ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Torino

INFN/TC-85/15
27 Agosto 1985

Gao Zhong-Ren and D. Crosetto:

A MICROCOMPUTER CONTROLLED PAL PROGRAMMER

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Torino.

INFN/TC-85/15
27 Agosto 1985

A MICROCOMPUTER CONTROLLED PAL PROGRAMMER

GAO ZHONG-REN[a] and D. CROSETTO[b]

a) - Institute of Physics of Chinese Academy of Sciences, Peking.

b) - Istituto Nazionale di Fisica Nucleare, C.so M. D'Azeglio 46, 10125 Torino.

ABSTRACT.

PAL (Programmable Array Logic) is a new type of bipolar integrated circuit which can be used to replace the conventional SSI/MSI gates and flip-flops in implementing complex logic functions. Because PAL is programmable, it's much more flexible than conventional TTL logic and can greatly simplify prototyping. Typical chip count reduction gained by using PALs is greater than 4:1.

Our PAL programmer is controlled by a microcomputer; all operations are prompted; and all high voltages on socket pins are automatically removed after operating, so it can be operated easily and securely.

At present, the programmer works with an interface board compatible with the STD bus system, but it can be easily modified to adapt to any other microcomputer system.

1. - INTRODUCTION

PAL implements the familiar sum of products logic with an AND array of fusible links followed by a fixed OR array.

The programming of PAL implies blowing some fuses according to a predesigned pattern, thus configuring the AND and OR gates to perform desired logic functions.

Many semiconductor manufacturers have developed a whole family of PAL devices. There have been already defined 15 standard PAL types in 20 pin packages and 10 standard PAL types in 24 pin packages.

However, the programming of PAL is a complicated task, and the PAL programmers now available on the market are rather expensive and not convenient to use[1]. Thus, for our laboratory use, we decided to design and build this PAL programmer which can be employed to program all 25 types of standard PALs.

2. - PAL AND ITS PROGRAMMING.

There are three kinds of programmable AND-OR array bipolar devices, i.e. PROM, FPLA and PAL. The basic logic structure of PROM consist of a fixed AND array driving a programmable OR array (Fig. 1). It's easy to program, has low cost, and is most commonly used to store program and data in a computer.

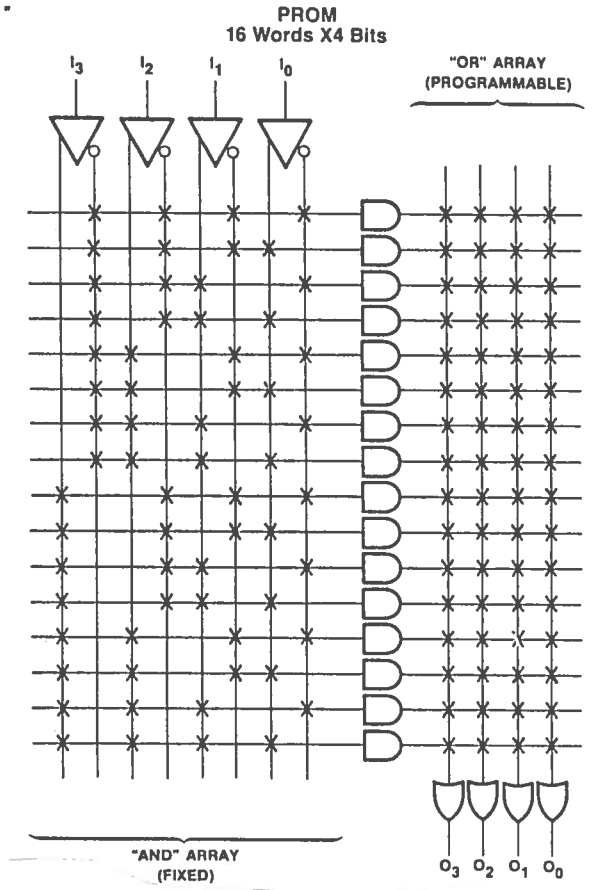

Figure 1. THE BASIC LOGIC STRUCTURE OF A PROM.

The basic logic structure of a FPLA consists of a programmable AND array driving a programmable OR array (Fig. 2). Clearly it's the most flexible device for implementing logic functions, but its flexibility is at the expense of high cost and programming complexity. That makes FLPA rather difficult to use.

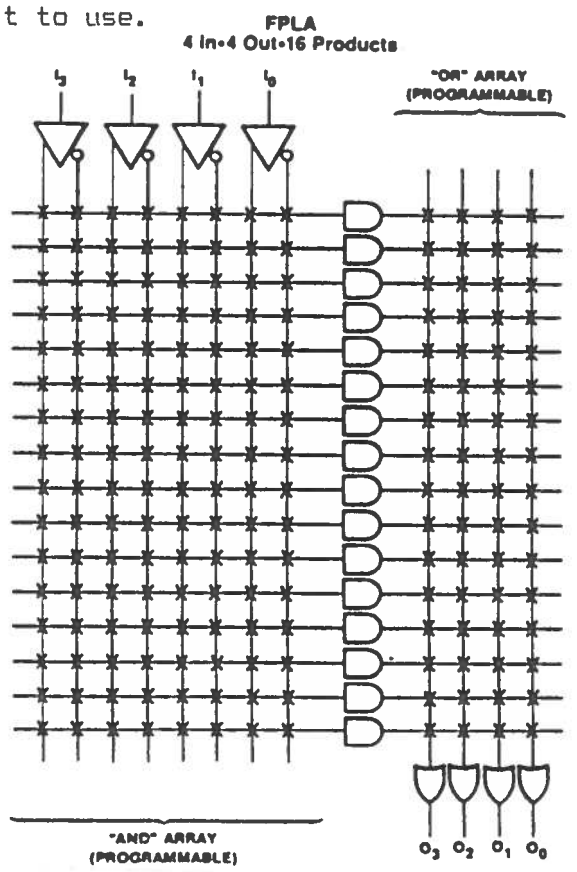

Figure 2. THE BASIC LOGIC STRUCTURE OF A FPLA.

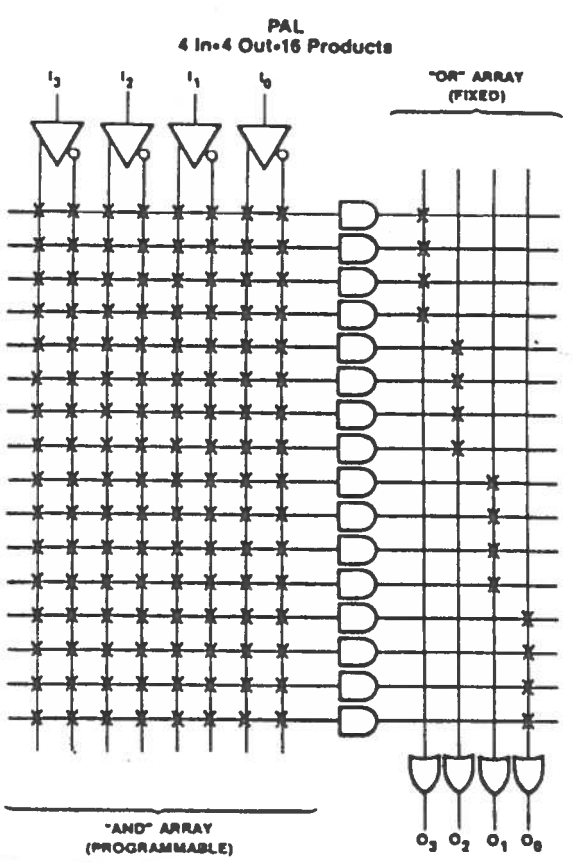

Figure 3.THE BASIC LOGIC STRUCTURE OF A PAL.

The basic logic structure of a PAL, as mentioned above, consists of a programmable AND array driving a fixed OR array (Fig. 3). The PAL combines much of the flexibility of PLA with a low cost and less complex programming.

PART NO.	INPUT	OUTPUT	PROGRAMMABLE I/O'S	FEEDBACK REGISTER	OUTPUT POLARITY	FUNCTIONS
10H8	10	8			AND-OR	AND-OR Gate Array
12H6	12	6			AND-OR	AND-OR Gate Array
14H4	14	4			AND-OR	AND-OR Gate Array
16H2	16	2			AND-OR	AND-OR Gate Array
16C1	16	2			BOTH[1]	AND-OR Gate Array
20C1	20	2			BOTH[1]	AND-OR Gate Array
10L8	10	8			AND-NOR	AND-OR Invert Gate Array
12L6	12	6			AND-NOR	AND-OR Invert Gate Array
14L4	14	4			AND-NOR	AND-OR Invert Gate Array
16L2	16	2			AND-NOR	AND-OR Invert Gate Array
12L10	12	10			AND-NOR	AND-OR Invert Gate Array
14L8	14	8			AND-NOR	AND-OR Invert Gate Array
16L6	16	6			AND-NOR	AND-OR Invert Gate Array
18L4	18	4			AND-NOR	AND-OR Invert Gate Array
20L2	20	2			AND-NOR	AND-OR Invert Gate Array
16L8	10	2	6		AND-NOR	AND-OR Invert Gate Array
20L10	12	2	8		AND-NOR	AND-OR Invert Gate Array
16R8	8	8		8	AND-NOR	AND-OR Invert Gate Array w/Reg's
16R6	8	6	2	6	AND-NOR	AND-OR Invert Array w/Reg's
16R4	8	4	4	4	AND-NOR	AND-OR Invert Array w/Reg's
20X10	10	10		10	AND-NOR	AND-OR-XOR Invert w/Reg's
20X8	10	8	2	8	AND-NOR	AND-OR-XOR Invert w/Reg's
20X4	10	4	6	4	AND-NOR	AND-OR-XOR Invert w/Reg's
16X4	8	4	4	4	AND-NOR	AND-OR-XOR Invert w/Reg's
16A4	8	4	4	4	AND-NOR	AND-CARRY-OR-XOR Invert w/Reg's

Table 1. STANDARDS PALS AND THEIR CHARACTERISTICS.

Table 1 and Fig. 4 and 5 show the whole standard PAL family now available commercially along with their characteristics. This variety of PAL types could be used to design circuits covering the spectrum of logic functions to fit almost all applications.

As an example for showing the power of PAL, we introduce the following application used in our CPU board (MICRO-LEARN GD1) design.

In the MICRO-LEARN GD1 board we need the following logic functions:

INTAK = M1 * IORQ.

/WRTLTH = /IORQ + /WR + /PORTSEL

/DATAIN = RD * MREQ * OBMEM + RD * IORQ * PORTSEL + M1 * IORQ * IEI * /IEO + /BUSAK *
/RD * MREQ + /BUSAK * /M1 * /IORQ * /MREQ + /BUSAK * /RD * IORQ * /M1

RDROM = RD * MREQ * BOOT.

RDRAM = /BOOT * RD * MREQ

WRRAM = WR * MREQ.

/RWEPFL = WR * IORQ.

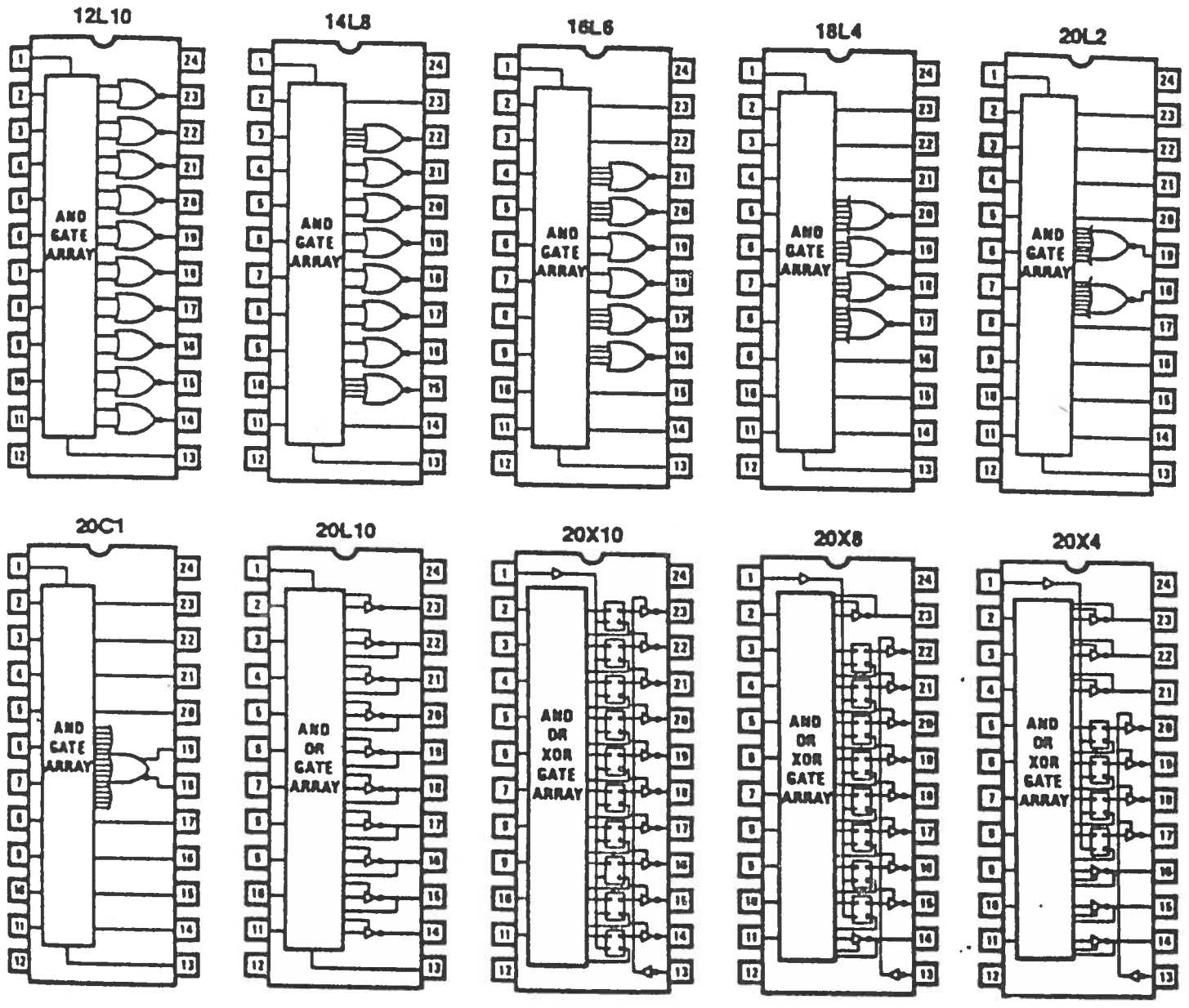

Figure 4. 24 PIN STANDARD PALS.

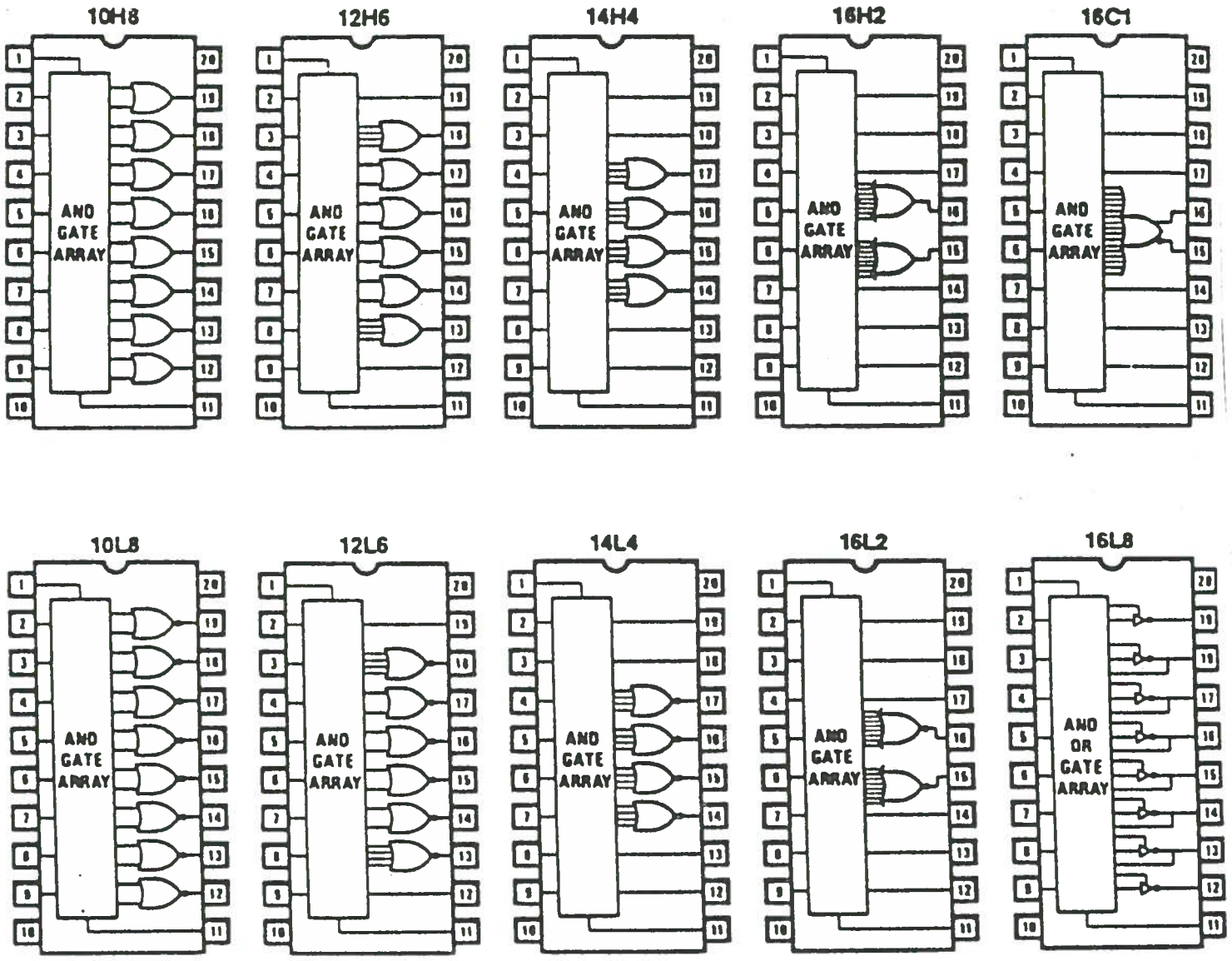

Figure 5. 20 PIN STANDARD PALS.

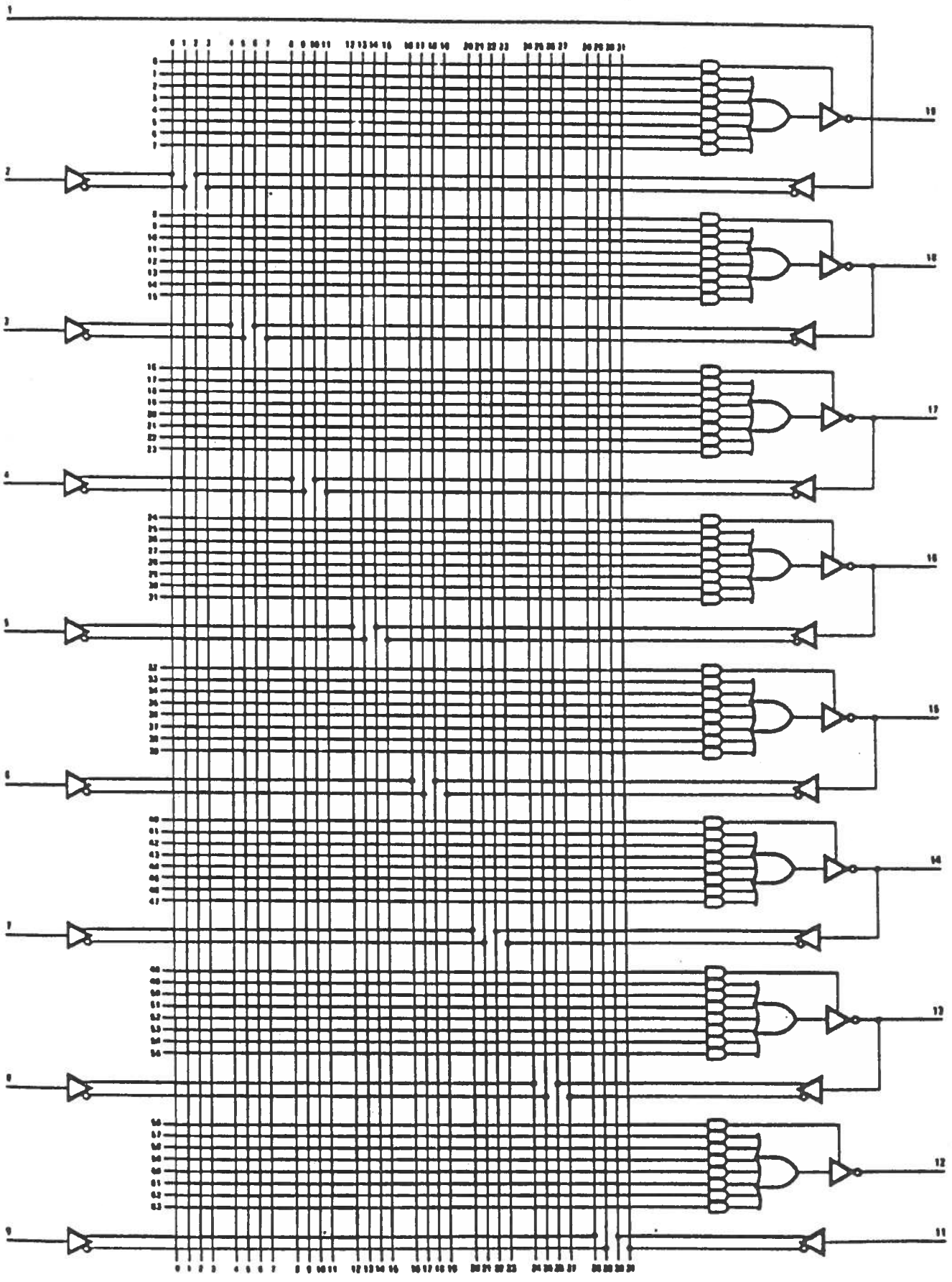

Figure 6. THE LOGIC DIAGRAM OF PAL 16L8.

Obviously, this is a complicated set of logic functions, with 11 inputs and 7 outputs. We selected PAL 16L8 for our use. It has 20 pins, 11 for inputs, 7 for outputs, and other two pins for Vcc and ground. The logic diagram of 16L8 is shown in Fig. 6.

Internally, PAL 16L8 has 64 product lines and 32 input lines; all together (64 * 32) fusible links in the AND array. In order to implement the logic function mentioned above, two procedures had to be followed. The first is to assign pins and design a programming pattern, the second is to have the PAL programmed according to the predesigned pattern. Figs. 7 and 8 show the pin assignment and the programming pattern, respectively (in Fig. 8 "/" represents the links blown during the programming procedure).

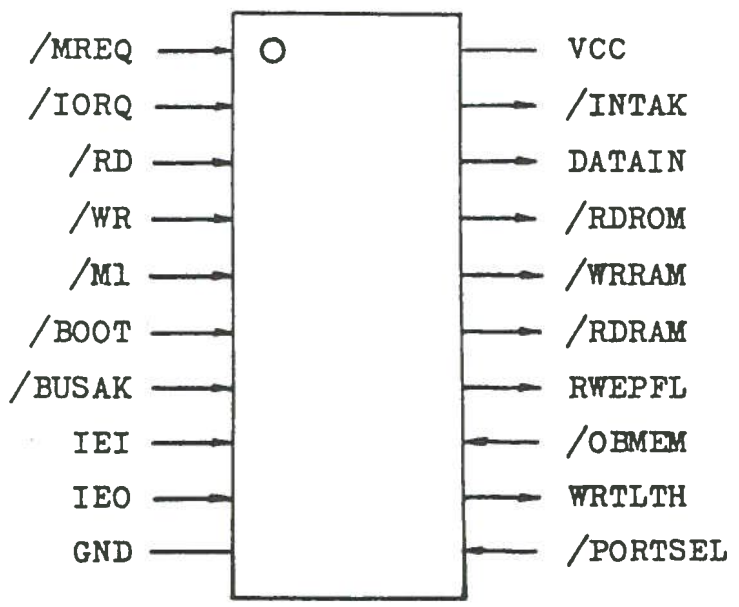

Figure 7. PIN ASSIGNEMENT OF 16L8 PAL.

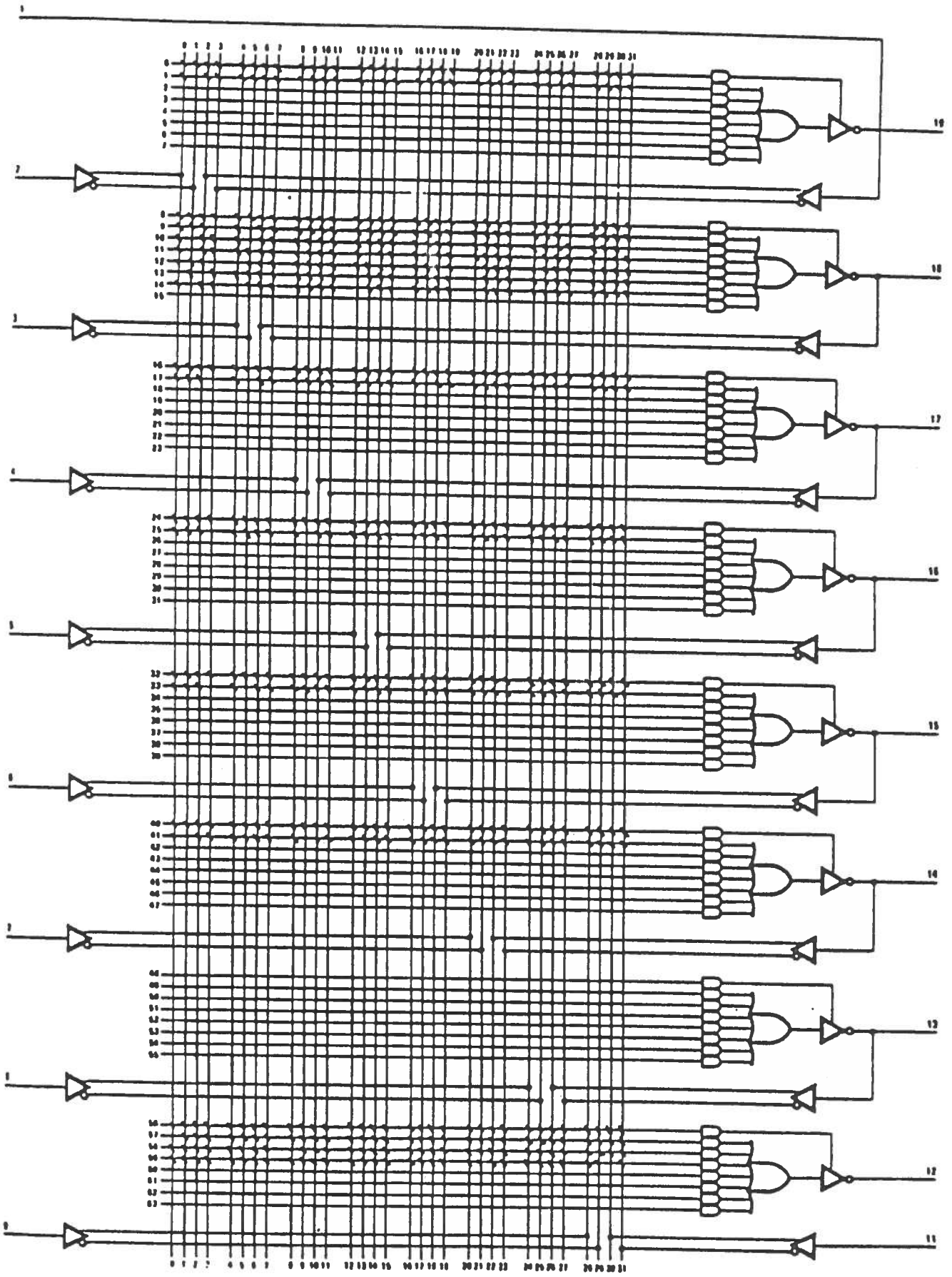

Figure 8. PROGRAMMING PATTERN OF 16L8 PAL.

The pin configuration of PAL in programming and in verifying are in Figs.9 and 10 respectively.

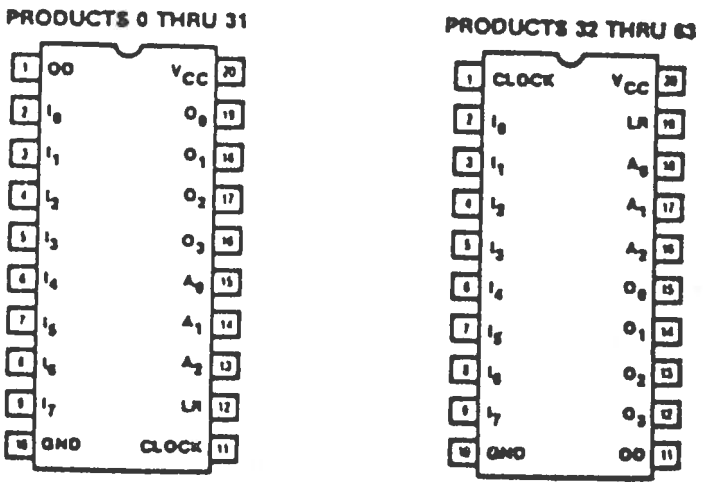

Figure 9. PIN CONFIGURATION OF 20 PIN PAL.

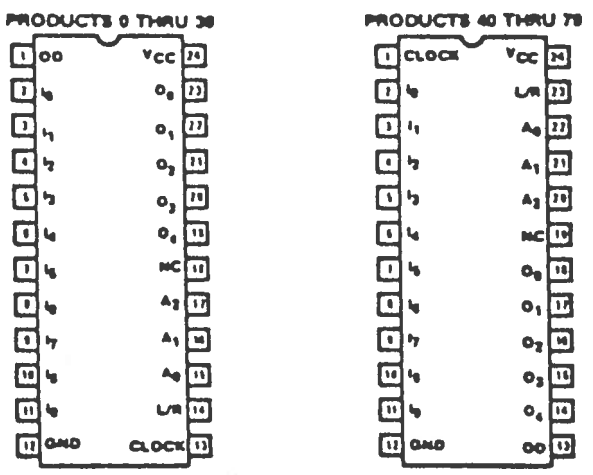

Figure 10. PIN CONFIGURATION OF 24 PIN PAL.

The corresponding waveforms and their parameters are shown in Figs. 11 and 12 and Table 2 respectively.

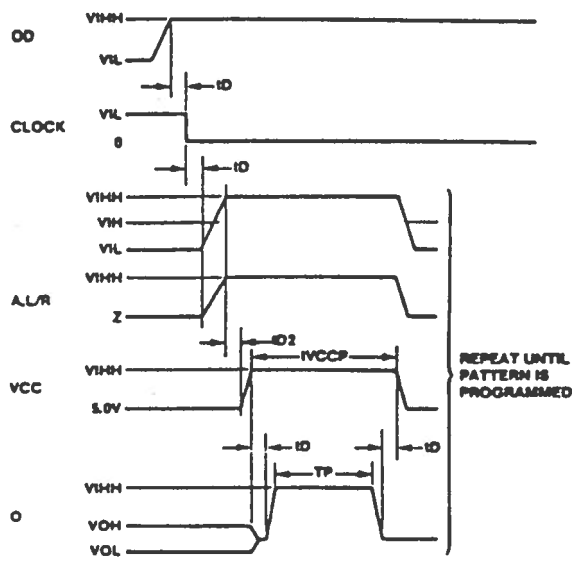

Figure 11. PROGRAMMING WAVEFORMS.

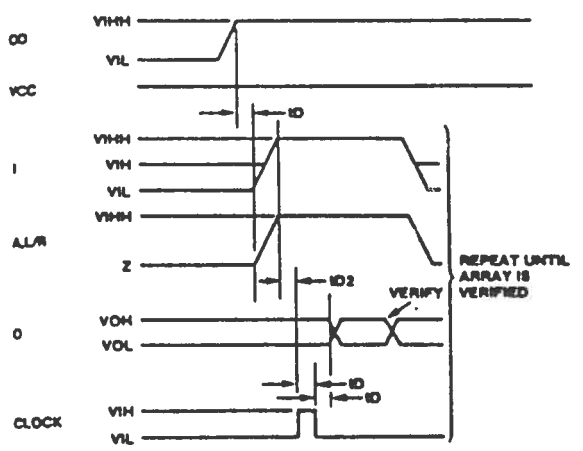

FIGURE 12. VERIFYING WAVEFORMS.

SYMBOL	PARAMETER		LIMITS MIN	TYP	MAX	UNIT
V_{IHH}	Program-level input voltage		11.5	11.75	12	V
I_{IHH}	Program-level input current	Output Program Pulse			50	mA
		OD, L/R			50	
		All other inputs			10	
I_{CCH}	Program Supply Current				900	mA
t_{VCCP}	Pulse Width of V_{CC} @ V_{IHH}				60	µS
T_P	Program Pulse Width		10	20	50	µS
t_D	Delay Time		100			ns
t_{D2}	Delay Time after L/R Pin		10			µS
	V_{CCP} Duty Cycle				20	%
t_{RP}	Rise time of output programming and address pulses		1	1.5	10	V/µS
V_{CCPP}	V_{CC}value for first verify		4.75	5.0	5.25	V

Table 2. PROGRAMMING PARAMETERS.

In order to program the PAL (by fusing the links), we need to build a current pulse generator, which can deliver current pulses according to Tables 3 and 4.

In Table 3 and 4 L = Low-level input voltage, V_{IL}; H = High-level input voltage, V_{IH}; HH = High-level program voltage, V_{IHH}; Z = High impedance.

INPUT LINE NUMBER	PIN IDENTIFICATION								
	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	Z
25	HH	H	HH	HH	HH	HH	HH	HH	Z
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	Z
29	H	HH	HH	HH	HH	HH	HH	HH	Z
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

PRODUCT LINE NUMBER	PIN IDENTIFICATION						
	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0, 32	Z	Z	Z	HH	Z	Z	Z
1, 33	Z	Z	Z	HH	Z	Z	HH
2, 34	Z	Z	Z	HH	Z	HH	Z
3, 35	Z	Z	Z	HH	Z	HH	HH
4, 36	Z	Z	Z	HH	HH	Z	Z
5, 37	Z	Z	Z	HH	HH	Z	HH
6, 38	Z	Z	Z	HH	HH	HH	Z
7, 39	Z	Z	Z	HH	HH	HH	HH
8, 40	Z	Z	HH	Z	Z	Z	Z
9, 41	Z	Z	HH	Z	Z	Z	HH
10, 42	Z	Z	HH	Z	Z	HH	Z
11, 43	Z	Z	HH	Z	Z	HH	HH
12, 44	Z	Z	HH	Z	HH	Z	Z
13, 45	Z	Z	HH	Z	HH	Z	HH
14, 46	Z	Z	HH	Z	HH	HH	Z
15, 47	Z	Z	HH	Z	HH	HH	HH
16, 48	Z	HH	Z	Z	Z	Z	Z
17, 49	Z	HH	Z	Z	Z	Z	HH
18, 50	Z	HH	Z	Z	Z	HH	Z
19, 51	Z	HH	Z	Z	Z	HH	HH
20, 52	Z	HH	Z	Z	HH	Z	Z
21, 53	Z	HH	Z	Z	HH	Z	HH
22, 54	Z	HH	Z	Z	HH	HH	Z
23, 55	Z	HH	Z	Z	HH	HH	HH
24, 56	HH	Z	Z	Z	Z	Z	Z
25, 57	HH	Z	Z	Z	Z	Z	HH
26, 58	HH	Z	Z	Z	Z	HH	Z
27, 59	HH	Z	Z	Z	Z	HH	HH
28, 60	HH	Z	Z	Z	HH	Z	Z
29, 61	HH	Z	Z	Z	HH	Z	HH
30, 62	HH	Z	Z	Z	HH	HH	Z
31, 63	HH	Z	Z	Z	HH	HH	HH

Table 3. SELECTION TABLE FOR 20 PIN PAL.

INPUT LINE NUMBER	PIN IDENTIFICATION										
	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	HH	HH	L	HH	HH	HH	HH	HH	HH	Z
25	HH	HH	HH	H	HH	HH	HH	HH	HH	HH	Z
26	HH	HH	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	HH	HH	H	HH	HH	HH	HH	HH	HH	HH
28	HH	HH	L	HH	HH	HH	HH	HH	HH	HH	Z
29	HH	HH	H	HH	HH	HH	HH	HH	HH	HH	Z
30	HH	HH	L	HH	HH	HH	HH	HH	HH	HH	HH
31	HH	HH	H	HH	HH	HH	HH	HH	HH	HH	HH
32	HH	L	HH	HH	HH	HH	HH	HH	HH	HH	Z
33	HH	H	HH	HH	HH	HH	HH	HH	HH	HH	Z
34	HH	L	HH	HH	HH	HH	HH	HH	HH	HH	HH
35	HH	H	HH	HH	HH	HH	HH	HH	HH	HH	HH
36	L	HH	HH	HH	HH	HH	HH	HH	HH	HH	Z
37	H	HH	HH	HH	HH	HH	HH	HH	HH	HH	Z
38	L	HH	HH	HH	HH	HH	HH	HH	HH	HH	HH
39	H	HH	HH	HH	HH	HH	HH	HH	HH	HH	HH

PRODUCT LINE NUMBER	PIN IDENTIFICATION							
	O_4	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0, 40	Z	Z	Z	Z	HH	Z	Z	Z
1, 41	Z	Z	Z	Z	HH	Z	Z	HH
2, 42	Z	Z	Z	Z	HH	Z	HH	Z
3, 43	Z	Z	Z	Z	HH	Z	HH	HH
4, 44	Z	Z	Z	Z	HH	HH	Z	Z
5, 45	Z	Z	Z	Z	HH	HH	Z	HH
6, 46	Z	Z	Z	Z	HH	HH	HH	Z
7, 47	Z	Z	Z	Z	HH	HH	HH	HH
8, 48	Z	Z	Z	HH	Z	Z	Z	Z
9, 49	Z	Z	Z	HH	Z	Z	Z	HH
10, 50	Z	Z	Z	HH	Z	Z	HH	Z
11, 51	Z	Z	Z	HH	Z	Z	HH	HH
12, 52	Z	Z	Z	HH	Z	HH	Z	Z
13, 53	Z	Z	Z	HH	Z	HH	Z	HH
14, 54	Z	Z	Z	HH	Z	HH	HH	Z
15, 55	Z	Z	Z	HH	Z	HH	HH	HH
16, 56	Z	Z	HH	Z	Z	Z	Z	Z
17, 57	Z	Z	HH	Z	Z	Z	Z	HH
18, 58	Z	Z	HH	Z	Z	Z	HH	Z
19, 59	Z	Z	HH	Z	Z	Z	HH	HH
20, 60	Z	Z	HH	Z	Z	HH	Z	Z
21, 61	Z	Z	HH	Z	Z	HH	Z	HH
22, 62	Z	Z	HH	Z	Z	HH	HH	Z
23, 63	Z	Z	HH	Z	Z	HH	HH	HH
24, 64	Z	HH	Z	Z	Z	Z	Z	Z
25, 65	Z	HH	Z	Z	Z	Z	Z	HH
26, 66	Z	HH	Z	Z	Z	Z	HH	Z
27, 67	Z	HH	Z	Z	Z	Z	HH	HH
28, 68	Z	HH	Z	Z	Z	HH	Z	Z
29, 69	Z	HH	Z	Z	Z	HH	Z	HH
30, 70	Z	HH	Z	Z	Z	HH	HH	Z
31, 71	Z	HH	Z	Z	Z	HH	HH	HH
32, 72	HH	Z	Z	Z	Z	Z	Z	Z
33, 73	HH	Z	Z	Z	Z	Z	Z	HH
34, 74	HH	Z	Z	Z	Z	Z	HH	Z
35, 75	HH	Z	Z	Z	Z	Z	HH	HH
36, 76	HH	Z	Z	Z	Z	HH	Z	Z
37, 77	HH	Z	Z	Z	Z	HH	Z	HH
38, 78	HH	Z	Z	Z	Z	HH	HH	Z
39, 79	HH	Z	Z	Z	Z	HH	HH	HH

Table 4. SELECTION TABLE FOR 24 PIN PAL.

Based on Monolithic Memories (MMI) PAL timing specifications, we designed and built our microcomputer controlled PAL programmer.

3. HARDWARE OF THE PAL PROGRAMMER.

The PAL programmer consists of a programming current generator and an interface, which is compatible with the STD-BUS. The block diagram of the PAL programmer is shown in Fig. 13.

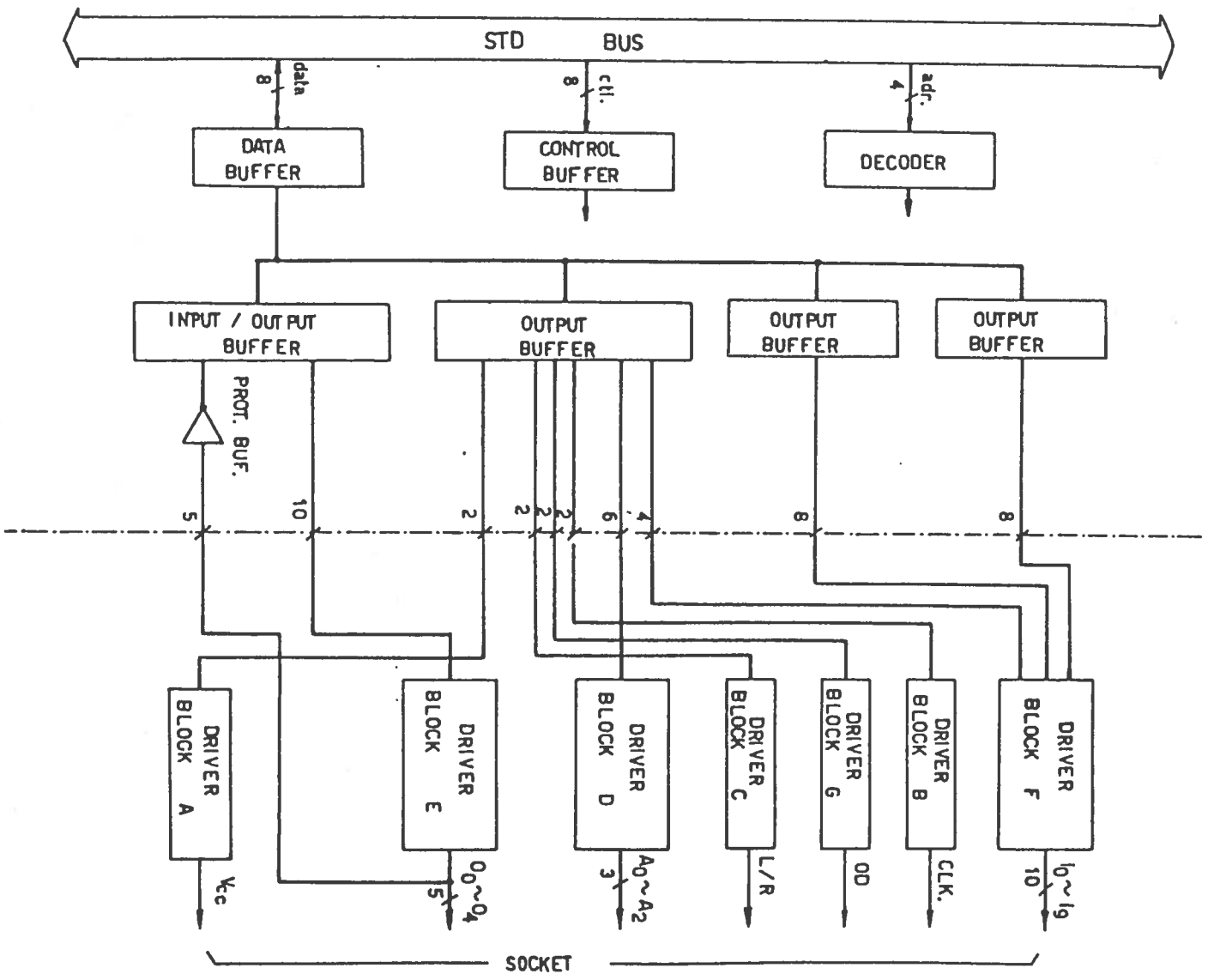

Figure 13. BLOCK DIAGRAM OF PAL PROGRAMMER.

The programming current generator is installed in a separete box on the panel of which there are four sockets with an indicator. Fig. 14 shows the arrangement of the four sockets, two for 20 pin PALs and two for 24 pin PALs. The indicator shows the operating state of the programmer; when it's lighted, the PAL is under operation; otherwise there is neither high voltage nor power supply on sockets so that one can insert or extract PALs with no danger.

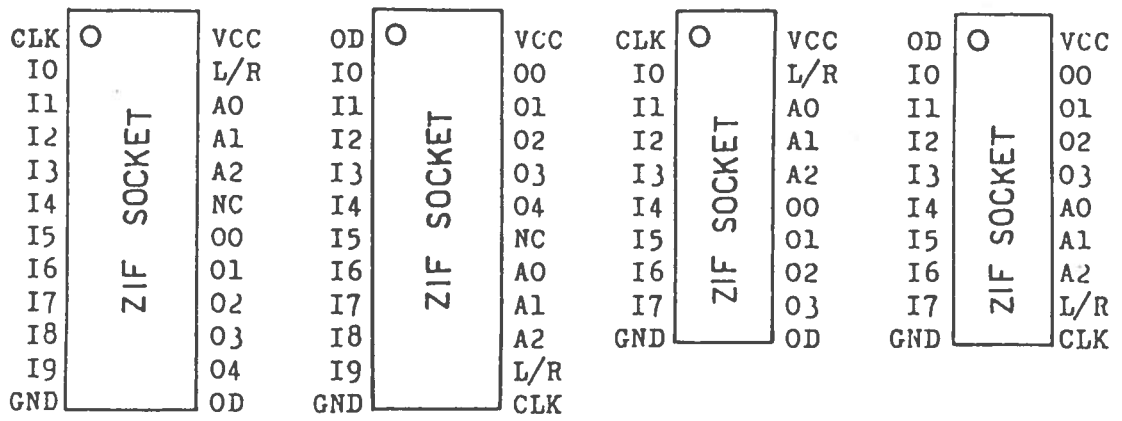

Figure 14. THE SOCKET ARRANGEMENT ON THE PANEL.

There are seven driver blocks in the programming current generator which can deliver above specified currents for programming and verifying.

Each driver has two inputs which are connected to the interface. Under program control four possible states are selectable at the driver output. It should be emphasized here that the pulse strings available for programming and verifying of PAL are all directly controlled by program except for the following 4 key points:

a) - The width of the "0" pulse in programming which is adjusted to 20 us, the typical value.

b) - The width of the "Vcc" pulse in programming which is adjusted to 30 us, the optimum value.

c) - The delay of the "0" pulse to the "Vcc" pulse in programming is 5 us which ensures that the 0 pulse is fully enclosed by the Vcc pulse.

d) - The width of clock pulse in verifying, which is adjusted to 600 us.

These fixed (adjustable in hardware) timing relations are very important. The programmer can still work properly even if the control program or the computer itself is changed. No time dependent pulse generating routines have to be written.

Thus, the time flexibility by software and fixed timing by hardware are combined succesfully, ensuring the reliability and stability of the PAL programmer.

In the following we will explain these drivers in more detail.

Driver block A contains one driver which delivers Vcc as shown in the schematics of Fig. 15.

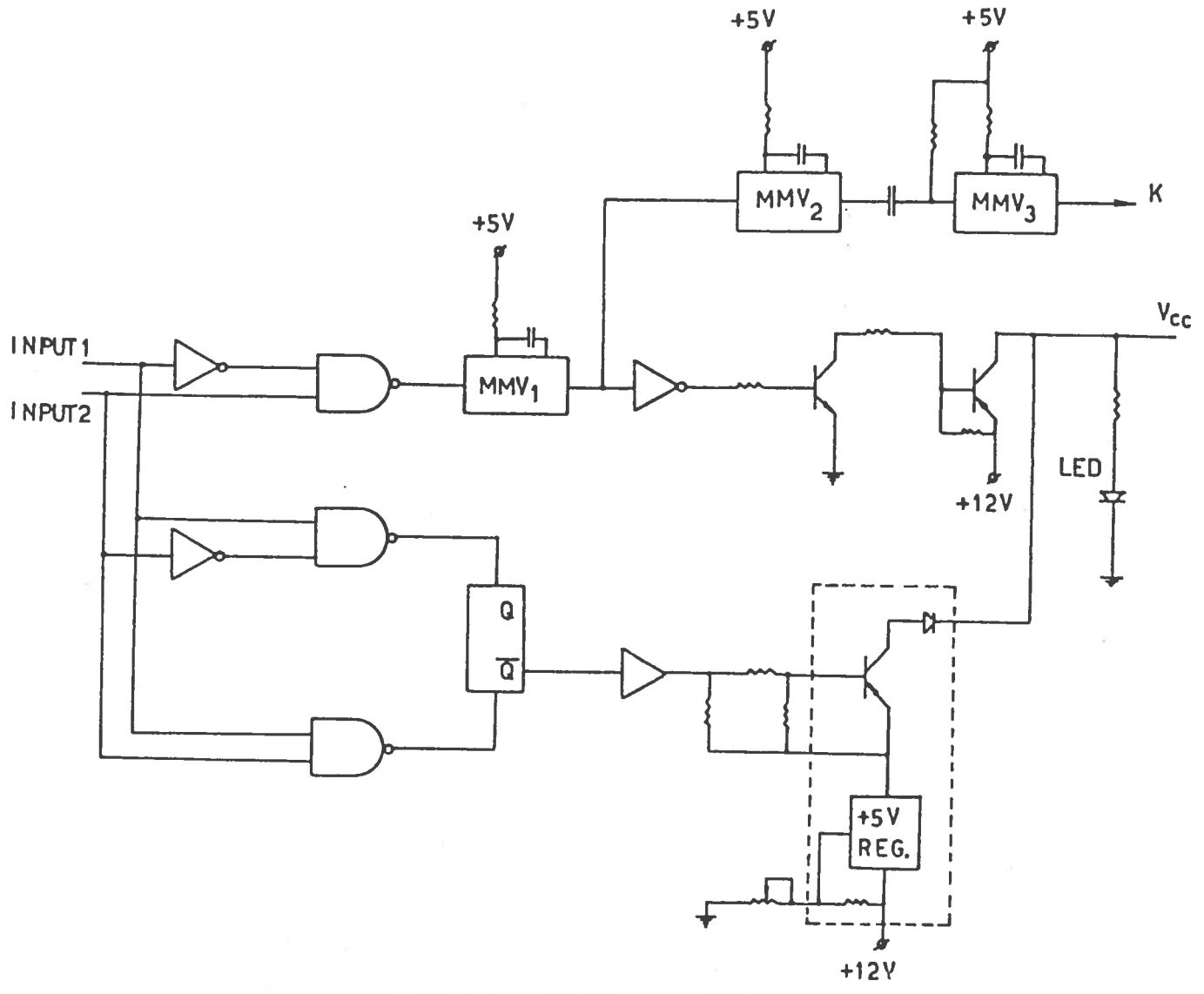

Figure 15. SCHEMATIC DIAGRAM OF DRIVER 'A'.

Driver A has four possible states, as shown in Table 5.

INPUT1	INPUT2	OUTPUT
L	L	high impedance
L	H	generates VCC pulse for programming
H	L	turn on +5V power supply
H	H	turn off +5V power supply

Table 5. STATES OF DRIVER 'A' (L = LOW TTL LEVEL; H = HIGH TTL LEVEL).

In Fig. 15 MMV1, MMV2 and MMV3 are used to yield the "0" pulse and the "Vcc" pulse of fixed width and fixed delay in PAL programming. The output K is connected to the input of the driver E.

In order to ensure an exact +5 Volt power supply, we make use of a regulator.

From Fig. 11, the "Vcc" pulse waveform in programming goes from +5 Volt to V_{IHH} (not from 0 Volt). A flip-flop register maintains +5Volt unchanged even during pulsing.

Because driver A offers max 900mA program supply current, and the regulator works all the time, the dissipation is rather high. Thus we have mounted part of the circuit in Fig. 15 (the part enclosed with dotted lines) on a radiator outside the box.

The LED in Fig. 15 is the indicator mounted on the box front panel (lighted when +5 Volt is on).

Driver block B contains one driver, which generates clock pulses in verifying and mantains 0 Volt during programming. Fig. 16 shows its schematics.

INPUT1	INPUT2	OUTPUT
L	L	generates clock pulse for verifying
L	H	H
H	L	L
H	H	GND

table 6. STATES OF DRIVER B (L = LOW TTL LEVEL; H = HIGH TTL LEVEL).

Driver B has four possible states, as shown in Table 6.

In Fig. 16 MMV is used to yield the clock pulse with fixed width in verifying.

In order to ensure exact 0 Volt in programming we use a reed-relay.

The three-state gate in fig. 16 is used to isolate the MVV output from transistors.

Driver block C contains one driver, which generates L/R pulse in both programming and verifying. Driver block D contains three drivers, which generate A0-A2 in both programming and verifying. Driver block F contains 10 drivers, which generate I0-I9 in both programming and verifying. Driver block G contains one driver, which generates the OD high voltage in both programming and verifying.

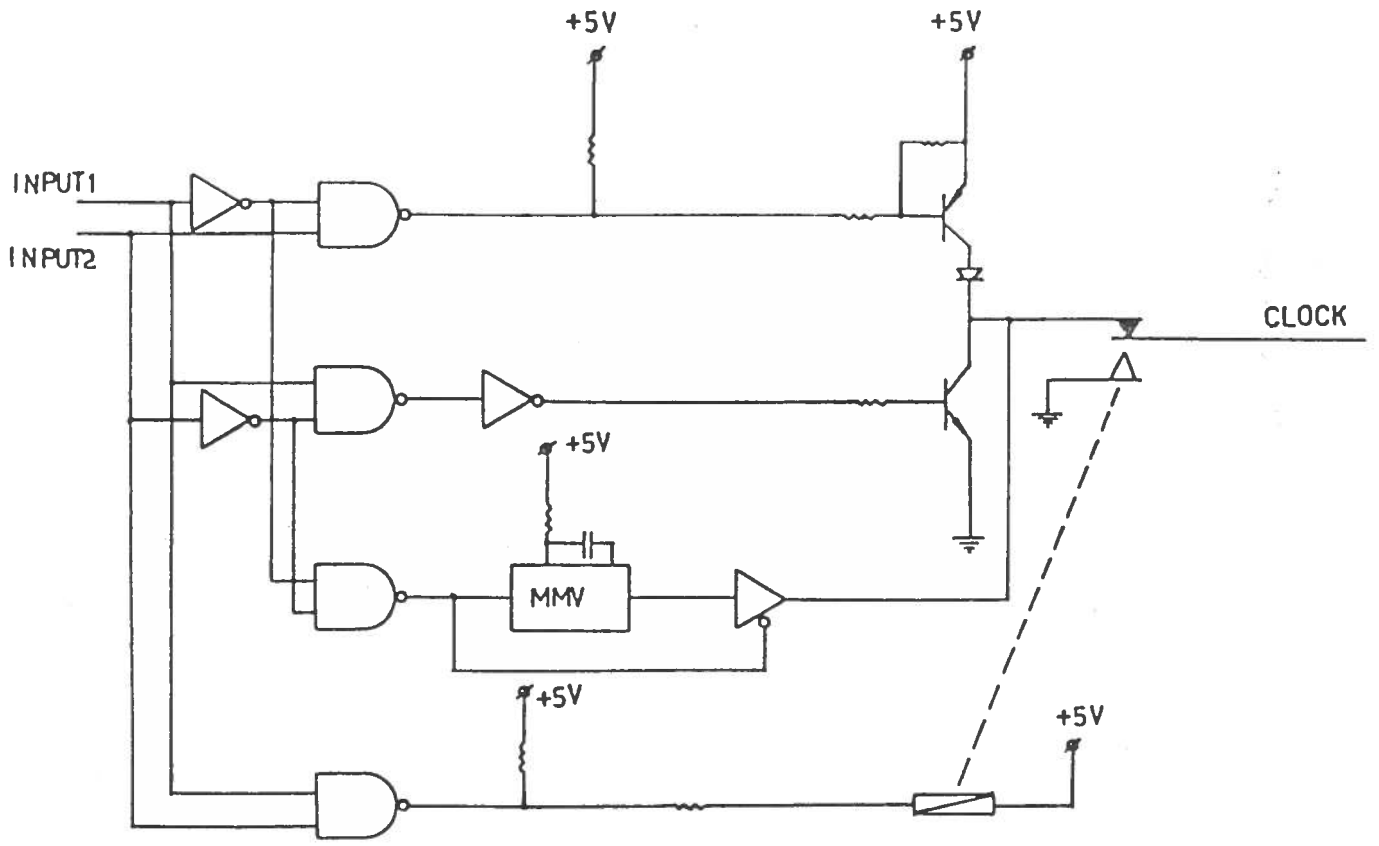

Figure 16. SCHEMATIC DIAGRAM OF DRIVER B

Fig. 17 shows the schematic of drivers C, D, F and G.

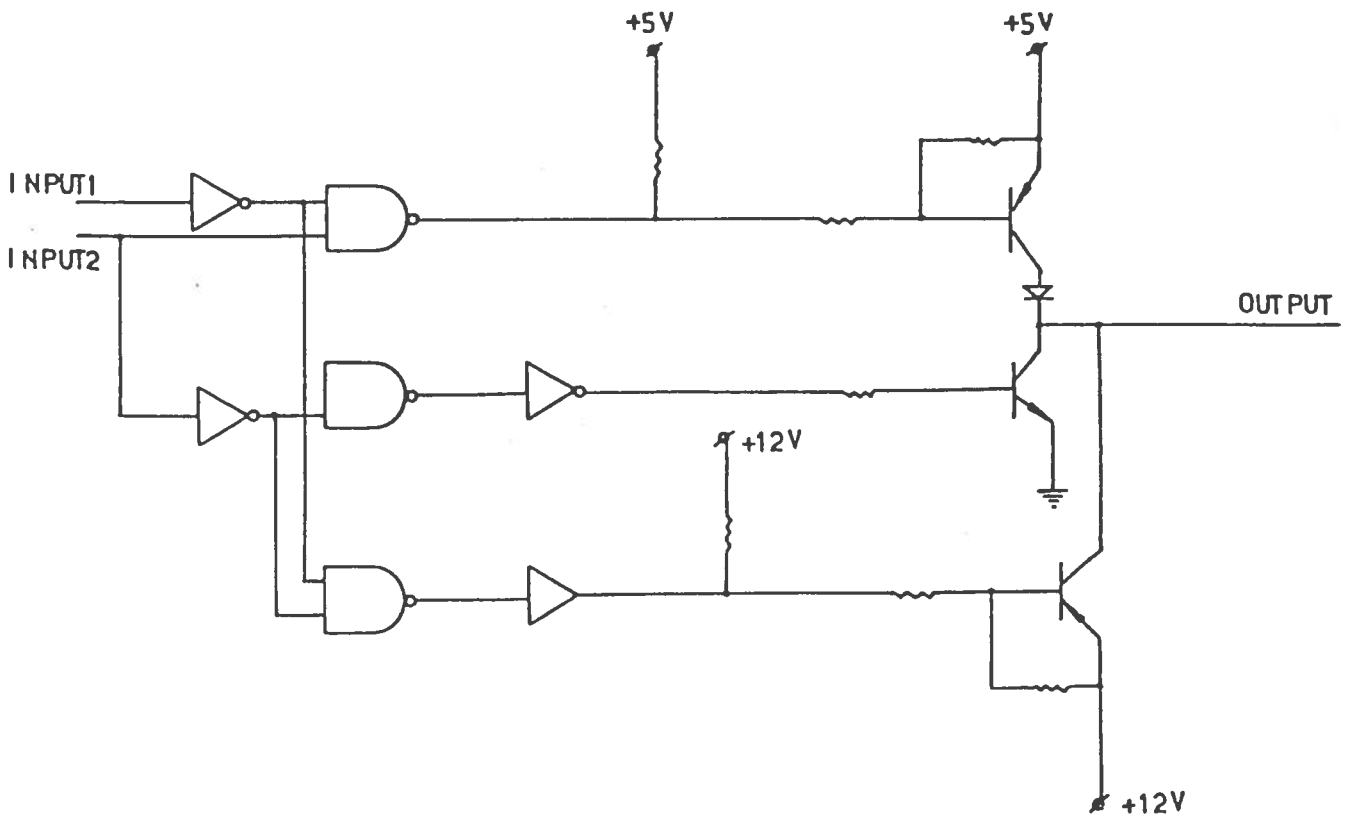

Figure 17. SCHEMATIC DIAGRAM OF DRIVERS C, D, F AND G.

Drivers C, D, F and G have four possible states.

INPUT1	INPUT2	OUTPUT
L	L	high impedance
L	H	H
H	L	L
H	H	VIHH

Table 7. STATES OF DRIVERS C, D, F AND G (L = LOW TTL LEVEL; H = HIGH TTL LEVEL).

Driver block E contains 5 drivers, which generate the "0" pulse in programming.

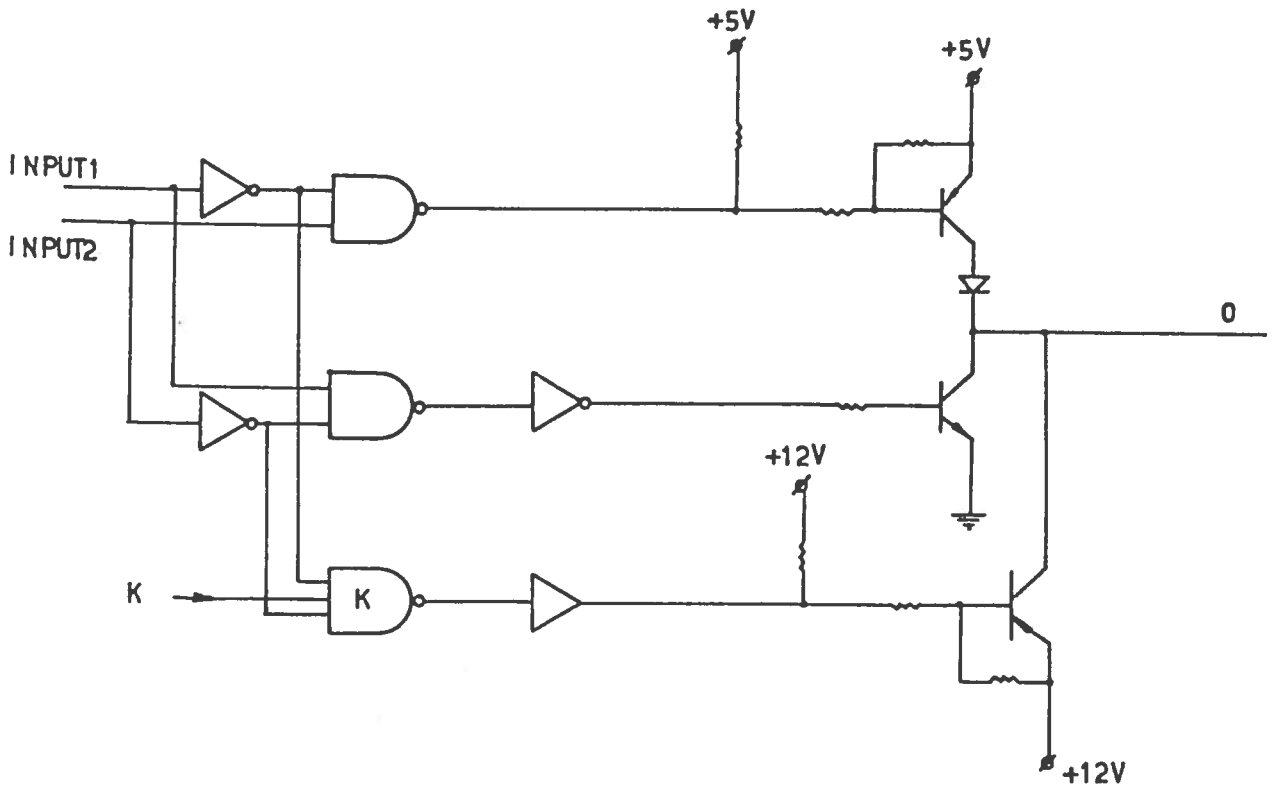

Figure 18. SCHEMATIC DIAGRAM OF DRIVER E.

Driver E has four possible states, as shown in Table 8.

INPUT1	INPUT2	OUTPUT
L	L	high impedance
L	H	H
H	L	L
H	H	keeps gate K opened for 0 pulse

Table 8. STATES OF DRIVER E (L = LOW TTL LEVEL; H = HIGH TTL LEVEL).

The microcomputer sees the PAL programmer as a peripheral device and controls it throgh the interface board. At present, the interface board is compatible with STD-BUS, but it could be easily modified to adapt to any other bus.

From Fig. 13, the interface board mainly consists of the following four parts:

a) - Decoder which can be used to select any contiguous 16-address block in 8-bit port-adressing (now it resides at 20H through 2FH).

b) - Buffers for the control-lines and for the 8-bit data-lines for the STD-BUS.

c) - Output buffers for driving the programming current generators.

d) - Input buffers with gate. During verifying, O0-O4 are used as inputs of PAL, so it's necessary to separate the input buffers from the output drivers.

4. SOFTWARE OF PAL PROGRAMMER.

The control program of PAL programmer is written in PASCAL, running under the CP/M operating system. Howvever, because it's essentially a stand alone routine, the software of the PAL programmer could fit other operating systems with only a few modifications.

Under program control, the PAL programmer can be used to program 20 pin and 24 pin standard PALs with the following 5 functions:

KEY IN - The user can key in the programming pattern from the keyboard;

LOAD IN - The pattern of a written PAL is loaded into the buffer.

CHECK - Checks the pattern in buffer before programming;

PROGRAM - Programs the PAL according to the pattern already in buffer;

VERIFY - Compares the written pattern in PAL with the contents in buffer.

The block diagram of the program is shown in Fig. 19.

The first feature of the software is to ensure security of operation. Immediately after power-on all driver outputs are in high impedance or grounded without software intervention. This is the so-called initialization state. The software makes the PAL programmer return to the initialization state after every operation, so that one can insert and extract PALs freely before or after operating without any danger (the LED indicator in the initialization state is OFF).

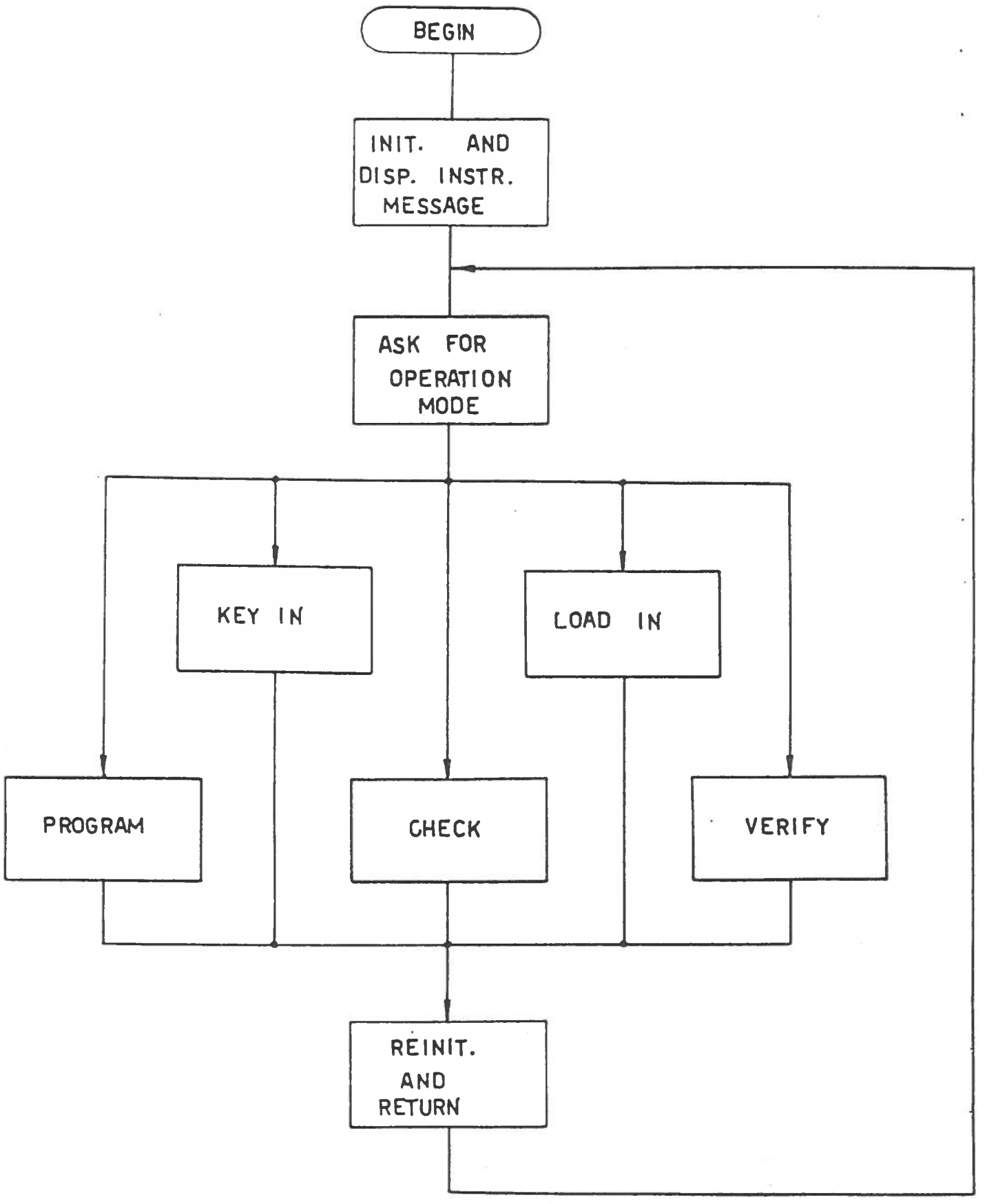

Figure 19. SOFTWARE BLOCK DIAGRAM.

The second feature of the Software is that of being convenient and easy to use. All operations are prompted. One can operate the PAL programmer without any training, just by doing what the instructions displayed on the screen tells one to do.

In the following we will explain the five commands in more detail. In order to enter the corresponding routine the first letter of the command must be typed.

KEY IN - In this command, one can type the product line number succeded by the input line numbers according to the predesigned pattern; the product line number followed by ":", the input line number followed by ",".

In addition, in the command KEY IN we have three more subcommands:

A - Clear all patterns in buffer;

nD - Delete pruduct line n;

nL - Load all input line numbers to product line n.

CHECK - In this command, one can type any key to check four product lines. When the line number exceeds the last, it will return to the beginning.

LOAD IN - This command loads the contents of a programmed PAL into the buffer.

PROGRAM - This command programs the PAL according to the pattern in the buffer. Using the commands LOAD IN and PROGRAM we can copy PALs easily.

VERIFY - Compares the contents in PAL with the pattern in the buffer automatically. If there is any mismatching, there will be some message displayed on the screen indicating the incorrect locations. One can press any key to go on.

The "ESCAPE" key can be used to exit from all five commands.

During programming and verifying the pin assignment of the PAL must be changed for different parts of the programming routine. Thus we have two sockets for 20 pin PALs and two sockets for 24 pin PALs (see Fig. 9 and Fig. 10). During operation, one should change the PAL from one socket to another according to the intructions displayed on the screen.

ACKNOWLEDGEMENTS

GAO ZHONG-REN has carried out this work with the support of a grant from the International Centre for Theoretical Physics (ICTP) Program for training and Research in Italian Laboratories, Trieste, Italy. He would also like to express his gratitude to Prof. Abdus Salam, Director of ICTP, Prof. G. Furlan, Head of the Programme, and Prof. G. Denardo for

providing him the opportunity to do research in the field of microprocessors. A special thanks to Prof.sa A. Marzari Chiesa and Prof. S. Costa, Director of the Istituto di Fisica Superiore dell'Universita' di Torino, for providing the hospitality. The authors gratefully acknowledge Prof. A. Werbrouck for suggestions, criticism and proofreading of the guide.

REGISTERED TRADEMARKS.

PAL, PALASM and HAL are "MMI"registered trademarks.

REFERENCES.

(1) - Programmers and Development Systems are available from the following vendors:

Data I/O, Structured Design, STAG, DIGELEC, PROLOG, KONTRON.