VI 開発支援ツールの使い方

学習目標
1　マイコン応用システムの各開発段階において、どの開発支援ツールを使用すべきかを学習させる。
2　ロジックアナライザの機能と基本的な使い方を理解させる。
3　インサーキットエミュレータの機能と基本的な使い方を理解させる。
4　シミュレータの機能と具体的な使い方を理解させる。
5　プロトコルアナライザ、ROMライタの機能と基本的な使い方を理解させる。

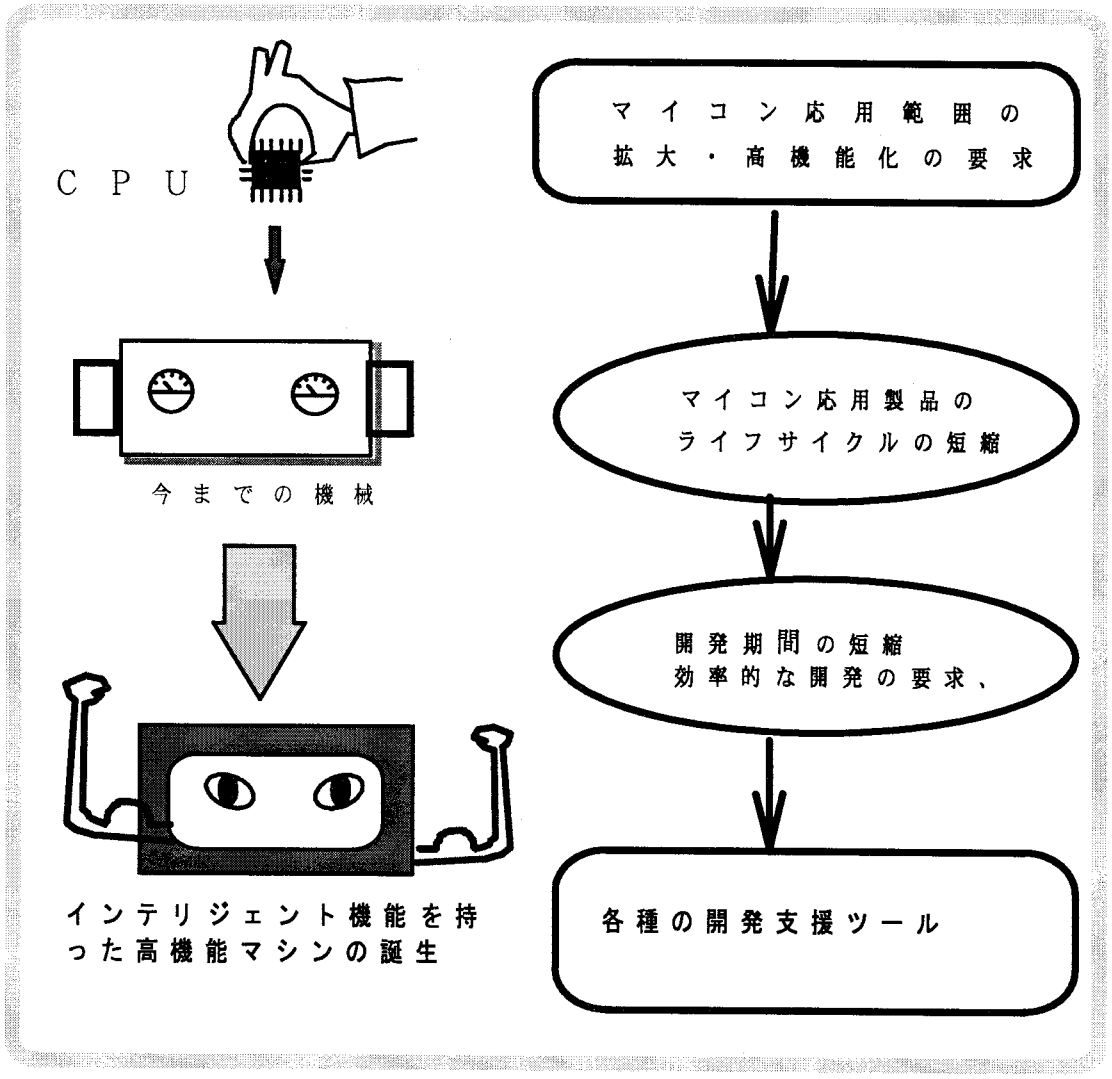
学 習 の 項 目 と 内 容

項　　目	内　　容
1　支援ツールの種類	代表的な開発支援ツール マイコン応用システムの開発作業 開発段階で使用するツール ツールを使う考え方
2　ロジックアナライザの使い方	ロジックアナライザの外観 ロジックアナライザの用途 ロジックアナライザの機能 ロジックアナライザのセッティング
3　インサーキットエミュレータの使い方	ＩＣＥの外観 ＩＣＥの用途 ＩＣＥの機能
4　シミュレータの使い方	シミュレータの用途
5　プロトコルアナライザの使い方	プロトコルアナライザの外観 プロトコルアナライザの用途 プロトコルアナライザの機能
6　ROMライタの使い方	ROMライタの外観 ROMライタの用途 ROMライタの機能 ROM化の準備

1 支援ツール

マイコンの普及によりこれまでの機械に簡単にコンピュータ制御を組み込むことができるようになった。これは、それまでの機械に高度な機能を組み込むことを意味しており、低コストで、高機能の機械・インテリジェントマシンが作れるようになった。

マイコン応用システムの需要の拡大に伴い 更なる高機能化の要求が生まれた。そして、それが更にマイコン応用製品のライフサイクルの短縮に繋がった。開発期間は、短くなり効率のよい開発環境が要求されるようになった。そして、各種の開発ツールが使われるようになった。



図Ⅵ－１　マイコン搭載によるマシンの高機能化と開発期間の短縮

ソフトウェア開発の部分では、従来の情報処理システム開発と同じツールが用いられる。しかし、マイコン応用システムでは、ハードウェアと密接に関連する部分が多いためマイコン応用システム固有の開発環境ツールを使用することが多い。

(1) 代表的な開発支援ツール

① ロジックアナライザ

主にハードウェアのテストを行うツールである。多数の信号を同時にしかも高速にサンプリングして記憶し、後でディジタル信号のタイミング波形や０と１のステイトテーブルで画面に表示して確認することができる。

② インサーキットエミュレータ（ＩＣＥ）

主にハードウェアとソフトウェアの結合環境のテストに威力を発揮するが、ハードウェアとソフトウェアの結合テストでも大きな力を発揮する。ターゲットマシンのＣＰＵソケットにプローブと呼ばれるコネクタを差し込んで接続し、ＣＰＵやメモリの動作を代行しながらハードウェアとソフトウェアのデバック作業を支援する。

③ シミュレータ

主にソフトウェアのテストに威力を発揮する。ターゲットマシンの環境を開発マシン上に擬似的に作り出すソフトウェアツールである。

④ プロトコルアナライザ

ターゲットマシンと周辺装置の間のシリアル通信を、タイミング波形やステートテーブルで確認することができる。また、ターゲットマシンと接続する周辺装置の代わりにターゲットマシンとの通信を擬似的に行うエミュレーション機能を持っている。

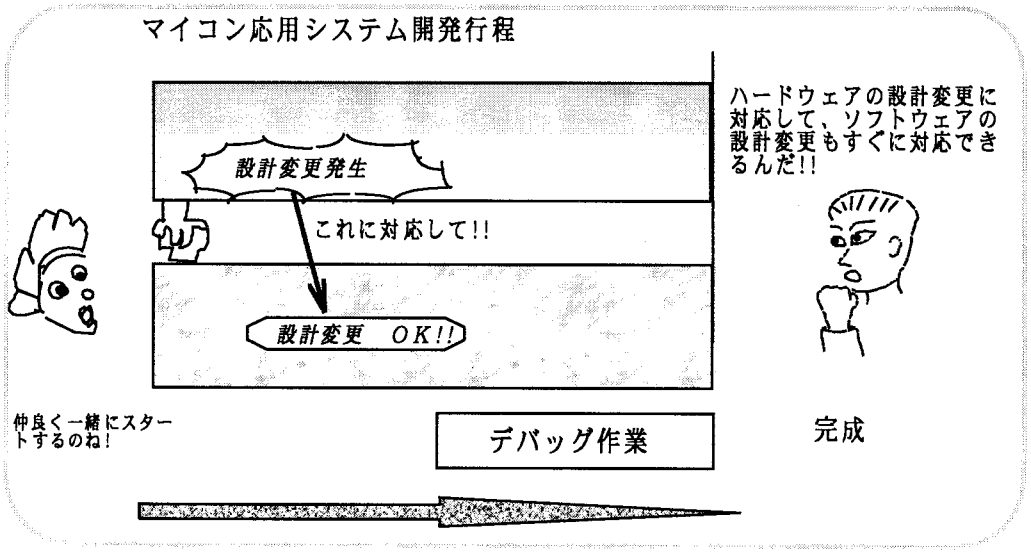
⑤ ＲＯＭライタ

開発したプログラムをＲＯＭに書き込んでターゲットマシンに組み込むための装置。

⑥ クロスコンパイラ

ターゲットマシン用のソースプログラムをＣＰＵの異なるプログラム開発マシンでコンパイルし、ターゲットマシン用のオブジェクトプログラムを生成するためのソフトウェアツールである。

(2) マイコン応用システムの開発作業



図Ⅵ－２　マイコンシステムの開発工程

マイコン応用システムの開発作業は、ハードウェア開発とソフトウェア開発を並行して進める。それはソフトウェアのテストを行うためにはハードウェアが必要になり、ハードウェアのテストを行うためにもソフトウェアが必要になるからである。もし、ソフトウェアテストときにハードウェアが使用できない場合はテスト用のハードウェアを準備・開発することになる。また、逆にハードウェアのテストときに、ソフトウェアが使えない場合は、ハードウェアのテスト用のソフトウェアが必要になる。もし、並行でなくそれぞれ単独で作業を行うと、ハードウェア又はソフトウェアの設計変更が、もう一方の設計変更につながり開発期間の延長を生じさせることになる。

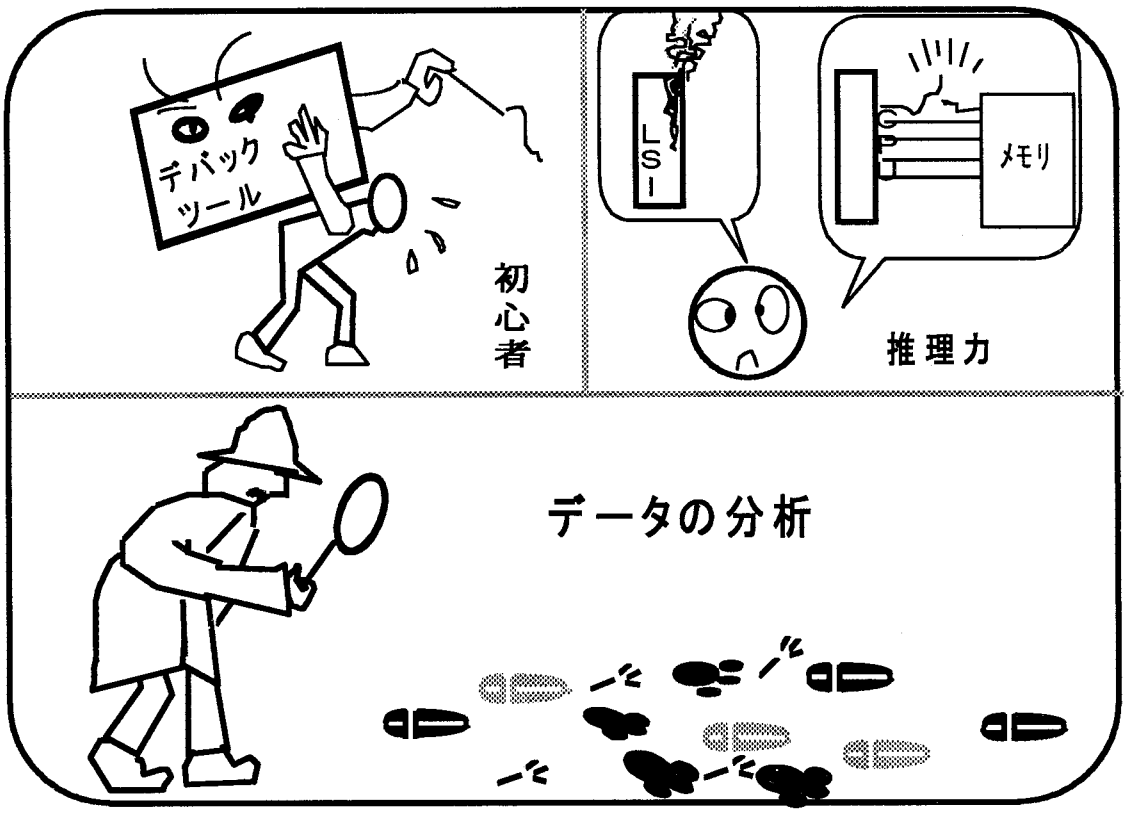
デバッグ作業を効率よく行うためには、ハードウェア、ソフトウェアそれぞれの部分だけで行えるテストは、事前に十分にテストを行いデバッグを済ませておく必要がある。そのためには最小限のテスト用ソフトウェア、テスト用ハードウェアを準備する必要がある。

ハードウェア開発	・ テスター ・ オシロスコープ ・ ロジックアナライザ ・ ＜ＰＬＤライタ＞
ソフトウェア開発	・ エディタ ・ コンパイラ ・ アセンブラ ・ リンカ ・ ＜クロスコンパイラ＞ ＜クロスリンカ＞ ＜クロスアセンブラ＞
デバッグ作業	・ シュミレータ ・ インサーキットエミュレータ（ＩＣＥ） ・ ＲＯＭライタ

図Ⅵ－３　各工程で必要な主なツール

(3) ツールを使うときの考え方

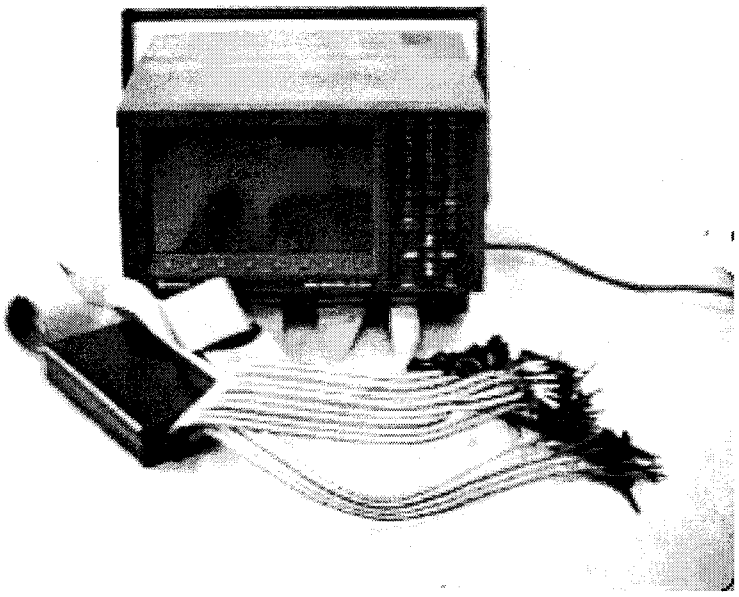
初心者は、ツールを使うとその使い方だけで手一杯になってしまいがちである。また、デバッグ作業とツールとの関係を固定的に考えがちである。デバッグ作業は想像的に行わなければならない。推理小説を読んで犯人を特定するために、いろいろと想像し推理するのと同じである。ただし、そのとき推理の裏付けとなる証拠をつかむためにデバッグツールを使って証明するのである。また、その逆の方法も大事である。皆目見当のつかないバグを探すためには、ベテランの刑事がやるように地道にデータを集め確認することも必要である。その地道な作業を行う際も、デバッグツールを上手に活用しなければならない。どんなデータを取り出すのかが先であり、そのために使えるツールは何かを後から考える方法がよい。ツールのためのデバッグでなくデバッグのためのツールである。



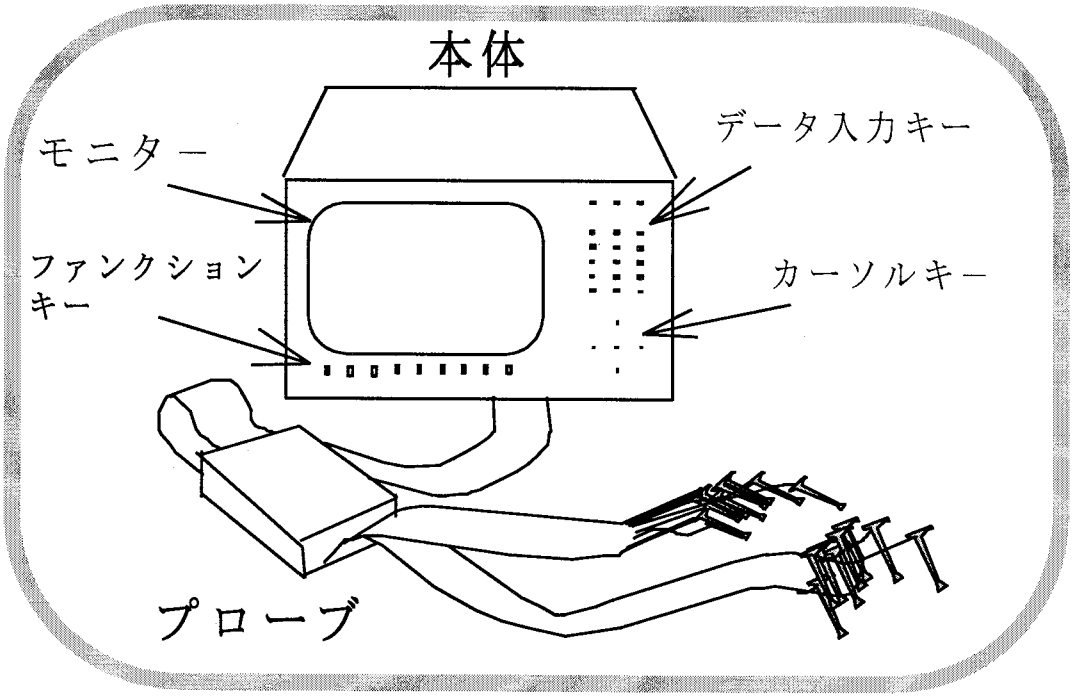
図Ⅵ－４　ツールを使うときの考え方

2 ロジックアナライザの使い方

(1) ロジックアナライザの外観



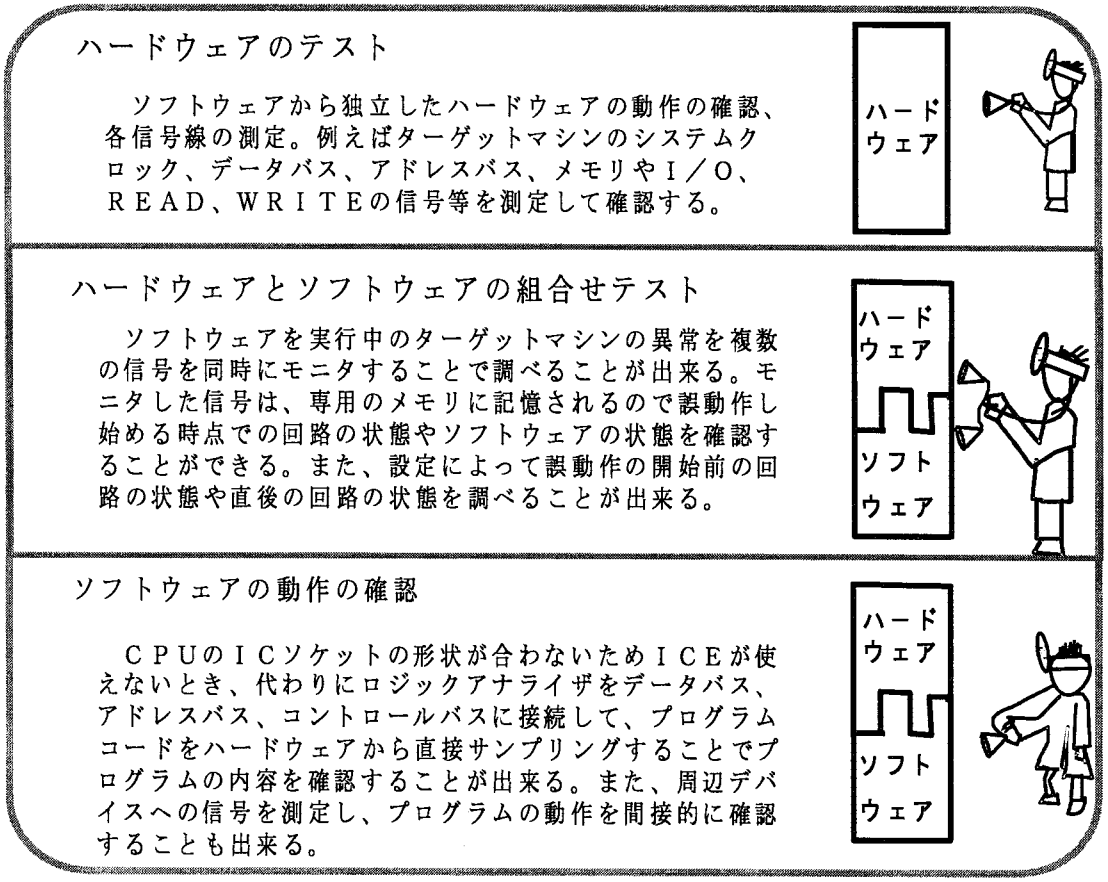
図VI－５　ロジックアナライザの写真



図VI－６　ロジックアナライザの外観図

(2) ロジックアナライザの用途

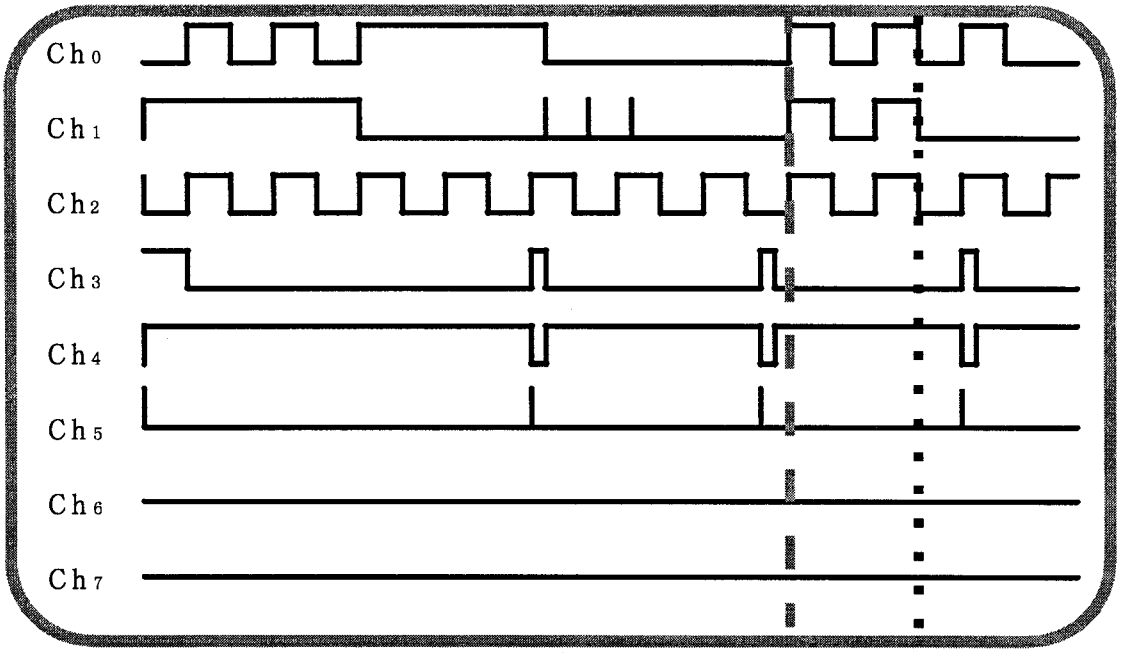
ロジックアナライザは、ターゲットマシンの複数の端子の信号を同時にモニタして分析する装置である。したがって、ロジックアナライザによるデバックは、ハードウェアのテストが中心となるが、ハードウェアとソフトウェアの組合せテストやソフトウェアの動作確認テストにも利用できる。



図Ⅵ－７　ロジックアナライザの用途

(3) ロジックアナライザの機能

多くの信号(入力信号経路をチャネルと呼ぶ。)を同時にサンプリングし記憶することができる。サンプリングしたデータは、タイミングチャート又はステートテーブルに表示して解析することができる。



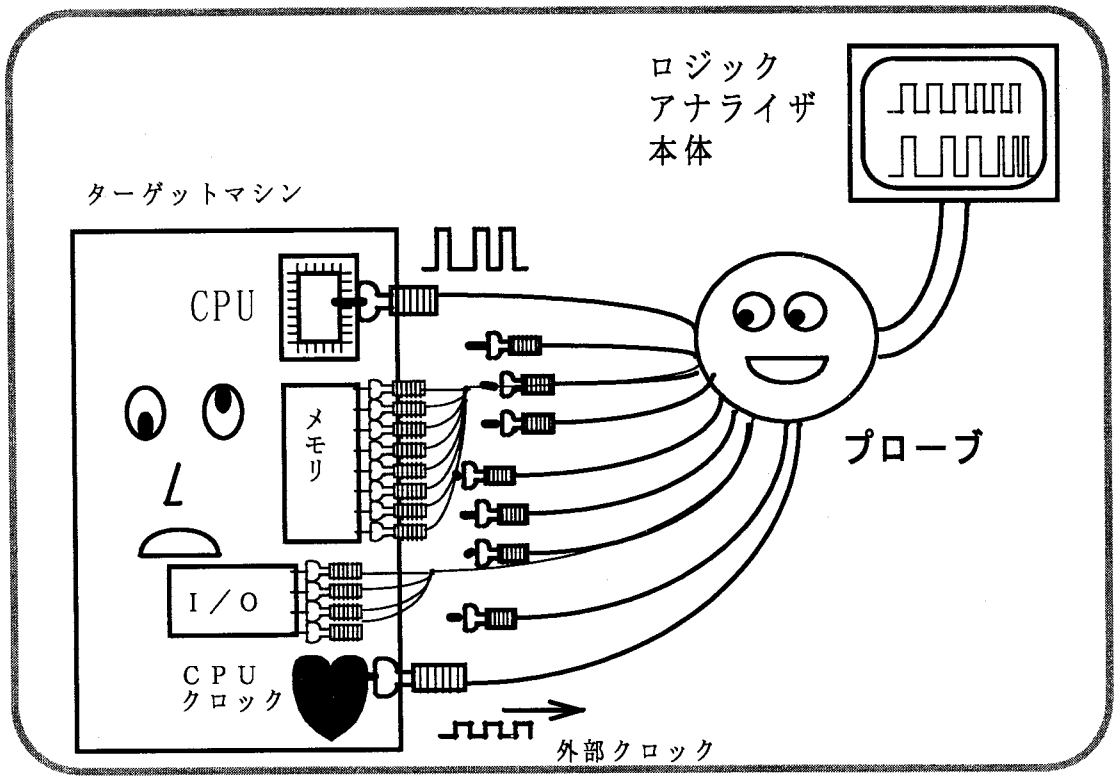
図VI－8　タイミングチャート

	HEX表示	BIN表示
0	3 1	0011 0001
1	0 0	0000 0000
2	E D	1110 1101
3	C D	1100 1101
4	7 C	0111 1100
5	8 5	1000 0101
6	3 E	0011 1110
7	8 0	1000 0000
8	D 3	1101 0011
9	8 3	1000 0011

図VI－9　ステートテーブル

(4) ロジックアナライザのセッティング

① 被測定物との接続



図Ⅵ－１０ ターゲットマシンとの接続

a 外部クロックは、ターゲットマシンのＣＰＵクロック等外部クロックとして使用したいターゲットマシンの端子につなぐ。なければターゲットのグラウンド端子につなぐ。

b データプローブは、測定したい信号のあるテストピン、コネクタ、ＩＣ端子等につなぐ。

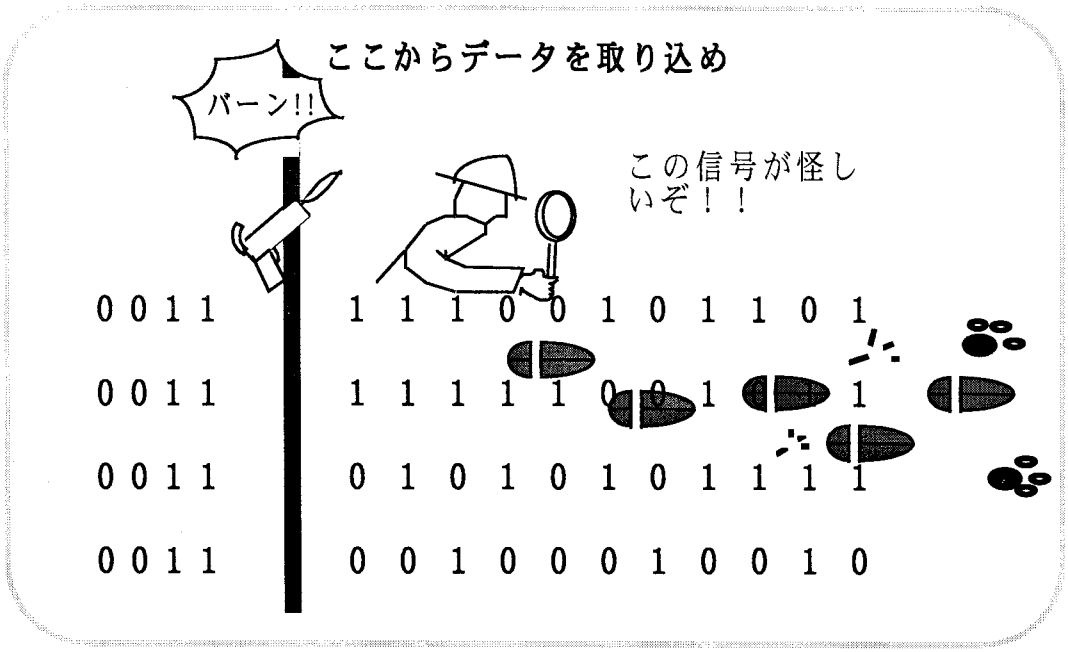
② ロジックアナライザの初期化

設定項目はたくさんあるので基本的なものを取り上げる。

a スレッショルド電圧レベルの設定： チャネルごとのスレッショルド電圧レベルと、０Ｖと５Ｖのどちらかを Ｈとみなすか決める。

b クロックの設定： データを取り込むタイミングの基準となるクロック(サンプルクロック)をターゲットマシン上の信号(外部クロック)とするか、ロジックアナライザ内部で発生したクロックを使用するかどうかを決める。内部クロックの場合その周期も決める。

c トリガの設定： 解析したい現象を取り込むために、その現象の現れるタイミングを表すトリガ信号を設定する。



図VI－１１　トリガの働き

③　ステートテーブルと逆アセンブル表

サンプリングした複数の信号を16進数、10進数、８進数、２進数などの数字に表示したものをステートテーブルという。オプションでアセンブラの命令に逆アセンブルする機能を持つものもある。この機能によってデータバスやアドレスバス、実行したプログラムの命令コードを調べることができる。

	HEX表示	BIN表示	逆アセンブルリスト
0	3 1	0011 0001	LD　SP, E000
1	0 0	0000 0000	
2	ED	1110 1101	
3	CD	1100 1101	CALL　857CH
4	7 C	0111 1100	
5	8 5	1000 0101	
6	3 E	0011 1110	LD　A, 80
7	8 0	1000 0000	
8	D 3	1101 0011	OUT　(83H), A
9	8 3	1000 0011	
10	0 6	0000 0110	LD　B, 08H

図VI－１２　逆アセンブル

④　スレッショルド機能

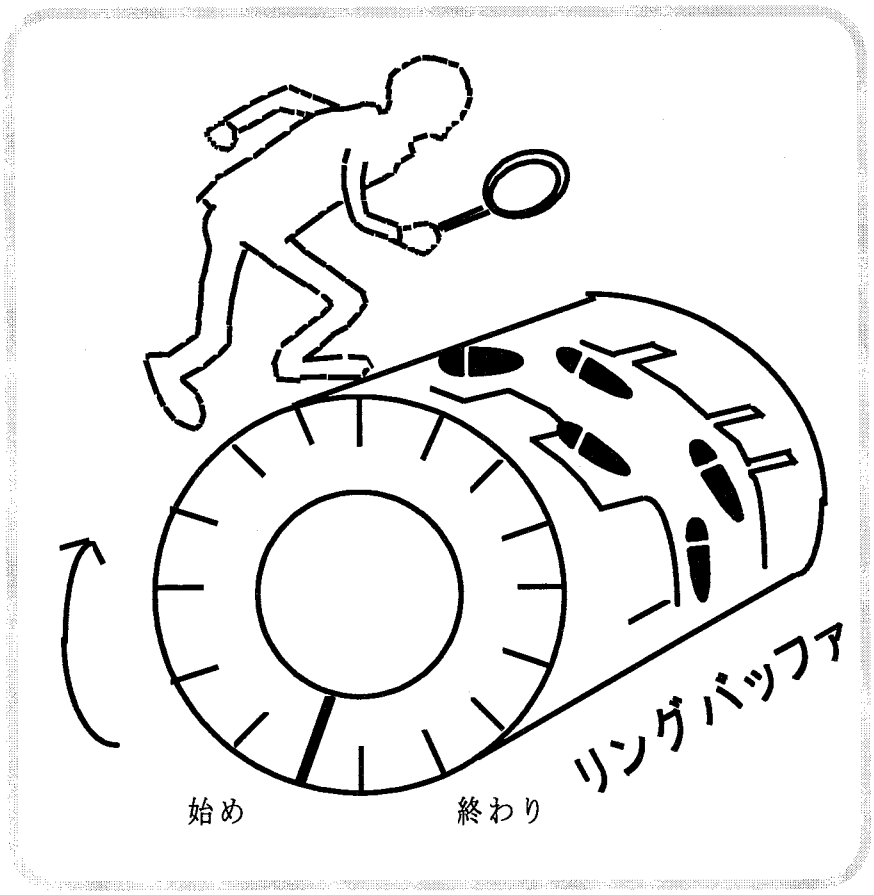
測定する信号は厳密にいうとディジタル信号に近いアナログ信号である。これをスレッショルドと呼ばれる電圧レベルを境にしてHighとLowの信号にみなしている。

⑤　サンプリング周期の設定

測定する信号の性質によってサンプリング周期・サンプリングクロックを自由に設定することができる。その目安は、測定する信号の変化の２倍以上(サンプリング周期でいうと半分以下)で測定しなければならない。また、非周期的な信号であるデータバスやアドレスバスの内容を確認するためには、外部クロックを目的に合わせて選ばなければならない。例えば、メモリからＣＰＵへ送られるデータを測定したければ、データ取り込みタイミングを表す入力信号・外部クロックとしてメモリのＯＥ端子（ＲＤ端子）の信号を設定しなければならない。

⑥　メモリ機能

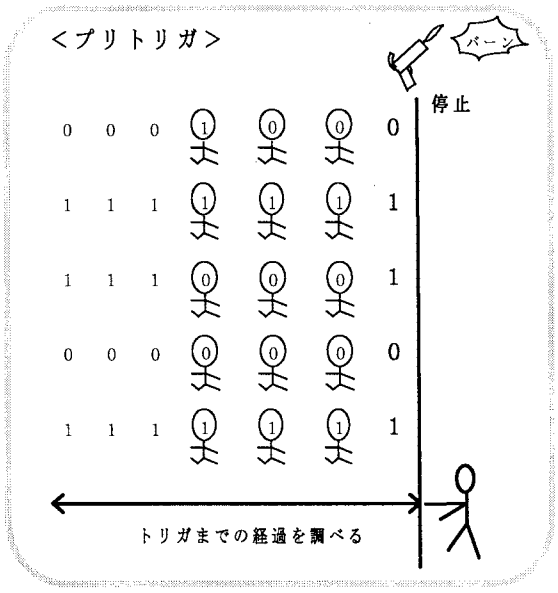
サンプリングしたデータを記憶しておく機能である。これにより高速にサンプリングしたデータを何度でも表示して解析することができる。また、この記憶装置はリング状になったメモリ(リングバッファ)が使われる。これは、一定以上のデータを書き込むと最も古いデータが上書きされる仕組みであり、常に新しいデータが記憶される。



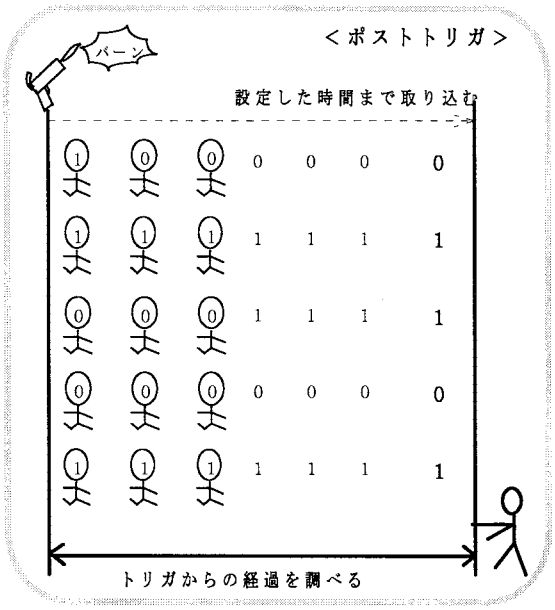
図Ⅵ－１３　リングバッファ

⑦　トリガ機能

トリガ(Trigger)とは、拳銃の引き金という意味であるが、ここでは測定信号の取込みを停止するためのきっかけという意味に使う。測定したい信号がある条件を満たしたところで解析をするのである。調べるところ(時点)が、トリガの前にあるのか、後ろにあるのかによってプリトリガとポストトリガを使い分ける。プリトリガは、トリガが発生するまでの経過を調べる場合に使う。トリガが発生するとその時点でのサンプルデータをメモリに書き込んで停止するので、その直前のデータが多く残る。ポストトリガが発生すると、設定した遅延時間の間サンプリングを続け、その後で停止する。また、トリガには信号のレベルによるレベルトリガと、信号の０か１への立ち上がりや、その逆のたち下がりによるエッジトリガがある。



図Ⅵ－１４　プリトリガ



図Ⅵ－１５　ポストトリガ

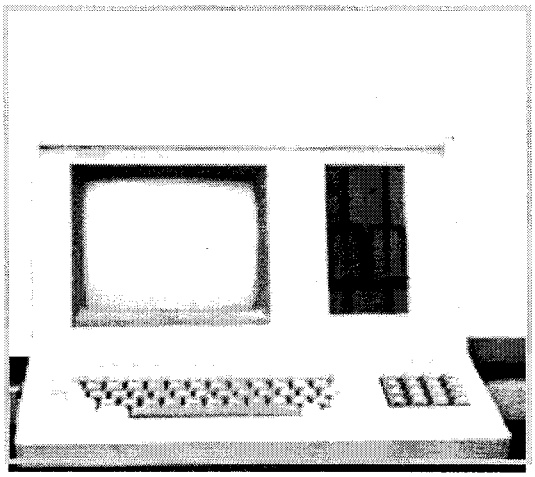
⑧　探索機能と比較機能

サンプリングデータから、設定データ値と等しいものの個数と位置を見つける探索機能や、あらかじめメモリテーブルにデータパターンを登録しておき、サンプリングデータと比較する機能がある。

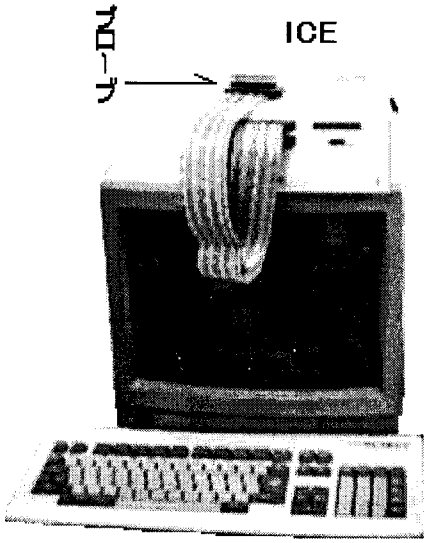
3 インサーキットエミュレータ(ＩＣＥ)の使い方

(1) ＩＣＥの外観

インサーキットエミュレータのターゲットマシンのＣＰＵの代わりにＣＰＵソケットに差し込み、ＣＰＵの動作を代行し、ハードウェアやソフトウェアのデバッグ作業を支援するツールである。

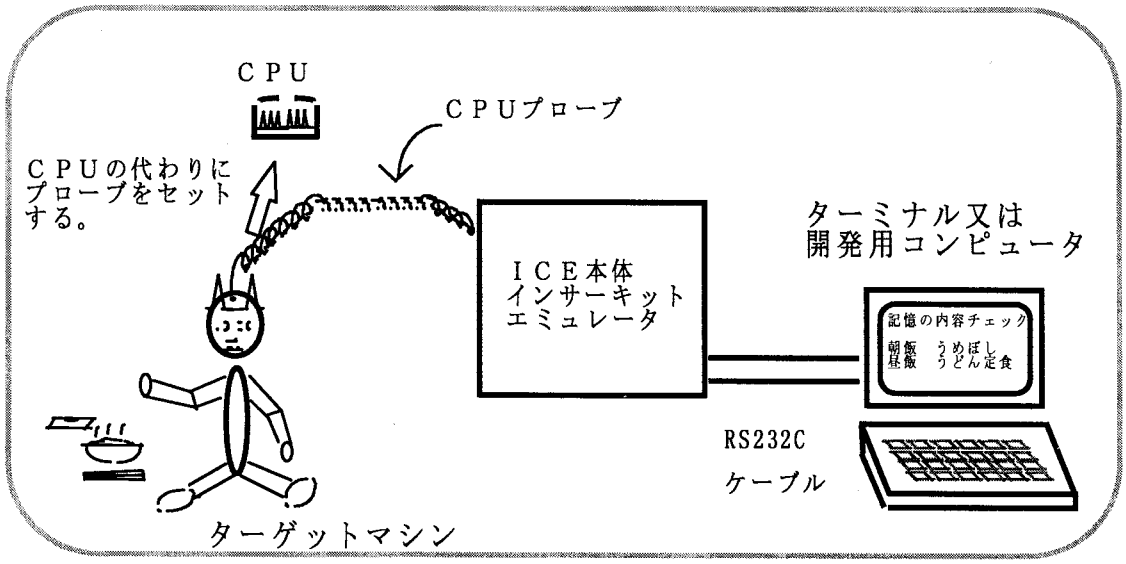


図Ⅵ－１６　ﾓﾆﾀｰ搭載タイプのＩＣＥ



図Ⅵ－１７　ｺﾝﾋﾟｭｰﾀ接続タイプのICE

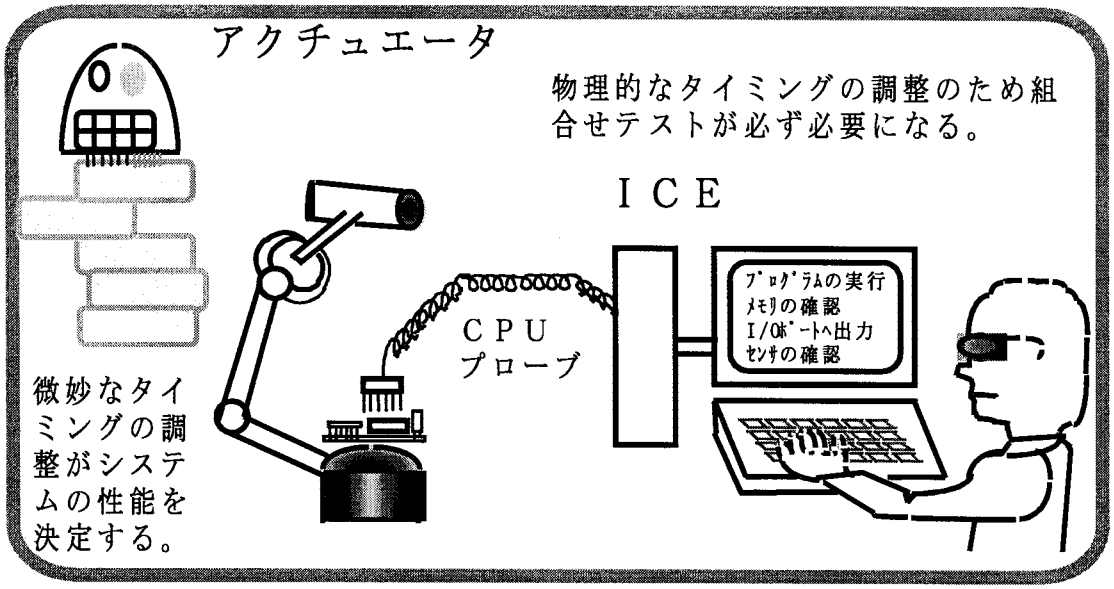
ＩＣＥは、ターゲットマシンとの接続に使うＣＰＵプローブの部分とＩＣＥ本体部分の二つの部分からできている。この本体にモニターとキーボードが付いているタイプと、直接コンピュータと繋いで使うタイプとがある。直接コンピュータに繋がれば　高機能のデバッカーソフトウェアが使用できる。



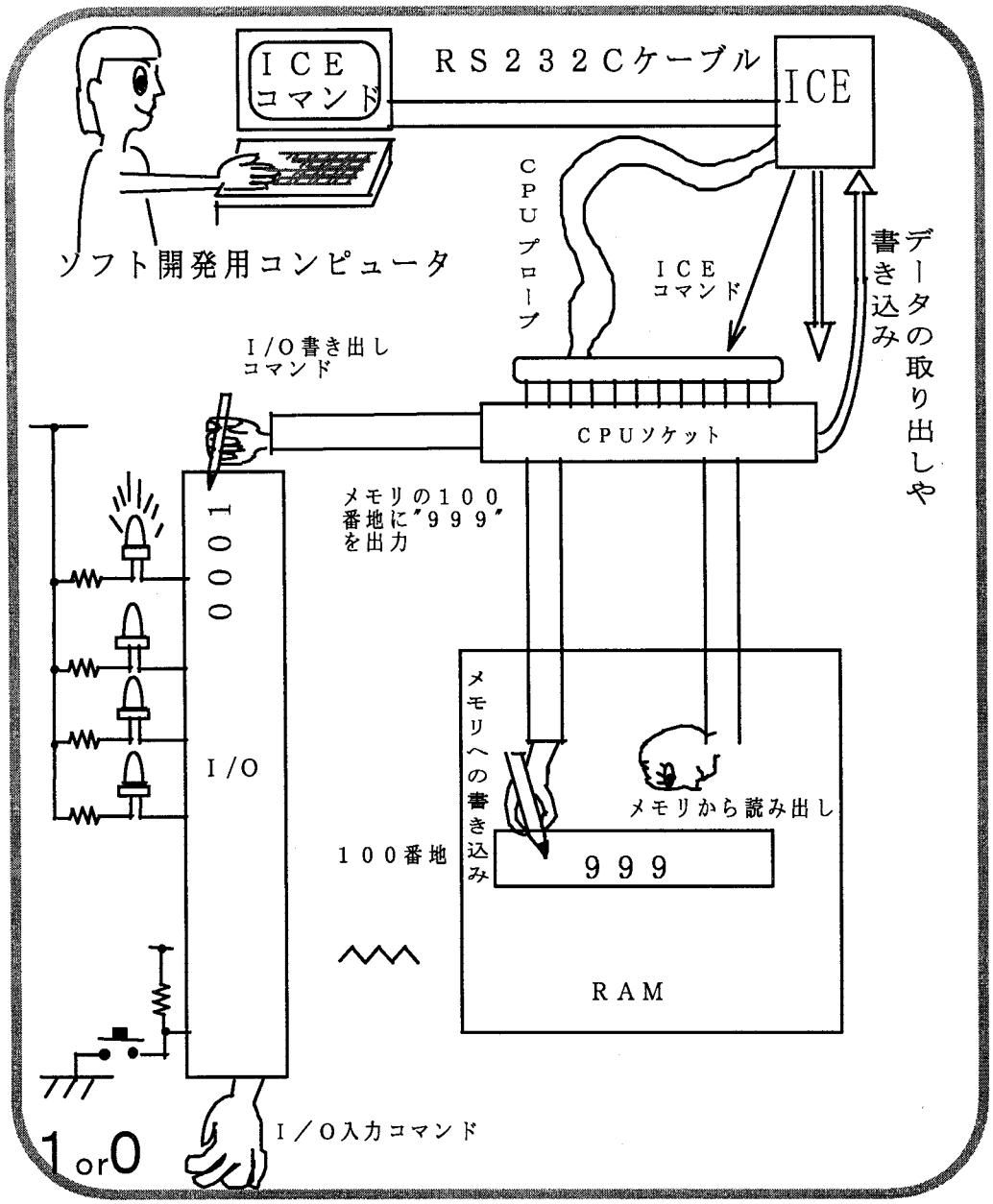
図Ⅵ－１８　ＩＣＥ

(2) ＩＣＥの用途

センサやアクチュエータ等のハードウェア部分を含むシステムでは、物理的なタイミング部分をデバッグするために組合せテストが必要になる。そのときＩＣＥを使うと、ソフトウェアの微妙な調整ができ、作業効率がグンとよくなる。また、ＯＳを搭載しないシステムやキーボード、ディスクがないターゲットシステムでのソフトウェア開発作業を容易にする。



図Ⅵ－１９　ＩＣＥの用途（ａ）



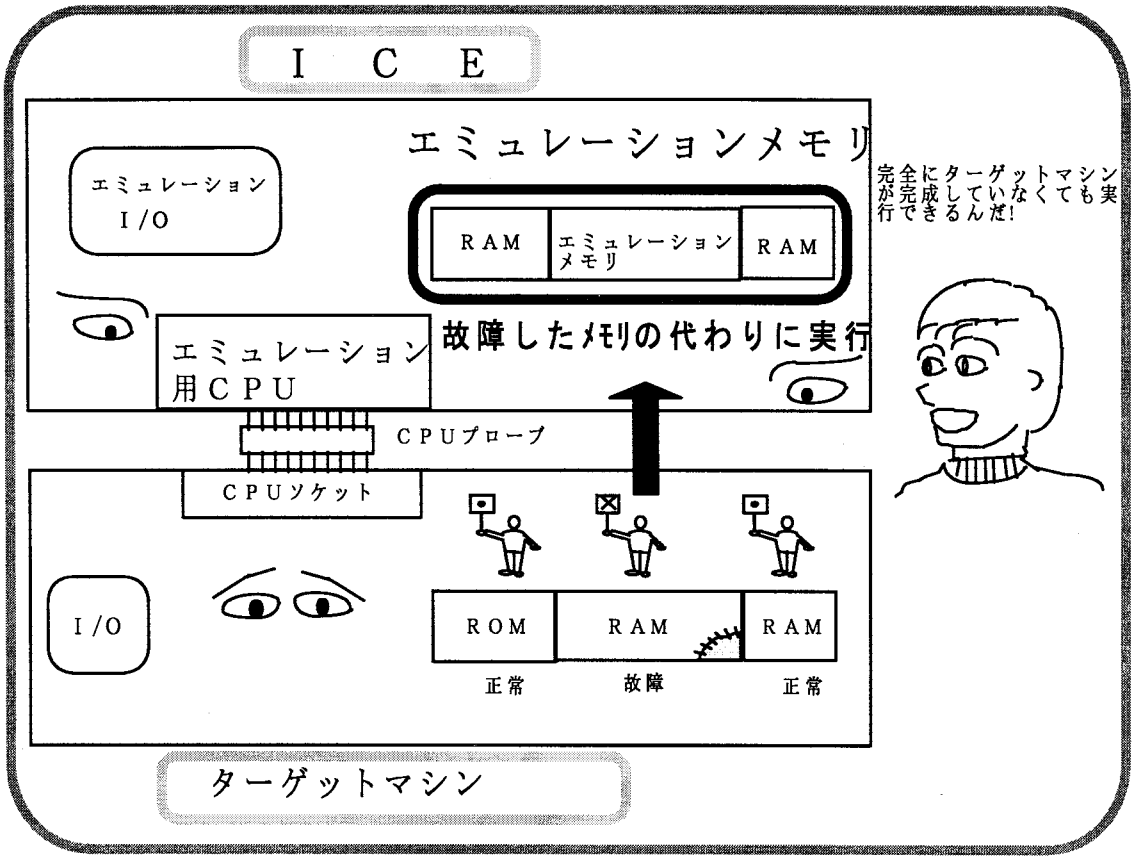
図Ⅵ－２０　ＩＣＥの用途（ｂ）

○ ＩＣＥコマンドでメモリへデータを書き込み、そのデータを読み出して確認することでメモリの動作テストができる。

○ 同様にＩ/Ｏポートにデータを出力し、ＬＥＤやロジックアナライザで確認することが出来る。

○ Ｉ/Ｏの入力ピンにスイッチをつけ入力の確認を行うこともできる。

(3) ＩＣＥの機能



図Ⅵ－２１　ＩＣＥの用途（ｃ）

ＩＣＥにはターゲットシステムのＣＰＵの代わりに実行するエミュレーションＣＰＵがある。ターゲットシステムのＣＰＵソケットにプローブをつないで、ターゲットシステムのＣＰＵの代わりに実行することにより総合的なデバッグを実現するものである。

また、ＩＣＥにはエミュレーション用のメモリやＩ/Ｏがあり、ターゲットシステムのメモリの代わりにＩＣＥのエミュレーションメモリでプログラムを実行させることができる。このとき部分的に故障したメモリの部分にだけエミュレーションメモリを配置(マッピング)することができる。

また、ＲＯＭなどデータを書き込みできないエリアへの書込みがあった場合には、割込みを発生させて停止させ、間違ったソフトウェアのデバッグを行うことができる。

4 シミュレータの使い方

主にソフトウェアのモジュール単位のテストやソフトウェアの論理テストに威力を発揮するツールである。ＣＰＵの異なるターゲットマシンの環境をソフトウェア的に作り出す。これにより高機能のプログラム開発マシンで、テスト→プログラム修正の繰返しサイクルが効率のよくできる作業環境が実現できる。

(1) シミュレータの用途

図Ⅵ－２２　シミュレータ

① モジュールテスト

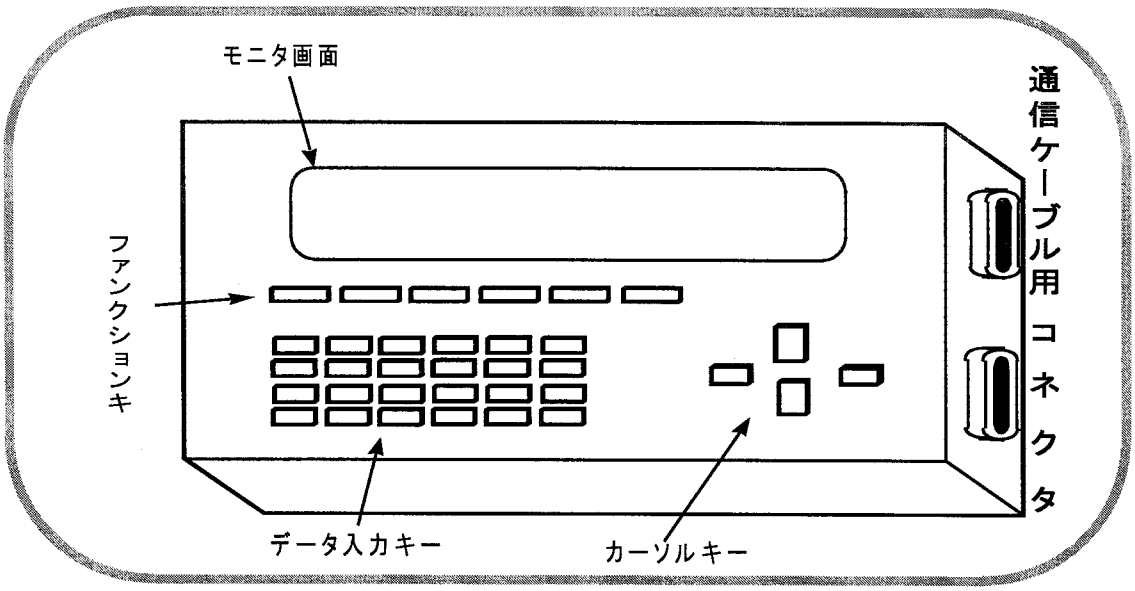
初期のテストの段階では関連するソフトウェア、ハードウェア環境から影響を受けない状態でテストする方が効率のよいテストができる。この段階では、高機能のプログラム開発マシン上でシミュレータを使いプログラムを作成しながら、モジュールテストを行う方が効率がよい。

② ソフトウェアの論理テスト

モジュールをつないだ総合テストの段階で、入出力を仮想的なエミュレーションＩ/Ｏに割り当てることにより、ハードウェアから切り離した状態で論理的なテストを行うことができる。また、高機能のシュミレータでは入出力のタイミングを設定することにより、ハードウェアの仕様の検討を行うこともできる。

5 プロトコルアナライザの使い方

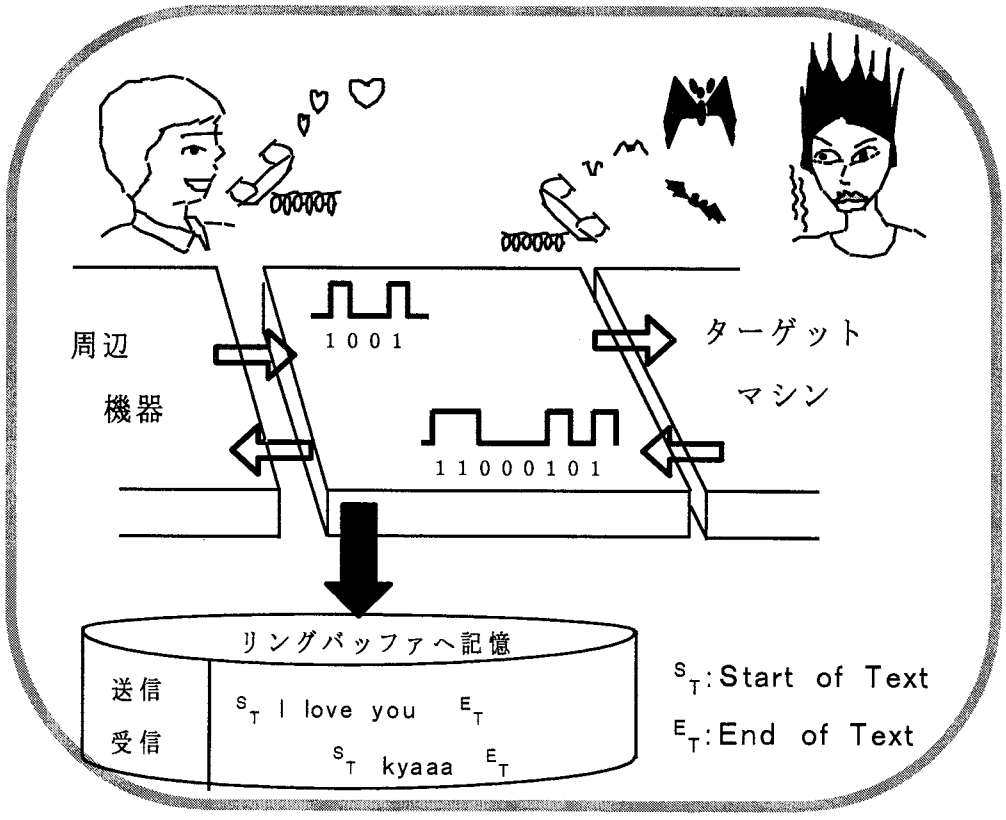
(1) プロトコルアナライザの外観



図VI－２３　プロトコルアナライザの外観

(2) プロトコルアナライザの用途

ターゲットマシンと周辺機器との間の通信のモニタやデバックに使われる。



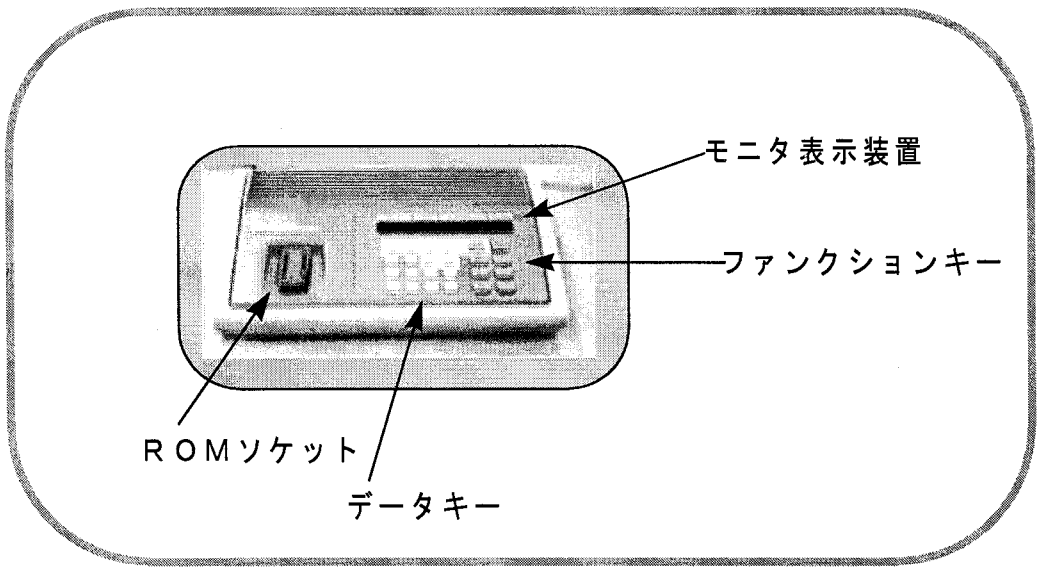
図VI－２４　プロトコルアナライザの用途

(3) プロトコルアナライザの機能

ロジックアナライザと同様にトリガ機能とメモリ機能を持っていて主にシリアル通信の内容の確認に用いられる。また、通信のエミュレーション機能を持っていてターゲットシステムの周辺機器の代わりに模擬の通信を行い通信機能のテストを行うことができる。

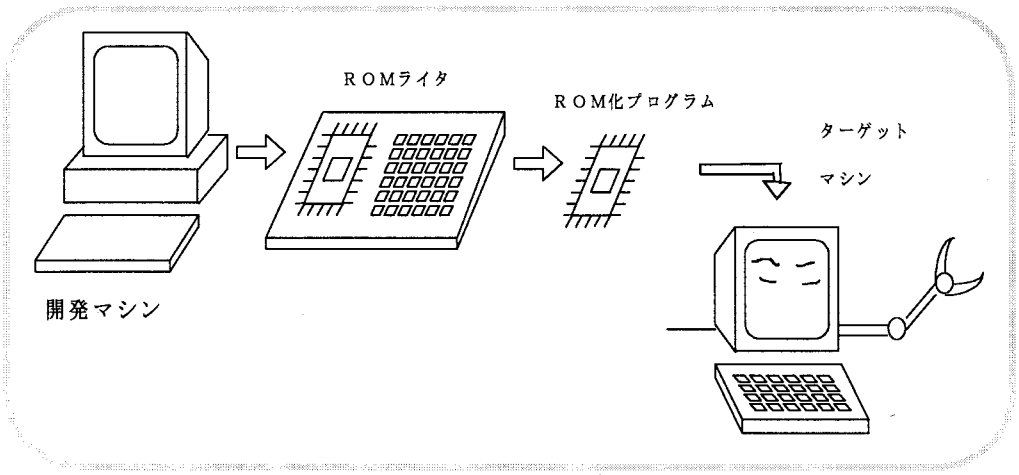
6 ＲＯＭライタの使い方

(1) ＲＯＭライタの外観



図Ⅵ－２５　ＲＯＭライタの外観

(2) ＲＯＭライタの用途



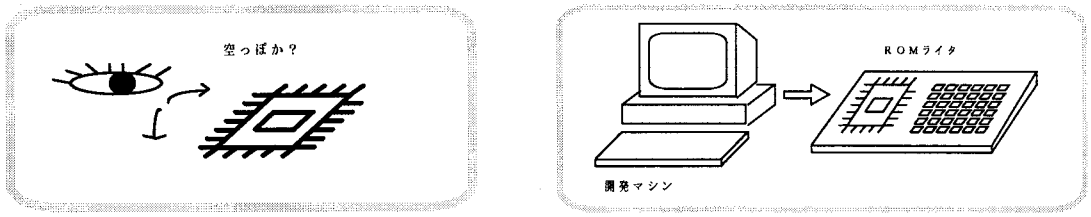
図Ⅵ－２６　ＲＯＭライタの用途

ＲＯＭライタは、開発されたプログラムをＲＯＭに書き込んで、ターゲットマシンに組み込むための装置である。

(3) ＲＯＭライタの機能

① ブランクチェック
これから書き込むＲＯＭに少しでも何かが書き込んであると正しく書き込むことが出来ない。そこで書き込む前に完全に空っぽであることをチェックする。

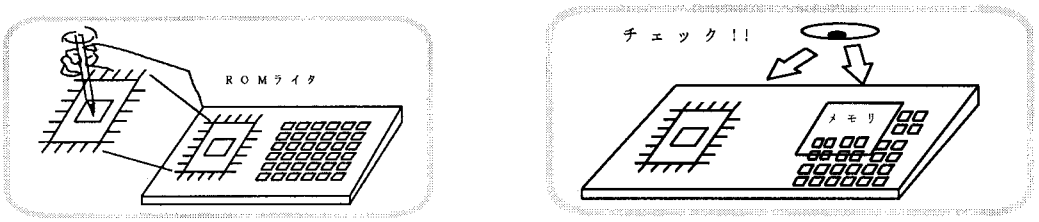
② ＲＯＭライタへダウンロード
開発したプログラムをＲＯＭライタのメモリに転送する機能



図Ⅵ－２７　ＲＯＭライタの機能

③ ＲＯＭへの書込み
ＲＯＭライタのメモリに入っている内容をＲＯＭに書き込む。

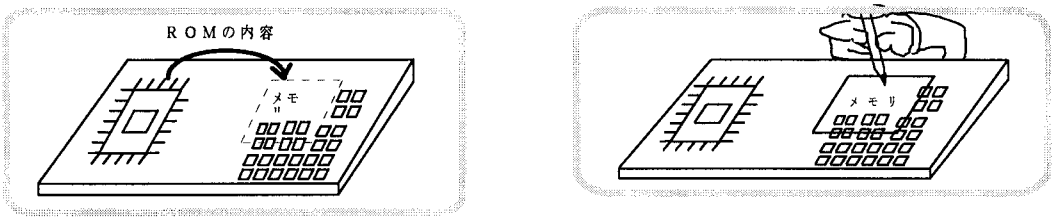
④ ベリファイ機能
ＲＯＭの内容とメモリの内容を照合して、書き込まれた内容と同じかどうかをチェックする。



図Ⅵ－２８　ＲＯＭライタの機能

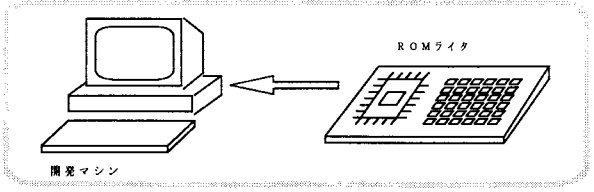
⑤ ＲＯＭの読込み
ＲＯＭの内容をＲＯＭライタのメモリへ写し取ることができる。

⑥ 内容の変更
ＲＯＭライタのメモリ上の内容の変更とその内容を表示することができる。



図Ⅵ－２９　ＲＯＭライタの機能

⑦ ＲＯＭライタからのアップロード
ＲＯＭライタのメモリの内容を開発マシンに転送することができる。

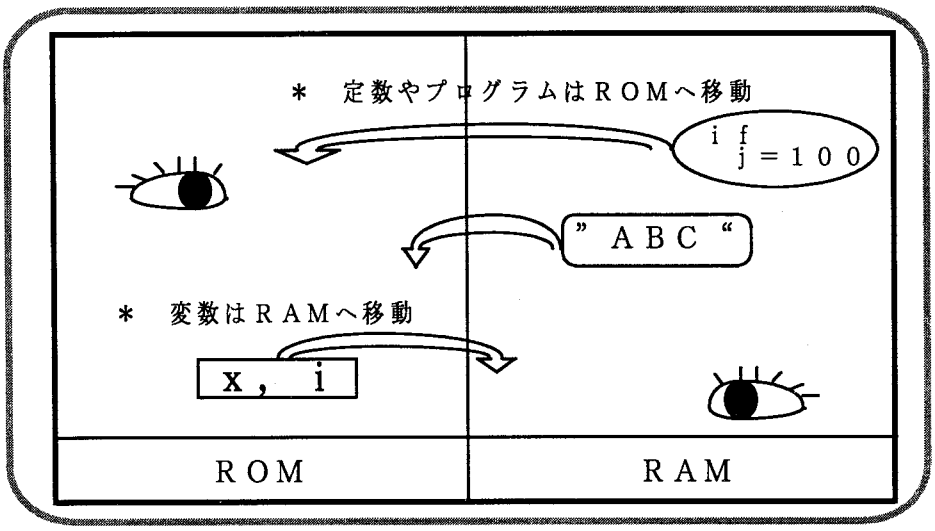


図Ⅵ－３０　ＲＯＭライタの機能

(4) ＲＯＭ化の準備

① 書き込む内容の準備

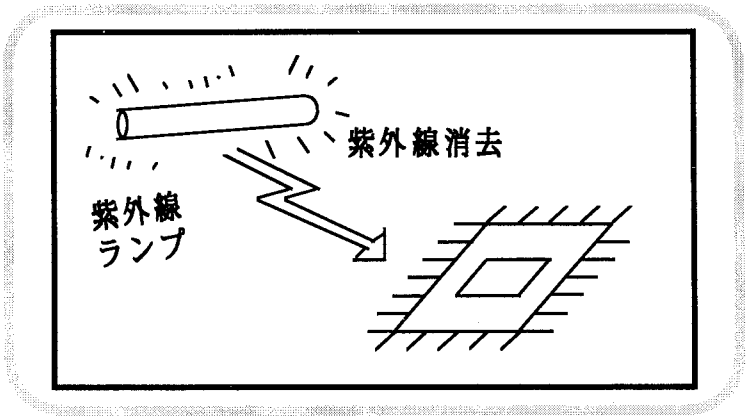
ＲＯＭ化するプログラムは、変数はＲＡＭのエリアに、定数やプログラムコードはＲＯＭのエリアに割り当てられるようにアドレスを割付けなければならない。アセンブラで開発したプログラムはプログラマが意識してアドレスを付けられるが、コンパイラを使った場合には、パラメータでプログラムコードエリア(ＣＯＤＥ＝＊＊＊＊)と、データエリア(ＤＡＴＡ＝＊＊＊＊)を指定する必要がある。また、ＲＯＭライタにロードできるような形式(ＨＥＸ形式のファイルが多い。)にしておかなければならない。



図Ⅵ－３１ ＲＯＭ化の準備

② ＲＯＭの準備

書き込む前のＲＯＭの内容は、空になっていなければならない。ＥＰＲＯＭの場合は、紫外線イレーサでＲＯＭの内容を消去しておく必要がある。また、書き込んだ後にＲＯＭの窓にプロテクトシールを貼る必要がある。



図Ⅵ－３２ ＲＯＭの消去