

# パーソナルコンピュータ PET 2001-8 の拡張

——メモリ増設と P-ROM (2708用) ライタのインターフェース——

高浪 五男\*・八田 信\*\*・井上 克司\*・原戸 博幸\*

## Expanding the Personal Computer PET 2001-8

——Increase of Memory and Interface of P-ROM Writer for 2708——

Itsuo TAKANAMI, Makoto HATTA, Katsushi INOUE and Hiroyuki HARATO

### Abstract

PET 2001 is a high performance and inexpensive popular personal computer with high speed BASIC ROM. The PET has several bus connectors being open to the users. Using one of these connectors, we increase memory of PET2001-8 by static RAM 2114 and dynamic RAM 4116. Furthermore, we interface P-ROM writer for 2708 with PET.

### 1. はじめに

PET 2001は高速 BASIC 言語を ROM に内蔵した高性能、低価格のパーソナルコンピュータで、国内で最も人気のある機種の一つである。この PET はユーザーに開放された種々のバスコネクタを持っており、極めて拡張性に富んでいる。この特長を利用して種々の拡張が考えられる。この報告では 8KRAM を内蔵している PET 2001-8 のスタチック RAM 及びダイナミック RAM によるメモリーの拡張法と P-ROM ライターのインターフェースについて述べる。

### 2. 拡張ボード

PET の種々の拡張を効率的に行なうために、まずマザボード型式の拡張ボードを作成し、このボードのコネクタを通して種々の拡張を行なう。

PET 2001-8 のバス・コネクタは向って右下にある。コネクタの片面は 40 ピンで、両面合わせて 80 ピンである。ピンのピッチは 0.1 インチ (2.54mm) で、上面のすべてのピンはグラウンドになっている。下面のピンは手前から A1, A2, …, A40 のような番号になっている。

図 1 は拡張ボードのバッファ回路である。PET のバス・コネクタのピン番号に続くカッコ内の記号は対応する信号名を意味する。バス・コネクタからの信号

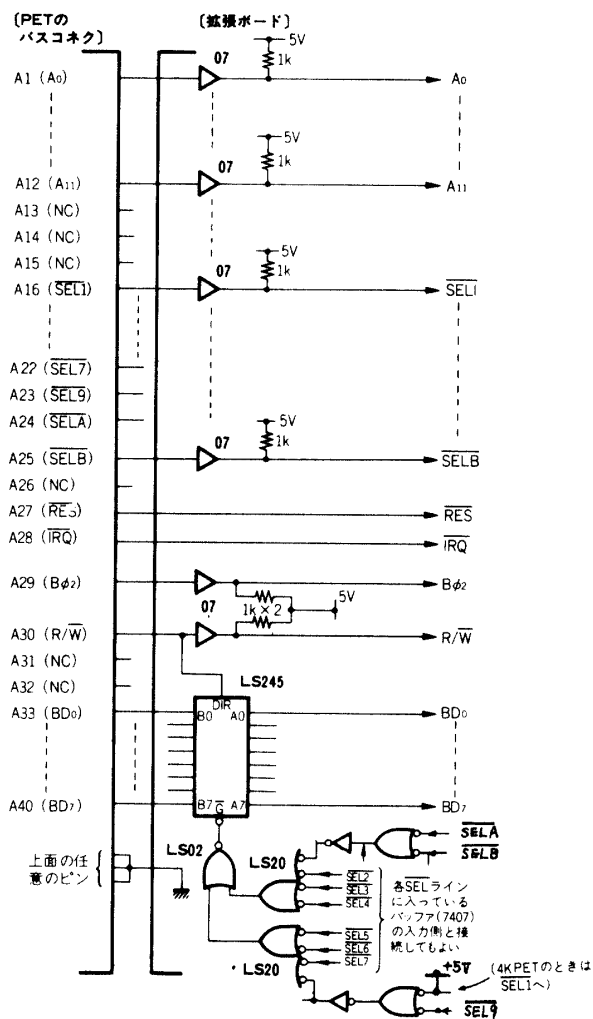


Fig.1 Buffer circuit in the expansion board

\* 電子工学科

\*\* 大学院電気工学専攻

は、バッファを介して拡張ボードと接続される。拡張ボードの電源は専用の電源を設ける。この電源には、+5V 以外に必要な電圧、たとえば ±12V、あるいは ±26V、-5V 等の端子を設けておくことと便利である。

拡張ボードと PET との接続はできる限り短い距離で行なう。図1の回路は簡単であるので配線さえ間違えなければ確実に動作するはずである。

### 3. メモリーの増設

#### 3.1 スタチック型 RAM 2114 による方法

まずスタチック型 RAM 2114 (1K×4ビット) を用いたもっとも簡単で確実な方法を述べる。

図2は 8K バイトのメモリーを増設し、そのアドレスを (8192)<sub>10</sub> から (16383)<sub>10</sub> 番地 (添字の 10 は 10 進数を意味する) に割り当てた場合の回路である。PET 2001-8 のアドレス・ラインは A<sub>0</sub>~A<sub>11</sub> と 4K バイトごとにデコードされた SEL1, SEL2, ..., SELB とからなっている。7442 は BCD- デシマル (10進) のデコーダで A, B, C, D の各入力端子に与えられる 4 ビットの BCD コードを、対応する「0」から「9」までの各出力端子にデコードする。たとえば、DCBA = "LLLL" のときは出力端子 0 のみが "L" で、他の端子はすべて "H" となり、DCBA = "LLHH" のときは出力端子 3 のみが "L" で他の端子はすべて "H" となる。

さて、PET のマニュアルによれば、SEL2 = "L" となるのは (8192)<sub>10</sub> から (12287)<sub>10</sub> 番地、そして SEL3 = "L" となるのは (12288)<sub>10</sub> から (16383)<sub>10</sub> 番地が、それぞれセレクトされたときのみに限られていることから、SEL2 = "L" のとき 7442 の出力端子 0 から 3 まで、SEL3 = "L" のとき 7442 の出力端子 4 から 7 までのそれぞれがセレクトされるようにした。このとき、SEL2 = SEL3 = "H" のときは出力端子 0~7 のどれもが "L" にならなければよいこと、SEL2 = "L" と SEL3 = "L" は同時には起り得ないことから、7442 の D, C 両端子に対する SEL ラインの論理関係は次のようになる。

SEL2	SEL3	D	C
L	L	*	*
L	H	L	L
H	L	L	H
H	H	H	*

ここで、\*印は "don't care 条件 ("L" または "H" のどちらでもかまわない)" を意味する。これより、回路実現が簡単になるように "don't care" 項を選ぶと、次のような論理式 (以下、正論理による) になる。

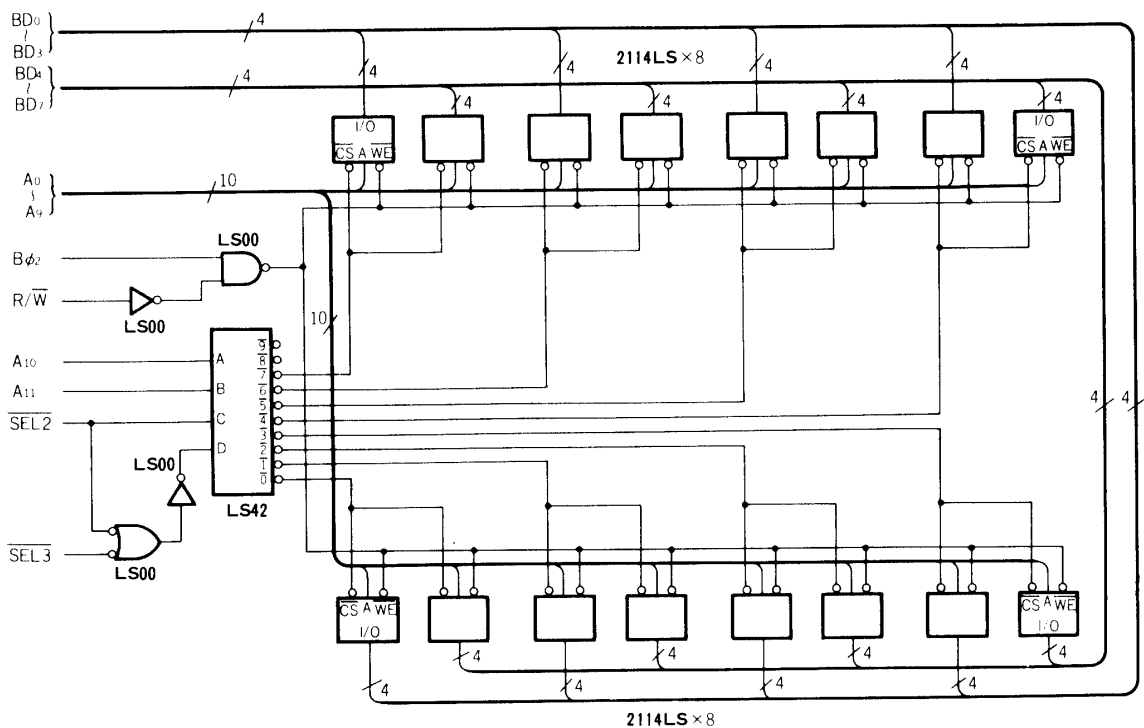


Fig. 2 8k byte memory board using static RAM 2114

$$C = \overline{\text{SEL2}}$$

$$D = \overline{\text{SEL2}} \cdot \overline{\text{SEL3}}$$

次に2114のWE端子は $B\phi_2 = "H"$ で $R/\overline{W} = "L"$ のときのみ $\overline{WE} = "L"$ として書き込みを可能にするから、

$$\overline{WE} = B\phi_2 \cdot \overline{R/\overline{W}}$$

よって、

$$\overline{WE} = B\phi_2 \cdot \overline{R/\overline{W}}$$

となる、以上のことから、7442のC, D端子、及び2114のWE端子への論理回路は図2のようになる。なお、この図において、SEL2, SEL3の代わりにどのSEL信号をつなぐかによってメモリのアドレス空間割り当てを4Kバイトごとに変更することができる。

メモリ・ボードは配線の誤りがなければ、確実に動作するはずである。動作が正常であればPETのスクリーン(CRT)上にフリー・エリアとして15359が表示される。なお、拡張ボードとこのメモリ・ボードの消費電流は合計で約0.8Aであった。

### 3.2 ダイナミック型RAM 4116による方法

前節ではスタチック型のRAMを用いた8Kバイト拡張ボードについて述べた。これを用いて更にメモリの増設を行なうことができるが、それに伴い消費電流が増大し、それに見合う電源を必要とすることとメモリからの発熱が相当増加する。そこで、ここでは消費電力の小さいダイナミック型RAMによって、一挙に24Kバイトまでメモリを増設し(実際には32Kバイトの容量があるが、そのうちの24Kバイトを使う)、PETのRAM領域を32Kバイトにする方法を述べる。(PET 2001-8にはすでに8KバイトのRAMが(0000)<sub>16</sub>から(1FFF)<sub>16</sub>番地まで内蔵されているので、(2000)<sub>16</sub>から(7FFF)<sub>16</sub>番地までを増設する)

#### 3.2.1 回路構成

ダイナミック型RAMは小型大容量、低消費電力という優れた性質をもっているが、反面これの制御信号はスタチック型RAMにくらべて格段に複雑である。表1は4116(または相当品)のリード、ライト、リフレッシュ、及び非選択の各サイクルにおける状態を示す。

表より、ライト、リフレッシュ及び非選択の各サイクルでは、Doutがフローティングになるため、DoutとDinを直接結んでデータ・バスに接続でき、回路が簡単になる。リフレッシュサイクルはRASオンリ・リフレッシュを用いる。

次に、(a), (b), (c)の3通りの非選択サイクルのう

Table 1 Status of D. RAM 4116 in read, write, refresh and chip non-selecting cycles

	リード・ サイクル	ライト・ サイクル	リフレッ シュ・サ イクル	非選択サイクル		
				(a)	(b)	(c)
RAS	A	A	A	H	H	A
CAS	A	A	H	H	A	H
WRITE	H	L	X	X	X	X
D	H/L	Z	Z	Z	Z	Z

H=High Level, L=Low Level, Z=Floating.

X=Don't Care, A=Active Cycle.

ち、メモリ・チップの消費電力を少なくするため(b)の方法を用いる。ただこの方法は制御が複雑で、チップの非選択にはRASを"H"にし、リフレッシュ時にはCASを"H"にすると同時にすべてのチップのRASを"L"にする必要がある。しかし、この方法は選択されないチップにはRASがインアクティブであるので、選択されないチップは電力を消費しないという利点をもつ。

以上のことを考慮して作られたメモリ・ボードを図3に示す。この回路は文献(1)のものを参考にし、PET用に若干の修正と改良を行ったものである。

改良点は、文献(1)ではリフレッシュ・コントローラ(i 3242)へのCOUNT信号を作る方法である。すなわち、上記文献の回路では、COUNT信号が"L"になるタイミングが単安定マルチバイブレータの動作によって決まっていたため、このタイミングが $\phi_2$ 信号の"L"になる直前、またはその近くなる可能性があり(このとき、i 3242のリフレッシュ・アドレス・カウンタの遅れがRASの立下りに間に合わなくなる)、そうならないように単安定マルチバイブレータのCR定数を調整する必要があった。そこで、図3の回路ではDフリップフロップ(74LS74)を用いて、このタイミングが必ず $\phi_2$ 信号の立ち上りで起きるようにした。これによって、CR定数は大体の値でよく無調整になった。

次に修正点は、リフレッシュ・コントローラ(i 3242)に与えるアドレス・ラインである。

16KダイナミックRAM用のリフレッシュ・コントローラ(i 3242)には14本のアドレス・ライン(A<sub>0</sub>~A<sub>13</sub>)を与える必要があるが、PET 2001-8にはA<sub>0</sub>~A<sub>11</sub>の12本しかバス・コネクタに出力されていない。基板からハンダづけによってA<sub>12</sub>, A<sub>13</sub>を取り出すことでこの問題は解決できるが、作業が面倒であり、また故障の心配もあるので、ここではコネクタからの信

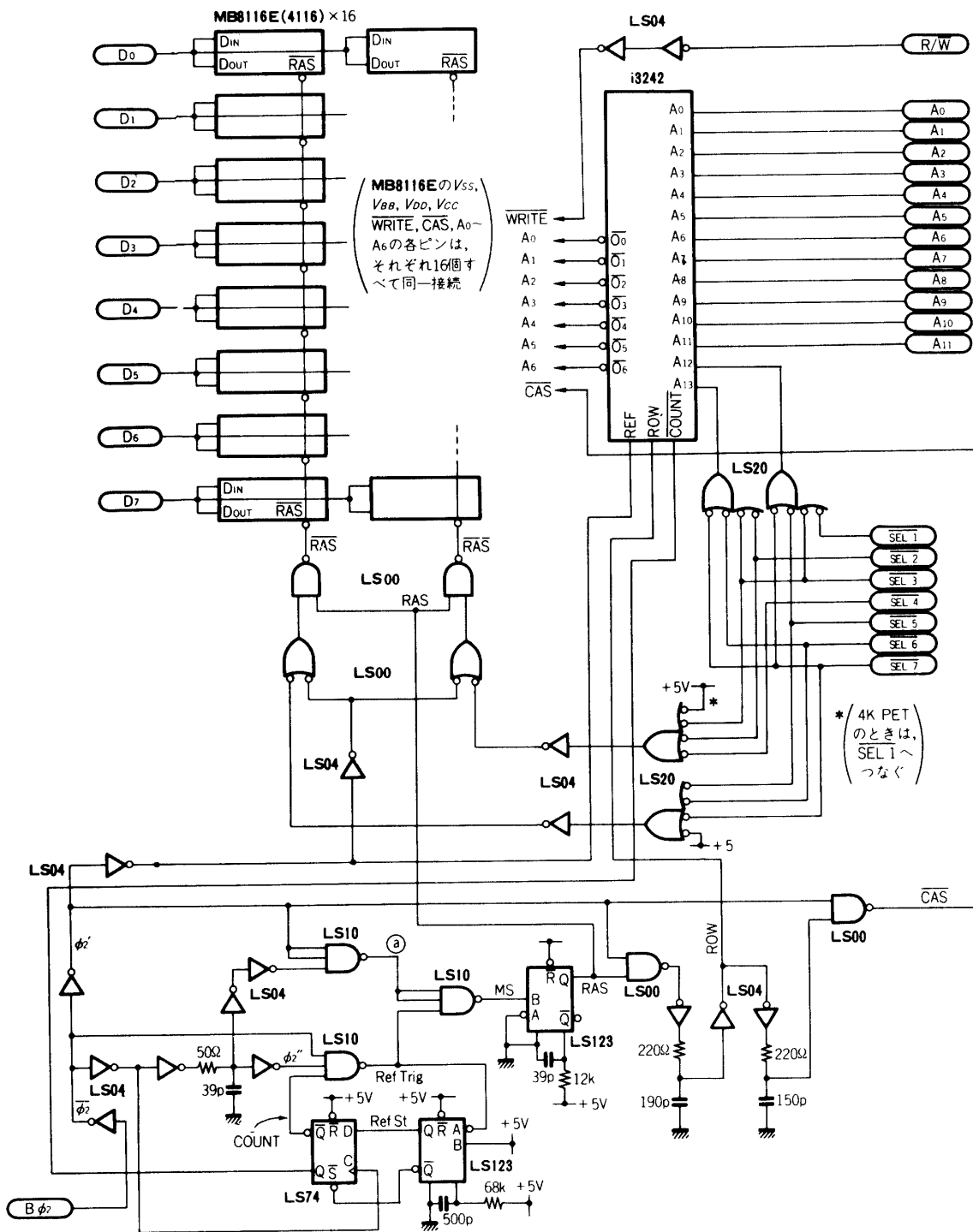


Fig.3 24k byte memory board using dynamic RAM 4116

号で  $A_{12}$ ,  $A_{13}$  を再生することにした。そのため 4K バイトごとにデコードして出力されている SEL ラインを用いる。

SEL1 はアドレス  $(1000)_{16} \sim (1FFF)_{16}$  が、SEL2 は  $(2000)_{16} \sim (2FFF)_{16}$ , ... , 等々がセレクトされたときのみそれぞれ "L" になることを考慮すると、

- ・  $A_{12}$  が "H" になるのは SEL1, SEL3, SEL5, SEL7 のいずれかが "L" のときである、
- ・  $A_{13}$  が "H" になるのは SEL2, SEL3, SEL6, SEL7 のいずれかが "L" のときである、

なることがわかる。これにより 4 入力 NAND ゲート (74 LS20) を用いれば  $A_{12}$ ,  $A_{13}$  の再生が簡単に実

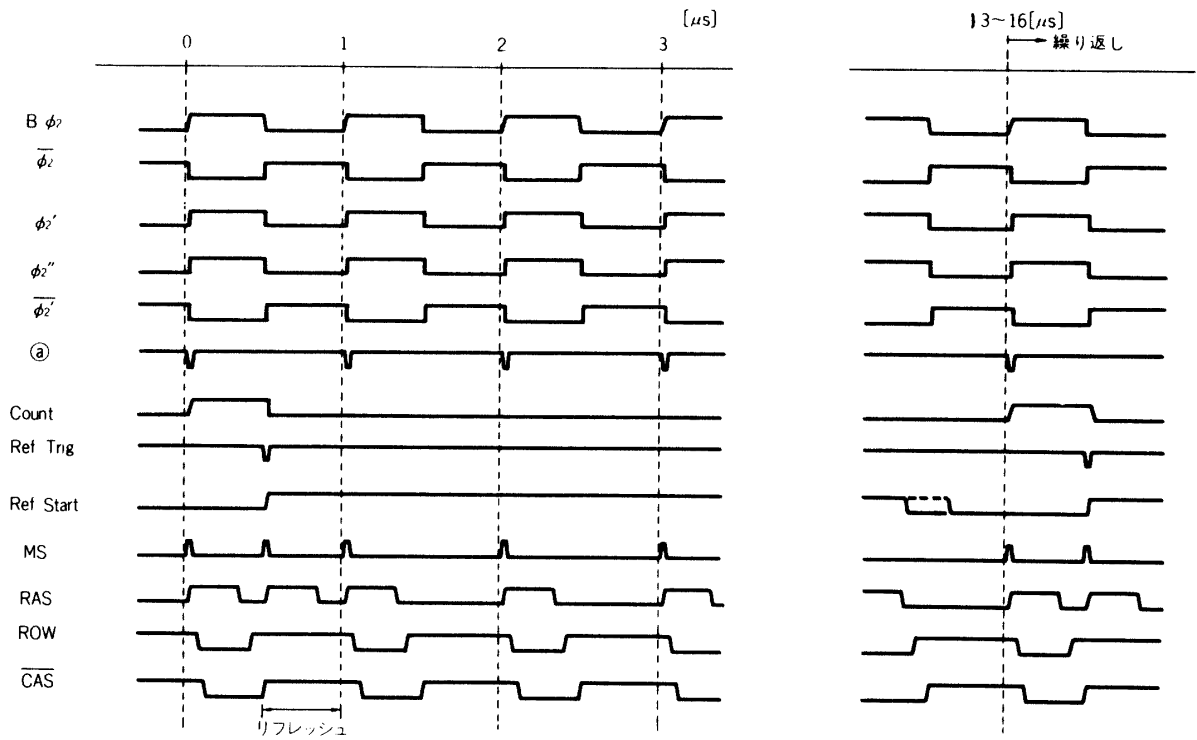


Fig.4 Timing chart

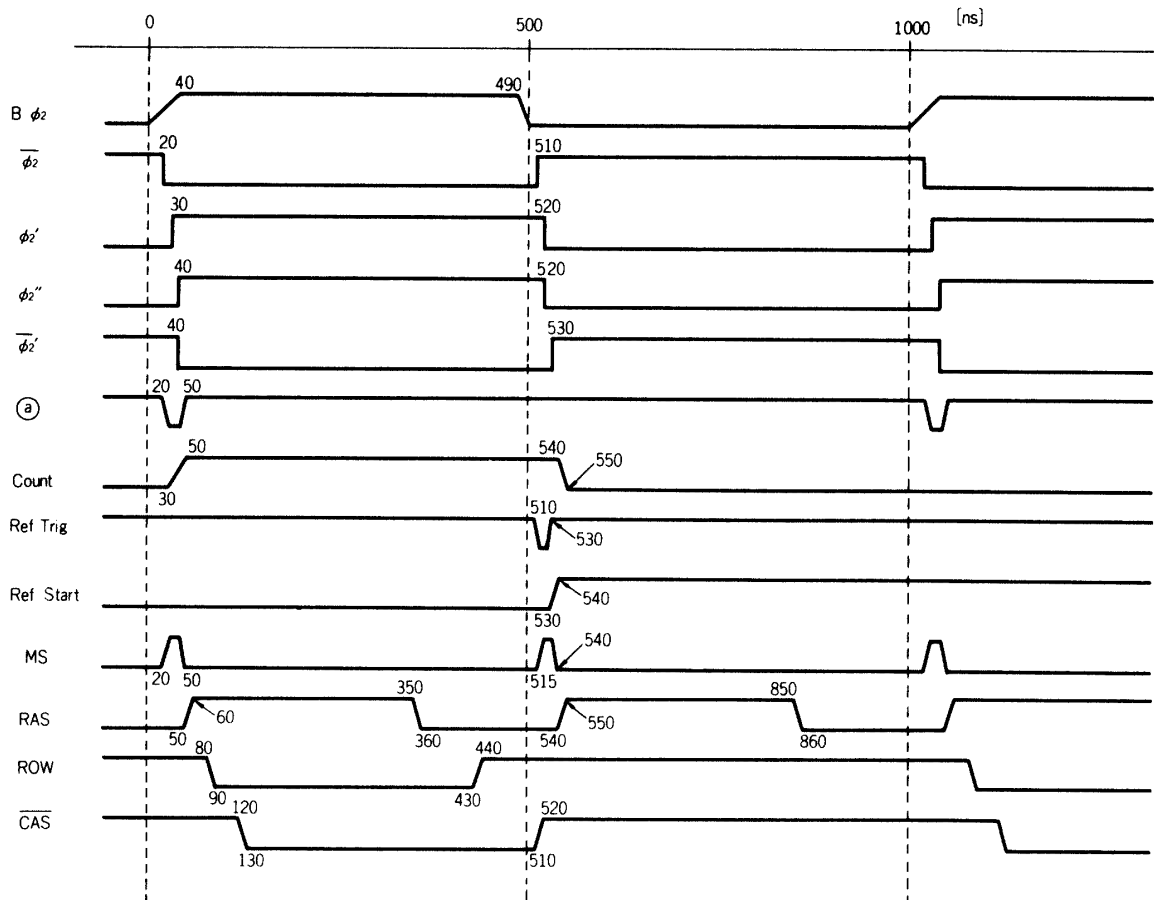


Fig.5 Expanded timing chart between 0 to 1  $\mu\text{s}$  in Fig. 4

現できることがわかる。

3.2.2 タイミング・チャート

図4および図5はタイミング・チャートで、図4は1回のリフレッシュから次の回のリフレッシュまでのタイミングを示し、図5は1回のリフレッシュの近傍(0~1μs)のようすを拡大したもので、実測値による詳細なタイミングを示している。リフレッシュはサイクル・スチール・リフレッシュ方式を採用しているので、リフレッシュ・スケジューラ的设计が容易になっている。リフレッシュ間隔は最長 2ms になっているので、128 個のリフレッシュ・アドレスを分散して行なうためには、約 15μs に1回のリフレッシュサイクルとなるが、それ程厳密なものではなく、かなりのマージンがある。

各信号のタイミングは使用した 16K ダイナミック型 RAM MB8116E (4116 相当) の規格を十分満足しているのがわかる。また、前節で述べた改良点の COUNT 信号がリフレッシュの始まる直前の φ<sub>2</sub> の立ち上がり時点で“H”にしたがって  $\overline{\text{COUNT}}$  信号は“L”になっているのが見られる。

製作上の注意点は月並なことであるが、IC 各について電源ラインに接続される各ピンとグラウンド・ピン間にバイパス・コンデンサを挿入し、また、基板のグラウンド・ラインにできるだけノイズが乗らないよ

うにための導線をグラウンド母線として用いるようにする。

できあがったメモリ、ボードに各種のメモリ・テストを行なった結果、完全に動作していることを確認した。長時間にわたる稼動にもかかわらず、発熱量も少なく、ダイナミック型 RAM の消費電力の少ないのが確認できた。

4. P-ROM (2708 用) ライターの接続

システムの拡張において、種々の開発したプログラムを ROM 化するのが望ましい場合がある。しかし、市販されている P-ROM ライターはかなり高価である。そこで、ここでは拡張ボードのコネクタを通して P-ROM (2708 用) ライターを接続する方法を述べる。このライターは極めて安価であり、PET の BASIC 言語を有効に用いているので、プログラムが大変見やすく、種々のコメント文を CRT に表示できるので、操作の間違いがなくなり、使いやすくなっているのが特長である。ここでは2708用についてのみ述べるが、同様の方法によって他のタイプの P-ROM の ROM ライターも容易に作製できる。

4.1 回路構成

図6に ROM ライターの回路図を示す。回路は大変シンプルであるので、動作を簡単に説明するにとどめ

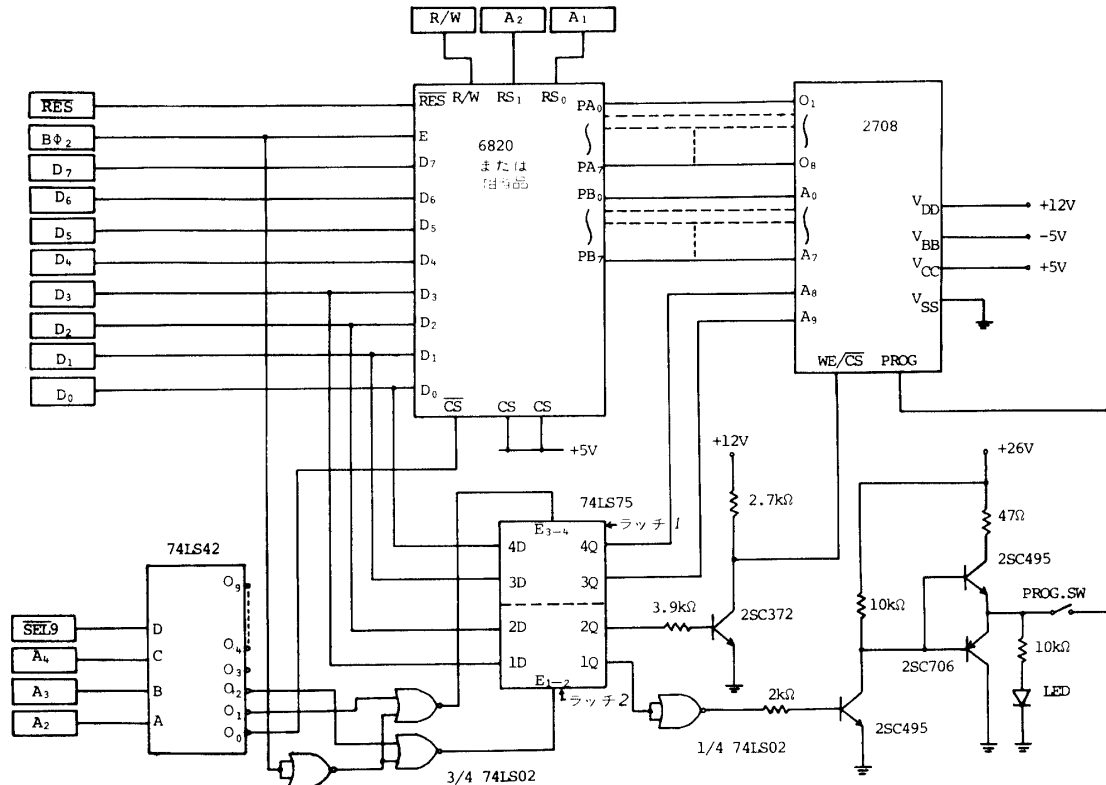


Fig.6 ROM writer circuit

る。

PIA 6820 の A ポートは書き込み時は出力ポートに、内容照合時は入力ポートに設定される。 B ポートは出力ポートとして動作し、 P-ROM のアドレスの下位 8 ビット ( $A_6 \sim A_7$ ) の設定をする。 ラッチ 1 (74LS 75 の上半分) は P-ROM のアドレスの上位 2 ビット ( $A_8$  と  $A_9$ ) の設定をする。 ラッチ 2 (74LS75 の下半分)

	アドレス
PIA の A ポート	$(9000)_{16} = (36864)_{10}$
	$(9001)_{16} = (36865)_{10}$
B ポート	$(9002)_{16} = (36866)_{10}$
	$(9003)_{16} = (36867)_{10}$
ラッチ 1	$(9004)_{16} = (36868)_{10}$
ラッチ 2	$(9008)_{16} = (36872)_{10}$

ここで、添字の 16 は 16 進数を、添字の 10 は 10 進数を意味する。

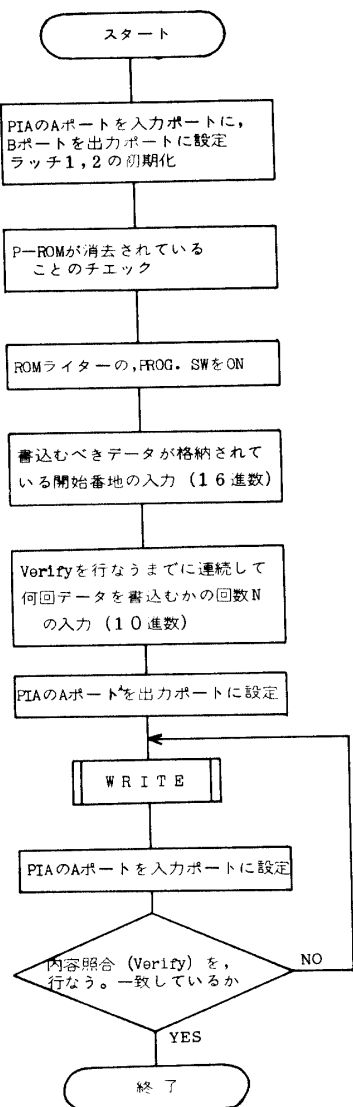


Fig.7a Total Flow-chart

は P-ROM のリード/ライトの設定とデータ書き込み時における P-ROM (2708) の PROG 端子へのパルス (幅は約 0.9msec) の供給を行なう。

PIA の A ポート、 B ポート、 ラッチ 1, 2 へのアドレスの割り当ては一応次のようにした。

#### 4.2 ソフトウェア

図 7 にフローチャートを、図 8 にプログラムを示す。プログラムは最も時間のかかる P-ROM への書き込み動作のみ機械語で書き、それ以外はすべて BASIC 言語で書いている。したがって、従来いろいろ発表されている機械語のみによる ROMライターにくらべて大変見やすいプログラムになっており、操作途中のコメント表示が豊富になされるので、操作の間違いがなくなるという特長をもっている。

プログラムを簡単に説明する。  $(50)_{16}$ ,  $(51)_{16}$  番地は P-ROM に書き込むべきデータが格納されている開始番地をポイントしている。  $(52)_{16}$  番地は連続した書

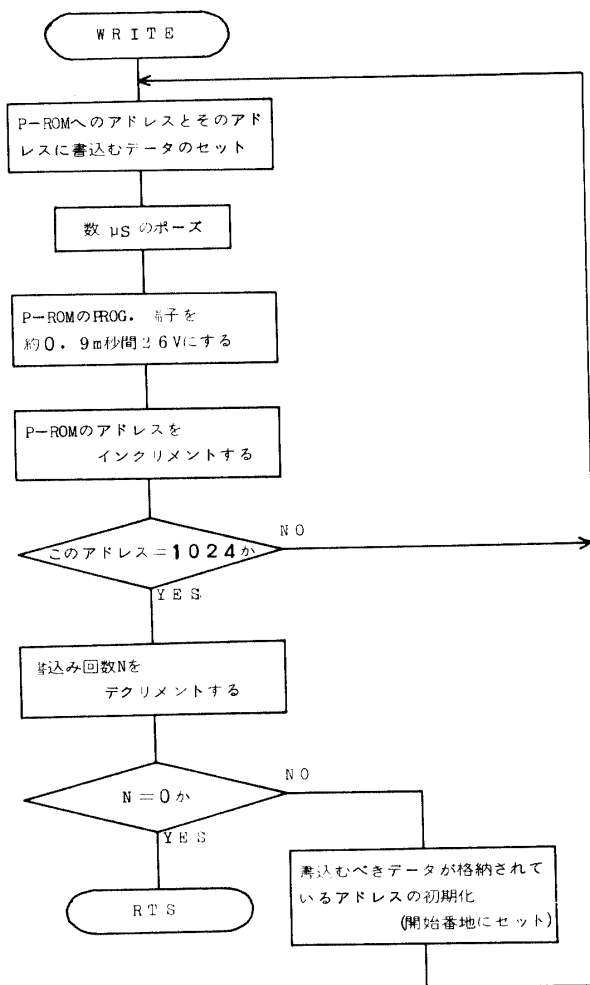


Fig.7b Flow-chart in writing period

```

1 REM** P-ROM LIGHTER **
5 GOSUB430
10 POKE36865,0:POKE36864,0:POKE36865,4:REM** PA IS IN **
20 POKE36867,0:POKE36866,255:POKE36867,4:REM** PB IS OUT **
30 POKE36868,0:POKE36872,4:REM** RESET OF THE LATCH 1 & 2 **
40 REM** CHECK OF ERASE **
50 FOR A=0 TO 1023:GOSUB400:PRINTA,PEEK(36864)
60 IF PEEK(36864)<>255 THEN PRINT"? ERASE ERROR":STOP
70 NEXTA
80 PRINT"ERASE OK !":PRINT"TURN THE SWITCH ON !"
90 REM** INPUT **
100 PRINT"TYPE THE START ADDRESS IN HEX.":INPUT S$
110 PRINT"TYPE THE REPEAT NUMBER IN DEC.":INPUT N
120 POKE36865,0:POKE36864,255:POKE36865,4:REM** PA IS OUT **
130 GOSUB 250:POKE36872,0
140 PRINT"WRITING !"
145 POKE36872,0
150 SYS(4096):REM** WRITE **
160 POKE36865,0:POKE36864,0:POKE36865,4:REM** PA IS IN **
170 PRINT"VERIFYING !"
175 POKE36872,4
180 FOR A=0 TO 1023:GOSUB 400:PRINTA,PEEK(36864)
190 IF PEEK(36864)<>PEEK(A+A) THEN PRINT"? VERIFY ERROR":PRINT"RE":GOTO130
200 NEXT A
210 PRINT"CONGRATULATION !":PRINT"VERIFY OK !":PRINT"TURN THE SWITCH OFF !"
220 END
240 REM** HEX TO DEC & SET ADDRESS **
250 AB=0
260 FOR I=1 TO 2:C=0
270 FOR J=1 TO 2
280 B=ASC(MID$(S,J+2*1-2,1))-48
290 IF B>10 THEN B=B-7
300 C=C+16^(2-J)*B
310 NEXT J
320 POKE36872-1,C:A0=AB+256^(2-1)*C
330 NEXT I
340 RETURN
399 REM** SET PB & LATCH 1 **
400 POKE 36868,A/256
410 POKE 36866,A-INT(A/256)*256
420 RETURN
430 FOR I=4096TO4161
440 READ M
450 POKEI,M
460 NEXT I
470 RETURN
500 DATA 160, 0, 162, 0, 177, 80, 141, 0, 144, 140, 2, 144, 142, 4, 144
510 DATA 72, 104, 234, 234, 169, 8, 141, 8, 144, 169, 50, 141, 76, 128, 72
520 DATA 104, 206, 76, 128, 208, 249, 169, 0, 141, 8, 144, 200, 208, 216, 230
530 DATA 81, 232, 224, 4, 208, 209, 198, 82, 240, 10, 56, 165, 81, 235, 4
540 DATA 133, 81, 76, 0, 16, 96
    
```

Fig.8 Program list

```

. D
*** P-ROM WRITER ***
1000 1042

1000 A0 00      LDY #000
1002 A2 00      LDY #000
1004 B1 50      LDA ($50),Y
1006 8D 0090    STA $9000
1009 8C 0290    STY $9002
100C 8E 0490    STX $9004
100F 48         PHA
1010 68         PLA
1011 EA         NOP
1012 EA         NOP
1013 A9 08      LDA #08
1015 8D 0890    STA $9008
1018 A9 32      LDA #32
101A 8D 4C80    STA $204C
101D 48         PHA
101E 68         PLA
101F CE 4C80    DEC $204C
1022 D0 F9      BNE $101D

1024 A9 00      LDA #00
1026 8D 0890    STA $9008
1029 C8         INY
102A D0 D8      BNE $1004

102C E6 51      INC $51
102E E8         INX
102F E0 04      CPX #04
1031 D0 D1      BNE $1004

1033 C6 52      DEC $52
1035 F0 0A      BEQ $1041

1037 38         SEC
1038 A5 51      LDA $51
103A E9 04      SBC #04
103C 85 51      STA $51
103E 4C 0010   JMP $1000

1041 60         RTS
    
```

Fig.9 Machine subroutine for writing operation

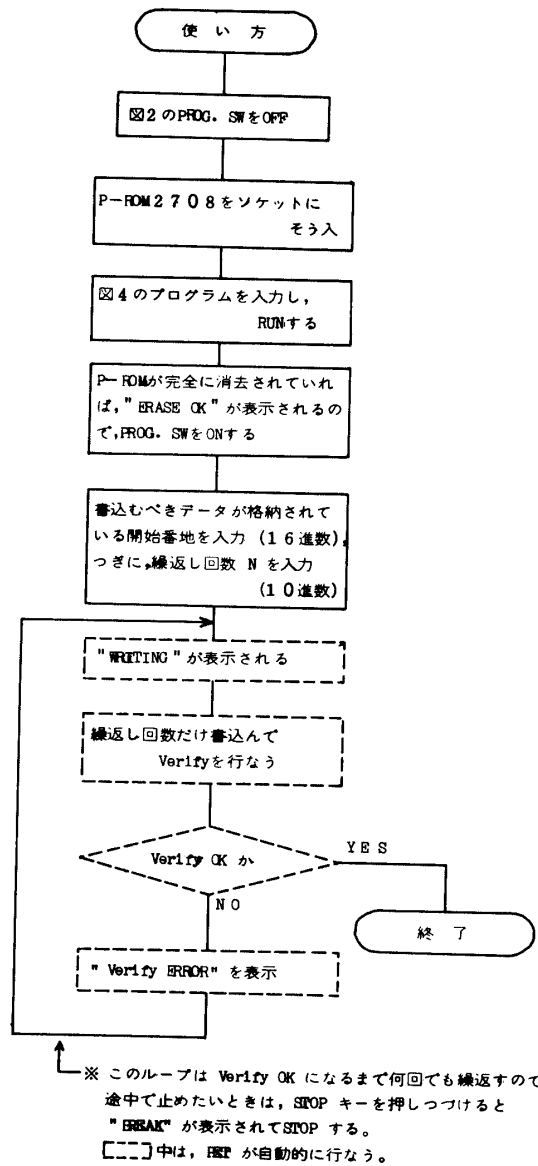


Fig.10 Flow-chart for use

き込み動作の繰返し回数Nを格納しており、この値はユーザがキーボードより適当に入力する。これは次のような理由によっている。

2708 の PROG 端子に与えるべき +26V のパルスのパルス幅は各アドレスについて1回 1msec 以内と規定され、しかも、各アドレスについてパルス幅の加算値が 100msec になるように全アドレスを走査しながら繰返しパルスを与えるようになっている。この条件を満たすようにパルスを与えると、アドレスは1024個あるので、 $1024 \times 100\text{msec} = 102.4\text{sec}$  は少なくともかかることになる。しかし、実際には10回前後の書き込みで済むようである。一方、内容照合は BASIC 言語で行なっているので、一回書き込みごとに内容照合



をしていたのでは照合に時間がかかる。この二つの時間のかねあいをみて、連続した書き込み回数をキーボードより入力するようにした。

書き込み動作を行なう機械語プログラムは文番号 500 以降の DATA 文で入力される。この機械語部分の逆アセンブル・リストを図 9 に示す。

#### 4.3 使い方

図 10 にこの ROM ライターの使い方をフローチャート型式で示す、繰返し回数  $N$  は 10 前後の値を入力すれば大体一度の書き込み動作で書き込みが完了するようである。

#### 5. むすび

マイクロ・コンピュータとしては、すぐれたソフトウェアを内蔵する PET を有効に用いるために、まずメモリの増設を行ない、PET の一応用として、P-ROM ライターの作製とその PET への接続法を述べた。

ソフトウェアの開発には相当の時間を要するとき、すぐれたソフトウェアをもつパーソナル・コンピュータをいろいろなものに応用できれば、ソフトウェア開発の手間も省ける。このとき、用いるパーソナル・コンピュータのハードとソフトウェアの理解及びモニターの解読を必要とする。この点、PET2001-8 は最も拡張性に富んだ機種の一つであり、今後、種々の応用を考えている。

#### 参 考 文 献

- 1) 松本吉彦, “メモリ・システムの設計①~⑤” トランジスタ技術, 1978. 11 (p.278), 1978. 12 (p.287), 1979. 1 (p.273), 1979. 2 (p.267), 1979. 3 (p.267) CQ 出版社
- 2) 小川晃 “6800 用 PROM ライターとイレーサの製作”, RAM, 1978. 12 (p.22), 広済堂出版
- 3) MCS 6500 Microcomputer Family Hardware Manual. MOS TECHNOLOGY, INC. (1976)

(昭和 54 年 10 月 15 日 受理)