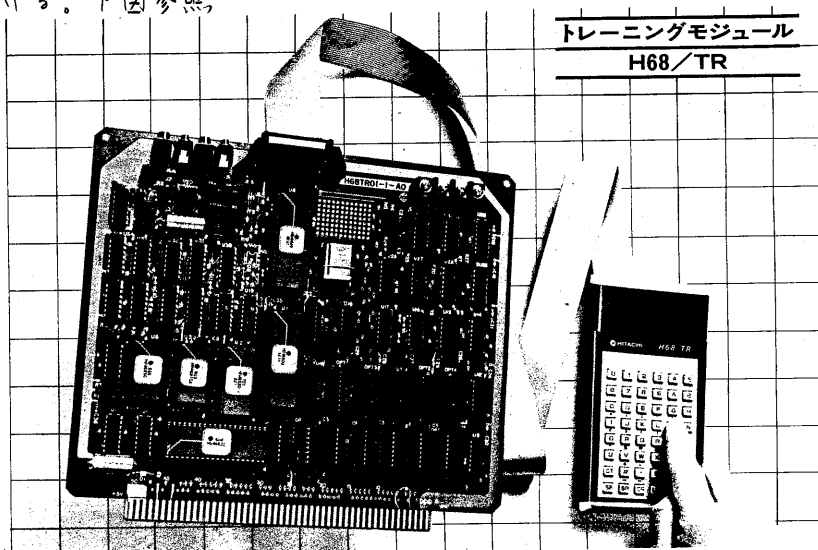


# トレーニングモジュールの機能設定

西野秀毅, 吉村一馬 (日立システム開発研究所)  
 小野寺徹, 渡雅男 (日立武蔵)

## 1 概要

米国のMITS社は1975年1月にAltair 8800マイクロコンピュータキットを発売した。これをきっかけにマイクロコンピュータキットが各社から発売された。これらマイクロコンピュータキットの用途は主に、マイクロコンピュータハード及びソフトの学習、マイクロコンピュータ応用システム開発、アマチュア向けキット、等である。日立トレーニング・モジュール(H68/TR)もこれら用途を目的とした製品である。H68/TRは基本ボードとポケッタアルコンソールから構成されている。下図参照。



ポケッタアルコンソールにおける特徴は、①開閉電車ケースの採用、②フルキーボード(英数字入力可)の採用、③14桁7セグメントけい光表示管による変形英数字記号等の表示機構の採用、等である。ポケッタアルコンソールにおいては必要な機能の多くをソフト処理する事により、部品点数をできるだけ少なくしている。

基本ボードにおける特徴は、①4Kビット・スタティックRAM, 32Kビット・マスクROMの採用によってメモリ容量の大きな増大(最大RAM3Kバイト, マスクROM4Kバイト)を計った事、②オーディオカセットレコーダを補助記憶装置として使用できるためのインタフェース回路を標準装備している事、(この回路中には基本的READ/WRITE機能の外に、2台までの自動スタート、ストップ制御機能も含まれる。)③基本ボードからの拡張を容易にするためのいくつかの機構を備えている事、たとえば i) アドレスフルデコード方式, ii) 100ピン共通バスコネクタ, iii) 日立シングルボード・コンピュータとコンパクトなボードサイズ, コネクタ仕様, iv) 4枚ボード挿入可能なカード・ケース、等である。

以上のハード機能によって、①ラベル処理ができるアセンブラ、②アセンブラ、ソースプログラムの修正、保存のためのテキスト・エディタ、③フレイク・

ポイント及びステップアップの機能を持ったモニタ, 等のソフトサポートが可能になった。

## 2. H68/TR アセンブラ

### 2.1 アセンブラ・ソース・プログラム

- i) 全OPコードのニモニク記述
- ii) ラベル記述ができる。
- iii) オペランドにおける定数は16進及び10進形式のみに限る。
- iv) アセンブラ命令として次の5命令がある。ORG, RMB (Reserve Memory Byte), EQU, FCB (Form Constant Byte), END。
- v) ラベル形式はL\*\*である。ここで\*\*は01~FFの16進形式の数値である。

```

ORG 40
RMB 2
ORG $200
LDAA 40
ADDA 41
BPL L01
STAA 41
BRA L02
L01 STAA 40
L02 SWI
END

```

アセンブラコーディング例

### 2.2 アセンブル方式の特徴

我々は1パスアセンブル方式を採用した。1回をアセンブルできるステップ数は2パスアセンブル方式より少なくなるが, この事は大きなデメリットではない。なぜなら, 我々のシンボルテーブルにおいては1ラベル当り2バイトしか必要でない。アセンブラがマスクROM化されている事1パスアセンブル方式によって, アセンブル時の操作は極めて単純なものになっている。

### 2.3 アセンブラ・ソース・プログラムの入力方式

ここではシステム設計にとって重要な所をピックアップして述べる。

#### 2.3.1 キーボード・スイッチにおけるキートップの設計

アセンブラ・ソース・プログラムを入力するためにはどうしても英数字キーが必要である。そこで我々はTTY式のフルキーボードよりはるかに安価な関数電卓用の48キー・スイッチを改良することにした。ここで問題になるのはキートップの設計である。この設計に関しては大きく分けて次の2通りの方式がある。

##### i) アルファニューメリック方式

##### ii) 命令キー方式

これらの2通りの方式によるキートップを図1, 図2に示す。アセンブラ・ソースプログラムを表示したりケースにおいては, 命令キー方式のアセンブラはアルファニューメリック方式のアセンブラより約300バイト小さくて済む。これはニューモニクコードのテーブル(231バイト)とテーブルサーチのプログラム(約70バイト)が不要になるからである。しかしH68/TRのようにアセンブラ・ソースが表示できるシステムでは, 命令キー方式の場合, 命令コードをニモニク表示するためのテーブル(約200バイト)が必要になる。従って

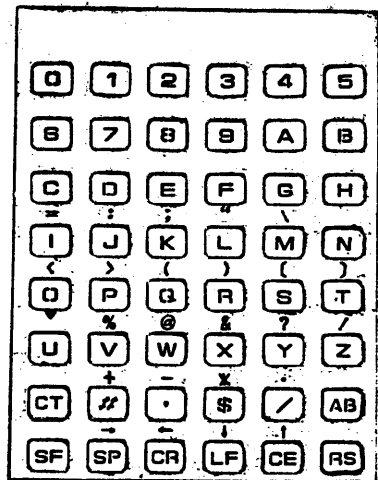


図1 アルファニューメリック方式

この場合アルファニューメリック方式のアセンブラは、命令キー方式のアセンブラよりも約100バイト大きくなる。他方アルファニューメリック方式の方は次のような利点がある。

- 1段シフトで済むためキー操作が単純
- 標準記号がサポートできる。
- TTYインタフェース回路を作れば、TTYインタフェースを持つコンソールをH68/TR用コンソールとして使用できる。

以上の主要な理由により、アルファニューメリック方式が採用された。

### 2.4 キーボードスイッチ読み取り機構

この機構の特徴は、この回路の一部がけり光表示管表示回路の一部を構成していることである。けり光表示管の表示方式はダイナミック方式であるが、この表示桁選択信号が同時にキーボードのスクアン信号になっている。図3における

HD74154 (4-line-to-16-line Decoder / Demultiplexer) がこのスクアン信号を出す。図においては表示桁選択信号線は省略されている。この機構で重要な役割りを果たしているタイマー割込み及びHD74148について簡単に説明する。MPUには1.2kHzの信号がIRQ端子(マスク可能割込み端子)にインポートされている。

ABA/ADC	ADD/AND	ASL/ASR	BCC/BCS	BEO/BGE	BGT/BHT
0	1	2	3	4	5
BIF/BIE	BLS/BLT	BMI/BNE	BPL/BRA	BSR/BVC	BVS/CBA
6	7	8	9	A	E
CLC/CLT	CLR/CLV	CMP/COM	CPX/DAA	DEC/DES	DEX/EOR
C	D	E	F	G	H
INC/INS	INY/JMP	JSR/LDA	LDS/LDX	LSR/NEG	NOP/ORA
I	J	K	L	M	N
PCH/PUL	ROL/ROR	RTY/RTS	SBA/SBC	SEC/SEI	SEY/STA
O	P	Q	R	S	T
STS/STX	SUB/SWT	TAB/TAP	TBY/TBA	TST/TSX	TXS/WAI
U	V	W	X	Y	Z
	AES/ORG	EQV/FCB	END/\$	>/<	
SFT 1	+	-	*	/	ABT
SFT 2	=	#	SP	CR	RES

図2 命令キー方式

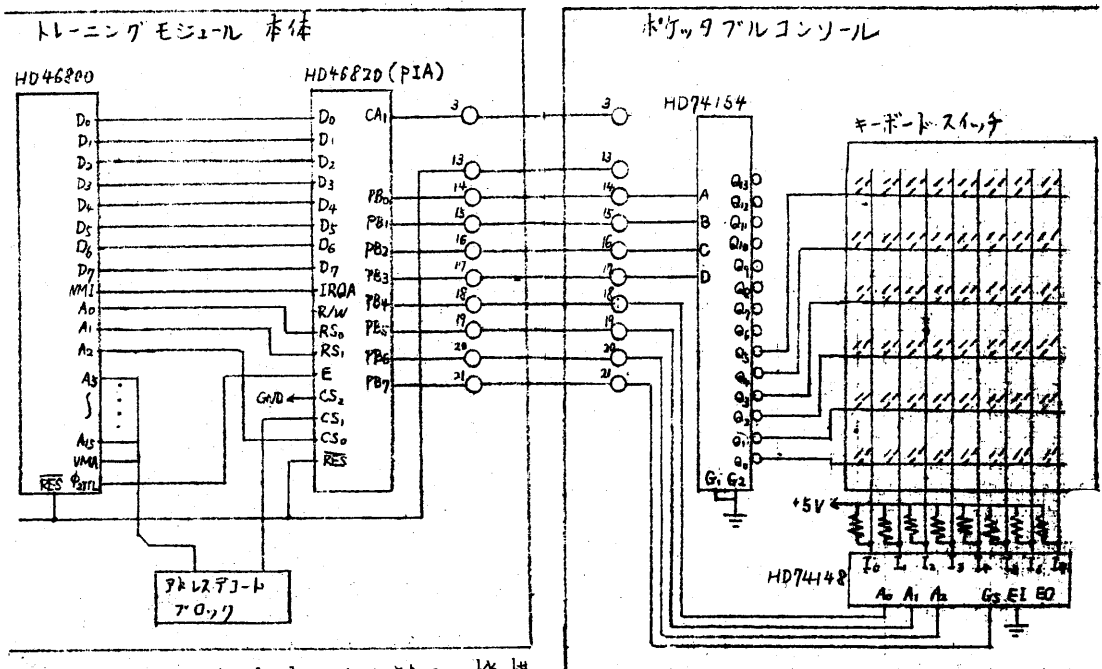


図3 キーボードスイッチ読み取り機構

これはユーザがハードウェアタイマとして利用する以外にも、モニタかけり光表示管表示におけるダイナミック制御にも利用している。タイマ期込サ1回毎にモニタはHD74154の(D, C, B, A)に信号を送る。(D, C, B, A)は(0, 0, 0, 0)から(1, 1, 1, 0)までを周期的に変動する。既に述べたように(D, C, B, A)をデコードした情報(Q<sub>0</sub>, Q<sub>1</sub>, Q<sub>2</sub>, ..., Q<sub>13</sub>)の中でQ<sub>0</sub>~Q<sub>5</sub>がキーボードのスキャン情報として利用されている。

HD74148はエンコーダとしての機能以外に、I<sub>0</sub>~I<sub>7</sub>のいずれかが1つが"0"であれば、G<sub>S</sub>が"0"になる。もしQ<sub>4</sub>とI<sub>5</sub>の交点の所のキーが押されているとすれば、Q<sub>4</sub>が"0"になるとI<sub>5</sub>が"0"になり、この事によって(A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub>)が(0, 1, 0)になり、G<sub>S</sub>は"0"になる。

以下キーボード読取り手順について述べる。(PIA操作のための必要な手順は省略する。

- ① PIAのPB7を読む。
- ② PB7が1であれば①に分歧, 0であれば③に分歧する。
- ③ 14m秒待つ。(キースイッチのバウンス時間回避)
- ④ PIAのPB7を読む。
- ⑤ PB7が1であれば④に分歧, 0であれば⑥に分歧する。
- ⑥ PB4, PB5, PB6の値と、スキャン信号を与えている(D, C, B, A)の値からテーブルを調べ、該当キーコードを見つける。

## 2.5 アセンブラ・ソース・プログラムの表示方式

アセンブラ・ソース・プログラムの印字又は表示方式をどのようにするかは今回の1つの検討テーマであった。印字方式に関しては、低価格の5×7ドット方式の放電又はサーマルプリンタを利用する方式がある。しかしこれでもベーシックシステムに標準装備するにはやはり価格的に無理があり、拡張機能として考えるのが自然である。我々は以下に述べるけり光表示管によるアセンブラ・ソース・プログラム表示方式を採用した。

### 2.5.1 表示デバイスの選択

我々はけり光表示管を選択したが、これは他の表示デバイス(発光ダイオード表示デバイス、液晶表示デバイス)より低価格であるからである。消費電力は5mW/digと他の表示デバイスより極めて低いが、H<sub>08</sub>/TRの全体の消費電力(約6W)を考慮すれば何ともし無視できる。

次に表示桁数をいかに少なくするか1つの問題点である。これはアセンブラにおけるスタートメント仕様、及びテキストエディタのコマンド仕様と関連する。さらにできればけり光表示管の標準品を使い、電卓ケースに収まらなければならない。我々の電卓ケースに収まりうる標準品として最長の14桁のものを採用した。これによりアセンブラのチャートのスタートメントは表示可能(あれば好ましい空白(ラベル, オブコード等)などはなし)になったが、テキストエディタのChangeコマンドでは無理なフォーマットになってしまった。

### 2.5.2 表示デバイスによる文字表示方式

開発当時発売されていたけり光表示管は7セグメント(小数点用ドットセグメントを含めれば8セグメント)であった。7セグメント方式は本来数値表示を意図していたためアルファニューメリック表示には適してはいない。セグメント数を増加させればセグメント、16セグメントになれば、一応アルファニューメリック

記号は表示できる。しかしす下に述べたように電卓ケースに収まる14桁にすれば、セグメント間隔が狭きすぎ技術的に困難であり、さらに標準品でなくなるため価格的にも採用は困難になる。

ここで以下に示すいずれかの方針を選択しなければならぬ。

- ① アセンブラのサポートを見送る。
- ② アセンブラのサポートは行なうが、アセンブラ・ソースの表示は行なわぬ。
- ③ 7セグメントでアルファベット表示を行う。

我々は結果として③を選んだ。図4に7セグメントによる記号表示法を示す。

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I	J
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z	#	,	\$	.
U	V	W	X	Y	Z	#	,	\$	.
=	+	-	/	*	(	)	!	%	&
=	+	-	/	*	(	)	!	%	&
<	>	↑	←	\	[	]	_	.	
<	>	↑	←	\	[	]	_	.	

図4 7セグメントによるアルファベット表示

これらの表示は初め奇異に見えるがすぐに慣れることが判明した。我々の簡単な実験では、アセンブラ・ソース・ステートメントで必要な記号程度ならば30分程度の学習をほとんどの人が読めることが示された。表記方法の原則を示す。

- ① できるだけ字形が似ていること。
- ② アルファベットはできるだけ大文字表現にする。
- ③ 誤読される危険のある形はできるだけ排除する。

## 2.6 アセンブラ・ソース・プログラムの保存・修正方式

アセンブラを使用してデバッグする事を想定すれば、アセンブラ・ソース・プログラム・レベルでのプログラム保存・修正機能は重要な意味を持つ。通常レジスタシステムにおいては、紙テープ・マスでこの機能がサポートされている。しかしこれはトレーニングモジュールの価格を考慮すれば好ましくなり。いくつかのトレーニングキットにおいて家庭で使用されているオーディオカセットレコーダが“補助記憶装置”として使用されている。しかもこのためのインタフェース回路は複雑でない。オーディオカセットを補助記憶装置として使用する上で信頼性が十分あることが判明した。

以下においてテキストエディタの言語仕様、及びその処理方式に関する検討は省略する。

### 2.6.1 レコーディング・フォーマット

オーディオカセットテープにおけるレコーディング・フォーマットは1975年11月にミズリー州カンサスシティで開催された「BYTE」マガジンの後援のシンポジウムで公表された「カンサスシティ規格」に従っている。これは以下のようなものである。

- (1) マーク（論理“1”）は2.4kHz信号8周期。
- (2) スペース（論理“0”）は1.2kHz信号4周期。
- (3) 1文字の構成はスタートビットとして1個のスペース、8個のデータビット、ストップビットとして2個以上のマークからなっている。
- (4) 文字と文字の間は、不特定時間のマーク信号が入る。
- (5) データはテープの先頭から30秒後に録音する。
- (6) 文字のLSB (least significant bit) が最初に転送され、MSB (most significant bit) が最後に転送される。
- (7) データはいくつかのブロックから構成される。ブロック長は任意でよい。ブロックの先頭には少くとも5秒間のマークが必要である。

### 2.6.2 オーディオ・カセット・インタフェース

オーディオ・カセット・インタフェース設計上における検討項目として、オーディオ・カセットからのアナログ信号をデジタル化した後に1.2kHz、2.4kHzの矩形波を識別するということがある。矩形波のばうつきは約20%程度までを許容したい。テープをローカルに見れば、そこでの1.2kHzと2.4kHzの波長の比は2:1である。なぜなら電圧変動、メカニカルな原因によって短時間にテープ物動速度の急激な変動があるとは考えられない。従って波長の絶対的長さを想定せず、2:1の比のみを用いる単純な復調方式があれば最も好ましい。現在この方式による突発的回路は提案されていない。他の方式による簡単なものにはワンショットを利用した次のようなものがある。1.2kHz又は2.4kHzパルスの立上りから一定時間(1/2400 × 3/4秒)おきのパルスのレベルによって1.2kHzか2.4kHzを判断する。あるワンショットでは

$$t = 0.28 R_T C_{exp} (1 + 0.7 / R_T)$$

の式で与えられるものをこの一定時間にセットすればよい。ここで $R_T$ は外部タイミング抵抗、 $C_{exp}$ は外部タイミング容量である。このようにワンショットは抵抗、容量にセンシティブなためあまり好ましくない形式といえる。基本的方針として我々は一定時間の測り方はカウンタを利用することにした。次の回は我々が採用したオーディオカセットインタフェース回路である。

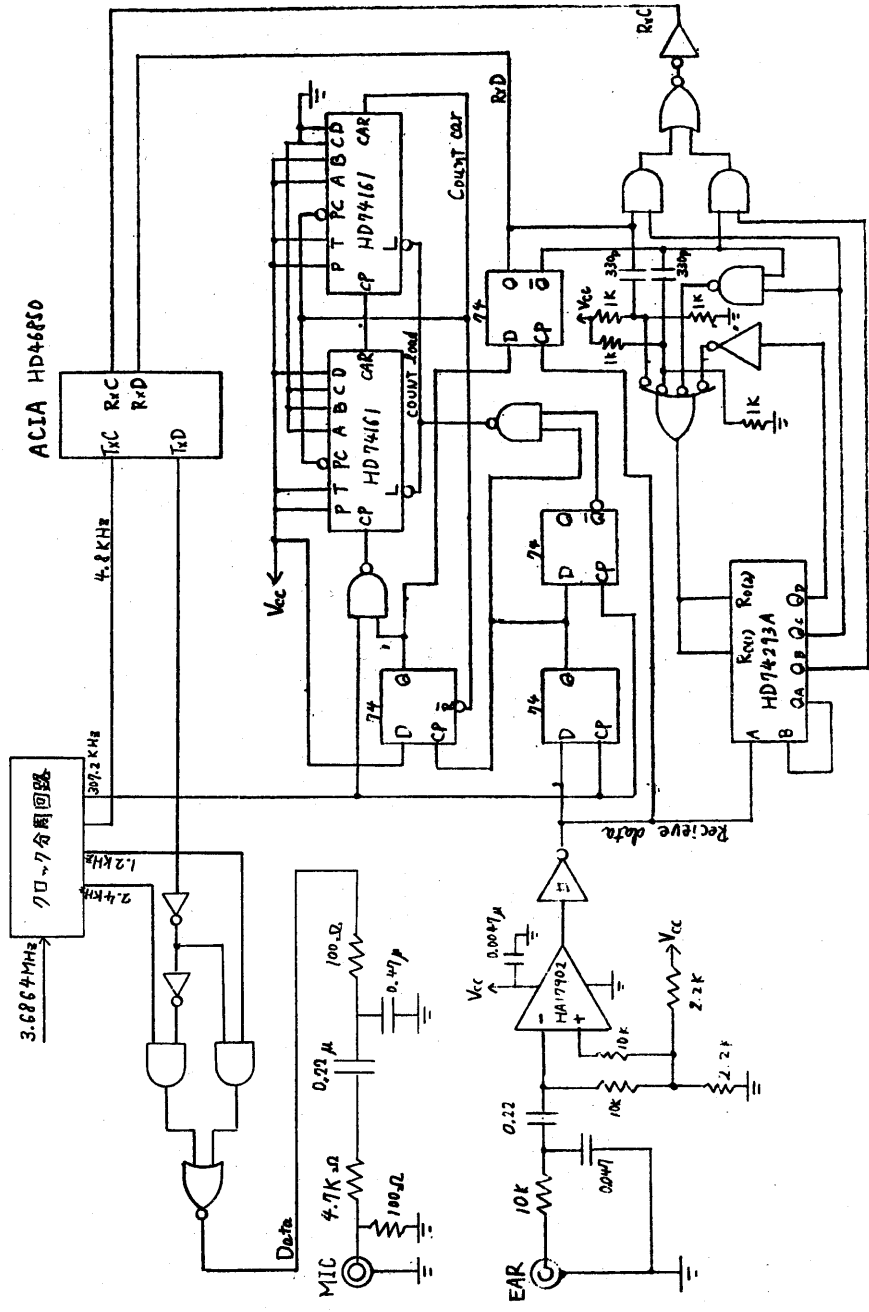


図5 オーダイオ データレイトイングフェイス回路

### 3 システム割込み機構

次のモニタ機能がMPUの割込み機能を用いて実現されている。

- (1) システムリセット (リセット割込み)
- (2) デバッグ用ステップラン (Non Maskable 割込み)
  - 1命令実行毎にユーザにコントロールが渡り、レジスタ類の参照ができる。対象プログラムの所在 (RAM, ROM, PROM) は問われない。
- (3) アボート機能 (Non Maskable 割込み)
  - 命令実行を中断してコントロールをユーザに渡し、レジスタ類の参照ができる。
- (4) ハードウェアタイマ (Maskable 割込み)
  - 1.2 KHzの頻度で割込みが発生する。これをMPUで計測する事によりタイマ割込み機構が実現できる。システムではこのタイミングでだけ光表示管に表示データを転送し、必要なdutyを確保している。又このタイミングを利用することにより割込み機構を持たないキーをあらかも割込み機構が付加されているかのようにあつかうことができる。つまり一般のプログラムの実行中にユーザのキーインを受付けることができる。
- (5) ブレイクポイント設定 (ソフトウェア割込み)
  - RAMに格納されているプログラムの任意の所でプログラムの実行を中断することができる。これはユーザの指定した所にソフトウェア命令を埋込むことにより実現される。

#### 3.1 デバッグ用ステップラン機構

基本的考え方は次のようなものである。HD46800 MPUにおいては割込み処理プログラム (多くはモニタ) から割込まれたプログラムに復帰する場合 RTI 命令によって行なわれる。RTI命令の実行サイクルは10である。従ってRTI命令実行直前に、10サイクル経過後に割込みを引起す機構を起動すれば、RTI命令実行直後に実行される命令の所で割込みがかかる。ここでの割込みは Non Maskable でないといけない。こうすることにより割込みをマスクしているプログラムをもステップランすることができる。

#### 図6 参照

#### 3.2 アボート機構

この機構もステップランと同様な意味で Non Maskable 割込みを利用する。Non Maskable はユーザにも解放されているため、割込み要因の判定ができなければならない。このため我々のシステムでは割込み要因設定レジスタとしてPIAのステータス・レジスタが使われている。ユーザのキーイン信号 (アボート信号) はまずPIAの割込み受付端子に入り、ここでPIAのステータス・レジスタに割込みフラグがセットされる。次にPIAからMPUに割込みがかかる。

#### 3.3 ハードウェアタイマ

割込み要因設定はアボート機構と同様にPIAのステータスレジスタを利用する。この割込みは Maskable である。

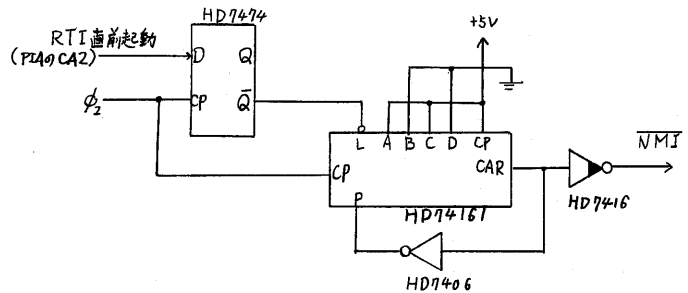
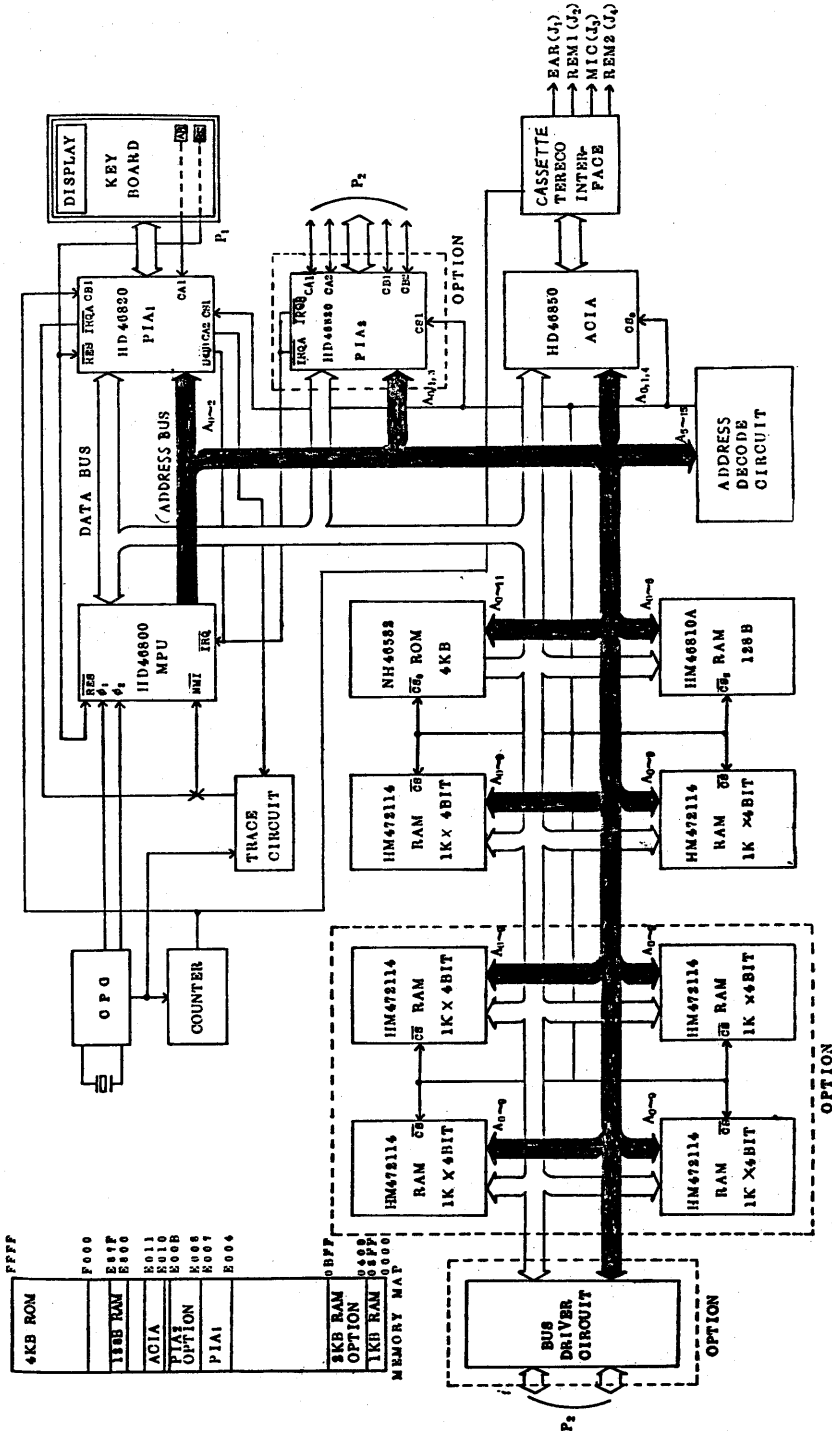


図6 ステップラン機構



#### 4 H68/TRシステム構成図とメモリマップ

メモリマップで示される4KB ROMにはモニタ、アセンブラが格納されている。  
 E\*\*\*は現在モニタワーク・エリア及びPIA, ACIA というI/Oインタフェース・エリアになっている。ユーザエリアは0000～から使用される。  
 以下にシステム構成図を示す。



H 68 / TR システム構成



## 5 おわりに

今回H68/TRのアセンブラサポートに関するシステム機能面、特にハード機能面を中心に述べた。この他に白ボックスとしてソフトウェア機能(モニタ、アセンブラ、テキスト・エディタ)、ハード・ソフト両面に亘る拡張サポート機能、システムの信頼性(特にオーディオ・カセット・レコーダを補助記憶装置として使う上での信頼性)等に関する検討が残されている。これらについては今後機会があれば述べたいと思う。最後に本開発期間中御指導願った日立武蔵工場の大沢晃副技師長、豊沢弘毅主任技師に深く感謝致します。