

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。

4ビット・シングルチップ・マイクロコントローラ

μ PD17108Lは、ROM1Kバイト (512×16)、RAM16×4ビット、I/Oポート16本で構成されているタイニ・マイクロコントローラです。

電源電圧1.5Vからの低電圧での動作が可能のため、リチウム電池1本または乾電池2本を電源とする製品への応用にたいへん便利です。

CPUには、直接データ・メモリを操作することのできる汎用レジスタを取り入れた17Kアーキテクチャを採用しているため、たいへん効率のよいプログラミングが可能です。また、すべての命令は16ビット長1語で構成されています。

特 徴

- プログラム・メモリ (ROM) : 1Kバイト (512×16ビット)
- データ・メモリ (RAM) : 16×4ビット
- I/Oポート : 16本 (N-ch オープン・ドレイン出力 4本)
- 命令実行時間 : 40 μ s (200 kHz動作時)
- 命令種類 : 24種類 (すべて1語長)
- スタック・レベル : 1レベル
- スタンバイ機能 (STOP, HALT モード)
- 低電圧でのデータ・メモリのデータ保持可能 (MIN. 1.5 V)
- システム・クロック用 RC 発振回路… コンデンサ内蔵タイプ (外付けは抵抗のみ)
- 動作電源電圧 : 1.5 ~ 3.6 V (200 kHz 動作時)

用 途

- 家電製品やおもちゃなどの電子制御化

オーダ情報

| オーダ名称 | パッケージ | 品質水準 |
|--|---------------------------------|--------------|
| μ PD17108LCS- $\times\times\times$ | 22ピン・プラスチック・シュリンク DIP (300 mil) | 標準 (一般電子機器用) |
| μ PD17108LGS- $\times\times\times$ | 24ピン・プラスチック SOP (300 mil) | // |

備考 $\times\times\times$ はROMコード番号です。

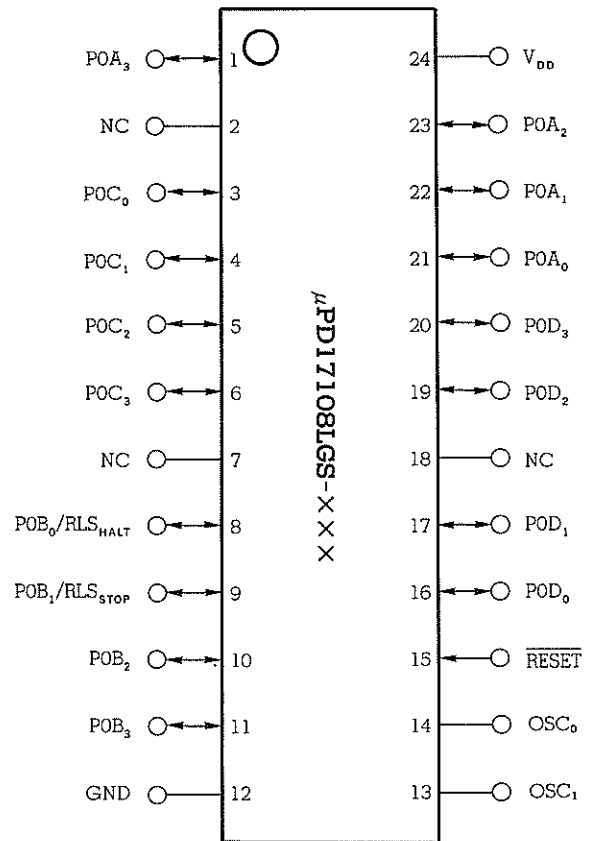
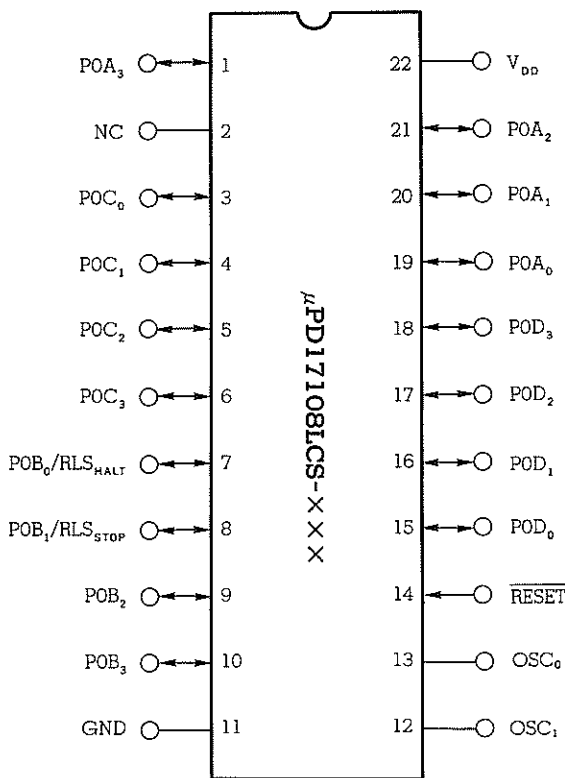
品質水準とその応用分野の詳細については当社発行の資料「NEC 半導体デバイスの品質水準」(IEI-620)をご覧ください。

本資料の内容は、後日変更する場合があります。

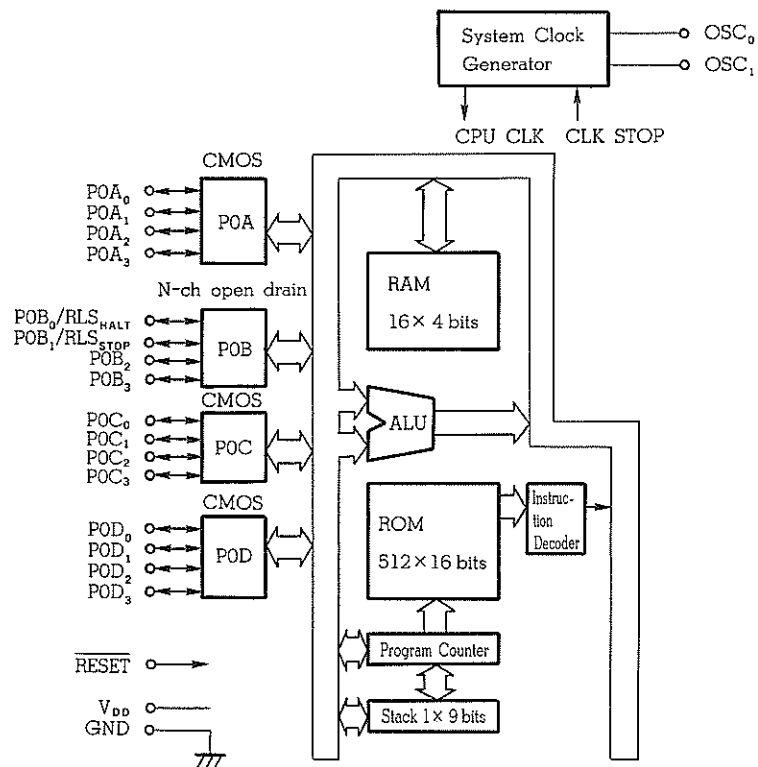
端子接続図 (Top View)

22ピン・プラスチック・シュリンク DIP

24ピン・プラスチック SOP



ブロック図



端子機能

端子機能一覧

○ポート端子

| 端子名称 | 入出力 | 機 能 | リセット時 |
|---------------------------------------|-----|--|--|
| POA ₀ - POA ₃ | 入出力 | CMOS (プッシュプル) 4ビット入出力ポート (ポート 0A) | ハイ・インピーダンス (入力モード) |
| POB ₀ /RLS _{HALT} | 入出力 | HALT モード解除用 | <ul style="list-style-type: none"> • オープン・ドレイン時 ハイ・インピーダンス (入力モード) • プルアップ抵抗選択時 ハイ・レベル (入力モード) |
| POB ₁ /RLS _{STOP} | | STOP モード解除用 | |
| POB ₂ , POB ₃ | 入出力 | <ul style="list-style-type: none"> • N-ch オープン・ドレイン 4ビット入出力ポート (ポート 0B) • ビット単位で内蔵プルアップ抵抗選択可能(マスク・オプション) • オープン・ドレイン時 9V 耐圧 | |
| POC ₀ - POC ₃ | 入出力 | CMOS (プッシュプル) 4ビット入出力ポート (ポート 0C) | ハイ・インピーダンス (入力モード) |
| POD ₀ - POD ₃ | 入出力 | CMOS (プッシュプル) 4ビット入出力ポート (ポート 0D) | ハイ・インピーダンス (入力モード) |

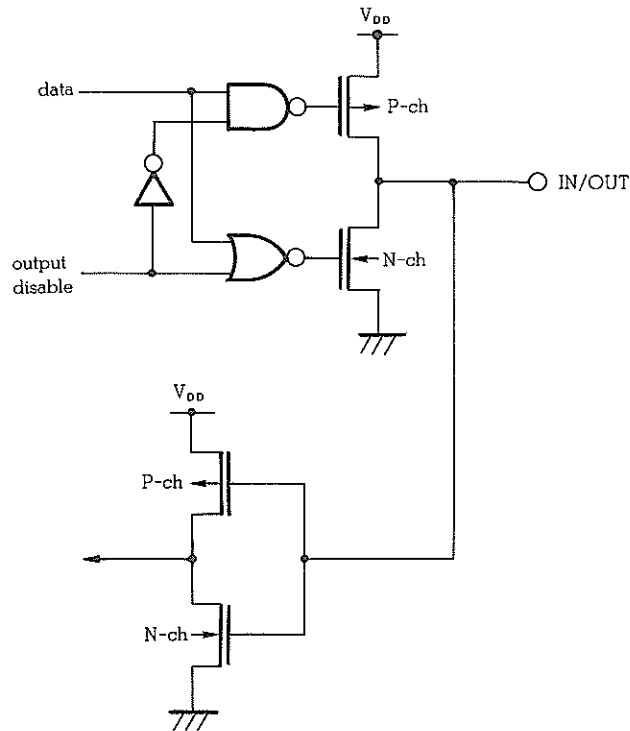
○ポート端子以外

| 端子名称 | 入出力 | 機 能 |
|-------------------------------------|-----|--|
| RESET | 入 力 | <ul style="list-style-type: none"> • システム・リセット入力端子 • 内蔵プルアップ抵抗選択可能 (マスク・オプション) |
| V _{DD} | — | 正電源端子 |
| GND | — | GND 電位端子 |
| OSC ₀ , OSC ₁ | — | システム・クロック発振用抵抗接続端子 |

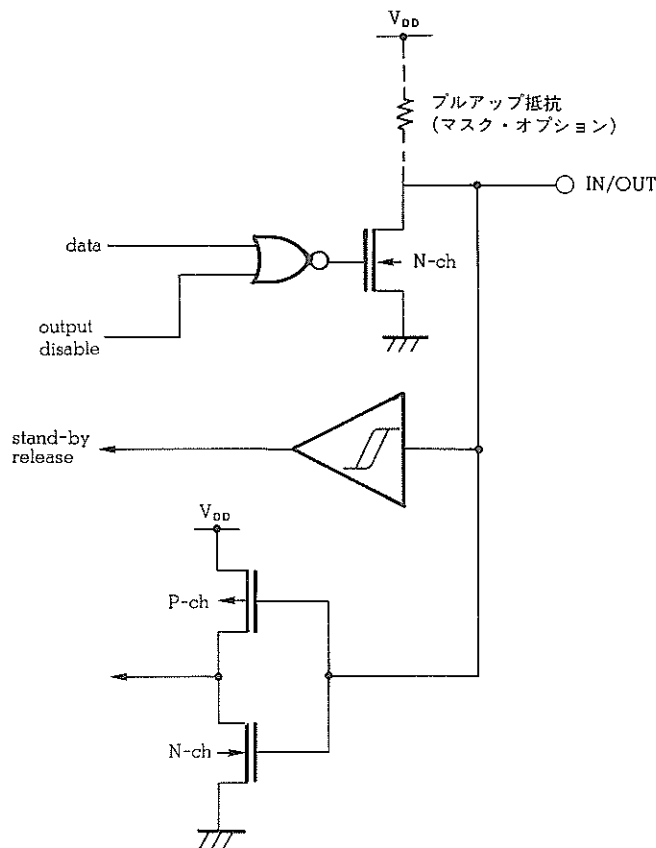
端子の等価回路

μPD17108Lの各端子の等価回路を一部簡略化した形式を用いて示します。

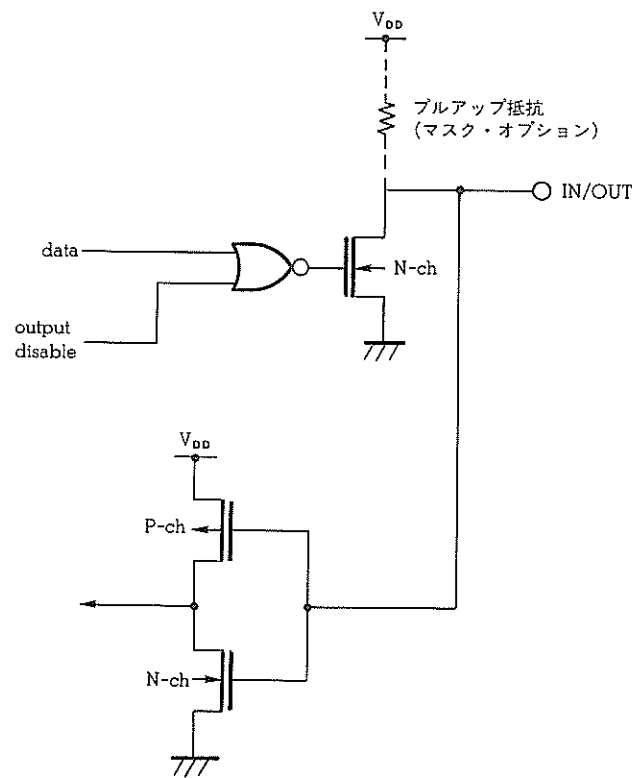
(1) POA, POC, POD



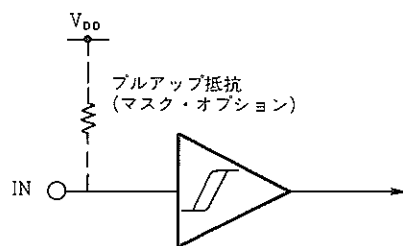
(2) POB₀, POB₁



(3) POB₂, POB₃



(4) RESET



★ 未使用端子の処理

未使用端子は、以下のように処理をしてください。

| 端子名 | | 処理方法 | | |
|--------------------------|--------------------------------------|--|---------------------------|--|
| | | マイコン内部 | マイコン外部 | |
| ポート | 入力モード | POA, POC, POD | — | プルアップ抵抗を介して V_{DD} に接続または、プルダウン抵抗を介して GND に接続 ^注 |
| | | POB | マスク・オプションによるプルアップ抵抗を内蔵しない | |
| | マスク・オプションによるプルアップ抵抗を内蔵する | | | |
| | 出力モード | POA, POC, POD (CMOSポート) | — | オープン |
| POB (N-ch オープン・ドレイン・ポート) | | マスク・オプションによるプルアップ抵抗を内蔵しないで、ロウ・レベルを出力する | | |
| | マスク・オプションによるプルアップ抵抗を内蔵して、ハイ・レベルを出力する | | | |

注 外部でプルアップまたはプルダウンする場合には、ポートのドライブ能力や消費電流に注意してください。また、高い抵抗値でプルアップまたはプルダウンする場合には、その端子にノイズが乗らないように注意してください。応用回路にもよりますが、プルアップまたはプルダウンの抵抗値は、数十kΩ程度が一般的です。

注意 端子の出力レベルは、プログラムの各ループ内で繰り返し設定することによって固定することを推奨します。

RESET 端子の使用上の注意

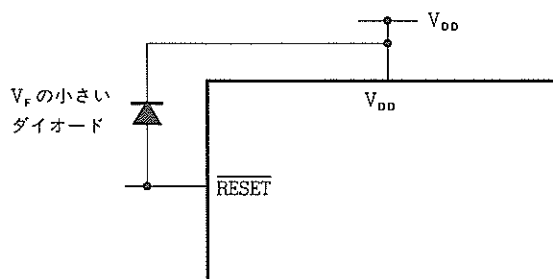
RESET 端子は、端子機能に示した機能のほかに、μPD17108L の内部動作をテストするテスト・モードを設定する機能 (IC テスト専用) を持っています。

この端子に V_{DD} を越える電圧を印加すると、テスト・モードに設定されます。このため、通常動作時であっても V_{DD} を越えるようなノイズが加わった場合にはテスト・モードに入ってしまう、通常動作に支障をきたすことがあります。

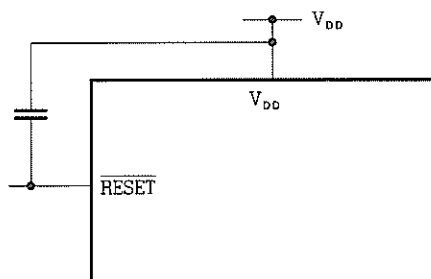
たとえば、RESET 端子の配線の引き回しが長い場合などでは、RESET 端子に布線間ノイズが加わって上記の問題を起こしてしまうことがあります。

したがって、できるだけ布線間ノイズを抑えるような配線を行ってください。どうしてもノイズが抑えられない場合には、下図のような外付け部品によるノイズ対策を実施してください。

○ V_{DD} との間に V_F の小さいダイオードを接続



○ V_{DD} との間にコンデンサを接続



目 次

| | |
|---|---|
| 1. プログラム・カウンタ (PC) … 9 | |
| 1.1 プログラム・カウンタ (PC) の構成 … 9 | |
| 1.2 プログラム・カウンタ (PC) の機能 … 9 | |
| 2. スタック … 10 | |
| 3. プログラム・メモリ (ROM) … 11 | |
| 4. データ・メモリ (RAM) … 12 | |
| 4.1 データ・メモリ (RAM) の構成 … 12 | |
| 4.1.1 汎用データ・メモリの機能 … 12 | |
| 4.1.2 ジェネラル・レジスタの機能 … 12 | |
| 4.1.3 ポート・レジスタの機能 … 12 | |
| 4.1.4 システム・レジスタの機能 … 13 | |
| 5. ALUブロック … 16 | ★ |
| 5.1 ALUブロックの構成 … 16 | |
| 5.2 ALUブロックの機能 … 16 | |
| 5.2.1 ALUの機能 … 16 | |
| 5.2.2 一時記憶レジスタ A および B の機能 … 20 | |
| 5.2.3 ステータス・フリップフロップの機能 … 20 | |
| 5.2.4 2進4ビット演算 … 21 | |
| 5.2.5 BCD 演算 … 21 | |
| 5.2.6 ALU ブロック処理手順 … 22 | |
| 5.3 算術演算 (2進4ビット加減算およびBCD加減算) … 23 | |
| 5.3.1 CMP フラグ = 0, BCD フラグ = 0 のときの加減算 … 23 | |
| 5.3.2 CMP フラグ = 1, BCD フラグ = 0 のときの加減算 … 23 | |
| 5.3.3 CMP フラグ = 0, BCD フラグ = 1 のときの加減算 … 24 | |
| 5.3.4 CMP フラグ = 1, BCD フラグ = 1 のときの加減算 … 24 | |
| 5.3.5 算術演算使用時の注意 … 25 | |
| 5.4 論理演算 … 25 | |
| 5.5 ビット判断 … 26 | |
| 5.5.1 True ビット (1) 判断 … 26 | |
| 5.5.2 False ビット (0) 判断 … 27 | |
| 5.6 比較判断 … 27 | |
| 5.6.1 “等しい” の判断 … 28 | |
| 5.6.2 “等しくない” の判断 … 28 | |
| 5.6.3 “以上” の判断 … 29 | |
| 5.6.4 “未満” の判断 … 29 | |
| 5.7 回転処理 … 30 | |
| 5.7.1 右回転処理 … 30 | |

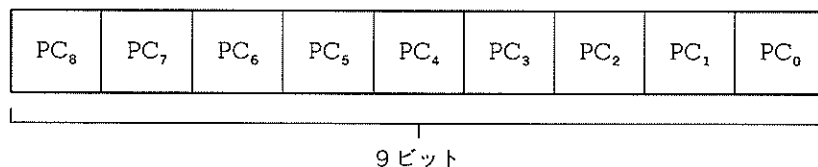
| | | |
|-------|---|----|
| | 5.7.2 左回転処理 | 31 |
| 6. | ポート | 32 |
| 6.1 | ポート 0A (POA ₀ - POA ₃) | 32 |
| 6.2 | ポート 0B (POB ₀ /RLS _{HALT} , POB ₁ /RLS _{STOP} , POB ₂ , POB ₃) | 32 |
| 6.3 | ポート 0C (POC ₀ - POC ₃) | 32 |
| 6.4 | ポート 0D (POD ₀ - POD ₃) | 33 |
| ★ | 6.5 ポート・レジスタの操作時の注意 | 34 |
| 7. | スタンバイ機能 | 35 |
| 7.1 | HALT モード | 35 |
| 7.2 | STOP モード | 35 |
| 7.3 | スタンバイ・モードの設定と解除 | 35 |
| ★ | 7.4 スタンバイ・モード中のハードウェアの状態 | 36 |
| 7.5 | スタンバイ・モードの解除タイミング | 36 |
| 8. | リセット | 38 |
| 8.1 | システム・リセット | 38 |
| 9. | アセンブラ予約語 | 39 |
| 9.1 | マスク・オプション疑似命令 | 39 |
| 9.1.1 | OPTION, ENDOP 疑似命令 | 39 |
| 9.1.2 | マスク・オプション定義疑似命令 | 39 |
| 9.2 | 予約シンボル | 41 |
| 10. | 命令セット | 42 |
| 10.1 | 命令セット一覧表 | 42 |
| 10.2 | 命令一覧表 | 43 |
| ★ | 10.3 アセンブラ (AS17K) 組み込みマクロ命令 | 45 |
| 11. | 電気的特性 | 46 |
| 12. | 特性曲線 (参考値) | 49 |
| 13. | 外形図 | 52 |
| 14. | 半田付け推奨条件 | 55 |
| 15. | μ PD17108L と μ PD17108, 17P108 の違い | 56 |
| 16. | タイニ・マイクロコントローラ・ファミリー一覧表 | 56 |
| 付録 | 開発ツール | 57 |

1. プログラム・カウンタ (PC)

1.1 プログラム・カウンタ (PC) の構成

プログラム・カウンタは図 1-1 に示すように、9 ビットのバイナリ・カウンタで構成されています。

図 1-1 プログラム・カウンタ



1.2 プログラム・カウンタ (PC) の機能

プログラム・カウンタはプログラム・メモリ (ROM) すなわちプログラムのアドレスを指定するカウンタです。

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 (BR)、サブルーチン・コール命令 (CALL)、リターン命令 (RET) 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、その番地の命令が実行されます。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 (No Operation) 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。

2. スタック

μPD17108Lのスタックは、サブルーチン・コール時にプログラムの戻り番地を退避するためのレジスタで、1レベルあります。

図2-1にPC、スタックおよびBR、CALL命令のオペランドの関係を示します。

図2-1 PC、スタックおよびBR、CALL命令のオペランドの関係

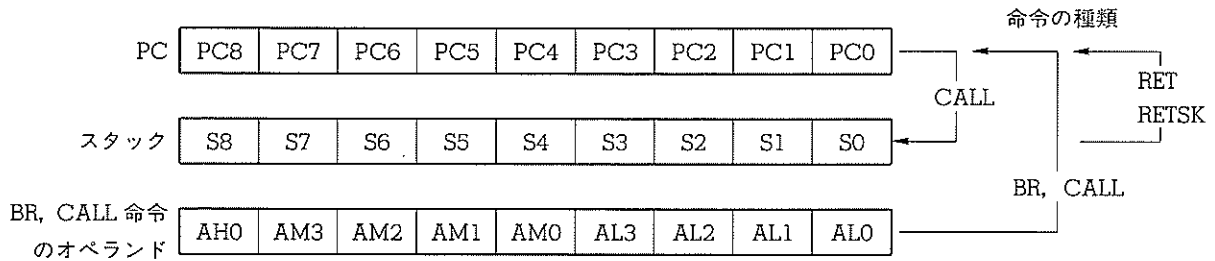
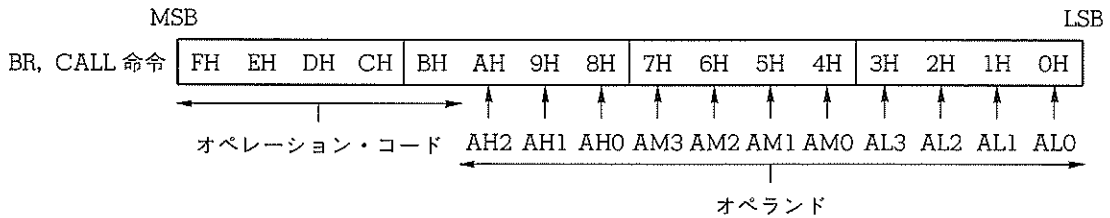


図2-1において、AHn, AMn, ALn(0 ≤ n ≤ 3)とは、それぞれ16ビット命令の以下の部分のことを言います。

図2-2 16ビット命令の構成



なお、アセンブラ (AS17K) を使用せず BR, CALL 命令を使用するときには AH2, AH1 は必ず "0" を指定してください。

リセット入力時には、プログラム・カウンタは全ビット "0" にクリアされます。

3. プログラム・メモリ (ROM)

図3-1にプログラム・メモリ (ROM) の構成を示します。

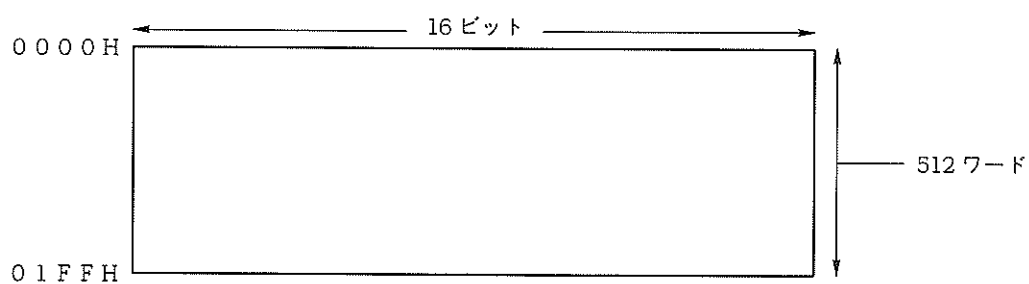
図3-1に示すように、512ワード×16ビットで構成されています。

プログラム・メモリ (ROM) は16ビット単位で“番地 (アドレス)” が付けられており、0000H 番地から 01FFH 番地まであり、プログラム・カウンタ (PC) により番地を指定します。

“命令” はすべて16ビット長の“一語命令” であるため、プログラム・メモリ (ROM) の1つの番地に1つの命令を格納することができます。

0000H 番地は、リセット・スタート番地になっています。

図3-1 プログラム・メモリ・マップ



4. データ・メモリ (RAM)

データ・メモリ (RAM) は、演算、制御など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、読み出しが行えます。

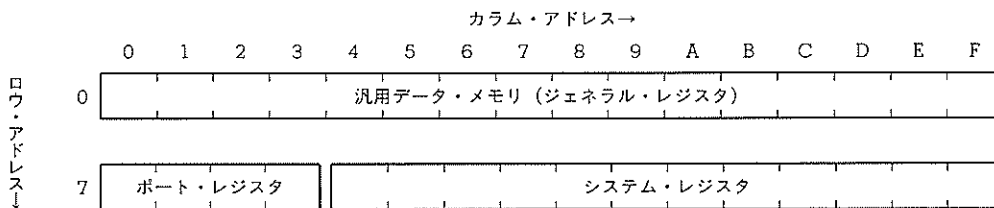
4.1 データ・メモリ (RAM) の構成

図 4-1 にデータ・メモリ (RAM) の構成を示します。

データ・メモリは4ビット単位“1 ニブル”で構成されており、それぞれの4ビット単位のデータごとに番地 (アドレス) が割り付けられています。アドレスは上位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。

また、データ・メモリは機能別に以下に示す汎用データ・メモリ、ポート・レジスタ、およびシステム・レジスタの各ブロックに分かれます。

図 4-1 データ・メモリの構成



4.1.1 汎用データ・メモリの機能

汎用データ・メモリは、データ・メモリからシステム・レジスタ (SYSREG) とポート・レジスタを除いた部分です。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。

4.1.2 ジェネラル・レジスタの機能

ジェネラル・レジスタは、システム・レジスタのレジスタ・ポインタ (RP) で指定されるデータ・メモリ中の任意の同一ロウ・アドレス (16 ニブル) を指します。μPD17108Lでは、レジスタ・ポインタが“0”に固定されていますので、汎用データ・メモリが同時にジェネラル・レジスタとしても使用できます。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。

4.1.3 ポート・レジスタの機能

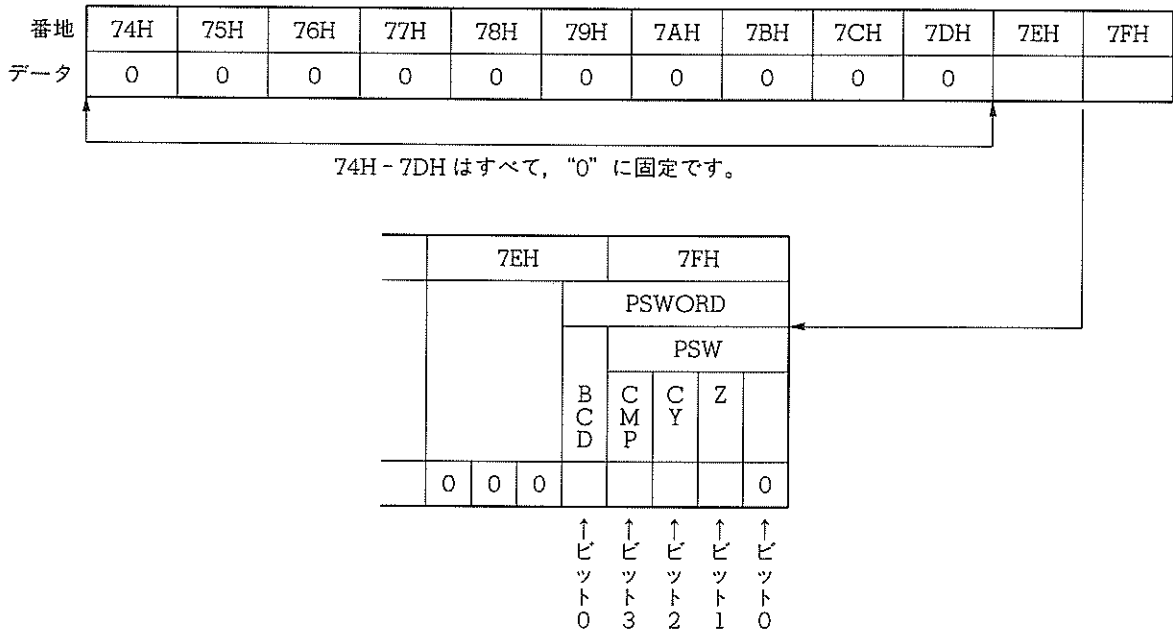
ポート・レジスタは、入出力ポートの出力データの設定や、入力データの読み込みを行うときに使用します。

ポートに対応するポート・レジスタに一度でもデータを書き込むと、そのポートは出力モードに設定され、値を書き換えなにかぎりその値を出力し続けます (リセットをかけないかぎり出力モードは維持されます)。また、入出力どちらのモードにもかかわらず、ポート・レジスタに対し読み込みを実行すると、ポート・レジスタの値ではなく常に端子の状態が読めます。

4.1.4 システム・レジスタの機能

システム・レジスタは、CPUの制御を行うレジスタです。μPD17108Lのシステム・レジスタはプログラム・ステータス・ワード (PSWORD) だけです。

図 4-2 システム・レジスタ・マップ

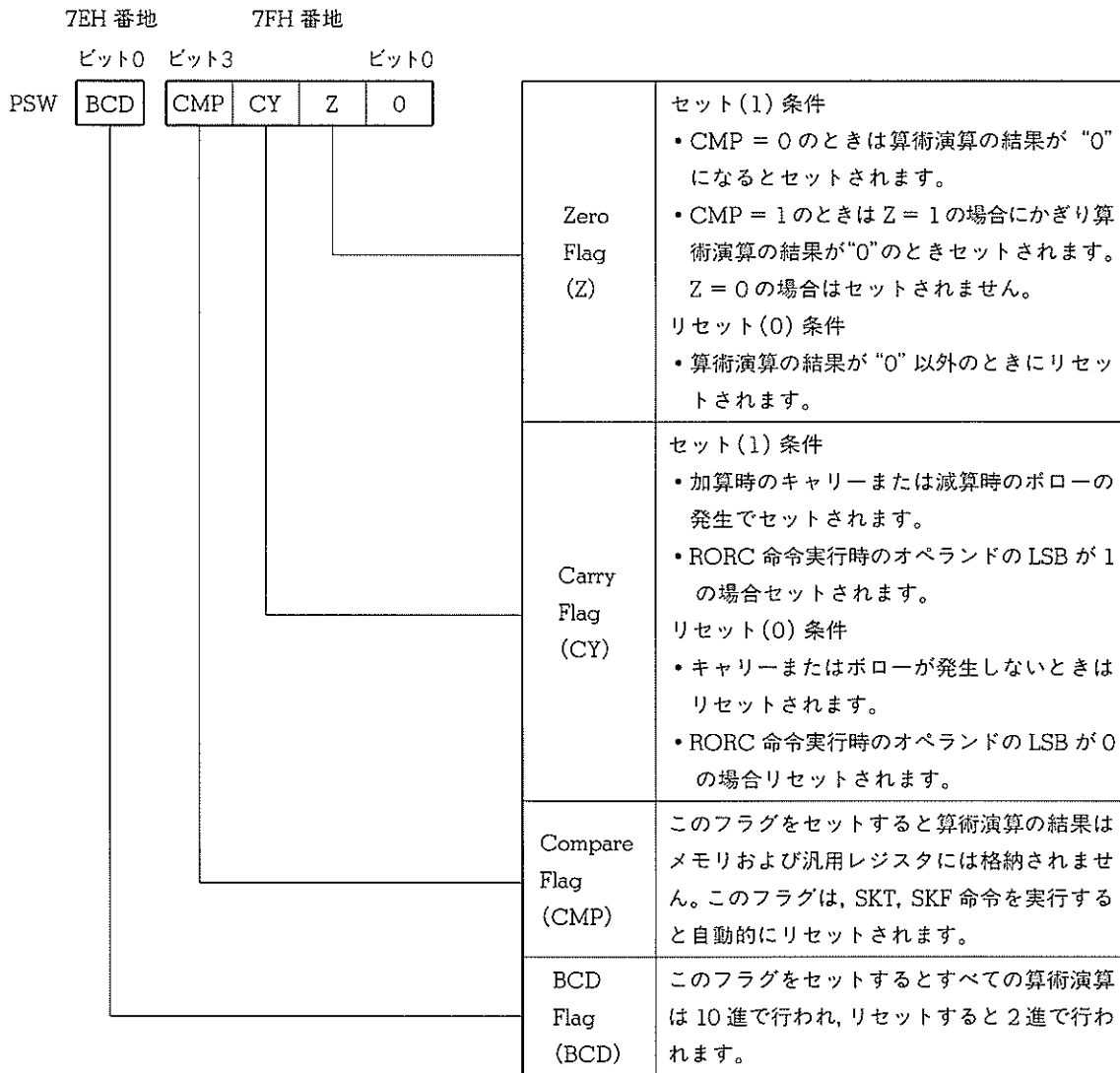


7FH番地 (PSW) の4ビットと7EH番地のビット0は、プログラム・ステータス・ワードに割り当てられています。

7EH番地のビット0はBCDフラグ、7FH番地のビット3はCMPフラグ、ビット2はCYフラグ、ビット1はZフラグがマッピングされています。

また、7EH番地の上位3ビットと7FH番地のビット0は、“0”に固定されています。

図 4-3 プログラム・ステータス・ワードの構成



CY フラグは比較命令 (SKE/SKNE/SKGE/SKLT) では変化しませんが、算術演算時にはたとえ CMP フラグがセットされていても演算の結果により変化します。

なお、プログラム・ステータス・ワードの各ビットは、リセット信号入力時に "0" に初期化されます。

プログラム・ステータス・ワード内の Z フラグは、CMP フラグの値によって表 4-1 のように変化します。

表 4-1 Z フラグの変化

★

| 条 件 | CMP = 0 のとき | CMP = 1 のとき |
|---------------------|-------------|-------------|
| 算術演算の結果が“0”になったとき | Z ← 1 | Z は変化しない |
| 算術演算の結果が“0”以外になったとき | Z ← 0 | Z ← 0 |

CMP = 1 にセットされている場合、Z フラグの値が“1”のときに算術演算の結果が OH になると Z フラグの値は“1”に保持されたままですが、演算結果がいったん OH 以外になると Z フラグの値は“0”にリセットされ、再び算術演算をしてその結果が OH になっても Z フラグの値は“1”にセットされません。

すなわち、CMP フラグと Z フラグを“1”にセットして複数回の比較作業（減算比較）を行ったあと、Z フラグを参照して Z フラグが“1”のままであればすべての比較結果が 0（一致）になっていたことを意味し、“0”になっていれば比較結果が 1 回以上異なっていたことを意味します。

12ビット・データの比較の例

； M001, M002, M003 に格納された 12ビットのデータが、456H に等しいか？

CMP456 :

```

SET2  CMP, Z
SUB   M001, #4 ; M001, M002, M003 に格納された
SUB   M002, #5 ; データは壊れない。
SUB   M003, #6 ;
; CLR1  CMP
SKT   Z ; CMP はビット判断命令で自動的にクリア
BR    DIFFER ; ≠ 456H
BR    AGREE ; = 456H

```

★ 5. ALU ブロック

ALU は 4 ビット・データの算術演算、論理演算、ビット判断、比較判断および回転処理を行います。

5.1 ALU ブロックの構成

図 5-1 に ALU ブロックの構成を示します。

図 5-1 に示すように ALU ブロックは 4 ビットのデータ処理を行う ALU 本体と、ALU の周辺回路である一時記憶用レジスタ A、B と、ALU の状態を制御するステータス・フリップフロップと、BCD 演算使用時の 10 進補正回路から構成されています。

ステータス・フリップフロップは図 5-1 に示すように、ゼロ・フラグ用 FF、キャリー・フラグ用 FF、コンペア・フラグ用 FF および BCD フラグ用 FF から構成されています。

ステータス・フリップフロップはシステム・レジスタの中のプログラム・ステータス・ワード (PSWORD: アドレス 7EH, 7FH) の各フラグであるゼロ・フラグ (Z), キャリー・フラグ (CY), コンペア・フラグ (CMP) および BCD フラグ (BCD) と 1 対 1 に対応しています。

5.2 ALU ブロックの機能

ALU はプログラムに書かれた命令により、算術演算、論理演算、ビット判断、比較判断および回転処理を行います。表 5-1 に各演算、判断、および回転処理命令の一覧を示します。

表 5-1 に示した各命令を実行することにより 4 ビット単位の演算、判断および回転処理または 1 桁の BCD 演算が 1 命令で実行できます。

5.2.1 ALU の機能

算術演算には加算と減算があります。算術演算はジェネラル・レジスタの内容とデータ・メモリの内容との演算、またはデータ・メモリの内容とイミディエト・データとの演算が行えます。また、算術演算は 2 進数による 4 ビットの演算と 10 進数による 1 桁の演算 (BCD 演算) が可能です。

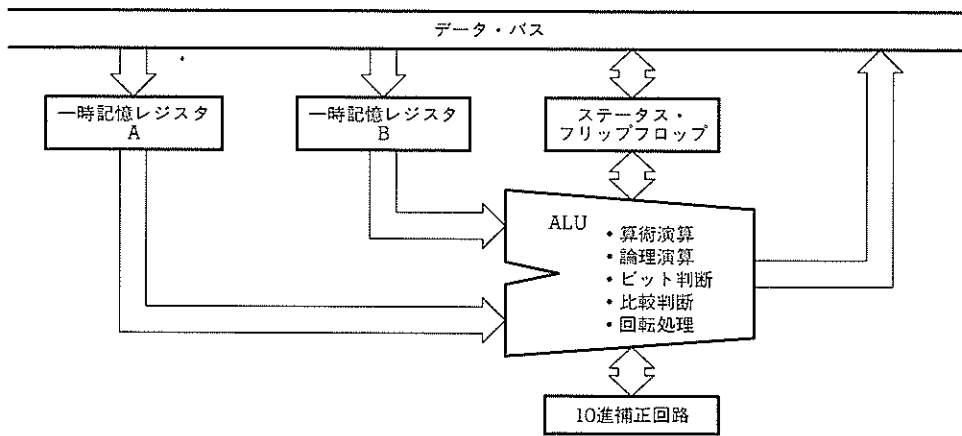
論理演算には論理積 (AND), 論理和 (OR) および排他的論理和 (XOR) があります。論理演算は、ジェネラル・レジスタの内容とデータ・メモリの内容との演算、またはデータ・メモリの内容とイミディエト・データとの演算が行えます。

ビット判断は、データ・メモリの 4 ビット・データのうち “0” であるビットまたは “1” であるビットの判断を行います。

比較判断はデータ・メモリの内容とイミディエト・データとの比較を行い、“等しい”, “等しくない”, “以上” および “未満” の判断を行います。

回転処理はジェネラル・レジスタの 4 ビット・データを下位ビットの方向へ 1 ビット、シフトします (右へ回転する)。

図 5-1 ALU ブロックの構成



| | | | | | |
|------|--------------------------|----------------|----------------|----------------|----------------|
| アドレス | 7EH | 7FH | | | |
| 名称 | プログラム・ステータス・ワード (PSWORD) | | | | |
| ビット | b ₀ | b ₃ | b ₂ | b ₁ | b ₀ |
| フラグ | BCD | CMP | CY | Z | 0 |

| | | | |
|-------------------|-------------------|------------------|-----------------|
| ステータス・フリップフロップ | | | |
| BCD フラグ用 FF | CMP フラグ用 FF | CY フラグ用 FF | Z フラグ用 FF |

| | |
|-------|----------------------|
| 機能の概要 | |
| → | 算術演算結果が 0 であることを示す |
| → | 算術演算時のキャリーまたはボローを格納 |
| → | 算術演算結果を格納するかを指定 |
| → | 算術演算時に 10 進補正を行うかを指定 |

表 5 - 1 ALU 処理命令一覧 (1/2)

| ALU 機能 | 命 令 | 動 作 | 説 明 |
|-----------------------|----------------------------|--|--|
| 算 術 演 算 | 加 算 | ADD r, m | $(r) \leftarrow (r) + (m)$ ジェネラル・レジスタとデータ・メモリの内容を加算。 結果をジェネラル・レジスタへ格納。 |
| | | ADD m, #n4 | $(m) \leftarrow (m) + n4$ データ・メモリとイミディエト・データの内容を加算。 結果をデータ・メモリへ格納。 |
| | | ADDC r, m | $(r) \leftarrow (r) + (m) + CY$ ジェネラル・レジスタとデータ・メモリの内容を CY フラグと ともに加算。結果をジェネラル・レジスタへ格納。 |
| | | ADDC m, #n4 | $(m) \leftarrow (m) + n4 + CY$ データ・メモリとイミディエト・データの内容を CY フラグ とともに加算。結果をデータ・メモリへ格納。 |
| | 減 算 | SUB r, m | $(r) \leftarrow (r) - (m)$ ジェネラル・レジスタの内容からデータ・メモリの内容を減 算。結果をジェネラル・レジスタへ格納。 |
| | | SUB m, #n4 | $(m) \leftarrow (m) - n4$ データ・メモリの内容からイミディエト・データを減算。 結果をデータ・メモリへ格納。 |
| | | SUBC r, m | $(r) \leftarrow (r) - (m) - CY$ ジェネラル・レジスタの内容からデータ・メモリの内容と CY フラグを減算。結果をジェネラル・レジスタへ格納。 |
| | | SUBC m, #n4 | $(m) \leftarrow (m) - n4 - CY$ データ・メモリの内容からイミディエト・データと CY フラ グを減算。結果をデータ・メモリへ格納。 |
| 論 理 演 算 | 論 理 和 | OR r, m | $(r) \leftarrow (r) \vee (m)$ ジェネラル・レジスタとデータ・メモリの内容を OR。 結果をジェネラル・レジスタへ格納。 |
| | | OR m, #n4 | $(m) \leftarrow (m) \vee n4$ データ・メモリとイミディエト・データの内容を OR。 結果をデータ・メモリへ格納。 |
| | 論 理 積 | AND r, m | $(r) \leftarrow (r) \wedge (m)$ ジェネラル・レジスタとデータ・メモリの内容を AND。 結果をジェネラル・レジスタへ格納。 |
| | | AND m, #n4 | $(m) \leftarrow (m) \wedge n4$ データ・メモリとイミディエト・データの内容を AND。 結果をデータ・メモリへ格納。 |
| | 排 他 的 論 理 和 | XOR r, m | $(r) \leftarrow (r) \oplus (m)$ ジェネラル・レジスタとデータ・メモリの内容を XOR。 結果をジェネラル・レジスタへ格納。 |
| | | XOR m, #n4 | $(m) \leftarrow (m) \oplus n4$ データ・メモリとイミディエト・データの内容を XOR。 結果をデータ・メモリへ格納。 |
| ビ ッ ト 判 断 | True | SKT m, #n CMP ← 0, if (m) ∧ n = n, then skip | データ・メモリの内容のうち, n で指定されたビットがすべて True (1) ならスキップ。結果は格納されない。 |
| | False | SKF m, #n CMP ← 0, if (m) ∧ n = 0, then skip | データ・メモリの内容のうち, n で指定されたビットがすべて False (0) ならスキップ。結果は格納されない。 |
| 比 較 判 断 | 等 しい | SKE m, #n4 $(m) - n4$, skip if zero | データ・メモリの内容がイミディエト・データと等しいと きスキップ。結果は格納されない。 |
| | 等 しく ない | SKNE m, #n4 $(m) - n4$, skip if not zero | データ・メモリの内容がイミディエト・データと等しくな いときスキップ。結果は格納されない。 |
| | 以 上 | SKGE m, #n4 $(m) - n4$, skip if not borrow | データ・メモリの内容がイミディエト・データより以上の ときスキップ。結果は格納されない。 |
| | 未 満 | SKLT m, #n4 $(m) - n4$, skip if borrow | データ・メモリの内容がイミディエト・データより未満の ときスキップ。結果は格納されない。 |
| 回 転 | 右 回 転 | RORC r $\rightarrow CY \rightarrow (r)_{b3} \rightarrow (r)_{b2} \rightarrow (r)_{b1} \rightarrow (r)_{b0}$ | ジェネラル・レジスタの内容を CY フラグとともに右へ回転。 結果をジェネラル・レジスタへ格納。 |

表 5 - 1 ALU 処理命令一覧 (2/2)

| ALU 機能 | プログラム・ステータス・ワード (PSWORD) による動作のちがい | | | | |
|------------------|------------------------------------|--------------------|--------------------|---|-----------------------------------|
| 算 術 演 算 | BCD フラグの値 | CMP フラグの値 | 演算動作 | CY フラグ | Z フラグ |
| | 0 | 0 | 2 進演算 結果を格納する | キャリー または ポローの 発生で セット, 発生しな ければ リセット | 演算結果 0000B でセット 0000B 以外はリセット |
| | 0 | 1 | 2 進演算 結果を格納しない | | 演算結果 0000B で状態保持 0000B 以外はリセット |
| | 1 | 0 | 10 進演算 結果を格納する | | 演算結果 0000B でセット 0000B 以外はリセット |
| | 1 | 1 | 10 進演算 結果を格納しない | | 演算結果 0000B で状態保持 0000B 以外はリセット |
| | | | | | |
| 論 理 演 算 | Don't care (保持) | Don't care (保持) | 変わらない | Don't care (保持) | Don't care (保持) |
| | | | | | |
| ビット 判 断 | Don't care (保持) | リセット される | 変わらない | Don't care (保持) | Don't care (保持) |
| | | | | | |
| 比 較 判 断 | Don't care (保持) | Don't care (保持) | 変わらない | Don't care (保持) | Don't care (保持) |
| | | | | | |
| 回 転 | Don't care (保持) | Don't care (保持) | 変わらない | ジェネラル レジスタの b ₀ の値 | Don't care (保持) |
| | | | | | |

5.2.2 一時記憶レジスタ A および B の機能

一時記憶レジスタ A および B は 4 ビット・データを一度に処理するために必要なレジスタであり、処理されるデータと、処理するデータを一時的に蓄えておくレジスタです。

5.2.3 ステータス・フリップフロップの機能

ステータス・フリップフロップは ALU の動作制御および、処理されたデータの状態を格納するフリップフロップです。ステータス・フリップフロップはシステム・レジスタのプログラム・ステータス・ワード (PSWORD) の各フラグと 1 対 1 に対応しているため、システム・レジスタを操作すれば、ステータス・フリップフロップも同時に操作されます。次にプログラム・ステータス・ワードの各フラグについて説明します。

(1) Z フラグ

算術演算の結果が 0000B になるとセット (1) され、0000B 以外になるとリセット (0) されます。ただし、CMP フラグの状態により次のようにセット (1) される条件が異なります。

(i) CMP フラグ=0 のとき

演算結果が 0000B であればセット (1) され 0000B 以外であればリセット (0) されます。

(ii) CMP フラグ=1 のとき

演算結果が 0000B であれば以前の状態を保持し、0000B 以外であればリセット (0) されます。算術演算以外では変化しません。

(2) CY フラグ

算術演算の結果、キャリーまたはボローが発生するとセット (1) され、発生しなければリセット (0) されます。

算術演算がキャリーまたはボローとともに演算を行う場合は CY フラグの内容を最下位ビットに演算します。回転処理 (RORC 命令) を行うときは、そのときの CY フラグの内容をジェネラル・レジスタの最上位ビット (b_7) とし、ジェネラル・レジスタの最下位ビットの内容が CY フラグの内容になります。

算術演算および回転処理以外では変化しません。

(3) CMP フラグ

CMP フラグがセット (1) されているときに実行された算術演算は、結果がジェネラル・レジスタおよびデータ・メモリに格納されません。

ビット判断命令を実行すると CMP フラグはリセット (0) されます。

比較判断、論理演算、回転処理には影響を与えません。

(4) BCD フラグ

BCD フラグがセット (1) されているときは、すべての算術演算が BCD 演算で行われます。

リセット (0) されているときは 2 進 4 ビット演算で行われます。

論理演算、ビット判断、比較判断、回転処理には影響を与えません。

これらのフラグは、プログラム・ステータス・ワード (PSWORD) を直接操作することにより値を変化させることも可能です。このとき、変化したフラグに対応するステータス・フリップフロップも同様に変化します。

5.2.4 2進4ビット演算

BCD フラグが0のとき、算術演算は、2進数による4ビット単位の演算を行います。

5.2.5 BCD 演算

BCD フラグが1のとき算術演算は、BCD 演算を行います。2進4ビット演算結果とBCD 演算結果の違いを表5-2に示します。BCD 加算実行時に加算結果が20以上になったとき、あるいはBCD 減算実行時に減算結果が-10~+9以外になったときにはデータ・メモリに1010B (0AH) 以上のデータが格納されます (表5-2の網掛け部分)。

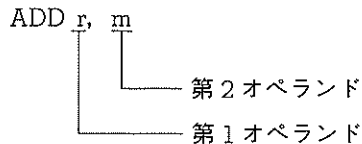
表5-2 2進4ビット演算結果とBCD 演算結果

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

5.2.6 ALU ブロック処理手順

プログラム上で算術演算命令、論理演算命令、ビット判断命令、比較判断命令および回路処理命令が実行されると、演算、判断または処理されるデータおよび処理するデータがそれぞれ一時記憶レジスタ A および B に格納されます。

処理されるデータとは、各命令の第 1 オペランドでアドレス指定されるジェネラル・レジスタの内容またはデータ・メモリの内容であり、4 ビットのデータです。処理するデータとは各命令の第 2 オペランドでアドレス指定されるデータ・メモリの内容または第 2 オペランドで直接指定されるイミディエト・データで 4 ビットのデータです。たとえば



命令において処理されるデータは r でアドレス指定されるジェネラル・レジスタの内容であり、処理するデータは m でアドレス指定されるデータ・メモリの内容となります。また

ADD m , # $n4$

命令は、処理されるデータは m でアドレス指定されるデータ・メモリの内容であり、処理するデータは # $n4$ で指定されるイミディエト・データになります。また回転処理命令である

RORC r

命令は、処理する方法が決まっているため処理されるデータのみ必要となり、 r でアドレス指定されるジェネラル・レジスタの内容になります。

次に、一時記憶レジスタ A および B に格納されたデータは、各命令に従い ALU で算術演算、論理演算、ビット判断、比較判断および回転処理を実行します。実行された命令が算術演算、論理演算および回転処理のときは、ALU で処理されたデータを、命令の第 1 オペランドで指定されるジェネラル・レジスタまたはデータ・メモリに格納して動作を終了します。また、実行された命令がビット判断および比較判断であるときは、ALU で処理された結果によりプログラム上の次の命令をスキップ(次の命令はノー・オペレーション (NOP 命令) 命令として実行されます)して動作を終了します。

ALU ブロック動作については次の点に注意が必要です。

- (1) 算術演算は、プログラム・ステータス・ワードの CMP フラグおよび BCD フラグの影響を受ける。
- (2) 論理演算は、プログラム・ステータス・ワードの CMP フラグおよび BCD フラグの影響は受けない。また Z フラグ、CY フラグには影響を与えない。
- (3) ビット判断はプログラム・ステータス・ワードの CMP フラグをリセットする。

5.3 算術演算（2進4ビット加減算およびBCD加減算）

表5-3に示すように、算術演算は、加算と減算に大別され、さらにキャリーとともに加算およびボローとともに減算とに分けられます。算術演算命令はこの4種類に分けられ、それぞれ、“ADD”、“ADDC”、“SUB”、“SUBC”命令を使用します。

“ADD”、“ADDC”、“SUB”、“SUBC”命令は、さらにジェネラル・レジスタとデータ・メモリの加減算およびデータ・メモリとイミディエト・データの加減算に分けられます。これは各命令のオペランドに記述する値により決定されます。すなわちオペランドが“r, m”であればジェネラル・レジスタとデータ・メモリの加減算になり“m, #n4”であればデータ・メモリとイミディエト・データの加減算になります。

算術演算命令はステータス・フリップフロップすなわち、システム・レジスタのプログラム・ステータス・ワード (PSWORD) の影響を受けます。プログラム・ステータス・ワード (PSWORD) のBCDフラグにより2進4ビット演算およびBCD演算を行い、CMPフラグにより、演算結果をどこにも格納しないことができます。

5.3.1 - 5.3.4 に各算術演算命令とプログラム・ステータス・ワード (PSWORD) について説明します。

表5-3 算術演算の種類

| | | | | |
|------|----|--------------------|--------------------|-------------|
| 算術演算 | 加算 | キャリーは無視 ADD | ジェネラル・レジスタとデータ・メモリ | ADD r, m |
| | | | データ・メモリとイミディエト・データ | ADD m, #n4 |
| | | キャリーとともに加算 ADDC | ジェネラル・レジスタとデータ・メモリ | ADDC r, m |
| | | | データ・メモリとイミディエト・データ | ADDC m, #n4 |
| | 減算 | ボローは無視 SUB | ジェネラル・レジスタとデータ・メモリ | SUB r, m |
| | | | データ・メモリとイミディエト・データ | SUB m, #n4 |
| | | ボローとともに減算 SUBC | ジェネラル・レジスタとデータ・メモリ | SUBC r, m |
| | | | データ・メモリとイミディエト・データ | SUBC m, #n4 |

5.3.1 CMPフラグ=0, BCDフラグ=0のときの加減算

2進4ビットの加減算を行い、結果をジェネラル・レジスタまたはデータ・メモリに格納します。

CYフラグは演算結果が1111Bを越えたとき（キャリーの発生）と、0000B未滿（ボローの発生）になるとセット(1)され、それ以外ではリセット(0)されます。

演算結果が0000Bになるとキャリーおよびボローの発生に関係なくZフラグをセット(1)し、0000Bでなければリセット(0)します。

5.3.2 CMPフラグ=1, BCDフラグ=0のときの加減算

2進4ビットの加減算を行います。

ただしCMPフラグがセット(1)されているため、演算結果がジェネラル・レジスタまたはデータ・メモリに格納されません。

演算結果によりキャリーまたはボローが発生するとCYフラグがセット(1)され、発生しなければリセット(0)されます。

Zフラグは、演算結果が0000Bであれば以前の状態を保持し、0000Bでなければリセット(0)されます。

5.3.3 CMP フラグ=0, BCD フラグ=1 のときの加減算

BCD 演算を行います。

演算結果は、ジェネラル・レジスタまたはデータ・メモリに格納されます。CY フラグは演算結果が 1001B (9D) を越えるか、0000B (0D) 未満になるとセット(1)され、0000B (0D) ~1001B (9D) であればリセット(0)されます。

Z フラグは、演算結果が 0000B (0D) になるとセット(1)され、0000B (0D) 以外になるとリセット(0)されます。

BCD 演算は、一度 2 進で演算された結果を 10 進補正回路で 10 進に変換する方法を用いています。この 2 進-10 進変換は **5.2.5 BCD 演算** の表 5-2 を参照してください。

したがって、BCD 演算を正しく実行するためには、次のことに注意してください。

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

0D~19D とは、CY フラグを考慮した値であり、16 進数で表すと

$$\begin{array}{c} 0, 0000B \sim 1, 0011B \text{ のことです} \\ \widetilde{CY} \quad \quad \quad \widetilde{CY} \end{array}$$

-10D~-1D とは同様に

$$\begin{array}{c} 1, 0110B \sim 1, 1111B \text{ のことです} \\ \widetilde{CY} \quad \quad \quad \widetilde{CY} \end{array}$$

(1), (2) 以外で 10 進演算を行うと CY フラグがセット(1)され、演算結果として 1010B (0AH) 以上のデータが出力されます。

5.3.4 CMP フラグ=1, BCD フラグ=1 のときの加減算

BCD 演算を行います。

演算結果はジェネラル・レジスタまたはデータ・メモリへ格納されません。

すなわち、CMP フラグ=1 のときと、BCD フラグ=1 のときの動作を同時に行います。

```
例  MOV  RPL,  #0001B ; BCD フラグをセット(1)
     MOV  PSW,  #1010B ; CMP フラグと Z フラグをセット(1), CY フラグをリセット(0)
     SUB  M1,   #0001B ; ①
     SUBC M2,   #0010B ; ②
     SUBC M3,   #0011B ; ③
```

このとき、①②③により M3, M2, M1 の 12 ビットの内容とイミューディエト・データの 321 とを、10 進で比較することが可能となります。

5.3.5 算術演算使用時の注意

プログラム・ステータス・ワード (PSWORD) に対して算術演算を行うときは、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る、という点に注意する必要があります。

すなわちプログラム・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、算術演算結果によりセット/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ません。

ただし、CMPフラグがセット(1)されているときは、算術演算結果が格納されないため、CYフラグ、Zフラグは通常どおりセット/リセットされます。

5.4 論理演算

表5-4に示すように、論理演算は論理和 (OR)、論理積 (AND) および排他的論理和 (XOR) が使用できます。論理演算命令はこの3種類に分けられ、それぞれ“OR”、“AND”および“XOR”命令を使用します。

“OR”、“AND”、“XOR”命令は、さらにジェネラル・レジスタとデータ・メモリの論理演算およびデータ・メモリとイミューディエト・データの論理演算に分けられます。これは算術演算と同様に命令のオペランドに記述された値“r, m”または“m, #n4”により決定されます。

論理演算は、プログラム・ステータス・ワード (PSWORD) のBCDフラグおよびCMPフラグの影響を受けません。またCYフラグおよびZフラグには何の影響も与えません。

表5-4 論理演算

| | | | |
|------|---------------|----------------------|------------|
| 論理演算 | 論理和 OR | ジェネラル・レジスタとデータ・メモリ | OR r, m |
| | | データ・メモリとイミューディエト・データ | OR m, #n4 |
| | 論理積 AND | ジェネラル・レジスタとデータ・メモリ | AND r, m |
| | | データ・メモリとイミューディエト・データ | AND m, #n4 |
| | 排他的論理和 XOR | ジェネラル・レジスタとデータ・メモリ | XOR r, m |
| | | データ・メモリとイミューディエト・データ | XOR m, #n4 |

表5-5 論理演算の真理値表

| 論理積 C=A AND B | | | 論理和 C=A OR B | | | 排他的論理和 C=A XOR B | | |
|------------------|---|---|-----------------|---|---|---------------------|---|---|
| A | B | C | A | B | C | A | B | C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

5.5 ビット判断

表 5-6 に示すように、ビット判断は True ビット (1) 判断および False ビット (0) 判断に分けられます。

True ビット (1) 判断および False ビット (0) 判断はそれぞれ “SKT” および “SKF” 命令を使用します。

“SKT”, “SKF” 命令はデータ・メモリに対してのみ行うことができます。

ビット判断は、プログラム・ステータス・ワード (PSWORD) の BCD フラグの影響を受けません。また CY フラグおよび Z フラグには何の影響も与えません。ただし、CMP フラグは “SKT” および “SKF” 命令が実行されるとリセット (0) されます。

5.5.1, 5.5.2 に True ビット (1) 判断および False ビット (0) 判断について説明します。

表 5-6 ビット判断命令

| | |
|-------|-------------------------------|
| ビット判断 | True ビット (1) 判断 SKT m, #n |
| | False ビット (0) 判断 SKF m, #n |

5.5.1 True ビット (1) 判断

True ビット (1) 判断命令 “SKT m, #n” は、データ・メモリの 4 ビットのうち、n で指定されたビットが “True (1)” であるかを判断します。n で指定されたビットがすべて “True (1)” であるとき、この命令の次の命令をスキップします。

```

例  MOV  M1,    #1011B
     SKT  M1,    #1011B ; ①
     BR   A
     BR   B
     SKT  M1,    #1101B ; ②
     BR   C
     BR   D
    
```

このとき、①では M1 のビット b_3, b_1, b_0 を判断し、すべて True (1) であるから B へ分岐します。

②では、M1 のビット b_3, b_2, b_0 を判断しますが、M1 の b_2 は False (0) であるため、C へ分岐します。

5.5.2 False ビット(0) 判断

False ビット(0) 判断命令“SKF m, #n”はデータ・メモリの4ビットのうちnで指定されたビットがFalse(0)であるかを判断します。nで指定されたビットがすべて“False(0)”であるとき、この命令の次の命令をスキップします。

```

例  MOV  M1,  #1001B ;
     SKF  M1,  #0110B ; ①
     BR   A           ;
     BR   B           ;
     SKF  M1,  #1110B ; ②
     BR   C           ;
     BR   D           ;
    
```

このとき①では、M1のビット b_2, b_1 を判断し、すべてFalse(0)であるためBへ分岐します。

②ではM1のビット b_3, b_2, b_1 を判断しますが、M1のビット b_3 はTrue(1)であるためCへ分岐します。

5.6 比較判断

表5-7に示すように、比較判断は“等しい”、“等しくない”、“以上”および“未満”の判断に分けられます。

“等しい”、“等しくない”、“以上”および“未満”の判断はそれぞれ“SKE”、“SKNE”、“SKGE”および“SKLT”命令を使用します。

“SKE”、“SKNE”、“SKGE”、“SKLT”命令は、データ・メモリとイミディエイト・データとの比較判断のみ行うことができます。ジェネラル・レジスタとデータ・メモリとの比較判断を行うときは、プログラム・ステータス・ワード(PSWORD)のCMPフラグおよびZフラグを用いて減算命令により行えます(5.3 算術演算(2進4ビット加減算およびBCD加減算)参照)。

比較判断は、プログラム・ステータス・ワード(PSWORD)のBCDフラグおよびCMPフラグの影響を受けません。またCYフラグおよびZフラグには何の影響も与えません。

5.6.1-5.6.4に“等しい”、“等しくない”、“以上”および“未満”の判断について説明します。

表5-7 比較判断命令

| | |
|------|----------------------|
| 比較判断 | 等しい SKE m, #n4 |
| | 等しくない SKNE m, #n4 |
| | 以上 SKGE m, #n4 |
| | 未満 SKLT m, #n4 |

5.6.1 “等しい”の判断

“等しい”の判断命令“SKE m, #n4”はデータ・メモリとイミューディアット・データの内容が“等しい”かを判断します。

データ・メモリとイミューディアット・データの内容が“等しい”とき、この命令の次の命令をスキップします。

```
例  MOV  M1,    #1010B
     SKE  M1,    #1010B ;①
     BR   A
     BR   B
     ;
     SKE  M1,    #1000B ;②
     BR   C
     BR   D
```

このとき、①では、M1の内容とイミューディアット・データの1010Bが等しいためBへ分岐します。

②ではM1の内容とイミューディアット・データの1000Bが等しくないためCへ分岐します。

5.6.2 “等しくない”の判断

“等しくない”の判断命令“SKNE m, #n4”は、データ・メモリとイミューディアット・データの内容が“等しくない”かを判断します。

データ・メモリとイミューディアット・データの内容が“等しくない”とき、この命令の次の命令をスキップします。

```
例  MOV  M1,    #1010B
     SKNE M1,    #1000B ;①
     BR   A
     BR   B
     ;
     SKNE M1,    #1010B ;②
     BR   C
     BR   D
```

このとき、①では、M1の内容とイミューディアット・データの1000Bが等しくないためBへ分岐します。

②では、M1の内容とイミューディアット・データの1010Bが等しいためCへ分岐します。

5.6.3 “以上”の判断

“以上”の判断命令“SKGE m, #n4”はデータ・メモリとイミューディエト・データの内容を比較し、データ・メモリの内容が、イミューディエト・データより“大きい”か、または“等しい”ときに、この命令の次の命令をスキップします。

```
例  MOV  M1,    #1000B
      SKGE  M1,    #0111B ;①
      BR   A
      BR   B
      ;
      SKGE  M1,    #1000B ;②
      BR   C
      BR   D
      ;
      SKGE  M1,    #1001B ;③
      BR   E
      BR   F
```

このとき、M1の内容は1000Bであるため①は“大きい”、②は“等しい”、③は“小さい”と判断され、それぞれB、D、Eに分岐します。

5.6.4 “未満”の判断

“未満”の判断命令“SKLT m, #n4”はデータ・メモリとイミューディエト・データの内容を比較し、データ・メモリの内容が、イミューディエト・データより“小さい”とき、この命令の次の命令をスキップします。

```
例  MOV  M1,    #1000B
      SKLT  M1,    #1001B ;①
      BR   A
      BR   B
      ;
      SKLT  M1,    #1000B ;②
      BR   C
      BR   D
      ;
      SKLT  M1,    #0111B ;③
      BR   E
      BR   F
```

このとき、M1の内容は1000Bであるため、①は“小さい”、②は“等しい”、③は“大きい”と判断され、それぞれB、C、Eに分岐します。

5.7 回転処理

回転処理には、右回転処理と左回転処理に分けられます。

右回転処理には“RORC”命令を使用します。

“RORC”命令は、ジェネラル・レジスタに対してのみ行うことができます。

“RORC”命令による回転処理は、プログラム・ステータス・ワード (PSWORD) の BCDフラグおよび CMP フラグの影響を受けません。また Zフラグには何の影響も与えません。

左回転処理は、加算命令である“ADDC”命令により行うことができます。

5.7.1, 5.7.2 で、回転処理について説明します。

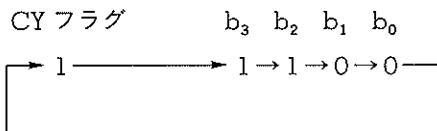
5.7.1 右回転処理

右回転処理命令“RORC r”はジェネラル・レジスタの内容を下位ビット方向に1ビット回転します。

このとき、ジェネラル・レジスタの内容の最上位ビット b_3 には CY フラグの内容が書き込まれ、最下位ビット b_0 の内容を CY フラグに書き込みます。

```
例 1. MOV PSW, #0100B ; CY フラグをセット (1)
      MOV R1, #1001B
      RORC R1 ; ①
```

このとき、次のように処理されます。



すなわち、CY フラグ→ b_3 , b_3 → b_2 , b_2 → b_1 , b_1 → b_0 , b_0 →CY フラグのように右に回転を行います。

```
2. MOV PSW, #0000B ; CY フラグをリセット (0)
   MOV R1, #1000B
   MOV R2, #0100B
   MOV R3, #0010B
   RORC R1
   RORC R2
   RORC R3
```

このとき、上記プログラムは R1, R2, R3 の12ビット・データを右に回転します。

5.7.2 左回転処理

左回転処理は加算命令である“ADDC r, m”命令を用いることにより行えます。

```
例  MOV  PSW,  #0000B : CY フラグをリセット(0)
     MOV  R1,   #1000B
     MOV  R2,   #0100B
     MOV  R3,   #0010B
     ADDC R3, R3
     ADDC R2, R2
     ADDC R1, R1
```

このとき、上記プログラムは R1, R2, R3 の12ビット・データを左に回転します。

6. ポート

6.1 ポート OA (POA₀ - POA₃)

4ビットの入出力ポートです。出力形式は、CMOS（プッシュプル）出力です。

ニブル単位で入出力を設定することができます。この設定は、リセット時は入力モード、そしてデータ・メモリの70H番地に配置されているポート・レジスタにデータを書き込むことによって出力モードになり、リセットをかけたかぎり出力モードは維持されます。

ポートへの出力は、ポート・レジスタを介して行います。一度、ポート・レジスタにデータを書き込むと、ポートOAの全端子は出力モードとなり、書き込んだデータを出力し続けます。データは、書き換えにかぎり保持されます。

ポート・レジスタを読み込む命令を実行すると、ポート・レジスタの内容ではなく、入出力どちらのモードにもかわらず、常に端子の状態^注が読めます。この際、ポート・レジスタの内容は変化しません。

6.2 ポート OB (POB₀/RLS_{HALT}, POB₁/RLS_{STOP}, POB₂, POB₃)

4ビットの入出力ポートです。出力形式は、N-ch オープン・ドレイン出力です。N-ch オープン・ドレイン出力は9Vの耐圧を持っていますので、電源電圧の異なる回路とのインタフェースに便利です。

ニブル単位で入出力を設定することができます。この設定は、リセット時は入力モード、そしてデータ・メモリの71H番地に配置されているポート・レジスタにデータを書き込むことによって出力モードになり、リセットをかけたかぎり出力モードは維持されます。

ポートへの出力は、ポート・レジスタを介して行います。一度、ポート・レジスタにデータを書き込むと、ポートOBの全端子は出力モードとなり、書き込んだデータを出力し続けます。データは、書き換えにかぎり保持されます。

ただし、ポート・レジスタに“1”を書くことにより、N-ch オープン・ドレイン出力端子はハイ・インピーダンス状態になります。このため、“1”を出力している端子は入力としても使用できます。

ポート・レジスタを読み込む命令を実行すると、ポート・レジスタの内容ではなく、入出力どちらのモードにもかわらず、常に端子の状態^注が読めます。この際、ポート・レジスタの内容は変化しません。

μPD17108LがHALTモードまたはSTOPモードのとき、POB₀、POB₁はそれぞれ疑似割り込み端子としてHALTモード、STOPモードを解除することができます（7. スタンバイ機能を参照してください）。

6.3 ポート OC (POC₀ - POC₃)

4ビットの入出力ポートです。出力形式は、CMOS（プッシュプル）出力です。

ニブル単位で入出力を設定することができます。この設定は、リセット時は入力モード、そしてデータ・メモリの72H番地に配置されているポート・レジスタにデータを書き込むことによって出力モードになり、リセットをかけたかぎり出力モードは維持されます。

ポートへの出力は、ポート・レジスタを介して行います。一度、ポート・レジスタにデータを書き込むと、ポートOCの全端子は出力モードとなり、書き込んだデータを出力し続けます。データは、書き換えにかぎり保持されます。

ポート・レジスタを読み込む命令を実行すると、ポート・レジスタの内容ではなく、入出力どちらのモードにもかわらず、常に端子の状態^注が読めます。この際、ポート・レジスタの内容は変化しません。

注 出力モード時は、出力データの内容および端子の外付け回路を考慮する必要があります。

6.4 ポート OD (POD₀ - POD₃)

4ビットの入出力ポートです。出力形式は、CMOS（プッシュプル）出力です。

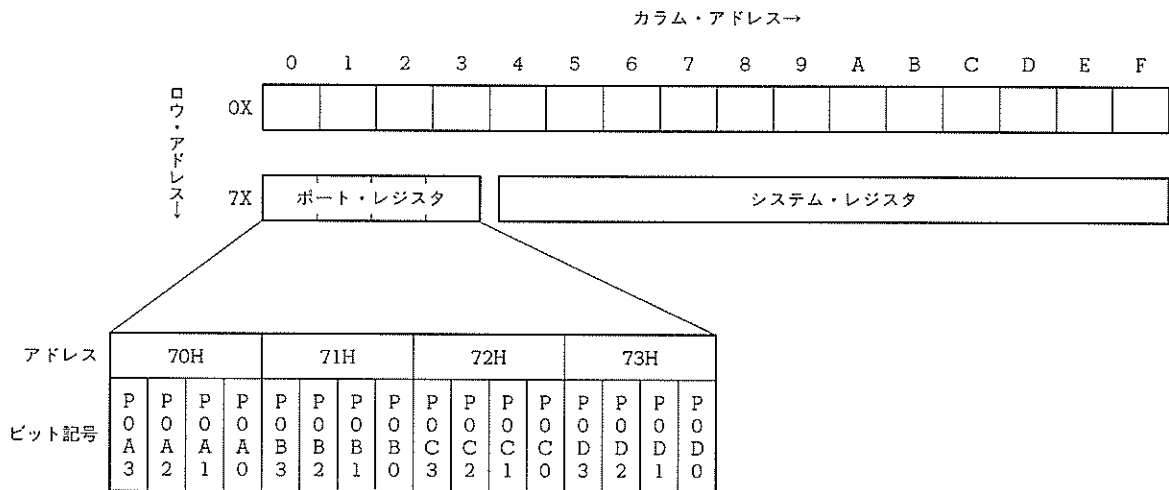
ニブル単位で入出力を設定することができます。この設定は、リセット時は入力モード、そしてデータ・メモリの73H番地に配置されているポート・レジスタにデータを書き込むことによって出力モードになり、リセットをかけないかぎり出力モードは維持されます。

ポートへの出力は、ポート・レジスタを介して行います。一度、ポート・レジスタにデータを書き込むと、ポート ODの全端子は出力モードとなり、書き込んだデータを出力し続けます。データは、書き換えないかぎり保持されます。

ポート・レジスタを読み込む命令を実行すると、ポート・レジスタの内容ではなく、入出力どちらのモードにもかかわらず、常に端子の状態^注が読めます。この際、ポート・レジスタの内容は変化しません。

注 出力モード時は、出力データの内容および端子の外付け回路を考慮する必要があります。

図 6-1 ポート・レジスタのマッピング



★ 6.5 ポート・レジスタの操作時の注意

μPD17108Lの入出力ポートは、出力モードであっても読み込み時には端子の状態を読み込みます。

したがってポート・レジスタに組み込みマクロ命令 (SETn/CLRn など) や AND/OR/XOR 命令などでビット操作すると、意図していない端子の状態も変化してしまうことがあります。

特にポート OB (N-ch オープン・ドレイン出力) を入出力混在させて使用する場合には注意が必要です。

ポート OB に CLR1 POB2 命令 (AND 71H, #1011B 命令と同じ) を実行するとポート・レジスタおよびマイコン内部の状態が変化する例を図 6-2 に示します。

たとえばポート OB の各端子を、POB₃ と POB₂ 端子は出力、POB₁ と POB₀ 端子は入力として使用し、POB₃ と POB₂ 端子からハイ・レベルが出力され、POB₁ と POB₀ 端子にロウ・レベルが入力されている場合を考えると、ポート OB の各状態は図 6-2 ① のようになります。

〔ポート OB の端子のうち入力として使用したい端子は、ハイ・レベルを出力しておく必要があります。また、μPD17103, 17103L, 17107, 17107L では、POB₃ 端子は存在しませんがプログラム上は仮想的に存在しているものとして扱います。〕

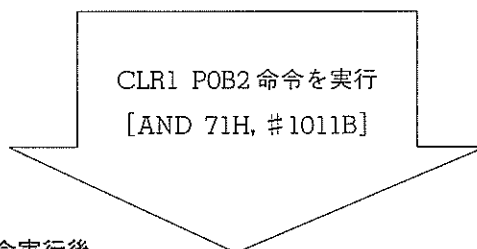
POB₂ 端子をロウ・レベルにするため、CLR1 POB2 命令を実行すると、ポート OB の各状態は図 6-2 ② のように変化します。本来、ハイ・レベルを出力しておかなければならない POB₁ と POB₀ 端子から、ロウ・レベルが出力されるようにポート・レジスタの値が変化しています。これはポート・レジスタではなく端子の状態に対して CLR1 POB2 命令が実行されたために生じた結果です。

この現象を防ぐには、変化させる端子だけではなく、すべての端子の状態を MOV 命令などで設定するようにします。この例で POB₂ 端子だけをロウ・レベルにするには、MOV 71H, 1011B 命令を使用すれば問題ありません。

図 6-2 CLR1 POB2 命令によるポート・レジスタの変化

① 命令実行前

| 状 態 | POB ₃ | POB ₂ | POB ₁ | POB ₀ |
|----------|------------------|------------------|------------------|------------------|
| ポート・レジスタ | 1 | 1 | 1 | 1 |
| マイコン内部 | H 出力 | H 出力 | H 出力 | H 出力 |
| 端 子 | H | H | L (入力) | L (入力) |



② 命令実行後

| 状 態 | POB ₃ | POB ₂ | POB ₁ | POB ₀ |
|----------|------------------|------------------|------------------|------------------|
| ポート・レジスタ | 1 | 0 | 0 | 0 |
| マイコン内部 | H 出力 | L 出力 | L 出力 | L 出力 |
| 端 子 | H | L | L | L |

H: ハイ・レベル L: ロウ・レベル

7. スタンバイ機能

μPD17108Lは、スタンバイ・モードとして HALT モードと STOP モードの2種類を用意しています。

7.1 HALT モード

HALT モードとは、システム・クロックの発振を継続させたまま PC (プログラム・カウンタ) の動作が停止している状態を設定するモードです。HALT モードは HALT 命令で設定でき、リセット信号 ($\overline{\text{RESET}}$) または POB₀ 端子へのハイ・レベル入力で解除できます。POB₀ 端子へのハイ・レベル入力による解除の場合、システム・クロックの発振安定待ちは行いません。また、この場合、解除後の実行は HALT 命令の次の命令からです。

なお、リセット信号 ($\overline{\text{RESET}}$) により強制的に解除した場合、通常システム・リセット (OH 番地スタート) を行います。

7.2 STOP モード

STOP モードとは、システム・クロックの発振を停止させ、低電源電圧でデータ保持が可能になっている状態を設定するモードです。STOP モードは STOP 命令で設定でき、リセット信号 ($\overline{\text{RESET}}$) または POB₁ 端子へのハイ・レベル入力で解除できます。また、POB₁ 端子へのハイ・レベル入力による解除の場合、解除後の実行は STOP 命令の次の命令からです。

なお、リセット信号 ($\overline{\text{RESET}}$) により強制的に解除した場合、通常システム・リセット (OH 番地スタート) を行います。

7.3 スタンバイ・モードの設定と解除

(1) HALT モードの設定と解除

HALT 命令は、オペランドの最下位ビットを表 7-1 のように設定することによってモードの解除条件を選択することができます。オペランドの上位3ビットは“0”に固定してください。

表 7-1 HALT 命令の設定, 解除条件

★

HALT 000 XB ←オペランドの4ビット・データ

| X | 設定, 解除条件 |
|---|--|
| 0 | HALT 命令を実行すると無条件で HALT モードになります。 リセット信号 ($\overline{\text{RESET}}$) でのみ解除できます。解除後は OH 番地より命令を実行します。 |
| 1 | POB ₀ 端子の状態がロウ・レベルのとき、HALT 命令を実行すると HALT モードになります。 リセット信号 ($\overline{\text{RESET}}$) で解除できます。解除後は OH 番地より命令を実行します。 また、POB ₀ 端子へのハイ・レベル入力によっても解除できます。この場合、解除後実行される命令は HALT 命令の次の命令からです。 POB ₀ 端子の状態がハイ・レベルのとき、HALT 命令を実行しても HALT 命令は無視 (NOP 命令扱い) され、HALT モードになりません。 |

(2) STOP モードの設定と解除

STOP 命令は、オペランドの最下位ビットを表 7-2 のように設定することによってモードの解除条件を選択することができます。オペランドの上位3ビットは“0”に固定してください。

★

表 7-2 STOP 命令の設定・解除条件

STOP 000 XB ←オペランドの4ビット・データ

| X | 設定・解除条件 |
|---|---|
| 0 | STOP 命令を実行すると無条件でSTOP モードになります。 すべての周辺回路はシステム・リセットをかけたときと同一の初期状態に戻ったあと、動作を停止します。 リセット信号 (RESET) でのみ解除できます。解除後は 0H 番地より命令を実行します。 |
| 1 | POB ₁ 端子の状態がロウ・レベルのとき、STOP 命令を実行すると STOP モードになります。 POB ₁ 端子の状態がハイ・レベルのとき、STOP 命令を実行しても STOP 命令は無視 (NOP 命令扱い) され、STOP モードにはなりません。 リセット信号 (RESET) で解除できます。解除後は 0H 番地より命令を実行します。 また、POB ₁ 端子へのハイ・レベル入力によっても解除できます。この場合、解除後実行される命令は STOP 命令の次の命令からです。 |

★

7.4 スタンバイ・モード中のハードウェアの状態

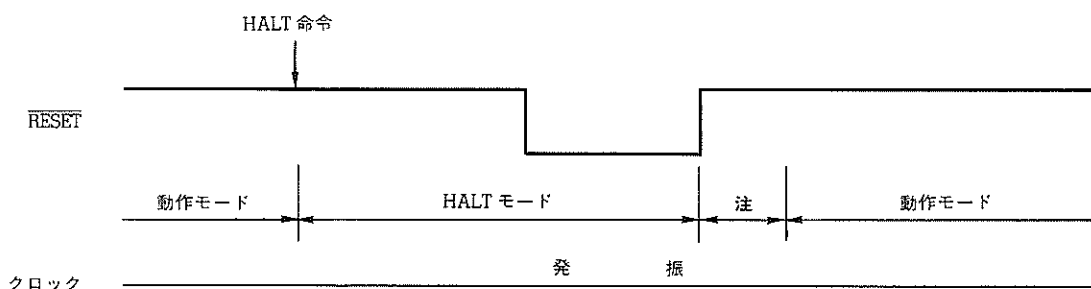
スタンバイ・モード中のハードウェアの状態は次のとおりです。

表 7-3 スタンバイ・モード中のハードウェアの状態

| ハードウェア | HALT 命令または STOP 0001B 命令 | STOP 0000B 命令 |
|--------------------------|------------------------------------|---------------------------|
| クロック発振回路 | HALT 命令の場合：発振継続 STOP 命令の場合：発振停止 | 発振停止 |
| プログラム・カウンタ | HALT, STOP 命令の次の番地 | 000H |
| データ・メモリ (00H-0FH) | 前のデータを保持 | 前のデータを保持 |
| プログラム・ステータス・ワード (PSWORD) | 前のデータを保持 | すべて "0" |
| ポート・レジスタ (71H-73H) | 前のデータを保持 (端子の入出力モードも保持) | 前のデータを保持 (端子はすべて入力モード) |

7.5 スタンバイ・モードの解除タイミング

図 7-1 RESET 入力による HALT モードの解除



HALT モードの RESET 入力による解除を行う場合、RESET 入力をハイ・レベルに戻したあと、動作モードになります。

注 この期間は待ち時間です。モードは HALT モードです。

OSC₁端子からのクロック・パルスを 8 カウントすると動作を開始します。

図 7-2 POB₀ 端子へのハイ・レベル入力による HALT モードの解除

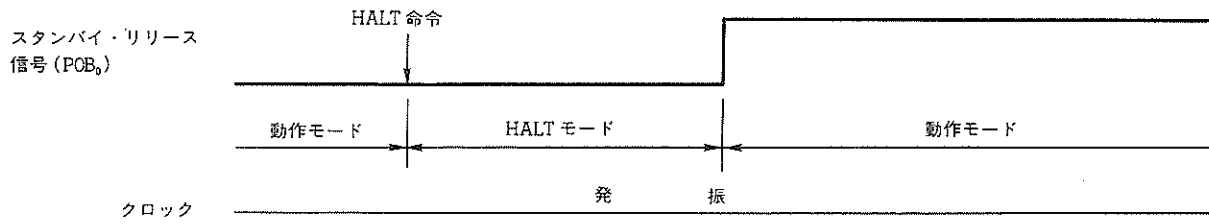
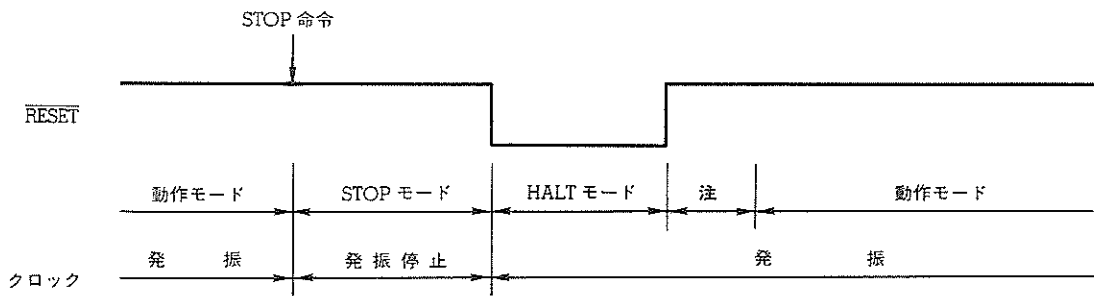


図 7-3 $\overline{\text{RESET}}$ 入力による STOP モードの解除

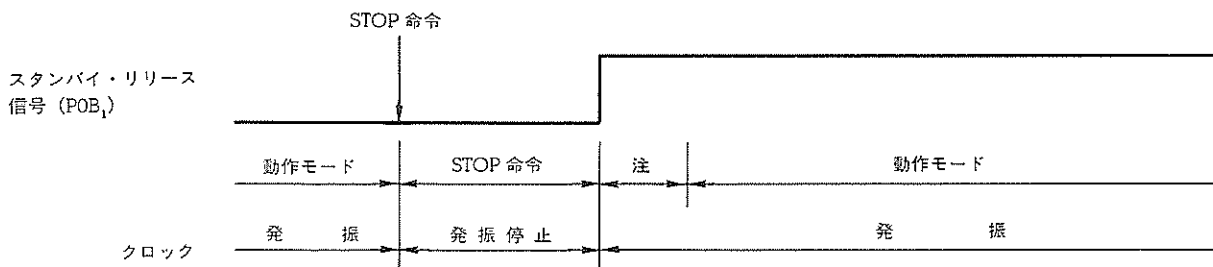


STOP モード中に $\overline{\text{RESET}}$ 入力がハイ・レベルからロウ・レベルに下がると同時に発振を開始します。

注 この期間は発振安定待ち時間です。モードは HALT モードです。

OSC₁ 端子からのクロック・パルスを 8 カウントすると動作を開始します。

図 7-4 POB₁ 端子へのハイ・レベル入力による STOP モードの解除



注 この期間は発振安定待ち時間です。モードは HALT モードです。

OSC₁ 端子からのクロック・パルスを 8 カウントすると動作を開始します。

★ 8. リセット

8.1 システム・リセット

RESET 端子にロウ・レベル信号を入力することによりシステム・リセットがかかり、ハードウェアが初期化されます。

なお、RESET 端子がロウ・レベルであっても電源電圧が供給されているかぎりシステム・クロックは発振します。

RESET 端子をハイ・レベルに戻すとリセットが解除され、8クロックの発振安定待ちのあと、動作モードになります。

表 8-1 リセット後のハードウェアの状態

| ハードウェア | | ・電源立ち上げ直後のリセット ・動作中のリセット | スタンバイ・モード中のリセット注 |
|--------------------------|--------|-----------------------------|------------------|
| プログラム・カウンタ | | 000H | 000H |
| データ・メモリ (00H-0FH) | | 不定 | リセット前のデータを保持 |
| プログラム・ステータス・ワード (PSWORD) | | すべて "0" | すべて "0" |
| ポート | 入出力モード | 入力 | 入力 |
| | 出カラッチ | 不定 | リセット前のデータを保持 |

注 STOP 0000B 命令を実行した場合には、命令を実行した時点でハードウェアが初期化されます。

9. アセンブラ予約語

9.1 マスク・オプション疑似命令

μPD17108Lのプログラムを作成する場合、アセンブラのソース・プログラム中にマスク・オプション疑似命令を使用してマスク・オプションを指定する必要があります。

マスク・オプションを設定するためには、アSEMBル時に AS17108L (μPD17108L用デバイス・ファイル) 中の D17108L. OPT ファイルをカレント・ディレクトリに入れておかなければなりませんので注意してください。

以下の端子すべてにマスク・オプションを指定してください。

- POB₀
- POB₁
- POB₂
- POB₃
- RESET

9.1.1 OPTION, ENDOP 疑似命令

OPTION 疑似命令から ENDOP 疑似命令までをマスク・オプション定義ブロックとします。マスク・オプション定義ブロックの記述形式を以下に示します。

このブロック内では表9-1に示す2つの疑似命令だけが記述可能です。

記述形式：

| シンボル欄 | ニモニック欄 | オペラント欄 | コメント欄 |
|---------|--------|--------|---------|
| [レーベル:] | OPTION | | [;コメント] |
| | ⋮ | | |
| | ENDOP | | |

9.1.2 マスク・オプション定義疑似命令

各端子のマスク・オプションを定義する疑似命令を表9-1に示します。

表9-1 マスク・オプション定義疑似命令一覧表

| 端子名 | マスク・オプション疑似命令 | オペラントの数 | オペラント名 |
|-------------------------------------|---------------|---------|---|
| POB ₃ - POB ₀ | OPTPOB | 4 | POBPLUP (プルアップ抵抗あり) OPEN (プルアップ抵抗なし) |
| <u>RESET</u> | OPTRES | 1 | RESPLUP (プルアップ抵抗あり) OPEN (プルアップ抵抗なし) |

OPTPOBの記述形式を以下に示します。オペラント欄には第一オペラントから順に POB₃, POB₂, POB₁, POB₀のマスク・オプションを定義します。

記述形式：

| シンボル欄 | ニモニック欄 | オペラント欄 | コメント欄 |
|---------|--------|--|---------|
| [レーベル:] | OPTPOB | (POB ₃), (POB ₂), (POB ₁), (POB ₀) | [;コメント] |

OPTRES の記述形式を以下に示します。

記述形式：

| シンボル欄 | 二モニツク欄 | オペランド欄 | コメント欄 |
|---------|--------|---------|---------|
| [レーベル:] | OPTRES | (RESET) | [;コメント] |

例 μPD17108L のアセンブル用ソース・ファイルで次のマスク・オプションを設定します。

POB₃ …プルアップ, POB₂ …プルアップ, POB₁ …オープン, POB₀ …オープン,
 RESET …プルアップ

```

                                :
; 17108L
マスク・オプション設定： OPTION
                                OPTPOB   POBPLUP, POBPLUP, OPEN, OPEN
                                OPTRES    RESPLUP
                                ENDOP
                                :
    
```

9.2 予約シンボル

μPD17108Lのデバイス・ファイル(AS17108L)内で定義されている予約シンボルの一覧を表9-2に示します。

表9-2 予約シンボル一覧表

| 名 前 | 属 性 | 値 | R/W | 説 明 |
|------|-----|---------|-----|-----------------|
| POA0 | FLG | 0.70H.0 | R/W | ポート0Aのビット0 |
| POA1 | FLG | 0.70H.1 | R/W | ポート0Aのビット1 |
| POA2 | FLG | 0.70H.2 | R/W | ポート0Aのビット2 |
| POA3 | FLG | 0.70H.3 | R/W | ポート0Aのビット3 |
| POB0 | FLG | 0.71H.0 | R/W | ポート0Bのビット0 |
| POB1 | FLG | 0.71H.1 | R/W | ポート0Bのビット1 |
| POB2 | FLG | 0.71H.2 | R/W | ポート0Bのビット2 |
| POB3 | FLG | 0.71H.3 | R/W | ポート0Bのビット3 |
| POC0 | FLG | 0.72H.0 | R/W | ポート0Cのビット0 |
| POC1 | FLG | 0.72H.1 | R/W | ポート0Cのビット1 |
| POC2 | FLG | 0.72H.2 | R/W | ポート0Cのビット2 |
| POC3 | FLG | 0.72H.3 | R/W | ポート0Cのビット3 |
| POD0 | FLG | 0.73H.0 | R/W | ポート0Dのビット0 |
| POD1 | FLG | 0.73H.1 | R/W | ポート0Dのビット1 |
| POD2 | FLG | 0.73H.2 | R/W | ポート0Dのビット2 |
| POD3 | FLG | 0.73H.3 | R/W | ポート0Dのビット3 |
| BCD | FLG | 0.7EH.0 | R/W | BCD演算フラグ |
| PSW | MEM | 0.7FH | R/W | プログラム・ステータス・ワード |
| Z | FLG | 0.7FH.1 | R/W | ゼロ・フラグ |
| CY | FLG | 0.7FH.2 | R/W | キャリー・フラグ |
| CMP | FLG | 0.7FH.3 | R/W | コンペア・フラグ |

10. 命令セット

10.1 命令セット一覧表

| 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 | RET | | | |
| | | RETSK | | | |
| | | 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 | | | | |
| 1 0 1 1 | B | SKNE | m, #n4 | SKLT | m, #n4 |
| 1 1 0 0 | C | BR | addr | CALL | addr |
| 1 1 0 1 | D | | | MOV | m, #n4 |
| 1 1 1 0 | E | | | SKT | m, #n |
| 1 1 1 1 | F | | | SKF | m, #n |

10.2 命令一覧表

凡 例

- 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, 上位2ビットは0固定)
- CMP : コンペア・フラグ
- CY : キャリー・フラグ
- h : ホールト解除条件
- 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)
- n4 : イミディエイト・データ (4ビット)
- PC : プログラム・カウンタ
- r : ジェネラル・レジスタ・カラム・アドレス
- SP : スタック・ポインタ
- s : ストップ解除条件
- (x) : xでアドレスされる内容

| 命令群 | 二モニック | 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 |
| 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 | $(r) \leftarrow (r) \nabla (m)$ | 00101 | m_R | m_C | r | |
| | m, #n4 | $(m) \leftarrow (m) \nabla n4$ | 10101 | m_R | m_C | n4 | |
| 判断 | SKT | m, #n | CMP ← 0, if (m) ∧ n = n, then skip | 11110 | m_R | m_C | n |
| | SKF | m, #n | CMP ← 0, if (m) ∧ 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 | | 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 | m, #n4 | $(m) \leftarrow n4$ | 11101 | m_R | m_C | n4 |
| 分岐 | BR | addr | $PC_{10-0} \leftarrow addr$ | 01100 | addr | | |
| サブルーチン | CALL | addr. | $SP \leftarrow SP - 1, ASR \leftarrow PC, PC_{10-0} \leftarrow addr$ | 11100 | addr | | |
| | RET | | $PC \leftarrow ASR, SP \leftarrow SP + 1$ | 00111 | 000 | 1110 | 0000 |
| | RETSK | | $PC \leftarrow ASR, SP \leftarrow SP + 1$ and skip | 00111 | 001 | 1110 | 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 |

10.3 アセンブラ (AS17K) 組み込みマクロ命令

★

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 ≤ n ≤ 4 |
| | SKFn | flag 1, ...flag n | if (flag 1) ~ (flag n) = all "0", then skip | 1 ≤ n ≤ 4 |
| | SETn | flag 1, ...flag n | (flag 1) ~ (flag n) ← 1 | 1 ≤ n ≤ 4 |
| | CLRn | flag 1, ...flag n | (flag 1) ~ (flag n) ← 0 | 1 ≤ n ≤ 4 |
| | NOTn | flag 1, ...flag n | if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0 | 1 ≤ n ≤ 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 ≤ n ≤ 4 |

11. 電気的特性

絶対最大定格 ($T_a = 25^\circ\text{C}$)

| 項目 | 略号 | 条件 | 定格 | 単位 | |
|------------|-----------|--|--------------------------|--------------------------|----|
| 電源電圧 | V_{DD} | | $-0.3 \sim +7.0$ | V | |
| 入力電圧 | V_i | POA, POC, POD, $\overline{\text{RESET}}$ | $-0.3 \sim V_{DD} + 0.3$ | V | |
| | | POB | 注1 | $-0.3 \sim V_{DD} + 0.3$ | V |
| | | | 注2 | $-0.3 \sim +11$ | V |
| 出力電圧 | V_o | POA, POC, POD | $-0.3 \sim V_{DD} + 0.3$ | V | |
| | | POB | 注1 | $-0.3 \sim V_{DD} + 0.3$ | V |
| | | | 注2 | $-0.3 \sim +11$ | V |
| ハイ・レベル出力電流 | I_{OH} | POA, POC, POD 1端子当たり | -5 | mA | |
| | | 全端子合計 | -15 | mA | |
| ロウ・レベル出力電流 | I_{OL} | POA, POB, POC, POD 1端子当たり | 30 | mA | |
| | | 全端子合計 | 100 | mA | |
| 動作温度 | T_{opt} | | $-40 \sim +85$ | $^\circ\text{C}$ | |
| 保存温度 | T_{stg} | | $-65 \sim +150$ | $^\circ\text{C}$ | |
| 許容損失 | P_d | $T_a = 85^\circ\text{C}$ | 22ピンプラスチックシュリンク DIP | 400 | mW |
| | | | 24ピンプラスチックSOP | 250 | |

注1. マスク・オプションによるプルアップ抵抗を内蔵した場合

2. マスク・オプションによるプルアップ抵抗を内蔵しない場合

注意 各項目のうち1項目でも、また一瞬でも絶対最大定格を越えると、製品の品質を損なう恐れがあります。つまり絶対最大定格とは、製品に物理的な損傷を与えかねない定格値です。必ずこの定格値を越えない状態で、製品をご使用ください。

容量 ($T_a = 25^\circ\text{C}$, $V_{DD} = 0\text{V}$)

| 項目 | 略号 | 条件 | MIN. | TYP. | MAX. | 単位 |
|-------|----------|-------------------|------|------|------|----|
| 入力容量 | C_{IN} | $f = 1\text{MHz}$ | | | 15 | pF |
| 入出力容量 | C_{IO} | 被測定端子以外は0V | | | 15 | pF |

DC 特性 ($T_a = -40 \sim +85^\circ\text{C}$, $V_{DD} = 1.5 \sim 3.6\text{V}$)

| 項 目 | 略 号 | 条 件 | | MIN. | TYP. | MAX. | 単 位 |
|------------------------|------------|--|--|----------------|------|---------------|------------------|
| ハイ・レベル入力電圧 | V_{IH1} | POA, POC, POD | | $0.7 V_{DD}$ | | V_{DD} | V |
| | V_{IH2} | $\overline{\text{RESET}}$ | | $0.8 V_{DD}$ | | V_{DD} | V |
| | V_{IH3} | POB | 注 1 | $0.8 V_{DD}$ | | V_{DD} | V |
| | V_{IH4} | | 注 2 | $0.8 V_{DD}$ | | 9 | V |
| ロウ・レベル入力電圧 | V_{IL1} | POA, POC, POD | | 0 | | $0.25 V_{DD}$ | V |
| | V_{IL2} | $\overline{\text{RESET}}$ | | 0 | | $0.15 V_{DD}$ | V |
| | V_{IL3} | POB | | 0 | | $0.15 V_{DD}$ | V |
| ハイ・レベル出力電圧 | V_{OH} | POA, POC, POD $I_{OH} = -200 \mu\text{A}$ | | $V_{DD} - 1.0$ | | | V |
| ロウ・レベル出力電圧 | V_{OL} | POA, POB, POC, POD $I_{OL} = 600 \mu\text{A}$ | | | | 0.5 | V |
| ハイ・レベル入力 リーク電流 | I_{L1H1} | POA, POC, POD, $V_{IN} = V_{DD}$ | | | | 5 | μA |
| | I_{L1H2} | POB | $V_{IN} = V_{DD}$ 注 1 | | | 5 | μA |
| | I_{L1H3} | | $V_{IN} = 9\text{V}$ 注 2 | | | 10 | μA |
| ロウ・レベル入力 リーク電流 | I_{L1L1} | POA, POC, POD, $V_{IN} = 0\text{V}$ | | | | -5 | μA |
| | I_{L1L2} | POB, $V_{IN} = 0\text{V}$ | | | | -5 | μA |
| ハイ・レベル出力 リーク電流 | I_{L0H1} | POA, POC, POD, $V_{OUT} = V_{DD}$ | | | | 5 | μA |
| | I_{L0H2} | POB | $V_{OUT} = V_{DD}$ 注 1 | | | 5 | μA |
| | I_{L0H3} | | $V_{OUT} = 9\text{V}$ 注 2 | | | 10 | μA |
| ロウ・レベル出力 リーク電流 | I_{L0L} | POA, POB, POC, POD, $V_{OUT} = 0\text{V}$ | | | | -5 | μA |
| RESET 端子の 内蔵プルアップ抵抗 | R_{RES} | | | 20 | 47 | 95 | $\text{k}\Omega$ |
| POB 端子の 内蔵プルアップ抵抗 | R_{POB} | | | 5 | 15 | 30 | $\text{k}\Omega$ |
| 電 源 電 流 注 3 | I_{DD1} | 動作モード | $V_{DD} = 3\text{V} \pm 10\%$ $f_{CC} = 200\text{kHz} \pm 20\%$ | | 65 | 150 | μA |
| | I_{DD2} | HALTモード | $V_{DD} = 3\text{V} \pm 10\%$ $f_{CC} = 200\text{kHz} \pm 20\%$ | | 55 | 130 | μA |
| | I_{DD3} | STOPモード | $V_{DD} = 3\text{V} \pm 10\%$ | | 0.1 | 10 | μA |

- 注 1. マスク・オプションによるプルアップ抵抗を内蔵した場合
 2. マスク・オプションによるプルアップ抵抗を内蔵しない場合
 3. 内蔵プルアップ抵抗に流れる電流は除く

データ・メモリ STOP モード低電源電圧データ保持特性 ($T_a = -40 \sim +85^\circ\text{C}$)

| 項目 | 略号 | 条件 | MIN. | TYP. | MAX. | 単位 |
|-----------|-------------------|---------------------------------|------|------|------|----|
| データ保持電源電圧 | V_{DDDR} | | 1.5 | | 3.6 | V |
| データ保持電源電流 | I_{DDDR} | $V_{\text{DDDR}} = 1.5\text{V}$ | | 0.1 | 5.0 | μA |

AC 特性 ($T_a = -40 \sim +85^\circ\text{C}$, $V_{\text{DD}} = 1.5 \sim 3.6\text{V}$)

| 項目 | 略号 | 条件 | MIN. | TYP. | MAX. | 単位 |
|--|--------------------------------------|----|------|------|------|----|
| CPU クロック・サイクル・タイム (命令実行時間) | t_{CY} | | 32 | | 160 | μs |
| POB_0 , POB_1 ハイ, ロウ・レベル幅 | t_{PBH} t_{PBL} | | 100 | | | μs |
| $\overline{\text{RESET}}$ ハイ, ロウ・レベル幅 | t_{RSH} t_{RSL} | | 100 | | | μs |

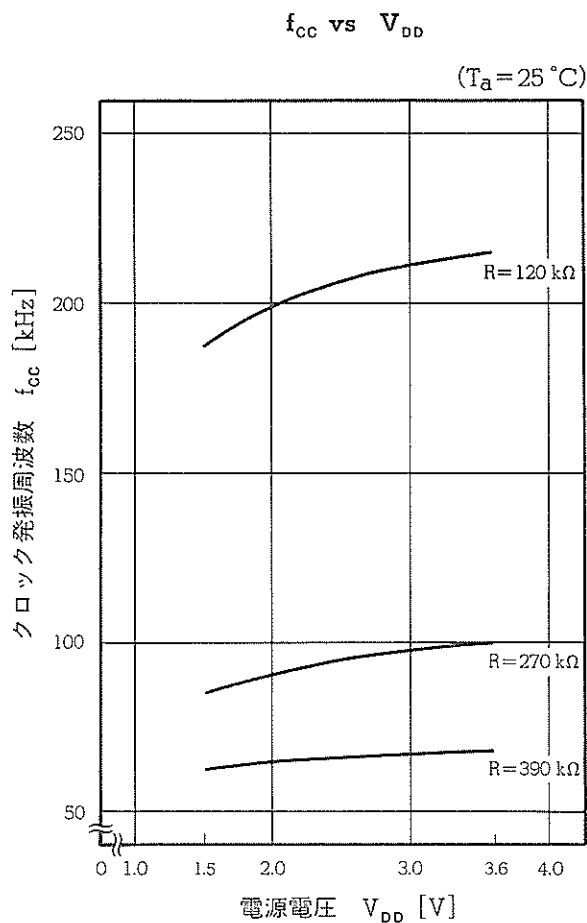
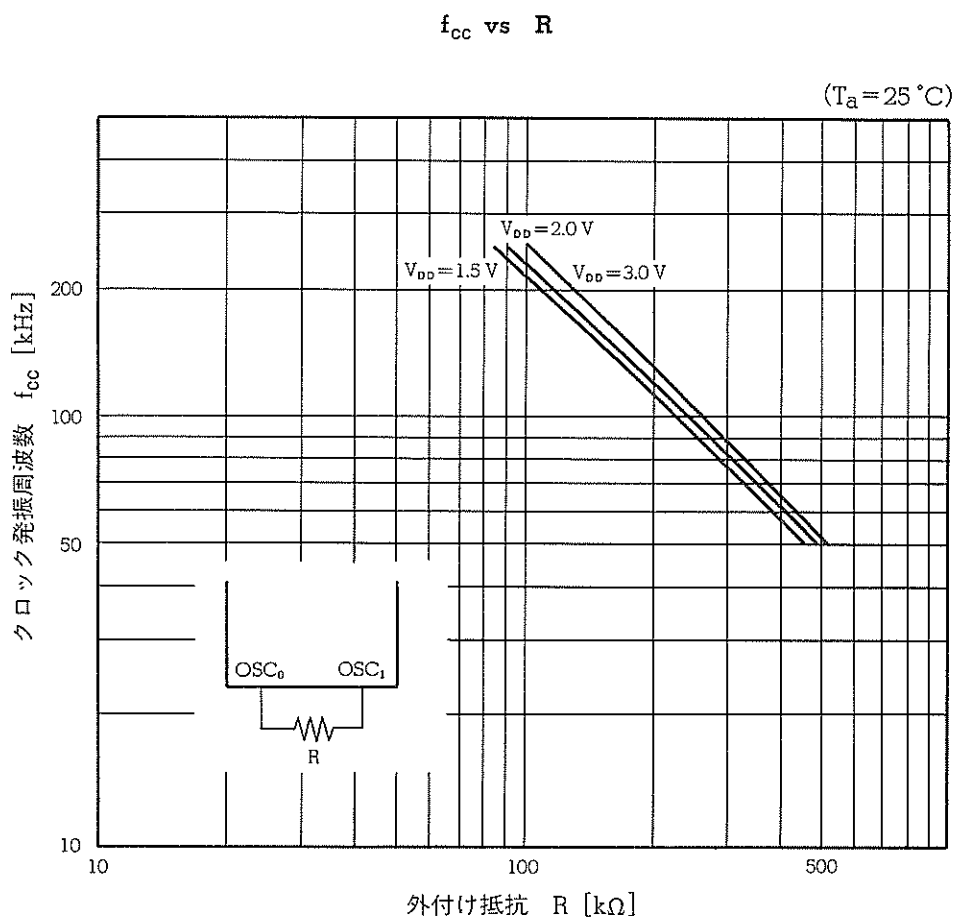
備考 $t_{\text{CY}} = 8/f_{\text{CC}}$ (f_{CC} はシステム・クロック発振周波数)

★ システム・クロック発振回路特性 ($T_a = -40 \sim +85^\circ\text{C}$)

| 項目 | 略号 | 条件 | MIN. | TYP. | MAX. | 単位 |
|--------------------|-----------------|---|------|------|------|-----|
| システム・クロック 発振周波数 | f_{CC} | $V_{\text{DD}} = 2.7 \sim 3.3\text{V}$, $R_{\text{OSC}} = 120\text{k}\Omega$ | 150 | 200 | 245 | kHz |
| | | $V_{\text{DD}} = 1.5 \sim 3.6\text{V}$, $R_{\text{OSC}} = 120\text{k}\Omega$ | 100 | 200 | 250 | kHz |
| | | $V_{\text{DD}} = 2.7 \sim 3.3\text{V}$, $R_{\text{OSC}} = 270\text{k}\Omega$ | 65 | 90 | 115 | kHz |
| | | $V_{\text{DD}} = 1.5 \sim 3.6\text{V}$, $R_{\text{OSC}} = 270\text{k}\Omega$ | 50 | 90 | 115 | kHz |
| | | $V_{\text{DD}} = 2.7 \sim 3.3\text{V}$, $R_{\text{OSC}} = 390\text{k}\Omega$ | 50 | 65 | 80 | kHz |

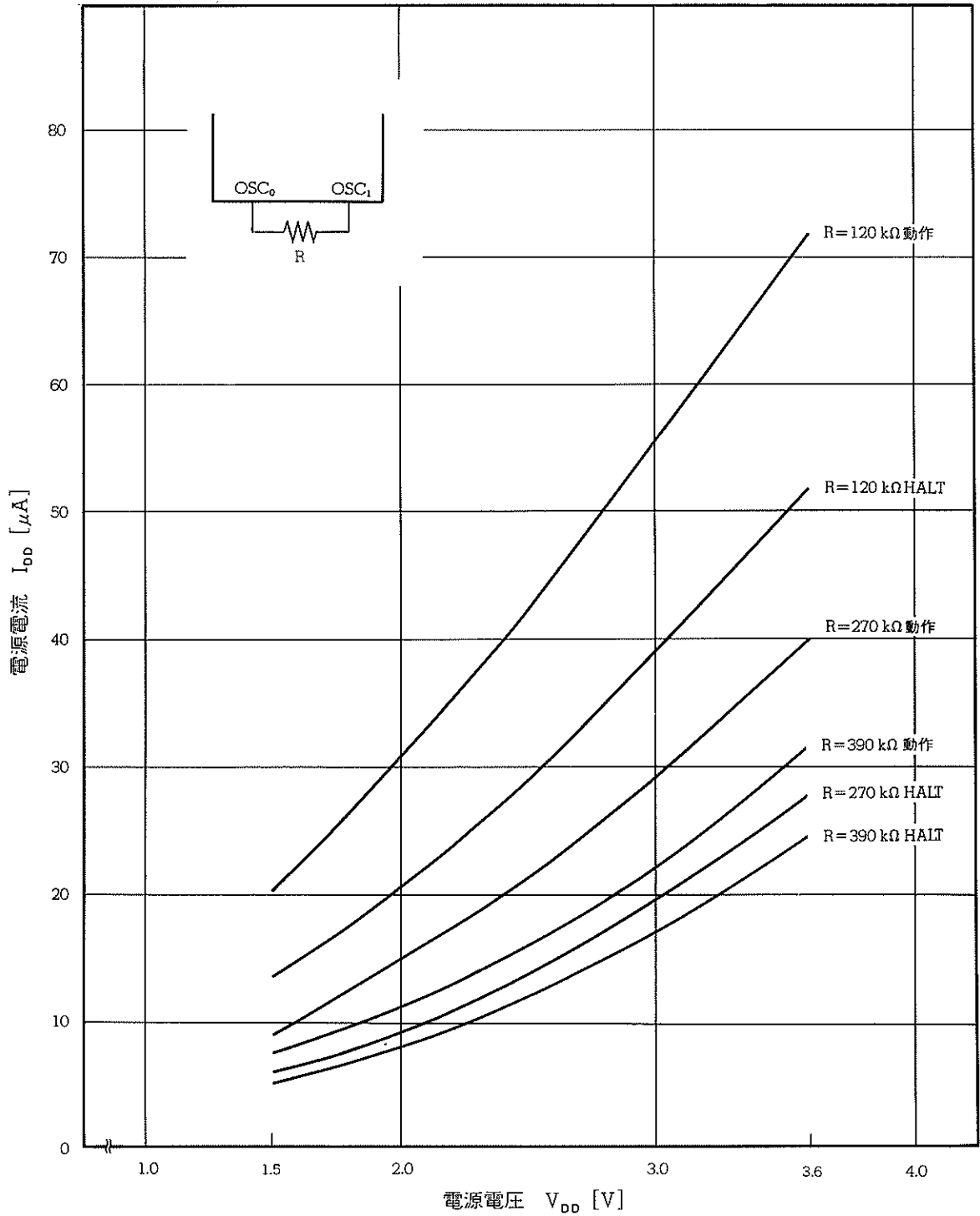
注意 上記の条件は抵抗の誤差を含みません。

12. 特性曲線 (参考値)



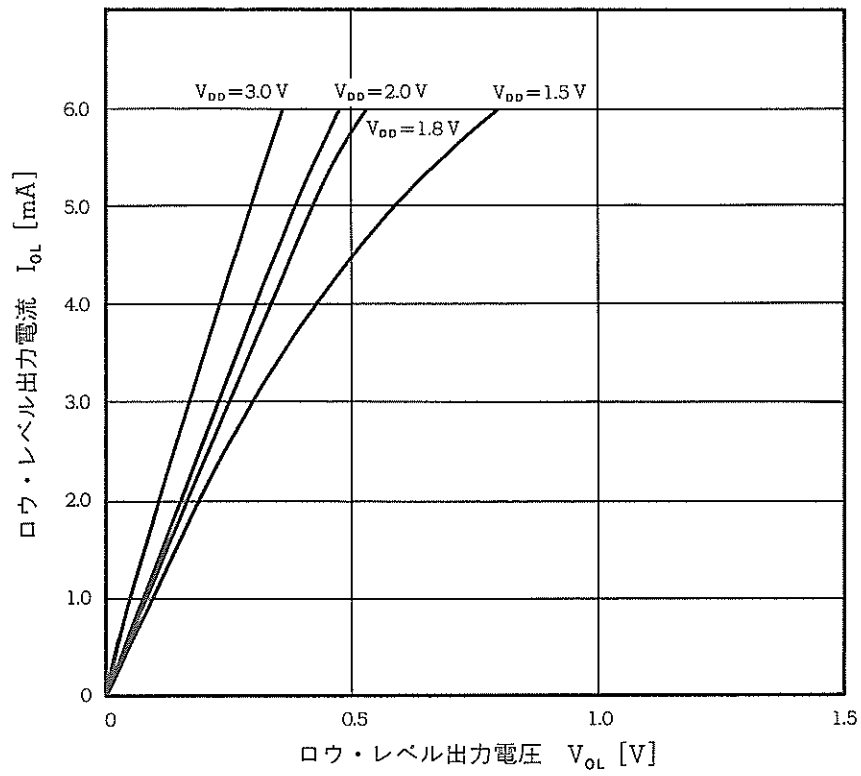
I_{DD} vs V_{DD}

($T_a = 25^\circ\text{C}$)



I_{OL} vs V_{OL}

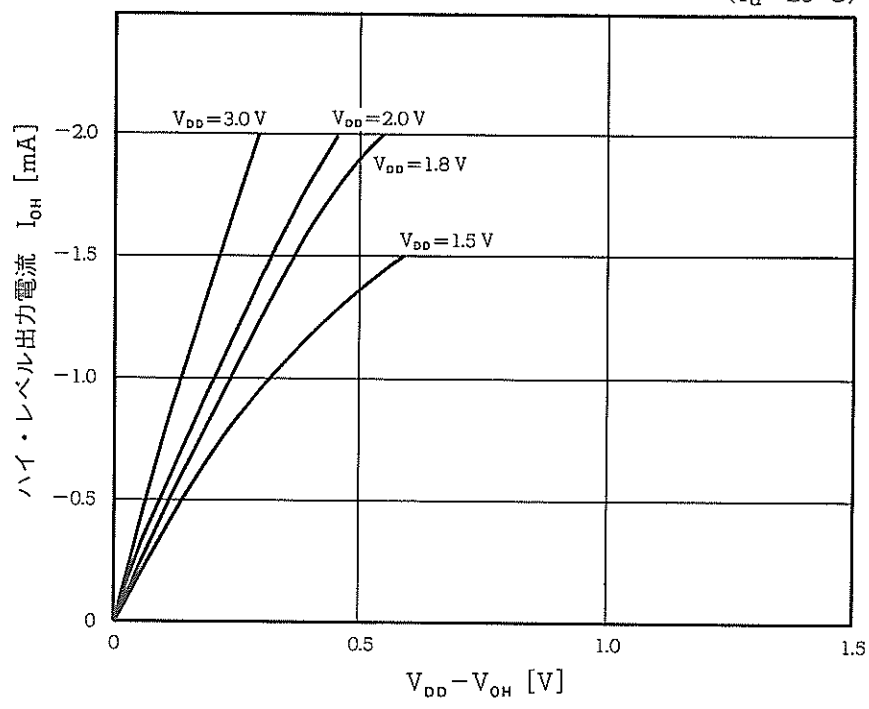
($T_a = 25^\circ C$)



注意 絶対最大定格は、1端子当たり 30 mA です。

I_{OH} vs ($V_{DD} - V_{OH}$)

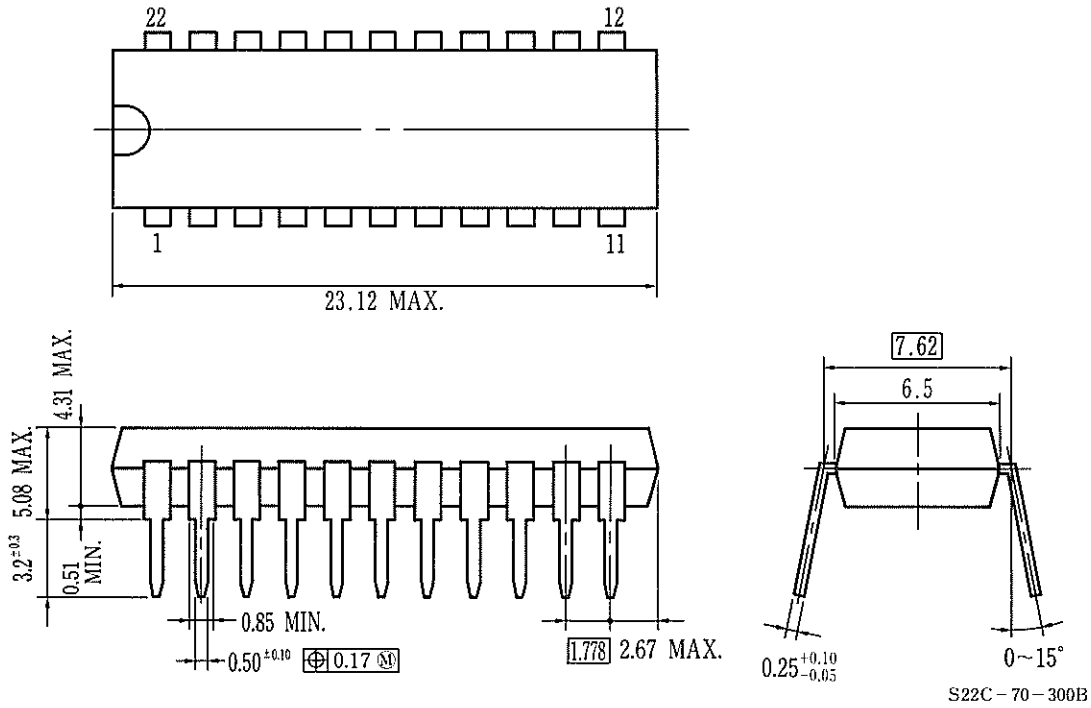
($T_a = 25^\circ C$)



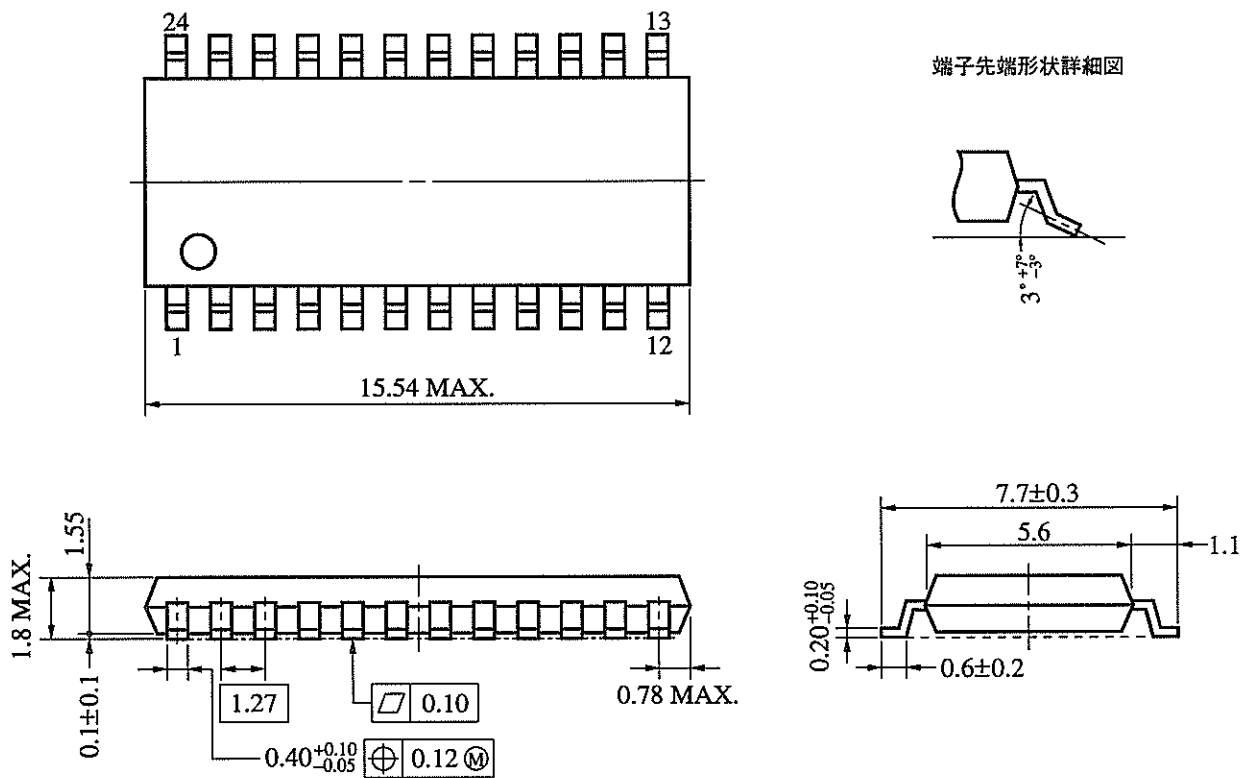
注意 絶対最大定格は、1端子当たり -5 mA です。

13. 外形図

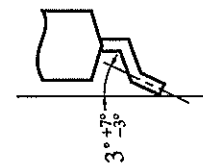
22ピン・プラスチック・シュリンク DIP (300 mil) 外形図(単位: mm)



24ピン・プラスチック SOP (300 mil) 外形図 (単位: mm)

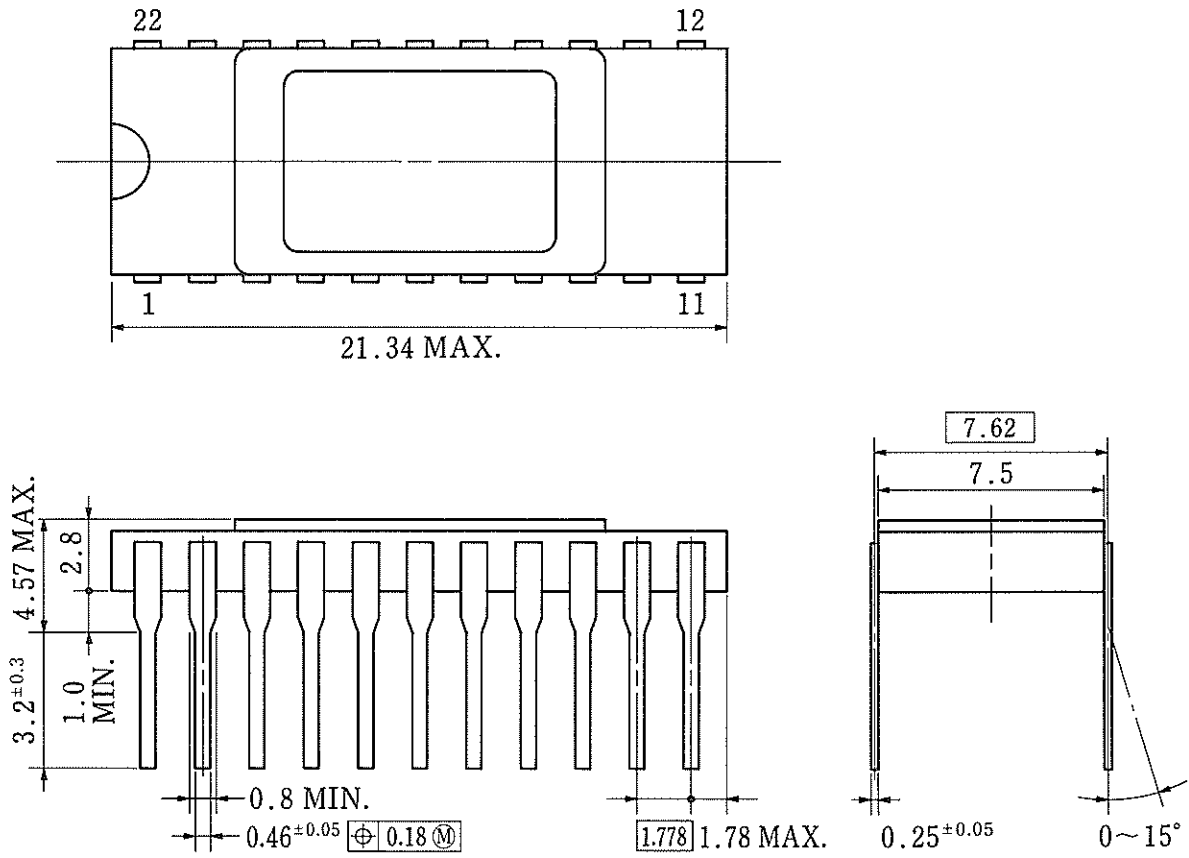


端子先端形状詳細図



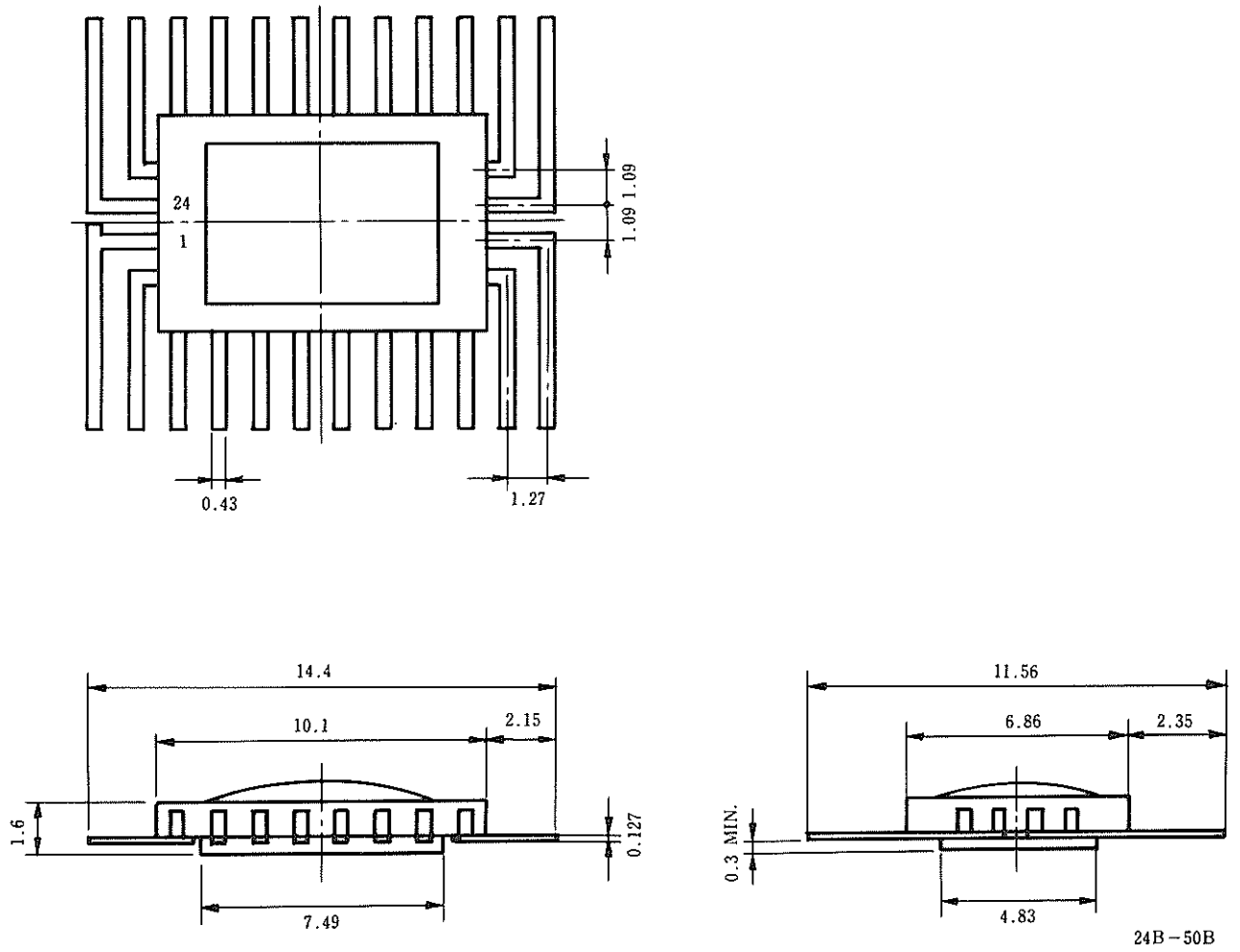
P24GM-50-300B-3

22ピン・セラミック・シュリンク DIP (300 mil) (ES用) 外形図(単位: mm)



P22D-70-300B

ES用24ピン・セラミック SOP 外形図 (参考) (単位: mm)



14. 半田付け推奨条件

★

μPD17108L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 14-1 表面実装タイプの半田付け条件

μPD17108LGS-×××：24ピン・プラスチック SOP (300 mil)

| 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：2回以内 <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避けください。 | IR35-00-2 |
| 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 <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避けください。 | VP15-00-2 |
| ウェーブ・ソルダリング | 半田槽温度：260℃以下、時間：10秒以内、回数：1回 予備加熱温度：120℃ MAX. (パッケージ表面温度) | WS60-00-1 |
| 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) | — |

注意 半田付け方式の併用は避けください (ただし、端子部分加熱方式は除く)。

表 14-2 挿入タイプの半田付け条件

μPD17108LCS-×××：22ピン・プラスチック・シュリンク DIP (300 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：300℃以下、時間：3秒以内 (リード1ピン当たり) |

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。

15. μPD17108L と μPD17108, 17P108 の違い

| 項 目 | μPD17108L | μPD17108 | μPD17P108 |
|---|--|--|------------------------------------|
| ROM | マスク ROM 1 K バイト (512×16ビット) | | ワン・タイム PROM 1 K バイト (512×16ビット) |
| POB ₀ -POB ₃ 端子の プルアップ抵抗 | マスク・オプション | | な し |
| RESET 端子の プルアップ抵抗 | | | |
| V _{PP} 端子, 動作モード 選択端子 | な し | | あ り |
| 発振安定待ち時間 (クロック数) | 8クロック | | 16クロック |
| 電源電圧 | 1.5~3.6 V (200 kHz 動作時) | 2.5~6.0 V (250 kHz 動作時) 4.5~6.0 V (1 MHz 動作時) | |
| 発振器特性 ^注 | μPD17108, 17108L, 17P108 で異なります。詳細については、それぞれのデータ・シートに記載されている特性曲線を参照してください。 | | |
| パッケージ | 22 ピン・プラスチック・シュリンク DIP (300 mil) 24 ピン・プラスチック SOP (300 mil) | | |

注 電源電圧および抵抗値が同じであれば、μPD17P108 は μPD17108, 17108L の発振周波数より約 1 割程度低い周波数で発振します。このため、μPD17P108 から μPD17108, 17108L にデバイス変更する場合には、抵抗値の変更が必要となります。

16. タイニ・マイクロコントローラ・ファミリー一覧表

| 項 目 | μPD17103 | μPD17104 | μPD17103L | μPD17104L | μPD17107 | μPD17108 | μPD17107L | μPD17108L |
|---------------------|--|--------------------------------------|------------------------|--------------------------------------|--|--------------------------------------|------------------------|--------------------------------------|
| ROM 容量 | 1 K バイト (512 × 16 ビット) | | | | | | | |
| RAM 容量 | 16 × 4 ビット | | | | | | | |
| 入出力ポート ^注 | 11 本 (3 本) | 16 本 (4 本) | 11 本 (3 本) | 16 本 (4 本) | 11 本 (3 本) | 16 本 (4 本) | 11 本 (3 本) | 16 本 (4 本) |
| システム・クロック | セラミック発振 | | | | RC 発振 | | | |
| 電源電圧 | 2.7-6.0V (2MHz 動作時) 4.5-6.0V (8MHz 動作時) | | 1.8-3.6V (2MHz 動作時) | | 2.5-6.0V (250kHz 動作時) 4.5-6.0V (1MHz 動作時) | | 1.5-3.6V (200kHz 動作時) | |
| パッケージ | • 16ピンDIP • 16ピンSOP | • 22ピン・ シュリンク DIP • 24ピンSOP | • 16ピンDIP • 16ピンSOP | • 22ピン・ シュリンク DIP • 24ピンSOP | • 16ピンDIP • 16ピンSOP | • 22ピン・ シュリンク DIP • 24ピンSOP | • 16ピンDIP • 16ピンSOP | • 22ピン・ シュリンク DIP • 24ピンSOP |
| PROM 製品 | μPD17P103 | μPD17P104 | μPD17P103 | μPD17P104 | μPD17P107 | μPD17P108 | μPD17P107 | μPD17P108 |

注 () 内はN-ch オープン・ドレイン出力の本数です。N-ch オープン・ドレイン出力は、マスク・オプションによりプルアップ抵抗の有無を選択できます。

付録 開発ツール

μPD17108L 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}) | <p>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。</p> <p>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/ATTM 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。</p> <p>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 SIMPLEHOSTTM 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。</p> <p>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。</p> |
| SE ボード (SE-17108) | SE-17108 は、μPD17108, 17108L, 17P108 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-17104CS) | EP-17104CS は、μPD17104, 17104L, 17P104, 17108, 17108L, 17P108 用のエミュレーション・プローブです。 |
| PROM プログラム (AF-9703 ^{注3} AF-9704 ^{注3} AF-9706 ^{注3}) | AF-9703, AF-9704, AF-9706 は、μPD17P108に対応した PROM プログラムです。プログラムアダプタ AF-9799 を接続することにより、μPD17P108をプログラミングすることができます。 |
| プログラムアダプタ (AF-9799 ^{注3}) | AF-9799は、μPD17P103, 17P104, 17P107, 17P108をプログラミングするためのアダプタです。AF-9703, AF-9704 または AF-9706 と組み合わせて使用します。 |

注 1. 廉価版：電源外付けタイプ

- 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。
- 安藤電気株式会社の製品です。詳細につきましては、安藤電気株式会社 (東京(03)3733-1151) までお問い合わせください。

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。 μPD17108L のプログラム開発には、この AS17K とデバイス・ファイル (AS17103L) 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 (AS17108L) | AS17108L は μPD17108L 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 (AS17K) 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 | μS5A10AS17103 ^注 |
| | | | | | 3.5 インチ 2HD | μS5A13AS17103 ^注 |
| | | IBM PC/AT | PC DOS | | 5 インチ 2HC | μS7B10AS17103 ^注 |
| | | | | | 3.5 インチ 2HC | μS7B13AS17103 ^注 |
| サポート・ソフトウェア (SIMPLEHOST) | SIMPLEHOST はインサートキット・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 | | 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 |

注 μS××××AS17103 には、AS17103, AS17104, AS17107, AS17108, AS17103L, AS17104L, AS17107L, AS17108L が入っています。

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。

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。

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 インダストリー半導体販売事業部 | 〒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(松下中目ビル) | 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 郡山支店 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 太田支店 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 長野支社 長野 (0262)35-1444 松本支店 松本 (0263)35-1666 上諏訪支店 上諏訪 (0266)53-5350 甲府支店 甲府 (0552)24-4141 埼玉支社 埼玉 (048)641-1411 立川支社 立川 (0425)26-5981 千葉支社 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 北陸支社 金沢 (0762)23-1621 | 福井支店 福井 (0776)22-1866 富山支店 富山 (0764)31-8461 京都支社 京都 (075)344-7824 神戸支社 神戸 (078)332-3311 中国支社 広島 (082)242-5504 鳥取支店 鳥取 (0857)27-5311 岡山支店 岡山 (086)225-4455 四国支社 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 九州支社 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】

| | | | |
|----------------------------|--------------------------------|-------------------|--|
| 半導体応用技術本部 マイクロコンピュータ技術部 | 〒210 川崎市幸区塚越三丁目484番地 | 川崎 (044)548-7923 | 半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお願い致します) |
| 半導体応用技術本部 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(松下中目ビル) | 名古屋 (052)242-2762 | |
| 半導体応用技術本部 西日本応用システム技術部 | 〒540 大阪市中央区城見一丁目4番24号(NEC関西ビル) | 大阪 (06) 945-3383 | |