MINIMUM HARDWARE AND SOFTWARE REQUIREMENTS

256K of System Memory
Dos 2.00 or Higher
One Floppy Drive

ABOUT THE PROGRAMMER

The Pocket Programmer for the IBM PC and compatibles from Intronics, Inc. is easy to use and install. The Pocket Programmer uses the printer port (Bi-Directional or not) of your PC and can use LPT1 or LPT2. The Pocket Programmer is under software control at all times and allows you the ability to change the parameters to suit your needs.

There are No personality modules or switch settings to change for different Eprom's. An external power pack provides all power for the Pocket Programmer.

Please use the following precautions when using the Pocket Programmer:

1. Never insert an Eprom in the ZIF socket until you have started the program and have selected the your Eprom type.

2. NOT all Eprom's with the same # are the same. Be sure of the type and of the Vpp Voltage you are programming. Please refer to #9 for further information.

3. Make sure the ZIF socket is Open before inserting an Eprom. If you insert an Eprom in then try to Open the ZIF socket, you WILL break the socket.

WHERE ARE THE INSTRUCTIONS ??????

The instructions for the pocket programmer are on disk. This saves cost which saves you money (just like Taiwanes do it). There is a file called EPROMDOC.TXT. You can print a copy of the documentation on your printer by typing " PRINTER " and pressing the Enter key. There is a total of nine pages of text. EpromDoc.txt is a Ascii file so you can list it on your screen or use your word processor to look at it.

THE POCKET PROGRAMMER 90 DAY WARRANTY

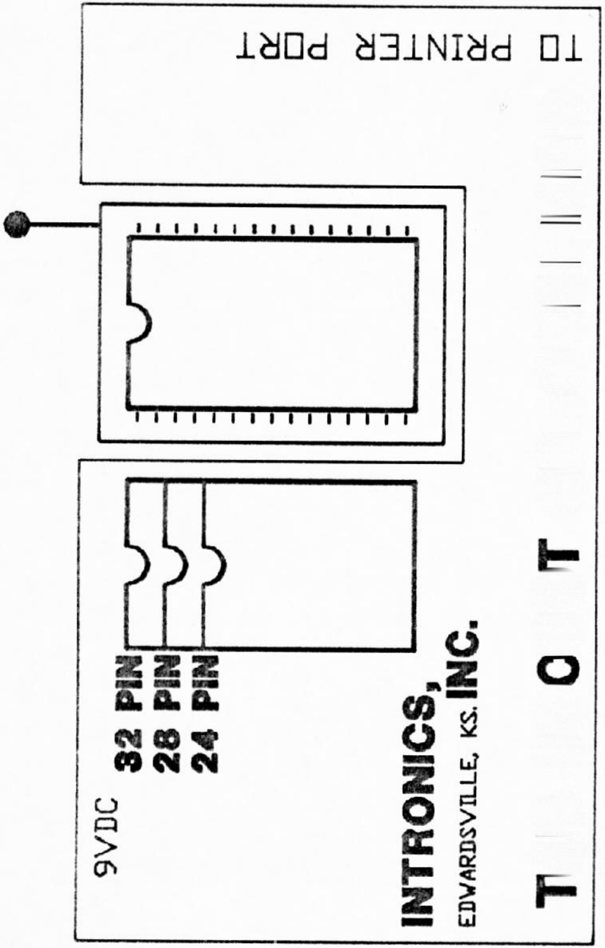
The Pocket Programmer comes with a 90 day LIMITED WARRANTY from the time of purchase.

INTRONICS, INC. shall have no liability or responsibility to the customer or any other person or entity with respect to any liability loss or damage caused or alleged to be caused directly or indirectly by "equipment" or "software" sold or furnished by INTRONICS, INC.

To get your Pocket Programmer repaired under warranty, send it postage-paid with letter explaining problem and proof of purchase to:

Intronics, Inc.
Box 13723
612 Newton St.
Edwardsville, Ks 66113
422-2094

Non-Warranty Repair charge is $35.00.

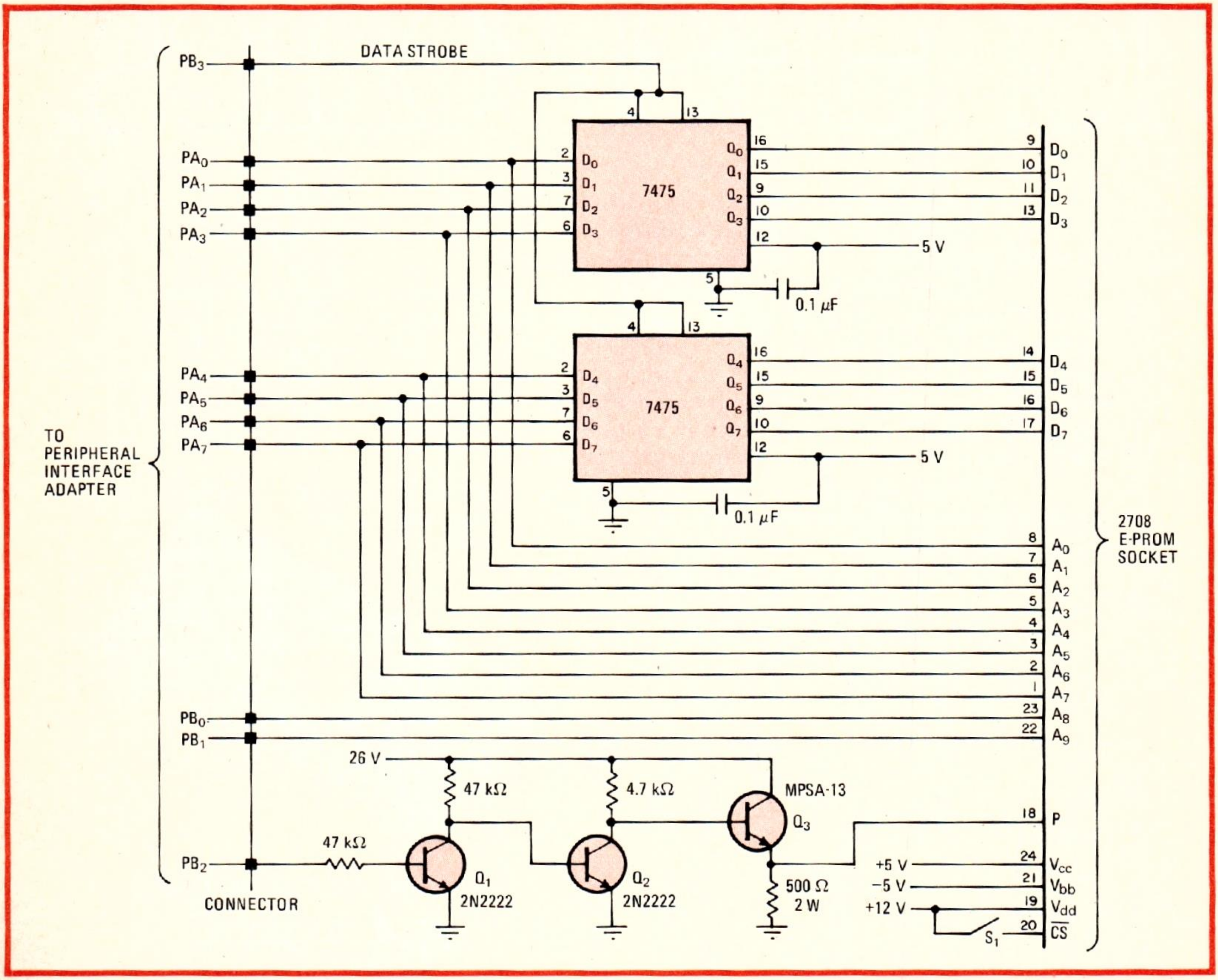


Low-cost interface automates E-PROM loading

by Henry Jan Stec
Panama Canal Co., Panama

This on-board interface and the accompanying routine equip a microprocessor-based system with the ability to program erasable programmable read-only memories. A specific circuit realization and program for the Motorola MEK6800D2 evaluation kit are presented here to illustrate the simplicity of this low-cost scheme, but in general any microprocessor system can be adapted for such a task.

The MEK6800D2 kit accepts two 2708 8-K E-PROMs that may be programmed with data stored in the microprocessor system's random-access memories. These 1-K-by-8-bit E-PROMs are most easily programmed in two operations that load half the memory at a time with upper and lower blocks of 512 bytes each. The loading is done with two quad latches, three transistors, and the



Loading blocks. Two-chip three-transistor interface and short software routine for MEK6800D2 evaluation kit simplify writing into an E-PROM. Under software control, the E-PROM is quickly programmed in 512-byte blocks with data transferred from random-access memory.

MEK6800D2 E-PROM PROGRAMMING ROUTINE

Location	Op code	Operand	Mnemonic	Comments
00	8E	A04F	LDS	
03	86	FF	LDAA	set up PIA's input/output lines
05	B7	8004	STAA DDRA	
08	B7	8006	STAA DDRB	
0B	86	04	LDAA	
0D	B7	8005	STAA CRA	
10	B7	8007	STAA CRB	
13	86	64	LDAA	number of loops = 100
15	B7	A041	STAA C1	
18	CE	03FF	LDX	
1B	8C	FFFF	CPX	end of current loop?
1E	27	4E	BEQ	to 6E
20	FF	A042	STX C2, C3	
23	FF	A044	STX C4, C5	
26	7A	A042	DEC C2	
29	7A	A042	DEC C2	
2C	2C	07	BGE	to 35 – does highest E-PROM address point to lower E-PROM?
2E	FE	A044	LDX C2, C3	
31	A6	00	LDAA	
33	20	02	BRA	to 37
35	86	FF	LDAA	
37	B7	8004	STAA ORA	output accumulator to PIA
3A	86	08	LDAA	
3C	B7	8006	STAA ORB	strobe latches
3F	7F	8006	CLR ORB	
42	FE	A044	LDX C4, C5	
45	B6	A045	LDAA C5	
48	B7	8004	STAA	output address bits 0 through 7
4B	B6	A044	LDAA C4	
4E	B7	8006	STAA ORB	output address bits 8 and 9
51	01		NOP	delay for address setup
52	01		NOP	
53	01		NOP	
54	01		NOP	
55	01		NOP	
56	86	04	LDAA	raise program pulse 1 ms
58	BB	8006	ADDA ORB	
5B	B7	8006	STAA ORB	
5E	C6	65	LDAB	pulse length adjust
60	5A		DECB	
61	26	FD	BNE	to 60
63	86	FC	LDAA	lower program pulse
65	BB	8006	ADDA ORB	
68	B7	8006	STAA ORB	
6B	09		DEX	
6C	20	AD	BRA	to 1B
6E	7A	A041	DEC C1	
71	26	A5	BNE	to 18
73	3F		SWI	stop

program shown, which initially resides in system RAM.

The program is loaded into the microprocessor system by its own bootstraps when the power is turned on. Because the program is short, there is plenty of room for many other utility programs in even a small (8-K) utility E-PROM.

Under software control, data to be written into the desired half of the 2708 E-PROM is passed from each RAM location to the peripheral-interface adapter's PA_i lines. For each location, the PIA strobes the data into the 7475 quad latches and the D_i bus. The program then sends the desired byte address through the PIA to the E-PROM's A_i bus and applies a programming pulse (P) via transistors Q_1 through Q_3.

In this way, the lower half of the E-PROM is programmed. Logic 1s are then applied to the quad latches so as not to disturb any data previously stored in the upper half.

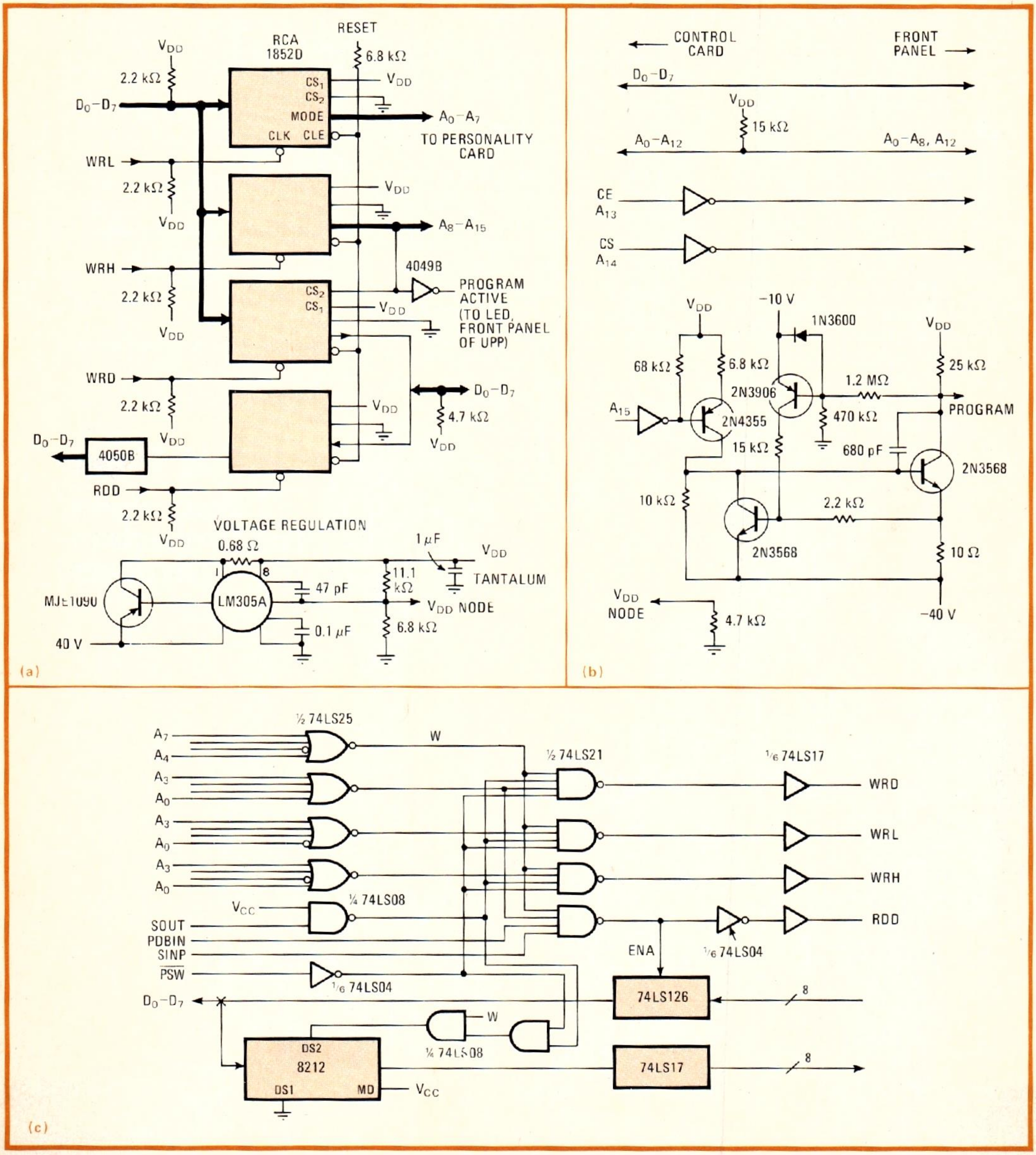
The actual mechanics of programming is simple. After the E-PROM is placed in its socket, the system is turned on and a system-reset pulse applied. The 26-volt supply required for E-PROM pulsing is then activated. Switch S_1 is closed and the program is executed, whereupon S_1 is opened and the 26-V supply turned off.

To program the upper half of the E-PROM, location 2C of the program must be modified so that it contains 2B and location 30, containing 42, is added to the listing. The time required to program an E-PROM is 2 minutes. ☐

Software-based controller [illegible]mplifies PROM programmer

by R. F. Hobson
Simon Fraser University, Burnaby, British Columbia, Canada

While Intel's popular Universal PROM Programmer (UPP) works effectively in its intended capacity as a system development tool, it has two major drawbacks. First, the so-called personality cards that are required for manually programming each type of programmable read-only memory are expensive and much too complicated to build. Second, it is restricted to Intel PROMs, so that the newer complementary-MOS erasable-PROM



Burning softly. Complexity of control board (a) and personality cards (b) in universal PROM programmer are reduced if software-based controller leads system's host computer through various read/write burn-in phases. Personality card is shown for the IM6604 PROM. S-100 bus interface (c) units UPP to 8080 host processor. Small program (table) guides 8080 through write-and-verify sequence.

WRITE-AND-VERIFY SEQUENCE: IM6604 E-PROM

Write/Verify Routine

```
        Statement                    Comment

COUNT:  DS 1
WRITE:  PUSH  B
        PUSH  PSW
        CALL  DOUT    ;   LATCH DATA BYTE.
        XRA   A
        STA   COUNT   ;   INITIALIZE COUNT.
        POP   PSW
        CALL  ADDRL   ;   LATCH ADDRESS.
        CALL  INFO0   ;   INITIALIZE CRT.
WLP:    CALL  PULSE   ;   SEND A PROG PULSE.
                      ;   VERIFY
        MOV   B,A     ;   SAVE DATA BYTE.
        LDA   COUNT
        INR   A       ;   BUMP UP COUNT.
        STA   COUNT
        CALL  READ    ;   GET CELL CONTENTS.
        CMP   B       ;   COMPARE WITH DATA.
     JZ BURN          ;   EXIT IF VERIFIED.
        CALL  INFO1   ;   UPDATE CRT.
        LDA   COUNT
        CPI   030H    ;   PULSE LIMIT . . .
        MOV   A,B
     JC WLP           ;   EXCEEDED?
        STC           ;   IF SO EXIT.
        POP   B
     RET
BURN:   ......        ;   LAST WRITE OK,
                      ;   BURN AND EXIT.
```

Pulse Routine

```
        Statement                    Comment

PROG    EQU  80H      ;   CONTROL BIT.
PULSE:  PUSH PSW
        PUSH B
        MOV    A,D    ;   ADDR/CTL BYTE.
        ORI  PROG     ;   SET PROGRAM BIT.
        MVI  B, OEH   ;   SET 14MS COUNT.
        CALL HOUT     ;   START PROG PULSE.
        CALL MSDLAY   ;   HOLD IT.
        XRI  PROG     ;   CLEAR PULSE BIT.
        CALL HOUT     ;   RESET PROG PULSE.
        MVI  B, 07H   ;   7MS COUNT.
        CALL MSDLAY   ;   WAIT (2/3 DC).
        POP  B
        POP  PSW
     RET
```

chips cannot be programmed. The personality cards can be simplified and the UPP peripheral device made more versatile, however, if a software-based controller guides the system's host computer through the various read/write phases required to program and verify the contents of PROMs.

The basic UPP interface has eight data-input and eight data-output lines, along with read-data, read-acknowledge, and read-status ports. Also included is a write-data line, a write high-address and write low-address line, and an interrupt line. A pulsed control signal required for programming each PROM location is handled via a 4-bit 4040 microprocessor on a control card in the UPP.

The best way to simplify such a peripheral is to have the host computer provide the timing, control, and logic necessary for programming, reading, and verifying the contents of PROMs and E-PROMs. The complexity of the UPP's control card is then reduced to that shown in (a) of the figure, where the 4040-based setup is replaced by four C-MOS I/O chips (RCA 1852D).

The host computer consequently sees one input port and three output ports. The input port is used for returning the contents of a selected memory word. Two of the output ports are used for latching address and control information; the remaining output line is used for a data-output latch.

Of the 16 PROM-address lines available, 12 are used to address up to 4 kilobytes of memory. The remaining four lines can be used for program and chip control. Consider the personality card of the 512-by-8-bit IM6604 PROM, for example (b in figure). There, line A_{15} is used for a program pulse enable, A_{14} and A_{13} are used for chip select and chip enable, respectively, while line A_{12} is used for a strobe pulse. The popular 27XX E-PROM series would require only two control lines. Because the 27XX chips are powered by 5 volts, while C-MOS devices require 10 V for programming, the 27XX's personality card would interface to the host computer through open-collector devices.

In general, then, a personality card will consist of a bidirectional data bus, the required number of address lines, and a pulser circuit. It is thus used mainly to route the bus lines to the proper front-panel pin positions on the UPP. The pulse circuit must be designed to be reset by the UPP's front-panel reset button. This can be accomplished by connecting the reset line to the 1852D's CLE inputs. For completeness, the program control line (on the control card) is also connected to the program LED on the front panel.

A typical S-100 bus interface for the modified UPP is shown in (c). I/O ports 32, 33, and 34 have been decoded for a data strobe (read), write low-address, and write high-address, respectively. Interface software must include a timing subroutine and program pulse and verification routines particular to the PROM that is programmed.

As for the programming required, the sequence in the table outlines the steps necessary for the write-and-verify operation in the IM6604. The program is written for the host 8080A microprocessor. ☐

Engineer's notebook is a regular feature in *Electronics*. We invite readers to submit original design shortcuts, calculation aids, measurement and test techniques, and other ideas for saving engineering time or cost. We'll pay $50 for each item published.

Teleprin[illegible] option uni[illegible] PR[illegible]M programmer to MC6800

by J. Padmanabhan and M. S. Swaminathan
O/E/N India Ltd., Mulanthuruthy, Cochin, India

Offering a simple way to unite the often-used MC6800 microprocessor system to the popular Prolog Series 90 PROM programmer, this modified teleprinter option and appropriate software provide an economical way to transfer large blocks of data rapidly. Here, what is essentially the modified 9102 teleprinter interface and interactive software for supervising data flow between microprocessor and PROM programmer do the basic job of the more expensive 9104 computer interface normally used to do the task.

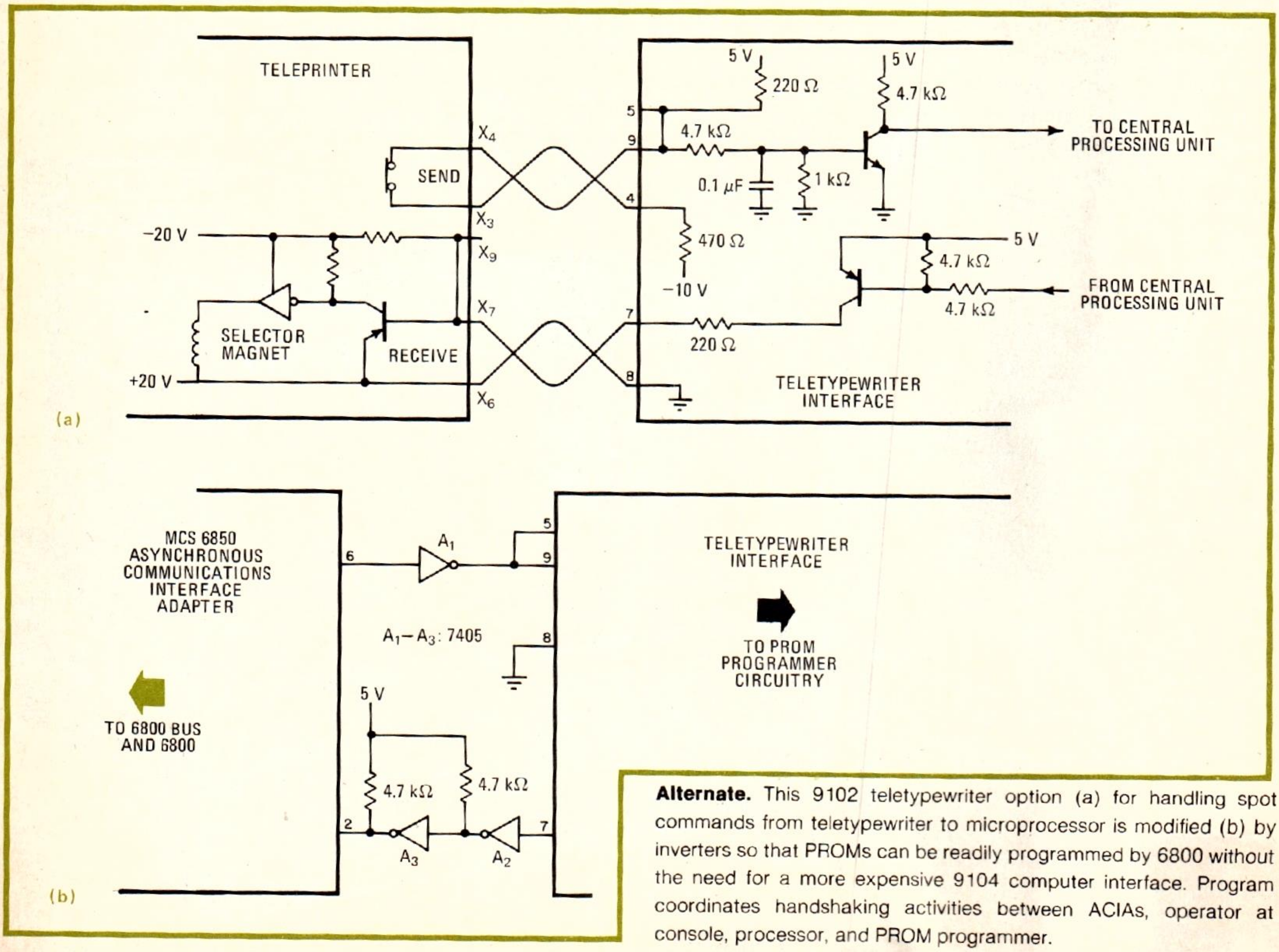
The 9102 option was designed to link the computer bus to a teleprinter through a 20-milliampere current loop, as shown in (a). In cases where it is used, it will offer an operator at a teleprinter console a convenient way to enter spot commands or data into a microprocessor or other device. But manual loading of 2 kilobytes of data or program, for example, is impractical because of the time required and the errors that may be generated because of human interaction.

On the other hand, the transferring of data stored in memory to the programmable read-only memories (PROMs) by means of the PROM programmer's "duplicate" facility can only be done if the master PROMs are of the same word size as the PROMs to be programmed. These problems are overcome by transferring the data from the 6800 to the programmer through the teletypewriter interface via an asynchronous communications interface adapter. The data transfer is controlled by the user at a console with the aid of suitable software.

Adding three open-collector inverters between the ACIA and the teletypewriter interface going to the PROM programmer (b) constitutes the hardware modification. Note that the teletypewriter console, not shown, would be connected to the 6800 bus through a separate ACIA and unmodified teletypewriter interface, also not shown.

Inverter A_1 connects the normally high ACIA output to a logic 0, which simulates the active-high teletypewriter signal that the PROM programmer requires. Note that the interconnecting loop now operates on voltage levels, not on a current mode. A_2 and A_3, acting in a similar role, provide buffering action in the reverse direction.

The software operates in an interactive mode and is transparent to both the operator and the PROM programmer. It can handle any console device besides the teleprinter (such as a video terminal or hex keyboard) operating at any baud rate and may be easily modified to accommodate any PROM size. As shown in the figure,



Alternate. This 9102 teletypewriter option (a) for handling spot commands from teletypewriter to microprocessor is modified (b) by inverters so that PROMs can be readily programmed by 6800 without the need for a more expensive 9104 computer interface. Program coordinates handshaking activities between ACIAs, operator at console, processor, and PROM programmer.

6800 ROUTINE: PROM PROGRAMMER-TO-PROCESSOR TELETYPEWRITER INTERFACE

LOCATION	MNEMONIC		COMMENTS
0001	CNTRG1	EQU 00C7	CONTROL REGISTER OF ACIA1
0002	CNTRG2	EQU 00C6	CONTROL REGISTER OF ACIA2
0003	DATRG1	EQU 0087	DATA REGISTER OF ACIA1
0004	DATRG2	EQU 0086	DATA REGISTER OF ACIA2
0005	START	LDA A#03	ACIA INITIALIZATION
0006		STA A CNTRG2	
0007		STA A CNTRG1	
0008		LDA A#05	
0009		STA A CNTRG2	
0010		STA A CNTRG1	
0011		LDX ADDBUF	STARTING ADDRESS IN X
0012		LDA B#01	
0013		JSR 1NECHO	INPUT*
0014		LDA B#02	
0015		JSR PGRTTY	OUTPUT CR, LF
0016		LDA B#01	
0017		JSR INECHO	INPUT P
0018		LDA B#02	
0019		JSR PGRTTY	OUTPUT CR, LF
0020		LDA B#06	
0021		JSR INECHO	INPUT ADDRESS FIELD
0022		LDA B#07	
0023		JSR PGRTTY	OUTPUT CR, LF, LF, ADD, SPACE
0024	NXTDTA	LDA B#02	
0025		JSR MPPGR	DATA TO PROGRAMMER
0026		LDA B#03	
0027		JSR PGRTTY	OUTPUT CR, LF & ONE CHARACTER
0028		COM A#0A	CHECK FOR LF
0029		BEQ ERROR	TO ERROR HANDLING
0030		COM A#0D	CHECK CR
0031		BEQ END	TO END OF PROGRAMMING
0032		LDA B#02	TWO MORE DIGITS OF ADDRESS
0033		JSR PGRTTY	
0034		BRA NXTDTA	GO FOR NEXT DATA
0035	END	LDA B#04	
0036		JSR PGRTTY	OUTPUT LF, /, CR, LF
0037		BRA RETURN	RETURN FROM PROGRAM
0038	ERROR	LDA B#03	ERROR HANDLING
0039		JSR PGRTTY	OUTPUT TWO DIGIT ADDRESS & SPACE
0040	RETURN	RTS	END OF PROGRAMMING

the program handles data transfer between the user and PROM programmer and vice versa, from the microprocessor to the PROM programmer and between the two ACIA ports.

In transferring information from the console to the PROM, the user initializes the program and defines the mode of operation and the address field. For full-duplex operation, these data characters have to be accepted from the user and transmitted to the PROM programmer, whereupon a data-valid signal will be sent back to the console. Because the number of characters for a given word of information varies, a loop-counter subroutine must be contained in the program.

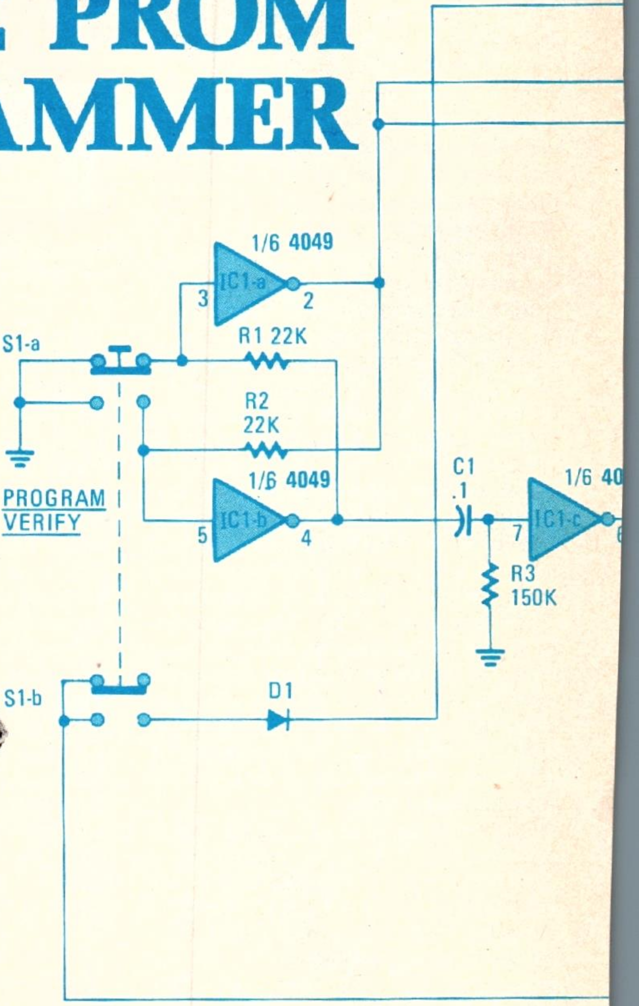
When the PROM programmer is ready to accept the next data, it transmits CR, LF, and the address of the next location. When transferring information from the microprocessor to the programmer, data is transferred as required, provided the operating mode and the address field has been previously defined. The data is available in computer memory in binary form, and the processor transmits it to the programmer as two hexadecimal ASCII characters for each byte sent. A data-valid command is then sent back to the user terminal at the completion of a data-transfer cycle.

The program identifies any error condition as well as the completion of programming by checking the third character transmitted to the console by the PROM programmer. If there are data errors, the programmer transmits CR, LF, LF, and the address of the current location (instead of CR, LF, and the address, which is the normal condition). The system then waits for user intervention. At the completion of programming (all good data), the programmer sends back CR, LF, CR, LF,/, CR, LF.

Note that the subroutines INECHO, PGRTTY, and MPPGR are all well-known teleprinter handling routines. In the interest of space, they are not listed here. ☐

Engineer's notebook is a regular feature in *Electronics*. We invite readers to submit original design shortcuts, calculation aids, measurement and test techniques, and other ideas for saving engineering time or cost. We'll pay $50 for each item published.

INEXPENSIVE PROM PROGRAMMER



T.E. LeVERE

Here's a simple circuit that will enable you to program the popular 74S-series of PROM's right on your own workbench.

MORE AND MORE ELECTRONIC DEVICES are being "programmed"—whether or not they contain a microprocessor. In many instances, as is the case with household appliances or games, those programs are not entered by hand or from a magnetic medium such as disk or tape, but are contained in integrated circuits called ROM's (for *R*ead *O*nly *M*emory).

The ROM is a type of memory device that is permanently programmed—while you can read the information it contains as many times as you like, you cannot change it. That's why it's called "read-only."

In general, there are two ways that ROM's are programmed. The first of these is called *mask programming* and is actually a part of the IC manufacturing process. That type of programming is useful only when large quantities of identical ROM's are to be produced.

The other type of ROM—and the one we'll discuss here—is actually known as a PROM (*P*rogrammable *R*ead *O*nly *M*emory). That type of memory IC is supplied with all its bits at either a high or low logic level. In programming it, you change those logic states to meet your requirements. Typically, that programming is accomplished by "blowing" (burning out) internal titanium-tungsten (Ti-W) fuse links, each one representing a bit. That is done by applying a specific excess voltage to the power input of the PROM IC after selecting those bits that are to be a logic-high, and those that are to

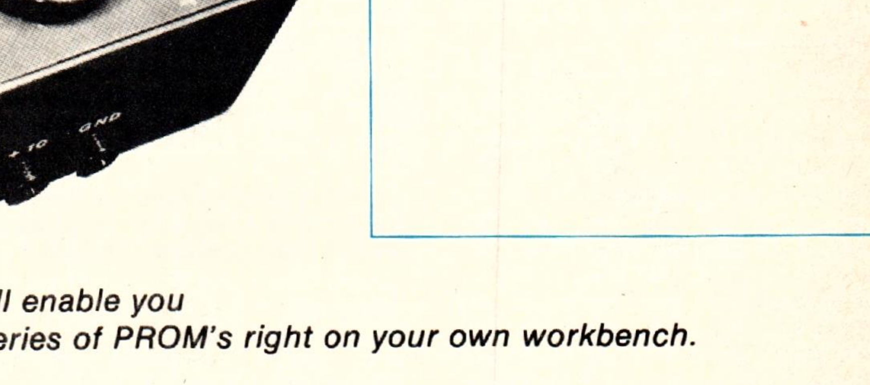


FIG. 1—TIMING SEQUENCE for programming a word of memory into the PROM.

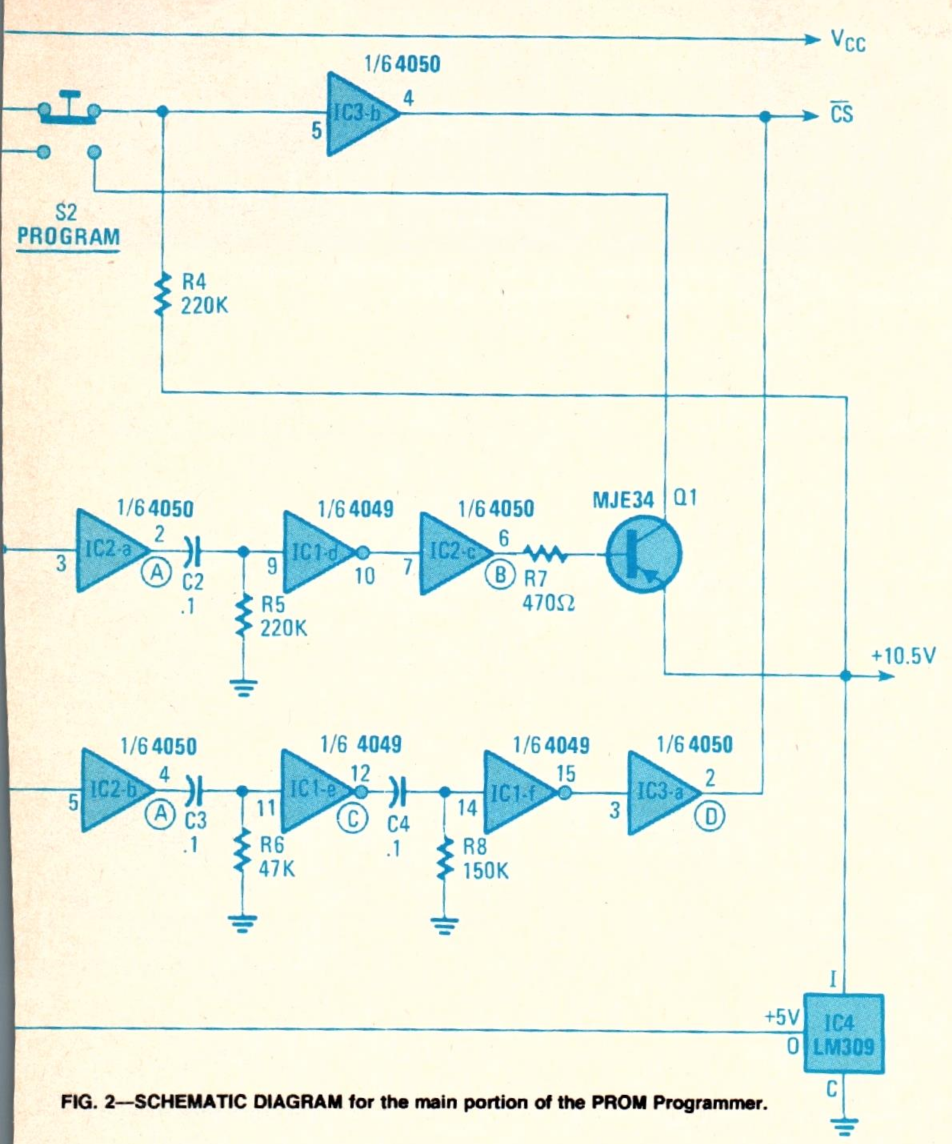


FIG. 2—SCHEMATIC DIAGRAM for the main portion of the PROM Programmer.

PARTS LIST—PROGRAMMER

Resistors, 1/4 watt, 5%
R1, R2—22,000 ohms
R3, R8—150,000 ohms
R4, R5—220,000 ohms
R6—47,000 ohms
R7—470 ohms
R9-R13—220 ohms
R14-R21—3900 ohms
Capacitors
C1-C4—0.1 μF, ceramic disc
Semiconductors
D1—1N914 or 1N4148
IC1—4049 CMOS hex inverter (RCA)
IC2, IC3—4050 CMOS hex buffer/driver (RCA)
IC4—LM309 or LM340 five-volt regulator (National)
Q1—MTE34 PNP power transistor, (Motorola; or Radio Shack 276-2027 or equivalent)
LED1-LED13—jumbo red LED (Radio Shack 276-041 or equivalent)
S1—DPDT pushbutton switch
S2—SPDT pushbutton switch
S3-S7—SPDT toggle switch
S8—two-pole, nine-position rotary switch
Miscellaneous: perforated construction board, IC sockets (including one for PROM), 10.5-volt power supply, enclosure, wire, etc.

be a logic-low.

While in theory it sounds simple, a good deal of care must be taken when "burning" a PROM. If the voltage is too low, the fuse links will not be destroyed; if it's too high, or applied for too long a time, undesirable side effects may result. If just *one* bit is programmed incorrectly, the entire PROM is ruined.

The circuit presented here for programming fusible-link PROM's is a simple one, involving only three IC's, one transistor, and a voltage regulator, together with the switches necessary to set up the program. It is intended for use with Texas Instruments' 74S-series of programmable memories.

The programming process

There were several reasons for selecting that family of PROM's to work with. First, they are readily available (check the advertisers in the back pages of **Radio-Electronics**). Second, they are relatively inexpensive—a few dollars for a 32-word type. Third, they require only a single five-volt supply and are available with either open-collector or three-state output configurations, and are rather fast (under 30 nanoseconds, generally). Finally, while several different memory configurations are available in that series, each type of PROM is programmed in the same manner, which means that the circuit can be used for a number of applications.

The circuit design is intended to meet the rather tight timing sequence needed for programming that family of PROM's. Basically, programming involves selecting the word of memory to be programmed by setting the address lines, grounding the bits that are to be programmed, and holding the bits that *are not* to be programmed at the supply voltage (five volts). The actual programming takes place when the supply voltage is briefly stepped-up to 10.5 volts and the IC is enabled by means of the chip-select $(\overline{CS})$ line. Details of that sequence are shown in Fig. 1. The upper line indicates the voltage applied to the V_{CC} pin, while the lower one shows the voltage at the $\overline{CS}$ pin of the IC.

Initially, the supply voltage (five volts) is applied to both the V_{CC} and $\overline{CS}$ pins, which disables the IC. The programming sequence is started by raising the voltage at the V_{CC} pin to 10.5 volts and then, after 500 microseconds (0.5 mS)—TI suggests a range of from 10 μS to 1 mS—$\overline{CS}$ is brought to ground and the memory enabled for 1 mS. After that, $\overline{CS}$ is returned to five volts and the 10.5 volts at the V_{CC} pin is reduced to five volts. At this point it is possible to ground the $\overline{CS}$ pin to verify that the programming took place as planned.

Texas Instruments makes several recommendations concerning the programming of the 74S– series. First, since the process involves burning out the Ti-W fuse links, it is recommended that V_{CC} be removed between programming sequences to avoid overheating the chip. Second, it is recommended that the 10.5-volt programming pulse have a duty cycle of no more than 30 percent of the entire programming cycle. With an automated programmer, that could be a problem but since our circuit is manually operated, it is nothing to be concerned about.

Circuit description

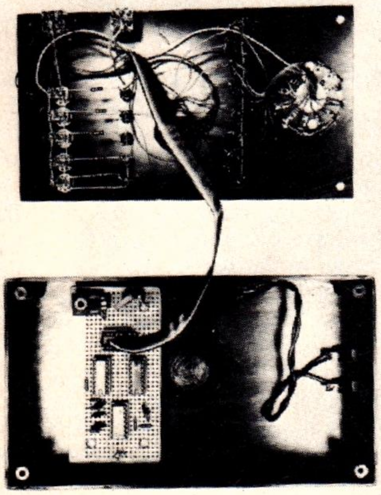
The main portion of the PROM programmer circuit is shown in Fig. 2. It uses two 4050 CMOS hex buffer/drivers, a 4049 CMOS hex inverter, one PNP power transistor and a five-volt regulator, such as an LM309 or LM340.

A power supply that can provide 10.5 volts DC (Fig. 3) is required to open the fusible links of the PROM. The regulator reduces that to 5 volts, which is needed at several points in the circuit.

Two switches are used in this section. The first, S1, supplies power to the PROM and/or initiates the programming sequence while S2 supplies 10.5 volts to the V_{CC} pin when it is required.

Programming is accomplished by depressing switch S2 and holding it down while switch S1 is pressed momentarily. Verification is obtained through the use of S1 alone. Details of that will be presented later.

In brief: the 4049 hex inverter is used to debounce switch S1 and to form several half-monostable circuits



INTERNAL view of the PROM programmer shows how the perfboard is mounted inside the case. The front panel is connected via ribbon cable.

PARTS LIST—POWER SUPPLY

Resistors, ½-watt, 10%
R22—1600 ohms
R23—220 ohms
Capacitors
C5—250 μF, 25 volts, electrolytic
C6—.1 μF, ceramic disc
C7—1 μF, tantalum
Semiconductors
BR1—bridge rectifier, 50 PIV
IC5—LM317T, adjustable voltage regulator
T1—18 volts, one amp
Miscellaneous: construction board, wire, solder, etc.

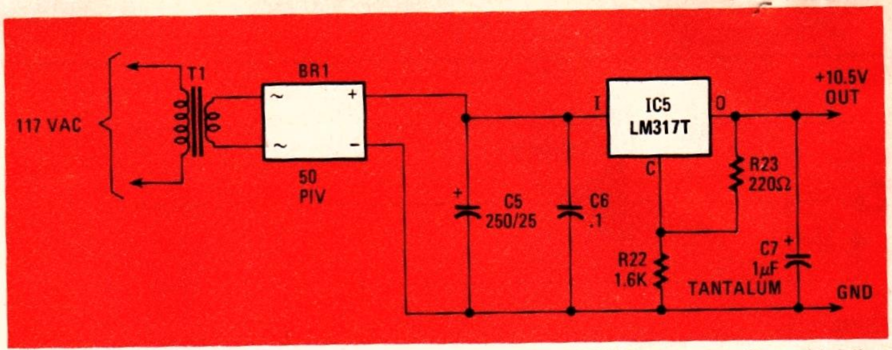


FIG. 3—SCHEMATIC DIAGRAM of the power supply for the PROM Programmer. It provides the 10.5 volts DC required to open the fusible links of the PROM.

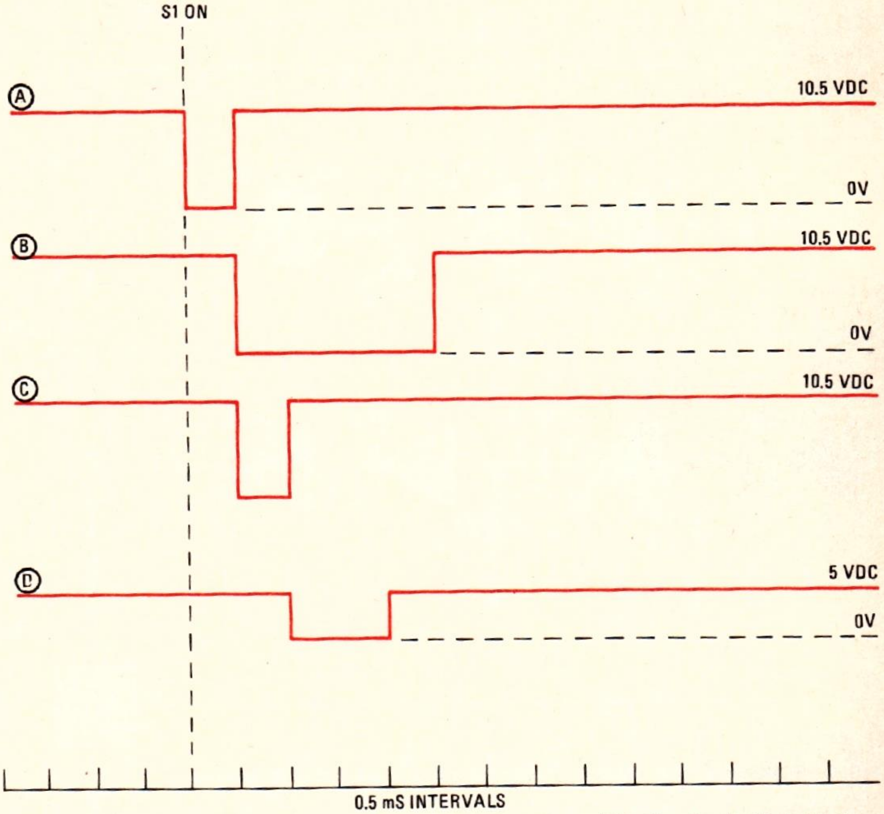


FIG. 4—TIMING DIAGRAM shows waveforms at specific points within the circuit. Circled letters correspond to circled letters in Fig. 2.

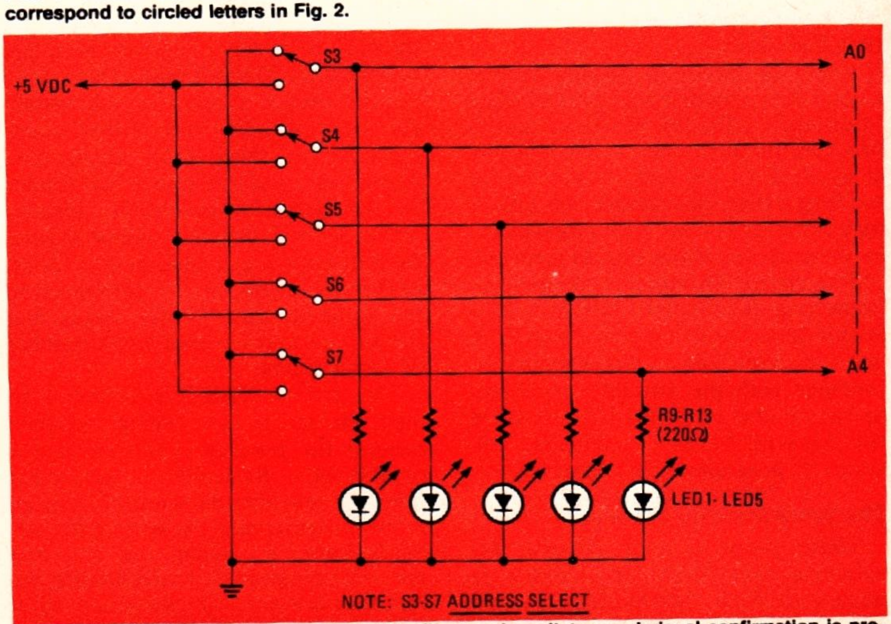


FIG. 5—PROM ADDRESSES are selected by five toggle switches and visual confirmation is provided by discrete LED's.

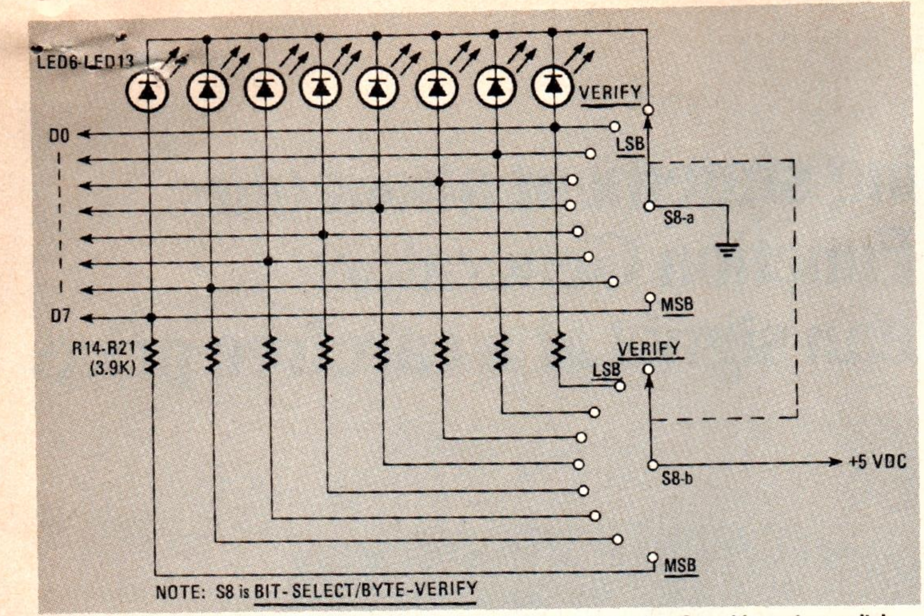


FIG. 6—A WORD IS PROGRAMMED sequentially one-bit at a time via a 9-position rotary switch.

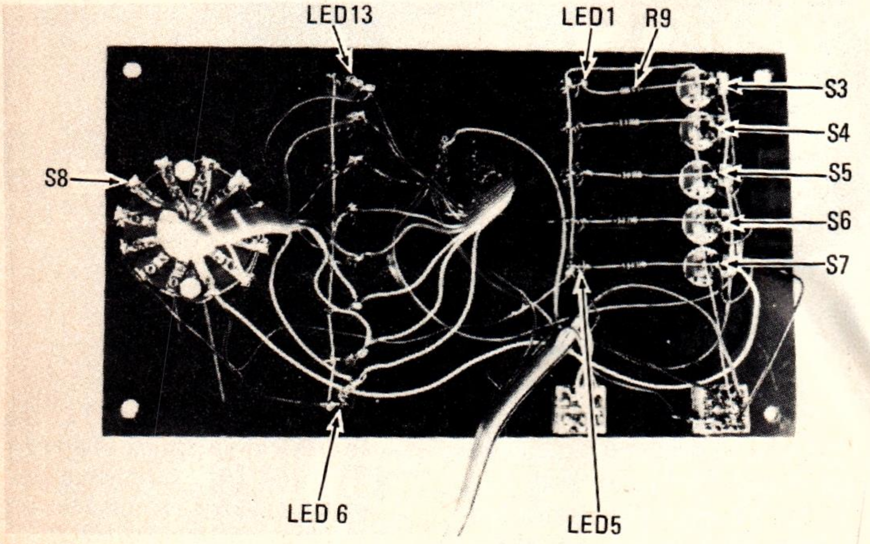


FIG. 7—PLACEMENT OF THE FRONT PANEL mounted components. Power switch S1 is located in the lower right corner and switch S2 is just to the left of S1. Connections to the PROM are made via a front-panel mounted DIP socket.

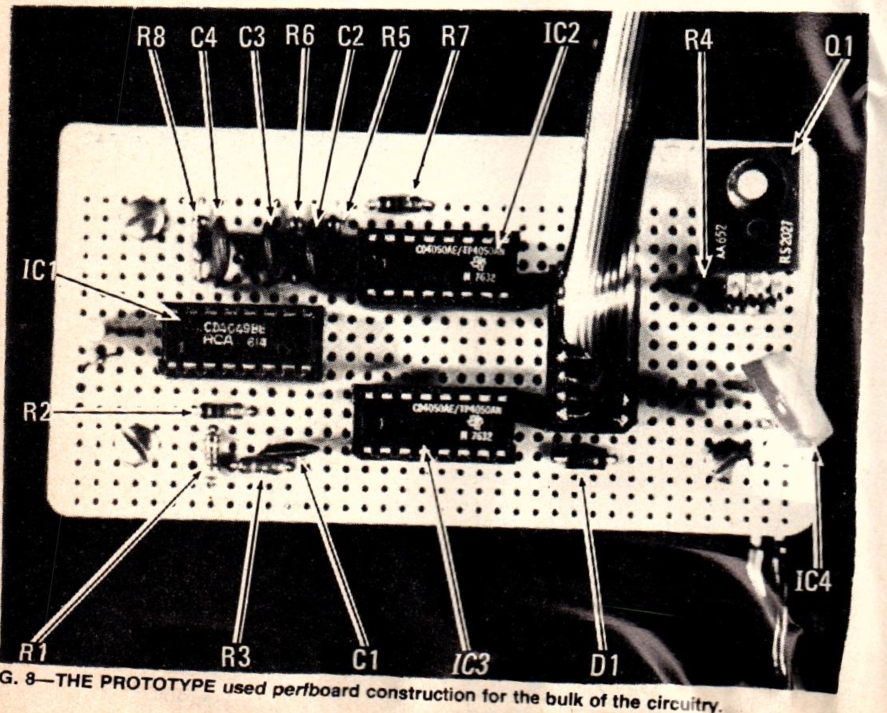


FIG. 8—THE PROTOTYPE used perfboard construction for the bulk of the circuitry.

used to time the programming sequence. One of the 4050 buffer/drivers is powered by 10.5 volts and acts as a waveform shaper for the signals from the 4049. The other is powered from the 5-volt regulator and is used as a level-translator to supply 5 volts to the V_{CC} and $\overline{CS}$ pins. The circled letters on the schematic (Fig. 2) correspond to those in Fig. 4, that shows the waveforms.

While the addresses and bit-patterns may be set using as crude a method as alligator clips, the switch circuits shown in Figs. 5 and 6 make programming much simpler.

Figure 5 illustrates an address-selector circuit using five SPDT switches. Each switch is tied to one address bit of the PROM and to an LED, that acts as a status indicator. The LED's allow you to make sure that the address you selected is correct—remember, programming one bit incorrectly will ruin the entire PROM!

To set up the word to be programmed, the circuit in Fig. 6 works quite well. Its major component is a double-pole, nine-position rotary switch. One pole of that switch is tied to the five-volt supply with eight of its positions going to the eight data outputs of the PROM through 3.9K resistors. The other pole of the switch is at ground potential, so that only one of the data outputs is grounded for programming purposes at any time.

The ninth switch position is used for program verification, and removes the data-output pins of the PROM from the 3.9K resistors and grounds the cathodes of the LED's. Any bit at a logic-1 will cause its associated LED to light.

With the switch wired in that manner, it is easiest to program from the most-significant bit to the least-significant bit and to verify the programming, in that order, afterwards. That is the switch is first moved to its most counterclockwise position, bits seven through zero are programmed as the switch is turned clockwise, and the switch is then turned to its final position for obtaining vertification of the programming.

Construction

Assembling the PROM programme is extremely simple and it may be bui on perforated construction board wit out difficulty. Component placeme is not critical. Refer to Figs. 7 and for component placement. A sm chassis box can be used to hold circuit board and a power supply, to mount the LED's and switches.

With two exceptions—both due human error—that setup has perfor faultlessly. It should work at least well for you.

Programmable source sets voltage of E-PROMs

by Ralph Tenny
George Goode & Associates, Dallas, Texas

Many erasable programmable read-only memories require varying voltages for different functions at pin 20. Intel's 2732 E-PROM, for example, multiplexes two functions on pin 20—the output enable and programming voltage input. It is desirable to have these inputs generated by a programmable source since the use of a relay to switch voltages is cumbersome, slowing down the circuit and adding a mechanical element to it. This programmable supply, controlled with two logic signals, provides an automatic selection of four voltages and has 0 volt as an off position.

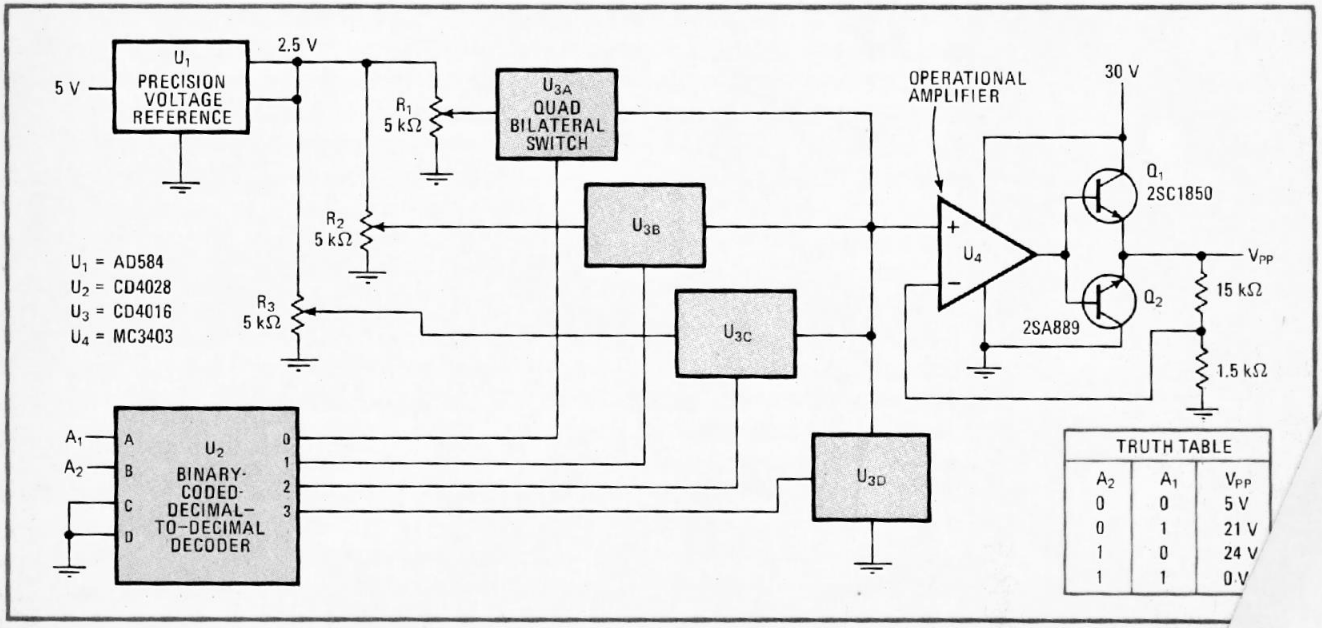
This logic-controlled programmable voltage source (see figure) is composed of three integrated circuits, a voltage reference, and a few discrete components and has a slew rate of around 1 V per microsecond. However, this slew rate is limited by the operational amplifier (U_4). In addition, a constant slew rate for both the positive and negative swings of the supply is maintained by transistor Q_2, which pulls the output voltage down rapidly.

The binary-coded–decimal-to-decimal decoder in the circuit takes two logic input signals and converts them into four output signals. These outputs enable the four sections of switch U_3 that gate the operational amplifier's (U_4) input. The op amp along with transistors Q_1 and Q_2 forms a voltage regulator whose output is 11 times the input reference voltage. This 2.5-V reference is provided by U_1 and is tapped by potentiometers R_1, R_2, and R_3, which in turn produce the reference voltages needed to generate the desired programming voltages.

All the voltages needed for pin 20 are produced by means of this supply (as shown in the truth table). As a result, all 5-V E-PROMs, including the I2732A and I2764, may be programmed with this circuit. The supply is driven by the B port pins of the peripheral interface adapter MC6821.

The PIA lines that drive inputs A_1 and A_2 are terminated with pull-up resistors. This configuration produces a reset input of 11 (to U_2) that forces the programming voltage to 0 V. The performance of this circuit is adequate for all E-PROMs except for Motorola's MC68766, which requires a fast 12-V/μs slew rate. ☐

Engineer's notebook is a regular feature in *Electronics*. We invite readers to submit original design shortcuts, calculation aids, measurement and test techniques, and other ideas for saving engineering time or cost. We'll pay $75 for each item published.



TRUTH TABLE

A_2	A_1	V_{PP}
0	0	5 V
0	1	21 V
1	0	24 V
1	1	0 V

Programmable supply. Three ICs, a voltage reference and a few discrete components form this logic-controlled programmab source with its approximately 1-V/μs slew rate. The supply is capable of providing four output voltages with 0 V as an off position.

NOTES ON PROGRAMMING PROCEDURE

The 8223 PROM specifications sheet recommends a 390-ohm burn resistor, as opposed to the 39-ohm value specified in "How To Program Read-Only Memories" (July 1975). The exact magnitude difference between the two values makes me suspicious of the article's procedure. —*Mark Coffman, Cincinatti, Ohio*

I have a question on one of the steps for programming a PROM (page 30). The note in step 1 states: "NEVER operate *S1* when *S2* is set to BURN." If this is correct, there is no way to blow the fuses.—*David Peterson, Edmonton, Alberta, Canada*

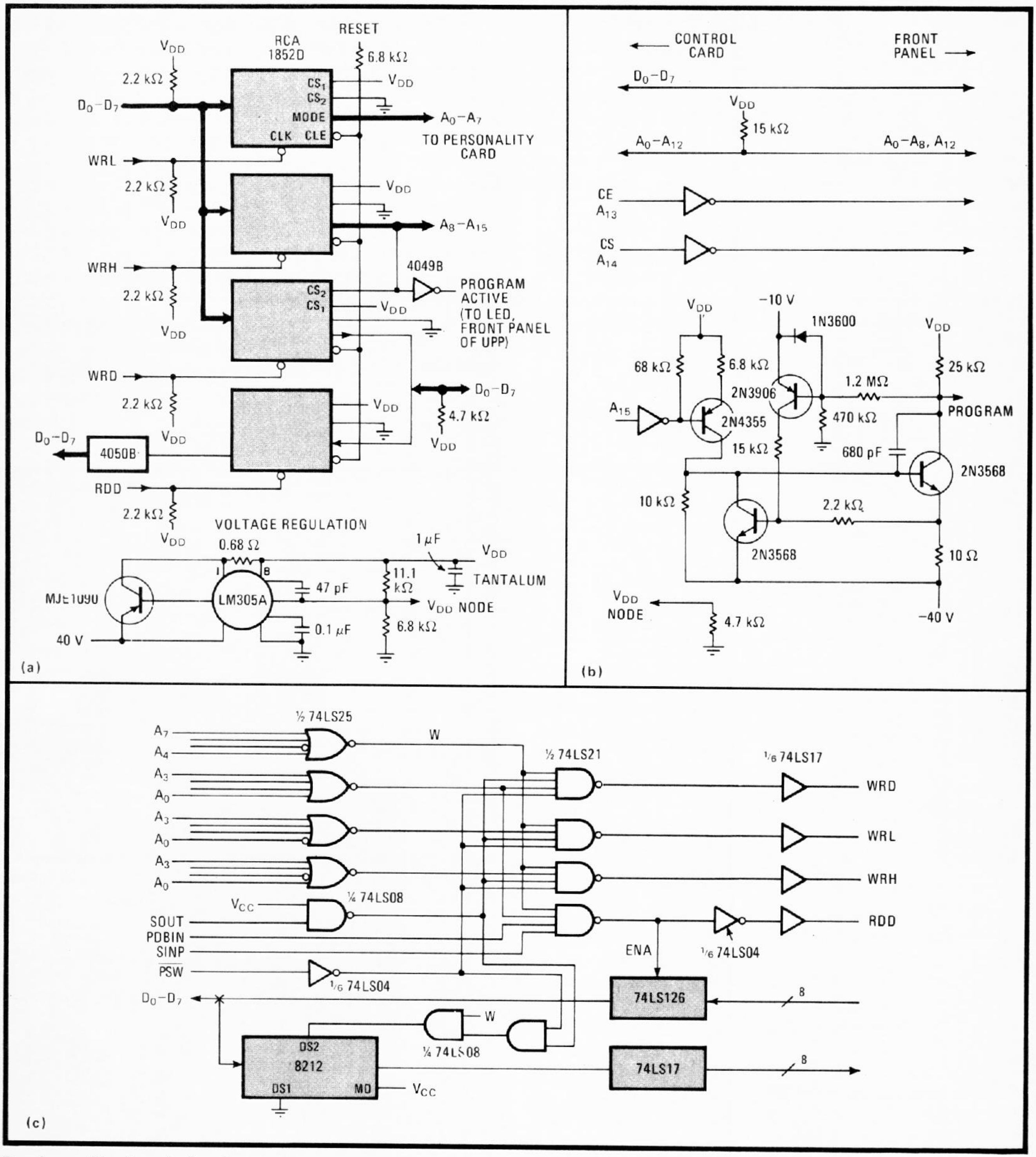
It has been our experience that most PROM's need a great deal more burn current than is possible to obtain using a 390-ohm resistor. We experimented with a number of PROM's and found that 39 ohms is the optimum burn resistor value. The 390-ohm value failed to give the desired results. (Perhaps the PROM's we had, from several sources, were manufacturer fallouts; if so, most of the PROM's our readers will be able to obtain are in this category.)

As for the note in step one of the programming procedure, it should read: "NEVER operate S1 when S2 is set to TEST."

Software— controller simplifies PROM programmer

by R. F. Hobson
Simon Fraser University, Burnaby, British Columbia, Canada

While Intel's popular Universal PROM Programmer (UPP) works effectively in its intended capacity as a system development tool, it has two major drawbacks. First, the so-called personality cards that are required for manually programming each type of programmable read-only memory are expensive and much too complicated to build. Second, it is restricted to Intel PROMs, so that the newer complementary-MOS erasable-PROM



Burning softly. Complexity of control board (a) and personality cards (b) in universal PROM programmer are reduced if software-based controller leads system's host computer through various read/write burn-in phases. Personality card is shown for the IM6604 PROM. S-100 bus interface (c) units UPP to 8080 host processor. Small program (table) guides 8080 through write-and-verify sequence.

WRITE-AND-VERIFY SEQUENCE: IM6604 E-PROM

Write/Verify Routine

```
Statement                        Comment

COUNT:   DS 1
WRITE:   PUSH  B
         PUSH  PSW
         CALL  DOUT    ;   LATCH DATA BYTE.
         XRA   A
         STA   COUNT   ;   INITIALIZE COUNT.
         POP   PSW
         CALL  ADDRL   ;   LATCH ADDRESS.
         CALL  INFO0   ;   INITIALIZE CRT.
WLP:     CALL  PULSE   ;   SEND A PROG PULSE.
                       ;   VERIFY
         MOV   B,A     ;   SAVE DATA BYTE.
         LDA   COUNT
         INR   A       ;   BUMP UP COUNT.
         STA   COUNT
         CALL  READ    ;   GET CELL CONTENTS.
         CMP   B       ;   COMPARE WITH DATA.
      JZ BURN          ;   EXIT IF VERIFIED.
         CALL  INFO1   ;   UPDATE CRT.
         LDA   COUNT
         CPI   030H    ;   PULSE LIMIT . . .
         MOV   A,B
      JC WLP           ;   EXCEEDED?
         STC           ;   IF SO EXIT.
         POP   B
      RET
BURN:    . . . . . .   ;   LAST WRITE OK,
                       ;   BURN AND EXIT.
```

Pulse Routine

```
Statement                        Comment

PROG     EQU  80H      ;   CONTROL BIT.
PULSE:   PUSH PSW
         PUSH B
         MOV     A,D   ;   ADDR/CTL BYTE.
         ORI  PROG     ;   SET PROGRAM BIT.
         MVI  B, OEH   ;   SET 14MS COUNT.
         CALL HOUT     ;   START PROG PULSE.
         CALL MSDLAY   ;   HOLD IT.
         XRI  PROG     ;   CLEAR PULSE BIT.
         CALL HOUT     ;   RESET PROG PULSE.
         MVI  B, 07H   ;   7MS COUNT.
         CALL MSDLAY   ;   WAIT (2/3 DC).
         POP  B
         POP  PSW
      RET
```

chips cannot be programmed. The personality cards can be simplified and the UPP peripheral device made more versatile, however, if a software-based controller guides the system's host computer through the various read/write phases required to program and verify the contents of PROMs.

The basic UPP interface has eight data-input and eight data-output lines, along with read-data, read-acknowledge, and read-status ports. Also included is a write-data line, a write high-address and write low-address line, and an interrupt line. A pulsed control signal required for programming each PROM location is handled via a 4-bit 4040 microprocessor on a control card in the UPP.

The best way to simplify such a peripheral is to have the host computer provide the timing, control, and logic necessary for programming, reading, and verifying the contents of PROMs and E-PROMs. The complexity of the UPP's control card is then reduced to that shown in (a) of the figure, where the 4040-based setup is replaced by four C-MOS I/O chips (RCA 1852D).

The host computer consequently sees one input port and three output ports. The input port is used for returning the contents of a selected memory word. Two of the output ports are used for latching address and control information; the remaining output line is used for a data-output latch.

Of the 16 PROM-address lines available, 12 are used to address up to 4 kilobytes of memory. The remaining four lines can be used for program and chip control. Consider the personality card of the 512-by-8-bit IM6604 PROM, for example (b in figure). There, line A_{15} is used for a program pulse enable, A_{14} and A_{13} are used for chip select and chip enable, respectively, while line A_{12} is used for a strobe pulse. The popular 27XX E-PROM series would require only two control lines. Because the 27XX chips are powered by 5 volts, while C-MOS devices require 10 V for programming, the 27XX's personality card would interface to the host computer through open-collector devices.

In general, then, a personality card will consist of a bidirectional data bus, the required number of address lines, and a pulser circuit. It is thus used mainly to route the bus lines to the proper front-panel pin positions on the UPP. The pulse circuit must be designed to be reset by the UPP's front-panel reset button. This can be accomplished by connecting the reset line to the 1852D's CLE inputs. For completeness, the program control line (on the control card) is also connected to the program LED on the front panel.

A typical S-100 bus interface for the modified UPP is shown in (c). I/O ports 32, 33, and 34 have been decoded for a data strobe (read), write low-address, and write high-address, respectively. Interface software must include a timing subroutine and program pulse and verification routines particular to the PROM that is programmed.

As for the programming required, the sequence in the table outlines the steps necessary for the write-and-verify operation in the IM6604. The program is written for the host 8080A microprocessor. ☐

PROM PROGRAMMING

Programmable-read-only-memories can perform many logic functions and at the same time reduce circuit complexity. The key to using a PROM, however, is the ability to program it.

A read-only memory (ROM) is a random access memory in which data is mask-programmed during the manufacturing process. A programmable ROM, or PROM, is also a read-only memory; however, memory patterns are programmed into it by the user after manufacture. Read-only memories act as code converters that accept input codes and generate arbitrarily assigned output patterns. They are logically equivalent to truth tables in which the number of input variables equals the number of ROM address inputs and the number of output functions equals the number of ROM outputs. For example, an 8K PROM organized as 1024 x 8 bits implements the truth table for eight functions of ten variables.

Techniques for programming PROMs differ according to the technologies used to implement the devices. Certain types of MOS PROMS (EPROMs) can be erased and reprogrammed, but bipolar TTL PROMs can be programmed only once. In either case, the user can custom-program PROMs and, when necessary, make system changes without facing the substantial mask charges, manufacturing turn-around time, and need to maintain inventories of different patterns that are required by ROMs.

Bipolar TTL PROMs offer access times in the 25 ns to 50 ns range and ECL devices are available with access times in the 10 ns to 20 ns range, while MOS is slower than either type. Although MOS has historically been available in larger bit configurations than bipolar, advances in bipolar memory technology are narrowing the gap. Isoplanar Schottky PROMs, for example, offer densities between those of PMOS and silicon-gate NMOS. They are also cost- and performance-competitive not only with ROMs, but also with standard TTL on a per-function basis.

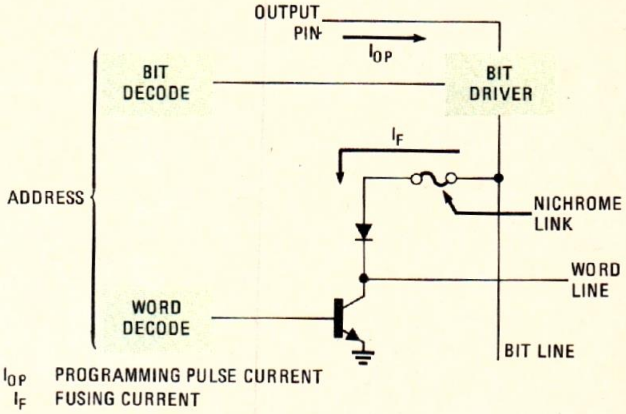


Fig. 1 Programming Current Path

Isoplanar Schottky PROMs in various sizes and configurations are available from Fairchild, with high performance guaranteed across both commercial and military temperature ranges. These devices include the 93417/93427 256 x 4-bit, the 93436/93446 512 x 4-bit, the 93438/93448 512 x 8-bit, the 93452/93453 1024 x 4-bit, and the 93450/93451 1024 x 8-bit PROMs. They are completely TTL compatible, include fully decoded addressing, and are available with open-collector or 3-state outputs. A 256 x 4-bit Isoplanar ECL PROM with a typical access time of 11 ns is also available.

Reprinted from "Fairchild Journal of Semiconductor Progress, Vol. 6 No. 2

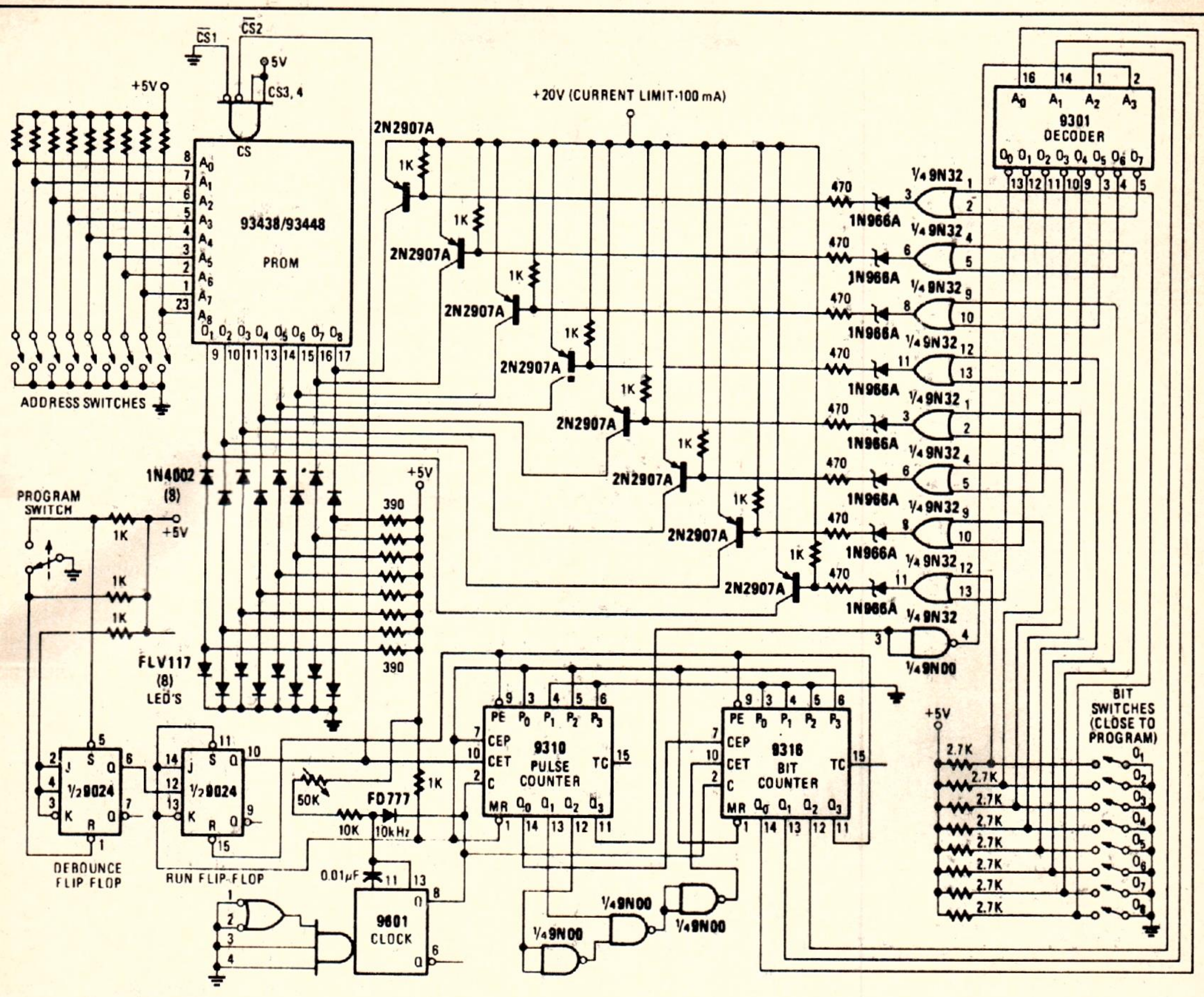


Fig. 2 PROM Programming Circuit

APPLICATIONS

Programmable ROMs are widely used today in microprogrammed computers ranging from the largest mainframe systems to microcomputers. They are also finding increased use as replacements for random logic in peripheral controllers, digital controllers, instruments, and terminals. Circuit applications include fixed data and instruction storage in computers, microprogrammed system control storage, look-up and decision tables, and address and priority mapping. Other applications include character/vector generation, encoding/decoding and sequential controllers. Many of these functions will, in time, be performed by field-programmable logic arrays (FPLAs); because of their large memory capabilities, however, PROMs will remain important elements in mass storage applications.

PROGRAMMING TECHNIQUE

The basic technique for programming a PROM having fusible links is to provide the amount of current necessary to permanently open the link associated with a selected bit, thereby programming the bit to a logic "0" or a logic "1", depending upon the circuit design.

Several materials are used to provide the programmable links within bipolar PROMs, including nichrome, tungsten, titanium, and doped polysilicon. The nichrome fuse link is the most popular because of extensive past experience with this material, high programming yield, and reliability. In Fairchild PROMs, the fuse is a thin-film nichrome link with a small, square notch that concentrates the fusing energy in a central neck during programming, ensuring a wide, clean gap.

Fairchild Isoplanar Schottky TTL PROMs are manufactured with all bits in the HIGH, or logic "1", state. When a programming current pulse (I_{OP}) is applied to an output pin, fusing current (I_F) is driven through the selected bit, as shown in *Figure 1*. Because of careful device design, almost all of the fusing energy is delivered to the nichrome link, opening the link and programming the bit to a LOW, or logic "0", state. Minimal losses to leakage paths and intermediate circuits permit the link to open rapidly with a low-energy pulse, which improves reliability. The nichrome links actually program on the rise time of the pulse; this permits the reduction of programming pulse width for high-speed, low-energy device programming. *Table 1* gives programming specifications for Fairchild PROMs.

PROGRAMMING PROCEDURE

The following steps, performed with reference to *Table 1*, can be used to program one bit at a time:

1. Apply the proper power by supplying $V_{CC} = +5$ V and ground.

2. Select the word to be programmed by applying the appropriate levels to the Address pins.

3. Select the chip for programming by disabling the outputs: apply a HIGH to the active LOW Chip Select inputs and a LOW to the active HIGH inputs, if present. All PROMs have active LOW Chip Select inputs; the 93438/93448 512 x 8-bit and 93450/93451 1024 x 8-bit PROMs have active HIGH Chip Select inputs as well.

4. Apply a programming pulse to the output pin associated with the bit to be programmed. The other outputs may be left open or tied to any HIGH. Note that only one output at a time can be programmed.

5. To verify a LOW in the bit just programmed, remove the pulse from the output pin and sense the pin after applying a LOW to the active LOW Chip Selects and a HIGH to the active HIGH Chip Selects, if any.

6. Repeat steps 2 through 5 as necessary for each bit to be programmed.

PROGRAMMERS

Most PROM programming is done with commercially available programmers. An alternative is simpler programmers with manual switches; for example, the circuit shown in *Figure 2*, designed for PROMs having eight outputs, is capable of sequentially programming all bits in words of up to 8-bit length. The address of the word to be programmed is entered by the address switches, and the desired bit pattern for that word is set up on the bit switches. The contents of the PROM at the selected address are displayed on the eight FLV117 LEDs as long as the program switch is open. When the program switch is open, the PROM is enabled and, for every bit in the HIGH state, the associated LED is on; for every bit in the LOW state, the associated LED is off. Closing the program switch disables the chip and all of the LEDs are simultaneously turned on by current supplied through the 390 Ω resistors. The 1N4002 diodes isolate the LEDs from the 21 V programming pulse.

One-half of a 9024 dual JK flip-flop is used as a switch debouncer, while the other half is the run flip-flop. The 9601 one-shot is connected as a 10 kHz oscillator. When the program is initiated by closing the program switch, the switch debouncer sets; this clocks the other half of the flip-flop into the run state and enables the pulse and bit counters to initiate programming pulses, preparing the PROM for programming. The pulse counter is preset to 5 to provide the requisite 20% duty factor, and the bit counter is preset to 8. To avoid overlap problems between the programming pulse, chip enable and scan, the bit counter advances when the pulse counter goes from state 3 to state 4. The bit to be programmed is decoded by the 9301 and wired-OR with the bit switch. The OR gate is a high-voltage driver that supplies the drive to the programming transistors upon selection by a closed bit switch. When the last bit has been programmed, the counter presets itself and resets the run flip-flop, completing the programming sequence for the selected word.

BOARD PROGRAMMING

It is often convenient to program PROMs mounted on a circuit board in wired-OR configurations. Fairchild devices are particularly convenient for board programming because only the Chip Select and output pins need to be accessed to program the part. *Figure 3* shows a circuit used for board programming. The programmer is connected to the output bus as shown and the Chip Selects are driven by a decoder with elevated voltage levels. Thus, all that is required for board programming is to raise V_{CC}, V_{EE}, and the Device Select inputs on the decoder to 7.6 V above their normal operating levels. The standard 21 V programming pulse then programs bits in the PROM having an active LOW Chip Select input of approximately 7.8 V, which is high enough to disable the PROM outputs. The following steps are used to program board-mounted PROMs with this circuit:

1. Connect the common output bus of PROMs 0 through n as shown in *Figure 3*.

2. On 93417, 93438, 93450, 93451, 93452, and 93453 PROMs, connect either $\overline{CS}_1$ or $\overline{CS}_2$ to 0 V. Connect the other active LOW Chip Select inputs to the active LOW outputs of the TTL decoder.

PARAMETER	SYMBOL	MIN	RECOMMENDED VALUE	MAX	UNITS	COMMENTS
Power Supply Voltage	V_{CC}	4.75	5.0	5.25	V	
Address Input	V_{IH}	2.4	5.0	5.0	V	Do not leave inputs open
	V_{IL}	0	0	0.4	V	
Chip Select	$\overline{CS}_1$, $\overline{CS}_2$	2.4	5.0	5.0	V	Either or both
	CS_3, CS_4	0	0	0.4	V	
Programming Pulse Voltage	V_{OP}	20	21	21	V	Applied to output to be programmed
Programming Pulse Current	I_{OP}			100	mA	If pulse generator is used, set current limit to this max value.
Programming Pulse Width	t_{pw}	0.05	0.18	50	ms	
Programming Pulse Duty Cycle			20	20	%	Maximum duty cycle to maintain $t_c < 85°C$
Programming Pulse Rise Time	t_r	0.5	1.0	3.0	μs	
Number of Required Pulses		1		4		
Case Temperature			25	85	°C	

Table 1. Programming Specifications

3. On 93436 and 93446 PROMs, connect the Chip Select inputs to the TTL decoder outputs.

4. On 93438, 93448, 93450 and 93451 PROMs, connect the CS_3 and CS_4 inputs to a HIGH or leave them unconnected.

5. To program a bit in one of the PROMs, simultaneously raise the TTL decoder supply voltages to V_{CC} = 12.6 V and V_{EE} = 7.6 V and select the desired PROM via the TTL decoder Address inputs (HIGH = 10.0 V, LOW = 7.6 V).

6. Close the bit switch associated with the bit to be programmed and raise the programming pulse voltage to 21 V. The selected PROM, the active LOW Chip Select input of which is at approximately 7.8 V, programs. The active LOW Chip Select inputs of all other PROMs are at approximately 10.6 V; these PROMs remain deselected, and do not program.

7. To verify a LOW in the bit just programmed, remove the programming pulse and sense the PROM output bus after simultaneously lowering the TTL decoder supply voltages to V_{CC} = 5.0 V and V_{EE} = 0 V and lowering the TTL decoder Address inputs to their normal levels (HIGH = 2.4 V, LOW = 0.4 V).

8. Repeat the procedure for other bits, following the normal programming sequence.

9. To select a different PROM on the board, change the TTL decoder address. **R-E**

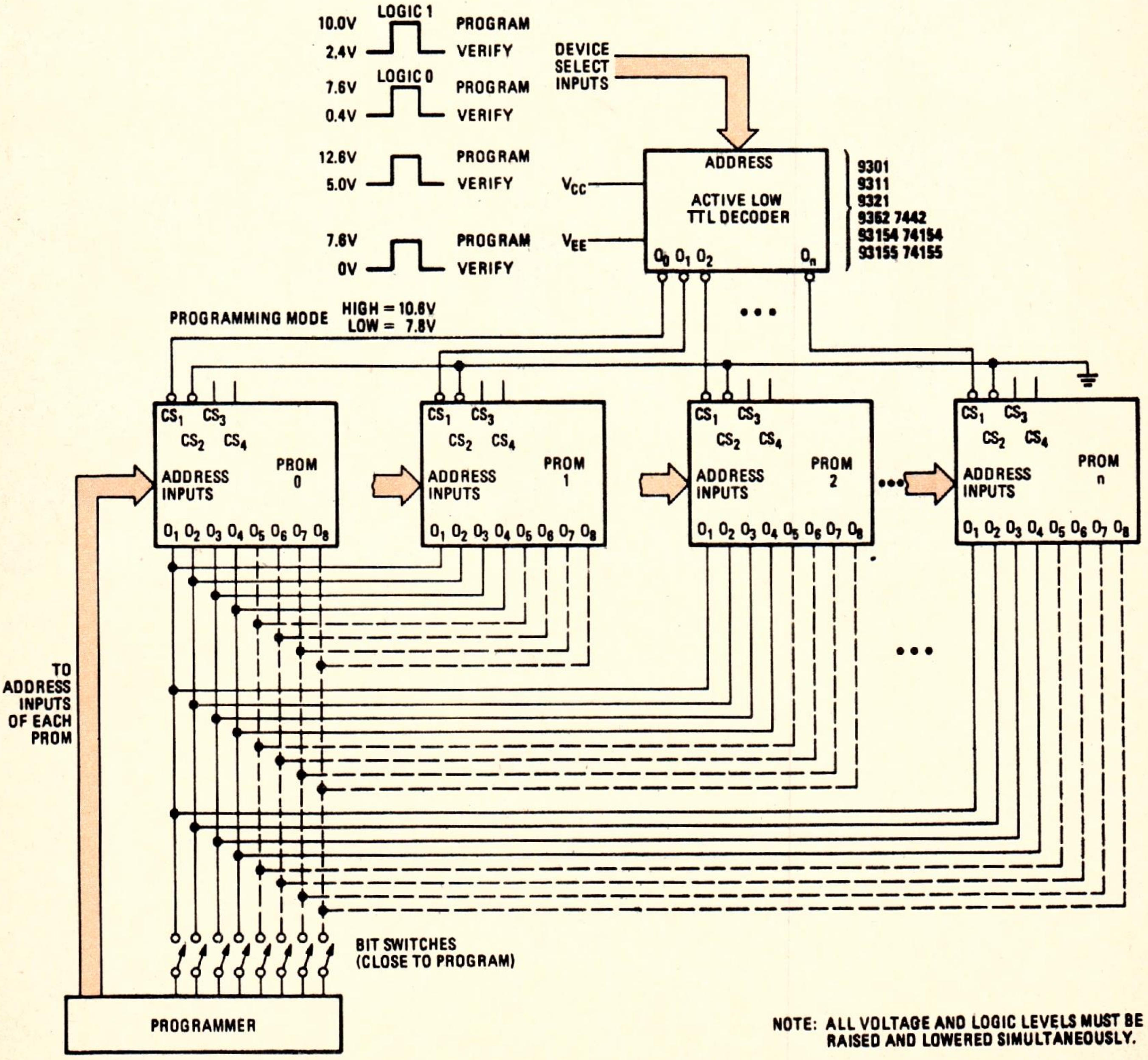


Fig. 3 Board Programming