

II マイコンの構成要素の働きと接続

学習目標

マイコンのバスの種類とその働きの概略、そしてマイコンの構成要素がバスによって接続されていることを理解させる。また、マイコンにおけるマイクロプロセッサやメモリ、入出力ポートの種類、役割、必要性などについても理解させる。

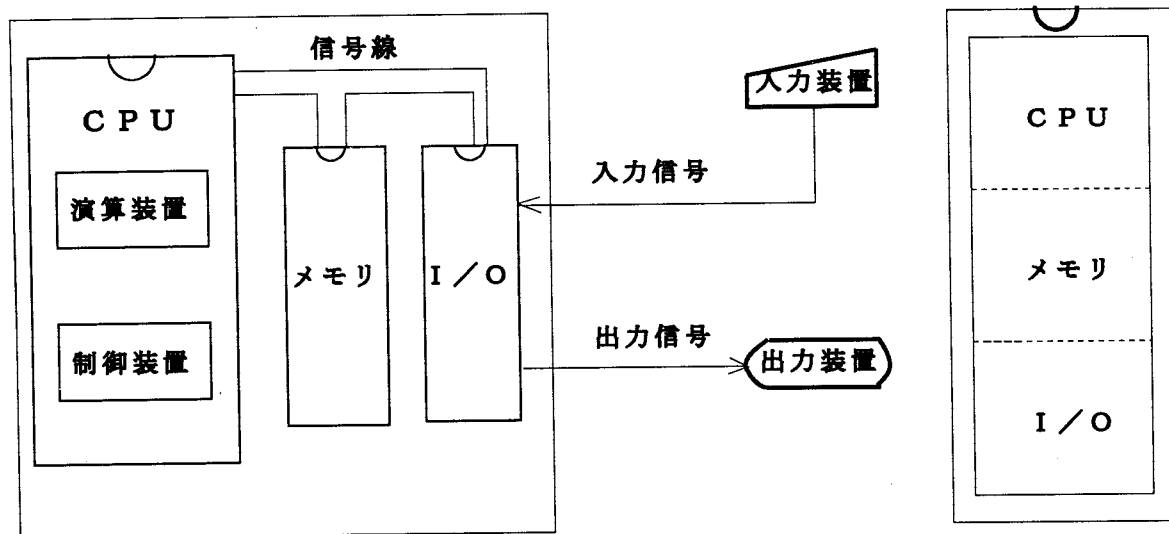
1 マイコンの基本構成要素とバス構成

(1) マイコンの基本構成要素

マイコンは三つの基本的な要素から成り立っている。まず、最も重要なのが四則演算や論理演算をしたり、比較判断をしたりするCPUと呼ばれる部分である。そして、そのCPUを動作させるためのプログラムを記憶したり、入力されたデータを格納したりするためのメモリ、最後は外部から信号を取り込んだり、外部機器をコントロールしたりするための入出力（I/O）部分である。

このCPU、メモリ、I/Oをマイコンの3要素といい、各種信号線（バス）で接続されている（図II-1参照）。

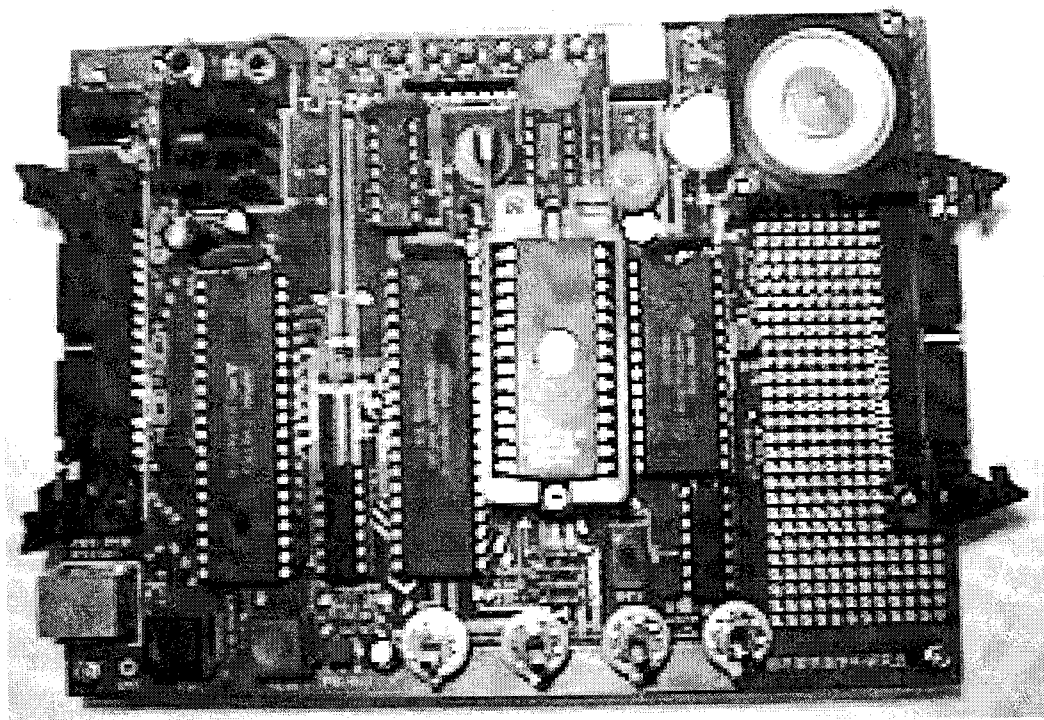
これらの三つのうちどれか一つが欠けても、マイコンシステムを構成することはできない。また、三つの要素があるからといって、必ず3個の部品からなるというわけでもない。これらの三つの要素が一つのICの中に組み込まれているものをワンチップマイコンという（写真II-1、II-2、II-3参照）。



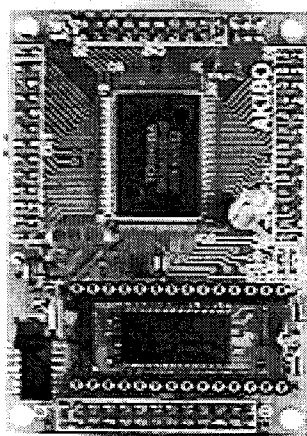
(1) ワンボードマイコン

(2) ワンチップマイコン

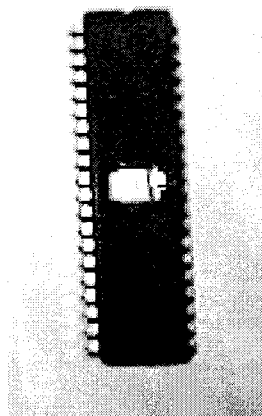
図II-1 マイコンの基本構成



写真II-1 ワンボードマイコン



写真II-2 ワンチップマイコンを搭載したボードマイコン



写真II-3 ワンチップマイコン

(2) バス構成

これらCPU、メモリ、I/Oを接続するのがアドレスバス、データバスという信号線の束である。アドレスバスとは、その名のとおりメモリやI/Oの番地を指定するための信号線で、データバスとはアドレスバスで決めたメモリやI/Oの番地にデータを書き込んだり、読み込んだりする際のデータの通り道である。

また、これとは別にメモリなのか、I/Oなのかを指定する信号線、データの読み込みか、書き込みかの指定をする信号線など何本かのコントロールバス（制御信号線）がある（図Ⅱ-2参照）。

① アドレスバスの働き

バスとは信号線の束である。アドレスバスとは番地を示す信号線の束である。マイコンは命令やデータをメモリから読み込む。また、センサの信号をI/Oから読み込む。このときCPUがどのメモリから読み込むのか、また、どのセンサからデータを読み込むのかの番地を指定する必要がある。この指定のためにアドレスバスが使われる。

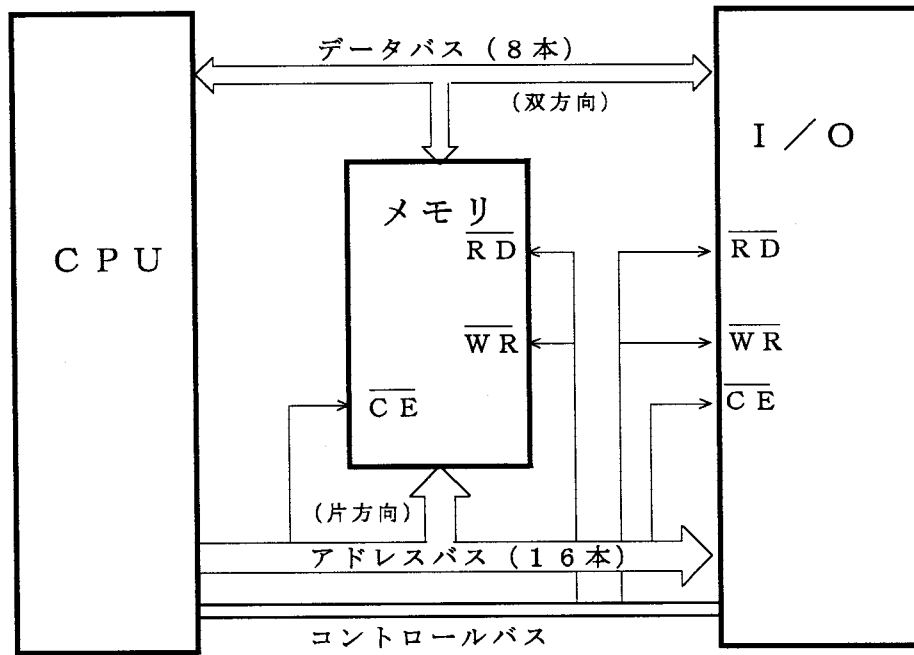
このときどれだけのメモリやI/Oを扱えるのかは、アドレスバスのビット数（本数）で決まる。例えば、8ビットCPUにはアドレスバスが16本あり、それぞれA0～A15という名前がついていて、 2^{16} つまり0000h番地～0FFFFh番地（hは16進数を示す添え字）までの64Kバイトのメモリ空間を管理できる（図Ⅱ-3参照）。アドレスバスが24ビットなら16Mバイトの指定ができる。一般にnビットのアドレスバスで 2^n バイトを指定できる。

② データバスの働き

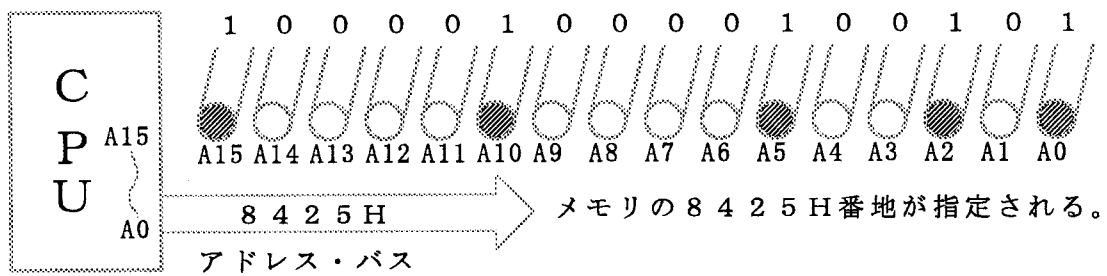
アドレスバスが番地を示す信号の束であるのに対して、データバスはデータを示す信号線の束である。メモリから読み込んだ命令やI/Oからのデータは、このデータバスを通してCPUに読み込まれる（リード動作）。また、CPUで処理したデータはメモリやI/Oに対して書き込まれる（ライト動作）。

このようにデータバスは、アドレスバスと異なり双方向にデータが流れる。

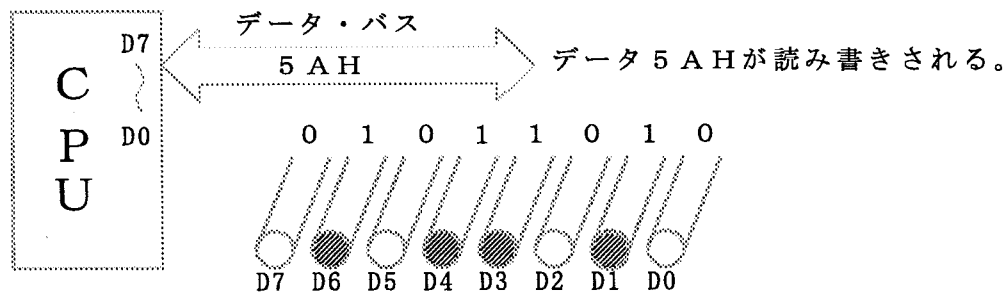
Z80や6809などの8ビットのCPUでは、D0～D7の8本の線があり、同時に処理できる単位が8ビットということである（図Ⅱ-4参照）。



図II-2 バス構成



図II-3 アドレスバスの説明



図II-4 データバスの説明

③ コントロールバスの働き

マイコンにはプログラムの命令やデータを読み書きするためのメモリと、外部装置のセンサなどからの信号を読み込んだり、外部装置へCPUで処理したデータを出力するために必要なI/Oがある。これらの番地の指定にアドレスバスが使われる。つまり、番地指定には、メモリ、I/Oにかかわらずアドレスバスが兼用される。

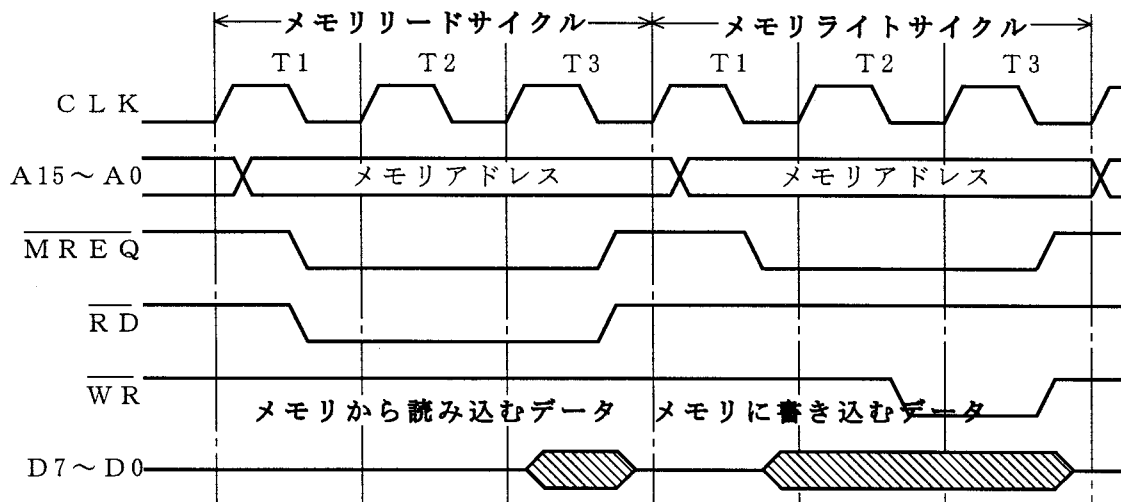
そこで、メモリから読み出したいのか、I/Oからセンサ入力を読みみたいのかという指定を別の信号線で指定する必要がある。そのための信号線が $\overline{\text{MREQ}}$ と $\overline{\text{IORQ}}$ である。メモリに対して読み書きしたいときは $\overline{\text{MREQ}}$ が、I/Oに対して読み書きしたいときは $\overline{\text{IORQ}}$ がLレベルになる。メモリかI/Oかを選択しているが、ハードウェア的には $\overline{\text{MREQ}}$ と $\overline{\text{IORQ}}$ のどちらの信号がアクティブ（有効）になるかという違いだけである。

以上の信号線で、メモリかI/Oか、そしてどのアドレスかは指定することができる。後はそのアドレスに対してデータを読み込むのか、書き込むのかという信号線が必要である。それが $\overline{\text{RD}}$ と $\overline{\text{WR}}$ である。データをCPUに読み込む場合は、 $\overline{\text{RD}}$ がLレベルとなり、書き込む場合は、 $\overline{\text{WR}}$ がLレベルになる。当然のことながら同時にLレベルになることはない。

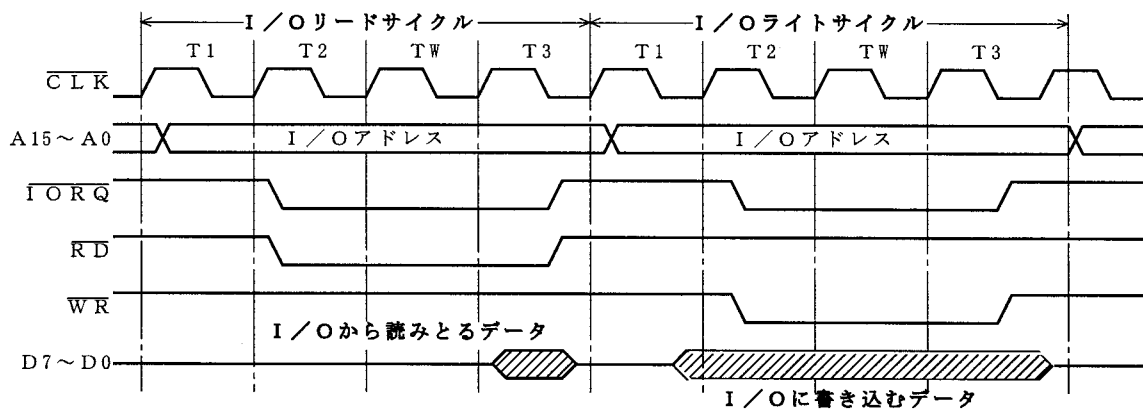
次に、 $\overline{\text{RESET}}$ 信号入力をLレベルにするとCPUは初期化状態になり、Hレベルにすると、メモリのアドレス0000hから命令を読み込んで動き出す。

$\overline{\text{MI}}$ 信号線は、CPUがメモリから命令を読み込んでいるときにLレベルになる信号線である。

これらの信号線の状態を表すタイミングチャートを図Ⅱ-5、図Ⅱ-6に示す。



図Ⅱ-5 メモリリードライトサイクル



図Ⅱ-6 I/Oリードライトサイクル

(3) マイコンの基本動作

マイコンの基本動作についてエアコンを例に説明しよう。温度センサからの信号や、人間が設定した温度スイッチの情報の入力、またヒータやクーラーのコントロール出力は、マイコンのI/Oに接続されている。

”温度センサの出力を読み込め”や”ヒータをONせよ”というプログラムはメモリに記憶しておく。

電源ONでエアコンが動き出すと、まずマイコンはアドレス0000番地の命令をメモリからCPUに読み込む。

CPUではこの命令は”温度センサの出力を読み込め”であると解釈し、そこで今度はCPUからアドレスバスに温度センサのI/Oアドレスを出力し、センサからのデータを取り込む。

そして、また次の命令”温度スイッチの指示値を読み込め”を読み込み、今度は温度設定スイッチのI/Oアドレスをアドレスバスに出力し、設定スイッチの状態を取り込む。

また、次の命令を読み込み、温度センサからのデータと設定スイッチからのデータとどちらが温度が高いかを判断する。

そして、温度センサから読み込んだ部屋の温度が低ければアドレス0007番地にジャンプする。アドレス0007番地の命令は、”ヒータをONにしろ”という命令であるので、アドレスバスにヒータのI/Oアドレスを出力し、ヒータをONにするという意味のOFFhというデータを出力する。これで室内温度が設定温度になるまで、ヒータがONされる。

このようにマイコンは、命令を読み込んで、その命令に従って別のメモリアドレスのデータを読み込んだり、I/Oアドレスにアクセスしてデータを入出力する。

そして、再び次の命令を読み込んで動作を繰り返す。

演習課題

- 1 マイコン構成要素の現物を実際のマイコンボード上で確認させる。
- 2 マイコンボード上でバスの存在を実感させる。
- 3 具体的なマイコン応用システムのカバーを外させ、実際に使用されている構成要素の型名を調べさせる。

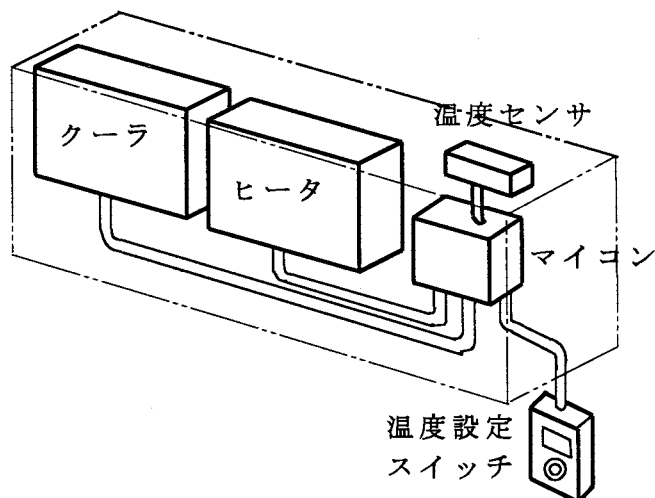


図 II-7 エアコンの内部ブロック図

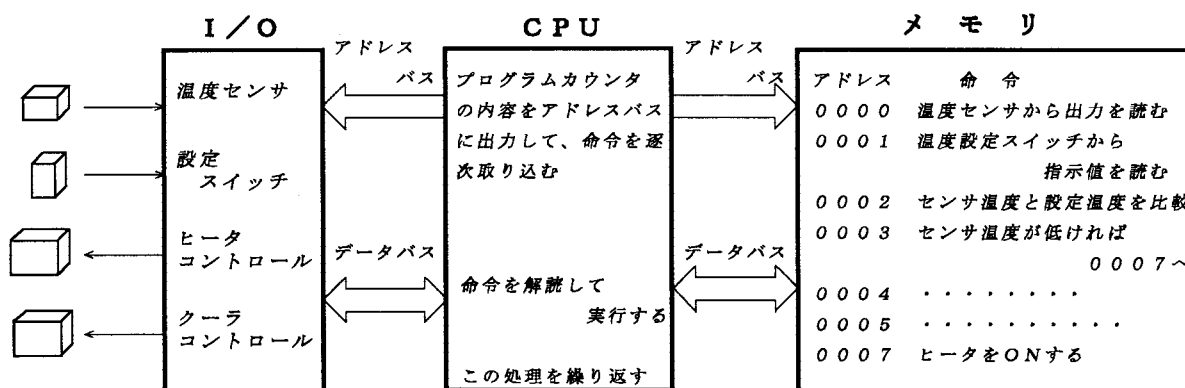


図 II-8 エアコンに内蔵されているマイコンの動作例

2 マイクロプロセッサ

最近、あらゆるシステムにマイコンが入り込み、もうだれもそのことでは驚かなくなつた。それだけ社会の中に定着したようである。また、そのCPUの進歩も著しく8ビットCPUから16ビット、更に32ビットへと次から次へと新しいシステムが発表されている。

このマイコンを構成する中で、中心的なデバイスがマイクロプロセッサであることはいうまでもないが、一般にこのマイクロプロセッサのことをインテル系ではCPU、モトローラ系ではMPUと表現される。ここではマイクロプロセッサのことをCPUとして説明する。

CPUにはインテル系、ザイログ系、モトローラ系など、更に8ビット、16ビット、32ビットなどがあり、その内部構成も様々であるが、すべてがプログラム内蔵方式と逐次処理方式であるので原理的には同じである。そこで、一般的なマイクロプロセッサの内部構成として説明するために、仮想的に拡張COMETをマイコンにしたASC (ASimple Computer) をブロック図で示し、基本部分の機能を説明する。

(1) マイクロプロセッサの内部構成

① 命令解読器 (デコーダ)

CPUがメモリに格納されているプログラムの命令部を取り出し、その命令がどんな命令であるかを解読する部分である。ここで命令を解読して、その命令に対応した信号をCPU制御部へ送る。

② 演算回路

論理・演算をする部分である。演算といっても四則演算や論理演算程度しかできない。これらの計算は、足し算や引き算を組み合わせで計算する。

③ 制御回路

命令解読器で解読された命令によって、CPU内部の各部に制御信号を送る部分である。人間でいえば中枢神経に当たる重要な部分である。

④ アキュムレータ (累算器)

CPUで一番重要なレジスタがアキュムレータである。CPUの各種演算は必ずこのアキュムレータに対して行われ、その演算結果も必ずアキュムレータに格納される。

⑤ プログラムカウンタ

次に、どのアドレスの命令を読み込むかを常に記憶しているのがプログラムカウンタである。メモリから命令を読み込むたびに1ずつ増えていく。また、ジャンプ命令が実行されたときは、そのジャンプ先のアドレスの値が入る。

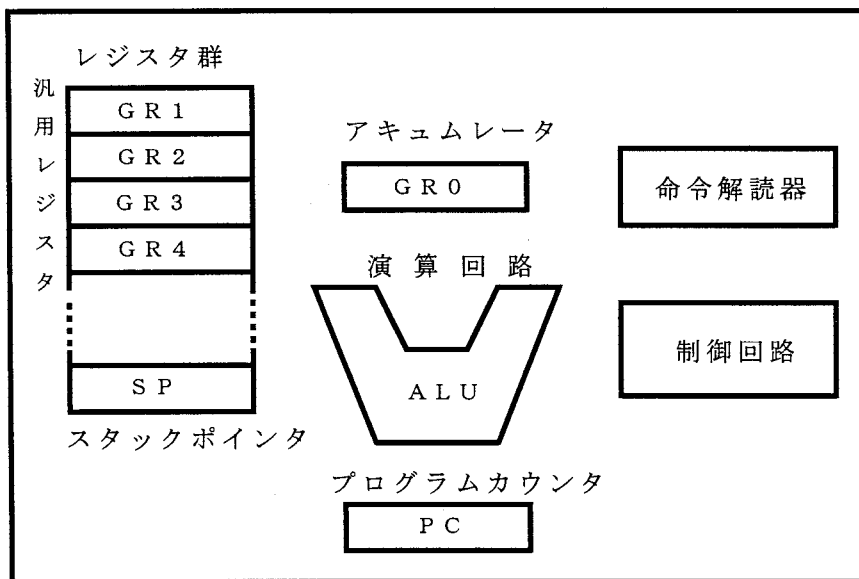
⑥ スタックポインタなど

レジスタの値を一時的に保存したり、サブルーチンコールなどの戻りアドレスなどを保存するために使われるメモリ領域のアドレスを示すカウンタである。

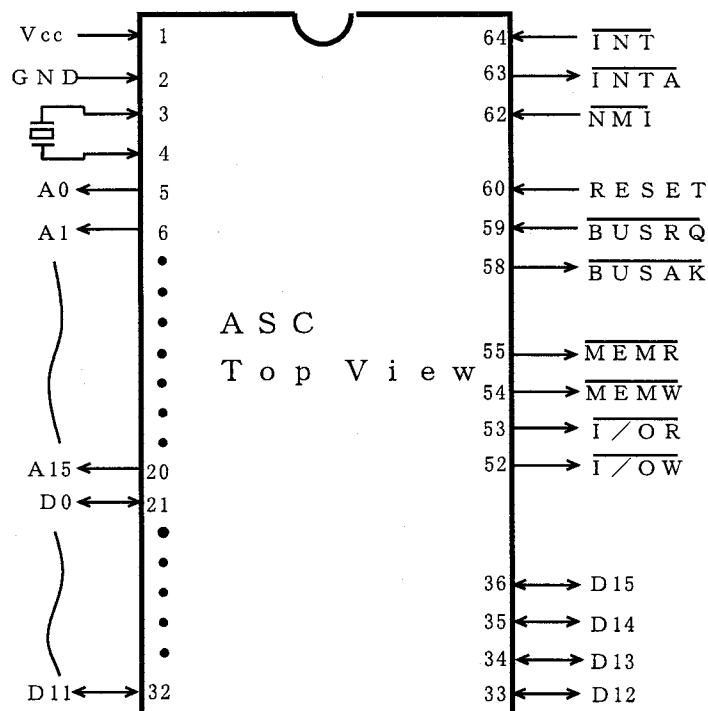
(2) マイクロプロセッサの端子と機能

人間の情報処理機能と比較して、コンピュータの入力、出力、記憶、演算、制御機能をコンピュータの五大機能という。この五つの機能のうちCPUは、制御機能と演算機能を受け持つ。

CPUの端子の各信号を仮想マイコンASCを例に図Ⅱ-10に示す。



図II-9 マイクロプロセッサの内部構成



図II-10 ASCの端子図

(3) 代表的なマイクロプロセッサの種類

今日において代表的と思われるCPUについて幾つか紹介し、それらの特徴などについて説明する。これらCPUの開発経過とその特徴、そしてクロック数の推移を表Ⅱ-1に、最近の代表的なCPUの外観写真を写真Ⅱ-4に示す。

① 4ビットから32ビットへ

マイコンが8ビット単位でデータを扱う場合、CPU内部の命令レジスタ、アキュムレータ、汎用レジスタ、ALUが8ビット幅であり、これらを結ぶ内部データバスも8ビット幅である。このようなものを8ビットCPUと呼び、その代表的なものにインテル系の8085、ザイログ系のZ80、モトローラ系の6809などがある。

16ビットCPUは、これらをすべて16ビット幅に拡張したもので、その代表的なものに8086がある。ただし、この8086は本来命令コードも16ビット幅に拡大すべきであるが、8ビット幅に据え置かれた。

32ビットCPUは、これらすべてを32ビット幅に拡大したもので、代表的なものに80386、i486がある。

8ビットCPUでも16ビット、32ビットCPUでも、メモリやI/Oのアドレスは1バイトごとに付けられる。ただし、16ビット、32ビットCPUでは2バイト分や4バイト分を同時にやり取りできるようになっている。このことは、16ビットのCPUは8ビットのCPUの半分の回数で同じ桁の計算ができるということになる。

しかし、8ビットと16ビット、32ビットCPUの違いは、単純に計算するビット数の違いだけでない。例えば、16ビットの8086は、整数のかけ算とわり算が1命令で実行できるので、8ビットCPUに比べてはるかに高速演算できる。更に、32ビットのi486は、小数の加減乗除を行えるFPU（浮動小数点演算ユニット）を内蔵し、高度な計算を超高速で処理できるようになっている。

② 動作周波数とプロセス

CPUは外部から入力されるクロック信号に同期して動作する。8ビットCPUのZ80のクロック信号の周波数は、当初2.5MHzで1秒間に250万回動作した。命令によって異なるが、一つの命令を実行するのに4クロックから23クロックかかった。

最新のCPUであるPentiumProでは、最高200MHzという高速クロックで動作する。つまり、1秒当たり2億回動作することになる。更に、1命令の実行に要するクロック数も少なくなるように設計されている。足し算、引き算といった基本的な命令は1クロックで処理できる。

CPUの能力の向上、複雑化に伴い、チップ上に作り込まれるトランジスタの数も飛躍的に増加し、表Ⅱ-1に示すように最新のPentiumProでは550万素子が集積化されている。

③ CISCとRISC

CPUの動作は十分に速いけれども、メモリから命令を取り出すのに時間がかかるので、CPUにウェイトがかかり実行速度が落ちてしまう。CPUとメモリ間は高速伝送が必要であることは明らかであるが、大体のソフトウェアの処理では、ほとんどが繰り返し処理で

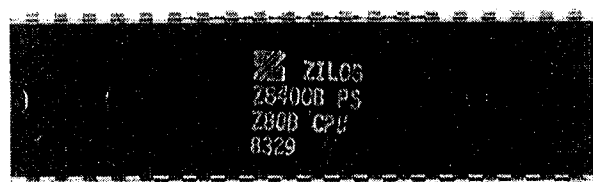
あるので、何度も何度も同じデータを取り出しに行き、そのたびにウェイトをかけられるのではCPUの性能が活かせない。そこで、CPUの内部に高速メモリを内蔵し、同じデータを何度も読みに行くのを避けたのが、キャッシュメモリである。システムメモリを百科事典とすると、キャッシュメモリはメモ帳のようなもので、百科事典の同じページを何度も読みにかずにすむようになる。

また、このCPUとメモリ間のデータ転送時間の差を解消するために、できるだけ少ない命令で仕事ができるようにし、データ流量を減らすことを目的として開発されたのが、CISC (Complex Instruction Set Computer) である。

全く別の発想でこの問題に対処するために、CPUとメモリ間のパイプを太くしてしまえばいいというように、大量の高速メモリを用意し、単機能の命令を大量に使い、その代わりに命令当たりの実行時間を短くして処理能力を向上させようというのが、RISC (Reduced Instruction Set Computer) である。具体的には、使用頻度の高い命令だけを採用し、内部の制御系を単純にして、高速化している。主にEWS (Engineering Work Station : EWS) で用いられている。

表Ⅱ-1 CPUの開発経過とその特徴

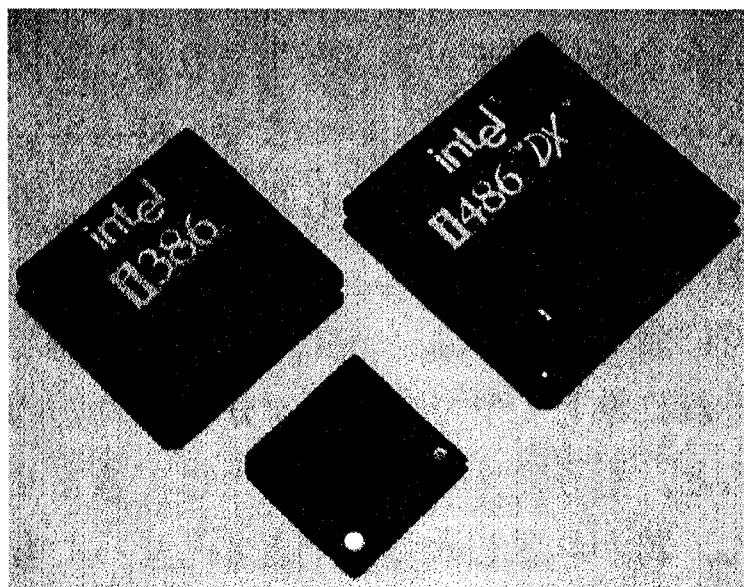
開発年	型 式	特 徴	トランジスタ数 クロック周波数
1971	4004	世界初のCPU。4ビットのデータを基本に扱う4ビットCPUで、電卓用に開発された。	2,200 0.75MHz
1974	8080	世界初のパソコンに应用可能な8ビットCPUである。64kバイトのメモリを使用可能。クロックは2MHz	5,500 2MHz
1978	8086	世界初の16ビットCPU。16ビット処理と1Mバイトのメモリ容量を持つ8086の登場でようやく本格的なパソコンができた。	2万9,000 4~10MHz
1982	80286	8086を高機能化してプロテクトモードにより、16Mバイトのメモリを扱えるようになっている。	12万8,000 6~20MHz
1985	80386	80286をベースに32ビット化したCPU。メモリ容量は4Gバイト	27万5,000 12~33MHz
1989	i486	機能的には80386と同等。ただし、8kバイトのキャッシュの内蔵や、命令実行ステップの短縮により、大幅なスピードアップを実現	120万 16~66MHz
1993	Pentium	i486をさらに高速化したCPU。コード用8kバイト、データ用8kバイトのキャッシュ、2命令同時実行、分岐予測、基本命令を専用回線で高速実行できる。	310万 60~166 MHz
1995	PentiumPro	Intelの最新CPU。x86命令をRISC命令に変換して実行する。レジスタリネーミング、アウトオブオーダー実行といった新技術が投入されている。	550万 150~200 MHz



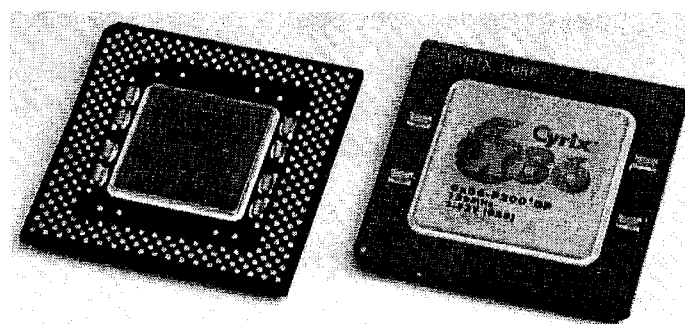
写真II-4 ザイログ社のZ80CPU (8ビット)



写真II-5 インテル社の8086CPU (16ビット)



写真II-6 米インテル社の80286、i386、i486CPU



写真II-7 米Intel社の200MHz動作Pentiumと米Cyrix社のCyrix6x86-P200+

3 メモリ

(1) メモリの役割

メモリは”1”か”0”かという情報を記憶する素子である。アドレスを指定してプログラムやデータを読み書きするための素子がメモリである。

このメモリも半導体技術の進歩で大容量化が進み、パソコンに数十Mバイトのメモリを積むのは当たり前という状況である。もともとコンピュータの構成には記憶装置すなわちメモリ素子なしでは成り立たないものであるが、このメモリ素子としてICメモリが使われている。メモリICは大きくROM (Read Only Memory) とRAM (Random Access Memory) とに分類される。

ROMが読み出し専用メモリであるのに対し、RAMは書き込みも読み出しもできるICメモリである。

(2) メモリの種類と特徴

メモリを機能により分けると、揮発性メモリと不揮発性メモリに分類できる。揮発性メモリは、電源を切るとデータが失われてしまうメモリのことである。読み書きやランダムアクセスが自在にできるので、RAM (Random Access Memory) とも呼ぶ。RAMは、記録する方法により、DRAM (Dynamic RAM) とSRAM (Static RAM)、画像メモリのような特殊用途向けのメモリとに分けることができる。

電源を切ると内容が失われるので、電源ONですぐに動作させるマイコンでは、RAMにプログラムを記憶させておくことはできない。主にRAMはデータ領域やスタックエリアとして使われる。

不揮発性メモリは、電源を切ってもデータが失われないメモリのことである。基本的には書き込みができず読み出し専用なので、ROM (Read Only Memory) とも呼ぶ。

ICメモリの種類と開発経緯を図Ⅱ-11と図Ⅱ-12とに示す。

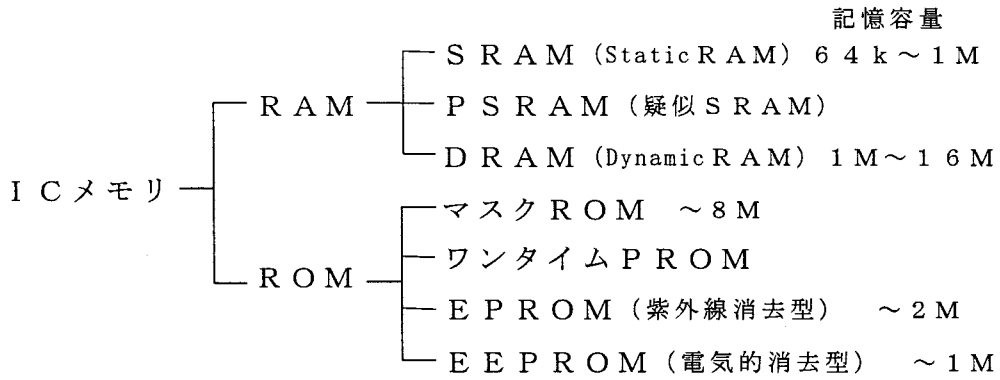


図 II-11 ICメモリの分類

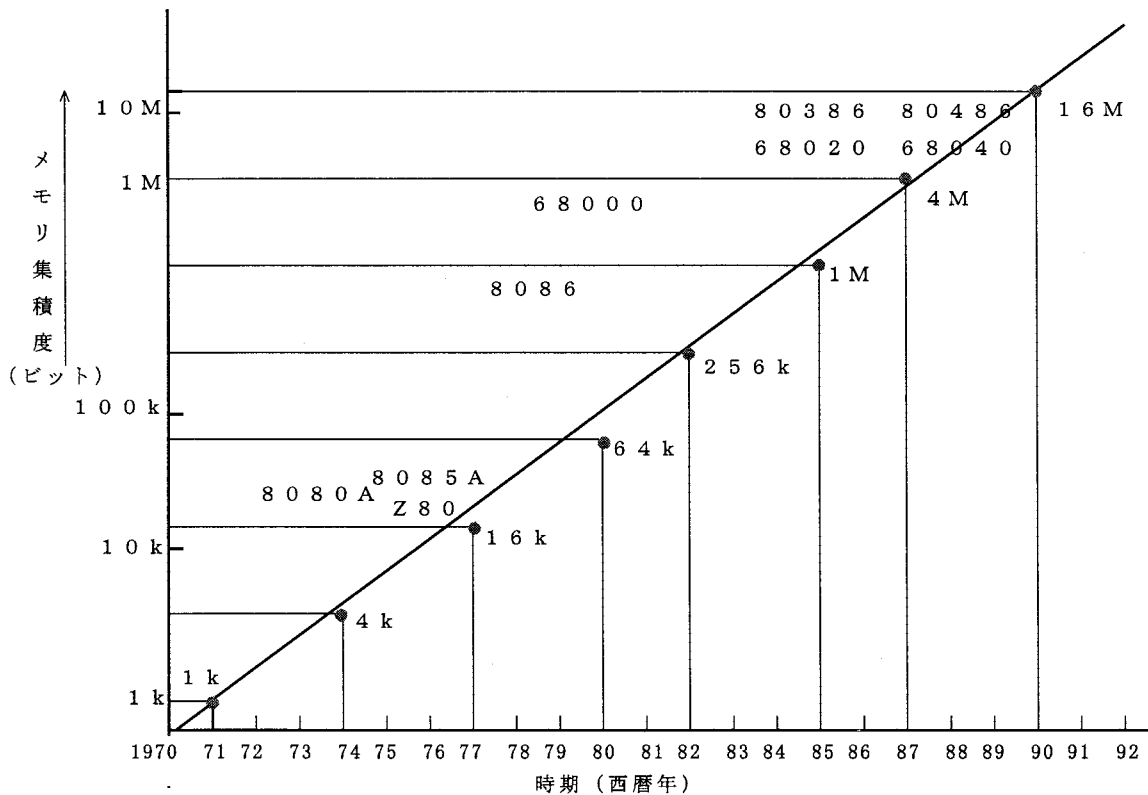


図 II-12 ICメモリの開発経緯

① RAM (ダイナミック型、スタティック型)

DRAMは”0”又は”1”の記憶部分(メモリ・セル)が一つのコンデンサと一つのスイッチングトランジスタにより構成されており、このコンデンサに電荷を充電することで”0”又は”1”の情報を記憶する。記憶部分がコンデンサであるため、リーク電流により電荷が減少してしまい、放っておくと内容が消えてしまう。このため、一定時間ごとに一度内容を読み出し、再度書き込みを行わなければならない。この作業をリフレッシュという。

同じRAMの仲間のDRAMとSRAMのセル構造を図Ⅱ-13に示すが、DRAMのセル構造はSRAMに比べ構造が簡単であることがわかる。この単純な構造のため、DRAMはSRAMに比べ大容量化しやすく、ビット当たりの単価も安い。そのため、パソコンなどの大容量システムでは主メモリとして使われるが、8ビット程度のマイコンシステムではあまり使われない。8ビットのCPUで使うRAMとしてはSRAMを使うのが一般的である。

SRAMはまさにフリップフロップを並べたメモリと考えることができる。フリップフロップの原理により”0”又は”1”を記憶しているため、DRAMのリフレッシュのような処理は必要ない。

② ROM (マスクROM、EPROM、ワンタイム)

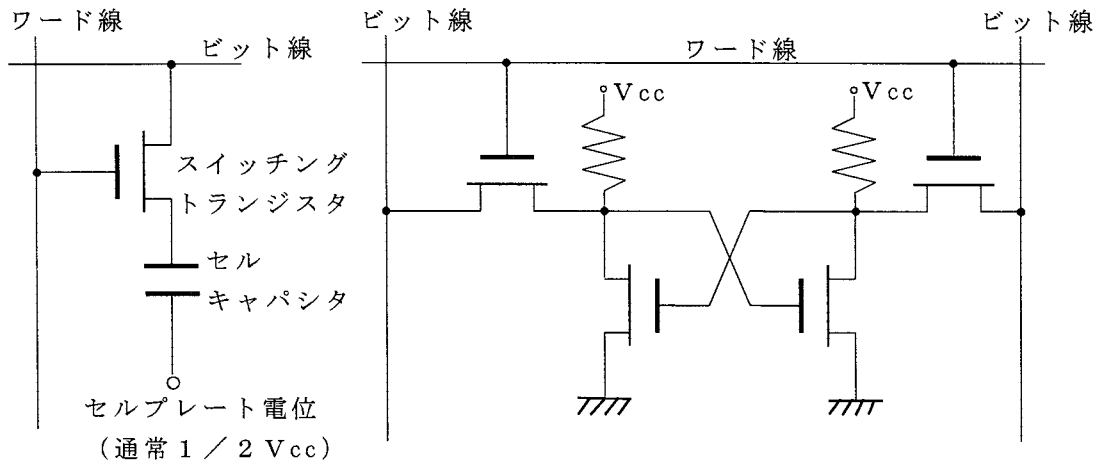
不揮発性メモリは、電源を切っても内容が消えないので、電源ONで動作する組み込み機器のためのプログラムや初期データ用のメモリとして使われる。

不揮発性メモリにはIC製造時(ICを作るときのフィルム)で内容が決まってしまうマスクROMと、ユーザ側でも専用回路(ROMライター)によりデータを書き込むことのできるEPROMとに分けられる。

更に、EPROMはデータの消去方法の違いにより、紫外線で消去(写真Ⅱ-9に示すように紫外線を照射するための窓がICの中心部にある。)が可能なUV EPROM、電氣的に消去が可能なEEPROM、そして、最近使われるようになってきたEEPROMの親戚でもある一括消去型のフラッシュROMに分けられる。

揮発性メモリは、更にDRAMとSRAMに分けられる。図II-13にDRAMとSRAMの内部構造図を、写真II-8にその外観を示す。

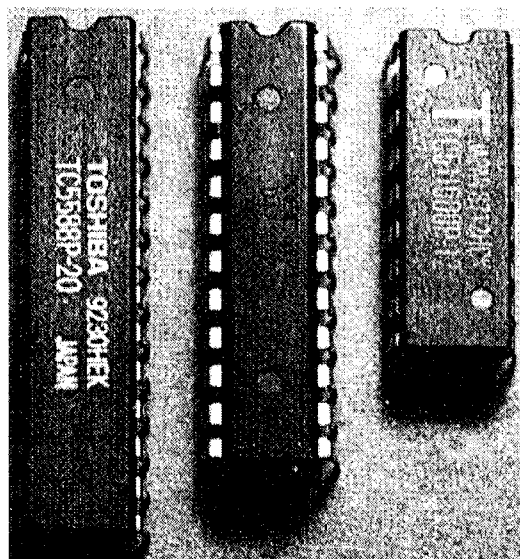
また、写真II-9にUV EPROMの外観を示す。



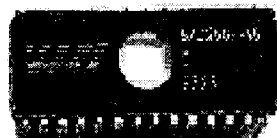
(1) DRAM

(2) SRAM

図II-13 DRAMとSRAMのメモリセル構造の違い



写真II-8 SRAMとDRAMの外観



写真II-9 EPROMの外観

4 入出力ポート

入出力ポートとは、コンピュータシステムと外部の何らかのシステムとの間で、データのやり取りを実現するための窓口である。すなわち、入力ポートより外界からデータを取り込み、コンピュータ内部で処理したデータを出力ポートより外部周辺機器に出力するというように入出力ポートはCPUと周辺機器との仲介（インタフェース）を行っている。

(1) ポートの必要性

マイコンシステムの場合、外部の情報をシステム内部に取り込み、決められた処理を行い、その結果を再び外部に知らせるといように、外部装置とデータのやり取りをする窓口が必ず必要になる。この外部とデータのやり取りをする窓口に当たる部分を入出力ポートという。

一般に、コンピュータの周辺機器の処理速度は、CPU内部でのデータ処理速度に比べはるかに遅い。入出力ポートはこのような処理速度の違いを吸収する一種のバッファ的な役割をするとともに、データ形式や信号レベルの変換や動作タイミングの調整なども行う。

CPUに接続するメモリと入出力ポートの接続回路の違いは、データの入出力動作に対して、基本的には \overline{MREQ} がアクティブになるか、 \overline{IOREQ} がアクティブになるかの違いだけである。アドレスバスに固有の番地をアクセスする信号が出力され、 \overline{RD} 又は \overline{WR} により、読み込みか、書き込みの選択をする部分は、メモリについても入出力ポートについても基本的には同じである（図II-5、図II-6参照）。

マイコンシステムでは、データバスが重要な働きをしている。アドレスをセットし、デコーダ回路で相手の一つだけを選び、コントロール信号で制御しながらデータバスを通じて選択された装置とデータのやり取りをする。

バスを使うことにより、マイコンは大量のデータの受け渡しを、装置間で信号線を増やさずに容易に行える。

この反面、時分割で使われるので、外部装置側にデータをラッチする機能とバスをバッファリングする機能が必要になる。

このようにCPUが外部のデータを読込みたいときに、外部からのデータをデータバスに出力するもの、またCPUが外部にデータを出力したいときに、データバスからデータを入力して外部に出力するものなど、入出力をコントロールするラッチやバッファが入出力コントローラである。

(2) 入力ポート

入力ポートからのデータは、アドレスが確定してから \overline{IORQ} と \overline{RD} 信号がアクティブになってT₃ステートの立ち下がりでデータが取り込まれる（図II-6参照）。

したがって、このタイミングで入力すべきデータをCPUのデータバスに乗せてやる必要がある。また、データバスは双方向性であるので、入力ポートに対してその要求がないときには電氣的に切り離された状態にしなければならない。これらのことを実現するために入力ポートでは、バスバッファがよく使われる。

外部装置からデータを取り込む場合、通常IN命令が使われる。例えば、「外部装置から入力ポート(OFAH)を介して、データを取り込む。」という場合は、IN A、(OFAH)と

いう命令を実行すると、入力装置から取り込まれたデータは、入力ポートを介してデータバスよりCPUに取り込まれる。

(3) 出力ポート

CPUが出力(OUT)命令を実行すると、図II-6に示すようにまずI/Oアドレスが出力され、続いてデータが出力される。このとき、I/Oに対する書き込みであることを伝えるために \overline{IORQ} と \overline{WR} 信号がアクティブになる。

このように出力データは、時間的にみれば瞬間的に現れる。よって、このデータを何らかの方法で固定しなければならない。

一番基本的な8ビットの平行出力ポートとしては、8ビットのD型ラッチ回路がよく使われる。

また、入力ポートの場合と同様に、「CPUで処理したデータを出力ポート(OFDH)を介して、出力装置へ書き出せ。」という命令であるOUT A、(OFDH)を実行すると、CPUからのデータがデータバスより出力ポートを介して、出力装置に出力される。

(4) 汎用のポート

各種周辺LSIは、それぞれメインとなるCPUと密接な関係を持ち、比較的汎用性の高いLSIファミリなどと呼ばれる。

データ処理の基本となる8ビットの平行データを扱う平行入出力用LSI、ビット列のシリアルデータと平行データを相互に変換するシリアル入出力用LSI、パルスをカウントしたり一定時間を作るカウンタ/タイマ用LSIなどは、マイコンシステムの中で必要なものである。