資料

マイクロコンピュータのハードウェアの基礎

昭和58年3月

JIPDEC

財団法人 日本情報処理開発協会

財団法人 図書 1.11.6 日本情報処理開発協会

この資料は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて，昭和５７年度に実施した「マイクロコンピュータの応用に関する調査研究」の一環としてとりまとめたものです。

序

わが国におけるマイクロプロセッサの生産量は，昭和５７年度の１年間だけで１億６千万個にのぼる。しかもこの数は年々増加し続けることが予想される。この膨大な数のチップが家電製品に，プラントに，そしてパソコンに組込まれて行く。あらためてマイコンの威力に驚かざるを得ない。

マイコンを応用する技術は，いまやわが国のほぼすべての産業が必要としている。システム，あるいは機器の高機能化，低価格化，高信頼化を図るうえでマイコンチップが必需品になりつつあるからである。

本書はまさにこの技術の基礎の習得を助けるためのものとして企画されたものである。街の書店に行けばマイコンに関する書物を多数見出すことができる。しかしそれらの多くは，アマチュア向けやビジネスマン向けのものであり，本書が意図するような，マイコン埋込み技術を包括的に述べたものは数少い。たとえあっても内容が古くなってしまったりする。この分野の技術はいうまでもなく日進月歩である。また，関連する分野が極めて多岐にわたっている。この種の書物の企画は，言うはやすく，行うは難しである。敢えてここに刊行するのは，多くの読者の目に触れることによって，適切な批判と助言をいただき，それをもとにしてよりよいものを作りあげようという考えがあるからに他ならない。たとえ拙くとも，全くないよりはずっとましであるし，将来への足がかりになる。それがわれわれの初心であったし，また，こうして一応の体裁を整えた今もその気持ちにかわりがない。

以上のようなわれわれの趣旨をご理解いただき，読者諸賢の力強いフィードバックをいただけるならば，これにまさる喜びはない。

なお，本書の作成にあっては，コアデジタル㈱松本　喬氏，萩原電気㈱萩原義昭氏他多くの方々にご協力いただいた。ここに謝意を表する次第である。

昭和５８年３月

田　村　浩一郎

（電子技術総合研究所パターン情報部数理基礎研究室長）

目　　次

第１部　マイクロコンピュータの基礎知識 …… 1

第１章　部品の名称と特徴 …… 1

1.1　抵　　抗 …… 1

1.2　コンデンサ …… 2

1.3　ダイオード …… 4

1.4　トランジスタ …… 12

第２章　パルス回路の基礎 …… 16

2.1　トランジスタのパルス特性 …… 16

2.2　クランプ回路 …… 17

2.3　クリップ回路 …… 18

2.4　積分回路 …… 20

2.5　微分回路 …… 21

2.6　マルチバイブレータ …… 22

第３章　論理代数 …… 33

3.1　ブール代数の基礎定理 …… 33

3.2　論理回路の簡単化 …… 35

第４章　デジタルＩＣについて …… 38

4.1　論理記号 …… 38

4.2　記号および形状 …… 38

4.3　論理レベルと正論理，負論理 …… 39

4.4　状態表示記号 …… 40

4.5　バッファ増幅器 …… 43

4. 6　フリップフロップ …… 44
第 5 章　TTLの種類と使い方の例 …… 46
5. 1　S N 7 4 0 0 …… 46
5. 2　S N 7 4 7 6 …… 47
5. 3　S N 7 4 7 4 …… 50
第 6 章　MOS ICについて …… 52
6. 1　MOSの仕様と特徴 …… 52
6. 2　Pチャネル MOS IC …… 54
6. 3　Nチャネル MOS IC …… 55
6. 4　コンプリメンタリ MOS IC …… 56
第 7 章　MOSとTTLとの接続 …… 58
7. 1　PチャネルMOSからのインターフェース …… 58
7. 2　NチャネルMOSからのインターフェース …… 60
7. 3　CMOSとTTLとのインターフェース …… 61
第 8 章　演算増幅器の基礎 …… 64
8. 1　演算増幅器とは …… 64
8. 2　演算増幅器の基本的性質 …… 65
8. 3　理想演算増幅器とは …… 66
8. 4　理想演算増幅器の等価回路 …… 68
第 9 章　ノイズについて …… 73
9. 1　TTLの耐ノイズ性 …… 73
9. 2　ノイズの分類 …… 74
第 10 章　接地とリターン回路 …… 76
10. 1　筐体アース …… 76

10. 2　電気部品のアース …… 77
10. 3　直流電源のリターン回路 …… 78
第11章　4ビットマイクロコンピュータ …… 80

第2部　マイクロコンピュータの実際 …… 87
第1章　8085の構造と機能 …… 87
1. 1　CPUチップの役割 …… 87
1. 2　命令の構成とアドレス指定方式 …… 94
第2章　マイクロコンピュータシステム …… 107
2. 1　マイクロコンピュータシステムの構成 …… 108
2. 2　メモリの構成 …… 114
2. 3　インタフェースの考え方 …… 121
2. 4　割　込　み …… 129
2. 5　DMA（Direct Memory Access） …… 143
2. 6　8085に接続するハードウェアとインタフェース …… 152

第1部　マイクロコンピュータの基礎知識

第 1 章　部品の名称と特徴

1.1　抵　　抗

電気回路の中で最も数多く使用されているのは抵抗である。

抵抗には多くの種類があるが，特に多く使用されているのはカーボン皮膜抵抗であり，抵抗値の表示法としてはカラーコードが使用されている。理由は，どの面が上になっても，その抵抗値が読み取れるようになるからである。各色に対応する数字を表１－１に示す。

表１－１　抵抗値の表示

黒	茶	赤	橙	黄	緑	青	紫	灰	白	金	銀
0	1	2	3	4	5	6	7	8	9	-1	-2
										5	10

抵抗には図１－１に示すようにふつう４本のカラーが塗ってある。このカラーの意味は

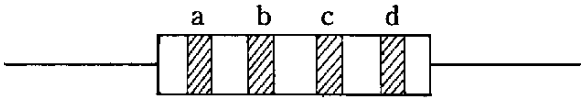


図１－１　抵抗値の表示

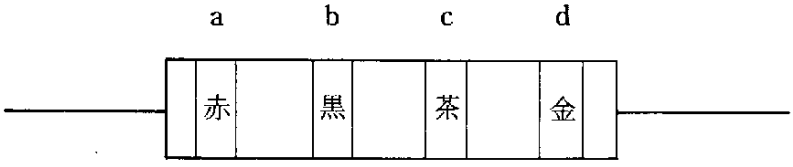
① a と b で抵抗値の２桁を表示している。

② c で抵抗値の桁数を表示しており 10^{n} となる。

③ d で抵抗値の誤差を表示している。

以上の３つの意味を実際の抵抗値を例にとって表示してみる。

例１



a は赤なので２を示す（表１－１参照）

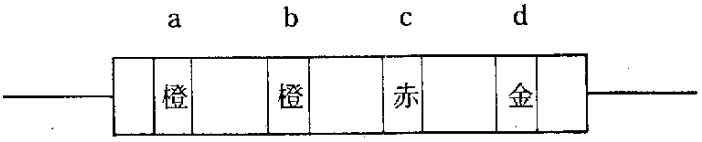
b は黒なので０を示す

c は茶なので１を示し 10^{1} の意味をもつ

dは金なので－1を示すが，この表示だけは抵抗値の誤差を表示しているので5％の意味を表わしている。

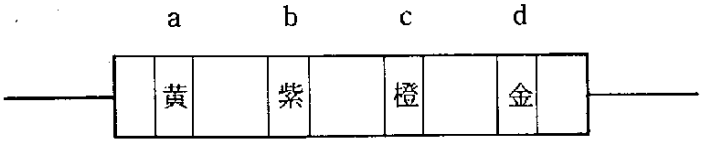
よって　　20×10¹＝200Ω　　誤差5％
　　↑↑　　↑　　↑
　　ab　　c　　d

例2



33×10²＝3300Ω　誤差5％
↑↑　　↑　　↑
ab　　c＝3.3kΩ　　d

例3



47×10³＝47000Ω　誤差5％
↑↑　　↑　　↑
ab　　c＝47kΩ　　d

ここで抵抗値の単位について説明する。

1 M(メグ) Ω ＝ 1000 K(キロ) Ω

1 K(キロ) Ω ＝ 1000 Ω

Mはメグとよみ$10^6\Omega$となる。

Kはキロとよみ$10^3\Omega$となる。

抵抗のシンボルマークとして が使用される。

1.2　コンデンサ

コンデンサは電荷を蓄える部品で，キャパシタともいう。容量が固定のものを固定容量コンデンサ，容量が可変のものを可変容量コンデンサという。

1.2.1　容量誤差

コンデンサの場合その容量は表示値の通常±数％～20％の範囲に入って

いるが，電解コンデンサやセラミックコンデンサでは±10%～±50%程度のものも使用されている。この誤差は少い方が良いが，誤差を小さくすることは価格の上昇に結びつくので，許容できる範囲を明確にすればよい。

1.2.2 使用電圧

コンデンサにかかる直流電圧と交流電圧の最大値を加算したものを使用電圧という。特に交流とことわっていない場合は直流の電圧であるから，交流やリップルの大きな直流に対しては注意を要する。定格値の80%以下で使用したほうがよい。温度が上昇すると定格電圧は小さくなる。一般にフイルムコンデンサは耐圧（使用に耐えられる最大電圧）が高く，電解コンデンサは耐圧が低いものが多い。耐圧と容量値は同一形状では反比例の関係にある。

1.2.3 損　　失

実際のコンデンサでは漏れ電流が流れる。これによって発熱してエネルギが失なわれ，損失が起きる。この損失の度合は次のようにして表わす。理想コンデンサでは漏れ電流がないため，加えた電圧に比べて位相が90度進んだ電流が流れる。しかし実際のものでは漏れのため90度よりδだけ遅れた位相の電流が流れる。このδを損失角という。δの正接をとって$\tan\delta$とあらわし，コンデンサの良さを表わす目安として用いる。場合によっては

$$Q = 1 / \tan\delta$$

という関数でQを用いることもある。Qは大きい方がよい。

1.2.4 使用温度範囲

温度が上昇すると$\tan\delta$が増大し，各種の特性が悪くなり，寿命が加速度的に短くなる。その他，急激な充電（コンデンサに電荷を蓄えること），放電（コンデンサに蓄えられた電荷を流出すること）はコンデンサの寿命を短くするし，極性の指定されたコンデンサはこれを守る必要がある。

1.2.5 コンデンサの容量

1F（ファラッドと読む）は1Aの電流を1秒間流し込んだ時，電圧が

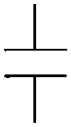
1Ｖ上昇する容量のことをいう。これは非常に大きい容量であって，ふつうの電子回路には１μＦ（マイクロファラッドと読む）程度のものが用いられる。

$$1\ \underset{\text{ファラッド}}{F} = 10^{6}\ \underset{\text{マイクロ ファラッド}}{\mu F}$$

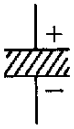
$$1\ \underset{\text{マイクロ ファラッド}}{\mu F} = 10^{6}\ \underset{\text{ピコ ファラッド}}{PF}$$

1.2.6　コンデンサのシンボル

一般のコンデンサ



極性のある電解コンデンサ（この場合は，－側に＋側より高い電圧を加えてはいけない）。



1.3　ダイオード

1.3.1　整流用ダイオード

ダイオードに流せる最大電流を容量といい，容量が１Ａ程度のものから数百Ａのものまで各種製造されている。

ダイオードの構造を図１－２に示す。ダイオードはｐ形半導体とｎ形半導体で接合面をつくり，リード線を設けて，樹脂，ガラスまたは金属ケースに封入したものである。

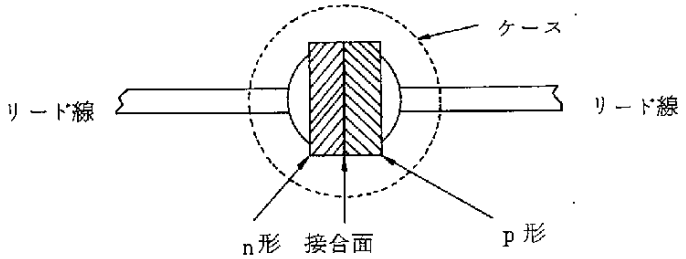


図１－２　ダイオードの構造

数A以下の小容量ダイオードは，樹脂またはガラス等でシールされており，数A以上のダイオードは，ハーメチックシールと呼ばれる金属性のケースに窒素ガスとともに封入されている。

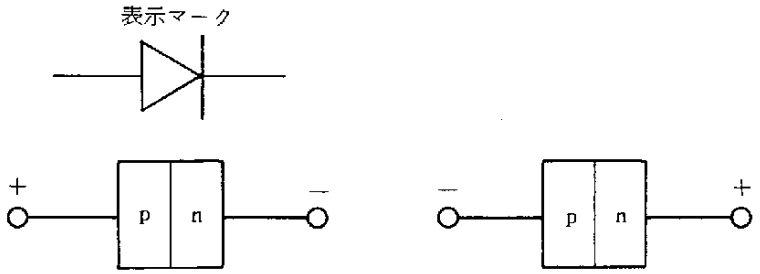


図1－3　ダイオードの正方向，逆方向

ダイオードは図1－3に示すようにpnに対して図1－3（a）のような方向の電圧が加わるとわずかな電圧（シリコンダイオードでは約0.7 V）以上で電流が流れ，逆に図1－3（b）のような方向の電圧が加わると電流はほとんど流れない。この様子を電圧・電流でグラフに書くと図1－4のようになる。

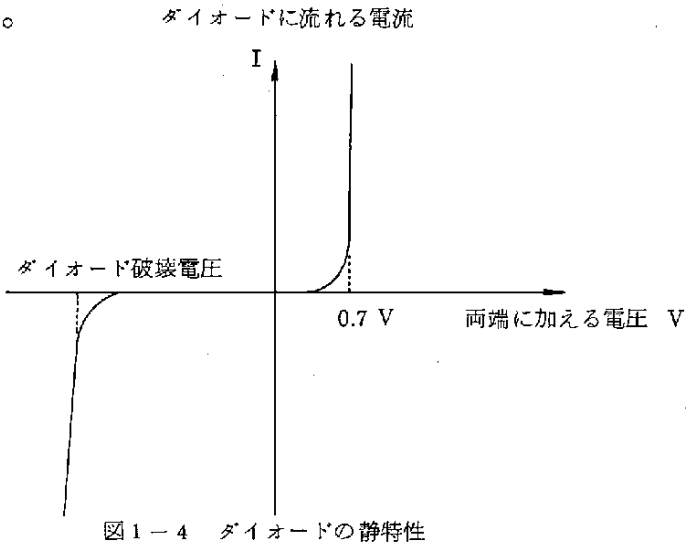


図1－4　ダイオードの静特性

これをダイオードの静特性と呼ぶ。すなわち正方向では約0.7 V以上の電圧で電流は急激に流れ，逆方向では，通常数百ボルトの点までは，電流

はほとんど流れない（常温で数A級のダイオードでは数μA以下）。しかしこの点を越えて逆電圧が印加されると，電流は急激に流れる。この場合は大きな熱が発生し，ダイオードを破壊するので，この電圧をこえない範囲でダイオードを使用しなくてはならない。この電圧はダイオード破壊電圧（ピーク逆電圧ともいう）と呼ばれる。

また，順方向に電流が流れるとき，約0.7Vの電圧降下があるため，流れる電流に比例して熱が発生する。例えば10Aを流すと7W，100Aを流すと70Wの熱が接合部に発生するが，外部に放散して接合部分を常に一定温度（最大150℃程度）以下にしないと連続してダイオードを使用できない。

ダイオードの定格は，カタログに表示されているが，小容量のものは定格電流付近で使用する場合，リード線による熱の放散も大きく影響するのでリード線の長さを約30㎜以上に保つように心掛けなければならない。また周囲温度も50℃をこえる場合は，定格以下で使用したほうが安全である。さらに正方向の電流はシリコンダイオードでは10ms位の短時間ならば，定格電流の50倍ぐらい流すことができるが，これをひんぱんに流すと破壊することがある。また，もちろんピーク逆電圧をこえる逆の電圧も短時間であっても破壊する恐れがある。このような電圧が発生する回路に使用する場合は，コンデンサやサージ・アブソーバとも呼ばれる素子を用いてダイオードを保護する方法がよくとられる。

1.3.2　信号用ダイオード

ダイオードの持つ静電容量は少ないが，接合部でのリカバリータイム（図1－5のtの時間でダイオードにかかる電圧が順方向から急激に逆方向に変化した時，接合部に残った電子の動作により起る現象でこの時間は短いほうが好ましい）が短いために100KHZ以上の高周波の整流にも使用できるものを信号用ダイオードと呼ぶ。

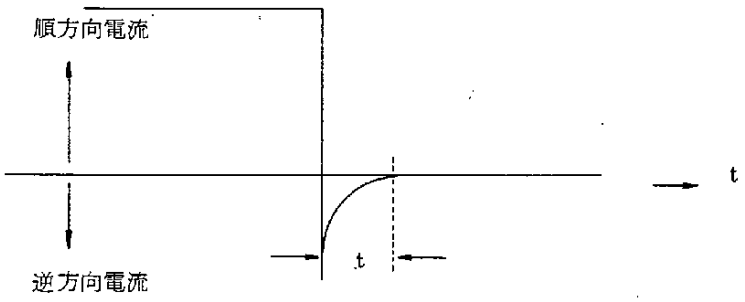


図1－5　リカバリータイム

信号用ダイオードは接触形と接合形の2種類がある。この信号用ダイオードは整流ダイオードと同じく，整流作用がある。一般にダイオードの静電容量は金属と半導体の接合，半導体のpn接合の所にできる。ダイオードが逆バイアス状態においても加えた交流電圧の周波数が高周波であると，ダイオードとしての特性を失い，この容量を通って電流が流れ，低周波では整流性があっても高周波になると整流性が出ないことがある。これを十分考慮したダイオードが信号用ダイオードである。ゲルマニウムを用いたものは順方向では0.2 V程度から電流が流れ，シリコンでは順方向において0.7 V程度から電流が流れる。小信号用にはゲルマニウムが良いが，逆方向性ではゲルマニウムは漏れ電流が多く，耐電圧も低く，温度特性も悪いので，シリコンを用いたものが現在では主流である。ポイント・コンタクトのダイオードでは，ミリ波，マイクロ波を整流できるものもある。接合形のものは接合容量が多いが，中波ラジオ音声信号の検波，テレビ映像信号の検波などに広く用いられている。

1.3.3　高速用ダイオード

p形，n形の半導体にはキャリアと呼ばれる電流を運ぶものがある。これが外側から加えられる電界によって正方向電界ならば接合面をキャリアが通過して電流が流れるが，逆方向電界であれば，接合面をキャリアが通過できず，電流が流れない。この電界が正から逆に変る速度が早いといままで接合面を通過中のキャリアが急には無くならないので逆方向に反転し

たときに若干のキャリアが遅れて接合面を通過する。これは通常50HZ程度の周波数では無視できる値であるが，20KHZ位の高い周波数になってくると，この値が問題となり，整流の効率を下げてしまう結果になる。そこで接合面付近でキャリアを早く消滅させるように，金などを接合面付近に拡散する方法がとられる。こうすると通常のダイオードに比べて，電界が反転した時に，接合面付近でキャリアが早く消滅する。このために20KHZ程度の周波数でも十分整流ができる。この金拡散を施したダイオードをファースト・リカバリー・ダイオード（F・R・D）と呼び20KHZ～50KHZ程度の周波数で動作させるインバータやコンバータによく使用されている。周波数がさらに高くなるとFRDも使用できないので，さらに高い周波数で整流できる，電力用のダイオードが必要となる。この種のダイオードとしてショットキー・バリヤ・ダイオードが使用される。このダイオードは，普通の半導体ダイオードのようなpn接合を持たず，p形に比べてキャリアの移動速度の速いn形半導体と金属を接合させて整流を行わせるために高周波の電力整流には適している。ただし，逆方向の耐電圧が低いので現状では60V程度の電圧にしか使用できない欠点もある。近年，コンバータやスイッチングレギュレータでは高周波化，小形化の傾向が進み，すでに100KHZ～200KHZの周波数でショットキーバリヤダイオードが使用されるようになっている。

1.3.4　定電圧ダイオード

ダイオードの静特性として逆方向の耐電圧以上の電圧が加わると急激に電流が流れるが，これを積極的に利用したダイオードを定電圧ダイオード（ツェナーダイオードともいわれる）と呼ぶ。整流用のダイオードは通常数百V以上の耐電圧をもつが，半導体材料を調整して数V～数十Vの低い耐電圧の素子をつくることができる。この場合，逆方向に多少の電流を流しても，その損失＝（印加している電圧）×（流れる電流）が数十mW～数W程度（例えば10Vで5mAであれば50 mW）であれば，ダイオードを

破壊させないで使用することができる。このとき，逆方向で破壊を起す電圧は素子により一定であるので，電流の変化に対して一定電圧を得る回路によく利用される。この定電圧ダイオードは構造的には整流用のダイオードとほとんど変りなく，接合面の材料が若干異なる。したがって，外形，形状もほとんど通常のダイオードと同じである。記号はダイオードマークの肩に定電圧特性を意味するカギのマークを付けてダイオードと区別する（図１－６）。また，静特性は図１－７に示すが，ダイオード破壊電圧に相当する電圧を定電圧（ツェナー電圧）という。

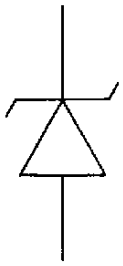


図１－６定電圧ダイオードのシンボル

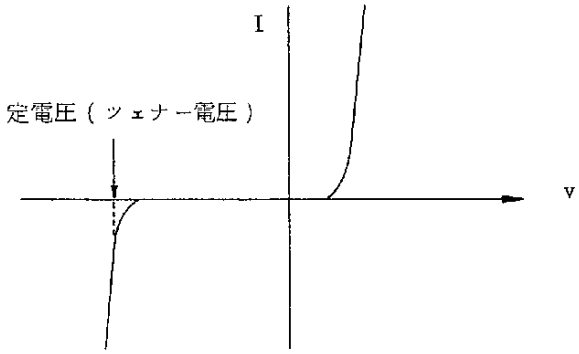


図１－７　定電圧ダイオード静特性

1.3.5　フォト・ダイオード

p形半導体とn形半導体を接合したpn接合を持ったダイオードや，半導体と金属を接合したショットキー・バリア・ダイオードに逆方向の電圧をかけ（このことを逆バイアス状態という），接合の部分に光を当てると入射光強度に比例した電流がダイオードに流れる。これを光電流という。このように光がダイオードにあたることで光電流が流れるようにしたダイオードのことをフォトダイオードという。一般のダイオードでもこの性質があるが，フォトダイオードは接合の所に光が入りやすいような構造になっている。通常はゲルマニウムまたはシリコンを用いたダイオードが一般的である。ただしこれらのダイオードで光をあてたとき光電流が流れるためには光の波長に制限がある。ゲルマニウムの場合は1.6 μmより短波長の光に，シリコンの場合には約1.1 μmより短波長の光に応答する。

半導体に一般に光があたって吸収されると，電子とホールのペアが作られる。p形またはn形半導体中にそれぞれ少しぐらいホールや電子が増加しても，もとからのホールや電子の数が多いため増加分は目立たない。しかし逆バイアス状態に電圧をかけたpn接合や，ショットキー・バリヤの所には空乏層といわれる高抵抗層ができ，ここには電荷は存在しない（図1－8）。この空乏層に光があたると，ホール（＋），電子（－）ができるが，もともとこの領域はホールや電子は少ないため，それらの増加の割合は非常に大きい。この空乏層には外部からかけた逆バイアスの電圧によって高電界が作られている。光で発生した電子とホールはこの電界によってそれぞれの極へ引かれ，外部回路には電流として現われる。

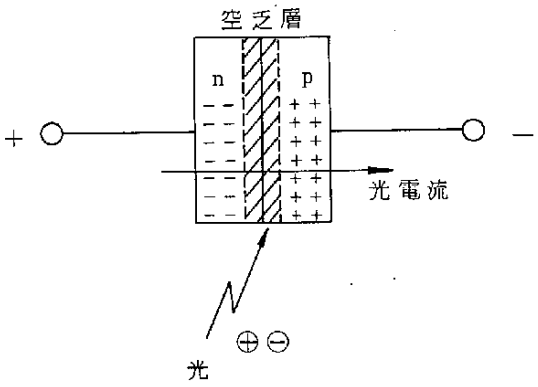


図1－8　光電流の流れ

フォトダイオードとして注意すべき性質は光電流が小さいため，光があたらないときの電流（暗電流といわれる）はできるだけ小さいことである。また，高速の光信号に応答するため，ダイオード逆バイアス時の接合の容量が小さいことが望まれる。特に高速の光信号に応答するために接合の中間にi層と呼ばれる高抵抗層をはさみ込んだpin構造にしたフォト・ダイオードがあり，1GHZ程度の周波数まで応答できる。同じく高周波まで利用でき，アバランシェ効果と呼ばれる電子ナダレを利用して高感度にしたアバランシェ・フォト・ダイオードがある。

1.3.6　発光ダイオード

最近はパイロットランプを用いた表示装置のかわりに発光ダイオード（LED）が多量に用いられるようになった。これはパイロットランプの寿命が1000時間程度しかないのに比べると半導体を用いた発光ダイオードは

ほぼ無限大の寿命をもつためである。また発光色もGa As半導体を利用した赤外発光ダイオード，Ga AsPを用いた赤色発光ダイオード，GaPを用いた緑色や，黄色発光ダイオードなど，豊富でかつ安価に市販されている。最近GaN，SiCを用いた青色の発光ダイオードも試験的につくられ，市販されるようになった。

一般に市販されている赤外，赤，緑，黄の発光ダイオードは前述のような材料を用い，それら材料のｐ形，ｎ形半導体を利用してｐｎ接合を形成したものが普通である。「整流ダイオード」の項でも述べたように，pn接合のため電圧電流特性は，シリコン，ゲルマニウムで作ったものと同様の整流特性を持っている。シリコンのpn接合では順方向電流が立ち上がるのが約0.7Ｖであるが，これらの発光ダイオードの場合は1.7～1.8Ｖ以上の電圧を順方向にかける必要がある。それは順方向電流の立ち上がりのときｐｎ接合の境界に材料の種類や特性によって決まる障壁ができるために通常のｐｎ接合よりも高電圧が必要となる。発光の原理は順方向に電流を流すことでｐ形半導体中のホールとｎ形半導体中の電子がｐｎ接合の境界領域で衝突して，それぞれの持っている正と負の電荷が消滅するが，その際，余分のエネルギーが光に変換されて放出される。シリコンやゲルマニウムダイオードでは材料のエネルギー構造が異なっていて光ではなく熱として余分のエネルギーを放出している。

電流がどのくらい光に変換されるかが実用上たいせつであるが，赤外用のもので10％程度，赤色発光で数％，緑色で0.1～0.05％程度である。これらのダイオードはTO18と呼ばれる金属円形台またはセラミック上にのせ，光が有効に利用できる構造にして，レンズまたは樹脂を用いて封じたものが多い。これらの発光ダイオードは信頼性が高く，長寿命，機械的にも強い，高輝度，高速動作が可能である，などの特徴を持っている。

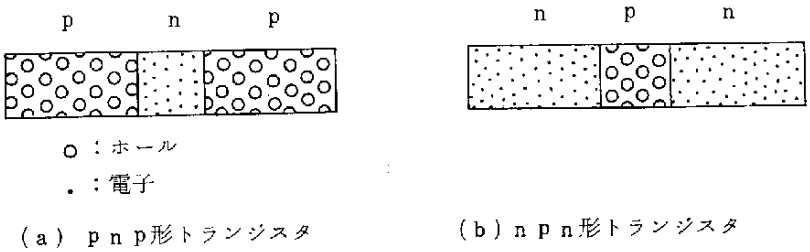
1.4　トランジスタ

抵抗，コンデンサなどの素子を受動素子というのに対して，トランジスタは能動素子といわれ，トランジスタと受動素子を組合わせて増幅，発振，スイッチなどの動作ができる。

トランジスタはゲルマニウムやシリコン（現在はほとんどシリコン）で作られた2個のpn接合を利用した素子を組合わすことにより，pnp，npnと2種類のトランジスタができる。トランジスタの分類の1つとして低周波用と高周波用に分類される。（明確な周波数の境界はないが，可聴周波数（10KHZ～15KHZ程度）を低周波と考えればよい。）低周波用はB，D，高周波用はA，Cの記号で表示される。A，Bはpnp形，C，Dはnpn形で，2SA，2SB，2SC，2SDの呼びかたで分類されている。最初の2は3本足の素子を表わし，2番目のSは半導体であることを示している。またA，B，C，Dの後に各トランジスタの番号が付けられる。

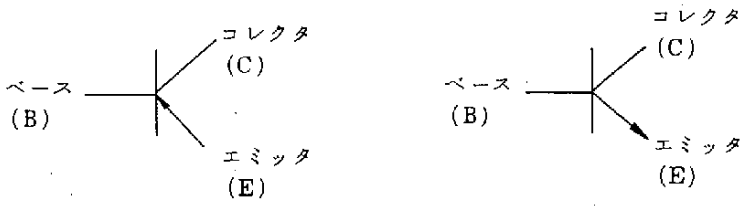
1.4.1　トランジスタの構成

トランジスタは図1－9（a）のように，非常に薄いn形半導体を2つのp形半導体の間にサンドイッチのように接合した構造になっている。これをpnp形トランジスタといい，これとは逆に図1－9（b）のようにn形半導体の間にp形半導体をはさみこんだものをnpn形トランジスタという。図1－10にトランジスタのシンボルを示す。



（a）pnp形トランジスタ　　（b）npn形トランジスタ

図1－9　トランジスタの構成



（a）pnp形トランジスタ　　（b）npn形トランジスタ

図1－10　トランジスタのシンボル

1.4.2　トランジスタの動作原理

いま，このpnp形トランジスタに図1－11（a）のように電池をつなぐとダイオードの逆バイアスに相当するから電流はほとんど流れない。次に図1－11（b）のように左側のp形の部分と中央のn形の部分にp形が＋，n形が－となるように別の電池をつなぐと左側のpn部分は順バイアスとなるためp形の部分のホールはn形の部分に向って注入される。n形

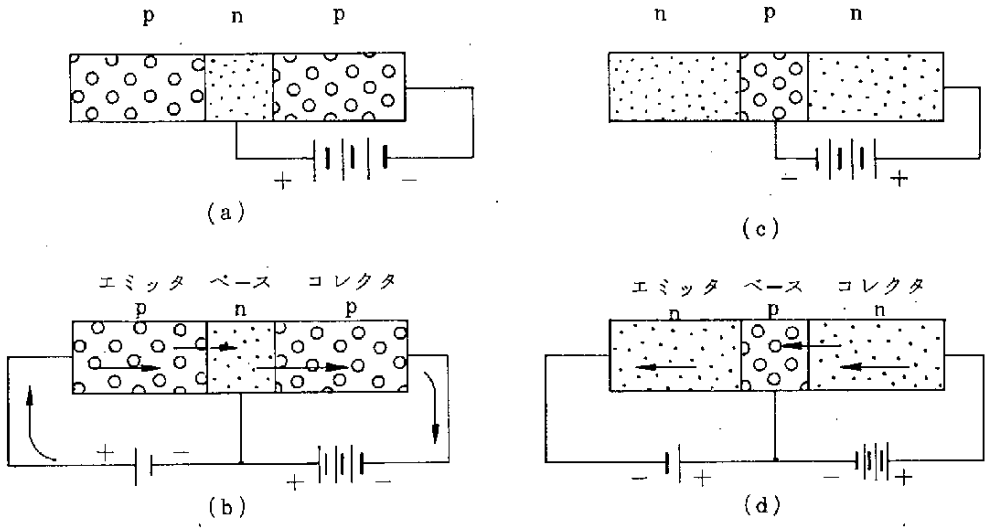


図1－11　トランジスタ動作図

の部分は非常に薄くつくられているので注入されたホールのほとんど全部が右側の境界（ｎｐ接合部）に達する。そしてここには右側の部分に引き寄せる電界があるので右側のｐ形の部分に入ってしまう。図１－11(b) に記した３つの部分の左側のホールを注入するｐ形部をエミッタ（Ｅ），中央のｎ形部をベース（Ｂ），右側のホールを集めるｐ形部をコレクタ（Ｃ）と呼ぶ。

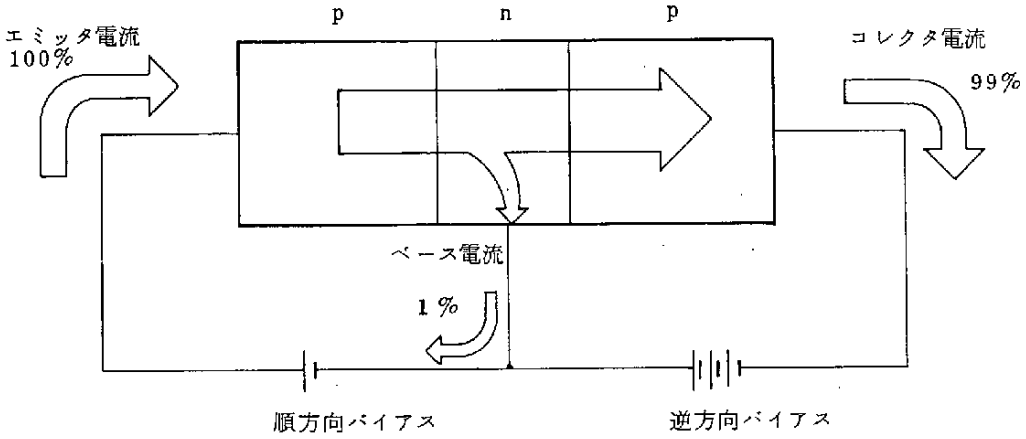


図１－１２　トランジスタ内部を流れる電流

ここでトランジスタにかける電圧と，それによって流れる電流を考えてみよう。エミッタとベースの間は順バイアスなので，低い電圧でもよく電流が流れる。またベースとコレクタの間は逆バイアスで使用するが，コレクタを流れる電流はコレクタにかける電圧にはあまり左右されず，エミッタに流れる電流によって大きく支配される。図１－１２はトランジスタの各部を流れる電流を示したもので，コレクタに流れる電流はエミッタに流れる電流の９８～９９%，エミッタからベースに流れる電流は１～２%で大部分がコレクタに流れる。

ここで図１－１３に示すように，エミッタに小さな交流信号を加え，コレクタに大きな負荷抵抗をつないだ状態を考えてみよう。エミッタとベースの間は順バイアス方向であるから抵抗が低く，小さい交流信号電圧を加えても相当の信号電流がエミッタに流入する。そして，この電流の大部分は

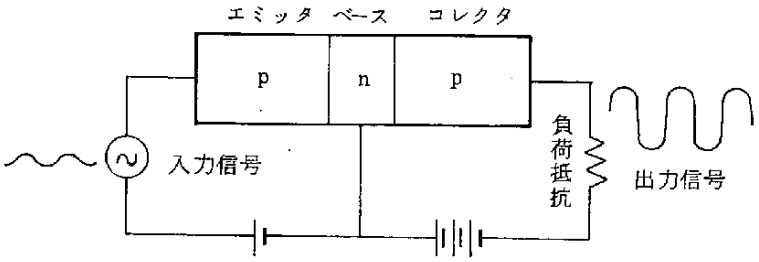


図１－１３　トランジスタの増幅作用

コレクタから負荷抵抗の中を流れ，負荷抵抗の両端には大きな信号電圧が現われる。すなわちトランジスタは増幅作用をすることがわかる。以上はｐｎｐ形トランジスタの場合を例にとったが，ｎｐｎ形トランジスタの場合でも原理は同様である。ただし，トランジスタ動作図１－１１（ｃ）（ｄ）のように電池のつなぎ方がｐｎｐ形と全く反対で，また，キャリアは電子である点が異なっている。なおトランジスタのシンボルは図１－１０に示すとおりでｐｎｐ形とｎｐｎ形の区別はエミッタ電流の方向によって区別し,pnp形では，エミッタの矢印をベースに向って書き，ｎｐｎ形では，矢印をベースから外の方向に向って書くようにしている。

第２章　パルス回路の基礎

パルスとは正弦波でない波形をいうが，一般には方形波がその主体である。パルス回路とは波形を整形したり，あるいは発生したりする回路で，その応用は電子計算機をはじめとして，データ通信，交換機などの広い分野に使用されている。パルス回路に使用される素子は，ダイオード，トランジスタをはじめサイリスタなどの半導体やリレー，磁心などが用いられるが，トランジスタ，ダイオードがその主体となる。ここではトランジスタとダイオードを使ったパルス回路について，その基礎的な回路を取りあげ，動作原理について述べる。

2.1　トランジスタのパルス特性

トランジスタを図２－１（ａ）に示すように接続して，ベースにパルス電

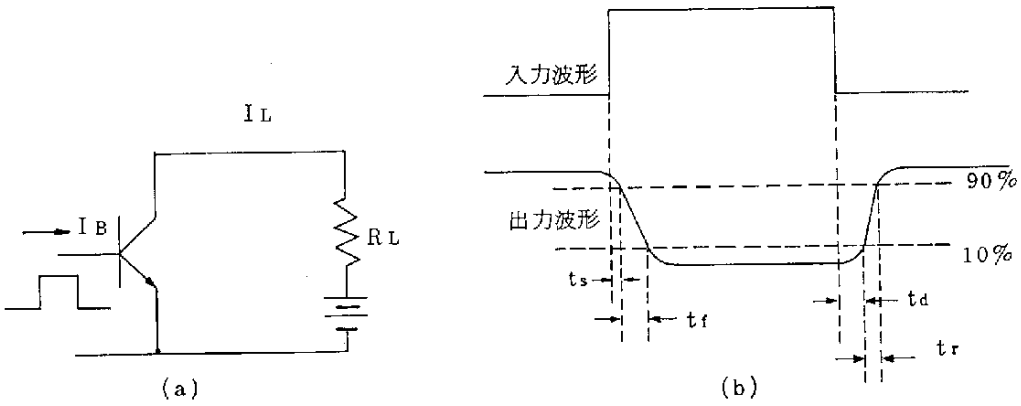


図２－１　スイッチ回路とパルス応答

流を加えるとパルスの加わっているときだけトランジスタはON状態になり，コレクタには図２－１（ｂ）のような波形が得られる。コレクタに現われる電流波形は一般に丸みをおび，加えた波形とは違ってくる。この違いは次のような４つの定数で表わされる。

①　遅延時間：td（delay　time）

入力パルスにより出力パルスの振幅が最大振幅の１０％に達する時間

② 立ち上がり時間：t_r （rise time）

出力パルスが最大振幅の10％から90％に達する時間

③ 蓄積時間：t_s （storage time）

入力パルスにより出力パルスの振幅が最大振幅の90％に減少するまでの時間

④ 立ち下がり時間：t_f （fall time）

出力パルスが最大振幅の90％から10％までに減少する時間

t_d, t_r, t_s, t_f は使用するトランジスタや回路条件によって異なるが，この時間が短いほど，高速スイッチングに適することになる。特にトランジスタではt_sが大きくなるので設計上の注意が必要である。

2.2 クランプ回路

入力波形（方形波）の振幅や波形を変えずに，そのままの状態で出力する回路をクランプ回路といい，入力波形を変えずに信号レベルだけを動かすので，レベルシフト回路ともいわれている（図2－2）。

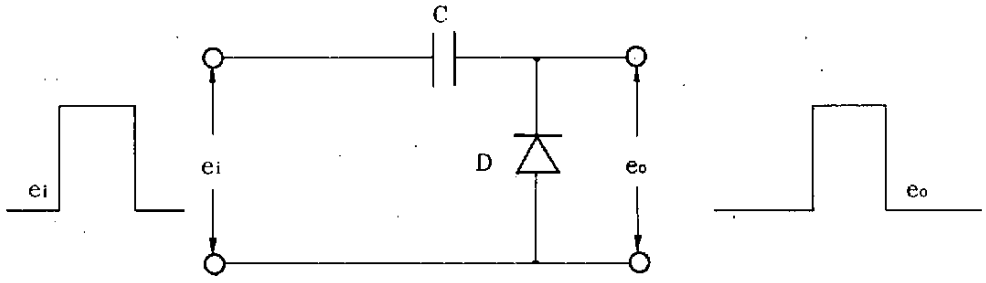


図2－2 クランプ回路

また入力信号が直流分を含んでいない入力波形に直流分を付け加えることから，直流再生回路ともいわれている。たとえば図2－2は出力電位は"0"に保たれ，図2－3の出力電位は"0"電圧の位置が"E"だけシフトし，あたかも，直流電圧を加えたようになるので，これを直流付加回路ともいう。

図2－4（a）では，スイッチング回路のダイオードD_1，D_2がクランプの役目をして，指定した電圧レベルを作る。この2つのダイオードが無い場

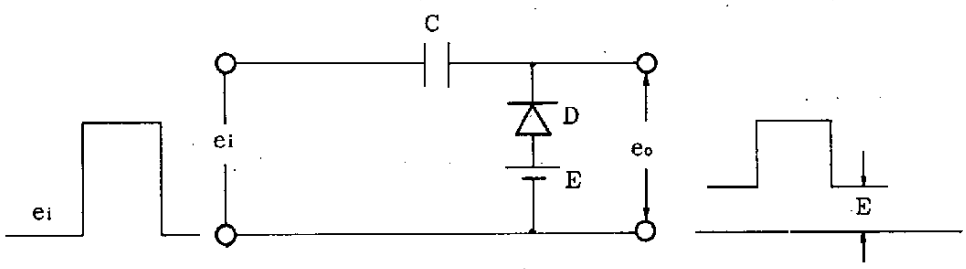


図２－３　直流付加回路

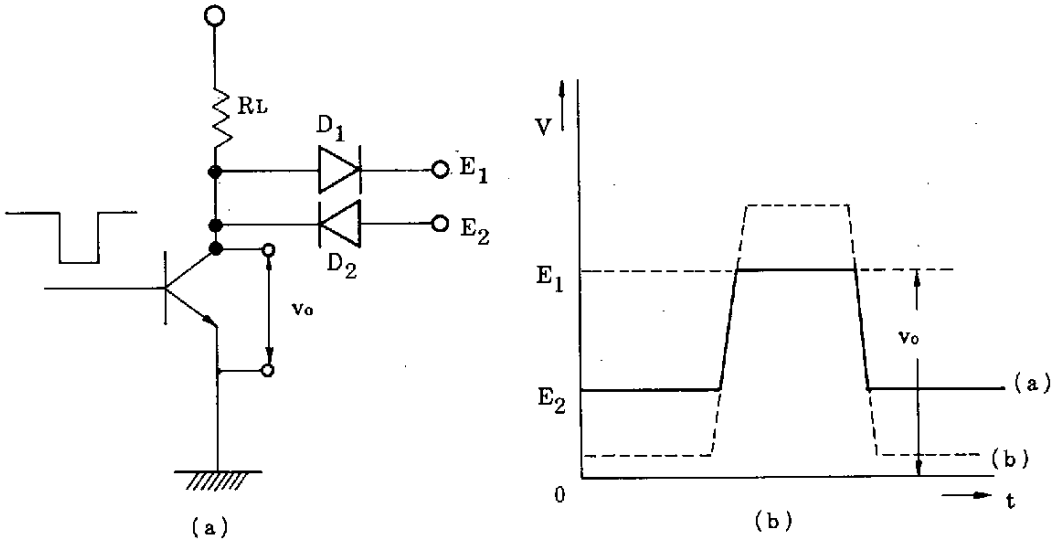


図２－４　クランプによる尖鋭な立ち上がり立ち下がりパルスの作り方

合は出力として図２－４（ｂ）の点線部分が得られることになる。D_1 は出力電圧 v_o を E_1 でおさえ，D_2 は出力電圧 v_o を E_2 よりも低くならないように働いている。このようにして，出力波形は，図のような尖鋭な立ち上がり，立ち下がり特性のよい方形波になる。

2.3　クリップ回路

通称クリッパとも呼ばれ，入力信号波形の歪んだ部分をクリップして整形するものである。すなわち，入力波形のあるレベル以上，またはレベル以下の波形を作り出す回路である。またこれは実動的に波形の立ち上がり，立ち下がりを改善することにもなる。

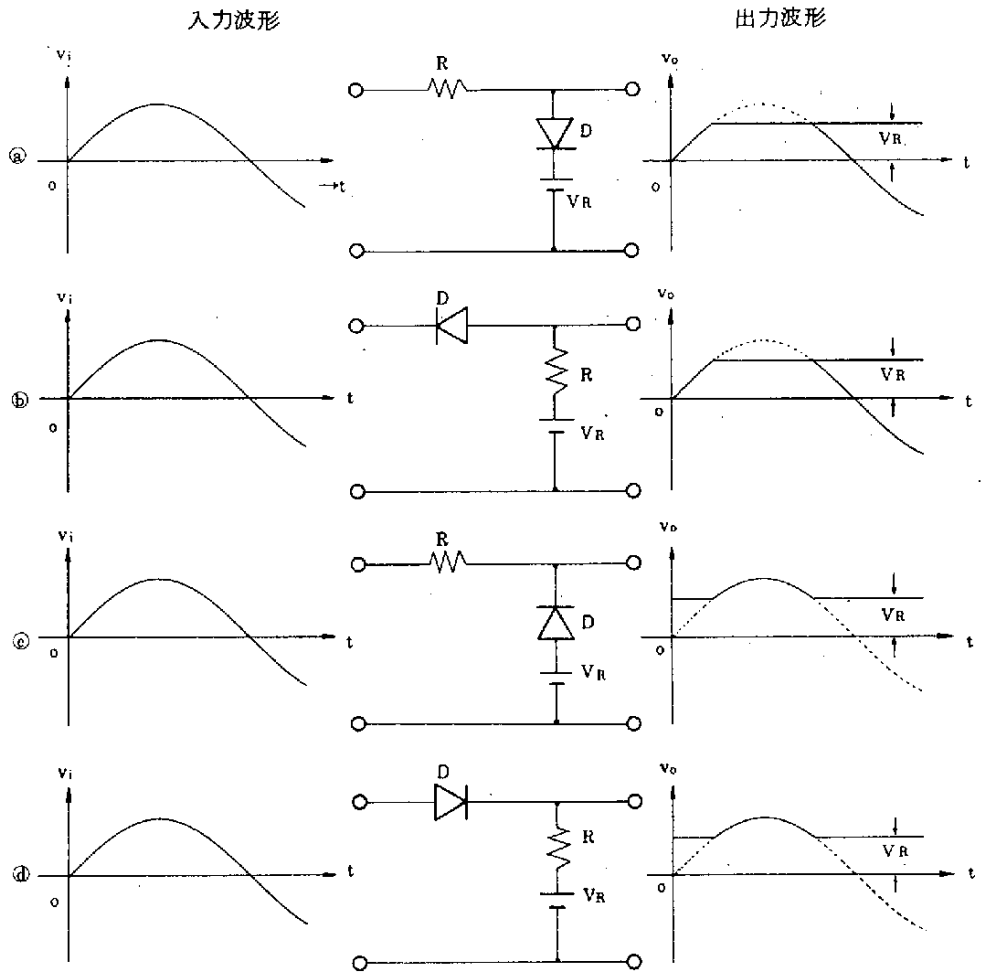


図２－５　クリップ回路のいろいろ

クリップ回路には並列型（図２－５（a），（c））と直列型（図２－５（b），（d））があり，回路構成すなわち，ダイオードを直列に入れるか，並列に入れるか，電源やダイオードの極性をどちらにするかによって分けら

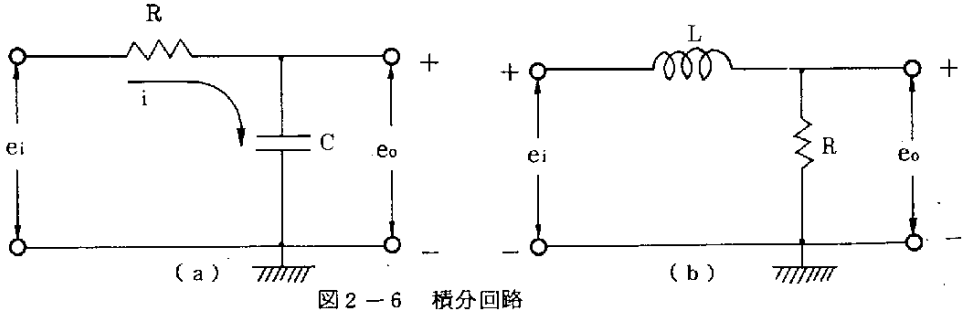
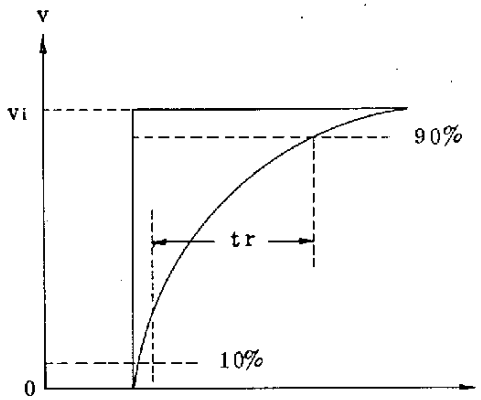


図２－６　積分回路

れる。図２－５のうち出力波形の点線の部分は入力波形がクリップされた部分である。

2.4　積分回路

積分回路はRC型(図2−6(a))とRL型（図２－６（b））とがある。ＲＬ型ではＲＣ型と同様な積分波形が得られるが，Ｌを設定することが難かしいことと，大きな値にすることが不便なので，あまり使用されていない。ここではRC型について述べる。



（ｃ）
図２－６　積分回路

ＲＣ型積分回路に図２－６（c）で表わされるようなステップ電圧 v_i を与えたときの出力端子電圧 e_o は，時定数 $\tau = CR$ をもった指数関数で表わされ，０Ｖから定常状態の電圧に向って上昇する。t_r は出力波形の立ち上がり時間である。出力電圧 e_o は

$$e_o = e_i \left(1 - \varepsilon^{-\frac{t}{\tau}}\right) \quad \text{ただし} \tau = CR \quad \text{RL回路では} \tau = R/L$$

となる。なぜなら，

図２－６（ａ）でコンデンサＣの両端の電圧 e_c は

$$e_c = \frac{1}{c} \int_0^t i\,dt$$

電流 i は e_i ／Ｒ よって，出力電圧 e_o は

$$e_o = e_c \fallingdotseq \frac{1}{CR} \int_0^t e_i\,dt$$

となる。

この結果からもわかるとおり，Ｃの両端の電圧は入力電圧 e_i の積分値の１／ＣＲ倍となる。

2.5　微分回路

パルス波（図２－７（ｂ））を入力信号として与えたときに図２－７(c)のようなシャープな波形をもつ出力信号，すなわち入力信号波形を時間について微分した信号を発生する回路が微分回路で，トリガ回路（ピストルの引金を引くという意味で，回路に動作の指示の開始信号を作るための細いパルスを作る回路）や細いパルスを作るのに用いられる。出力電圧の最大値は入

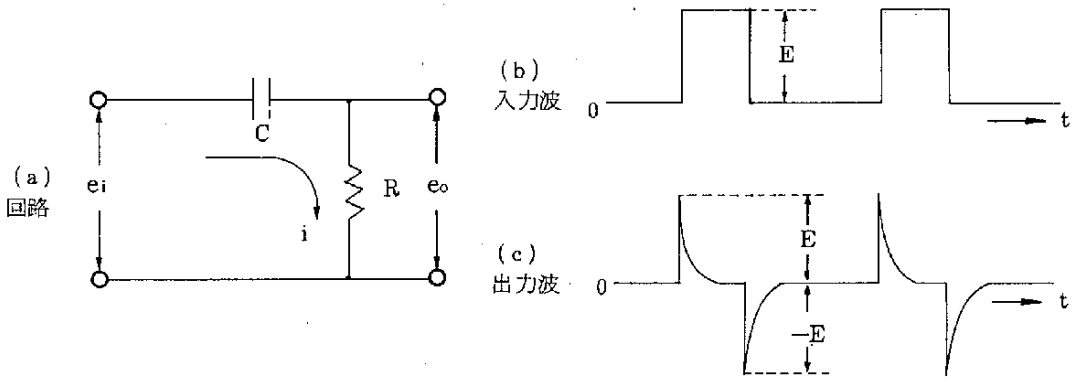


図２－７　微分回路と入，出力波

力の電圧値の２倍の出力となる。入力信号を $e_i = E \sin \omega t$ としこれを微分すると

$$\frac{de_i}{dt} = \omega E \cos \omega t$$

となる。このことから周波数応答が角周波数ωに比例するような特性をもつ回路を微分回路という。

図２－７（ａ）の図で抵抗Ｒの電圧降下 e_o は

$$e_o = \frac{j\omega CR}{1+j\omega CR} e_i$$

となり，ここで $\omega CR \ll 1$ とすれば

$$e_o \fallingdotseq j\omega CR\, e_i$$

となって，この回路の周波数特性は角周波数ωに比例するので微分回路とみなすことができる。

2.6　マルチバイブレータ

マルチバイブレータはパルス発生回路として使用されており，結合回路の構成（コレクタ結合型，エミッタ結合型）や回路の安定状態により，非安定（無安定）マルチバイブレータ，双安定（2安定）マルチバイブレータ，それに単安定マルチバイブレータがある。

2.6.1　非安定マルチバイブレータ

この回路はまったく安定状態がなく，自動的に動作し続け，片一方の状態からもう一方の状態へ，回路のもつ時定数によって交互に発振を続ける。

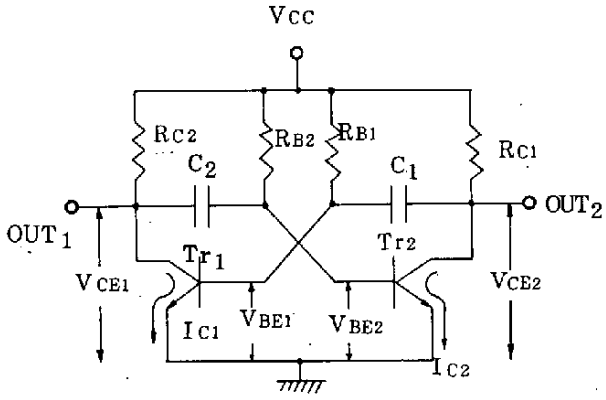


図2－8　非安定マルチバイブレータ回路

このようなことから，この回路（図2－8）を別名，自走マルチバイブレータとも呼んでいる。このマルチバイブレータは方形波出力を作り出し，その発振周波数はそれぞれ2つの時定数$C_1 R_{B1}$，$C_2 R_{B2}$で決められる。

図2－8において，トランジスタ1がOFF，トランジスタ2がONになった瞬間をt_1（図2－9）とする。トランジスタ1のベース電位V_{BE1}はカットオフベース電圧$V_{BE(0)}$より負に落ち込み，トランジスタ2のV_{BE2}は飽和ベース電圧V_{BE}（sat）となる。したがってこのt_1の状態では

$$I_{C1} = I_{CB0}$$

$$V_{CE1} = V_{CC} - I_{CB0} \cdot R_{C1}$$

I_{CB0}：コレクタしゃ断電流

V_{CE}（sat）：飽和コレクタ電圧

トランジスタ1がOFFであるから

〔 I_{C2} ＝設定値 〕

$V_{CE1} \fallingdotseq V_{CC}$

$V_{CE2} = V_{CC} - I_{C2}R_{C2} = V_{CE}$（sat） …… (1)

となり，次に $t_1 \rightarrow t_2$ に移るに従って，V_{BE1} は C_1 に蓄えられた負の電荷が R_{B1} を通じて放電されるため，図2－9（a）に示すとおり＋E_B に向

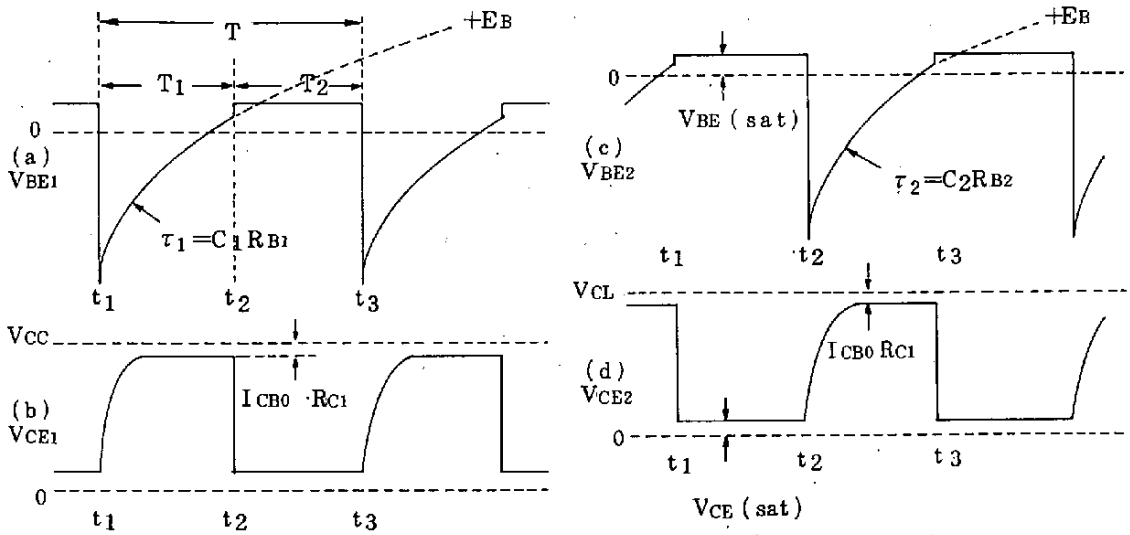


図2－9　非安定マルチバイブレータの波形関係

って上昇する。そして t_2 になった時点で，$V_{BE(0)}$ よりわずかに正になる。この時，$\triangle I_{B1}$ なるわずかなベース電流が流れる。したがって，トランジスタ1のコレクタには $\triangle I_{B1}$ の h_{FE} 倍になる $\triangle I_{C1}$ が流れ，トランジスタ1の V_{CE1} を $\triangle I_C R_{C1}$ だけ低下させる。この変化が C_2 を通してトランジスタ2のベースに加えられ，$\triangle I_{C1}$ R_{C1} だけ V_{BE2} が低くなる。そのために I_{B2} が $\triangle I_{B2}$ だけ減り，I_{C2} は $\triangle I_{B2} \cdot h_{FE} = \triangle I_{C2}$ だけ小さくなる。（h_{FE} ＝直流電流増幅率で I_C / I_B で定義される）

すると，V_{CE2} が $\triangle I_{C2} \cdot R_{C2}$ だけ上昇し，この変化が C_1 を通して，トランジスタ1のベースに入り，V_{BE1} を $\triangle I_{C1} \cdot R_{C1}$ だけ上昇させるので，

それに比例して$\triangle I_{B1}$がさらに流れ込み，$\triangle I_{C1}$を大きくし，V_{BE1}がベース飽和電圧$V_{BE}(sat)$に達するため，トランジスタ2をカットオフにする。

t_2の状態は　〔 I_{C1} ＝設定値 〕

$$V_{CE1} = V_{CC} - I_{C1} \cdot R_{C1} = V_{CE}(sat)$$

$$I_{C2} = I_{CB0}$$

$$V_{BC2} = V_{CC} - I_{CB0} \cdot R_{C2} \quad \cdots\cdots (2)$$

トランジスタ2がOFFであるから

$$V_{CE2} \fallingdotseq V_{CC}$$

となり，(1)式と(2)式の状態は完全に対称で，トランジスタ1とトランジスタ2が飽和とカットオフの状態を交互にくり返す。

回路値の設定：コレクタ負荷抵抗R_{C1}，R_{C2}は出力インピーダンスや負荷電流によって決められ，前述の(2)式から

$$R_{C1} = \frac{V_{CC} - V_{CE}(sat)}{I_{C1}} \fallingdotseq \frac{V_{CC}}{I_{C1}}$$

したがって

$$I_{C1} \fallingdotseq \frac{V_{CC}}{R_{C1}}$$

で与えられる。R_{B1}は

$$R_{B1} = \frac{V_{CC}}{I_{B1}}$$

トランジスタがONになった時の飽和条件は

$$I_{B1} \geqq I_{C1} / h_{FE}$$　　（ただしh_{FE}：電流増幅率）

であるから

$$R_{B1} \leqq \frac{V_{CC}}{I_{B1}}$$

$$\leqq \frac{(V_B - V_{BE}(sat)R_{C1}}{V_{CC} - V_{CE}(sat)} \cdot h_{FE}$$

$$\leqq \frac{E_B}{E_C} \cdot h_{FE} \cdot R_{C1}$$

（ただし，E_B，E_Cはそれぞれベース，コレクタの実効電源電圧）

もし$E_B = E_C$とすれば

$$R_{B1} \leqq h_{FE} \cdot R_{C1}$$

が与えられる。

発振周波数：対称波形を得るには$C_{B1} = C_{B2}$とすればよく，周期Tは

$$T = T_1 + T_2 \quad \cdots\cdots (3)$$

$$= 0.69 (R_{B1} \cdot C_1 + R_{B2} \cdot C_2) \quad \cdots\cdots (4)$$

で与えられる。対称型の回路の場合

$$R_{B1} = R_{B2} = R, \qquad C_1 = C_2 = C$$

とすれば，(4)式は

$$T = 1.38\,RC \quad \cdots\cdots (5)$$

で表わすことができる。

ではどうしてそうなるのか計算式で出してみると，まず

$$V_{BE1} = V_{BE}(sat)$$

$$= E_B - (E_B + V_{CC} - V_{CE}(sat)) \exp\left\{-\frac{T_1}{(C_1 R_{B1})}\right\}$$

で表わされ，これを変形すると

$$\frac{E_B - V_{BE}(sat)}{E_B + V_{CC} - V_{CE}(sat)} = \exp\left\{-\frac{T_1}{C_1 R_{B1}}\right\}$$

両辺の対数をとって

$$\log_e \frac{E_B - V_{BE}(sat)}{E_B + V_{CC} - V_{CE}(sat)} = -\frac{T_1}{C_1 R_{B1}}$$

よって

$$T_1 = C_1 R_{B1} \log_e \frac{E_B + V_{CC} - V_{CE}(sat)}{E_B - V_{BE}(sat)}$$

一般には$V_{CC} \gg V_{CE}(sat)$, $E_B \gg V_{BE}(sat)$であるから

$$T_1 \fallingdotseq C_1 \cdot R_{B1} \log_e \left(1 + \frac{V_{CC}}{E_B}\right) \quad \cdots\cdots (6)$$

となる。上式から$V_{CC} = E_B$であるから（図2－8では共通電源としている）(6)式は

$$T_1 = 2.3\, C_1 R_{B1} \log_{10} 2 \fallingdotseq 0.69\, C_1 \cdot R_{B1} \quad \cdots\cdots (7)$$

として表わすことができる。同様にT_2も(6)式より

$$T_2 \fallingdotseq C_2 \cdot R_{B2} \cdot \log_e \frac{E_B + V_{CC} - V_{CE}(sat)}{E_B - V_{BE}(sat)}$$

$V_{CC} \gg V_{CE}(sat)$, $E_B \gg V_{BE}(sat)$であるから

$$T_2 \fallingdotseq C_2 R_{B2} \cdot \log_e \left(1 + \frac{V_{CC}}{E_B}\right)$$

$V_{CC} = E_B$であるから

$$T_2 = 0.69\, C_2 \cdot R_{B2} \quad \cdots\cdots (8)$$

で表わされる。以上から，最初の(3)式にもどってみると，(7)と(8)式から

$$T = 0.69\,(C_1 R_{B1} + C_2 R_{B2})$$

となり(4)式を満足し(3)式は

$$T = 1.38\, RC$$

の結果になる。したがって

$$f = \frac{1}{T} = \frac{1}{1.38\, RC}$$

となる。

これまで発振周波数について述べたが，周波数はV_{CC}，E_Bという供給電圧と$C_1 R_{B1}$，$C_2 R_{B2}$なる時定数の関係で表わされる。図2－8のような回路では$E_B = V_{CC}$であり，周波数は時定数のみで決まる。したがっ

て前述のように

$$C_1 = C_2 = C$$

$$R_{B1} = R_{B2} = R_B$$

$$R_{C1} = R_{C2} = R_C$$

にすれば

$$T_1 = T_2$$

となり出力波形は対称な方形波を得ることができる。

2.6.2 双安定マルチバイブレータ

双安定（2安定）マルチバイブレータは，通称フリップフロップ回路とも呼ばれ，図2－10がその代表的な回路である。回路はふつう左右対称である。

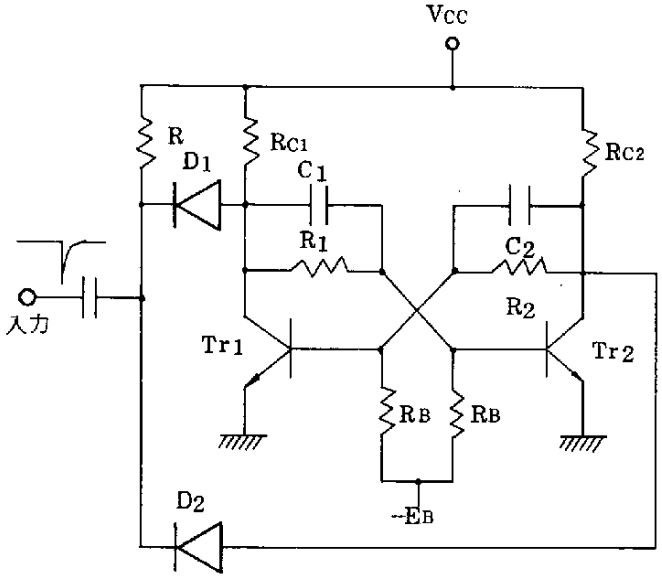


図2－10　双安定マルチバイブレータ回路図

入力から見てもわかるとおり電源を入れたときにトランジスタ1かトランジスタ2のどちらかがONになり，これは偶然の要素で決まる。トランジスタ1がON状態になるとトランジスタ1のコレクタ電圧V_{CE1}は

$$V_{CE1} \fallingdotseq V_{CE}(sat) \fallingdotseq 0$$

となる。

トランジスタ1がONであるからトランジスタ2がOFFになり，トランジスタ2のコレクタ電圧V_{CE2}は

$$V_{CE2} = V_{CC} - I_{CB0} \cdot R_{C2} \fallingdotseq V_{CC}$$

となる。

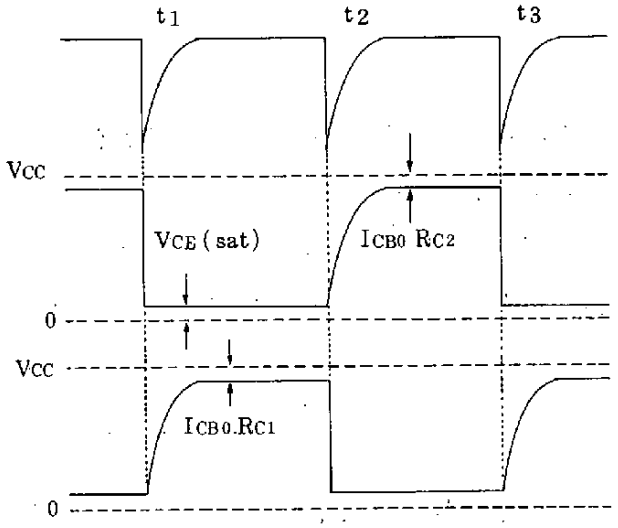


図2－11　コレクタ波形

このような状態では図2－11のようにt_1の時点で負のトリガパルスを加えると，逆バイアスされているダイオード1にはパルスは通過せず，ダイオード2のON状態からトランジスタ2のコレクタ電圧が下がり，C_2を通ってトランジスタ1のベース電位を低下させ，トランジスタ1のコレクタ電流は，さらに減少する。負のトリガパルスが入ると，すなわちt_1の時点から

V_{CE2}が低下 ⟶ V_{BE1}が低下 ⟶ I_{C1}が減少 ⟶ I_{C2}が増大 ⟶ V_{BE2}が上昇 ⟶ V_{CE1}が上昇 ⟶

となる。

スピードアップ・コンデンサC_1，C_2の役目：R_1，R_2にそれぞれ並列に入っているもので，スピードをアップする効果がある。

図2－12でC_{BE}はトランジスタがOFFのときのベース入力容量で，

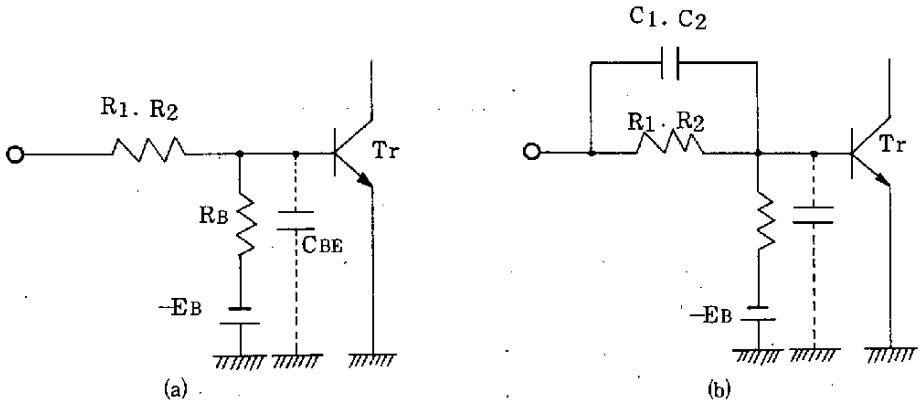


図2－12　スピードアップコンデンサの働き

トランジスタがOFFのとき（a）のようにC₁，C₂がないとR₁，R₂を通してベースに出力信号が入るが，CBEによってベースの立ち上がりが遅くなり，入力トリガ・パルスのくり返しが速い場合，動作が不安定となる。これをなくすために，R₁，R₂と並列にコンデンサC₁，C₂を入れる。

この場合

$$C_1 = \frac{C_{BE} \cdot R_B}{R_1}$$

$$C_2 = \frac{C_{BE} \cdot R_B}{R_2}$$

に選べば立ち上がりが補正できる。

2.6.3　単安定マルチバイブレータ

図2－13が基本回路であり，コレクタ結合型とエミッタ結合型があるが，ここでは一般的なコレクタ結合について述べる。

この動作は，あるトリガパルスの入力によって，ある時間幅をもった出力を出し，それを過ぎると再び元の安定状態にもどるもので，2.6.1の非安定マルチバイブレータと2.6.2の双安定マルチバイブレータを合せたものと同じ働きをする。図2－13の回路では，電源を供給すると，トランジスタ1がOFF，トランジスタ2がONになりこの時の安定状態は

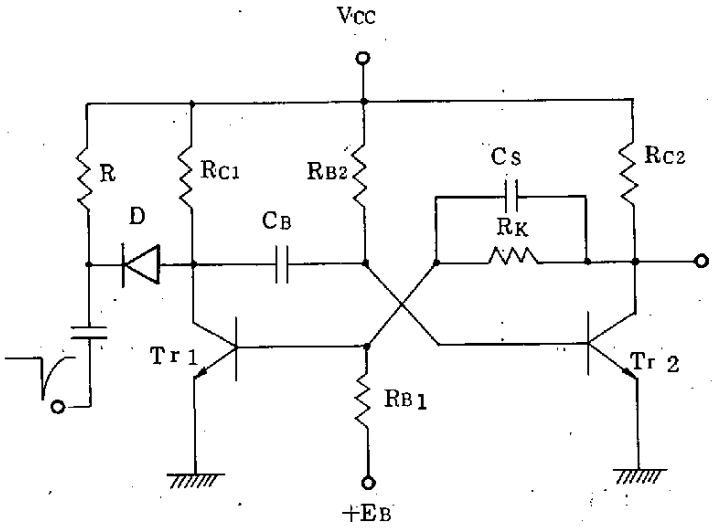


図2－13　単安定マルチバイブレータ回路

$$V_{CE1} = V_{CC} - I_{CB0} \cdot R_{C1} \fallingdotseq V_{CC}$$

$$V_{CE2} = V_{CC} - I_{C2} \cdot R_{C2} \fallingdotseq V_{CE}(sat)$$

になる。

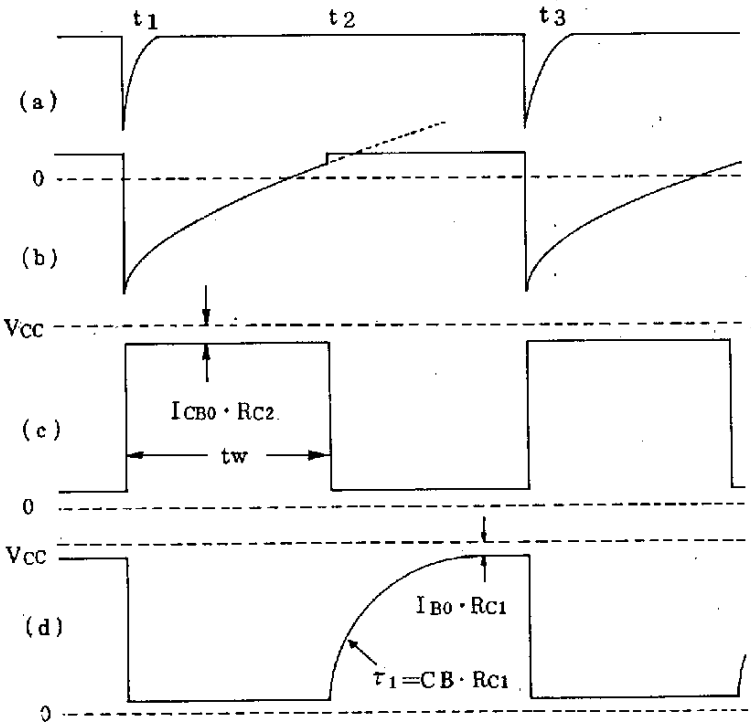
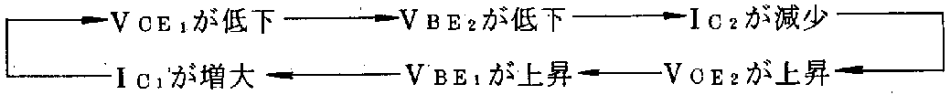


図2－14　波　形　図

さて図2－14（a）のt_1時点において負のトリガパルスを入れるとV_{CE_1}が低下する。そしてこの変化分がC_Bを通してトランジスタ2のベースに入り，V_{BE_2}を低下させる。それによってI_{C_2}が減少し，したがってI_{C_1}が増大し，電圧V_{CE_1}が下がる。動作ループは



となり，この作用でトランジスタ1とトランジスタ2が，それぞれON，OFFとなる。次にt_1からt_2に近づくと，$V_{BE_2} < V_{BE(0)}$　の場合はトランジスタ1がONとなっている。またV_{BE_2}はコンデンサC_Bによって時定数$C_B \cdot R_{B_2}$でV_{CC}に向って上昇する。そして$V_{BE(0)}$より正になると，t_2の時点でトランジスタ2がONになり，I_{C_2}が流れ，元の安定状態にもどることになる。

パルス幅：出力パルス幅twはトランジスタ2がカットオフ状態のときであり，実際の回路では$E_B = V_{CC}$として

$$tw = C_B \cdot R_{B_2} \log_e \left(1 + \frac{V_{CC}}{E_B}\right)$$

で与えられる。

$$tw = C_B \cdot R_{B_2} \log_e 2 \fallingdotseq 0.69\, C_B R_{B_2}$$

となり，取り出したパルス幅の目安が得られる。

回路値の設定：図2－13においてR_Cは

$$R_C = \frac{V_{CC} - V_{CE(sat)}}{I_C} = \frac{V_{CC}}{I_C}$$

$$\because \quad R_C = R_{C_1} = R_{C_2}$$

$$I_C = I_{C_1} = I_{C_2} \quad \text{とする}$$

ふつう　$R_{C_1} = R_{C_2} = R_C$，$I_{C_1} = I_{C_2} = I_C$　とするので

$$R_{B_2} \leqq h_{FE} \cdot R_C$$

で求めることができる。

以下R_KとR_{B1}を求める（トランジスタ1がON）と

$$R_K \geqq R_{B1} \cdot \frac{V_{BE(0)} + V_{CE(sat)}}{E_B - V_{BE(0)} - I_{CB0} \cdot R_{B1}}$$

$$R_K \leqq R_{B1} \cdot \frac{V_{CE(0)} - V_{BE(sat)}}{E_B + V_{BE(sat)} + I_{B1} \cdot R_{B1}}$$

$\because \quad I_{B1} \geqq I_C / h_{FE}, \quad V_{CE(0)} = V_{CC} - I_{CB0} \cdot R_C$

となる。

第3章　論　理　代　数

論理代数はブール代数とも呼ばれている。論理代数で取扱う変数としては，ふつうの代数と違い，2進法の“0”と“1”だけになる。

ディジタル技術を使用するうえで論理代数は必ずしも必要不可欠な知識ではないが，複雑な論理回路の動作を理解したり，あるいは各種の定理を用いて論理式を簡単化して回路を簡素化する場合に有効である。

3.1　ブール代数の基礎定理

ブール代数は基本演算の定義から次の3つの公理と，それから導かれる定理があるが，いずれも扱う値は“1”と“0”の2つだけである。

公理1　否　　定（NOT）

“1”の否定は“0”であり，“0”の否定は“1”である。

公理2　論 理 和（OR）

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 1$

この公理は2つの変数の論理和は，2つの変数のうち1つでも“1”があれば“1”ということを意味している。

公理3　論 理 積（AND）

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

この公理は2つの変数の論理積は2つの変数の両方が“1”のときに限って“1”となることを意味している。

以上の否定，論理和および論理積の3つの基本演算は，デジタル回路の設計に際して，もっとも重要なもので，この公理から次に述べる定理が導かれ，これを駆使してすべての論理を表現することができる。

次にブール代数を実用するうえで重要な基本定理をあげる。

(1) 交換の定理

2つの変数A，Bについて

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

が成立する。

(2) 結合の定理

変数A，B，Cについて

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

が成立する。

(3) 分配の定理

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + B \cdot C = (A + B)\ (A + C)$$

が成立する。とくに下式は通常の代数とは異なり，ブール代数特有のものである。

(4) 同一の定理

変数Aについて

$$A + A + A + \cdots\cdots + A = A$$

$$A \cdot A \cdot A \cdot \cdots\cdots \cdot A = A$$

が成立する。このことはAに“0”と“1”を代入すれば容易にわかる。

(5) 吸収の定理

$$1 + A = 1$$

$$0 \cdot A = 0$$

$$0 + A = A$$

$$1 \cdot A = A$$

$$A + A \cdot B = A \cdot (1 + B) = A$$

$$A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A$$

(6) 否定についての定理

$$A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

$$A = \bar{\bar{A}}$$

下式は多重否定の還元性を表わしている。

(7) ド・モルガン（De Morgan）の定理

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

この式は非常に重要な性質を表わしており，論理回路の簡単化に役立つ。つまり上式の両辺をさらに否定すると

$$A + B = \overline{\bar{A} \cdot \bar{B}}$$

$$A \cdot B = \overline{\bar{A} + \bar{B}}$$

となりこれは論理和と論理積とが完全に独立ではなく，否定を組み合わせると相互交換が可能で，否定と論理和，または否定と論理積のいずれかの組み合わせだけで，すべての論理を構成することができることを示している。

一般的に変数 n に対して

$$\overline{(A_1 + A_2 + \cdots\cdots + A_n)} = \bar{A}_1 \cdot \bar{A}_2 \cdot \cdots\cdots \cdot \bar{A}_n$$

$$\overline{(A_1 \cdot A_2 \cdot \cdots\cdots \cdot A_n)} = \bar{A}_1 + \bar{A}_2 + \cdots\cdots + \bar{A}_n$$

が成り立つ。またよく用いられる定理として

$$A + \bar{A} \cdot B = A + B$$

がある。

3.2 論理回路の簡単化

複雑な組合せで構成された論理回路を，装置全体として，もっとも経済的にできる構成をするため，与えられた論理式をできるだけ簡単にする必要がある。論理式の簡単化の方法として前に述べたブール代数による方法や真理

値表（入力が１と０との状態の全ての組合わせをとって出力の状態を表にあらわしたもの）から論理式を作る方法がある。では真理値表からどのように論理式を導けばよいかを述べる。

表３－１　真理値表

	№	1	2	3	4	5	6	7	8
入力	A	0	0	0	0	1	1	1	1
	B	0	0	1	1	0	0	1	1
	C	0	1	0	1	0	1	0	1
出力	X	0	1	0	1	1	1	0	0

表３－１に３入力の真理値表を示す。これはA，B，Cという３つの入力にそれぞれ“０”または“１”を入力したときに，出力がどうなって欲しいかを示している。図で表わすと図３－１のようになる。そしてこの中味を具体的にどのような回路で構成するかを今から考えてみよう。

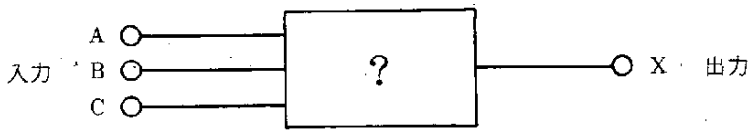


図３－１　入出力の関係

表中，出力Xが“１”を満足する条件を考えてみると，４つあることがわかる。つまり，№２，４，５，６の場合にX＝１となる。ここで表中の№２は$\overline{A}\cdot\overline{B}\cdot C$と表わせ，ほかも同様に“０”にはアルファベットの上に－を付けて表わし，“１”はアルファベットそのままで表わす。すると論理積の和としてXは

$$X=\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot B\cdot C+A\cdot\overline{B}\cdot\overline{C}+A\cdot\overline{B}\cdot C$$

と表わすことができる。

これを回路図で表わすと図３－２のようになる。

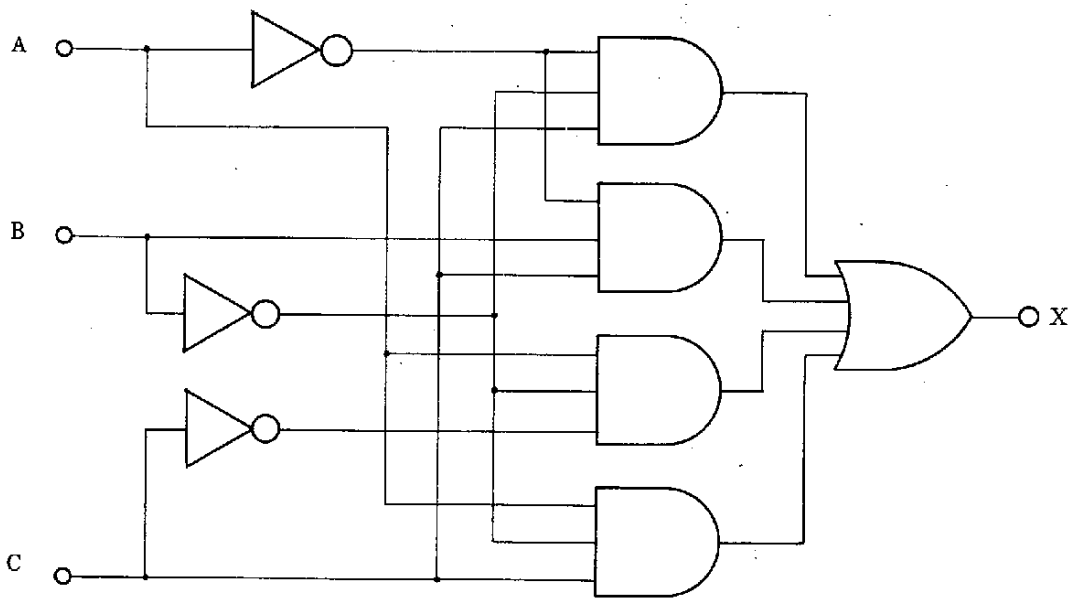


図３－２　真理値表による回路構成

ここで，ブール代数を用いて上式を簡単化してみる。

$$X=\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot B\cdot C+A\cdot\overline{B}\cdot\overline{C}+A\cdot\overline{B}\cdot C$$

$$=\overline{A}\cdot C\cdot(\overline{B}+B)+A\cdot\overline{B}(\overline{C}+C)$$

$$=\overline{A}\cdot C+A\cdot\overline{B}$$

となる。これを図で表わすと図３－３になる。

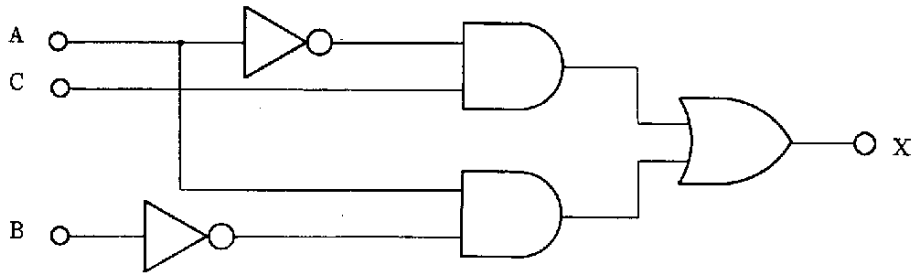


図３－３　簡単化された回路図

このように図３－２が図３－３のように簡単化される。

第4章　デジタルICについて

4.1　論理記号

わが国では，現在でも標準の論理記号がない。またその回路が正論理（一般に5V電源で使用している場合に動作状態が5Vになっているもののことをいう）の受渡しか，負論理（一般に5V電源で使用している場合に動作状態が0Vになっているもののことをいう，正論理と逆の論理）の受けわたしであるかなどは，論理記号だけではわからないことがある。

IC以外の素子を使用しているときは，その回路の特殊性に従ってこのような不便さもやむを得ない場合があるが，TTL，DTLのように素子そのものの機能が統一化され，互換性のあるものがメーカによって供給されることになると，装置メーカ，最終ユーザ共に統一された，しかも論理構成が一目でわかる記号体系が必要となってくる。アメリカで軍関係の電子部門で，各メーカが共通化された電子装置を納入する必要性から検討された結果，標準化された記号が1962年2月MIL－STD－806Bとして制定され，今日に及んでいる。

従来の論理装置はまずブロックダイヤグラムを書き，その各ブロックに対して個別の回路図を作成して全体の設計を行うやりかたが多かったのに対して，ICを使用する場合，ブロックダイヤグラムの論理素子記号にIC番号とピン番号を記入するのみで，ただちに回路図ができあがる。しかもこのMIL記号法を用いれば，正論理・負論理も自動的に行なわれて，まちがうこともないので，この記号法を全面的に採用することが好ましいと思われる。以下，この記号を解説し，以後の構成も全てMILの記号法によることとする。

4.2　記号および形状

記号および形状はANDゲート，ORゲート，フリップフロップなどが定

められ，それぞれの記号の書き方，大きさは任意であるが，各辺の比率がきまっている。内外のＩＣメーカはこれに従ってロジックの定規を作っている。

図４－１に記号の形状を示す。

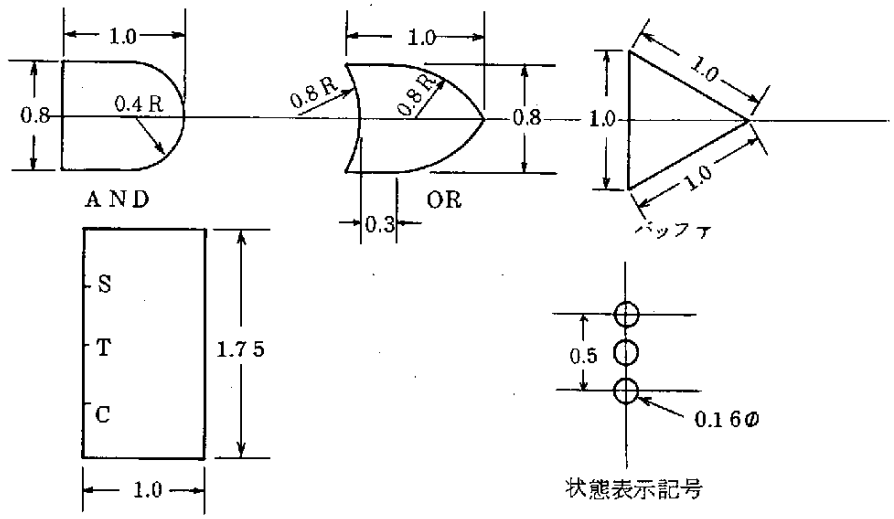
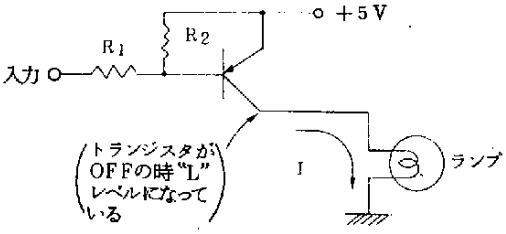


図４－１　記号の形状

4.3　論理レベルと正論理・負論理

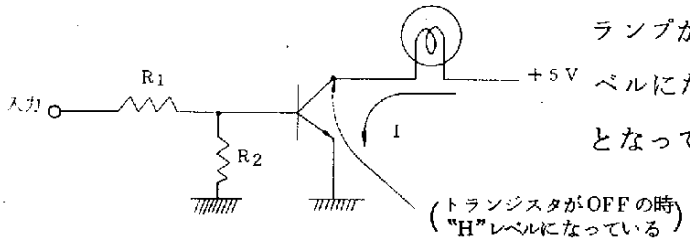
２つのデジタル量に対する定義は，電圧の高低で定める。これによって電流スイッチとの混同による間違いを避けることができる。

〔例１〕　論理レベルが高い時に仕事をさせる時



ランプが点灯するのは入力が“Ｌ”レベルになった時，トランジスタがＯＮとなって，＋５Ｖがランプに印加され電流が流れ点灯する。この時トランジスタの出力レベルは“Ｈ”となる。これが論理レベルが高い時に仕事をさせるということである。

〔例２〕　論理レベルが低い時に仕事をさせる時



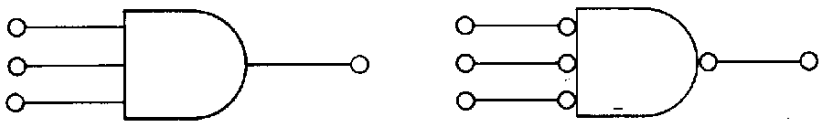
ランプが点灯するのは入力が＋５Ｖレベルになった時，トランジスタがＯＮとなってランプに電流Ｉが流れて点灯

する。この時トランジスタの出力レベルは“L”となる。これが論理レベルが低い時に仕事をさせるということである。

このように電流による仕事を表現する時，電流が流れてランプを点灯させることは同じだが，電圧レベルでいうと“H”レベルで仕事をする場合と“L”レベルで仕事をする場合とがある。MILの方法では高い電圧レベルを“H”とし，低い電圧レベルを“L”として論理“1”および“0”に対応させている。

正論理および負論理は従来の方法では系全体を正論理か，負論理かに統一してしまっていたが，MILでは“H”レベルを論理“1”とし，“L”レベルを論理“0”とするもの（例1の場合）を正論理，逆に“L”レベルを“1”，“H”レベルを“0”とするもの（例2の場合）を負論理として定義づけ，系全体が正論理か負論理かきめず，小円の状態表示記号（4.4で説明する）を併用して回路内でその都度使いわけができるようにしてある。

基本ゲートのANDについていえば，正論理では（図4－2(a)）入力の全



(a) 正 論 理　　(b) 負 論 理

図4－2. ANDの正論理と負論理

てが“H”レベルの時，出力が“H”となる。また負論理では（図4－2(b)）入力の全てが“L”レベルの時，出力が“L”となる。

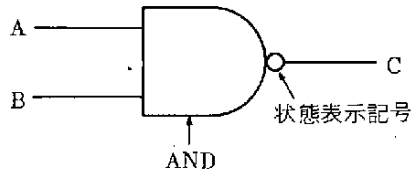
4.4 状態表示記号

図4－1の小円が状態表示記号で入力または出力の1つあるいはいくつかにつけて用いる。正論理では状態表示記号はつけないが，この記号がついた入力では“L”の状態で回路が“ACTIV”すなわち能動となるときめている。

正論理では“H”レベルでANDおよびORゲートが動作するから，端的にいえば，この記号を入出力に付けることにより，回路が負論理の受けわたしに変ったことを表示すると考えてよいことになる。

TTLの基本論理素子は，ほとんどがインバータ（否定）を内蔵したAND・NOTすなわちNANDまたは，OR・NOTすなわちNORであるのでこれを例にして説明する。

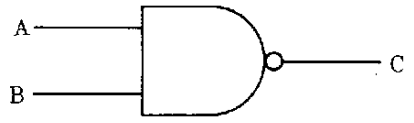
NANDの状態を表にすると図4－3のようになる。



A	B	C
H	H	L
H	L	H
L	H	H
L	L	H

図4－3. NAND記号および状態表

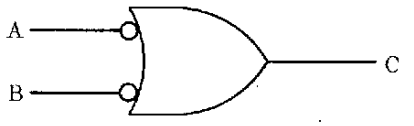
この図のとおりNANDとして取扱うと，入力は状態表示記号が付かず，正論理の扱いだから，真理値表は図4－4のとおりとなり，AND・NOT動



A	B	C
1	1	0
1	0	1
0	1	1
0	0	1

図4－4. NANDと真理値表

作を行ない，出力は“L”で“ACTIV”となる。この状態を，もし“H”を“0”論理として考えると図4－5の真理値表となる。



A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

図4－5. ORと真理値表

この真理値表より，A，Bのどちらかまたは両方に“1”が入力されると出力Cには“0”が出力されることになって，動作としてはORの働きとなる。いい変えるとAまたはBのどちらか，あるいは両方に“L”を与えると出力Cは"H"になる。よって信号の電気的動作は同じでも回路の表現の方法を変える事により信号のアクティブ状態を表現することができる。

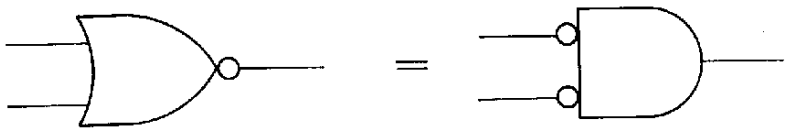


図4－6. NORと電気的動作が同じで表現方法を変えた例

NOR素子もNAND素子と同様に図4－6で示すように書き表わすことができる。こうした定義のもとでは，接続された回路の入出力で，どちらか一方のみに状態表示記号がついているときは，回路の論理としては間違っていることになる。例えば図4－7で信号A，B，C，DのANDを作ろうとするとき，入力NANDを3個使用して(a)の回路図を組んだとする。しかし(a)の回路ではA，BおよびC，DのAND出力には状態表示が付いているので“L”で“ACTIVE”となり，このままNANDに接続するとAND動作せず（A・B・C・D＝Eとはならない），(b)の回路となって論理的に

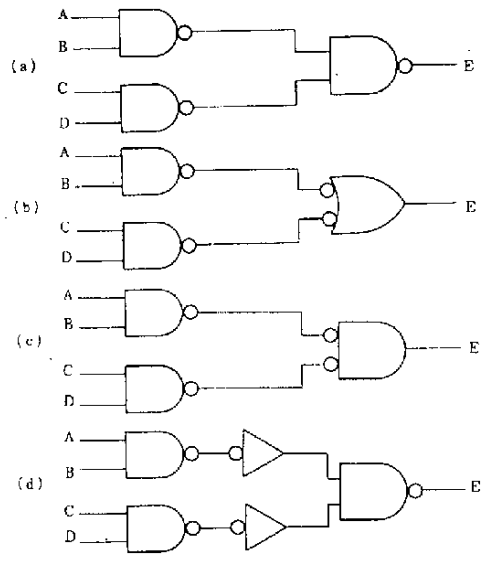


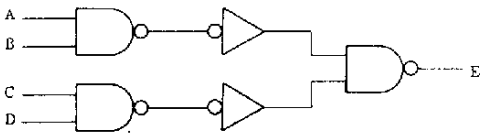
図4－7. 接続の例

$$(A \cdot B) + (C \cdot D) = E$$

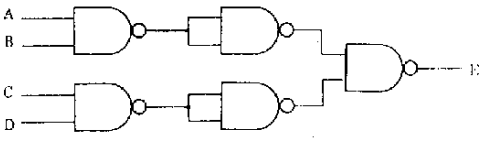
となってしまう。最終出力が正論理であるか，負論理であるかを別にして，もしA・B・C・DのANDをとろうとするなら，NORを1個使用して(c)の回路とするか，(d)のようにインバータを入れて位相反転をし，正論理にしてNANDを使用しなければ，論理的にあやまちをおかす。状態表示記号を使用すればこのような間違いを避けることができる。

4.5　バッファ増幅器

図4－1のバッファの記号は電圧・電流のレベル変換器，パルス増幅器，その他の増幅器を示す記号で，専用のインバータまたはNAND，NOR を用いても，同様の動作をさせることができる。通常は波形の整形の目的で使用される記号であるが，その他リレー，ランプ駆動回路なども含む。NANDおよびNORはインバータとして図4－8のように使うこともできるが，こ



(a)　$E = \overline{A \cdot B \cdot C \cdot D}$　　正しい表記



(b)　表記としては正しくないが実際のつなぎ方

図4－8.　NANDのインバータとしての使用法

の場合はNANDまたはNORをバッファとして書き換えておき，状態表示記号を前または後ろにつける。NANDまたはNORはそのときの接続状態によって，状態表示記号の小円を入力側または出力側につけて用いる。インバータとしての動作は図4－9の表示があるが，どれも同じことを表わして

いる。

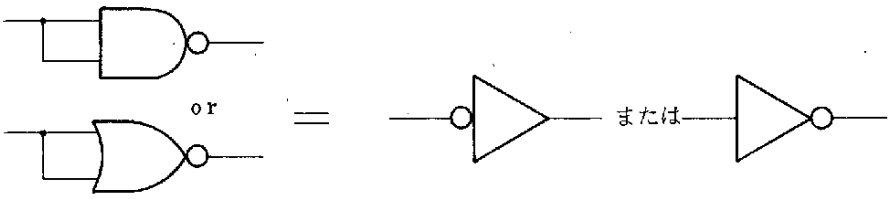


図４－９．　NANDおよびNORのインバータとしての動作

4.6　フリップフロップ

フリップフロップは論理的に安定な２つの状態が存在して，外部からの指令信号によって，このうちのどちらかの状態を保持する一種の記憶回路である。フリップフロップはＪ－Ｋ，Ｔ，Ｄ，Ｒ－Ｓなどの数種があって，それぞれ独自な使用法があるが，ここでは基本的なＲ－Ｓ－Ｔフリップフロップについて述べる。

入力信号として

セット（ｓｅｔ）……Ｓ

クリアまたはリセット（Ｃｌｅａｒ，Ｒｅｓｅｔ）……Ｃ，Ｒ

クロックまたはトリガ（Ｃｌｏｃｋ，Ｔｒｉｇｇｅｒ）……Ｔ

があり，出力としてＱ，$\bar{Q}$の２つを持っている。Ｑを論理の“１”に，$\bar{Q}$を論理の“０”に対応させる。Ｑが“Ｌ”の時にはこのフリップフロップには“０”が記憶されていることになる。この時$\bar{Q}$は必ず“Ｈ”となっている。このようにＱと$\bar{Q}$は互いにいつも論理が逆になっている。また，Ｑが“Ｈ”の時にはこのフリップフロップには“１”が記憶されていることになる。

Ｒ－Ｓ－Ｔフリップフロップの記号を図４－１０に示す。

Ｓ入力に正論理信号“１”が入った時，フリップフロップの今までの状態に関係なく，このフリップフロップの内容は“１”となる。同様にＲに“1”が入った時，フリップフロップの内容は“０”となる。図面の書きかたとして次の約束が定められている。すなわち

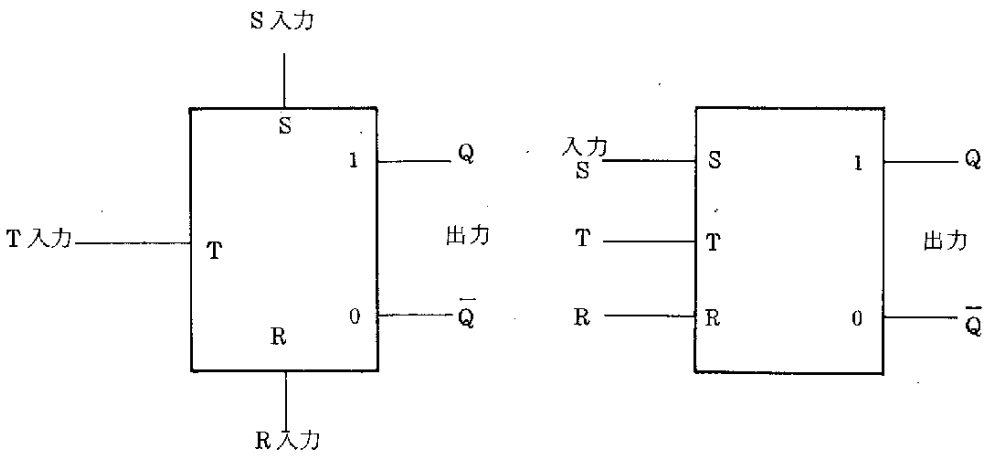


図4－10. R－S－Tフリップフロップの記号

① T入力を用いない時はTの入力を省略する。

② S入力は“1”出力の近くに書く。

③ R入力は“0”出力の近くに書く。

第５章　ＴＴＬの種類と使い方の例

ＴＴＬには単機能のゲート，フリップフロップなどを１個または数個集積したＳＳＩ（Small Scale Integration），かなり複合機能をもつＭＳＩ（Medium Scale Integration），大きな集積度をもつＬＳＩ（Large scale Integration）がある。厳密にはこれらの区別ははっきりしていない。なお，本書ではＴＴＬの系と呼び名をＴＩ社の７４シリーズを代表として取り上げることとする。他社のものも全く同様に応用することができる。ピン番号は上面から見たものである。

5.1　SN 7400

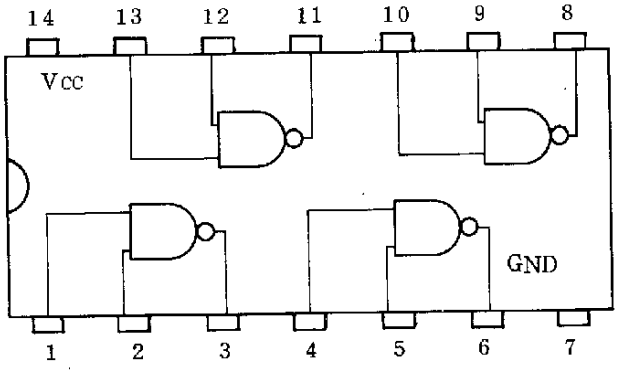
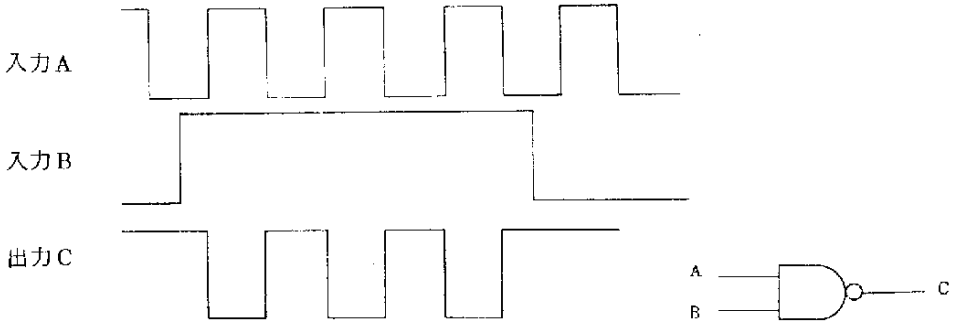


図５－１．　SN 7400 ロジック図

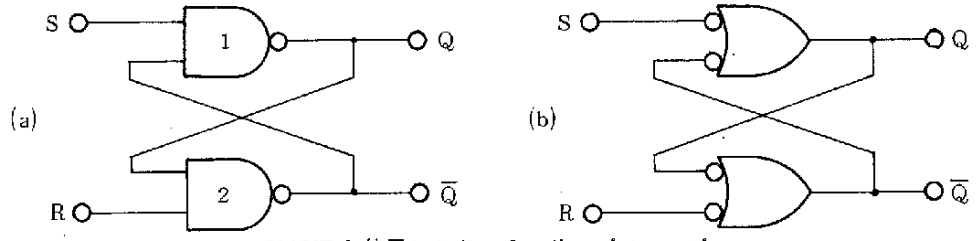
最も基本的なゲートの１つで，組合わせによって数多くの機能をもたせることができる。ＴＴＬでは，信号の受けわたしは直流結合で行われる。したがって時間に関係なく，入出力は直流的な相関関係で定まる。

〔例 1.〕　ゲート回路として使用する場合



NANDのA入力にクロックを入力しておく。B入力に“H”信号を入力した時だけ出力Cに入力Aの信号の反転したクロックが出力される。

〔例2〕 R－Sフリップフロップとして使用する場合



NANDを使用したR－Sフリップフロップ

NANDゲートを2個使用したR－Sフリップフロップを上図に示す。但し，入力は“L”の状態で規定されるので，書き方としては(a)よりも(b)の方が正しい。

S，R入力とも“H”，“H”の場合，このフリップフロップはある状態を保っている。今S入力が“L”になるとNAND 1は必ず出力が“H”となり，NAND 2はR入力が“H”だから2入力共“H”となり，出力すなわち$\overline{Q}$が“L”となる。ここでS入力が“H”となっても，$\overline{Q}$が“L”になっているからNANDの1の入力の一方に“L”入力が入ってQ出力は“H”の状態を保っている。R入力に“L”信号が入ると，これと反対の動作をして$\overline{Q}$が“H”となり，そのままの状態を保持する。S，R入力の方が同時に“L”となった時は，Q，$\overline{Q}$は両方とも“H”となる。そしてS，Rのどちらかが“H”になった時点で，一方が“H”，他方が“L”となる。

5.2 SN 7476

J－Kフリップフロップは，一般的にはJ入力とK入力の2入力をもち，J入力に“H”が加わりCKにパルスが加わった時の“H”から“L”へ変化した時（一般に立下がった時という）出力Qが“H”，出力$\overline{Q}$が“L”となる。またK入力に“H”が加わりCKの立下がりで$\overline{Q}$が“H”となりQが

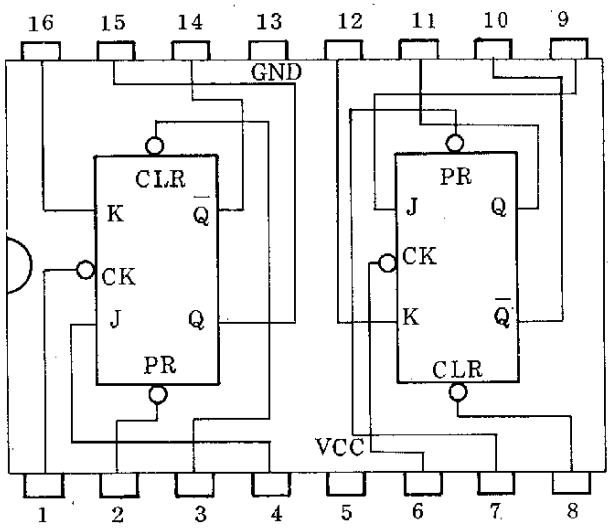
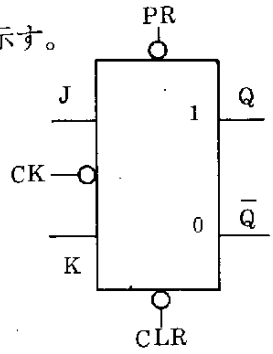


図５－２．　SN 7476

“L”となる。またJ入力，K入力共に“H”の時にCKが立下がるとQと$\overline{Q}$はCKがくる前の状態と逆の状態となる。すなわちQが“H”，$\overline{Q}$が“L”の時はQが“L”，$\overline{Q}$が“H”となる。J－Kフリップフロップの真理値表を図５－３に示す。



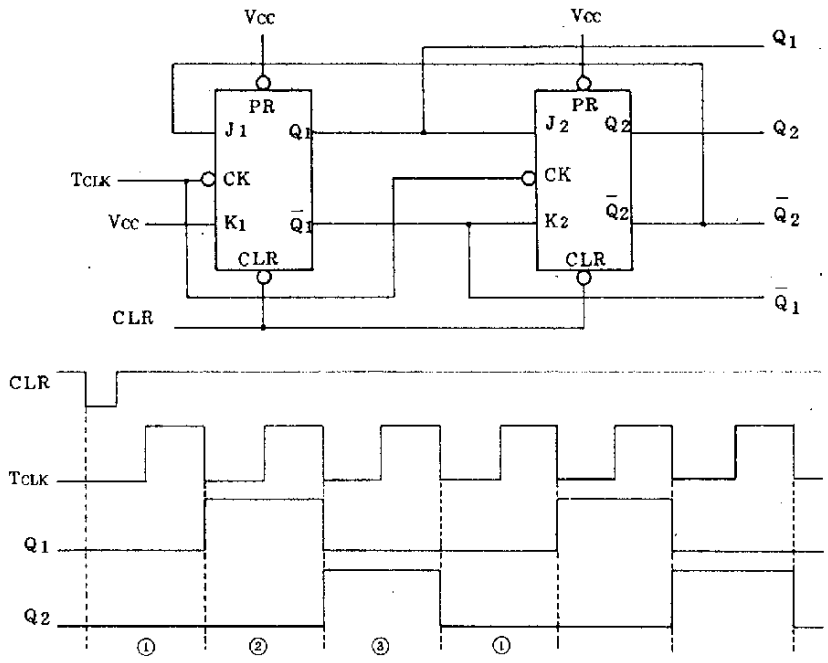
J	K	出力
0	0	Qn
0	1	0
1	0	1
1	1	$\overline{Qn}$

図５－３．　フリップフロップの真理値表

（ただし出力の変化はCKの立下りの時点でのみおこる）

SN 7476 にはPR（プリセット）機能とCLR（クリア）機能が付いている。PRの機能は状態表示記号が付いているので正論理で“L”の時，Q出力は無条件で“H”になる。CLR機能は同じく“L”の時，Q出力は無条件で“L”になる。また，PR，CLRの両方ともに“L”，“L”の信号が同時に入力されると，今までの状態をそのまま保っている。

〔例1〕　SN 7476を使用した3進カウンタ



電源を入れたときQ₁とQ₂のJ－Kフリップフロップは Qが“H”か“L”か不定なので，動作開始前に必ずCLR端子に負のパルスを両方のJ－Kフリップフロップに入力する。するとQ₁，Q₂のJKフリップフロップはともにリセットがかかり，Q₁，Q₂が“L”の状態になり，クロック入力の入力待ちとなる。ここでCLKの1番目の立下りがきた時の各入力状態を考えると，J₁入力は$\overline{Q}_2$と接続されているので“H”状態である。K₁入力はV_{CC}に接続されているので常に“H”状態である。

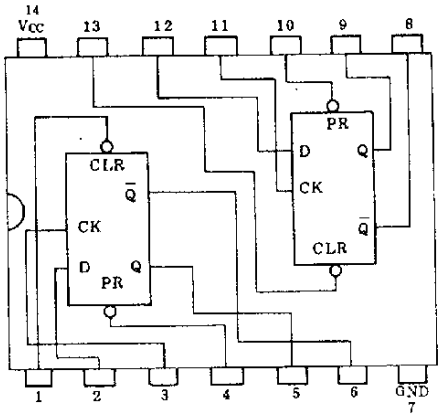
J₂入力はQ₁と接続されているので，Q₁は“L”となり，したがってJ₂入力は“L”となる。また，K₂入力は$\overline{Q}_1$と接続されているので入力は“H”となっている。この状態でCLKが立下がるとQ₁の出力は反転するが，Q₂の出力は前と同じ状態になる。（タイミングチャート②の状態）。

次に2番目のクロックの立下りの時はJ₁は“H”，K₁は“H”，J₂はは“H”，K₂は“L”となりQ₁，Q₂共に反転（タイミングチャート③

の状態）する。次に3番目のクロックの立下りの時の各入力状態は，J_1：“L”，K_1：“H”，J_2：“L”，K_2：“H”となり，Q_1はそのままの状態だがQ_2は反転する。そして再び①の状態になり3進カウンタとして動作する。

5.3 SN 7474

図5－4に示した真理値表のとおり，出力条件はクロック入力時のDの状



tn	tn＋1	
入力D	出力Q	出力$\bar{Q}$
0	0	1
1	1	0

図5－4. SN 7474と真理値表

態で一義的に定まるので，J－Kフリップフロップに比べ入力が1つですみ，記憶素子として使用すると便利である。このときの動作はクロック入力の立上り時間に関係せず，クロック入力がスレッショルドレベルをこえたとき，フリップフロップの内容が規定される。SN 7474にもクロックに関係なく，$\bar{Q}$，Qの状態をきめるクリア・プリセット入力がついているので，Dタイプのフリップフロップと R－S フリップフロップの組合せとして使用することができる。1つのICパッケージの中に4個のDフリップフロップが構成されている例としてSN 74175がある。このICはクロック端子とクリア端子が内部で全てに接続されていて，入力信号端子1つで4個とも同時にクリアしたデータの取込も同時にできるので，データの一時保持回路としてよく使用される。回路図を図5－5に示す。

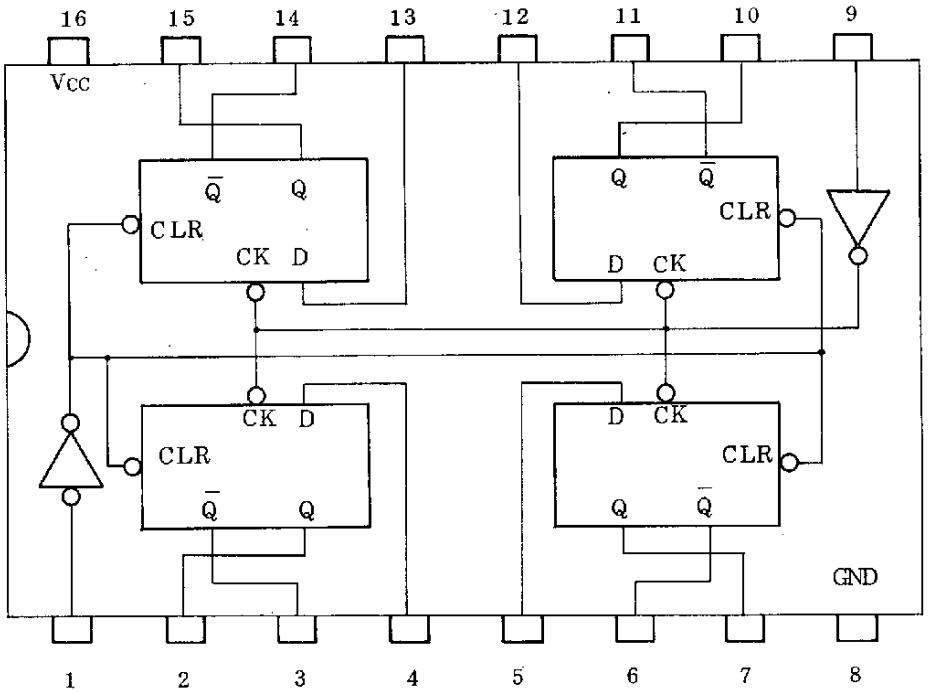


図5－5. SN 74175

以上，TTL のICのほんの一部について説明したが，実際には300種以上もあって全てを本項で説明することはできないが，それらについてはTTL ハンドブックを参照されたい。

第6章　MOS　ICについて

6.1　MOSの仕様と特徴

現在MOS　ICとしては表6－1に示すごとくPチャネルMOS　FETのみで構成されるもの，NチャネルMOS　FETのみで構成されるもの，およびPチャネルMOS　FETとNチャネルMOS　FETとを相補形に組み合せて構成されたものと，大きく分類して3種のものが市場に出ている。

表6－1.　MOS　IC　の分類

MOS IC	PチャネルMOS　IC	高スレッショルド
		低スレッショルド
	NチャネルMOS　IC	
	コンプリメンタリMOS IC（CMOS）	

これらのMOS　ICの特徴としては

① 入力抵抗がＴＴＬと比べて数桁高い。このためＴＴＬから駆動する際の入力電流は無視し得る。しかし入力容量は比較的大きいものがあり，入力インピーダンスとしてはそれほど高くない。したがってこの静電容量を高速にドライブするために大きなピーク電流を必要とし，周波数が高くなれば当然駆動するＩＣのＡＣ電力が問題となる。

② 出力インピーダンスはＴＴＬと比べて一桁位高い。

このためにＭＯＳからＴＴＬを駆動するとき，多数個並列に駆動することは難しくなる。

③ 動作速度はＴＴＬと比べると遅い。また，最低動作速度が規定されているものがある。このためにシステムによっては動作速度のマッチングを取る必要がある。

④　低電力消費である。

⑤　負荷として同一構造のMOS　FETを使用できるので，すべて同一のMOS　FETで回路を構成でき，製造工程も比較的簡単であるから，④とも関連して大規模集積化が容易である。

等があげられる。

また，PチャネルおよびNチャネルMOS　ICの出力回路形式としては，図6－1に示すように出力がオープンドレインのもの，ダブルエンデッドのもの，プッシュプルのものがある。図6－1(a)はオープンドレインの出力回路で，この出力のMOS　FETのドレイン・ソース間抵抗は，オフ時にはほぼ無限大となり，オン時には，大体1 kΩ以下となる。このオン時の抵抗は各デバイスごとに抵抗値または，出力電圧と電流で規定される。(b)はダブルエンデッド出力回路形式を示し，(a)のオープンドレイン形のものに負荷

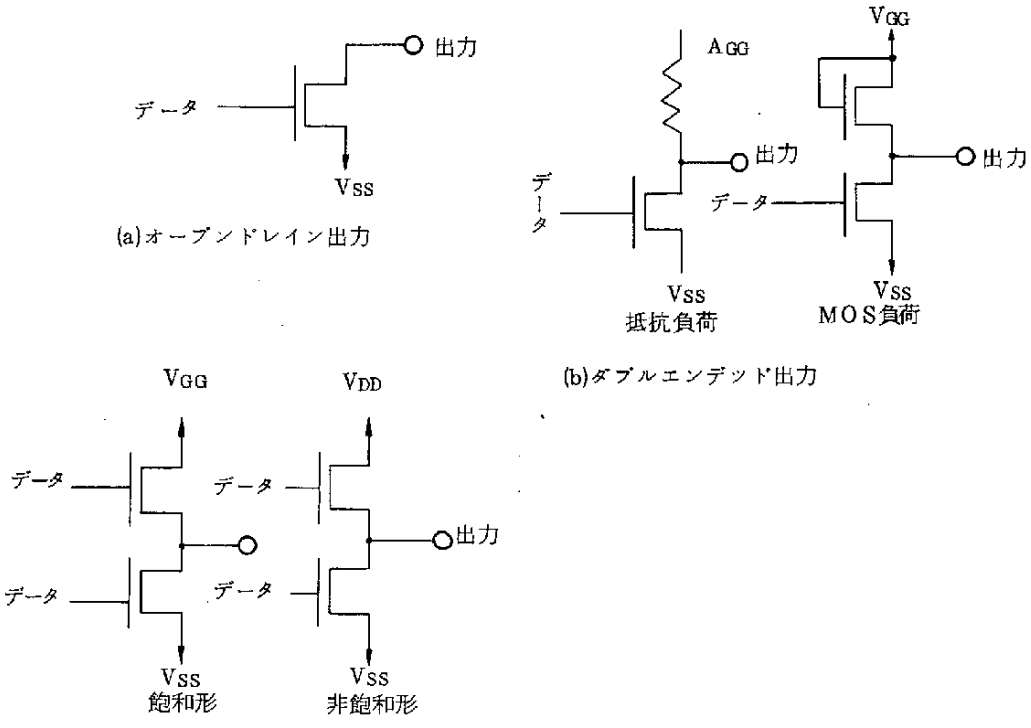


図6－1　各種出力回路形式

として抵抗値を追加したもので通常はMOS FETが受動素子として負荷に利用される。(c)はプッシュプル出力回路形式を示しV_{SS}－V_{GG}間をスイングする飽和形のものと，V_{SS}－V_{DD}間をスイングする非飽和形のものがある。

6.2 PチャネルMOS IC

表6－2に代表的なPチャネル高スレッショルドMOS IC，低スレッショルドMOS ICおよびTTLの供給電源電圧と論理レベルを示す。

表6－2. PチャネルMOS ICの電源電圧と出力レベル(V_{SS}基準)

	供給電圧(V)				論理レベル(V)	
	V_{CC}	V_{SS}	V_{DD}	V_{GG}	低レベル	高レベル
高スレッショルドMOS IC	－	0	−12	-12	−24～−9	−3 ～0
低スレッショルドMOS IC	－	0	− 5	-17	−17～−4.2	−1.5～0
TTL	5	－	－	－	0～ 0.4	2.4～5

この表からもわかるように，TTLでは供給電圧，出力レベル共に正のレベルにあるのに，MOSではサブストレート電圧(V_{SS})に対して供給電圧，出力レベル共に負のレベルであり，このままではTTLとMOSの接続を行うのに不都合である。MOS ICでは相互の電圧関係さえ変えなければ，基準はV_{SS}にとってもV_{DD}にとってもMOS ICとしての動作にはさしつかえないから，表6－2をV_{DD}を基準にしたレベルに変換すると表6－3のようになる。

表6－3. PチャネルMOS ICの電源電圧と出力レベル(V_{DD}基準)

	供給電圧(V)				論理レベル(V)	
	V_{CC}	V_{SS}	V_{DD}	V_{GG}	低レベル	高レベル
高スレッショルドMOS IC	－	12	0	−12	−12～ 3	9～12
低スレッショルドMOS IC	－	5	0	−12	−12～0.8	3.5～ 5
TTL	5	－	－	－	0～0.4	2.4～ 5

表６－３からもわかるように，低スレッショルドＭＯＳでは論理レベルはほとんどＴＴＬレベル近くになり，また電源もV_{SS} －５ＶはV_{CC}と共通に使える等の利点もあり，一般にＭＯＳではＴＴＬとのインターフェースの容易さからV_{DD}を基準とした電源供給法がとられるのが普通である。

6.3　ＮチャネルMOS　ＩＣ

ＮチャネルＭＯＳではキャリアとして電子を用いるので移動度がホールより２～３倍高く，ＰチャネルＭＯＳよりスピードが速くなるという予想があったが，製造上，エンハンストメントモードのMOS　FETを安定に製造するのが難かしく，MOS　ICとしての開発は遅れていた。しかし最近ようやく市場に出廻り始めている。ＰチャネルMOS　ＩＣとは反対にＮチャネルＭＯＳ　ＩＣでは，基板に対して電源電圧は正となり，TTLとのインターフェースも楽になると予想される。

表６－４．インターフェースに必要なTMS 4030のデータ

絶対最大定格

電源電圧V_{DD}，V_{CC}	−0.3 ～ 20V
入力電圧(クロックを含む)	−0.3 ～ 20V
動作温度範囲	0℃ ～ 70℃
保存温度範囲	−55℃ ～ 150℃

推奨動作条件における仕様			最小		最大	条　件
供給電圧〔V〕		V_{DD}	11.4	12.0	12.6	
		V_{SS}		0.0		
		V_{BB}	−2.7	−3.0	−3.3	
		V_{CC}	4.75	5	10.0	
入力電圧〔V〕	データ	V_{IH}	2.2		V_{CC}	
		V_{IL}	−0.6		0.6	
	クロック	V_{IH}	V_{DD}−0.6	V_{DD}	V_{DD} +0.6	
		V_{IL}	−0.6	0.0	0.6	
出力電圧〔V〕		V_{OH}	2.4		V_{CC}	Iout = −20mA
		V_{OL}	V_{SS}		V_{SS} +0.4	Iout = 3.2mA
入力容量〔PF〕		データ		4	6	チップセレクト
		アドレス		5	7	W/R DATA
		クロック		23	27	

表６－４に一例としてTMS 4030の仕様を示す。表からもわかる通り，データ入力レベルはほぼTTLレベルであり，クロック入力レベルは低レベルで0.6V以下，高レベルで11.4V以上となっていて，TTLとのインターフェースも容易である。また出力はプッシュプルでTTLとコンパチブルである。

6.4 コンプリメンタリMOS IC

CMOSはPチャネルFETとNチャネルMOS FETとをコンプリメンタリ接続したものを基本構造として，図６－２のインバータでは，入力が低レ

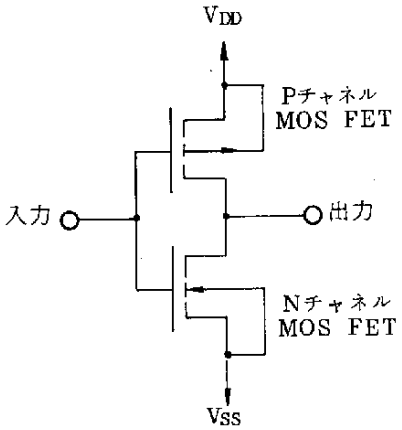


図６－２．コンプリメンタリMOS ICインバータ

ベルの時は上側のPチャネルFETがオンに，高レベルのときは下側のNチャネルFETがオンになり，どちらかのFETが必ずオフとなっているため，定常状態での電力消費は標準で10mW程度であり，過渡状態では必ず一方のFETを通して負荷の容量を充放電するため，速いスイッチング速度が得られる。また，図６－３の入出力特性にみるように，動作電源電圧V_{DD}範囲が広く，V_{DD}が増大すると論理のスレッショルドレベルも増大し，この値はV_{DD}／２になる。電源電圧に対する相対ノイズマージンも大きくとれ理想的であるが，そのためにはPチャネルFETとNチャネルFETのスレッショルドレベルの絶対値が同じで特性も同じである必要がある。

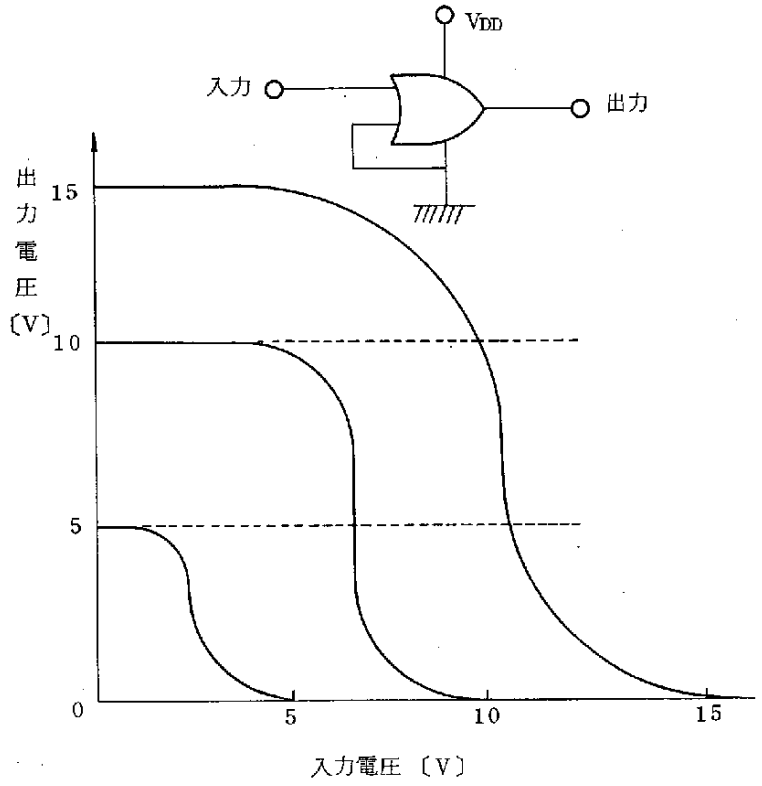


図6－3. TF 4001A の入出力特性

以上MOSの種類と特徴について述べてきたが，ICを使ったロジックシステムではなるべくディスクリート部品を使用しないことがシステムの信頼性を向上させることになるので，MOSのインターフェースに際してもなるべくディスクリート部品を使用しないことが望ましい。例外的に，ディスクリート部品を使用しなければTTLとインターフェースできないものもある。

第7章　MOSとTTLとの接続

7.1　PチャネルMOSからのインターフェース

PチャネルおよびNチャネルMOSからTTLへインターフェースする場合は，MOSの出力回路形式によって若干その方法が異ってくる。またPチャネルでは出力回路形式に加えて，高スレッショルドMOSか低スレッショルドMOSかでインターフェースが異なってくる。

7.1.1　Pチャネル高V_TMOSからTTLへのインターフェース

① オープンドレイン出力の場合

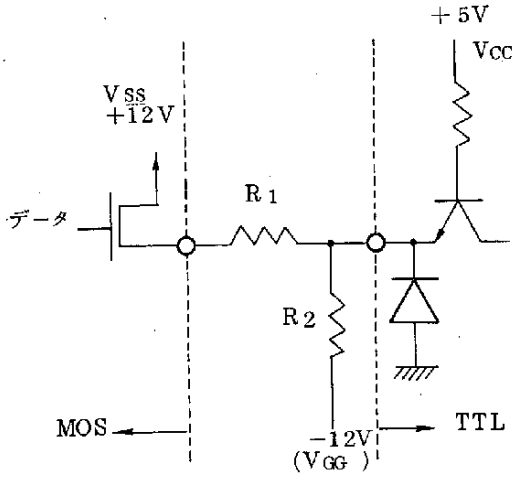


図7－1．オープンドレインPチャネル高V_TMOS→TTLインターフェース

図7－1はPチャネル高V_T MOSのオープンドレイン出力から，TTLへのインターフェースの例を示したものである。図中R_1，R_2はMOS ICの出力が高レベルのとき，TTLの入力が最大許容入力電圧（V_{IH}（MAX）＝5.5V）をこえないように抵抗分圧したもので，MOS出力FETがオフのとき，TTLの入力は低レベルとなり，V_{IL}＝0.4V　I_{IL}＝1.6mAを満足し，しかもTTL入力のクランプダイオードに流れる電流が12mAを越えない範囲でR_2を設定し，次にMOS出力FETがオンのときTTLの入力条件V_{IH}＝2.4V，I_{IH}＝40μAを満足し，同時にTTL入力電圧が5.5Vを越えないようにR_1を決定する。

② ダブルエンデッド出力の場合

図7－2はダブルエンデッド出力の場合のインターフェースの方法であるが，基本的にはオープンドレインの場合と同じでMOSの出力FETがオンになったときにTTLの入力条件$2.4V \leq V_{IH} \leq 5.5V$，$I_{IH}$

$= 40\mu A$（$V_{IH} = 2.4V$のとき）を満足し，MOS出力FETがオフのときには$V_{IL} \leqq 0.4V$，$I_{IL} = 1.6mA$および$I_{IL} \leqq 12mA$（$V_{IL} \leqq 0$のとき）を満足するように決定すれば良い。

③ プッシュプル出力の場合

図7－3にプッシュプル出力のMOSからのインターフェースの方法を示す。プッシュプル出力の場合，出力レベルが高レベルの時は上側，低レベルの時は下側のMOS FETがオンとなり，定常状態では必ず一方のFETはオフとなっている。オン状態の出力抵抗はカタログに規定されているから，R_1，R_2はTTLの入力条件を満足するように決めれば良い。

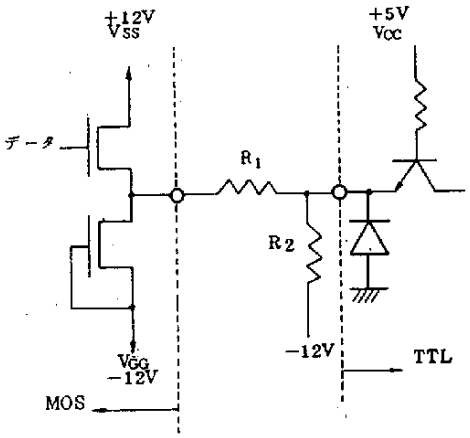


図7－2. ダブルエンデッドPチャネル高V_TMOS→TTLインターフェース

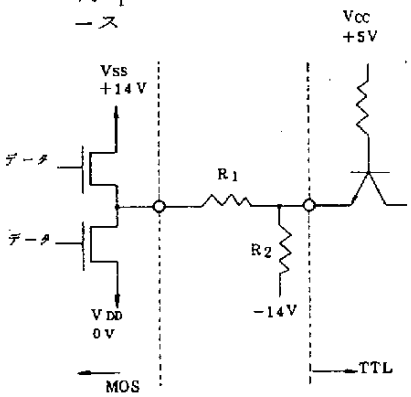


図7－3. プッシュプル出力Pチャネル高V_TMOS→TTLインターフェース

7.1.2 Pチャネル低V_T MOSからTTLへのインターフェース

① オープンドレイン出力の場合

低スレッショルドのPチャネルMOSの場合，V_{DD}を基準に電源を供給すれば，出力レベルがTTLの許容最大入力（V_{IH}（MAX）$= 5.5V$）を越えることはないので，Pチャネル高スレッショルドMOSからTTLへのインターフェースの場合の直列抵抗R_1が省略でき，TTLの低レベル入力条件（$V_{IL} \leqq 0.4V$，$I_{IL} = 1.6mA$）を満足させるために，プルダウン抵抗R_2のみを使用すれば良い。これを図7－4に示す。

このR_2の値は次のようになる。

$$R_2 \leqq \frac{(12+0.4)\ V}{1.6\,mA} \leqq 7.75\,[k\Omega]$$

なお，$R_2 = 7.5\,k\Omega$ を接続したときに，TTLの高レベルでの入力条件（$V_{IH} \leqq 2.4V$, $I_{IH} = 400\,\mu A$）を満足するように出力MOS FETのオン時の抵抗は，1.4 kΩ以下に作られている。

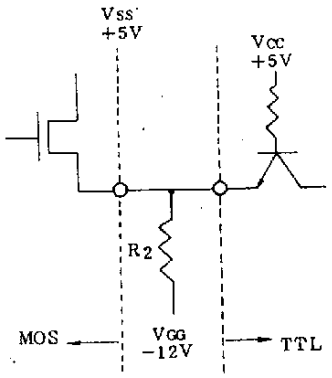


図7－4. オープンドレインPチャネル V_T MOS→TTLインターフェース

② ダブルエンデッド出力の場合

Pチャネル低V_TMOS ICで出力がダブルエンデッド形式の場合は図7－4のプルダウン抵抗R_2がMOS IC内部に設けられた形になり，通常TTLの入力条件が満足されるように出力MOS FETおよび負荷MOS FETが設計されているのでほとんどの場合，直接接続が可能である。

③ プッシュプル出力の場合

Pチャネル低V_TMOS ICで出力回路形式がプッシュプルの場合にも，ダブルエンデッド出力回路形式の場合と同様に，TTLの入力条件を満足するように，出力回路が設計されているので，TTLと直接接続可能である。またプッシュプル出力回路形式の方がパルスの立上り，立下り等の点でも有利であり，現在Pチャネル低V_T MOS ICの出力回路形式として最も一般的なものである。

7.2 NチャネルMOSからのインターフェース

NチャネルMOS ICのTMS 4030からTTLへインターフェースした例を図7－5に示す。TMS 4030の出力レベルは高レベル，低レベルでそれぞれ$V_{OH} \leqq 2.4V$，$I_{OH} = -2mA$

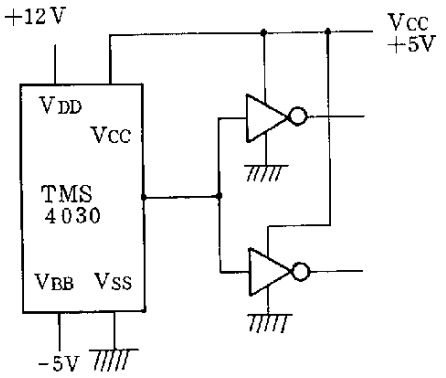


図7－5. NチャネルMOS IC→TTL

$V_{OL} \leqq 0.4V$, $I_{OL} = 3.2mA$ と規定されているから，SN 74 シリーズのTTL 2 個が直接駆動できる。

7.3 CMOSとTTLとのインターフェース

図7－6にインターフェースを考慮したCMOS ICの入力および出力特性を示す。

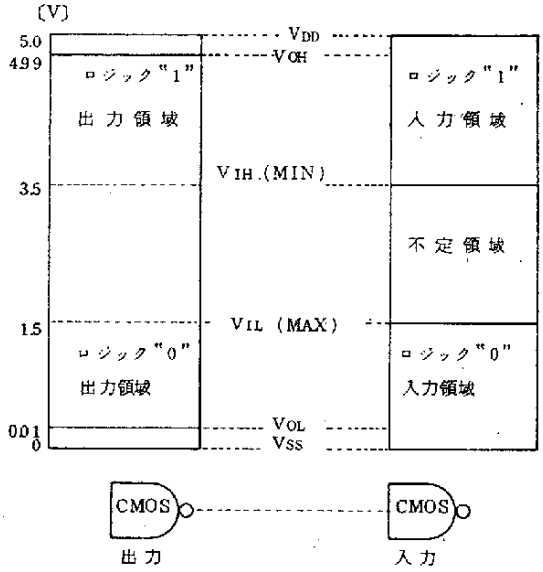


図7－6 インターフェースを考慮した入出力特性

CMOSは他の飽和形論理回路（TTL／DTL）とは特性が異なる。例えばCMOS ICをスイッチングさせるには電源電圧の半分の電圧レベルでスイッチするように設計されている。しかしながら，TTL／DTL素子はそうではなく，スレッショルドレベルが1.5Vである。よってICとのインターフェースに当って，論理振幅，出力駆動能力，直流入力電流，雑音余裕度，および動作速度などに注意して設計を進める必要がある。

7.3.1 TTLからCMOS ICへのインターフェース

TTLからCMOSをドライブする場合は，TTLの出力ドライブ能力，スイッチングレベルおよびレシーバ（CMOS）の入力特性を考慮してインターフェースをする。CMOSの入力特性の電流はI_{IL}，I_{IH}共非常に小さく10 pAである。また電圧は$V_{IH} = 3.5V$以上，$V_{IL} = 1.5V$以下であ

ればよい。したがってTTLの出力に同じ雑音余裕度を得るためには論理"0"は1.5V以下，論理"1"は3.5V以上なければならない。オープンコレクタ方式を採用しているロジックでは，プルアップ抵抗が必要になる。図7－7の方式でCMOSをドライブするときのプルアップ抵抗の設定方法をのべる。

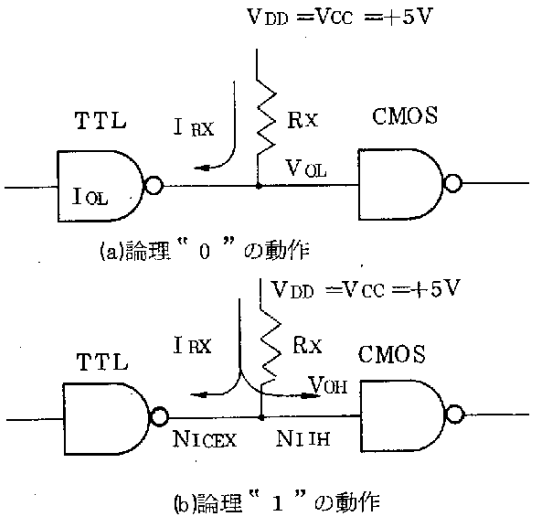


図7－7. TTLからCMOSへのインターフェース例

プルアップ抵抗の選択には，ファンアウト，論理"0"の時許容しうる最大コレクタ電流（I_{OL}（MAX）），論理"1"における，リーケージ（I_{CEX}），消費電力，電源電圧および伝搬遅延時間に対する考慮が必要である。

プルアップ抵抗に関する算出式は次式で表わす。

$$R_X\ (\mathrm{MIN}) = \frac{V_{DD} - V_{OL}\ (\mathrm{MAX})}{I_{OL}}$$

$$R_X\ (\mathrm{MAX}) = \frac{V_{CC}\ (\mathrm{MIN}) - V_{IH}\ (\mathrm{MIN})}{(N)\ I_{CEX}\ (\mathrm{MAX})}$$

N ：ドライバー側の数

7.3.2 CMOSからTTLへのインターフェース

この場合，ドライバー側（CMOS）のランク電流（I_{OL}）を考慮しなければならない。スタンダードTTLの入力特性の最悪条件は，論理"1"に対して$I_{IH} \leq 40\,\mu A$，$V_{IH} = 2.4V$，論理"0"に対して$I_{IL} \leq -1.6mA$，$V_{IL} = 0.4V$である。論理"1"に対してはだいたい十分に取れるが，論理"0"に対しては特殊なもの以外は直接接続はできない。CMOSで直

接ドライブできるTTLファミリーに74Lおよびショットキーダイオードを導入した74LSがある。

以上のようにインターフェース回路を設計する場合，いろいろ検討項目はあるが，いずれにしても駆動側の出力特性と，レシーバ側の入力特性を考慮して設計を進めなければならない。

第８章　演算増幅器の基礎

8.1　演算増幅器とは

演算増幅器とはもともとアナログ計算機に用いられる高利得，広帯域の増幅器のことであったが，これが集積回路化されるに至って，安価で，容易に入手できるようになり，現在では汎用増幅器として種々の回路に応用されている。

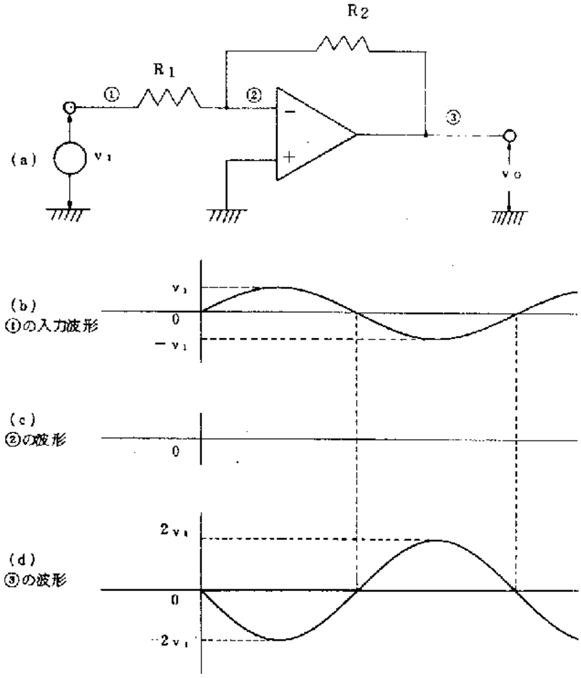


図８－１．演算増幅回路の基本

図８－１に示す回路が演算増幅器の最も基本となる回路である。ここで三角形で表わされているのが，演算増幅器で，通常２つの入力端子と１つの出力端子を有している。R_1，R_2は外づけされた抵抗である。いま入力端子に最大値 v_1〔V〕の正弦波を加えてみる。R_2／R_1の値を２倍とすれば，出力端子③には最大値 $2v_1$〔V〕で，位相が反転した正弦波信号が得られる。すなわち，入力 v_1の－２倍が出力電圧 v_oとなる。これを式で示すと。

$$v_o = k v_1 \qquad (k = -2) \cdots\cdots (1)$$

となる。すなわち入力信号 v_1の定数倍が出力として得られる。

8.2　演算増幅器の基本的性質

図８－２に示すような⊕入力端子と⊖入力端子および出力端子をもち，利

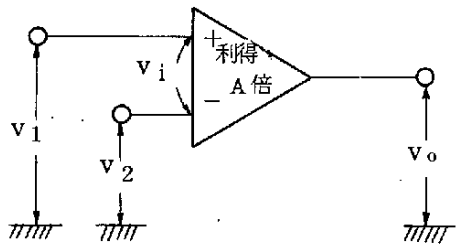


図８－２．増　幅　器

得がA倍の増幅器を考えてみる。この増幅器の出力電圧 v_o は次のように表わされる。

$$v_o = Av_i = A(v_1 - v_2) \quad \cdots\cdots(2)$$

すなわち，２つの入力端子間に加えられた電圧 v_i のA倍が出力となる。いま $v_2 = 0$ とすると出力は，

$$v_o = A v_1$$

となり⊕入力端子に加えられた電圧のA倍となる。また $v_1 = 0$ とすると

$$v_o = -A v_2$$

となり，⊖入力端子の電圧の－A倍，すなわち逆相出力が得られる。図８－２の増幅器の入力側に書かれている⊕，⊖の記号はこのような意味をもつている。また入力側の端子間の電圧は電位差０となる。

一方，出力電圧は有限の電圧がでている。(2)式より増幅器の入力電圧 v_i は

$$v_i = \frac{v_o}{A} \quad \cdots\cdots(3)$$

と得られる。式(3)で $v_o \neq 0$ の有限な値とすると $A = \infty$ のときに $v_i = 0$ となる。これが演算増幅器の基本的な性質である。すなわち演算増幅器は，利得が無限大の増幅器で，有限の出力の状態では，その入力端子間には電圧が生じない。

演算増幅器は非常に大きな利得をもっているから，図８－２に示すように直接 v_i （$\neq 0$）を入力に加えるような使い方はできない（直接 v_i を加えると出力は無限大となり，演算増幅器の出力回路が飽和して増幅器として作用しなくなる）。図８－１に示すように，外部に抵抗などの素子を接続して，利得を下げて使用する。図８－１では入力電圧 $v_i = 2\ V_{PP}$ に対して出力は $4\ V_{P-P}$ であるから，利得は２倍となっている。

以上述べたように演算増幅器の利得は外部回路により，下げられた状態では使用される。図８－１で抵抗 R_1，R_2 はこのような目的に使用されている。特に，出力電圧 v_o を R_2 を通して演算増幅器の入力へ戻している。このように出力の一部を入力側へ戻すことを帰還という。

8.3　理想演算増幅器とは

標準的な演算増幅器は図８－３に示すように正負一対の入力端子と一つの

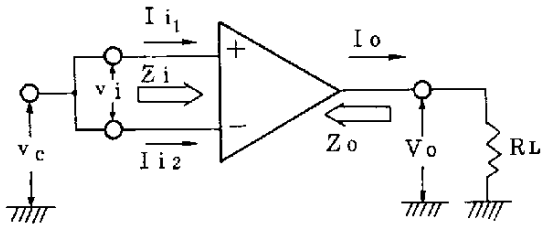


図８－３．演算増幅器

出力端子をもっている。

理想演算増幅器とは，次のような性質を有する演算増幅器をいう。

① 入力電圧 v_i に対する利得は無限大である。

入力電圧 v_i を差動入力電圧という。v_i に対する利得を差動利得と呼び，次式で表わす。

$$Ad = v_o / v_i \quad (Ad = \infty \text{理想演算増幅器}) \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

差動利得Adは無限大であるから，理想演算増幅器の入力端子間には0でない電圧をかけることはできない。逆にいえば，出力に有限振幅の電圧 v_o が生じているような正常な使用状態では必ず入力端子間の電圧 v_i は0になっている。

② 正負入力端子に同一電圧 v_c を加えた場合の出力電圧は0である。

正負入力端子に加えられる同一入力電圧 v_c を同相入力電圧という。v_c に対する利得を同相利得と呼び，次式で定義する。

$$Ac = v_o / v_c \quad (Ac = 0 : \text{理想演算増幅器}) \cdots\cdots\cdots\cdots\cdots\cdots (2)$$

同相利得Acの持つ意味は，演算増幅器の持つ種々のパラメータは温度などによって変化する。このようなパラメータの変化は等価的に同相入力とみなせる。したがって同相利得が0であれば，出力電圧はこれらパラメータの変化の影響を受けないことになる。

③ 入力端子に流入する電流は0である。

入力に電流が流れないということは演算増幅器の入力インピーダンスが無限大であることを意味している。すなわち

$$Zi = \infty$$

④ 出力電圧 V_o の大きさは，負荷抵抗 R_L の大きさに依存しない。

これは出力電流 I_o は負荷抵抗 R_L を小さくすれば，いくらでも取り出すことができ，演算増幅器の出力側回路は理想的な電圧源で構成されており，出力インピーダンスが0（$Z_o = 0$）であることを意味している。

⑤ 差動利得 Adは周波数に依存しない。

理想演算増幅器は直流より無限大周波数まで無限大利得を有しており，通常の増幅器のように周波数の増加にともなって利得が減少しない。

⑥　増幅器に外部より入力を加えない限り出力は0である。

演算増幅器には正負一対の入力端子があるが，この2つの入力端子が完全に平衡している場合には，入力信号を外部より加えない限り，出力には電圧が現われない。これを演算増幅器のオフセットが0であるという。

以上が基本的な理想演算増幅器の性質である。では実際の演算増幅器と理想演算増幅器との比較をしてみると図8－4のようになる。

	理想演算増幅器	実際の演算増幅器
差動利得　Ad	∞	100 dB（10^5倍）以上
同相利得　Ac	0	0 dB（1倍）以上
入力インピーダンス Zi	∞	300 kΩ以上
出力インピーダンス Zo	0	75 Ω
周波数帯域幅	∞	10 Hz（－3 dB）
入力換算オフセット電圧	0	1 mV

図8－4.　理想演算増幅器と実際の演算増幅器の比較

8.4　理想演算増幅器の等価回路

演算増幅器が理想的である場合には，簡単な等価回路表示ができ，演算増

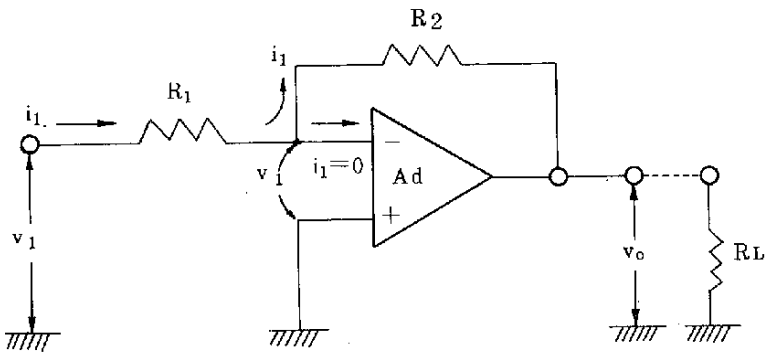


図8－5.　演算増巾器回路

幅器回路の解析設計が非常に容易になる。

次にナレータ，ノレータの概念について述べ，これを用いて演算増幅器の等価回路を表してみる。

図８－５の回路で，まず差動利得Adを有限として考えてみる。すなわち

$$v_o = -Ad\ v_i \quad \cdots\cdots(1)$$

が成り立つ。また演算増幅器の入力インピーダンスを無限大，すなわち $i_1 = 0$ とすると，R_1 を流れる電流 i_1 はすべて R_2 に流れるから，

$$v_1 - v_o = (R_1 + R_2)\ i_1 \quad \cdots\cdots(2)$$

$$v_i = v_1 - R_1\ i_1 \quad \cdots\cdots(3)$$

が成り立つ。式(1)，(2)，(3)より i_1，v_i を消去すると

$$v_o = \frac{\dfrac{R_2}{R_1 + R_2}Ad}{1 + \dfrac{R_1}{R_1 + R_2}Ad}\ v_1 \quad \cdots\cdots(4)$$

が得られる。また式(1)より v_i は

$$v_i = -\frac{v_o}{Ad} = \frac{\dfrac{R_2}{R_1 + R_2}}{1 + \dfrac{R_1}{R_1 + R_2}Ad}\ v_1 \quad \cdots\cdots(5)$$

となる。理想演算増幅器では，差動利得Adは無限大だから，式(4)，(5)で $Ad \to \infty$ とすると

$$v_o = -\frac{R_2}{R_1}\ v_1 \quad \cdots\cdots(6)$$

$$v_i = 0 \quad \cdots\cdots(7)$$

が得られる。すなわち図８－５の回路は利得が $-R_2/R_1$ であり，また常に理想演算増幅器の入力端子間の電圧 v_i は０となる。

ここで演算増幅器の入力端子に着目してみる。

理想演算増幅器は入力インピーダンスは無限大であるから，入力端子には電流が流れない。また式(7)に示すように，入力端子間の電圧も０となる。このように理想演算増幅器の入力端子間は電流も電圧も外部回路とは無関係に０となる。

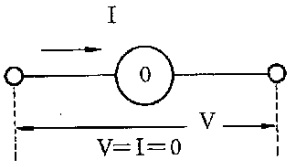


図８－６．ナレータ

図８－６.に示す記号はナレータと呼ばれる２端子素子で，その端子間電圧Ｖと端子を流れる電流Ｉは常に０であるという性質をもつ仮想の素子である。したがって理想演算増幅器の入力端子は，このナレータで置き換えることができる。図８－７はこのようにして図８－５の回路をナレータで置きかえたものである。この図より出力電圧 v_o を求める。

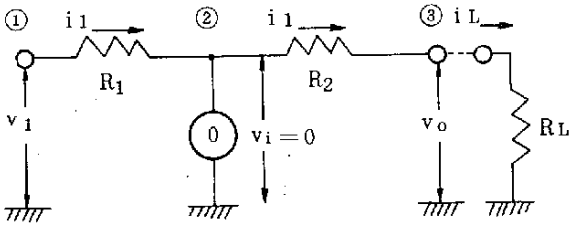


図８－７．ナレータを導入した等価回路

R_1 を流れる電流 i_1 は R_1 の両端（① － ② 間）の電圧を R_1 で割ったものであるから

$$i_1 = \frac{v_1 - v_i}{R_1} \qquad \cdots\cdots(8)$$

となる。ここでナレータの両端の電圧 v_i は０であるから結局 i_1 は

$$i_1 = \frac{v_1}{R_1} \qquad \cdots\cdots(9)$$

となる。ナレータには電流は流れないから，i_1 はすべて R_2 を流れることになる。したがって $v_i = 0$ であるから，出力電圧 v_o は R_2 の両端の電圧とな

るから

$$v_o = -R_2\, i_1 \quad \cdots\cdots(10)$$

と得られる。(10)に(9)を代入すると

$$v_o = -\frac{R_2}{R_1} v_1 \quad \cdots\cdots(11)$$

となり，式(6)の結果と一致する。

ナレータの導入は，それほど不自然ではなく，別段，矛盾のないように見えるが，計算を進めていくと，不都合なことが生じる。例えば図８－７で③の端子に抵抗R_Lを接続してみる。電流i_1は③の端子より負荷抵抗R_Lに流入するから，R_Lの両端の電圧は

$$v_o = R_L \cdot i_1 = \frac{R_L}{R_1} v_1 \quad \cdots\cdots(12)$$

となる。しかしR_Lの両端の電圧はv_oに等しいはずであるから，明らかに式(10)と(11)は異なっている。v_o'がv_oに等しくなるためには，R_Lには

$$i_L = \frac{v_o}{R_L} = -\frac{R_2}{R_L \cdot R_1} v_1 = -\frac{R_2}{R_L} i_1 \quad \cdots\cdots(13)$$

という電流が流れなければならない。i_Lとi_1は等しくないから，図８－７の端子③の点でキルヒホッフの電流則を満足しなくなる。そこでこの矛盾をとり除くために③の端子に電流通路を設け，この電流通路にi_1とi_Lの差の電流を流し，③の点でキルヒホッフの電流則を満足するようにする。図８－８にこの様子を示す。図８－８では③と接地点の間に図８－９に示

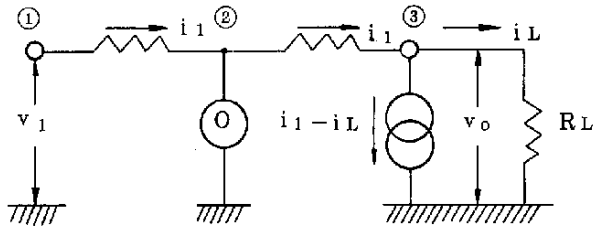


図８－８　電流に矛盾のない回路

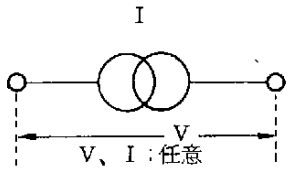


図８－９　ノレータ

すような2端子素子が導入されている。この2端子素子をノレータと呼ぶ。ノレータを流れる電流は図8－8では$i_1 - i_L$であり，またその両端の電圧はv_oである。一方i_Lおよびv_oは式(11)，(13)からわかるように，R_1，R_2，R_Lを変えることにより変わる。すなわちノレータの電流および電圧は，ノレータ自身が有している固有の電流および電圧ではなく，ノレータに接続されている周りの回路の電流・電圧によって決定される。したがってノレータの性質は流れる電流および両端の電圧が任意に定められるということである。

以上述べたように，ナレータ，ノレータをそれぞれ1個ずつ含む図8－8の回路は電流を見ても，電圧をみてもまったく矛盾がない演算増幅器の等価回路となる。

第9章　ノイズについて

最近の集積回路は，高集積・高速化・低消費電力化に向かって早いスピードで進んでいるが，各集積回路が進歩するにともなって，逆に幅の狭いパルス性のノイズに対しても応答してしまい，設計した回路を製作しても予定通りに動作せず，チェックに時間がかかり，非常に苦労をする事が多くなってきている。この様なことを少しでも少なくするためにTTLの耐ノイズ性について述べる。

9.1　TTLの耐ノイズ性

今日，広く普及しているデジタルICの代表はTTL SN 74シリーズとSN 74 LSシリーズである。まず，電気的特性に注目してみると，入出力関係から最悪ノイズ・マージンは0.4Vである。TTL SN 74とSN 74 LSの特性表を表9－1に示す。

表9－1.　TTL SN 74，SN 74 LSシリーズの特性

パラメータ		SN 74シリーズ			SN 74 LSシリーズ			単位
		MIN	TYP	MAX	MIN	TYP	MAX	
"H"レベル入力電圧 V_{IH}		2			2			V
"L"レベル入力電圧 V_{IL}				0.8			0.8	V
"H"レベル出力電圧 V_{OH}		2.4	3.4		2.7	3.4		V
"L"レベル出力電圧 V_{OL}	$I_{OL}=$ MAX		0.2	0.4		0.25	0.5	V
	$I_{OL}=$ 4 mA						0.4	V

図9－1(a)のように，論理信号線にV_Nというノイズ電圧が誘起する場合には，最悪の場合で0.4Vしか許容できないということである。またゲート2の入力電圧で判断する場合にはゲート1の出力電圧とノイズ電圧V_Nの和が0.8Vになるのが最悪の場合の許容限度となる。もっともSN 74シリーズのスレッショルド電圧の代表値は1.4Vであるから，ノイズマージンの代

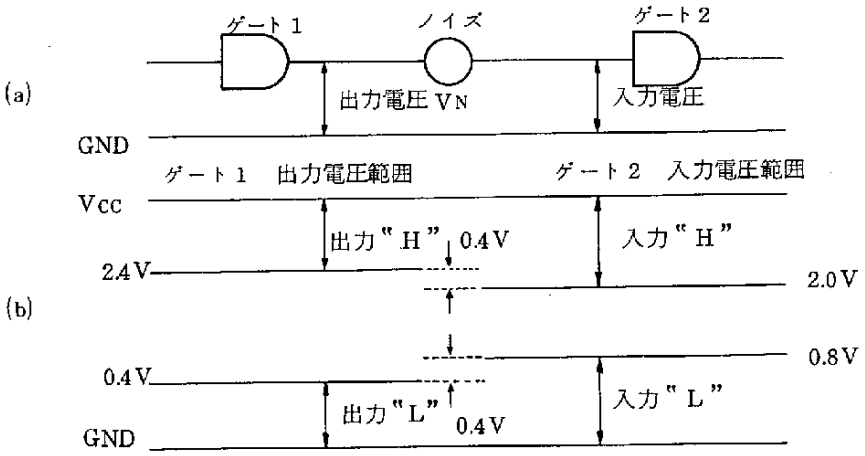


図9－1.　TTL SN 74シリーズのノイズマージン

表値はゲート1の出力電圧0.4Vを差し引いた1Vということになるが，メーカの保証値はあくまで最悪値0.4Vである。つぎにスイッチング特性に注目してみる。出力が"H"になる時と"L"になる時とでは遅延時間に差があり，さらにSN 74シリーズとSN 74 LSシリーズでは若干の差がある。ここでは簡単のために10 nsと考えておこう。この遅延時間というのは，入力信号を印加してからこれだけの時間がたてば，出力電圧が変化するから巾10 ns以上の入力信号を印加すれば，出力側に変化が生じるのである。つまりこの程度の細いパルスにも応答する。

以上のべたことをまとめると，ゲートの入力端子でみて電圧0.8 V以上，巾10 ns以上のノイズであれば，そのゲートは誤動作をする恐れがあるということである。

9.2　ノイズの分類

ノイズの分類方法は使用目的によりいろいろな分類ができる。ノイズ対策をたてる場合には，いくつかの分類法を併用するのが適当である。

①　成因による分類

- 雑音
  - 内部雑音
    - 電子管（電子的，マイクロホン的）
    - 抵抗（熱雑音，電流雑音）
    - 磁気的（バルクハウゼン，バルクハウゼン効果粒子雑音）
    - 電源（ハム，モータボーディング）
  - 外来雑音
    - 自然雑音（気象学的，天文学的）
    - 人工雑音（放電，摺動接触，持続振動）

② ノイズの性質による分類

- 雑音
  - ランダム雑音
    - インパルス雑音
    - 連続性雑音（ホワイトノイズ，3波雑音）
  - 周期性雑音
    - ハム，マイクロホン的，ハウリング，モータボーディング

③ 結合・伝播の形態による分類

- 結合・伝播
  - 静電誘導
  - 電磁誘導
  - 直接結合
    - 共通インピーダンス
    - 漏れ電流
  - 伝導妨害
  - 放射妨害

以上簡単な分類を述べたが詳細については，各専門誌を参照されたい。

第10章　接地とリターン回路

10.1　筐体のアース

電算機やレーダ装置等の設備や装置は，きちんと据付けられた機器に手をふれても「ピリッ」と電気がきて，びっくりすることはまずない。これに対して測定器を持ち出してきてデータを取ったり，あるいは実験室で実験をしたり，または工場で配線作業を終ったばかりの機器に初めて電気を入れたりする場合によく「ピリッ」と体験するものである。この「ピリッ」のある場合とない場合とは，どこが違うか，現実には大きな装置ほど「ピリッ」とこないのが普通である。特に据付工事に1ヶ月も2ヶ月もかかるような機器には，アース工事がきちんとしていてしっかりしたアースを筐体に結んでいる。小型のものは，手にぶらさげて運搬し，ほこりを払って電源を投入する。これでは「ピリッ」とこない方がおかしいくらいである。「ピリッ」とくるといった初歩的なミスはきわめて簡単に防ぐことができるのであって，筐体をアースに接続さえすればよいのである。心電計，あるいは洗濯機のようなものはそれぞれ指定のアース端子を利用して，しっかりアース線を結べばよい。しかしユーザならこれでよいが，メーカではこれではすまされない。アース端子の出し方を知らなければならない。普通筐体はメラミン塗装されているので，ボルトを取付ける際に，その部分の塗装をはいでおかないと，せっかく「アース端子を付けても導通なし」ということになってしまう。ボルトの取付方として次のような方法がある。

① ボルトをタップ止めにする。

筐体にいきなりタップを立てる方法なので一番推奨できる。

② ボルトを溶接する。

ただし，鉄の筐体に黄銅のボルトは溶接できないので，鉄のボルトを溶接しなければならない。

なおボルトの取付場所は，電気的にいえば，文字通り適当でよい。という

ことは筐体アースは「ポテンシャルを決めるアース」であって，「電流を流すアース」ではないからである。電気は金属の中はよく流れることは誰でも知っている。金属の中でも銀や銅は特に電気を通しやすいが，アルミニウム，鉄でも電気を流すことには変わりがない。そこで「鉄でできているのだから筐体だって電気は流れる」と考え，つい「アースを取ること」と「アースに電気を流すこと」を混同して筐体に電流を流している場合がかなりある。筐体に電気を流すと筐体の持っている抵抗によって，筐体の各部の電圧が違ってきていろいろな障害を与える。したがって筐体に電気を流すことは絶対にいけない。

10.2　電気部品のアース

コンデンサ，可変抵抗器，変圧器などの電気部品は可能な限り，そのケースとか，カバーをアースしなければいけない。なぜこのようなことをしなければいけないのだろうか。

ある種の機器を何台も使ったが，どうも特性が１台ごとに違う，また１台の機器でも測定して得られたデータが測定のたびごとに微妙な点で不安定であるといったことがよくある。それにボディエフェクトといって人がダイアルをつかんだだけでも特性が違ってしまうこともある。こんな場合，部品とアース間のストレーキャパシティが部品ごとに違うためであったり，シールド線のシールドと部品のケースが不完全に接続していたりしているため，データが不安定なこともある。また人体によってストレーキャパシティの大きさが変化して特性がかわることもよくある。

このような不安定なことをなくすためにも筐体自身はもちろん，シャーシとか電気部品のケースなどは電気的にしっかりアースしておく必要がある。アースしてあれば，対アースのストレーキャパシティがなくなる場合が多いし，あっても一定不変である。一定であれば，ストレーキャパシティにより特性が変動することもなく，安定なデータが得られるわけである。また，電

気磁気学的に考えても，ストレーキャパシティが一定であるということは，シールドがよいということとまったく同じなので特性が安定するとともに，シールドのよい機器を製作することもできるわけである。

ところで，電気部品を塗装したシャーシ上にそのまま取付けたものをよく見かける。そして「シャーシに付いているのだからアースされているはずだ」と考えがちである。しかし「塗装されていることは絶縁されていること」なのである。電気部品のケースは決して電気的にはシャーシにはつながっていないのである。したがって部品を取付ける場所は塗装をしないままにしておくことが絶対に必要である。

10.3　直流電源のリターン回路

直流電源のアース回路は，リターン回路を考える上で非常に簡単なものである。しかし問題が全然ないわけではない。次に考えられる問題点をあげてみると次のものがある。

① リターン回路の電圧降下の問題

② 電流計シャント（分流器），過負荷リレーの挿入方法

③ リモートセンシング

④ 出力インピーダンス

ここでは，リターン回路の電圧降下について述べる。

直流電源に負荷を加えた場合，リターン回路ができていなければ電流は流

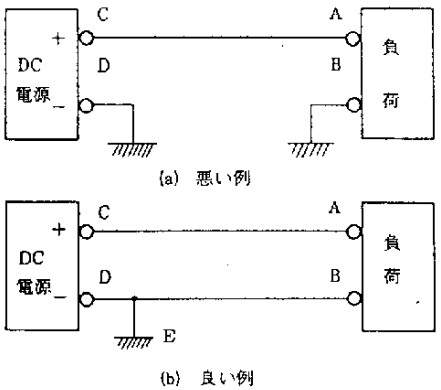


図 10－1. 直流電源のリターン回路

れないし，電流が流れなければ配線が不完全だということはすぐにわかる。また直流であるがためにいくらシャーシに流してもハムを発生することもない。このため何となく気楽にアースしてしまう。しかし気楽にアースする前に，やはりアースに落してから一体どこを通って元の電源にもどってくるか，一応，チェックする必要があろう。図10－1(a)は悪い例で，アースされた筐体の中をどの様に電流が流れてしまうかは不明となってしまう。(b)のようにきちんと帰り道を作らなければならない。だが5mAとか100mAとかいった程度の場合には何の問題も起きないので(a)のような配線をしてしまう。がやはり(b)のようにリターン回路はどうなっているか考え，しっかり配線しておくことを習慣づけるとよい。電流が大きくなって2Aとか5Aとかになってきたらリターン回路の電圧降下について，一応の注意が必要である。

例えば，コンピュータのようにICを多く使用する機器では，直流電圧が5Vで電力が500Wというと直流電流は100Aにもなる。こうして計算してみると，びっくりする人が案外多い。100Aともなると，これはちょっとしたビニール電線では電圧降下が大きくてまずいということは誰でも気が付く。そして図10－1(b)でCA間の電圧降下を計算して電線の太さを決める。が帰りのBD間の電圧降下はまるで考えなかったなどという実例もある。また電流が大きい電源たとえば5V 100Aの電源で電源電圧を測定しなさいというと，文字通り直流電源の出力端子とアース間（(b)図でCE間）の電圧を測ってくる人がいる。「100Aも流れているのだから途中の電線の電圧降下があって，実際の負荷にかかる電圧はもっと低いはずだ。だからもう1度測定し直しなさい」というと「わかりました」といって負荷の入力端子Aと，直流電源のアース端子E間の電圧を測ってくる。このような人はまったくリターン回路のことが頭にない人である。こういう人にはあまり怒ったりせず，ていねいに負荷端子A－B間の電圧を測らなければ正確な直流電圧にならないことを説明してやらねばならない。

第11章　4ビットマイクロコンピュータ

一般にマスク用CPUといわれているもので，実際に動作する時には1チップの中にCPU機能と動作をさせるためのプログラムをマスクして特定のアプリケーションに対応させるものである。アプリケーションはユーザの設計したものがマスクされるので，ユーザ独自のCPUが作れる。プログラムを開発するために1チップCPUと同等の機能をもち，外部にROMまたはRAMを接続するだけで，プログラムの動作の確認のできる装置が用意されている。これを評価用システムといい，各社独自のものを準備している。

ここではTMS1000シリーズ（PMOS　CPU）について述べてみる。

TMS 1000シリーズマイクロコンピュータのオン・チップ I／Oは，4入力，8～10並列出力，11ないし16個の個々にアドレス可能なラッチ付出力の能力がある。図11－1にTMS 1000シリーズの入／出力一覧表を示す。しかしながら複雑な計算やコントロール・アプリケーションではさらに入・出力

デバイス	TMS 1000	TMS 1200	TMS 1070	TMS 1210	TMS 1100	TMS 1300
パッケージピン数	28	40	28	40	28	40
入力ライン（k）	4		4		4	
独立ラッチ出力（R）	11	13	11	13	11	16
並列データのラッチ出力	8 bit		8 bit	8 or 10 bit	8 bit	

図11－1. TMS 1000シリーズ入／出力構成

を追加する必要があることがあり，このような場合には外部回路を使用することによって簡単に用意することができる。図11－2にTMS 1000の入出力電気特性を示す。

平均出力電流　O出力　－２４ｍＡ
　　R出力　－１４ｍＡ
最大出力電流　O出力　－４８ｍＡ
　　R出力　－２８ｍＡ

パラメータ		試験条件	MIN	TYP	MAX	単位
I_I 入力電流　K入力		$V_I = 0V$	50	300	500	μA
V_{OH} 高レベル出力電圧	O出力	$I_O = -10$ mA	-1.1	-0.6		V
	R出力	$I_O = -2$ mA	-0.75	-0.4		V
I_{OL} 低レベル出力電流		$V_{OL} = V_{DD}$			-100	μA
供給電圧			-14	-15	-17.5	V
高レベル入力電圧 V_{IH}	K		-1.3	-1	0.3	V
	INTまたはクロック		-1.3	-1	0.3	V
低レベル入力電圧 V_{IL}	K		V_{DD}		-4	V
	INTまたはクロック		V_{DD}	-15	-8	V

図 11 － 2.　TMS 1000 電気特性

〔例〕　ＴＭＳ 1000を使用したハード回路

時計をＴＭＳ 1000を使用して作る場合のハード設計について以下に述べる。

仕様　表　　示：4桁 24時間計とする。

入力キイ：時，分，調整用×２

セット完了用キイ×１

50／60 切換 SW×１

時計用クロック：AC電源より取込む

ブロック図により各入出力ポートの決定をする。

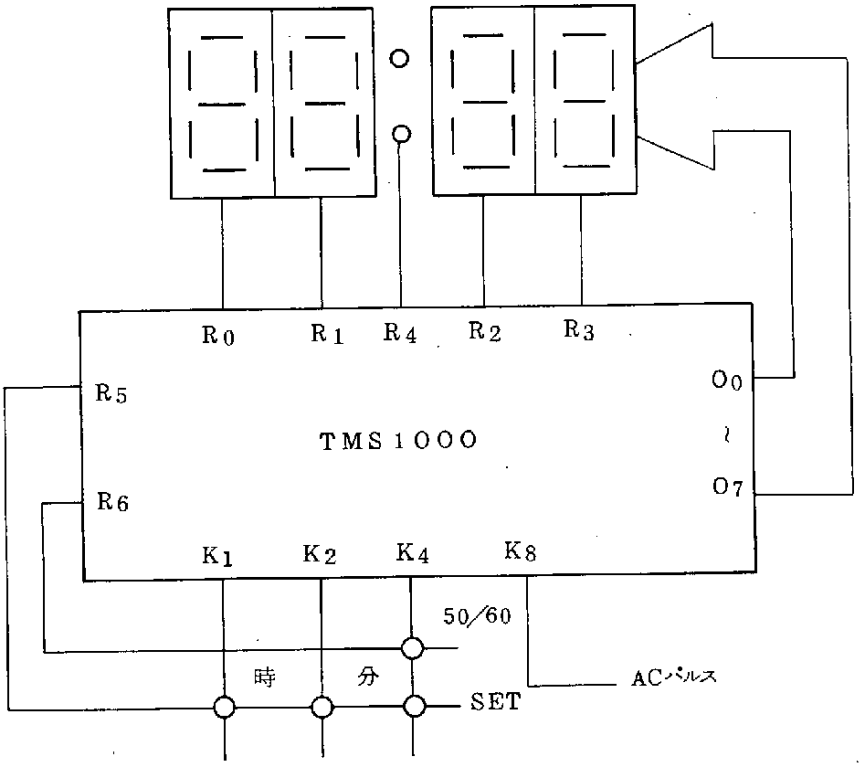


図11－3　TMS1000を使用したブロック図

注意事項：ブロック図はあくまでも入出力ポートを決めるためのもので論理が正しいものとはかぎらない。

以上のブロック図よりハードの図面を作成する。

◦使用部品の決定

回路図面より数量決定

◦各定数の決定

発振周波数：メーカが提示している資料に基いて決定する。この設計では

f＝300kHZを設定している。この時の1命令の実行時間は

$$1\text{命令実行時間} = \frac{1}{300 \times 10^{3}} \times 6 = 20\,\mu s$$

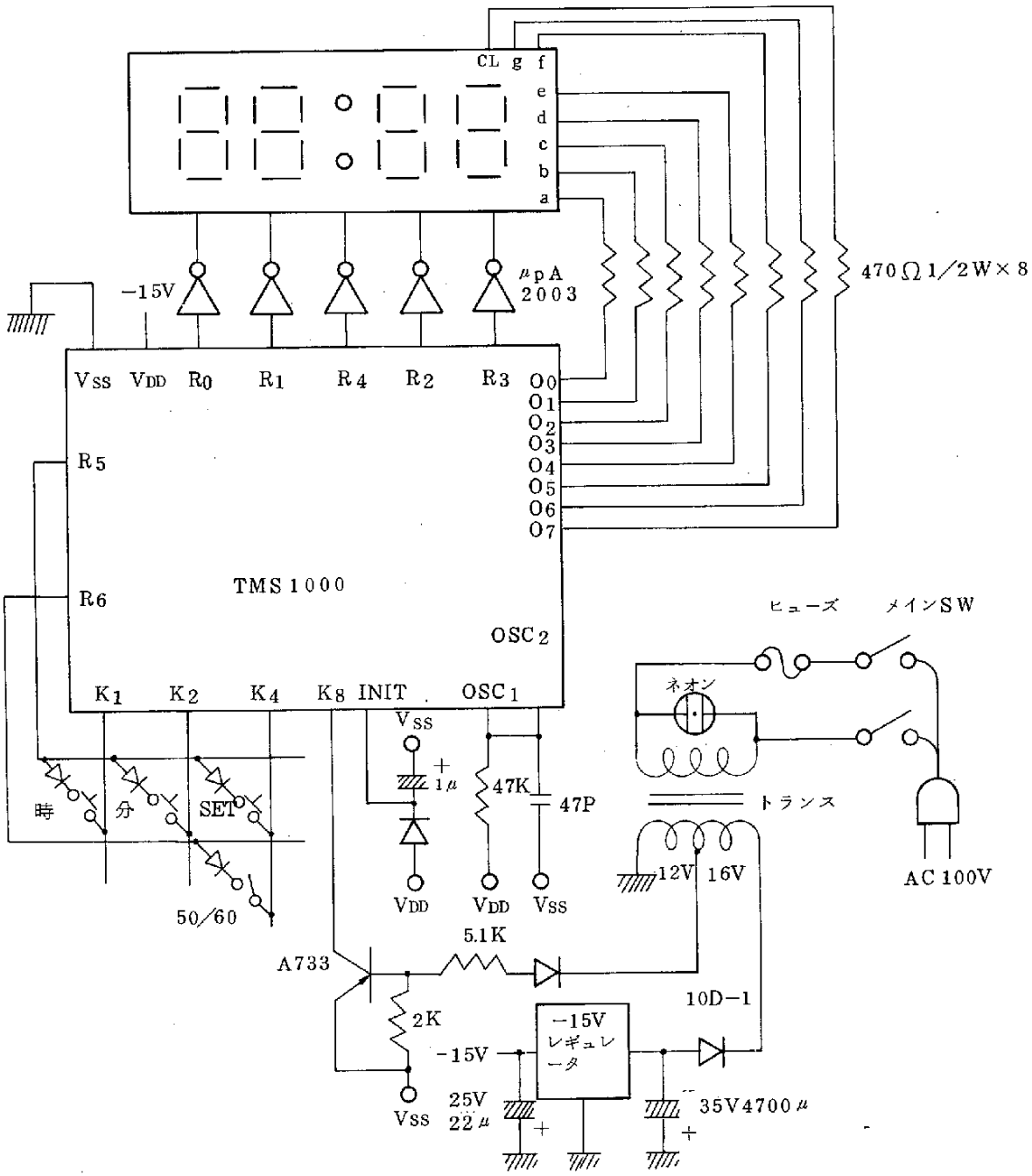


図11−4 時計用回路図面

INIT：C_{EXT}（μF）＝0.06×パワー立上り時間（ms）の関係によりコンデンサの容量が決まるので，パワーの立上り時間をおよそ20msとすると

$$C = 0.06 \times 20 = 1.2\,\mu F \fallingdotseq 1\,\mu F$$

となり，1μFのコンデンサを取付ける。この時のダイオードの取付の理由は，電源オフにした時にコンデンサ内に蓄えられた電荷を早く放電させるためのものである。

O出力の抵抗：O出力の最大出力電流は図11－2により48mAと規定されているので30mAを使用するとすると

$$R = \frac{15\ V}{30\ (mA) \times 10^{-3}} = 500\,\Omega \fallingdotseq 470\,\Omega$$

$$W = (15V / 470)^2 \times 470 = 0.48\,W \fallingdotseq 1/2\,W$$

となり，470Ωで1／2Wの抵抗を取付ける。

R出力のインバータの使用：7セグメントで8の数字を点灯させるとすると

$$I_R = \frac{15}{470} \times 7 = 223\,mA$$

となり，一般に使用されているTTLとインバータでは電流を引張りきれないのでμpA 2003を使用する。このICは1つのゲート当り500mAの電流を引き込むことができる。

K8入力の取込：AC 12Vをダイオードにより半波整流を行い，V_{SS}より低い電位の時にトランジスタをONさせてACの信号をK8に入力する。

〔参考文献〕

(1) 「制御用マイコン初歩から応用まで」 日刊工業新聞社

(2) 「電源回路設計の基礎」 オーム社

(3) 「電子実用回路ハンドブック」 CQ出版社

(4) 「インターフェース回路の設計」 CQ出版社

(5) 「インターフェースIC」 テキサスインスツルメンツ社

(6) 「制御機器のノイズ対策」 石井康太郎著 日刊工業新聞社

(7) 「アース回路」 伊藤健一著 日刊工業新聞社

(8) 「演算増幅器回路の設計」 藤井信生著 産報出版

(9) 「マイクロコンピュータハードウェア基礎技術」 横井与次郎著 ラジオ技術社

(10) 「SHARPZ-80 テクニカルマニュアル」 SHARP社

(11) 「ICハンドブック」 テキサスインスツルメンツ社

(12) 「MOSハンドブック」 モトローラ社

第2部　マイクロコンピュータの実際

第１章　８０８５の構造と機能

1.1　ＣＰＵチップの役割

現在最もポピュラーなＣＰＵチップとして８０８５をとりあげ，ＬＳＩとしての機能について説明する。

８０８５は，ＮチャネルＭＯＳの８ビットマイクロプロセッサである。

図１－１に示すような４０ピンＤＩＰ（デュアルインライン　パッケージ）に納められており，１つのＬＳＩチップとして作られている。

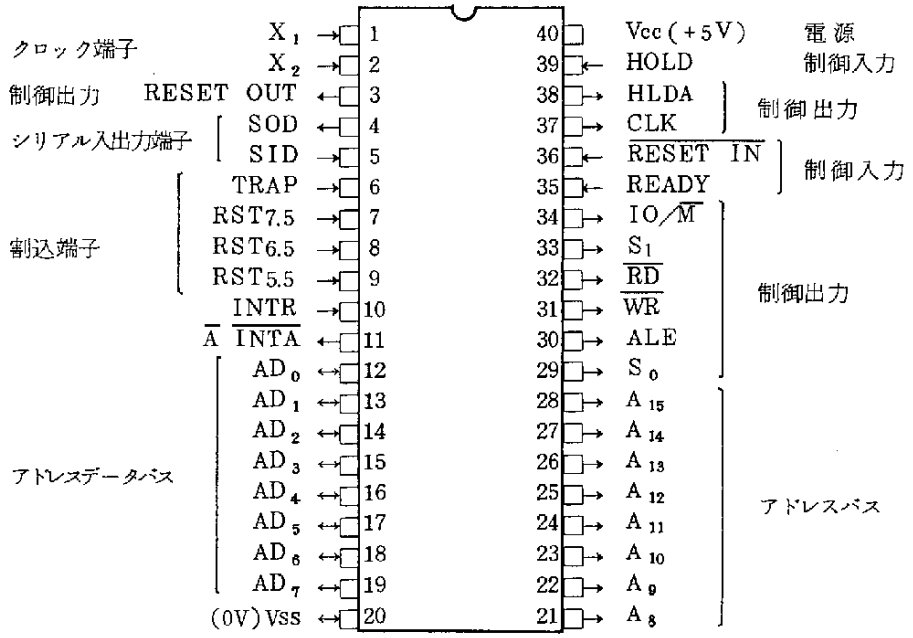


図１－１　８０８５の外観

８０８５には４０個の入出力端子があり，それぞれ図に示すような名称が付けられている。それらを機能で分類すると，アドレスバス，アドレスデータバス，タイミング・制御出力信号，制御入力信号，割込信号，シリアル入出力信号，電源，クロック端子などに分けることができる。８０８５の内部構成を図１－２に示す。

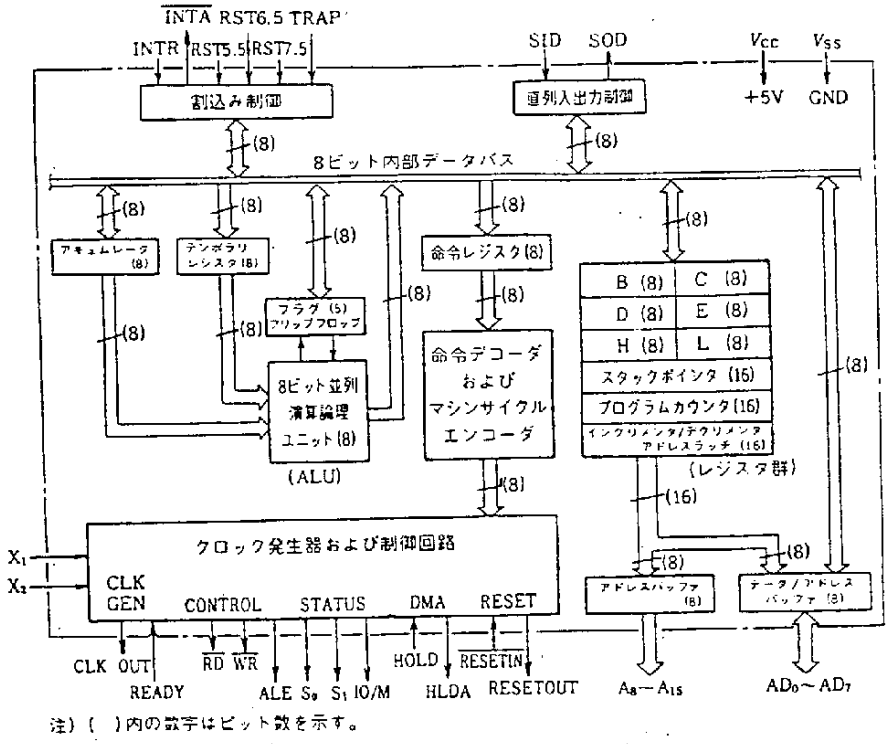


図１－２　８０８５の内部構成

このＣＰＵチップを機能別に分けると演算部，制御部，レジスタ部で構成されている。

(演算部)

この部分は算術論理演算を行うところで以下のようなもので構成されている。

◦２進演算回路

算術演算（加算，減算），論理演算（And，Or，Exclusive Or），比較などを行う。

◦シフト，ローティド回路

シフトや回転を行う。

◦１０進補正回路

２進演算の結果を２進化１０進数（ＢＣＤ）に変換を行う。

◦５ビットのフラグ

算術演算や論理演算を行った結果，その状態（演算のオーバーフロー，

サイン，パリティ，など）を表わす。

◦ 汎用ワーキングレジスタ

B，C，D，E，H，L，A（アキュームレータ）レジスタは，各々8ビット長のレジスタで直接命令によって操作でき，いろいろな目的のために使用できる。さらに，それらのレジスタは，BC，DE，HLと対にしてメモリの参照，倍精度（16ビット）の加算に用いられる。

Aレジスタは，他の汎用レジスタと違って演算結果の記憶（例外もあるが），入出力命令によるI／Oとのデータの入出力などに用いられる。

◦ テンポラリレジスタ

CPUが処理を行うときに使用されるレジスタで命令によって操作できない。

◦ プログラムカウンタ

プログラムカウンタは，次に実行すべき命令のアドレスを指定する16ビットのレジスタである。

◦ スタックポインタ

スタックとしてメモリの何番地まで使用しているかを示す16ビットのレジスタである。

(制御部)

◦ 命令レジスタ

メモリから読みだした命令を一時保持するレジスタである。

◦ 命令デコーダおよびマシンサイクルエンコーダ

命令レジスタにセットされた命令を解読し，その命令によって決められたシーケンス動作を制御する。

◦ クロック発生及び制御回路

◦ 割込み制御，直列入出力制御

X_1，X_2端子に外付けした水晶やRC回路で作られたクロック信号をもとに各制御信号をコントロールし，CPUがデータバスのデータを読みこ

むための$\overline{RD}$信号とか，データバスにデータを出力する$\overline{WR}$信号を出力したり，外部からの信号（ＲＥＳＥＴ　ＩＮ　ＨＯＬＤなど）を調べたりする。

８０８５の外部端子にでている制御信号は以下の通りである。

◦ X_1，X_2（入力）

クロックを内部で発生させるために必要な水晶またはＲＣ回路を接続するための端子である。

◦ ＣＬＫ（出力）

X_1，X_2に接続された発振回路により，ＣＰＵ内部で発生したクロック信号を出力する端子である。周辺回路へのクロック信号として使用できる。

◦ ＲＥＳＥＴ　ＩＮ（入力）

最も優先度の高い信号で，ＣＰＵをリセットする場合に用いる。プログラムカウンタや命令レジスタ制御信号はリセットされ初期状態となる。したがってＣＰＵは０番地の命令から実行する。

◦ ＲＥＳＥＴ　ＯＵＴ（出力）

ＣＰＵがリセット状態になっていることを示す出力信号である。周辺回路へのシステムリセット信号として利用できる。

◦ ＨＯＬＤ（入力）

ＣＰＵをＨＯＬＤ状態にして，アドレスバス，データバスおよび$\overline{RD}$，$\overline{WR}$，ＩＯ／$\overline{M}$信号をフローティング状態（高インピーダンス状態：アドレスバス，データバスがＣＰＵから切り離された状態）にする。後述するＤＭＡ（ダイレクトメモリアクセス）モードで使用される信号である。

◦ ＨＯＬＤＡ（出力）

ＣＰＵがＨＯＬＤになったことを示す信号である。

◦ ＲＥＡＤＹ（入力）

ＣＰＵのリード，ライト動作にＷＡＩＴ状態を作ることができる。これによって，アクセスタイムの遅いメモリやＩ／Ｏを接続することができる。

◦ ＡＬＥ（出力）

図１－３に示すように，８０８５ではアドレスの下位８ビットとデータバスを共用しているので，ＡＬＥ（Address　Latch　Enable)により，ＣＰＵ外部にアドレスラッチを外付けしてアドレスバス１６ビットを生成する。ＡＬＥのタイミングを図１－４に示す。

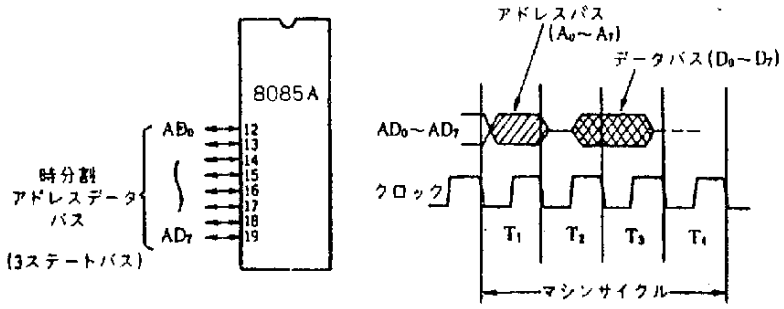


図１－３　アドレス・データバスAD_0～AD_7

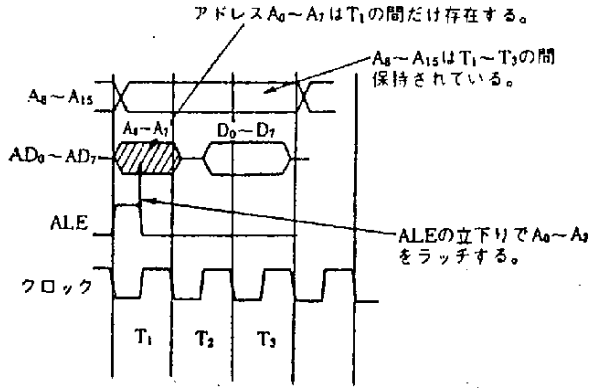


図１－４　ＡＬＥ信号のタイミング

また，この信号はステータス情報をラッチするのにも使用できる。

◦ $\overline{RD}$（出力・３ステート）

アドレスバスによって選択されたメモリ又はＩ／Ｏからデータバスを通じてデータを読みとるための信号である。ＨＯＬＤおよびＨＡＬＴモードの時はハイインピーダンスになる。

◦ $\overline{WR}$（出力・3ステート）

アドレスバスによって選択されたメモリ又はI／Oに対してデータバスを通じてデータを書きこむための信号である。HOLD又はHALTモードの時はハイインピーダンスになる。

S_0，S_1，IO／$\overline{M}$

S_0，S_1は，CPUが停止（HALT），書き込み（WRITE），読みだし（READ），命令フェッチ（FETCH）のどの状態にあるかを示すステータス情報である。

またIO／$\overline{M}$は，読みだし又は書き込みがメモリに対するものか，I／Oに対するものかを示す。図1－5に各信号に対するステータスを示す。

IO／$\overline{M}$	S_1	S_2	状　態
0	0	1	メモリ書込み
0	1	0	メモリ読取り
1	0	1	I／O書込み
1	1	0	I／O読取み
0	1	1	命令フェッチ
1	1	1	割込み受付け
*	0	0	停　止

*は3ステート（高インピーダンス）を示す。

図1－5　ステータス情報

◦ INTR（入力）

割込要求（Interrupt Request）を受け付けるための端子である。実行中の命令が終了する時又はHALT状態の時に受付けられる。INTR信号は，命令の最後の期間のみサンプリングされる。割込を受付けるとプログラムカウンタのインクリメントを止め，$\overline{INTA}$を出力し，割込応答のサイクルになる。そしてこのサイクル中に割込サービスルーチンへ分岐するためのRST命令やCALL命令を挿入することができる。

INTRによる割込要求に対してプログラムで割込を受付け可能にしたり禁止したりすることができるが，割込受付直後又はRESET時には割

込禁止となる。

割込については割込の項で詳細に述べる。

◦ $\overline{\text{INTA}}$（Interrupt Acknowledge）

割込要求に対する応答信号である。

この信号によってＲＳＴ命令やCall命令を挿入する。

◦ＲＳＴ7.5，ＲＳＴ6.5，ＲＳＴ5.5（入力）

割込要求信号を受付ける端子である。前述のＩＮＴＲと違いすでに再スタート番地が決められており，割込の要求信号には優先順位がある。優先順位はＲＳＴ7.5が一番高く，次がＲＳＴ6.5，ＲＳＴ5.5が最下位である。

また，先ほどのＩＮＴＲと比較すると，ＩＮＴＲはＲＳＴ5.5より低くなる。

ＲＳＴの入力にはＣＰＵ内部にプログラマブルなマスクがあり，これによって割込の受付けをコントロールできる。

また信号の形態で見ると，ＲＳＴ7.5はエッジトリガセンス（信号の立上がりで割込があったと判断する）であるのに対し，ＲＳＴ6.5，ＲＳＴ5.5はレベルセンス（信号がＨレベルになっていると割込があったと判断する）点が違っている。前者は短いパルス信号で割込を要求することができる。

◦ＴＲＡＰ（入力）

先ほどから述べてきた割込要求信号のうち最も優先順位の高い割込端子である。この割込要求信号は，前述の割込マスクにもこの割込可能Ｆ／Ｆの状態に関係なく受付けられ，再スタートが実行される。この割込は，電源断とか，バスのエラーを検出したとか，重大なエラーを防止する目的に利用される。ＴＲＡＰはレベルでもエッジトリガでも検出される。図１－６に割込要求信号をまとめた。

名　　称	優先順位	再スタート番地	トリガ形態	マスク
TRAP	1	24 (16)	立ち上りおよびHレベル検出	なし
RST7.5	2	3C (16)	立ち上り	可能
RST6.5	3	34 (16)	Hレベル検出	可能
RST5.5	4	2C (16)	Hレベル検出	可能
INTR	5	プログラムにより定める	Hレベル検出	可能

図１－６　ＣＰＵの持つ割込

◦ＳＩＤ（入力）　ＳＯＤ（出力）

ＳＩＤ（Serial　Input　Data）は直列データ入力用端子であり，ＳＯＤ（Serial　Output　Data）は直列データの出力用端子である。

ＣＰＵの命令によってプログラムで制御されるシリアルポートとして利用される。

◦Ｖcc（入力）

＋５Ｖ電源の入力端子

◦Ｖss

グランド端子

1.2　命令の構成とアドレス指定方式

マイクロプロセッサの機種によって命令の種類，形式，アドレス指定方式などには違いがある。しかし，その内容には同じ思想が根底に流れているのである。

命令には，何に対してどのような処理を行うかという情報を必ず含んでいる。

命令のどのような処理を行うかを示している部分を操作部（オペレーションコード）と呼んでいる。

一方，何に対してかを示す部分をオペランドと呼び，一般にオペランドの内容を示すメモリの番地が使用されるのでアドレス部とも呼ばれている。その形式を図１－７(a)に示す。

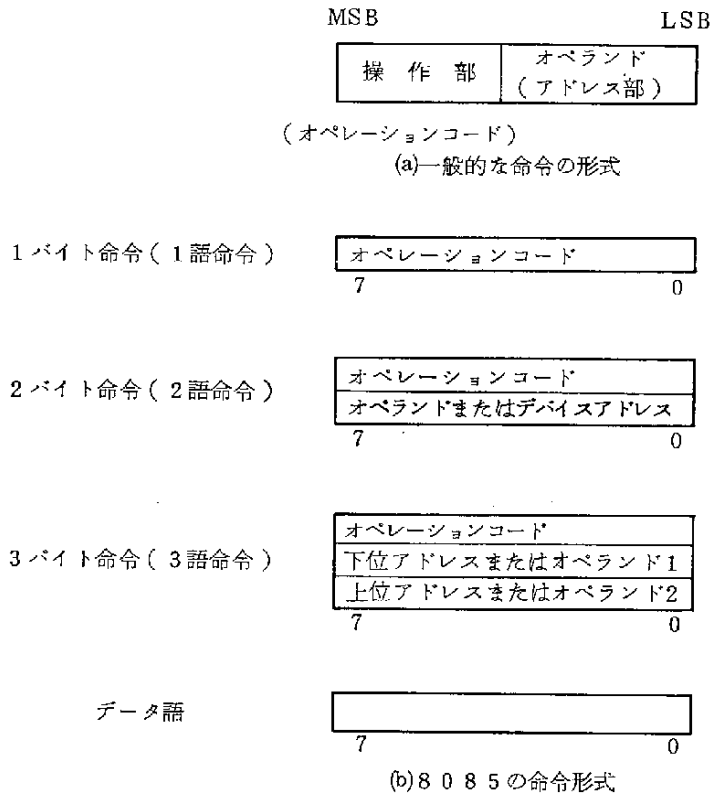


図 1 － 7

8085では図1－7(b)に示すように，すべての命令は1バイト，2バイト，3バイトのいずれかである。この図の1バイト命令はすべてのビットがオペレーションコードという訳ではなく，命令によってオペランドの指定が含まれているものもある。

8085の命令セットについて説明する前に命令のもつ一般的なアドレス指定方式について触れておこう。

一般にミニコンピュータでは16ビットが主流であるが，マイクロコンピュータでは4, 8, 16ビット，さらには自分で何ビットにでも構成できるビットスライスまである。

ここで各メーカーのCPUが特徴を発揮する部分はアドレス部で，この部分にアドレス変更（Address Modification）の機能をコード化して持たせ，種々のアドレス指定が行えるようになっている。この機能がCPUの持つア

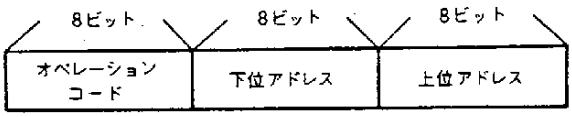
ーキテクチュアの中の重要な部分で，マイクロコンピュータがミニコンピュータに劣る点である。

アドレスの指定方式には，絶対アドレス，相対アドレス，インデックスアドレスなどがあり，またその中で直接，間接がある。

ここでは8085の命令フォーマットと対応させアドレス指定方式を説明する。

i) 直接アドレス指定

8085では3バイトの命令であって，図1－8(a)のように2バイト目，3バイト目の内容（16ビット）で直接データの格納されている番地を示す。



直接アドレス指定（8085）

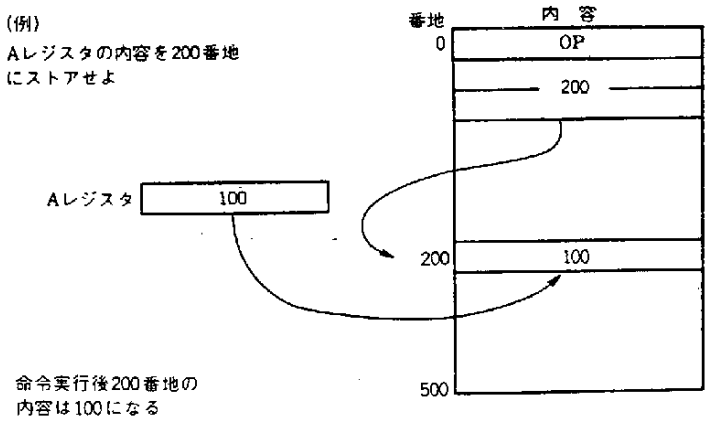
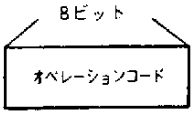


図1－8(a) 直接アドレス指定

ii) 間接アドレス指定

8085では図1－8(b)が示すように1バイトの命令で，命令が16ビットのレジスタ対を指定し，そのレジスタ対の内容が，データの格納されている番地を指定する。上記指定方式をレジスタ間接アドレス指定とい

う。



レジスタ間接アドレス指定（8085）

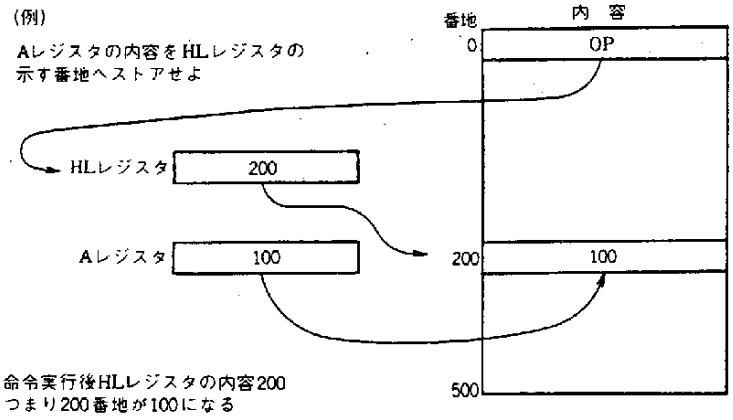
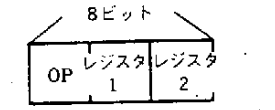


図１－８(b)　間接アドレス指定



レジスタアドレス指定（8085）

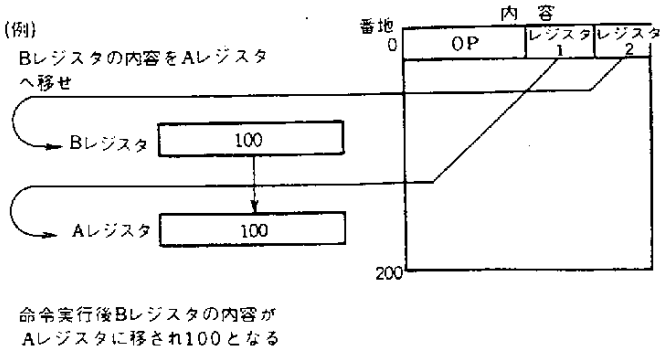


図１－８(c)　レジスタアドレス指定

ⅲ）インプライド（Implied）アドレス指定

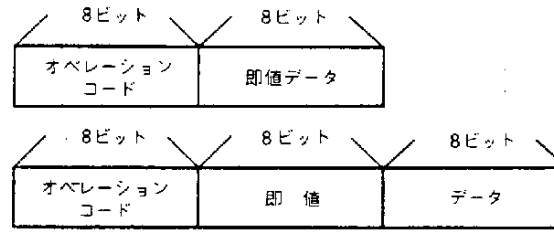
その命令によってオペランドとなるレジスタが決っているもので，8085では1バイト命令となる。

ⅳ）レジスタアドレス指定

8085では1バイトの命令であって，オペランドとなるレジスタまたはレジスタ対を指定するフィールドがその中に含まれているものである。(図1－8(c)）

ⅴ）即値アドレス指定（Immediate Address）

8085では2バイト，3バイトの命令であって，図1－8(d)のように2バイト目または2バイト目，3バイト目のデータそのものがオペランドとなる。



即値アドレス指定方式

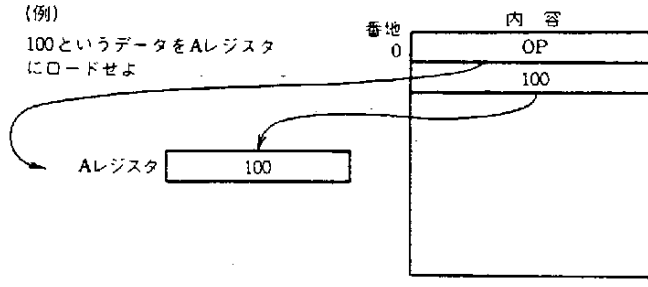
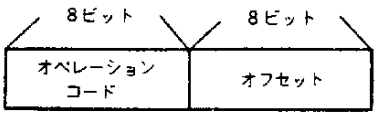


図1－8(d) 即値アドレス指定

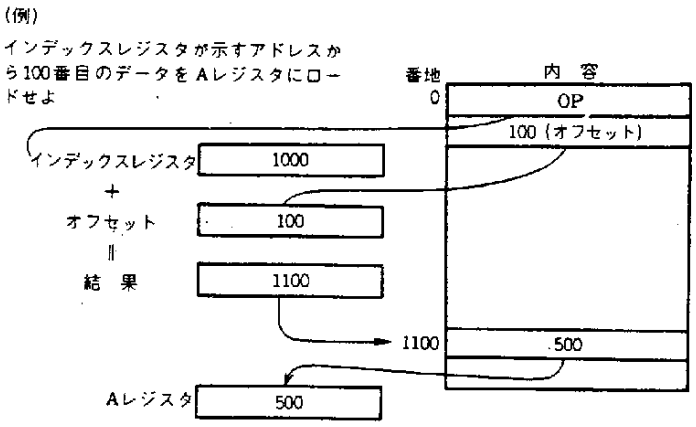
ⅵ）インデックスアドレス指定

8085はこの機能はもっていないが，一般にインデックスレジスタの内容にはオペランドの内容（8ビットで0～255）を加算し，メモリの番地を指定する方法で図1－8(e)にその一例を示す。

この指定方式を採用しているマイクロプロセッサには6800,Z80などがある。



インデックスアドレス指定



命令実行後インデックスレジスタと命令のオペランド(オフセット)が加算されたアドレス1100番地の内容500がAレジスタにロードされる.

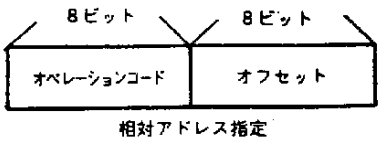
図1－8(e) インデックスアドレス指定

Vii) 相対アドレス指定

8085はこの機能をもっていないが，一般にプログラムカウンタ(PC)にオフセット分を加えアドレスを指定する。

この指定方式をうまく利用するとプログラムのリロケータブル化(任意のアドレスへプログラムを配置しても実行できる)が簡単に行うことができる。この方式を採用しているマイクロプロセッサには，6800,Z80などがある。その一例を図1－8(f)に示す。

一般にコンピュータが採用しているいくつかのアドレス指定方式について説明したが，8085は，その中で，直接アドレス，レジスタ間接アドレス，レジスタアドレス，インプライドアドレス，即値アドレスの5つの方式を採用している。



相対アドレス指定

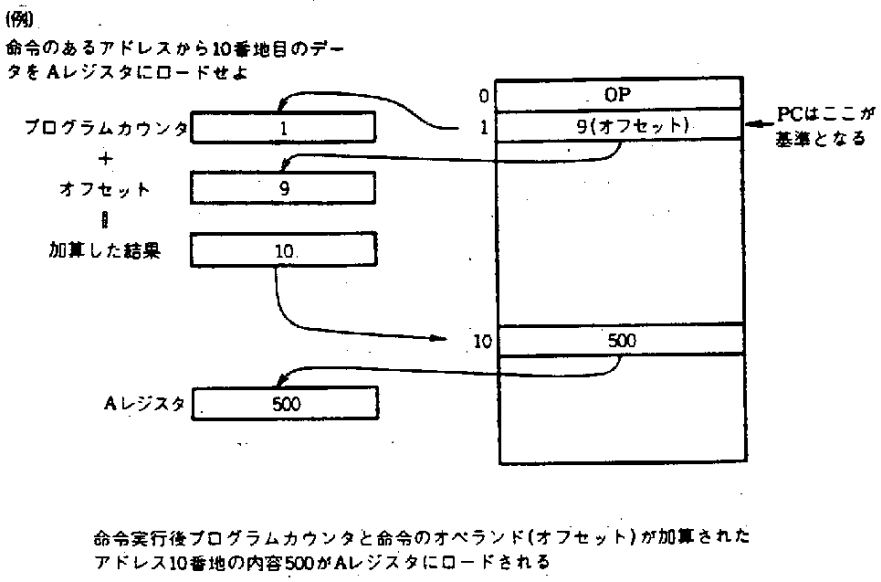


図1－8(f)　相対アドレス指定

(d)　命令の種類と命令セット

マイクロプロセッサは，Aレジスタメモリの番地からデータを転送せよとか，AレジスタとBレジスタを加算せよとか，その他はじめから組み込まれたいくつかの命令をもっている。

8085の命令を機能的な面で分類すると次のようになる。

データ転送命令

算術・論理演算命令

分岐命令

サブルーチンリンク命令

スタック操作命令

入出力，マシン制御命令

ここでは，命令の詳細な説明は省くことにする。

ⅰ）データ転送命令

これは，レジスタ間，あるいはメモリとレジスタ間でデータの転送を行う命令であり，ＭＯＶ，ＭＶＩ，ＬＤＡ，ＳＴＡ，ＬＸＩなどがある。

ⅱ）算術・論理演算命令

これは，レジスタまたはメモリのデータに対して加算，減算，論理積，論理和，排他的論理和などの演算およびシフト，回転，１０進補正，インクリメント／ディクリメント，フラグ操作，比較命令がある。

これらの命令を実行すると先に説明したフラグが影響をうける。

これらの命令としてＡＤＤ，ＳＵＢ，ＡＮＡ，ＩＮＲ，ＣＭＰ，ＳＴＣなどがある。

ⅲ）分 岐 命 令

これは，制御を移す命令であり，プログラムカウンタの内容を次に実行すべき命令のメモリアドレスへ変更することによって行われる。分岐には，先に述べたフラグの条件によって行われるものと，無条件に行われるものとある。

ＪＭＰ（無条件），ＪＮＺ，ＪＣ（条件つき）などの命令がある。

ⅳ）サブルーチン・リンク命令

サブルーチン　プログラムの中にあるまとまった同じ処理がある場合図１－９(a)のように書いても良いがメモリ効率が悪く不経済である。この場合図１－９(a)の中に処理Ａを図１－９(b)のようにメインプログラムの外に書き，必要に応じてメインプログラムから呼びだして使用することがよく行われる。このようなプログラムをサブルーチンと呼び，サブルーチンへ制御を移す命令ＣＡＬＬ（コール）および戻る命令ＲＥＴ（リターン）が用意されている。

また，サブルーチンの中にもあるまとまった処理があれば図１－１０の

のように，さらにサブルーチンとして使用することが行われる。これをサブルーチンのネスティングと呼んでいる。

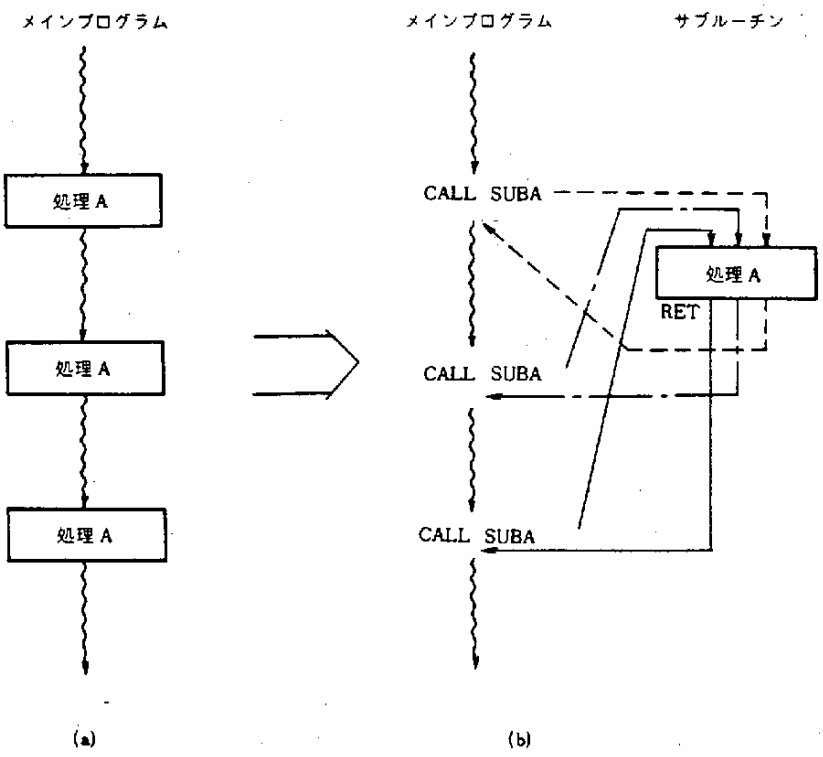


図１－９　サブルチーン

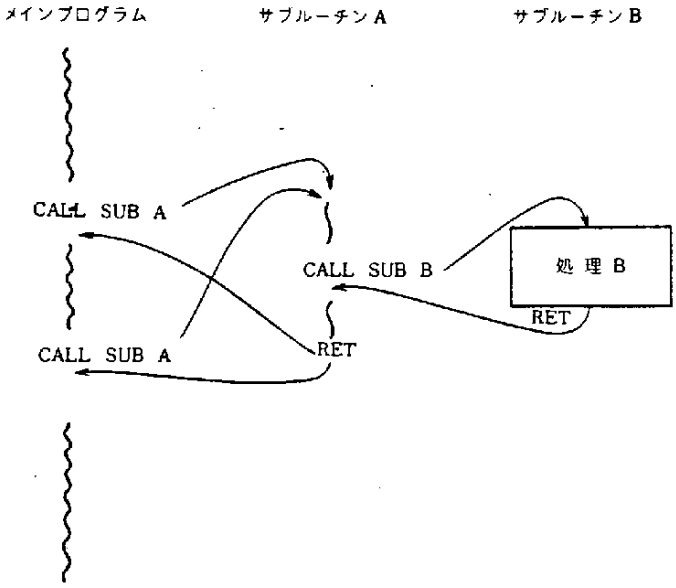


図１－１０　サブルーチンのネスティング

以上のように，ＣＡＬＬは，そのサブルーチンの処理が終ったら元のプログラム，すなわちＣＡＬＬの次の命令へもどって来ることを前提としている。この点が前述のＪＭＰ命令と違うところである。

したがって，ＣＡＬＬ命令を実行する場合次の命令が格納されているアドレスを一時的に退避しておく必要が生ずる。この退避領域としてスタックがあり，ＣＡＬＬ命令が実行されると，自動的にＣＡＬＬ命令の次のアドレスが転送されるのである。その様子を図１－１１に示す。また，ＣＡＬＬ命令，ＲＥＴ命令にも，ＪＭＰ命令の場合と同様に条件付きと無条件の命令がある。

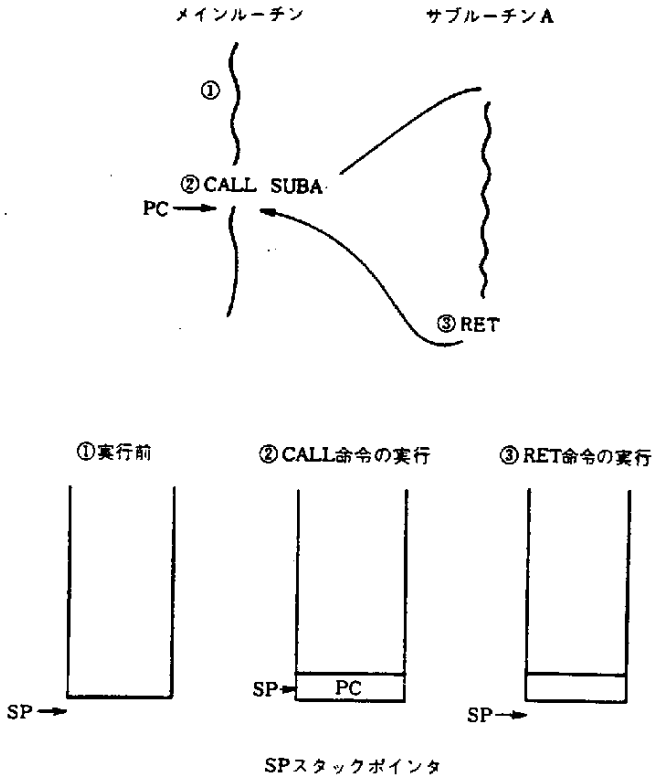


図１－１１　サブルーチンによるスタックの変化

V）スタック操作命令

スタック　スタックは，メモリ（ＲＡＭであることが必要）に，データやレジスタの内容を一時的に貯えるのに用いる。８０８５ではこのスタック

をコントロールするために前に述べたＳＰ（スタックポインタ）というレジスタを持っており，任意にレジスタの内容をスタックに入れたり，出したりすることができ，さらに前項で説明したサブルーチンリンク命令の実行時に重要な役割をもっている。

スタックにデータを入れることをＰＵＳＨ（プッシュ）といい，スタックからデータをとりだすことをＰＯＰ（ポップ）という。図１－１２にこの様子を示す。

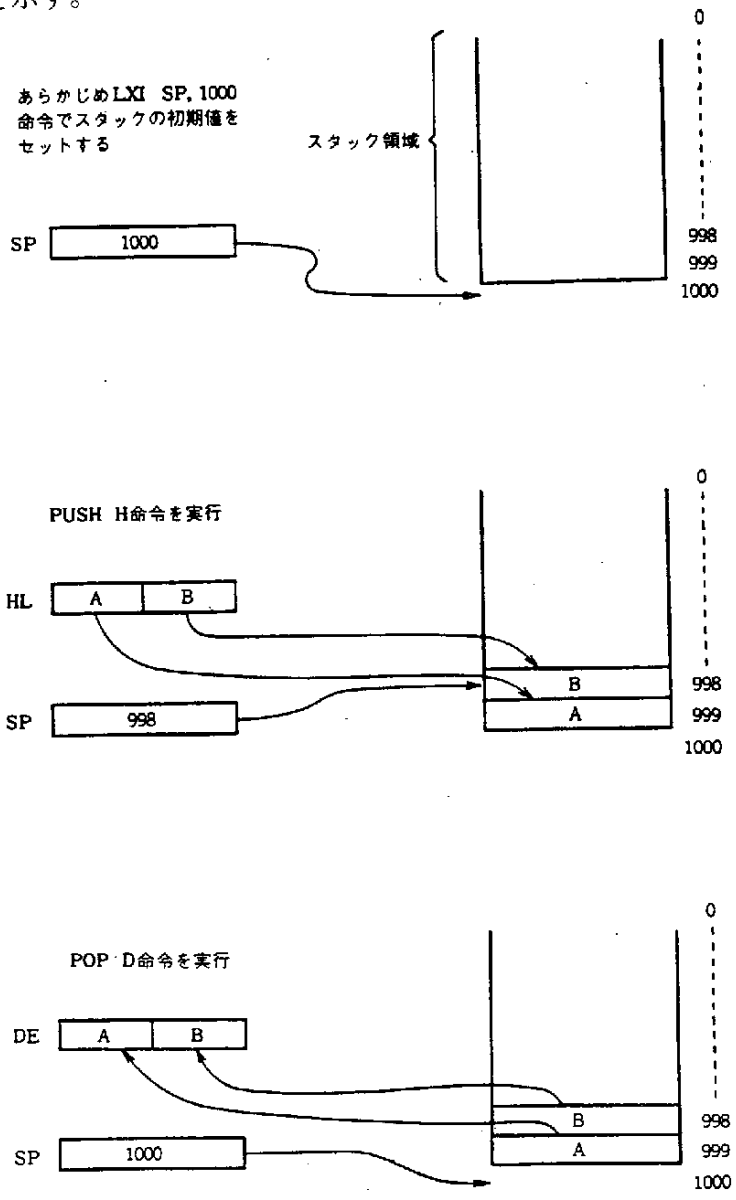


図１－１２　スタック操作命令によるスタックの変化

表1－1　8085の命令

命令群	ニーモニック	命令コード			ステート	オペレーション
ロード・ストア命令	MOV r1,r2	01DDDSSS			4	(r1)←(r2)
	MOV M,r	01110SSS			7	(M)←(r)
	MOV r,M	01DDD110			7	(r)←(M)
	MVI r	00DDD110	B2		7	(r)←(B2)
	MVI M	00110110	B2		10	(M)←(B2)
	LXI B	00000001	B2	B3	10	(B)←(B3),(C)←(B2)
	LXI D	00010001	B2	B3	10	(D)←(B3),(E)←(B2)
	LXI H	00100001	B2	B3	10	(H)←(B3),(L)←(B2)
	LXI SP	00110001	B2	B3	10	$(SP)_H$←(B3),$(SP)_L$←(B2)
	STAX B	00000010			7	[(B)(C)]←(A)
	STAX D	00010010			7	[(D)(E)]←(A)
	LDAX B	00001010			7	(A)←[(B)(C)]
	LDAX D	00011010			7	(A)←[(D)(E)]
	SHLD	00100010	B2	B3	16	[(B3)(B2)]←(L),[(B3)(B2)+1]←(H)
	LHLD	00101010	B2	B3	16	(L)←[(B3)(B2)],(H)←[(B3)(B2)+1]
	STA	00110010	B2	B3	13	[(B3)(B2)]←(A)
	LDA	00111010	B2	B3	13	(A)←[(B3)(B2)]
	XTHL	11100011			16	(L)←(SP),(H)←(SP+1)
	XCHG	11101011			4	(H)←(D),(L)←(E)
	PCHL	11101001			6	(PC)←(H)(L)
	SPHL	11111001			6	(SP)←(H)(L)
レジスタ増減命令	INR r	00DDD100			4	(r)←(r)+1
	DCR r	00DDD101			4	(r)←(r)-1
	INR M	00110100			10	(M)←(M)+1
	DCR M	00110101			10	(M)←(M)-1
	INX B	00000011			6	(B)(C)←(B)(C)+1
	INX D	00010011			6	(D)(E)←(D)(E)+1
	INX H	00100011			6	(H)(L)←(H)(L)+1
	INX SP	00110011			6	(SP)←(SP)+1
	DCX B	00001011			6	(B)(C)←(B)(C)-1
	DCX D	00011011			6	(D)(E)←(D)(E)-1
	DCX H	00101011			6	(H)(L)←(H)(L)-1
	DCX SP	00111011			6	(SP)←(SP)-1
演算・論理操作命令	ADD r	10000SSS			4	(A)←(A)+(r)
	ADC r	10001SSS			4	(A)←(A)+(r)+C
	SUB r	10010SSS			4	(A)←(A)-(r)
	SBB r	10011SSS			4	(A)←(A)-(r)-C
	ANA r	10100SSS			4	(A)←(A)∧(r)
	XRA r	10101SSS			4	(A)←(A)∨(r)
	ORA r	10110SSS			4	(A)←(A)∨(r)
	CMP r	10111SSS			4	(A)-(r)
	ADD M	10000110			7	(A)←(A)+(M)
	ADC M	10001110			7	(A)←(A)+(M)+C
	SUB M	10010110			7	(A)←(A)-(M)
	SBB M	10011110			7	(A)←(A)-(M)-C
	ANA M	10100110			7	(A)←(A)∧(M)
	XRA M	10101110			7	(A)←(A)∨(M)
	ORA M	10110110			7	(A)←(A)∨(M)
	CMP M	10111110			7	(A)-(M)
	ADI	11000110	B2		7	(A)←(A)+(B2)
	ACI	11001110	B2		7	(A)←(A)+(B2)+C
	SUI	11010110	B2		7	(A)←(A)-(B2)
	SBI	11011110	B2		7	(A)←(A)-(B2)-C
	ANI	11100110	B2		7	(A)←(A)∧(B2)
	XRI	11101110	B2		7	(A)←(A)∨(B2)
	ORI	11110110	B2		7	(A)←(A)∨(B2)
	CPI	11111110	B2		7	(A)-(B2)
	DAD B	00001001			10	(H)(L)←(H)(L)+(B)(C)
	DAD D	00011001			10	(H)(L)←(H)(L)+(D)(E)
	DAD H	00101001			10	(H)(L)←(H)(L)+(H)(L)
	DAD SP	00111001			10	(H)(L)←(H)(L)+(SP)
	DAA	00100111			4	10進 (BCD) 補正
	CMA	00101111			4	(A)←($\overline{A}$)
	STC	00110111			4	C←"1"
	CMC	00111111			4	C←$\overline{C}$
回転命令	RLC	00000111			4	(An+1)←(An),(A0)←(A7),C←(A7)
	RRC	00001111			4	(An-1)←(An),(A7)←(A0),C←(A0)
	RAL	00010111			4	(An+1)←(An),(A0)←C,C←(A7)
	RAR	00011111			4	(An-1)←(An),(A7)←C,C←(A0)

命令群	ニーモニック	命令コード			ステート	オペレーション
ジャンプ命令	JMP	11000011	B2	B3	10	ジャンプ(PC)←(B3)(B2)
	JNZ	11000010	B2	B3	10/7	Z＝"0"
	JZ	11001010	B2	B3	10/7	Z＝"1"
	JNC	11010010	B2	B3	10/7	C＝"0"
	JC	11011010	B2	B3	10/7	C＝"1"
	JPO	11100010	B2	B3	10/7	P＝"0"
	JPE	11101010	B2	B3	10/7	P＝"1"
	JP	11110010	B2	B3	10/7	S＝"0"
	JM	11111010	B2	B3	10/7	S＝"1"
コール命令	CALL	11001101	B2	B3	18	(SP-1)(SP-2)←(PC),(SP)←(SP)-2,(PC)←(B3)(B2)
	CNZ	11000100	B2	B3	18/9	Z＝"0"
	CZ	11001100	B2	B3	18/9	Z＝"1"
	CNC	11010100	B2	B3	18/9	C＝"0"
	CC	11011100	B2	B3	18/9	C＝"1"
	CPO	11100100	B2	B3	18/9	P＝"0"
	CPE	11101100	B2	B3	18/9	P＝"1"
	CP	11110100	B2	B3	18/9	S＝"0"
	CM	11111100	B2	B3	18/9	S＝"1"
	RST ×	11AAA111			12	(SP-1)(SP-2)←(PC),(SP)←(SP)-2,(PC)←(00000000AAA000)
リターン命令	RET	11001001			10	(PC)←(SP+1)(SP),(SP)←(SP)+2
	RNZ	11000000			12/6	Z＝"0"
	RZ	11001000			12/6	Z＝"1"
	RNC	11010000			12/6	C＝"0"
	RC	11011000			12/6	C＝"1"
	RPO	11100000			12/6	P＝"0"
	RPE	11101000			12/6	P＝"1"
	RP	11110000			12/6	S＝"0"
	RM	11111000			12/6	S＝"1"
スタック操作命令	PUSH B	11000101			12	(SP-1)←(B),(SP-2)←(C),(SP)←(SP)-2
	PUSH D	11010101			12	(SP-1)←(D),(SP-2)←(E),(SP)←(SP)-2
	PUSH H	11100101			12	(SP-1)←(H),(SP-2)←(L),(SP)←(SP)-2
	PUSH PSW	11110101			12	(SP-1)←(A),(SP-2)←(F),(SP)←(SP)-2
	POP B	11000001			10	(C)←(SP),(B)←(SP+1),(SP)←(SP)+2
	POP D	11010001			10	(E)←(SP),(D)←(SP+1),(SP)←(SP)+2
	POP H	11100001			10	(L)←(SP),(H)←(SP+1),(SP)←(SP)+2
	POP PSW	11110001			10	(F)←(SP),(A)←(SP+1),(SP)←(SP)+2
割込制御命令	DI	11110011			4	INTE F/F リセット
	EI	11111011			4	INTE F/F セット
入出力命令	OUT	11010011	B2		10	DB←(A),AB15-8,AB7-0←(B2)
	IN	11011011	B2		10	(A)←DB,AB15-8,AB7-0←(B2)
その他	HLT	01110110			5	HALT
	NOP	00000000			4	ノーオペレーション
	RIM	00100000			4	A7←SID
	SIM	00110000			4	SOD←(A7)

DDD or SSS	0	1	2	3	4	5	6	7
	B	C	D	E	H	L	M	A

r＝A，B，C，D，E，H，L

データ・バスとフラグ

DB_0	C
DB_1	"1" or "0"
DB_2	P
DB_3	"0"
DB_4	AC
DB_5	"1" or "0"
DB_6	Z
DB_7	S

命令＼フラグ	C	Z	S	P	AC
INR		×	×	×	×
DCR		×	×	×	×
ADD	×	×	×	×	×
ADC	×	×	×	×	×
SUB	×	×	×	×	×
SBB	×	×	×	×	×
ANA	0	×	×	×	1
XRA	0	×	×	×	0
ORA	0	×	×	×	0
CMP	×	×	×	×	×
ADI	×	×	×	×	×
ACI	×	×	×	×	×
SUI	×	×	×	×	×
SBI	×	×	×	×	×
ANI	0	×	×	×	1
XRI	0	×	×	×	0
ORI	0	×	×	×	0
CPI	×	×	×	×	×
RLC	×				
RRC	×				
RAL	×				
RAR	×				
STC	1				
CMC	C				
DAA	×	×	×	×	×
DAD	×				
POP PSW	×	×	×	×	×

vi) 入出力・マシン制御命令

8085は，6800のようにメモリではなく，入出力機器をアドレス指定して，アキュームレータ（Aレジスタ）と入出力機器とデータ転送を行う命令ＩＮ，ＯＵＴがある。

また，マシン制御命令として，割込許可Ｆ／Ｆのセット（ＥＩ）やリセット（ＤＩ）を行うもの，ＣＰＵを停止（ＨＬＴ）させるもの，何も実行しない（ＮＯＰ）がある。

以上命令の概要を述べたが，表１－１に8085の命令を示しておく。

第２章　マイクロコンピュータシステム

これまで主として8085を取り上げてマイクロプロセッサの動作を説明してきた。マイクロコンピュータは，そのマイクロプロセッサ（ＣＰＵ）を中心に，図２－１に示すような要素から構成される。

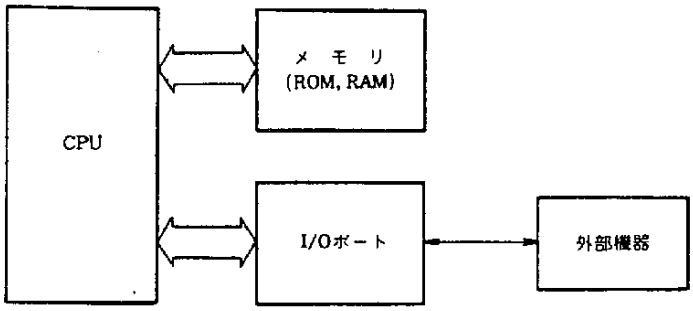


図２－１　マイクロコンピュータの構成

ＣＰＵに処理手順を与えるプログラムは，２進コードでＲＯＭに書並べておく。

ＲＯＭ（Read Only Memory）は，いったん書込んだ内容は電源が切れてもそのまま記憶し，動作中にも内容を書直すことのできない読出し専用のメモリである。

これに対し，ＲＡＭ（Random Access Memory＝通常は読書き可能メモリの意味）は電源が切れたとき内容は消失するが，動作中自由に読出しや書込みができるので，処理途中のデータや演算結果を記憶する目的に使用される。

一方，ＣＰＵは外部に接続された機器からデータを受取ったり，また機器の制御を行う。このような外部機器に対するデータの入出力動作は，すべてＩ／Ｏポートを通して行われる。

ＣＰＵは有機的に結合されたこれらの基本要素を制御しながら，それぞれとの間で命令やデータを送受して処理を進めて行くことになる。その処理に要する時間や複雑さに応じて，各要素の構成，結合や制御の方法が決定される。

2.1　マイクロコンピュータシステムの構成

(a)　ＣＰＵの構成

8085は前身の8080に比較すると，簡素化されているものの，そのままではメモリやＩ／Ｏを接続（小規模構成とする場合にはＲＡＭ，Ｉ／Ｏ8155やＲＯＭ，Ｉ／Ｏ8755がある）できるようにはなっていない。普通は図２－２に示すような２個のＩＣ（他のＩＣでも可能）の助けを借りて，メモリやＩ／Ｏポートの制御を行うことになる。

図２－２の中のＩＣ　ＬＳ373は時分割アドレスデータバスからアドレスバス下位８ビットの生成，ＬＳ２５７はメモリやＩ／Ｏに対するリードライト信号の生成を行っている。

実際のCPUモジュールとシステムバスの接続回路は，マイクロコンピュータシステムの規模により異なる。

小規模なシステムでは，図２－２に示したような8085を中心に３個のＩＣよりなるＣＰＵモジュールに，そのままメモリやＩ／Ｏポートの回路を接続することができる。しかし，接続されるメモリやＩ／Ｏポートの数が多くなれば，システムバスの駆動能力を増すためにバッファを入れるのが望ましくなる。さらにシステムの規模が大きくなったり，機能の拡張を図るような場合には，バスを基板外に取出すこともある。ＣＰＵ，メモリなどそれぞれのモジュールはＭＯＳの素子で構成される場合が多いので，それらの保護やバスの性能・信頼性の面からも，バッファを介してシステムバスに接続するのが普通である。

メモリやＩ／Ｏに対するリード，ライト制御信号をはじめ，割込み，ホールド，リセットなど，システムを制御する信号群をまとめてコントロールバスと呼ぶ。8085は２ケのＩＣと結合されてバス構造のＣＰＵシステムを制御することができる。

したがって，図２－２の構成は，最小限のＣＰＵブロック（モジュール）といえる。このＣＰＵモジュールとシステムを結ぶ信号をまとめると，図

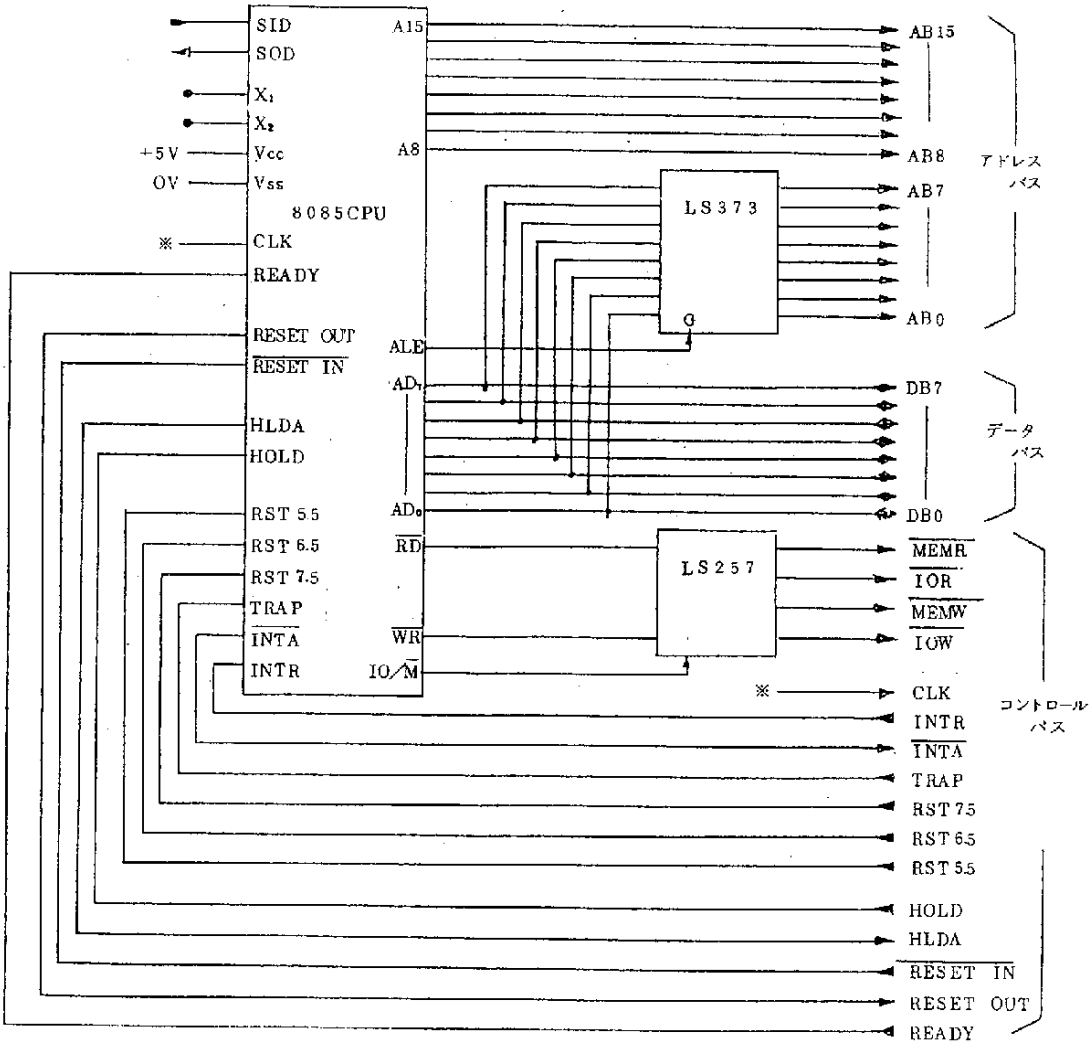


図2－2　8085マイクロコンピュータの
CPUモジュール

2－3のようになる。図2－2のアドレス，データ，コントロールの3群のバスをまとめて，システムバスと称する。

信号名	意味	入出力
AB0～AB15	アドレスバス	出
DB0～DB7	データバス	入／出
$\overline{\text{MEMR}}$	メモリ読出しストローブ	出
$\overline{\text{MEMW}}$	メモリ書きこみストローブ	出
$\overline{\text{IOR}}$	I／O入力ストローブ	出
$\overline{\text{IOW}}$	I／O出力ストローブ	出
INTR	割込み要求	入
$\overline{\text{INTA}}$	割込アクノレジ	出
TRAP	トラップ割込	入
RST 7.5	割込要求	入
RST 6.5	〃	入
RST 5.5	〃	入
HOLD	ホールド要求	入
HLDA	ホールドアクノレジ	出
READY	レディ制御(ウェイト要求)	入
RESET IN	リセット	入
CLK	システム同期用クロック	出

図2－3　システムバスの信号一覧

(b)　バス方式のシステム構成

8085を例に，マイクロプロセッサを使ってバス構造のCPUモジュールを構成するところまでを説明してきた。マイクロコンピュータでは，このCPUモジュールにメモリやI／Oポートを図2－4のように接続するのが一般的になっている。

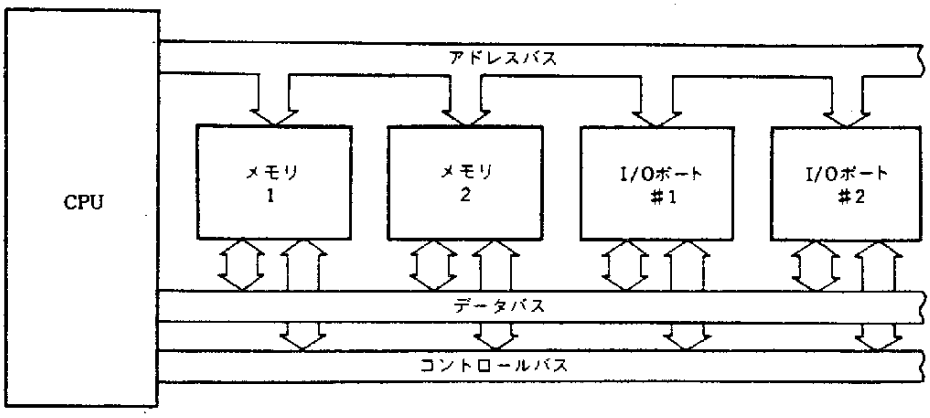


図2－4　バス方式のシステム構成

アドレスバス，データバス，コントロールバスを共通にして各要素を並列に接続するもので，バス方式と呼ばれる。データバスはＣＰＵとメモリ，Ｉ／Ｏポートの間をどちら向きにもデータが転送するので，双方向バスになっている。

(c)　マイクロコンピュータの基本動作

マイクロコンピュータの基本動作は，ＣＰＵとメモリやＩ／Ｏポートの間で命令やデータを送受することである。ＣＰＵが入出力を行う方法は，それが命令であるかデータであるか，また対象がメモリなのかＩ／Ｏで区別されている，というにすぎない。

先に述べた8085システムでは，図２－５に示すような方法でデータの入出力が行われる。

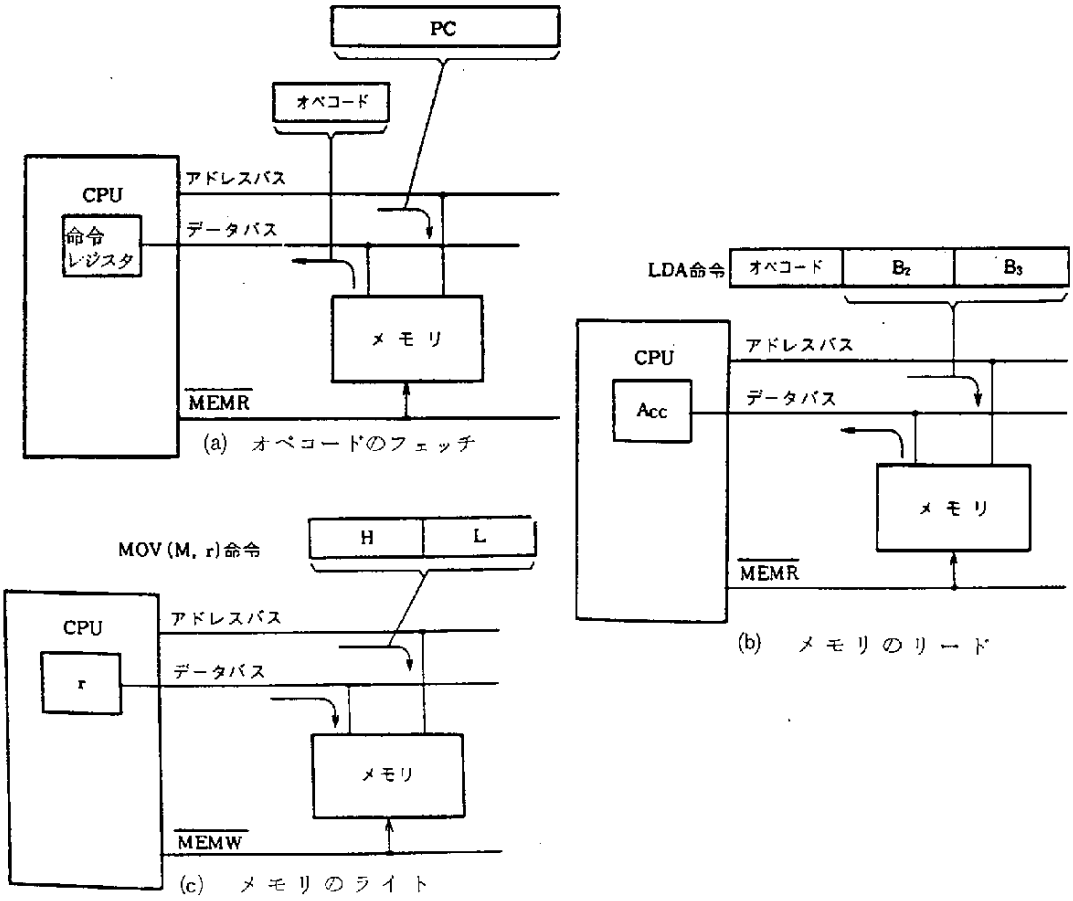


図２－５　メモリ，Ｉ／Ｏに対するリード・ライト動作

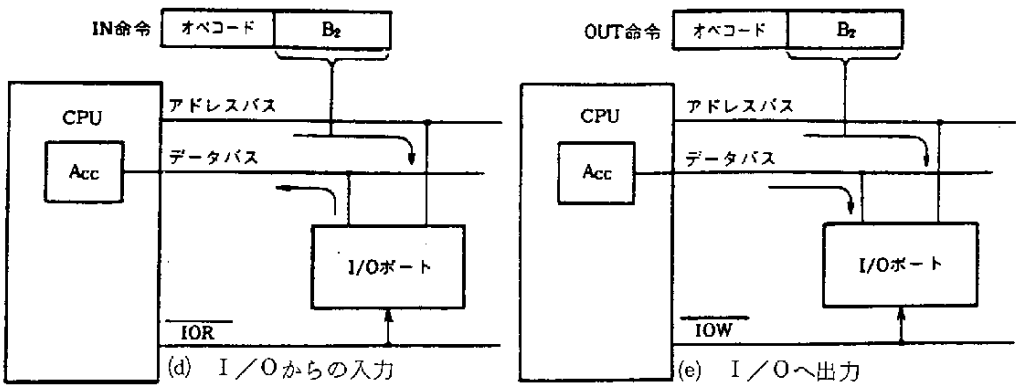


図2－5　メモリ，I／Oに対するリード・ライト動作

① CPUはメモリまたはI／Oポートのアドレス（番地，番号）をアドレスバスに出力する。命令フェッチサイクルにはプログラムカウンタ(PC）の値が，メモリ参照命令の実行サイクルには参照すべきメモリの番地が，それぞれアドレスバスAB0～AB15に出力される。またI／Oの時の入出力命令の実行サイクルでは，指定されたI／Oポートの番号がアドレスバスの上位8ビットにそれぞれに出力される（そのうち一方，普通下位8ビットの分を用いる）。

② メモリやI／Oポートは常にアドレスバスの内容を解読（デコード）しており，自分の番地が指定されたときのみCPUとデータの送受④を行う。

③ CPUがデータを出力する場合は，まずデータバス上に送出し，メモリまたはI／Oポートにその受取りを要求する出力ストローブ信号（$\overline{\text{MEMW}}$，$\overline{\text{IOW}}$）を送る。

またCPUがデータを読込む場合は入力ストローブ信号（$\overline{\text{MEMR}}$，$\overline{\text{IOR}}$）を出力して，メモリやI／Oポートにデータをバスに送出するように要求する。

④ アドレス指定されたメモリまたはI／Oポートは，出力ストローブ信号が送られた場合にはデータバス上のデータを読込み，入力ストローブ

信号が送られたときはデータをデータバス上に出力する。

メモリやＩ／Ｏポートが（ＣＰＵと）正しくデータを送受するには，ストローブ信号が出力されている限られた期間内に入出力動作を行わなければならない。

またメモリやＩ／Ｏポートの動作が遅くてこれらのタイミングに間に合わないような場合は，確実に入出力ができるまでREADY信号をリセットして，ＣＰＵを待ち状態（ウェイト）にしておくことができる。

図２－６にリード・ライトのタイミング図を示す。

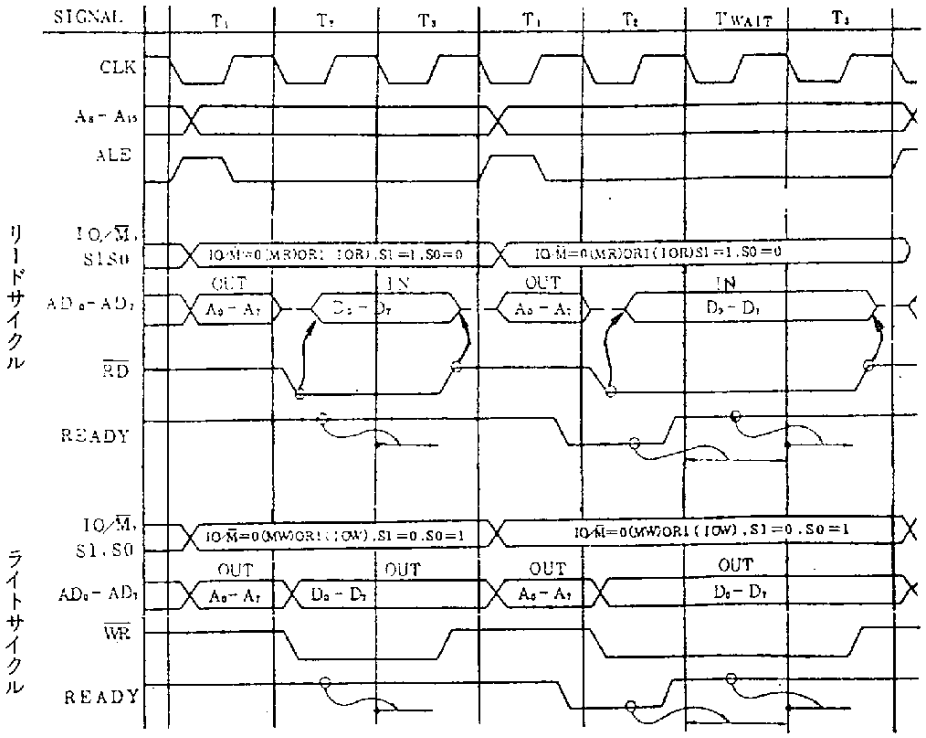


図２－６　リード・ライト動作のタイミング（８０８５）

(d)　マイクロコンピュータの種類と特徴

実際にマイクロコンピュータのハードウェアがどのような姿になるかはもちろんシステムの目的や規模によって異なる。

図２－７のように，マイクロプロセッサ，ＲＯＭ，ＲＡＭおよびＩ／Ｏポートが１個のＬＳＩチップ上に集積されたものは，ワンチップ・マイクロコンピュータと呼ばれる。どの程度の機能を集積できるかはＬＳＩ技術にかかっており，現状では，ＲＯＭ２Ｋバイト，ＲＡＭ128バイト，８ビットのＩ／Ｏのポート３個といった程度である（8049／8039の場合）。

メモリやＩ／Ｏを増設可能なものもあるが，１チップのＣＰＵに比べ機能が劣るのは避けられない。しかし，何よりも低価格で，他に多くのＩＣを必要としないから，機器に組込まれて専用コントローラの役割を果すことが多くなる。

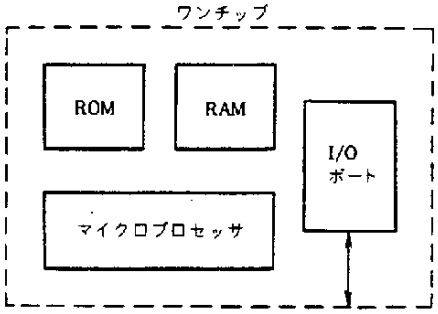


図２－７　ワンチップ・マイクロコンピュータ

これに対し，8085などＬＳＩのＣＰＵチップを使ったマイクロコンピュータは，数個～数100個のＩＣから構成することになる。小規模なものは，１枚の基板上にＣＰＵ，ＲＡＭ，Ｉ／Ｏポート（シリアル，パラレル）といった基本的な機能ブロックを搭載したワンボード・マイクロコンピュータという形をとる。メモリは１０数Ｋバイトとし，拡張性を考慮して基板外にバスを引出せるように設計するのが普通なので，システム設計はかなり自由になる。低価格でコンパクトにまとめられ，機器組込用として広く用いられている。

さらに拡張性や汎用性を重視するマイクロコンピュータでは，機能（モジュール）ごとに基板を分け，必要に応じて増設を行うという方法を採用する。基板間はマザーボード方式のバスによって結ばれ，実装のためのコストはかさむが，システム設計の自由度は最も大きい。

2.2　メモリの構成

現在，マイクロコンピュータシステムのプログラムメモリ（ＲＯＭ），データメモリ（ＲＡＭ）には，ほとんどの場合半導体メモリ（ＩＣメモリ）が用いられる。本節では，最初にこのＩＣメモリの種類や構造，特徴などに関して概略を記述する。次にＩＣメモリ素子の一般的な動作を説明し，

それらを使ったマイクロコンピュータのシステムメモリ構成法について述べる。

(a) ＲＯＭ，ＲＡＭの構造と特徴

まずＲＯＭでは，チップを製造する際のＩＣマスクによって記憶内容が決定され，それ以後の変更が不可能なマスク式ＲＯＭがある。集積度が高くビット当りの価格は低くなるものの，専用マスクを用いて製造するから，そのマスク製作に費用がかかり大量生産を行う場合以外は使用できない。

これに対し，ユーザ側で情報を書込むことが可能なＲＯＭをＰＲＯＭ（Programmable ROM）と呼ぶ。マスク式ＲＯＭより集積度が劣りビット当りの価格は高くなるが，応用に合わせて自由にプログラムを書込めるので，マイクロコンピュータの応用では広く使用されている。

ＰＲＯＭは，原理上１回に限り書込みを行えるヒューズ型ＲＯＭと，書込んだ内容を消去して何度でも書換えることができるＥＰＲＯＭ（Erasable ＰＲＯＭ）の２種類に分けられる。前者では，情報を記憶する単位（記憶セル）が例えば図２－８のようにヒューズのついたトランジスタから成っており，過電流でこのヒューズを切断するか否かによって情報を書込む。バイポーラ型であるから集積度は上がらないが，動作は高速である。

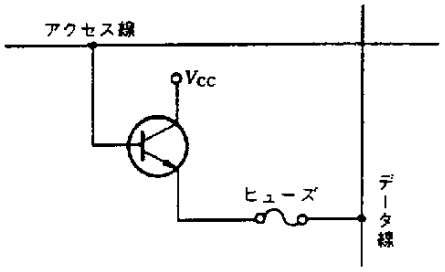


図２－８　ヒューズ形ＲＯＭの構造

ＥＰＲＯＭで現在最も多く使用されているのは，ＦＡＭＯＳ（Floating gate Avalanche injection ＭＯＳ）形と呼ばれるＲＯＭである。メモリセルは図２－９のような構造になっている。書込みの際，このソース，ドレイン間に高電圧（５０Ｖ程度）を印加してなだれ現象を

生じさせ，フローティングゲートに電子を蓄積する。電子が十分に蓄積された場合にはこのトランジスタがＯＮ状態となるので，フローティングゲートのチャージ状態によって情報が記憶されることになる。強力な紫外線を照射すると，フローティングゲートの電子が光電流として絶縁層を抜けて放出されるから，トランジスタはＯＦＦ状態に復帰し，消去可能というわけである。

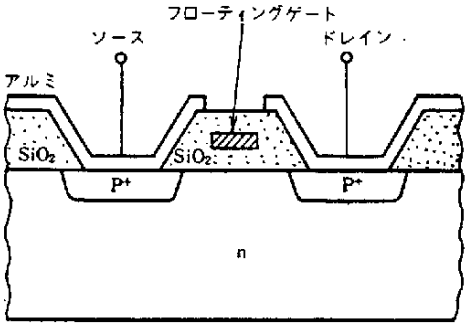


図２－９　ＦＡＭＯＳ形ＲＯＭの構造

一方，ＲＡＭはその構造からスタティック方式のものとダイナミック方式のものに分けられる。スタティックＲＡＭは，図２－１０に示すようなフリップフロップのメモリセルが集積されたものである。

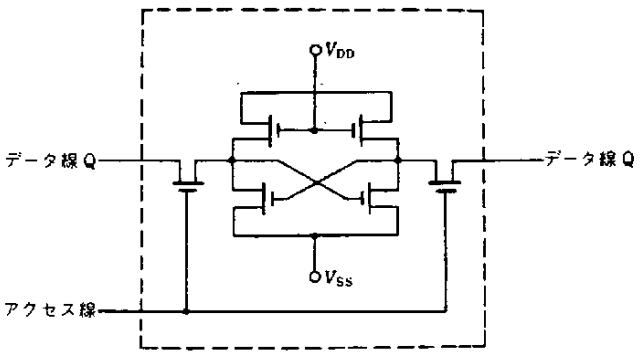


図２－１０　スタティックＲＡＭの構造

これに対し，図２－１１に示すようなＭＯＳトランジスタ３個から成るメモリセルを集積したものがダイナミックＲＡＭで，トランジスタのゲートに接続されたキャパシタンスが充電されているか否かによって情報を記

憶する仕組みになっている。

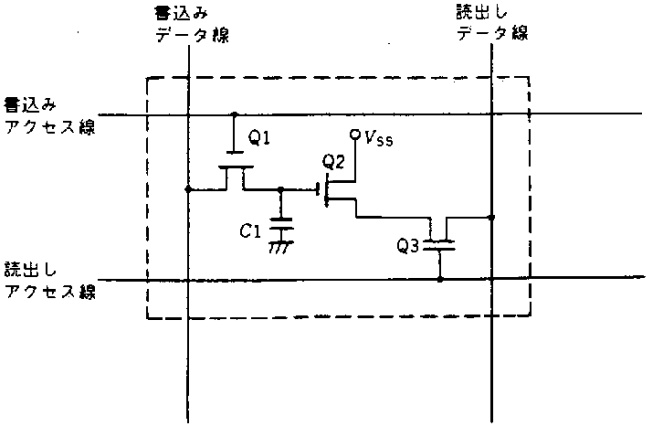


図２－１１　ダイナミックＲＡＭの構造

書込みにより蓄積した電荷は数１０ミリ秒後に放電し，記憶した情報を保てなくなってしまうので，周期的に再書込みを行って充電しなければならない。この動作をリフレッシュと呼んでいる。

ダイナミックＲＡＭは，リフレッシュを行うための回路が必要となる煩わしさはあるが，スタティック型に比べメモリセルを構成するトランジスタ数は半分で済むから集積度が高く，安価である。このため，大容量のメモリを必要とするシステムではダイナミック型がよく用いられるが，マイクロコンピュータでは使い易さの点で有利なスタティック型が多く使われている。しかし，Ｚ８０のようにメモリのリフレッシュ機能を内蔵したＣＰＵもあり，そうした場合にはダイナミックＲＡＭを簡単に使用することができる。

(b)　メモリ素子の構成と動作

メモリは，記憶セルのアレイから成る記憶部と，アドレスデコーダ，入出力制御部で構成される。実際のメモリでは，図２－１２に示すように記憶セルがマトリクス状に配列されていることが多い。入力されるアドレス信号を２つに分けてデコードし，Ｙ（行）デコーダによって１行の記憶セルが選択された後，その行中の１つがＸ，Ｙ両アドレスの交差点により記

憶セルを選択する方法が採用されるのは，記憶容量（記憶セルの数）が大きくなった時にアドレスデコード部が複雑になるのを抑えるためである。

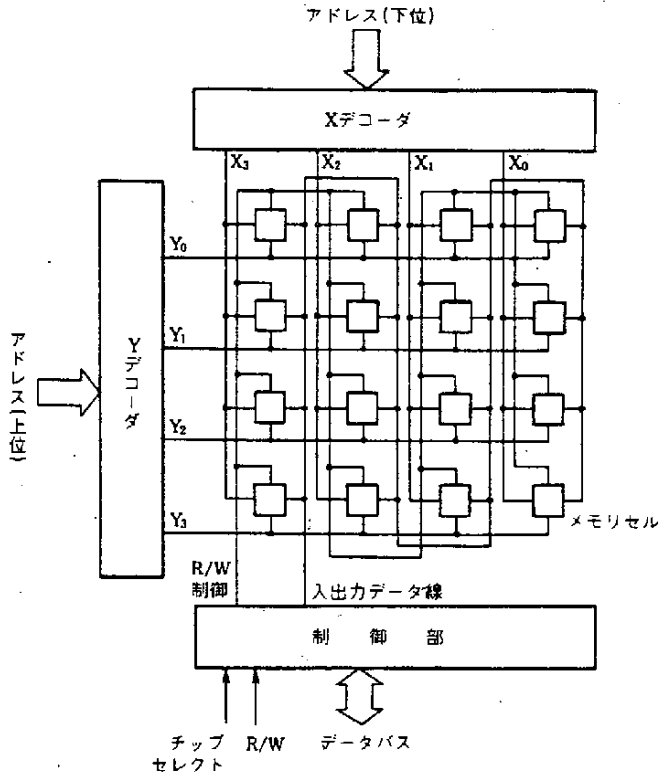


図２－１２　メモリチップの構成

選択された記憶セルでは，制御部からのコントロール信号に従って情報の入出力が行われる。ＲＡＭとＲＯＭは，基本的には図２－１３のような制御構成になっており，ＲＡＭにデータを書込むか読出すかはリードライト指定信号（Ｒ／Ｗ）によって定まる。書込みが指定された場合には入力ゲートを開いて記憶セルに書込みを行い，読出しが指令されれば出力バッファを開いて記憶内容を出力するわけである。ＲＯＭは読出し専用なので書込み回路は不要である。

チップセレクト信号（チップイネーブル信号）は，後述するようにシステムのメモリは多数のメモリチップから構成されることになるので，読出しまたは書込みをどのチップに対して行うか選択するために使用する。通常システムのアドレス信号の一部をデコードした信号をチップセレクト信号

に使用し，これによって選択されたチップのみが先に述べたデータの読出し，書込み動作を行う。

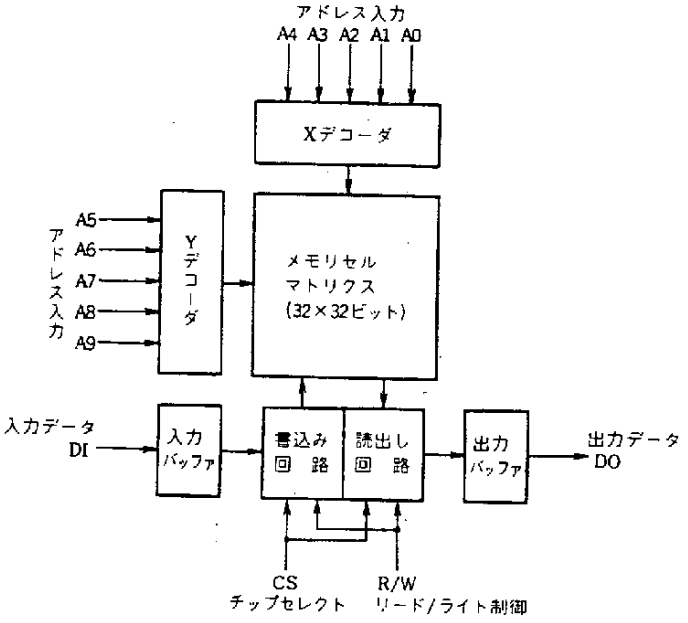


図２－１３　メモリの制御構成

(c)　システムメモリの構成

それでは，マイクロコンピュータのシステムメモリの構成法について説明しよう。

ＣＰＵが呼びだせるメモリ番地の範囲をアドレス空間と呼び，これと対応するようにメモリチップを組込むことによってシステムメモリが構成されることになる。

８ビットのＣＰＵは１６本のアドレス線をもっているので，そのアドレス空間は $2^{16} = 65{,}536$ バイトになる。しばしば，１０２４バイトを便宜上１Ｋバイトといい表わすが，それに従えば上記のアドレス空間は６４Ｋバイトと表現することができる。

今，２ＫバイトのＲＯＭと２５６バイトのＲＡＭを必要とするマイクロコンピュータのメモリ構成を考えてみよう。

ＣＰＵのアドレス空間は０番地からの６４Ｋバイトで，メモリはこの中

のどこに割付けても良い（アドレス空間を分割してメモリやＩ／Ｏポートを振分けることを割付けという）。ただし，ＣＰＵをリセットしたときプログラムカウンタが０に設定され，プログラムの実行が０番地から始まるので，０番地からの領域には通常ＲＯＭを割付ける。

ここでは図２－１４のようにＲＯＭを割付けることにしょう。

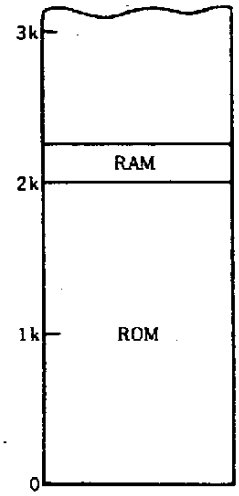


図２－１４　メモリの割付け

メモリチップの内部構成は１０２４×８ビットなどと表わされるが，これはチップ内の１つの番地が８個の記憶セル（８ビット＝１バイト）から成り，その番地が１０２４つまり１Ｋバイトあることを示している。したがって，１０２４×８のＲＯＭと，２５６×４のＲＡＭを用いて図２－１４のようなメモリシステムを構成するには，ＲＯＭチップ，ＲＡＭチップともに２個ずつ必要となる。その例を図２－１５に示す。ただし，ここではコントロール線は省略してある。

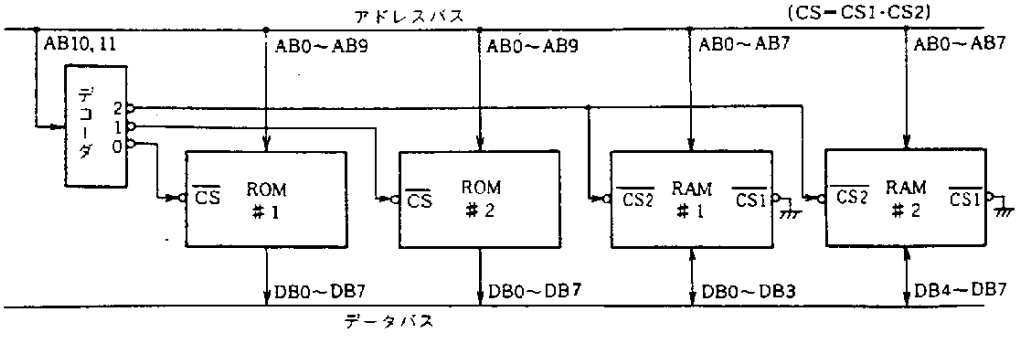


図２－１５　メモリの接続例

2.3 インタフェースの考え方

ＣＰＵはＩ／Ｏポートを通していろいろな装置や機器と情報の受渡しを行う。一般に入出力機器をＣＰＵのバスに接続するためのハードウェアと，それをコントロールするソフトウェアを含めてインタフェースと称している。

マイクロコンピュータシステムを設計する際の煩わしさは，このインタフェースをどのように構成するか，という問題に帰着することが多い。

とくに，リアルタイム処理に関する要求の大きいシステムでは，ハードウェアとソフトウェアの双方に跨るインタフェースの考え方が，システムの効率を左右することになる。ここでは，その一般的な構成，入出力の方法，といった基本的な考え方を述べる。さらに，割込みやＤＭＡなどと称する入出力制御方式について説明する。

(a) インタフェースの構成と動作

Ｉ／Ｏポートは，図２－１６のようにバス方式でＣＰＵに接続するのが普通である。ＣＰＵがどのＩ／Ｏポートとデータの入出力を行うかは，それぞれのＩ／Ｏポートごとに付けられた番号（アドレス）によって指定する。

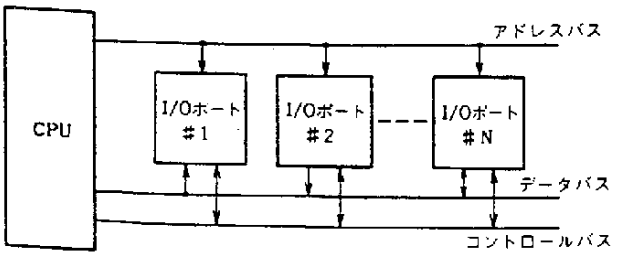


図２－１６　バス方式のＩ／Ｏ構成

通常Ｉ／Ｏポートのアドレスは図２－１７(a)に示すようなＩ／Ｏアドレス空間内のいずれかに割付けられる。この場合，８０８５の例では，入出力命令の２バイト目がＩ／Ｏポートのアドレスを示し，実行サイクルにおいてそれがアドレスバスの上半分下半分の８ビットに同じ値が出力されるので

0～255番までのI／Oポートを指定できることになる。

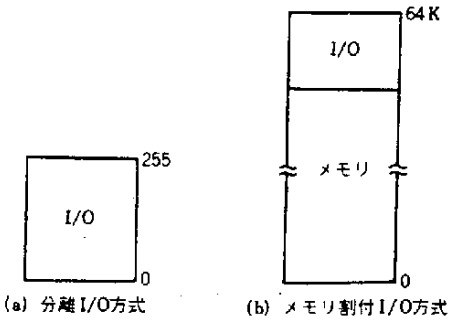


図2－17　I／Oのアドレス空間

I／Oポートの構成を図2－18に示す。

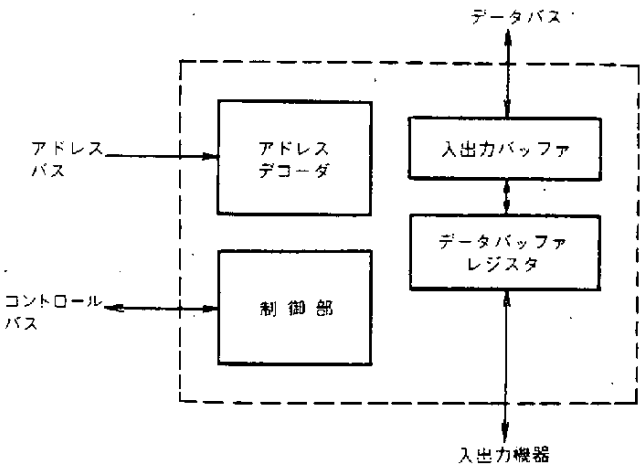


図2－18　I／Oポートの構成

I／Oポートはアドレスバスの信号をデコードする機能をもち，自分の番号が出力されている間に入力または出力を要求するコントロール信号が与えられれば，CPUとの間にデータを送受する。すなわち，入力ストローブ信号が与えられれば，ゲートを開いてバッファレジスタの内容をデータバスに送り出す。逆に出力ストローブ信号が与えられたときには，データバスの内容をバッファレジスタに取込む動作を行うわけである。

バッファレジスタの役割は，CPUと外部機器の一方から他方へデータを受渡す間の一時記憶を行うことになる。通常，両者は異なった速度で非同期に動作しており，同じ形式のデータを送受するとも限らないからである。

ところで，ＣＰＵとＩ／Ｏポートの間のデータ転送方法は，メモリに対するそれと基本的に異なった点はない。Ｉ／Ｏ専用の入出力命令が使用され，ハードウェア上は両者に対するコントロール信号が分けられているというにすぎないのである。したがって，メモリとＩ／Ｏをことさら区別する必要もなく，図２－１７(b)に示すようにＩ／Ｏポートをメモリのアドレス空間内に割付け，同じ扱いにすることも考えられる。このような方法をメモリ割付けＩ／Ｏ（memory mapped Ｉ／Ｏ）方式と呼ぶ。ＣＰＵによっては，Ｉ／Ｏがすべてメモリ割付けとなるものもある（6800など）。

(b) 入出力の方法

インタフェースのデータバッファを介してＣＰＵが入出力機器とデータを送受することは既に述べた。多くの入出力機器はＣＰＵとは独立，非同期に動作し，速度も１秒に数100Ｋバイトを転送する磁気テープやドラムのような高速のものから，機械的な構造のために１回の動作に何秒も要する低速なものまで，広い範囲に及んでいる。あるいは，キーボードやスイッチのように動作時間を予測できない場合もある。

このようにＣＰＵと速度差の大きな入出力機器との間にうまくデータを送受するためには，まず以下に述べるような方法が考えられる。入出力機器はデータを転送する準備や入出力動作が完了したとき，それをＣＰＵに知らせるためにインタフェース内部の状態フラグ（ステータス）をＯＮにする。

マイクロコンピュータでは，これを制御信号としてＣＰＵに伝えるのではなく，図２－１９のようにデータバスに接続されたインタフェース内部のステータス・レジスタにセットして，データ・レジスタと全く同様の方法でＣＰＵから読取れるようにするのが普通である。

したがってＣＰＵが入出力機器とデータを送受するには，図２－２０に示すようにまずステータス・レジスタの内容を入力し，状態フラグがＯＮ（入出力機器がready）であれば，データレジスタに対しデータの入出

力を実行すれば良い。また状態フラグがＯＮでなければ，入出力機器は準備や前回の動作が完了していないこと（ busy)を示すから，待ちループをまわって状態フラグがＯＮとなるのを待つか，あるいは他の処理に移ることになる。この方法では，プログラム（ ＩＮ命令，ＯＵＴ命令）によって入出力機器の状態を調べ，レディ状態を確認した後にデータの入出力を行うので，プログラムＩ／Ｏ（ programmed Ｉ／Ｏ）方式と呼んでいる。

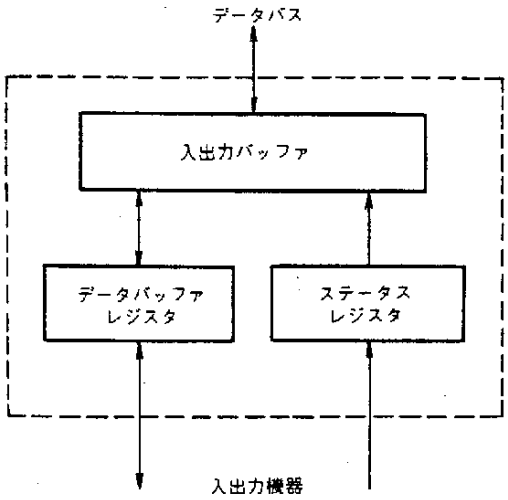


図２－１９　ステータスレジスタ

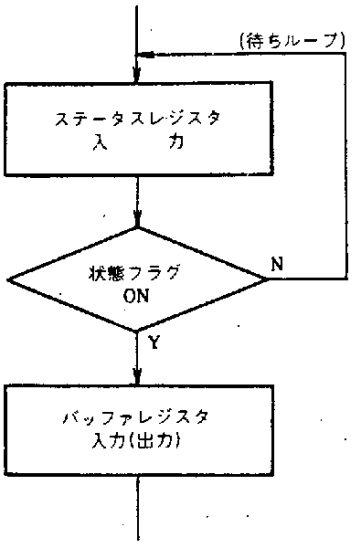


図２－２０　入出力の方法

プログラムＩ／Ｏは，インタフェースの構成やプログラムの上では簡単であるが，Ｉ／Ｏ処理の高速性が要求されるシステムにおいては，割込みやＤＭＡなどの方式を採用せねばならない場合も生ずる。これらの考え方については後述する。

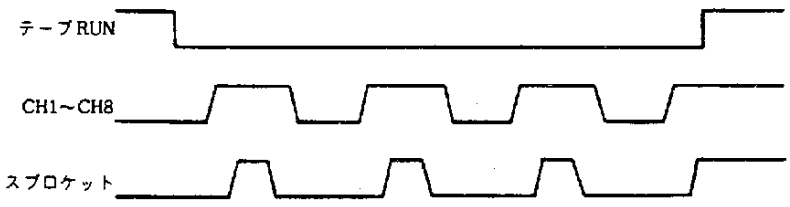


図２－２１　ＰＴＲの動作

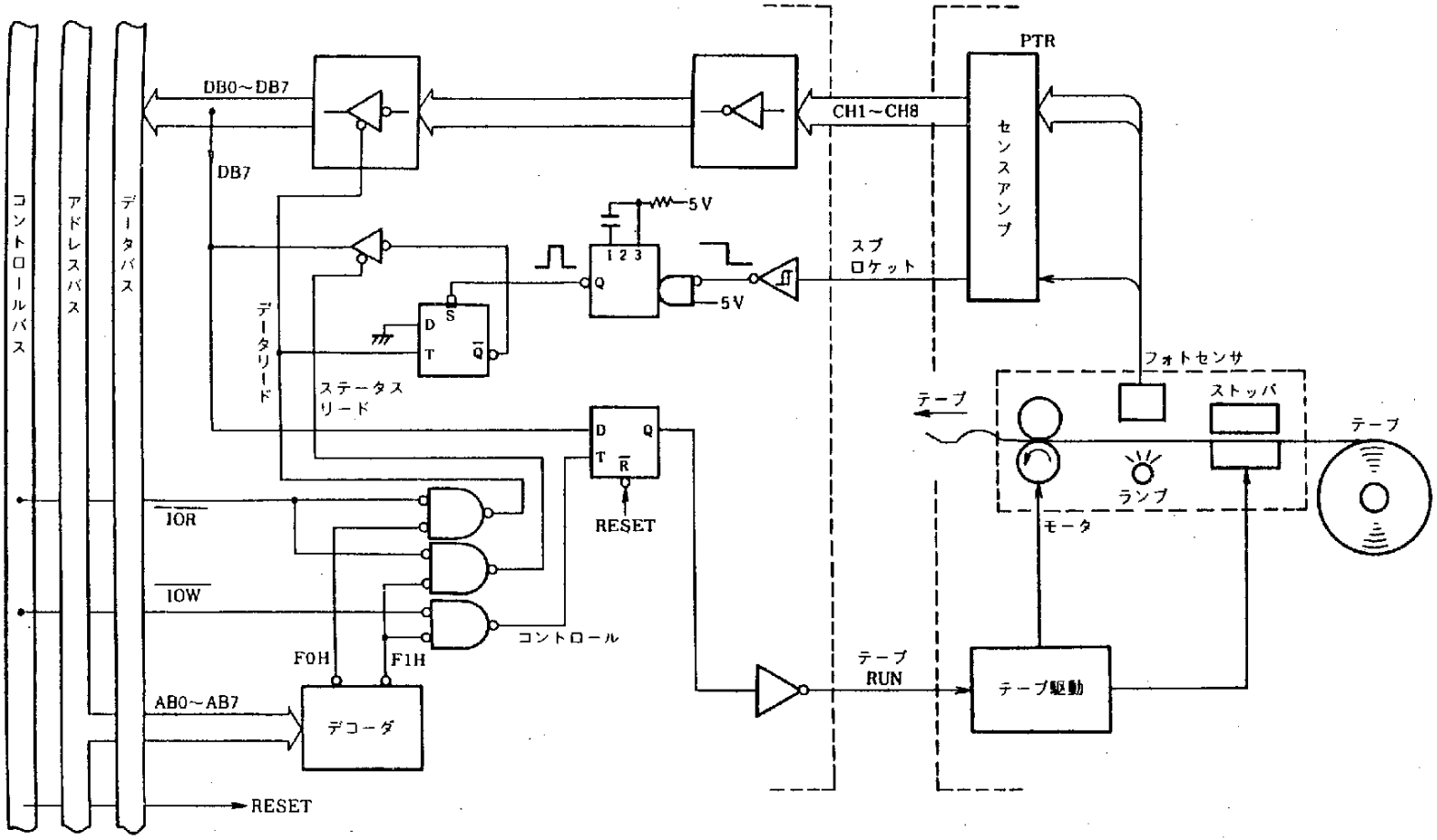
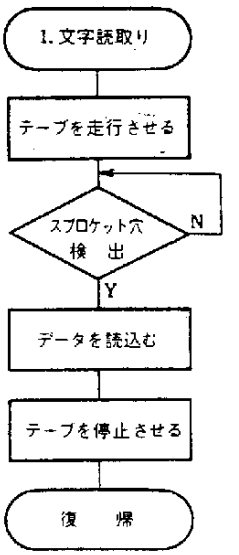


図2－22　PTRとのインタフェース例

ここで，プログラムやデータの入力装置としてよく利用される紙テープリーダ（ＰＴＲ）の例を取り上げて，インタフェースの方法を説明しょう。

一般に使用される紙テープはテープの駆動用のスプロケット孔ごとに８ビットのデータ（１文字）を表わすようになっており，データ各ビットは孔があいているとき１，孔があいていないとき０となる。図２－２１に示すように，ＰＴＲはテープ駆動信号ＲＵＮがlowのときテープを走行させ，光電検出されたデータ信号をコンピュータに送るものとしよう。これを図２－２２のような構成のインタフェース回路を用いて前に取り上げた8085システムに接続する。この例では，アドレスＦ０Ｈのポートがデータ入力用に，アドレスＦ１Ｈがテープ駆動コマンドの出力ポートおよびステータス入力ポートとして割付けられている。図２－２３は紙テープを走行させ，先に説明したプログラムＩ／Ｏ方式でデータを１文字読取るプログラムを示す。一般に，データバッファの入出力も，機器やインタフェースに対する動作指令（コマンド）やステータスの入出力も，ＣＰＵからは全く同じ扱い方がなされる。




```
PTRD: MVI   A, 80H
      OUT   0F1H
      IN    0F1H
      RLC
      JNC   $ -3
      IN    0F0H
      PUSH  PSW
      XRA   A
      OUT   0F1H
      POP   PSW
      RET
```

図２－２３　入出力制御例（ＰＴＲ）

(c)　並列転送と直列転送

ＰＴＲの例のように，並列のデータ信号が一度に転送される方式を並列転送（parallel transmission）と呼ぶ。これに対し，同じ紙テープをテレタイプを用いて読取る場合，テレタイプからコンピュータに送られてくるデータ信号は図２－２４(b)に示すように１ビットずつ順次送信されるパルス信号でそのような転送方式を直列転送（ serial transmission ）と呼んでいる。

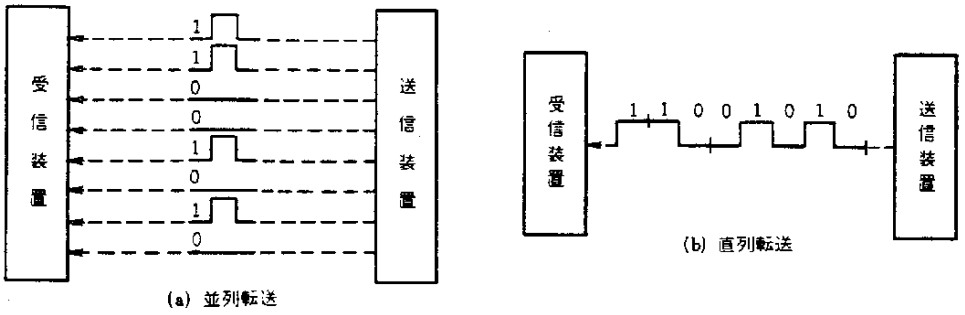


図２－２４　並列転送と直列転送

当然のことながら直列方式では転送速度が遅くなるが，信号線は１対で済むので，テレタイプなどの端末装置の他，モデム（変復調装置）を介して遠方へデータを伝送する場合など，通信システムにおいて広く用いられている。

直列転送の速度は，１秒間に伝送されるビット（エレメント）数にボー（ baud ）という単位を付して表わし，モデムでは1200ボー，2400ボー4800ボーなどの速度のものがよく用いられている。

マイクロコンピュータのバスは並列入出力であるから，直列方式の入出力機器やモデムを接続するためには，図２－２５に示すようにインタフェースでデータの直並列変換を行わなければならない。実際のインタフェースの構成は，直並列変換および種々の伝送フォーマットに従った制御機能をもつＵＳＡＲＴ（ Universal Synchronous／Asynchronous Receiver／Transmitter ）と呼ぶＬＳＩを利用して，簡単に行うことが多い。その例は８２５１の項で詳しく説明を行うことにする。

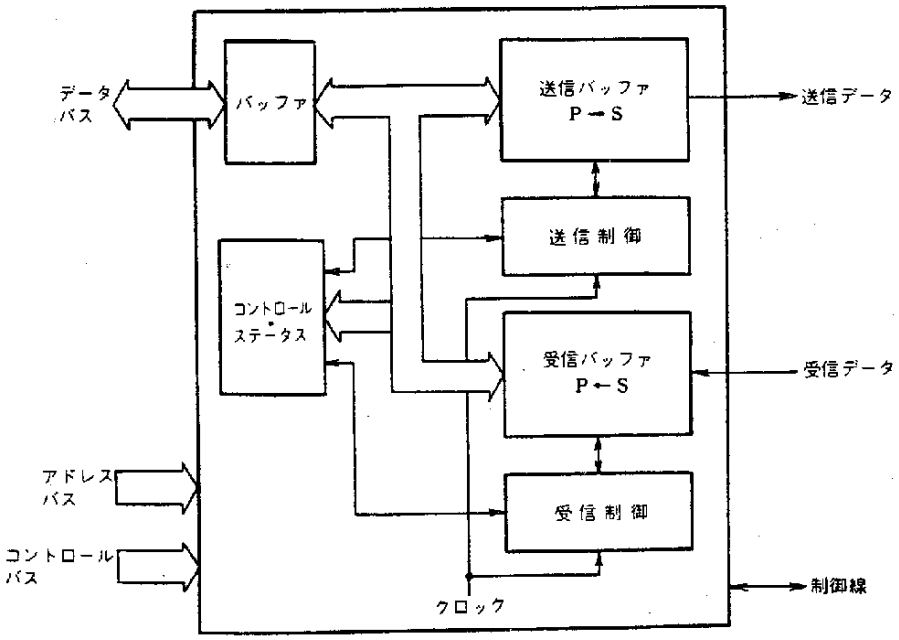


図２－２５　直並列変換インタフェース

2.4 割 込 み

(a) 割込みの考え方

プログラムＩ／Ｏ方式では，ＣＰＵは外部機器とデータを入出力するために，常にそのレディ，ビジィ状態を調べている必要がある。ＣＰＵは同時に２つ以上のプログラムを処理することはできないから，動作速度の遅い機器が接続された場合には，実際にデータの送受を行うわずかな時間を除く大半の期間が待ちループで無駄に消費されてしまう。もちろん，合間に他の処理を差しはさむことはできるが，そのプログラムを実行できる時間は限られることになる。もしも他の処理に多くの時間を費すようであれば，入出力機器が動作を完了しているにもかかわらず，ＣＰＵはなかなかそれを知ることができないわけであるから，機器に対する処理（サービス）が遅れる。数多くの入出力機器が接続されていたり，突発的に起る事象を監視するような場合も同様で，ＣＰＵが１つ１つ順番にそれぞれの状態を見ること（ポーリング）を繰返していたのでは，迅速なサービスが期待できなくなる。

このような場合，サービスの必要となった外部の入出力機器からその時点で直ちにＣＰＵに知らせる方法があれば，ＣＰＵはサービスを要求されるまで機器の状態を気にすることなく，他の処理に専念できるだろう。この方法が以下に説明する割込み（ Interrupt ）と呼ばれるもので，とくにリアルタイム性の強いプロセス制御の分野に応用されることが多いマイクロコンピュータやミニコンピュータのシステムでは，頻繁に用いられるＩ／Ｏ制御方式である。

割込み方式では，図２－２６に示すように，主導権を与えられたＩ／ＯがＣＰＵを制御することにより入出力ルーチンが実行され，複数の並行したプログラムを効率良く切分けて処理を進めることになる。これまでにも述べてきたように

① Ｉ／Ｏの動作が完了したり，状態が変ったりして，ＣＰＵによるサービ

スが必要となった場合

② オペレータの操作など，突発的な事象によるプログラムの変更

③ システムのハードウェアの異常や停電が生じた場合

などに対しては，割込みを用いることが多い。他にも，プログラム実行中に未定義の命令やアドレスが検出されたり，演算結果が異常となった場合にも同様の概念で割込み処理で対処することがある。

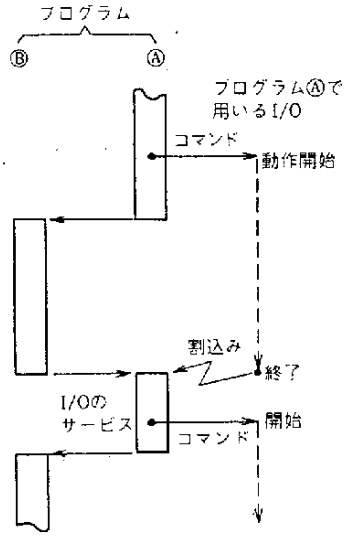


図２－２６　割込みによる処理の例

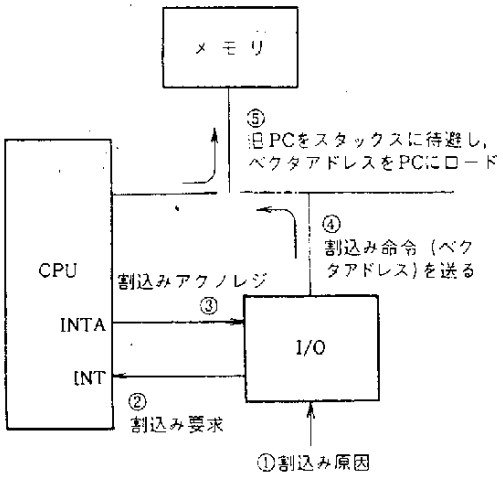


図２－２７　割込みの基本動作

割込みの動作や処理の方法はＣＰＵによってそれぞれ特徴をもっているが，図２－２７を参照してその基本動作を説明しよう。Ｉ／Ｏは先に述べたような理由で割込要求信号（ＩＮＴＲ）を出力し，自らに対する処理ルーチンの実行をＣＰＵに要求する。

ＣＰＵがこの信号を検出し，割込み要求がなされていることを知ると，進行中のプログラムを中断して，割込みに対する応答サイクルを開始する。

このサイクルでは，ＣＰＵは割込みを行ったＩ／Ｏに対し，そのサービスルーチンの開始番地を与えるように要求する。この開始番地のことを，ベクタ（Vector）アドレスと呼んでいる。ベクタアドレスとして，サービスルーチンの開始番地を直接に示す方法の他，開始番地へのポインタの存

在する番地を示す方法もある。ただし，8085では後者のようなベクタアドレスの与え方はできない（Ｚ８０などのＣＰＵでは可能）。割込みＩ／Ｏからベクタアドレスが与えられると，ＣＰＵは今まで実行中であったプログラムカウンタ（ＰＣ）の内容をスタックにプッシュした後，新たな実行開始番地としてベクタアドレスをＰＣにセットしてＩ／Ｏサービスルーチンの実行を開始する。割込みのサービスルーチンを終了して復帰命令が実行されると，スタックにプッシュされていた以前のＰＣの値がロード（ポップ）され，中断されていた元のプログラムの状態に戻ることができる。

ところで，システムによって割込みの要因は１つであるとは限らないので，その要因ごとにいくつかの割込み要求ラインに分け，それぞれに応じた処理ルーチンを用意しておくことが多い。それらを割込みレベルと呼んでいる。複数の割込みレベルをもつシステムでは，同時に２つ以上の割込要求が発生する可能性がある。それに備え，どの割込みレベルから先に受付けるのかという優先順位（ priority ）を，あらかじめハードウェア上で決めておくのが普通である。例えば，停電には最も緊急に対処する必要があるし，オペレータが重要な操作を行うような場合には，それらの割込みレベルの優先度を高くしておくのが普通である。割込みレベルやその優先順位の設定方法は，ＣＰＵやシステムの個性によって決められる。ノンマスカブル割込み（Non Maskable Interrupt，命令で後に述べるようなマスクをすることができない割込み）の入力ラインが用意されているＣＰＵにおいては，通常それに最も高い優先度が与えられている。

システムにおいて優先度を設定するには，次のような方法が考えられる。

その１つは，図２－２８のように多数の割込みラインをプライオリティ・エンコーダを介してＣＰＵに接続する方法で，同時に複数の割込み要求が生じた場合，例えばレベル０に接続されたＩ／Ｏの優先度が最も高く，レベル７が最も低くなる。その具体的な例については後に触れることにする。

もう１つの方法は，図２－２９に示すようにイモズル式（ Daisy Chain）

にＩ／Ｏを接続し，ＣＰＵの近くに接続されたＩ／Ｏほど優先度を高くするものである。

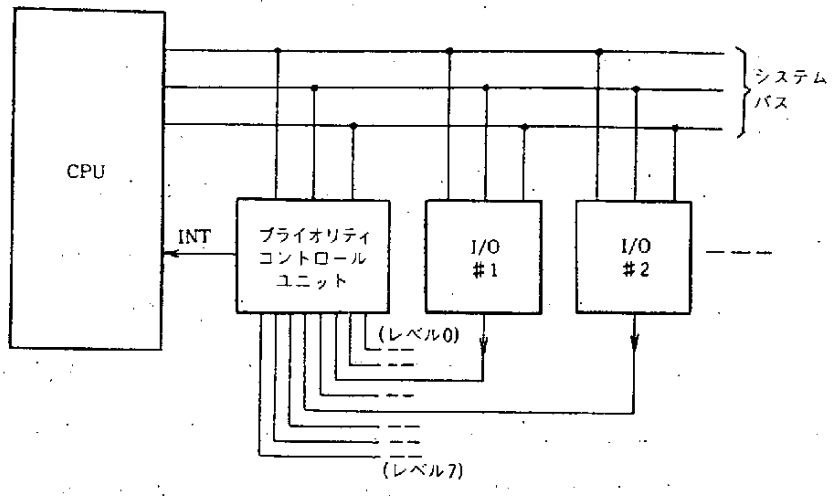


図２－２８　マルチラインの割込み方式

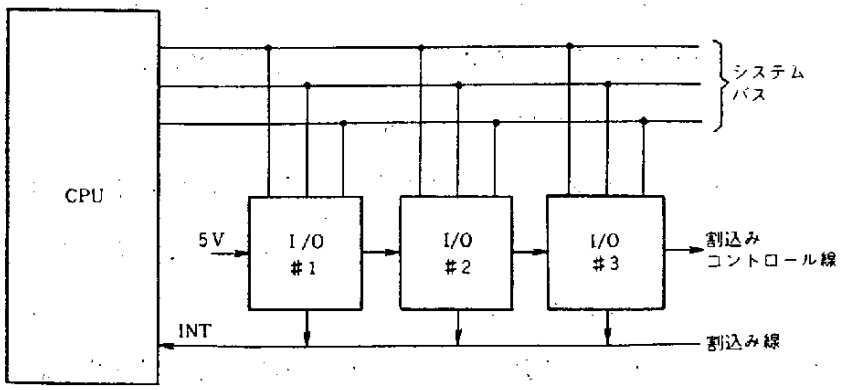


図２－２９　デージィチェーンの割込み方式

このデージィ・チェーン方式では，各Ｉ／Ｏは割込コントロール線によって直列につながれ，このラインの入力がhighのＩ／ＯポートだけがｃｐｕにＩＮＴ信号を送れるようにしておく。ＣＰＵに近い側のＩ／Ｏに割込みの必要が生じたとき，そのＩ／Ｏから割込みコントロール線の出力をlowとし，それが最後のＩ／Ｏまで順番に伝えられるようになっていれば，ＣＰＵに近い側のＩ／Ｏほど優先的に割込みを行えることになる。

システムに多くの割込み要因がある場合，優先順位を設定するとともにプログラムでそれぞれの割込み要求を制御する必要が生ずる。このため，割込み要求を行う各レベルまたは要因の側に，図２－３０のような割込み

マスク回路を設けておくことが多い。ＣＰＵがある割込みを受付けたくない場合には，その割込みのマスクをセットして禁止したり，保留しておいたりすることができる。

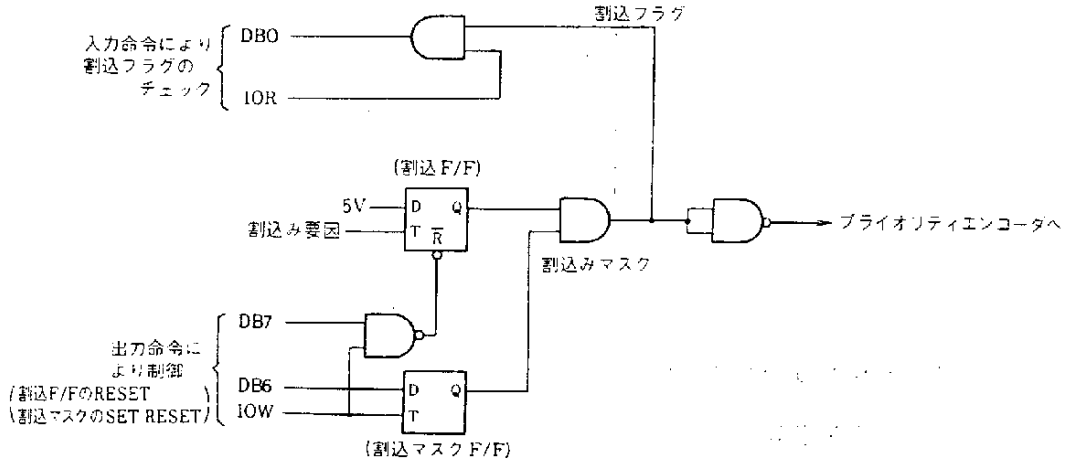


図２－３０　割込みマスク回路

この図で，割込フラグの役割は，同じ割込レベルに多くの割込要因を接続した場合にどのＩ／Ｏからの割込であるかを調べるために用意されたものである。

同時に発生した割込み要求が優先順位に従って受付けられることは前に述べたが，１つの割込みが受付けられた後，その処理プログラムを実行中に，他の割込み要求が行われることがある。そうした場合，図２－３１のような多重割込みを受入れるかどうかは，割込み処理ルーチン中での割込みマスクの制御次第である。実行中の割込み処理をそれによって中断されたくない割込みに対しては，マスクをセットしてその割込み要求が出力されないようにしておけば良いし，緊急度のより高い割込みは許すことになる。すなわち，処理プログラムを実行する優先度は，プログラムによって決められることに注意する必要がある。

さて，これまでに割込みはハードウェア上の優先順位とプログラムによる制御によって受付けられること，またその処理は割込みを行ったＩ／Ｏからベクタアドレスを ＣＰＵに送ることで開始されることを説明してきた。

そこで，8085での割込みの実際的な例を取上げてみよう。

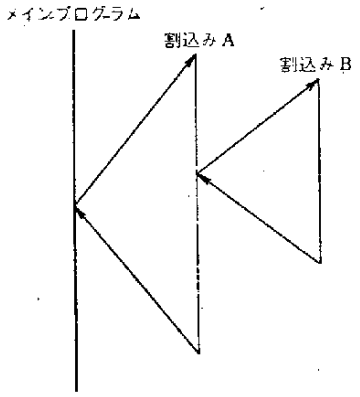


図2－31　多重割込み

(b)　8085の割込み制御

i)　INTR

8085では図2－32に示すように5本の割込み線が用意されている。先ず，INTRについて説明する。この割込み方式はZ80においても許されている。

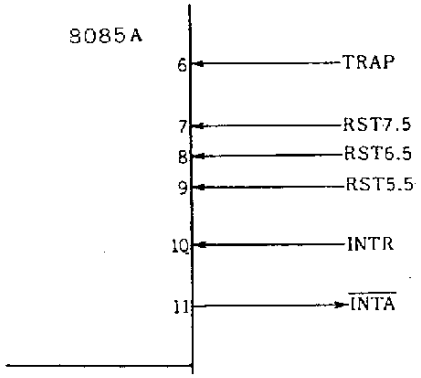


図2－32　8085の割込みに関係した端子

8085は図2－33に示すように各命令の最後でINTR線の状態をサンプルしている。あらかじめEI命令が実行されてCPU内部の割込みイネブル(INTE)F／Fがセットされており，かつINTR線がアクティブになっていれば割込みを受入れる。8085内のINTR入力回路を図2－34に示す。

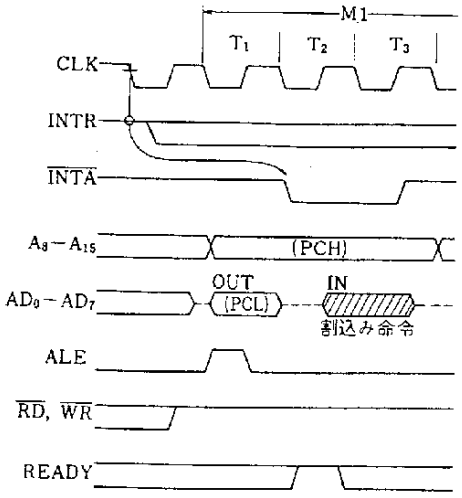


図２－３３　８０８５の割込み認識サイクル

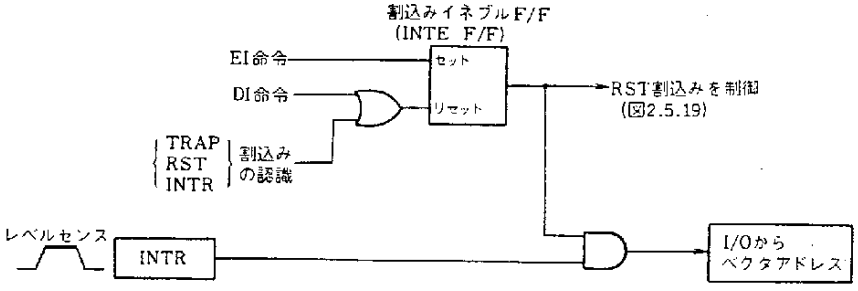


図２－３４　割込みイネブルＦ／ＦとＩＮＴＲ入力線

ＣＰＵが割込みを受入れるとＩＮＴＥ　Ｆ／Ｆを自動的にリセットし（以後，ＥＩ命令が実行されるまで割込みは受付けない），割込み応答サイクルを開始する。

通常のサイクルがプログラムカウンタ（ＰＣ）の示すメモリから次に実行すべき命令のオペコードをフェッチしてくることで始まるのに対し，割込み応答サイクルではＩＮＴＡ信号を出力してＩ／Ｏに実行すべき命令（割込み命令）を与えるように要求する。この間ＰＣのインクリメントはされない。

割込みに対して８０８５が自動的に行うのは以上に述べたＩＮＴＥ　Ｆ／Ｆのリセット，割込み応答サイクルの開始，これだけである。

したがって割込みを行ったＩ／Ｏは，この特別のＭ１サイクルでどん

な命令のオペコードを与えても良い。しかし通常の割込みの目的からは，今のＰＣの値を保存し割込み処理ルーチンのベクタアドレスを与える命令を与える必要があるだろう。このためにＰＳＴ命令やＣＡＬＬ命令が選ばれる。ＰＳＴ命令は８つの飛先アドレスを１バイトのオペコードから発生させられるから，最も簡単な割込み命令としてよく用いられる。

Ｉ／ＯからＩＮＴＡ信号に合わせてＲＳＴ命令のコードをデータバス上に送り出せば，ＣＰＵはそれを解読・実行し，与えられた番地へジャンプしてそこからのプログラムの実行を開始する。以前のＰＣの値がスタックへプッシュされるのはＲＳＴ命令の機能である。

ＣＡＬＬ命令を用いれば任意のベクタアドレスを与えることができるが，割込応答サイクルでＣＡＬＬ命令のオペコードに引続き２バイトのアドレスをＩ／Ｏから与えなければならない。いずれにせよ，割込みによって正常なプログラムの流れを強制的に中断し，Ｉ／Ｏから命令を与えて処理ルーチンへジャンプさせるにすぎないのである。またそこまでがハードウェアの役割になる。

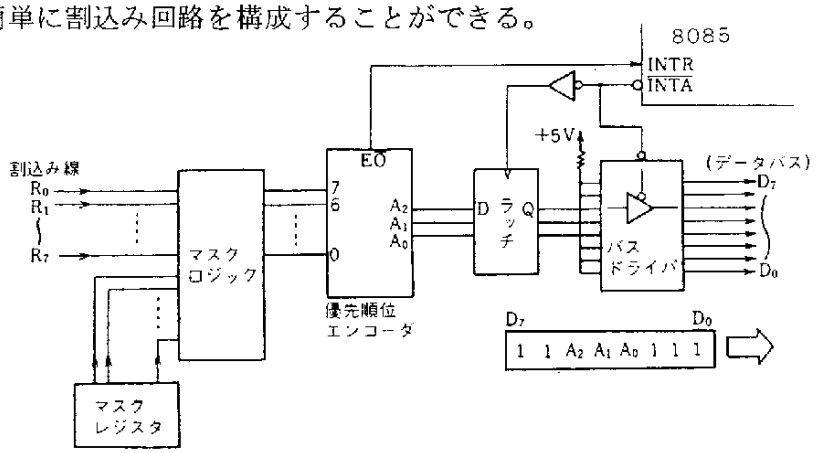
図２－３５は８レベルの割込み要求からそれぞれ個別のアドレスへＲＳＴ命令によりジャンプさせる回路例である。メモリ上の各飛先番地には通常そのレベルの割込み処理ルーチンへのＪＵＭＰ命令が書込まれることになるだろう。

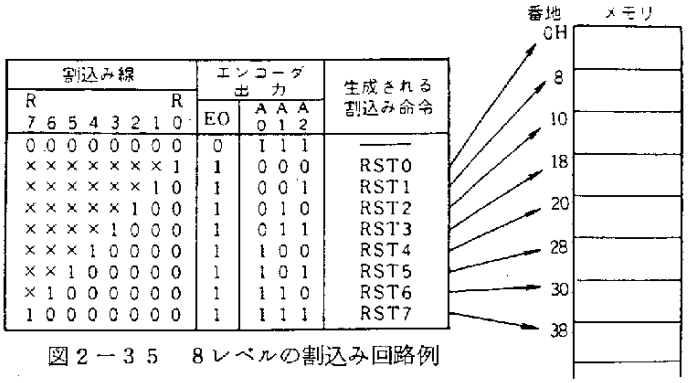
同図では割込みレベルごとのマスク回路を介して割込み線が優先順位エンコーダに接続されており，割込みレベル０（R_0）が最高優先度になっている。

ii） ＲＳＴ７.５，ＲＳＴ６.５，ＲＳＴ５.５　（ＲＳＴ割込み）

ＲＳＴ７.５，ＲＳＴ６.５，ＲＳＴ５.５の３本の割込みは，受付けられるとＣＰＵ内で自動的にＲＳＴ型命令が生成される。つまり図２－３５に示したハードウェアがすべてＣＰＵチップに内蔵されているわけで，Ｉ／Ｏからベクタアドレスを与える必要がないから小規模なシステムではこれだ

けで簡単に割込み回路を構成することができる。

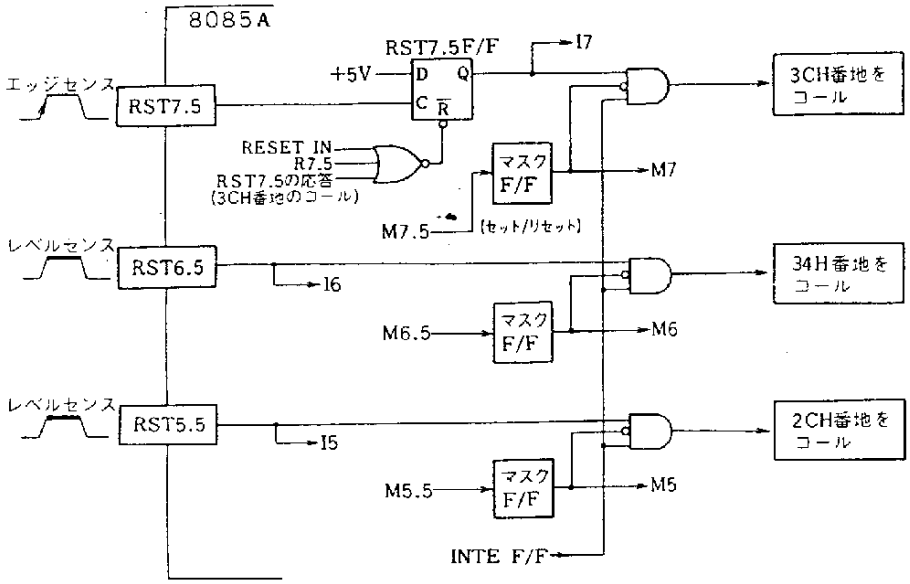




割込み線 R7	R6	R5	R4	R3	R2	R1	R0	エンコーダ出力 EO	A0	A1	A2	生成される割込み命令
0	0	0	0	0	0	0	0	0	1	1	1	―
×	×	×	×	×	×	×	1	1	0	0	0	RST0
×	×	×	×	×	×	1	0	1	0	0	1	RST1
×	×	×	×	×	1	0	0	1	0	1	0	RST2
×	×	×	×	1	0	0	0	1	0	1	1	RST3
×	×	×	1	0	0	0	0	1	1	0	0	RST4
×	×	1	0	0	0	0	0	1	1	0	1	RST5
×	1	0	0	0	0	0	0	1	1	1	0	RST6
1	0	0	0	0	0	0	0	1	1	1	1	RST7

図２－３５　８レベルの割込み回路例

ＲＳＴ5.5,ＲＳＴ6.5,ＲＳＴ7.5の飛先アドレスはそれぞれ２ＣＨ,３４Ｈ，３ＣＨでちょうどＲＳＴ5,ＲＳＴ6,ＲＳＴ７の中間になっている。ＲＳＴ割込みのＣＰＵ内のロジックを図２－３６に示す。



(注) R7.5, M7.5, M7, I7などはRIM命令，SIM命令を参照のこと

図２－３６　ＲＳＴ割込みの内部ロジック

ＲＳＴ割込みにはＩＮＴＲよりも高い優先度が与えられており，３つの中ではＲＳＴ7.5が最高優先で以下ＲＳＴ6.5,ＲＳＴ5.5の順になっている。またＩＮＴＲと同様ＣＰＵ内のＩＮＴＥ　Ｆ／Ｆによって制御されるマスカブル割込みであるが，さらに各ラインごとに割込みマスクが付けられている。これらのマスクはＳＩＭ命令により制御することができる。図２－３７参照。

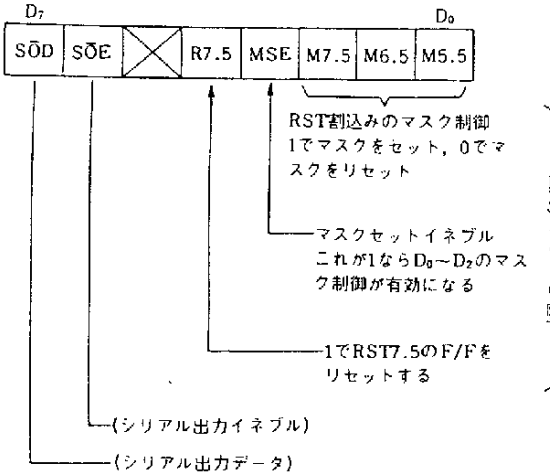


図２－３７　ＳＩＭ命令によるＲＳＴ割込み制御

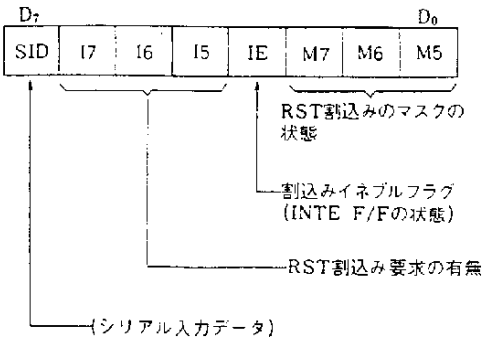


図２－３８　ＲＩＭ命令実行後のアキュムレータの内容

ＲＳＴ6.5,ＲＳＴ5.5に入力される割込要求信号はＩＮＴＲと同様highレベルのときＣＰＵに認識されるから，ＣＰＵがこれらを受入れて割込処理ルーチン中で確認されるまでhighに保たねばならない。これに対しＲＳＴ7.5のみは図２－３６に見られる通り信号の立上りエッジで内部的にラッチされるので，要求信号は単一パルスでもかまわない。ＲＳＴ7.5のラッチは割込みが受入れられてベクタアドレス（ＢＣＨ）へジャンプしたとき，Ｒ7.5を１としてＳＩＭ命令が実行されたとき，ＣＰＵがリセットされたとき，のいずれかでリセットされる。

ＲＳＴ7.5Ｆ／Ｆの状態やＲＳＴ6.5,ＲＳＴ5.5入力線の状態は，ＲＩＭ命令を実行することで図２－３８に示すようにアキュムレータへロードすることができる。ＲＳＴ割込みのマスクの状態もまたＲＩＭ命令により読み取ることができる。ＳＩＭ命令とＲＩＭ命令は8080にはなく，8085

でＲＳＴ割込みとシリアル入出力線が新たに設けられたことにより追加された命令である。

８０８５はこの２つの命令を除いて８０８０とソフトコンパチブルになっている。

iii） ＴＲＡＰ

ＴＲＡＰはプログラムでマスクできない最高優先度の割込みで，普通はハードウェアの異常や停電など非常事態のサービスのために用いる。ＩＮＴＥ　Ｆ／Ｆがリセットされていても受入れられるからＤＩ命令もこの割込みに対しては無効である。

ＴＲＡＰが受付けられるとＲＳＴ割込みと同様，ＲＳＴ型のベクタアドレスがＣＰＵ内で自動的に生成され，２６Ｈ番地（ＲＳＴ 4.5と言える）へジャンプする。

ＴＲＡＰの入力ロジックは図２－３９に示すようになっており，割込み信号がhighに立上りかつhighレベルを最小１００ns保つという条件でＣＰＵに認識される。ＴＲＡＰを受付けてからそのサービスの実行中に，ＴＲＡＰ入力のノイズなどによって再び割込みを受付けてしまうような誤動作を防ぐために，一度割込みが受付けられるとその後ＴＲＡＰ入力がlowに落ち着き再び認識条件を満足するまでは受付けられないようになっている。

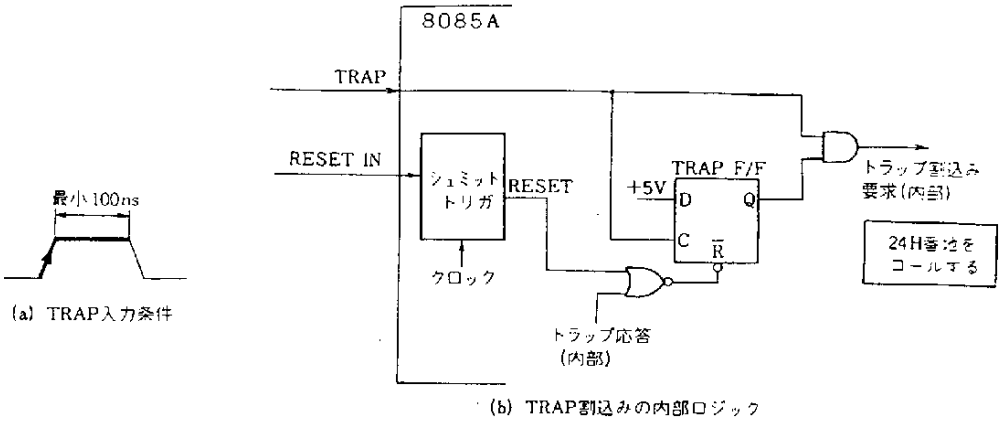


図２－３９　ＴＲＡＰ割込み

ＴＲＡＰ割込みが受付けられるとＩＮＴＥ　Ｆ／Ｆは自動的にリセットされる。そして割込みルーチンから復帰に当りＩＮＴＥ　Ｆ／Ｆの状態を自動的に復元する機能を持った命令が用意されている。

そこで，８０８５ではＴＲＡＰが受付けられた後も以前のＩＮＴＥ　Ｆ／Ｆの状態が保存され，ＲＩＭ命令を実行したときＩＥビットにこれを表現するようになっている。ただし以前の状態を読み出せるのはＴＲＡＰ受け付け後の１回のＲＩＭ命令だけである。ＴＲＡＰ割込みから復帰するときは，ＲＩＭ命令で以前のＩＮＴＥ　Ｆ／Ｆの状態を読み出し，それを元通りの状態に戻してから復帰しなければならない。

最後に，ＩＮＴＲ，ＲＳＴ，ＴＲＡＰの各割込みの特性を図２－４０にまとめた。

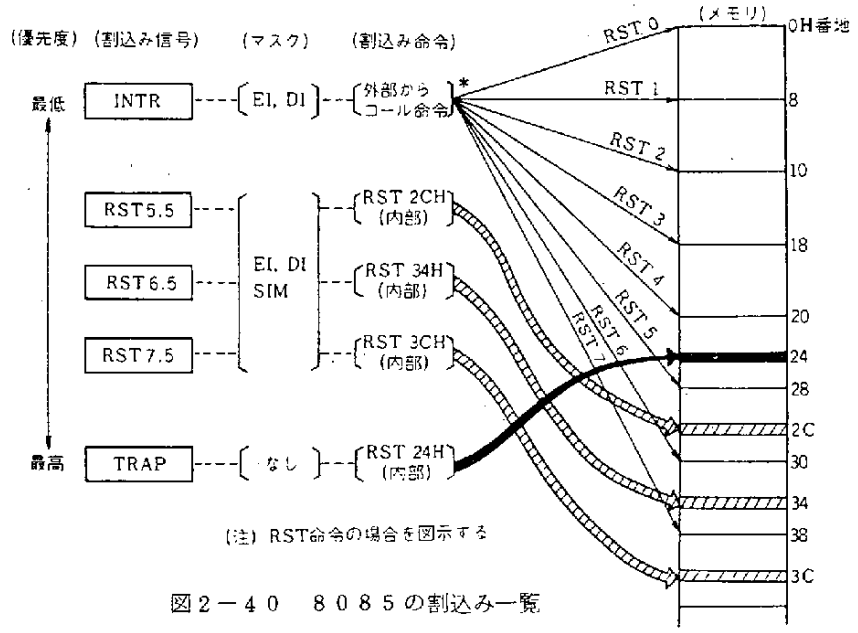


図２－４０　８０８５の割込み一覧

(c)　割込み処理プログラムの基本

割込み制御のプログラムはそのシステムのソフトウェアの土台となるべきもので，システム設計の思想にまで発展する複雑な問題を多くはらんでいる。しかしながらその糸口を見つけ出すことすらかなりの準備を要することである。ここでは割込み処理ルーチンを構成する最も基本的な規則を簡単に説明するにとどめよう。

割込み処理プログラムの基本型は図２－４１のようになる。割込み前のプログラムカウンタの退避，割込処理ルーチンの先頭へのジャンプまではハードウェアの役割であった。割込み処理のプログラムはＣＰＵのアキュムレータ，フラグ類（プロセッサステータスワード，ＰＳＷ）の退避，つまりスタックへプッシュすることに始まる。通常のプログラムでアキュムレータを操作しないことはまず考えられないから，その保存は割込み前に実行中であったプログラムを再開するために最低必要な条件である。次に割込み処理の中で使用する汎用レジスタを同じ理由で退避する。

こうした手続きはサブルーチンと同じである。割込みルーチンで注意すべきことは，プログラムでサブルーチンを呼び出す場合と異り，それがいつどのような条件で呼び出されるか予想できない点である。

次に，１つの割込みレベルに多数の割込み要因が接続されている場合には，Ｉ／Ｏごとの割込みフラグを読出していずれが割込み要求を発しているのかを解析する。１つのレベルに１つの割込み要因しかなければ，ベクタアドレスによって割込み要因は自明だからこの必要はない。

割込み要因が判別できればそれに対する処理も明かとなり，その処理中にもより優先度の高い割込みは受付けられるように割込みマスクを操作する。

現在の割込みが受け入れられたときにＩＮＴＥ　Ｆ／ＦはリセットされているからＥＩ命令を実行する必要がある。ただこうした多重割込みを許可するに先立ち，現在の割込みを引き起こした割込み要因はリセットするかマスクしておかねばならない。さもないと再び割込みが受け付けられてしまうからである。

高速な応答が必要な割込みＩ／Ｏが他にある場合には，それ以外の割込みが受け付けられてから上述の多重割込みを許可するまでの割込みディセーブルの期間を短くするように設計する必要がある。優先度の高い割込みがなければ割込み処理の途中で割込みマスクを操作しなくとも，割込みルーチンから復帰する命令の直前でＥＩ命令を実行すればよい。ＥＩ状態にして復帰

しないとそれ以後割込みはすべて受付けられなくなる。

割込み要因に対するサービスの実行は必要最小限にとどめ，データの一時記憶（バッファ），フラグなどプログラム上の伝達方法によりメインルーチンに処理を引継ぐのが原則である。システムの中での各処理の関連を明かにし易いばかりでなく，Ｉ／Ｏが高速になればそれに追従できなくなるからである。

サービスの実行後は割込みマスクの状態を復元し，スタックへ退避した汎用レジスタやＰＳＷを元通りに（ポップ）して割込みルーチンから復帰する。プッシュしたレジスタ群は逆の順番ですべてポップしておくことはもちろんである。

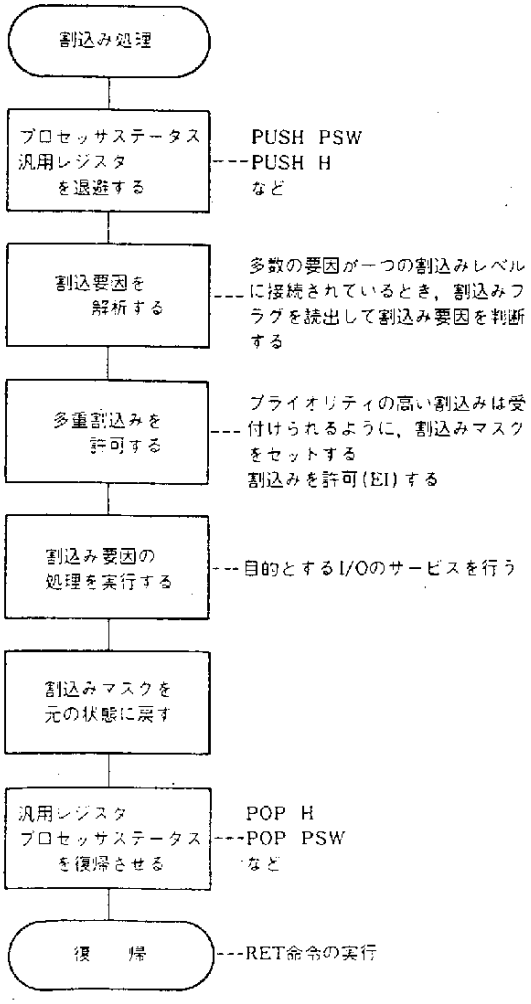


図２－４１　割込みプログラムの基本形

このように割込み処理プログラムでは本来の目的である割込み要因に対するサービスの実行の他に，多くの前・後処理が必要である。割込みの頻度が増えてくるとこうした目的以外の処理（オーバヘッド）に要する時間が馬鹿にならなくなってくる。Ｉ／Ｏに対するシステムの応答性を高めるにはこのオーバヘッド時間を減らすことが重要である。

2.5 ＤＭＡ（Direct Memory Access）

これまでにプログラムＩ／Ｏ，割込みＩ／Ｏという２つの入出力制御の方法を説明した。これらではいずれもプログラムされた入出力命令をＣＰＵが実行することによりデータを転送する。

ＣＰＵでデータ転送を制御する際に問題となるのは，１つのデータを数１０～数100 μ秒程の時間内に処理しなければならなくなってくると転送プログラムの実行時間が無視できなくなることである。そのような高速な機器の動作に何とか遅れないでデータ転送を制御できたとしても，その間ＣＰＵは転送処理に専念せざるを得なくなる。とくに多量のデータを転送する場合には著しくシステムの効率が低下するであろう。

さらに高速な機器であればプログラム制御は不可能ともなる。

以下に説明するＤＭＡ（Direct Memory Access）と呼ばれる方式はＣＰＵを介さないで直接Ｉ／Ｏとメモリ間でデータ転送を行うもので，プログラム処理での速度の制約を解決し，システムの効率を極度に高めようとする技術である。（図２－４２）

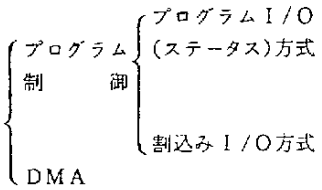


図２－４２　Ｉ／Ｏ機器とのデータ転送

図２－４３はＤＭＡの特性と他の方法を比較した例である。

	１つのデータ転送を処理する時間	１秒のうち転送の制御に関わる時間	ＣＰＵの全処理に占める割合
プログラムＩ／Ｏ	数10μs ＋(レディ待ち時間)	１秒の大半	ほとんど100％
割込みI/O	約100μs	0.1	１０％
Ｄ　Ｍ　Ａ	１～２μs	２ｍ秒	0.2％

図２－４３　１秒間に１Ｋバイトのデータ転送する場合のＣＰＵの処理効率

(a)　ＤＭＡの考え方

Ｉ／Ｏ機器との間のあるまとまった量のデータ転送（ブロック転送と呼ぶ）をプログラムで制御する場合，一般に図２－４４に示すようになるだろう。バッファ領域のアドレスと転送バイト数を予め設定しておきＩ／Ｏ機器の動作をスタートさせる。機器の送受信準備が整うごとに入出力プログラムを実行し，メモリとの間でデータを転送する。機器のレディ状態をＣＰＵが調べながらこの入出力プログラムを起動するのがプログラムＩ／Ｏ方式，機器が割込み信号を発して入出力プログラムの起動を促がすのが割込みＩ／Ｏ方式であった。

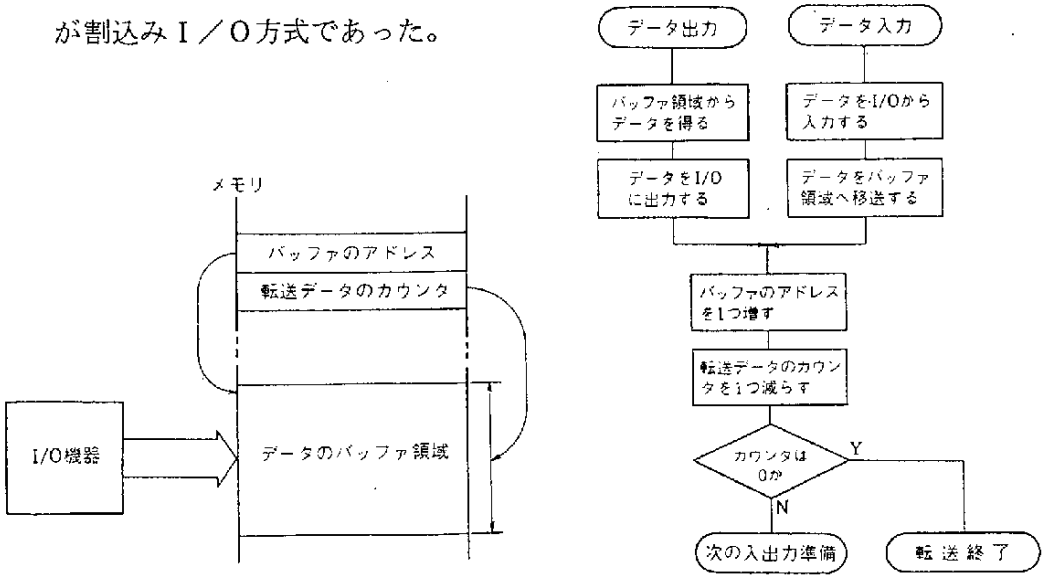


図２－４４　ブロック転送の制御

図２－４４に示すようにＩ／Ｏ機器とメモリ間のデータ転送のみを行うにしても，ＣＰＵは１バイトごと（転送サイクル）に少くも１０ステップほどの命令群を実行する必要があり，２０～３０μ秒よりも速い転送サイクルを制御することはできない。

ＤＭＡは図２－４４と同じ処理を専用のハードウェアによって制御する。プログラムを介在させることなくＩ／Ｏ機器とシステムメモリ間にデータを直接転送するから，極めて高度な入出力制御が可能になる。

マイクロコンピュータの補助記憶装置として馴染み深いＩ／Ｏにフロッピーディスクやハードディスク，磁気テープ，磁気ドラムなどがある。これらは１バイト当り１～１０数μ秒でデータを読書きする必要があり，ＤＭＡ方式のインタフェースによって接続されるのが普通である。

また図２－４５に示すようなデータ通信の制御や，アナログ現象を高速にサンプリングしそれをディジタル値に変換して収集するような計測・制御分野の応用において，ＤＭＡ転送がよく利用される。

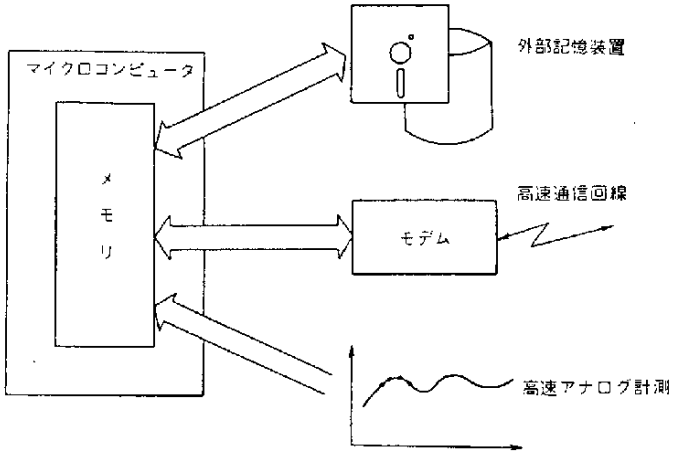


図２－４５　高速Ｉ／Ｏ機器の例

次にＤＭＡの一般的な実現方法について説明する。先にＤＭＡは専用のハードウェア（ＤＭＡコントローラと呼ぶ）がＩ／Ｏとメモリ間のデータ転送を制御すると述べたが，そのためには転送の間システムのバスとメモリをＤＭＡコントローラの支配下に置かなければならない。通常バスとメ

モリはＣＰＵが使用しているので，ＤＭＡ転送を行う際に両者がぶつからないように制御する必要がある。この目的でＣＰＵにＨＯＬＤ信号が用意されている。

図２－４６はＤＭＡ方式の概略のシステム構成を示す。ＤＭＡコントローラはその内部に図２－４７のような制御用レジスタを持っており，前に図２－４４で述べた機能に対応している。

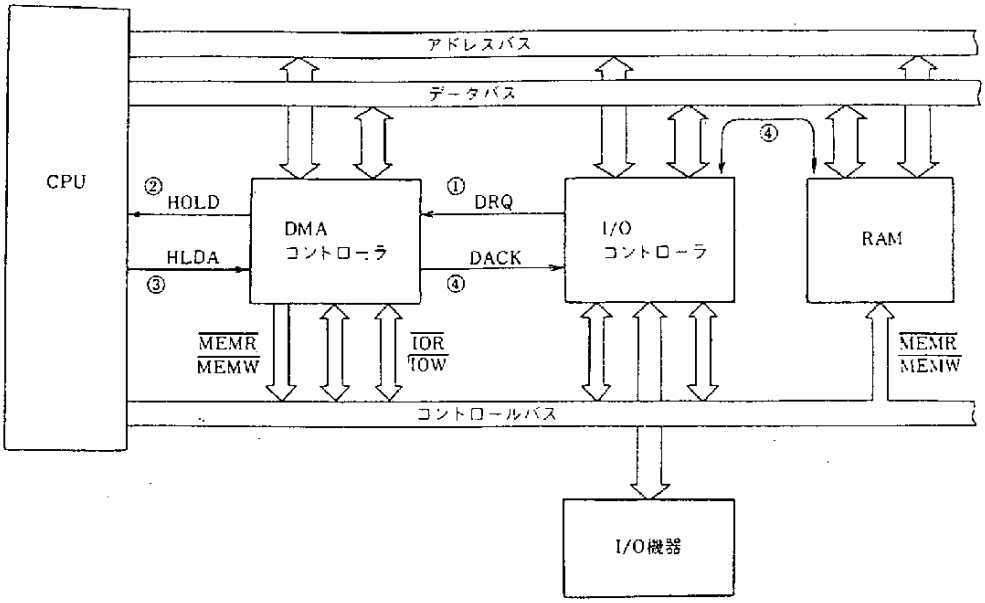


図２－４６　ＤＭＡの構成と基本動作

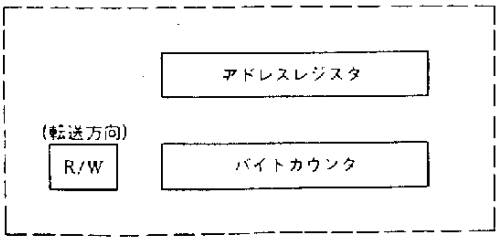


図２－４７　ＤＭＡコントロールに必要なレジスタ

つまりアドレスレジスタはデータを転送すべきメモリのアドレスを，バイトカウンタは転送すべきデータ数をそれぞれ保持している。またＲ／Ｗフラグはメモリからデータを読出してＩ／Ｏへ出力する（Ｒ）のか，Ｉ／Ｏからデータを入力してメモリへ書込む（Ｗ）のかを指示する。したがってＤＭＡ転送を開始するに先立ってＣＰＵからこれらのレジスタにＤＭＡ

開始アドレス，転送バイト数，転送方向を書き込んでおかなくてはならない。然る後Ｉ／Ｏコントローラに機器の動作を開始させるコマンドを出力する。

以下図２－４６を参照しながら説明を進める。

① Ｉ／Ｏ機器がスタートしデータ転送の準備が整うとＩ／ＯコントローラはＤＭＡコントローラにＤＭＡ要求（ＤＲＱ）を発する。

② ＤＭＡコントローラはＤＲＱを受信するとＣＰＵにホールド要求（ＨＯＬＤ）を行う。

③ ＨＯＬＤを受けたＣＰＵは自らシステムバスを電気的に切離す。つまりアドレス，データ，コントロール線の端子をハイインピーダンスとするように作られている。これによってシステムバスやメモリ，Ｉ／Ｏを使用する権利はそれらの制御機能を持った他のコントローラ（マスタと称する）に移される。同時にＣＰＵはホールド状態に入ったことを示す確認信号（ＨＬＤＡ）を出力する。

④ ＣＰＵがホールド状態になったことを確認したＤＭＡコントローラはアドレスにアドレスレジスタの内容を，またＩ／ＯコントローラにＤＭＡアクノレジ信号（DACK）を出力する。そして指定された転送方向に従ってメモリ，Ｉ／Ｏそれぞれにリード（$\overline{\text{MEMR}}$または$\overline{\text{IOR}}$）とライト（$\overline{\text{IOW}}$ $\overline{\text{MEMW}}$）コントロール信号を発する。

⑤ １バイトのデータ転送が終了するとアドレスレジスタの内容に１を加え，バイトカウンタを１つ減らす。バイトカウンタの値が０になるまで同様にしてＤＭＡ転送が行われる。

⑥ 転送が完了するとＤＭＡコントローラはホールド要求を解除し，システムバスは再びＣＰＵとつながる。ＣＰＵはホールド状態に入った後停止していた処理を再開する。

さて以上に説明したようなＤＭＡの転送サイクルは１～２μ秒であり，それと同程度の速度のＩ／Ｏであれば図２－４８に示すように１ブロックの

データを連続して転送してしまうことができる。これをバースト（burst）モードのDMAと呼ぶ。同図中でTC（ターミナルカウント）はバイトカウントが0になったこと，つまり最後のデータを示す信号である。

このようにDMA転送はCPUの動きとは無関係に進められるので，一連の転送が終了するまでCPUはそれに関与する必要がない。

DMAコントローラからの割込みやステータスによって転送終了を知り，必要なデータの処理に移ればよい。

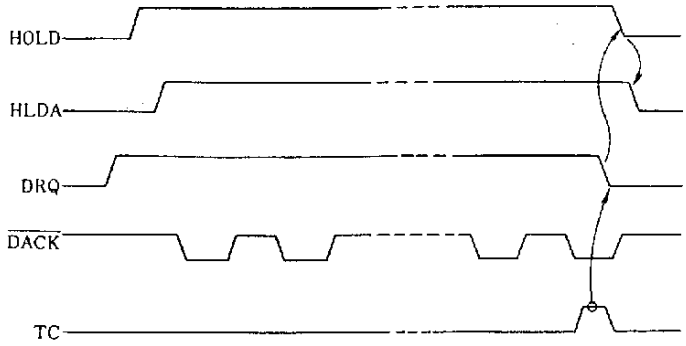


図2－48　バーストモード（ブロック転送）DMA

それに対して1バイト転送するごとにCPUのホールドを解除し，小きざみに転送を行う方法を一般にサイクルスチール（cycle steal）モードと呼んでいる。I／Oの速度にもよるが，CPUがDMAとは関係なく進めているプログラムの実行に与える影響は少なくなる。また複数のI／O機器がそれぞれ別のDMAチャネルに接続されて動作している場合，それらのDMA転送を並行して進めることができる。

厳密な意味でのサイクルスチールはまさしくCPUのサイクルを盗み取ってDMA転送を行うものであろう。通常，命令の解読や演算の実行サイクルでは処理がCPU内部でなされ，CPUはシステムバスを使用しないからこの期間を利用して1度に少しずつDMA転送を行う。制御は複雑になるがCPUの動作にはまったく影響を与えず，システムの効率は最高になる。

DMAコントローラは普通のCPUの周辺LSIとして用意されている

ことが多く，それらを用いればDMAの制御部は簡単に構成できる。

(b) 8085とDMA

8085にコンパチブルなDMAコントローラとして40ピンのLSI，8257－5が用意されている。基本的な機能は2.5 (a)の中の①～⑥で説明した通りの一般的なものである。8257固有の主な特徴を上げると次のようになる。

① 4つのDMAチャネルをもっている。

② 4つのチャネルの優先順位を固定かロータリ式のいずれかにプログラムできる。ロータリ式とはDMAサイクルごとに順番に優先順位が移り変わる方式である。

③ バイトカウンタが0になった後はそのチャネルを自動的にDMAディスエイブルとすることができる。

④ チャネル2だけは指定により1ブロックの転送のたびにアドレスレジスタやバイトカウントの初期値が自動的に再ロードされ，連続してブロック転送できるようになっている。

②以下は操作上の問題であるからここでは触れないこととし，基本的なことがらのみを述べる。

図2－49に8085と8257の接続例を示す。アドレスバスが複雑になっているのは8257がDMA制御時にメモリアドレスの上位半分をデータバスを介して出力するためである。1バイトごとの転送サイクルの最初にデータバスに出力されるアドレス上位をラッチしてメモリアドレスを構成する必要がある。

8275内のレジスタは図2－50に示す通りである。DMA転送を行うには目的のチャネルのアドレスレジスタとターミナルカウントレジスタ（バイトカウントとR／Wフラグ）の制御に必要なデータを書込み，そのチャネルをイネブルにする。チャネルの転送終了はステータスレジスタに表示される。

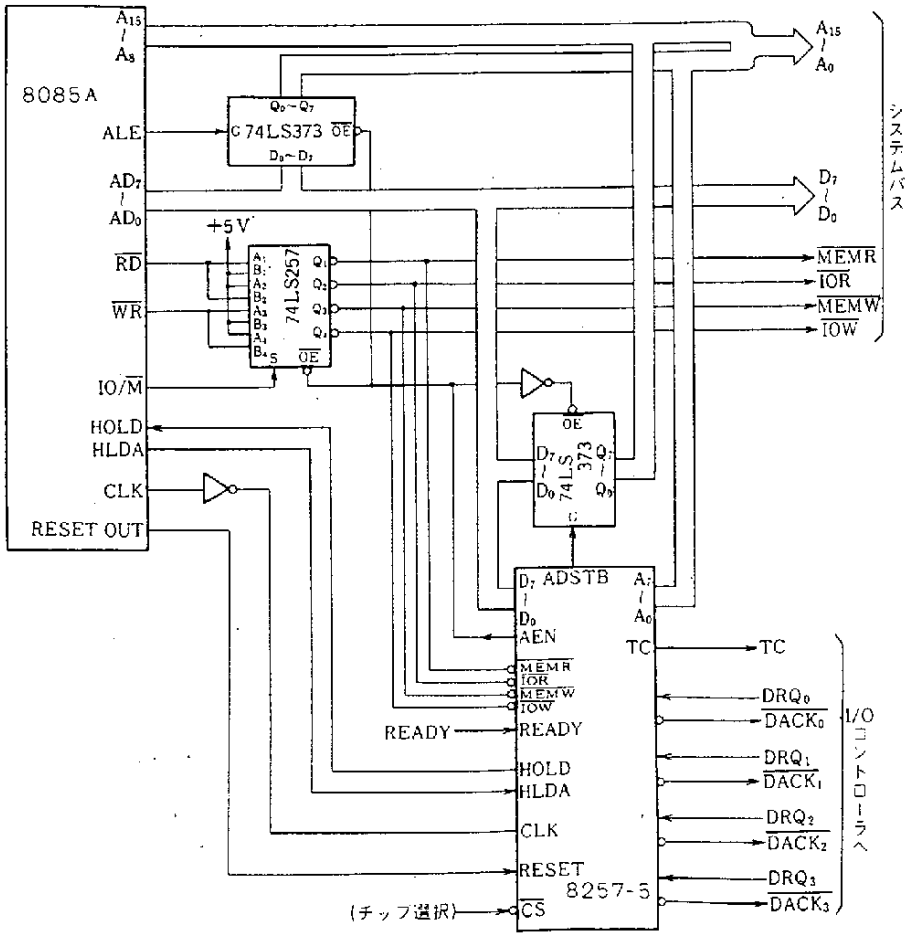


図２－４９　８０８５とＤＭＡコントローラ８２５７の接続

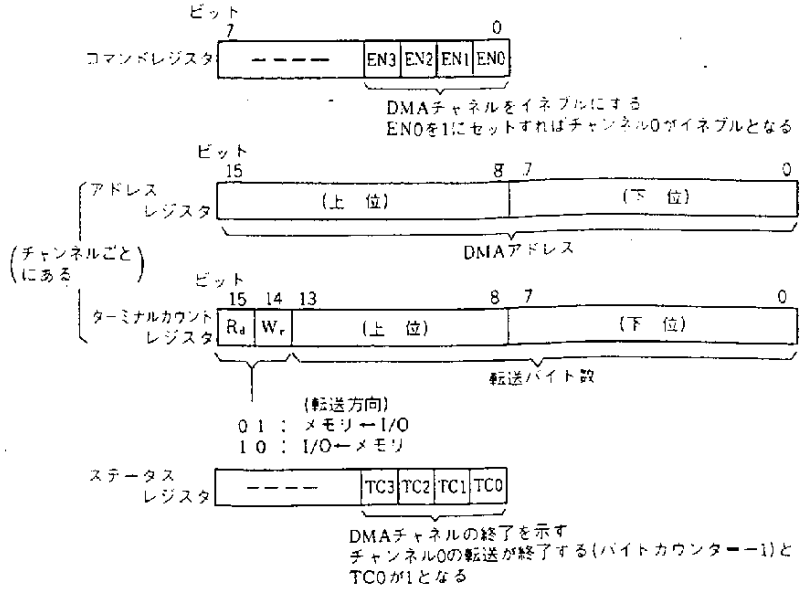


図２－５０　８２５７内のレジスタ

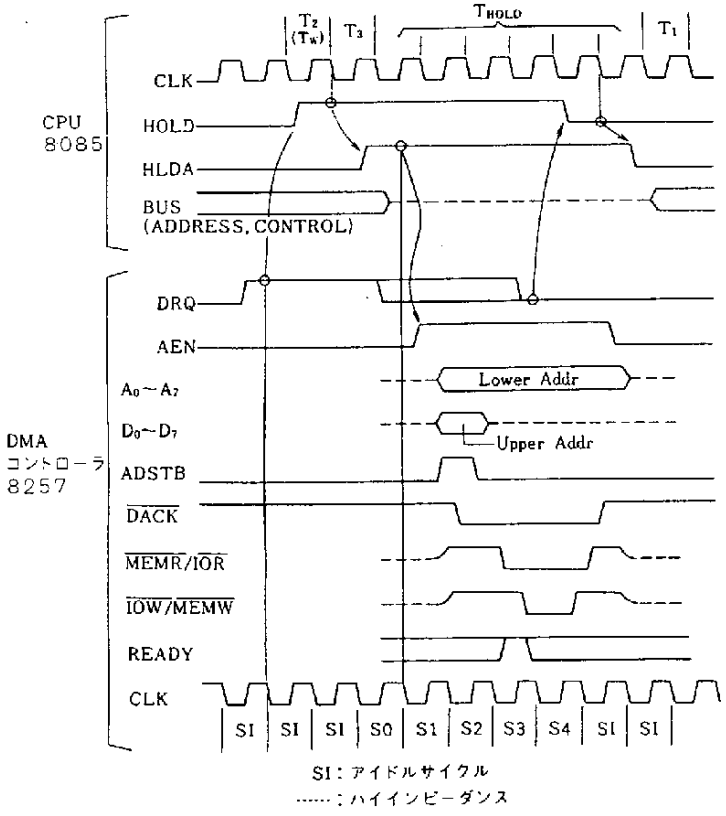


図２－５１　ＤＭＡコントロールのタイミング例

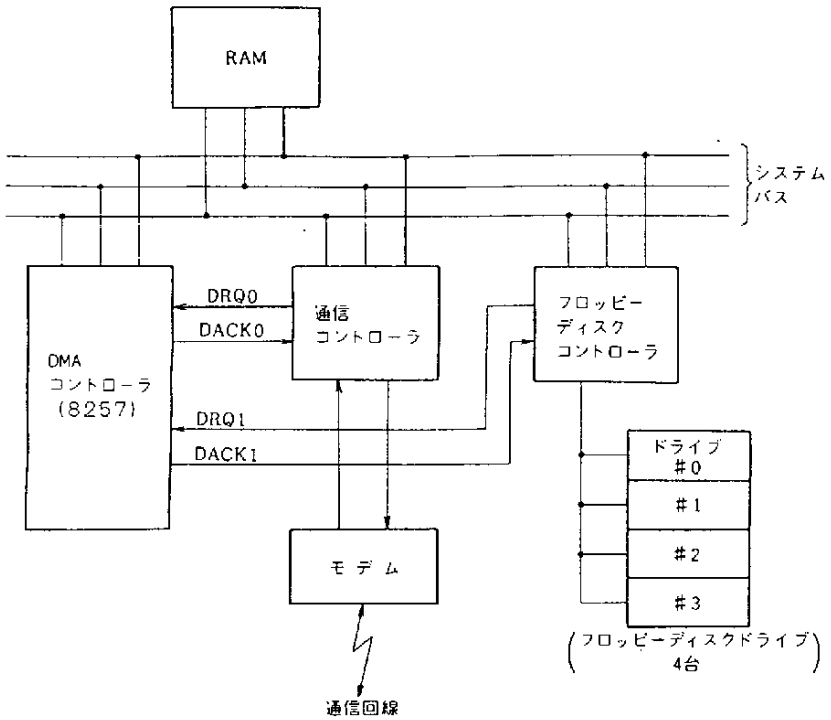


図２－５２　ＤＭＡの応用例

図２－５１には８０８５と８２５７のDMA制御のタイミングを示す。８０８５は各マシンサイクルのT_2またはT_Wサイクルが終了した時点でホールド要求をサンプルする。８０８５と８２５７間にシステムの制御機能が移行する様子が理解されるだろう。(a) 中の①～⑥を参照されたい。

最後にDMAの応用の一例を図２－５２に示した。

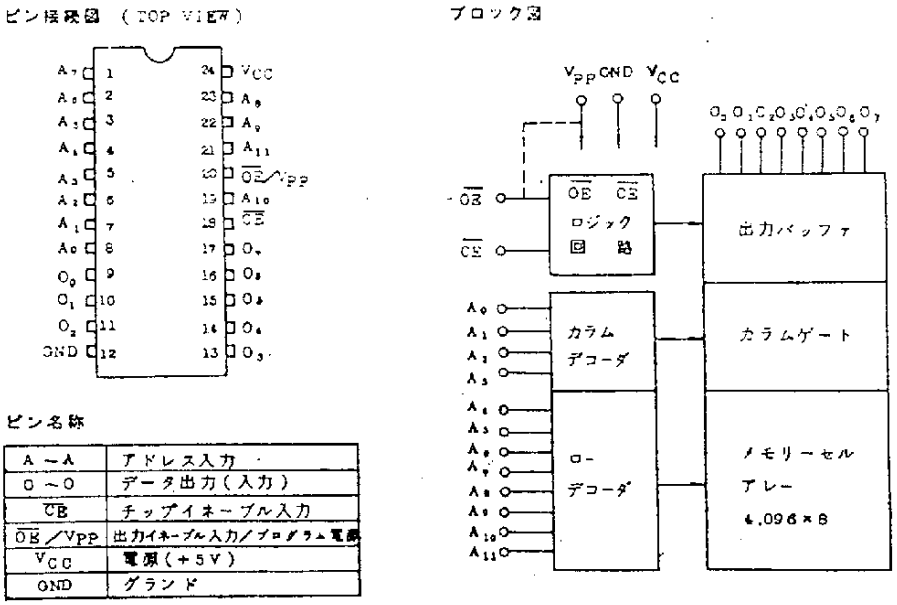
2.6　８０８５に接続するハードウェアとインタフェース

(a)　メモリ

メモリ素子の構成や機能については 2.2 で説明した。以下では，まず代表的なROMとRAMを例に取り，その概略を紹介する。次に８０８５とメモリの接続例を説明することにする。

(i)　ROM

2048×８ビットのEPROM２７３２のブロック図を図２－５３に示す。信号の意味については，2.2「メモリの構成」において説明した通りである。これにプログラムを書込むときは，２５Vの電圧を印加する。



ピン名称

A ～A	アドレス入力
O ～O	データ出力（入力）
$\overline{CE}$	チップイネーブル入力
$\overline{OE}$／VPP	出力イネーブル入力／プログラム電源
V_{CC}	電源（＋５V）
GND	グランド

図２－５３　２７３２の信号とブロック図

(ii) RAM

RAMの例として，スタティック形で1024×4ビットの2114を取上げる。図2－54はそのピン接続信号とブロック図である。

(b) 8085とメモリの接続例

8085とメモリ（I／Oでも同じ）を接続する際に注意すべきことは，メモリの動作速度（アクセスタイム）が遅い場合に，それに応じた待ち状態（ウエイト）をCPUに要求する必要があるという点である。

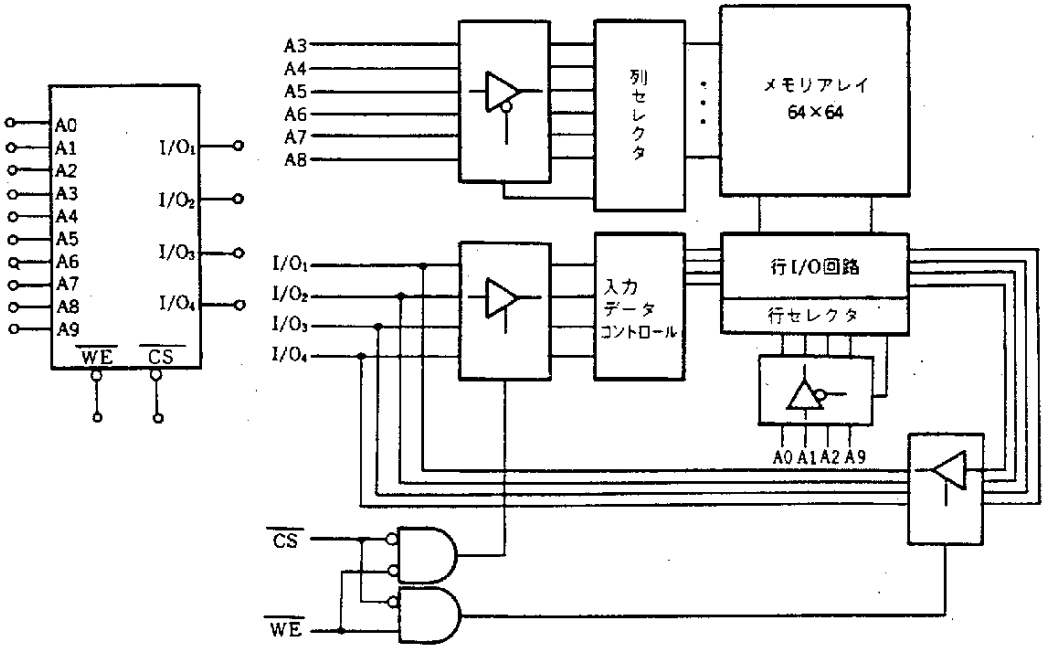
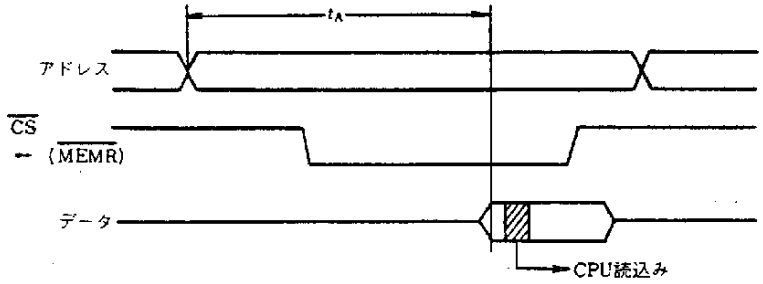


図2－54　2114の信号とブロック図



メモリのアクセス時間 t_A がCPUの読込みタイミングに間に合わない場合は，ウェイトを要求する必要がある。

図2－55　メモリのアクセス時間

図２－５５に示すようにＣＰＵのＲＥＡＤＹを制御して時間を調節しなければならない。１つのウェイトステートを置く最も簡単な方法は8085のＲＥＡＤＹ端子とWAIT端子を結線することである。

遅いメモリがアクセスされたときだけウェイトを挿入するには，アドレスとステータス情報によりフリップフロップをセット（ＲＥＡＤＹ信号リセット）し，クロック信号のタイミングに従ってそれをリセットするような回路が１例として考えられる。

先に紹介した2732, 2114 を8085に接続し，ＲＯＭ８Ｋバイト，ＲＡＭ１Ｋバイトのメモリを構成する例を，図２－５６に示す。2732, 2114ともに動作は高速であるから，ウェイト要求を行う必要はない。

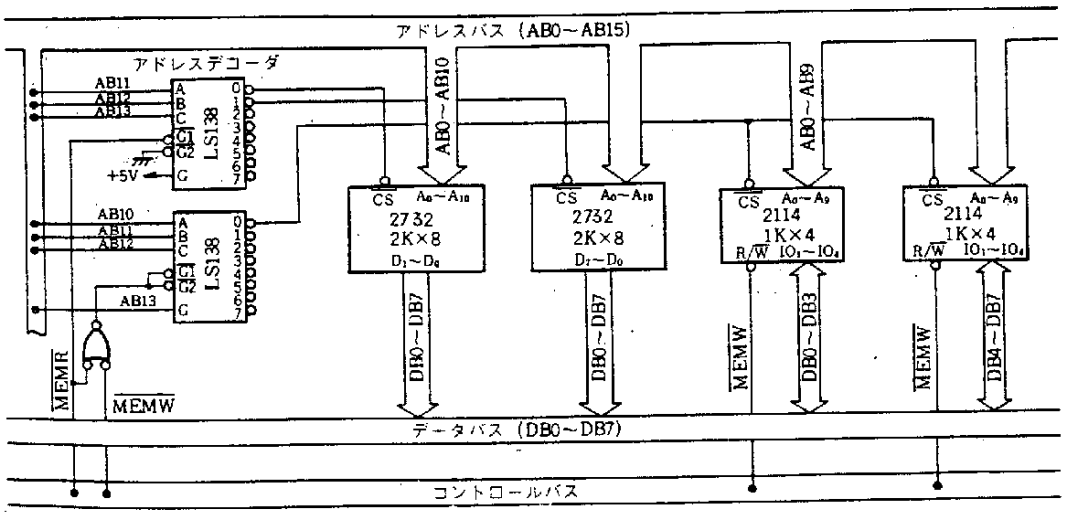


図２－５６　２７３２，２１１４を用いたシステムメモリの構成例

(c)　入出力インタフェース

マイクロコンピュータには，図２－５７に示すように，ＴＴＹ，ＣＲＴディスプレーなどいろいろな入出力端末装置が接続される。また，プロセスの制御に応用する場合には，その対象とさまざまな形で情報を交換することになる（このような入出力をプロセス入出力と俗称しているが定義はあいまいである）。

ここでは，いくつかの典型的な例をあげて，マイクロコンピュータの入出力の方法を説明する。

最初に取上げるのは，多機能，汎用化された2つの代表的なデバイス，8255, 8251 によるインタフェースである。これらの特徴はプログラム可能な点にあり，接続する機器に合わせて入出力の形式をＣＰＵから設定することができるので，機器ごとに専用インタフェースを設ける必要がなくなる。

その次に，プロセス入出力の例を説明する。プロセス入出力の項は，アナログ入出力とディジタル入出力に分けてある。

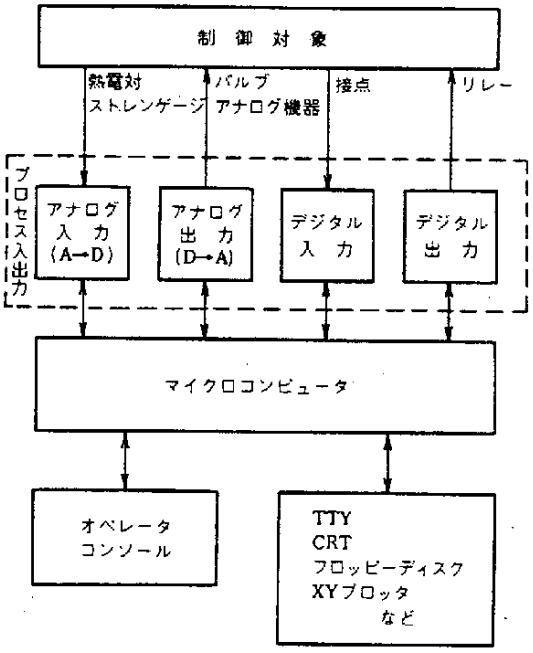


図２－５７　マイクロコンピュータ　システムの構成例

(i)　8255：プログラマブル周辺インタフェース

8255 は図２－５８に示されるように３つの８ビットポートからなる汎用入出力インタフェース素子である。３つのポートA，B，Cはそれぞれ入力，出力のいずれにも使えるようになっており，３通りの動作の仕方（モード）を選ぶことができる。どのように用いるかはＣＰＵからのプログラム（モード設定）によって規定されるのである。

３つのモードはそれぞれ次のような動作になる。

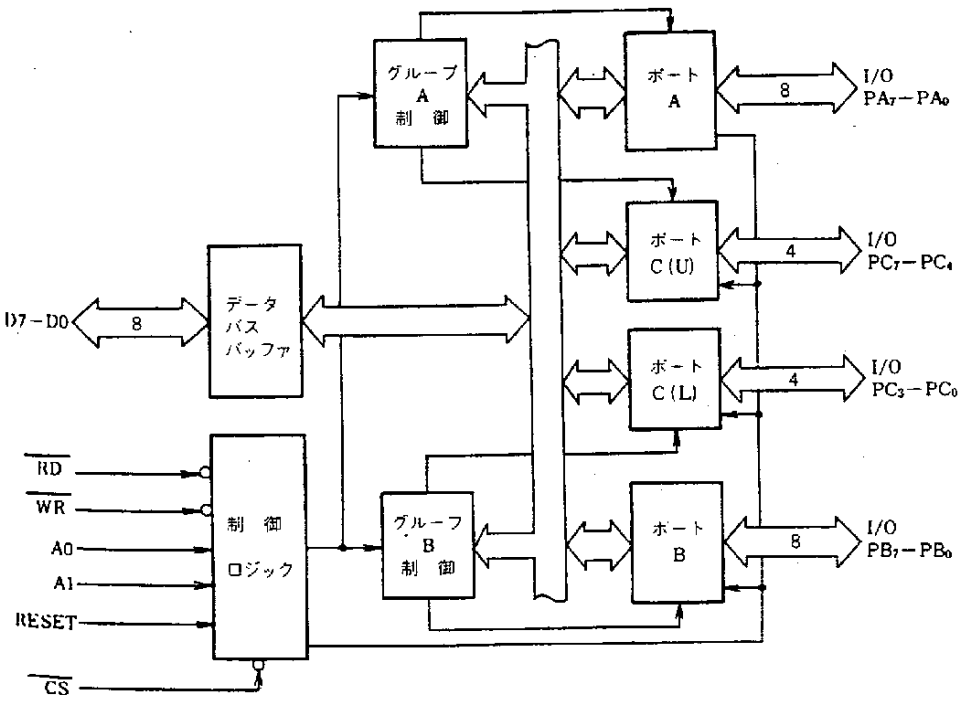


図２－５８　８２５５のブロック図

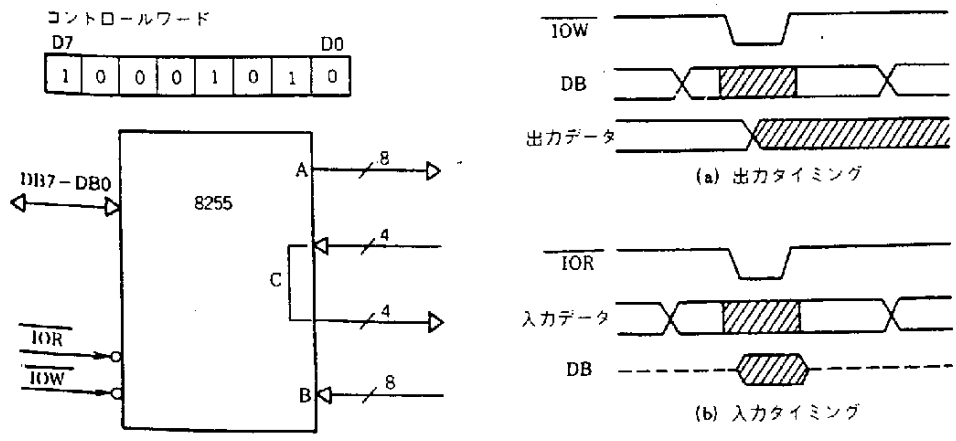


図２－５９　８２５５モード０の動作

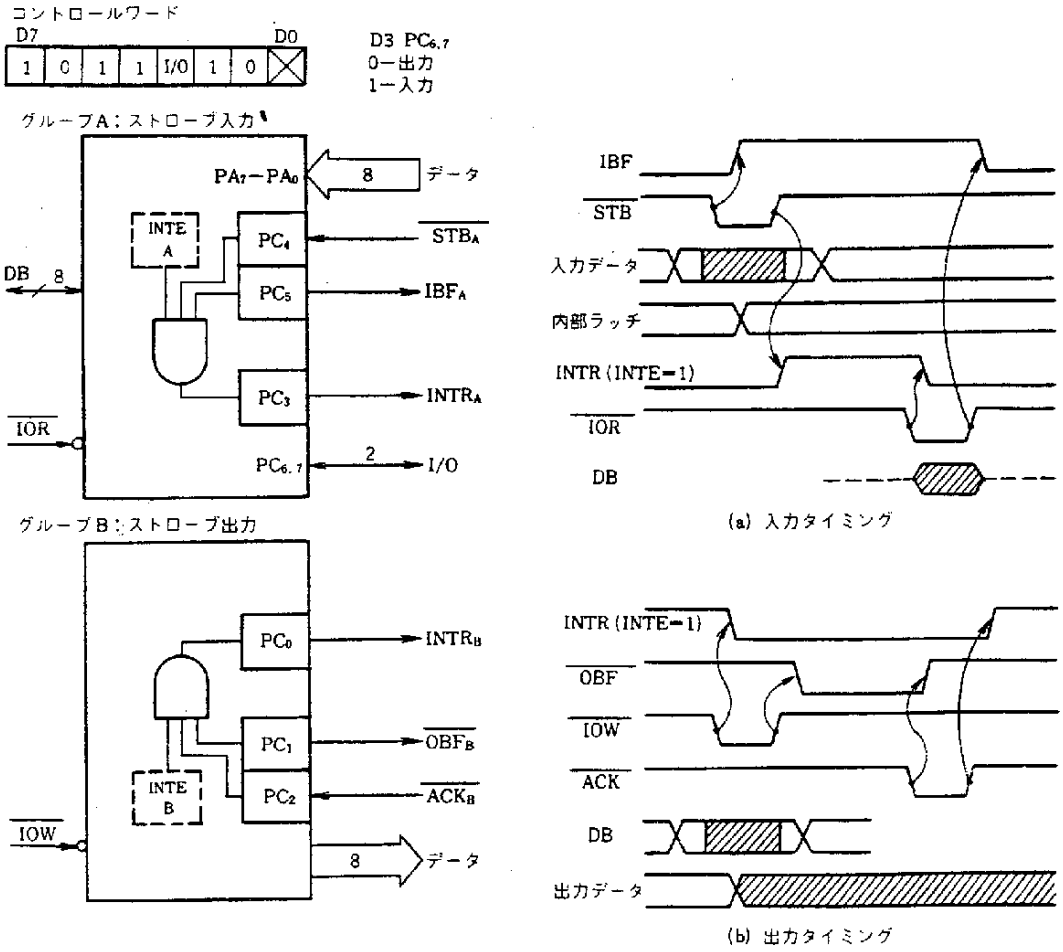
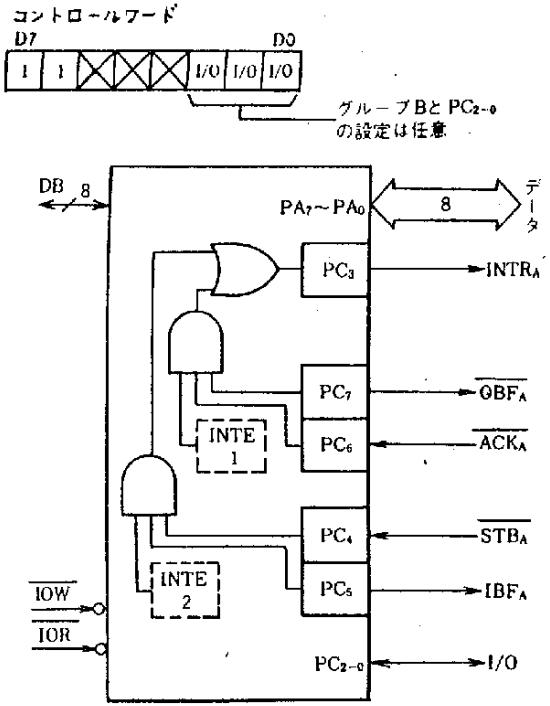
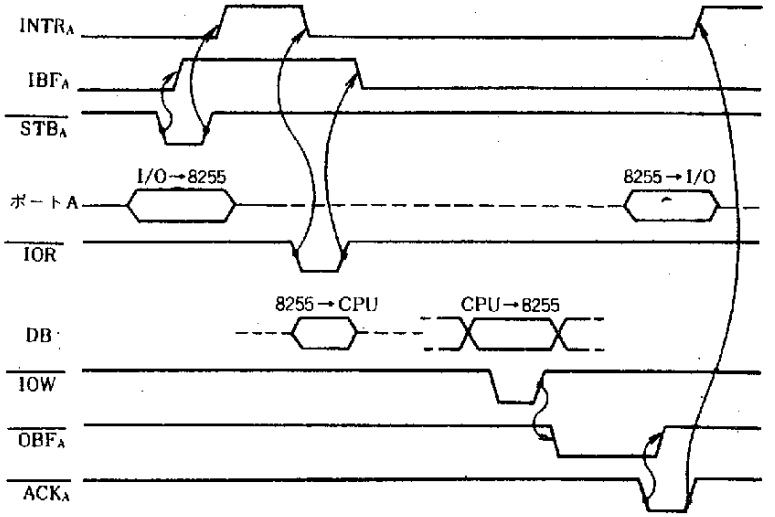


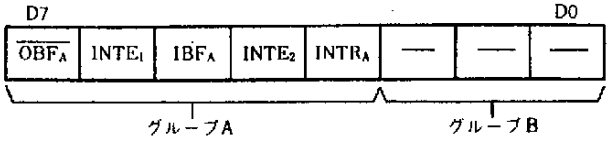
図２－６０　８２５５モード１の動作

①モード０（図２－５９）　８ビットのポートA，Bと，４ビットのポートC（上半），C（下半）を，それぞれ入力にも出力にも指定でき





(a) 入出力タイミング



(b) モード2ステータス

図2－61　8255モード2の動作

る。

②モード1（図2－60）　グループAとグループBの2つに分けられる。各グループで8ビットのデータポートは入力，出力いずれにも指定でき4ビットのポートはデータポートに対する制御信号，ステータスとして用いられる。

③モード2（図2－61）　グループAのポートAを入・出力両方向のデータポートとして使用し，ポートCの5ビットはポートAに対する制御信号，ステータスとして働く。

これらのモード設定およびポートの入・出力指定は，あらかじめ図2－62のような構成のコントロール・ワードを8255のレジスタに書込むことにより行われる。グループA，Bごとに動作モードをプログラムできる。

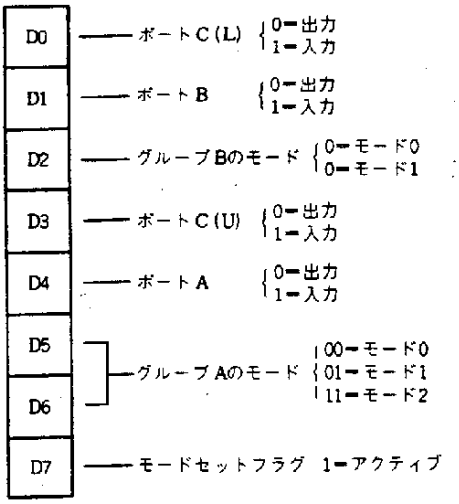


図2－62　8255のコントロール・ワードフォーマット

8255の簡単な使用例として，図2－63に示すようなキーボード，プリンタとのインタフェースを取上げてみよう。この例ではグループAをモード1の入力ポート，グループBをモード1の出力ポートとして使

用するので，あらかじめコントロールワード‘Ｂ４Ｈ’（図２－６０参照）を，アドレスＦ３に割付けられたコントロールレジスタに書込んでおかねばならない。これでキーボードからデータを入力する場合は図２－６４のようなプログラムを実行すればよい。また，図２－６５はプリ

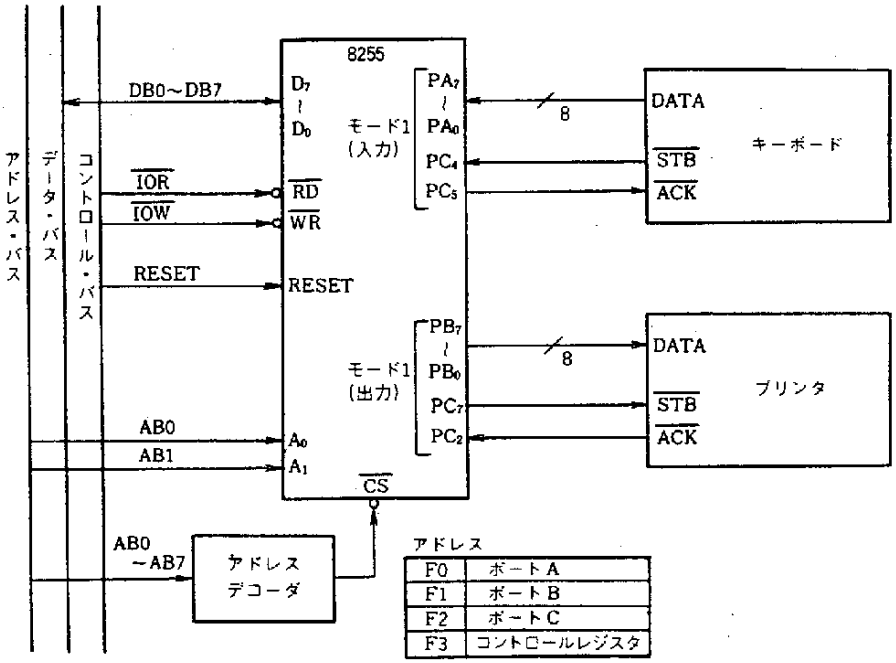
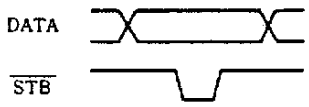
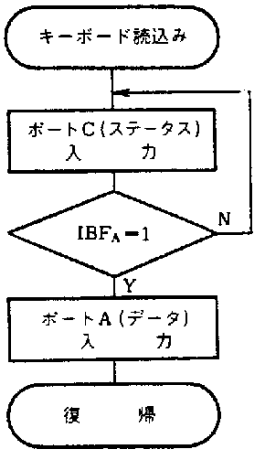


図２－６３　８２５５を用いたキーボード，プリンタとのインタフェース

ンタにデータを出力するプログラムの例である。この例のプリンタに与えるストローブ信号ＳＴＢのように，ポートＣのあるビットに接続されたコントロール信号の出力をＯＮ，ＯＦＦする場合には，ビットセット／リセットと呼ばれる８２５５の特別な機能を利用することができる。この機能は，ポートＣの任意のビットをセット(1)，リセット(0)することができるというもので，図２－６６に示すフォーマットに従ってコントロールレジスタへ出力すればよい。一方，各グループからの割込み要求信号ＩＮＴＲをＣＰＵのＩＮラインに接続すれば割込み方式で使用でき

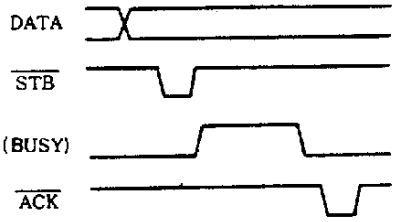


(a) キーボードの信号

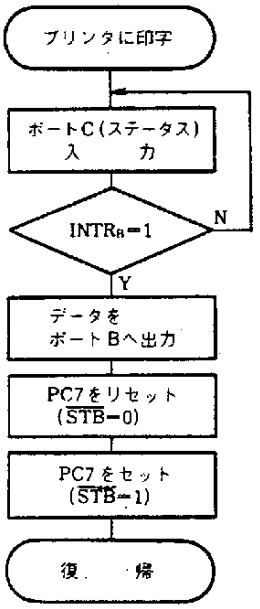


(b) プログラム

図２－６４　キーボードからの入力制御例



(a) プリンタの信号



(b) プログラム

図２－６５　プリンタの出力制御例

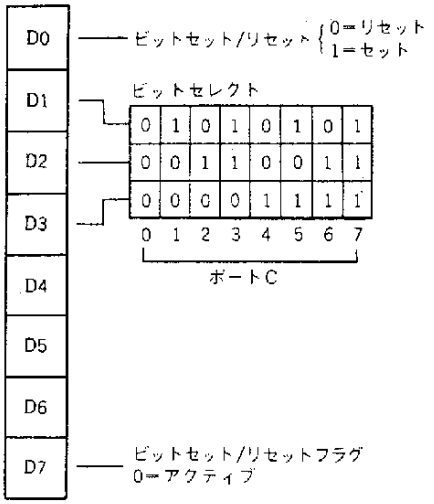


図２－６６　８２５５のビットセット／リセット　フォーマット

ることになるが，その制御にもビットセット／リセット機能を利用する。

８２５５内の割込みマスク（ＩＮＴＥ）に対応したポートＣのビットをセットしておけばＩＮＴＲ信号が出力されるようになり，リセットした場合は禁止される。図２－６０の例でストローブ出力のポートＢを割込み方式で使うには，ポートＣのビット０（ＰＣ０）をセットしておけばよいわけである。ただし，先のプリンタ出力のプログラム例では，ステータスとしてＩＮＴＲを検出している。

ところで，８２５５のモード１，モード２における動作のように，出力データの受取りを要求する信号と，それに対する受信確認信号を相互に交換しながら情報を転送する方法を，一般にハンドシェーク（握手，Hand shake）方式と呼ぶ。ハンドシェイク方式は，送信側と受信側の動作速度が異っていても確実にデータを受渡しできるので，非同期で動作する機器を接続する際の有力な方法としてよく用いられる。とくにモード２の動作は，共通バス方式のシステムにおけるハンドシェイク転送に適していると考えられる。

(ii) ８２５１：プログラマブル通信インタフェース

ＣＰＵとＴＴＹ，ＣＲＴなどの入出力機器や通信装置（モデム）を接続する場合，これらがいずれも直列データを送受する方式であるため，並列データと直列データを変換するインタフェースが必要となる。通常，通信インタフェース，ＵＳＡＲＴ（Universal Synchronous Asynchronous Receiver／Transmitter）などと呼ばれるものである。

８２５１は，現在用いられている直列データ転送に広範囲に適用可能なプログラマブル通信インタフェース素子で，内部の構成を図２－６７に示す。ＴＴＹなど低速，非同期方式の伝送から，高速のモデムのような同期方式に至るまで動作させることができる。非同期方式とは，図２－６８(a)のようにキャラクタごとに制御ビットを付加する方式で，同図(b)のように同期符号（ＳＹＮＣキャラクタ）に続いてデータを送る方式を

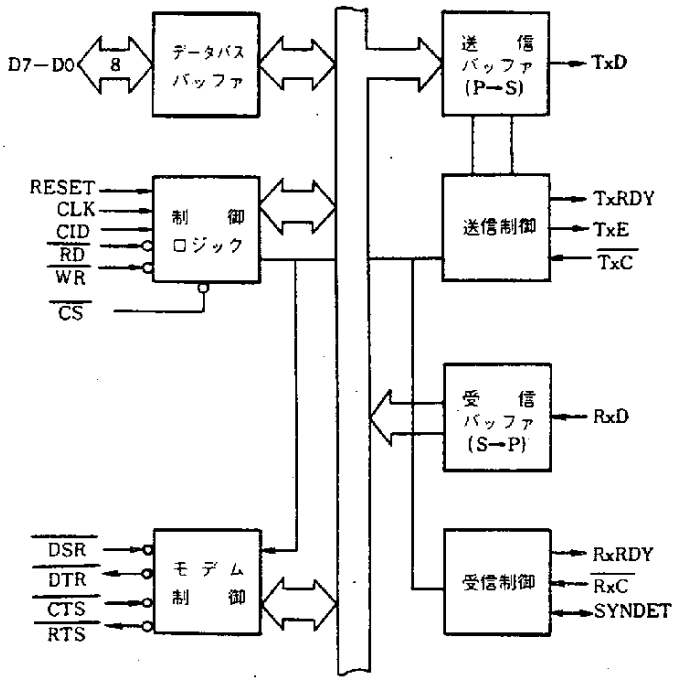


図２－６７　８２５１のブロック図

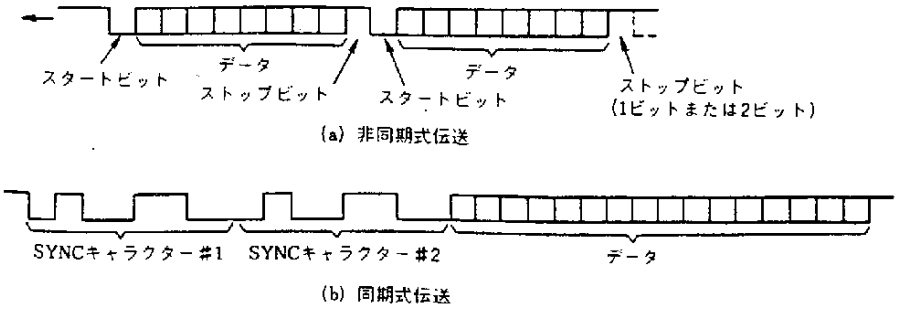


図２－６８　非同期式伝送と同期式伝送

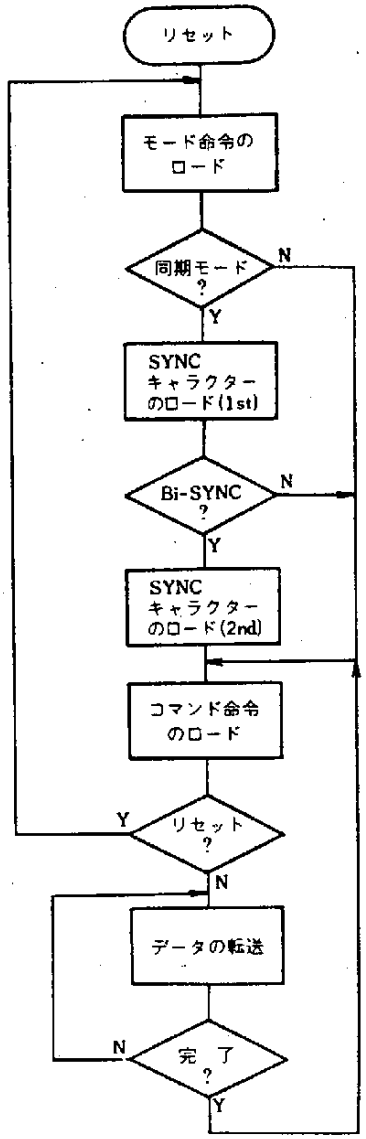


図２－６９　８２５１の初期設定

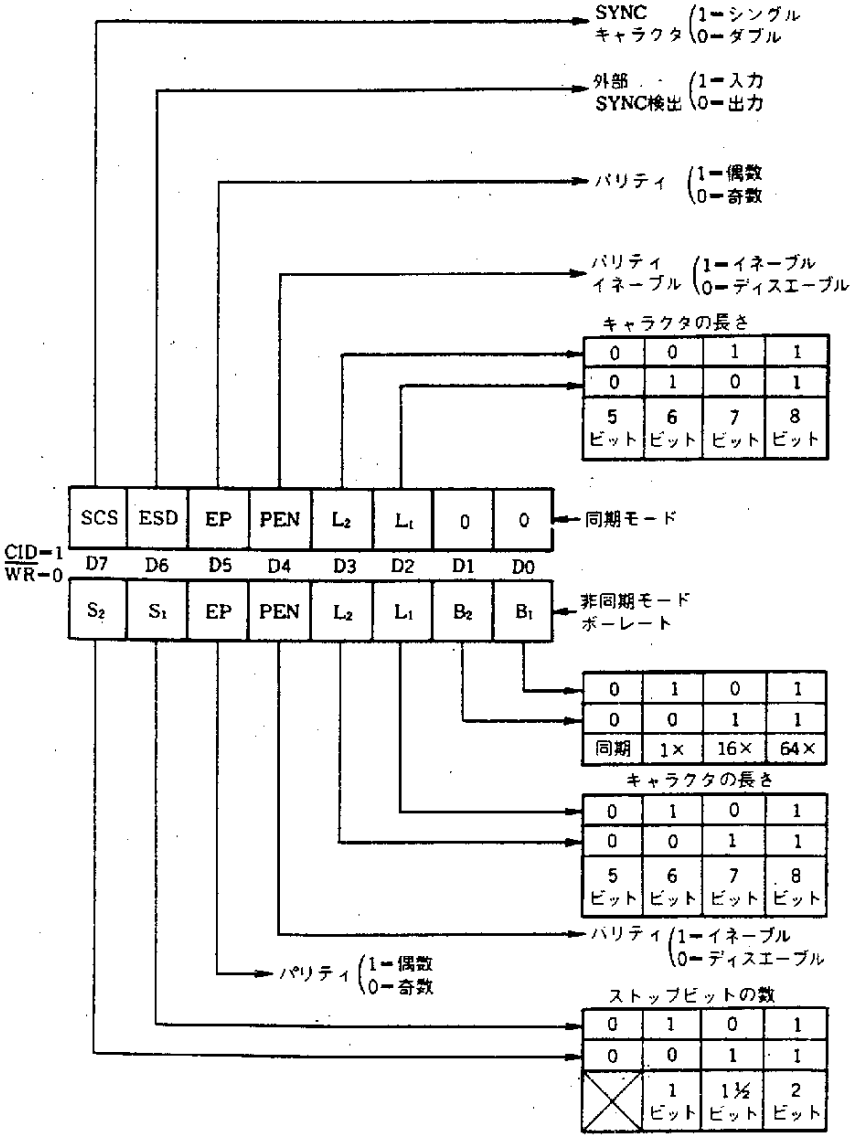


図２－７０　８２５１のモード設定，フォーマット

同期方式という。

８２５１をいずれの方式で用いるかは，リセットに引続き図２－６９の初期設定を実行することにより規定される。最初のモード設定は通信に必要な一般的動作特性を指定するために行うもので，図２－７０に示す

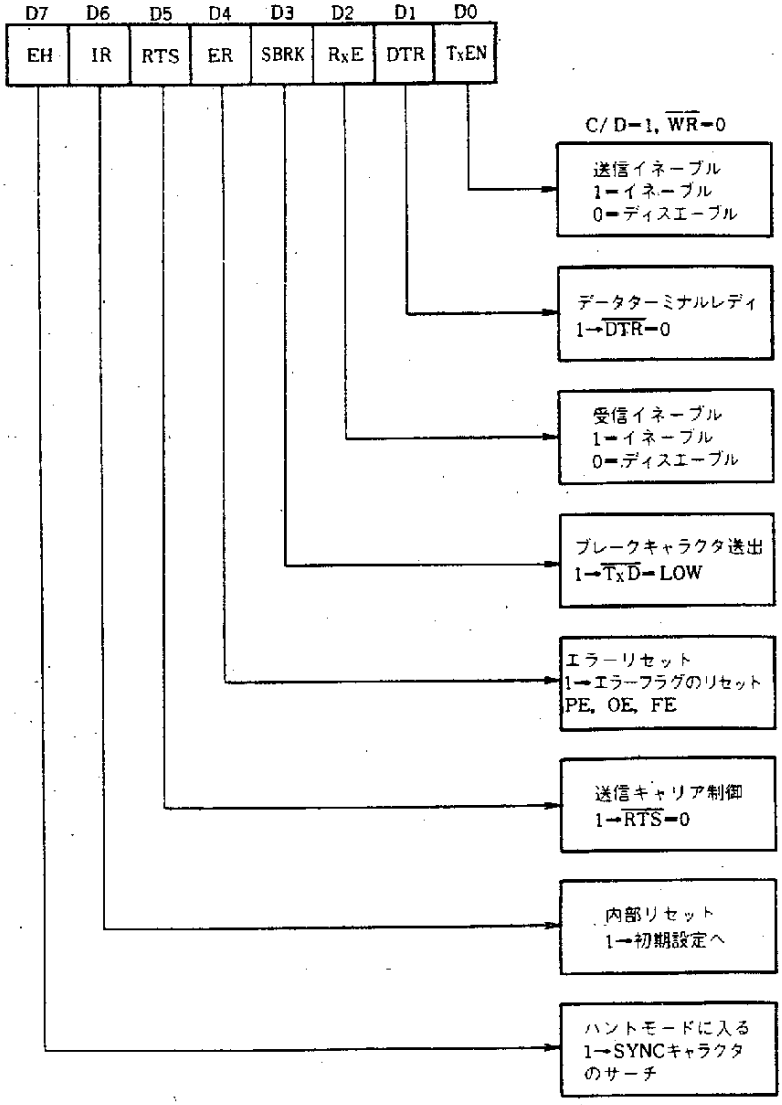


図２－７１　８２５１のコマンド，フォーマット

フォーマットに従う。同期モードを指定した場合には次にＳＹＮＣキャラクタを書込む。これでデータ通信の準備が整ったわけで，あとは図２－７１のようなコマンドを与えて，データの送受信やモデムのコントロールを開始すればよい。８２５１のステータス（図２－７２）はいつでも

読出すことができる。

8251を用いてＴＴＹやＣＲＴを接続するには図２－７３のようにする。クロックTxC,RxCの周波数は，ＴＴＹのボーレイトが１１０とすれば，先に述べたモード命令によりボーレイト（B_2, B_1）を1倍にプログラムしたとき110H_Z，16 倍ならば1.76KH_Z とする。

その他の応用として，同期方式の機器，非同期方式のモデムとの接続例を図２－７４，図２－７５に示す。

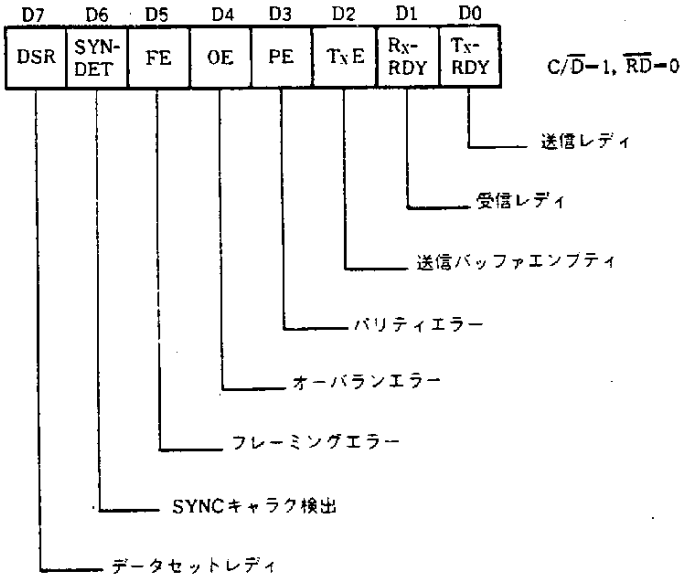


図２－７２　８２５１のステータス

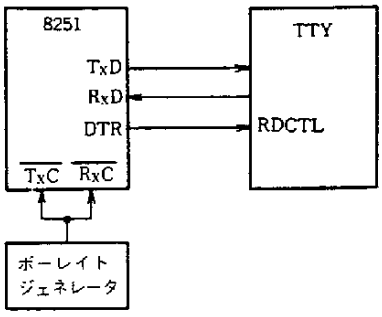


図２－７３　８２５１とＴＴＹの接続

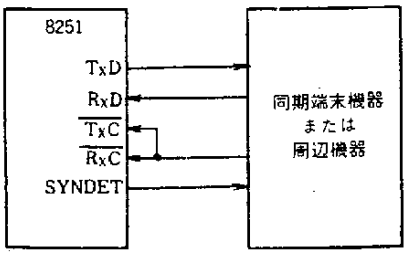


図２－７４　同期式機器の接続

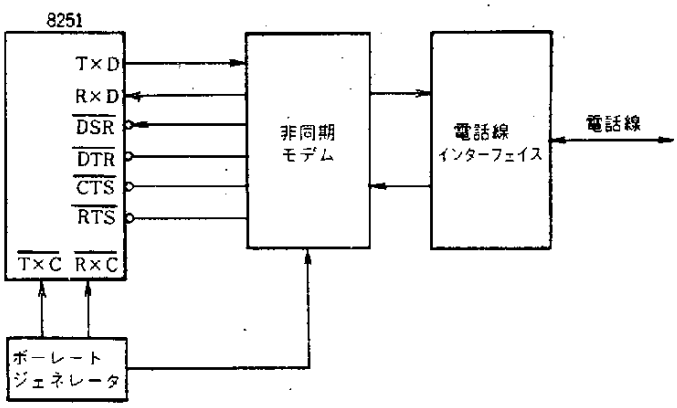
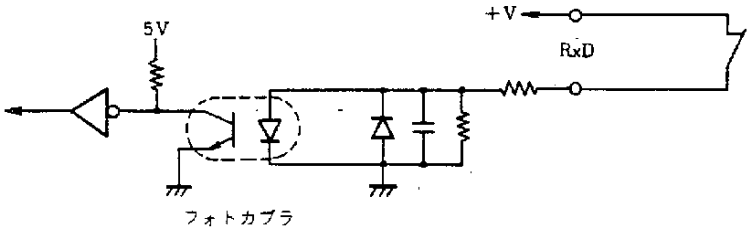


図２－７５　非同期式モデムの接続



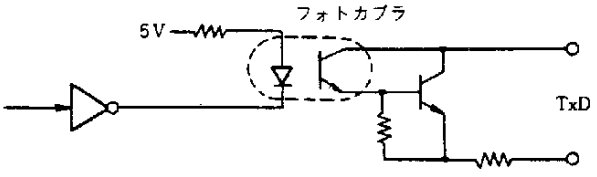


図２－７６　カレントループ接続回路

以上のいずれの場合にも，実際には，相手側の仕様に応じた信号レベルの変換回路や，バッファ回路を介して接続されることになる。

図２－７６は，カレントループと呼ばれる接続回路の例を示す。後に紹介するＡＳＲ－３３などは，このタイプの回路によって接続される。

また，図２－７７は，モデムのインタフェース回路に採用されている。ＲＳ－232－Ｃ と呼ぶ規格である。これは直列データ転送の信号の質を規定したもので，最近ではモデムの接続ばかりでなく，ＴＴＹ，ＣＲＹなどの端末装置にもこのインタフェースを標準装備したものが多く，直列転送方式の標準ということができる。この規格にマッチした送信，受信用ＩＣが用意されているので，通信インタフェース素子とＲＳ－２３２－Ｃに従った機器やモデムとは，これを利用して接続するのが簡単である。

回路形態		片側接地
最大ケーブル長		15m
最大転送速度		20Kボー
ドライバ	無負荷時　最大出力電圧	±25V
	負荷時　最小出力電圧	±5V～±15V
	電源断等　最小出力抵抗	R_0 = 300 Ω
	短絡時　最大出力電流	Isc＝± 500 mA
	出力スルーレート	30V/μSMAX
レシーバ	入力抵抗	R_{IN}＝3～7KΩ
	最大入力スレッショルド	－3～＋3V
	最大入力電圧	±25V

図２－７７　ＥＩＡ規格，ＲＳ－２３２－Ｃの主要な特性

(iii) プロセス入出力

汎用の入出力インタフェースの構成法については既に説明したが，マイクロコンピュータを用いてプロセスの制御を行う際には，その対象機器や回路により入出力信号の種類や特性はさまざまであり，それらに応じたハードウェアを介して接続する必要がある。

プロセスの制御は，いろいろなセンサ，接点，計器から情報をマイクロコンピュータに入力し，また逆に，コンピュータの信号で記録計や制御器，各種継電器を働かせることにより行われる。これらの入出力をプロセス入出力と呼ぶ。ここでは，アナログ信号とデジタル信号の入出力に分け，基礎的なことがらを説明する。

(イ) アナログ入出力

マイクロコンピュータは，熱電対，可変抵抗，ストレンゲージ（図2－78）などいろいろなセンサや計測器からアナログ信号を入力することがある。一方，X－Yプロッタやブラウン管，バルブなどのようにアナログ機器を制御する場合も多い。マイクロコンピュータとこのようなアナログ信号とのインタフェースにはディジタル符号をアナログ電圧（電流）に変換するD－A変換器や，アナログ電圧（電流）をディジタル符号に変換するA－D変換器を使用する。

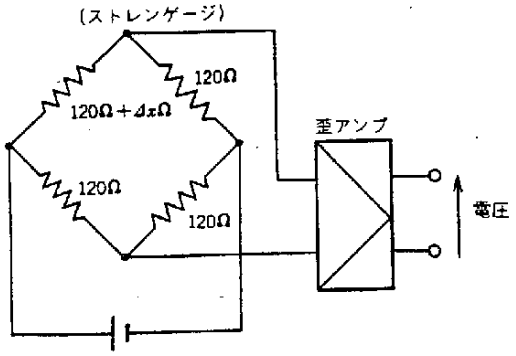


図2－78　ストレン・ゲージ

A－D変換，D－A変換を行う場合には，一般に次のようなことを考慮しなければならないだろう。

①分解能（ビット数）　8ビットから16ビットまでの変換器がある。例えばフルスケール20V に対して5mVの精度を得るには，12ビットの変換器が必要であるという目安になる。

②アナログ入出力のフルスケール値は±10V 程度のものが多い。

③変換時間　1～20μs のものが多い。

④動作性能　直線性や温度変化による誤差などが問題となる。

⑤デジタルコード　バイナリ（オフセットバイナリ），2の補数，BCDなどのデジタルコードが使用されており，プログラミングの容易なコードを選択する。

まず，D－A変換器について説明しよう。D－A変換器で最も多く用いられているのは図2－79のようなラダー回路で，出力端からi番目のビットのスイッチが切替わると出力に2－（i－1）の重みの影響を与える。出力は電流で，このような重みの入力の和を表わしている。通常，IC化されたD－A変換器を使用し，図2－80のように簡単にアナログ出力を作り出せる。チャネル数を増す場合は，その数だけのD－A変換器を並べるのが普通である。

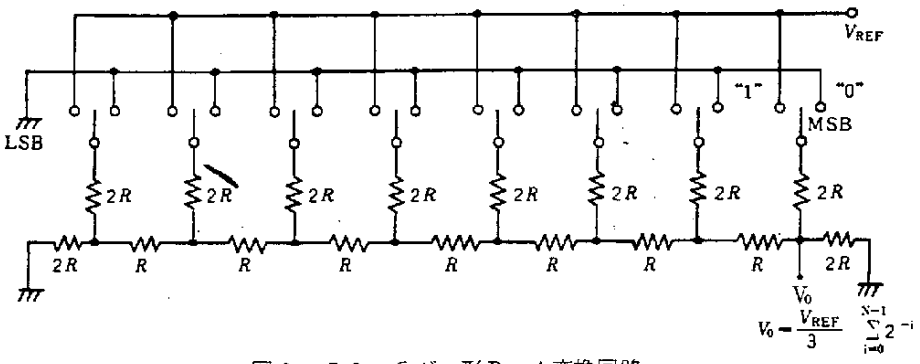


図2－79　ラダー形D－A変換回路

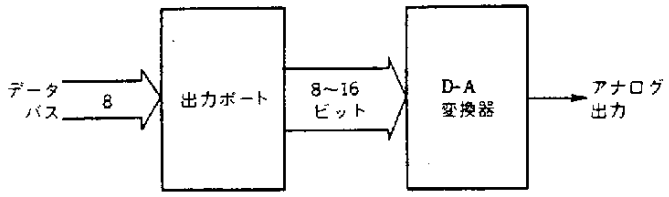


図2－80　D－A変換モジュール

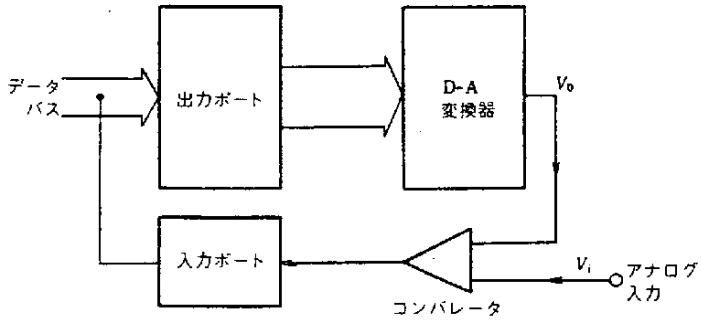


図２－８１　Ｄ－Ａ変換器を利用した逐次比較形Ａ－Ｄ変換器の構成

ところで，アナログ信号を入力する場合のＡ－Ｄ変換器もまた上述のＤ－Ａ変換器を利用して作ることができる。図２－８１のような構成で，Ｄ－Ａ変換器の出力とアナログ入力を比較しながら，ＭＳＢから順にデジタルコードを決定していけばよいのである。このような方法を逐次比較方式と呼ぶ。

変換のフローを図２－８２に示す。プログラムで逐次比較を実行する場合には変換に数百μs を要することになるが，より高速に変換するには，同様の処理を行うシーケンスコントローラを内蔵したＡ－Ｄ変換器を使用する。市販のＡ－Ｄ変換器は，スタートコマンドにより変換動作を開始し，12ビットの変換を20～30μsで行うものが多い（高速のものでは数μs ）。

実際のアナログ入力インタフェースは，このＡ－Ｄ変換器を中心として，一般に図２－８３のように構成される。各種のセンサから得られる信号はさまざまな電圧や電流の値に及んでいるから，Ａ－Ｄ変換器にみあったレベルに直すために，増幅や非線形変換を行う回路を介さねばならないこともある（シグナル・コンディショニング）。

多チャネルの入力をＡ－Ｄ変換するときは，高価なＡ－Ｄ変換器をチャネル入力ごとに並べるのではなく，図２－８３のようにアナログマルチプレクサを切換えてチャネルを選択するのが普通である。

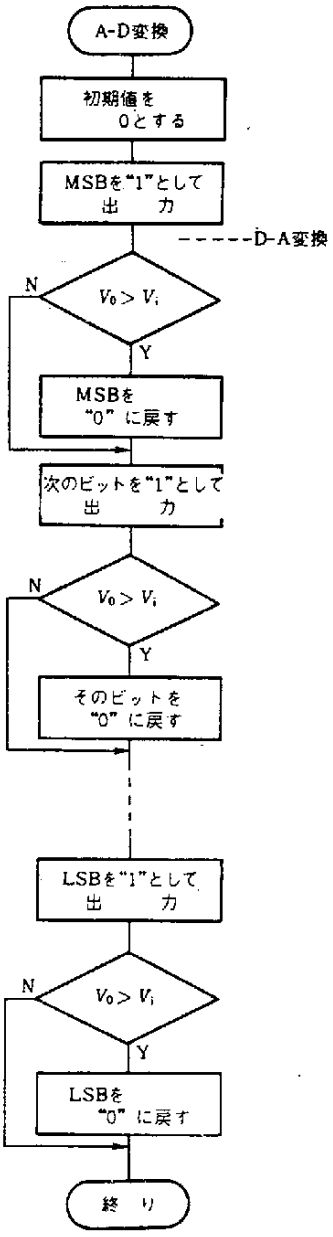


図２－８２　Ａ－Ｄ変換プログラム

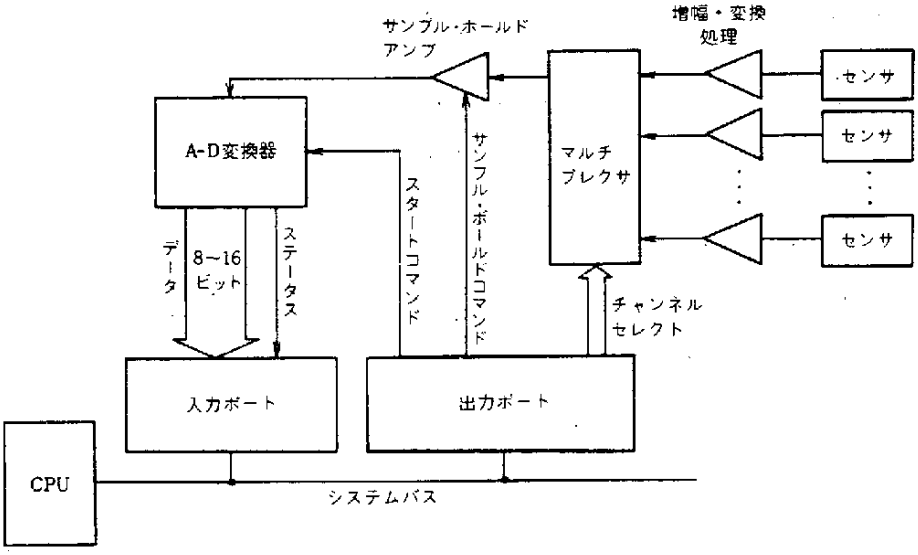


図２－８３　Ａ－Ｄ変換モジュールの構成

サンプル・ホールド回路は，Ａ－Ｄ変換中に変換器の入力電圧が変化するのを防ぐために挿入される。例えば，変換に25μs を要する12ビットＡ－Ｄ変換器に1.5Hz以上の信号が入力されると，変換中に入力がＬＳＢ幅以上変化することになって変換値に誤差を生ずる。このような場合には，変換を行っている間だけ入力電圧をホールドするわけである。

アナログ入出力のインタフェースに際しては，実用上注意しておくべき問題もある。当然のことながら，コンピュータのデジタル回路と外部アナログ回路が接続されるわけであるから，危険性や相互に及ぼす悪影響をできるだけ防ぐような回路構成や配線を行う必要がある。また，Ａ－Ｄ，Ｄ－Ａ変換素子はモノリシックＩＣ化される傾向にはあるが，現在ではまだハイブリッド形が多いので，とかく信頼性の面でトラブルを発生しがちである点も注意を要する。

(ロ)　デジタル入出力

デジタル入力には，接点やスイッチの状態，計測器などが考えられ

る。また，デジタル出力は，リレー，ソレノイド，モータ，記録計，表示器などを対象とするものである(図２－８４)。入出力回路は，送受する信号の電圧，電流レベル，速度，質などに応じて異なることになる。

インタフェース回路の構成例を示す前に，プロセス入出力における一般的な問題を上げてみよう。

①電圧，電流のレベルを合わせるとともに，駆動能力が不足しないようにする。

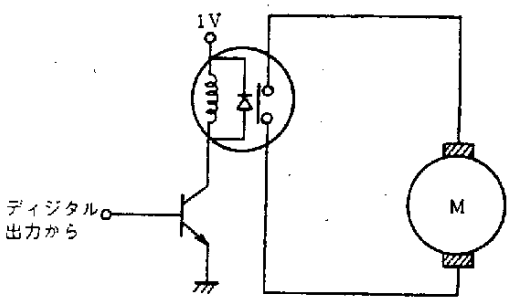


図２－８４　モータの駆動

②外部接続ラインは長くなることが多いので浮遊容量が増す。応答を速くするにはできるだけ低い電圧で駆動し，電流を十分に流すようにする。

③入力信号の質や，駆動する負荷の特性に注意する。入力信号が歪やノイズを含むものであればフィルタを入れたり波形整形を行う。

④リレー，ソレノイド，モータなど誘導性負荷をドライブする場合には，そのリアクションの対策が必要である。このような負荷を大電流でスイッチングしたり，ＡＣラインの制御を行うようなときは，コンピュータ側の駆動回路と負荷回路を電気的に絶縁しておくのが安全である。駆動回路のノイズが誤動作の原因となるし，負荷側に異常が生ずればコンピュータの全体に被害がおよび危険となるからである。

このような問題に関し，フォトカプラは，比較的高速で使い易いので，電力素子や接点信号の入出力インタフェースによく用いられる。

よく用いられるプロセス入出力回路の例を示そう。図２－８５は

電磁リレーやソレノイドを駆動する回路の例である。負荷のスイッチ回路をオン，オフするリレー，電気信号を機械的な直線運動に変換するソレノイドは，産業分野で広く利用されているものである。

一方，フォトカプラは接点信号などの入力回路にも利用されることが多い。図２－８６は，パルス性入力信号の入力回路例を示す。電力計や流量計などから得られる信号はパルス信号である。

最後に，トライアックを用いたＡＣラインの制御方法，およびＡＣラインからの入力方法の一例を図２－８７に示す。

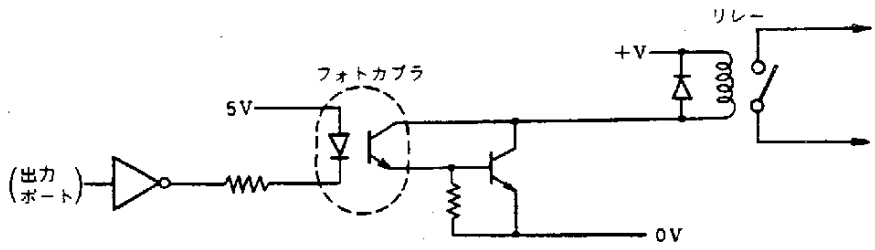


図２－８５　リレーの駆動回路例

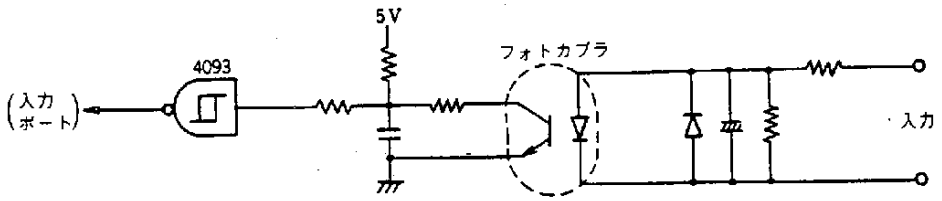


図２－８６　フォトカプラを介した入力回路の例（パルス信号の場合）

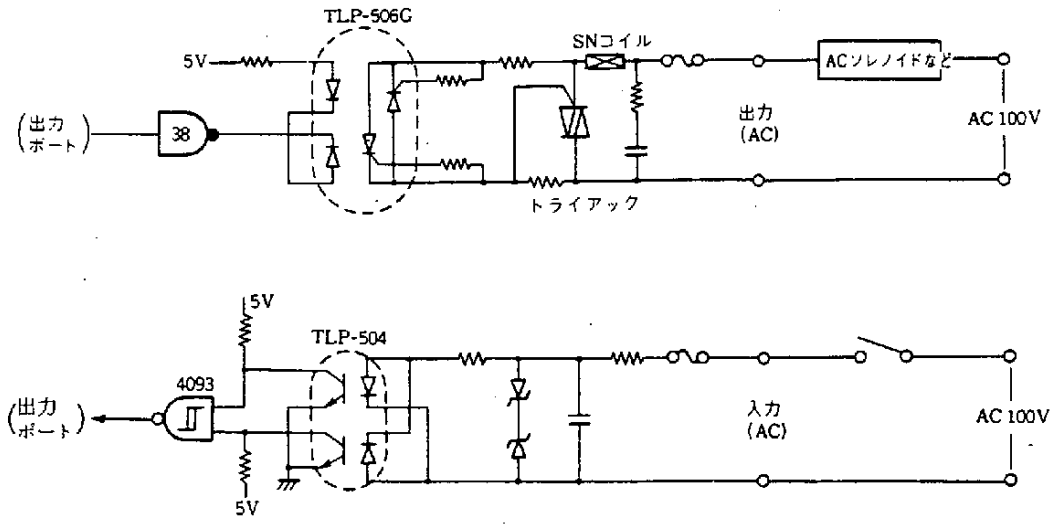


図２－８７　ＡＣラインの入出力回路例

——禁無断転載——

昭和58年3月発行

発行所　財団法人　日本情報処理開発協会
東京都港区芝公園3丁目5番8号
機械振興会館内
Tel（434）8211（代表）

印刷所　株式会社　タケミ印刷
東京都千代田区神田司町2－16