込みを受け付けるような多重割り込みを2レベルまで行うことができます。

割り込みスタックの内容は、割り込み処理ルーチンからの専用復帰命令である“RETI”命令を実行することによりシステム・レジスタのバンク・レジスタとインデックス・イネーブル・フラグに復帰されます。

“4. スタック”の項も参照してください。

11.3 割り込み受け付け動作

11.3.1 割り込み受け付け動作と優先順位

割り込みを受け付けるまでの動作を以下に示します。

- (1) 各周辺ハードウェアは、割り込み条件が満たされる（たとえば、INT_{NC}端子に立ち下がり信号が入力される）と、各割り込み要求ブロックに割り込み要求信号を出力します。
- (2) 各割り込み要求ブロックは、各周辺ハードウェアからの割り込み要求信号を受け付けると、対応する IRQ × × × フラグ（たとえばINT_{NC}端子であればIRQNC）をセット(1)します。
- (3) 各インタラプト・リクエスト・フラグがセットされたときに各 IRQ × × × に対応するインタラプト・パーミッション・フラグ、たとえばIRQNCフラグであればIPNCフラグがセット(1)されていると、各割り込み要求ブロックから“1”を出力します。
- (4) 各割り込み要求ブロックから出力された信号はOR回路を介して割り込みイネーブル・フリップフロップへ入力されます。

この割り込みイネーブル・フリップフロップは、“EI”命令によりセット(1)され、“DI”命令によりリセットされます。

割り込みイネーブル・フリップフロップがセットされているときに、各割り込み要求ブロックから“1”が出力されると、割り込みイネーブル・フリップフロップから“1”が出力され、割り込みが受け付けられます。

割り込みが受け付けられると図11-1に示したように割り込みイネーブル・フリップフロップの出力は、AND回路を介して各割り込み要求ブロックへ入力されます。

各割り込み要求ブロックへ入力された信号によりインタラプト・リクエスト・フラグがリセットされ、かつ各割り込みに対するベクタ・アドレスが出力されます。

このとき、割り込み要求ブロックから“1”が出力されていると、割り込み受け付け信号は次段に伝達されないため、同時に複数の割り込み要求が発行されたときは、以下に示す優先順位で割り込みが受け付けられます。

(DMA) > INT_{NC}端子 > タイマ > $\overline{V_{SYNC}}$ 端子 > シリアル・インタフェース

この優先順位を“ハードウェア優先順位”と呼びます。

図11-2に割り込み受け付け動作のフロー・チャートを示します。

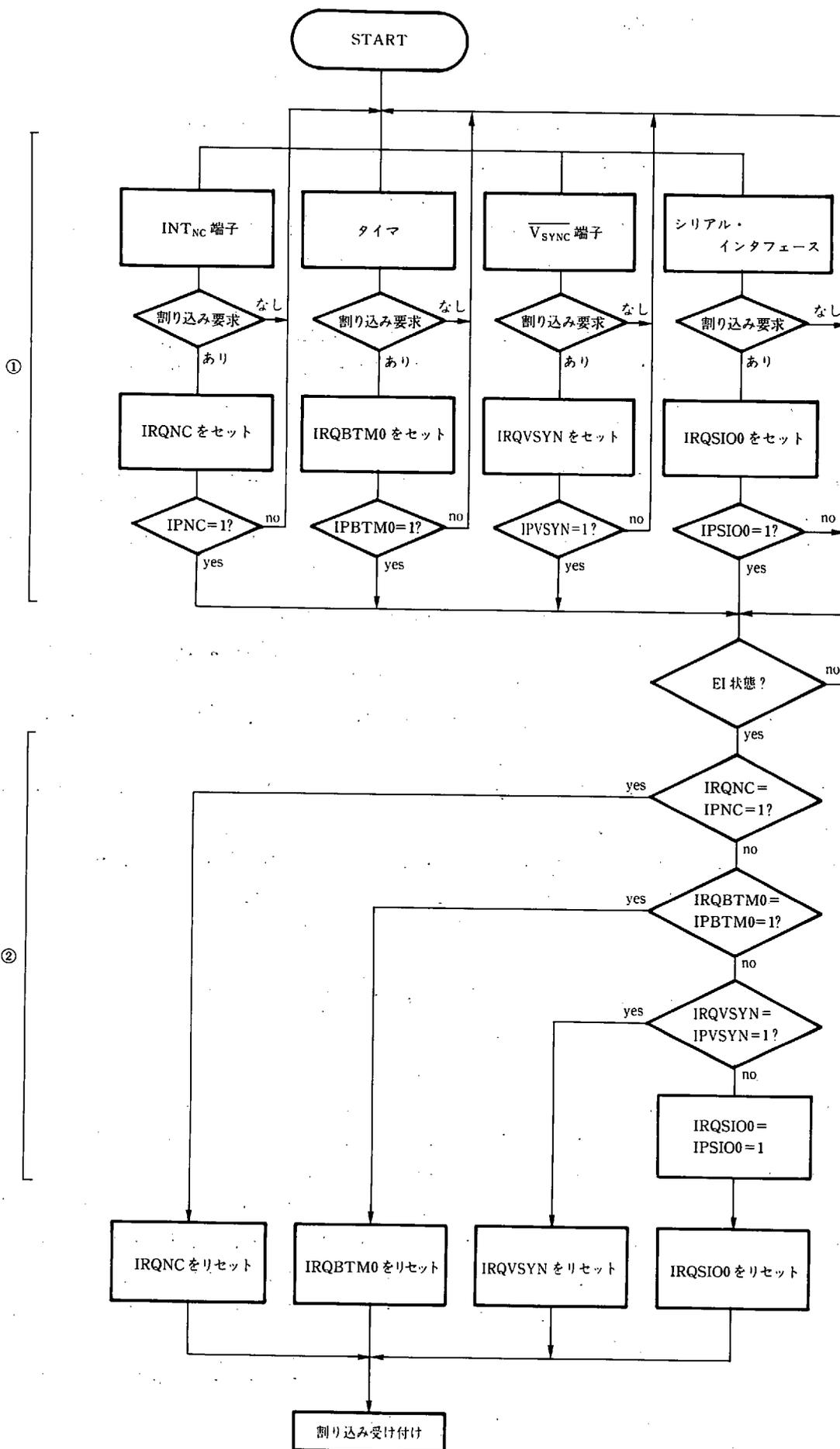
図11-2に示す①の処理は常に並列に処理されるため、まったく同時に複数の割り込み要求が発生したときは、各インタラプト・リクエスト・フラグは同時にセットされます。

ところが②の処理は各インタラプト・パーミッション・フラグにより優先順位をもって処理されます。

すなわち、インタラプト・パーミッション・フラグがセットされていなければその割り込み要因に対する割り込みは受け付けられません。インタラプト・パーミッション・フラグはプログラムでセットまたはリセットできるため、インタラプト・パーミッション・フラグをリセットしておけば、ハードウェア優先順位の高い割り込みを禁止することが可能です。

このインタラプト・パーミッション・フラグによる割り込みを“マスクابل割り込み”と呼びます。またマスクابل割り込みは、ハードウェア優先の高い割り込み要因をプログラムで禁止できるため、“ソフトウェア優先順位”とも呼びます。

図 11-2 割り込み受け付け動作のフロー・チャート



11.3.2 割り込み受け付け時のタイミング・チャート

図 11-3 に割り込み受け付け時のタイミング・チャートを示します。

図 11-3 の(1)は1種類の割り込みによるタイミング・チャートです。

(1)の(a)はインタラプト・リクエスト・フラグが最後にセット(1)された場合であり、(1)の(b)はインタラプト・パーミッション・フラグが最後にセット(1)された場合のタイミング・チャートです。

どちらの場合も割り込みの受け付けは、インタラプト・リクエスト・フラグ、割り込みイネーブル・フリップフロップおよびインタラプト・パーミッション・フラグのすべてがセットされた時点で行われます。

最後にセットされたフラグもしくはフリップフロップが“MOV T DBF, @AR”命令の第1命令サイクルまたは、スキップ条件を満たした命令の場合は、それぞれ“MOV T DBF, @AR”命令の第2命令サイクルおよびスキップした命令(NOPになる)実行後に割り込みが受け付けられます。

割り込みイネーブル・フリップフロップのセットは“EI”命令が実行された次の命令サイクルでセットされます。

図 11-3 の(2)は複数の割り込みを使用するときのタイミング・チャートを示しています。

複数の割り込みを使用するときはインタラプト・パーミッション・フラグがすべてセットされていれば、ハードウェアで優先されている割り込みが先に受け付けられますが、プログラムでインタラプト・パーミッション・フラグを操作することにより、ハードウェアの優先度を変えることが可能になります。

なお、図 11-3 に示した“割り込みサイクル”とは、割り込みが受け付けられてからインタラプト・リクエスト・フラグのリセット、ベクタ・アドレスの指定、プログラム・カウンタの退避等を行うための特別なサイクルであり、1命令実行時間分(2 μs, IDC 動作時は12 μs)必要となります。詳しくは“11.4 割り込み受け付け後の動作”の項を参照してください。

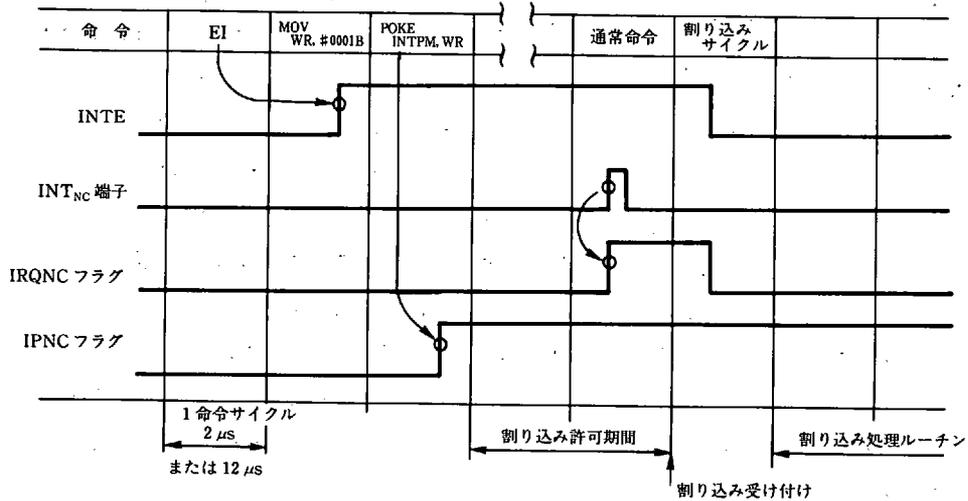
インタラプト・リクエスト・フラグは、“EI”命令やインタラプト・パーミッション・フラグとは関係なく、周辺ハードウェアの割り込み要求によりセット(1)されるため、プログラムによりインタラプト・リクエスト・フラグを検出することにより割り込み要求の有無を知ることができます。

図 11-3 割り込み受け付けタイミング・チャート (1/2)

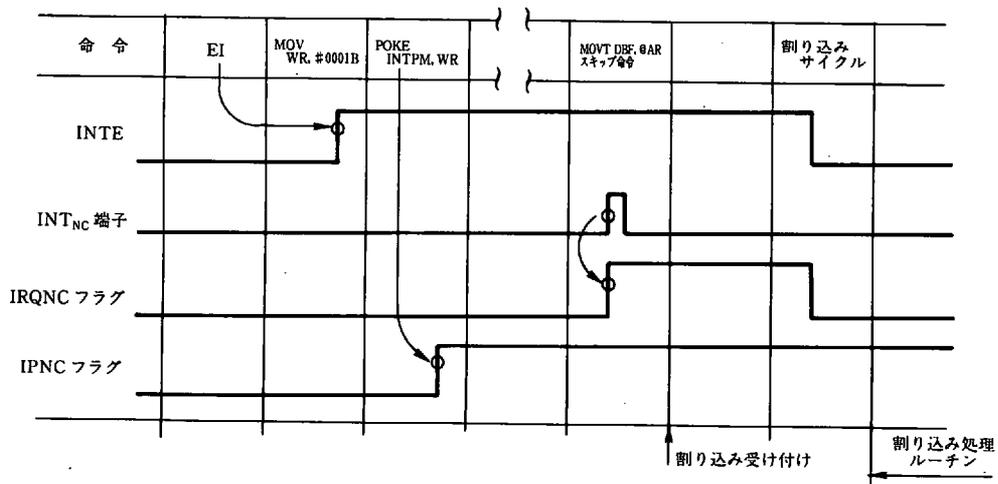
(1) 1種類の割り込み (例: INT_{NC} 端子の立ち上がり) を使用時

(a) インタラプト・パーミッション・フラグによる割り込みマスク時間がない場合

① 割り込み受け付け時が“MOV”命令およびスキップ条件を満たした命令でない通常命令の場合



② 割り込み受け付け時が“MOV”命令か、または“スキップ条件を満たした命令”の場合



(b) インタラプト・パーミッション・フラグによる割り込み保留期間がある場合

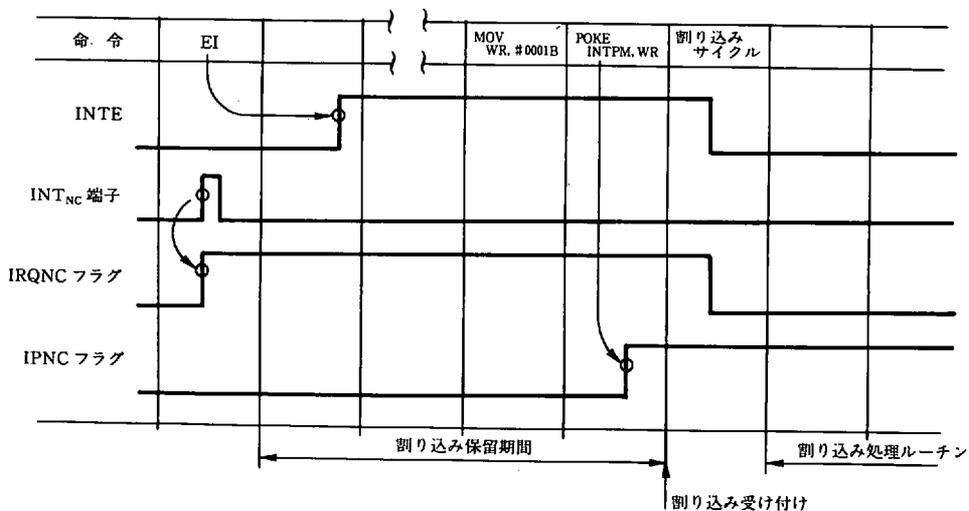
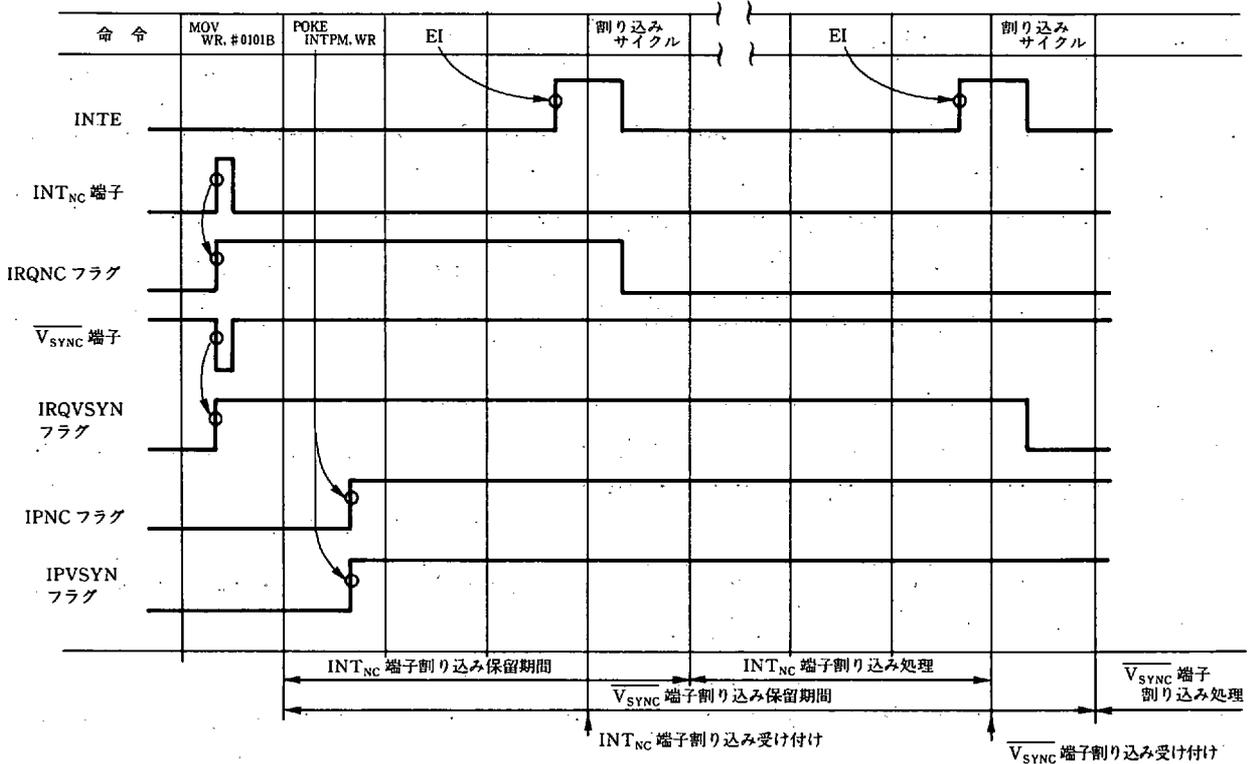


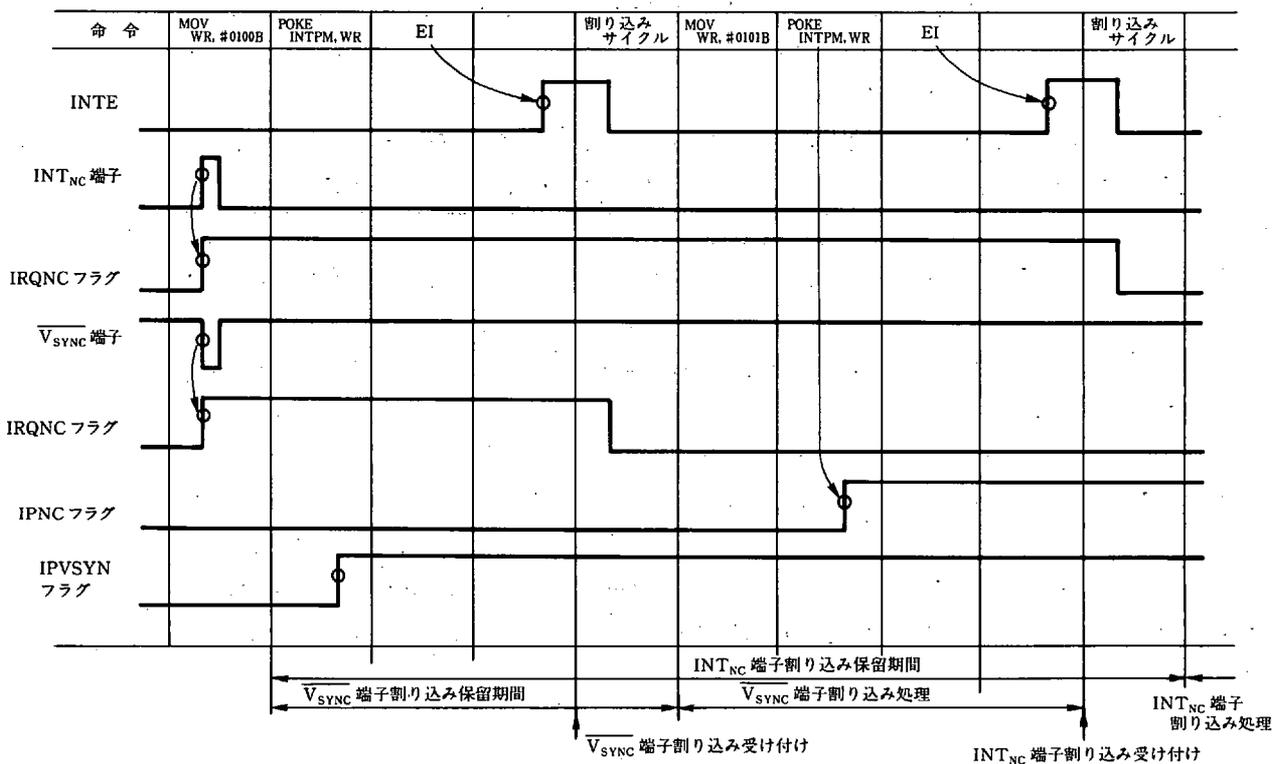
図 11-3 割り込み受け付けタイミング・チャート (2/2)

(2) 複数の割り込み (例: INT_{NC} 端子の立ち上がり, $\overline{V}_{\text{SYNC}}$ 端子の立ち下がり) を使用時

(a) ハードウェア優先



(b) ソフトウェア優先



11.4 割り込み受け付け後の動作

割り込みが受け付けられると、以下に示す処理が順次自動的に実行されます。

- (1) 割り込みイネーブル・フリップフロップおよび受け付けられた割り込み要求に対応するインタラプト・リクエスト・フラグをリセットする。すなわち割り込み禁止状態になります。
- (2) スタック・ポインタの内容を-1する。
- (3) プログラム・カウンタの内容をスタック・ポインタで指定されるアドレス・スタック・レジスタへ退避する。
このときのプログラム・カウンタの内容は、割り込みが受け付けられた時点の次のプログラム・メモリ・アドレスになります。たとえば分岐命令であれば分岐先のアドレスとなり、サブルーチン・コール命令であればコールしたアドレスになります。スキップ命令でスキップ条件が満たされたときは、次の命令を“NOP”命令として実行してから割り込みが受け付けられるため、プログラム・カウンタの内容はスキップしたアドレスになります。
- (4) バンク・レジスタ (BANK: アドレス 79H) の下位 2 ビットと、インデクス・イネーブル・フラグ (IXE: アドレス 7FH のビット b_0) を、割り込みスタックへ退避する。
- (5) 受け付けられた割り込みに対応するベクタ・アドレス・ジェネレータの内容をプログラム・カウンタに転送する。すなわち割り込み処理ルーチンへ分岐する。

上記(1)-(5)の処理は通常の命令の実行を伴わない特別な 1 命令サイクル (2 μ s, IDC 動作時は 12 μ s) の間に実行されます。

この命令サイクルを“割り込みサイクル”と呼びます。すなわち割り込みが受け付けられてから、対応するベクタ・アドレスへ分岐するまでに 1 命令サイクルの時間が必要になります。

11.5 割り込み処理ルーチンからの復帰処理

割り込み処理ルーチンから、割り込みが受け付けられたときの処理へ復帰させるには、専用命令である“RETI”命令を使用します。“RETI”命令が実行されると以下に示す処理が順次自動的に実行されます。

- (1) スタック・ポインタで指定されるアドレス・スタック・レジスタの内容をプログラム・カウンタに復帰する。
- (2) 割り込みスタックの内容をバンク・レジスタの下位 2 ビットおよびインデクス・イネーブル・フラグのビット b_0 に復帰する。
- (3) スタック・ポインタの内容を+1する。

上記(1)-(3)の処理は“RETI”命令が実行される 1 命令サイクル中に処理されます。“RETI”命令とサブルーチン・リターン命令である“RET”および“RETSK”命令との違いは、上記(2)のバンク・レジスタおよびインデクス・イネーブル・フラグの復帰動作の違いのみです。

11.6 割り込み処理ルーチン

割り込みは、割り込みが許可されているプログラム領域であれば、割り込み要求が発行された時点でそのとき実行しているプログラムに関係なく受け付けられます。

したがって、割り込み処理を実行したのちにもとのプログラムに復帰させるときは、あたかも割り込み処理が行われていないような状態に戻しておく必要があります。

たとえば割り込み処理中に算術演算等が実行されると、キャリー・フラグ等の内容は割り込みが受け付けられる前と変化してしまう可能性があり、復帰後のプログラムは判断を誤ることになります。

このため、少なくとも割り込み処理ルーチン内で操作する可能性があるシステム・レジスタやコントロール・レジスタ等については、割り込み処理ルーチンの中で退避および復帰操作を行う必要があります。

なお、割り込み処理中に他の割り込みを許可する場合（多重割り込み）の処理については“11.9 多重割り込み”の項も参照してください。

11.6.1 退避処理

例に割り込みルーチンでの退避処理例を示します。

システム・レジスタの中でハードウェアにより自動的に退避されるものはバンク・レジスタとインデックス・イネーブル・フラグのみであるため、例に示すように必要であればプログラムにより他のシステム・レジスタ等を退避します。

例に示すように、システム・レジスタ等の退避や復帰処理は“POKE”命令と“PEEK”命令を使用すると便利です。

“PEEK”および“POKE”命令以外にも転送命令（LD r, m や ST m, r 等）を使用する方法もありますが、割り込みが受け付けられた時点でジェネラル・レジスタのロウ・アドレスが一定でない場合などは転送命令による退避方法を使用すると退避するためのデータ・メモリ・アドレスを指定することが困難です。

なぜなら、ジェネラル・レジスタ自身を退避するために転送命令を使用すると、ジェネラル・レジスタのアドレスが一定でない場合退避するアドレスも一定でなくなってしまうため、割り込み許可ルーチン中は少なくともジェネラル・レジスタを固定して使用しなければならなくなるためです。

ところが“PEEK”命令や“POKE”命令で制御されるレジスタ・ファイルはジェネラル・レジスタの内容とは無関係にアドレス指定され、かつレジスタ・ファイルの40H-7FH番地は、そのとき選択されているバンクのデータ・メモリと重なっているため、バンクを指定するだけで各システム・レジスタ等を退避することができます。

例ではウィンドウ・レジスタとジェネラル・レジスタ・ポインタを“PEEK”および“POKE”命令で退避したのち、ジェネラル・レジスタをBANK0のロウ・アドレス07Hに再指定したあとに“ST”命令で他のシステム・レジスタ等を退避しています。

また、図11-4に“PEEK”および“POKE”命令による退避動作例を示します。

11.6.2 復帰処理

例に復帰処理例を示します。

復帰処理は、11.6.1で説明した退避処理と反対の動作を行えばよいことになります。

また、割り込みが受け付けられたときは当然割り込み許可状態（EI状態）であったはずですから“RETI”命令を実行する前に“EI”命令を実行する必要があります。

“EI”命令は次の“RETI”命令が実行されたあとに割り込みイネーブル・フリップフロップをセット(1)します。したがって、割り込み受け付け前のプログラムに復帰したのちに割り込み許可状態になります。

11.6.3 割り込み処理ルーチンでの注意

割り込み処理ルーチンでは以下に示す点に注意してください。

(1) ハードウェアにより退避されるデータについて

バンク・レジスタおよびインデックス・イネーブル・フラグは、割り込みスタックに退避された後すべて“0”にリセットされます。

(2) ソフトウェアにより退避したデータについて

ソフトウェアにより退避したデータは、退避後でもリセットされません。

特に、プログラム・ステータス・ワードであるBCDフラグ、コンペア・フラグ、キャリー・フラグおよびゼロ・フラグや、メモリ・ポインタ・イネーブル・フラグについては割り込み受け付け前の値をそのまま保持しているため必要に応じてイニシャライズしてください。

例 割り込み処理ルーチン内の状態退避方法

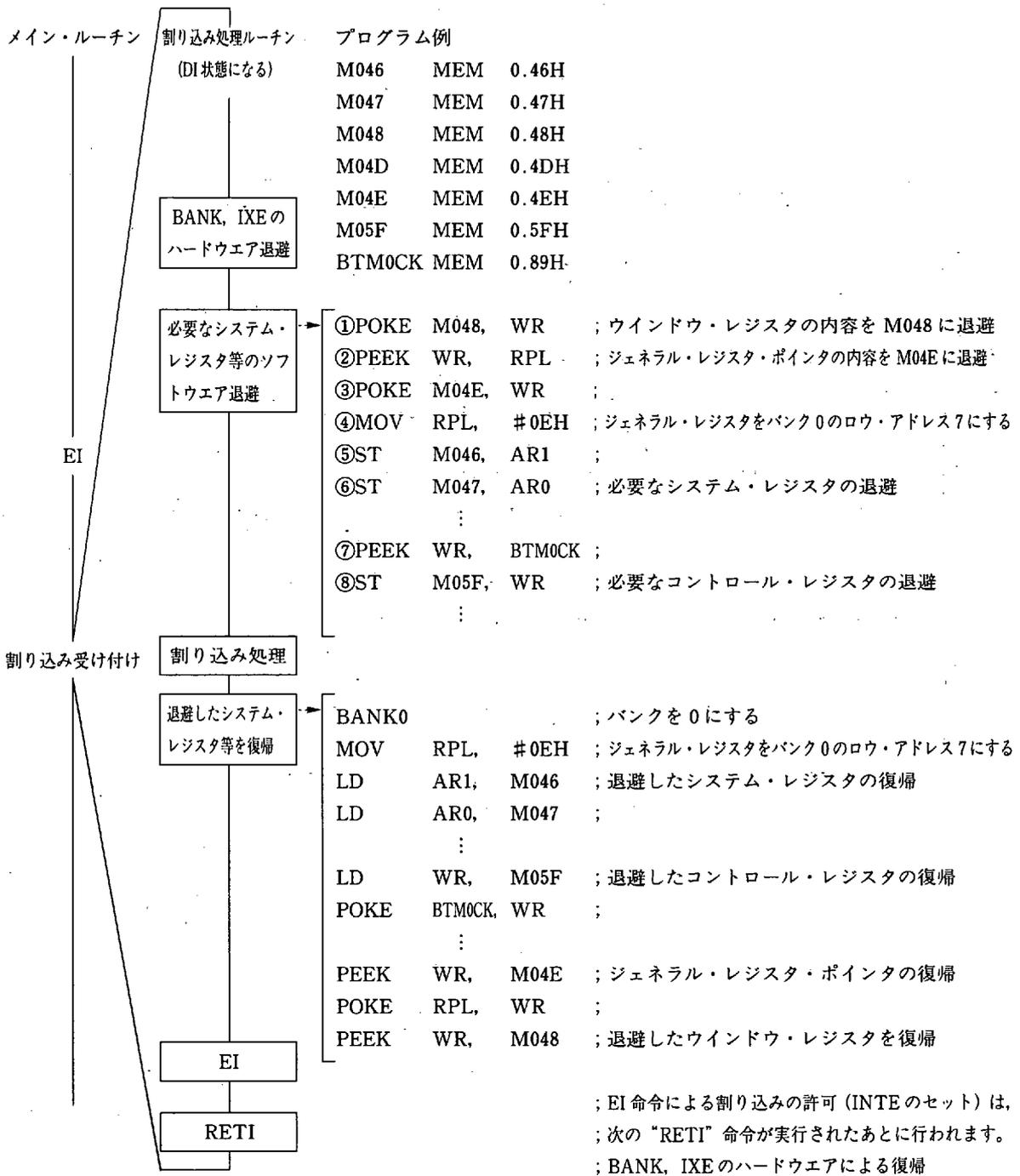
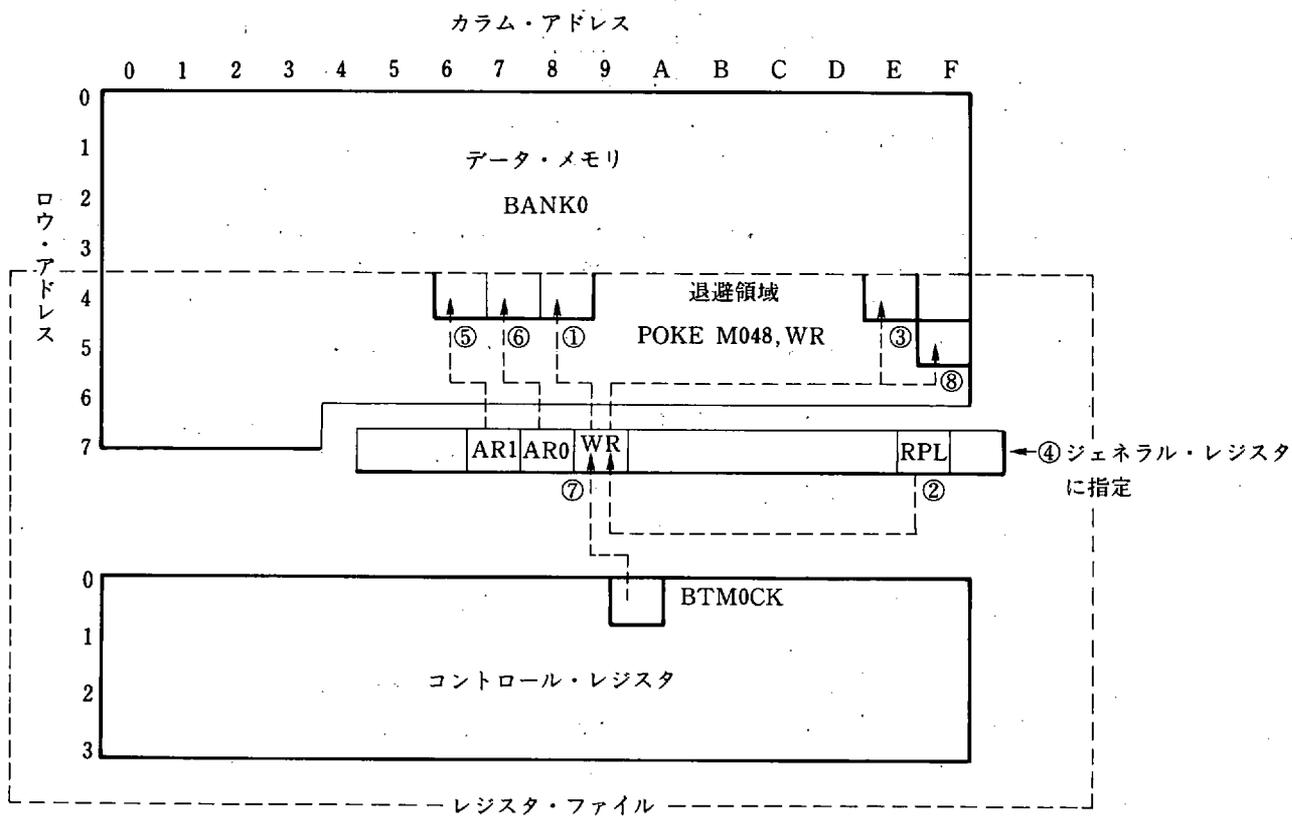


図 11-4 ウィンドウ・レジスタを用いたシステム・レジスタやコントロール・レジスタの退避動作

①～⑧は例のプログラム例の No. です。

①～⑧は例のプログラム例の No. です。



11.7 外部割り込み (INT_{NC} 端子, $\overline{V}_{\text{SYNC}}$ 端子)

外部割り込みは INT_{NC} 端子および $\overline{V}_{\text{SYNC}}$ 端子による 2 系統があります。

これらの端子に加えられた信号の立ち上がりまたは立ち下がりエッジにより割り込み要求を発行します。

11.7.1 構成

図 11-5 に INT_{NC} 端子と $\overline{V}_{\text{SYNC}}$ 端子の構成を示します。

図 11-5 に示すように、INT_{NC} 端子および $\overline{V}_{\text{SYNC}}$ 端子から入力された信号は、それぞれ INTNC ラッチおよび INTVSYN ラッチに入力されるとともに各エッジ検出回路に入力されます。

エッジ検出回路は各端子から入力された信号と、IEGNC および IEGVSYN フリップフロップからの入力により割り込み要求信号を出力します。

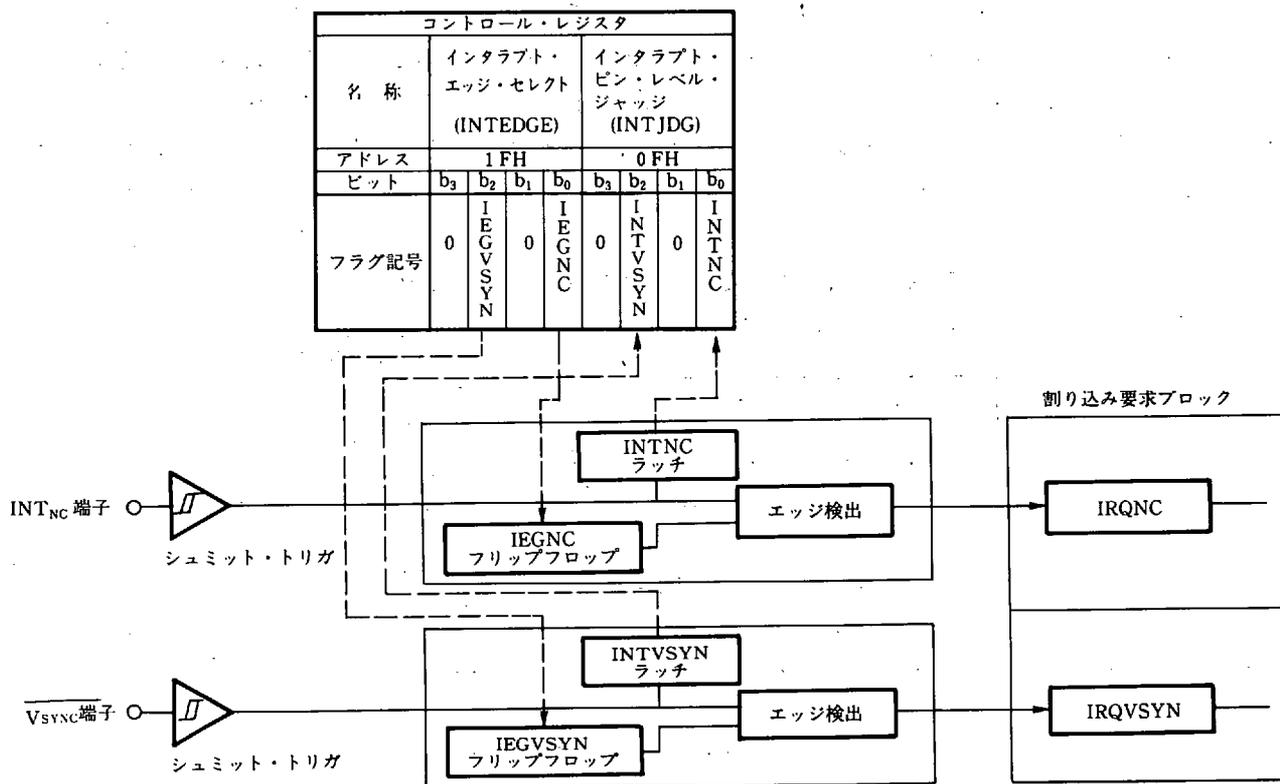
IEGNC および IEGVSYN フリップフロップは、コントロール・レジスタのインタラプト・エッジ・セレクト・レジスタ (INTEEDGE: アドレス 1FH) の IEGNC フラグおよび IEGVSYN フラグと 1 対 1 に対応しています。

INTNC および INTVSYN ラッチは、コントロール・レジスタのインタラプト・ピン・レベル・ジャッジ・レジスタ (INTJJDG: アドレス 0FH) の INTNC フラグおよび INTVSYN フラグと 1 対 1 に対応しています。

INT_{NC} 端子および $\overline{V}_{\text{SYNC}}$ 端子はノイズによる誤動作を防ぐため、シュミット・トリガ入力となっており、かつ、1 μs 未満のパルス入力は受け付けられません。

また、INT_{NC} 端子は受け付けパルス幅の設定が可能です (9.10 参照)。

図 11-5 INT₀ 端子および INT₁ 端子の構成



11.7.2 機能

INT_{NC} 端子および $\overline{V}_{\text{SYNC}}$ 端子は、各端子に加えられた立ち下がりまたは立ち上がりエッジにより割り込み要求を発行します。

立ち上がりエッジおよび立ち下がりエッジの選択は、コントロール・レジスタのインタラプト・エッジ・セレクト・レジスタの IEGNC および IEGVSYN フラグにより行います。

表 12-2 に IEGNC および IEGVSYN フラグと、割り込み要求発行エッジの関係を示します。

ここで以下の点に注意が必要です。

IEGNC および IEGVSYN フラグにより割り込み要求発行エッジを切り替えると、切り替えた瞬間に割り込み要求信号が発行される場合がある

という点です。これは表 11-3 に示すように、たとえば今 IEGNC フラグが“0”（立ち下がりエッジ）に設定されており、INT_{NC} 端子からハイ・レベルが入力されているとします。このとき IEGNC フラグをセット(1)すると、エッジ検出回路は立ち上がりエッジが入力されたと判断し、割り込み要求を発行してしまうため注意してください。

割り込み要求が発行されたあとの動作は“11.2 割り込みの機能”の項を参照してください。

INT_{NC} 端子および $\overline{V}_{\text{SYNC}}$ 端子に入力された信号は、図 11-5 に示したようにそれぞれ INTNC ラッチおよび INTVSYN ラッチに入力されるため、INTNC フラグおよび INTVSYN フラグを読み出すことにより入力された信号レベルを検出することが可能です。

この INTNC フラグおよび INTVSYN フラグは割り込みとは無関係にセットまたはリセットされるため割り込み機能を使用しない場合は2ビットの汎用入力ポートとして使用できることとなります。

また割り込みを許可していなければ、インタラプト・リクエスト・フラグ (IRQNC, IRQVSYN) を読み込むことにより立ち上がりまたは立ち下がりエッジを検出できる汎用ポートとして使用できます。ただしこの場合はインタラプト・リクエスト・フラグは自動的にリセットされませんので、プログラムによりリセットする必要があります。

表 11-2 IEGNC および IEGVSYN フラグと割り込み要求発行エッジ

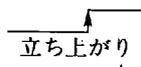
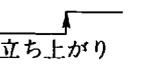
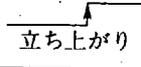
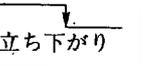
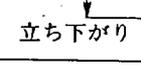
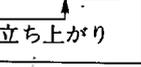
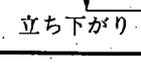
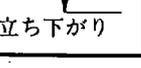
各フラグの値		各端子の割り込み要求発行エッジ	
IEGNC	INTVSYN	INT _{NC} 端子	$\overline{V}_{\text{SYNC}}$ 端子
0	0	 立ち上がり	 立ち上がり
0	1	 立ち上がり	 立ち下がり
1	0	 立ち下がり	 立ち上がり
1	1	 立ち下がり	 立ち下がり

表 11-3 IEGNC フラグの変更による割り込み要求の発行

IEGNC, IEGVSYN フラグ の変化	INT _{NC} 端子, $\overline{V_{SYNC}}$ 端子 の状態	割り込み要求発行の有無	IRQNC フラグの状態
1 → 0 (立ち下がり) (立ち上がり)	ロウ・レベル	発行されない	状態保持
	ハイ・レベル	発行される	セットされる
0 → 1 (立ち上がり) (立ち下がり)	ロウ・レベル	発行される	セットされる
	ハイ・レベル	発行されない	状態保持

11.8 内部割り込み (タイマ, シリアル・インタフェース)

内部割り込みには, タイマ, シリアル・インタフェースの2系統があります。

11.8.1 タイマ割り込み

タイマ割り込みは一定時間ごとに割り込み要求を発行することができます。

選択できる時間は 100 ms, 20 ms および 5 ms の3種類です。

詳しくは“12. タイマ機能”の項を参照してください。

11.8.2 シリアル・インタフェースの割り込み

シリアル・インタフェースの割り込みは, シリアル・アウトまたはシリアル・イン動作終了時に割り込み要求を発行することができます。

割り込み要求の発行は, おもにシリアル・クロックにより行います。

詳しくは“16. シリアル・インタフェース”の項を参照してください。

11.9 多重割り込み

多重割り込みとは、図 11-6 に示すようにある割り込み要因 A や B に対する割り込み処理中に、別の割り込み要因である C や D の割り込みを処理する割り込み方法です。

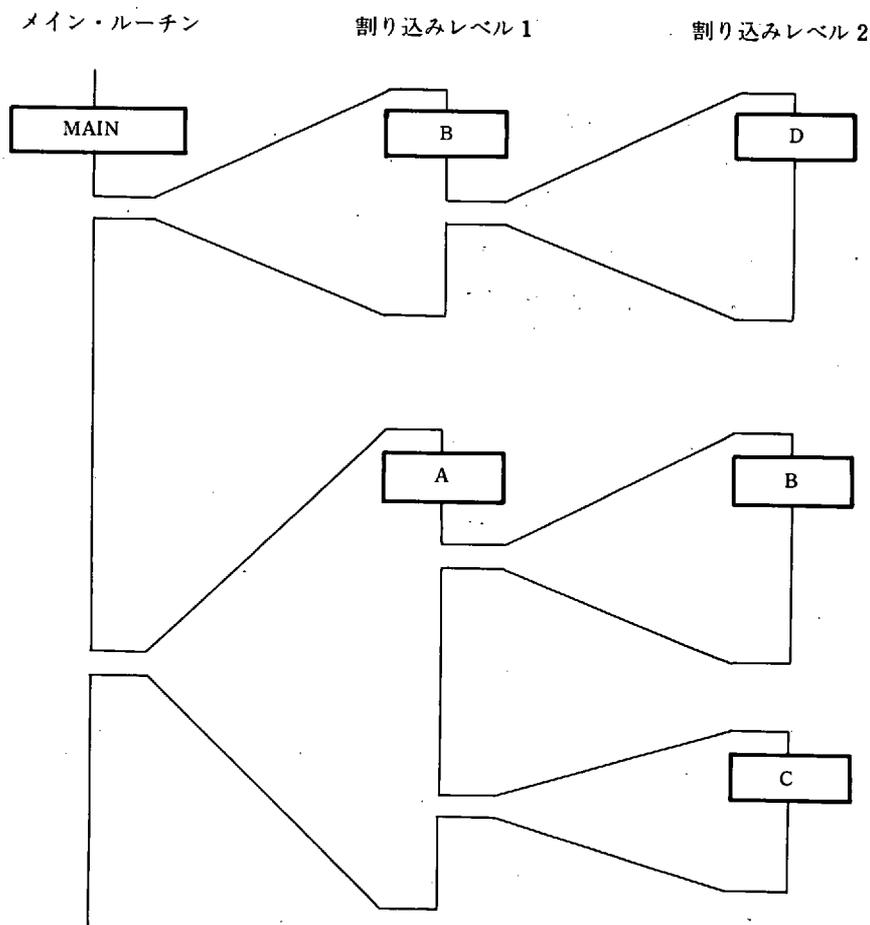
このときの割り込みの深さを割り込みレベルと呼びます。

多重割り込みを使用するときには、以下の点に注意が必要です。

- (1) 割り込み要因の優先度
- (2) 割り込み用スタックによる割り込みレベルの制限
- (3) アドレス・スタック・レジスタによる割り込みレベルの制限
- (4) システム・レジスタおよびコントロール・レジスタ等の退避

上記(1), (2), (3), (4)について 11.9.1 - 11.9.4 に説明します。

図 11-6 多重割り込み例



11.9.1 割り込み要因の優先順位

多重割り込みを使用する場合は割り込み要因の優先順位を決定しておく必要があります。

たとえば割り込み要因が A, B, C および D であるときの優先順位は

$$A=B=C=D$$

でもかまいませんし、

$$A<B<C<D$$

でもかまいません。

ただし、 $A=B=C=D$ の場合には、メイン・ルーチンでは常に A, B, C および D の割り込みが受け付けられますが、たとえば C の割り込みが受け付けられれば他の A, B および D の割り込みは禁止されるため多重割り込みの意味はなくなってしまいます。

$A<B<C<D$ の優先順位を持つときは A または B の割り込み処理実行中であっても C の割り込みは優先的に処理されなければならず、D の割り込みはさらに C の割り込み処理実行中であっても優先されなければなりません。

前述したような優先順位は“11.3 割り込み受け付け動作”の項で説明したように、ハードウェアによる優先順位でもかまいませんし、インタラプト・パーミッション・フラグを用いたソフトウェアによる優先順位でもかまいません。

多重割り込みにおいて優先順位を決定しておく必要性は、たとえば割り込み要因 A および B があり、A の要因は 10 ms ごとに要求を発行し、その割り込み処理時間が 4 ms であるとし、B の要因は 2 ms ごとに要求を発行し、その処理時間は 1 ms であるとし、

このとき A および B に優先順位がなかったと仮定すると、たまたま B の割り込み処理中に A の割り込み要求により A の割り込み処理が実行されると、B の割り込み処理は数回分行われなくなってしまいます。

一般に割り込みは緊急性が高い処理に使用されることが多いため、前述したような場合等は $A<B$ の優先順位を付けて B の割り込み処理中は A の割り込み処理を禁止しておき、また A の割り込み処理中であっても B の割り込みは受け付けるようなプログラムが必要となります。

緊急性のない目的で多重割り込みを使用するときは、必ずしも優先順位を付けておく必要はありませんが、“11.9.2” および “11.9.3” 項に示すような多重割り込みレベルの制限を越える割り込み要因数が存在する場合は必ず、割り込みレベルを越えないように優先順位を決定しておく必要があります。

11.9.2 割り込み用スタックによる割り込みレベルの制限

割り込み用スタックには、システム・レジスタのバンク・レジスタとインデクス・イネーブル・フラグの内容が自動的に退避されます。

割り込みスタックの動作を図 11-7 の (a) に示します。

バンク・レジスタとインデクス・イネーブル・フラグは割り込み用スタックに退避されると同時にすべてリセットされます。

割り込み用スタックは 2 レベルであるため、2 レベルを越える多重割り込みを行うとバンク・レジスタおよびインデクス・イネーブル・フラグは図 11-7 の (b) に示すように正常に復帰できなくなります。

すなわち、2 レベルを越える多重割り込みは使用できないことになります。

ただし、図 11-8 に示すようにたとえば割り込みが許可されているメイン・ルーチンにおいて必ずバンク・レジスタおよびインデクス・イネーブル・フラグが固定されておりかつ、割り込みの優先度が明確になっているような多重割り込みにおいては、サブルーチン・リターン命令である“RET”命令を使用することにより 2 レベル以上の

多重割り込みが可能になります。

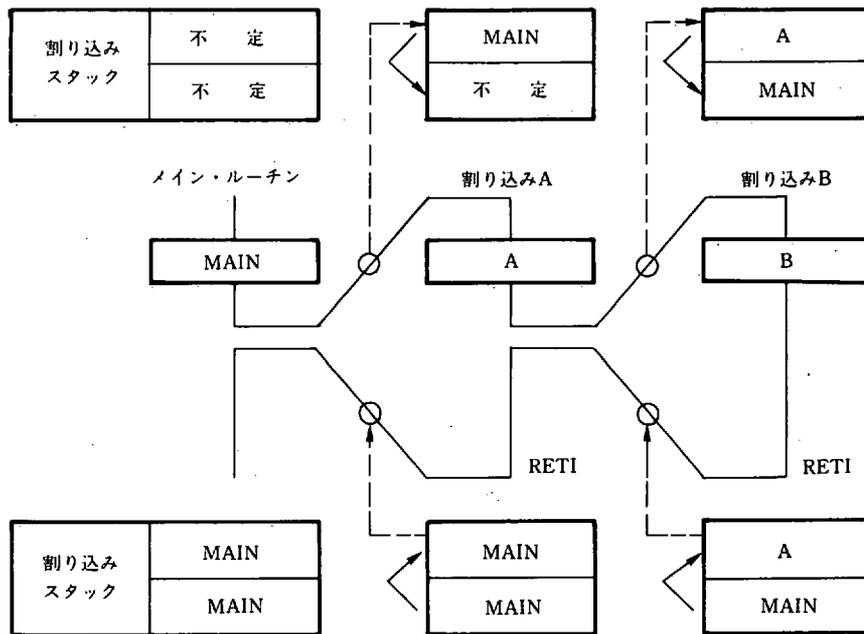
2レベルを越える多重割り込みでは、図11-8および図11-9に示すようにデバイスの動作とエミュレータでの動作が異なるため、注意が必要です。

すなわち、デバイスの割り込みスタック時の動作は“はき捨て型”であり、エミュレータの動作は“回転型”になっています。

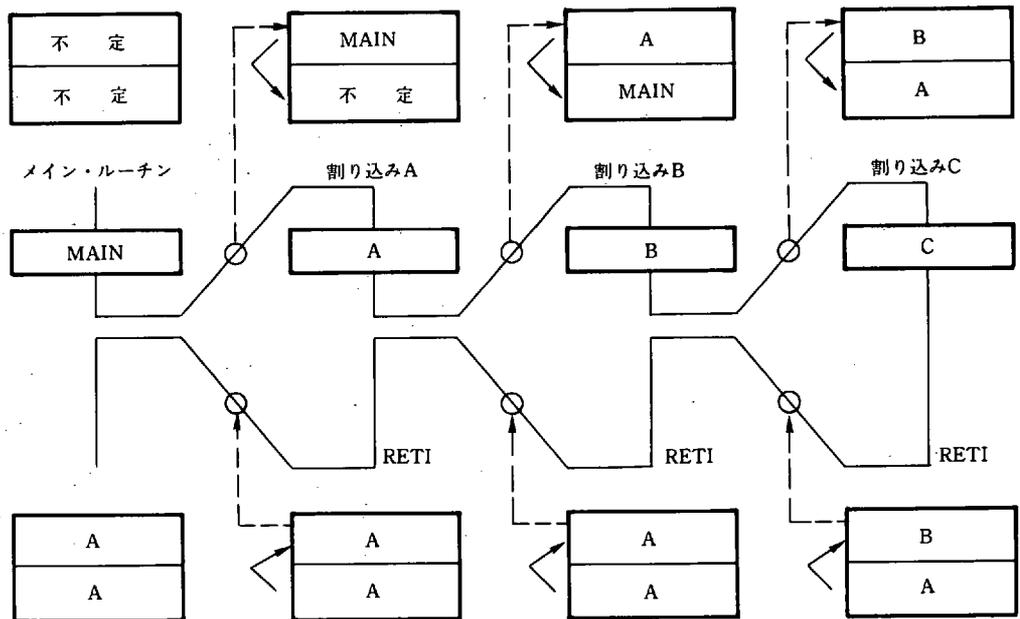
したがって2レベルを越える多重割り込みを使用するときの最終復帰命令には“RET”命令を使用してください。なお、“RETI”命令と“RET”命令は割り込み用スタックの復帰処理以外は同等の処理を行う命令です。

図 11-7 多重割り込み時の割り込みスタック動作

(a) 2レベルの多重割り込み

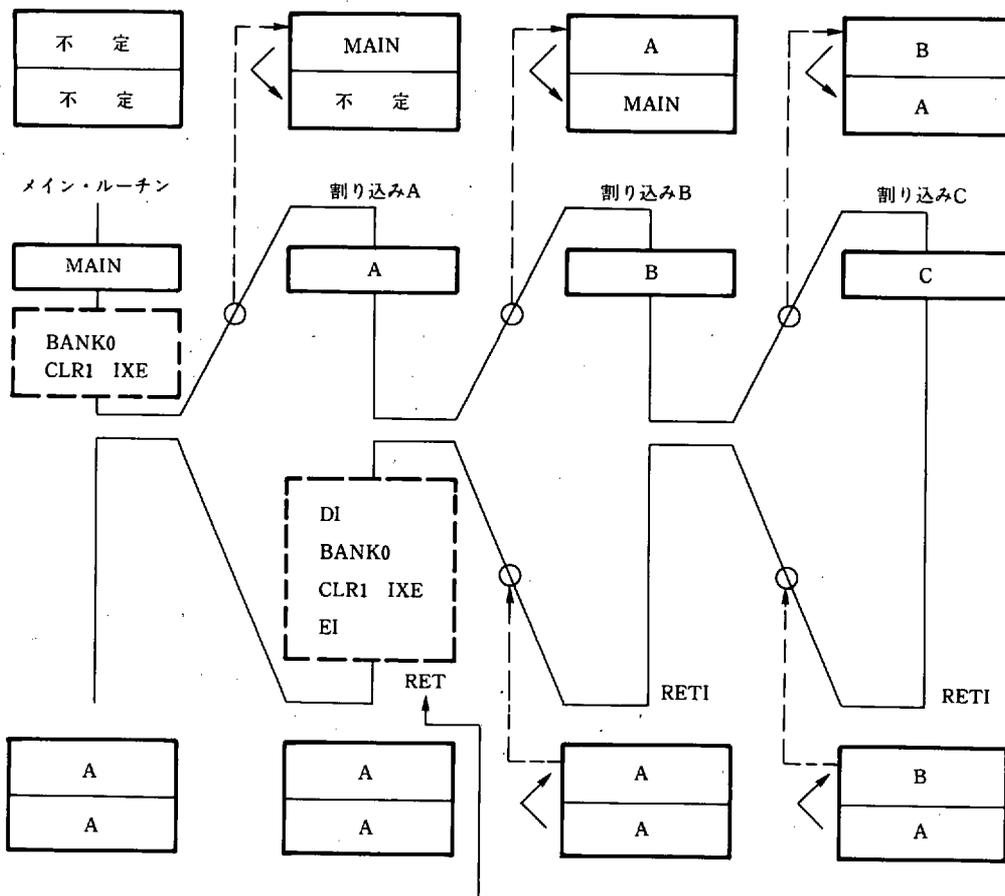


(b) 3レベルの多重割り込み



↑
この時点でメイン・ルーチンに復帰すると割り込みAの BANK と IXE が復帰されてしまい、メイン・ルーチンの動作が正常に行われなくなる。

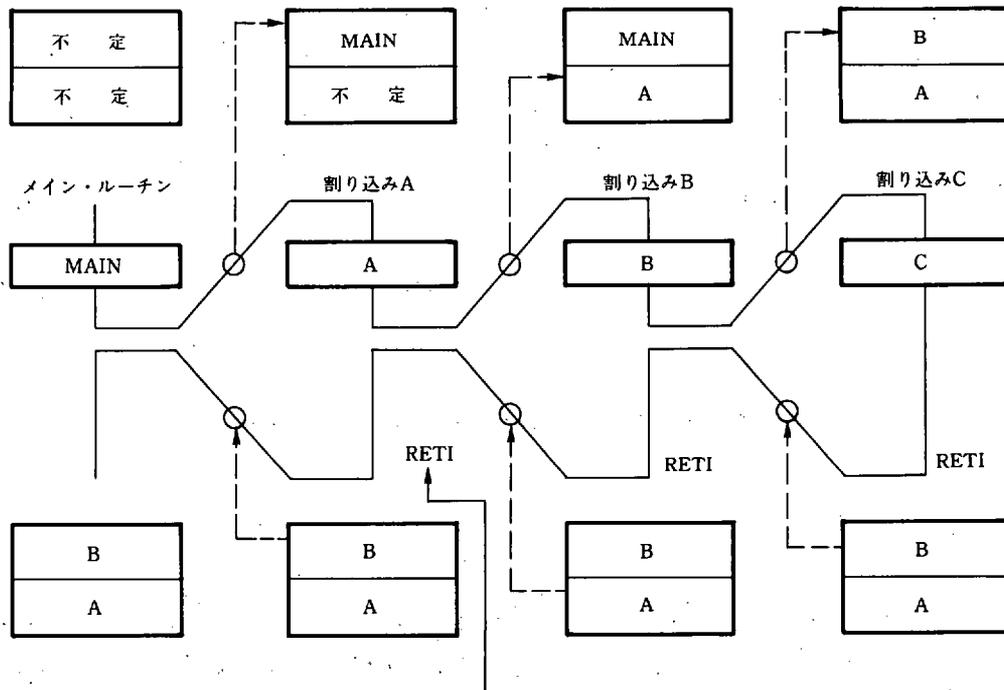
図 11-8 3レベルの多重割り込み使用例



割り込み A の優先順位は割り込み B および C より必ず低くし、かつ割り込み A を許可するメイン・ルーチンではバンク・レジスタとインデックス・イネーブル・フラグを常に一定（この例では BANK0, IXE=0）にしておけば、割り込み A の処理終了時点でメイン・ルーチンのバンク・レジスタおよびインデックス・イネーブル・フラグを指定したあとに“RET”命令を使用することにより 3レベルの多重の割り込みが可能になります。

また、割り込み A でのバンク・レジスタとインデックス・イネーブル・フラグがメイン・ルーチンとまったく等しいときは“RETI”命令を使用することができますが、図 11-9 に示すように 17K シリーズのエミュレータでの動作が異なるため“RETI”命令によるデバッグはできません。

図 11-9 17K シリーズのエミュレータ使用時の割り込みスタック動作



エミュレータで“RETI”命令を使用すると割り込み B のバンク・レジスタとインデクス・イネーブル・フラグの内容を復帰してしまいます。

11.9.3 アドレス・スタック・レジスタによる割り込みレベルの制限

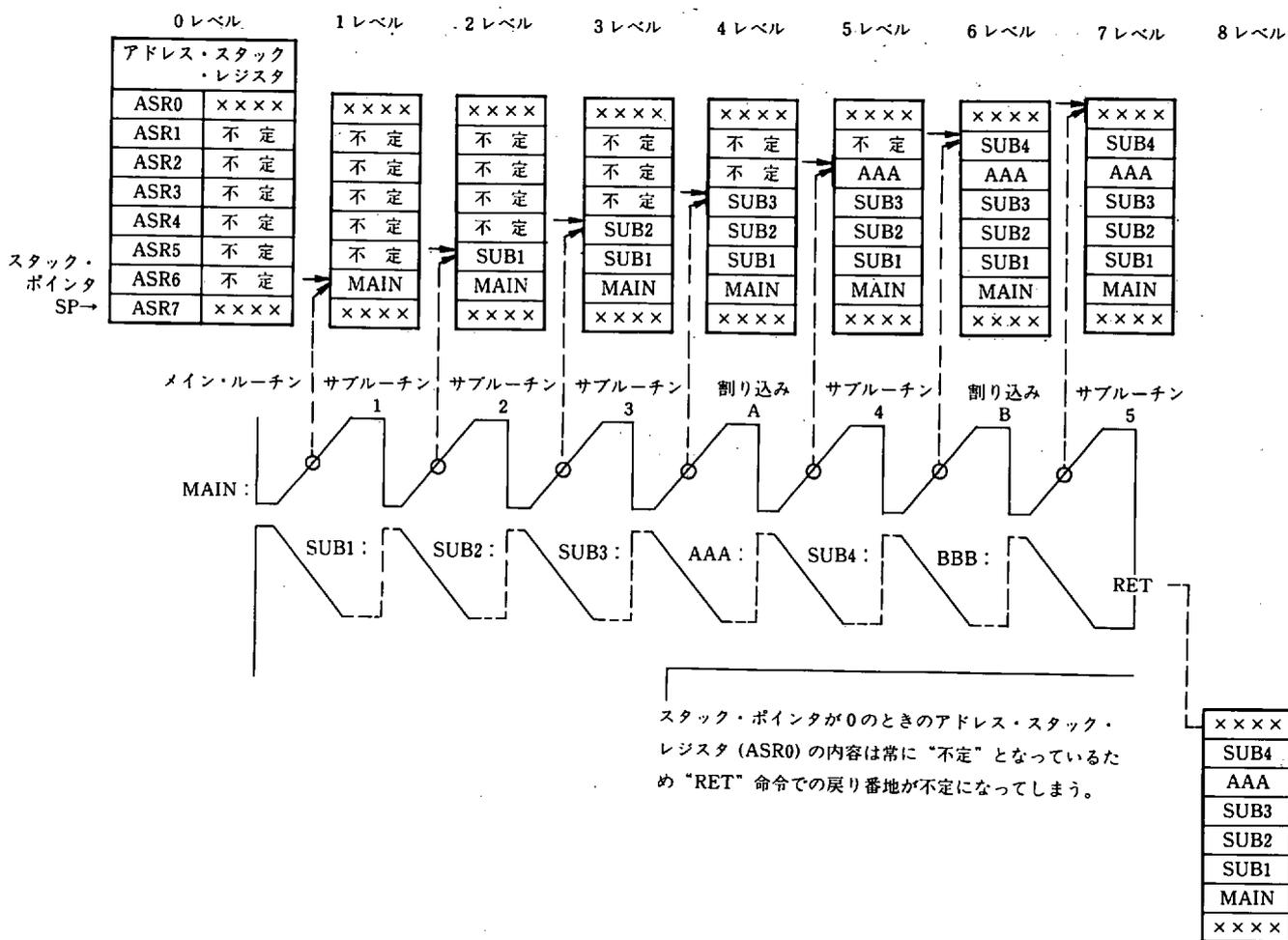
アドレス・スタック・レジスタには、割り込み処理からの復帰時の戻り番地が自動的に退避されます。

アドレス・スタック・レジスタは“4. スタック”の項で説明したように ASR0-ASR5 の 6 レベルが使用できます。割り込み要因は INT_{NC} 端子、タイマ、 $\overline{V_{SYNC}}$ 端子、およびシリアル・インタフェースの 4 要因であるため、割り込みだけでアドレス・スタック・レジスタを使用するうえでは、多重割り込みレベルに制限はないといえます。

ところが、アドレス・スタック・レジスタは、サブルーチン・コール時の戻り番号を退避するときにも使用されるため、サブルーチン・コールで使用されたアドレス・スタック・レジスタのレベル分だけ多重割り込みのレベルが制限されます。

たとえば図 11-10 に示すようにサブルーチン・コールで 4 レベル使用しているときには多重割り込みは 2 レベルしか使用できません。

図 11-10 アドレス・スタック・レジスタの動作



11.9.4 システム・レジスタおよびコントロール・レジスタ等の退避

多重割り込みを使用するときは、割り込み処理中に内容が変化するシステム・レジスタや、コントロール・レジスタの内容をあらかじめ退避しておく必要があります。

このとき、これらの内容の退避領域は各割り込み要因に対して別々に確保する必要があります。

また、現在受け付けられている割り込みおよびその割り込みより優先順位の低い割り込みを禁止し、優先順位の高い割り込みを許可する必要があります。

このとき、優先順位の高い割り込みは緊急性が高いため、割り込みの許可を最優先で行わなければなりません。

したがって、システム・レジスタやコントロール・レジスタの内容の退避は、“優先順位の高い割り込みを許可する処理”の次に行う方法が望ましいといえます。

以下に割り込み処理ルーチンにおける“優先順位の高い割り込みの許可”および“システム・レジスタやコントロール・レジスタ等の内容の退避”処理の例を示します。

例 多重割り込み時の割り込み許可および退避処理例

INT_{NC} 端子、 $\overline{V_{SYNC}}$ 端子およびタイマ割り込みを以下の優先順位（ソフトウェア優先）で使用する

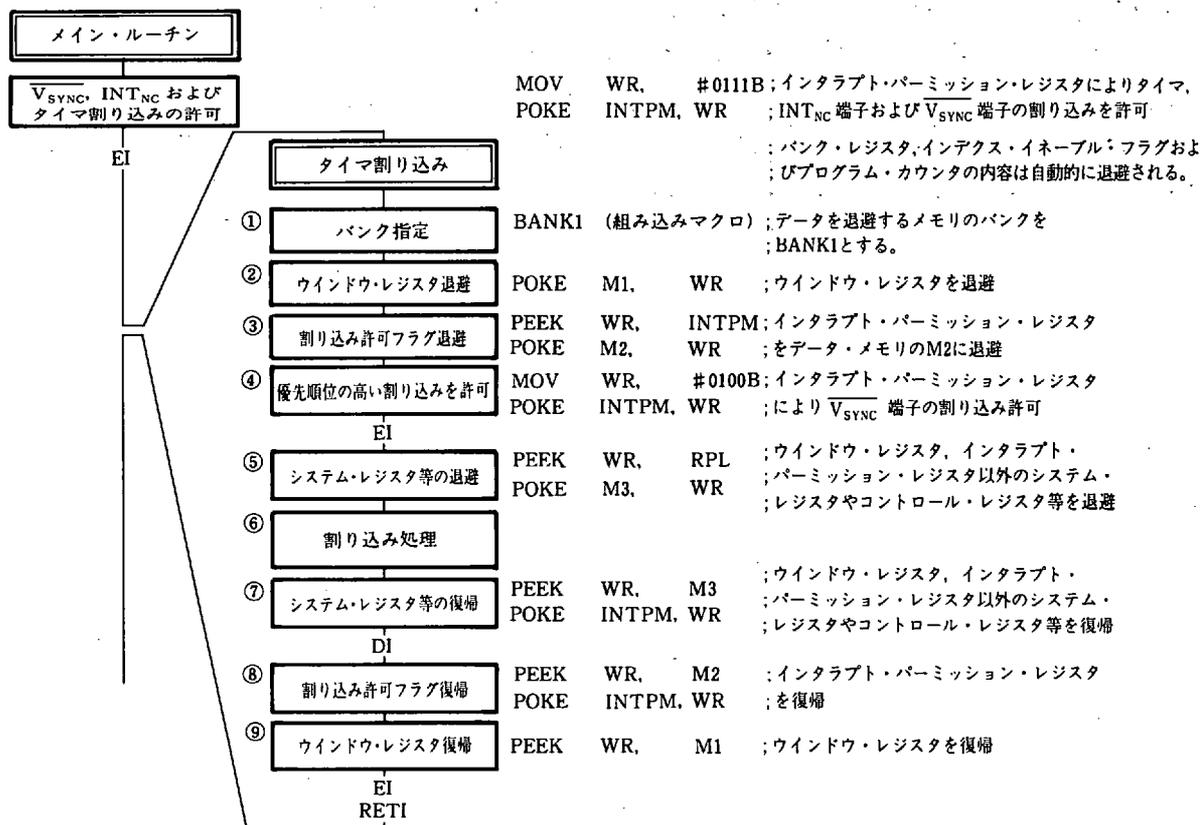
$\overline{V_{SYNC}}$ 端子 > タイマ > INT_{NC} 端子

また、1レベル目でタイマ割り込みが受け付けられたとします。

このときのプログラム例およびフロー・チャートを以下に示します。

フロー・チャート

プログラム例



①ではシステム・レジスタ等の内容を退避しておくデータ・メモリのバンクを指定します。

このとき、割り込みが受け付けられるとバンクはBANK0となるため、データを退避するメモリがBANK0であればこの命令は必要ありません。

②でウインドウ・レジスタの内容をデータ・メモリのM1に退避します。

このとき、“POKE”命令を使用しているため、データ・メモリM1のアドレスは40H以上である必要があります。ウインドウ・レジスタはこれ以後のデータ退避のワーク・エリアとして使用するため、最初に退避しておく必要があります。

③では、割り込みが受け付けられたときの各割り込み許可フラグ (IPNC, IPBMT0, IPVSYN) を退避します。この退避は、たとえばこの例の場合はメイン・ルーチンに復帰するときはINT_{NC} 端子、 $\overline{V_{SYN}}$ 端子およびタイマのすべての割り込みを許可する必要があるが、タイマ割り込みの優先順位はINT_{NC} 端子より高いためINT_{NC} 端子の割り込み処理中に受け付けられたときは、INT_{NC} 端子の割り込みを禁止して復帰させなければならないためです。

④ではタイマ割り込みより優先順位の高い $\overline{V_{SYN}}$ 端子の割り込みを許可し、このあと“EI”命令によりすべての割り込みを許可します。

①、②、③および④はすべての割り込みを禁止した状態で処理する必要があるため、この間は、優先順位が一番高い $\overline{V_{SYN}}$ 端子による割り込みも禁止されることになります。

⑤および⑥でシステム・レジスタやコントロール・レジスタ等の退避および復帰を行います。このときは優先順位が高い割り込みは許可されていてもかまいません。

なぜなら、優先順位の高い $\overline{V_{SYN}}$ 端子の割り込みが受け付けられたときに、同様の退避処理を行えば、 $\overline{V_{SYN}}$ 端子の割り込み処理から復帰したときはシステム・レジスタやコントロール・レジスタの内容は変化しないためです。

⑦および⑧で割り込み許可フラグの復帰とウインドウ・レジスタの復帰を行います。

このときはすべての割り込みは禁止されていなければなりません。

なぜなら割り込みを許可する⑦の命令が“EI”状態で実行されると、たまたまタイマ割り込み要求が発行されていた場合は⑧のウインドウ・レジスタの復帰を行わずに再度②でウインドウ・レジスタの退避を行ってしまい、ウインドウ・レジスタの内容が復帰できなくなってしまうためです。

12. タイマ機能

タイマ機能は、プログラムを作成するうえでの時間管理に使用します。

12.1 タイマの構成

図 12-1 にタイマの構成を示します。

図 12-1 に示すように、タイマはタイマ・キャリア・フリップフロップ (タイマ・キャリア FF) ブロックと、タイマ割り込みブロックの 2 系統で構成されています。

タイマ・キャリア FF およびタイマ割り込みの時間を設定するクロック生成回路は、8 MHz 分周器、セクタ A、セクタ B、バイアス回路およびコントロール・レジスタであるタイマ・モード・セレクト・レジスタ (BTM0CK: アドレス 09H) により構成されています。

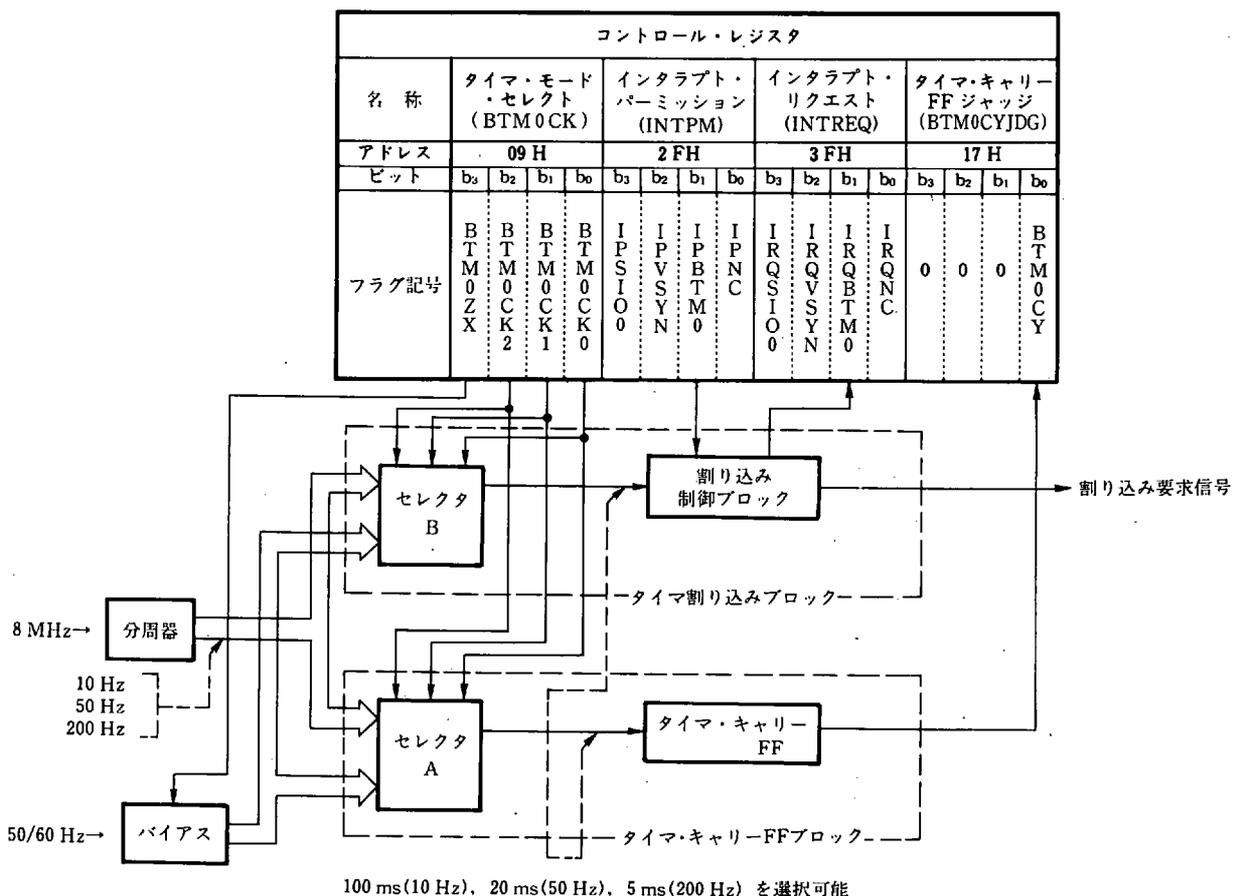
12.1.1 タイマ・キャリア FF ブロックの構成

図 12-1 に示すように、タイマ・キャリア FF ブロックはセクタ A、タイマ・キャリア FF およびコントロール・レジスタのタイマ・キャリア FF ジャッジ・レジスタ (BTM0CYJDG: アドレス 17H) により構成されています。

12.1.2 タイマ割り込みブロックの構成

図 12-1 に示すように、タイマ割り込みブロックはセクタ B、割り込み制御ブロックおよびコントロール・レジスタのインタラプト・パーミッション・レジスタ (INTPM: アドレス 2FH) とインタラプト・リクエスト・レジスタ (INTREQ: アドレス 3FH) により構成されています。

図 12-1 タイマの構成



12.2 タイマの機能

タイマは、タイマ・キャリー FF の検出およびタイマ割り込みの 2 つの機能があります。

タイマ・キャリー FF の検出は一定時間ごとにセットされるタイマ・キャリー FF の状態をプログラムで検出することにより時間管理を行い、タイマ割り込みは一定時間ごとに割り込みをかけることにより時間管理を行います。

タイマ・キャリー FF をセット (1) するタイミングおよびタイマ割り込みを発行するタイミングは、それぞれセレクト A およびセレクト B から出力されるタイマ時間設定パルスにより制御されます。

このタイマ時間設定パルスはタイマ・モード・セレクト・レジスタにデータを設定することにより 10 Hz (100 ms)、50 Hz (20 ms) および 200 Hz (5 ms) が選択できます。

セレクト A、セレクト B に入力するタイム・ベースとして、デバイスの動作周波数の 8 MHz を分周したパルスを使用する内部タイマ・モードと P0B₂/TMIN 端子に入力された 50 Hz または 60 Hz を使用する外部タイマ・モードの選択もタイマ・モード・セレクト・レジスタで設定します。

P0B₂/TMIN 端子に入力されたパルスを 5 分周するか 6 分周するかを選択もタイマ・モード・セレクト・レジスタで設定します。

タイマ時間設定パルスはタイマ・キャリー FF とタイマ割り込みとを組み合わせて設定します。

図 12-2 にタイマ・モード・セレクト・レジスタとタイマ時間設定パルスの関係を示します。

内部タイマ・モードではタイマ時間設定パルスは、デバイス動作周波数である 8 MHz を分周して作成しているため、8 MHz がずれているとタイマ時間設定用パルスも同じ割合でずれることになります。

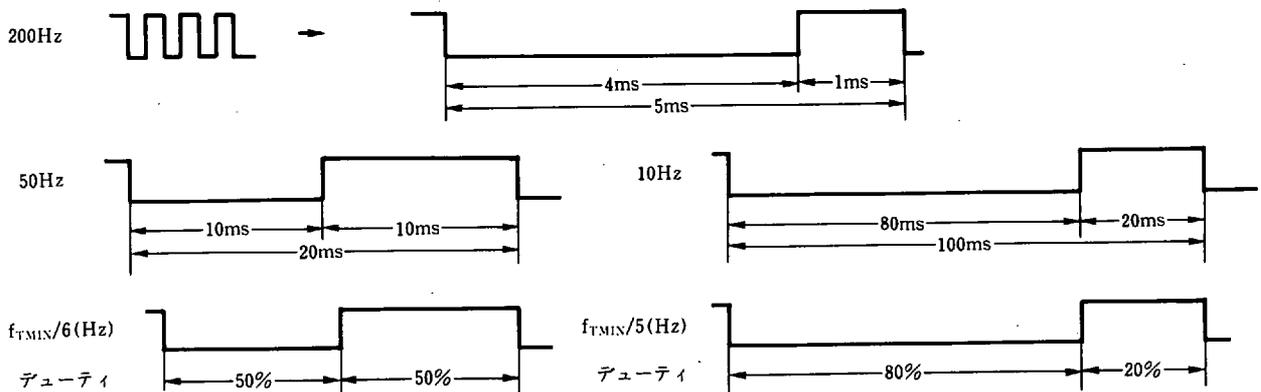
図 12-2 タイマ・モード・セレクト・レジスタとタイマ時間設定パルスの関係

コントロール・レジスタ				
名称	タイマ・モード・セレクト (BTM0CK)			
アドレス	09H			
Read/Write	R/W			
ビット	b ₃	b ₂	b ₁	b ₀
フラグ記号	B T M 0 Z X	B T M 0 C K 2	B T M 0 C K 1	B T M 0 C K 0

タイマ・モード・セレクト (BTM0CK)			タイマ・キャリア・FFセット用パルスの周波数(時間)選択		タイマ割り込み用パルスの周波数(時間)選択	
0	0	0	10 Hz (100 ms)	内部タイマ	200 Hz (5 ms)	内部タイマ
0	0	1	200 Hz (5 ms)	内部タイマ	10 Hz (100 ms)	内部タイマ
0	1	0	10 Hz (100 ms)	内部タイマ	50 Hz (20 ms)	内部タイマ
0	1	1	200 Hz (5 ms)	内部タイマ	50 Hz (20 ms)	内部タイマ
1	0	0	$f_{TMIN}/5$ Hz ($5/f_{TMIN}$ s)	外部タイマ	200 Hz (5 ms)	内部タイマ
1	0	1	200 Hz (5 ms)	内部タイマ	$f_{TMIN}/5$ Hz ($5/f_{TMIN}$ s)	外部タイマ
1	1	0	$f_{TMIN}/6$ Hz ($6/f_{TMIN}$ s)	外部タイマ	200 Hz (5 ms)	内部タイマ
1	1	1	200 Hz (5 ms)	内部タイマ	$f_{TMIN}/6$ Hz ($6/f_{TMIN}$ s)	外部タイマ

f_{TMIN} は P0B₂/TMIN 端子の入力周波数 (50 Hz または 60 Hz)

0	バイアス回路停止
1	バイアス回路動作



12.3 タイマ・キャリー・フリップフロップ (タイマ・キャリー FF)

タイマ・キャリー FF は、タイマ・モード・セレクト・レジスタにより設定されたタイマ・キャリー FF セット用パルスの立ち上がりエッジによりセット (1) されます。

タイマ・キャリー FF の内容はタイマ・キャリー FF ジャッジ・レジスタの最下位ビット (BTM0CY フラグ) と 1 対 1 に対応しており、タイマ・キャリー FF がセット (1) されると、BTM0CY フラグも同時にセット (1) されます。

BTM0CY フラグは“PEEK”命令によりウインドウ・レジスタに内容を読み込むことによりリセット (0) されず (Read & Reset)。

BTM0CY フラグがリセット (0) されると、同時にタイマ・キャリー FF もリセット (0) されます。

すなわち、プログラムで BTM0CY フラグを読み込むことによりタイマ・モード・セレクト・レジスタによって設定した時間のタイマを作成できます。

12.3.1 にプログラム例を示します。

タイマ・キャリー FF を使用するときには以下の点に注意する必要があります。

タイマ・キャリー FF は、パワーオン・リセット時はセット禁止状態になっており、その後一度“PEEK”命令により BTM0CY フラグの内容を読み込むまでセットされません。

すなわちパワーオン・リセット後最初に BTM0CY フラグを読み込むと必ず“0”が読み込まれ、その後はタイマ・モード・セレクト・レジスタによって設定された時間ごとにセット (1) されることとなります。

また、タイマ・キャリー FF は CE リセットのタイミングも制御しています。

すなわち、CE 端子がロウ・レベルからハイ・レベルに変化すると、次にタイマ・キャリー FF がセットされるタイミングに同期して CE リセットがかかります。

したがって、システム・リセット時 (パワーオン・リセット時および CE リセット時) に BTM0CY フラグの内容を読み込むことにより、停電検出を行うことが可能となります。詳しくは“12.4 タイマ・キャリー FF 使用時の注意”および“14. リセット機能”の項を参照してください。

BTM0CY フラグは読み出し専用フラグであるため“POKE”命令による書き込み動作を行ってもデバイス動作上は何の影響も受けません。ただし、17K シリーズのアセンブラを使用すると“エラー”が発生します。

12.3.1 BTM0CY フラグによるタイマ使用例

以下にプログラム例を示します。

```

例 INITFLG  NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
          ; 組み込みマクロ
          ; タイマ・キャリー FF のセット時間を100 ms に設定

LOOP1:
  MOV  M1, #0110B

LOOP2:
  SKT1  BTM0CY ; 組み込みマクロ
          ; BTM0CY フラグをテストし, "0" なら NEXT へ分岐

  BR   NEXT

  ADD  M1, #0100B ; データ・メモリ M1 に 4 を加算
  SKT1  CY       ; 組み込みマクロ
          ; CY フラグをテストし,
  BR   NEXT     ; "0" なら NEXT へ分岐
  [ 処理 A ]    ; "1" なら処理 A を実行
  MOV  M1, #0110B

NEXT:
  [ 処理 B ]    ; 処理 B を実行して LOOP へ分岐
  BR   LOOP
    
```

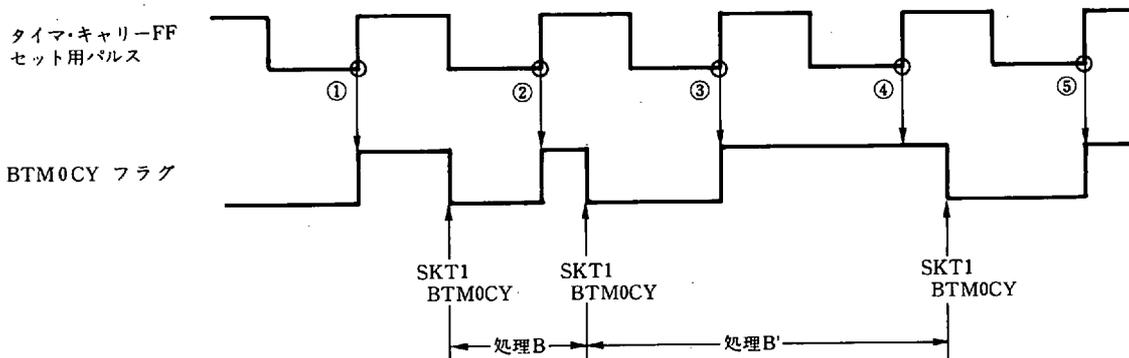
上記プログラムは処理 A を 1 秒ごとに実行します。

ここでプログラム作成上以下の(1)に示す点に注意が必要です。

- (1) BTM0CY フラグを検出する時間は、タイマ・キャリー FF がセット (1) される時間より短くなくてはならない。

これは、上記の例を基にして説明すると、図 12-3 に示すように、処理 B の時間が 100 ms 以上あると、タイマ・キャリー FF のセットを取り逃がすためです。

図 12-3 BTM0CY フラグの検出とタイマ・キャリー FF



②でセットされた BTM0CY フラグを検出後、処理 B' の時間が長いので、③でセットされた BTM0CY フラグの状態を取り逃がしてしまう。

12.3.2 BTM0CY フラグによるタイマの誤差

BTM0CY フラグによるタイマの誤差には BTM0CY フラグの検出時間による誤差と、タイマ・キャリー FF のセット時間を変更したときの誤差があります。

以下の(1)および(2)にそれぞれについて説明します。

(1) BTM0CY フラグの検出時間による誤差

12.3.1 項で説明したように、BTM0CY フラグを検出する時間はタイマ・キャリー FF がセット(1)される時間より短い間隔で検出する必要があります。

すなわち BTM0CY フラグを検出する時間間隔を t_{CHECK} とし、タイマ・キャリー FF がセットされる時間間隔 (100 m または 5 ms) を t_{SET} とすると、以下に示す関係である必要があります。

$$t_{CHECK} < t_{SET}$$

このとき図 12-4 に示すように BTM0CY フラグを検出するときのタイマの誤差は以下のようになります。

$$0 < \text{誤差} < t_{CHECK}$$

図 12-4 BTM0CY フラグの検出時間による誤差

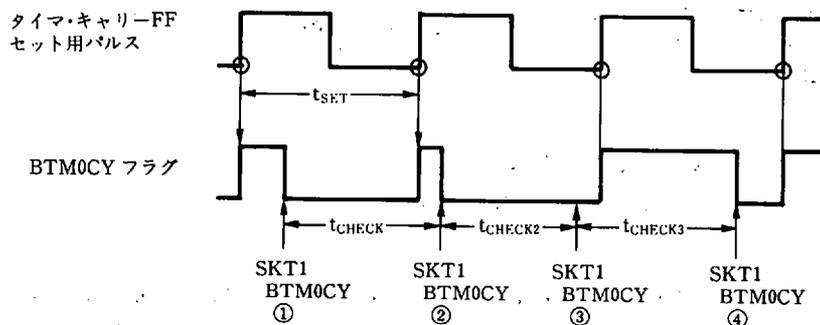


図 12-4 に示すように、まず②で BTM0CY フラグを検出すると“1”であるためタイマを更新します。次に③で検出すると“0”であるため、④で再度検出するまでタイマは更新されません。すなわち、このときのタイマは t_{CHECK3} の時間だけ長くなることになります。

(2) タイマ・キャリー FF のセット時間を変更したときの誤差

タイマ・キャリー FF のセット時間はタイマ・モード・セレクト・レジスタの BTM0CK2, BTM0CK1 および BTM0CK0 フラグにより設定します。

図 12-1 および図 12-2 に示したように、タイマ時間設定パルスは 200 Hz, 10 Hz および外部タイマの 3 種類を選択できます。

このとき、これらの 3 種類のパルスは独立に動作しています。

したがって BTM0CK2, BTM0CK1 および BTM0CK0 フラグによってタイマ時間設定パルスを切り替えると、以下に示す例のように誤差が生じます。

例

; ①

INITFLG NOT BTM0ZX, NOT BTM0CK2, BTM0CK1, NOT BTM0CK0

; 組み込みマクロ

処理 A

; タイマ・キャリー FF セット用パルスを 10 Hz (100 ms) に設定

; ②

SET1 BTM0CK0 ; 組み込みマクロ

; タイマ・キャリー FF セット用パルスを 200 Hz (5 ms) に設定

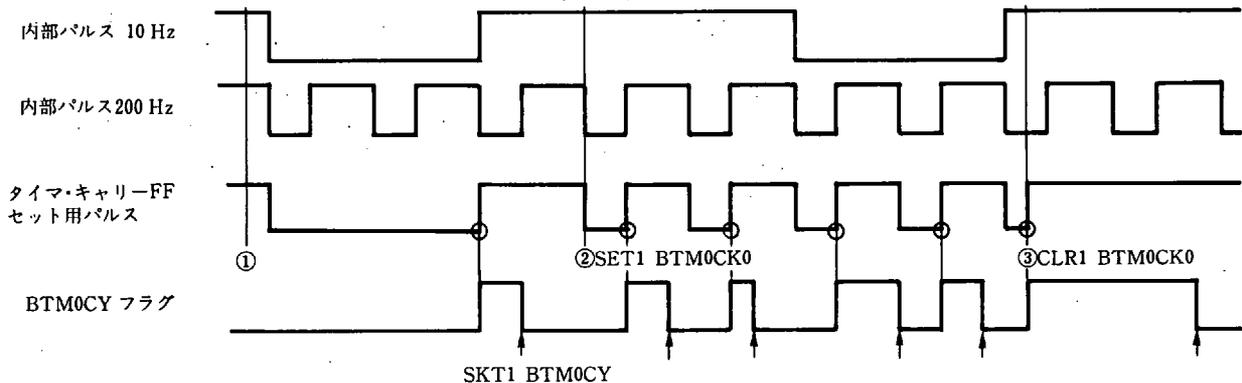
処理 A

; ③

CLR1 BTM0CK0 ; 組み込みマクロ

; タイマ・キャリー FF セット用パルスを 10 Hz (100 ms) に設定

このとき、タイマ・キャリー FF セット用パルスは以下のように切り替わります。



上図に示すように、タイマ・キャリー FF のセット時間を切り替えることにより、切り替わったパルスが立ち下がった場合は BTM0CY フラグは以前の状態を保持 (図の ②) しますが、切り替わったパルスが立ち上がったときは BTM0CY フラグはセット (1) されます (図の ③)。

図 12-5 に示すようにタイマ・キャリー FF セット時間を切り替えたときの最初の BTM0CY フラグのセットされるまでの誤差は以下ようになります。

- $t_{SET} < \text{誤差} < t_{CHECK}$

t_{SET} : 切り替わったタイマ・キャリー FF セット時間

t_{CHECK} : BTM0CY フラグを検出する時間

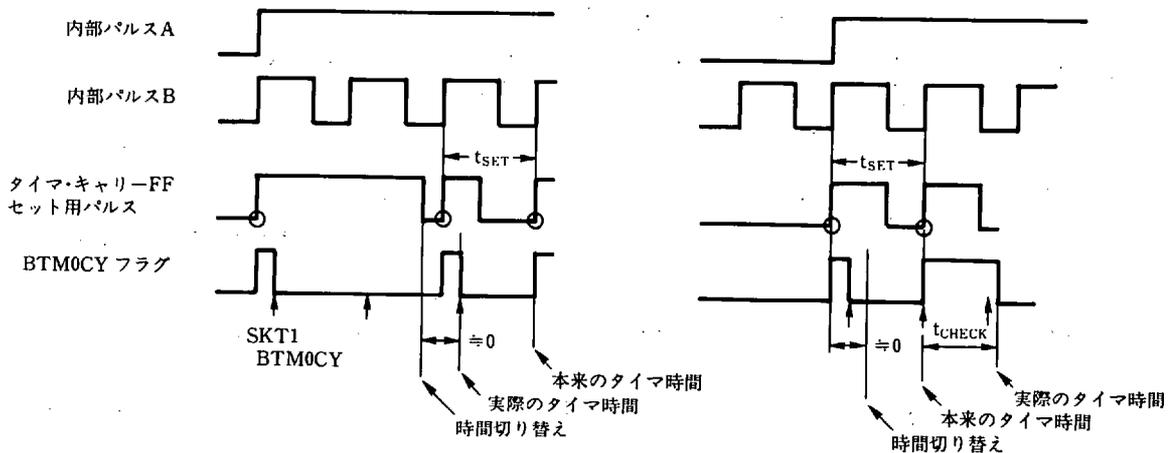
なお、4 Hz, 10 Hz, 200 Hz および 1 KHz の内部パルスにはそれぞれ位相差を設けてありますが、この位相差は切り替えたパルスの時間より短くなるため上記誤差の中に含まれることになります。

各パルスの位相差については、“12.6 タイマ割り込み使用時の注意”の項を参照してください。

図 12-5 タイマ・キャリー FF セット時間を A → B に切り替えたときの誤差

(a) - t_{SET} の誤差

(b) t_{CHECK} の誤差



タイマ時間を切り替えたすぐあとに BTM0CY フラグを検出すると“1”となるため t_{SET} の誤差になる。

BTM0CY フラグを検出したすぐあとにタイマ時間の切り替えを行うと、BTM0CY フラグは1回分リセットされた状態になるため t_{CHECK} の誤差が出る。

12.4 タイマ・キャリー FF 使用時の注意

タイマ・キャリー FF は、タイマ機能だけでなく CE リセット時のリセット同期信号に使用しています。

すなわち、CE 端子がロウ・レベルからハイ・レベルに変化したあとに次のタイマ・キャリー FF セット用パルスが立ち上がると、CE リセットがかかります。

このとき、以下に示す点に注意が必要です。

- (1) タイマの更新処理時間および BTM0CY フラグの検出時間間隔の和は、タイマ・キャリー FF セット時間より短い必要があります。
- (2) パワーオン・リセット後は CE リセットに関係なく常時一定のタイマが動作するプログラムを作成したときは、CE リセットごとに補正を行う必要があります。
- (3) BTM0CY フラグの検出と CE リセット時のリセット同期信号は、BTM0CY フラグの検出が優先されます。したがって、両方が重なった場合は CE リセットは 1 回分遅れてかかります。

上記(1)、(2)および(3)についてそれぞれ 12.4.1 - 12.4.3 項に説明します。

12.4.1 タイマの更新処理時間および BTM0CY フラグの検出時間間隔

12.3.1 項でも説明したように BTM0CY フラグを検出する時間間隔 t_{SET} はタイマ・キャリー FF がセットされる時間より短くする必要があります。

このとき、たとえ BTM0CY フラグを検出する時間間隔が短くてもタイマの更新処理時間が長いときは、CE リセットがかかるとタイマの処理が正常に実行されない場合があります。

したがって、以下に示す条件を満足させる必要があります。

$$t_{CHECK} + t_{TIMER} < t_{SET}$$

t_{CHECK} : BTM0CY フラグを検出する時間間隔

t_{TIMER} : タイマ更新処理時間

t_{SET} : タイマ・キャリー FF をセットする時間

以下に例を示します。

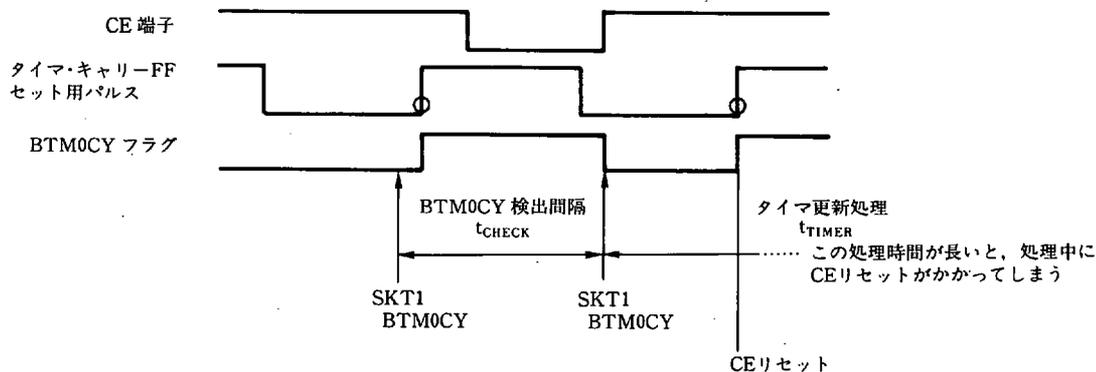
例 タイマの更新処理および BTM0CY フラグ検出時間間隔例

```

START :                               ; プログラム・アドレス 0000H
      INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
      ; 組み込みマクロ
      ; タイマ・キャリー FF のセット時間を 100 ms に設定

TIMER :
      ; ①
      SKT1   BTM0CY   ; 組み込みマクロ
      ; BTM0CY フラグをテストし
      BR     AAA     ; "0" なら AAA へ分岐
      [ タイマ更新 ]
      BR     TIMER
AAA :
      [ 処理 A ]
      BR     TIMER
    
```

上記プログラムのタイミング・チャートを以下に示します。



12.4.2 CEリセット時におけるタイマ・キャリーFFの補正

例にCEリセット時のタイマ補正例を示します。

例に示すように、CEリセット時にタイマの補正が必要な場合として、“タイマ・キャリーFFを停電検出に使用しており、かつタイマ・キャリーFFを時計用のタイマ等に使用している場合”が考えられます。

タイマ・キャリーFFはパワーオン・リセット時はリセット(0)されており、以後一度“PEEK”命令によりBTM0CYフラグを読み出すまでセット禁止状態になっています。

またCE端子がロウ・レベルからハイ・レベルに変化するとタイマ・キャリーFFセット用パルスの立ち上がりエッジに同期してCEリセットがかかります。このとき、BTM0CYフラグはセット(1)されてスタートします。

したがってシステム・リセット(パワーオン・リセットおよびCEリセット)時にBTM0CYフラグの状態を検出することにより“0”であればパワーオン・リセットであり、“1”であればCEリセットであると判断できます(停電検出)。

このとき、時計用のタイマなどはCEリセット時でも動作を継続していなければなりません。

ところが、停電検出のためにBTM0CYフラグを読み込むことによりBTM0CYフラグがリセット(0)されてしまうため、BTM0CYフラグのセット(1)状態を1回見逃がしてしまいます。

このため、停電検出によりCEリセットと判断されたときは時計用のタイマ更新を行う必要があります。

以下に例を示します。

また、停電検出については“14.6 停電検出”の項も参照してください。

例 CEリセット時のタイマ補正例

タイマ・キャリーFFにより停電検出と時計の更新を行うとき

```

START :                               ;プログラム・アドレス 0000H
      [ 処 理 A ]
      ;①
      SKT1    BTM0CY    ;組み込みマクロ
                          ;BTM0CYフラグをテストし
      BR      INITIAL  ;“0”ならINITIALへ分岐(停電検出)
BACKUP :
      ;②
      [ 100ms分時計更新 ] ;バックアップ(CEリセット)であるため時計の補正
LOOP :
      ;③
      [ 処 理 B ] ;処理Bを実行しながら
      SKF1    BTM0CY    ;BTM0CYフラグをテストして時計を更新する。
      BR      BACKUP
      BR      LOOP
INITIAL :
      INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
                          ;組み込みマクロ
                          ;停電(パワーオン・リセット)であるため、タイマ・キャリーFFの
                          ;セット時間を100msに設定し、処理Cを実行する。
      [ 処 理 C ]
      BR      LOOP
    
```

上記プログラムのタイミング・チャートを図12-6に示します。

図 12-6 タイミング・チャート

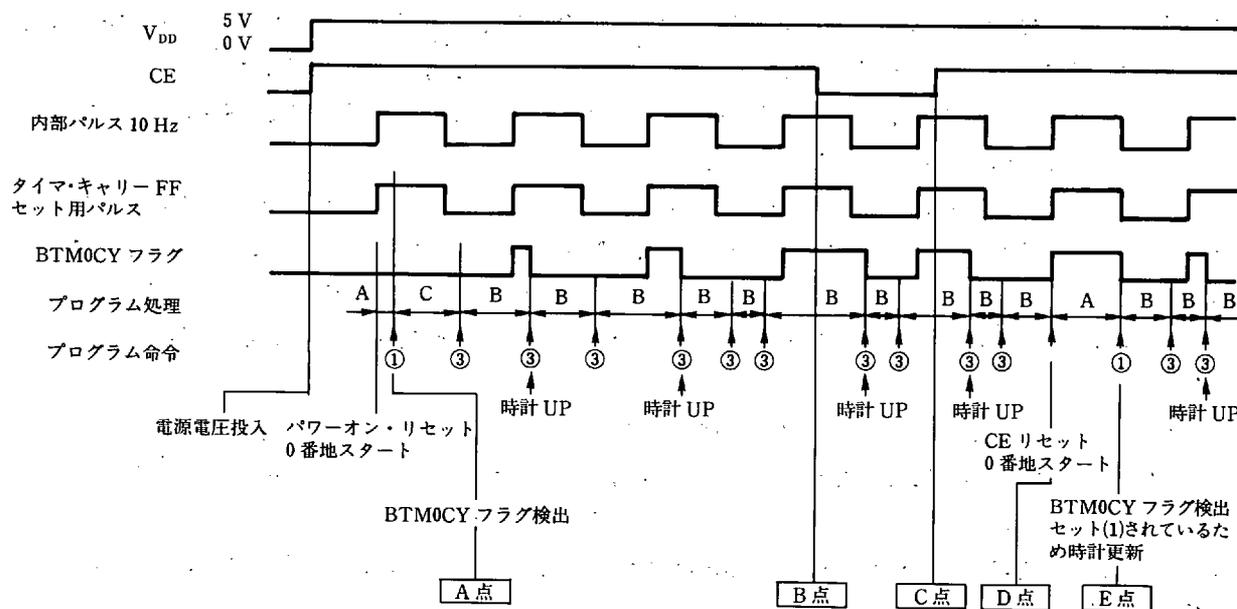


図 12-6 に示すように、パワーオン・リセット時は内部 10 Hz パルスの立ち上がりにより 0000H 番地からプログラムがスタートします。

次に A 点で BTM0CY フラグを検出すると、電源投入時であるため BTM0CY フラグはリセット (0) されており停電 (パワーオン・リセット) であると判断されます。

したがって、“処理 C” を実行しタイマ・キャリー FF セット用パルスを 100 ms に設定します。

A 点で一度 BTM0CY フラグの内容を読み出しているため、以後 100 ms ごとに BTM0CY フラグがセット (1) されます。

次に B 点で CE 端子がロウ・レベルとなり C 点でハイ・レベルになっても、クロック・ストップ命令が実行されていないかぎりプログラムは“処理 B”を実行しながら時計のカウント・アップを行います。

C 点で CE 端子がロウ・レベルからハイ・レベルに立ち上がったため、次のタイマ・キャリー FF セット用パルスの立ち上がりである D 点で CE リセットがかかりプログラムは 0000H 番地からスタートします。

このとき E 点で BTM0CY フラグを検出すると BTM0CY フラグはセット (1) されているためバック・アップ (CE リセット) と判断されます。

また、図から明らかなように E 点で時計を 100 ms 分更新しなければ、CE リセットがかかるたびに時計が 100 ms ずつ遅れることとなります。

また、E 点で停電検出を行うときに処理 A が 100 ms 以上かかっていると、BTM0CY フラグのセットを 2 回見逃がすことになるため、処理 A は必ず 100 ms 未満に行う必要があります。

したがって、停電検出のための BTM0CY フラグの検出はプログラムが 0000H 番地からスタートしたあと、タイマ・キャリー FF のセット時間未満で行う必要があります。

12.4.3 BTM0CY フラグの検出とCE リセットが重なったとき

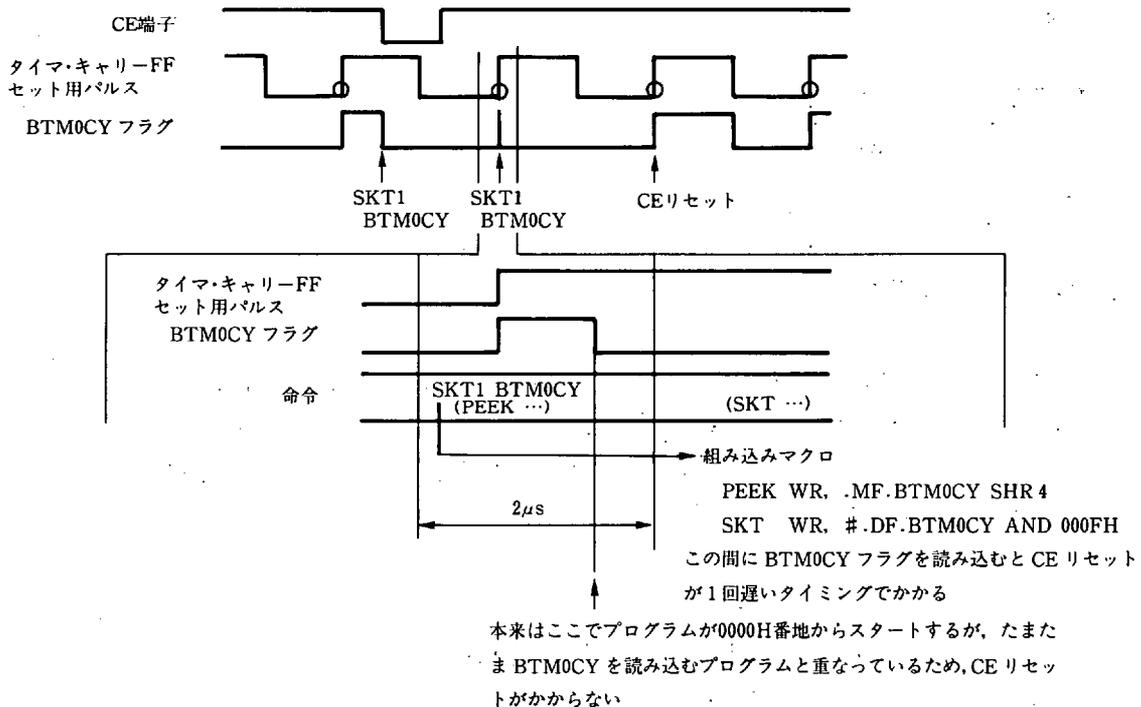
12.4.2 項でも説明したように CE リセットは BTM0CY フラグがセット (1) されると同時にかかります。

このとき、たまたま BTM0CY フラグの読み込み命令と CE リセットが重なると、BTM0CY フラグの読み込み命令が優先されます。

したがって、CE 端子がロウ・レベルからハイ・レベルに切り替わった次の BTM0CY フラグのセット (タイマ・キャリー FF セット用パルスの立ち上がり) と、BTM0CY フラグの読み込み命令が重なったときは、さらに “その次の BTM0CY フラグがセットするタイミング” で CE リセットがかかります。

この動作を図 12-7 に示します。

図 12-7 CE リセットと BTM0CY フラグの読み込み命令が重なったときの動作



したがって、BTM0CY フラグを周期的に検出するようなプログラムで、かつ BTM0CY フラグの検出時間間隔と BTM0CY フラグのセット時間が一致するようなプログラムになっていると永遠に CE リセットがかからないことがあります。

すなわち、以下に示す点に注意が必要です。

1 命令サイクルは 2 μs (1/500 kHz) であるため、たとえば 500 命令に 1 回の割合で BTM0CY フラグを検出するようなプログラムは、2 μs × 500 = 1 ms ごとに BTM0CY フラグを読み出すことになります。

このとき、タイマ時間設定パルスは 5 ms および 100 ms のどちらを選択しても一度 BTM0CY フラグのセットと検出が一致すると永遠に CE リセットがかからなくなります。

すなわち、以下に示す条件を満足するような周期的なプログラムは作成しないでください。

$$\frac{t_{SET} \times 500}{x} = n \quad (n: \text{自然数})$$

t_{SET} : BTM0CY フラグのセット時間

x : BTM0CY フラグを読み出す命令の周期 x ステップ

となる n が存在する x ステップのプログラムは作成しないでください。

また、以下にこの条件を満足してしまうプログラム例を示します。このようなプログラムは作成しないでください。

例

```

    処 理 A
INITFLG  NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, BTM0CK0
           ; 組み込みマクロ
           ; タイマ・キャリー FF セット用パルスを 5 ms に設定

LOOP :
; ①
SKT1     BTM0CY ; 組み込みマクロ
BR       BBB
AAA :
    496 ステップ
BR       LOOP
BBB :
    496 ステップ
BR       LOOP
    
```

この例では、①の BTM0CY フラグ読み込み命令は500命令ごとに繰り返されるため、①の命令のタイミングでたまたま BTM0CY フラグがセットされると、以後 CE リセットがかからなくなります。

また、IDC 動作中は命令実行時間が12 μs (1/83.33 kHz) であるため以下に示す条件を満足するような周期的なプログラムは作成しないでください。

$$\frac{t_{SET} \times 83.33}{x} = n \quad (n: \text{自然数})$$

t_{SET} : BTM0CY フラグのセット時間

x : BTM0CY フラグを読み出す命令の周期 x ステップ

12.5 タイマ割り込み

タイマ割り込みは、タイマ・モード・セレクト・レジスタにより設定されたタイマ割り込み用パルスの立ち下がりエッジにより割り込み要求が発行されます。

タイマ割り込み要求はインタラプト・リクエスト・レジスタの IRQBTM0 フラグと 1 対 1 に対応しており、タイマ割り込み要求が発行されると IRQBTM0 フラグがセット (1) されます。

すなわち、タイマ割り込み用パルスが立ち下がると、IRQBTM0 フラグがセット (1) されます。

タイマ割り込みは、“11. 割り込み”の項でも説明したように割り込み要求の発行以外に、すべての割り込みの許可命令である“EI”命令の実行およびタイマ割り込みの許可を設定する必要があります。

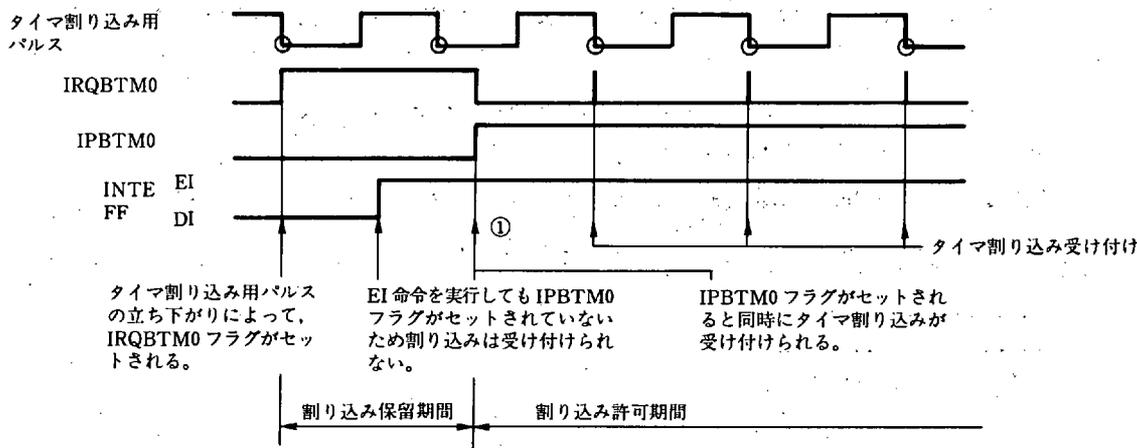
タイマ割り込みの許可はインタラプト・パーミッション・レジスタの IPBTM0 フラグをセット (1) することにより設定します。

したがってタイマ割り込みは“EI”命令が実行されており、かつ IPBTM0 フラグがセット (1) されているときに IRQBTM0 フラグがセット (1) されると受け付けられます。

タイマ割り込みが受け付けられるとプログラムの流れはプログラム・メモリ・アドレスの 0003H 番地に移ります。割り込みが受け付けられた時点で IRQBTM0 フラグはリセット (0) されます。

図 12-8 にタイマ割り込み用パルスと IRQBTM0 フラグの関係を示します。

図 12-8 タイマ割り込み用パルスと IRQBTM0 フラグの関係



ここで注意する点は図 12-8 の ① 点に示すように“DI”命令または IPBTM0 フラグによりタイマ割り込みが禁止されているときに IRQBTM0 フラグが一度セットされると、次に“EI”命令の実行および IPBTM0 フラグがセットされた時点でタイマ割り込みが受け付けられるという点です。

このような場合は、IRQBTM0 フラグに“0”を書き込むことにより割り込み要求を解除することができます。

また、逆に IRQBTM0 フラグに“1”を書き込むと、割り込み要求が発行されたことと等価になります。

タイマ割り込みが受け付けられたときはスタックが 1 レベル使用されます。

このとき、バンク・レジスタの内容と、インデックス・イネーブル・フラグの内容は自動的に退避されます。

割り込み処理ルーチンからの復帰には専用命令である“RETI”命令を使用します。

詳しくは“4. スタック”および“11. 割り込み”の項を参照してください。

12.5.1 および 12.5.2 に、タイマ割り込みを用いた使用例およびタイマ割り込みの誤差について説明します。

また、他の割り込み (INT_{NC} 端子、 $\overline{V}_{\text{SYNC}}$ 端子、シリアル・インタフェース) との関係は“11. 割り込み”の項を参照してください。

12.5.1. タイマ割り込みを用いたタイマの使用例

以下に例を示します。

例

```

BR      AAA      ;AAA に分岐
TIMER :          ;プログラム・アドレス 0003H
ADD     M1, #0001B ; M1 に 1 を加算
SKT1   CY       ; CY フラグをテスト
BR     BBB      ; キャリーが出なければリターン


処 理  A


BBB :
EI
RETI
AAA :
INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
          ; 組み込みマクロ
          ; タイマ割り込み用パルス を 5 ms に設定
MOV     M1, #0000B ; M1 の内容を 0 にクリア
SET1   IPBTM0     ; タイマ割り込みの許可を設定
EI     ; すべての割り込みの許可を設定
LOOP :


処 理  B


BR     LOOP

```

上記プログラムは処理Aを80msごとに実行します。

このとき注意する点は、割り込みが受け付けられると、自動的にDI状態になるという点と、DI状態であってもIRQBTM0フラグはセット(1)されるという点です。

すなわち、処理Aの時間が5ms以上あると“RETI”命令で復帰してもすぐに割り込みが受け付けられてしまい、処理Bは実行されなくなってしまいます。

12.5.2 タイマ割り込みの誤差

12.4で説明したように、タイマ割り込みが許可されていればタイマ割り込み用パルスが立ち下がるたびに割り込みが受け付けられます。

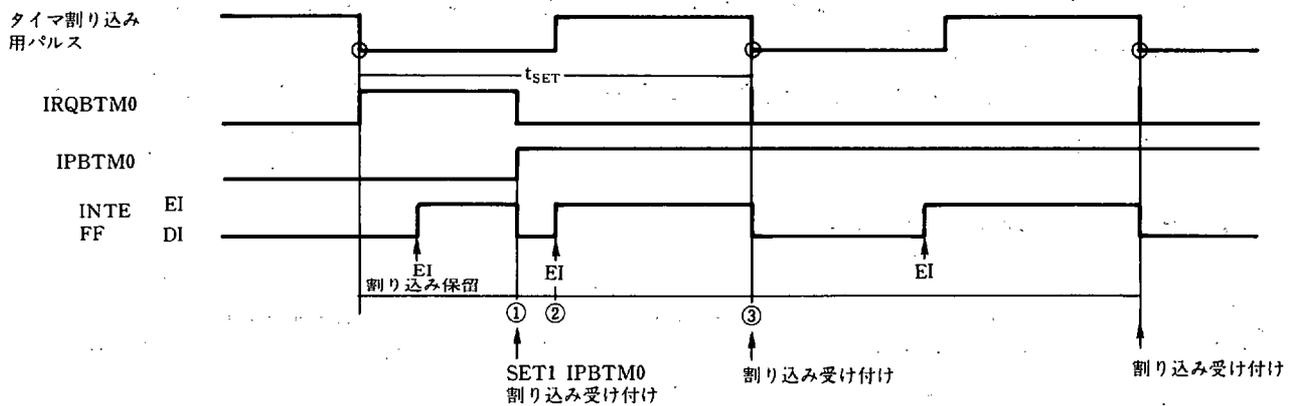
したがって、タイマ割り込み使用時のタイマの誤差は以下の操作を行ったときのみ発生します。

- (1) タイマ割り込みを許可したときの最初の割り込み受け付け時
- (2) タイマ割り込み用パルスの時間を変更したときの最初の割り込み受け付け時
- (3) IRQBTM0 フラグに書き込み操作を行ったとき

図 12-9 に各操作時の誤差を示します。

図 12-9 タイマ割り込みの誤差 (1/2)

(a) タイマ割り込みを許可したとき



上記①点で IPBTM0 フラグをセットしてタイマ割り込みを許可すると即座に割り込みが受け付けられます。

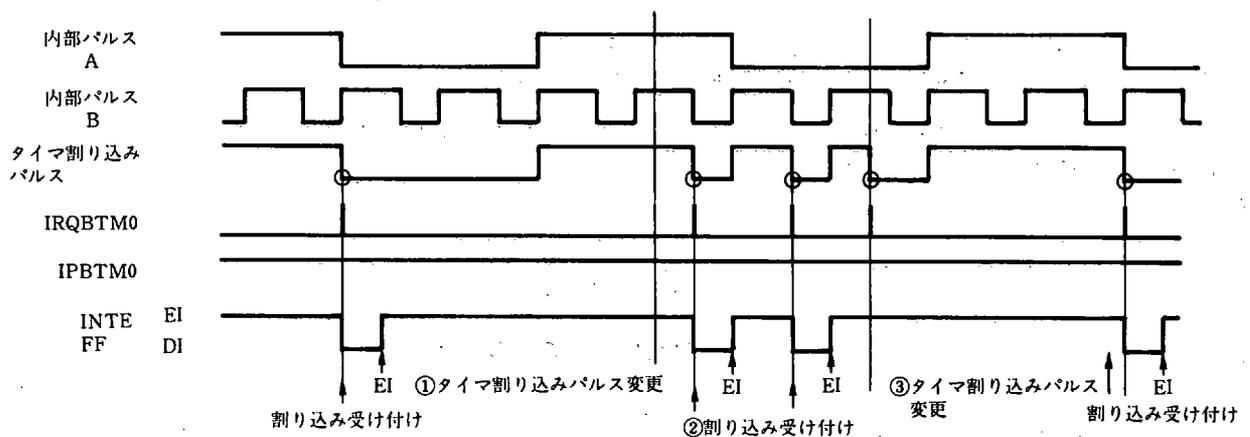
このときの誤差は $-t_{SET}$ になります。

次に②点で“EI”命令により割り込みを許可すると③点のタイマ割り込み用パルスの立ち下がりでの割り込みがかかります。

このときの誤差は $-t_{SET} < \text{誤差} < 0$ になります。

図 12-9 タイマ割り込みの誤差 (2/2)

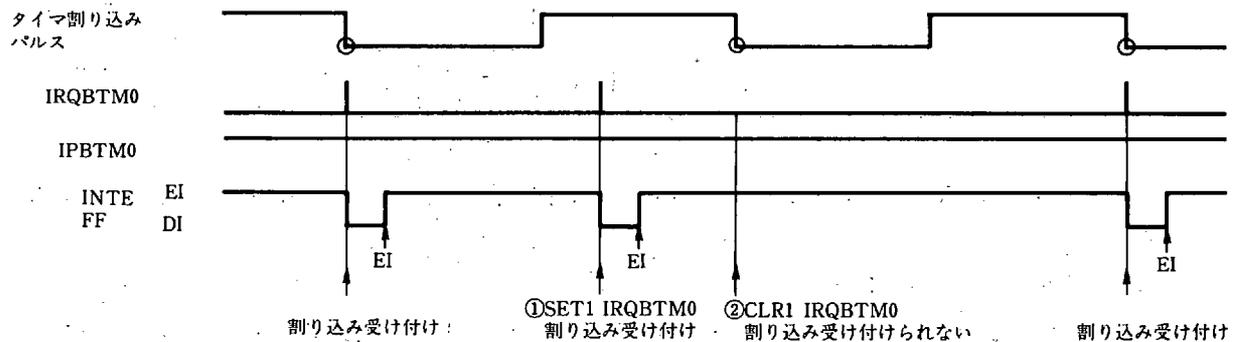
(b) タイマ割り込み用パルスを切り替えたとき



①でタイマ割り込みパルスをBに変更してもタイマ割り込み用パルスが立ち下がらないため、次の②で割り込みが受け付けられます。

③でタイマ割り込みパルスをAに変更するとタイマ割り込み用パルスが立ち下がるため、即座に割り込みが受け付けられます。

(c) IRQBTM0 フラグを操作したとき



①でIRQBTM0フラグをセットすると即座に割り込みが受け付けられます。

②でIRQBTM0フラグのリセットとタイマ割り込み用パルスの立ち下がりが重なると割り込みは受け付けられません。

12.6 タイマ割り込み使用時の注意

タイマ割り込みを使用して、パワーオン・リセットされたあとは常時一定のタイマが動作するようなプログラム、たとえば時計用のプログラム等を作成するときは、タイマ割り込み処理時間を一定時間以内に終了させる必要があります。

以下に例を基に説明します。

例

```

BR      AAA      ;リセット後, AAA へ分岐
TIMER :          ;プログラム・アドレス 0003H
ADD     M1, #0100B ; M1 の内容に 0100B を加算
SKT1   CY        ; キャリーが出れば時計処理
BR      AAA
; ①


時 計 処 理


EI
RETI
AAA :
INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
          ; 組み込みマクロ
          ; タイマ割り込み時間を 250 ms, タイマ・キャリー FF セット時間を 100 ms に設定
SET1    IPBTM0   ; 組み込みマクロ
EI      ; タイマ割り込みを許可


処 理 A

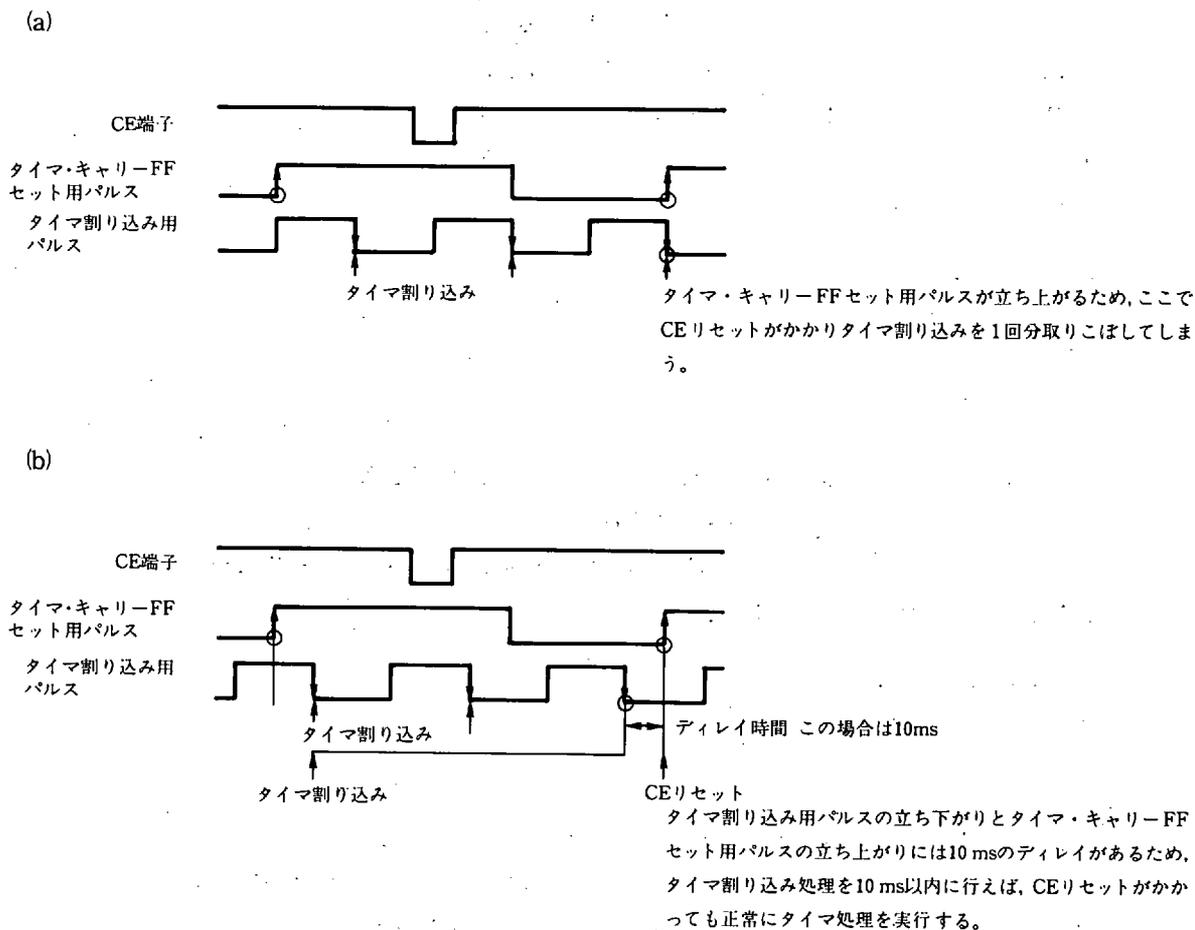

BR      AAA
    
```

この例では、処理Aを実行しながら1秒ごとに時計処理①を実行します。
 ここで図 12-10(a)に示すようにCE端子がロウ・レベルからハイ・レベルに変化すると、タイマ・キャリーFFセット用パルスの立ち上がりに同期してCEリセットがかかります。
 このとき、たまたまタイマ割り込み要求の発行とタイマ・キャリーFFのセットが重なるとCEリセットが優先されます。CEリセットがかかると、タイマ割り込み要求 (IRQBTM0フラグ) はリセットされてしまうため、タイマ処理を1回分取りこぼしてしまいます。

例に示したタイマ割り込みの取りこぼしを防ぐため、実際には図 12-10 の (b) に示すように“タイマ・キャリア FF セット用パルスの立ち下がり”と“タイマ割り込み用パルスの立ち下がり”にはディレイを設けてあります。

したがって図 12-10 の (2) に示したように、例においては時計処理の時間を 10ms 以内に行うことにより CE リセットによるタイマ割り込みの取りこぼしはなくなります。

図 12-10 タイミング・チャート



13. スタンバイ機能

スタンバイ機能は、バックアップ時にデバイスの消費電流を減少させる目的で使用します。

13.1 スタンバイ・ブロックの構成

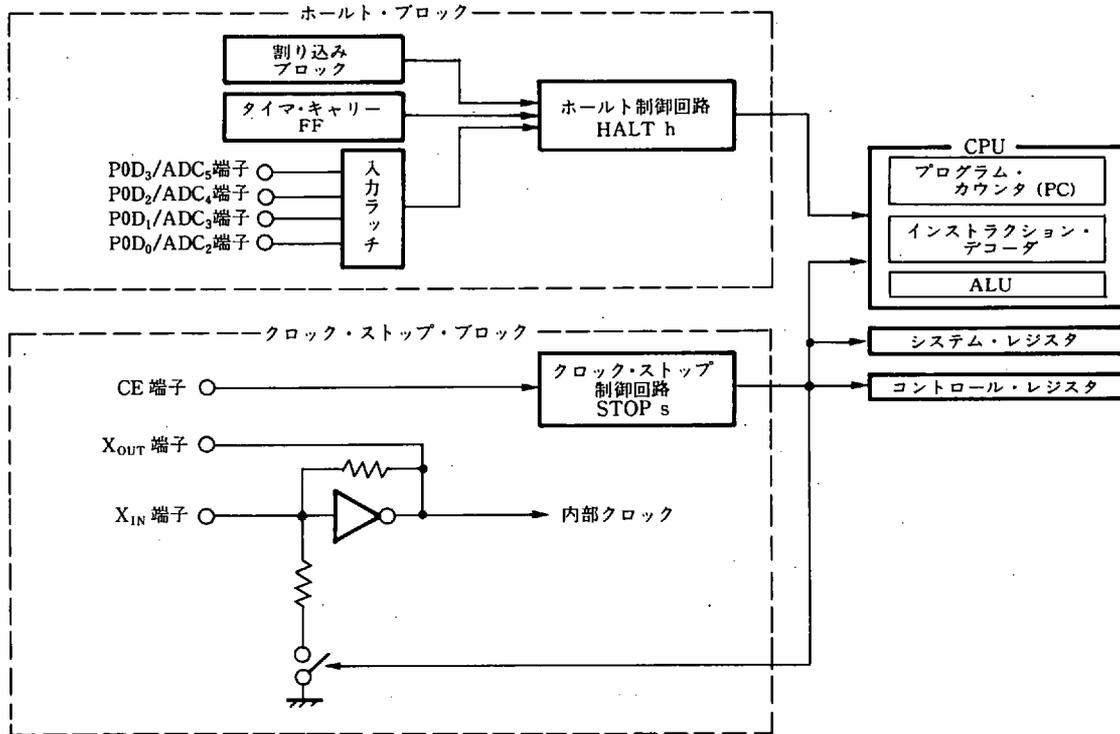
図 13-1 にスタンバイ・ブロックの構成を示します。

図 13-1 に示すようにスタンバイ・ブロックはホールド制御ブロックとクロック・ストップ制御ブロックに分けられます。

ホールド制御ブロックはホールド制御回路のほかに割り込み制御ブロック、タイマ・キャリー FF およびキー入力端子である P0D₀/ADC₂ 端子 - P0D₃/ADC₅ 端子から構成されており CPU (プログラム・カウンタ、インストラクション・デコーダおよび ALU ブロック) の動作を制御します。

クロック・ストップ制御ブロックはクロック・ストップ制御回路により 8 MHz 水晶振動子の発振回路、CPU、システム・レジスタおよびコントロール・レジスタを制御します。

図 13-1 スタンバイ・ブロックの構成



13.2 スタンバイ機能

スタンバイ機能は、デバイスの動作を一部またはすべて停止させることによりデバイスの消費電流を減少させる機能です。

スタンバイ機能は、ホールド機能とクロック・ストップ機能に分けられます。

ホールド機能は専用命令である“HALT h”命令を使用してCPUの動作を停止させることによりデバイスの消費電流を減少させます。

クロック・ストップ機能は、専用命令である“STOP s”命令を使用して8 MHz 水晶振動子の発振回路を停止させることによりデバイスの消費電流を減少させます。

ホールド機能とクロック・ストップ機能の他にCE端子によるデバイスの動作モードの設定があります。

13.3にCE端子によるデバイス動作モードの設定を説明します。

13.4および13.5にホールド機能およびクロック・ストップ機能を説明します。

備考 μPD17062の“STOP s”命令のオペランドsには0000Bを指定します。したがって実際の命令は“STOP 0000B”になります。

13.3 CE 端子によるデバイス動作モード

CE 端子は、外部から入力される信号の入力レベルおよび立ち上がりエッジにより以下に示す機能を制御します。

- (1) クロック・ストップ命令の有効、無効の制御
- (2) デバイスのリセット

13.3.1-13.3.2 に上記 (1), (2) を説明します。

13.3.1 クロック・ストップ命令の有効、無効の制御

クロック・ストップ命令である“STOP s”命令は、CE 端子がロウ・レベルのときのみ有効になります。

CE 端子がハイ・レベルのときに実行された“STOP s”命令は、ノー・オペレーション命令 (NOP) として実行されます。

13.3.2 デバイスのリセット

CE 端子をロウ・レベルからハイ・レベルに立ち上げるによりデバイスにリセット (CEリセット) をかけることができます。

リセットには CE リセットのほかに電源電圧 V_{DD} を投入したときのパワーオン・リセットがあります。

詳しくは“14. リセット機能”の項を参照してください。

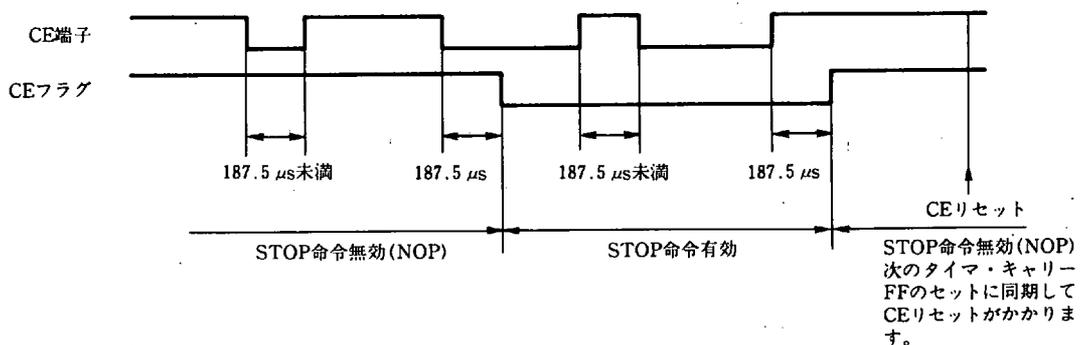
13.3.3 CE 端子への信号入力

CE 端子は、ノイズによる誤動作を防止するため187.5 μs未満のロウ・レベルまたはハイ・レベルは受け付けません。

また CE 端子に入力された信号の入力レベルは、コントロール・レジスタの CE フラグ (アドレス 07H のビット b_0) により検出することができます。

図 13-2 に入力信号と CE フラグの関係を示します。

図 13-2 CE 端子への入力信号と CE フラグの関係



13.4 ホールト機能

ホールト機能は“HALT h”命令を実行することによりCPUの動作クロックを停止します。

すなわち“HALT h”命令が実行されると、プログラムは“HALT h”命令で停止し、以後ホールト状態が解除されるまで“HALT h”命令で停止したままになります。

したがって、ホールト状態中のデバイス消費電流はCPUの動作電流分だけ減少します。

ホールト状態からの解除はタイマ・キャリーFF、割り込みおよびキー入力により行います。

タイマ・キャリーFF、割り込みおよびキー入力の解除条件は“HALT h”命令のオペランドである“h”により指定します。

“HALT h”命令はCE端子の入力レベルに関係なく有効です。

13.4.1 - 13.4.5 にホールト状態、ホールト解除条件および各ホールト解除条件について説明します。

13.4.1 ホールト状態

ホールト状態中はCPUの動作がすべて停止します。

すなわちプログラムの実行が“HALT h”命令で停止している状態になります。

ただし、周辺ハードウェアは“HALT h”命令以前に設定されている動作をそのまま継続します。

13.4.2 ホールト解除条件

図 13-3 にホールト解除条件を示します。

図 13-3 に示すようにホールト解除条件は“HALT h”命令のオペランド“h”で指定する4ビット・データにより設定します。

ホールト状態は、オペランド“h”に“1”で指定された条件が満たされると解除されます。

ホールト状態が解除されると“HALT h”命令の次の命令から実行を始めます。

このとき一度に複数の解除条件が設定されていると、設定されている条件のどれか1つでも満たされれば、ホールト状態は解除されます。

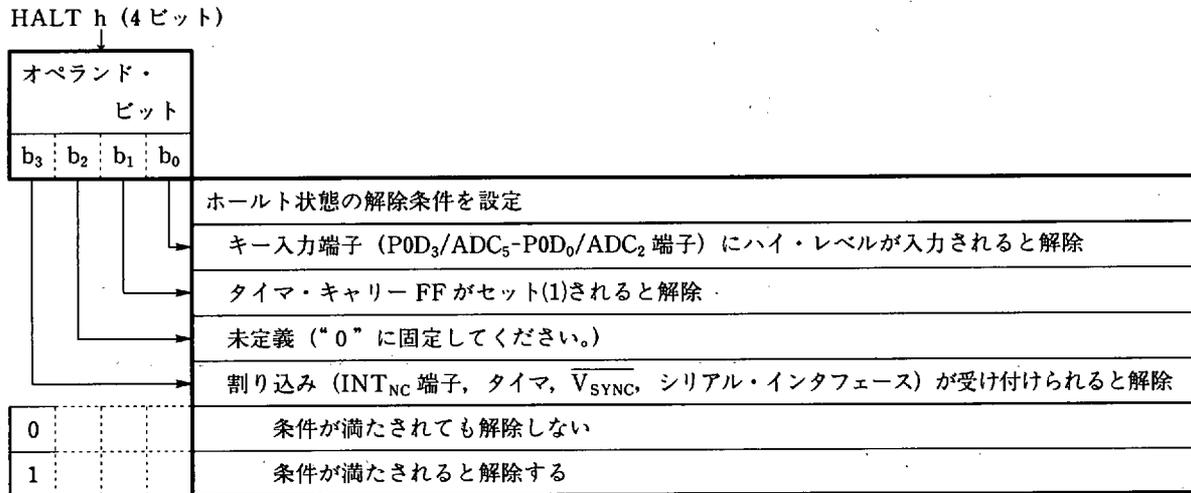
また、デバイスにリセット(パワーオン・リセットまたはCEリセット)がかかったときは、ホールト状態は解除されリセット動作を行います。

ホールト解除条件“h”に0000Bが設定されると、どの解除条件も設定されません。

このときはデバイスにリセット(パワーオン・リセットまたはCEリセット)がかかるとホールト状態が解除されます。

13.4.3 - 13.4.6 にタイマ・キャリー FF, 割り込みおよびキー入力の各ホールト解除条件について説明します。

図 13-3 ホールト解除条件



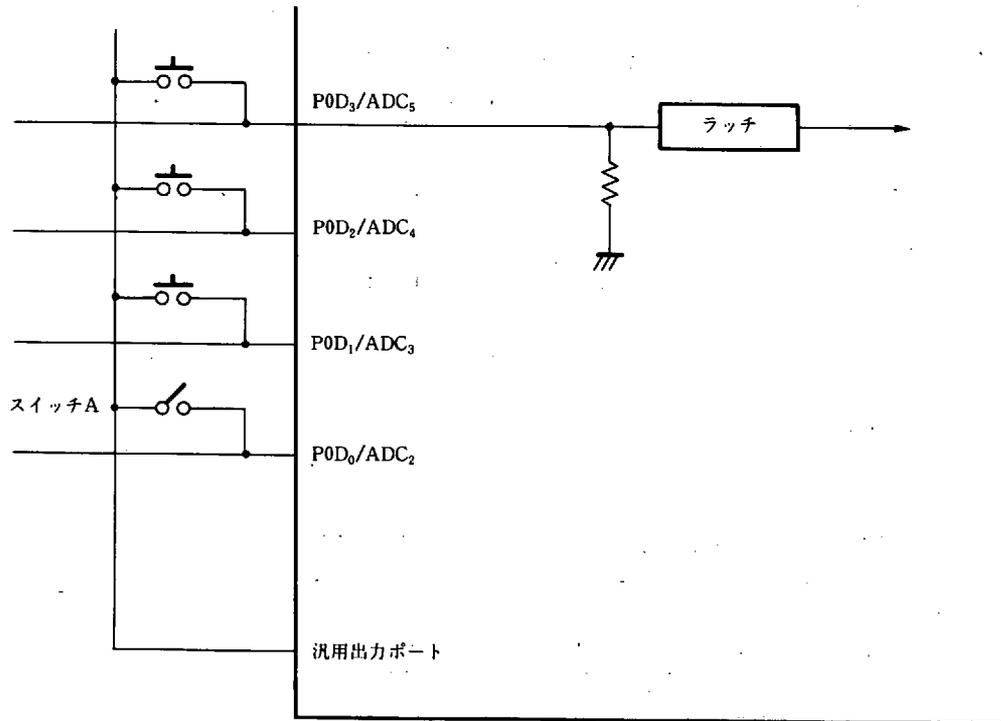
13.4.3 キー入力によるホールド解除

キー入力によるホールド解除条件は、“HALT 0001B” 命令で設定します。

キー入力によるホールド解除条件を設定すると、P0D₀/ADC₂ - P0D₃/ADC₅ 端子の4本のうちどれか1本にでもハイ・レベルが入力されたときにホールド状態を解除します。

汎用出力ポートをキー・ソース信号とする場合や P0D₀/ADC₂ - P0D₃/ADC₅ を A/D コンバータとして使用する場合の注意などを次の(1)-(3)に示します。

(1) 汎用出力ポートをキー・ソース信号とする場合の注意

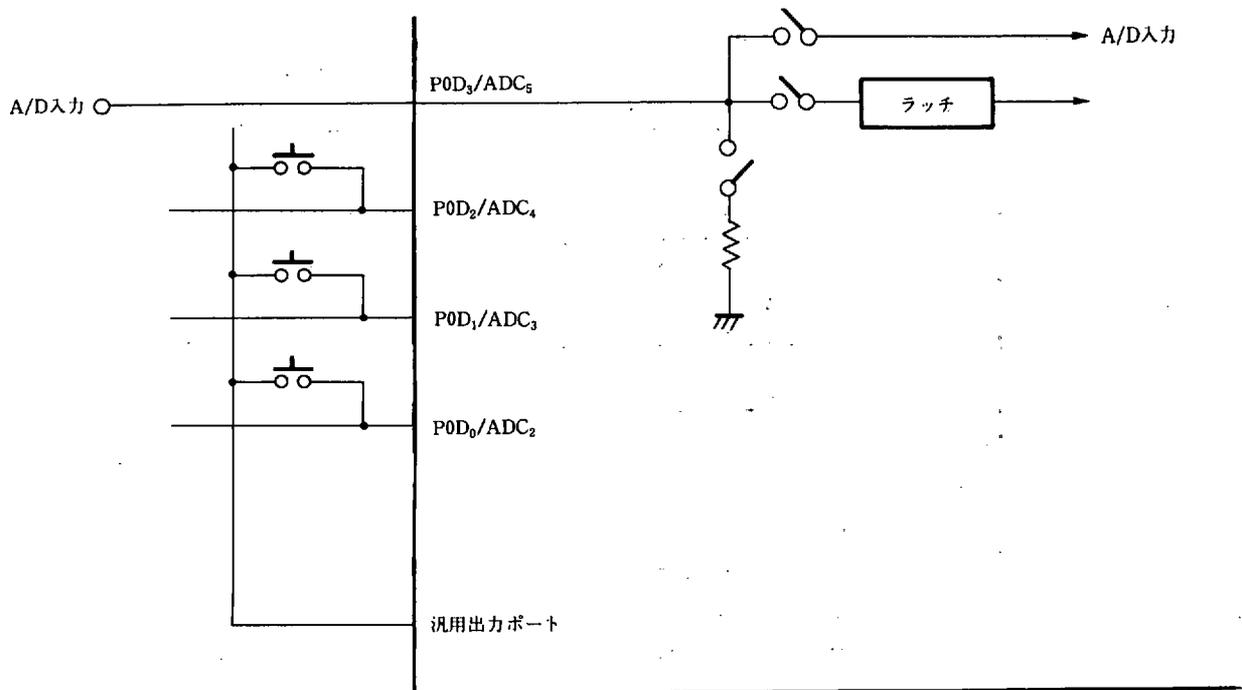


キー・ソース信号用の汎用出力ポートをハイ・レベルにしたあとに、“HALT 0001B” 命令を実行します。このとき上図に示すスイッチAのようにオルタネート・スイッチを用いていると、スイッチAが閉じている間は常に P0D₀/ADC₂ 端子にハイ・レベルが加わるためホールド状態はただちに解除されてしまいます。

したがって、オルタネート・スイッチを使用するときは十分注意してください。

P0D₀/ADC₂ - P0D₃/ADC₅ 端子は内部で自動的にプルダウンされます。

(2) P0D₀/ADC₂ - P0D₃/ADC₅ 端子を A/D コンバータとして使用する場合の注意



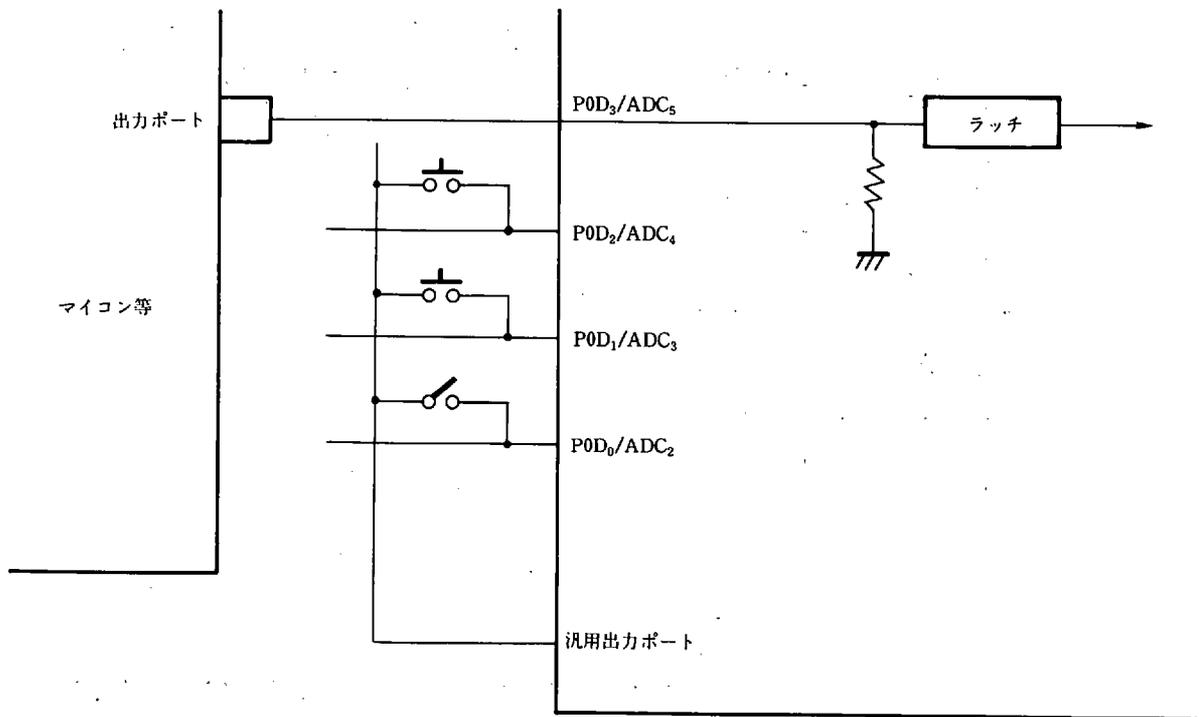
P0D₀/ADC₂ - P0D₃/ADC₅ 端子を A/D コンバータとして選択すると、選択された端子 (同時には 1 本しか選択できない) は入力ラッチから切り離され内部の A/D コンバータ入力へ接続されます。

このとき、A/D コンバータに選択した時点でその端子にたまたまハイ・レベルが入力されていると、ラッチ回路はハイ・レベルのまま保持されてしまいます。

この状態で“HALT 0001B”命令を実行すると、入力ラッチがハイ・レベルであるため命令が実行されてもただちにホールド状態が解除されてしまいます。

これを避けるためには、“HALT 0001B”命令を実行する前に入力ポートに設定し、A/D コンバータの入力にはロウ・レベルを設定してください。

(3) その他



P0D₀/ADC₂ - P0D₃/ADC₅ 端子は、プルダウン抵抗付きの汎用入力ポートとしても使用できます。
 したがって、ホールド解除は上図に示すように他のマイコン等により行うことも可能です。

13.4.4 タイマ・キャリー FF によるホールド解除

タイマ・キャリー FF によるホールド状態の解除は、“HALT 0010B” 命令で設定します。

タイマ・キャリー FF によるホールドの解除が設定されると、タイマ・キャリー FF がセット (1) されると同時にホールド状態が解除されます。

タイマ・キャリー FF はコントロール・レジスタの BTM0CY フラグ (アドレス 17H のビット b₀) と 1 対 1 に対応しており、一定時間ごと (5 ms, 100 ms) にセット (1) されます。

したがって、一定時間ごとにホールド状態を解除することが可能です。

以下に使用例を示します。

例

```

HLTTMR DAT 0010B ; シンボル定義
INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, NOT BTM0CK0
          ; 組み込みマクロ
          ; タイマ・キャリー FF のセット時間を 100 ms に設定

LOOP1:
  MOV     M1, #0110B
LOOP2:
  HALT    HLTTMR ; ホールド解除条件をタイマ・キャリー FF のセットに設定
  SKT1    BTM0CY ; 組み込みマクロ
  BR      LOOP ; BTM0CY フラグがセットされていなければ LOOP2 へ分岐
  ADD     M1, #0001B ; M1 の内容に 0001B を加算
  SKT1    CY ; 組み込みマクロ
  BR      LOOP2 ; キャリーが出れば処理 A を実行
  処 理 A
  BR      LOOP1

```

上記の例では 100 ms ごとにホールド状態が解除され、1 秒ごとに処理 A を実行します。

13.4.5 割り込みによるホールド解除

割り込みによるホールド状態の解除は“HALT 1000B”命令で設定します。

割り込みによるホールド状態の解除が設定されると、割り込みが受け付けられると同時にホールド状態が解除されます。

割り込み要因には、INT_{NC} 端子、タイマ、 $\overline{V_{SYNC}}$ 、シリアル・インタフェースの4要因があります。

したがって、どの割り込み要因でホールドを解除するかは、あらかじめプログラムで指定しておく必要があります。

また、割り込みが受け付けられるためには、各割り込み要因からの割り込み要求の発行以外に、すべての割り込みの許可 (EI 命令) および各割り込みごとの許可 (インタラプト・パーミッション・フラグがセットされている) がすべて満たされる必要があります。

したがって、割り込み要求が発行されてもその割り込みが許可されていなければ割り込みは受け付けられず、ホールド状態も解除されません。

割り込みの受け付けによりホールド状態が解除されると、プログラムの流れは各割り込みのベクタ・アドレスへ移ります。割り込みの処理後、“RETI”命令を実行するとプログラムの流れは“HALT”命令の次の命令に復帰します。

以下に使用例を示します。

例

```

HLTINT DAT 1000B ;シンボル定義
START : ;アドレス 0000H
BR MAIN ;
NOP
INTTM : ;タイマ割り込みベクタ・アドレス (0003H)
BR INTTIMER ;タイマ割り込み処理である INTTIMER に分岐
INT0 : ;INTNC 端子割り込みベクタ・アドレス (0004H)


処理 A

 ;INTNC 端子による割り込み処理
EI
RETI
INTTIMER :


処理 B

 ;タイマによる割り込み処理
EI
RETI
MAIN :
SET2 IPBTM0, IPNC ;組み込みマクロ
;INTNC 端子およびタイマの割り込みを許可
SET1 BTM0CK2 ;組み込みマクロ
LOOP : ;タイマ割り込みの時間間隔を 5ms に設定


処理 C

 ;メイン・ルーチン処理
EI ;すべての割り込みを許可
HALT HLTINT ;割り込みによるホールド解除の設定
;①
BR LOOP

```

上記の例では、タイマによる割り込み受け付け時にはホールド状態を解除して処理Bを実行し、INT_{NC} 端子による割り込み受け付け時には処理Aを実行します。

またホールド状態が解除されるたびに処理Cを実行します。

ホールド状態中にまったく同時に INT_{NC} 端子による割り込み要求とタイマによる割り込み要求が発行された場合は、ハードウェア優先順位の高い INT_{NC} 端子の処理Aを実行します。

処理Aの実行後、“RETI”が実行されると①の“BR LOOP”命令に復帰しますが、“BR LOOP”命令は実行されず、すぐタイマ割り込みが受け付けられます。

タイマ割り込み処理である処理Bの実行後、“RETI”命令が実行されると、“BR LOOP”命令を実行します。

13.5 クロック・ストップ機能

クロック・ストップ機能は“STOP s”命令を実行することにより 8 MHz 水晶振動子の発振回路を停止します(クロック・ストップ状態)。

このクロック・ストップ機能により、μD17062の消費電流は10 μA MAX.まで減少させることができます。

“STOP s”命令のオペランド“s”には“0000B”を指定します。

“STOP s”命令はCE端子がロウ・レベルのときのみ有効となり、CE端子がハイ・レベルの間に実行されてもノー・オペレーション命令(NOP)として実行されます。

すなわち、CE端子がロウ・レベルのときに“STOP s”命令を実行する必要があります。

クロック・ストップ状態を解除するためには、CEリセットにより行います。

13.5.1-13.5.3にクロック・ストップ状態、クロック・ストップ状態の解除およびクロック・ストップ命令使用時の注意を説明します。

13.5.1 クロック・ストップ状態

クロック・ストップ状態は水晶振動子の発振回路が停止するため、CPUおよび周辺ハードウェア等のデバイス動作はすべて停止します。

クロック・ストップ状態中はデバイスの電源電圧 V_{DD} を約2.2Vまで下げても停電検出回路は動作しません。したがって低電圧でのデータ・メモリ・バックアップが可能になります。

13.5.2 クロック・ストップ状態の解除

クロック・ストップ状態の解除はCE端子をロウ・レベルからハイ・レベルに立ち上げる(CEリセット)か、デバイスの電源電圧 V_{DD} を一度2.2V以下に下げた後から4.5Vまで立ち上げる(パワーオン・リセット)ことにより行います。

図13-4および図13-5にそれぞれCEリセット時とパワーオン・リセット時の解除動作を示します。

パワーオン・リセットによるクロック・ストップ状態の解除を行うと停電検出回路が動作します。

図 13-4 CE リセットによるクロック・ストップの解除

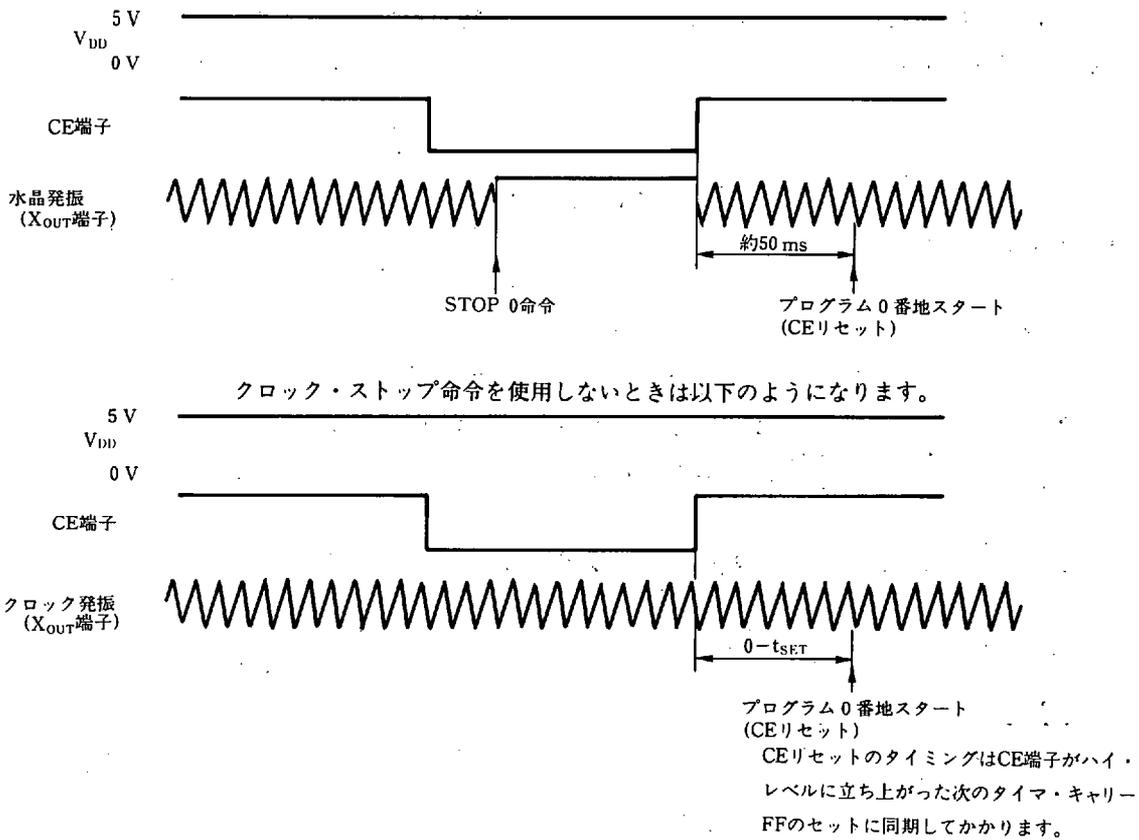
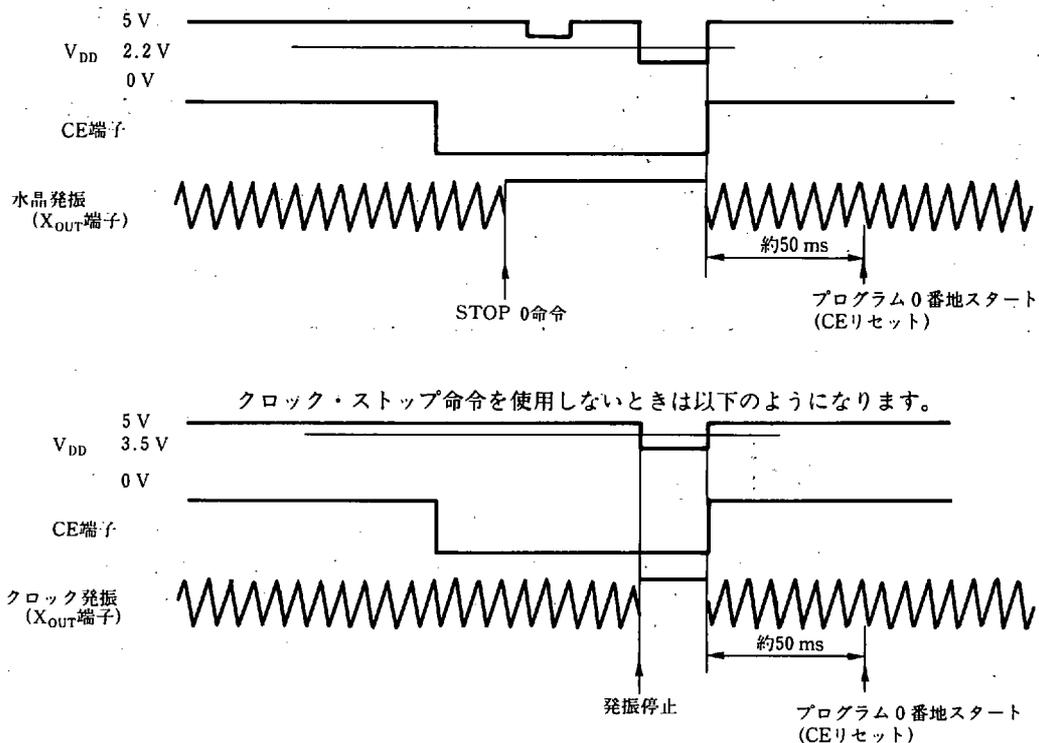


図 13-5 パワーオン・リセット時のクロック・ストップ解除



13.5.3 クロック・ストップ命令使用時の注意

クロック・ストップ命令 (STOP s命令) は CE 端子がロウ・レベルのときのみ有効になります。

したがって、CE 端子がたまたまハイ・レベルであったときの処理をプログラム上考慮しておく必要があります。以下の例を基に説明します。

例

```

XTAL DAT 0000B ;クロック・ストップ条件のシンボル定義
CEJDG :
; ①
SKF1 CE ;組み込みマクロ
; CE端子の入力レベルを検出
BR MAIN ;CE =ハイ・レベルならメイン処理へ分岐
[ 処 理 A ] ; CE =ロウ・レベルの処理
; ②
STOP XTAL ;クロック・ストップ
; ③
BR $-1
MAIN :
[ メイン処理 ]
BR CEJDG
    
```

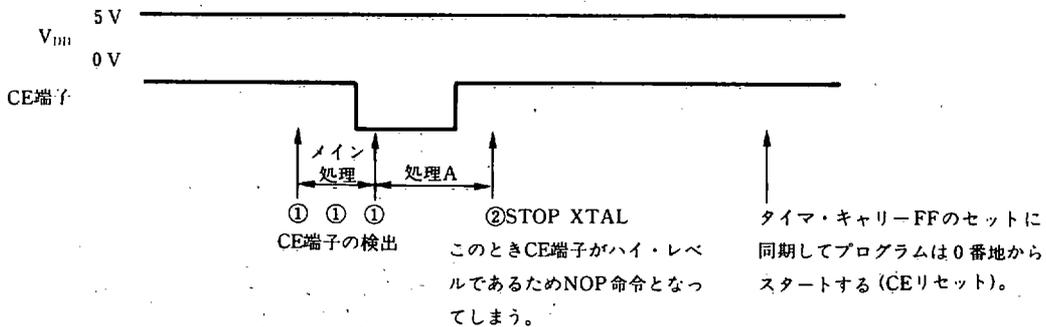
上記の例では①でCE端子の状態を検出し、CE端子がロウ・レベルであれば処理Aを実行後②のクロック・ストップ命令“STOP XTAL”を実行します。

ところが、下図に示すように②の“STOP XTAL”命令実行中にCE端子がハイ・レベルになってしまうと“STOP XTAL”命令はノー・オペレーション命令 (NOP) として動作してしまいます。

このとき、仮に③の分岐命令“BR \$-1”がなかったとすると、プログラムはメイン処理に移ってしまい誤動作することが考えられます。

したがって、プログラム上は③のように分岐命令を挿入するか、またはメイン処理に移っても誤動作しないようなプログラムにしておく必要があります。

また②のように分岐命令を用いたときは、CE端子がハイ・レベルのままであっても、次のタイマ・キャリーFFのセットと同期してCEリセットがかかります。



13.6 ホールトおよびクロック・ストップ時のデバイス動作

13.6.1 ホールトおよびクロック・ストップ時の各端子の状態

表 13-1 にホールト状態中とクロック・ストップ状態中の CPU および周辺ハードウェアの動作を示します。

表 13-1 に示すようにホールト状態中は命令の実行が停止する以外はすべての周辺ハードウェアは通常動作を継続します。

また、クロック・ストップ状態中はすべての周辺ハードウェアが動作を停止します。

周辺ハードウェアの動作状態を制御するコントロール・レジスタは、ホールト状態中は通常通り動作（イニシャライズされない）しますが、クロック・ストップ状態（STOP s 命令が実行された時点）中は所定の値にイニシャライズされます。

したがって、言い替えば各周辺ハードウェアは、ホールト状態中はコントロール・レジスタに設定されている動作を継続し、クロック・ストップ状態中は所定の値にイニシャライズされたコントロール・レジスタに従い動作状態が決定されます。

コントロール・レジスタのイニシャライズされる値については“9. レジスタ・ファイル (RF)”の項を参照してください。

また、以下に例を示します。

例 Port0A の P0A₀/SDA と P0A₁/SCL 端子を出力ポートに設定し、P0A₂/SCK と P0A₃/SO 端子をシリアル・インタフェースとして使用しているとき

```

HLTINT  DAT 1000B ; シンボル定義
XTAL    DAT 0000B ;
INITFLG P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
        ; 組み込みマクロ
; ①
SET2    P0A0, P0A1;
INITFLG SIO0CH, NOT SB, SIO0MS, SIO0TX
        ;
SET2    SIO0CK1, SIO0CK0
; ②
SET2    SIO0IMD1, SIO0IMD0
CLR1    IRQSIO0
SET1    IPSIO0
EI
; ③
SET1    SIO0NWT
; ④
HALT    HLTINT
; ⑤
STOP    XTAL
    
```

上記の例では、①で P0A₀, P0A₁ 端子からハイ・レベルを出力し②でシリアル・インタフェースの条件設定を行い、③でシリアル通信をスタートしています。

このとき④の“HALT”命令が実行されると、シリアル通信はそのまま継続し、シリアル・インタフェースによる割り込みが受け付けられた時点でホールト状態を解除します。

また④の“HALT”命令のかわりに⑤の“STOP”命令が実行されると“STOP”命令が実行された時点で①、②および③で設定したコントロール・レジスタのフラグがすべてイニシャライズされてしまうため、シリアル通信を中断し Port0A のすべての端子は汎用入力ポートに設定されます。

表 13-1 ホールト状態とクロック・ストップ状態中のデバイス動作

周辺ハードウェア	状 態			
	CE 端子=ハイ・レベル		CE 端子=ロウ・レベル	
	ホールト時	クロック・ストップ時	ホールト時	クロック・ストップ時
プログラム・カウンタ	HALT 命令のアドレスでストップ	STOP 命令は無効(NOP)	HALT 命令のアドレスでストップ	0000H にイニシャライズされ、ストップ
システム・レジスタ	保 持		保 持	イニシャライズ注
周辺ハードウェアレジスタ	”		”	保 持
コントロール・レジスタ	”		”	イニシャライズ注
タイマ	通常動作		通常動作	動作停止
A/D コンバータ	”		”	”
D/A コンバータ	”		”	”
シリアル・インタフェース	動作停止		動作停止	”
汎用入出力兼用ポート	通常動作		通常動作	入力ポート
汎用入力ポート	”		”	入力ポート
汎用出力ポート	”		”	保 持
IDC	HALT 命令実行時の状態を保持		動作停止	動作停止

注 イニシャライズされる値については“8. システム・レジスタ (SYSREG)”および“9. レジスタ・ファイル (RF)”の項を参照してください。

13.6.2 ホールト状態およびクロック・ストップ状態中の各端子の処理上の注意

ホールト機能は、たとえば時計のみを動作させるときなどの消費電流を減らす目的で使用されます。また、クロック・ストップ機能はデータ・メモリのみ保持するために消費電流を減らす目的で使用されます。したがって、ホールト状態およびクロック・ストップ状態中は極力消費電流を減少させる必要があります。このとき、消費電流は各端子の状態により大きく異なることがあるため、表 13-2 に示すような注意が必要です。

表 13-2 ホールトおよびクロック・ストップ状態中の各端子の状態と注意 (1/2)

端子機能		端子記号	各端子の状態と処理上の注意	
			ホールト状態	クロック・ストップ状態
汎用 入 出 力 ポ ー ト	Port0A	P0A ₃ /SO P0A ₂ /SCK P0A ₁ /SCL P0A ₀ /SDA	ホールト以前の状態をそのまま保持します。 (1) 出力端子に指定されているとき ハイ・レベル出力中に外部でプルダウンされていたり、ロウ・レベル出力中に外部でプルアップされていると消費電流が増加します。 N-ch オープン・ドレイン出力 (P0A ₁ , P0A ₀ , P1A ₃ - P1A ₀) は特に注意してください。 (2) 入力端子に指定されているとき フローティング状態になっているとノイズにより消費電流が増加します。 (3) Port0D (P0D ₃ /ADC ₇ - P0D ₀ /ADC ₄) プルダウン抵抗を内蔵しているため、外部でプルアップされていると消費電流が増加します。 ただし、A/D コンバータに選択されている端子はプルダウン抵抗はオフします。 (4) Port0B (P0B ₃ /HSCNT - P0B ₀ /SI) Port1B (P1B ₃ /P1B ₀) P0B ₃ /HSCNT 端子が HSYNC カウンタとして動作しているときおよび P1B ₃ 端子が外部タイマ入力として動作しているときは内蔵の自己バイアス回路が動作して消費電流が増加します。	すべて汎用入力ポートに指定されます。 このとき P0A ₁ /SCL および P0A ₀ /SDA 端子を除いたすべての入力ポートは、外部でフローティング状態であってもノイズによる消費電流は増加しない回路になっています。 P0A ₁ /SCL および P0A ₀ /SDA 端子はノイズなどで消費電流が増加しないように外部でプルダウンまたはプルアップする必要があります。 Port0D (P0D ₃ /ADC ₅ - P0D ₀ /ADC ₂) は内部でプルダウンされます。
		Port0B		
	Port1B	P1B ₃ P1B ₂ P1B ₁ P1B ₀		
	Port1C	P1C ₃ /ADC ₁ P1C ₂ P1C ₁		
汎用 入 力 ポ ー ト	Port0D	P0D ₃ /ADC ₅ P0D ₂ /ADC ₄ P0D ₁ /ADC ₃ P0D ₀ /ADC ₂		
汎用 出 力 ポ ー ト	Port0C	P0C ₃ P0C ₂ P0C ₁ P0C ₀		汎用出力ポートに指定されます。 出力されている内容はそのまま保持されます。 したがってハイ・レベル出力中に外部でプルダウンされていたり、ロウ・レベル出力中にプルアップされていると消費電流が増加します。

表 13-2 ホールトおよびクロック・ストップ状態中の各端子の状態と注意 (2/2)

端子機能	端子記号	各端子の状態と処理上の注意	
		ホールト状態	クロック・ストップ状態
割り込み	INT _{NC}	フローティング状態になっていると外部ノイズ等により消費電力が増加します。	
IDC	RED GREEN BLUE BLANK \overline{H}_{SYNC} \overline{V}_{SYNC}	出力端子は HALT 命令実行時の状態を保持します。IDCEN フラグがセットされているときは消費電流が増加します。	IDC ディスエーブルになります。 各端子は以下のようになります。 RED, GREEN, BLUE はロウ・レベル出力, \overline{H}_{SYNC} 端子および \overline{V}_{SYNC} 端子はフローティング状態となっても消費電流は増加しない回路になっています。
D/A コンバータ	PWM ₃ PWM ₂ PWM ₁ PWM ₀	汎用出力ポートと同様の注意が必要です。	すべての端子がロウ・レベルを出力します。
A/D コンバータ	ADC ₀	フローティング状態になります。	
クロック発振回路	X _{IN} X _{OUT}	クロック発振回路の発振波形により消費電流が変化します。 振幅が大きいほど消費電流は小さくなります。 発振振幅は使用する水晶振動子や負荷コンデンサにより左右されるため、評価が必要です。	X _{IN} 端子が内部でプルダウンされ、X _{OUT} 端子はハイ・レベルを出力します。

14. リセット機能

リセット機能は、デバイス動作の初期化を行うために使用します。

14.1 リセット・ブロックの構成

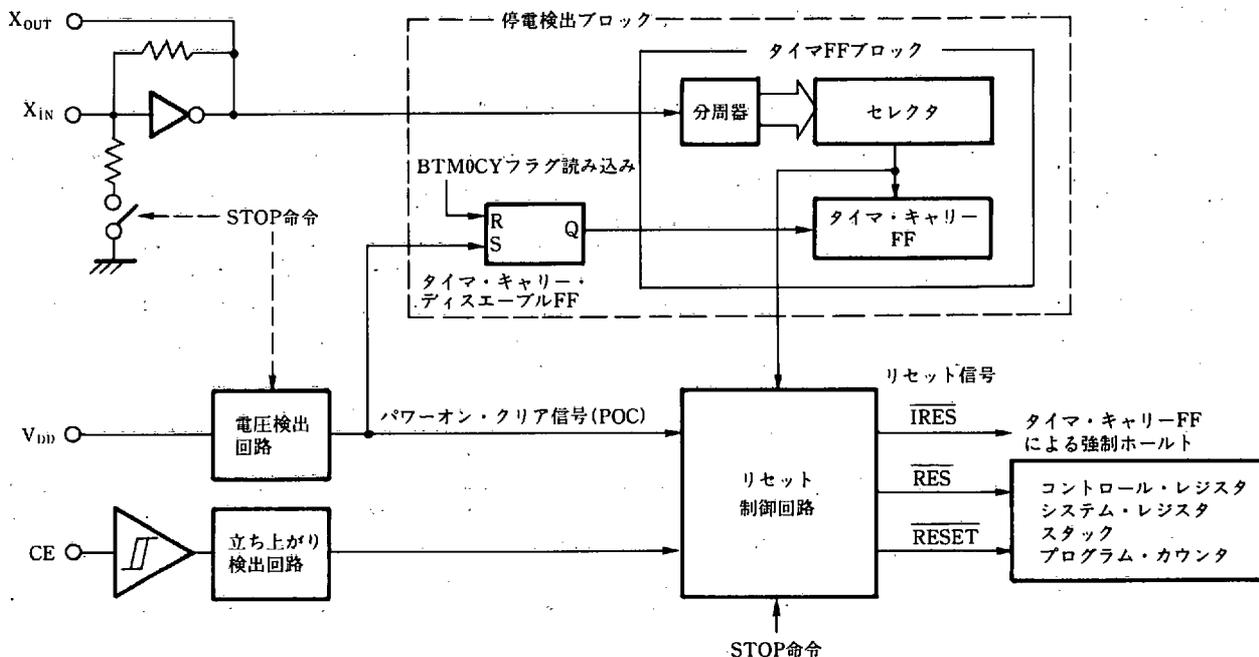
図 14-1 にリセット・ブロックの構成を示します。

デバイスのリセットは電源電圧 V_{DD} の投入によるリセット（パワーオン・リセットまたは V_{DD} リセット）と CE 端子によるリセット（CE リセット）に分けられます。

パワーオン・リセット・ブロックは V_{DD} 端子から入力された電圧を検出する電圧検出回路、停電検出回路およびリセット制御回路から構成されています。

CE リセット・ブロックは CE 端子に入力された信号の立ち上がりを検出する回路とリセット制御回路から構成されています。

図 14-1 リセット・ブロックの構成



14.2 リセットの機能

パワーオン・リセットは電源電圧 V_{DD} が一定電圧以下から立ち上がったときにかかり、CE リセットは CE 端子がロウ・レベルからハイ・レベルに立ち上がったときにかかります。

パワーオン・リセットは、プログラム・カウンタ、スタック、システム・レジスタおよびコントロール・レジスタ等の初期化を行い、プログラムを 0000H 番地から実行します。

CE リセットは、プログラム・カウンタ、スタック、システム・レジスタおよびコントロール・レジスタの一部などの初期化を行い、プログラムを 0000H 番地から実行します。

パワーオン・リセットと CE リセットの違いはおもに、初期化されるコントロール・レジスタと 14.6 項で説明する停電検出回路の動作です。

パワーオン・リセットと CE リセットは、図 14-1 に示したリセット制御回路から出力されるリセット信号 \overline{IRES} 、 \overline{RES} および \overline{RESET} 信号により制御されます。

表 14-1 に \overline{IRES} 、 \overline{RES} および \overline{RESET} 信号とパワーオン・リセットおよび CE リセットの関係を示します。

また“13. スタンバイ”の項で説明したクロック・ストップ命令 (STOP) 実行時もしリセット制御回路が動作します。

14.3 および 14.4 項にそれぞれ CE リセットとパワーオン・リセットについて説明します。

14.5 項に CE リセットとパワーオン・リセットの関係を説明します。

表 14-1 内部リセット信号と各リセットの関係

内部リセット信号	出力される信号			各リセット信号により制御される内容
	CE リセット時	パワーオン・リセット時	クロック・ストップ時	
\overline{IRES}	×	○	○	デバイスを強制的にホールド状態にする。ホールドの解除はタイマ・キャリー FF のセットにより行う。
\overline{RES}	×	○	○	コントロール・レジスタの一部を初期化する。
\overline{RESET}	○	○	○	プログラム・カウンタ、スタック、システム・レジスタおよびコントロール・レジスタの一部を初期化する。

14.3 CE リセット

CE リセットは CE 端子をロウ・レベルからハイ・レベルに立ち上げるにより行います。

CE 端子がハイ・レベルに立ち上がると、次のタイマ・キャリー FF セット用パルスの立ち上がりエッジに同期して $\overline{\text{RESET}}$ 信号が出力され、デバイスにリセットがかかります。

CE リセットがかかると、 $\overline{\text{RESET}}$ 信号によりプログラム・カウンタ、スタック、システム・レジスタおよびコントロール・レジスタの一部を初期値にイニシャライズしてプログラムを 0000H 番地から実行します。

初期値については各項を参照してください。

CE リセットはクロック・ストップを使用するときと使用しないときではリセット動作が異なります。

これを 14.3.1 および 14.3.2 項にそれぞれ示します。

14.3.3 に CE リセット時の注意を示します。

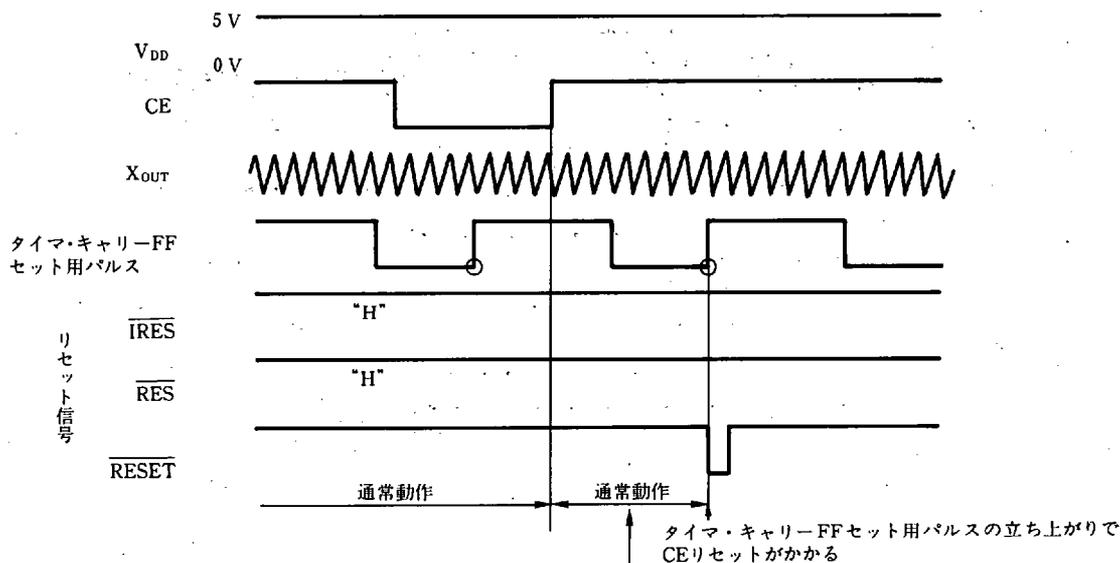
14.3.1 クロック・ストップ (STOP 命令) を使用しないときの CE リセット

図 14-2 に動作を示します。

クロック・ストップ (STOP 命令) を使用しないときは、コントロール・レジスタのタイマ・モード・セレクト・レジスタはイニシャライズされません。

したがって CE 端子がハイ・レベルになったあと、そのとき選択されているタイマ・キャリー FF セット用パルス (5 ms, 100 ms) の立ち上がりエッジにより $\overline{\text{RESET}}$ 信号が出力され、リセットがかかります。

図 14-2 クロック・ストップを使用しないときの CE リセット動作



そのとき選択されているタイマ・キャリーFFセット時間が t_{SET} であるとこの期間 t は CE 端子の立ち上げタイミングにより $0 < t < t_{\text{SET}}$ になる
また、この期間はプログラムは動作を継続する

14.3.2 クロック・ストップ (STOP 命令) を使用するときの CE リセット

図 14-3 に動作を示します。

クロック・ストップを使用するときは、“STOP s” 命令が実行された時点で $\overline{\text{IRES}}$ 、 $\overline{\text{RES}}$ および $\overline{\text{RESET}}$ 信号が出力されます。

このとき $\overline{\text{RES}}$ 信号によりコントロール・レジスタのタイマ・モード・セレクト・レジスタが 0000B に初期化されるため、タイマ・キャリー FF セット用信号は 100 ms に指定されます。

また、CE 端子がロウ・レベルの間は $\overline{\text{IRES}}$ 信号が出力され続けるためタイマ・キャリー FF による解除の強制ホールド状態になります。

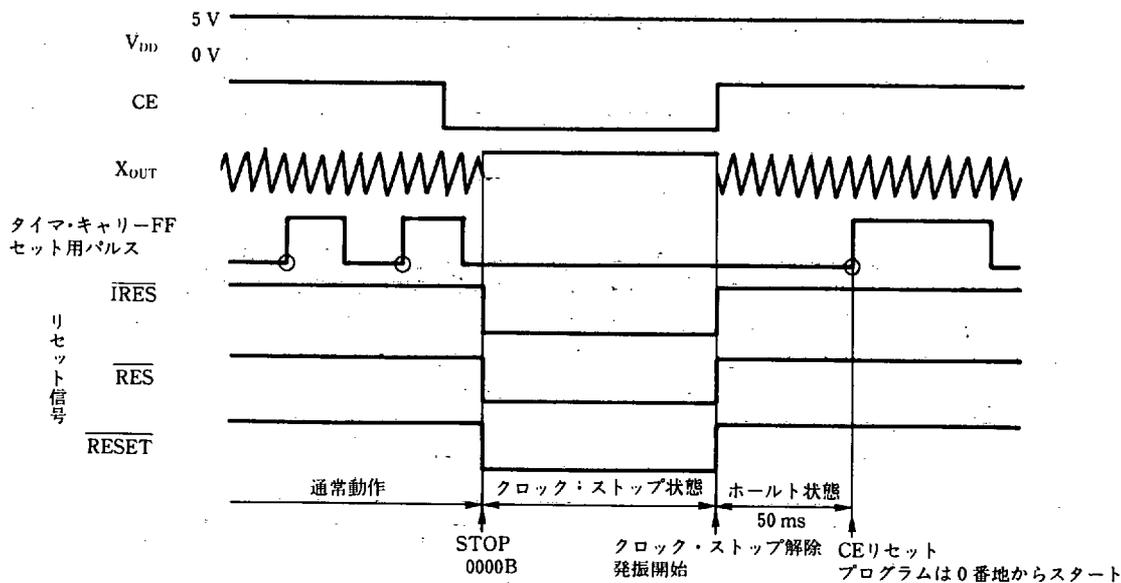
ただし、クロック自身が停止しているためデバイスは動作を停止しています。

CE 端子がハイ・レベルに立ち上がるとクロック・ストップ状態が解除され発振が開始します。

このとき $\overline{\text{IRES}}$ 信号によりタイマ・キャリー FF による解除のホールド状態になっているため、CE 端子が立ち上がったあと、タイマ・キャリー FF セット用パルスが立ち上がるとホールド状態が解除されプログラムが 0 番地からスタートします。

タイマ・キャリー FF セット用パルスは 100 ms に初期化されているため、CE 端子がハイ・レベルに立ち上がってから 50 ms 後に CE リセットがかかります。

図 14-3 クロック・ストップを使用するときの CE リセット動作



14.3.3 CEリセット時の注意

CEリセットは、実行中の命令に関係なくかかるため以下の(1)および(2)に示す点に注意が必要です。

(1) 時計などのタイマ処理を実行する時間

タイマ・キャリー FF やタイマ割り込みを使用して時計のプログラムを作成する場合は、そのプログラムの処理時間を一定時間以内に終了させる必要があります。

詳しくは、“12.4 タイマ・キャリー FF 使用時の注意”および“12.6 タイマ割り込み使用時の注意”の項を参照してください。

(2) プログラムに使用しているデータやフラグなどの処理

1命令で処理できないデータやフラグなどにおいて、CEリセットがかかっても内容が変化してはいけないようなもの、たとえばラスト・チャンネルなどはその内容を書き換えるときに注意が必要です。

以下に例を基に説明します。

例1.

; ①

LCTUNE :

初期受信処理

M1, M2の内容のチャンネルを受信

; ラスト・チャンネルの受信

MAIN :

チャンネル変更

; メイン処理

; チャンネルが変更されたら、変更後のチャンネルをジェネラル・レジスタ R1, R2 に代入する。

; ②

ST M1, R1

; ラスト・チャンネルの書き換え

; ③

ST M2, R2

BR MAIN

上記の例1において、いま仮にラスト・チャンネルが、“12H”であったとすると、データ・メモリ M1, M2の内容はそれぞれ“1H”, “2H”になります。

このときCEリセットがかかると、①でラスト・チャンネル 12 chを受信します。

また、メイン処理でチャンネルが変更されるときは、変更されたチャンネルを②, ③でM1, M2に書き直します。

ここで、仮にチャンネルが“04H”に変更されたとすると②, ③でM1, M2に“0H”, “4H”を書き込みます。ところが、②を実行した時点でたまたまCEリセットがかかると③を実行しないでリセットがかかってしまいます。

したがって、ラスト・チャンネルは“02H”になってしまい、①で02 chを受信してしまいます。

このような場合は、以下の例2に示すようなプログラムで対策します。

例 2.

```

; ④
SKT1  FLG1          ; FLG1 フラグが "1" であれば,
BR     LCTUNE
ST     M1, R1       ; M1, M2 に再度書き込みを行う。
ST     M2, R2
CLR1   FLG1
; ①
LCTUNE :
    初期受信処理          ; ラスト・チャンネルの受信
    M1, M2 の内容のチャンネルを
    受信
MAIN :
    チャンネル変更        ; チャンネルが変更されたら, 変更後のチャンネルをジェネラル・
                           ; レジスタ R1, R2 に代入する。
; ⑤
SET1   FLG1          ; ラスト・チャンネルを書き換える間は FLG1 フラグをセットする。
; ②
ST     M1, R1       ; ラスト・チャンネルの書き換え
; ③
ST     M2, R2
CLR1   FLG1
BR     MAIN
    
```

上記の例 2 では、②、③ でラスト・チャンネルを書き換えるときに FLG1 フラグをセットしています。したがって、③ でたまたま CE リセットがかかっても ④ で再度書き込みを行います。

14.4 パワーオン・リセット

パワーオン・リセットはデバイスの電源電圧 V_{DD} を一定電圧 (パワーオン・クリア電圧と呼ぶ) 以下から立ち上げるにより行います。

電源電圧 V_{DD} がパワーオン・クリア電圧以下のときは、図 14-1 に示した電圧検出回路からパワーオン・クリア信号 (POC) が出力されます。

パワーオン・クリア信号が出力されると、水晶発振回路を停止しデバイス動作を停止します。

また、パワーオン・クリア信号出力中は \overline{IRES} 、 \overline{RES} および \overline{RESET} 信号が出力されます。

電源電圧 V_{DD} がパワーオン・クリア電圧を越えるとパワーオン・クリア信号が切れ、水晶発振を開始すると同時に \overline{IRES} 、 \overline{RES} および \overline{RESET} 信号も切れます。

このとき、 \overline{IRES} 信号によりタイマ・キャリー FF による解除のホールド状態になっているため、次のタイマ・キャリー FF セット用信号の立ち上がりエッジでパワーオン・リセットがかかります。

タイマ・キャリー FF セット用信号は \overline{RESET} 信号により 100 ms に初期化されているため、電源電圧 V_{DD} がパワーオン・クリア電圧を越えてから 50 ms 後にリセットがかかりプログラムが 0 番地からスタートします。

この動作を図 14-4 に示します。

パワーオン・リセット時のプログラム・カウンタ、スタック、システム・レジスタおよびコントロール・レジスタの初期化は、パワーオン・クリア信号が出力された時点で行われます。

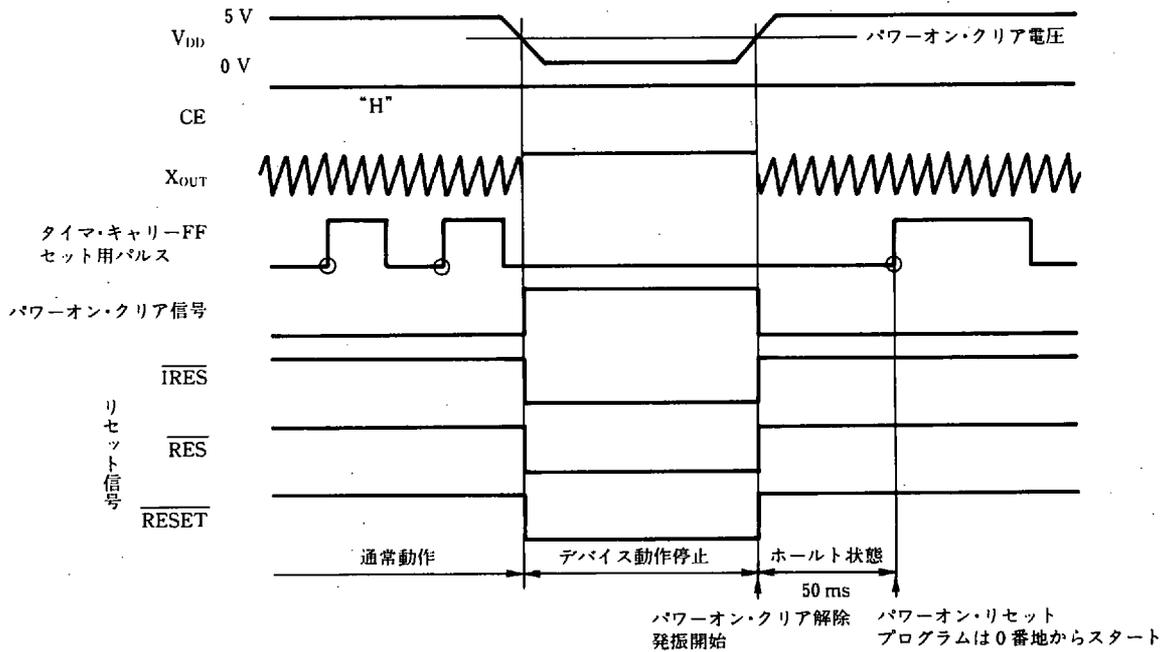
また、初期値については各項を参照してください。

パワーオン・クリア電圧は通常動作時は 3.5 V (規格値) であり、クロック・ストップ状態中は 2.2 V (規格値) になります。

このときの動作を 14.4.1 および 14.4.2 に示します。

また電源電圧 V_{DD} を 0 V から立ち上げたときの動作を 14.4.3 に示します。

図 14-4 パワーオン・リセットの動作



14.4.1 通常動作時のパワーオン・リセット

図 14-5 の (a) に動作を示します。

図 14-5 の (a) に示すように CE 端子の入力レベルに関係なく電源電圧 V_{DD} が 3.5 V より下がるとパワーオン・クリア信号が出力されデバイスの動作が停止します。

次に電源電圧 V_{DD} が再度 3.5 V 以上に立ち上がると、50 ms のホールド後プログラムが 0000H 番地からスタートします。

通常動作時とは、クロック・ストップ命令を使用していないときで、ホールド命令によるホールド状態中も含まれます。

14.4.2 クロック・ストップ状態中のパワーオン・リセット

図 14-5 の (b) に動作を示します。

図 14-5 の (b) に示すように電源電圧 V_{DD} が 2.2 V より下がるとパワーオン・クリア信号が出力されデバイス動作が停止します。

ただし、クロック・ストップ状態であるため、見かけ上デバイス動作は変化しません。

次に電源電圧 V_{DD} が 3.5 V 以上に立ち上がると 50 ms のホールド後プログラムが 0000H 番地からスタートします。

14.4.3 電源電圧 V_{DD} が 0 V から立ち上がったときのパワーオン・リセット

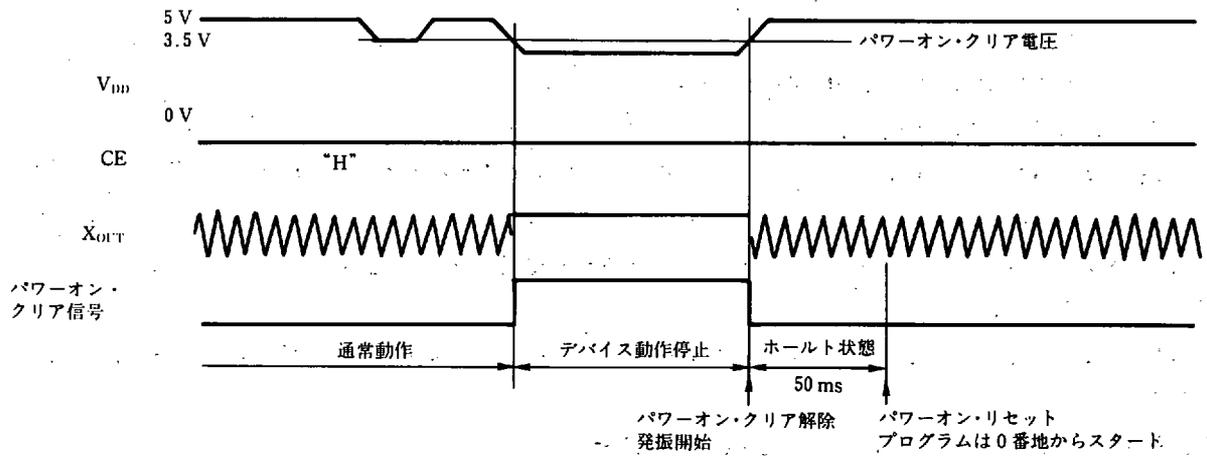
図 14-5 の (c) に動作を示します。

図 14-5 の (c) に示すように電源電圧 V_{DD} が 0 V から 3.5 V に立ち上がるまでパワーオン・クリア信号が出力されます。

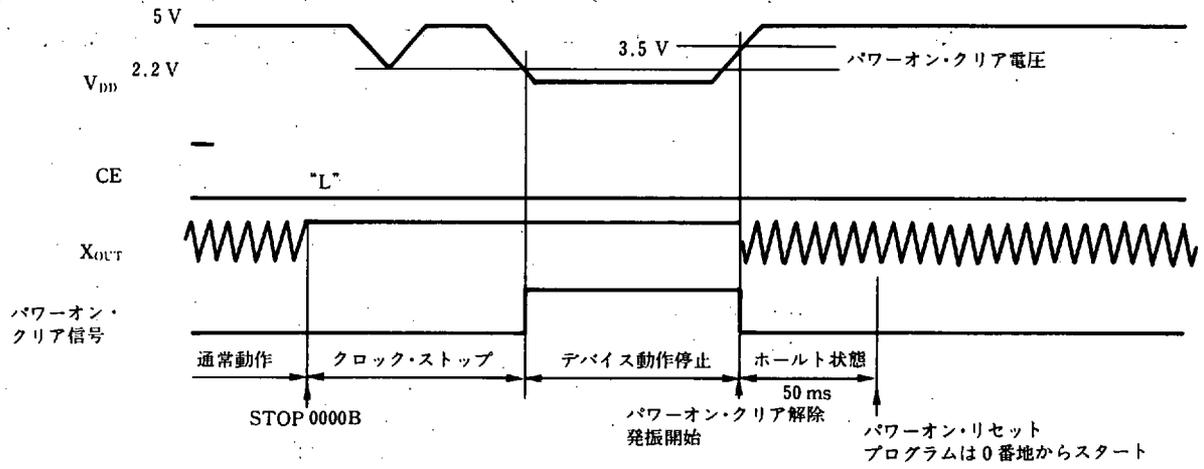
電源電圧 V_{DD} がパワーオン・クリア電圧を越えると水晶発振回路が動作を開始し、50 ms のホールド後にプログラムが 0000H 番地からスタートします。

図 14-5 パワーオン・リセットと電源電圧 V_{DD}

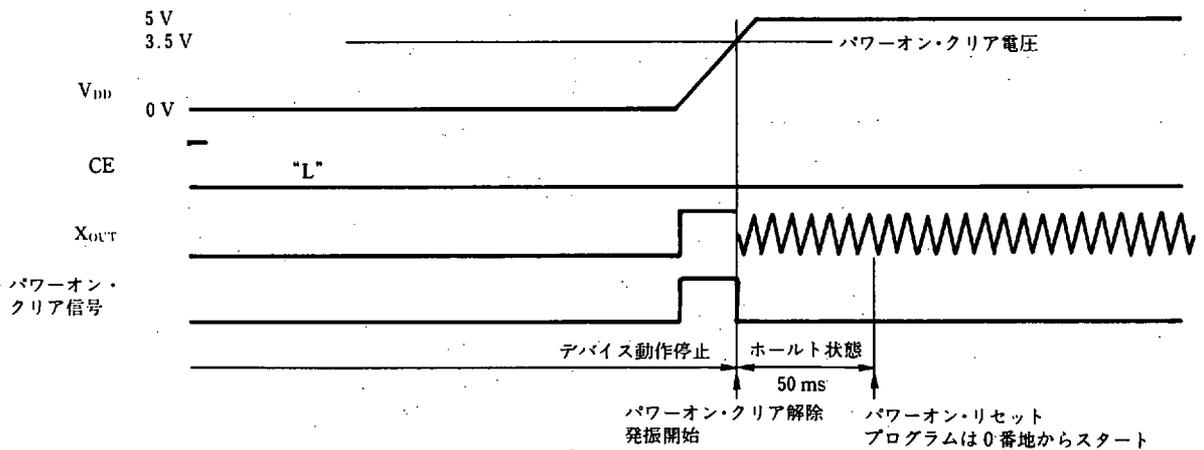
(a) 通常動作時 (ホールド状態も含む)



(b) クロック・ストップ時



(c) 電源電圧 V_{DD} を 0V から立ち上げたとき



14.5 CEリセットとパワーオン・リセットの関係

最初の電源電圧投入時等は、パワーオン・リセットとCEリセットが同時にかかる可能性があります。

このときのリセット動作を14.5.1-14.5.3に示します。

また、14.5.4に電源電圧立ち上げ時の注意を示します。

14.5.1 V_{DD} 端子とCE端子が同時に立ち上がったとき

図14-6の(a)に動作を示します。

このときは、パワーオン・リセットによりプログラムが0000H番地からスタートします。

14.5.2 パワーオン・リセットの強制ホールド中にCE端子が立ち上がったとき

図14-6の(b)に動作を示します。

このときは、14.5.1と同様にパワーオン・リセットによりプログラムが0000H番地からスタートします。

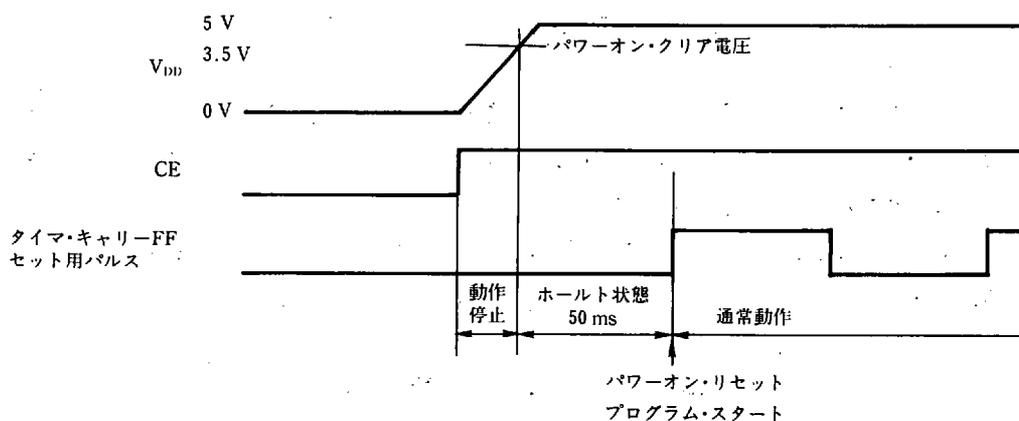
14.5.3 パワーオン・リセット後にCE端子が立ち上がったとき

図14-6の(c)に動作を示します。

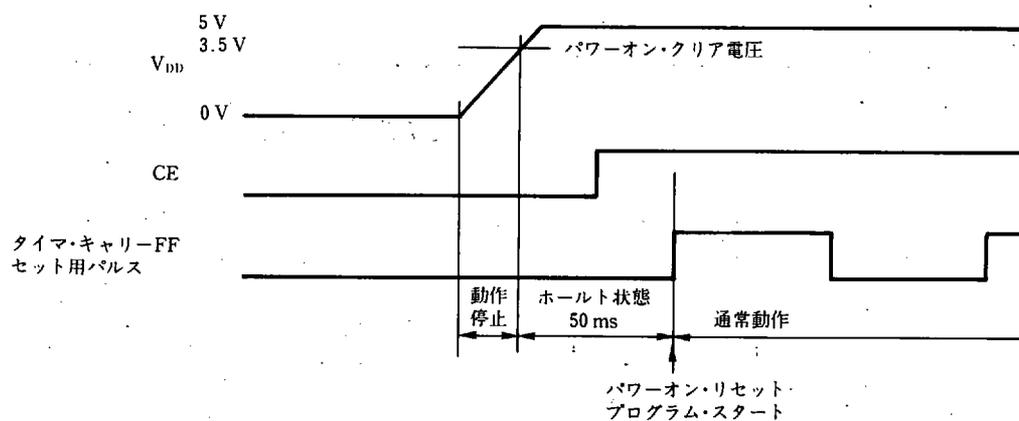
このときは、パワーオン・リセットによりプログラムが0000H番地からスタートし、さらにCEリセットにより次のタイマ・キャリーFFセット用信号の立ち上がりでプログラムが0000H番地から再度スタートします。

図 14-6 パワーオン・リセットと CE リセットの関係

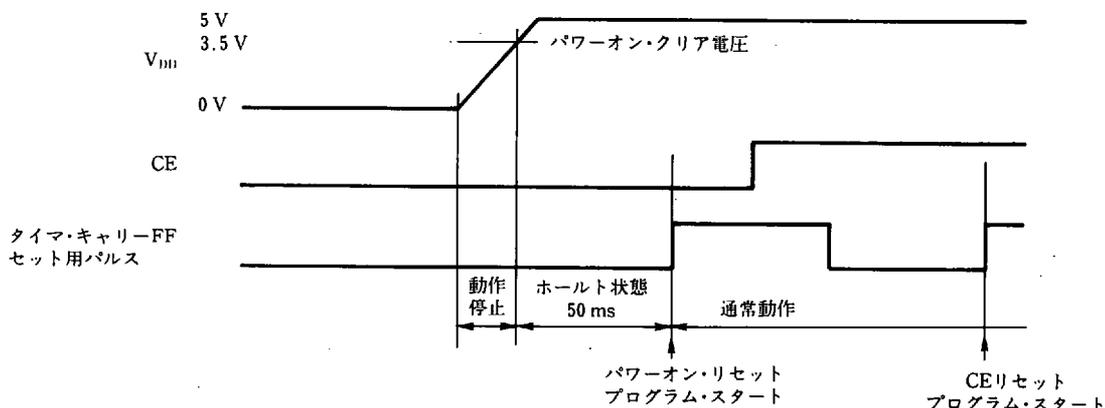
(a) V_{DD} と CE 端子が同時に立ち上がったとき



(b) ホールド状態中に CE 端子が立ち上がったとき



(c) パワーオン・リセット後に CE 端子が立ち上がったとき



14.5.4 電源電圧立ち上げ時の注意

電源電圧を立ち上げるときは、以下の(1)および(2)に示す注意が必要です。

(1) 電源電圧をパワーオン・クリア電圧以下から立ち上げるとき

電源電圧を立ち上げるときは、必ず一度3.5 V以上に立ち上げる必要があります。

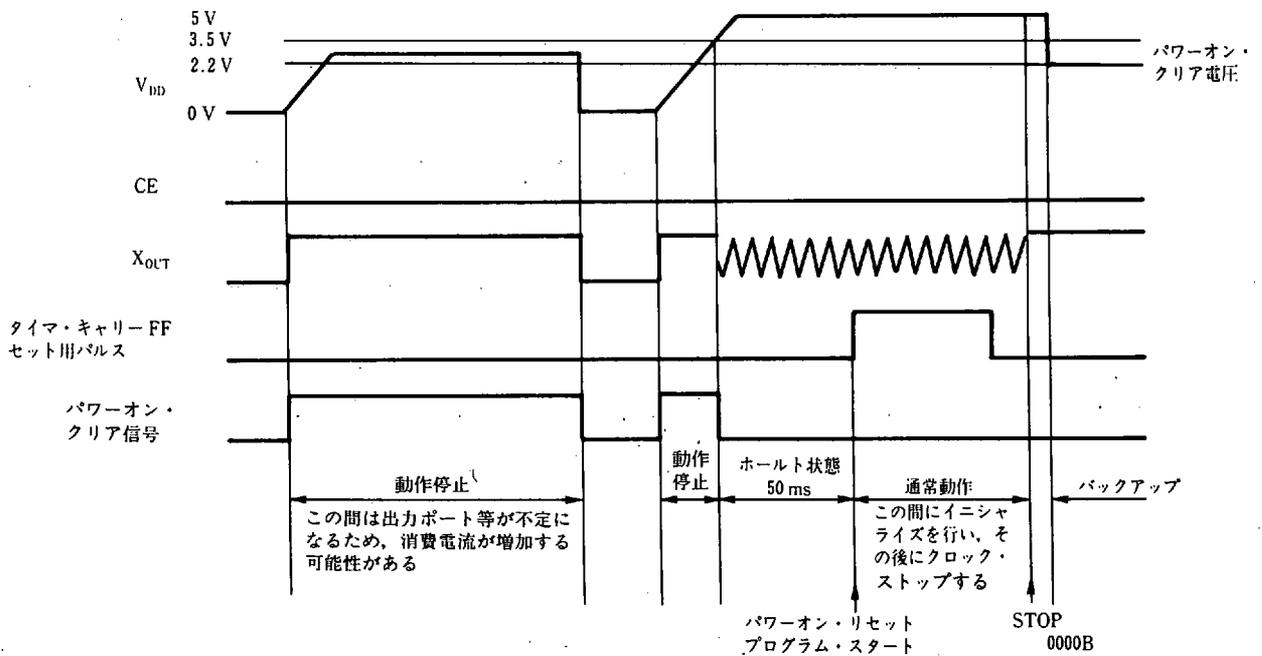
これを図 14-7 に示します。

図 14-7 に示すように、たとえばクロック・ストップを使用して $V_{DD}=2.2\text{ V}$ でバックアップするようなプログラムにおいて、 V_{DD} 投入時に 3.5 V 未満の電圧しか印加されなかったとすると、パワーオン・クリア信号が出力されたままになりプログラムは動作しません。

このとき、デバイスの出力ポート等は不定な値を出力するため、場合により消費電流が増加します。

すなわち、電池等によるバックアップ時はバックアップ時間が大幅に減少してしまいます。

図 14-7 V_{DD} 立ち上げ時の注意



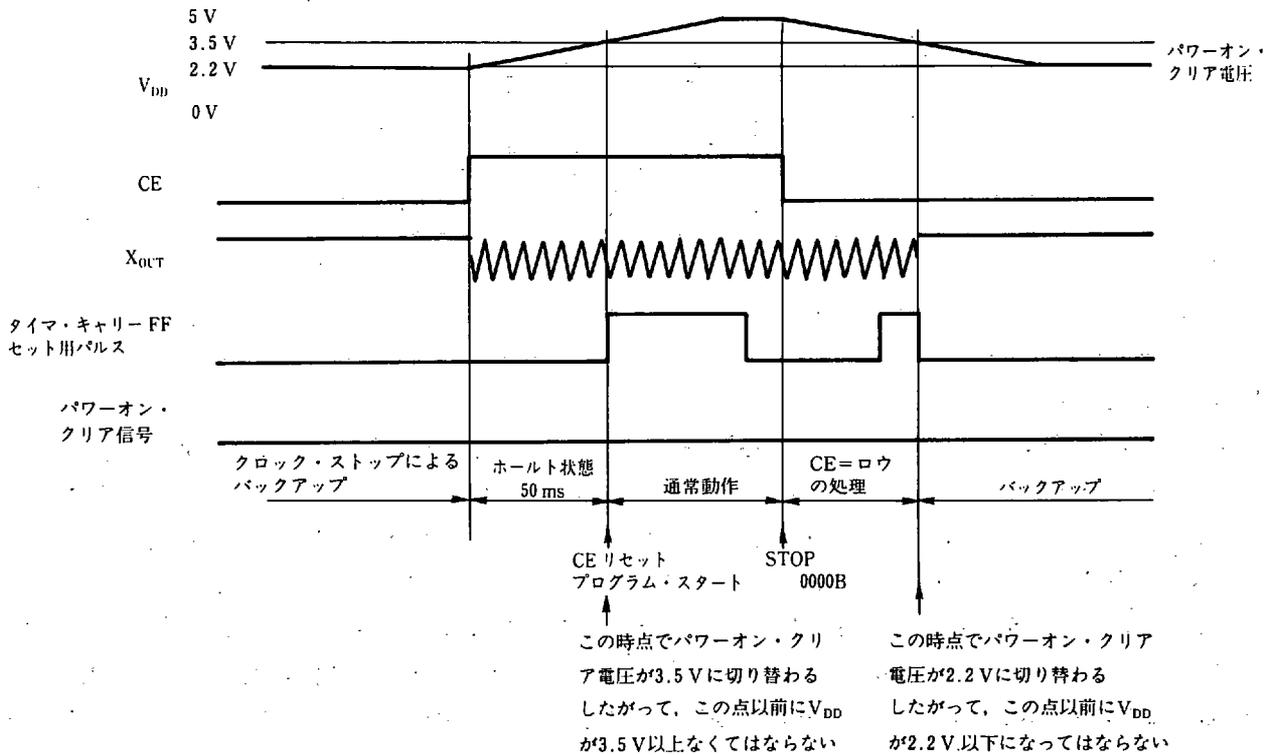
(2) クロック・ストップ状態からの復帰時

クロック・ストップを使用して電源電圧を2.2Vでバックアップしているときに、バックアップ状態から復帰するときは、CE端子がハイ・レベルになってから50ms以内にV_{DD}を3.5V以上に立ち上げる必要があります。

図14-8に示すようにクロック・ストップ状態からの復帰はCEリセットにより行いますが、CE端子が立ち上がったから50ms後にパワーオン・クリア電圧が3.5Vに切り替わるため、この時点でV_{DD}が3.5V以上ないとパワーオン・リセットがかかってしまいます。

また、V_{DD}を立ち下げるときも同様に注意が必要です。

図14-8 クロック・ストップ状態からの復帰

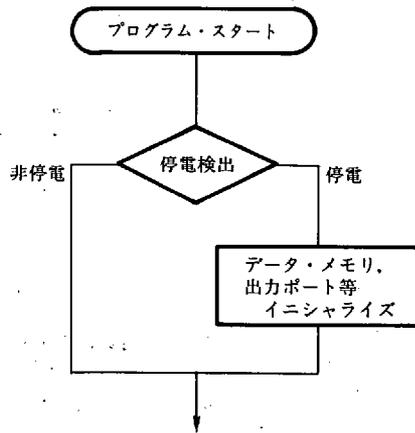


14.6 停電検出

停電検出は図 14-9 に示すようにデバイスにリセットがかかったときに電源電圧 V_{DD} の投入であったかまたは CE 端子によるリセットであったかを判断するために使用します。

電源電圧投入時は、データ・メモリや出力ポート等の内容が“不定”になっているため、停電検出によりこれらの内容を初期化します。

図 14-9 停電検出フロー・チャート



14.6.1 停電検出回路

停電検出回路は図 14-1 に示したように電圧検出回路と、電圧検出回路の出力（パワーオン・クリア信号）によりリセットされるタイマ・キャリー・ディスエーブル・フリップフロップおよびタイマ・キャリー FF で構成されます。

タイマ・キャリー・ディスエーブル FF は、パワーオン・クリア信号によりセット (1) され、BTM0CY フラグ (アドレス 17H のビット b_0) を読み込む命令が実行されるとリセット (0) されます。

タイマ・キャリー・ディスエーブル FF がセット (1) されているときは、BTM0CY フラグはセット (1) されません。すなわち、パワーオン・クリア信号が出力される（パワーオン・リセット時）と、BTM0CY フラグはリセットされた状態でプログラムがスタートし、以後 BTM0CY フラグを読み込む命令が実行されるまでセット禁止状態になります。

一度 BTM0CY フラグを読み込む命令が実行されれば、以後タイマ・キャリー FF セット用パルスの立ち上がりごとに BTM0CY フラグがセットされます。したがってデバイスにリセットがかかったときに BTM0CY フラグの内容を検出し、リセット (0) されていればパワーオン・リセット（停電）であり、セット (1) されていれば CE リセット（非停電）であることを判断できます。

停電を検出できる電圧はパワーオン・リセットのかかる電圧と同じになるため、クロック発振時で約 3.5V、クロック・ストップ時で約 2.2V になります。

図 14-10 に BTM0CY フラグの状態遷移を示します。

また、図 14-11 に図 14-10 のタイミング・チャートおよび BTM0CY フラグの動作を示します。

図 14 - 10 BTM0CY フラグの状態遷移

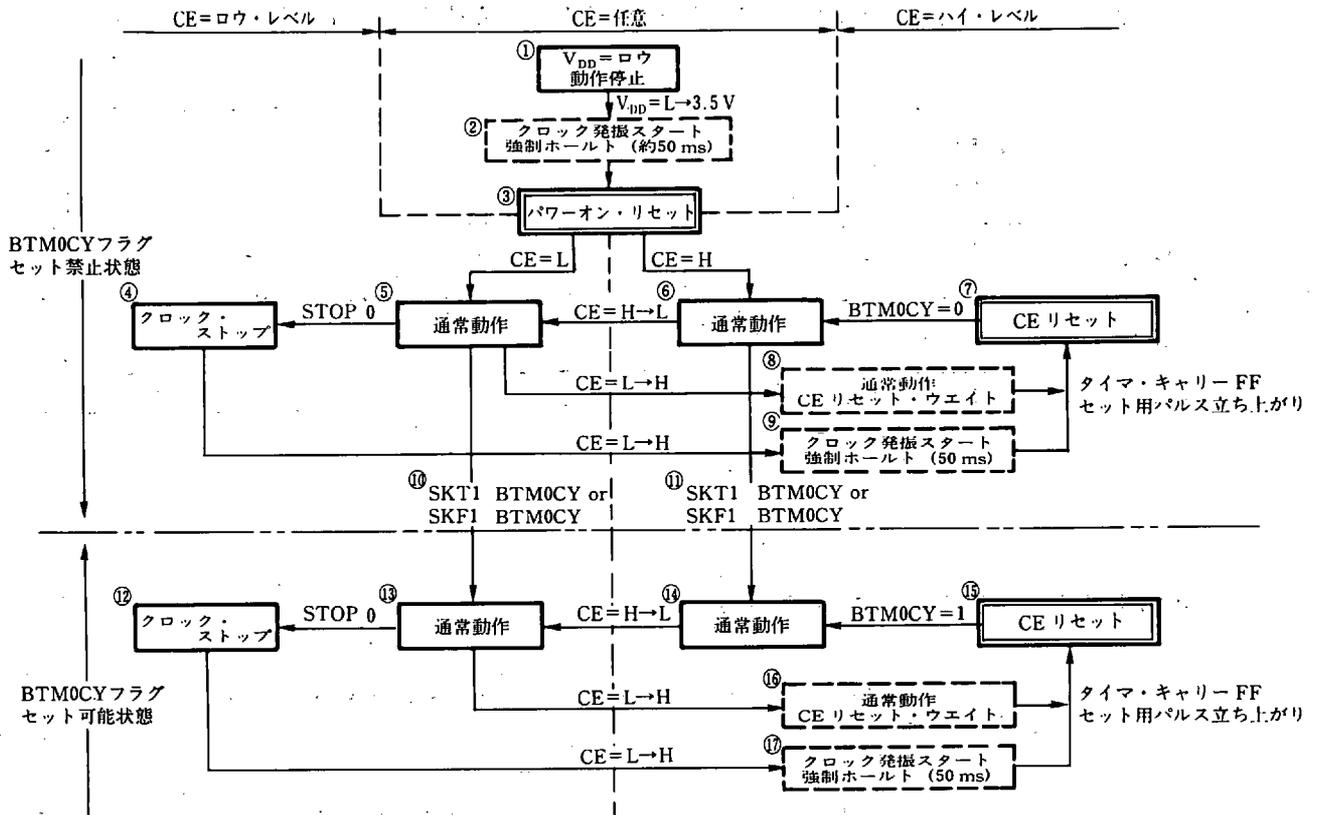
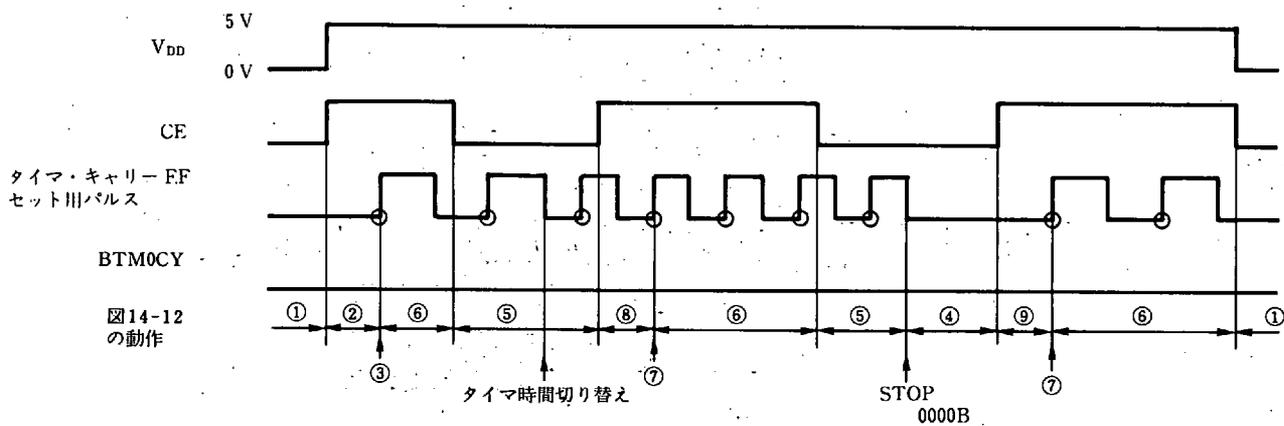
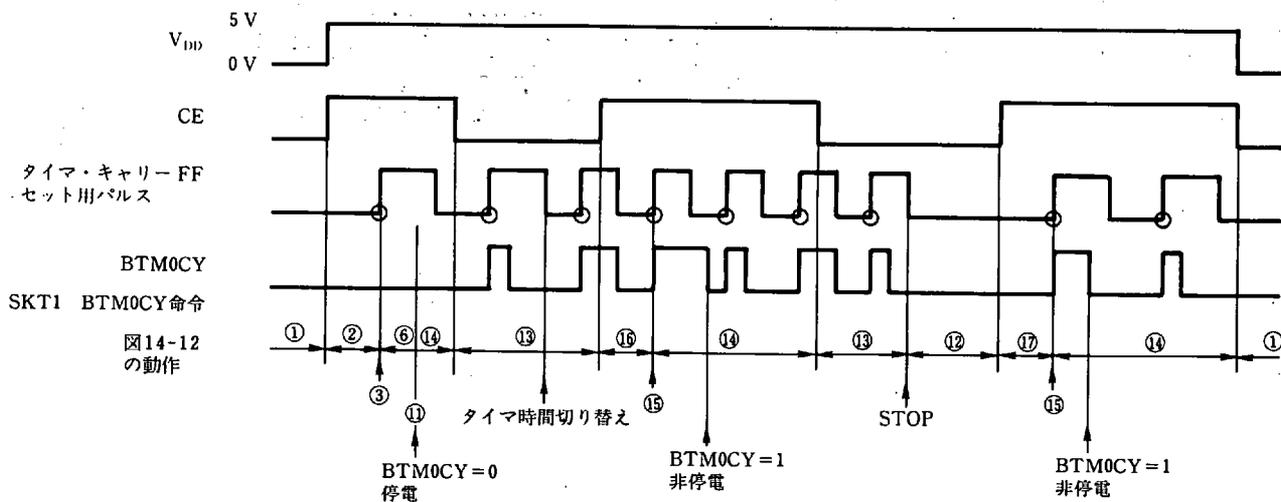


図 14-11 BTM0CY フラグの動作

(a) BTM0CY フラグを一度も検出しないとき (SKT1 BTM0CY または SKFI BTM0CY を実行しない)



(b) BTM0CY フラグによる停電検出を行うとき



14.6.2 BTM0CY フラグによる停電検出時の注意

BTM0CY フラグで時計のカウント等を行うときは以下の点に注意が必要です。

(1) 時計の更新

タイマ・キャリー FF を使用して時計等のプログラムを作成するときは、停電検出後に時計を更新する必要があります。

これは停電検出時に BTM0CY フラグを読み込むために BTM0CY フラグがリセット (0) され、1 回分時計のカウントを取り逃がすためです。

(2) 時計更新処理時間

時計の更新を行うときは、その処理を次のタイマ・キャリー FF セット用パルスが立ち上がるまでに終了させる必要があります。

これは、時計更新処理中に CE 端子がハイ・レベルに立ち上がると時計更新処理を最後まで実行しないで CE リセットがかかってしまうためです。

上記(1)および(2)については“12.4.2 CE リセット時におけるタイマ・キャリー FF の補正”の項を参照してください。

また、停電時の処理を行うときは以下の点に注意が必要です。

(3) 停電検出を行うタイミング

BTM0CY フラグで時計のカウントなどを行うときは、停電検出のための BTM0CY フラグの読み込みをプログラムが 0000H 番地からスタートしてから次のタイマ・キャリー FF セット用パルスが立ち上がる時間以内に行う必要があります。

これは、タイマ・キャリー FF セット時間をたとえば 5 ms に設定し、停電検出をプログラムがスタートしてから 6 ms 後に行うとすると、1 回分 BTM0CY フラグを取り逃がしてしまうためです。

“12.4.2 CE リセット時におけるタイマ・キャリー FF の補正”の項を参照してください。

また、下記の例に示すように、停電検出およびイニシャル処理は、タイマ・キャリー FF セット時間以内で行う必要があります。

これは停電処理およびイニシャル処理中に CE 端子が立ち上がり CE セットがかかると、これらの処理が途中で中断されてしまい、不具合が発生する可能性があるためです。

また、イニシャル処理でタイマ・キャリー FF セット時間を変更する場合は変更する命令はイニシャル処理の一番最後に 1 命令で行う必要があります。

これも、下記の例に示すようにイニシャル処理よりさきにタイマ・キャリー FF のセット時間が切り替わると CE リセットによりイニシャル処理が最後まで実行できない可能性があるためです。

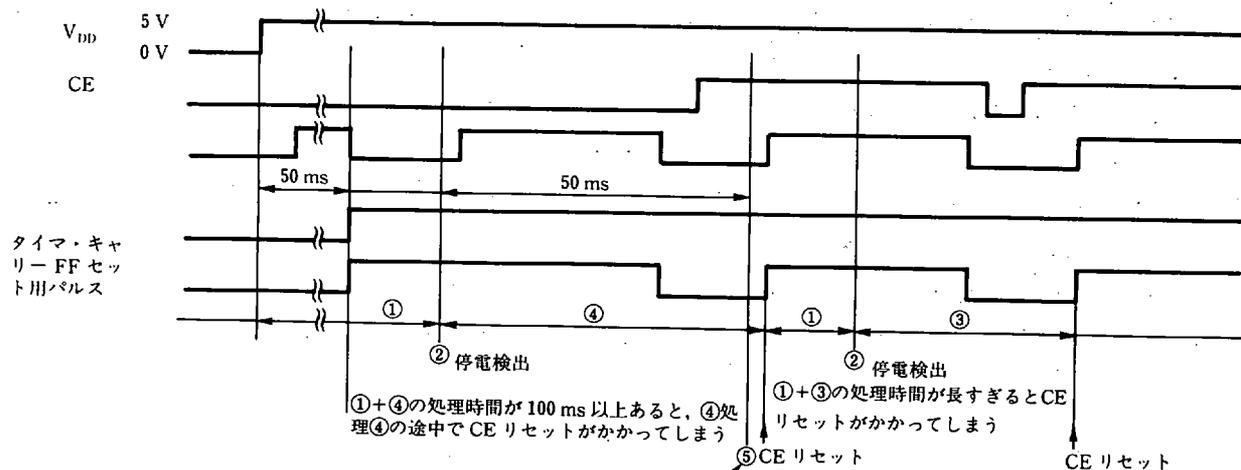
例

プログラム例

```

START:                                ;プログラム・アドレス 0000H
; ①
    リセット時の処理 ;
; ②
    SKT1 BTM0CY      ; 停電検出
    BR  INITIAL
BACKUP:
; ③
    時計更新
    BR  MAIN
INITIAL:
; ④
    イニシャル処理
; ⑤
    INITFLG NOT BTM0ZX, NOT BTM0CK2, NOT BTM0CK1, BTM0CK0
                                ; 組み込みマクロ
                                ; タイマ・キャリー FF セット時間を 5 ms に設定
MAIN:
    SKT1 BTM0CY
    BR  MAIN
    時計更新
    
```

動作例



タイマ・キャリー FF セット時間を切り替えるタイミングにより、すぐに CE リセットがかかってしまう場合がある。したがって、④より先に⑤を実行すると停電処理④が最後まで実行されないような不具合が生じる。

15. 汎用ポート

汎用ポートは、外部回路へのハイ・レベル、ロウ・レベルまたはフローティング信号の出力および外部回路のハイ・レベル、ロウ・レベル信号の読み込みを行います。

15.1 汎用ポートの構成と分類

図 15-1 に汎用ポートのブロック図を示します。

また、表 15-1 に汎用ポートの分類を示します。

図 15-1 に示すように、汎用ポートはデータ・メモリの各バンクのアドレス 70H 番地から 73H 番地 (ポート・レジスタ) によりデータ設定を行う Port0A (P0A) - Port1C (P1C) から構成され、BANK0 と BANK2 は同一のポート・レジスタが割り当てられています。

各ポートは、それぞれ各汎用ポート端子、たとえば Port0A は P0A₃ 端子 - P0A₀ 端子から構成されています。

また、表 15-1 に示すように汎用ポートは入出力兼用ポート (入出力ポート)、入力専用ポート (入力ポート) および出力専用ポート (出力ポート) に分類されます。

入出力ポートは 1 ビット単位 (1 端子単位) で入出力を指定できるビット I/O ポートと 3 ビット単位 (3 端子単位) で入出力を指定できるグループ I/O ポートに分類されます。

図 15-1 汎用ポートのブロック図

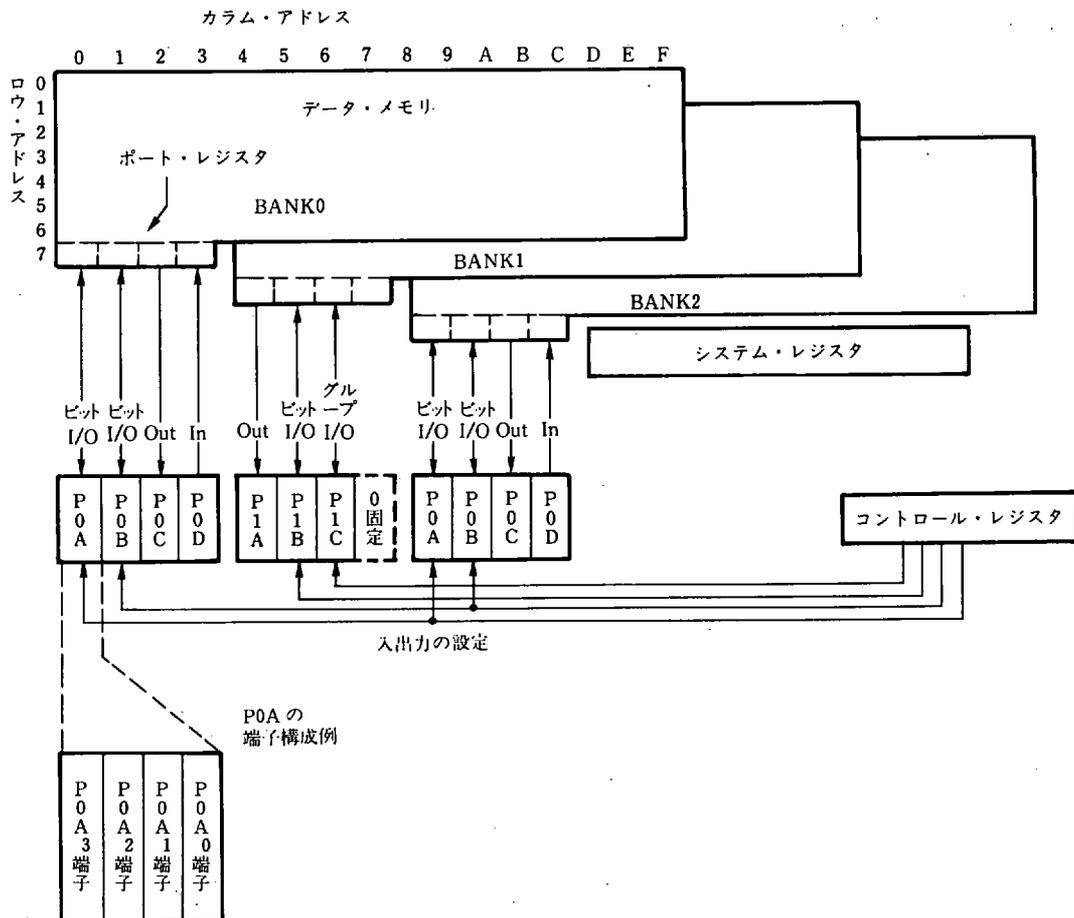


表 15-1 汎用ポートの分類

汎用ポートの分類		対象ポート	データ設定方法
汎 用 ポ ー ト	入出力兼用ポート	ビット I/O Port0A Port0B Port1B	ポート・レジスタ
		グループ I/O Port1C	ポート・レジスタ
	入力専用ポート	Port0D	ポート・レジスタ
	出力専用ポート	Port0C Port1A	ポート・レジスタ

15.2 汎用ポートの機能概略

汎用出力ポートまたは出力ポートに設定されている汎用入出力ポートは、ポート・レジスタにデータを設定することにより、対応する各端子からハイ・レベルまたはロウ・レベルを出力します。

また、汎用入力ポートまたは入力ポートに設定されている汎用入出力ポートは、ポート・レジスタの内容を読み込むことにより、対応する各端子に加えられている入力信号レベルを検出できます。

汎用入出力ポートの入力ポートおよび出力ポートの切り替えは各ポートに対応するコントロール・レジスタにより行います。

すなわち、プログラムで入出力を切り替えることが可能です。

P0A - P0D, P1A - P1C は、パワーオン・リセット時に汎用ポートに設定されるため、他の周辺ハードウェアを兼用している端子は対応するコントロール・レジスタにより独立して設定します。

以下の 15.2.1 - 15.2.4 にポート・レジスタの機能と、各ポートの機能概略を示します。

15.2.1 汎用ポート・データ・レジスタ (ポート・レジスタ)

ポート・レジスタは各汎用ポートの出力データの設定や入力データの読み込みを行います。

ポート・レジスタは、データ・メモリ上に配置されているため、すべてのデータ・メモリ操作命令で操作することができます。

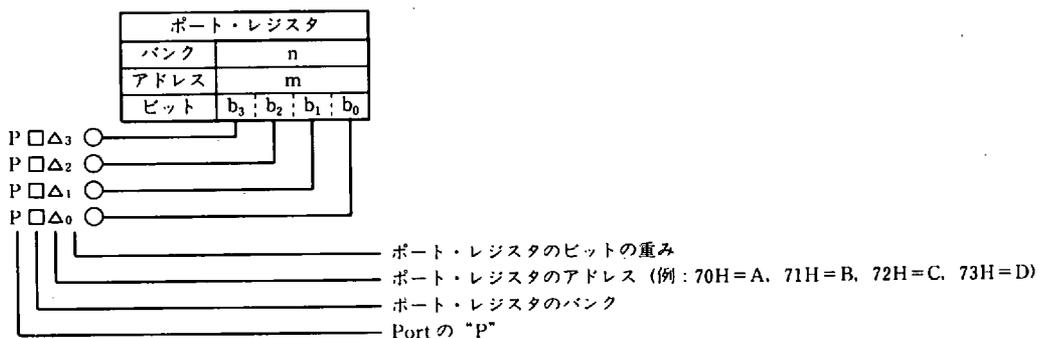
図 15-2 にポート・レジスタと対応する各端子の関係を示します。

汎用出力ポートに設定されている端子に対応するポート・レジスタにデータを設定することにより各端子の出力を設定します。

また、汎用入力ポートに設定されている端子に対応するポート・レジスタを読み込むことにより各端子の入力状態を検出します。

各ポート (各端子) とポート・レジスタの関係を表 15-2 に示します。

図 15-2 ポート・レジスタと各端子の関係



ポート・レジスタにはアセンブラ上で予約語が定義されています。

この予約語はフラグ (ビット) 単位で定義されているため、アセンブラ組み込みマクロ命令を使用できます。

ポート・レジスタにはデータ・メモリ型の予約語は定義されていないので注意してください。

15.2.2 汎用入出力ポート (P0A, P0B, P1B, PIC)

P0A, P0B, P1B および PIC の入出力の切り替えは、それぞれ P0A ビット I/O セレクト・レジスタ (RF アドレス 37H 番地), P0B ビット I/O セレクト・レジスタ (RF アドレス 36H 番地), P1B ビット I/O セレクト・レジスタ (RF アドレス 35H 番地), PIC グループ I/O セレクト・レジスタ (RF アドレス 27H 番地) により行います。

P0A, P0B, P1B および PIC の入出力データの設定は、それぞれポート・レジスタの P0A (データ・メモリ・アドレス: BANK0 または BANK2 の 70H 番地), P0B (データ・メモリ・アドレス: BANK0 または BANK2 の 71H 番地), P1B (データ・メモリ・アドレス: BANK1 の 71H 番地), PIC (データ・メモリ・アドレス: BANK1 の 72H 番地) により行います。

表 15-2 を参照してください。

詳細は 15.3 項を参照してください。

15.2.3 汎用入力ポート (P0D)

P0D の入力データの読み込みは、ポート・レジスタの P0D (データ・メモリ・アドレス: BANK0 または BANK2 の 73H 番地) により行います。

表 15-2 を参照してください。

詳細は 15.4 項を参照してください。

15.2.4 汎用出力ポート (P0C, P1A)

(1) P0C, P1A

P0C および P1A の出力データの設定は、それぞれポート・レジスタの P0C (データ・メモリ・アドレス: BANK0 または BANK2 の 72H 番地), P1A (データ・メモリ・アドレス: BANK1 の 70H 番地) により行います。

表 15-2 を参照してください。

詳細は 15.5 項を参照してください。

表 15-2 各ポート（端子）とポート・レジスタの関係

ポ ー ト	端 子		データ設定方法				
	記 号	入 出 力	ポート・レジスタ（データ・メモリ）				
			バンク	アドレス	記 号	ビット記号 (予約語)	
Port0A (P0A)	P0A ₃	入出力 (ビット I/O)	BANK0 BANK2	70H	P0A	b ₃	P0A3
	P0A ₂					b ₂	P0A2
	P0A ₁					b ₁	P0A1
	P0A ₀					b ₀	P0A0
Port0B (P0B)	P0B ₃	入出力 (ビット I/O)		71H	P0B	b ₃	P0B3
	P0B ₂					b ₂	P0B2
	P0B ₁					b ₁	P0B1
	P0B ₀					b ₀	P0B0
Port0C (P0C)	P0C ₃	出 力		72H	P0C	b ₃	P0C3
	P0C ₂					b ₂	P0C2
	P0C ₁					b ₁	P0C1
	P0C ₀					b ₀	P0C0
Port0D (P0D)	P0D ₃	入 力	73H	P0D	b ₃	P0D3	
	P0D ₂				b ₂	P0D2	
	P0D ₁				b ₁	P0D1	
	P0D ₀				b ₀	P0D0	
Port1A (P1A)	P1A ₃	出 力	BANK1	70H	P1A	b ₃	P1A3
	P1A ₂					b ₂	P1A2
	P1A ₁					b ₁	P1A1
	P1A ₀					b ₀	P1A0
Port1B (P1B)	P1B ₃	入出力 (ビット I/O)		71H	P1B	b ₃	P1B3
	P1B ₂					b ₂	P1B2
	P1B ₁					b ₁	P1B1
	P1B ₀					b ₀	P1B0
Port1C (P1C)	P1C ₃	入出力 (グループ I/O)	72H	P1C	b ₃	P1C3	
	P1C ₂				b ₂	P1C2	
	P1C ₁				b ₁	P1C1	
	対象端子なし				b ₀	注	

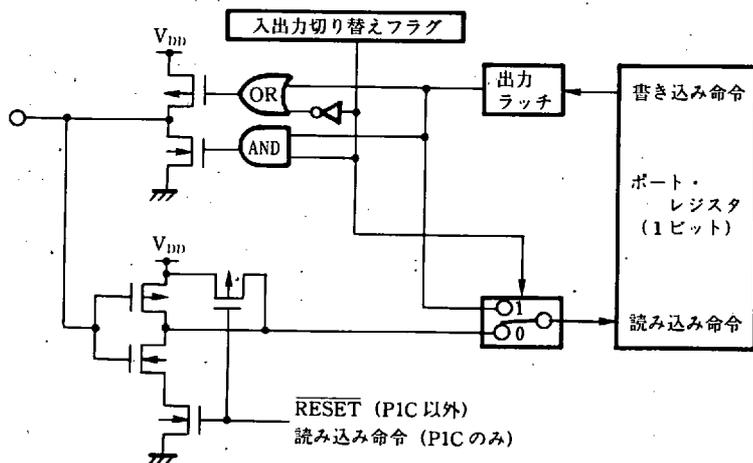
注 72H の b₀には何も割り当てられていません。b₀を読み出すと常に0が読み出されます。

15.3 汎用入出力ポート (P0A, P0B, P1B, P1C)

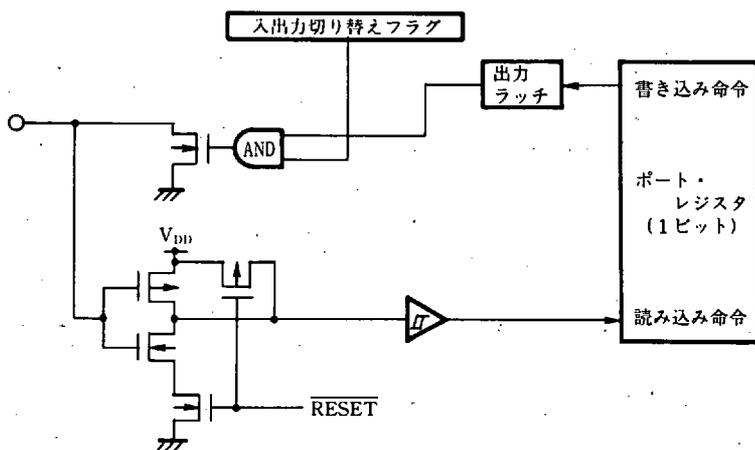
15.3.1 入出力ポートの構成

以下の(1)-(3)に入出力ポートの構成を示します。

- (1) P0A (P0A₃, P0A₂ 端子)
- P0B (P0B₃, P0B₂, P0B₁, P0B₀ 端子)
- P1B (P1B₃, P1B₂, P1B₁, P1B₀ 端子)
- P1C (PIC₃, PIC₂, PIC₁ 端子)



- (2) P0A (P0A₁, P0A₀ 端子)



15.3.2 入出力ポートの使用法

入出力ポートは、コントロール・レジスタのP0A, P0B, P1BおよびP1Cの各I/Oセレクト・レジスタにより入力および出力の設定を行います。

ビットI/Oポート (P0A, P0B, P1B) は1ビット単位 (1端子単位) で入出力を設定でき、グループI/Oポート (P0C) は3ビット単位 (3端子単位) で入出力を設定できます。

出力データの設定および入力データの読み込みは、それぞれ対応するポート・レジスタにデータを書き込むかデータを読み込む命令を実行することにより行います。

15.3.3項に各ポートのI/Oセレクト・レジスタを説明します。

15.3.4および15.3.5項に入力および出力ポートとして使用する方法を示します。

15.3.4 入出力ポート (P0A, P0B, P1B, P1C) を入力ポートとして使用する場合

各ポートの I/O セレクト・レジスタにより入力として使用する端子を選択します。

このとき、PIC は、3 ビット単位 (3 本単位) でのみ入出力の設定ができます。

入力ポートに指定された端子はフローティング (Hi-Z) 状態になり、外部信号の入力待ちになります。

入力データの読み込みは、各端子に対応するポート・レジスタの内容を読み込む命令、たとえば "SKT" 命令などを実行することにより行えます。

ポート・レジスタは、各端子にハイ・レベルが入力されているときは "1" が読み込まれ、ロウ・レベルが入力されているときは "0" が読み込まれます。

入力ポートに指定されているポート・レジスタに書き込み命令、たとえば "MOV" 命令などを実行すると、出力ラッチの内容が書き換わります。

15.3.5 入出力ポート (P0A, P0B, P1B, P1C) を出力ポートとして使用する場合

各ポートの I/O セレクト・レジスタにより出力として使用する端子を選択します。

このとき、PIC は、3 ビット単位 (3 本単位) でのみ入出力の設定ができます。

出力ポートに指定された端子は出力ラッチの内容を各端子から出力します。

出力データの設定は、各端子に対応するポート・レジスタの内容に書き込む命令、たとえば "MOV" 命令などを実行することにより行えます。

各端子にハイ・レベルを出力するときは "1" を書き込み、ロウ・レベルを出力するときは "0" を書き込みます。

また、入力ポートに指定することによりフローティング (Hi-Z) 状態にすることができます。

出力ポートに指定されているポート・レジスタを読み込む命令、たとえば "SKT" 命令などを実行すると、出力ラッチの内容が読み込まれます。

ただし、P0A₁ 端子および P0A₀ 端子の 2 本は端子の状態がそのまま読み込まれるため、出力ラッチの内容と読み込んだ内容が異なる場合があるため注意が必要です。

15.3.6 項を参照してください。

15.3.6 入出力ポート (P0A₁ および P0A₀ 端子) 使用時の注意

以下の例に示すように、P0A₁ 端子と P0A₀ 端子を出力として使用する場合は、出力ラッチの内容が書き換わる場合があるため注意が必要です。

例

```
INITFLG  NOT P0ABIO3, NOT P0ABIO2, P0ABIO1, P0ABIO0
          ; P0A1, P0A0 端子を出力に設定

INITFLG  NOT P0A3, NOT P0A2, P0A1, P0A0
; ①          ; P0A1, P0A0 端子にハイ・レベルを出力

CLR1     P0A1      ; P0A1 端子にロウ・レベルを出力
; マクロ展開

AND      .MF.P0A1 SHR 4, #.DF. (NOT P0A1 AND 0FH)
```

上記 ① の命令を実行したときに、たまたま P0A₀ 端子が外部でロウ・レベルに引っ張られていると、“CLR1” 命令で P0A₀ 端子の出力ラッチの内容を“0”に書き換えてしまいます。

15.3.7 入出力ポート (P0A, P0B, P1B, PIC) のリセット時の状態

(1) パワーオン・リセット時

すべて入力ポートに指定されます。

出力ラッチの内容は“不定”であるため、出力ポートに切り替える前には必要に応じてプログラムでイニシャライズする必要があります。

(2) CE リセット時

すべて入力ポートに指定されます。

出力ラッチの内容は保持されます。

(3) クロック・ストップ時

すべて入力ポートに指定されます。

出力ラッチの内容は保持されます。

PIC 以外の入出力ポートは、クロック・ストップ時の $\overline{\text{RESET}}$ 信号出力により 15.3.1 に示したように入力バッファのノイズによる消費電流の増加を防いでいます。

(4) ホールト状態中

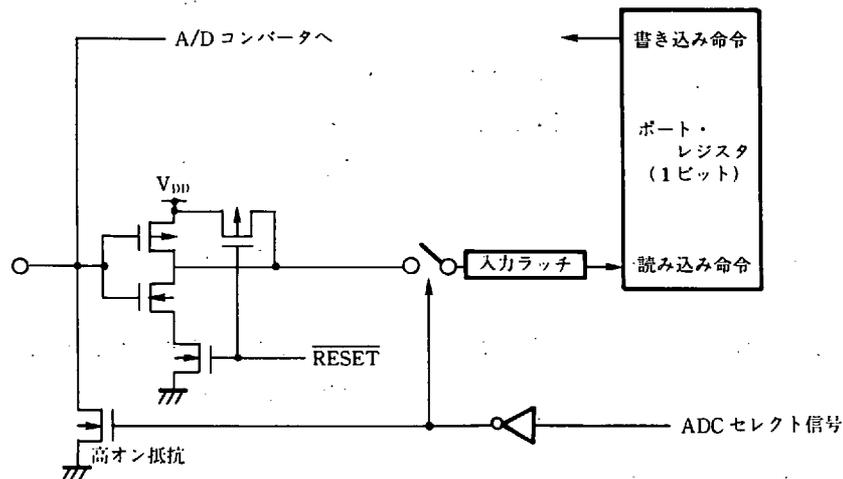
以前の状態を保持します。

15.4 汎用入力ポート (P0D)

15.4.1 入力ポートの構成

以下に入力ポートの構成を示します。

(1) P0D (P0D₃, P0D₂, P0D₁, P0D₀)



15.4.2 入力ポート (P0D) の使用例

入力データの読み込みは、各端子に対応するポート・レジスタの内容を読み込む命令、たとえば“SKT”命令などを実行することにより行えます。

ポート・レジスタは、各端子にハイ・レベルが入力されているときは“1”が読み込まれ、ロウ・レベルが入力されているときは“0”が読み込まれます。

ポート・レジスタに書き込み命令、たとえば“MOV”命令などを実行しても何も変化しません。

15.4.3 入力ポート (P0D) 使用時の注意

P0Dは、汎用ポートとして使用する場合は内部でプルダウンされています。

15.4.4 入力ポート (P0D) のリセット時の状態

(1) パワーオン・リセット時

すべて汎用入力ポートに指定されます。

(2) CE リセット時

すべて汎用入力ポートに指定されます。

(3) クロック・ストップ時

すべて汎用入力ポートに指定されます。

クロック・ストップ時は RESET 信号が出力されるため、15.4.1 に示したように入力バッファのノイズによる消費電流の増加を防いでいます。

P0D は、内部でプルダウンされます。

(4) ホールト状態中

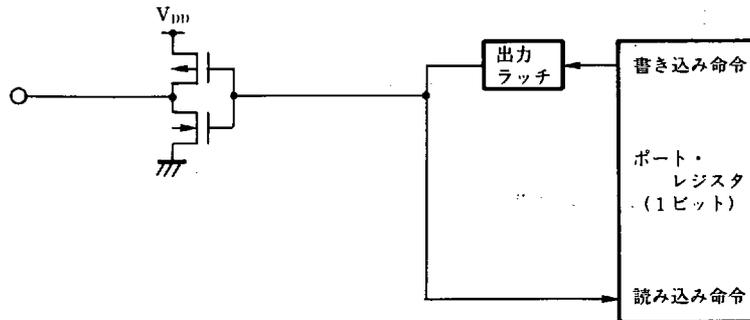
以前の状態を保持します。

15.5 汎用出力ポート (P0C, P1A)

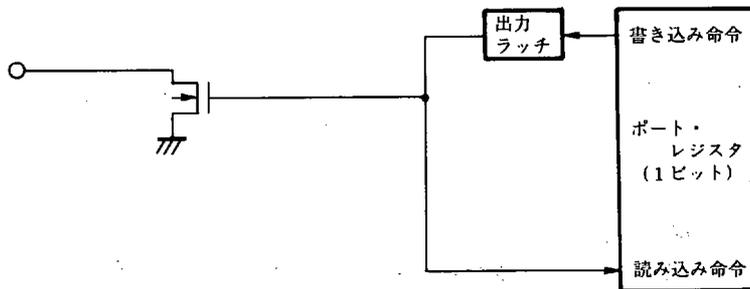
15.5.1 出力ポート (P0C, P1A) の構成

以下の(1)および(2)に出力ポートの構成を示します。

(1) P0C (P0C₃, P0C₂, P0C₁, P0C₀ 端子)



(2) P1A (P1A₃, P1A₂, P1A₁, P1A₀ 端子)



15.5.2 出力ポート (P0C, P1A) の使用例

出力ポートは出力ラッチの内容を各端子から出力します。

出力データの設定は、各端子に対応するポート・レジスタの内容に書き込む命令、たとえば“MOV”命令などを実行することにより行います。

各端子にハイ・レベルを出力するときは“1”を書き込み、ロウ・レベルを出力するときは“0”を書き込みます。

ただし、P1A₃, P1A₂, P1A₁, P1A₀ 端子は N-ch オープン・ドレイン出力のためハイ・レベル出力時は、フローティング (Hi-Z) になります。

ポート・レジスタを読み込む命令、たとえば“SKT”命令などを実行すると、出力ラッチの内容が読み込まれます。

15.5.3 出力ポート (P0C, P1A) のリセット時の状態

(1) パワーオン・リセット時

出力ラッチの内容を出力します。

出力ラッチの内容は“不定”であるため、一定期間 (プログラムでイニシャライズされるまで) “不定”な値が出力されます。

(2) CE リセット時

出力ラッチの内容を出力します。

出力ラッチの内容は保持されるため、CE リセット時は出力データは変化しません。

(3) クロック・ストップ時

出力ラッチの内容を出力します。

出力ラッチの内容は保持されるため、クロック・ストップ時は出力データは変化しません。

したがって、必要に応じてプログラムでイニシャライズしてください。

(4) ホールト状態中

出力ラッチの内容を出力します。

出力ラッチの内容は保持されるため、ホールト状態中は出力データは変化しません。

16. シリアル・インタフェース

μPD17062 は外部とのデータ通信用にチャンネル 0 (CH0), チャンネル 1 (CH1) の 2 組のシリアル・インタフェース用端子を有しています。

CH0 用の端子は, SDA, SCL の 2 本で構成されクロック同期方式の 2 線式シリアル入力またはシリアル出力, そして 2 線式バス^注の 3 つのモードで動作させることができます。SDA, SCL 端子は, シリアル・インタフェースとして使用しないときは, 汎用のポートとして使用することができます。

CH1 用の端子は $\overline{\text{SCK}}$, SO, SI の 3 本で構成され, 2 線式シリアル I/O 入力, シリアル I/O 出力および 3 線式シリアル I/O 入出力モードで動作させることができます。

なお, CH0 と CH1 を同時に動作させることはできません。CH0, CH1 のどちらを使用するかは SMODE (Serial Interface Mode Register) の SIO0CH フラグ (レジスタ・ファイル: 08H, b₃) で指定します。

また, ハードウェアでサポートしている 2 線式バス・モードは, シングル・マスタ用です。したがって調停 (アービトラージ) 機能はありませんので, ソフトウェアで対応してください。

注 2 線式バス・モードは, I²C バスとして使用できます。

表 16-1 シリアル・インタフェース用外部端子一覧

CH	端子名	機 能		
		2 線式バス・モード	シリアル I/O モード	ポート I/O 設定レジスタ
0	P0A ₀ /SDA	シリアル・データ入出力	シリアル・データ入出力	P0ABIO0
	P0A ₁ /SCL	シフト・クロック入出力	シフト・クロック入出力	P0ABIO1
1	P0A ₂ / $\overline{\text{SCK}}$	使用不可	シフト・クロック入出力	P0ABIO2
	P0A ₃ /SO		シリアル・データ出力	P0ABIO3
	P0B ₀ /SI		シリアル・データ入力	P0BBIO0

16.1 シリアル・インタフェース・モード・レジスタ

シリアル・インタフェース・モード・レジスタは, シリアル・インタフェースの動作モードを指定するレジスタで, 使用するチャンネル, プロトコル, クロック, 送受信を設定します。

シリアル・インタフェース・モード・レジスタはレジスタ・ファイル上の 08H 番地に配置されています。

なお, シリアル・インタフェース・モード・レジスタのすべてのフラグは, パワーオン・リセット時に “0” となります。

図 16-1 シリアル・インタフェース・モード・レジスタの構成

ビット位置	b ₃	b ₂	b ₁	b ₀
フラグ名称	SIO0CH	SB	SIO0MS	SIO0TX

表 16-2 CH0の動作モード一覧

シリアル・インタフェース・モード・レジスタ			ポート0A I/O指定		SDA 端子	SCL 端子	動作モード
SB	SIO0MS	SIO0TX	P0ABIO0	P0ABIO1			
0	0	0	0	0	SD-IN	CK-IN	シリアル I/O-SI, EXT-CLK
0	0	0	0	1	SD-IN	OUT-PORT	シリアル I/O-SI, INT-CLK(SOFT-CLK)
0	0	0	1	0	OUT-PORT	IN-PORT	IOOUT-PORT+IIN-PORT
0	0	0	1	1	OUT-PORT	OUT-PORT	2OUT-PORT
0	0	1	×	0	SD-OUT	CK-IN	シリアル I/O-SO, EXT-CLK
0	0	1	×	1	SD-OUT	OUT-PORT	シリアル I/O-SO, INT-CLK(SOFT-CLK)
0	1	0	0	×	SD-IN	CK-OUT	シリアル I/O-SI, INT-CLK
0	1	0	1	×	OUT-PORT	CK-OUT	CLK-OUT+IOOUT-PORT
0	1	1	×	×	SD-OUT	CK-OUT	シリアル I/O-SO, INT-CLK
1	0	0	0	0	SD-IN	CK-IN	BUS-SLAVE-RX
1	0	0	0	1	SD-IN	OUT-PORT	BUS-MASTER-RX(SOFT-CLK)
1	0	0	1	0	OUT-PORT	IN-PORT	IOOUT-PORT+IIN-PORT
1	0	0	1	1	OUT-PORT	OUT-PORT	2OUT-PORT
1	0	1	×	0	SD-OUT	CK-IN	BUS-SLAVE-TX
1	0	1	×	1	SD-OUT	OUT-PORT	BUS-MASTER-TX(SOFT-CLK)
1	1	0	0	×	SD-IN	CK-OUT	BUS-MASTER-RX
1	1	0	1	×	OUT-PORT	CK-OUT	CLK-OUT+IOOUT-PORT
1	1	1	×	×	SD-OUT	CK-OUT	BUS-MASTER-TX

備考 × : don't care

表 16-3 CHI の動作モード一覧

シリアル・インタフェース・モード・レジスタ			ポート I/O 指定			SI 端子	SCK 端子	SO 端子	動作モード
SB	SIO0MS	SIO0TX	P0ABIO2	P0ABIO3	P0BBIO0				
0	0	0	0	0	0	SD-IN	CK-IN	IN-PORT	シリアル I/O-SI, EXT-CLK, IIN-PORT
0	0	0	0	0	1	SD-IN	CK-IN	OUT-PORT	シリアル I/O-SI, EXT-CLK, IOOUT-PORT
0	0	0	0	1	0	OUT-PORT	IN-PORT	IN-PORT	IOOUT-PORT + 2IIN-PORT
0	0	0	0	1	1	OUT-PORT	IN-PORT	OUT-PORT	2OUT-PORT + IIN-PORT
0	0	0	1	0	0	SD-IN	OUT-PORT	IN-PORT	シリアル I/O-SI, INT-CLK(SOFT-CLK), IIN-PORT
0	0	0	1	0	1	SD-IN	OUT-PORT	OUT-PORT	シリアル I/O-SI, INT-CLK(SOFT-CLK), IOOUT-PORT
0	0	0	1	1	0	OUT-PORT	OUT-PORT	IN-PORT	2OUT-PORT + IIN-PORT
0	0	0	1	1	1	OUT-PORT	OUT-PORT	OUT-PORT	3OUT-PORT
0	0	1	0	0	×	SD-IN	CK-IN	SD-OUT	シリアル I/O-SI/SO, EXT-CLK
0	0	1	0	1	×	OUT-PORT	CK-IN	SD-OUT	シリアル I/O-SO, EXT-CLK, IOOUT-PORT
0	0	1	1	0	×	SD-IN	OUT-PORT	SD-OUT	シリアル I/O-SI/SO, INT-CLK(SOFT-CLK)
0	0	1	1	1	×	OUT-PORT	OUT-PORT	SD-OUT	シリアル I/O-SO, INT-CLK(SOFT-CLK), IOOUT-PORT
0	1	0	×	0	0	SD-IN	CK-OUT	IN-PORT	シリアル I/O-SI, INT-CLK, IIN-PORT
0	1	0	×	0	1	SD-IN	CK-OUT	OUT-PORT	シリアル I/O-SI, INT-CLK, IOOUT-PORT
0	1	0	×	1	0	OUT-PORT	CK-OUT	IN-PORT	CLK-OUT, IOOUT-PORT, IIN-PORT
0	1	0	×	1	1	OUT-PORT	CK-OUT	OUT-PORT	CLK-OUT, 2OUT-PORT
0	1	1	×	0	×	SD-IN	CK-OUT	SD-OUT	シリアル I/O-SI/SO, INT-CLK
0	1	1	×	1	×	OUT-PORT	CK-OUT	SD-OUT	シリアル I/O-SO, INT-CLK, IOOUT-PORT
1	×	×	×	×	×	-	-	-	設定禁止

備考 × : don't care

16.1.1 SIO0CH

シリアル・インタフェースのチャンネルを選択するフラグです。

SIO0CH フラグが“0”のときシリアル・インタフェースのハードウェアはCH0に接続され、“1”のときCH1に接続されます。

選択されなかったチャンネルの外部端子は汎用ポートとして使用できます。

表 16-4 シリアル・インタフェースのチャンネル設定

SIO0CH	選択されるチャンネル
0	CH0
1	CH1

16.1.2 SB

シリアル・インタフェースのプロトコルを指定するフラグです。

SB フラグが“0”のときはシリアル I/O モードが、“1”のときは2線式バス・モードが指定されます。

なお、CH1は2線式バス・モードをサポートしていませんので、CH1を使用するときはSBフラグは必ず“0”にしてください。

表 16-5 シリアル・インタフェースのプロトコル指定

SB	プロトコル
0	シリアル I/O モード
1	2線式バス・モード

16.1.3 SIO0MS

SIO0MS はシリアル・インタフェースのクロックの方向を指定するフラグです。

“0”のときは外部クロックが選択され，“1”のときは内部クロックが選択されます。内部クロックを選択したときの周波数はシフト・クロック周波数レジスタ (RF: 39H) によって行います。

2線式バス・モード時に SIO0MS が“0”ならスレーブ動作が，“1”ならマスタ動作が指定されます。

表 16-6 SIO0MS フラグの機能一覧

SIO0MS	機 能
0	2線式バス・モード : スレーブ動作 シリアル I/O モード : 外部クロック動作
1	2線式バス・モード : マスタ動作 シリアル I/O モード : 内部クロック動作

16.1.4 SIO0TX

2線式バス・モードが指定されているときに SIO0TX が“0”のときは受信モードが，“1”のときは送信モードが指定されます。

CH0のシリアル I/O モードが、指定されているとき、SIO0TX フラグを“0”にすると SI モード (SDA端子は入力モード) が，“1”にすると SO モード (SDA端子は出力モード) が指定されます。

CH1のシリアル I/O モードが、指定されているとき、SIO0TX フラグは SO 端子がシリアル・インタフェースとして使用されるかどうかを指定します。SIO0TX フラグが“1”なら SO 端子として使用され，“0”なら汎用ポートとして使用されます。

表 16-7 SIO0TX の機能一覧

SIO0TX	機 能
0	2線式バス・モード : RX (受信) モード CH0 シリアル I/O モード : SI モード CH1 シリアル I/O モード : P0A ₃ を汎用ポートとして使用
1	2線式バス・モード : TX (送信) モード CH0 シリアル I/O モード : SO モード CH1 シリアル I/O モード : P0A ₃ を SO 端子として使用

16.2 クロック・カウンタ

クロック・カウンタはそのとき選択されているシリアル・インタフェースのシフト・クロック端子 (CH0の場合は P0A₁/SCL 端子, CH1の場合は P0A₂/SCK 端子) 上のクロックをカウントするラップ・アラウンド・カウンタで、シフト・クロックを1-9のカウント数で繰り返しカウントします。つまりカウンタの初期値は0で、クロックの立ち上がりが検出されるごとに+1され、9までカウントされると次は1に戻るというカウントを繰り返します。

クロック・カウンタは、次の場合にリセットされ、“0”になります。

(1) 2線式バスモードの場合

- (a) パワーオン・リセット時
- (b) STOP 命令が実行され、システム・クロックが停止した場合
- (c) スタート・コンディションを検出した場合
- (d) シリアル・インタフェースの動作モードが2線式バス・モードからシリアル I/O モードに切り替えられた場合

(2) シリアル I/O モードの場合

- (a) パワーオン・リセット時
- (b) STOP 命令が実行され、システムクロックが停止した場合
- (c) ウェイト・レジスタにデータが書き込まれた場合
- (d) シリアル・インタフェースの動作モードがシリアル I/O モードから2線式バス・モードに切り替えられた場合

クロック・カウンタの内容が“8”または“9”になったかどうかは、ステータス・レジスタによってソフトウェアでテストすることができます。送信モードのときのクロックの休止、または2線式バス・モードでの受信モード時のクロックの休止要求は、ウェイト・レジスタを操作することにより行えます。

16.3 ステータス・レジスタ

ステータス・レジスタは4ビットの読み出し専用レジスタで、2線式バス・モード時のスタート、ストップ状況、および現在のクロック・カウンタの内容を保持します。

図 16-2 ステータス・レジスタの構成

ビット位置	b ₃	b ₂	b ₁	b ₀
フラグ名称	SIO0SF8	SIO0SF9	SBSTT	SBBSY

16.3.1 SBBSY (Serial Bus Busy) フラグ

SBBSY フラグはステータス・レジスタ (RF: 28H) の b₀ (LSB) に配置されており、2線式バス・モードでのビジィ信号検出フラグです。

SBBSY フラグはシリアル・モード・レジスタの SB フラグにより2線式バス・モードが選択されているときのみ有効で、スタート条件が検出されると“1”にセットされ、ストップ条件が検出されると“0”にリセットされます。

シリアル・モード・レジスタによりシリアル I/O モードが選択されると、SBBSY フラグは“0”にリセットされ、以後2線式バス・モードが選択されるまで“0”を保持します。

つまりシリアル I/O モードでは、SBBSY フラグは変化しません。

2線式バス・モードで、送信も受信も行っていない場合、この SBBSY フラグをテストすることにより、他のデバイスが通信を行っているかどうかを判定できます。

16.3.2 SBSTT (Serial Bus Start Test) フラグ

SBSTT フラグはステータス・レジスタの b₁ に配置されており、2線式バス・モードでのスタート条件検出フラグです。

SBSTT フラグはシリアル・モード・レジスタの SB フラグにより2線式バス・モードが選択されているときのみ有効で、スタート条件が検出されると“1”にセットされ、クロック・カウンタの内容が“9”になると“0”にリセットされます。

16.3.3 SIO0SF9 (Serial I/O Shift 9 Clock) フラグ

SIO0SF9 フラグはステータス・レジスタの b₂ に配置されているフラグで、クロック・カウンタの内容が“9”になると“1”にセットされ、クロック・カウンタが“0”または“1”になると“0”にリセットされます。

2線式バス・モードのマスター・モードの場合、スレーブがアクノリッジを返したかどうかを保持するフラグの内容は、このフラグが“1”になってから、次に“1”になるまでの間に読み出さなければなりません。

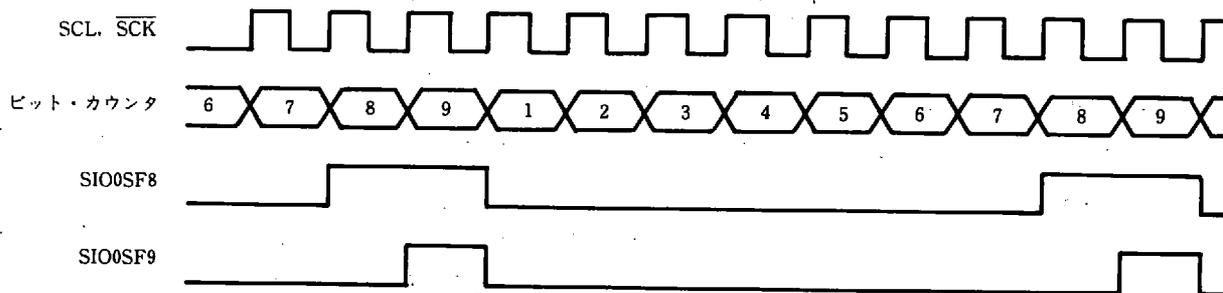
SIO0SF9 フラグは、シリアル・モード・レジスタの内容には影響されません。つまりシリアル I/O モードでもクロック・カウンタの内容が“9”になると SIO0SF9 フラグはセットされます。

16.3.4 SIO0SF8 (Serial I/O Shift 8 Clock) フラグ

SIO0SF8 フラグはステータス・レジスタの b_3 に配置されているフラグで、クロック・カウンタの内容が“8”になると“1”にセットされ、クロック・カウンタが“0”または“1”になると“0”にリセットされます。

プリセッタブル・シフト・レジスタを読み出す操作は、この SIO0SF8 フラグが“1”のときに行ってください。SIO0SF8 フラグは、シリアル・モード・レジスタの内容には影響されません。

図 16 - 3 SIO0SF8, SIO0SF9 の動作



16.4 ウェイト・レジスタ

μPD17062ではシフト・クロックが入力されてもシリアル・インタフェース・ハードウェアを動作させない状態を設定することができます。この状態をウェイト・モードと呼び、ウェイト・レジスタにより設定します。

ウェイト・レジスタは、シリアル・インタフェースで通信中にどのタイミングで通信を休止（ウェイト）するかを指定する SIO0WRQ0 フラグ、SIO0WRQ1 フラグと、現在ウェイト中かどうかを示す SIO0NWT フラグ、および 2線式バス・モードでアクノリッジを返すかまたは返されたかどうかを示す SBACK フラグの4ビットで構成されています。

ウェイト・レジスタはレジスタ・ファイル上に配置されており、ウインドウ・レジスタを介し“PEEK”、“POKE”命令で操作します。

ウェイト・レジスタはパワーオン・リセット時、および“STOP”命令が実行されてシステム・クロックが停止した場合には、すべてのフラグが“0”にリセットされます。

図 16-4 ウェイト・レジスタの構成

ビット位置	b ₃	b ₂	b ₁	b ₀
フラグ名称	SBACK	SIO0NWT	SIO0WRQ1	SIO0WRQ0

16.4.1 SIO0WRQ1, SIO0WRQ0 (Serial I/O Wait Request) フラグ

シリアル・インタフェースのハードウェアに対してどのタイミングでウェイトするかを予約（指定）するフラグです。μPD17062では、ウェイトの概念を2線式バス・モードのスレーブからのものだけでなく、2線式バス・モードの送信側、およびシリアル I/O モードの内部クロック動作にも拡張しています。

なお、ウェイト中は、クロック・カウンタおよびプリセッタブル・シフト・レジスタへのシフト・クロックは禁止されます。つまりウェイト中にシフト・クロック端子のレベルに変化があったとしても、クロック・カウンタは更新されず、プリセッタブル・シフト・レジスタの内容はシフトされません。

表 16-8 ウェイト・タイミング一覧表

SIO0WRQ1	SIO0WRQ0	ウェイト・モード	2線式バス・モード	シリアル I/O モード
0	0	ノー・ウェイト	ウェイトしない	ウェイトしない
0	1	データ・ウェイト	クロック・カウンタの内容が“8”のときのシフト・クロックの立ち下がりでウェイトする。	クロック・カウンタの内容が“8”になればシフト・クロックをハイ・レベルの状態ですべてウェイトする。
1	0	アクリッジ・ウェイト	クロック・カウンタの内容が“9”のときのシフト・クロックの立ち下がりでウェイトする。	クロック・カウンタの内容が“9”になればシフト・クロックをハイ・レベルの状態ですべてウェイトする。
1	1	アドレス・ウェイト	スタートコンディション検出後、クロック・カウンタの内容が“8”になったときのシフト・クロックの立ち下がりでウェイトする。	設定禁止

(1) 2線式バス・モードのスレーブ動作の場合のウェイト

SIO0WRQ1, SIO0WRQ0 で指定されたタイミングになると SCL 端子を出力モードに切り替え、ロウ・レベルを出力します。

もしノー・ウェイト (SIO0WRQ1=SIO0WRQ0=0) が指定されていると、この動作は行いません。

ウェイトの解除は、ウェイト・レジスタの SIO0NWT フラグに“1”を書き込むことにより行います。

たとえば、いまデータ・ウェイト・モード (SIO0WRQ1=0, SIO0WRQ0=1:クロック・カウンタの内容が“8”のときのシフト・クロックの立ち下がりでウェイト) を指定して、ウェイトしているときに SIO0NWT フラグに“1”を書き込むことによりウェイトは解除されます。そして次にクロック・カウンタの内容が“8”になったときのシフト・クロックの立ち下がりで再度ウェイトが行われます。

スレーブ動作で、まだ通信が開始されていないときは、通常アドレス・ウェイト・モード (SIO0WRQ1=SIO0WRQ0=1) を指定します。このウェイト・モードでは、スタート・コンディションが検出されたあと、最初にクロック・カウンタの内容が“8”になったときのシフト・クロックの立ち下がりでウェイトを行います。つまり、このモードはスレーブ・アドレスが送信されてアクリッジのための (9番目) のクロックが立ち上がる前にウェイトを行うモードです。このウェイトが行なわれている間にプリセットプル・シフト・レジスタ (PSR) の内容を読み出し、そのアドレスが自局に割り当てられたものかどうかの判断を行います。

ウェイト中かどうかは SIO0NWT フラグをテストすることにより知ることができます。

(2) 2線式バス・モードのマスタ動作の場合のウェイト

2線式バス・モードのマスタ動作の場合のウェイトとは送信の中断を意味します。このモードの場合、SIO0WRQ1, SIO0WRQ0 フラグに指定されたタイミングになるとシフト・クロックはロウ・レベルに固定されます。

たとえば、いまアクノリッジ・ウェイト・モード (SIO0WRQ1 = 1, SIO0WRQ0 = 0: クロック・カウンタの内容が“9”のときのシフト・クロックの立ち下がりでウェイト) を指定して、ウェイトしているときにフラグをテストすることによりレシーバがアクノリッジを返したかどうかをテストすることができます。また、このウェイト中に次に送信するデータをプリセッタブル・シフト・レジスタセットすることも可能です。

ウェイトの解除はスレーブ動作のときと同様に、SIO0NWT フラグへ“1”を書き込む動作で実行されます。またノー・ウェイトが指定されていると、ウェイト動作は行いません。

(3) シリアル I/O モードの内部クロック動作の場合のウェイト

この場合のウェイトは、2線式バス・モードのマスタ動作の場合とウェイトとはほぼ同義です。つまりこのウェイトも送信の中断を意味します。ただ1つ異なる点は、2線式バス・モードのマスタ動作の場合、シフト・クロックをロウ・レベルにしてウェイトしたのに対し、このシリアル I/O モードの場合にはシフト・クロックをハイ・レベルにしてウェイトする点です。

なお、シリアル I/O モードが指定されている場合、クロック・カウンタはウェイト・レジスタへのデータの書き込み動作によって“0”にリセットされます。このため、たとえばデータ・ウェイト・モードを指定して、次にアクノリッジ・ウェイト・モードに指定し直した場合、クロック・カウンタは“0”にリセットされてからカウントを開始し“9”になったときにシフト・クロックはハイ・レベルの状態で停止します。

つまりシリアル I/O モードの内部クロック動作モードでは、ウェイト・レジスタにデータを書き込むことにより、クロック・カウンタをリセットしてから送信を開始します。

もしノー・ウェイトが指定された場合は、シフト・クロックは連続的に出力されます。

(4) シリアル I/O モードの外部クロック動作の場合のウェイト

シリアル I/O モードの外部クロック動作の場合は、SIO0WRQ1, SIO0WRQ0 で指定されたタイミングで単にクロック・カウンタの更新およびプリセッタブル・シフト・レジスタのシフト動作が禁止されるだけです。たとえばデータ・ウェイト・モードが指定されているときには、クロック・カウンタの内容が“8”になったクロックの立ち下がりでウェイトがかかり、それ以後に入力されたシフト・クロックによってはクロック・カウンタは更新されません。また、プリセッタブル・シフト・レジスタもデータをシフトすることはありません。

ウェイト後のデータ入力を可能にするには、他の場合と同様に SIO0NWT フラグに“1”を書き込みます。つまり、このウェイト・レジスタにデータが書き込まれることによりクロック・カウンタは“0”にリセットされ、またウェイトは解除されます。

16.4.2 SIO0NWT (Serial I/O No Wait) フラグ

SIO0NWT フラグにデータを書き込むことにより、ウェイトの解除、または強制ウェイトを行うことができます。

(1) SIO0NWT に “0” を書き込んだ場合

この場合には強制ウェイトが行われます。つまり、クロック・カウンタおよびプリセットブル・シフト・レジスタへのクロックの供給は禁止されます。

また、もしそのときシリアル・インタフェース・モード・レジスタの SIO0MS フラグが “1” に設定されていればシフト・クロック動作もその状態で停止します。

(2) SIO0NWT に “1” を書き込んだ場合

この場合にはウェイトの解除が行われます。つまり、クロック・カウンタおよびプリセットブル・シフト・レジスタへのクロックは供給され、もしそのときシリアル・インタフェース・モード・レジスタの SIO0MS フラグが “1” に設定されていればシフト・クロック動作はウェイト直前の状態から引き続き行われます。

16.4.3 SBACK (Serial Bus Acknowledge)

SBACK はシリアル・インタフェースの動作モードによりその動作が異なります。

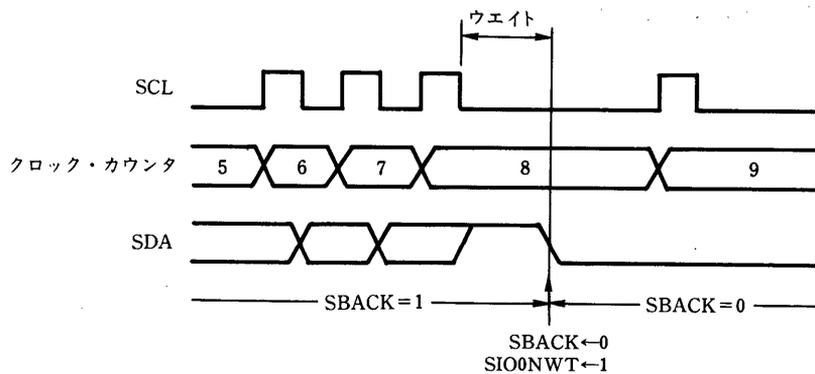
以下に SBACK の動作を説明します。

(1) 2線式バス・モードの受信動作 (SIO0TX=0) の場合

この場合 SBACK フラグに設定されたデータは、アクノリッジ出力のタイミングで自動的に SDA 端子へ出力されます。また、この場合には、SBACK フラグの内容はウェイト・レジスタに対して行われた “POKE 命令” によってのみ変化します。このため連続してアクノリッジを返す場合は、SBACK フラグに “0” を書き込んでおけば自動的に “0” をアルノリッジとして送信します。したがって1バイト受信ごとに SBACK フラグを操作する必要はありません。

なお、SBACK フラグへのデータの書き込みはデータの9ビット目がセットされてから次のデータの9ビット目がセットされるまでに行わなければなりません。通常の使用方法では、8ビット目の立ち下がりまたは9ビット目の立ち下がりでもウェイトをかけるため、そのウェイト時に SBACK フラグにデータを書き込むようにします。

図 16-5 ウェイト中に SBACK を書き換えたときのタイミング図



(2) 2線式バス・モードの送信動作 (SIO0TX=1) の場合

この場合、受信側からのアクノリッジの内容が SBACK フラグにセットされます。つまり SBACK フラグの内容を読み出すことにより受信側のアクノリッジの状態が判断できます。

SBACK フラグの判定は、1 バイトのデータの 9 ビット目がセットされてから次のデータの 9 ビット目がセットされるまでに行わなければなりません。通常の使用法では、9 ビット目の立ち下がりでウェイトをかけ、そのウェイト中に SBACK フラグを読み出すようにします。

SBACK フラグは送信動作時でも POKE 命令によりデータを書き込むことができます。

(3) シリアル I/O モードの場合

この場合は SBACK フラグの内容はシフト・クロックによる影響は受けません。つまり SBACK フラグはシリアル・インタフェースから完全に切り離されています。したがってこのとき SBACK フラグは 1 ビットのフラグとしてデータ記憶用に用いることも可能です。

16.5 プリセッタブル・シフト・レジスタ (PSR)

プリセッタブル・シフト・レジスタはシフト・クロック端子 (CH0 の場合は P0A₁/SCL 端子, CH1 の場合は P0A₂/SCK 端子) 上のクロック信号の立ち下がりに同期してシリアル・データ出力端子 (CH0 の場合は P0A₀/SDA 端子, CH1 の場合は P0A₃/SO 端子) に PSR の最上位ビットの内容を出力し, クロックの立ち上がりに同期してシリアル・データ入力端子 (CH0 の場合は P0A₀/SDA 端子, CH1 の場合は P0B₀/SI 端子) 上のデータを PSR の最下位ビットに読み込むという動作を行う 8 ビットのプリセッタブル・シフト・レジスタです。

ウェイト状態では PSR へシフト・クロックは供給されません。つまりウェイト状態では, PSR はシフト・クロック端子上にクロックが (内部または外部より) 供給されたとしてもデータのシフト動作は行いません。

ウェイト状態でないときの PSR の動作は 2 線式バス・モードとシリアル I/O モードで異なります。

PSR へのデータの書き込みおよび読み出しはデータ・メモリ上の DBF の下位 8 ビット (データ・メモリ・アドレス: 0EH, 0FH) を介して PUT 命令および GET 命令で行います。

(1) 2 線式バス・モードでの PSR の動作

2 線式バス・モードが指定されているときの PSR へのシフト・クロックの供給はクロック・カウンタが“1”から“8”までの間だけ行われます。たとえば 2 線式バス・モードで 9 ビット (8 ビットのデータ+1 ビットのアクノリッジ) のデータを受信する場合, PSR へは最初の 8 ビットのデータのみが読み込まれます。このとき 9 ビット目のデータは, ウェイト・レジスタの SBACK フラグに読み込まれます。

また, 2 線式バス・モードで PSR の内容を送信する場合には, クロック・カウンタの内容が“1”から“8”までの間は PSR の内容がシリアル・データ端子に出力され“9”の期間 (正確には 8 ビット目のクロックの立ち上がりから 9 ビット目のクロックの立ち上がりまでの期間) には SBACK フラグの内容が出力されます。

上記の動作は μPD17062 に内蔵されているシリアル・インタフェースのハードウェアを用いた場合 (内部クロックあるいは外部クロックを用いた場合も) だけでなく, シフト・クロック端子と兼用しているポート (P0A₀) を出力ポートにしてソフトウェアによりクロックを生成しても同様に動作します。

なお, 送信動作の場合, SDA 端子に出力されたデータは次のシフト・クロックの立ち上がりに同期して PSR へ再び読み込まれます。つまり送信動作でシフト・クロックを 8 発出力すると PSR には送信中の端子上のデータが格納されていることとなります。もし送信中にデータの衝突などがなければ, PSR には送信前とまったく同じデータが格納されます。このため送信前のデータと送信後の PSR 内のデータを比較することにより, 正しく送信されたかどうかを確認することができます。

なお, 上記の説明はすべてウェイト状態でない場合の PSR の動作についてであり, ウェイト状態では PSR はシフト動作を行いません。

(2) シリアル I/O モードでの PSR の動作

シリアル I/O モードが指定されているときの PSR へのシフト・クロックの供給はクロック・カウンタの内容には無関係であり, ウェイト状態でないかぎりシフト・クロック端子上のクロックによってシフト動作を行います。

PSR はウェイト状態ではシフト動作はしません。このためシリアル・インタフェースとして PSR を使用せず, かつ PSR を単なるデータ記憶として使用するような場合には, 必ずウェイト状態にしておかなければなりません。

シリアル I/O モードでは, PSR へのデータの書き込み, および読み出しは, 必ずシフト・クロックがハイ・レベルの期間か, またはウェイト状態のときに行ってください。もし他のタイミングで行うと PSR は正しい動作をしません。通常内部クロックを用いる場合は, 8 ビット目のクロックの立ち上がりでウェイトさせ, そのウェイト中に PSR を操作するようにします。また, 外部クロック動作の場合には, 送信側がシフト・クロックをハイ・レベルにしている保証がある期間に PSR の操作を行うようにします。

16.6 シリアル・インタフェース割り込みソース・レジスタ (SIO0IMD)

割り込みソース・レジスタ (SIO0IMD) は、シリアル・インタフェースでの通信時にどのタイミングで CPU に割り込みを発生させるかを指定する 4 ビットのレジスタです。

SIO0IMD はレジスタ・ファイル上の 38H 番地に配置されています。

SIO0IMD レジスタの構成を図 16-6 に示します。SIO0IMD の上位 2 ビットにはレジスタは割り当てられていません。SIO0IMD の上位 2 ビットを読み出すと、それぞれ “0” が読み出されます。

図 16-6 シリアル・インタフェース割り込みソース・レジスタの構成 (RF : 38H)

ビット位置	b ₃	b ₂	b ₁	b ₀
フラグ名	SIO0IMD3 (0)	SIO0IMD2 (0)	SIO0IMD1	SIO0IMD0

表 16-9 シリアル・インタフェース割り込みソース・レジスタの機能一覧

SIO0IMD1	SIO0IMD0	機 能
0	0	シフト・クロックの 7 ビット目の立ち上がりで割り込み要求発生
0	1	シフト・クロックの 8 ビット目の立ち上がりで割り込み要求発生
1	0	スタート・コンディションが検出された直後のシフト・クロックの 7 ビット目の立ち上がりエッジで割り込み要求発生
1	1	ストップ・コンディションを検出したら割り込み要求発生

16.7 シフト・クロック周波数レジスタ (SIO0CK)

シフト・クロック周波数レジスタは、シリアル・インタフェースの内部クロックの周波数を設定するための4ビットのレジスタです。

シフト・クロック周波数レジスタはレジスタ・ファイルの 39H 番地に配置されています。

シフト・クロック周波数レジスタの構成を図 16-7に示します。シフト・クロック周波数レジスタの上位2ビットにはレジスタは割り当てられていません。シフト・クロック周波数レジスタの上位2ビットを読み出すと、それぞれ“0”が読み出されます。

図 16-7 シフト・クロック周波数レジスタの構成 (RF : 39H)

ビット位置	b ₃	b ₂	b ₁	b ₀
フラグ名	SIO0CK3 (0)	SIO0CK2 (0)	SIO0CK1	SIO0CK0

表 16-10 シリアル・インタフェースの内部クロック周波数一覧

SIO0CK1	SIO0CK0	内部クロック周波数
0	0	100 kHz
0	1	200 kHz
1	0	500 kHz
1	1	1 MHz

17. D/A コンバータ

17.1 PWM 端子

μPD17062 は 4 本の 6 ビット PWM 出力端子を備えており、周波数 15.625 kHz のパルス信号のデューティを 64 段階に可変することができます。したがって外部にロウ・パス・フィルタを付加することにより D/A コンバータを構成することができます。また PWM 端子は 1 ビットの出力ポートとして使用することも可能です。

D/A コンバータとして使用するとき、その出力値は、出力データ設定用ラッチ PWMR に設定します。出力データ設定用のラッチ (PWMR0, PWMR1, PWMR2, PWMR3) は、それぞれ周辺アドレスの 05H, 06H, 07H, 08H に割り当てられており DBF を介してデータの読み書きができます。PWMR のアドレスと各端子の対応を表 17-1 に示します。

表 17-1 PWMR のアドレスと対応する端子

周辺装置	周辺アドレス	対応する端子
PWM0	05H	PWM ₀
PWM1	06H	PWM ₁
PWM2	07H	PWM ₂
PWM3	08H	PWM ₃

PWMR はおのおの 7 ビットで構成されています。PWMR の構造と DBF との対応を図 17-1 に示します。PWMR の最上位ビットは、PWM 信号出力端子として使用するか、出力ポートとして使用するかを指定するビットで、下位 6 ビットは PWM 信号の出力値を設定するビットです。

PWMR 端子の出力波形を図 17-2 に示します。

図 17-1 PWMR の構造と DBF との対応

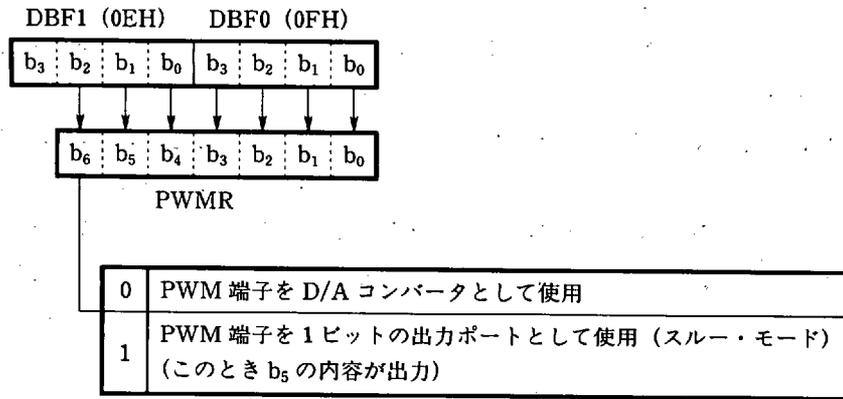
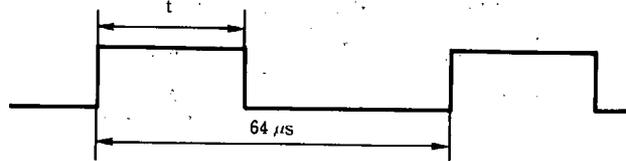


図 17-2 PWM 端子の出力波形



$t = n + 0.75 (\mu s)$ (n : PWMR に設定された値)

18. PLL 周波数シンセサイザ

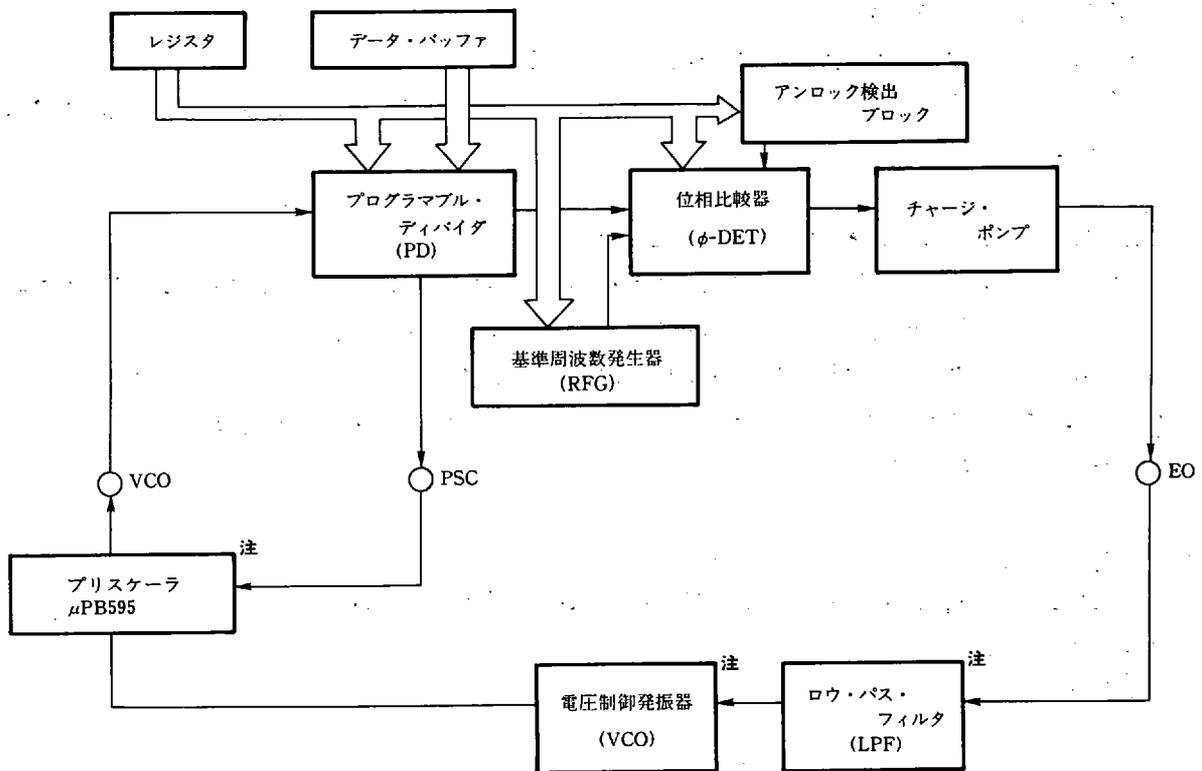
18.1 PLL 周波数シンセサイザの構成

図 18-1 に PLL 周波数シンセサイザのブロック図を示します。

図 18-1 に示すように PLL 周波数シンセサイザは、プログラマブル・ディバイダ (PD)、位相比較器 (ϕ -DET)、基準周波数発生器 (RFG) およびチャージ・ポンプから構成されています。これらのブロックと、外部のロウ・パス・フィルタ (LPF) および電圧制御発振器 (VCO) を接続することにより PLL 周波数シンセサイザを構成できます。

各ブロックについては、18.3-18.5項を参照してください。

図 18-1 PLL 周波数シンセサイザのブロック図



注 外部回路

18.2 PLL 周波数シンセサイザの各ブロックの機能概略

PLL 周波数シンセサイザは、VCO 端子から入力された信号をプログラマブル・ディバイダで分周し、基準周波数との位相差を EO 端子から出力します。

PLL 周波数シンセサイザは、CE 端子がハイ・レベルのときのみ動作します。CE 端子がロウ・レベルのときはディスエーブル状態になります。PLL のディスエーブル状態については18.6項を参照してください。

以下の(1)-(4)に各ブロックの機能概略を示します。

(1) プログラマブル・ディバイダ (PD)

VCO 端子から入力された信号を分周します。分周方式は NEC オリジナルのパルス・スワロ方式です。分周値の設定はデータ・バッファ (DBF) を介して行います。

18.3項を参照してください。

(2) 基準周波数発生器 (RFG)

位相比較器 (ϕ -DET) で比較する基準周波数を生成します。

基準周波数は PLL レファレンス・モード・セレクト・レジスタ (アドレス 13H 番地) により選択できます。

18.4項を参照してください。

(3) 位相比較器 (ϕ -DET) とアンロック検出ブロック

プログラマブル・ディバイダ (PD) から出力された信号と基準周波数発生器 (RFG) からの信号を比較し、位相差を出力します。また、PLL のアンロック状態を検出することが可能です。

PLL のアンロック状態の検出は、PLL アンロック FF ディレイ・コントロール・レジスタ (アドレス 32H 番地) および PLL アンロック FF ジャッジ・レジスタ (アドレス 22H 番地) により制御します。

18.5項を参照してください。

(4) チャージ・ポンプ

位相比較器 (ϕ -DET) から出力された信号をハイ・レベル、ロウ・レベルおよびフローティング出力として EO 端子から出力します。

18.5項を参照してください。

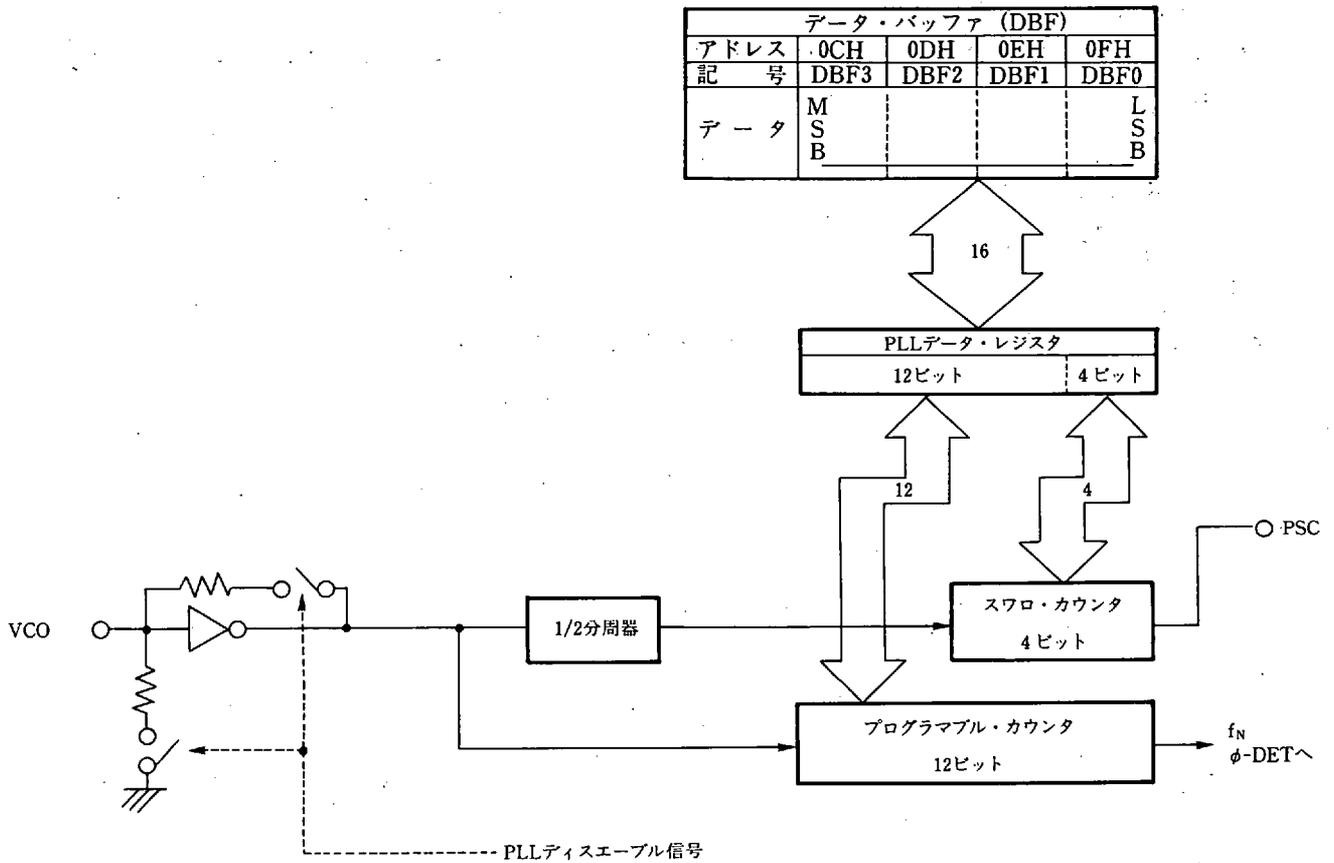
18.3 プログラマブル・ディバイダ (PD) と PLL モード・セレクト・レジスタ

18.3.1 プログラマブル・ディバイダの構成

図 18-2 にプログラマブル・ディバイダ (PD) の構成を示します。

図 18-2 に示すように、プログラマブル・ディバイダは、スワロ・カウンタおよびプログラマブル・カウンタから構成されています。

図 18-2 プログラマブル・ディバイダの構成



18.3.2 プログラマブル・ディバイダ (PD) とデータ・バッファ (DBF)

プログラマブル・ディバイダは、VCO 端子から入力された信号を、スワロ・カウンタおよびプログラマブル・カウンタに設定された値により分周します。

スワロ・カウンタおよびプログラマブル・カウンタは、それぞれ4ビットおよび12ビットのバイナリ・ダウン・カウンタで構成されています。

スワロ・カウンタおよびプログラマブル・カウンタへの分周値の設定はデータ・バッファ (DBF) を介して PLL データ・レジスタ (PLLR: アドレス 41H 番地) に設定します。

PLL データ・レジスタへのデータの設定および読み込みは“PUT PLLR,DBF”および“GET DBF,PLLR”命令で行います。

分周する値は“N値”と呼びます。

PLL データ・レジスタ (PLLR) に設定された値“N”と、プログラマブル・ディバイダにより分周されて出力される信号の周波数“ f_N ”は以下のようになります。

パルス・スワロ方式

$$f_N = \frac{f_{in}}{N} \quad N: 16\text{ビット}$$

各分周方式時の分周値 (N値) の設定については18.7項を参照してください。

18.4 基準周波数発生器 (RFG)

18.4.1 基準周波数発生器 (RFG) の構成と機能

図 18-3 に基準周波数発生器の構成を示します。

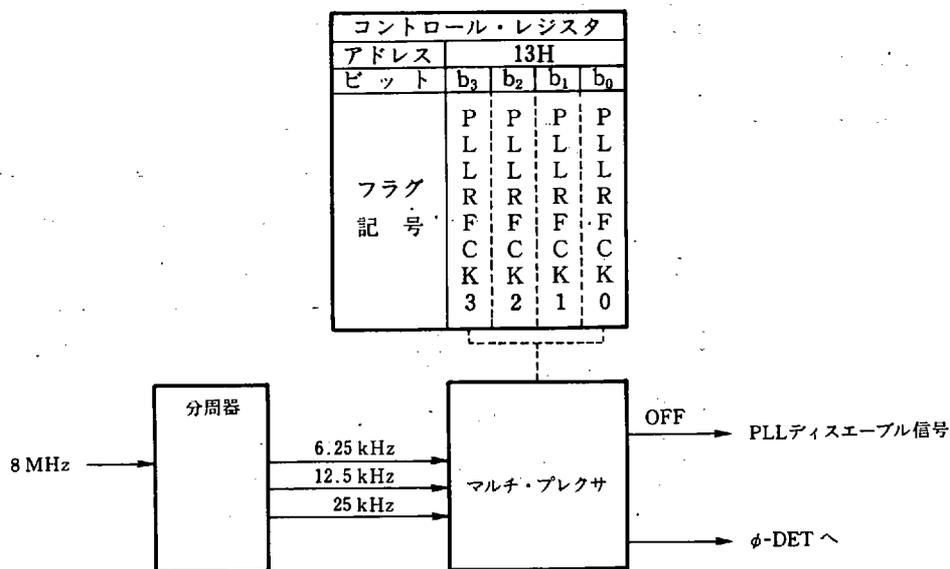
図 18-3 に示すように基準周波数発生器は、クロック発振の 8 MHz を分周して PLL 周波数シンセサイザの基準周波数 “ f_r ” を発生します。

基準周波数 f_r は 6.25, 12.5, 25 kHz の 3 種類を選択できます。

基準周波数 f_r の選択は PLL レファレンス・モード・セレクト・レジスタ (アドレス 13H 番地) により行います。

18.4.2項に PLL レファレンス・モード・セレクト・レジスタの構成と機能を示します。

図 18-3 基準周波数発生器 (RFG) の構成



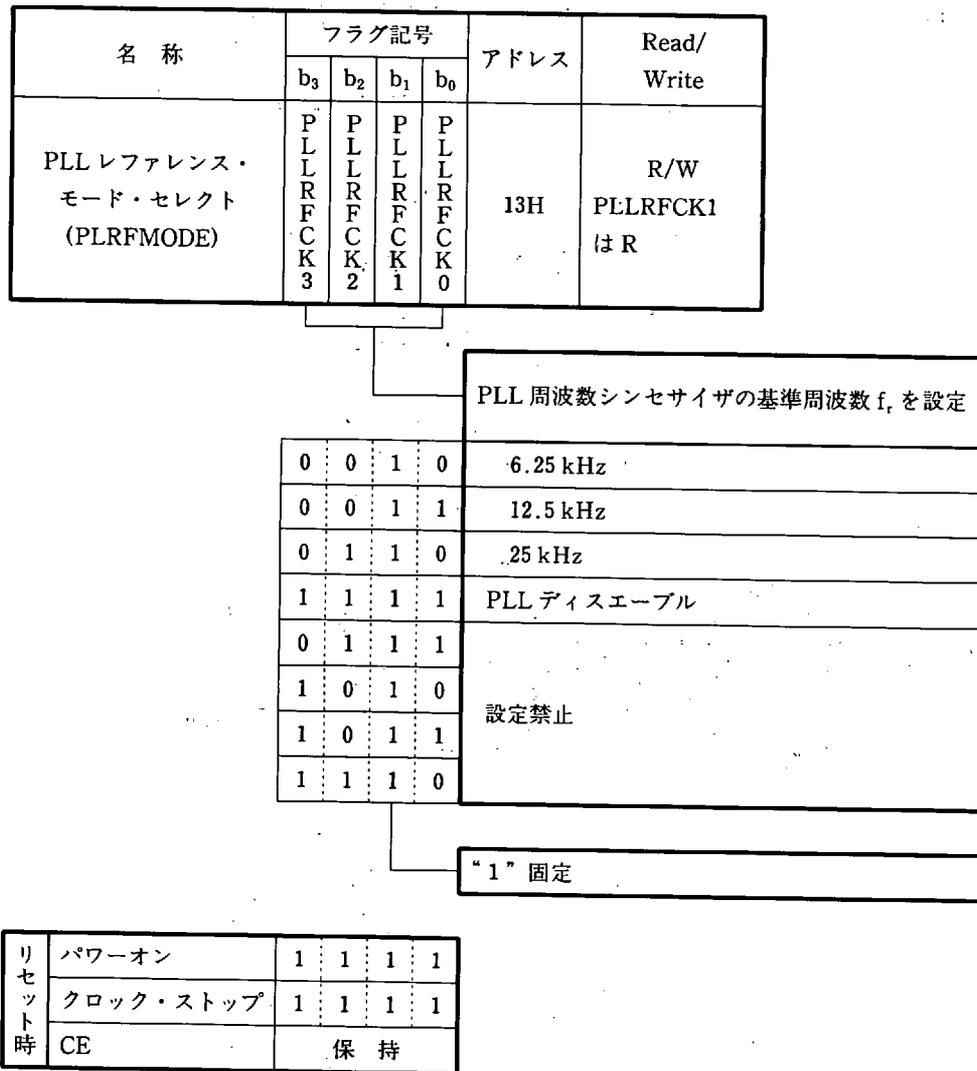
18.4.2 PLL レファレンス・モード・セレクト・レジスタの構成と機能

図 18-4 に PLL レファレンス・モード・セレクト・レジスタの構成と機能を示します。

PLL レファレンス・モード・セレクト・レジスタにより PLL ディスエーブルを選択したときは、VCO 端子が内部でプルダウンされます。また、EO 端子はフローティングとなります。

PLL ディスエーブルについては、18.6項を参照してください。

図 18-4 PLL レファレンス・モード・セレクト・レジスタの構成と機能



18.5 位相比較器 (φ-DET), チャージ・ポンプおよびアンロック検出ブロック

18.5.1 位相比較器 (φ-DET), チャージ・ポンプおよびアンロック検出ブロックの構成

図 18-5 に位相比較器 (φ-DET), チャージ・ポンプおよびアンロック検出ブロックの構成を示します。

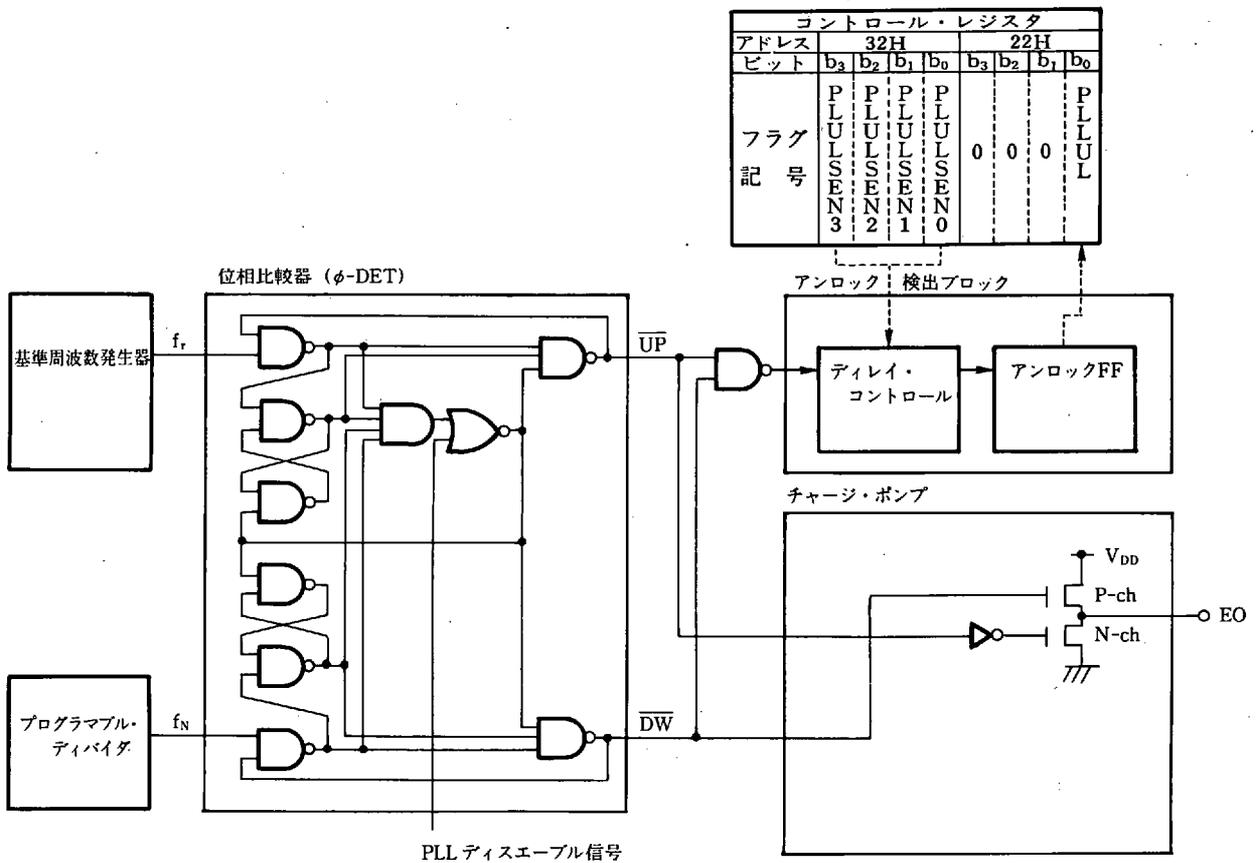
位相比較器は, プログラマブル・ディバイダ (PD) の分周周波数出力 “ f_N ” と基準周波数発生器の基準周波数出力 “ f_r ” の位相を比較し, アップ要求信号 (\overline{UP}) およびダウン要求信号 (\overline{DW}) を出力します。

チャージ・ポンプは位相比較器の出力をエラー・アウト端子 (EO 端子) から出力します。

アンロック検出ブロックは, デイレイ・コントロール回路およびアンロック FF から構成されており, PLL 周波数シンセサイザのアンロック状態を検出します。

18.5.2, 18.5.3 および 18.5.4 にそれぞれ位相比較器, チャージ・ポンプおよびアンロック検出ブロックの動作を示します。

図 18-5 位相比較器, チャージ・ポンプおよびアンロック検出ブロックの構成



18.5.2 位相比較器 (ϕ -DET) の機能

図 18-5 に示したように位相比較器は、プログラマブル・ディバイダ (PD) の分周出力 " f_N " と基準周波数 " f_r " の位相を比較し、アップ要求信号 (\overline{UP}) およびダウン要求信号 (\overline{DW}) を出力します。

すなわち、分周周波数 f_N が基準周波数 f_r より低い周波数であればアップ要求を出力し、分周周波数 f_N が基準周波数 f_r より高い周波数であればダウン要求を出力します。

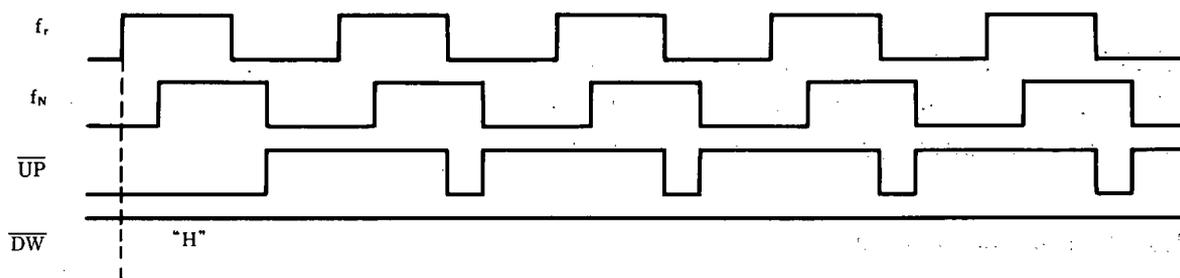
図 18-6 に基準周波数 f_r 、分周周波数 f_N 、アップ要求 \overline{UP} 、ダウン要求 \overline{DW} 信号の関係を示します。

PLL ディスエーブル時は、アップ要求およびダウン要求とも出力されません。

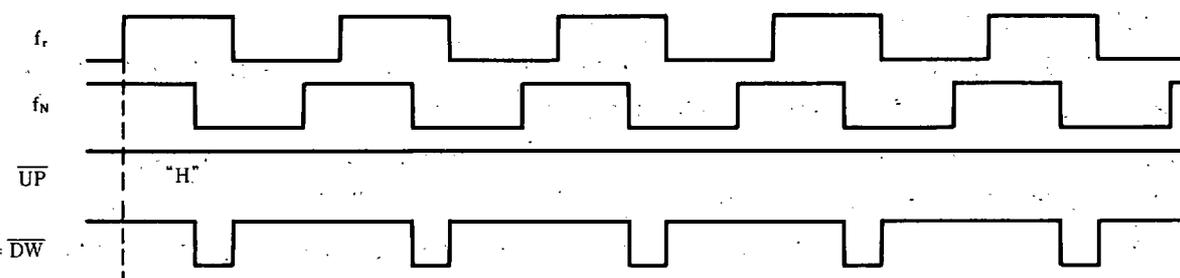
アップ要求およびダウン要求はそれぞれチャージ・ポンプおよびアンロック検出ブロックへ入力されます。

図 18-6 f_r , f_N , \overline{UP} , \overline{DW} 信号の関係

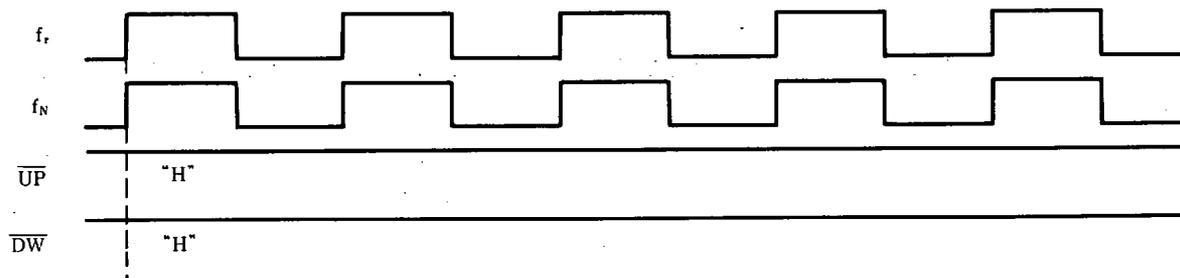
(1) f_N が f_r より位相が遅れているとき



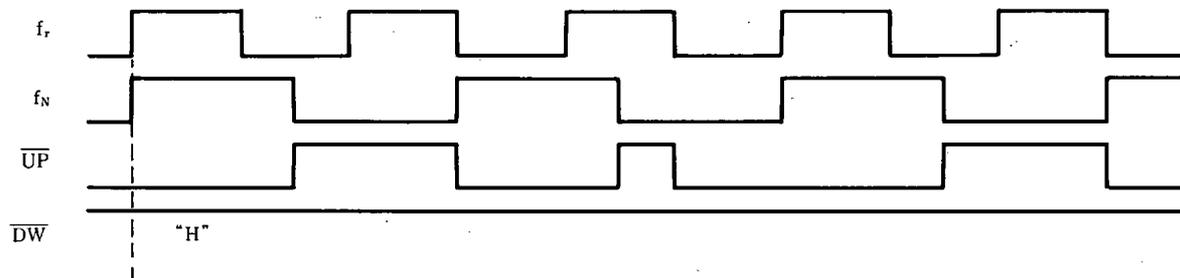
(2) f_N が f_r より位相が進んでいるとき



(3) f_N と f_r の位相が同じとき



(4) f_N が f_r より周波数が低いとき



18.5.3 チャージ・ポンプ

図 18-5 に示したようにチャージ・ポンプは、位相比較器 (ϕ -DET) からのアップ要求信号 (\overline{UP}) およびダウン要求信号 (\overline{DW}) をエラー・アウト端子 (EO 端子) から出力します。

したがって、エラー・アウト端子の出力と分周周波数 f_N および基準周波数 f_r の関係は次のようになります。

基準周波数 $f_r >$ 分周周波数 f_N のとき：ロウ・レベル出力

基準周波数 $f_r <$ 分周周波数 f_N のとき：ハイ・レベル出力

基準周波数 $f_r =$ 分周周波数 f_N のとき：フローティング

18.5.4 アンロック検出ブロック

図 18-5 に示したようにアンロック検出ブロックは、位相比較器 (ϕ -DET) のアップ要求信号 (\overline{UP}) およびダウン要求信号 (\overline{DW}) から、PLL 周波数シンセサイザのアンロック状態を検出します。

すなわち、アンロック状態中はアップ要求またはダウン要求信号のどちらか一方がロウ・レベルを出力するため、このロウ・レベル信号によりアンロック状態を検出します。アンロック状態であると、アンロック・フリップフロップ (FF) がセット(1)されます。

アンロック FF の状態は、PLL アンロック FF ジャッジ・レジスタ (アドレス 22H 番地) により検出します。

アンロック FF は、そのとき選択されている基準周波数 f_r の周期でセットされます。

また、PLL アンロック FF ジャッジ・レジスタの内容を PEEK 命令で読み込むとリセットされます。

したがって、アンロック FF の検出は基準周波数 f_r の周期 $1/f_r$ より長い周期で検出する必要があります。

アンロック・ディレイ・コントロール回路は、位相比較器のアップ要求およびダウン要求信号にディレイをかけることによりアンロック FF がセットされる状態を制御します。

すなわち、ディレイを大きくすれば、分周周波数 f_N と基準周波数 f_r の位相のずれが大きくてもアンロック FF はセットされません。

アンロック・ディレイ・コントロール回路のディレイ時間の設定は、PLL アンロック FF ディレイ・コントロール・レジスタ (アドレス 32H 番地) により行います。

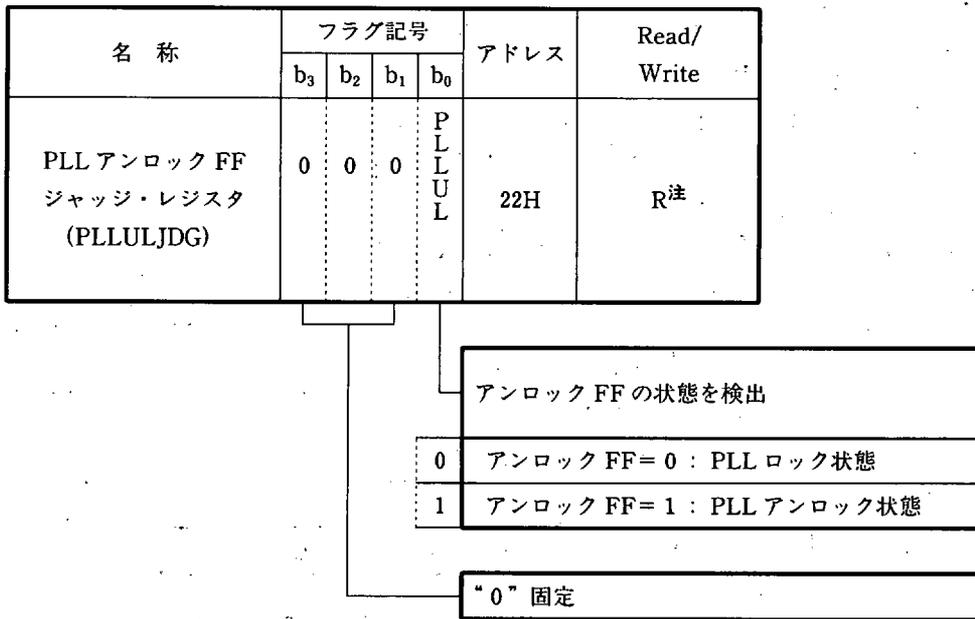
次に PLL アンロック FF ジャッジ・レジスタおよび PLL アンロック FF ディレイ・コントロール・レジスタの構成と機能を示します。

(1) PLL アンロック FF ジャッジ・レジスタ (PLLULJDG)

このレジスタは読み出し専用レジスタです。“PEEK”命令でウインド・レジスタ (WR) に内容を読み出すことによりリセットされます。

また、アンロック FF は基準周波数 f_r の周期でセットされるため、このレジスタをウインド・レジスタに読み込む場合は、基準周波数の周期 $1/f_r$ より遅い周期で読み込む必要があります。

図 18-7 PLL アンロック FF ジャッジ・レジスタ (PLLULJDG) の構成と機能



リ セ ッ ト 時	パワーオン	0	0	0	不 定
	クロック・ストップ				保 持
	CE				保 持

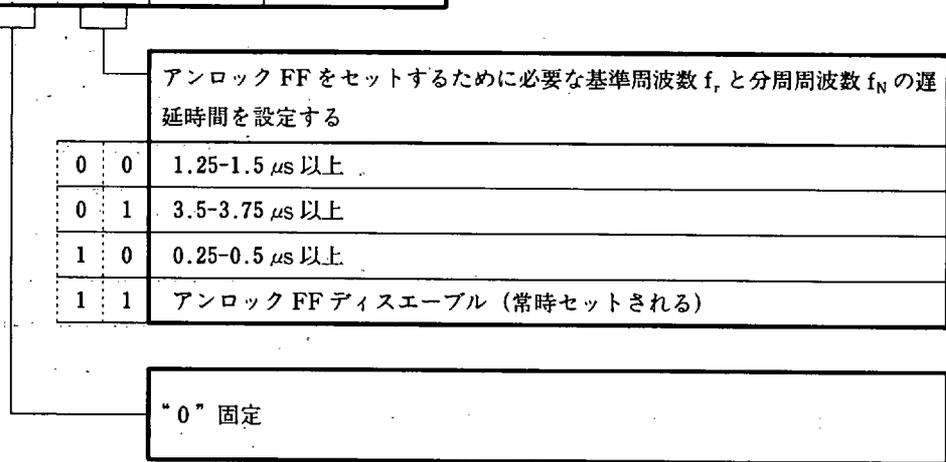
備考 PLLULJDG は、“PEEK”命令で内容を読み出すとリセットされます。

(2) PLL アンロック FF ディレイ・コントロール・レジスタ (PLULSEN)

アンロック FF ディスエーブル状態を設定するとアンロック FF は常にセットされた状態になるため、PLL アンロック FF ジャッジ・レジスタにより PLL のロック状態を検出すると常にアンロック状態 (PLLUL フラグ=1) となりますので注意してください。

図 18-8 PLL アンロック FF ディレイ・コントロール・レジスタ (PLULSEN) の構成と機能

名称	フラグ記号				アドレス	Read/Write
	b ₃	b ₂	b ₁	b ₀		
PLL アンロック FF ディレイ・コントロール (PLULSEN)	P L U L S E N 3	P L U L S E N 2	P L U L S E N 1	P L U L S E N 0	32H	R/W



リ セ ッ ト 時	パワーオン	0	0	0	0
	クロック・ストップ			0	0
	CE			保 持	保 持

18.6 PLL ディスエーブル状態

PLL 周波数シンセサイザは、CE 端子がロウ・レベルの間は動作を停止（ディスエーブル）します。PLL レファレンス・モード・セレクト・レジスタ (PLRFMODE：アドレス 13H 番地) により PLL ディスエーブルが選択されているときも同様に動作を停止します。

表 18-1 に PLL ディスエーブル時の各ブロックの動作を示します。

PLL レファレンス・モード・セレクト・レジスタは CE リセット時にイニシャライズされない（以前の状態を保持する）ため、CE 端子がロウ・レベルになり PLL ディスエーブルとなった後 CE 端子がハイ・レベルに立ち上がると以前の状態に復帰します。

したがって、CE リセット時に PLL ディスエーブルとする必要がある場合は、プログラムでイニシャライズする必要があります。

パワーオン・リセット時は PLL ディスエーブルになります。

表 18-1 PLL ディスエーブル時の各ブロックの動作

各ブロック	CE 端子=Low または PLRFMODE=1111B
VCO 端子	内部でプルダウン
プログラマブル・カウンタ	分周停止
基準周波数発生器	出力停止
位相比較器	出力停止
チャージ・ポンプ	エラー・アウト端子をフローティング

18.7 PLL 周波数シンセサイザのデータ設定方法

PLL 周波数シンセサイザを制御するためには以下に示すデータが必要です。

- (1) 基準周波数 : f_r
- (2) 分周値 : N

次に PLL データ設定方法を示します。

(1) 基準周波数 f_r の設定

PLL レファレンス・モード・セレクト・レジスタにより基準周波数を設定します。

(2) 分周値 N の計算方法

分周比 N は次の式で計算します。

$$N = \frac{f_{UCO}}{P \times f_r}$$

f_{UCO} : VCO 端子の入力周波数

f_r : 基準周波数

P : プリスケーラの分周比

(3) PLL データ設定例

以下に西ヨーロッパ TV の 02 ch を受信するためのデータ設定方法を示します。プリスケーラは μPB595 を使用し、分周比 P は 8 として計算します。

受信周波数 : 48.25 MHz

基準周波数 : 6.25 kHz

中間周波数 : 38.9 MHz

分周値 N は

$$N = \frac{f_{UCO}}{P \times f_r} = \frac{48250 + 38900}{8 \times 6.25} = 1743 \text{ (10進)}$$

$$= 06CFH \text{ (16進)}$$

PLL データ・レジスタ (PLL R : アドレス 41H) および PLL レファレンス・モード・セレクト・レジスタ (PLRFMODE : アドレス 13H) へのデータは以下のように設定します。

PLL R			
0000	0110	1100	1111
0	6	C	F

PLRFMODE
0010
6.25 kHz

19. A/D コンバータ

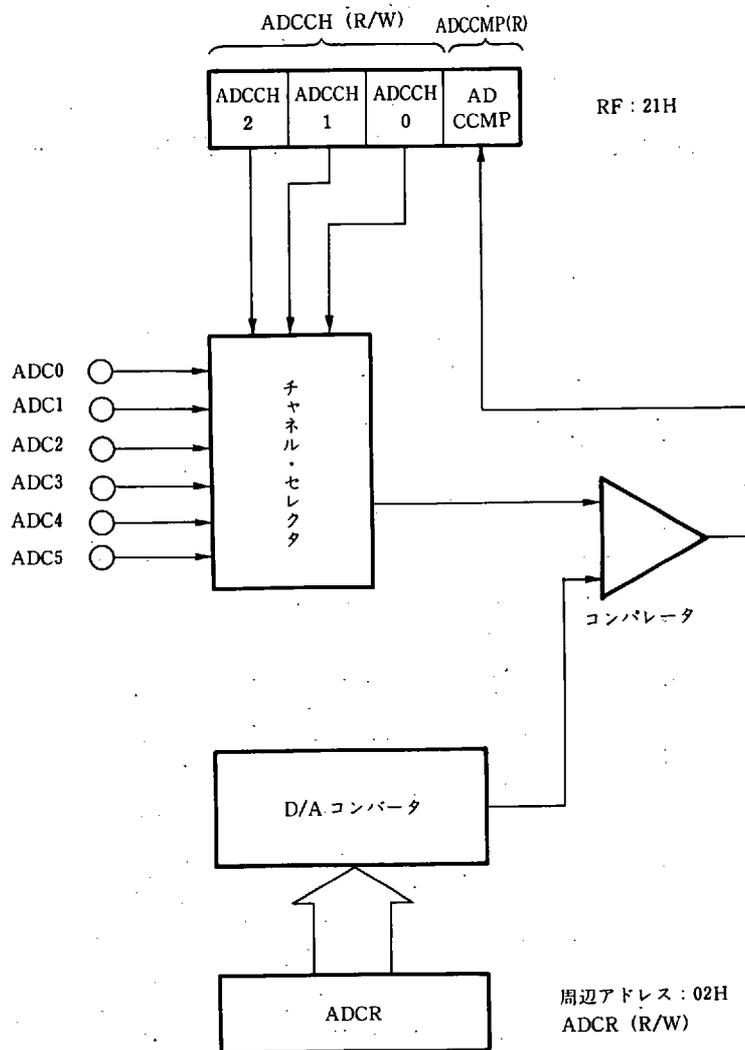
μPD17062 にはプログラムによる逐次比較方式の 4 ビット A/D コンバータが内蔵されています。

19.1 動作原理

μPD17062 の A/D コンバータは抵抗ストリング方式の 4 ビット D/A コンバータとコンパレータとから構成されています。

D/A コンバータへのデータ設定は周辺アドレスの 02H 番地に配置されている 4 ビットのレジスタ ADCR で行います。比較結果はレジスタ・ファイル上の ADCCMP フラグで判定します。

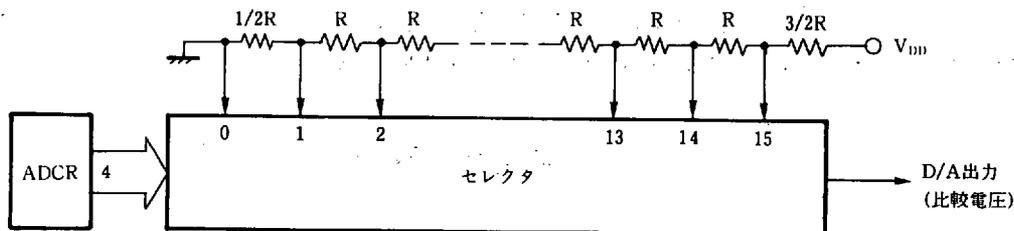
図 19-1 A/D コンバータの構造



19.2 D/A コンバータの構造

μPD17062 の A/D コンバータに使用されている D/A コンバータは V_{DD} と GND 間に16段の抵抗を直列に接続し、その接続点の電圧を選択する抵抗ストリング方式の D/A コンバータです。D/A コンバータの構造を図 19-2 に示します。

図 19-2 D/A コンバータの構造



この構造による D/A コンバータでは、ADCR に 0000B の値を設定したときには GND レベルが出力され、0001B の値が設定されると $1/32 \times V_{DD}$ の電圧が出力されます。同様に n (10進数) の値が設定されたときの比較電圧 V_{REF} は次式で表せます。

$$V_{REF} = V_{DD} \times \frac{2n-1}{32} \quad (\text{ただし } 15 \geq n \geq 1)$$

表 19-1 D/A コンバータの比較電圧

設定データ (ADCR)		比較電圧 (V_{REF})	
16進	2進	$\times V_{DD}$	$V_{DD} = 5V$
0	0000	0	0 [V]
1	0001	1/32	0.15625
2	0010	3/32	0.46875
3	0011	5/32	0.78125
4	0100	7/32	1.09375
5	0101	9/32	1.40625
6	0110	11/32	1.71875
7	0111	13/32	2.03125
8	1000	15/32	2.34375
9	1001	17/32	2.65625
A	1010	19/32	2.96875
B	1011	21/32	3.28125
C	1100	23/32	3.59375
D	1101	25/32	3.90625
E	1110	27/32	4.21875
F	1111	29/32	4.53125

19.3 比較電圧設定レジスタ (ADCR)

ADCR は A/D コンバータの比較電圧を設定するための 4 ビットのレジスタで周辺アドレスの 02H 番地に配置されています。ADCR レジスタへのデータの読み書きは、“PUT”、“GET” 命令を用いデータ・バッファを介して行います。このとき ADCR は 4 ビットですが DBF とのデータ転送は 8 ビット単位で行われます。すなわち DBF1 (0EH) と DBF0 (0FH) の 8 ビットを介し、データ転送を行います。つまり ADCR レジスタの内容を読む場合、“GET DBF, ADCR” を実行すると DBF0 には ADCR レジスタの内容が転送され、DBF1 は 0000B になります。また “PUT ADCR, DBF” を実行すると、DBF0 のデータが ADCR レジスタに設定され、DBF1 のデータは何であってかまいません。

ADCR は電源投入時の値は不定です。クロック・ストップ時、CE リセット時は以前のデータが保持されています。

19.4 比較判定レジスタ (ADCCMP)

ADCCMP は ADC 端子の入力電圧と比較電圧 (V_{REF}) をコンパレータにより比較した結果が格納されるレジスタで、レジスタ・ファイルの 21H 番地の b_0 (LSB) に配置されています。ADCCMP は 1 ビットの読み出し専用のレジスタであり書き込むことはできません。ADCCMP を読み出すときは、“PEEK” 命令を用いウインドウ・レジスタに読み出します。このときウインドウ・レジスタの上位 3 ビットには ADC 端子選択データが読み出されています。

ADCCMP はウインドウ・レジスタに読み出したとき次のように設定されます。

入力電圧 < 比較電圧のとき ADCCMP = 0

入力電圧 \geq 比較電圧のとき ADCCMP = 1

19.5 ADC 端子選択レジスタ (ADCCHn)

ADCCHn は A/D コンバータの入力端子を選択するレジスタです。レジスタ・ファイルの 21H 番地の上位 3 ビットに配置されています。ADCCHn と実際に選択される端子の関係を表 19-2 に示します。

表 19-2 ADC 端子の選択

(MSB)			(LSB)	
b ₃	b ₂	b ₁	#0	(RF : 21H)
			└──┬──┘	ADCCMP
ADCCH2	ADCCH1	ADCCH0	選択される端子	
0	0	0	ADC ₀	
0	0	1	PIC ₃ /ADC ₁	
0	1	0	P0D ₀ /ADC ₂	
0	1	1	P0D ₁ /ADC ₃	
1	0	0	P0D ₂ /ADC ₄	
1	0	1	P0D ₃ /ADC ₅	
1	1	0	該当する端子なし	
1	1	1	(設定禁止)	

PIC₃/ADC₁ を A/D コンバータとして使用するときは PIC を入力ポートに指定してください。

P0D₀/ADC₂ - P0D₃/ADC₅ はプルダウン抵抗付きの端子ですが A/D コンバータに選択されると、その端子のプルダウン抵抗は切り離されます。

また PIC, P0D とも A/D コンバータに選択された端子をポートとして読み出すと “0” が読み出されます。

19.6 A/D 変換のプログラム例

次に逐次比較法による A/D 変換のプログラム例を示します。
 変換後の結果は DBF0 に格納されています。

プログラム例

```
DBF0B3   FLG 0.0FH.3
DBF0B2   FLG 0.0FH.2
DBF0B1   FLG 0.0FH.1
DBF0B0   FLG 0.0FH.0
```

START :

BANK0

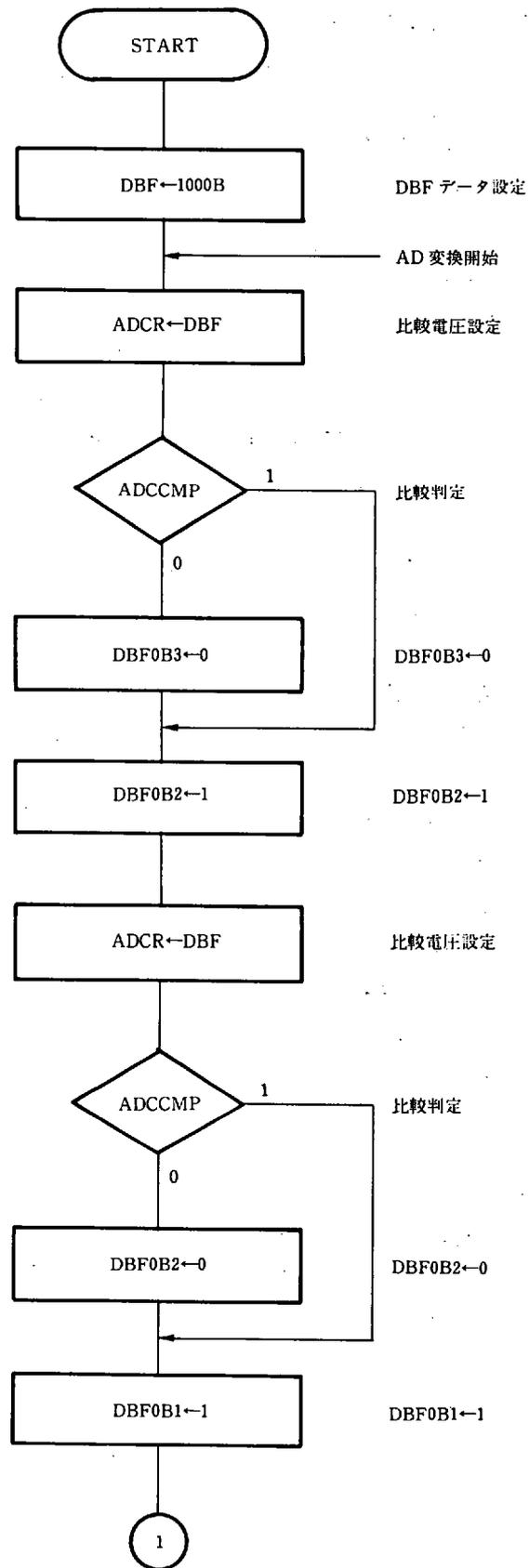
```
INITFLG  DBF0B3, NOT DBF0B2, NOT DBF0B1, NOT DBF0B0 ; DBF データ設定
PUT       ADCR, DBF ; 比較電圧設定
SKT1     ADCCMP ; 比較判定
CLR1     DBF0B3 ; DBF0B3 ← 0
SET1     DBF0B2 ; DBF0B2 ← 1
PUT       ADCR, DBF ; 比較電圧設定
SKT1     ADCCMP ; 比較判定
CLR1     DBF0B2 ; DBF0B2 ← 0
SET1     DBF0B1 ; DBF0B1 ← 1
PUT       ADCR, DBF ; 比較電圧設定
SKT1     ADCCMP ; 比較判定
CLR1     DBF0B1 ; DBF0B1 ← 0
SET1     DBF0B0 ; DBF0B0 ← 1
PUT       ADCR, DBF ; 比較電圧設定
SKT1     ADCCMP ; 比較判定
CLR1     DBF0B0 ; DBF0B0 ← 0
```

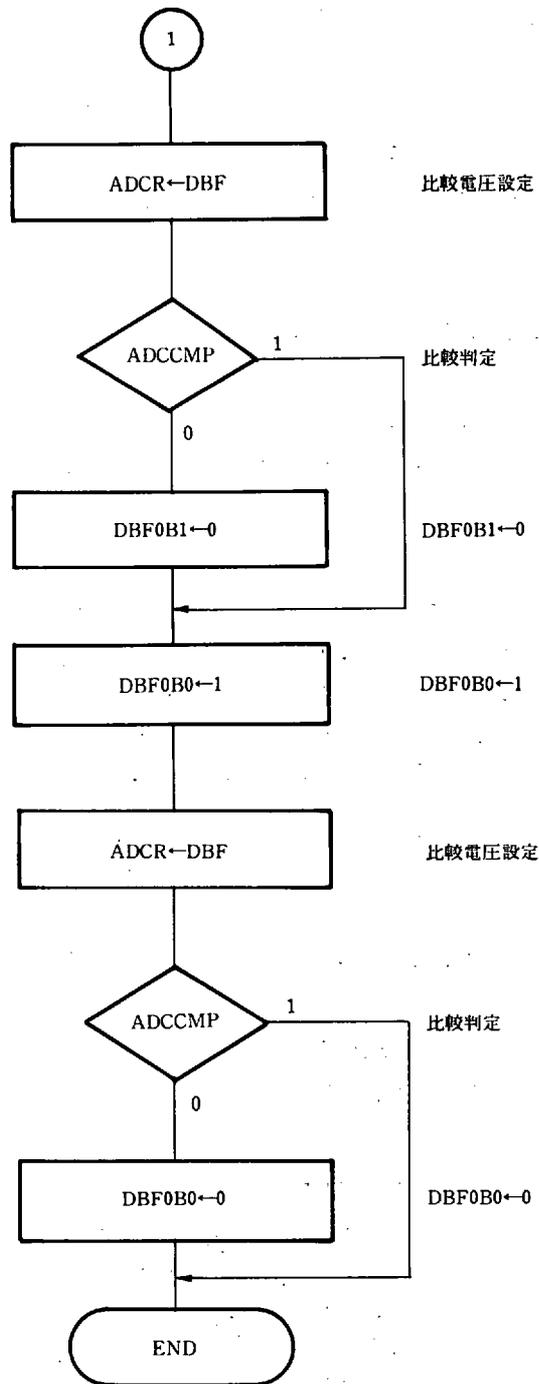
END :

変換ステップ数 : 17ステップ

変換時間 : 34 μs (DMA モード・オフのとき)
 204 μs (DMA モード・オンのとき)

フロー・チャート





20. IDC (Image Display Controller)

IDCはTV画面上にチャンネル番号や、音量表示、タイマ時刻などを表示する機能です。その表示パターンはユーザ・プログラマブルでCROM領域に表示パターンを定義します。

実際に表示するパターンはVRAMに格納します。VRAMはデータ・メモリのBANK1, BANK2に割り当てられています。

20.1 仕様概要および制限事項

(1) 1画面表示文字数：最大97文字

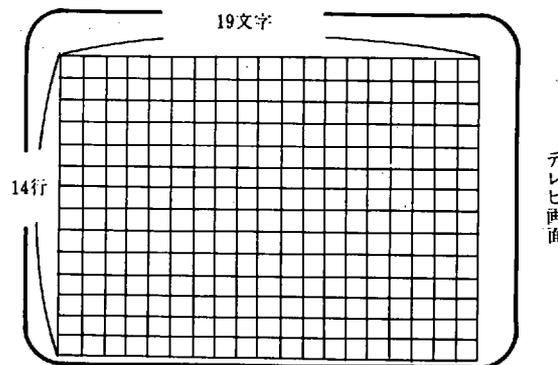
1行に対してコントロール・コードを1回使用した場合。コントロール・データの使用回数により最大表示文字数は変わります。

文字サイズ	最大1行表示文字数	コントロール・データ使用回数(1行に対する)
1倍(最小)	19文字	3回まで
2倍	9文字	6回まで
3倍	5文字	5回まで
4倍	4文字	4回まで

コントロール・データ使用回数3回までというのは、1行に3回の色指定が可能ということになります。

(2) 表示位置可変範囲：横19文字，縦14行以内

TV画面に対して、以下の範囲内となります。

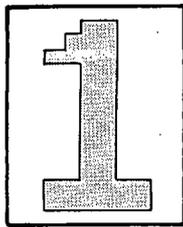


(3) 文字単位で色指定可能 (8色：白，黒含む)

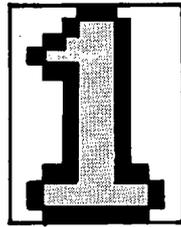
R, G, B独立 (コントロール・コードによる注)

注 コントロール・データは1行に3つまで指定できます。

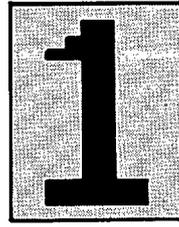
(4) 文字単位でラウンディング, ふちどり, 白ヌキ設定可



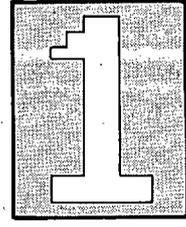
ふちどりなし



ふちどりあり



ラウンディング



白ヌキ



: R, G, Bによる色指定



: ブランク (黒)

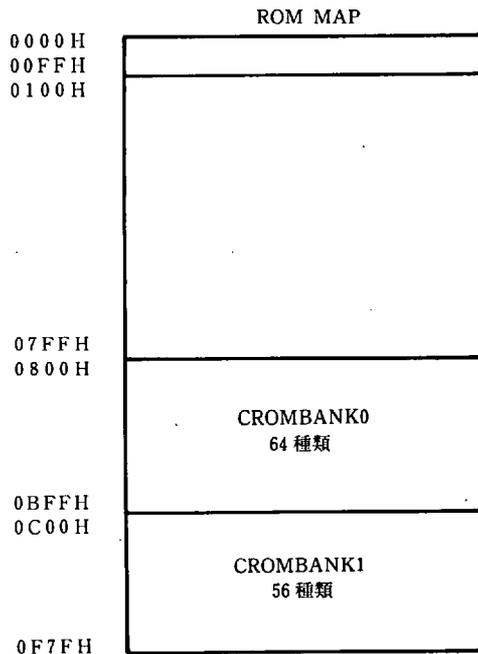


: 背景 (TV画面)

(5) フォントの種類 : 120種類 (ユーザ・プログラマブル)

ただし1画面に同時に表示できるのは64種類。

キャラクタ・パターン・データは, プログラム・メモリ上に配置してあり (CROM), 120種類のキャラクタ・パターン・データを設定 (ただし, 1画面に同時に表示できるのは 64 種類 …… 同一 CROMBANK 内)。



1画面にCROMBANK0とCROMBANK1に設定してあるキャラクタの表示はできません。

- (6) 縦、横独立におのおの4サイズ文字設定可
ただし、縦サイズは1行ごとの設定、横サイズは文字ごとに設定可能（コントロール・コードによる^{注1}）。
- (7) 10×15ドットの文字構成
文字と文字の間に隙間なし。^{注2}
- (8) キャラクタ・パターン・データはプログラム・メモリ空間に配置
キャラクタ・パターン・データが少ない場合、CROM領域もプログラム・エリアとして使用できます。
- (9) キャラクタ・データは、データ・メモリ空間に配置
データ転送、データ読み出し/書き込みは通常のデータ・メモリのデータと同じに扱えます。

注1. コントロール・データは1行に3つまで指定できます。

2. 文字と文字の間に隙間を設けないため2つ以上のキャラクタを合わせて漢字などの文字や図形の表示を行うことができます。

20.2 DMA

DMA は、Direct Memory Access の略であり、CPU を介さずに、メモリの内容を直接周辺装置に転送するモードです。

μPD17062 では、DMA モードは IDC を動作させるときに使用します。

μPD17062 のインストラクション・サイクルは 2 μs ですが、DMA モード中は見かけ上、インストラクション・サイクルは 12 μs となります。これは、実際のインストラクション・サイクルが 12 μs になるのではなく、IDC 用のデータ転送のため 5 インストラクション (10 μs) を使用し、残りの 2 μs、つまり 1 インストラクションで通常の命令を実行するためです。DMA モード中は、5 インストラクション・サイクル飛びに通常のインストラクションを実行することになります。

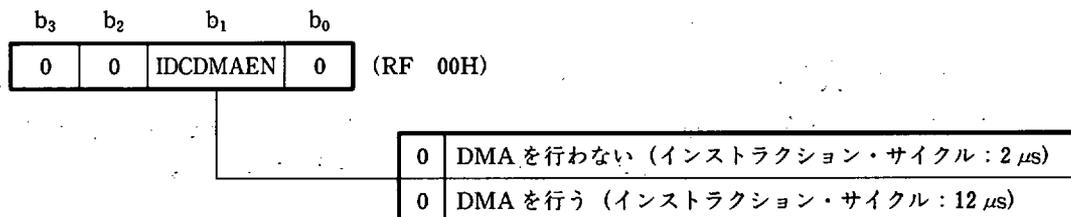
このため、IDC 使用時は、見かけ上の命令処理時間は、12 μs となります。プログラムの都合上、12 μs インストラクション・サイクルと 2 μs インストラクション・サイクルが混在して問題になる場合は、そのルーチンを実行するときに、DMA モードのみを設定し、IDC の動作を停止しておいてください。この場合、IDC 用データの転送のための 5 インストラクションは、クロックのみ動作し、μPD17062 自身は何の処理もしません。

また、DMA モードになると、6 命令中の 5 命令の ROM のアドレスは、プログラム・カウンタではさされず CROM アドレス・ポインタでさされ、また RAM のアドレスは VRAM アドレス・ポインタでさされるようになります。

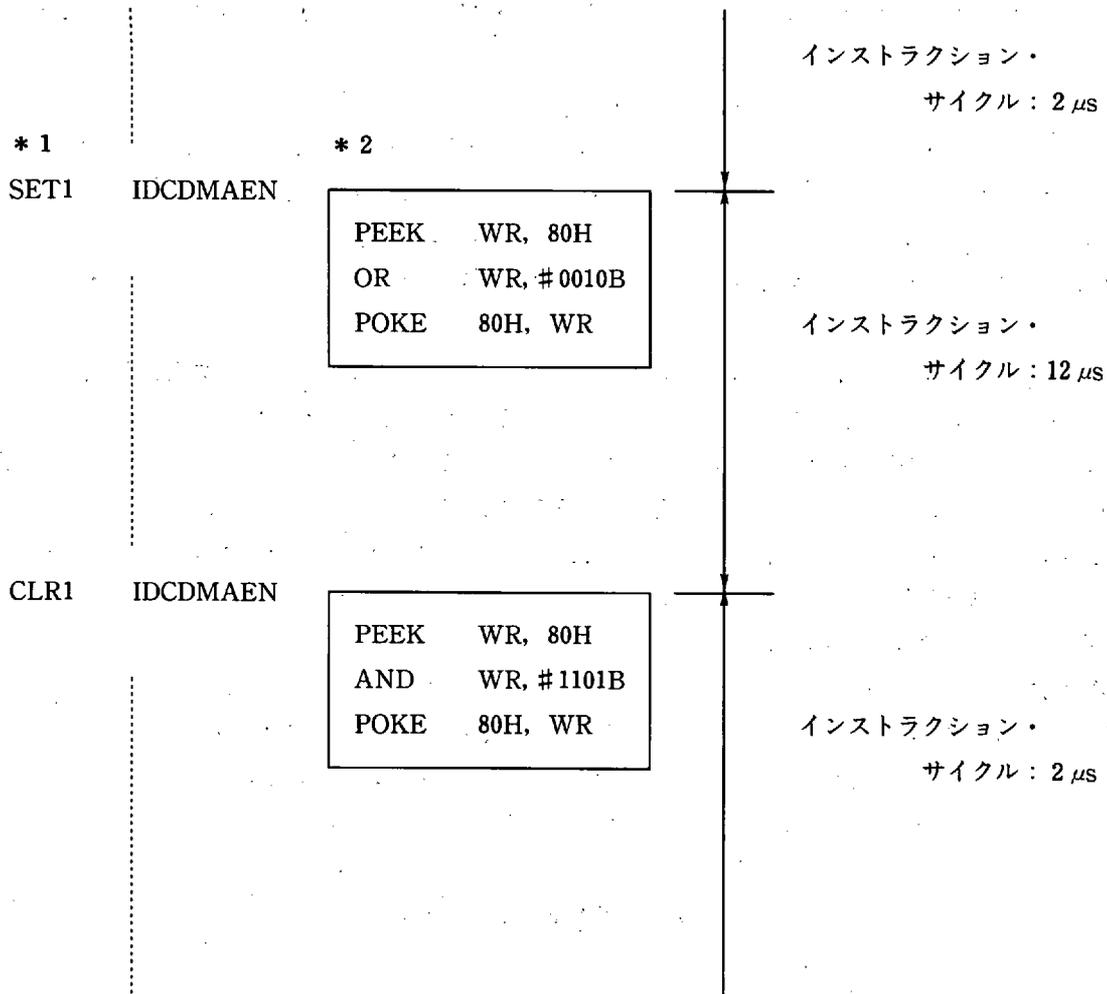
DMA を行うには、IDCDMAEN フラグを操作します。

IDCDMAEN フラグはレジスタ・ファイル上に位置されており、読み書き可能な 1 ビットのフラグです。このフラグをセットすると DMA リクエストは、どの割り込みより優先されて受け付けられ、DMA モードに移行します。このフラグをリセットすると DMA リクエストは、受け付けられなくなり、また DMA モード中にリセットすると、リセットした命令の次のインストラクション・サイクルから DMA は解除されます。

表 20-1 IDCDMAEN フラグ



プログラム例

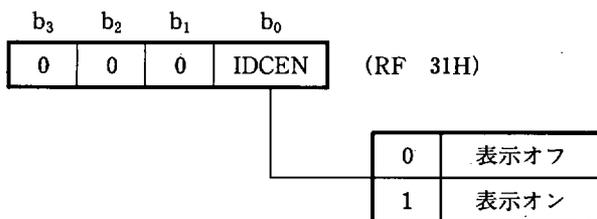


備考 “SET1”や“CLR1”はμPD17062の命令ではありません。これらは17Kシリーズのアセンブラで用意されている組み込みマクロ命令で、1ビットのフラグのセット/リセットを行います。*1のようにソース・プログラム上に記載し、アセンブルを行うと*2のように展開されます。

20.3 IDC イネーブル・フラグ

IDC の動作を開始 (表示オン) させるためには, IDCEN フラグを操作します。IDCEN (IDCイネーブル) フラグは, レジスタ・ファイルの 31H の最下位ビット (#0) に割り当てられています。

表 20-2 IDCEN フラグ



(1) 表示オン時の注意事項

- (a) IDCEN に “1” をセットするとき (表示開始) は, IDCDMAEN フラグ (RF, 00H, #1) をオンしたあとの垂直同期信号 (Vsync) がハイ・レベルの間 (垂直帰線間: $\overline{\text{Vsync}}$ =ロウ・レベル) に行わなければなりません。
- (b) VRAM のデータ設定は, 必ず IDCEN が “0” のとき (表示オフ) に設定するようにしてください。

プログラム例

```

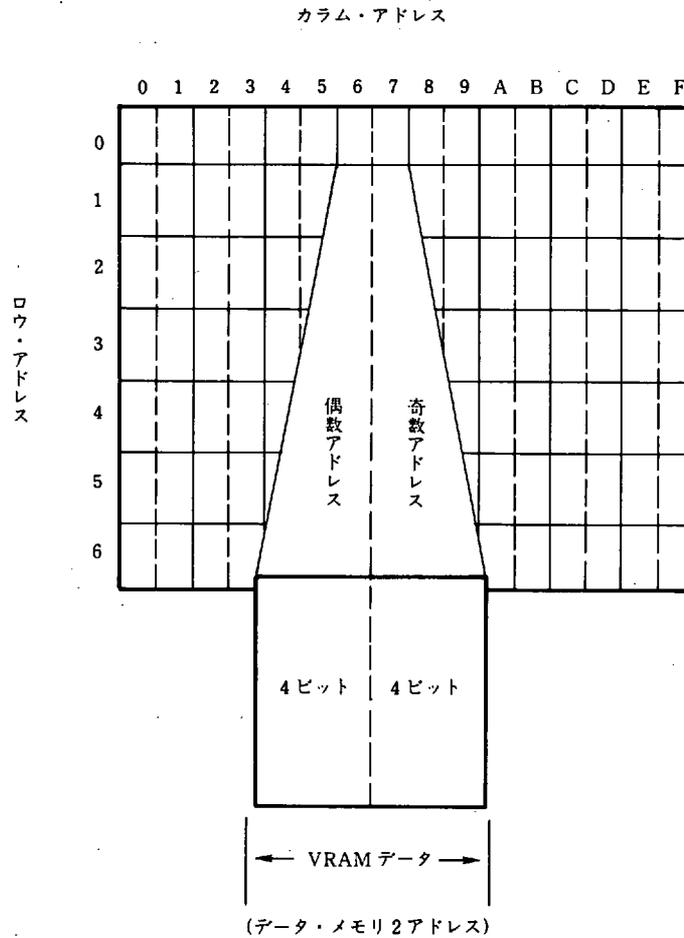
SET1    IDCDMAEN    ; DMA モードにセット
      ⋮
CLR1    IDCEN      ; 表示オンの状態から VRAM データを設定する場合は、
      ⋮              ; 一度 IDCEN をリセットします (表示オフ)。
      ⋮
      [ VRAM にデータをセット ] ; VRAM のデータを設定します。
      ⋮
LOOP
SKF1    INTVSYN    ;  $\overline{\text{Vsync}}$ =ロウ・レベルを確認し, IDCEN をセットします。
BR      LOOP
SET1    IDCEN      ; 表示オン
      ⋮
    
```

20.4 VRAM

VRAMとは、IDCによりテレビなどの画面に写し出される画像パターンを選択するためのデータが格納されているメモリです。μPD17062では、VRAMデータはデータ・メモリのBANK1, BANK2に配置されており、データ・メモリの隣り合った偶数アドレスと奇数アドレスの2アドレス（8ビット）で1つのデータになります。

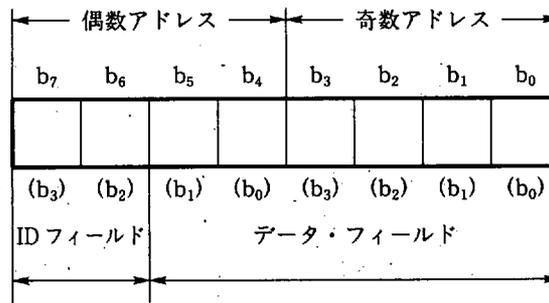
BANK1, BANK2は、おのおの112ニブル（合計224ニブル：224×4ビット）のデータ・メモリが配置されています。つまり、VRAMデータとしては112データが設定可能です。

図 20-1 VRAM の構成



VRAMデータは8ビットで構成されます。8ビット中、上位2ビットはIDフィールドと呼び、VRAMのデータの種類を示します。下位6ビットは、データ・フィールドと呼び、実際に表示する表示データや、制御データです。

図 20-2 VRAM データの構成



20.4.1 ID フィールド

ID フィールドは、データ・フィールドのデータの種類を示します。
 データ・フィールドに設定するデータは、次の3種類があります。

- (1) キャラクタ・パターン選択データ
- (2) キャリッジ・リターン・データ
- (3) コントロール・データ選択データ

表 20-3 ID フィールド

ID フィールド		データ・フィールドのデータの種類
b ₇	b ₆	
0	0	キャラクタ・パターン選択データ
0	1	BANK1 へのキャリッジ・リターン (戻り番地) (BANK1 から BANK1, および BANK2 から BANK1)
1	0	コントロール・データ
1	1	BANK2 へのキャリッジ・リターン (戻り番地) (BANK2 から BANK2)

20.4.2 キャラクタ・パターン選択データ

キャラクタ・パターンとは、テレビなどの画面に写し出される画像自体のデータであり、プログラム・メモリ上の CROM 領域 (プログラム・メモリ上の 0800H - 0F7FH) に配置されています。キャラクタ・パターン選択データは、CROM 上のアドレスの一部 (b₉ - b₄) となります。すなわちデータ・フィールドのデータ 6 ビットが CROM のアドレス b₉ - b₄ を指し示します。

ただし、CROM は BANK0 と BANK1 に分かれており、レジスタ・ファイル上の CROMBNK (30H, b₀) の値によって、6 ビットの VRAM データが同じでも指し示す CROM アドレスが異なりますので注意してください。つまり、もしデータ・フィールドのデータが 0 (000000B) であれば CROM のアドレスは 080×H (10000000××××B) または 0C0×H (11000000××××B) を指し示します。080×H か 0C0×H かは CROM の BANK を指定することにより決まります。VRAM キャラクタ・パターン選択データが実際に指し示す CROM アドレスを表 20-4 に示します。

表 20-4 VRAM データ (キャラクタ・パターン選択データ) と CROM アドレスの対応表

VRAM データ (8ビット)	CROM アドレス		VRAM データ (8ビット)	CROM アドレス	
	BANK0	BANK1		BANK0	BANK1
00H	0800H-080EH	0C00H-0C0EH	20H	0A00H-0A0EH	0E00H-0E0EH
01H	0810H-081EH	0C10H-0C1EH	21H	0A10H-0A1EH	0E10H-0E1EH
02H	0820H-082EH	0C20H-0C2EH	22H	0A20H-0A2EH	0E20H-0E2EH
03H	0830H-083EH	0C30H-0C3EH	23H	0A30H-0A3EH	0E30H-0E3EH
04H	0840H-084EH	0C40H-0C4EH	24H	0A40H-0A4EH	0E40H-0E4EH
05H	0850H-085EH	0C50H-0C5EH	25H	0A50H-0A5EH	0E50H-0E5EH
06H	0860H-086EH	0C60H-0C6EH	26H	0A60H-0A6EH	0E60H-0E6EH
07H	0870H-087EH	0C70H-0C7EH	27H	0A70H-0A7EH	0E70H-0E7EH
08H	0880H-088EH	0C80H-0C8EH	28H	0A80H-0A8EH	0E80H-0E8EH
09H	0890H-089EH	0C90H-0C9EH	29H	0A90H-0A9EH	0E90H-0E9EH
0AH	08A0H-08AEH	0CA0H-0CAEH	2AH	0AA0H-0AAEH	0EA0H-0EAEH
0BH	08B0H-08BEH	0CB0H-0CBEH	2BH	0AB0H-0ABEH	0EB0H-0EBEH
0CH	08C0H-08CEH	0CC0H-0CCEH	2CH	0AC0H-0ACEH	0EC0H-0ECEH
0DH	08D0H-08DEH	0CD0H-0CDEH	2DH	0AD0H-0ADEH	0ED0H-0EDEH
0EH	08E0H-08EEH	0CE0H-0CEEH	2EH	0AE0H-0AEEH	0EE0H-0EEEH
0FH	08F0H-08FEH	0CF0H-0CFEH	2FH	0AF0H-0AFEH	0EF0H-0EFEH
10H	0900H-090EH	0D00H-0D0EH	30H	0B00H-0B0EH	0F00H-0F0EH
11H	0910H-091EH	0D10H-0D1EH	31H	0B10H-0B1EH	0F10H-0F1EH
12H	0920H-092EH	0D20H-0D2EH	32H	0B20H-0B2EH	0F20H-0F2EH
13H	0930H-093EH	0D30H-0D3EH	33H	0B30H-0B3EH	0F30H-0F3EH
14H	0940H-094EH	0D40H-0D4EH	34H	0B40H-0B4EH	0F40H-0F4EH
15H	0950H-095EH	0D50H-0D5EH	35H	0B50H-0B5EH	0F50H-0F5EH
16H	0960H-096EH	0D60H-0D6EH	36H	0B60H-0B6EH	0F60H-0F6EH
17H	0970H-097EH	0D70H-0D7EH	37H	0B70H-0B7EH	0F70H-0F7EH
18H	0980H-098EH	0D80H-0D8EH	38H	0B80H-0B8EH	設定禁止
19H	0990H-099EH	0D90H-0D9EH	39H	0B90H-0B9EH	
1AH	09A0H-09AEH	0DA0H-0DAEH	3AH	0BA0H-0BAEH	
1BH	09B0H-09BEH	0DB0H-0DBEH	3BH	0BB0H-0BBEH	
1CH	09C0H-09CEH	0DC0H-0DCEH	3CH	0BC0H-0BCEH	
1DH	09D0H-09DEH	0DD0H-0DDEH	3DH	0BD0H-0BDEH	
1EH	09E0H-09EEH	0DE0H-0DEEH	3EH	0BE0H-0BEEH	
1FH	09F0H-09FEH	0DF0H-0DFEH	3FH	0BF0H-0BFEH	

プログラム例

VRAM データ

	0	1	2	3	4	5	6	7	8	9	A	B
0	8	0	0	0	0	1	4	0				
1												

CROM データ

0800H

“C”

080FH ; コントロール・データ 1

0810H

“H”

081FH ; コントロール・データ 2

0C00H

“V”

0C0FH ; コントロール・データ 1

0C10H

“O”

0C1FH ; コントロール・データ 2

上記のように CROM データと VRAM データを設定しておいた場合、CROM のバンクにより画面に表示される内容が異なります。

CROM のバンクはレジスタ・ファイル上の CROMBNK (30H, b₀) で指定します。

上記の例では次のようになります。

(1) CROMBNK = 0

画面上には、“CH” の文字が表示されます。この場合のコントロール・データは、“コントロール・データ 1” の内容になります。

(2) CROMBNK = 1

画面上には、“VO” の文字が表示されます。この場合のコントロール・データは、“コントロール・データ 1” の内容になります。

20.4.3 キャリッジ・リターン・データ

画面に表示している行の先頭キャラクタを指定している VRAM データのアドレスを指定するデータのことをキャリッジ・リターン・データと呼びます。

キャリッジ・リターン・データによって表示 1 行の終了を意味します。

キャリッジ・リターン・データが 2 回続くと 1 画面の終了を意味します。

キャリッジ・リターン・データには、BANK1 へのキャリッジ・リターンと BANK2 へのキャリッジ・リターンの 2 種類があります。データ・フィールドのキャリッジ・リターンが BANK1 であるか BANK2 であるかは ID フィールドのデータで決まります。ID フィールドが 01B であれば BANK1 へのキャリッジ・リターン、11B であれば BANK2 へのキャリッジ・リターンです。

キャリッジ・リターン・データは 6 ビットで構成され上位 3 ビットは VRAM のロウ・アドレスを指し、下位 3 ビットは VRAM のカラム・アドレスの上位 3 ビットを指します。VRAM のカラム・アドレスの最下位ビットは“0”に固定されています。したがってキャリッジ・リターン・データが 010011B であれば、VRAM のロウ・アドレスが 010B (2H)、カラム・アドレスが 0110B (6H)、すなわち 26H へのリターン・データを意味します。

図 20-3 キャリッジ・リターン・データの構成

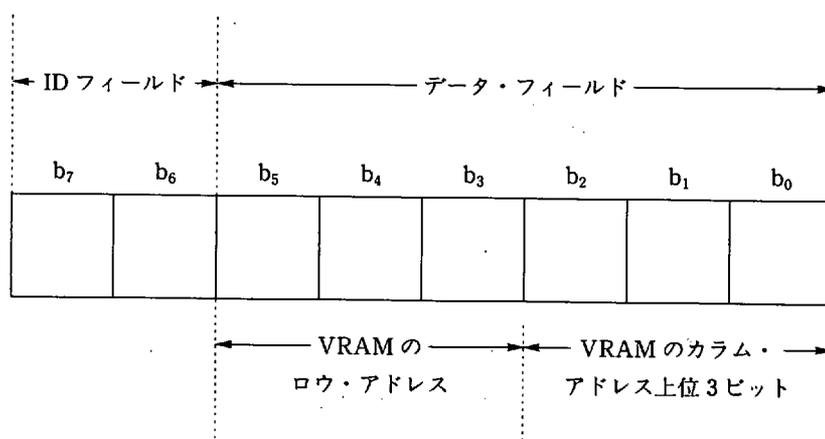


図 20-4 キャリッジ・リターン・データ一覧
(ID フィールドを含む 8 ビット)

BANK 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	40		41		42		43		44		45		46		47	
1	48		49		4A		4B		4C		4D		4E		4F	
2	50		51		52		53		54		55		56		57	
3	58		59		5A		5B		5C		5D		5E		5F	
4	60		61		62		63		64		65		66		67	
5	68		69		6A		6B		6C		6D		6E		6F	
6	70		71		72		73		74		75		76		77	

BANK 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C0		C1		C2		C3		C4		C5		C6		C7	
1	C8		C9		CA		CB		CC		CD		CE		CF	
2	D0		D1		D2		D3		D4		D5		D6		D7	
3	D8		D9		DA		DB		DC		DD		DE		DF	
4	E0		E1		E2		E3		E4		E5		E6		E7	
5	E8		E9		EA		EB		EC		ED		EE		EF	
6	F0		F1		F2		F3		F4		F5		F6		F7	

20.4.4 コントロール・データ選択データ

コントロール・データとは、キャラクタ・パターン画面上での文字サイズ、表示位置、色を指定するためのデータで、CROM 中（アドレスが×××FH）に格納されています。

CROM 中のどのコントロール・データを選択するかを VRAM 上で指定するのがコントロール・データ選択データです。

データ・フィールドの 6 ビットは CROM アドレスの $b_9 - b_4$ に対応します。またキャラクタ・パターン選択データの時と同様に CROM バンクの指定が必要です。CROM バンクの指定は、レジスタ・ファイル上の CROMBNK (30H, b_0) で行います。

図 20-5 コントロール・データと CROM アドレスの関係

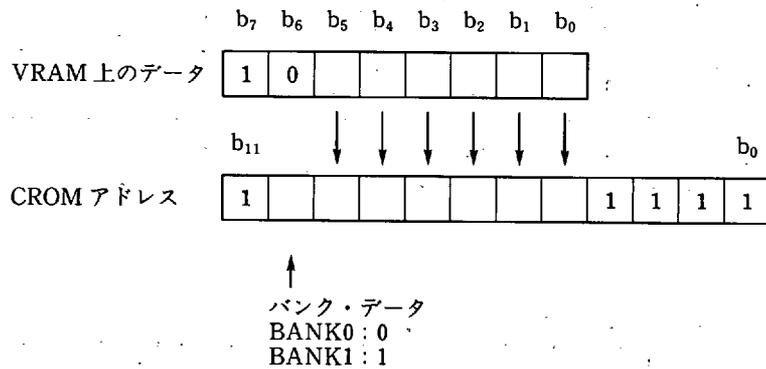


表 20-5 VRAM データ (コントロール・データ選択データ) と CROM アドレスの対応表

VRAM データ (8ビット)	CROM アドレス		VRAM データ (8ビット)	CROM アドレス	
	BANK0	BANK1		BANK0	BANK1
80H	080FH	0C0FH	A0H	0A0FH	0E0FH
81H	081FH	0C1FH	A1H	0A1FH	0E1FH
82H	082FH	0C2FH	A2H	0A2FH	0E2FH
83H	083FH	0C3FH	A3H	0A3FH	0E3FH
84H	084FH	0C4FH	A4H	0A4FH	0E4FH
85H	085FH	0C5FH	A5H	0A5FH	0E5FH
86H	086FH	0C6FH	A6H	0A6FH	0E6FH
87H	087FH	0C7FH	A7H	0A7FH	0E7FH
88H	088FH	0C8FH	A8H	0A8FH	0E8FH
89H	089FH	0C9FH	A9H	0A9FH	0E9FH
8AH	08AFH	0CAFH	AAH	0AAFH	0EAFH
8BH	08BFH	0CBFH	ABH	0ABFH	0EBFH
8CH	08CFH	0CCFH	ACH	0ACFH	0ECFH
8DH	08DFH	0CDFH	ADH	0ADFH	0EDFH
8EH	08EFH	0CEFH	AEH	0AEFH	0EEFH
8FH	08FFH	0CFFH	AFH	0AFFH	0EFFH
90H	090FH	0D0FH	B0H	0B0FH	0F0FH
91H	091FH	0D1FH	B1H	0B1FH	0F1FH
92H	092FH	0D2FH	B2H	0B2FH	0F2FH
93H	093FH	0D3FH	B3H	0B3FH	0F3FH
94H	094FH	0D4FH	B4H	0B4FH	0F4FH
95H	095FH	0D5FH	B5H	0B5FH	0F5FH
96H	096FH	0D6FH	B6H	0B6FH	0F6FH
97H	097FH	0D7FH	B7H	0B7FH	0F7FH
98H	098FH	0D8FH	B8H	0B8FH	設定禁止
99H	099FH	0D9FH	B9H	0B9FH	
9AH	09AFH	0DAFH	BAH	0BAFH	
9BH	09BFH	0DBFH	BBH	0BBFH	
9CH	09CFH	0DCFH	BCH	0BCFH	
9DH	09DFH	0DDFH	BDH	0BDFH	
9EH	09EFH	0DEFH	BEH	0BEFH	
9FH	09FFH	0DFFH	BFH	0BFFH	

20.4.5 VRAM データ設定時の注意点

- (1) VRAM データ設定時は必ず IDCEN フラグを“0”にしてから行ってください。
- (2) VRAM データの先頭は必ず BANK1 の 00H から行います。
- (3) VRAM データは、BANK1, BANK2 の 7×H には設定しないでください。
- (4) 1画面の先頭は必ずコントロール・データを設定してください。またプログラムミスを防ぐために、各行の先頭は、コントロール・データを設定するようにしてください。もし設定しなかった場合は、直前に設定されたデータが有効になります。
- (5) データの設定
 - (a) キャラクタ・パターン選択データは、画面の左上から表示する順に、VRAM の若いアドレスから順に設定します。
 - (b) コントロール・データは、1行に3回までしか使用できません。
 - (c) コントロール・データで修飾されるのは、コントロール・データ選択データ以後のキャラクタ・パターン・データです。

ただし、水平スタート位置データ、垂直スタート位置データは、そのコントロール・データ選択データのすぐあとのキャラクタに対してのみで、そのあとのキャラクタは続けて出力されます。
 - (d) 1行の最後は、必ずキャリッジ・リターン・データを設定してください。
 - (e) 1画面のデータの最後は必ずキャリッジ・リターン・データを2つ続けます。

20.5 CROM (キャラクタ ROM)

CROM データには IDC のパターン・データとコントロール・データを格納しています。CROM はプログラム・メモリと共用しており、容量は 2K ステップ (1920×16ビット) あります。CROM として使用していないエリアは、通常のプログラム・エリアとして使用できます。

ROM 上の CROM エリアは 0800H - 0F7FH 番地です。ただし、CROM エリアは CROM BANK0 と、CROM BANK1 に分割されており、CROM として使用するときのみバンクの概念が必要です。ただし、プログラム・エリアとして使用するときにはバンクの概念は必要ありません。CROM BANK0 は 0800H - 0BFFH の 1K ワードで、CROM BANK1 は、0C00H - 0F7FH の 896 ワードとなります。

CROM のバンク切り替えは、レジスタ・ファイル上の CROMBNK フラグ (30H, b₀) で行います。

表 20-6 CROM バンク

CROMBNK フラグ	CROM バンク	CROM アドレス
0	BANK0	0800H-0BFFH
1	BANK1	0C00H-0F7FH

備考 CROM のバンク切り替えは、IDCEN フラグが“0”のときに行うようにします。

レジスタ・ファイルの 30H 番地は読み書き可能ですが、CROMBNK フラグ (b₀) 以外は常に“0”となります。CROM データはプログラム・メモリ領域に配置されているため、データ長は16ビットになります。CROM データには、次の2種類があります。

- (1) キャラクタ・パターン・データ
- (2) コントロール・データ

20.5.1 キャラクタ・パターン・データ

文字や図形のパターン用データです。実際の1キャラクタは横10ドット、縦15ドットで構成され、そのキャラクタ・パターン・データは、16ビット×15ステップで構成されます。横10ドットのデータが CROM1 ステップに対応します。CROM 上でアドレス ××0H-××EH の15ステップが1組のキャラクタ・パターン・データとなります。またキャラクタ・パターン・データにはふちどりのありなしでデータの構成が異なります。

図 20-6 にキャラクタ・パターン・データの構造を示します。

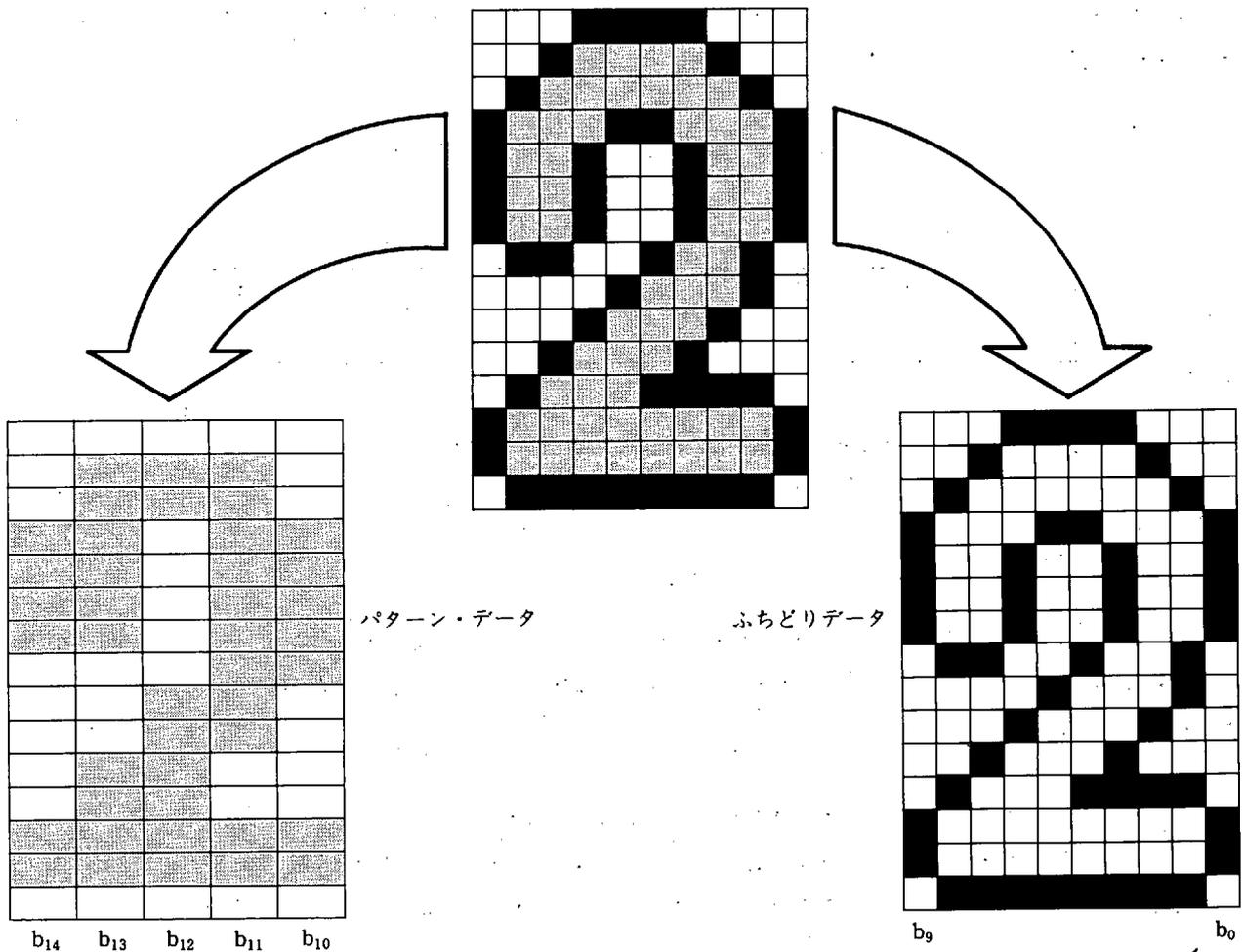
ふちどりのありなしを、最上位ビットで選択します。ふちどりが無い場合は“0”に、ある場合は“1”に設定します。

ふちどりが無い場合は、下位10ビットに実際に表示するキャラクタ・パターンのドット・イメージで設定します。b₉ が表示画面の左に対応し、b₀ が右に対応します。点灯するドットに対応するビットを“1”に、消灯するドットに対応するビットを“0”に設定します。

ふちどりが有る場合は、図 20-6 のようにキャラクタ・パターン・データは5ビットとなります。このとき実際の表示パターンの2ドットがキャラクタ・パターン・データ1ビットに対応します。それにふちどりデータ10ビットを重ねて (ふちどりデータは1ドット単位で設定) ふちどりが有る場合のキャラクタ・パターンを形成します。

17K シリーズのアセンブラを使用すれば、DCP 疑似命令を用いて容易にキャラクタ・パターンを定義することができます。DCP 疑似命令を使用すれば、ふちどりのありなしに関わらず自動的に図 20-6 に示すデータを生成します。

図 20-8 ふちどりがあある場合のキャラクタ・パターン例



	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀	b ₉	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
× × × 0 H	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	
× × × 1 H	1	0	1	1	1	0	0	0	1	0	0	0	0	1	0	
× × × 2 H	1	0	1	1	1	0	0	1	0	0	0	0	0	0	1	
× × × 3 H	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	
× × × 4 H	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	
× × × 5 H	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	
× × × 6 H	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	
× × × 7 H	1	0	0	0	1	0	0	1	1	0	0	1	0	0	1	
× × × 8 H	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1	
× × × 9 H	1	0	0	1	1	0	0	0	0	1	0	0	0	1	0	
× × × A H	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	
× × × B H	1	0	1	1	0	0	0	1	0	0	0	1	1	1	1	
× × × C H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	
× × × D H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	
× × × E H	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	

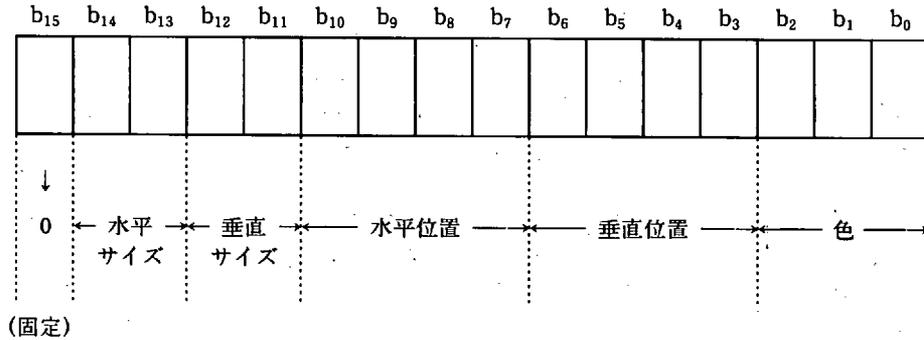
ROM アドレス
(CROM
エリア内)

↑ パターン・データ ふちどりデータ
"1" (ふちどりがあある)

20.5.2 コントロール・データ

コントロール・データは、キャラクタ・パターンの表示位置、サイズ、色を指定するためのデータで、CROM領域の×××FHに格納します。1データは16ビットで構成され最上位ビットは必ず“0”にします。コントロール・データの構成は図20-9のようになります。

図20-9 コントロール・データの構造



コントロール・データは各キャラクタ・パターン・データの間についていますが、コントロール・コードの前後のアドレスにあるキャラクタ・パターン・データとは何ら関係なく、VRAMに設定されるデータによりどのコントロール・データでも指定できます。

(1) 水平サイズ・データ (コントロール・データの b_{14} , b_{13} の2ビット)

水平サイズ・データは、キャラクタ画像の水平方向の大きさを決定するデータです。各キャラクタごとに4種類のサイズが設定できます (ただし1行に3回まで)。設定内容は表20-7のようになります。

表20-7 水平サイズの設定

水平サイズ・データ		大きさ	1キャラクタの 横方向幅	1行の最大 表示キャラクタ数
b_{14}	b_{13}			
0	0	1倍	2.5 μ S	16
0	1	2倍	5.0 μ S	8
1	0	3倍	7.5 μ S	5
1	1	4倍	10.0 μ S	4

(2) 垂直サイズ (コントロール・データの b_{12} , b_{11} の 2 ビット)

垂直サイズはキャラクタ画像の垂直方向の大きさを決定するデータです。各行ごとに 4 種類の大きさが設定できます。設定内容は表 20-8 のようになります。

垂直サイズ・データは、行の先頭で指定された垂直サイズ・データがその行では有効になり、その行中で他のコントロール・データの垂直サイズ・データは無視されます。

表 20-8 垂直サイズの設定

垂直サイズ・データ		大きさ	1 キャラクタ の縦方向幅 (インタレース)	縦方向の 最大表示 キャラクタ数
b_{12}	b_{11}			
0	0	1 倍	15H	12
0	1	2 倍	30H	6
1	0	3 倍	45H	4
1	1	4 倍	60H	3

(3) 水平位置データ (コントロール・データの b_{10} - b_7 の 4 ビット)

水平位置データは、図 20-10 の横方向 16 桁のうち、どの位置から表示させるかを決定するデータです。実際の表示位置は横方向 19 文字ですが、水平表示開始位置は画面の左から 16 文字以内でしか指定できません。

行の初めは絶対桁で (図 20-10 の 0-15 桁) で指定します。データとしては、 b_{10} を MSB とし、 b_7 を LSB とする 4 ビットのデータであり、水平位置データは 0H-FH の値となります。水平位置データは、0 桁目が "0" で、15 桁目が "FH" に対応します。

1 行中でキャラクタとキャラクタの間を開けるときは開けるキャラクタ数だけ、水平位置データで指定します。つまり、今表示しているキャラクタの何キャラクタ後に表示するかを 16 進数で設定します。

たとえば、図 20-10 の "A", "C" の水平位置データは、"A" は 8H になり、"C" は 1H となります。"C" のコントロール・データの水平位置データを "0" にすると、"C" は 9 桁目に表示されます。また "A" の次にコントロール・データを使用しなければ、"C" は 9 桁目に表示されます。

備考 ここでいう何キャラクタ分というのは、水平サイズ・データが "00" のときの例で、水平サイズ・データを変えれば、そのサイズのキャラクタ数で数えることとなります。

たとえば、水平サイズ・データを 2 倍に指定すれば、1 行は 8 桁になってしまいます。

(4) 垂直位置データ (コントロール・データの $b_6 - b_3$ の4ビット)

垂直位置データは、図 20-10 の縦方向12行のうちどの位置から表示させるか決定するデータです。データとしては、 b_6 を MSB とし、 b_3 を LSB とする 4 ビットのデータで、0H-DH で指定します (EH, FH のデータは設定しないでください)。設定データは 0 行目が "0" に、13 行目が "DH" に対応します。

画面初めのキャラクタは絶対行 (図 20-10 の 0-11 行) で指定します。行と行の間を開けるときは開ける行数分だけ垂直位置データで指定します。つまり今出しているキャラクタの何行後に表示するかを 16 進数で設定します。

たとえば図 20-10 の "A" の垂直位置データは 6H となり、"B" の垂直位置データは 1H となります。また "D" の垂直位置データは "0H" と設定します。

備考 1 行というのは垂直サイズ・データが "00" の 1 倍のときの例で垂直サイズ・データが変われば、そのサイズでカウントされてしまいます。たとえば、垂直サイズ・データを 2 倍にすれば、1 画面は 6 行になってしまいます。

図 20-10 表示位置

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	桁															
0 行																
1 行																
2 行																
3 行																
4 行																
5 行																
6 行									A		C					
7 行																
8 行				B												
9 行					D											
10 行																
11 行																
12 行																
13 行																

(5) 色データ (コントロール・データの $b_2 - b_0$ の3ビット)

色データはキャラクタの色を指定するデータで、指定された出力端子 (R, G, B 端子) からデータが出力されます。色データと各端子との対応を表 20-9 に示します。

また、各色データの設定と実際に出力される色の関係は表 20-10 のようになります。

表 20-9 色データ

b_2	b_1	b_0
R	G	B

表 20-10 キャラクタの色

色データ			キャラクタの色
R	G	B	
0	0	0	黒
0	0	1	青
0	1	0	緑
0	1	1	シアン
1	0	0	赤
1	0	1	マゼンタ
1	1	0	黄
1	1	1	白

20.5.3 アセンブラでの定義方法

17 K シリーズのアセンブラでは DCP 疑似命令を用いて表示パターンを容易に定義することができます。以下に DCP 疑似命令の記述方法を説明します。

(1) 記述形式

シンボル欄	ニモニック欄	オペランド欄	コメント欄
[レーベル:]	DCP	式, '表示パターン'	[; コメント]

(2) 説明

(a) 式は“0”または“1”の値をとり、第2オペランドで記述された表示パターンにふちどりをを行うか否かを指定します。

“0” : ふちどりを行わない

“1” : ふちどりをを行う

式の評価値が“0”または“1”以外の場合はエラーとなります。

(b) 表示パターンは、“0”、“#”、“ ” (空白) の3種類の文字のみを使用し、10文字記述します。

これら3種類以外の文字を記述した場合や、記述した文字数が10文字でない場合は、いずれもエラーとなります。これら3種類の文字はそれぞれ表示パターンの1ドットに相当し、次のように対応します。

“0” : 点灯

“#” : ふちどり

“ ” : ブランク

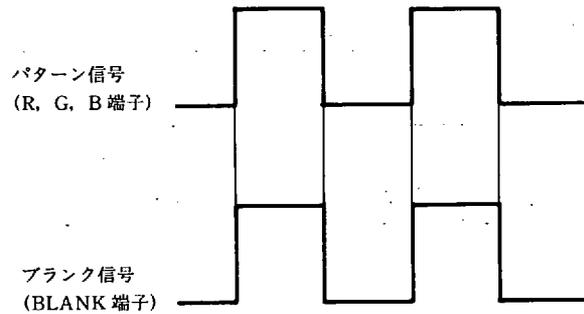
なお、第1オペランドの式の評価値が“0”のときは表示パターンに“#”を使用することはできません。

20.6 BLANK, R, G, B 端子

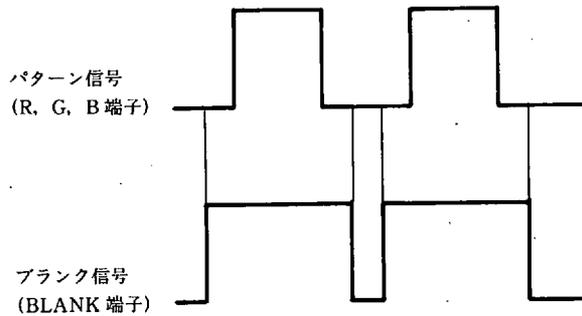
いずれも CMOS プッシュプル形式の出力端子で、アクティブ・ハイの信号が出力されます。BLANK 端子からは、放送画像を消す信号を出力し、R, G, B 端子からは、キャラクタ・パターン・データが出力されます。ふちどりのときは、BLANK 信号とキャラクタ・パターン信号 (R, G, B 信号の OR 信号) は同じ信号が出力され、ふちどりのないときは、BLANK 信号はキャラクタ・パターン信号を囲む波形が BLANK 端子より出力されます。

図 20-11 IDC の出力波形

(a) ふちどりのない場合



(b) ふちどりの場合

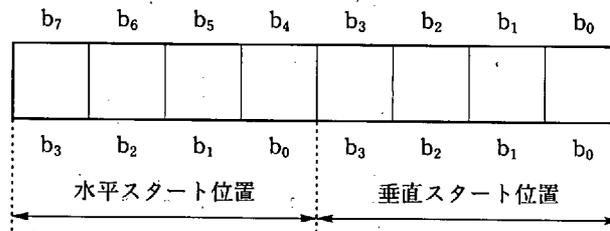


20.7 表示開始位置の設定

IDCの表示開始位置(画面の左上)は、IDCスタート位置設定レジスタにデータを設定することにより、水平方向、垂直方向にそれぞれ16カ所に設定できます。すなわち画面全体の表示位置を移動させることができます。IDCスタート位置設定レジスタは垂直スタート位置設定レジスタと水平位置スタート位置設定レジスタからなり、それぞれ4ビットずつ、合計8ビットで構成されています。IDCスタート位置設定レジスタは、周辺アドレス01Hが割り当てられており“GET”や“PUT”命令によってデータの設定や確認を行います。

ただしIDCスタート位置設定レジスタの設定はIDCENフラグが“0”のときに行ってください。

図 20 - 12 IDC スタート位置設定レジスタの構成



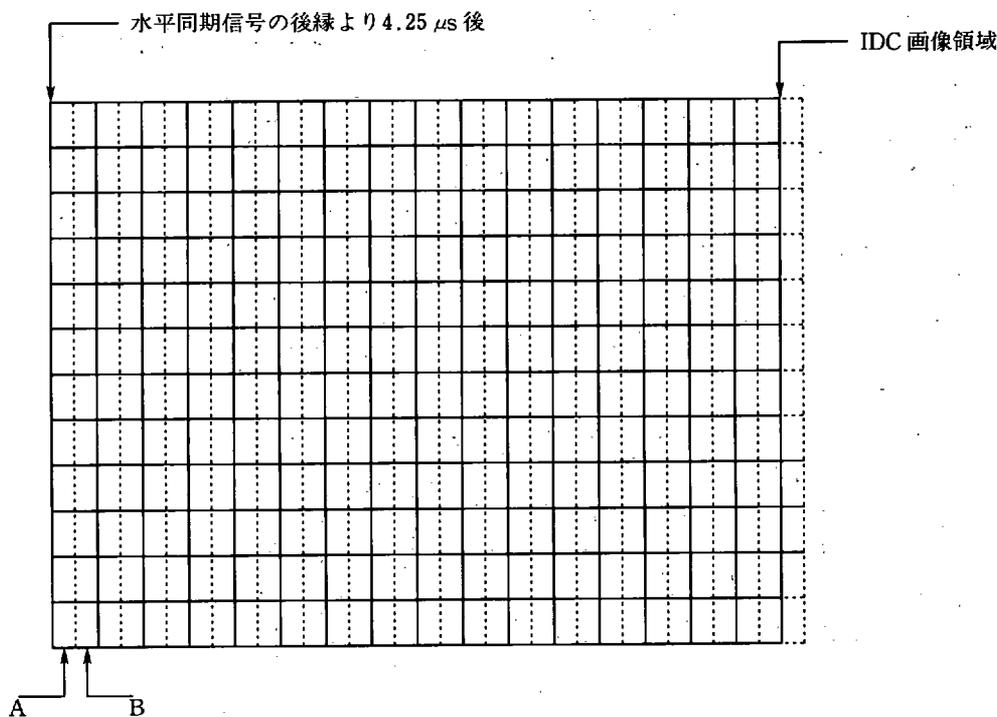
20.7.1 水平スタート位置設定レジスタ

水平スタート位置設定レジスタに設定するデータが“0H”のとき水平スタート位置は水平同期信号の後縁より4.25 μs後に設定されます。このデータが“1”増えるごとに水平スタート位置は右に250 nsずれます。つまり次式のように表せます。

$$\text{水平スタート位置} = 4.25 \mu\text{s} + 250 \text{ ns} \times (\text{水平スタート位置設定データ})$$

図 20-13 において、水平スタート位置設定データが“0H”のときの位置がAで、水平スタート位置設定データを“1”にすると右へ250 ns（最小サイズのキャラクタの1ドット分）移動し、Bの位置となります（データが“0”のときの画面を実線で、データが“1”のときの画面を点線で示します）。

図 20-13. 水平方向の移動



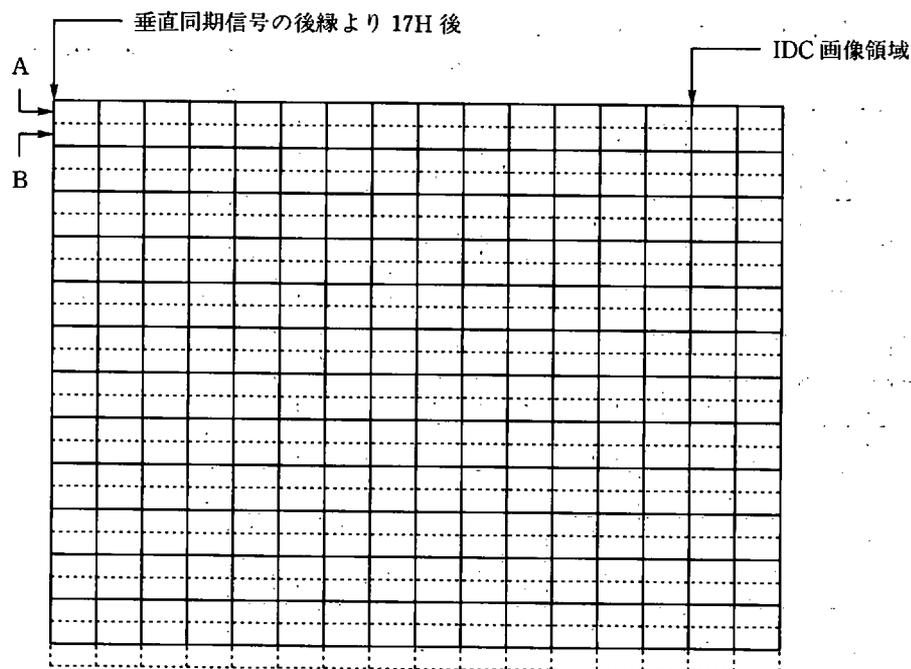
20.7.2 垂直スタート位置設定レジスタ

垂直スタート位置設定レジスタに設定するデータが“0H”のとき、垂直スタート位置は垂直同期信号の後縁より17H（インタレース）後に設定されます。このデータが“1”増えるごとに垂直スタート位置は下に1Hずれます。つまり次式のように表せます。

$$\text{垂直スタート位置} = 17H + 1H \times (\text{垂直スタート位置設定データ})$$

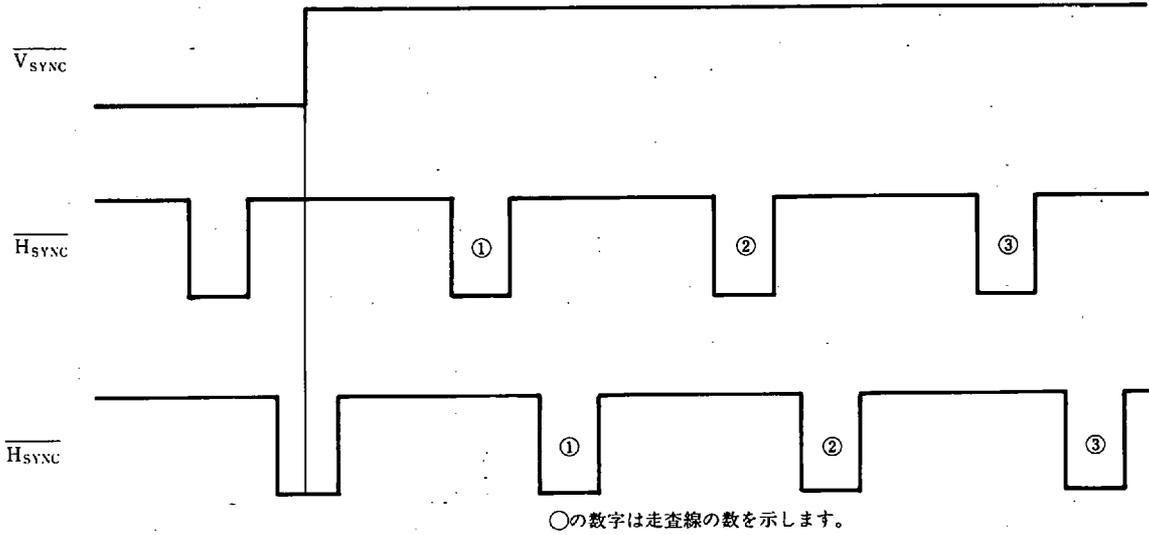
図 20-14 において、垂直スタート位置設定データが“0H”のときの位置がAで、垂直スタート位置設定データを“1”にすると下へ1H移動し、Bの位置となります（データが“0”のときの画面を実線で、データが“1”のときの画面を点線で示します）。

図 20-14 垂直方向の移動



表示文字の垂直スタート位置は、垂直スタート位置レジスタで決定されます。このときの垂直スタート位置（水平走査線の数）はμPD17062に入力される $\overline{V_{SYNC}}$ 、 $\overline{H_{SYNC}}$ 信号の状態により図20-15のように決定されます。つまり $\overline{V_{SYNC}}$ 信号の立ち上がり後初めての $\overline{H_{SYNC}}$ 信号を1H目と数えます。

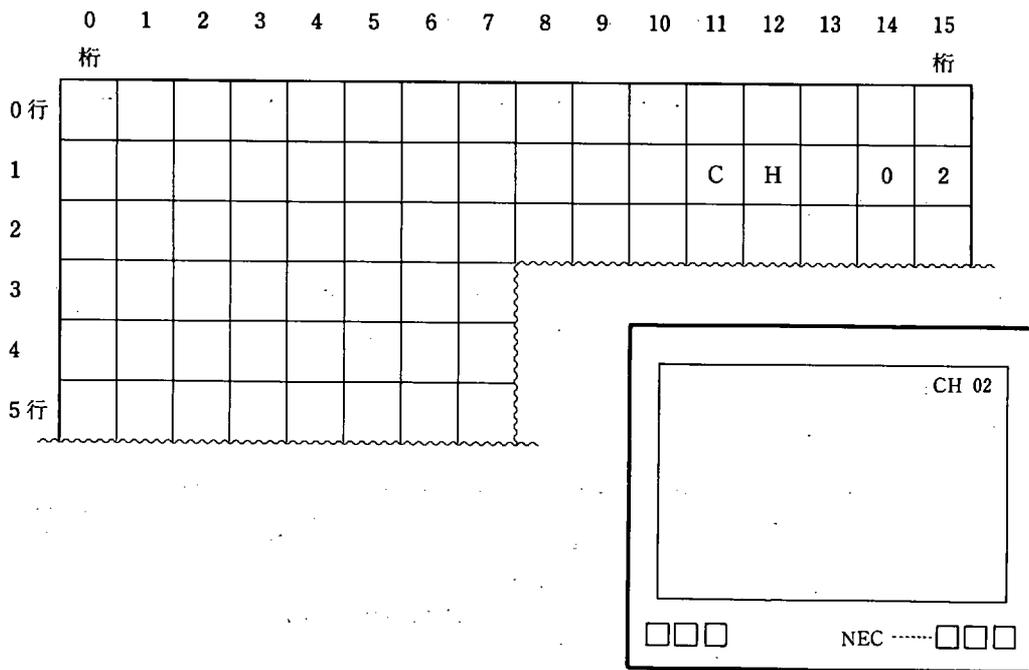
図20-15 垂直スタート位置の数え方



20.8 プログラム例

下図に示すような画面表示を行う場合のプログラム例を以下に示します。

表示内容



TV 画面上の表示

VRAM の RAM 名を以下のように定義します (仮)。

; **RAM SET**

VRAM0 MEM 2.00H VRAM マップ (BANK2)

VRAM1 MEM 2.01H

VRAM2 MEM 2.02H

VRAM3 MEM 2.03H

VRAM4 MEM 2.04H

VRAM5 MEM 2.05H

VRAM6 MEM 2.06H

VRAM7 MEM 2.07H

VRAM8 MEM 2.08H

VRAM9 MEM 2.09H

VRAMA MEM 2.0AH

VRAMB MEM 2.0BH

VRAMC MEM 2.0CH

VRAMD MEM 2.0DH

VRAME MEM 2.0EH

VRAMF MEM 2.0FH

⋮

	0	1	2	3	4	5	6	7
0	VRAM0	VRAM1	VRAM2	VRAM3	VRAM4	VRAM5	VRAM6	VRAM7
1
2
	8	9	A	B	C	D	E	F
0	VRAM8	VRAM9	VRAMA	VRAMB	VRAMC	VRAMD	VRAME	VRAMF
1
2

①の時点での VRAM (BANK2) の内容は、以下のようになります。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	0	0	C	0	D	8	1	0	0	0	2	4	0	4	0
1																

この例における CROM の内容を以下に示します。

```

ROM ADDRESS          CROM DATA
0800                ; *****
                   ; *** Image Display Controller data set ***
                   ; *****
0800                ORG 0800H
                   ; *****
                   ; *** 0 ***
                   ; *****
0800 0000          DCP 0, ' ' ; "0"
0801 007C          DCP 0, ' 00000 '
0802 00FE          DCP 0, ' 0000000 '
0803 01C7          DCP 0, ' 000 000 '
0804 0183          DCP 0, ' 00 00 '
0805 0183          DCP 0, ' 00 00 '
0806 0183          DCP 0, ' 00 00 '
0807 0183          DCP 0, ' 00 00 '
0808 0183          DCP 0, ' 00 00 '
0809 0183          DCP 0, ' 00 00 '
080A 0183          DCP 0, ' 00 00 '
080B 0183          DCP 0, ' 00 00 '
080C 01C7          DCP 0, ' 000 000 '
080D 00FE          DCP 0, ' 0000000 '
080E 007C          DCP 0, ' 00000 '
                   ; ** CD1 ** ; **コントロール・データ1**
080F 058A          DW 0000010110001010B ; 水平サイズ: 1倍, 垂直サイズ: 1倍
                                           ; 水平位置 : 11桁, 垂直位置 : 1行
                                           ; 色 : 緑 (G), ふちどり: なし
    
```

;*****
 ;*** 1 ***
 ;*****

0810	0000	DCP 0,	'	'	; "1"
0811	0006	DCP 0,	'	00	'
0812	000E	DCP 0,	'	00	'
0813	001E	DCP 0,	'	0000	'
0814	0076	DCP 0,	'	0000	'
0815	00C6	DCP 0,	'	00	'
0816	0186	DCP 0,	'	00	'
0817	0006	DCP 0,	'	00	'
0818	0006	DCP 0,	'	00	'
0819	0006	DCP 0,	'	00	'
081A	0006	DCP 0,	'	00	'
081B	0006	DCP 0,	'	00	'
081C	0006	DCP 0,	'	00	'
081D	0006	DCP 0,	'	000000	'
081E	0006	DCP 0,	'	000000	'

; ** CD2 **

081F 0082 DW 0000000010000010B

; **コントロール・データ2**
 ; 水平サイズ: 1倍, 垂直サイズ: 1倍
 ; 水平位置 : 1桁, 垂直位置 : 0行
 ; 色 : 緑 (G), ふちどり: なし

;*****
 ;*** 2 ***
 ;*****

ROM ADDRESS

0820	0000	DCP 0,	'	'	; "2"
0821	007C	DCP 0,	'	00000	'
0822	00FE	DCP 0,	'	0000000	'
0823	01C7	DCP 0,	'	000 000	'
0824	0183	DCP 0,	'	00 00	'
0825	0003	DCP 0,	'	00	'
0826	0007	DCP 0,	'	000	'
0827	000E	DCP 0,	'	000	'
0828	0038	DCP 0,	'	000	'
0829	00E0	DCP 0,	'	000	'
082A	01C0	DCP 0,	'	000	'
082B	0180	DCP 0,	'	000	'
082C	0180	DCP 0,	'	000	'
082D	01FF	DCP 0,	'	000000000	'
082E	01FF	DCP 0,	'	000000000	'

; ** CD2 **

082F 0000 DW 0000000000000000B ; NO USE

; *****
; *** 3 ***
; *****

0830 0000 DCP 0, ' ; "3"
0831 007C DCP 0, ' 00000 '
0832 00FE DCP 0, ' 0000000 '
0833 01C7 DCP 0, ' 000 000 '
0834 0183 DCP 0, ' 00 00 '
0835 0003 DCP 0, ' 00 '

⋮

; *****
; *** C ***
; *****

08C0 0000 DCP 0, ' ; "C"
08C1 007F DCP 0, ' 00000 '
08C2 00FF DCP 0, ' 0000000 '
08C3 01C0 DCP 0, ' 000 000 '
08C4 0180 DCP 0, ' 00 00 '
08C5 0180 DCP 0, ' 00 '
08C6 0180 DCP 0, ' 00 '
08C7 0180 DCP 0, ' 00 '
08C8 0180 DCP 0, ' 00 '
08C9 0180 DCP 0, ' 00 '
08CA 0180 DCP 0, ' 00 '
08CB 0180 DCP 0, ' 00 00 '
08CC 01C0 DCP 0, ' 000 000 '
08CD 00FF DCP 0, ' 0000000 '
08CE 007F DCP 0, ' 00000 '

08CF 0000 DW 0000000000000000B ; NO USE

; *****
 ; *** H ***
 ; *****

ROM ADDRESS

08D0	0000	DCP 0,			; "H"
08D1	0183	DCP 0,	00	00	
08D2	0183	DCP 0,	00	00	
08D3	0183	DCP 0,	00	00	
08D4	0183	DCP 0,	00	00	
08D5	0183	DCP 0,	00	00	
08D6	0183	DCP 0,	00	00	
08D7	01FF	DCP 0,	00000000		
08D8	01FF	DCP 0,	00000000		
08D9	0183	DCP 0,	00	00	
08DA	0183	DCP 0,	00	00	
08DB	0183	DCP 0,	00	00	
08DC	0183	DCP 0,	00	00	
08DD	0183	DCP 0,	00	00	
08DE	0183	DCP 0,	00	00	
;					
08DF	0000	DW	0000000000000000B		; NO USE

21. 水平同期信号カウンタ

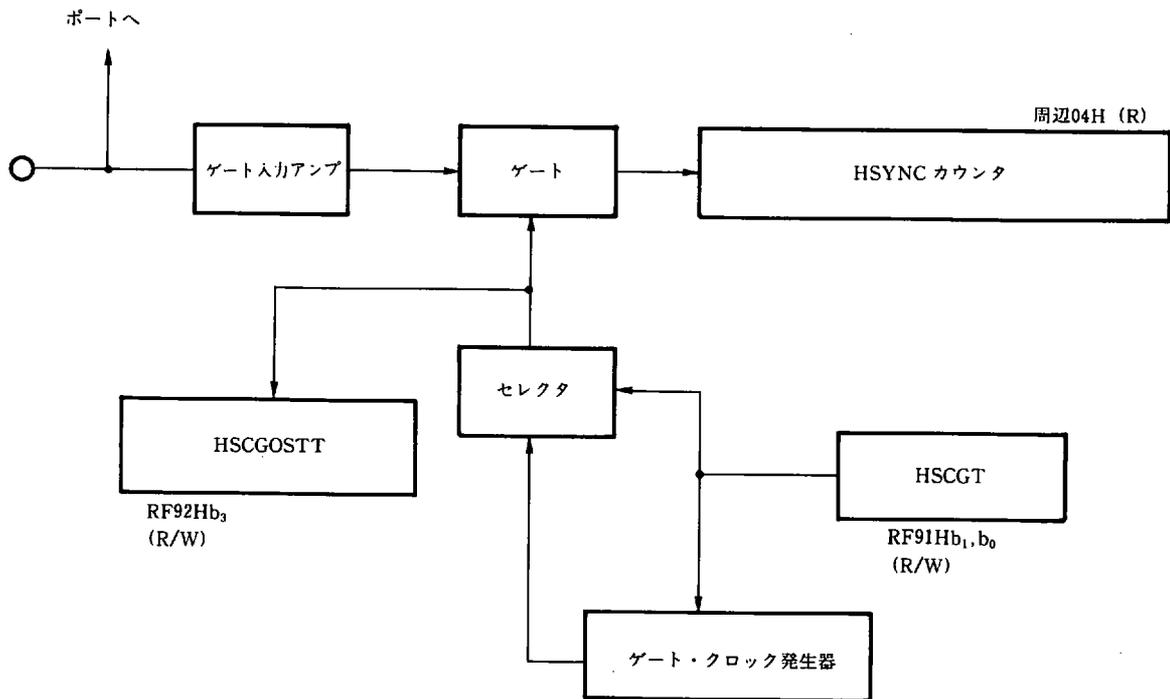
21.1 水平同期信号カウンタの構成

水平同期信号カウンタは、テレビなどの水平同期信号の周波数を計測するためのカウンタです。テレビ放送を受信している場合には規定の水平同期信号が出力されることを利用して、受信している周波数に放送局があるかないかを検出するときなどに利用することができます。

水平同期信号カウンタは、6ビットのHSYNCカウンタ(HSC)とゲート・クロック発生器、ゲート・コントロール・レジスタ(HSCGT)、ゲート入力アンプ、テスト・ゲート・オープン・レジスタ(HSCGOSTT)より構成されています。

P0B₃/HSCNT端子より入力された信号は、自己バイアスされた入力アンプで増幅され、ゲート・コントロール・レジスタにより、一定時間開くように制御されたゲートを通り、6ビットのHSYNCカウンタでカウントされます。ゲートが閉じるとHSYNCカウンタは、カウントをやめ、テスト・ゲート・オープン・レジスタに“1”をセットします。HSYNCカウンタは、読み出し専用のレジスタとなっているので、HSYNCカウンタを読み出せば、ゲートが開いている間にカウントされたパルス数が分かります。したがってゲートの開いている時間(1.69ms)でHSYNCカウンタから読み出した値を割ることでその周波数を求めることができます。またP0B₃/HSCNT端子はI/Oポートと兼用しており、ポートのP0B₃に割り当てられています。水平同期信号カウンタとして使用するとき、P0B₃を入力ポートに設定して使用してください。ポートとして使用するときHSCGTを0000Bにしてください。水平同期信号カウンタ入力として使用している場合にP0B₃を読むと、必ず“0”になります。

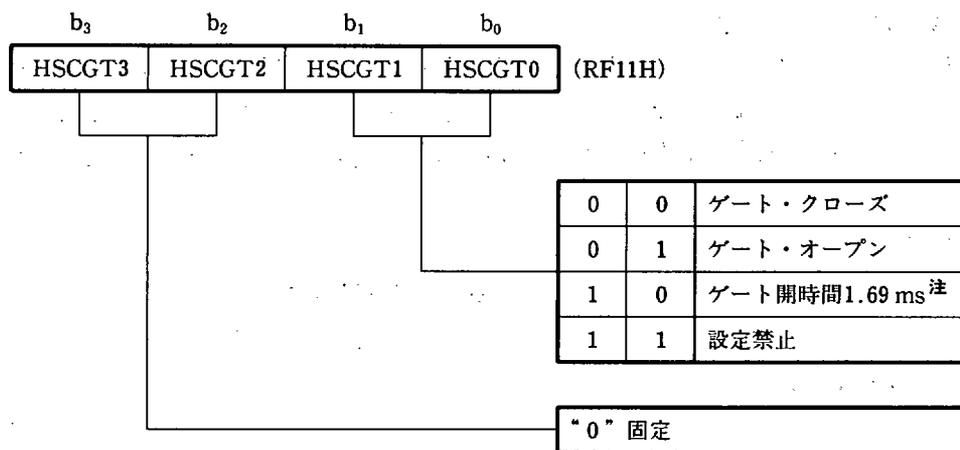
図 21-1 水平同期信号カウンタのブロック図



21.2 ゲート・コントロール・レジスタ (HSCGT)

ゲート・コントロール・レジスタは、ゲートを制御するための2ビットのレジスタで HSCGT1 フラグと HSCGT0 フラグで構成されており、レジスタ・ファイルの 11H 番地に配置されています。“PEEK”, “POKE” 命令により読み書き可能で、システム・レジスタのウインドウ・レジスタを介して行います。

ゲート・コントロール・レジスタを設定することにより次のようなモードになります。



注 このモードを選択したときのみゲート・クロック発生器が動作します。

21.2.1 ゲート・クローズ・モード

このモードでは、ゲートは閉じられたままで、HSC、ゲート・クロック発生器は動作しません (HSYNC カウンタの値は変化しません)。また水平同期信号カウンタ入力のバイアスもオフされるため、ポートとして使用するときには必ずこのモードにしてください。

パワーオン・リセット時およびクロック・ストップ時はこのモードになります。

21.2.2 ゲート・オープン・モード

このモードを設定すると、ゲートが開き、HSYNC カウンタをリセットした後、入力信号をカウントし始めます。HSYNC カウンタの値がオーバーフローすると“0”になります。

このモードのとき、入力端子はバイアスされます。

21.2.3 ゲート 1.69 ms モード

このモードに設定されると、HSYNC カウンタはリセットされ、 $\frac{3.375}{2}$ ms 遅れでカウントを開始します ($\frac{3.375}{2}$ ms の誤差は 0 - 62.5 μs)。ゲート時間は1.69 ms です。このモードのとき、入力端子はバイアスされます。

また、ゲートが開いたときや閉じるときに、ハイ・レベルの場合は、それを 1 発としてカウントします。

21.3 HSYNC カウンタ (HSC)

周辺アドレス 04H に割り当てられており、“GET” 命令でデータ・バッファを介してデータを読み出す読み出し専用の 6 ビット・バイナリ・カウンタです。

6 ビットなので、オーバフローしたときは、また 00H からカウントを始めます。

パワーオン・リセット時およびクロック・ストップ時、HSYNC カウンタの値は 00H にリセットされます。

(1) ゲート・オープン・ビット (HSCGOSTT)

HSCGOSTT は、レジスタ・ファイル 12H の MSB (b₃) に、割り当てられており、 $\overline{\text{Hsync}}$ 入力のゲートが開いているときは常にハイ・レベルになります。ただし、ゲート 1.69 ms モードを選択したときは、ゲート・クロックが来なくてもデータを設定した時点からハイ・レベルが出力されますので注意してください。

21.4 水平同期信号カウンタの使用例

水平同期信号カウンタを使用する場合の、プログラム例を以下に示します。

1.69 ms ゲート・オープンの場合

```

    CLR1      P0BBIO3      ; P0B3 を入力モードに設定します。
  
```

```

    PEEK  WR, 0B6H
    AND   WR, #0111B
    POKE  0B6H, WR
  
```

LOOP :

```

    PEEK  WR, #92H      ; ゲートが一度閉じるのを確認します。
    SKF   WR, #1000B
    BR    LOOP
  
```

```

    MOV   WR, #0010B   ; 1.69 ms 間ゲート・オープンモードに設定します。
    POKE  91H, WR
  
```

LOOP2 :

```

    PEEK  WR, #92H      ; ゲートが閉じるのを確認します。
    SKF   WR, #1000B
    BR    LOOP2
  
```

```

    GET   DBF, HSC      ; HSYNC カウンタの内容を取り込みます。
  
```

22. μPD17062 命令

22.1 命令セット概要

b ₁₄ -b ₁₁ \ b ₁₅		0		1	
BIN.	HEX.				
0 0 0 0	0	ADD	r, m	ADD	m, #n4
0 0 0 1	1	SUB	r, m	SUB	m, #n4
0 0 1 0	2	ADDC	r, m	ADDC	m, #n4
0 0 1 1	3	SUBC	r, m	SUBC	m, #n4
0 1 0 0	4	AND	r, m	AND	m, #n4
0 1 0 1	5	XOR	r, m	XOR	m, #n4
0 1 1 0	6	OR	r, m	OR	m, #n4
0 1 1 1	7	INC	AR		
		INC	IX		
		MOV	DBF, @AR		
		BR	@AR		
		CALL	@AR		
		RET			
		RETSK			
		EI			
		DI			
		RETI			
		PUSH	AR		
		POP	AR		
		GET	DBF, p		
		PUT	p, DBF		
		PEEK	WR, rf		
POKE	rf, WR				
RORC	r				
STOP	s				
HALT	h				
NOP					
1 0 0 0	8	LD	r, m	ST	m, r
1 0 0 1	9	SKE	m, #n4	SKGE	m, #n4
1 0 1 0	A	MOV	@r, m	MOV	m, @r
1 0 1 1	B	SKNE	m, #n4	SKLT	m, #n4
1 1 0 0	C	BR	addr (ページ 0)	CALL	addr (ページ 0)
1 1 0 1	D	BR	addr (ページ 1)	MOV	m, #n4
1 1 1 0	E			SKT	m, #n
1 1 1 1	F			SKF	m, #n

22.2 凡 例

AR	: アドレス・レジスタ
ASR	: スタック・ポインタで示されるアドレス・スタック・レジスタ
addr	: プログラム・メモリ・アドレス (11ビット)
BANK	: バンク・レジスタ
CMP	: コンペア・フラグ
CY	: キャリー・フラグ
DBF	: データ・バッファ
h	: ホールト解除条件
INTEF	: インタラプト・イネーブル・フラグ
INTR	: 割り込み時スタックに自動退避されるレジスタ
INTSK	: 割り込みスタック・レジスタ
IX	: インデクス・レジスタ
MP	: データ・メモリ・ロウ・アドレス・ポインタ
MPE	: メモリ・ポインタ・イネーブル・フラグ
m	: m_R , m_C で示されるデータ・メモリ・アドレス
m_R	: データ・メモリ・ロウ・アドレス (上位)
m_C	: データ・メモリ・カラム・アドレス (下位)
n	: ビット・ポジション (4ビット)
n_4	: イミーディエト・データ (4ビット)
PAGE	: ページ (プログラム・カウンタのビット11)
PC	: プログラム・カウンタ
p	: 周辺アドレス
p_H	: 周辺アドレス (上位3ビット)
p_L	: 周辺アドレス (下位4ビット)
r	: ジェネラル・レジスタ・カラム・アドレス
rf	: レジスタ・ファイル・アドレス
rf_R	: レジスタ・ファイル・アドレス (上位3ビット)
rf_C	: レジスタ・ファイル・アドレス (下位4ビット)
SP	: スタック・ポインタ
s	: ストップ解除条件
WR	: ウィンドウ・レジスタ
(×)	: ×でアドレスされる内容

22.3 命令セット一覧

命令群	ニモニック	オペランド	オペレーション	命令コード			
				オペ・コード	オペランド		
加算	ADD	r, m	$(r) \leftarrow (r) + (m)$	00000	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) + n4$	10000	m_R	m_C	n4
	ADDC	r, m	$(r) \leftarrow (r) + (m) + CY$	00010	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) + n4 + CY$	10010	m_R	m_C	n4
	INC	AR	$AR \leftarrow AR + 1$	00111	000	1001	0000
IX		$IX \leftarrow IX + 1$	00111	000	1000	0000	
減算	SUB	r, m	$(r) \leftarrow (r) - (m)$	00001	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) - n4$	10001	m_R	m_C	n4
	SUBC	r, m	$(r) \leftarrow (r) - (m) - CY$	00011	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) - n4 - CY$	10011	m_R	m_C	n4
論理演算	OR	r, m	$(r) \leftarrow (r) \vee (m)$	00110	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) \vee n4$	10110	m_R	m_C	n4
	AND	r, m	$(r) \leftarrow (r) \wedge (m)$	00100	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) \wedge n4$	10100	m_R	m_C	n4
	XOR	r, m	$(r) \leftarrow (r) \oplus (m)$	00101	m_R	m_C	r
		m, #n4	$(m) \leftarrow (m) \oplus n4$	10101	m_R	m_C	n4
判断	SKT	m, #n	$CMP \leftarrow 0$, if $(m) \wedge n = n$, then skip	11110	m_R	m_C	n
	SKF	m, #n	$CMP \leftarrow 0$, if $(m) \wedge n = 0$, then skip	11111	m_R	m_C	n
比較	SKE	m, #n4	$(m) - n4$, skip if zero	01001	m_R	m_C	n4
	SKNE	m, #n4	$(m) - n4$, skip if not zero	01011	m_R	m_C	n4
	SKGE	m, #n4	$(m) - n4$, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m, #n4	$(m) - n4$, skip if borrow	11011	m_R	m_C	n4
回転	RORC	r	$\rightarrow CY \rightarrow (r)_{b3} \rightarrow (r)_{b2} \rightarrow (r)_{b1} \rightarrow (r)_{b0} \rightarrow$	00111	000	0111	r
転送	LD	r, m	$(r) \leftarrow (m)$	01000	m_R	m_C	r
	ST	m, r	$(m) \leftarrow (r)$	11000	m_R	m_C	r
	MOV	@r, m	if MPE=1: $(MP, (r)) \leftarrow (m)$ if MPE=0: $(BANK, m_R, (r)) \leftarrow (m)$	01010	m_R	m_C	r
		m, @r	if MPE=1: $(m) \leftarrow (MP, (r))$ if MPE=0: $(m) \leftarrow (BANK, m_R, (r))$	11010	m_R	m_C	r
		m, #n4	$(m) \leftarrow n4$	11101	m_R	m_C	n4
MOV _T	DBF, @AR	$SP \leftarrow SP - 1$, $ASR \leftarrow PC$, $PC \leftarrow AR$, $DBF \leftarrow (PC)$, $PC \leftarrow ASR$, $SP \leftarrow SP + 1$	00111	000	0001	0000	

命令群	ニモニク	オペランド	オペレーション	命令コード			
				オペコード	オペランド		
転送	PUSH	AR	SP ← SP-1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP+1	00111	000	1100	0000
	PEEK	WR, rf	WR ← (rf)	00111	rf _R	0011	rf _C
	POKE	rf, WR	(rf) ← WR	00111	rf _R	0010	rf _C
	GET	DBF, p	DBF ← (p)	00111	p _H	1011	p _L
	PUT	p, DBF	(p) ← DBF	00111	p _H	1010	p _L
分岐	BR	addr	PC ₁₀₋₀ ← addr, PAGE ← 0	01100	addr		
			PC ₁₀₋₀ ← addr, PAGE ← 1	01101			
	@AR	PC ← AR	00111	000	0100	0000	
サブルーチン	CALL	addr	SP ← SP-1, ASR ← PC, PC ₁₁ ← 0, PC ₁₀₋₀ ← addr	11100	addr		
			@AR	SP ← SP-1, ASR ← PC, PC ← AR	00111	000	0101
	RET		PC ← ASR, SP ← SP+1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP+1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP+1	00111	100	1110	0000
割り込み	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
その他	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

22.4 組み込みマクロ命令

17Kシリーズのアセンブラ (AS17K) には、下記のマクロ命令があらかじめ、組み込みマクロ命令として用意されています。詳しくはアセンブラのユーザーズ・マニュアルをご覧ください。

凡 例

flag n : FLG 型シンボル
 < > : < >内は省略可能

	ニモニック	オペランド	オペレーション	n
組 み 込 み マ ク ロ	SKTn	flag 1, ...flag n	if (flag 1) ~ (flag n) = all "1", then skip	$1 \leq n \leq 4$
	SKFn	flag 1, ...flag n	if (flag 1) ~ (flag n) = all "0", then skip	$1 \leq n \leq 4$
	SETn	flag 1, ...flag n	(flag 1) ~ (flag n) ← 1	$1 \leq n \leq 4$
	CLRn	flag 1, ...flag n	(flag 1) ~ (flag n) ← 0	$1 \leq n \leq 4$
	NOTn	flag 1, ...flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	$1 \leq n \leq 4$
	INITFLG	<NOT>flag 1, ...<<NOT>flag n>	if description=NOT flag n, then (flag n) ← 0 if description=flag n, then (flag n) ← 1	$1 \leq n \leq 4$
	BANKn		(BANK) ← n	$0 \leq n \leq 2$

23. アセンブラの予約シンボル

アセンブラを使用したときの μPD17062 の予約シンボルを以下に示します。

23.1 システム・レジスタ

シンボル名	属性	値	R/W	説明
AR3	MEM	0.74H	R	アドレス・レジスタのビット15-12
AR2	MEM	0.75H	R	アドレス・レジスタのビット11-8
AR1	MEM	0.76H	R/W	アドレス・レジスタのビット7-4
AR0	MEM	0.77H	R/W	アドレス・レジスタのビット3-0
WR	MEM	0.78H	R/W	ウインドウ・レジスタ
BANK	MEM	0.79H	R/W	バンク・レジスタ
IXH	MEM	0.7AH	R	インデクス・レジスタのビット10-8
MPH	MEM	0.7AH	R	メモリ・ポインタのビット6-4
MPE	FLG	0.7AH.3	R/W	メモリ・ポインタ・イネーブル・フラグ
IXM	MEM	0.7BH	R/W	インデクス・レジスタのビット7-4
MPL	MEM	0.7BH	R/W	メモリ・ポインタのビット3-0
IXL	MEM	0.7CH	R/W	インデクス・レジスタのビット3-0
RPH	MEM	0.7DH	R	レジスタ・ポインタのビット6-3
RPL	MEM	0.7EH	R/W	レジスタ・ポインタのビット2-0
PSW	MEM	0.7FH	R/W	プログラム・ステータス・ワード
BCD	FLG	0.7EH.0	R/W	BCD演算フラグ
CMP	FLG	0.7FH.3	R/W	コンペア・フラグ
CY	FLG	0.7FH.2	R/W	キャリー・フラグ
Z	FLG	0.7FH.1	R/W	ゼロ・フラグ
IXE	FLG	0.7FH.0	R/W	インデクス・イネーブル・フラグ

23.2 データ・バッファ

シンボル名	属性	値	R/W	説明
DBF3	MEM	0.0CH	R/W	データ・バッファのビット15-12
DBF2	MEM	0.0DH	R/W	データ・バッファのビット11-8
DBF1	MEM	0.0EH	R/W	データ・バッファのビット7-4
DBF0	MEM	0.0FH	R/W	データ・バッファのビット3-0

23.3 ポート・レジスタ

シンボル名	属性	値	R/W	説明
P0A3	FLG	0.70H.3	R/W	ポート0Aのビット3
P0A2	FLG	0.70H.2	R/W	ポート0Aのビット2
P0A1	FLG	0.70H.1	R/W	ポート0Aのビット1
P0A0	FLG	0.70H.0	R/W	ポート0Aのビット0
P0B3	FLG	0.71H.3	R/W	ポート0Bのビット3
P0B2	FLG	0.71H.2	R/W	ポート0Bのビット2
P0B1	FLG	0.71H.1	R/W	ポート0Bのビット1
P0B0	FLG	0.71H.0	R/W	ポート0Bのビット0
P0C3	FLG	0.72H.3	R/W	ポート0Cのビット3
P0C2	FLG	0.72H.2	R/W	ポート0Cのビット2
P0C1	FLG	0.72H.1	R/W	ポート0Cのビット1
P0C0	FLG	0.72H.0	R/W	ポート0Cのビット0
P0D3	FLG	0.73H.3	R ^注	ポート0Dのビット3
P0D2	FLG	0.73H.2	R ^注	ポート0Dのビット2
P0D1	FLG	0.73H.1	R ^注	ポート0Dのビット1
P0D0	FLG	0.73H.0	R ^注	ポート0Dのビット0
P1A3	FLG	1.70H.3	R/W	ポート1Aのビット3
P1A2	FLG	1.70H.2	R/W	ポート1Aのビット2
P1A1	FLG	1.70H.1	R/W	ポート1Aのビット1
P1A0	FLG	1.70H.0	R/W	ポート1Aのビット0
P1B3	FLG	1.71H.3	R/W	ポート1Bのビット3
P1B2	FLG	1.71H.2	R/W	ポート1Bのビット2
P1B1	FLG	1.71H.1	R/W	ポート1Bのビット1
P1B0	FLG	1.71H.0	R/W	ポート1Bのビット0
P1C3	FLG	1.72H.3	R/W	ポート1Cのビット3
P1C2	FLG	1.72H.2	R/W	ポート1Cのビット2
P1C1	FLG	1.72H.1	R/W	ポート1Cのビット1

注 入力専用ポートですが、ポートに出力する命令を記述してもアセンブラ、IE-17Kはエラー・メッセージを出力しません。また、実際にデバイスで実行させても動作上は何も変化は起こりません。

23.4 レジスタ・ファイル

シンボル名	属性	値	R/W	説明
IDCDMAEN	FLG	0.80H.1	R/W	DMA イネーブル・フラグ
SP	MEM	0.81H	R/W	スタック・ポインタ
CE	FLG	0.87H.0	R	CE ピン・ステータス・フラグ
SIO0CH	FLG	0.88H.3	R/W	SIO0 チャンネル・セレクト・フラグ
SB	FLG	0.88H.2	R/W	SIO0 モード・セレクト・フラグ
SIO0MS	FLG	0.88H.1	R/W	SIO0 クロック・モード・セレクト・フラグ
SIO0TX	FLG	0.88H.0	R/W	SIO0 TX/RX セレクト・フラグ
BTM0ZX	FLG	0.89H.3	R/W	タイマ0 インタラプト・モード・セレクト・フラグ
BTM0CK2	FLG	0.89H.2	R/W	タイマ0 キャリー FF モード・セレクト・フラグ
BTM0CK1	FLG	0.89H.1	R/W	タイマ0 キャリー FF モード・セレクト・フラグ
BTM0CK0	FLG	0.89H.0	R/W	タイマ0 キャリー FF モード・セレクト・フラグ
INTVSYN	FLG	0.8FH.2	R	Vsync ピン・ステータス・フラグ
INTNC	FLG	0.8FH.0	R	INT _{NC} 端子の状態
HSCGT3	FLG	0.91H.3	R/W	Hsync カウンタ・モード・セレクト・フラグ (ダミー: 0)
HSCGT2	FLG	0.91H.2	R/W	Hsync カウンタ・モード・セレクト・フラグ (ダミー: 0)
HSCGT1	FLG	0.91H.1	R/W	Hsync カウンタ・モード・セレクト・フラグ
HSCGT0	FLG	0.91H.0	R/W	Hsync カウンタ・モード・セレクト・フラグ
HSCGOSTT	FLG	0.92H.3	R	Hsync カウンタ・ゲート・オープン・フラグ
PLLRFCK3	FLG	0.93H.3	R/W	PLL レファレンス・クロック・セレクト・フラグ
PLLRFCK2	FLG	0.93H.2	R/W	PLL レファレンス・クロック・セレクト・フラグ
PLLRFCK1	FLG	0.93H.1	R/W	PLL レファレンス・クロック・セレクト・フラグ
PLLRFCK0	FLG	0.93H.0	R/W	PLL レファレンス・クロック・セレクト・フラグ
INTNCMD3	FLG	0.95H.3	R/W	INT _{NC} 端子ステータス・フラグ (ダミー)
INTNCMD2	FLG	0.95H.2	R/W	INT _{NC} 端子ステータス・フラグ
INTNCMD1	FLG	0.95H.1	R/W	INT _{NC} 端子ステータス・フラグ
INTNCMD0	FLG	0.95H.0	R/W	INT _{NC} 端子ステータス・フラグ
BTM0CY	FLG	0.97H.0	R	タイマ0 キャリー FF ステータス・フラグ
SBACK	FLG	0.98H.3	R/W	シリアル・バス・アクノリッジ・フラグ
SIO0NWT	FLG	0.98H.2	R/W	SIO0 ノー・ウェイト・フラグ
SIO0WRQ1	FLG	0.98H.1	R/W	SIO0 ウェイト・リクエスト・フラグ
SIO0WRQ0	FLG	0.98H.0	R/W	SIO0 ウェイト・リクエスト・フラグ
IEGVSYN	FLG	0.9FH.2	R/W	Vsync インタラプト・エッジ・セレクト・フラグ
IEGNC	FLG	0.9FH.0	R/W	INTNC インタラプト・エッジ・セレクト・フラグ
ADCCH2	FLG	0.0A1H.3	R/W	A/D コンバータ・チャンネル・セレクト・フラグ
ADCCH1	FLG	0.0A1H.2	R/W	A/D コンバータ・チャンネル・セレクト・フラグ
ADCCH0	FLG	0.0A1H.1	R/W	A/D コンバータ・チャンネル・セレクト・フラグ
ADCCMP	FLG	0.0A1H.0	R/W	A/D コンバータ・ジャッジ・フラグ
PLLUL	FLG	0.0A2H.0	R	PLL アンロック FF フラグ
PICGIO	FLG	0.0A7H.0	R/W	Port 1C I/O セレクト・フラグ

シンボル名	属性	値	R/W	説明
SIO0SF8	FLG	0.0A8H.3	R	SIO0 シフト8クロック・フラグ
SIO0SF9	FLG	0.0A8H.2	R	SIO0 シフト9クロック・フラグ
SBSTT	FLG	0.0A8H.1	R	シリアル・バス・スタート・テスト・フラグ
SBBSY	FLG	0.0A8H.0	R	シリアル・バス・ビジー・フラグ
IPSIO0	FLG	0.0AFH.3	R/W	SIO0 インタラプト・パーミッション・フラグ
IPVSYN	FLG	0.0AFH.2	R/W	Vsync インタラプト・パーミッション・フラグ
IPBTM0	FLG	0.0AFH.1	R/W	タイマ0 インタラプト・パーミッション・フラグ
IPNC	FLG	0.0AFH.0	R/W	INTNC インタラプト・パーミッション・フラグ
CROMBNK	FLG	0.0B0H.0	R/W	CROM バンク・セレクト・フラグ
IDCEN	FLG	0.0B1H.0	R/W	IDC イネーブル・フラグ
PLULSEN3	FLG	0.0B2H.3	R/W	PLL アンロック・タイム・セレクト・フラグ (ダミー:0)
PLULSEN2	FLG	0.0B2H.2	R/W	PLL アンロック・タイム・セレクト・フラグ (ダミー:0)
PLULSEN1	FLG	0.0B2H.1	R/W	PLL アンロック・タイム・セレクト・フラグ
PLULSEN0	FLG	0.0B2H.0	R/W	PLL アンロック・タイム・セレクト・フラグ
P1BBIO3	FLG	0.0B5H.3	R/W	P1B3 I/O セレクト・フラグ
P1BBIO2	FLG	0.0B5H.2	R/W	P1B2 I/O セレクト・フラグ
P1BBIO1	FLG	0.0B5H.1	R/W	P1B1 I/O セレクト・フラグ
P1BBIO0	FLG	0.0B5H.0	R/W	P1B0 I/O セレクト・フラグ
P0BBIO3	FLG	0.0B6H.3	R/W	P0B3 I/O セレクト・フラグ
P0BBIO2	FLG	0.0B6H.2	R/W	P0B2 I/O セレクト・フラグ
P0BBIO1	FLG	0.0B6H.1	R/W	P0B1 I/O セレクト・フラグ
P0BBIO0	FLG	0.0B6H.0	R/W	P0B0 I/O セレクト・フラグ
P0ABIO3	FLG	0.0B7H.3	R/W	P0A3 I/O セレクト・フラグ
P0ABIO2	FLG	0.0B7H.2	R/W	P0A2 I/O セレクト・フラグ
P0ABIO1	FLG	0.0B7H.1	R/W	P0A1 I/O セレクト・フラグ
P0ABIO0	FLG	0.0B7H.0	R/W	P0A0 I/O セレクト・フラグ
SIO0IMD3	FLG	0.0B8H.3	R/W	SIO0 インタラプト・モード・セレクト・フラグ (ダミー:0)
SIO0IMD2	FLG	0.0B8H.2	R/W	SIO0 インタラプト・モード・セレクト・フラグ (ダミー:0)
SIO0IMD1	FLG	0.0B8H.1	R/W	SIO0 インタラプト・モード・セレクト・フラグ
SIO0IMD0	FLG	0.0B8H.0	R/W	SIO0 インタラプト・モード・セレクト・フラグ
SIO0CK3	FLG	0.0B9H.3	R/W	SIO0 シフト・クロック・セレクト・フラグ (ダミー:0)
SIO0CK2	FLG	0.0B9H.2	R/W	SIO0 シフト・クロック・セレクト・フラグ (ダミー:0)
SIO0CK1	FLG	0.0B9H.1	R/W	シリアル・クロックの選択
SIO0CK0	FLG	0.0B9H.0	R/W	シリアル・クロックの選択
IRQSIO0	FLG	0.0BFH.3	R	SIO0 割り込み要求フラグ
IRQVSYN	FLG	0.0BFH.2	R	Vsync 割り込み要求フラグ
IRQBTM0	FLG	0.0BFH.1	R	タイマ0 割り込み要求フラグ
IRQNC	FLG	0.0BFH.0	R	INT _{NC} 端子割り込み要求フラグ

23.5 周辺ハードウェア・レジスタ

シンボル名	属性	値	R/W	説明
IDCORG	DAT	01H	R/W	IDC スタート位置設定レジスタ
ADCR	DAT	02H	R/W	A/D コンバータ V _{REF} データ・レジスタ
SIO0SFR	DAT	03H	R/W	SIO0 レジスタ
HSC	DAT	04H	R	Hsync カウンタ・データ・レジスタ
PWMR0	DAT	05H	R/W	PWM データ・レジスタ 0
PWMR1	DAT	06H	R/W	PWM データ・レジスタ 1
PWMR2	DAT	07H	R/W	PWM データ・レジスタ 2
PWMR3	DAT	08H	R/W	PWM データ・レジスタ 3
AR	DAT	40H	R/W	アドレス・レジスタ
PLL	DAT	41H	R/W	PLL データ・レジスタ
AR_EPA1	DAT	8040H	-	CALL/BR/MOVT 命令のオペランド (EPA ビット・オン)
AR_EPA0	DAT	4040H	-	CALL/BR/MOVT 命令のオペランド (EPA ビット・オフ)

23.6 その他

シンボル名	属性	値	説明
DBF	DAT	0FH	PUT 命令, GET 命令, MOVT 命令の固定オペランド値
IX	DAT	01H	INC 命令の固定オペランド値

24. 電気的特性

絶対最大定格 (Ta = 25 ± 2 °C)

項目	略号	定 格	単 位
電 源 電 圧	V _{DD}	-0.3 ~ +6.0	V
入 力 電 圧	V _I	-0.3 ~ V _{DD} + 0.3	V
出 力 電 圧	V _O	-0.3 ~ V _{DD} + 0.3 (PIA ₃ -PIA ₀ , PWM ₃ -PWM ₀ を除く)	V
出 力 吸 収 電 流	I _O	10 (PIAを除く)	mA
出 力 耐 圧	V _{BDS}	13 (PIA, PWM)	V
動 作 温 度	T _{opt1}	-20 ~ +70	°C
	T _{opt2}	-40 ~ +85 (IDC停止時)	
保 存 温 度	V _{stg}	-55 ~ +125	°C

推奨動作範囲 (Ta = -40 ~ +85 °C)

項目	略号	条 件	MIN.	TYP.	MAX.	単 位
電 源 電 圧	V _{DD1}	CPU, PLLおよびIDC動作時 Ta = -20 ~ +70 °C	4.5	5.0	5.5	V
	V _{DD2}	CPUおよびPLL動作, IDC停止時	4.5	5.0	5.5	V
	V _{DD3}	CPUのみ動作, PLLおよびIDC停止時	4.0	5.0	5.5	V
デ ー タ 保 持 電 圧	V _{DDR}	水晶発振停止	3.0		5.5	V
出 力 耐 圧	V _{BDS}	PIB ₃ -PIB ₁	0.0		12.5	V
入 力 振 幅	V _{in1}	VCO	0.7		V _{DD}	V _{P-P}
電源電圧立ち上がり時間	t _{rise}	V _{DD} : 0 → 4.0 V			500	ms

AC 特性 ($T_a = -40 \sim +85^\circ\text{C}$, $V_{DD} = 5\text{V} \pm 10\%$, $RH \leq 70\%$)

項目	記号	条件	MIN.	TYP.	MAX.	単位
動作周波数	f_{in1}	VCO 正弦波入力 $V_{in} = 0.7 V_{P-P}$	0.7		20	MHz
	f_{in2}	TMIN	45		65	Hz
	f_{in3}	HSCNT	10		20	kHz
IDC ジッタ	IDC _G			4.0	8.0	ns
IDC 水平開始位置	IDC _{HP}	$\overline{H_{SYNC}}$ 後縁より		4.25		μs
IDC 水平開始位置	IDC _{VP}	$\overline{V_{SYNC}}$ 後縁より		17		H

A/D コンバータ特性 ($T_a = -40 \sim +85^\circ\text{C}$, $V_{DD} = 5\text{V} \pm 10\%$, $RH \leq 70\%$)

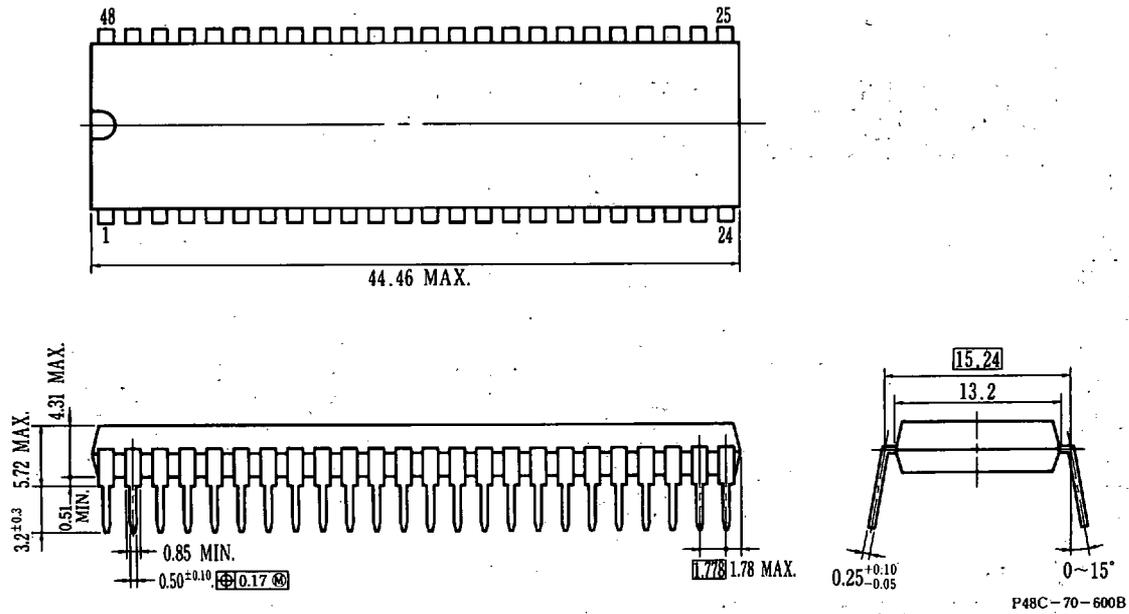
項目	記号	条件	MIN.	TYP.	MAX.	単位
A/D 変換分解能					4	bit
A/D 変換総合誤差		$T_a = -10 \sim +50^\circ\text{C}$	±0.5		±1.0	LSB
A/D 入力インピーダンス			1.0			MΩ

DC 特性 ($T_a = -40 \sim +85^\circ\text{C}$, $V_{DD} = 5\text{V} \pm 10\%$, $RH \leq 70\%$)

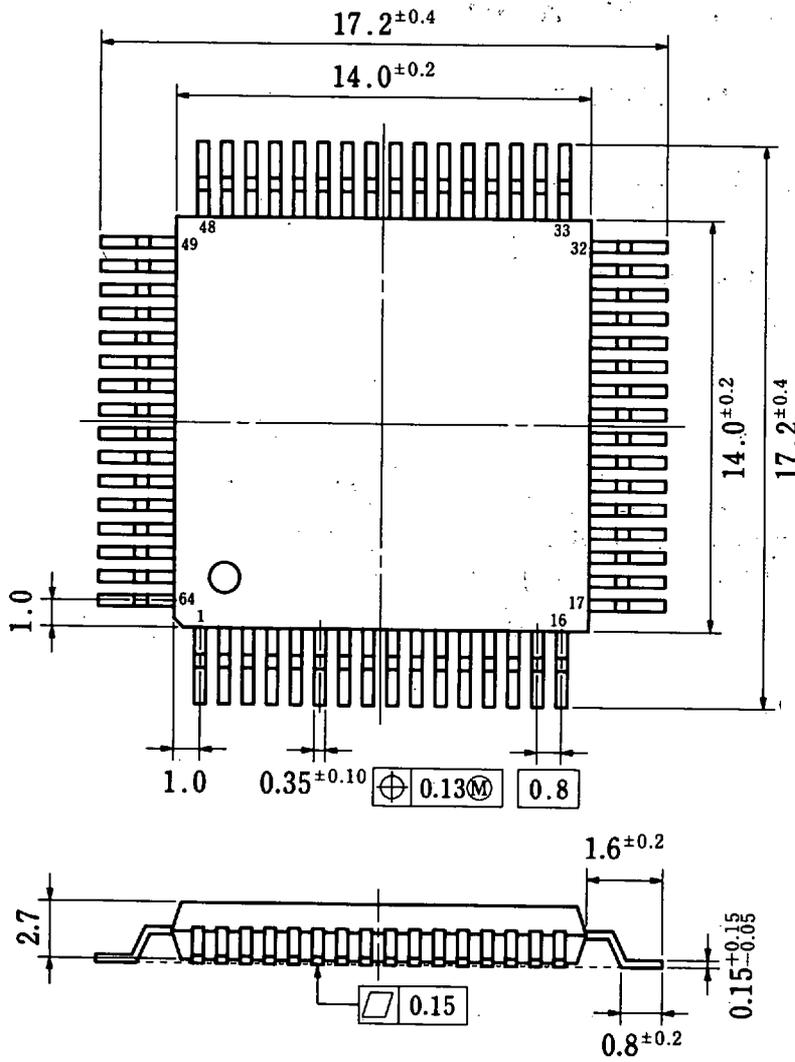
項目	記号	条件	MIN.	TYP.	MAX.	単位
電源電圧	V_{DD1}	CPU, PLLおよびIDC動作時 $T_a = -20 \sim +85^\circ\text{C}$	4.5	5.0	5.5	V
	V_{DD2}	CPUおよびPLL動作, IDC停止時	4.5	5.0	5.5	V
	V_{DD3}	CPUのみ動作, PLLおよびIDC停止時	4.0	5.0	5.5	V
電源電流	I_{DD4}	CPU動作, PLL, IDC停止, HALT命令使用時 (5msごとに20命令実行)		1.0	3.0	mA
データ保持電圧	V_{DDR}	タイマFFによる停電検出方法使用 水晶発振停止時	3.0		5.5	V
データ保持電流	I_{DDR}	水晶発振停止時 $T_a = 25^\circ\text{C}$		1.5	10	μA
高レベル入力電圧	V_{IH1}	P0A, P0B, P0D, P1B, P1C	$0.7 V_{DD}$			V
	V_{IH2}	CE, INT _{NC} , $\overline{V_{SYNC}}$, $\overline{H_{SYNC}}$	$0.8 V_{DD}$			mA
低レベル入力電圧	V_{IL1}	P0A, P0B, P0D, P1B, P1C			$0.3 V_{DD}$	V
	V_{IL2}	CE, INT _{NC} , $\overline{V_{SYNC}}$, $\overline{H_{SYNC}}$			$0.2 V_{DD}$	mA
高レベル出力電流	I_{OH1}	P0A, P0B, P0C, P1B, P1C, RED, GREEN, BLUE, BLANK $V_{OH} = V_{DD} - 1\text{V}$	-1.0	-2.0		mA
低レベル出力電流	I_{OL1}	P0A, P0B, P0C, P1B, P1C, RED, GREEN, BLUE, BLANK $V_{OH} = V_{DD} - 1\text{V}$	1.0	2.0		mA
	I_{OL4}	P1A $V_{OL} = 1\text{V}$	15	22		mA
高レベル入力電流	I_{IH1}	P0Dプルダウン時 $V_{IH} = V_{DD}$	20	70	150	μA
	I_{IH2}	VCO $V_{IH} = V_{DD}$	0.1	0.8	1.3	mA
出力オフ・リーク電流	I_{IL1}	P1A, PWM $V_{OH} = 12.5\text{V}$			0.5	μA
	I_{IL2}	EO $V_{OH} = V_{DD}$, $V_{OL} = 0\text{V}$			±1	μA
出力耐圧	V_{BDS}	P1A, PWM			12.5	V

25. 外形図

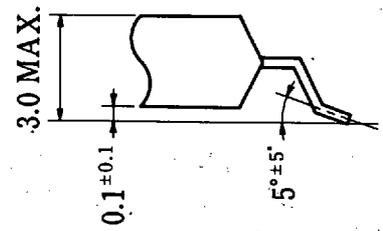
48ピン・プラスチック・シュリンク DIP (600 mil) 外形図(単位: mm)



64ピン・プラスチック QFP (□14) 外形図(単位: mm)



端子先端形状詳細図



S64GC-80-3BE

26. 半田付け推奨条件

μPD17062の半田付け実装は、次の推奨条件で実施してください。

半田付け推奨条件の詳細は、インフォメーション資料「半導体デバイス実装マニュアル」(IEI-616)を参照してください。

なお、推奨条件以外の半田付け方式および半田付け条件については、当社販売員にご相談ください。

表 26-1 表面実装タイプの半田付け条件

μPD17062GC-XXXX-3BE : 64ピン・プラスチック QFP (□14 mm)

半田付け方式	半田付け条件	推奨条件記号
赤外線リフロ	パッケージ・ピーク温度：235℃、時間：30秒以内 (210℃以上)、回数：1回 制限日数：7日間 ^注 (以降は125℃プリベーク 20時間必要)	IR35-207-1
VPS	パッケージ・ピーク温度：215℃、時間：40秒以内 (200℃以上)、回数：2回以内 制限日数：7日間 ^注 (以降は125℃プリベーク 20時間必要) 〈留意事項〉 (1) 2回目のリフロは1回目のリフロによるデバイス温度が常温に戻ってから開始してください。 (2) 1回目のリフロ後の水によるフラックス洗浄はお避けください。	VP15-207-2
端子部分加熱	端子部温度：300℃以下、時間：3秒以内 (デバイスの一辺当たり)	—

注 ドライパック開封後の保管制限日数で、保管条件は25℃、65%RH以下。

注意 半田付け方式の併用はお避けください (ただし、端子部分加熱方式は除く)。

表 26-2 挿入タイプの半田付け条件

μPD17062CU-XXXX : 48ピン・プラスチック・シュリンク DIP (600 mil)

半田付け方式	半田付け条件
ウェーブ・ソルダリング (リード部のみ)	半田槽温度：260℃以下、時間：10秒以内
端子部分加熱	端子部温度：260℃以下、時間：10秒以内

注意 ウェーブ・ソルダリングはリード部のみとし、噴流半田が直接本体に接触しないようにしてください。

付録 開発ツール

μPD17062 のプログラムを開発するために、次の開発ツールを用意しています。

ハードウェア

名 称	概 要
インサーキット・エミュレータ (IE-17K IE-17K-ET ^{注1} EMU-17K ^{注2})	IE-17K, IE-17K-ET, EMU-17K は, 17K シリーズ共通のインサーキット・エミュレータです。 IE-17K および IE-17K-ET は, ホスト・マシンである PC-9800 シリーズまたは IBM PC/AT TM と RS-232-C を介して接続して使用します。EMU-17K は, ホスト・マシンである PC-9800 シリーズの拡張用スロットに実装して使用します。 各品種専用のシステム・エバリュエーション・ボード (SE ボード) と組み合わせて使用することにより, その品種に対応したエミュレータとして動作します。マン・マシン・インタフェース・ソフトウェアである SIMPLEHOST TM を使用すると, さらに高度なデバッグ環境を実現できます。 また, EMU-17K は, データ・メモリの内容をリアルタイムで確認できるという機能を備えています。
SE ボード (SE-17002)	SE-17002 は, μPD17002, 17062 用の SE ボードです。単体でシステム評価に, インサーキット・エミュレータと組み合わせてデバッグに使用します。
エミュレーション・プローブ (EP-17002CU)	EP-17002CU は, 48ピン・シュリンク DIP (600 mil) 用のエミュレーション・プローブです。SE ボードとターゲット・システムを接続します。
エミュレーション・プローブ (EP-17002GC)	EP-17002GC は, 64ピン QFP (□14 mm) 用のエミュレーション・プローブです。EV-9400GC-64 ^{注3} とともに使用することで, SE ボードとターゲット・システムを接続します。
変換ソケット (EV-9200GC-64 ^{注3})	EP-17002GC とターゲット・システムを接続するために使用します。

注 1. 廉価版：電源外付けタイプ

2. 株式会社アイ・シーの製品です。詳細につきましては, 株式会社アイ・シー (東京(03)3447-3793) までお問い合わせください。
3. EP-17002GC をご購入になると, EV-9200GC-64 が 1 個付属されています。また, EV-9200GC-64 を 5 個 1 組で別売もしています。

ソフトウェア

名称	概要	ホスト・マシン	OS		供給媒体	オーダ名称
17K シリーズ アセンブラ (AS17K)	AS17K は 17K シリーズ共通の アセンブラです。μPD17062 の プログラム開発には、この AS 17K とデバイス・ファイル(AS 17062) を組み合わせて使用し ます。	PC-9800 シリーズ	MS-DOS™		5 インチ 2HD	μS5A10AS17K
					3.5 インチ 2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5 インチ 2HC	μS7B10AS17K
					3.5 インチ 2HC	μS7B13AS17K
デバイス・ファイル (AS17062)	AS17062 は μPD17062 用のデ バイス・ファイルです。17K シリーズ共通のアセンブラ(AS 17K) と組み合わせて使用しま す。	PC-9800 シリーズ	MS-DOS		5 インチ 2HD	μS5A10AS17062
					3.5 インチ 2HD	μS5A13AS17062
		IBM PC/AT	PC DOS		5 インチ 2HC	μS7B10AS17062
					3.5 インチ 2HC	μS7B13AS17062
サポートソフトウェア (SIMPLEHOST)	SIMPLEHOST は、インサー キット・エミュレータとパーソナル コンピュータを用いてプログラム 開発を行うときに Windows™ 上でマン・マシン・インタフェー スを行うソフトウェアです。	PC-9800 シリーズ	MS-DOS	Windows	5 インチ 2HD	μS5A10IE17K
					3.5 インチ 2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5 インチ 2HC	μS7B10IE17K
					3.5 インチ 2HC	μS7B13IE17K

備考 対応している OS のバージョンは次のとおりです。

OS	バージョン
MS-DOS	Ver. 3.30~Ver. 5.00A ^注
PC DOS	Ver. 3.1~Ver. 5.0 ^注
Windows	Ver. 3.0~Ver. 3.1

注 MS-DOS の Ver. 5.00/5.00A, PC DOS の Ver. 5.0 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。

(× ㄇ)

CMOSデバイスの一般的注意事項

①静電気対策 (MOS全般)

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

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

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

②未使用入力の処理 (CMOS特有)

注意 CMOSデバイスの入力レベルは固定してください。

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

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

③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。

○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。

注意：本製品はI²Cバス・インタフェース回路を内蔵しています。

I²Cバス・インタフェースを使用される場合には、カスタム・コードをご発注いただく時に、事前にその旨ご申告下さい。申告に基づき、以下の特典が受けられます。

日本電気株式会社のI²Cバス対応部品をご購入いただくことにより、これらの部品をI²Cシステムに使用する実施権がフィリップス社I²C特許に基づき許諾されることとなります。ただし、これらのI²Cシステムはフィリップス社によって設定されたI²C標準規格に合致しているものとします。

Purchase of NEC I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

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(仕様を決定した者)が判定してください。

- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- この製品を使用したことにより、第三者の工業所有権等にかかわる問題が発生した場合、当社製品の構造製法に直接かかわるもの以外につきましては、当社はその責を負いませんのでご了承ください。
- 当社は、航空宇宙機器、海底中継器、原子力制御システム、生命維持のための医療用機器などに推奨できる製品を標準的には用意しておりません。当社製品をこれらの用途にご使用をお考えのお客様、および、『標準』品質水準品を当社が意図した用途以外にご使用をお考えのお客様は、事前に販売窓口までご連絡頂きますようお願い致します。

当社推奨の用途例

標準：コンピュータ、OA機器、通信機器、計測機器、工作機械、産業用ロボット、AV機器、家電等
 特別：輸送機器(列車、自動車等)、交通信号機器、防災/防犯装置等

- この製品は耐放射線設計をしておりません。

M4 92.6

SIMPLEHOSTは日本電気株式会社の商標です。

MS-DOS, Windowsは、米国マイクロソフト社の商標です。

PC/AT, PC DOSは米国IBM社の商標です。

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

コンシューマ半導体販売事業部 OA半導体販売事業部 インダストリアル半導体販売事業部	〒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 半導体販売部	〒460 名古屋市中区栄四丁目14番5号(松下中ビル)	名古屋 (052)242-2755
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 札幌 (011)231-0161 東北支社 仙台 (022)261-5511 岩手支店 盛岡 (0196)51-4344 山形支店 山形 (0236)23-5511 山形支店 郡山 (0249)23-5511 いわき支店 いわき (0246)21-5511 長岡支店 長岡 (0258)36-2155 水戸支店 水戸 (0292)26-1717 神奈川支社 横浜 (045)324-5511 群馬支店 高崎 (0273)26-1255 群馬支店 太田 (0276)46-4011 宇都宮支店 宇都宮 (0286)21-2281	小山支店 小山 (0285)24-5011 長野支店 長野 (0262)35-1444 松本支店 松本 (0263)35-1666 上諏訪支店 諏訪 (0266)53-5350 甲府支店 甲府 (0552)24-4141 埼玉支店 埼玉 (048)641-1411 立川支店 立川 (0425)26-5981 千葉支店 千葉 (043)238-8116 静岡支店 静岡 (054)255-2211 沼津支店 沼津 (0559)63-4455 浜松支店 浜松 (053)452-2711 北陸支店 金沢 (0762)23-1621	福井支店 福井 (0778)22-1866 富山支店 富山 (0764)31-8461 京都支店 京都 (075)344-7824 神戸支店 神戸 (078)332-3311 中国支店 広島 (082)242-5504 鳥取支店 鳥取 (0857)27-5311 岡山支店 岡山 (086)225-4455 四国支店 高松 (0878)36-1200 新居浜支店 新居浜 (0897)32-5001 松山支店 松山 (0899)45-4111 九州支店 福岡 (092)271-7700 北九州支店 北九州 (093)541-2887

【本資料に関する技術お問い合わせ先】

コンシューマ半導体販売事業部 AVシステム技術部	〒210 川崎市幸区塚越三丁目484番地	川崎 (044)548-8886	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)
半導体応用技術本部	〒460 名古屋市中区栄四丁目14番5号(松下中ビル)	名古屋 (052)242-2762	
中部応用システム技術部	〒460 名古屋市中区栄四丁目14番5号(松下中ビル)	名古屋 (052)242-2762	
半導体応用技術本部	〒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	