Zapper

A Computer Driven EROM Programmer

G H Gable
50 Cliftwood Dr
Halesite NY 11743

One of the most fascinating and useful products of recent technology is the read only memory (often abbreviated as ROM) and especially useful for the experimental systems designer is the erasable and electrically programmable read only memory, variously abbreviated EROM or EPROM.

In designing my first microprocessor based system, a read only memory was a must to contain the operating system and the floating point arithmetic firmware. I did extensive research into read only memory systems and after a week or so I was ready to make a specification. I had previously chosen the processor for the system to be the MOS Technology 6502 which requires a memory access time of about 500 ns when running with a 1 MHz clock. It was very desirable to have the read only memory meet this specification for two reasons. First, because of the dynamic nature of the 6502, it does not wait for slow memory very readily. Second, and by far most important, I wanted my arithmetic routines in read only memory to run as fast as possible since I would be using them very often. These considerations ruled out the older 1702 type memories as too slow.

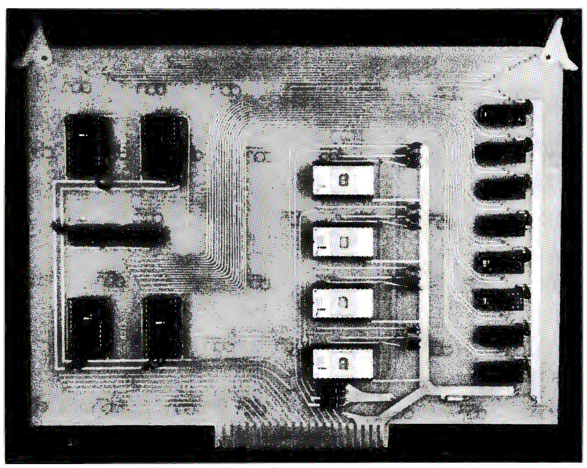


Photo 1: The prototype computer read only memory board with four EROM parts in the center. The eight sockets on the right hold the 1 K bytes of programmable memory which is selected by a jumper on the pins in the lower center of the board (off pin 24 of the lowest EROM). The 8212s on the left are the address bus drivers (lower pair). 8212s are abundantly used because they are the author's favorite all purpose medium scale integration chip.

The choice was obvious as soon as I read about the Intel 2708. It had all the requisite features: fast (450 ns) access time, large array (1024 8 bit words) on a single chip, and easy straightforward programming. When I designed this programmer the going price was $100; currently the prices have dropped to about $10, making this chip even more desirable.

The chip is also numbered 8708 to fit into Intel's 8000 line which includes the 8080. The 2708 and the 8708 are identical as far as I know. They are definitely interchangeable at a pin level. There is also a variation of the design called the 2704/8704 which is arranged as an array of 512 8 bit words. The 2704/8704 is electrically and logically identical to the 2708/8708 but contains only half as much memory. The high order address line is not defined for the 2704/8704. (Rumor has it that 2704/8704 parts are identical to 2708s but wired into the package with the high order address bit unconnected.)

System Design

My design called for 4 K bytes of read only memory resident firmware which could be built up over a period of time as the operating system and arithmetic routines were debugged. My approach to this was to prototype the eventual firmware in normal

 December 1978 © BYTE Publications Inc

programmable memory and then transfer it to read only memory after debugging. I designed a 4 K byte read only memory board (photo 1) which has four 2708 PROM chips plus 1 K bytes of programmable memory. The programmable memory can be jumper selected to occupy any 1 K page on the board. This allows for prototyping a routine in the actual address space that it will eventually occupy. The system has worked out extremely well.

It was my original intention to have the read only memory programmed by professionals offsite. My impression was that 2708 programming was somewhat complex and that a programmer board for a limited number of burns was not very practical. After learning more about the 2708 my attitude changed. A little thought convinced me that a computer driven programmer could be simply constructed at minimum cost. It would be very convenient to be able to program the chips in my own computer and to be able to make changes and corrections with a short turnaround time.

Programming the EROM

When initially received, and after each erasure, all the bits of the 2708 are in the "1" state (output high). The content of the 2708 is programmed by selectively changing state to "0" in the desired bit locations. Programming a given byte requires the address of the byte on the address input pins and the data byte on the data pins, all at TTL levels (+5 V) with the write enable pin held at +12 V, a program pulse of +26 V at 20 mA is applied to the program pin. The 2708 specifications require that the program pulse be between 100 μs and 1000 μs wide. A series of pulses are required to program a particular address. Intel recommends that one pulse be administered to each address location in a loop. The number of times the loop must be repeated is a function of the pulse width. The final accumulated program current time to each address must be greater than 100 ms. Such a scheme is a natural for computer control.

The Zapper programming board shown in photo 2 and figure 1 is designed to have the address and data multiplexed to it through a peripheral interface adapter (PIA) with at least eleven output lines. I use the peripheral interface adapter that is available on my MOS Technology KIM-1 single board computer to drive the Zapper. If you do not have one of these PIAs I recommend either the MOS Technology 6520 or the Motorola 6820. The address and data are passed through the lower eight lines (PA0-PA7) while three of the upper lines (PB0-PB2) control the multiplexing and programming current.

The driving computer is expected to direct the following sequence of events which will program one address location in the 2708:

- PB0 is brought high to enable the upper 8212 (IC1) eight bit latch.
- The lower eight bits of the address are loaded on PA0-PA7 and thus into the 8212.
- PB0 is brought low latching the low address onto the outputs of IC1 which are wired to the address inputs of the 2708.
- PB1 is brought high to enable the lower 8212 (IC2). The upper two bits of the address are loaded on PA0-PA1 and latched when PB1 goes low.
- The data byte is loaded on PA0-PA7 and latched by the PIA.
- PB2 is brought high for the pulse time gating the program current to the EROM.

This sequence is repeated the required number of times to program the EROM.

A Note About UV Bulbs:

MSC Macalaster is located at Rte 111 and Everett Tpk, Nashua NH 03060.

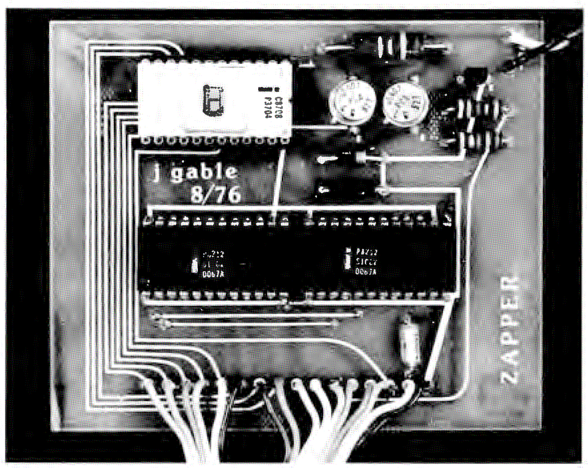


Photo 2: The Zapper board with the EROM in the upper left corner. Data from the PIA as well as logic power and ground come in via the ribbon cable at the bottom which is connected directly to the computer. Program power comes in on the cable in the upper right corner from an external power supply.

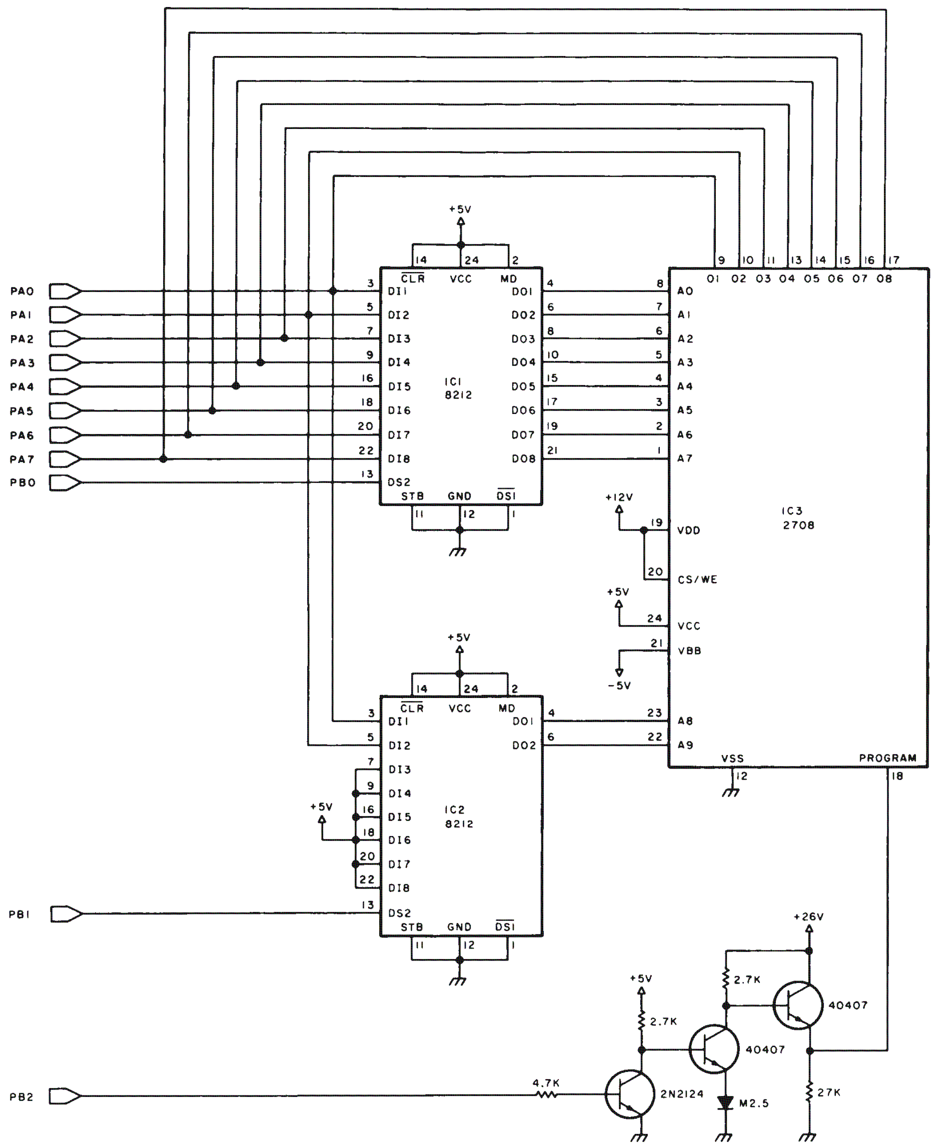


Figure 1: The address and data information for the Zapper is multiplexed through the PIA ports PA0-PA7 while control signals are presented on PB0-PB2. PB0 is connected to the enable pin of the upper 8212 which latches the lower eight bits of the address. The high two bits of the address are loaded and latched on the lower 8212 by PB1. The data byte is latched by the PIA. When PB2 goes high, program power is gated to the program pin of the 2708 by the 3 transistor high current gate in the lower right.

```
                        ;       PROGRAM ZAPPER
                        ;
                        ;       THIS LISTING WAS PREPARED
                        ;       FROM A HAND ASSEMBLED SOURCE
                        ;
0200  A9  FF            ZAP     LDA  $FF      INITIALIZE CYCLE
0202  85  05                    STA  CYR      COUNT TO 255
0204  8D  01  17                STA  PADD     SET PIA DIRECTION
0207  A9  7F                    LDA  $7F      REGISTERS FOR
0209  8D  03  17                STA  PBDD     OUTPUT PORTS
020C  A9  00                    LDA  $00      SET CONTROL PORTS
020E  8D  02  17                STA  PBD      TO "OFF"
0211  20  71  02                JSR  MSG      WAIT FOR PROG POWER
0214  A2  00                    LDX  $00      SET INDIRECT INDEX
0216  A5  00            NXCY    LDA  BSL      TRANSFER STARTING
0218  85  10                    STA  LRL      ADDRESS TO LOCATION
021A  A5  01                    LDA  BSH      REGISTER
021C  85  11                    STA  LRH
021E  20  3F  02        NXAD    JSR  BURN     BURN PULSE
0221  E6  10                    INC  LRL      INCREMENT
0223  D0  02                    BNE  A1       LOCATION REGISTER
0225  E6  11                    INC  LRH
0227  A5  10            A1      LDA  LRL
0229  C5  02                    CMP  BEL      COMPARE LOCATION
022B  D0  F1                    BNE  NXAD     REGISTER WITH END
022D  A5  11                    LDA  LRH      ADDRESS TO CHECK
022F  C5  03                    CMP  BEH      FOR END OF BLOCK
0231  D0  EB                    BNE  NXAD
0233  C6  05                    DEC  CYR      DECREMENT CYCLE
0235  D0  D5                    BNE  NXCY     COUNT
0237  A9  07                    LDA  $07      RING TTY BELL
0239  20  A0  1E                JSR  OUTCH    WHEN DONE AND
023C  4C  4F  1C                JMP  MONITOR  RETURN TO MONITOR
023F
```

Listing 1: The Zapper program programmable memory starting address (BSL,BSH) and ending address plus one (BEL,BEH) are set before execution. The driving program sets up the PIA ports as outputs and insures that the control lines (PBD) are off (zeros) before programming power is applied. The loop through the addresses in the location register (LRL,LRH) supplies a burn pulse for each location. The cycle is repeated so that each location receives 255 pulses. The end of the program is signaled by the Teletype bell or terminal signal.

```
                        ;
                        ;       PROGRAM ZAPPER/BURN
                        ;
023F  A9  01            BURN    LDA  $01      OPEN LOW ADDRESS
0241  8D  02  17                STA  PBD      BUFFER
0244  A5  10                    LDA  LRL      GET LOW ADDRESS AND
0246  8D  00  17                STA  PAD      PUT IN BUFFER
0249  A9  02                    LDA  $02      OPEN HIGH ADDRESS
024B  8D  02  17                STA  PBD      BUFFER
024E  A5  11                    LDA  LRH      GET HIGH ADDRESS AND
0250  8D  00  17                STA  PAD      PUT IN BUFFER
0253  A9  00                    LDA  $00      LATCH ADDRESS
0255  8D  02  17                STA  PBD      BUFFERS
0258  A1  10                    LDA  LRL(IX)  GET DATA AND
025A  8D  00  17                STA  PAD      LATCH
025D  E6  04                    INC  WR       WAIT FOR DATA TO SETTLE
025F  A9  04                    LDA  $04      TURN ON PROGRAM
0261  8D  02  17                STA  PBD      POWER
0264  A0  43                    LDY  $43      WAIT 600US
0266  E6  04            AGN     INC  WR
0268  88                        DEY
0269  D0  FB                    BNE  AGN
026B  A9  00                    LDA  $00      TURN OFF PROGRAM
026D  8D  02  17                STA  PBD      POWER
0270  60                        RTS           RETURN
0271
```

Listing 2: The burn subroutine multiplexes the address and data through the PIA port (PAD) controlled by the control lines (PBD). A 600 µs programming pulse is applied after the address and data have been latched. The INC WR instruction does nothing more than provide a 5 µs delay. It is used first to let the data lines to the EROM settle before the programming pulse is applied. Later it is used in the pulse timing loop simply to cut down the number of iterations.

Software

The driving software, as shown in listings 1 to 4, implements the above sequence of events in a double loop. The inside loop, listing 2, works its way through all the addresses to be programmed and gives each location a 600 µs programming pulse. The outer loop, listing 1, repeats the process 255 times giving a total program current time of 153 ms to each bit. This is sufficient time to program the 2708. The start and end plus one addresses of the programmable memory block are loaded in BSL, BSH and BEL, BEH registers respectively before execution is begun. Data is programmed into the same relative addresses in the read only memory as they are found in the programmable memory; ie: the low ten bits of the address are the same.

Notice that the 2708 can be partially programmed. If the memory block to be copied is less than 1024 bytes long, only the appropriate bytes are programmed. The remaining locations are unchanged. The block to be programmed can start and end anywhere in the 1 K page. This is a very useful feature as it allows firmware to be developed over a period of time. The partially programmed read only memory can be used in the meantime. Incidentally, listings 2 and 3 are subroutines only for the sake of modularity and the whim of the author. They are called at only one point each.

It is very important that the +26 V programming power be off at the power supply until the computer has had a chance to latch PB2 low. After this initialization, a pause is built in to allow the operator to turn on the power supply before continuing. This pause is implemented by waiting for input from a terminal in subroutine MSG, listing 3. The application of program power before the computer has initialized the Zapper board will usually result in some random location being burned with some random data.

Erasing the EROM

The 2708 is very easily erased using an ultraviolet light source. Intel specifications indicate that an integrated dose of 10 watt-sec/cm^2 at a wavelength of 2537 angstroms is required to erase the 2708. A quick glance at the *CRC Handbook of Chemistry and Physics* shows that 2537 angstroms is the most persistent spectral line of mercury (Hg). This means that any mercury vapor lamp will do the trick. I use a nice packaged source from MSC Macalaster (Catalog #3400) which slips over the top of the read only memory. (When using the unit, discard the filters which come with it, and be sure you shield your eyes from the lamp.) The chip

```
                         ;
                         ;      PROGRAM ZAPPER/MSG
                         ;
0271  A2  00             MSG    LDX  $00      OUTPUT THE MESSSAGE
0273  BD  B1  02         M1     LDA  SMG+X    FROM DATA BLOCK
0276  C9  00                    CMP  $00      (SMG) UNTIL "NUL"
0278  F0  07                    BEQ  RET      CHARACTER FOUND
027A  20  A0  1E                JSR  OUTCH
027D  E8                        INX
027E  4C  73  02                JMP  M1
0281  20  5A  1E         RET    JSR  GETCH    WAIT FOR KEYSTROKE
0284  60                        RTS           BEFORE RETURNING

0285  0D  0A  0A         SMG    DATA           CR/LF/LF
0288  54  55  52                               "TURN ON 26V--
028B  4E  20  4F                               PUSH ANY KEY"
028E  4E  20  32                               CR/LF/LF/NUL
0292  36  56  20
0295  2D  2D  20
0298  50  55  53
029B  48  20  41
029E  4E  59  20
02A2  4B  45  59
02A5  0D  0A  0A
02A8  00
02A9
```

Listing 3: The MSG routine effectively causes a pause so that programming power may be turned on after the Zapper board has been initialized. Execution is resumed when any key on the Teletype is pressed.

HEXADECIMAL LOCATION	SYMBOL	COMMENTS
0000	BSL	STARTING ADDRESS OF
0001	BSH	PROGRAMMABLE MEMORY.
0002	BEL	ENDING ADDRESS PLUS ONE OF
0003	BEH	PROGRAMMABLE MEMORY.
0004	WR	WAIT REGISTER.
0005	CYR	CYCLE COUNT REGISTER.
0010	LRL	LOCATION REGISTER.
0011	LRH	
1700	PAD	PA PORT DATA REGISTER.
1701	PADD	PA PORT DIRECTION REGISTER.
1702	PBD	PB PORT DATA REGISTER.
1703	PBDD	PB PORT DIRECTION REGISTER.
1EA0	OUTCH	TTY OUTPUT ROUTINE.
1F5A	GETCH	TTY INPUT ROUTINE.

Listing 4: External symbol table. The PIA registers (PAD, PADD, PBD, PBDD) are those assigned on the KIM-1 board. OUTCH and GETCH respectively output and input one character each to or from a terminal. They are part of the KIM-1 operating system.

should be stuck in a piece of conducting foam while erasing. An exposure time of 30 to 40 minutes will yield a fresh chip ready to be programmed again. If you want to make your own eraser, use the GE #G4S11 4 W mercury vapor lamp with a GE #89C504 ballast. Both of these items are usually available at commercial electrical supply houses. The exposure time is about 40 minutes with the 2708 placed 1 cm from the bulb.

My experience shows that each successive time a 2708 is erased the exposure time to completely erase it increases. As the total energy needed to erase it is cumulative, extra short exposures can be given as needed. A little program to check each byte for all ones will assure that the memory is fully erased.

It is also convenient to remember that any 1 bit in the EROM can be changed to 0. Sometimes a single byte needs to be modified and this can occasionally be done without erasing the EROM and reprogramming it. This has been the case for me more often then statistics would dictate. Someone else must not be nearly so lucky.■

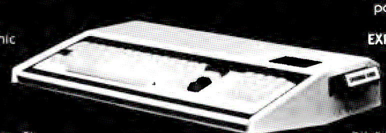
REFERENCES

1. *Intel Data Catalog,* Intel Corp, 1975.
2. *Memory Design Handbook*, Intel Corp, 1975.
3. *8080 Microcomputer Systems User's Manual,* Intel Corp, September 1975.

Intel Corporation
Literature Dept
3065 Bowers Av
Santa Clara CA 95051

THE PS-80

KEYBOARD: fully professional. Full 128 ASCII upper & lower case characters with 79 keys including a 16 key numeric pad.
CPU: Z-80 Processor
GRAPHICS: 64 pre-defined graphic chars. and 64 user defined chars. alternately all 128 graphic chars. may be user defined. Resolution 240 x 512 points, cursor control, and 64 chars. by 30 lines.
MEMORY: 4K byte power-on monitor, 8k byte user RAM, and 8k byte PROM external cartridge with Microsoft BASIC standard. Cartridges allow you to change operating systems & languages without any modification to the standard hardware by simply inserting a PROM cartridge in a side slot.
I/O: serial RS232 300/1200 baud port, full parallel port, dual cassette recorder port at 300/1200 baud.
EXPANSION: up to 32k RAM on board I/O to S-100 extension box for additional memory and any other S-100 peripheral boards.
PRICE: $895 (does not include CRT or cassette.)
8K RAM $895.00 available soon
16K RAM $1150.00 available now.
32K RAM $1395.00 available now.



EDUCATIONAL AND CLUB DISCOUNTS AVAILABLE
PERSONAL CONSULTANT AND DEALER INQUIRIES INVITED
See us at booth 233 at the West Coast Computer Faire.
For complete information write to: PERSONAL SYSTEMS CONSULTING P.O. Box 20286 El Cajon, California 92021
(714) 443-5353
Photography by James & Linda Neufeld
Ad Design Joanne De Vore Kwik Kopy Printing #215

Circle 303 on inquiry card.

Circle 201 on inquiry card.

MMike's
Program Library for
NORTH STAR ★ COMPUTERS
$500 — ONE TIME FEE

Get ALL the programs we write or update FREE for life! Includes many business application programs (Osburne & Associates), utilities, public domain programs, and modifications to North Star Extended Basic that allow appending as an executable statement and the addition of 8" floppy disks drives to your present North Star System. (Hard disk units to be available in near future). Call or write for free copy of Micro Mike's Program Library Newsletter.

Complete Business Systems

with modem connection to our Program Library

1. Horizon II Computer
2. 32KRAM
3. ADM-3A Terminal
4. IP-125 Printer
5. DC Hayes Modem Board
6. All Connectors & Cables

$4995.00

Expandable anytime by addition of 8" drives.

North Star Owners

move up to

8" Floppy Disks

Completely compatible with your present system. Includes all software necessary to run your present programs on either 8" or 5¼" drives or both.

$995 1st drive & controller
$795 each additional drive

Micro Mike's
905 Buchanan, Amarillo, Texas 79101
806-372-3633

KIMSI

The KIM to S-100 bus Interface/Motherboard

- Combines the power of the 6502 with the flexibility of the S-100 bus
- Attaches to any unmodified KIM
- Complete interface logic and fully buffered motherboard in one unit
- On-board regulation of power for KIM
- Eight slots of S-100 compatibility for additional RAM, Video and I/O boards, PROM Programmers, Speech processors . . .
- Includes all parts, sockets for ICs, one 100 pin connector, and full Assembly/Operating documentation

◆ *Kit $125, Assembled $165*
◆ *All units shipped from stock*

FORETHOUGHT PRODUCTS
87070 Dukhobar Rd. # F
Eugene, Oregon 97402

Circle 140 on inquiry card.

Office of Programming and Adjustment, where he underwent a battery of tests which proved that he was five years older. His Minnie was sent to one of many laboratories where highly skilled technicians made new program chips and inserted the chips in the Minnie to replace the ones which had served Twombly well for five years. It was late afternoon when Twombly left; an hour after that, one of the technicians approached the lab chief with an almost microscopic program chip in the palm of his hand.

"We have a condition red, I think," he told the chief. "This is the alternate program entertainment chip from Twombly's Minnie."

"Carson," said the chief, "that simply cannot be. He couldn't get out of the building without a full complement of chips; the master computer wouldn't let him through the door."

Carson, his face almost as red as the little dot on the chip which meant alternate program, said: "He had a full complement of chips. I got the wrong one in. He got an experimental chip I was designing for my wife's Minnie."

"What kind of an experimental chip?" asked the chief in tones that made Carson's flesh creep.

"You might call it a babysitting chip," said the technician, "although it doesn't just sit. I can tell you that we're in a great deal of trouble if he activates that chip. We have to prevent that."

"Condition red," sighed the chief. "We have to key into his Minnie by way of the house computer, but we'll have to get authorization from Washington. I'll notify Harris; it's his problem. He won't like it much."

"I don't think we have time. He'll most likely activate the entertainment chip after he finishes dinner; Twombly is predictable."

"We have to take time. After that J E Lewyt scandal, where the untouchability of our beloved director was found wanting, we've been under very rigid orders about invading the privacy of private computers. We've got to get authorization."

They got it after a three hour delay, but as Carson feared, it was too late. When the special code got them access to the Twombly house computer, it reported that Twombly had activated the alternate program entertainment chip. The chief sighed and requested a complete readout from the time of activation.

CHIP ACTIVATED 2030 HOURS. SEQUENCE COMPLETED: UNDRESSING, BATHING, DRYING, POWDERING, DIAPERING. AS INSTRUCTED BABY HAS BEEN PUT TO BED

A[1:100] in ALGOL (the 1 here is the lower bound on subscripts, which may be arbitrary in ALGOL, although it is always 1 in FORTRAN).

(13) **REAL.** In ALGOL, the **REAL** attribute refers to representation as a floating point number. *[Note that the attribute* **FLOAT** *performs this function in PL/I, and that* **REAL** *in PL/I is used only to distinguish real from complex numbers. . .***BL***]*

(14) **INTEGER.** BASIC assumes that all numbers are real; integers will be treated as if they are real numbers, which usually works the way we want it to, although some operations like division must be watched carefully. In ALGOL, all integers must appear in **integer** statements.

(15) **COMMON.** In PL/I, all main routine variables are common (called "global" in PL/I parlance) to internal subroutines (ie: the subroutine is declared by a PROCEDURE statement within the boundaries of the calling PROCEDURE and its END) unless it is redefined in the subroutine. The EXTERNAL attribute is used to share variables between external procedures. In ALGOL, any variable in a main program may automatically be used in any of its subroutines, unless there is another variable declared in the given subroutine that has the same name.

(16) **Assignment statements.** In ALGOL, the symbol := is used where = is used in FORTRAN, BASIC, and PL/I. In addition, = is used where .EQ. is used in FORTRAN. Some versions of BASIC permit, and some require, the word **LET** at the beginning of every assignment statement.

(17) **Semicolons.** Every statement in ALGOL ends with a semicolon unless it is followed by **end.** Every PL/I statement is followed by a semicolon.

There are hundreds of other differences between the various algebraic languages, but these are the basic ones which are required to be able to read published algorithms in FORTRAN, ALGOL, BASIC, and PL/I. Most such algorithms, with a few notorious exceptions, are presented in such a way as to use only the rules described above. The reader whose appetite has been stimulated by the possibilities of algebraic languages might do well to supplement his small system knowledge by renting a small amount of time (perhaps $100 worth) on a large system and trying out various features of FORTRAN, PL/I, and the like. This is, of course, in addition to the use of cross assemblers and cross compilers, which still require large systems to produce small system object code.■

Circle 227 on inquiry card.

SUPERKIM

MICROPRODUCTS HAS THEM

MICROPRODUCTS announces Superkim, its new singleboard control computer for commercial, industrial and hobbyist uses Superkim can accommodate 4K RAM and 16K EPROM onboard

This super controller can be applied to any situation where intelligent control is desirable: such as any manufacturing or production line process where automation is possible or to automatic tool operation or to real time data collection The board has a large prototype area suitable for mounting Analog to Digital, Digital to Analog converters, relays and other interface devices

Superkim is totally compatible with all KIM-1* software and most KIM-1 and APPLE II hardware interfaces It has TTY, RS232 and audio cassette interfaces as well as an onboard 5 volt, 3 amp and 12 volt regulator, rectifier diodes and filter capacitor. Superkim has eight latched priority interrupts which are individually resettable under software control. This feature is absolutely essential for implementing highly useful real time systems

MICROPRODUCTS can supply a hardware interface and a software downloading routine for the APPLE II and a firmware receiver routine, located in a 2716 EPROM, for installation in the Superkim. This greatly facilitates software development for the Superkim because of the powerful **MICROPRODUCTS**/APPLE II Co-resident assembler, the **MICROPRODUCTS**/APPLE II to PR-40 interface and the large memory available in the APPLE II. The software can be instantaneously transmitted from your APPLE II software development system to your Superkim RAM for instant checkout and use

*KIM-1 is a product of MOS Technology

APPLE II

DEVICES BY **MICROPRODUCTS**

MICROPRODUCTS/APPLE II PR-40 and Centronics Interface	49.95
MICROPRODUCTS/APPLE II 8-bit parallel output port card	44.95
MICROPRODUCTS/APPLE II co-resident assembler on cassette	19.95
MICROPRODUCTS/APPLE II co-resident assembler on floppy disc	25.00
MICROPRODUCTS/APPLE II EPROM programmer	99.95
MICROPRODUCTS/APPLE II 5 volt EPROM socket adapter	14.95
MICROPRODUCTS/APPLE II modem	129.95
MICROPRODUCTS Superkim single board control computer	395.00
MICROPRODUCTS 16 character alphanumeric LCD	795.00
MICROPRODUCTS/KIM-1 improved keyboard with double sided, gold plated pc board	30.00

Dealer inquiries invited
California residents add 6% sales tax

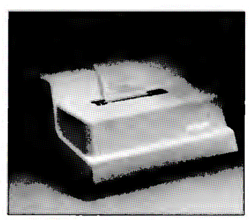
MICROPRODUCTS
2107 Artesia Blvd. / Redondo Beach / CA 90278
(213) 374-1673

Why Pay More?

Why pay for more printer than you need? Our series 40 printers offer more features for less bucks than any other commercial quality printer on the market today. A complete stand-alone 40 column impact dot matrix printer with a 64 character ASCII set. Includes power supply, casework and interface electronics. Single quantity price for the parallel ASCII interface model is $425. Serial RS232/current loop interface models start at $575. OEM discounts available.

For more information write to:

MPI 2099 West 2200 South, Salt Lake City, Utah 84119 or call (801) 973-6053.



mpi

Circle 265 on inquiry card.