PO-I1600-0000-A

PRINCIPLES OF OPERATION

FOR

ILLIAC IV PROCESSING ELEMENT MEMORY

BY

THEOFANIS ECONOMIDIS

APRIL 1973

INSTITUTE FOR ADVANCED COMPUTATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

continued...

LIST OF TABLES

SECTION 1

INTRODUCTION

The Processing Element Memory (PEM), which is a subunit of the ILLIAC IV Processing Unit (PU), is used to store 2048 words, each of which is 64 bits long. It consists of four memory boards, one control board, one power board, one power base, and one signal base.

Each memory board is organized into eight substacks, each of which consists of 16 Fairchild MμL 4100 integrated circuit (IC) packages. Each MμL 4100 contains 256 storage cells, which are organized as 256 one-bit words. To access a full memory word (64 bits), one substack on each board is accessed. Write operations affecting only 32 bits of a storage location are also possible; these are carried out by selecting substacks on only two memory boards instead of all four.

The control board provides binary address decoding and all necessary control and timing signals for operation of the PEM. The power board and base are used to provide the necessary power to the PEM; this power originates in external power supplies. The signal base provides all of the signal interconnections between the PEM and the MLU.

PEM activity is limited to read and write operations. Data that is written into the PEM may originate at the Control Unit (CU), Processing Element (PE), or Input/Output Subsystem (IOSS). CU write operations are carried out through the PE. Data read from the PEM may be sent to the Control Unit Buffer (CUB), PE, or IOSS. All data movement between the PEM and these other units is carried out through the Memory Logic Unit (MLU). The MLU also provides the PEM with the memory initiate, read/write controls, and address information used by the PEM to execute a read or write operation. Figure 1-1 illustrates the functional relationship between the PEM and the MLU, CU, CUB, PE, and IOSS. As shown by this block diagram, the PEM interfaces solely with the MLU.

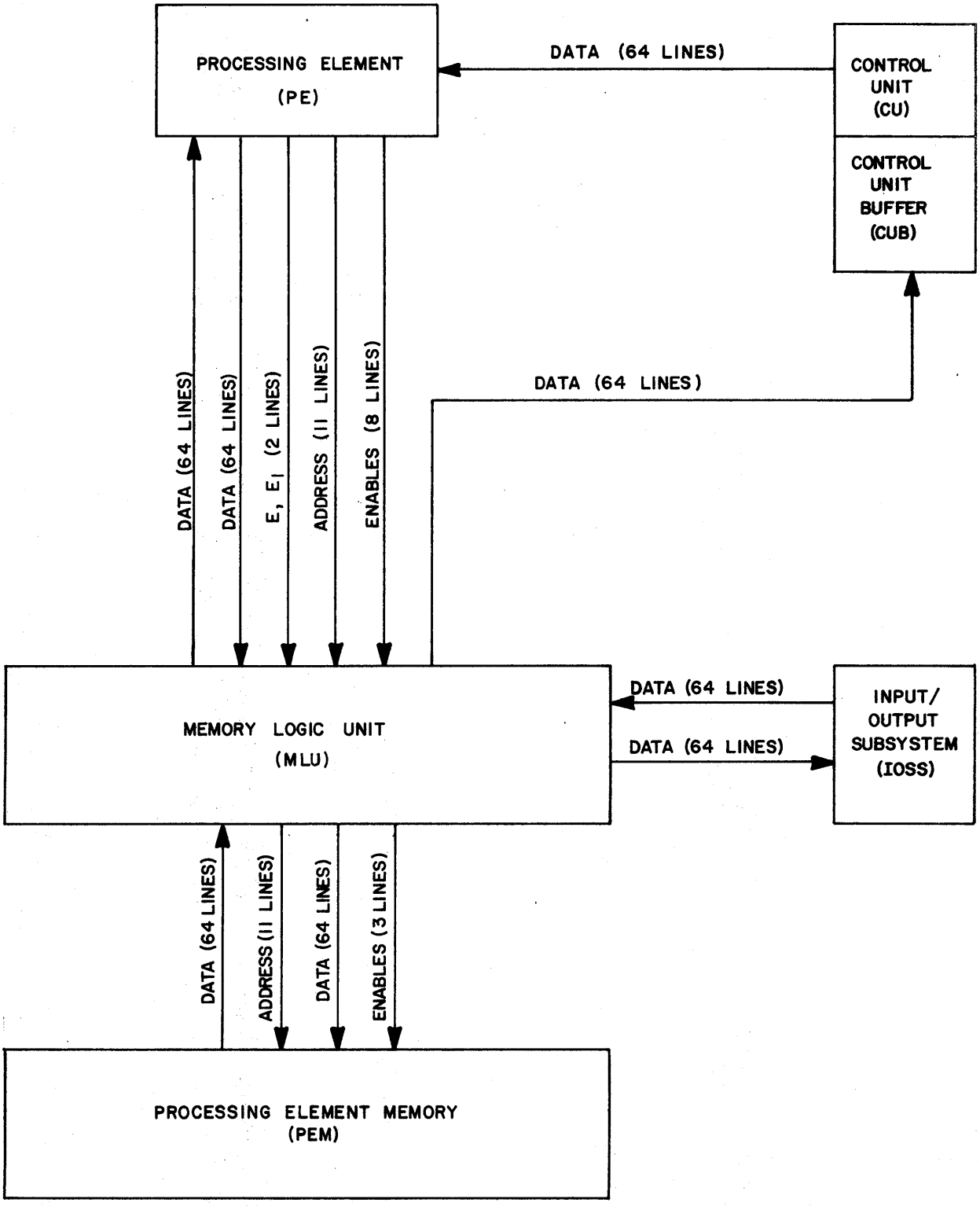


Figure 1-1. PEM Functional Interface

The PEM-to-MLU interface consists of 64 data lines into the PEM (for write data), 64 data lines out of the PEM (for read data), 11 address lines into the PEM, and one memory initiate line and two write control lines into the PEM (enables).

The 11 address lines are used to access one of the 2048 memory locations in the PEM. These lines are derived, via the PE and MLU, from bits 7 through 17 (PEM subaddress field) of the Memory Service Unit (MSU) configuration control registers (MC1 and MC2), which are part of the CU. Details regarding the origin and routing of this address information are provided in subsection 3.1.1 of this manual as well as in the ILLIAC IV Systems Characteristics and Programming Manual published by Burroughs Corporation.

The memory initiate line (one of three enables) is generated by MLU memory control logic when the MLU receives a memory select signal from the PE. The PEM uses the memory Initiate input to enable its control and timing logic. All PEM timing is referenced to the arrival of the memory Initiate signal. Details regarding the generation of this signal are included in the MLU Principles of Operation Manual.

The two write control inputs are used by the PEM to control its write logic for 32-bit mode or 64-bit mode write operations. One write control input controls the write logic for data bits 00 through 07 and 40 through 63 (designated the outer word) and the other controls the write logic for data bits 08 through 39 (designated the inner word). If both write control inputs are true, the write logic on all four memory boards is enabled, allowing a full 64-bit word to be stored. If only one write control input is true, the write logic on only two of the memory boards is enabled (memory boards 1 and 2 for the inner word or memory boards 3 and 4 for the outer word), allowing only 32 bits to be written. Details regarding the use of the write control signals in the PEM are provided in subsection 3.2.2 of this manual. The generation of these signals by the MLU is discussed in the MLU Principles of Operation Manual.

Address information for read and write memory cycles is provided by the CU through the PE.

Power for the PEM logic circuits is provided by an external power supply via a power supply shunt regulator. There is a separate regulator for each PU.

The PEM is housed as an integral part of its PU. Figure 1-2 illustrates the physical relationship of a PEM to the other major components of its PU, namely a PE, MLU, and power supply shunt regulator.

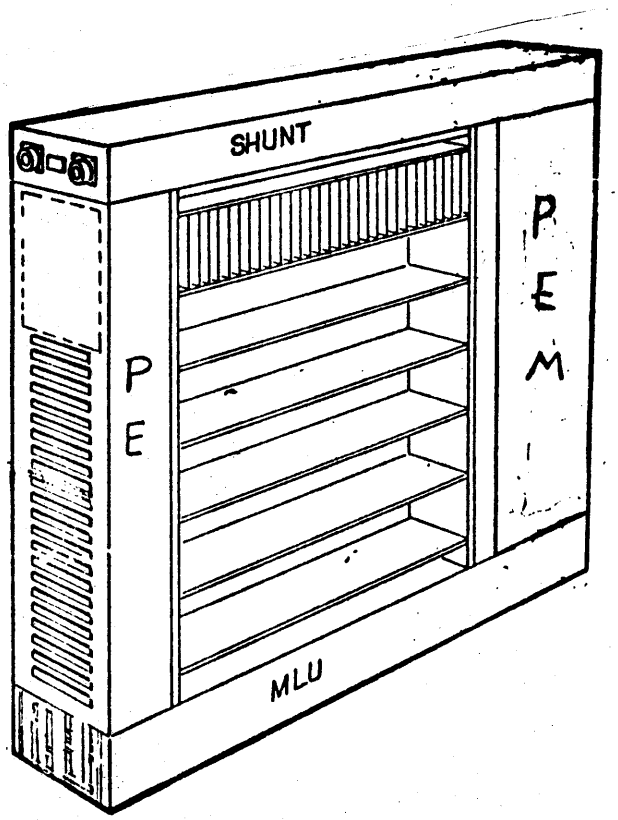


Figure 1-2. ILLIAC IV PU Subunits.

SECTION 2

SUMMARY OF PEM LOGIC CHARACTERISTICS

The PEM uses three families of integrated circuits: Complementary Transistor Micrologic (CTμL), Transistor Transistor Micrologic (TTμL), and Memory Micrologic (MμL).

2.1 CTμL CHARACTERISTICS

The CTμL circuits have the following general characteristics:

a) High Speed. Each logic decision is represented by a square pulse with a typical propagation delay of 3.0 ns (Figure 2-1) and logic levels from -0.5 V to +2.5 V at 25°C ambient temperature.

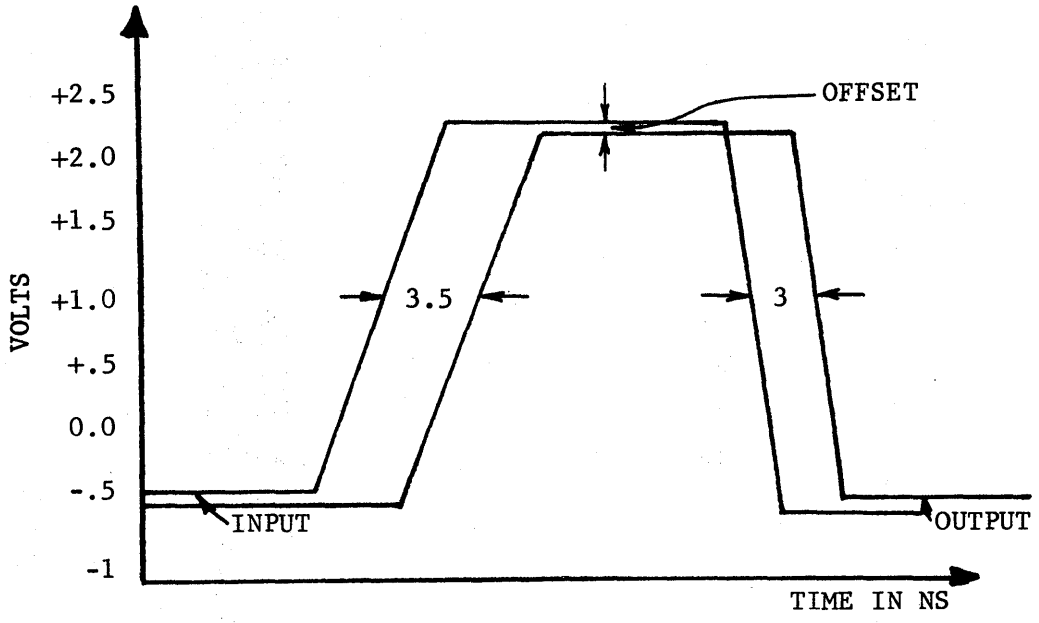


Figure 2-1. CTμL AND-OR Gate Logic Level.

b) Wired-OR. Two or more CTμL outputs may be wired together to form an OR function (Table 2-1).

Table 2-1. CTμL Wired-OR Truth Table

CTμL OUTPUT		OR FUNCTION
1	2	
0	0	0
0	1	1
1	0	1
1	1	1

c) Positive Logic. Since the basic gate employs the complementary PNP and NPN transistors (Figure 2-2), it can be said that the CTμL is a form of current mode logic and retains the flexibility of discrete AND-OR functions (1•1 = 1, 1•0 = 0, 0•0 = 0, where 1 represents a logic high and 0 represents a logic low).

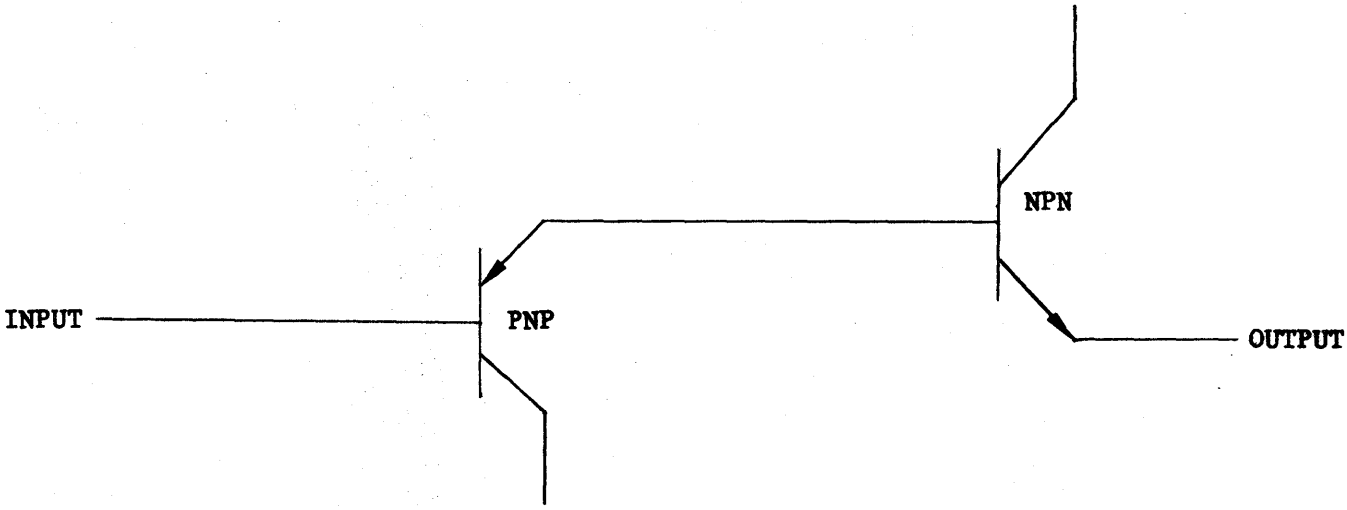


Figure 2-2. Configuration of Transistors Employed in the CTμL Gate.

d) Nonsaturating Circuit. The transistors are biased in such a way that they are always ON, thus eliminating the need to overcome a threshold; this makes it possible for the output to follow the input almost immediately (3 ns delay). The CTμL AND-OR gate can be considered as a noninverting amplifier, whose voltage gain is less than one. Since it is impossible to force the base-emitter voltages of the complementary PNP and NPN transistors to follow all the possible conditions, the output levels are offset (Figure 2-1) from the input. The amount of offset is a function of loading, temperature, and input voltage. A signal passing through a succession of CTμL AND-OR gates is most offset in the first gate, the offset decreasing sequentially on succeeding gates because of the reduced input level. However, in order to preserve sufficient system noise immunity, appropriate noise immunity (voltage) levels must be reestablished at intervals by inserting, once every three CTμL AND-OR gates, a restoring element such as inverter, buffer, or flip-flop (Figure 2-3). The current gain of the CTμL family is considerable, making them especially useful for driving loads involving a considerable amount of input capacitance.

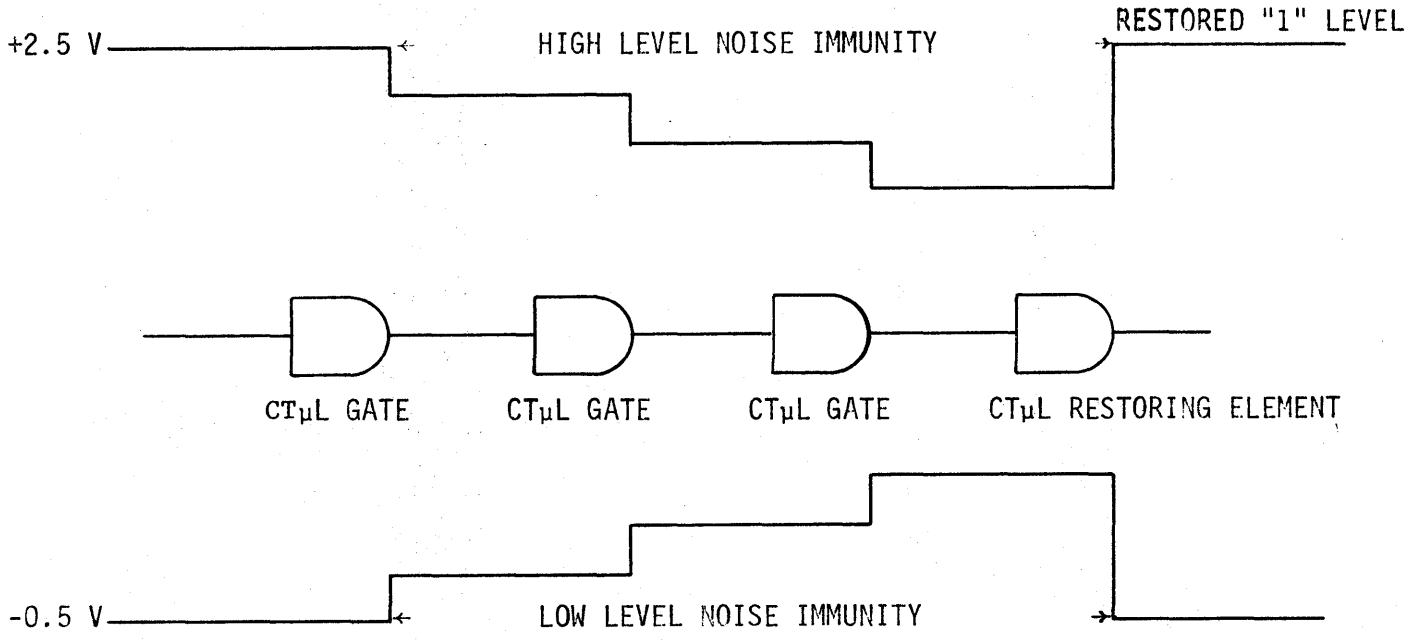


Figure 2-3. Configuration of CTμL AND-OR Gate Offset Level and Noise Immunity.

e) Two Power Supplies. The CTμL gates require two power supplies with voltages:

1. V_{cc} = +4.5 V ±10% applied at pin 12
2. V_{ee} = -2.0 V ±10% applied at pin 11

(Pin 5 is connected to ground.)

CTμL AND-OR gates, functioning on positive logic with a high level of approximately +2.5 V and a low level of about -0.5 V, provide a true (high) output, when all inputs are true (high). If any input is false (low) the output will also be false (low). Table 2-2 shows the logic output of a CTμL AND-OR gate as a function of the inputs. The gate is called AND-OR because its output follows the rules of the logic AND function and it can be tied to any other CTμL element output to perform the wired-OR function (Table 2-1).

Table 2-2. Two-Input CTμL Gate Truth Table

INPUT		OUTPUT
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

The significant feature of this type of logic circuit is the use of both PNP and NPN transistors to provide a noninverted logic function. In the static state, with no connection to inputs A and B (Figure 2-4), resistors R1 and R2 provide a forward current bias path for the base of Q1 sufficient to hold the transistor ON. Q2 will be held ON in the same manner by the base current through R3 and R4. With both PNP transistors thus conducting, a negative potential of approximately -2 V, less the voltage dropped across Q1 and Q2 (in parallel), is applied to the base of Q3. Q3 is thus forward biased just enough to keep it out of the cut-off state.

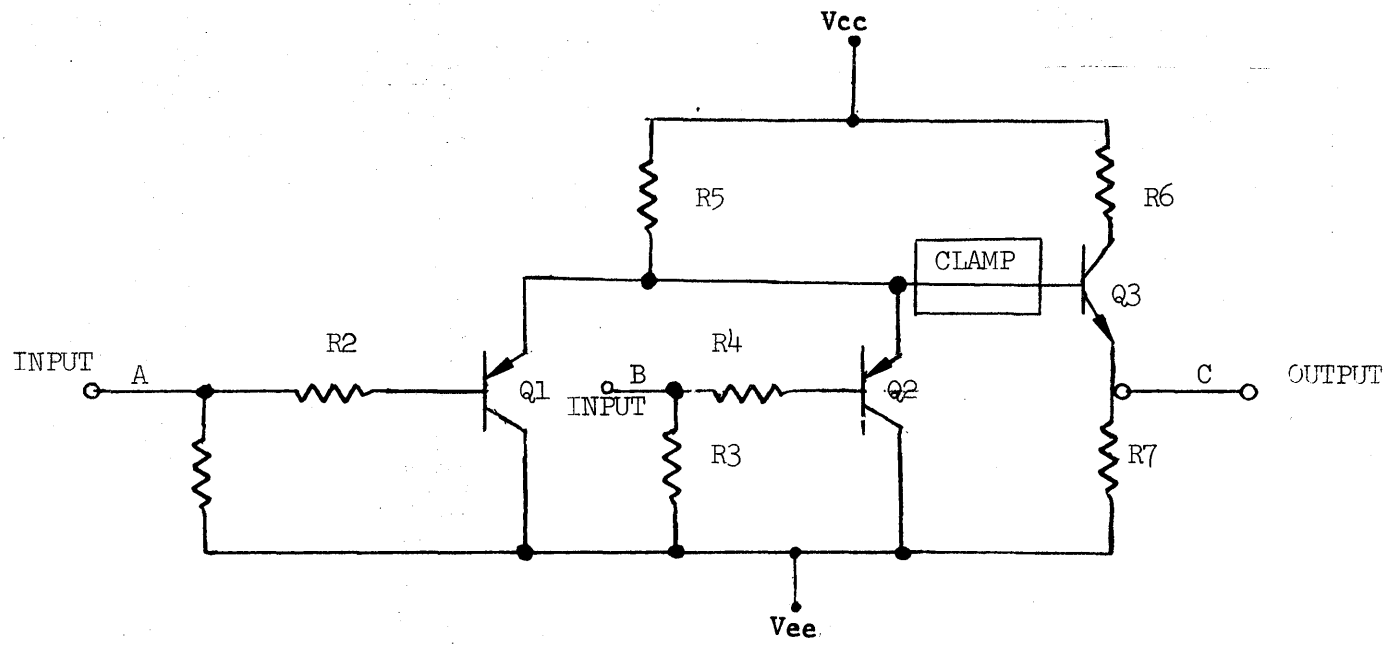


Figure 2-4. Basic Diagram of Two-Input CTμL AND-OR Gate (Fairchild).

As long as either input is left open (held negative), that transistor (Q1 or Q2) will remain in the ON state, thus holding the output at a low level. When positive signals are applied to both inputs, the output will be positive, but it always tends to "follow" whichever of the inputs is the least positive.

Saturation of Q3 by applying a relatively large positive signal to both inputs is prevented by the clamping circuit shown in Figure 2-5. If the signal, applied to the base of the output transistor Q3 (Figure 2-4), exceeds the positive limit determined by the clamp bias divider, the associated emitter diode of Qc1 (Figure 2-5) will begin to conduct, thus drawing off the excess current to keep the base at a potential of -2 V. The output transistor Q3 is thus held between the limits of cut-off and saturation, minimizing switching delays due to storage time.

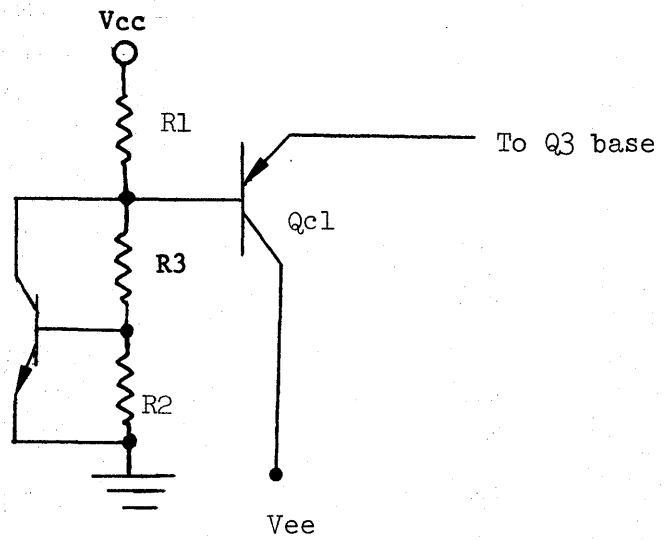


Figure 2-5. CTμL Clamping Circuit.

2.2 TTμL CHARACTERISTICS

The TTμL family of devices provides greater noise immunity. It is also more effective when driving high-capacitance loads, since in both logic states it provides low output impedance. Only one type of TTμL device is used in the PEM: the TTμL 9016.

The TTμL 9016 circuit has the following general characteristics:

a) Inverting Element. The TTμL 9016 package is an inverter. If the input is high the output will be low and vice versa.

b) The input is a current source and the output is a current sink. The input to a TTμL circuit is connected to an emitter of an NPN transistor (Figure 2-6), the base of which is connected to V_{cc}. It therefore acts as a current source in that current will tend to flow out of the input toward ground if a path is provided. The output of a TTμL circuit acts as a current sink by providing a very low impedance path to ground through a saturated transistor. In order to drive a TTμL input with a CTμL output (also a current source), a "current sink" resistor is connected to ground between them (Figure 2-7).

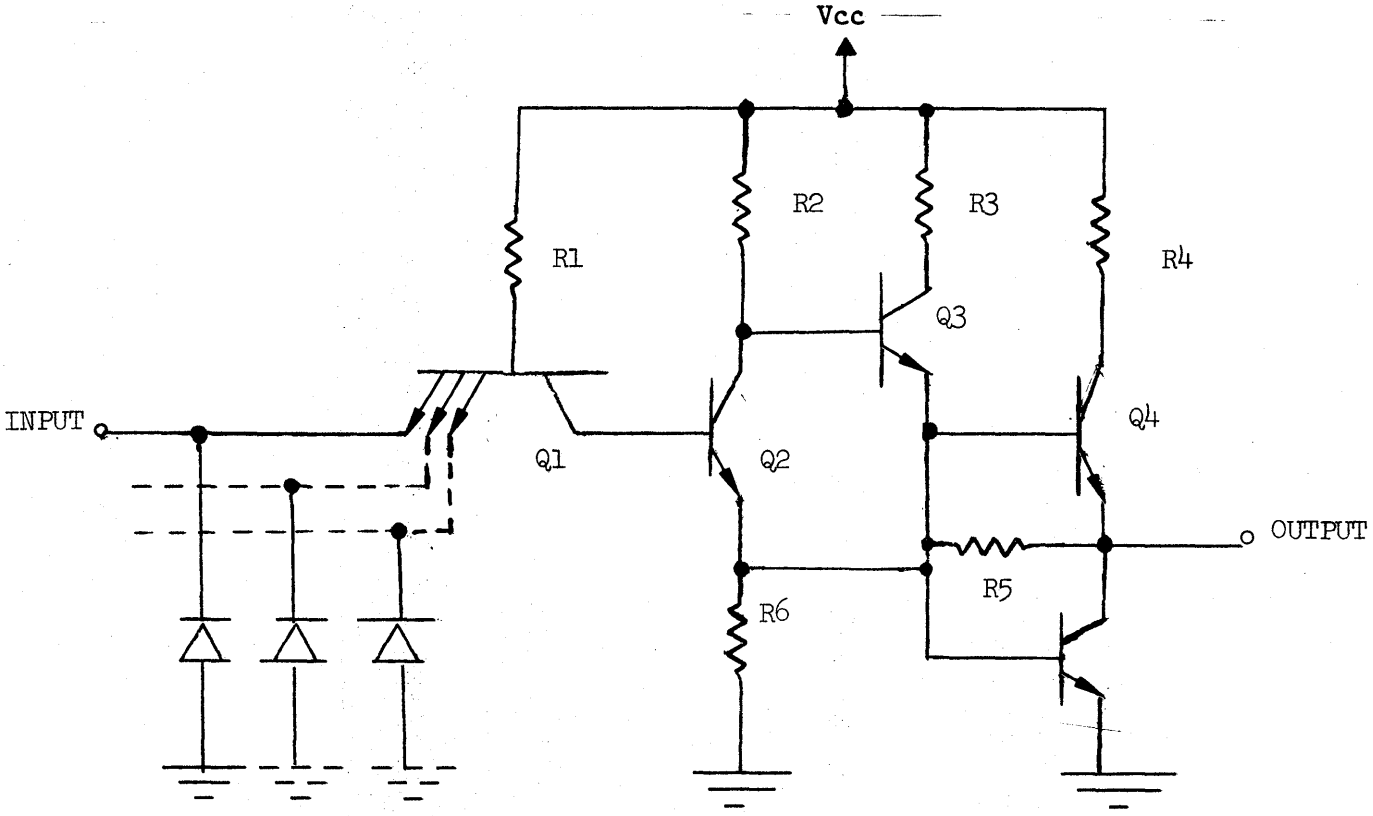


Figure 2-6. TTμL Basic Gate Circuit (Fairchild).

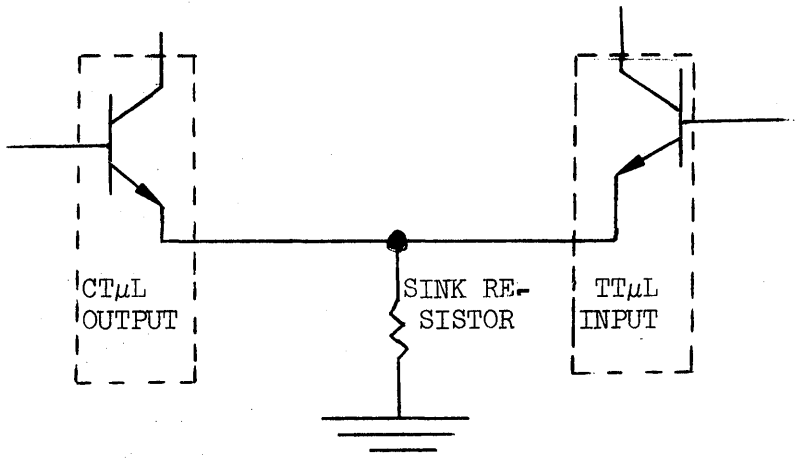


Figure 2-7. Diagram of a CTμL Output Driving a TTμL Input.

When the CTμL output transistor is turned ON (is fully conducting), the impedance presented by it is sufficiently low to provide as much current as this resistor can conduct. The TTμL input, having a much higher impedance, is therefore effectively deprived of its current-sink path and the input voltage level is held high.

When the CTμL output transistor is turned OFF (almost cut off), it presents a relatively higher impedance path to the sink resistor, thereby allowing current to flow from the TTμL input circuit through the resistor to ground.

c) Medium Speed. The TTμL device is relatively slower than the CTμL device for the following reasons:

1. It requires additional time for threshold detection.
2. It is a saturating element in that the output transistor switches completely from the cut-off state to the saturated state. The storage time of the output transistor in either the cut-off or saturated state introduces an extra delay into the response of the output to any change in the input. The total circuit delay amounts to approximately 7 ns per logic decision.

d) Excellent Noise Immunity. The Fairchild TTμL device (9016) provides high noise immunity and has the ability to reshape pulses of marginal amplitude. Figure 2-8 provides information about the characteristics of the output and input of a TTμL device, respectively, and safety margins for noise signals.

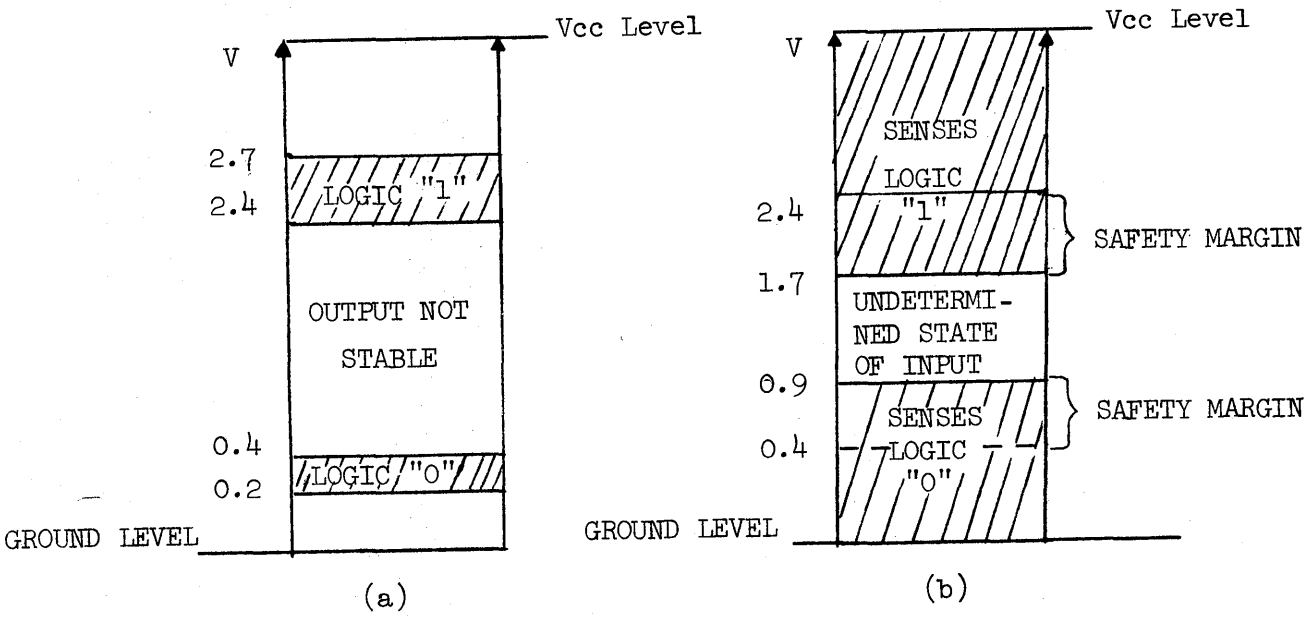


Figure 2-8. TTμL Logic Level Output (a), Input (b).

When the signal represented by Figure 2-8(a) becomes an input to a TTμL device, the safety margin due to noise pulse amounts to about 0.7 V for logic ONE and 0.5 V for logic ZERO, as shown in Figure 2-8(b).

2.3 MμL 4100 CHARACTERISTICS

The nucleus of the PEM is the memory package MμL 4100 (Figure 2-9) built by Fairchild Camera and Instrument Corporation. The package can provide 256 bits organized as a 16×16 array of flip-flops [1][1]. This array and associated read/write circuits are shown in Figure 2-10.

The basic cell of the memory chip (Figure 2-11) consists of a flip-flop. One of the transistors in the flip-flop is ON (conducting) while the other one is forced to remain in the OFF state (not conducting). When the "OFF" transistor is forced to conduct by an external signal, the transistor which is already in the ON state returns fo the OFF state. Therefore, the flip-flop can be in either one of two states (ON or OFF); it will remain in one of the states until an external signal causes it to change states.

The flip-flops can be used to store logic ONE's or ZERO's. If T^t represents the value of the external signal that changes the state of the flip-flop at time t, then the state of the flip-flop at time t+1 is given by the equation:

$$X^{t+1} = T^{-t} \cdot X^t + T^t \cdot X^{-t}$$

which is an exclusive OR, as shown by the truth table of Table 2-3.

Table 2-3. Truth Table of Trigger Flip-Flop

T^t	X^t	X^{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

[1] A number in square brackets refers to a reference document listed at the end of this report.

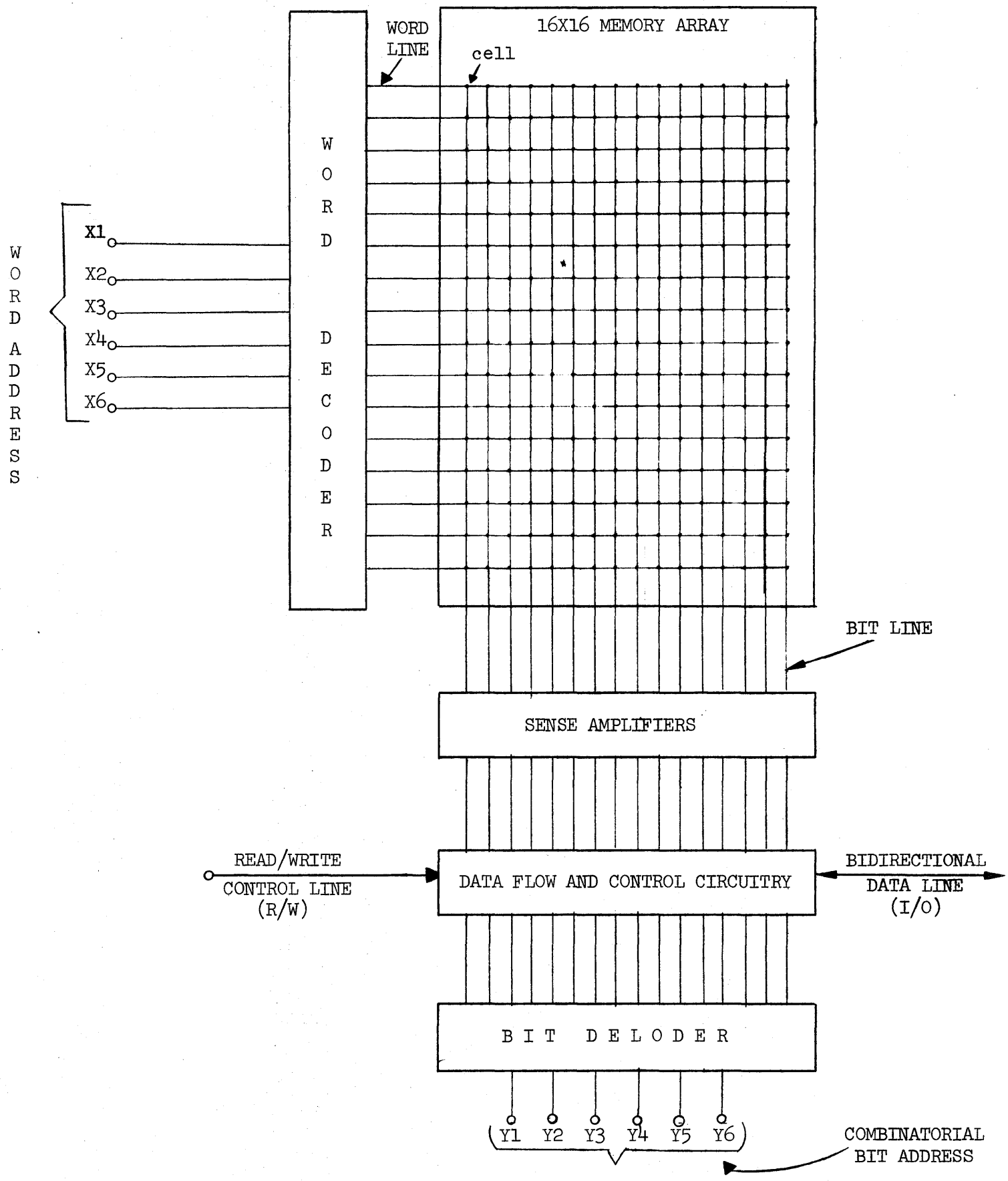


Figure 2-9. Block Diagram of MμL4100

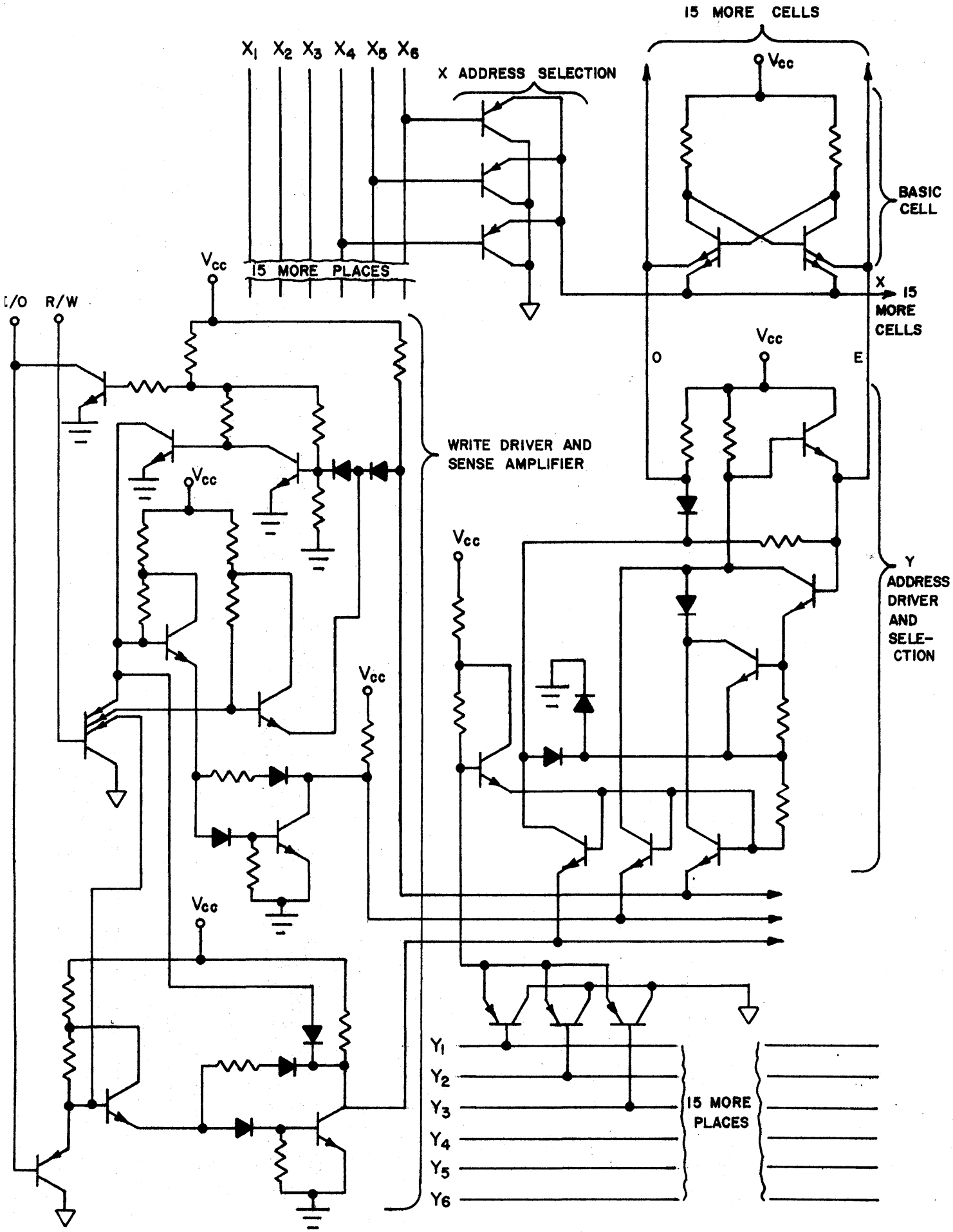


Figure 2-10. Configuration of MμL 4100.

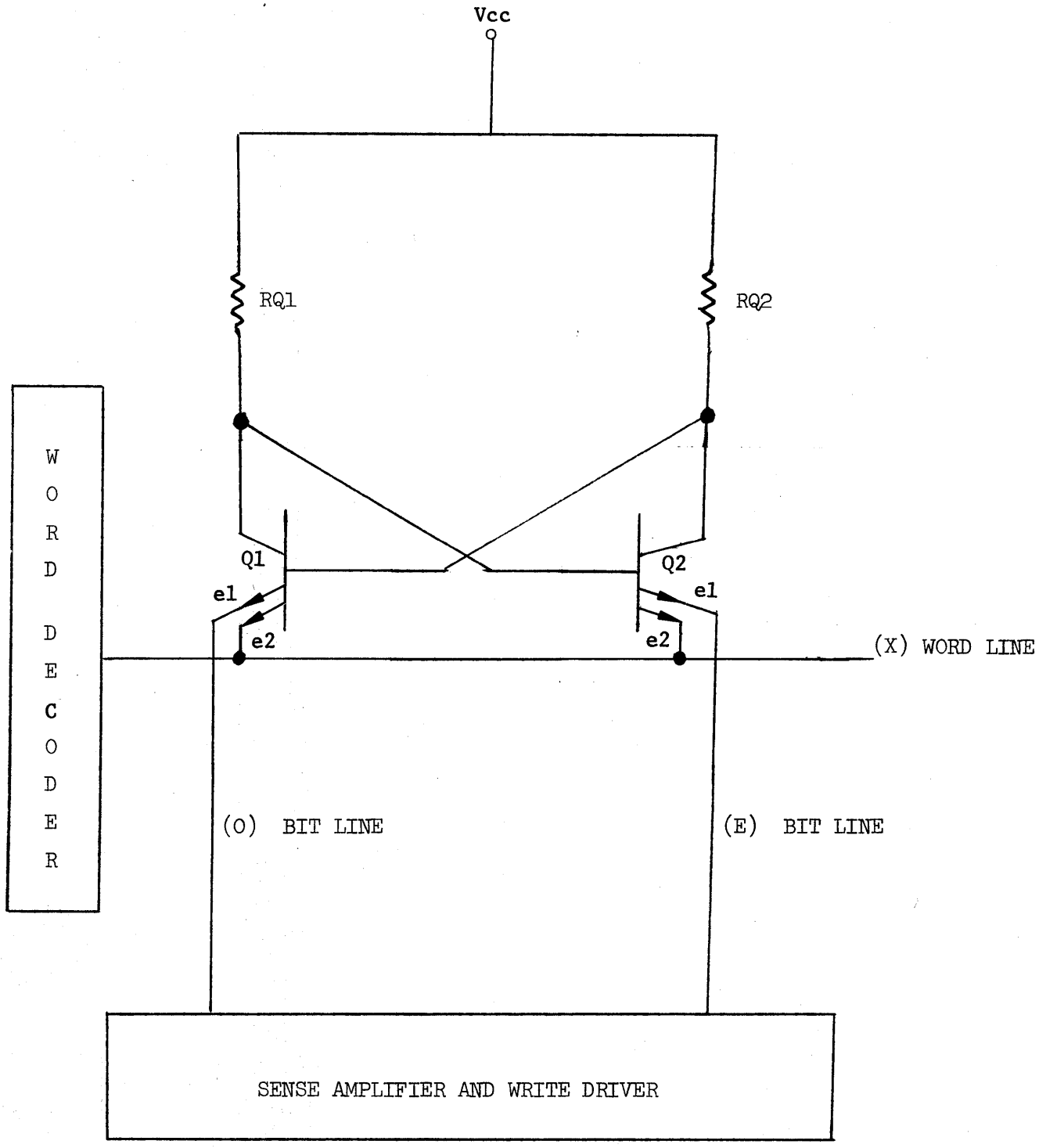


Figure 2-11. Basic Cell of MμL4100 (Fairchild)

In order to select one of the 256 memory cells for a read or write operation, the X_i (i = 1, 2, 3, 4, 5, 6) and Y_j (j = 1, 2, 3, 4, 5, 6) address lines are enabled simultaneously. These lines operate on the "3-out-of-6 code" (combinatorial address); this code is discussed in subsection 3.1.1.

The X_i address lines select one of the 16 word lines inside the memory package, while the Y_j address lines select the appropriate write driver on a write cycle or the appropriate sense amplifier on a read cycle. The read/write control line controls the direction of the data flow.

NOTE To avoid any confusion, it must be understood that the term OFF as used in subsequent discussion means that the transistor under discussion is in a very low state of conduction compared to that of the ON transistor.

Write Logic ONE

When the memory cell is selected, line X (Figure 2-11) is raised. In this case E is also raised and both emitters of Q2 (Q2e1 and Q2e2) become positive; consequently, conduction through Q2 decreases. Thus the potential at the base of Q1 rises and Q1 starts conducting more heavily. This reduces the amount of current available to the base of Q2, because the impedance presented by the collector of Q1 is less than that presented by the base of Q2. At this point, line 0 is more negative than line E and current flows through the emitter Q1e1 to line 0. Transistor Q1 will remain in the ON state, and in this state the cell is regarded as containing a logic ONE.

Write Logic ZERO

To write a logic ZERO in the cell discussed above, line X must also go high. This time, however, line 0 is more positive than line E and Q2 is turned ON. This forces Q1 to be turned OFF.

READ

When the cell is to be read, only line X becomes positive and thus any current which is going to flow is now flowing through Q1e1 and Q2e1. A sense amplifier (Figure 2-11) determines which one of the two lines (0,E) draws more current and therefore detects the state of Q1 and Q2. In the first case discussed, where a logic ONE was stored and Q1 is in

the ON state, line 0 is more positive and the sense amplifier will detect a logic ONE. In the second case, line E is more positive and a logic ZERO is detected.

Unselected Cell

When a cell is not selected either for reading from or writing into, lines X and E become negative. Current flows through Q1e1, Q1e2, Q2e1, and Q2e2, but because at the instant that lines X and E became negative one of the transistors was conducting more heavily than the other, and therefore the base of the latter was deprived of sufficient current to make it conduct as heavily as the former, it is held in a less conductive state than that of the former. Conversely, the low current path thus provided through the collector of the latter permits a relatively large base current to flow through the former, thereby holding this transistor in the ON state and the other in the OFF state.

* * *

The MμL 4100 is a high speed, LSI, bipolar 256-bit, read/write random access, nondestructive readout memory device, hermetically sealed in a 16-lead, dual in-line package (Figure 2-12). Its basic characteristics are:

a) TTμL Compatible. The MμL 4100 can be driven directly by a TTμL device.

b) Maximum Read Access Time. Time from X_i (i = 1, 2, 3, 4, 5, 6) address at the pins of MμL 4100 until the data is available at I/O pin of MμL 4100 is about 100 ns.

c) The Y_j (j = 1, 2, 3, 4, 5, 6) address is delayed from the X_i (i = 1, 2, 3, 4, 5, 6) address by approximately 35-45 ns.

d) The read/write pulse is required to be present only during the write operation, while during the read operation it is just a low level.

e) The write pulse is required to precede the Y_j (j = 1, 2, 3, 4, 5, 6) address by at least 10 ns.

f) The write pulse may be removed about 40 ns prior to removing the X_i and X_j addresses from MμL 4100.

g) Output Wired-OR Capability. Figure 2-10 shows that the I/O line of MμL 4100 is connected to an uncommitted collector. This feature allows the collectors (I/O lines) of several MμL 4100's to be "OR-tied", thereby providing for word expansion.

h) Partial Decoding on Chip. The binary address is converted into the combinatorial address, which is decoded inside the MμL 4100 in order to select the proper word and bit lines.

i) Power Requirements. V_{cc} = +4.8 V ±10% and ground.

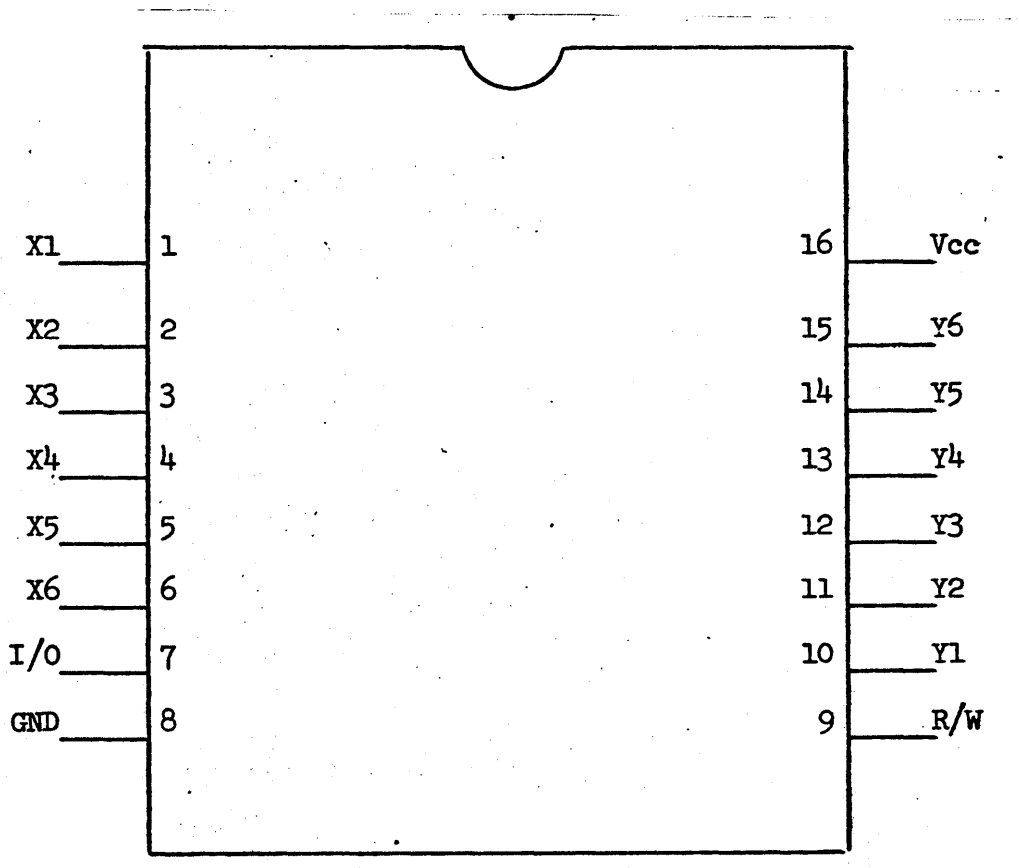
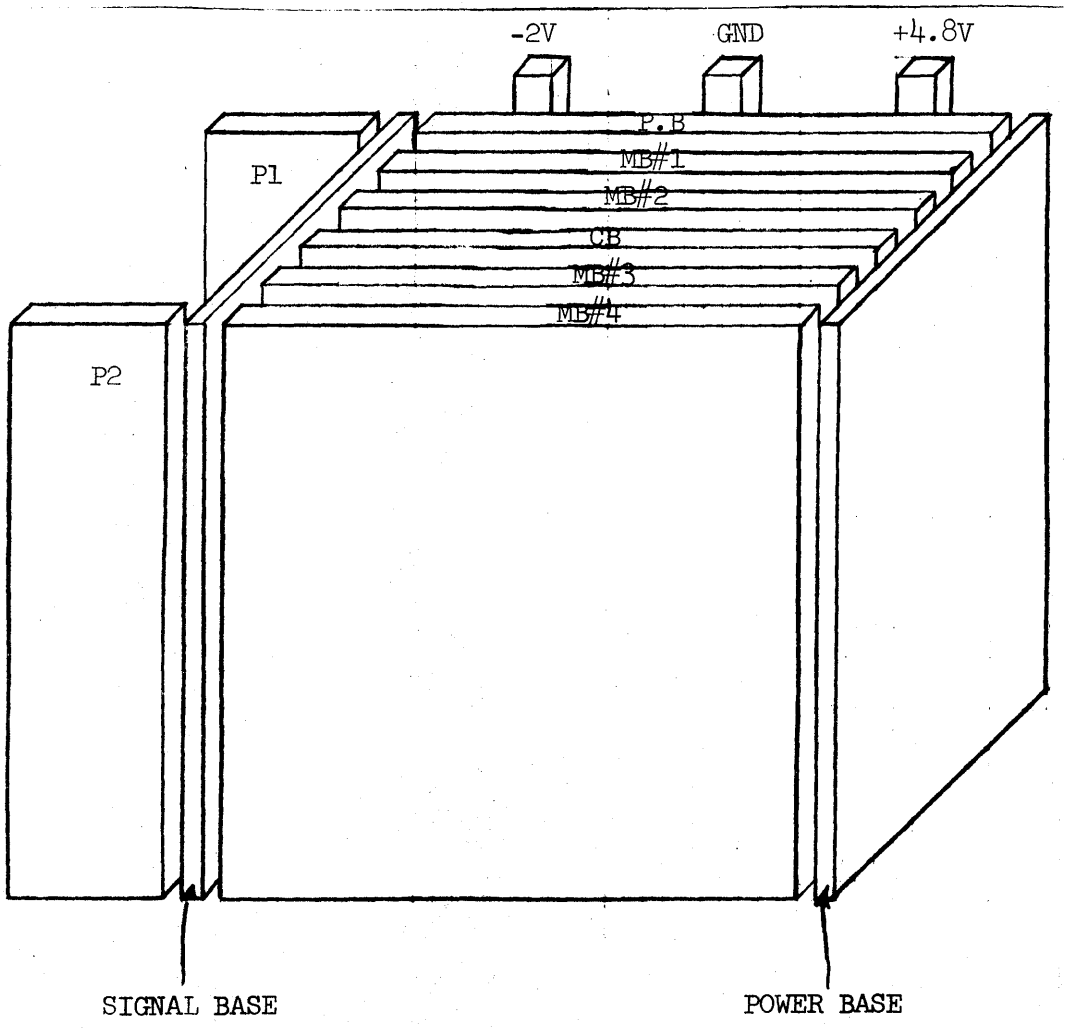


Figure 2-12. MμL 4100 Pin Configuration.

SECTION 3

PEM ORGANIZATION

Figure 3-1 shows the physical relationship of the four memory boards, one control board, one power board, one power base, and one signal base in each PEM. The two paddle boards that are connected to the signal base are also shown. The functional responsibilities of the principal logic contained on these boards are discussed in the following subsection.



MB#1 = MEMORY BOARD #1
MB#2 = MEMORY BOARD #2
C.B. = CONTROL BOARD
MB#3 = MEMORY BOARD #3
MB#4 = MEMORY BOARD #4
P.B. = POWER BOARD
P1 = PADDLE BOARD #1
P2 = PADDLE BOARD #2

Figure 3-1. Exploded View of ILLIAC IV PEM.

3.1 CONTROL BOARD

The control board logic provides the control and timing signals needed by other PEM logic to carry out a read or write operation; the logic responsible for this is called the timing generator. Another set of logic on the control board is responsible for translating the 11 bits of binary address received from the MLU into the X and Y components of a combinatorial address. This logic is referred to as the X and Y combinatorial address decoders. Figure 3-2 is a block diagram showing these functional blocks.

3.1.1 Binary Address Decoding

Before examining the control board's address decoding logic, the two address schemes (binary and combinatorial) will be discussed.

PEM Binary Address

Addressing in ILLIAC IV is mainly a function of the Memory Service Unit (MSU) whose users are the Input/Output Subsystem (IOSS), the Final Station (FINST), and the Instruction Look-Ahead (ILA). The FINST and ILA are sections of the control unit. The FINST requests and their purposes are as follows:

a) FINST Own Request: This is for PE read/write operation request.

b) FINST Request A: This is for the Advanced Station (ADVAST) LOAD(X), BIN(X), or STORE(X) request.

c) FINST Request B: This is a request for transfer from PE to ACAR (in the MLU that has been described as a transfer operation from PE to CUB).

The address as it is defined by the MSU Configuration Control registers (MC1 for instruction addresses and MC2 for data addresses) has the format shown in Figure 3-3 at FINST and ICR. Table 3-1 describes the one-quadrant array configuration.

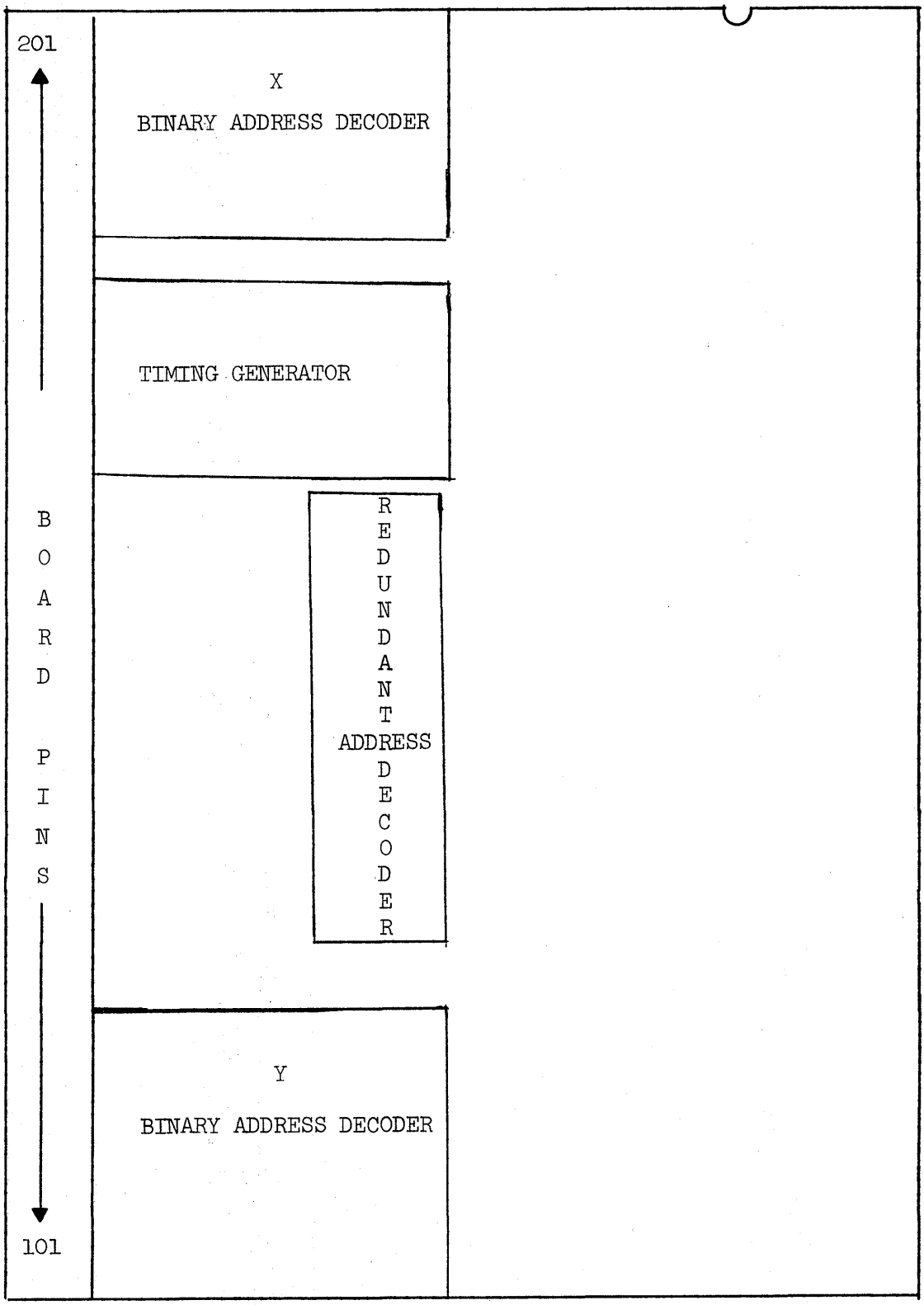


Figure 3-2. ILLIAC IV Control Board Functional Block Location Diagram (Component Side).

2 6	7 17	18 20	21 23	24
FOR FUTURE EXPANSION	PEM SUBADDRESS (11 BITS)	SELECTS THE PEM WITHIN THE P.U.C.	SELECTS THE P.U.C.	HALF WORD DESIGNATOR IN ICR

Figure 3-3. Address Format in a One-Quadrant Array.

Table 3-1. One-Quadrant Array Configuration

	← PROCESSING UNIT CABINET (PUC) →							
	0	1	2	3	4	5	6	7
↑	00	01	02	03	04	05	06	07
	10	11	12	13	14	15	16	17
P	20	21	22	23	24	25	26	27
E	30	31	32	33	34	35	36	37
M	40	41	42	43	44	45	46	47
	50	51	52	53	54	55	56	57
	60	61	62	63	64	65	66	67
	70	71	72	73	74	75	76	77
↓								

For this discussion, bits 7 through 23 are of primary interest. Bit 24 is used only to designate which half of the word is used (when it is ZERO the left half of the word is used) and bits 2 through 6 are for future memory expansion; if four quadrants were used, two extra bits would be needed to define which one of the four quadrants was to be accessed. Table 3-2, for illustration, shows specified address and components selected according to the format of Figure 3-3.

Table 3-2. Example of One-Quadrant Address Format

ADDRESS BITS (7-23) 8	SELECTED COMPONENTS IN THE QUADRANT		
	PUC	PEM	SUBADDRESS
0 0 0 0 0 0	0	0	0 0 0 0
0 0 0 1 1 1	1	1	0 0 0 1
3 7 7 7 7 7	7	7	3 7 7 7

As shown by the array configuration of Table 3-1, there are eight PU cabinets in the quadrant, each one containing eight PEM's. Every PU is assigned a two-digit octal number. The first digit (0-7) specifies the PEM in the PU cabinet and the second digit (0-7) specifies the PU cabinet and the quadrant. PE requests will select eight PUC's and all the PEM's in each PUC. Therefore, only 11 bits are sent on to the PEM (bits 7-17). These 11 bits are called the PEM subaddress.

The PEM subaddress leaves the MSU, passes through the FINST and, using the common data bus (CDB), it reaches the PE memory address register (MAR). The address may be indexed by S or X registers of the PE before it gets into the MAR. From the MAR, the address passes through the UP Converter of the MLU for CTμL compatibility and then is applied to the paddle board of the PEM. On PE and MLU signal lists, address bits A5 through A15 are designated as W05 through W15 (Table 3-3).

Table 3-3. Signal Name Representation

CONTROL UNIT			PROCESSING ELEMENT			MEMORY LOGIC UNIT		
SIGNAL NAME	LEVEL	LOGIC	SIGNAL NAME	LEVEL	LOGIC	SIGNAL NAME	LEVEL	LOGIC
TVW-WXX--0	HIGH	1	PYW-WXX--0	HIGH	0	MYW-WXX--0	HIGH	1
TVW-WXX--0	LOW	0	PYW-WXX--0	LOW	1	MYW-WXX--0	LOW	0
TVW-WXX--1	HIGH	0	PYW-WXX--1	HIGH	1	MYW-WXX--1	HIGH	0
TVW-WXX--1	LOW	1	PYW-WXX--1	LOW	0	MYW-WXX--1	LOW	1

Figure 3-4 shows that each address bit changes name three times until it arrives at the paddle board of the PEM. There is no particular reason for this, other than that the design of the different subunits of ILLIAC IV took place at different times and by different people, all of whom complied with the requirements of the design for 11 bits of address, but assigned names they felt fit best. Apparently the numbering of the subaddress in the CU follows the one-quadrant address format; in PE and MLU it follows the four-quadrant address format, and in the paddle board of the PEM an extra A was added to differentiate the signals inside the PEM from those outside the PEM. This last distinction is called for because, as is shown later, there is a logic redefinition of the address bits inside the PEM. This logic redefinition results from the fact that, while the MLU follows the negative logic convention (Table 3-2), inside the PEM (control board) positive logic is used (Figure 3-5). In order to achieve this redefinition and be consistent with the logic notation of the CU and PE, keeping in mind that all the signals to the PEM pass through the UP Converter of the MLU for CTμL compatibility wherein an inversion takes place, the address bits are buffered and inverted in the control board to provide both the TRUE and COMPLEMENT forms; both forms are needed to decode the binary into the combinatorial address (3-out-of-6 code). It must be mentioned, though, that the CU follows a negative logic notation and, therefore, if its output is a high level signal, it is considered to represent a logic ZERO. This signal, however, is brought through the CDB into the PE where it is inverted and then applied to the MAR, the output of which is brought into the MLU. An inversion takes place in the MAR; consequently, its output is the same level as the input to the operand select gates. The PE follows a positive logic notation, but, in order to be consistent with the CU and MLU signal notation, the input and output signals have been assigned the names shown in Table 3-3.

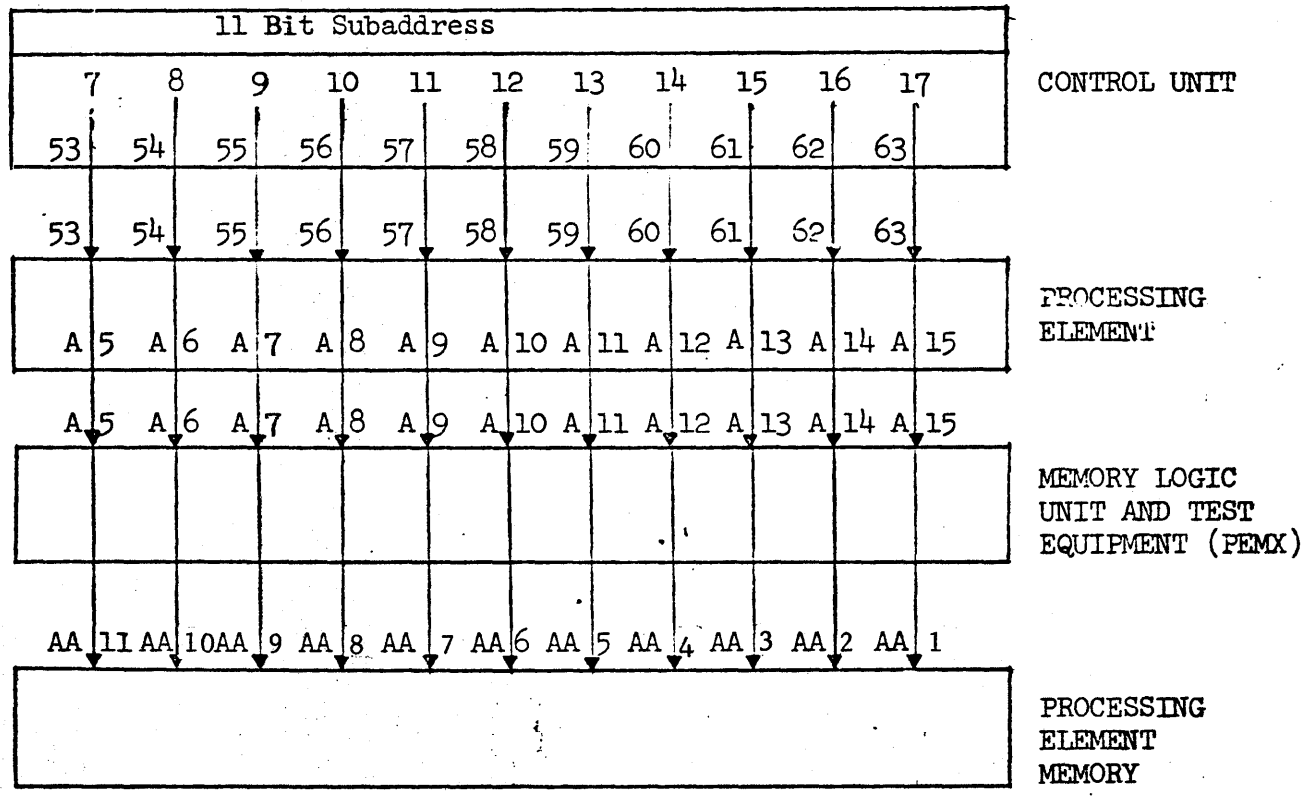


Figure 3-4. Address Chain from CU to PEM.

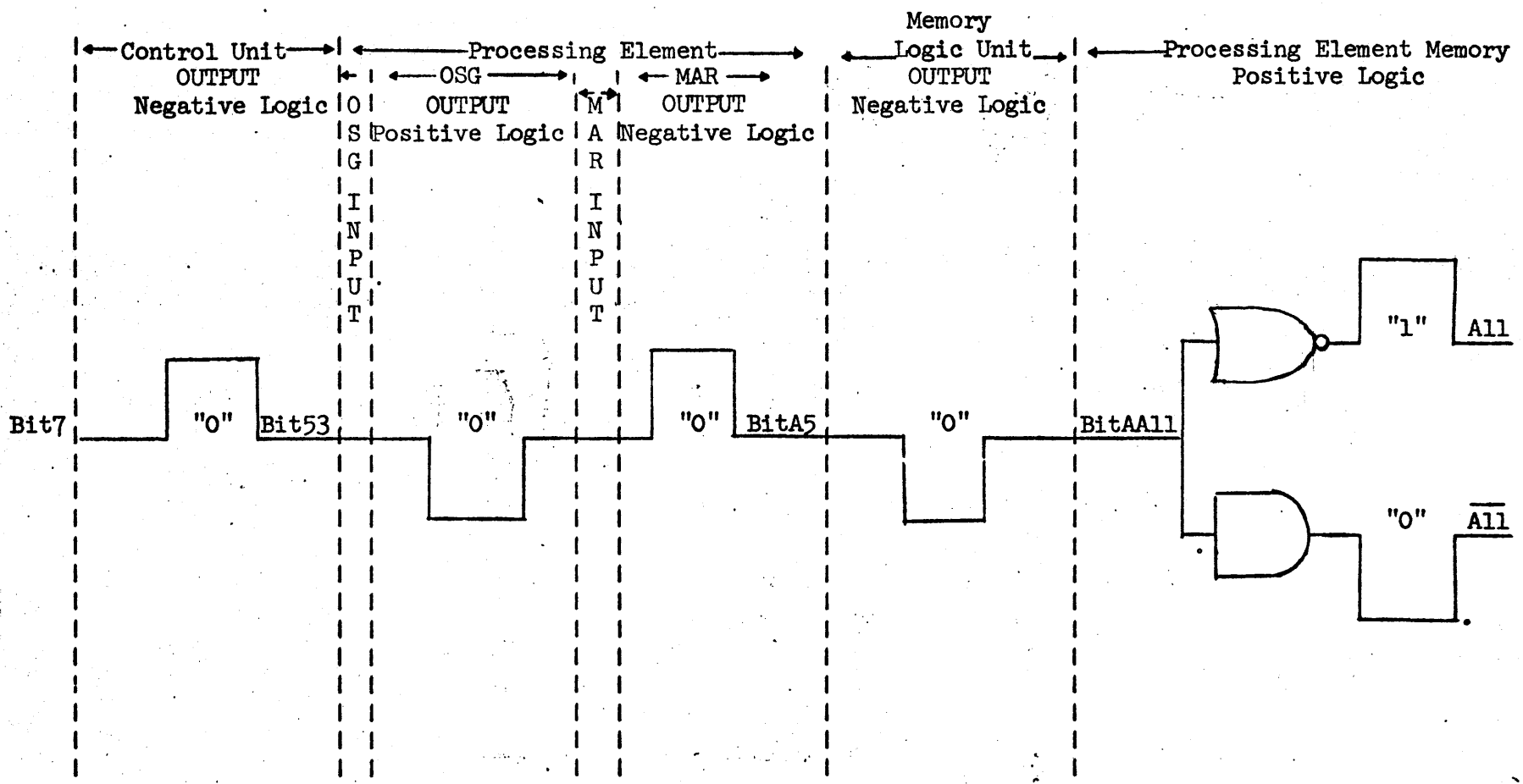


Figure 3-5. Logic Representation of the Address Chain

PEM Combinatorial Address

The PEM storage locations are accessed by using a 3-out-of-6 combinatorial code. This addressing scheme was selected in an attempt to minimize the number of interconnections required between the memory package and the external drivers. In order to reduce the number of external interconnections, decoding logic is provided inside the memory package. This logic decodes the combinatorial address provided by the binary address decoders.

The six least significant bits of the binary address are used for the X direction and the remaining five most significant binary address bits are used for the Y direction. However, in order to access the cell inside the MμL 4100 there are actually only four binary bits available for each direction (X, Y). Thus there are 16 combinations (2^4) available for each direction; these combinations allow access to all 256 cells of the MμL 4100. If only four inputs for each direction were used, the decoding logic inside the memory package would require 2^4 AND gates of four inputs each as well as extra circuitry to provide the true and complement form of the four binary bits for each direction. By using six inputs instead of four, a combinatorial code may be used [2], allowing a reduction in the number of gates needed.

The advantage to using the binary address would be the need for fewer input pins (pads), but it would also increase the complexity of the decoding circuitry inside the memory package.

Since there are four binary bits of address available for each direction, the decoding scheme will have to take into account 2^4 combinations; these combinations are defined by the Binomial Theorem. According to this theorem:

$$(X+\alpha)^{\nu} = X^{\nu} + \binom{\nu}{1} X^{\nu-1}\alpha + \binom{\nu}{2} X^{\nu-2}\alpha^2 + \ldots + \binom{\nu}{\nu} a \qquad (1)$$

but

$$\binom{\nu}{\mu} = \frac{\nu!}{(\nu-\mu)!\mu!} \quad \text{where } \nu \geq \mu$$

gives the number of combinations of ν binary bits (variables) taken μ at a time and therefore (1) above can be written as

$$(X+\alpha)^{\nu} = \frac{\nu!}{\nu!0!} X^{\nu} + \frac{\nu!}{(\nu-1)!1} X^{\nu-1}\alpha + \frac{\nu!}{(\nu-2)!2!} X^{\nu-2}\alpha^2$$

$$+ \frac{\nu!}{(\nu-3)!3!} X^{\nu-3}\alpha^3 + \ldots + \frac{\nu!}{0!\nu!} a^{\nu} \qquad (2)$$

for $X = \alpha = 1$ and $\nu = 4$. Equation (2) gives:

$$(2)^4 = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 16 \text{ combinations} \qquad (3)$$

Equation (3) justifies the statement that, if the decoding scheme inside the MμL 4100 uses the straight binary address, it will require 2^4 AND gates of four inputs each. If $\nu = 6$ and, of course, $X = \alpha = 1$, Equation (2) gives:

$$(2^6) = \binom{6}{0} + \binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{6}{4} + \binom{6}{5} + \binom{6}{6} = 64 \text{ combinations} \qquad (4)$$

In order to access the 256 cells of the MμL 4100, only 16 combinations are needed in each direction; therefore, not all the terms of series (4) are needed. By inspection, it is seen that the $\binom{6}{3}$ term provides 20 combinations, $\binom{6}{3} = \frac{6!}{3!3!} = 20$; this provides 16 combinations to access the cell inside the package with four extra combinations. These extra combinations may be used if the redundancy option is implemented.

NOTE Use of the combinatorial addressing scheme also makes possible implementation of redundant memory packages. This would allow use of MμL 4100 packages that contain malfunctioning data cells; valid data cells in a redundant package could be substituted for malfunctioning data cells in the primary memory packages. This would require address manipulation outside the memory packages to disable bit lines associated with malfunctioning data cells and automatically access the substitute data cells. This redundancy is not presently used, but the use of a combinatorial address scheme and the redundant address decode logic included on the PEM control board make this a potential feature for the PEM.

By bringing six lines into the MμL 4100, three of which are always high, there is no need for logic inversion circuits to provide the complement form of the address bits as there would be if the straight binary decoding form (four pads) were adopted.

In order to be able to bring six address lines into the memory package for each direction, when only four lines for each direction are received from the MLU, a code conversion is needed to generate these six lines. Eight out of the 11 binary address bits received from the MLU are used to generate the combinatorial address. Four binary address bits are decoded to provide the six-bit combinatorial address for the X direction; the other four are decoded to provide the six-bit combinatorial address for the Y direction. The remaining three binary address bits are used to generate the necessary controls for selecting the proper substack. The equations of the combinatorial address lines and the three substack select address lines will be given in terms of the binary address lines later when the address chain is described. Table 3-4 compares the binary and combinatorial addresses by identifying the various address bit names as they appear at the MLU, PEM paddle board, PEM control board, and PEM memory boards.

Table 3-4. Binary and Combinatorial Address Interface

BINARY ADDRESS				COMBINATORIAL ADDRESS			
MLU SIGNAL NAME	PEM SIGNAL NAME	PADDLE BOARD NUMBER	CONTROL BOARD INPUT PIN NO.	SIGNAL NAME	OUTPUT PIN	SIGNAL NAME	INPUT PIN
MYW-W05--0	AA11	P1	145	X1I	183	X1	181
				X1Ø	182		
MYW-W06--0	AA10	P2	181				
MYW-W07--0	AA9	P1	178	X2I	185	X2	184
				X2Ø	186		
MYW-W08--0	AA8	P2	175	X3I	189	X3	187
				X3Ø	188		
MYW-W09--0	AA7	P1	172				
MYW-W10--0	AA6	P2	184	X4I	179	X4	178
				X4Ø	180		
MYW-W11--0	AA5	P1	187	X5I	177	X5	175
				X5Ø	176		
MYW-W12--0	AA4	P2	199				
MYW-W13--0	AA3	P1	196	X6I	173	X6	172
				X6Ø	174		
MYW-W14--0	AA2	P2	193	X7I	191	X7	190
				X7Ø	192		
MYW-W15--0	AA1	P1	190	X8I	195	X8	193
				X8Ø	194		
				X9I	197	X9	196
				X9Ø	198		
				X1Ø	2Ø1	X10	199
				X1ØØ	2ØØ		
				Y1	1Ø1	Y1	1Ø1
				Y2	102	Y2	1Ø2
				Y3	103	Y3	103
				Y4	104	Y4	104
				Y5	105	Y5	105
				Y6	106	Y6	106

3.1.2 X and Y Binary Address Decoder Logic

Before the logic redefinition takes place (Figure 3-4), the address is subdivided into two unequal parts, consisting of the six lower order bits, which are used to generate all the necessary address lines to access the MμL 4100 in the X direction, and the five higher order bits, which are used to generate the address needed to access the MμL 4100 in the Y direction. At this point, the address, in addition to being logically redefined, is assigned the notation shown in Table 3-5. Bits AA1 through AA4 generate the six X combinatorial address lines (Table 3-6), and AA7 through AA10 generate the six Y combinatorial lines (Table 3-7).

The three remaining address bits (AA11, AA6, and AA5) are decoded in such a manner that AA5 and AA6 will generate the controls for the selection of the memory board substacks along the X direction (Table 3-8) and AA11 will generate the controls for the selection of the substacks along the Y direction. Because these three address bits control the selection of a single substack and are not involved in the binary address decoder, they will be explained more thoroughly as part of the discussion of the memory board organization.

Table 3-5. Logic Redefinition of Binary Address

ADDRESS BITS AT PEM PADDLE BOARD (NEGATIVE LOGIC)	ADDRESS BITS IN PEM's CONTROL BOARD	
	Inverter Output	Buffer Output
AA1	A1	$\overline{A1}$
AA2	A2	$\overline{A2}$
AA3	A3	$\overline{A3}$
AA4	A4	$\overline{A4}$
AA5	A5	$\overline{A5}$
AA6	A6	$\overline{A6}$
AA7	A7	$\overline{A7}$
AA8	A8	$\overline{A8}$
AA9	A9	$\overline{A9}$
AA10	A10	$\overline{A10}$
AA11	A11	$\overline{A11}$

Table 3-6. Truth Table of Binary and Combinatorial X Address

BINARY ADDRESS								COMBINATORIAL ADDRESS						WORD LINE (INSIDE THE MμL 4100)
AA4		AA3		AA2		AA1								
$\overline{A4}$	A4	$\overline{A3}$	A3	$\overline{A2}$	A2	$\overline{A1}$	A1	X1	X2	X3	X4	X5	X6	
0	1	0	1	0	1	0	1	0	0	1	1	1	0	0
0	1	0	1	0	1	1	0	0	1	0	1	1	0	1
0	1	0	1	1	0	0	1	0	1	1	0	1	0	2
0	1	0	1	1	0	1	0	0	1	1	1	0	0	3
0	1	1	0	0	1	0	1	0	0	1	0	1	1	4
0	1	1	0	0	1	1	0	0	0	0	1	1	1	5
0	1	1	0	1	0	0	1	0	1	1	0	0	1	6
0	1	1	0	1	0	1	0	0	1	0	1	0	1	7
1	0	0	1	0	1	0	1	1	0	0	1	1	0	8
1	0	0	1	0	1	1	0	1	1	0	1	0	0	9
1	0	0	1	1	0	0	1	1	0	1	0	1	0	10
1	0	0	1	1	0	1	0	1	1	1	0	0	0	11
1	0	1	0	0	1	0	1	1	0	0	0	1	1	12
1	0	1	0	0	1	1	0	1	0	0	1	0	1	13
1	0	1	0	1	0	0	1	1	0	1	0	0	1	14
1	1	1	0	1	0	1	0	1	1	0	0	0	1	15

Table 3-7. Truth Table of Binary and Combinatorial Y Address

BINARY ADDRESS								COMBINATORIAL ADDRESS						BIT LINE (INSIDE THE MμL 4100)
AA10		AA9		AA8		AA7								
$\overline{A10}$	A10	$\overline{A9}$	A9	$\overline{A8}$	A8	$\overline{A7}$	A7	Y1	Y2	Y3	Y4	Y5	Y6	
0	1	0	1	0	1	0	1	0	0	1	1	1	0	0
0	1	0	1	1	0	0	1	0	1	0	1	1	0	1
0	1	0	1	1	0	1	0	0	1	1	0	1	0	2
0	1	0	1	0	1	0	1	0	1	1	1	0	0	3
0	1	1	0	0	1	1	0	0	0	1	0	1	1	4
0	1	1	0	1	0	0	1	0	0	0	1	1	1	5
0	1	1	0	1	0	1	0	0	1	1	0	0	1	6
0	1	1	0	0	1	0	1	0	1	0	1	0	1	7
1	0	0	1	0	1	0	1	1	0	0	1	1	0	8
1	0	0	1	0	1	1	0	1	1	0	1	0	0	9
1	0	0	1	1	0	0	1	1	0	1	0	1	0	10
1	0	0	1	1	0	1	0	1	1	1	0	0	0	11
1	0	1	0	0	1	0	1	1	0	0	0	1	1	12
1	0	1	0	0	1	1	0	1	0	0	1	0	1	13
1	0	1	0	1	0	0	1	1	0	1	0	0	1	14
1	0	1	0	1	0	1	0	1	1	0	0	0	1	15

Table 3-8. Truth Table of Binary Address and Substack Select Controls (X Direction)

BINARY ADDRESS				SUBSTACK SELECT CONTROLS				SELECTED SUBSTACKS ON MEMORY BOARDS
AA6		AA5						
$\overline{A6}$	A6	$\overline{A5}$	A5	X7	X8	X9	X10	
0	1	0	1	0	0	0	1	3 and 7
0	1	1	0	0	0	1	0	2 and 6
1	0	0	1	0	1	0	0	1 and 5
1	0	1	0	1	0	0	0	0 and 4

The combinatorial X and Y address equations can be derived from Tables 3-6 and 3-7, respectively. Using the positive logic notation, as it appears in Tables 3-6 and 3-7, the following equations are obtained:

$$
\left.
\begin{aligned}
X1 &= \overline{A4} \\
X2 &= \overline{A1}\cdot\overline{A2} + \overline{A2}\cdot A4 + \overline{A1}\cdot A3 \\
X3 &= \overline{A1}\cdot\overline{A2} + A1\cdot A4 + \overline{A2}\cdot A3 \\
X4 &= \overline{A1}\cdot A2 + \overline{A1}\cdot A4 + A2\cdot A3 \\
X5 &= A1\cdot A2 + A2\cdot A4 + A1\cdot A3 \\
X6 &= \overline{A3} \\
\\
Y1 &= \overline{A10} \\
Y2 &= \overline{A7}\cdot\overline{A8} + \overline{A8}\cdot A10 + \overline{A7}\cdot A9 \\
Y3 &= A7\cdot\overline{A8} + A7\cdot A10 + \overline{A8}\cdot A9 \\
Y4 &= \overline{A7}\cdot A8 + \overline{A7}\cdot A10 + A8\cdot A9 \\
Y5 &+ A7\cdot A8 + A8\cdot A10 + A7\cdot A9 \\
Y6 &= \overline{A9}
\end{aligned}
\right\} \qquad (5)
$$

From Table 3-8, the X direction substack select controls are derived as follows:

$$\begin{aligned} X7 &= \overline{A5}\cdot\overline{A6} \\ X8 &= A5\cdot\overline{A6} \\ X9 &= \overline{A5}\cdot A6 \\ X10 &= A5\cdot A6 \end{aligned} \tag{6}$$

Since the binary bit AA11 can take on only two values, namely "1" or "0", it is decoded as follows:

$$\begin{aligned} Y7 &= \overline{A11} \\ Y8 &= A11 \end{aligned} \tag{7}$$

When Y7 is true, the right half of the memory stack is selected; when Y7 is false, Y8 is true and therefore the left half of the memory stack is selected.

X Binary Address Decoder and Combinatorial Address Driver

Figure 3-6 shows the logic necessary for decoding the binary address bit AA1 into the combinatorial X1 address and the driver for that line into the memory board. Figure 3-7 illustrates X address timing with respect to Initiate. It is apparent, however, that the only difference in the logic of the X1 through X10 address lines in the control board is the configuration of the decoder, which is derived from the logic equations given above for each one of the address lines. Though the logic required for the combinatorial address differs, the driver is the same for X1 through X10.

After the binary address (X1-X10) has been decoded into the combinatorial form, the line is driven by two CTμL 9856 buffers, whose enable signals are CL1A and CL2A. These two buffers comprise a latch, which is needed to reduce the access time by the amount of delay presented by a CTμL 9856.

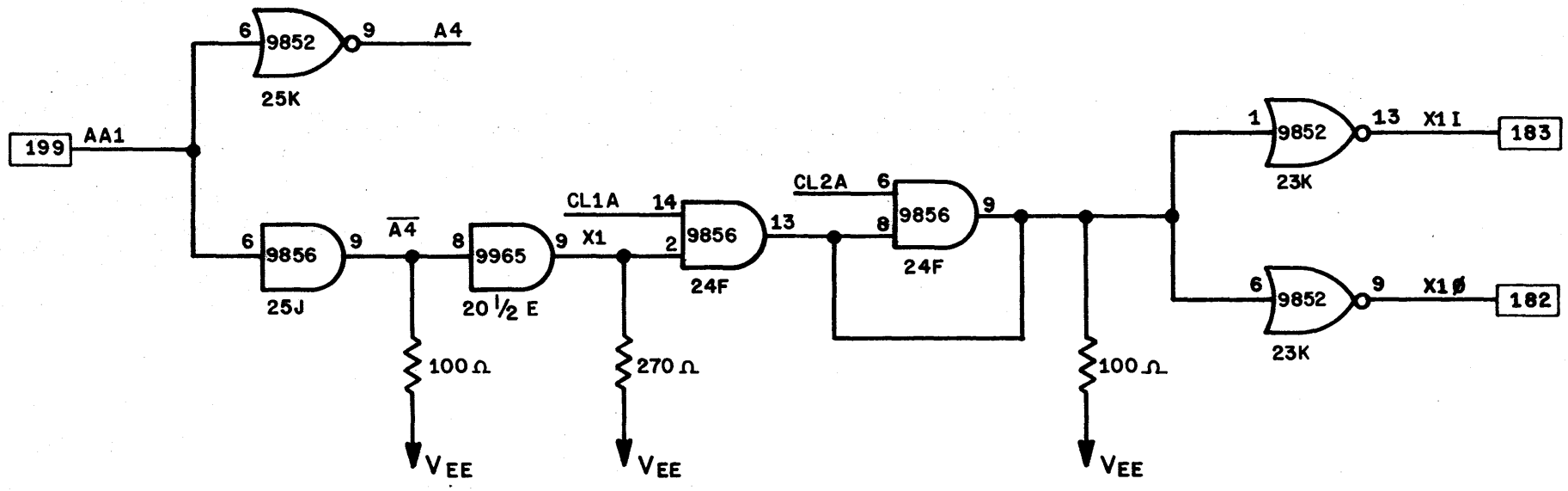
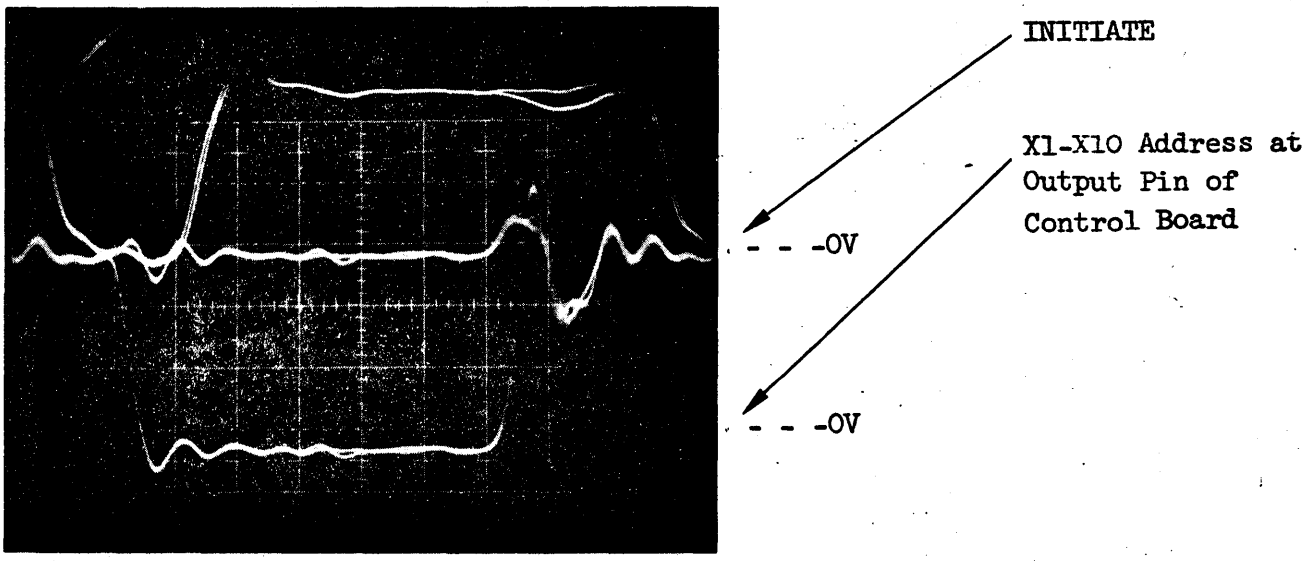


Figure 3-6. X Binary Address Decoder and Combinatorial Address Driver (X1 Line)



1 V/cm, 20ns/cm

Figure 3-7 . X1-X10 Address versus Initiate at Control Board

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

It is worth noting that access time is defined as the time required by the memory to provide data to the pin of the memory board (data output register) as measured with respect to Initiate.

CL1A enables the address to pass through the first CTμL 9856 buffer of the latch and thus to bypass the second CTμL 9856 buffer. If the address bit is a logic ONE, this signal should remain positive until at least 5 ns after CL2A arrives at the input of the second CTμL 9856 buffer. The reason for requiring CL1A to be present for at least 5 ns after the rise of CL2A is to guarantee that when CL1A is removed, the output of the second CTμL 9856 buffer (pin 9) will continue to be high (assuming a logic ONE for that address bit). If the latch is not allowed to reset before CL2A becomes stable at pin 9, it will remain set for as long as CL2A is asserted. Control logic timing assures that CL2A will hold the address bit in the latch for 165 ns after the arrival of Initiate. It can be said, then, that CL1A is used to enable the address into the register (latch), thereby establishing the leading edge of the address at the memory package, while CL2A is used to hold the address for the specified period, thereby determining the turn-off time of the address at the MμL 4100.

Once the address is available at pin 9 of the second buffer of the latch, the output of that buffer is fanned out and applied to a pair of CTμL 9852 inverters; this inversion is required so that when each address bit passes through the TTμL 9016 hex-inverter of the X address driver on the memory board (see Figure 3-21), it will be at the correct level when it arrives at the memory package. In this way, if a particular address line is required to be low at the MμL 4100, it will leave the pin of the control board in a high state and will arrive at the MμL 4100 after passing through the TTμL 9016 inverter in a low state. These inversions apply only to X1 through X6 address lines. Because X7 through X10 are the substack enables and since only one line among those four is high at a time, the one which is true at the output of the buffer of the latch (pin 9) leaves the control board in a low state, while the other three leave the control board in a high state, forcing the MμL 4100's of the memory substacks they control to receive all the addresses in a low state (see X address driver of memory board, subsection 3.2.1).

The address bit is fanned out to two CTμL 9852 drivers in order to split the address into INNER and OUTER parts for selecting the proper half of the PEM and thus to reduce the load significantly in driving the X address buffers of the memory board.

The use of pull-down resistors is needed to fulfill the requirement of the CTμL family of devices.

Y Binary Address Decoder and Combinatorial Address Driver

Figure 3-8 shows the logic for decoding the binary bit AA10 into the combinatorial Y1, and the driver of the Y1 address line into the memory board. Figure 3-9 illustrates Y address timing with respect to Initiate. As was mentioned in the discussion of the X binary address decoder, the decoder for the Y address is the result of logic equation (5). There are two differences between the X binary address decoders and the Y binary address decoders. One difference is that the drivers of Y1 through Y6 lines require an extra control signal (CLY) to enable the address into the memory board; the other difference is that the drivers of Y7 and Y8 lines require a pair of CTμL 9856 buffers for splitting the load.

Control pulses CL1B and CL2B perform the same controlling operation as CL1A and CL2A, respectively. CL2B must be present on the second buffer of the latch at least 5 ns before CL1B is removed in order to ensure that the removal of CL1B will not affect the operation of the latch.

The Y direction address will be present as long as CL2B is high. It can be said that CL1B is the enable for Y1 through Y8 address, while CL2B is the signal that maintains the address at the MμL 4100 as long as it is needed.

Due to a problem, however, in the early production of MμL 4100's which required the Y address to be delayed at the input of MμL 4100 by approximately 35-45 ns with respect to the X address, an extra clock pulse (CLY) is needed to strobe the Y1-Y6 address precisely at the correct time to meet this requirement. Enabling the Y1-Y6 address could not be achieved simply by using CL1B, because of the skew involved.

It should be noted that the Y direction drivers are noninverting CTμL 9856's, while the drivers for the X direction address lines are inverters. The reason for this can be found in the description of the X and Y address drivers of the memory board, where the X address driver uses an inverter prior to accessing the MμL 4100, and therefore another inversion is required.

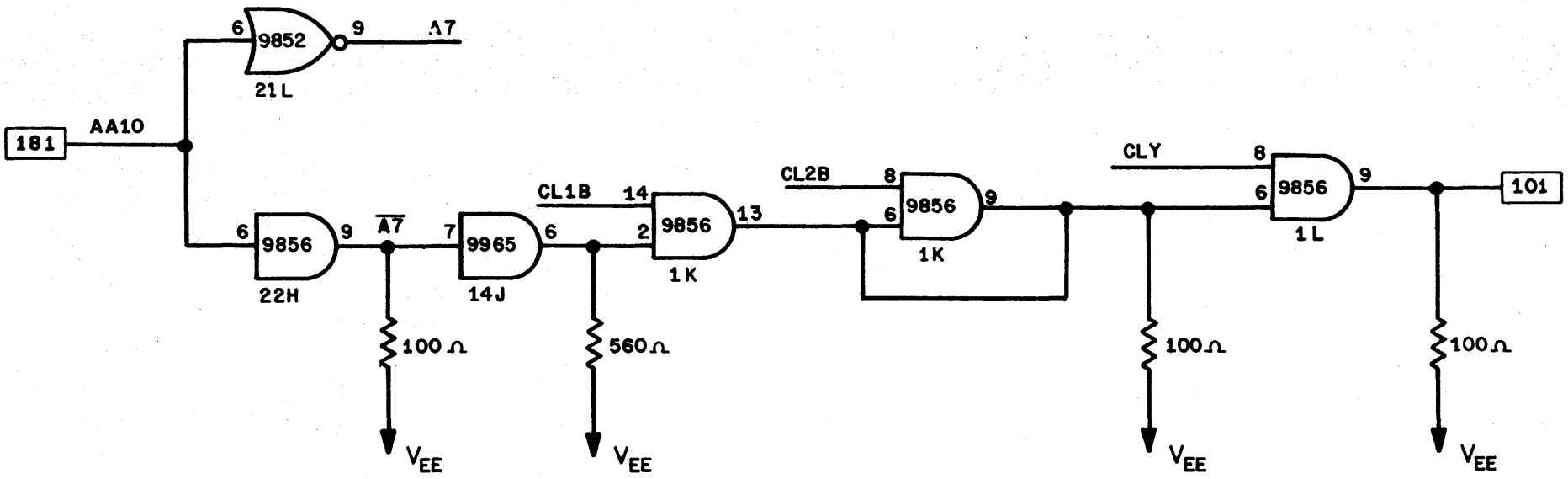
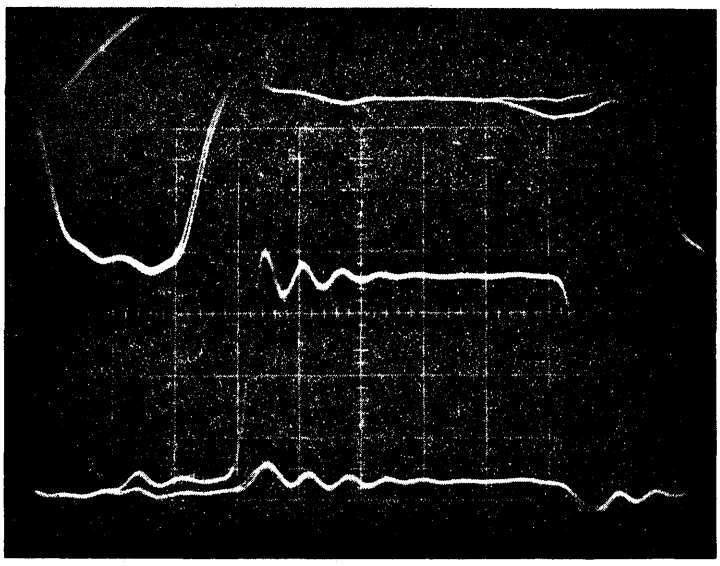
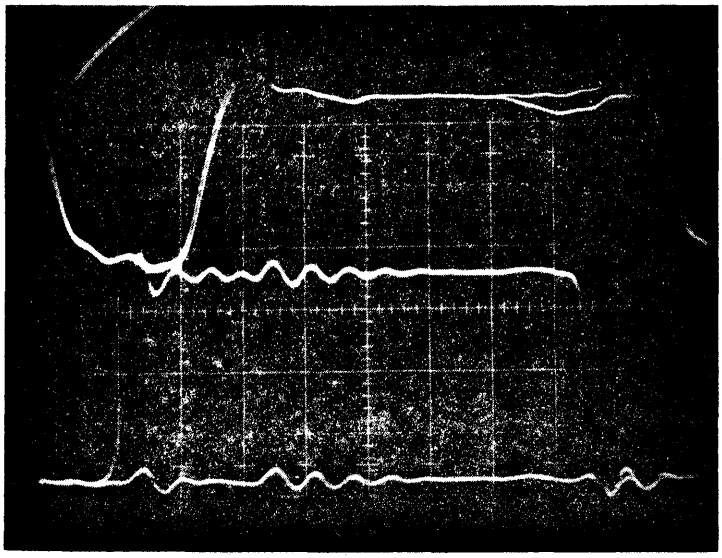


Figure 3-8. Y Binary Address Decoder and Combinatorial Address Driver (Yl Line)





1 V/cm, 20ns/cm

Figure 3-9. Y1 - Y8 Address versus Initiate at Control Board

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

3.1.3 Redundant Address Decoder

When describing the MμL 4100 device in subsection 3.1.1, it was said that one of the advantages of the combinatorial address is to allow the address to access a spare MμL 4100 if the storage location specified by that address is faulty. This would require proper wiring interchanges between the additional memory package and an external code converter, whose output would be TRUE only when one of the addresses corresponding to a bad bit line is selected.

Although the logic for this feature has been implemented on the control board, its function will not be described here because the ILLIAC IV PEM, as it now stands, does not use memory boards utilizing the redundancy feature.

3.1.4 Timing Generator

This portion of the control board generates all the necessary control signals for the internal operation of the PEM. Table 3-9 shows timing generator input and output signals and identifies their respective functions.

Definition of Timing Generator Input Signals

a. Initiate (IØ & IØN). This is the main signal used to initiate the PEM's read or write operation. Because the timing of the control signals in the PEM is very critical, the measurement and adjustment of these signals is performed with respect to the Initiate signal (at the 1 V level); the leading edge (falling edge) is assumed to be present at the control board (pin 147) at time TO. When a read or write operation is requested, the MLU provides the PEM with a low level, 50 ns minimum width pulse at a frequency of about 5 MHz. This signal results from the Initiate memory cycle signal (FIMC-----1) in the memory control card of the MLU. Whenever a read operation is begun, Initiate arrives at the PEM in about 10 ns with respect to FIMC-----1; when a write operation is begun, it occurs at about 75 ns with respect to FIMC-----1.

Table 3-9. Control Board Timing Generator Input/Output Signals List

INPUT SIGNAL			OUTPUT SIGNAL			
SIGNAL NAME	SIGNAL NAME	CONTROL BOARD INPUT PIN NO.	SIGNAL NAME	PURPOSE OF SIGNAL	CONTROL BOARD OUTPUT PIN NO.	REMARKS
MINITPL--1	INITIATE	147	CL1A	To turn on X1-X10 address	-	It is used internal to the control board.
MWOUTEN--1	AWØ	151	CL2A	To turn off X1-X10 address	-	It is used internal to the control board.
MWINNEN--1	AWI	146	CL1B	To turn on Y1-Y8 address in Y address latch	-	It is used internal to the control board.
			CL2B	To turn off Y1-Y8 address in Y address latch	-	It is used internal to the control board.
			CLY	To turn on and off Y address	-	To achieve 35-45 ns delay of Y address from X address.
			CLW	To stretch WØ & WØI	-	
			WØI	To hold CL3 low on WRITE cycle; to generate CL5	-	Both WØ & WØI are used internal to the control board.
			CL3	To enable data out during READ operation	162	It is used during READ operation only.
			CL4	To hold data out until 120 ns from next Initiate	165	It is used during READ operation, but is also present during WRITE operation.
			CL5	To enable data in during WRITE operation	127	It is present during WRITE operation only.
			CL6I CL6I	WRITE pulse needed to WRITE data in INNER memory boards	118 150	To WRITE into memory data bits 8-39.
			CL6Ø	WRITE pulse needed to WRITE data in OUTER memory boards	120 152	To WRITE into memory data bits 0-7 and 40-63. Both CL6I & CL6Ø are present during WRITE operation only.
			WØ WI	To strobe CL6Ø To strobe CL6I		

When the signal enters the control board timing generator, it is logically redefined in the same fashion as is the binary address, and, after passing through a 25 ns delay line, it is used as the basis for the precise timing and pulse shaping of the write inner (AWI) and write outer (AWØ) enables and the timing generator output signals (Table 3-9) as well. Because the time varies from signal to signal, a tapped 100 ns delay line is included which provides a relatively simple method for adjusting the falling and trailing edges of any critical signal (the taps normally allow variations in delay in steps of 10 ns). It must be mentioned, however, that the control board should be able to operate in any PEM of the ILLIAC IV system without presenting any significant problems due to tight timing specifications. The variations in the different components used in the PEM require the control signals to be within reasonable limits, but through experience it has been shown that the suggested times in the PEM control board timing chart of Drawing FD41044 have satisfied the requirements of even the worst-case memory boards in the different PEM's. It should be understood, therefore, that the same control signal is not always picked up from the same delay tap on different control boards, but instead some variations should be expected.

b. Write Outer Enable (AWØ). This is a low level, 50 ns wide pulse appearing at control board pin 151 at the same time as Initiate; it is used to enable CL6Ø control signal during the write operation into PEM memory boards 3 and 4 which accommodate the OUTER word (data bits 0-7 and 40-63).

c. Write Inner Enable (AWI). This is a low level, 50 ns wide pulse which may be coincident with Initiate when it appears at control board pin 146. It is used to enable CL6I control signal during the write operation into PEM memory boards 1 and 2, which accommodate the INNER word (data bits 8-39).

Definition of Timing Generator Output Signals

a. Enables of X and Y Combinatorial Address Signals (CL1A, CL1B). These are two high level signals, each about 25 ns wide, which are used to enable the X1-X6 (CL1A) and Y1-Y6 (CL1B) into the combinatorial address register. Because these signals are the result of Initiate, where IØ, delayed by 25 ns through the delay line shown in Figure 3-10, and IØN, driven directly to those gates whose outputs provide CL1A and CL1B, there is no way to adjust these signals through adjustable timing taps. Instead, CL2A and CL2B can be adjusted in such a way that the trailing edge of CL1A and CL1B will remain stable at least 5 ns after the rise (leading edges) of CL2A and CL2B, respectively.

In order to achieve a fast access time, CL1A and CL1B have been timed to be present when X1-X6 and Y1-Y6 address bits arrive at the address register of the control board. The waveforms in Figure 3-10 are readings from one of the 64 PEM's of the ILLIAC IV Quadrant. The numbers at the leading and trailing edges of each waveform specify the time those transitions occur with respect to Initiate, whose trailing edge is measured at 50 percent amplitude and at 0 ns. This point should be kept in mind, since the waveform photos that are provided are only for those control signals that are applied to the memory boards through the control board.

b. X and Y Combinatorial Address Hold Signals (CL2A, CL2B). These are two high level signals, each about 135 ns wide, which are used to hold the combinatorial address stable for about 135 ns. Both signals must be present at least 5 ns before the removal of CL1A and CL1B in order to allow the latch (address register) in the control board to set and provide a stable address until CL2A and CL2B are removed (Figure 3-11). CL2A OFF can be adjusted through tap T10, whose output is a function of CT0, which in turn can be adjusted accordingly through CT8ON. By adjusting CT0, the leading edge of CL2A is affected; this provides a method for adjusting the leading edge of CL2A with respect to the trailing edge of CL1A. Because the leading edge of CL2B is adjusted through circuitry common to CL2A and CL2B, its trailing edge is adjusted through the adjustable timing tap shown in Figure 3-11 (T10, T20, T30). The same figure provides waveforms at different gates to show logic level versus time in nanoseconds as referred to Initiate.

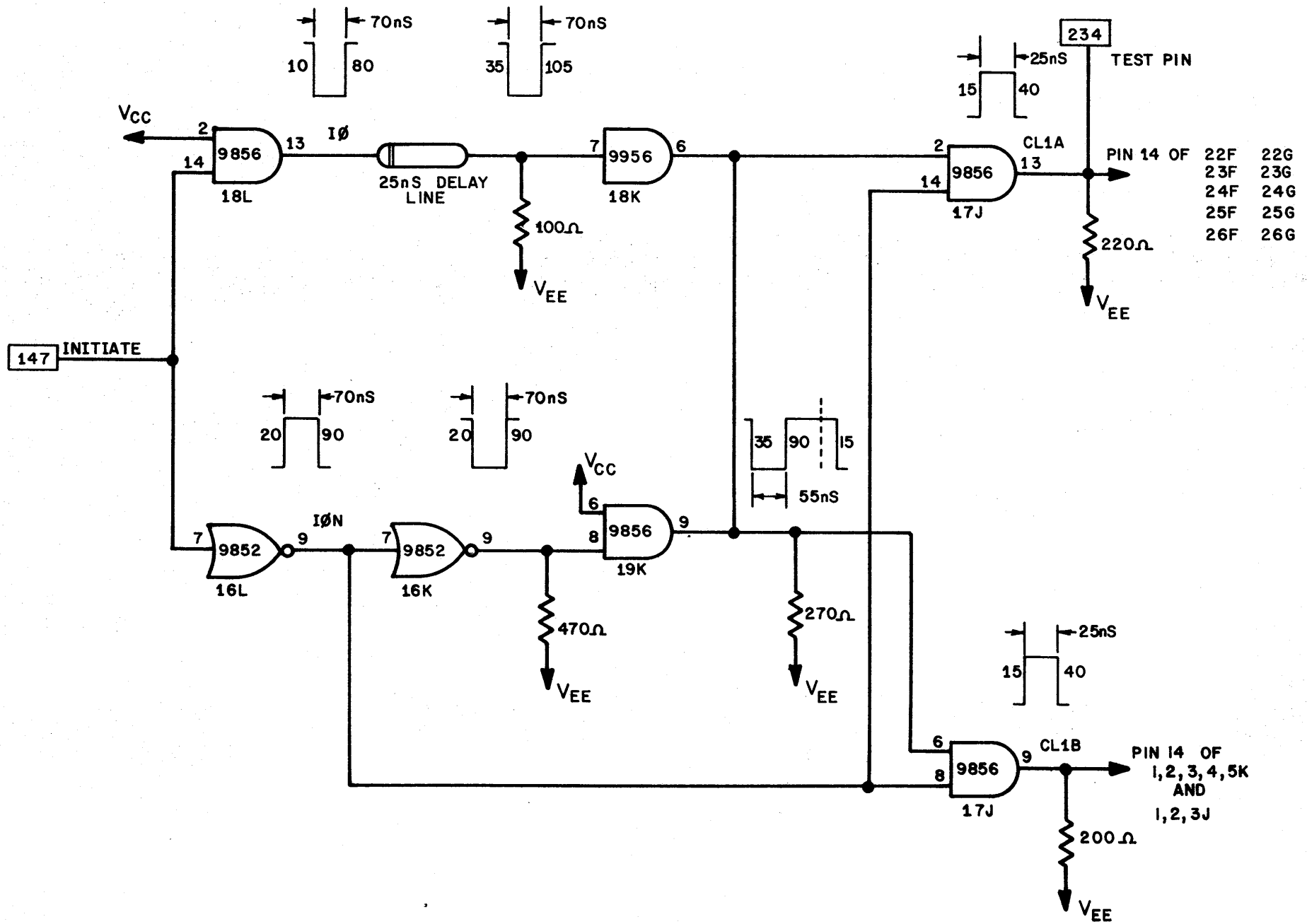


Figure 3-10. CL1A and CL1B Logic.

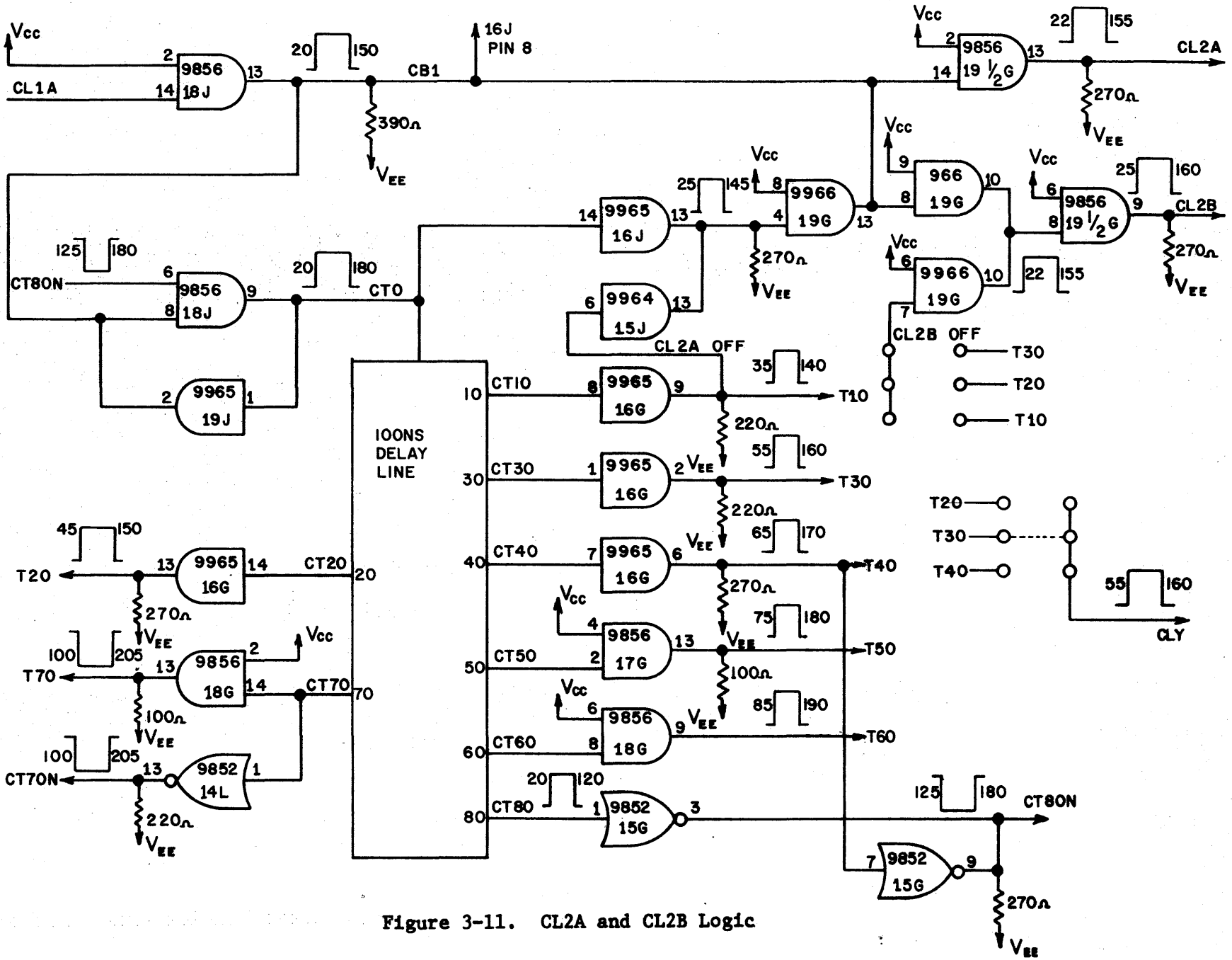


Figure 3-11. CL2A and CL2B Logic

CLY is used to control the times at which the Y1-Y6 address is turned on and off; this control is made necessary by a special characteristic of MμL 4100, which requires that the Y1-Y6 address must follow the X1-X6 address by 35 to 45 ns.

c. Data-out Enable Signal (CL3). This is a high level signal, about 60 ns wide, which is used to enable the data to be gated out of the memory board's data-out register within the specified access time of 188 ns. It has to remain stable for not more than 35 ns after the removal of Y1-Y10 address; it must also be present for at least 5 ns after the appearance of the leading edge of CL4 at control board pin 165. These measurements may be made at control board pin 162 for CL3 and pins 101-106 for Y1-Y6 address.

This signal has been adjusted to be present during the read operation only; its leading edge (50 percent amplitude) has to occur between 135-145 ns after the Initiate of the current read cycle. The trailing edge of CL3 should occur between 190-205 ns with respect to the current Initiate. However, experience has shown that 200 ns, with respect to Initiate, gives the best results. This means that the leading edge of CL4 must be present no later than 195 ns with respect to the current Initiate. Both the leading and trailing edges of CL3 can be adjusted through the timing taps CL3 ON and CL3 OFF, respectively. Because the leading edge of CL3 did not originally have the 10 ns skew it presently has, an optional capacitor (Figure 3-12) could be used to achieve a finer adjustment in case the adjustable timing tap could not bring the timing within the specified limits. The value of this capacitor would vary from one control board to another; for some control boards, there would be no capacitor at all. It has been observed that a 10 pf capacitor could introduce a delay of about 1 ns in the leading edge of CL3.

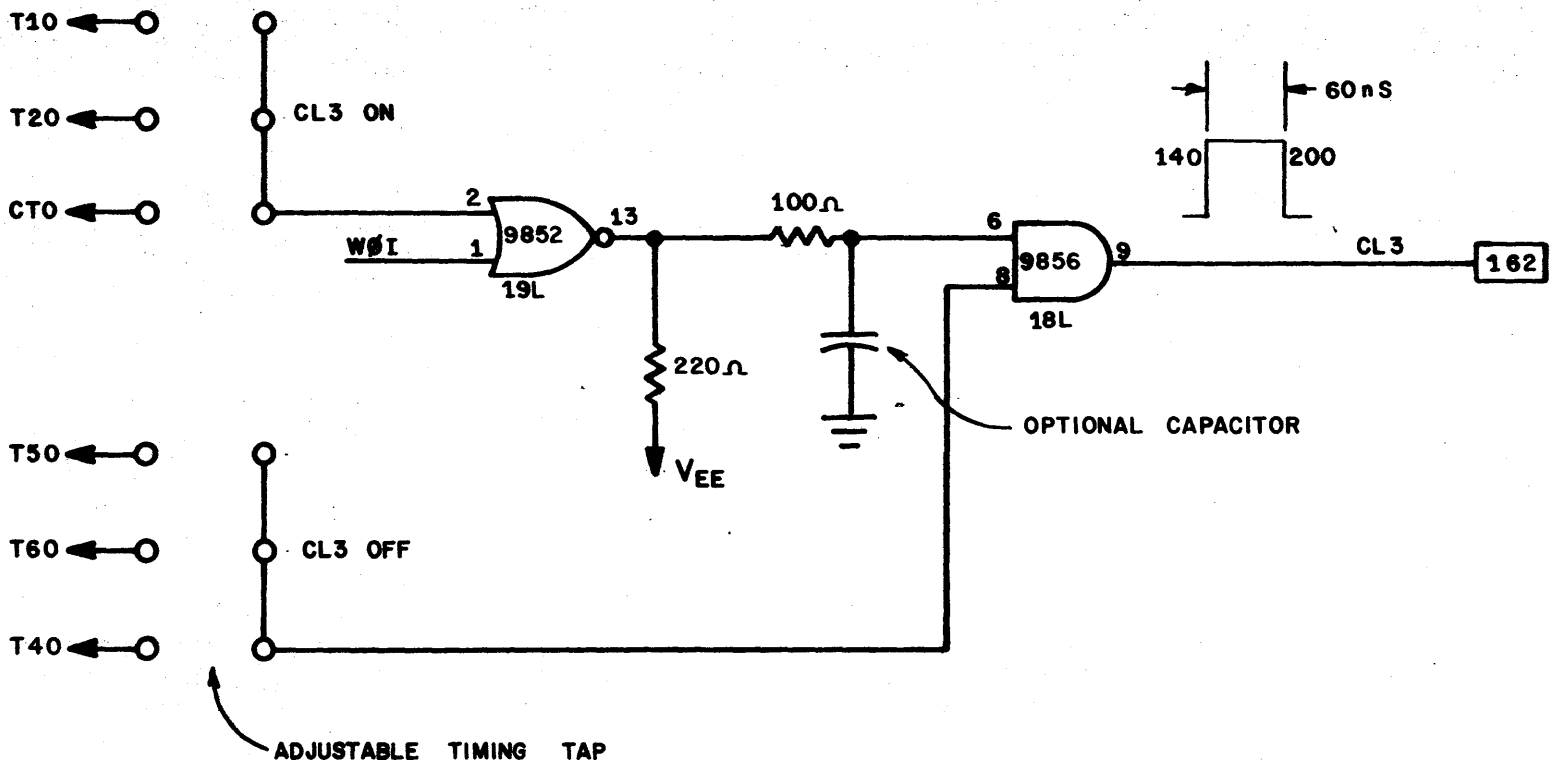


Figure 3-12. Data-out Enable Signal (CL3) Logic.

CL3 is forced to be present only during read operations, because the WRITE OUTER or INNER (WØI), which is a high level pulse during write operations, in both the 32- and 64-bit mode of operation, is applied to the NOR gate 19L pin 1. The output of this gate will remain low as long as WØI is present. If a glitch (transient) is present on CL3 during a write operation, CLW ON should be adjusted through its corresponding timing tap in order to reshape WØI (see logic for CL5) to such a degree that it will force NOR gate 19L pin 13 to remain low during the write operation. This adjustment may require a new adjustment of CL5, since this signal is generated directly from WØI. The waveform at the output of the CTμL buffer (AND gate) 18L pin 9 shown in Figure 3-12 indicates the actual reading from one of the 64 PEM's of the ILLIAC IV Quadrant. Figure 3-13 gives the waveform of CL3 at pin 162 of the control board together with Initiate.

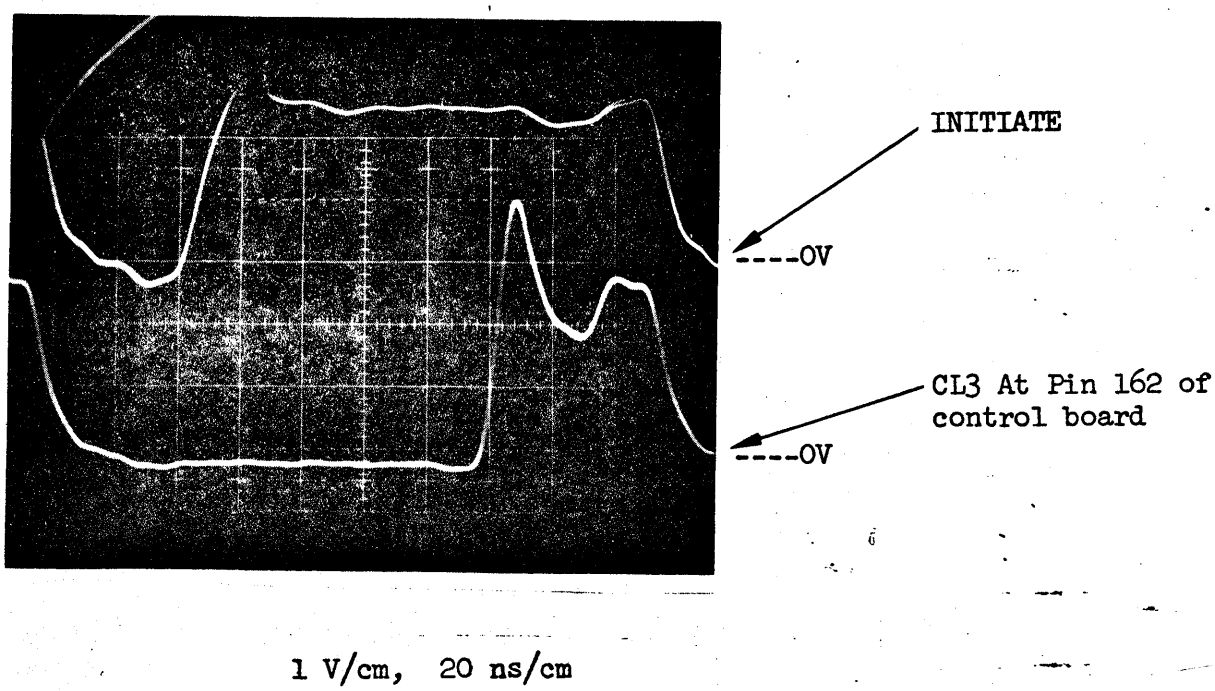
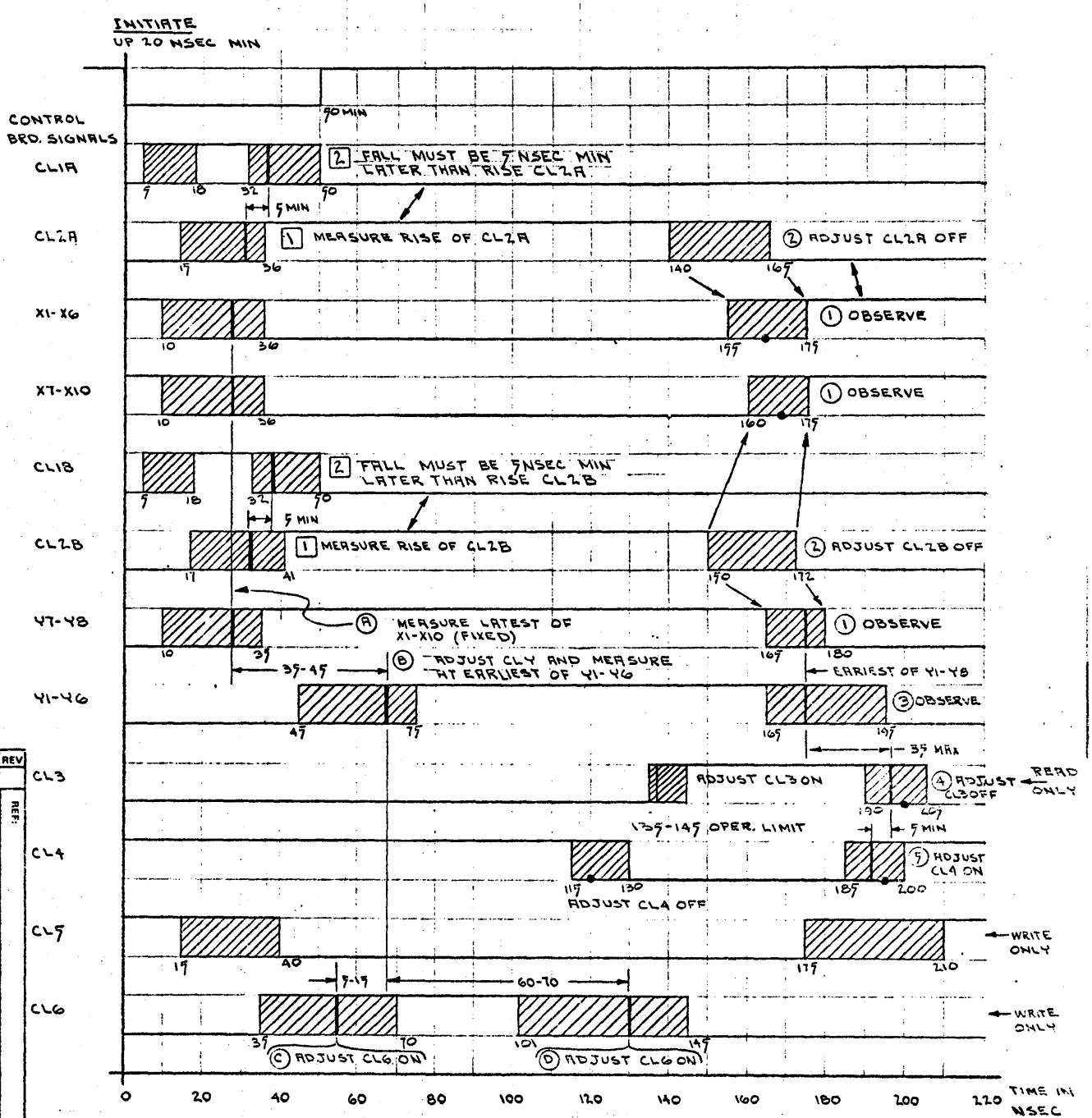


Figure 3-13. CL3 versus Initiate.

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

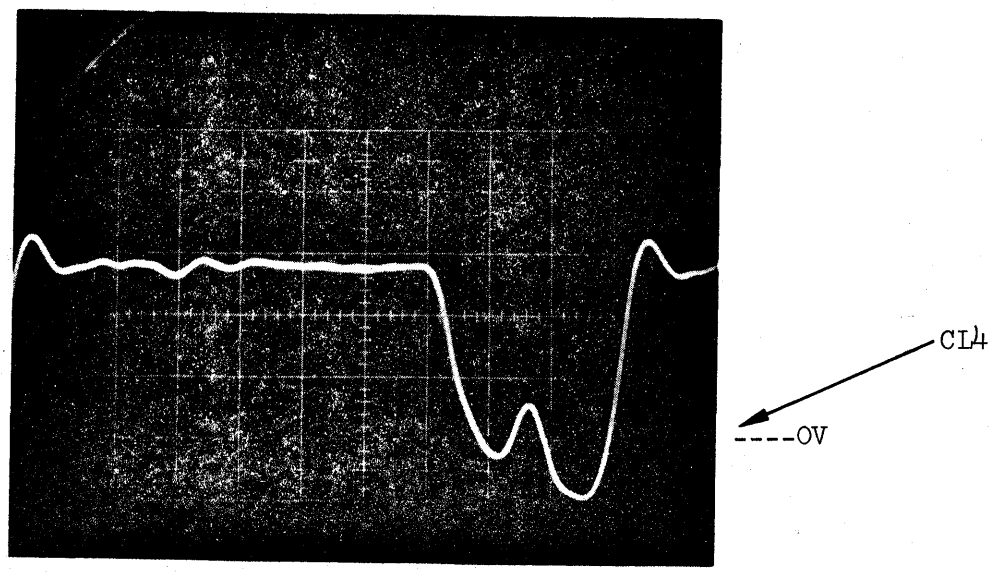


NOTES:

1. INITIATE STARTS AT TIME 0 NSEC
2. ALL SIGNALS ARE MEASURED WITH RESPECT TO INITIATE
3. USE SAMPLING SCOPE ONLY
4. THE DOTTED TIMES (•) ARE SUGGESTED VALUES.

Figure 3-15A. Control Board Timing Chart

d. Data-out Hold Signal (CL4). This is a high level pulse, about 145 ns wide, which is used to hold the read data out of the latch of the memory board's data-out register from the time CL3 is removed until 120 ns of the next memory cycle. The need for CL4 to be present for at least 5 ns prior to CL3 removal is explained in the description of the data-out register. Because CL4 will be present to hold the data stable until 120 ns after the next Initiate, it can be said that it is present during both read and write operations. Figure 3-14 (p. 3-34) shows the logic that produces CL4, with test pin 232 for testing and two timing taps for adjusting the leading and trailing edges of the signal according to the control board timing chart (Figure 3-15A). The waveforms at the outputs of CTμL gate at 15J pin 10 and the CTμL inverter at 19L pin 9 indicate an actual reading from one of 64 PEM's of the ILLIAC IV Quadrant. Figure 3-15 gives the waveform of CL4 as it was photographed at control board pin 165, with the trailing edge of the Initiate at 0 ns.



1 V/cm 20 ns/cm

Figure 3-15. CL4 at Control Board Pin 165.

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

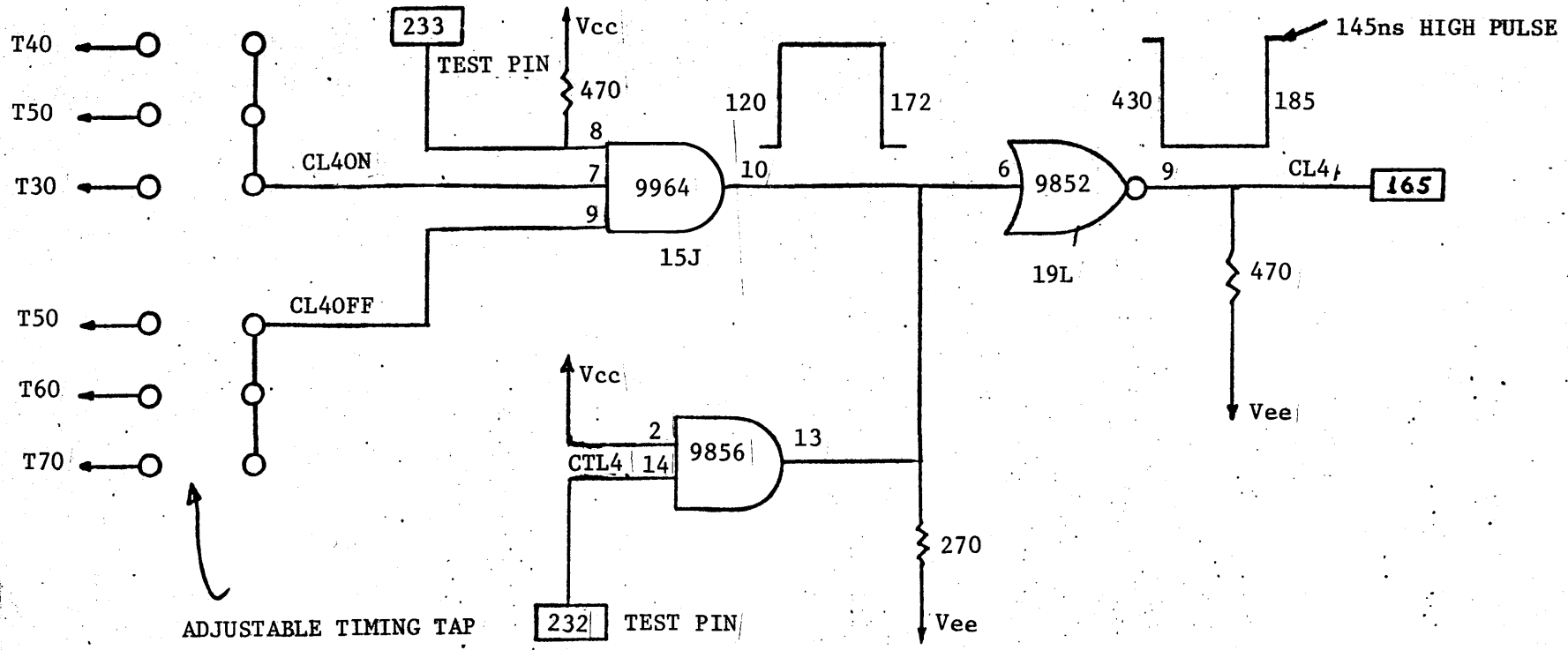


Figure 3-14. Data-out Hold Signal (CL4) Logic

e. Write Driver Enable Signal (CL5). This is a low level signal, about 180 ns wide, which is used to strobe the data to be written into the memory board. The signal is at low level in order to allow the base of the write transistor of the driver in the memory board to be controlled by the input data. CL5 will remain at high level during a read operation, because the WRITE INNER or OUTER (OI) is at high level only during write operations. Figure 3-16 shows the logic used for the formation of CL5 with waveforms at different gates as they were measured in one of the 64 PEM's of the ILLIAC IV Quadrant; this diagram illustrates the method by which CL5 is shaped within the specified timing limits of control board timing chart (FD 41044).

CL5 should be at low level only during write operations, as Figure 3-17 shows. Because there is no timing tap available that allows the leading and trailing edges of CL5 to be adjusted directly, CLW ON (T40, T50) can be adjusted accordingly in order to change the time specifications of CLW; this will in turn adjust WØI and, consequently, CL5. At this point, however, it must be mentioned that any misadjustment of WØI might result in a glitch on CL3 during a write operation; this must not occur. It is recommended that whenever adjustment of CL5 or CL3 is performed, these signals be checked only during the write and read operations, respectively.

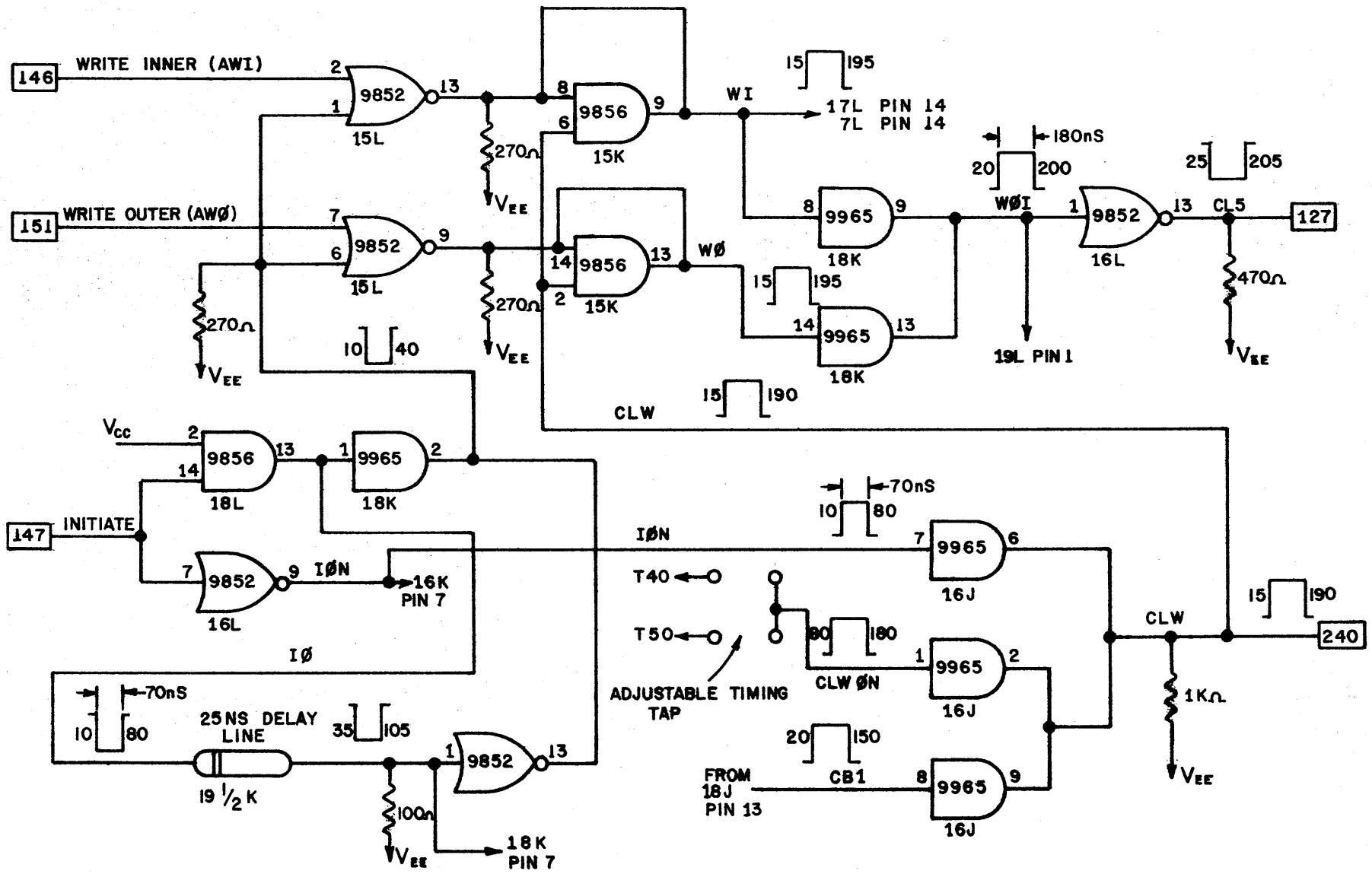
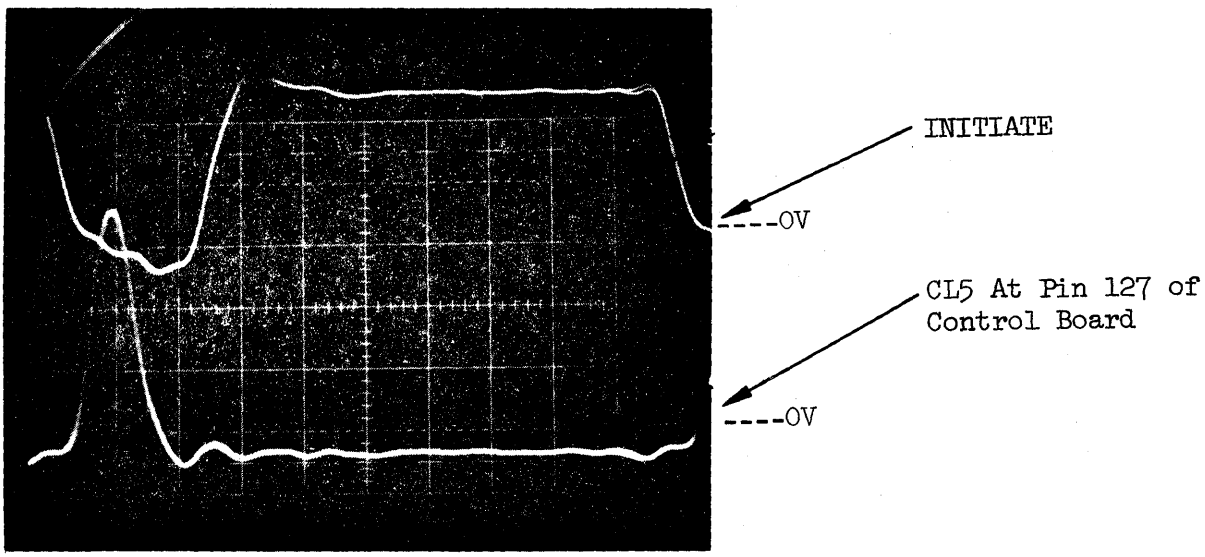


Figure 3-16. WRITE Driver Enable Signal (CL5) Logic.



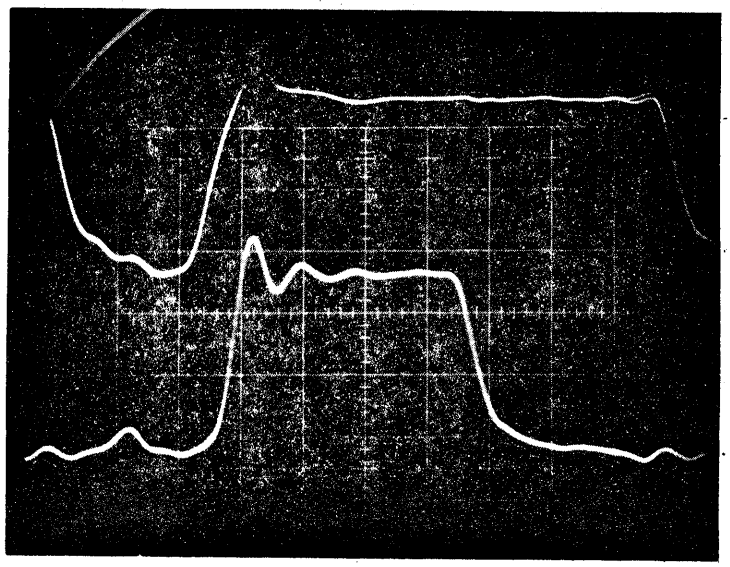
1 V/cm, 20 ns/cm

Figure 3-17. CL5 versus Initiate

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

f. Write Enable Signal (CL6). This is a high level signal about 75 ns wide, which is used to enable the data to be written into the MμL 4100. The leading edge of CL6 (Figure 3-18) should be present 5-15 ns prior to the leading edge of the Y1-Y6 address. The signal must remain stable for about 75 ns with a suggested leading edge at 55 ns and trailing edge at 130 ns with respect to Initiate during the write operation only, provided that the WRITE INNER Enable (WI) for CL6I and WRITE OUTER Enable (WØ) for CL6Ø are present. The reason for having a pair of write enables is explained later during the description of the memory board write driver. Figure 3-18 (p. 3-39) shows that portion of the logic involved in the formation of CL6I and CL6Ø that is not shown in Figure 3-16.

The waveforms at the input of CTμL gate at 15J, pins 1 and 2 indicate the reading at the adjustable timing taps with CL6 OFF connected to the tap at CTO and CL6 ON connected to the tap at T20. Figure 3-19 gives the waveforms of Initiate and CL6 as they were photographed at control board pins 147 and 120, respectively, of one of the 64 PEM's of the ILLIAC IV Quadrant.



1 V/cm, 20 ns/cm

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

Figure 3-19. Photo of CL6 versus Initiate.

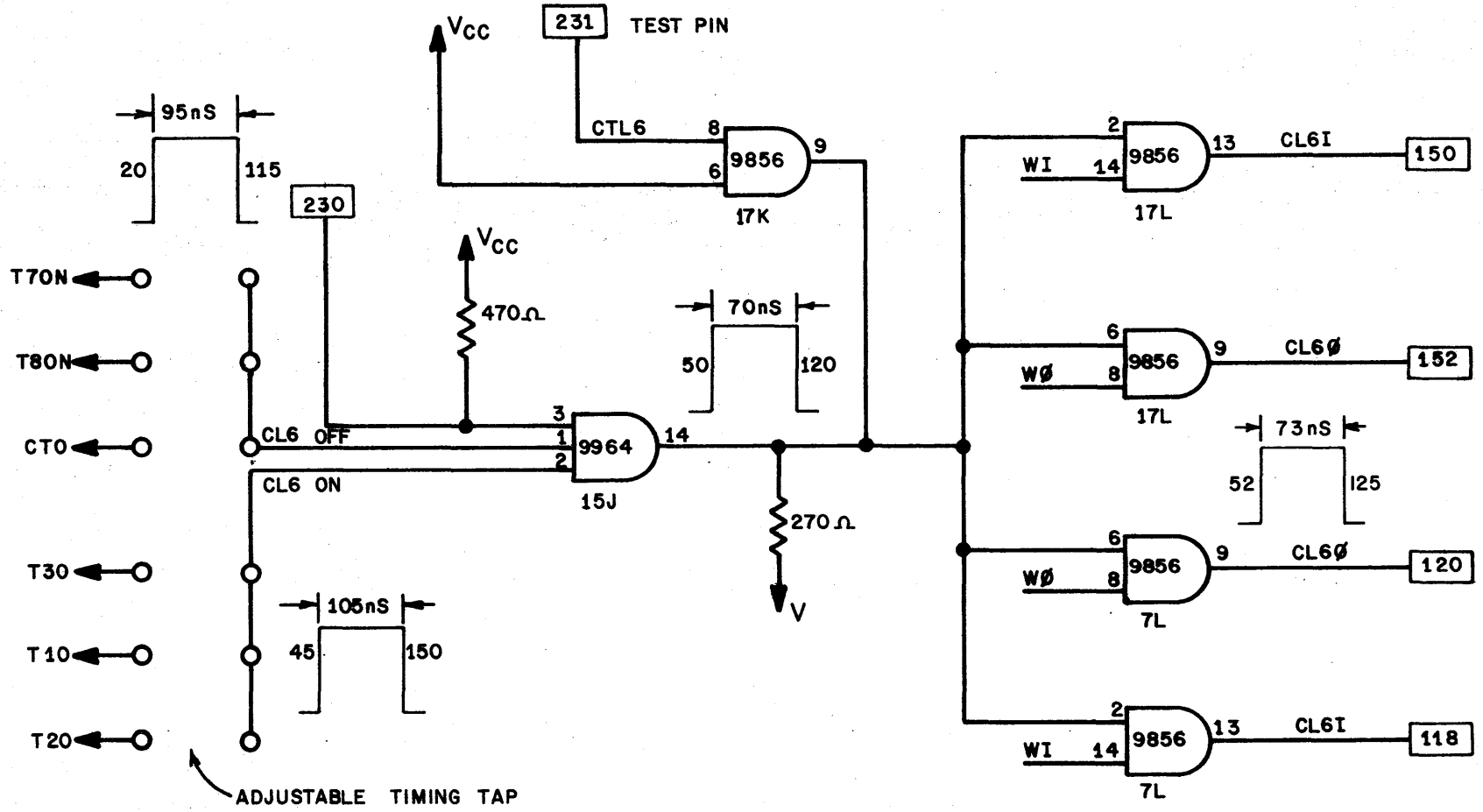


Figure 3-18. WRITE Enable Signal (CL6) Logic.

3.2 MEMORY BOARD

Since the four memory boards in the PEM are identical, the operation of only one will be described. Because the boards are interchangeable, the position of the memory board in the PEM affects only the order of bits that are written into or read out from that board.

Memory board logic may be classified into four functional areas: memory stack, write control, write driver, and data output register. These areas are shown in Figure 3-20.

3.2.1 Memory Stack

This is the portion of the memory that contains 128 MμL 4100 packages, which are organized into eight substacks of 16 packages each. Only one substack on each board may be selected at a time for any read or write operation.

Each MμL 4100 package contains 256 cells and each cell can accommodate one bit of data (logic ZERO or ONE). Since the MμL 4100 is an array of 16×16 flip-flops, it can be said that the MμL 4100 has a capacity of 256 words, each one bit in length and, consequently, that the capacity of the memory substack amounts to 256 words, each of which is 16 bits long. This means, however, that the I/O lines of the MμL 4100's in the same substack are not tied together, but instead are brought separately into the data-out register or write driver. In order to achieve word expansion, the I/O line of each MμL 4100 of a substack is OR-tied with the I/O lines of the corresponding MμL 4100 packages in the other seven substacks. Each memory board, therefore, has a storage capacity of 256 by 8 words, each 16 bits long.

The selection of each substack in the X direction is determined by the state of address bits A5 and A6 which, after being decoded, generate the substack select controls X7, X8, X9, and X10. These four controls select substacks 0 and 4, 1 and 5, 2 and 6, and 3 and 7, respectively. The state of address bit A11 determines the state of select controls Y7 and Y8, each of which selects substacks 0 through 3 and 4 through 7, respectively.

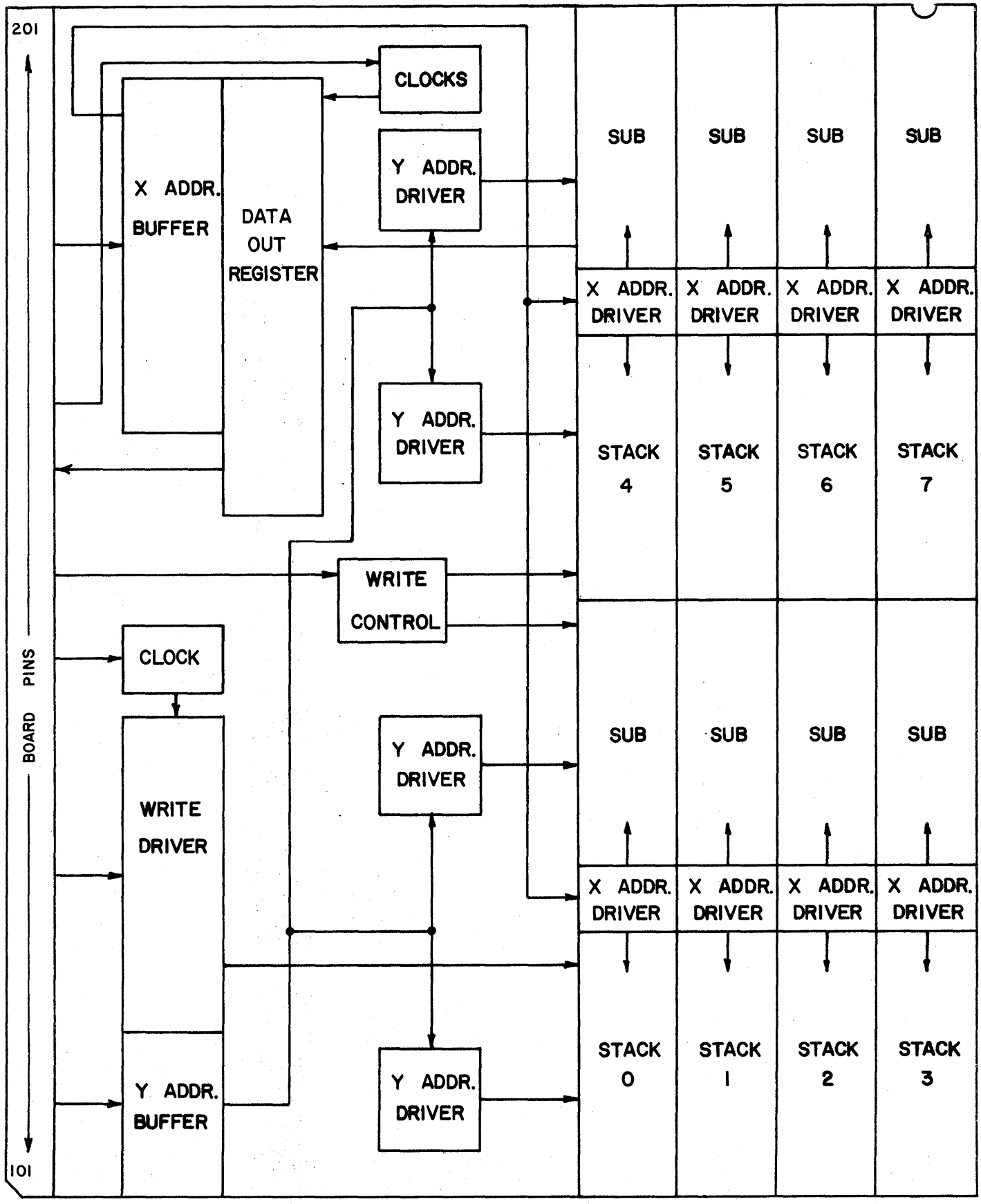


Figure 3-20: ILLIAC IV Memory Board Functional Block Diagram

(Component Side of Board)

The memory stack also contains eight X address and four Y address drivers. Each X address driver is used to drive the six combinatorial X address lines to 16 MμL 4100 packages of its corresponding substack, while each Y address driver is used to drive the six combinatorial Y address lines to 32 MμL 4100 packages located in four substacks. In this way, Y address drivers 1 and 2 drive the Y address lines for 64 MμL 4100 packages located on substacks 0, 1, 2, and 3, and Y address drivers 3 and 4 drive the Y address lines for the remaining 64 MμL 4100 packages located on substacks 4, 5, 6, and 7. In other words, to select one substack, the six X address lines of that particular substack are driven by the corresponding X address driver, while the six Y address lines for the same substack are driven by Y address drivers 1 and 2 for substacks 0 through 3 and by Y address drivers 3 and 4 for substacks 4 and 7. Since only one substack on a memory board is accessed whenever a read or write operation is performed, we can say that all substacks are identical to one another and the only difference is that they require different select controls in order to be accessed. Figures 3-21 and 3-23 illustrate the logic used for the X and Y address buffers and drivers, respectively, for substack 0. Figure 3-22 illustrates the timing for the X1-X6 address with respect to Initiate.

a. X Address Buffer and Driver (Figure 3-21). Table 3-10 shows the X address interface between the control board and a memory board through the signal base. The CTμL 9816 gate is a noninverting driver. It can provide more current than a regular gate; it has excellent noise immunity and high driving capability. The gate is used to drive the X address line into eight substacks (CTμL 9965). For every substack there are three CTμL 9965 AND-OR packages.

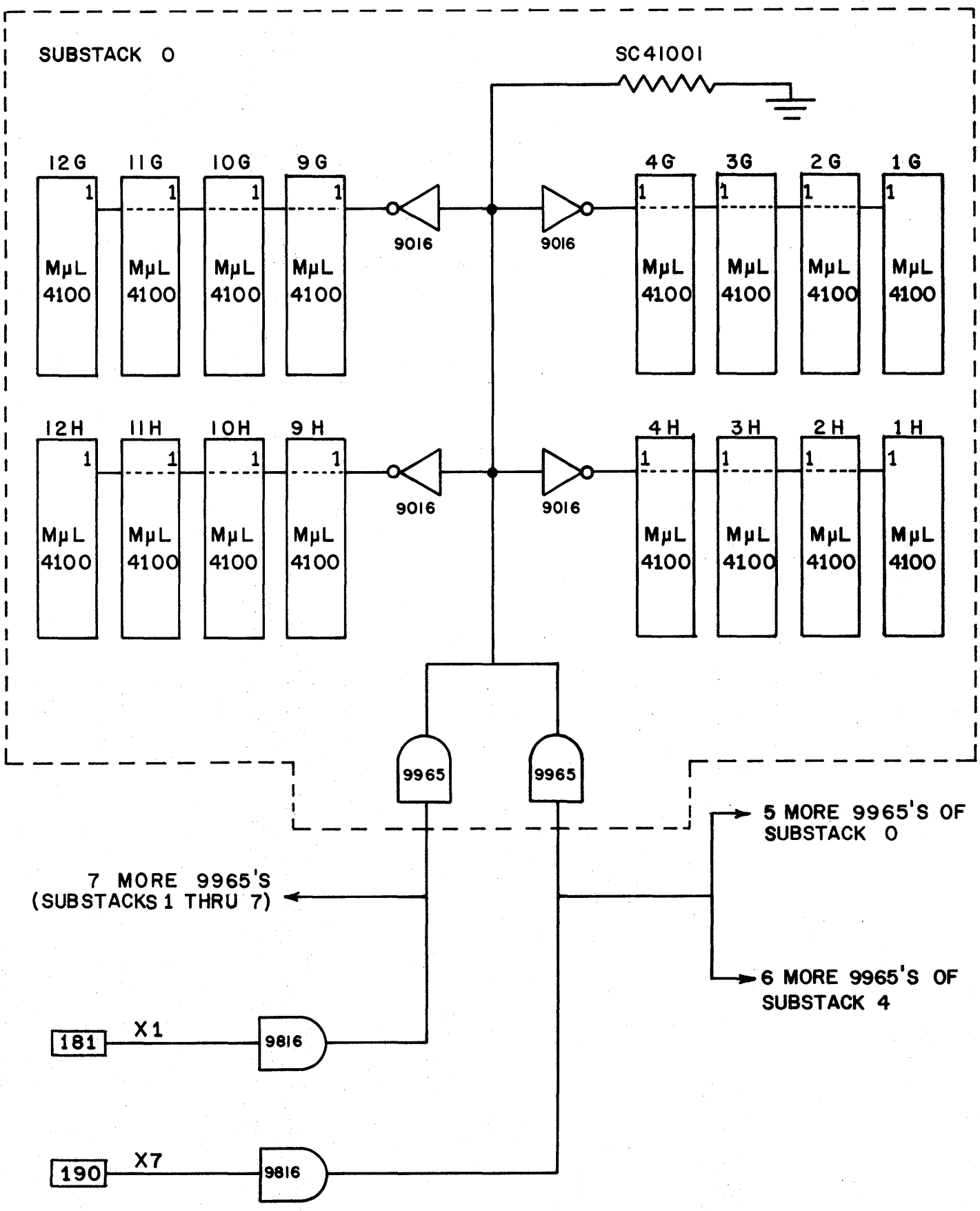
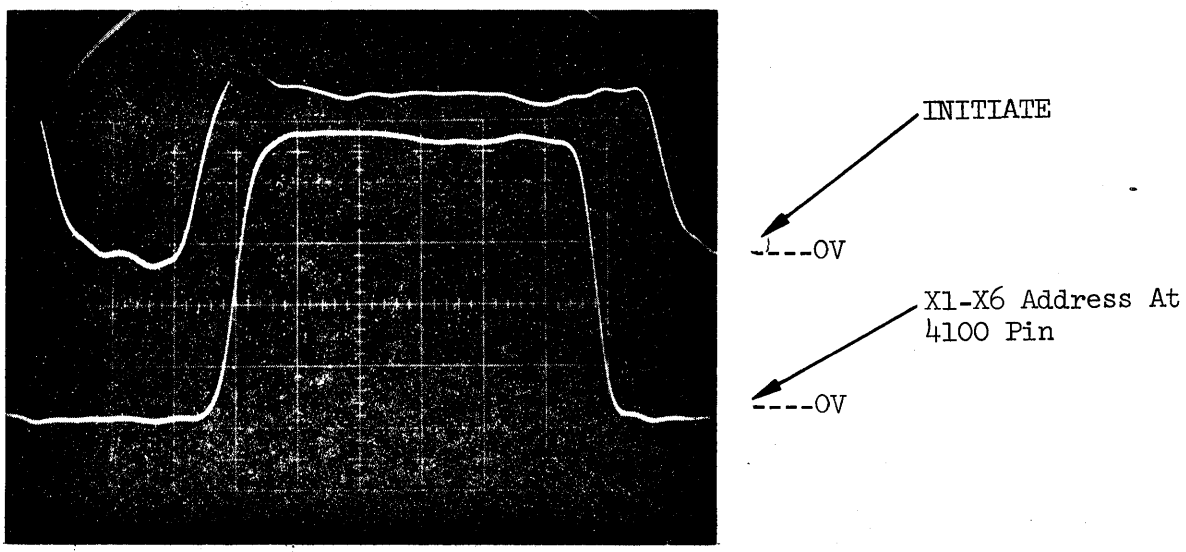


Figure 3-21. X Address Buffer and Driver (One Address Line & Control)



1 V/cm, 20 ns/cm

Figure 3-22. X1-X6 Address at Memory Board MμL **4100 Pin versus Initiate**

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

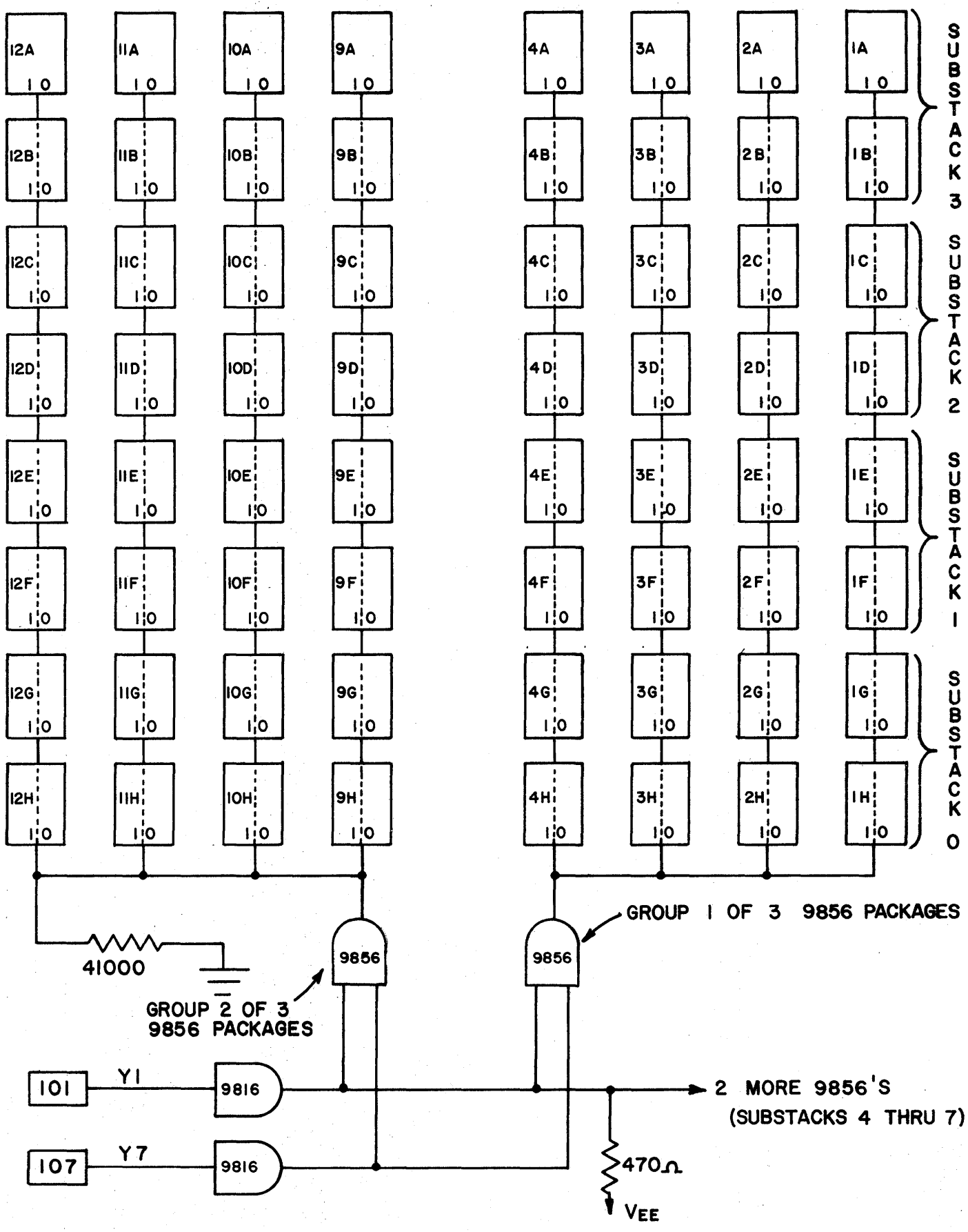


Figure 3-23. Y Address Buffer and Drivers (One Address Line & Control)

Table 3-10. X Address Interface between Control Board and Memory Board

SOURCE OF SIGNAL			DESTINATION OF SIGNAL			REMARKS
SIGNAL NAME	CONTROL BOARD		SIGNAL NAME	MEMORY BOARD		
	DEVICE	BOARD PIN NO.		DEVICE	BOARD PIN NO.	
Address Line X1 IN	CTμL 852	183	Address Line X1	CTμL 816	181	
" " X1 OUT	↑	182	" " X2	↑	184	In the control board
" " X2 IN		185	" " X3		187	the address line is
" " X2 OUT		186	" " X4		178	split into INNER and
" " X3 IN		189	" " X5		175	OUTER parts so that
" " X3 OUT		188	" " X6		172	the address will be
" " X4 IN		179	Select Control X7		190	available for either
" " X4 OUT		180	" " X8		193	mode of operation
" " X5 IN		177	" " X9	↓	196	(64-or 32-bit).
" " X5 OUT		176	" " X10	CTμL 816	199	
" " X6 IN		173				The memory board re-
" " X6 OUT		174				ceives the address
Select Control X7 IN		191				independently of board
" " X7 OUT		192				position and therefore
" " X8 IN		195				it needs only six
" " X8 OUT		194				address lines and four
" " X9 IN		197				select controls.
" " X9 OUT		198				
" " X10 IN		201				
" " X10 OUT	CTμL 852	200				

Each package consists of four single input gates and is used to drive two X address lines and the substack select control X7 (for substack 0) in such a way that the output of the gate which drives X7 is OR-tied to the output of the gate, which drives one of two address lines per CTμL 9956 package. The two OR-tied outputs of each package represent the state of one address line. This address line in turn is brought into four TTμL 9016 packages (hex inverters), each of whose outputs drive four MμL 4100's in a row. The use of the TTμL 9016 device was dictated by the fact that due to problems in the early production stages of the MμL 4100 the X address lines needed to be driven by a saturating device such as the TTμL 9016.

As stated earlier, however, whenever a CTμL output drives a TTμL input, a sink resistor is needed (Figure 2-7). For this reason, a resistor (SC41001, 100 Ω — Figure 3-21) is connected between ground and the line which connects the output of CTμL 9965 with the input of TTμL 9016. In the description of the MμL 4100 it was said that in order to access one of the 16 word lines, six combinatorial address lines are needed, three of which must be high. For example, to select word line 5, address lines X4, X5, and X6 must be high at the input of MμL 4100 while address lines X1, X2, and X3 should remain in the low state. Because of the use of the TTμL 9016, which is an inverting element, address lines X4, X5, and X6 must be sent into the memory board in a low state, while the remaining three address lines must be in a high state. This is accomplished by the use of the CTμL 852 gate of the control board which guarantees that the address lines which need be in a low state at the input of MμL 4100 will leave the control board pin in a high state.

At this point, the substack select control X7 (for substack 0) must leave the control board pin in a low state. In this way, address lines X4, X5, and X6 being in a low state at the output of CTμL 9965 will be OR-tied with X7, which will also be in a low state and, after being inverted by TTμL 9016, will arrive at pins 4, 5, and 6 of MμL 4100 in a high state. Address lines X1, X2, and X3 at the output of CTμL 9965 are in a high state and, therefore, even though they are OR-tied with X7, which is in a low state, they will arrive at pins 1, 2, and 3 of MμL 4100 in a low state because of the inversion through TTμL 9016.

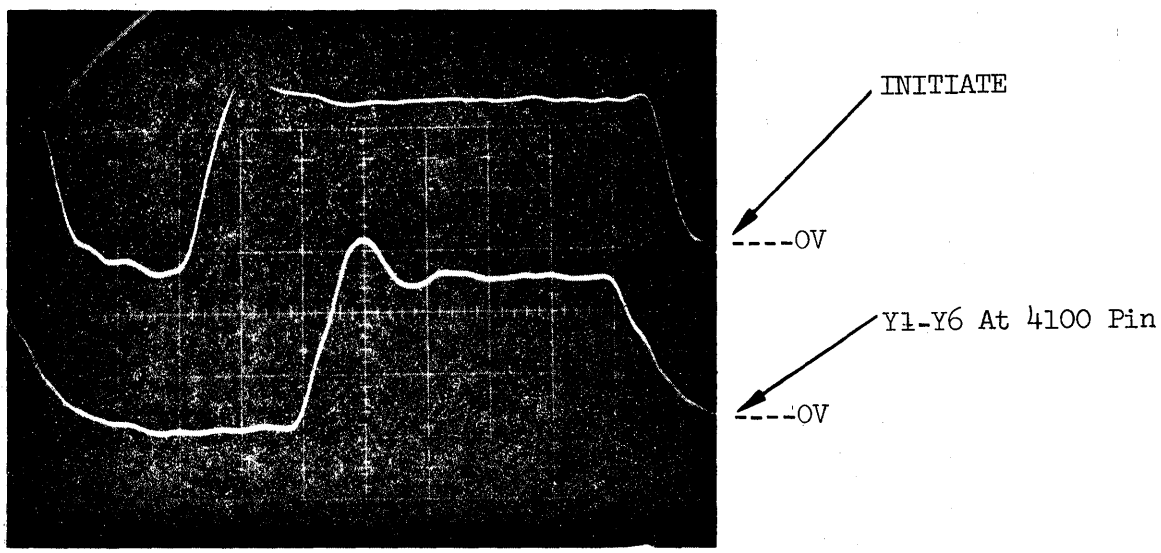
It is apparent that the six combinatorial address lines will be in the same state for the remaining seven substacks as they are for the substack under description; since only one substack is selected out of the eight, the other seven substacks must be prevented from being selected. The memory board organization does not allow selection of only one substack by the X address alone; instead the X address bits select a pair of substacks.

As stated before, select control signals X7, X8, X9, and X10 select substacks 0 and 4, 1 and 5, 2 and 6, and 3 and 7, respectively. Since for this example substack 0 is to be selected, X7 is also brought into the driving group of substack 4, allowing substacks 0 and 4 to be selected. When X7 is TRUE in the control board (high state) prior to being inverted by CTμL 852 of the control board, the other controls (X8, X9, and X10) are in a low state and, therefore, they are brought into the memory board in a high state. These three controls are OR-tied with the six combinatorial address lines at the outputs of CTμL 9965's and they force the inputs to TTμL 9016 to be in a high state. At this point, all the inputs to MμL 4100's of the substacks being selected by X8, X9, and X10 will be in a low state and, therefore, no selection of those MμL 4100's will take place. It is important to remember that selection of a MμL 4100 requires coincident X and Y addresses and, since two substacks are selected via the X address, the Y address must be prevented from being applied to the undesired substack of the pair selected via the X address. This is accomplished by the Y address drivers.

b. Y Address Buffer and Driver (Figure 3-23). Table 3-11 shows the Y address interface between the control board and a memory board. Figure 3-24 illustrates the timing for the Y1-Y6 address with respect to Initiate at a memory board. The CTμL 9816 gates are used to drive the Y address lines through CTμL 9856 buffers into the whole memory stack. The output of the CTμL 9816 gate representing any one of the six Y address lines or the substack select controls (Y7, Y8) is brought into four two-input CTμL 9856 gates. One of the inputs to CTμL 9856 is one of the six combinatorial Y address lines and the other is the substack select control Y7 for substacks 0 through 3 or Y8 for substacks 4 through 7. Since there are six Y address lines, there are three CTμL 9856 packages repeated four times.

Table 3-11. Y Address Interface between Control Board and Memory Board

SOURCE OF SIGNAL			DESTINATION OF SIGNAL			REMARKS
SIGNAL NAME	CONTROL BOARD DEVICE	CONTROL BOARD BOARD PIN NO.	SIGNAL NAME	MEMORY BOARD DEVICE	MEMORY BOARD BOARD PIN NO.	
Address Line Y1	CTμL 856	101	Address Line Y1	CTμL 816	101	In the control board the select controls are split into INNER and OUTER parts in order to select the appropriate substack in either the 64- or 32-bit mode of operation.
" " Y2		102	" " Y2		102	
" " Y3		103	" " Y3		103	
" " Y4		104	" " Y4		104	
" " Y5		105	" " Y5		105	
" " Y6		106	" " Y6		106	
Select Control Y7 IN		107	Select Control Y7		107	
" " Y7 OUT		108	" " Y8	CTμL 816	111	
" " Y8 IN		113				
" " Y8 OUT	CTμL 856	112				



1 V/cm, 20 ns/cm

Figure 3-24. Y1-Y6 Address at Memory Board versus Initiate

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

The output of the first group of three CTμL 9856 packages drives the 32 MμL 4100's located in the right half of substacks 0 through 3 and the second group of three CTμL 9856 packages drives the left half of substacks 0 through 3. The Y address lines from the output of CTμL 9816 to the input of CTμL 9856 are the same for the four groups; what differs, though, is the substack select controls. It can be said that Y7 enables the six Y address combinatorial bits to pass through groups 1 and 2 of CTμL 9856, while Y8 enables the same Y address bits to pass through groups 3 and 4 of CTμL 9856, thereby allowing substacks 4 through 7 to be selected. At this point, it is worth noting that whenever X7 is in a high state, Y8 is in a low state because these two controls are the result of the state of the single binary address bit AA11 (A11, $\overline{A11}$).

In the description of the operation of the X address drivers, it was assumed that word line 5 of substack 0 was to be selected. To continue this example, it will also be assumed that bit line 5 of the MμL 4100's of substack 0 is to be selected. Table 3-7 shows that address lines Y4, Y5, and Y6 must be in a high state at input pins 13, 14, and 15 of MμL 4100, while address lines Y1, Y2, and Y3 should remain in a low state. Y7 must be high in order to enable these address bits through groups 1 and 2; if Y7 is high, Y8 will be in a low state and will prevent the Y address from selecting any of the MμL 4100's in substacks 4 through 7. In this way, even though substacks 0 and 4 have been selected via the X address, the Y address together with Y7 substack select control selects only substacks 0 through 3, ignoring substack 4, which has already been selected via the X address. At this point, substack 0 has both X and Y addresses present; it can therefore be said that it has been selected for the read or the write operation.

3.2.2 Write Control

In order to write into the MμL 4100, a high level pulse is required. This pulse, called the write control pulse (CL6), is generated on the control board and is brought into the memory board at pins 118 and 150 (CL6 INNER) and 120 and 152 (CL6 OUTER). The requirement for providing four pins for the write control is imposed by the fact that the memory board is interchangeable through all four positions within the PEM and by so doing it is guaranteed that the memory board will be able to write data in either mode of operation (64-or 32-bit mode). CL6 INNER, therefore, if it is present, will be available for those memory boards at positions 1 and 2, while CL6 OUTER, if it is present, will be available for those memory boards at positions 3 and 4 (reference Figure 3-1).

Figure 3-25 shows that no matter what the position of the memory board in the PEM, CL6 will be available to two board pins which bring it to four CTμL 9856 buffers. Each buffer drives the write control into two substacks (32 MμL 4100's). It must be clarified that the memory board will not receive the write control (CL6) during the write operation in 32-bit mode, if it is in such a position in the PEM (inner or outer) that no write operation is requested. This, of course, is determined by the state of the two write controls (WRITE INNER and WRITE OUTER) from the MLU. If, for example, only the WRITE INNER control is present, that means that the write operation is going to be performed only on those two memory boards which are located at positions 1 and 2.

At this point, a very interesting situation exists. The two memory boards which do not receive the write control cannot write because, even though data from MLU will be available to these two boards and the write enable signal (CL5) will be low so that the I/O data lines will follow the write data, their corresponding MμL 4100's will not write this data because the write control (CL6 OUTER) is not present. At this time, when both MLU and PEM are in the write mode of operation, the enable of data out (CL3) is not present so that the data that is not accepted by the MμL 4100's of board positions 3 and 4 will not be able to be read out of the memory; therefore, it can be said that the absence of the write control from two memory boards causes no undesirable effects whatsoever.

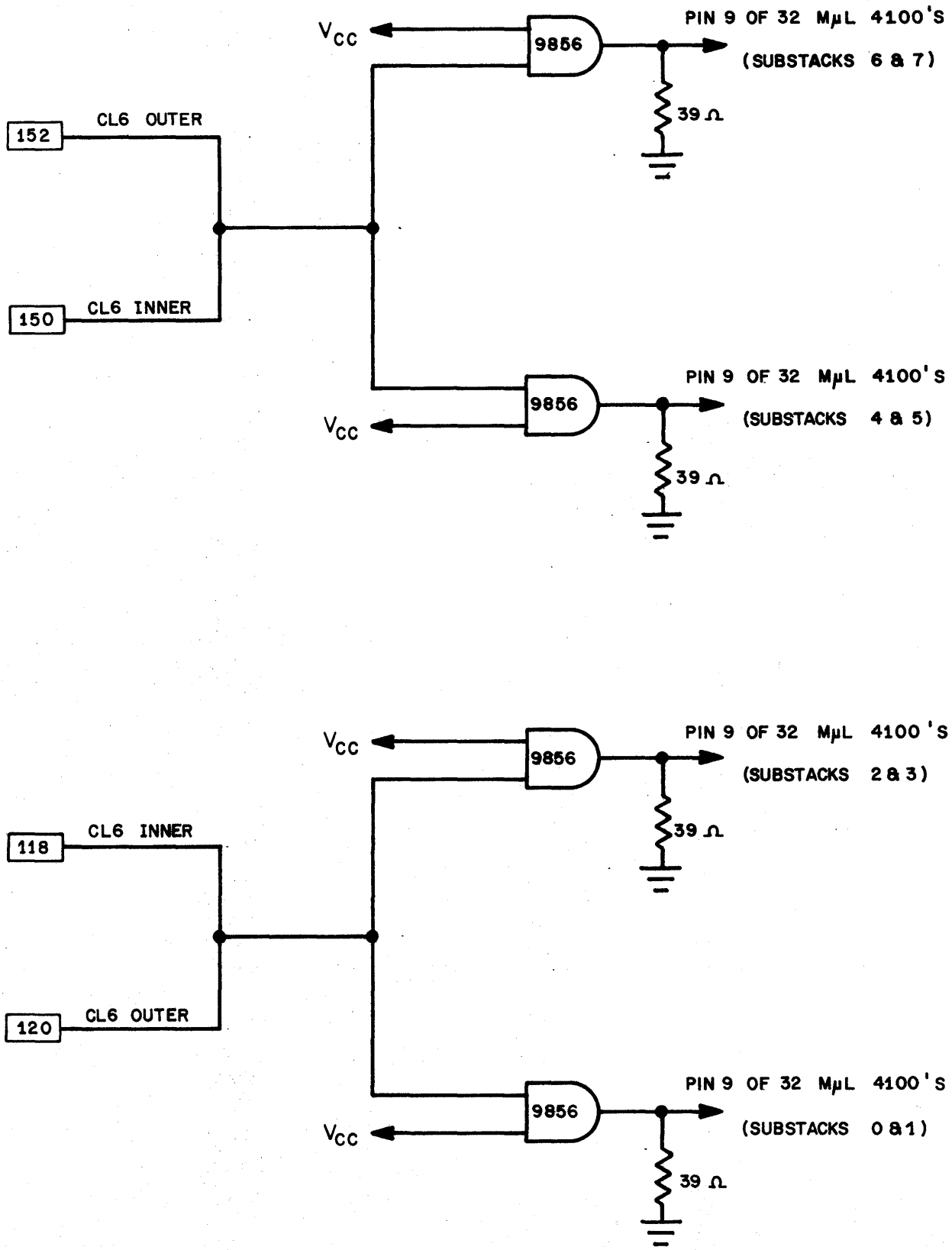
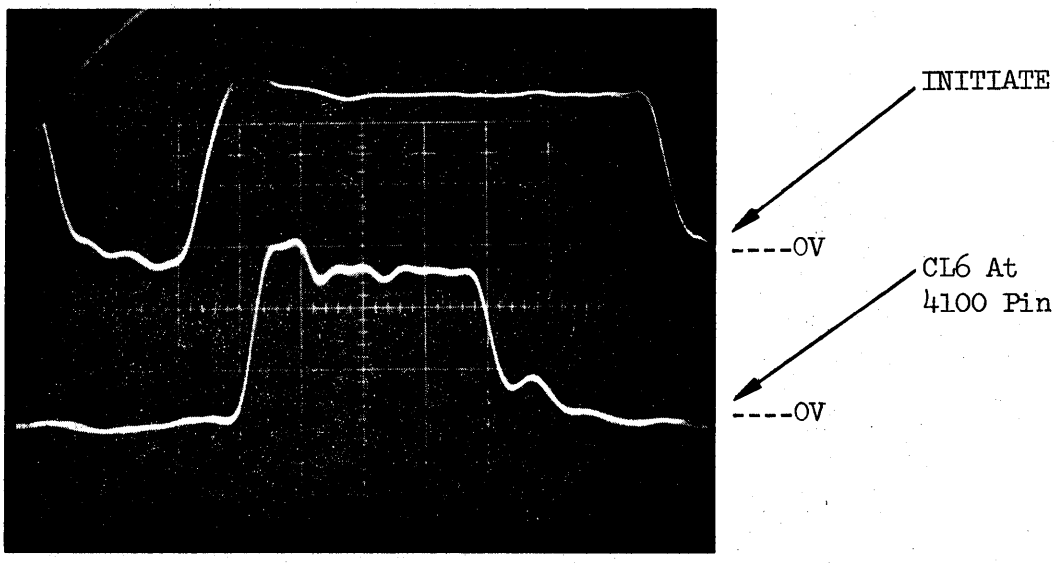


Figure 3-25. Memory Board WRITE Control



1 V/cm, 20 ns/cm

Figure 3-26. WRITE Control Signal CL6 at Memory Board MμL **4100 Pin versus Initiate**

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

3.2.3 Write Driver

It was said previously that whenever a read or write operation is requested, only one substack is selected per memory board. Because the same address is applied to all boards, the same substack is selected simultaneously on four memory boards, no matter what the mode of operation is (64-bit or 32-bit mode).

Each substack has 16 MμL 4100's and therefore it can accommodate 16 bidirectional (write/read) data lines. Each of the data lines is tied to pin 7 of eight MμL 4100's, each of which belongs to eight different substacks. For example, pin 7 of the 0^{th} MμL 4100 of substack 0 is connected to pin 7 of the 0th MμL 4100 of substacks 1 through 7 (Figure 3-29).

The difference here, however, is that even though these eight MμL 4100's located at eight substacks represent the same data bit for the same memory board, the same MμL 4100 of the other memory boards has been assigned a different data bit. In this way any given read/write pin of a particular MμL 4100 may represent four data bits, depending upon the position of the board within the PEM. Since the memory boards are interchangeable through all four positions within the PEM, the numbering of the data bits depends on the position of the board within the PEM and not on the board itself.

Since the PEM is part of the PU of ILLIAC IV, it follows the ILLIAC IV word format when it comes to numbering the memory data bits. Tables 2-1 and 2-2 of reference [4] show how the word is subdivided into inner and outer words. These tables will help explain the reason why the memory data has been assigned the numbers shown in Table 3-12. Figure 3-30 shows the memory board substack layout. Since there are eight substacks on each memory board, the whole memory stack can be viewed as being composed of eight substacks exactly the same as that shown in Figure 3-30. This is shown in Figure 3-29, where the same MμL 4100 number is repeated as many times as the number of substacks. The I/O data pin (pin 7) of each of those identically numbered MμL 4100's on each memory board is OR-tied to a common data line to provide the data bit as shown in Table 3-12.

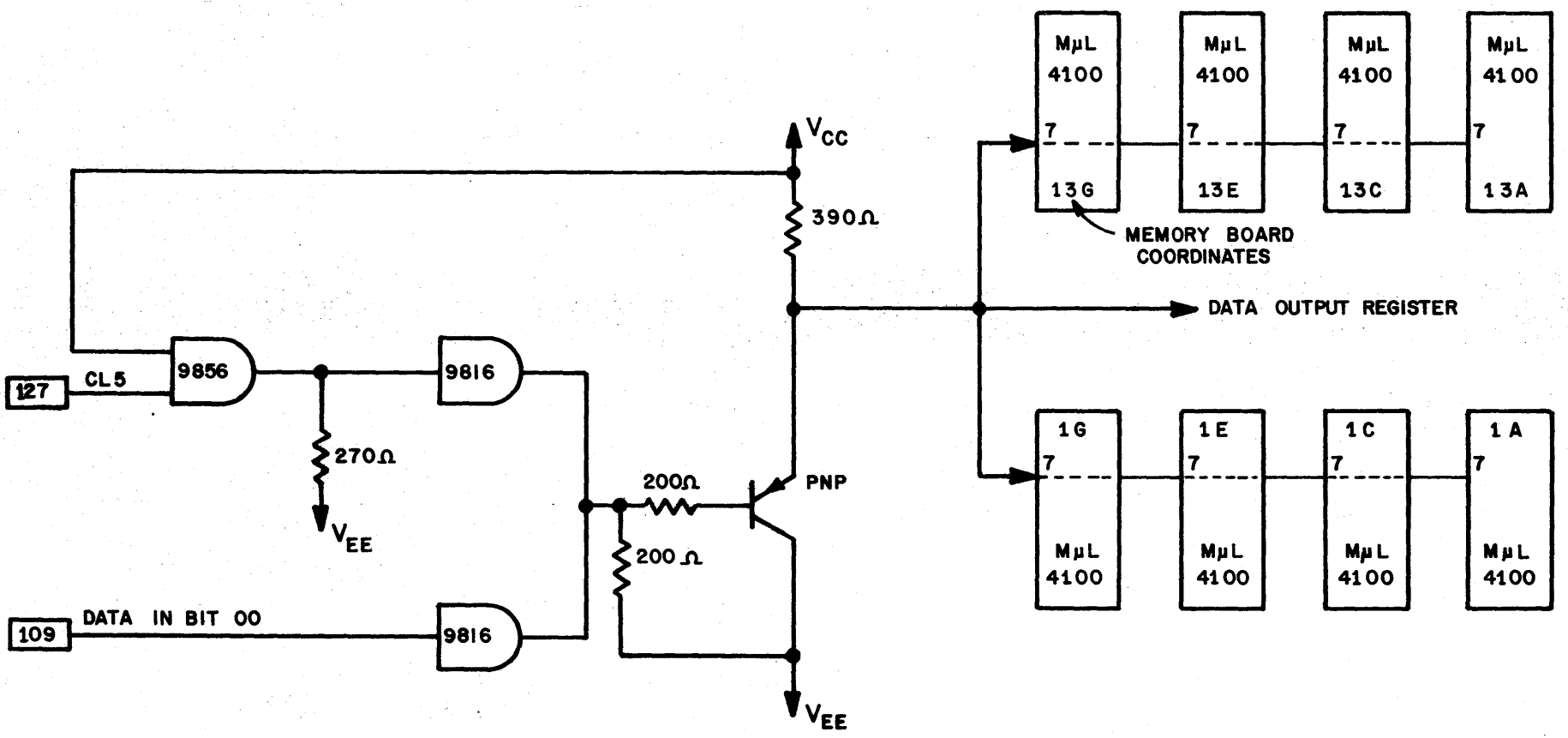
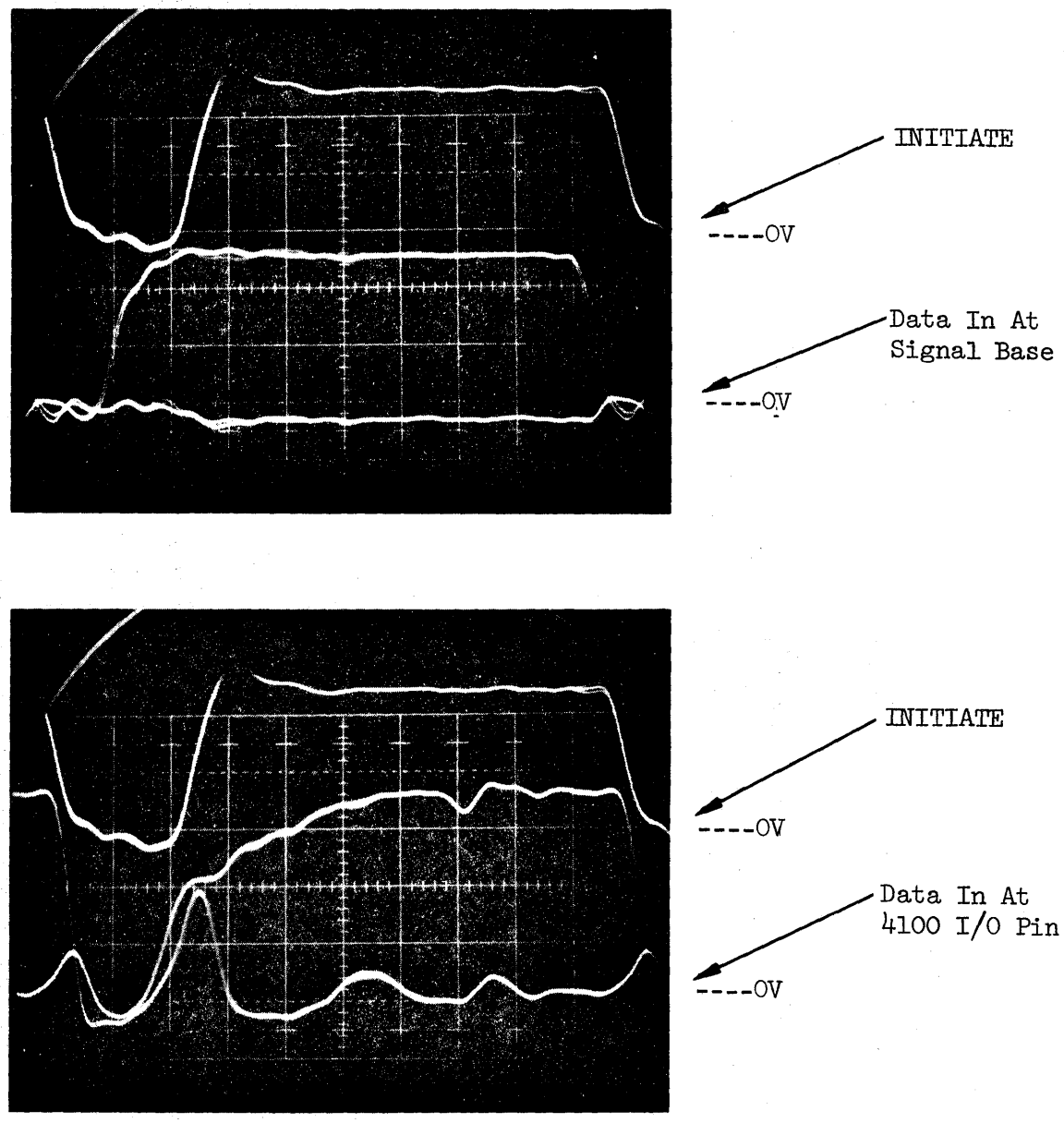


Figure 3-27. Memory Board WRITE Driver (One Bit Line).



1 v/cm, 20 ns/cm

Figure 3-28. Data In during WRITE operation

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

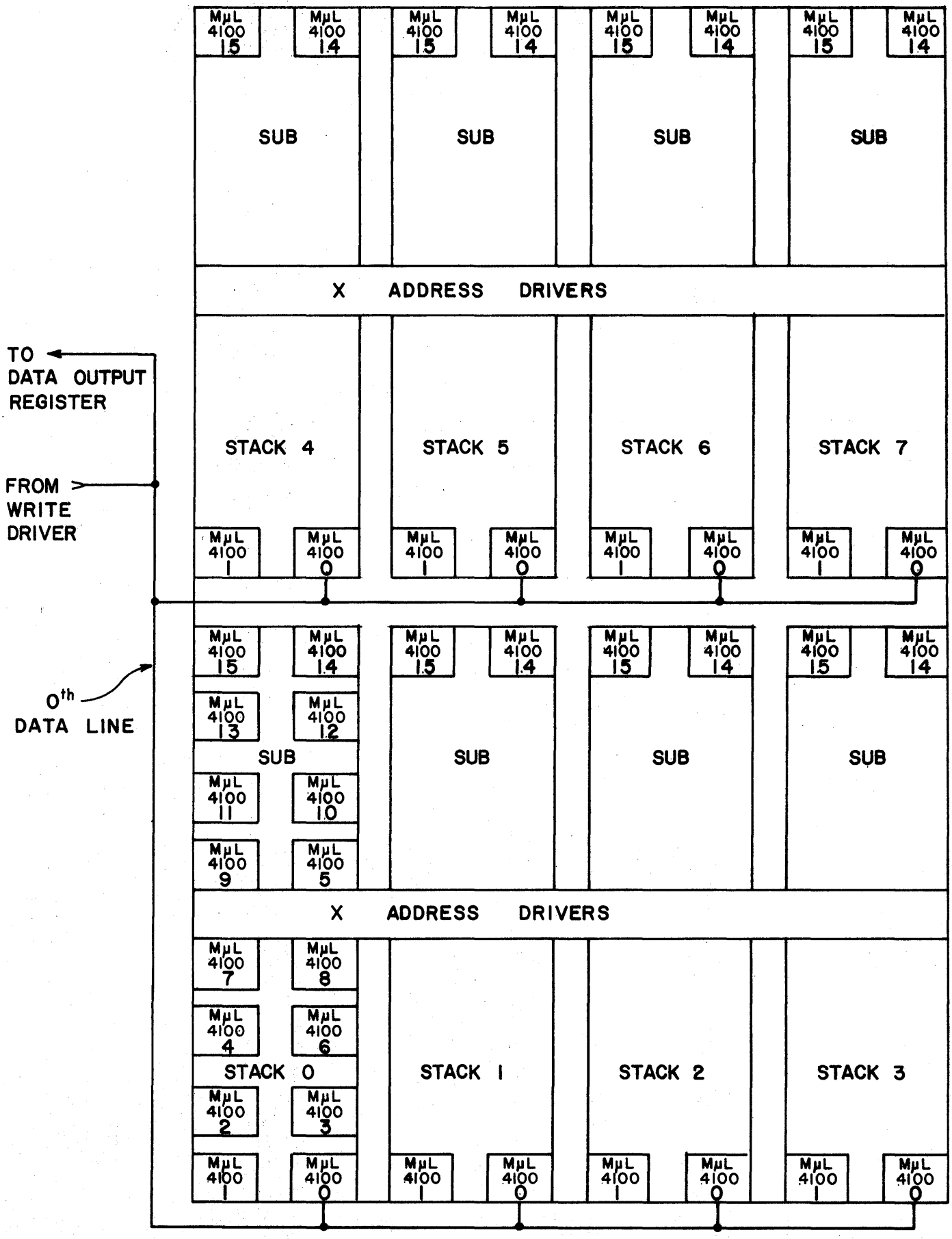


Figure 3-29. Memory Board 0th Data Line Layout

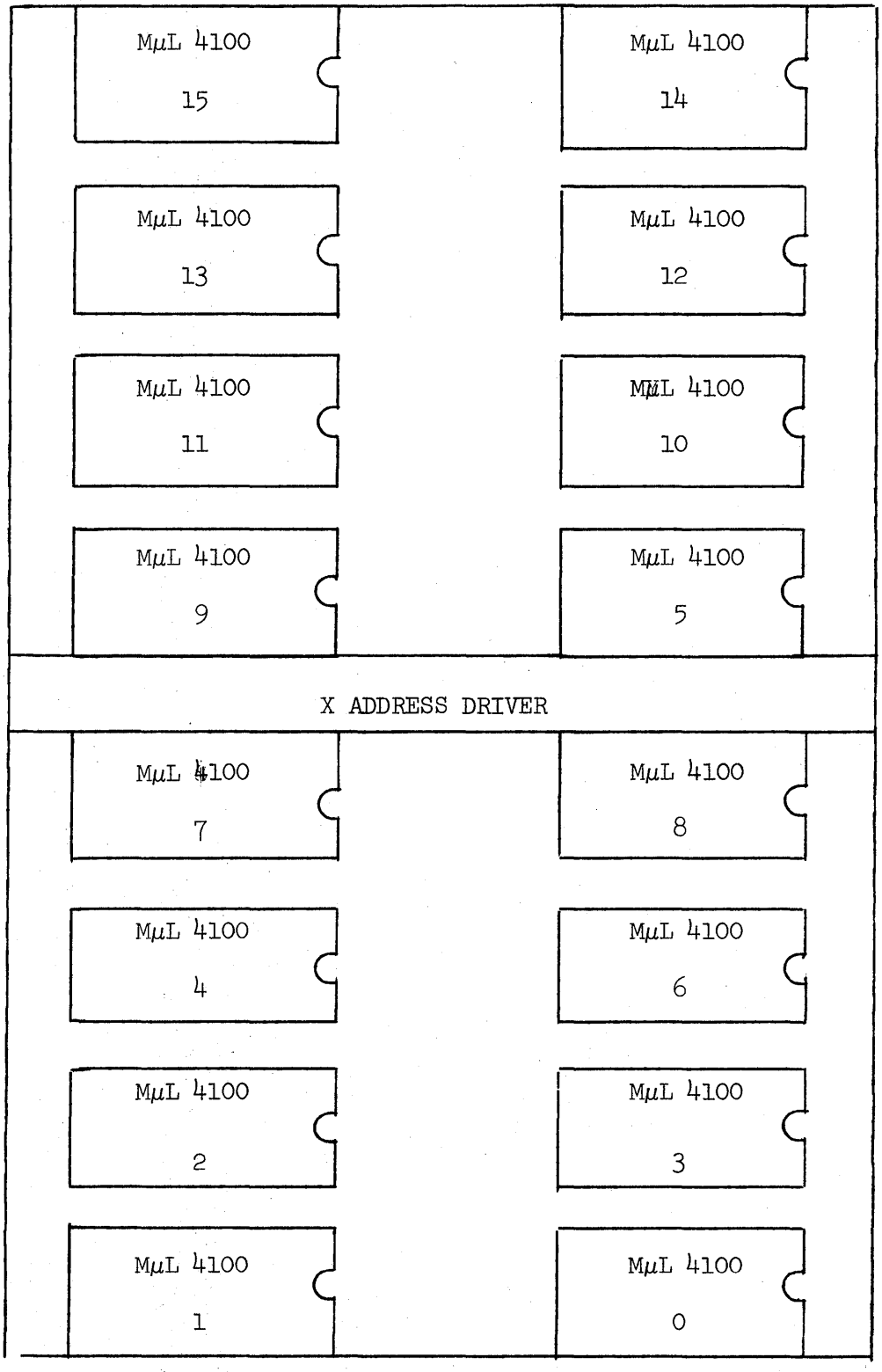


Figure 3-30. Memory Board Substack Layout

Table 3-12. Numbering Scheme of Memory Board I/O Data Bits

MμL 4100 NUMBER WITHIN SUBSTACK	DATA INPUT PIN	DATA OUTPUT PIN	INNER WORD BITS		OUTER WORD BITS		REMARKS
			BOARD POSITION #1	BOARD POSITION #2	BOARD POSITION #3	BOARD POSITION #4	
0	109	157		08	00		The numbering scheme of memory board data bits follows the ILLIAC IV word format convention in that bits 0-7 and 40-63 constitute the OUTER word, while bits 8-39 constitute the INNER word.
	110	158	09			01	
1	112	156		10	02		
	113	155	11			03	
2	115	161		12	04		
	116	160	13			05	
3	121	163		14	06		
	128	164	15			07	
4	123	167		16	40		
	124	166	17			41	
5	125	183		18	42		
	126	182	19			43	
6	128	168		20	44		
	129	169	21			45	
7	130	174		22	46		
	131	173	23			47	
8	132	176		24	48		
	133	177	25			49	
9	134	180		26	50		
	135	179	27			51	
10	136	188		28	52		
	137	189	29			53	
11	138	185		30	54		
	139	186	31			55	
12	140	194		32	56		
	141	195	33			57	
13	142	192		34	58		
	143	191	35			59	
14	154	200		37	61		
	153	201	36			60	
15	148	197		38	62		
	149	198	39			63	

Because the data lines on each memory board are directly associated with the MμL 4100 number on each substack, they can be assigned the same number as that assigned to the MμL 4100's. The numbering of these MμL 4100's is not sequential, because of layout limitations. The data arrives at the PEM in the order established by the MLU (MIR cards). Normally, MμL number 5 should occupy the position that MμL 4100 number 6 does, but because pins 125 and 126 are closer to the position of MμL 4100 number 5 than to position number 6 and, since the memory board is a two-layer board, any juggling around of different lines will inevitably require more board space. It is important to understand this because, due to layout limitations, many gates have been located on the board where space was available. The organization of these gates depends upon the importance of their respective signals from the timing point of view. The data lines have been kept relatively short inside the memory board in order to avoid any reflections due to a discontinuity in the characteristic impedance.

During the write operation (Figure 3-27), the write enable signal (CL5) is kept low so that the controlling signal will be the data-in bit. As an example, it will be assumed that bit 00 is a low level signal. Since CL5 is low, too, the base of the PNP write transistor will be negative enough to force the transistor to conduct. This transistor, which can be characterized as an emitter follower, will shift the level of data bit 00 by approximately 500 millivolts and I/O pin 7 of MμL 4100 will receive a low level signal (about -400 mV). After the write control has been present at pin 9 for at least 10 ns, the data will appear at the I/O pin and the selected memory cell will WRITE in bit 00.

If bit 00 happened to be a high level signal, the PNP write transistor would be forced to the cut-off state; thus the V_{cc}, less the voltage drop across the 390 Ω resistor, will appear at pin 7 of MμL 4100. The level of this signal is approximately 3.3 V. The voltage drop across the 390 Ω resistor is due to current flow through the PNP transistor and through the diode to the input CTμL 9966 gate, which, being a CTμL gate, is considered a current sink.

3.2.4 Data Output Register

Figures 3-27 and 3-31 show that each data line is associated with two main circuits, namely, the Write Driver during the write operation and the Data Output Register during the read operation. Table 3-12 shows the correspondence between a MμL 4100 position in the memory board substack with its corresponding data input and output pins and the data bits each MμL 4100 accommodates.

Previously it was said that during a write operation the enable of write data (CL5) is always low so that the I/O pin of the MμL 4100 will be able to follow the data when it appears on the base of the level shifter PNP write transistor. During a read operation, the write control signal CL6 becomes a low level, which forces CL5 to reverse polarity and therefore to become a positive pulse. CL5 at this time is the controlling signal because the two CTμL 9816 buffers, whose outputs are connected to the base of the PNP write transistor through a 200 Ω resistor, are tied together and form an OR function. The PNP transistor is brought into the cut-off state, bringing the I/O data line to a high state until the MμL 4100 is selected again for a read or write operation.

When a read cycle begins and the MμL 4100 is selected, the I/O pin of the MμL 4100 is in the high state. If a high level (ONE) was written into the selected MμL 4100, the I/O pin remains high. If a low level (ZERO) was written in the MμL 4100, the I/O pin goes low (-400 mV). This bit of data passes through the diode, which shifts its level by approximately 500 mV and is brought into one of the two OR-tied CTμL 9966's. At this time the enable data-out signal (CL3) is present and enables the data to two CTμL 9816 gates (pins 7 and 8). The data comes out of the CTμL 9816 buffer, which has excellent noise immunity and high driving capability, and is brought into the signal base. In this way, the data is available at the board pin within 188 ns from the time the Initiate signal was present on the control board. This period, which is called access time, is the minimum time possible because of the use of the CTμL 9966 gate and the CTμL 9816 buffer.

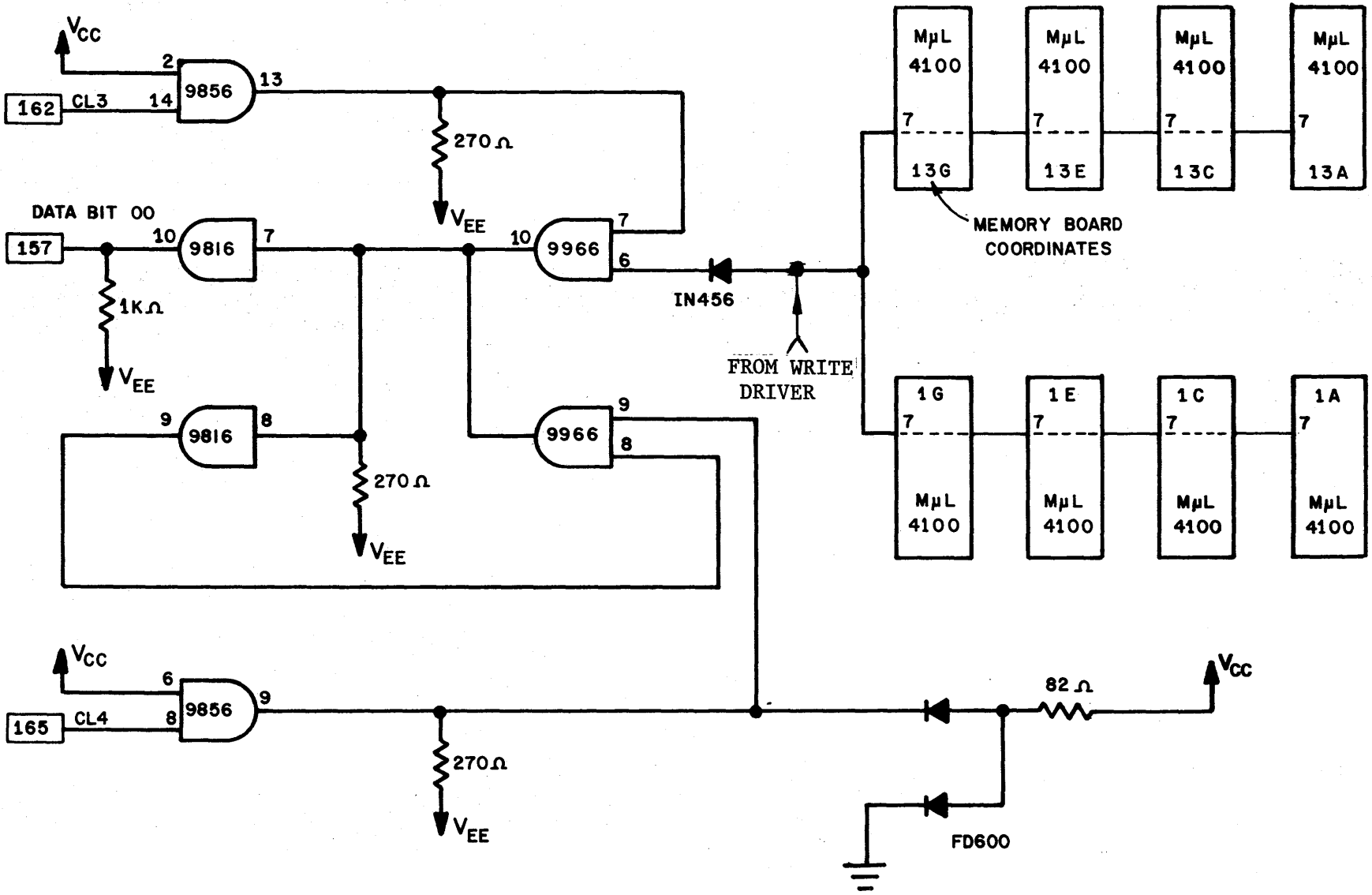


Figure 3-31. Memory Board Data Output Register (One Bit Line).

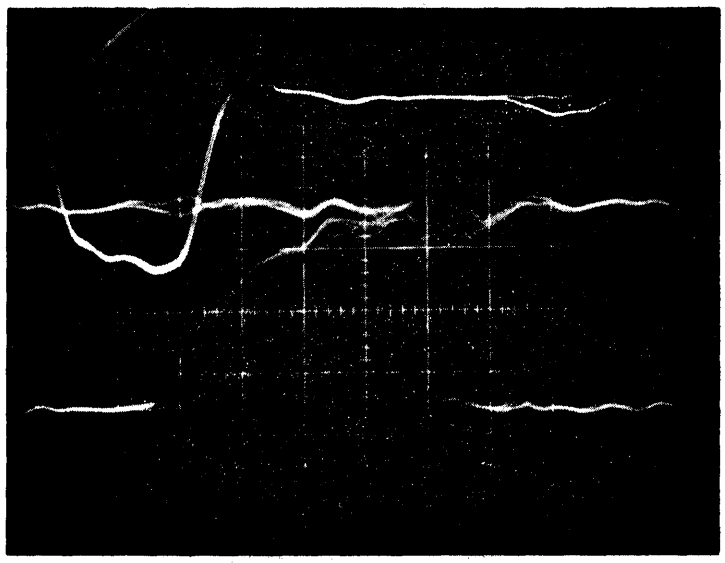
CL3 enables the data out of the PEM but it cannot hold the data as long as it needs to become stable. This necessitates the use of the data hold control signal (CL4) in order to hold the data in a stable state for about 120 ns after the next Initiate signal is present on the control board. Figure 3-31 shows that after the data is enabled (gated out) by CL3, it is applied to two CTμL 9816 buffers of the data register. Output 10 brings the data to board pin 157 (for bit 00) and output 9 of the other buffer brings the data back to input 8 of CTμL 9966 gate. As long as CL4 is not present this gate does practically nothing. This idle state lasts for about 60 ns because that is the width of CL3. Before CL3 is removed from input 7 of CTμL 9966 (about 5 ns), CL4 appears on input 9 of CTμL 9966. This pulse enables the data that is present on input 7 to go through and, since the two CTμL 9966 gates are OR-tied together, output 10 will continue providing stable data even if CTμL 9966 has no signal on inputs 6 and 7.

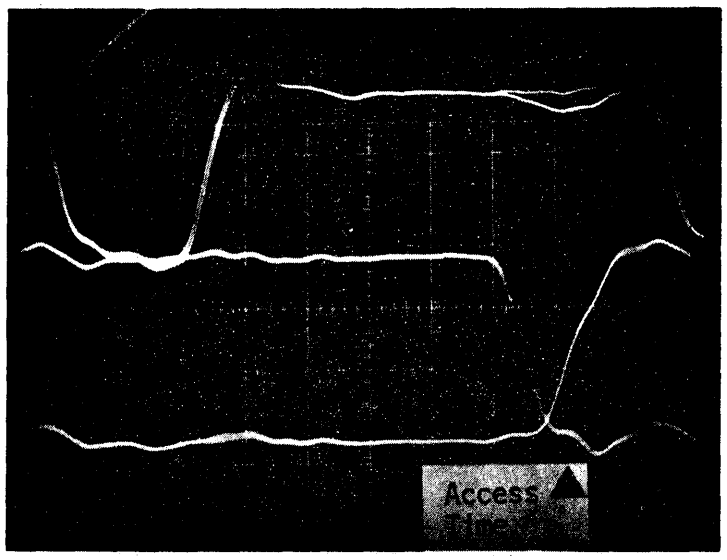
This data will be available for about 120 ns after the arrival of the next Initiate, because at that time CL4 is removed from input 9 of CTμL 9966, which along with the CTμL 9916 buffer, constitutes a feedback loop. The circuit, which consists mainly of two FD600 diodes with one diode connected to V_{cc} and ground and the other connected to V_{cc} and CL4, is used to clamp CL4 to a slightly higher than normal logic ZERO level in order to avoid glitches, which can be trapped in the data-out latch. This glitch is an internal characteristic peculiar to the CTμL 9966 gates.

Earlier, in the discussion of the read and write operations, it was stated that the I/O line for any MμL 4100 package remains high during the time that the package is not selected. This statement may call for some clarification. If the PEM is not requested to perform a read or write operation, no memory board substacks will be selected. At such a time, a small amount of current flows through the 390 Ω resistor, which is connected to V_{cc} and to the emitter of the PNP write transistor, which in turn is connected to the data line, and provides a voltage drop, forcing the I/O data line to remain high (the PNP transistor is not completely cut off). This external 390 Ω resistor is needed to fulfill a requirement imposed by the peculiar characteristic of MμL 4100's which, in order to achieve word expansion, require each I/O data line (on chip),

which is connected to an uncommitted collector gate, to have an external pull-up resistor. When a substack is selected, the I/O data line will follow the data to be written into the MμL 4100 on a write cycle, while on a read cycle the I/O data line will be in the state of the data bit which is being read out from the selected MμL 4100. Thus the I/O data lines of every group of eight MμL 4100's that are tied together to the memory board data I/O line form an AND function.

The MμL 4100 is known to be a TTμL-compatible device and, therefore, its I/O data line at pin 7 is considered to provide a signal at a TTμL level. However, during the write operation the data is buffered into the MμL 4100 by a CTμL 9816 gate (buffer). At this point, the signal from the CTμL level is shifted to the corresponding TTμL level by the emitter-follower PNP write transistor, which provides a one-diode voltage drop; because of its performance, this transistor acts as a level shifter. Figure 2-8 shows signal level requirements. During the read operation, the data being read out from the MμL 4100 is at the TTμL level, but, by using the IN 456 diode, which provides a sufficient voltage drop, the signal is shifted to the CTμL level prior to entering the CTμL 9966 gate. The use of pull-down resistors (resistors are connected to -2.0 V), as shown in Figures 3-27 and 3-31, is required because of characteristics peculiar to the CTμL family of devices.





1 V/cm, 20 ns/cm

Figure 3-32. Data Out during READ operation

Photographs will be provided at a future time. If copies are needed now, please contact Becky Vogl.

3.3 SIGNAL BASE

This board is used to provide the interface for all signals exchanged between the PEM and MLU. Figure 3-1 shows the position of the signal base with respect to the other boards in the PEM. The signal base receives address and control signals from the MLU through a pair of paddle boards (P1 and P2). The signal base sends these signals on to the control board during read and write operations. Write data (64 or 32 bits) is also received from the MLU through the paddle boards and signal base; it is then distributed to the memory boards. Read data (64 bits) is sent from the memory boards to the MLU through the signal base and paddle boards.

The connection of the signal base with the paddle boards and memory and control boards is performed via cam-operated connectors located on the signal base. This feature makes it possible for the memory and control boards to be plugged into the PEM and, because the pins on the signal base and the pluggable boards have been assigned the same numbers, the memory boards can be interchanged through all four positions without any problem; when bit numbers were assigned to the boards, the position of the memory board was taken into account but not the memory board itself.

Because the logic levels of the signals between MLU and PEM are different, the ECL level signals from MLU pass through the Up Converter cards in the MLU to attain CTμL logic levels prior to their arrival in the PEM, whereas all CTμL level data bits from the PEM are brought into the Down Converter cards of the MLU to obtain ECL levels.

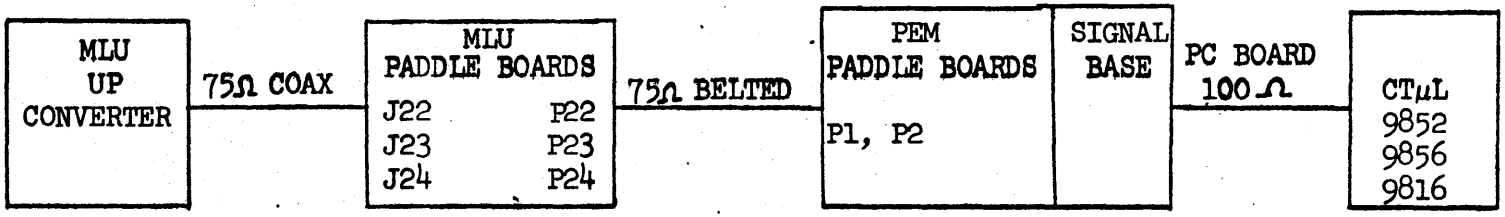
During a write operation (Figure 3-33a), the PEM receives 11 bits of address, three control signals, and 64 or 32 bits of data. These signals (Tables 3-13, 3-14, and 3-15) leave the MLU Up Converter through connectors J16, J17, and J18 via 8-inch, 75 Ω coaxial cables and are brought into MLU paddle boards P22, P23, and P24 at connectors J22, J23, and J24. From the other side of these MLU paddle boards, these signals are driven through 75 Ω belted cables to PEM paddle boards P1 and P2. At this point, it should be mentioned that each of the three MLU paddle boards has four straps (ribbons) with the straps of P22 and two straps of P23 connected to PEM paddle board P1 and the straps of P24 and the other two straps of P23 connected to PEM paddle board P2. In the belted cables (ribbons) the signal lines run adjacent to the ground lines in order to provide the

minimum possible inductance and keep the characteristic impedance at 75 Ω. The signal grounds to the ground lines of the belted cables are distributed through the signal base and PEM paddle boards P1 and P2 because, as is explained later in the description of power distribution, the ground grids on the memory and control boards provide signal grounds, which connect to the signal base through the power distribution board of the PEM.

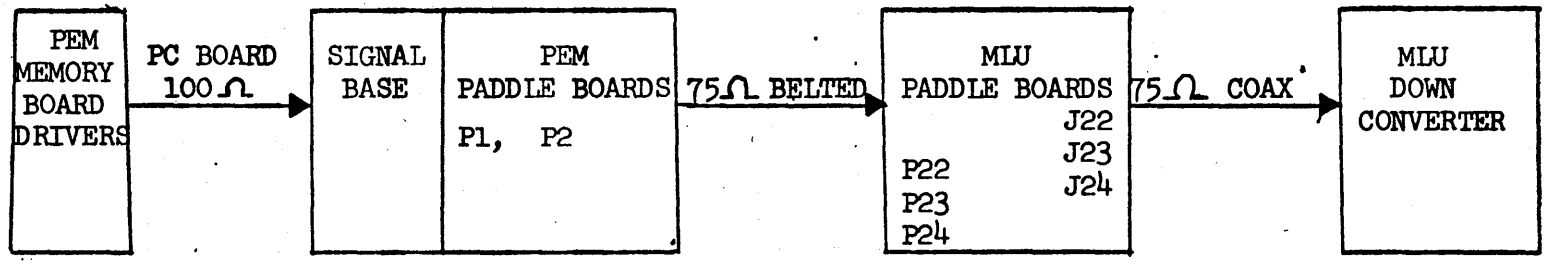
On PEM paddle boards P1 and P2, each signal is terminated by a 75 Ω resistor in order to keep the characteristic impedance in balance. The address and control signals are then brought through the signal base into the control board. The data is likewise brought through the signal base to the memory boards.

During read operations (Figure 3-33), the PEM receives the control signals to initiate the read operation and the address of the location containing the requested data. The data is driven out of each memory board by CTμL 9816 gates to PEM paddle boards P1 and P2 (Tables 3-16 and 3-17) and from there via 75 Ω belted cables into MLU paddle boards P22, P23, and P24 at connectors J22, J23, and J24. From that point, the data is driven through 75 Ω coaxial cables to the Down Converters of the Memory Information Register (MIR) cards of the MLU, where the signals are down-converted to ECL level.

Tables 3-18, 3-19, 3-20, and 3-21 present the interface between the PEM and the PEM Exerciser.



(a)



(b)

Figure 3-33. PEM-MLU Interface (a) WRITE Operation
(b) READ Operation

Table 3-13. Address and Control Signals from Up Converter Cards to PEM Control Board

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL	
	MLU (Up Converter)	CONNECTOR & PIN NO.	PEM PADDLE BOARD	PEM CONTROL BOARD PIN NO.
MYW-WO5--0	J16 - C03	J22 - C39	P1	145
06	J16 - D14	J24 - C33	P2	181
07	J17 - C03	J22 - B38	P1	178
08	J17 - D14	J24 - D42	P2	175
09	J16 - C27	J22 - A47	P1	172
10	J16 - B02	J24 - C29	P2	184
11	J17 - C27	J22 - B24	P1	187
12	J17 - B02	J24 - D06	P2	199
13	J17 - D36	J22 - A11	P1	196
MYW-W14--0	J18 - C27	J24 - C15	P2	193
MYW-W15--0	J18 - B02	J22 - B20	P1	190
MWOUTEN--1	J17 - A27	J24 - B30	P2	151
MWINNEN--1	J17 - B06	J22 - D38	P1	146
MINITPL--1	J17 - B44	J24 - B36	P2	147

Table 3-14. WRITE Data from Up Converter Cards to PEM (OUTER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	MLU (Up Converter)	CONNECTOR & PIN NO.	PEM PADDLE BOARD	PEM MEMORY BOARD #	PIN NO.
MOWFWOO--1	J17 - B26	J23 - C45	P2	3	109
↑ 01	↑ - C15	↑ - D44	↑	4	110
02	- B24	- C41		3	112
03	- A11	- C39		4	113
04	- A47	- D36		3	115
05	- A29	- C35		4	116
06	↓ - D32	- C27		3	121
07	J17 - B48	- D26		4	122
40	J18 - A23	- D24		3	123
41	↑ - D44	- C23		4	124
42	- C17	- C21		3	125
43	- B12	- D20		4	126
44	- D48	- C17		3	128
45	- C29	- C15		4	129
46	- B36	- D14		3	130
47	- A45	- D12		4	131
48	- B26	- C11		3	132
49	- C15	- C09		4	133
50	- B24	- D08		3	134
51	- A11	- D06		4	135
52	- A47	- C05		3	136
53	- A29	↓ - C03		4	137
54	- B32	J23 - D02		3	138
55	- B48	J24 - B48		4	139
56	- A27	↑ - A47		3	140
57	- B06	- A45		4	141
58	- A21	- B44		3	142
59	- A15	- B42		4	143
60	- B44	- A27		4	153
61	- C45	- B26		3	154
↓ 62	↓ - D36	↓ - A35	↓	3	148
MOWFW63--1	J18 - C47	J24 - A33	P2	4	149

Table 3-15. WRITE Data from Up Converter Cards to PEM (INNER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	MLU (Up Converter)	CONNECTOR & PIN NO.	PEM PADDLE BOARD	PEM MEMORY BOARD #	PIN NO.
MOWFW08--1	J16 - A23	J23 - A45	P1	2	109
↑ 09	↑ - D44	↑ - B44	↑	1	110
10	- C17	- A41		2	112
11	- B12	- A39		1	113
12	- D48	- A36		2	115
13	- C29	- A35		1	116
14	- B36	- A27		2	121
15	- A45	- B26		1	122
16	- B26	- B24		2	123
17	- C15	- A23		1	124
18	- B24	- A21		2	125
19	- A11	- B20		1	126
20	- A47	- A17		2	128
21	- A29	- A15		1	129
22	- D32	- B14		2	130
23	- B48	- B12		1	131
24	- A27	- A11		2	132
25	- B06	- A09		1	133
26	- A21	- B08		2	134
27	- A15	- B06		1	135
28	- B44	- A05		2	136
29	- C45	↓ - A03		1	137
30	↓ - D36	J23 - B02		2	138
31	J16 - C47	J22 - D48		1	139
32	J17 - A23	↑ - C47		2	140
33	↑ - D44	- C45		1	141
34	- C17	- D44		2	142
35	- B12	- D42		1	143
36	- D48	- C27		1	153
37	- C29	- D26		2	154
↓ 38	↓ - B36	↓ - C35	↓	2	148
MOWFW39--1	J17 - A47	J22 - C33	P1	1	149

Table 3-16. READ Data from PEM to MIR Cards (OUTER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	MEMORY BOARD #	PIN NO.	PEM PADDLE BOARD	CONNECTOR & PIN NO.	MLU (MIR)
MIWFW00--1	3	157	P2	J24 - A21	J08 - C27
↑ 01	4	158	↑	↑ - B20	↑ - C21
02	3	156		- A23	- C29
03	4	155		- B24	- D32
04	3	161		- A15	- D30
05	4	160		- A17	- C35
06	3	163		- B12	↓ - B36
07	4	164		- A11	J08 - A39
40	3	167		- B06	J09 - C27
41	4	166		- B08	↑ - C21
42	3	183		- D30	- C39
43	4	182		- D32	- D32
44	3	168		- A05	- D30
45	4	169		- A03	- C35
46	3	174		- D44	↓ - B36
47	4	173		- C45	J09 - A39
48	3	176		- C41	J10 - C27
49	4	177		- C39	↑ - C21
50	3	180		- C35	- C29
51	4	179		- D36	- D32
52	3	188		- C23	- D30
53	4	189		- C21	- C35
54	3	185		- C27	↓ - B36
55	4	186		- D26	J10 - A39
56	3	194		- D14	J11 - C27
57	4	195		- D12	↑ - C21
58	3	192		- C17	- C29
59	4	191		- D18	- D32
60	4	201		- C03	- D30
61	3	200		- C05	- C35
↓ 62	3	197	↓	↓ - C09	↓ - B36
MIWFW63--1	4	198	P2	J24 - D08	J11 - A39

Table 3-17. READ Data from PEM to MIR Cards (INNER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	MEMORY BOARD #	PIN NO.	PEM PADDLE BOARD	CONNECTOR & PIN NO.	MLU (MIR)
MIWFW08--1	2	157	P1	J22 - C21	J04 - C27
09	1	158		- D20	- C21
10	2	156		- C23	- C29
11	1	155		- D24	- D32
12	2	161		- C15	- D30
13	1	160		- C17	- C35
14	2	163		- D12	- B36
15	1	164		- C11	J04 - A39
16	2	167		- D06	J05 - C27
17	1	166		- D08	- C21
18	2	183		- B30	- C29
19	1	182		- B32	- D32
20	2	168		- C05	- D30
21	1	169		- C03	- C35
22	2	174		- B44	- B36
23	1	173		- A45	J05 - A39
24	2	176		- A41	J06 - C27
25	1	177		- A39	- C21
26	2	180		- A35	- C29
27	1	179		- B36	- D32
28	2	188		- A23	- D30
29	1	189		- A21	- C35
30	2	185		- A27	- B36
31	1	186		- B26	J06 - A39
32	2	194		- B14	J07 - C27
33	1	195		- B12	- C21
34	2	192		- A17	- C29
35	1	191		- B18	- D32
36	1	201		- A03	- D30
37	2	200		- A05	- C35
38	2	197		- A09	- B36
MIWFW39--1	1	198	P1	J22 - B08	J07 - A39

Table 3-18. WRITE Data from PEMX to PEM (OUTER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	PEMX BACKPLANE PIN NO.	CONNECTOR & PIN NO.	PEM PADDLE BOARD	PEM MEMORY BOARD #	PIN NO.
MOWFW00--1	1302-08	J5A - 645	P2	3	109
↑ 01	1302-19	↑ - 044	↑	4	110
02	1402-08	- 641		3	112
03	1402-20	- 639		4	113
04	1302-09	- D36		3	115
05	1302-18	- C35		4	116
06	1402-07	- C27		3	121
07	1402-21	- D26		4	122
40	1102-13	- D24		3	123
41	1102-20	- C23		4	124
42	1202-14	- C21		3	125
43	1202-22	- D20		4	126
44	1102-12	- C17		3	128
45	1102-22	- C15		4	129
46	1202-11	- D14		3	130
47	1202-23	- D12		4	131
48	1102-38	- C11		3	132
49	1102-49	- C09		4	133
50	1202-38	- D08		3	134
51	1202-50	- D06		4	135
52	1102-39	- C05		3	136
53	1102-48	↓ - C03		4	137
54	1202-37	J5A - D02		3	138
55	1202-51	J6A - B48		4	139
56	1102-43	↑ - A47		3	140
57	1102-50	- A45		4	141
58	1202-44	- B44		3	142
59	1202-52	- B42		4	143
60	1102-42	- A27		4	153
61	1102-52	- B26		3	154
↓ 62	1202-41	↓ - A35	↓	3	148
MOWFW63--1	1202-53	J6A - A33	P2	4	149

Table 3-19. WRITE Data from PEMX to PEM (INNER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	PEMX BACKPLANE PIN NO.	CONNECTOR & PIN NO.	PEM PADDLE BOARD	PEM MEMORY BOARD #	PIN NO.
MOWFW08--1	1302-13	J5A - A45	P1	2	109
↑ 09	1302-20	↑ - B44	↑	1	110
10	1402-14	- A41		2	112
11	1302-22	- A39		1	113
12	1302-12	- B36		2	115
13	1302-22	- A35		1	116
14	1402-11	- A27		2	121
15	1402-23	- B26		1	122
16	1302-38	- B24		2	123
17	1302-49	- A23		1	124
18	1402-48	- A21		2	125
19	1402-50	- B20		1	126
20	1302-39	- A17		2	128
21	1302-38	- A15		1	129
22	1402-37	- B14		2	130
23	1402-51	- B12		1	131
24	1302-43	- A11		2	132
25	1302-50	- A09		1	133
26	1402-44	- B08		2	134
27	1402-52	- B06		1	135
28	1302-42	- A05		2	136
29	1302-52	↓ - A03		1	137
30	1402-41	J5A - B02		2	138
31	1402-53	J4A - D48		1	139
32	1102-08	↑ - C47		2	140
33	1102-19	- C45		1	141
34	1202-08	- D44		2	142
35	1202-20	- D42		1	143
36	1102-09	- C27		1	153
37	1102-18	- D26		2	154
↓ 38	1202-07	↓ - C35	↓	2	148
MOWFW39--1	1202-21	J4A - C33	P1	1	149

Table 3-20. READ Data from PEM to PEMX (OUTER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	PEM MEMORY BOARD #	PIN NO.	PEM PADDLE BOARD	CONNECTOR & PIN NO.	PEMX BACKPLANE PIN NO.
MIWFW00--1	3	157	P2	J6A - A21	1302-05
↑ 01	4	158	↑	↑ - B20	1302-23
02	3	156		- A23	1402-04
03	4	155		- B24	1402-18
04	3	161		- A15	1302-35
05	4	160		- A17	1302-53
06	3	163		- B12	1402-34
07	4	164		- A11	1402-48
40	3	167		- B06	1102-07
41	4	166		- B08	1102-25
42	3	183		- D30	1202-13
43	4	182		- D32	1202-28
44	3	168		- A05	1102-37
45	4	169		- A03	1102-55
46	3	174		- D44	1202-43
47	4	173		- C45	1202-58
48	3	176		- C41	1102-06
49	4	177		- C39	1102-24
50	3	180		- C35	1202-01
51	4	179		- D36	1202-17
52	3	188		- C23	1102-36
53	4	189		- C21	1102-54
54	3	185		- C27	1202-31
55	4	186		- D26	1202-47
56	3	194		- D14	1102-15
57	4	195		- D12	1102-29
58	3	192		- C17	1202-06
59	4	191		- D18	1202-24
60	4	201		- C03	1102-45
61	3	200		- C05	1102-59
↓ 62	3	197	↓	↓ - C09	1202-36
MIWFW63--1	4	198	P2	J6A - D08	1202-54

Table 3-21. READ Data from PEM to PEMX (INNER Word)

SIGNAL NAME	SOURCE OF SIGNAL		DESTINATION OF SIGNAL		
	PEM MEMORY BOARD #	PIN NO.	PEM PADDLE BOARD	CONNECTOR & PIN NO.	PEMX BACKPLANE PIN NO.
MIWFW08--1	2	157	P1	J4A - C21	1302-07
↑ 09	1	158	↑	↑ - D20	1302-25
10	2	156		- C23	1402-13
11	1	155		- D24	1402-28
12	2	161		- C15	1302-37
13	1	160		- C17	1302-55
14	2	163		- D12	1402-43
15	1	164		- C11	1402-58
16	2	167		- D06	1302-06
17	1	166		- D08	1302-24
18	2	183		- B30	1402-01
19	1	182		- B32	1402-17
20	2	168		- C05	1302-36
21	1	169		- C03	1302-54
22	2	174		- B42	1402-31
23	1	173		- A45	1402-47
24	2	176		- A41	1302-15
25	1	177		- A39	1302-29
26	2	180		- A35	1402-06
27	1	179		- B36	1402-24
28	2	188		- A23	1302-45
29	1	189		- A21	1302-59
30	2	185		- A27	1402-36
31	1	186		- B26	1402-54
32	2	194		- B14	1102-05
33	1	195		- B12	1102-23
34	2	192		- A17	1202-04
35	1	191		- B18	1202-18
36	1	201		- A03	1102-35
37	2	200		- A05	1102-53
↓ 38	2	197	↓	↓ - A09	1202-34
MIWFW39--1	1	198	P1	J4A - B08	1202-48

SECTION 4

POWER DISTRIBUTION

The PEM, being a subunit of the Processing Unit (PU), receives power from external PU power supplies in the following way. In each PU cabinet there are two power supplies providing +4.8 V and ground, two power supplies providing -2.0 V and ground, and eight power supplies used to provide +4.52 V. These power supplies provide power for eight PU's, which belong to one of the eight PU cabinets of the Quadrant. The ground of all the power supplies in each cabinet is connected to the ground of the controller of the PU cabinet, which in turn connects to the ground of all the other controllers (eight) which along with the ground of the B6700 Control Computer are connected to Earth, thus establishing a ground (Earth) tree.

From the PU cabinet the above voltages are brought into the Processing Unit in two groups. The first group brings +4.8 V, ground, and -2.0 V, and the second group brings +4.52 V only. In the upper part of the PU is a power supply shunt regulator which uses the 4.52 V to provide +1.32 V and -3.20 V to the MLU. Distribution of these voltages is discussed in the MLU Principles of Operation Manual.

On the Power Supply Shunt Regulator of each Processing Element there are two power busses used to transfer the power (grouped voltages) into the individual subunits of the Processing Unit, namely, Processing Unit, Memory Logic Unit, and PEM. At the other end of the two power busses, where the voltages are distributed to the subunits of the Processing Unit, the two grounds are connected together to make sure that all three subunits have a common, solid ground. From this point the ground is brought into the Processing Element, Memory Logic Unit, and PEM; the -2.0 V is brought into the PEM only, whereas the +4.8 V is distributed to both the PEM and MLU, because the MLU uses +4.8 V for the circuits, which provide DOWN and UP conversion on the signal level, whenever the signal logic level changes (CTμL, ECL).

The power for the PEM (Figure 4-1) is distributed from the Power Bus through the three designated leads into the power distribution board, which contains large ground and voltage planes of 2 ounces of copper. In order to reduce the voltage drop, the power base picks up the power from the power distribution board through approximately 110 contact pins and distributes the power to the control and memory boards (Figures 4-1, 4-2) through its cam-operated connectors.

Each memory board and the control board as well use three intermeshed power grids for V_{cc} = +4.8 V, V_{ee} = -2.0 V, and ground for power distribution to CTμL, TTμL, and MμL 4100 devices. In order to achieve low DC characteristic impedance, these power lines (V_{cc}, V_{ee}, and ground) run parallel to one another. Figures 4-1 and 4-2 show only the main power busses, but the reader should realize that there are power distribution lines picking up the power from their corresponding power busses on each board. Any sudden variation in voltage due to spikes of high frequency noise is greatly minimized by the user of high-frequency filter (bypass) capacitors, connected between V_{cc}/V_{ee} and ground and also between V_{cc} and V_{ee}.

The power grid network as it has been implemented on the control and memory boards provides an effective ground shield, which helps to obtain, in the case of relatively long transmission signal lines, a characteristic impedance of about 100 Ω [3]. It has been calculated (by Dr. Frank Greene) that the total power dissipation is distributed as follows:

1) One Memory Board:
   a. CTμL, TTμL, and WRITE transistors require 3.66 amperes at +4.8 V and 2.98 amperes at -2.0 V.
   b. MμL 4100's require 16.00 amperes at +4.8 V.
2) Control Board:
   a. Because the Control Board uses only CTμL devices, it requires 4.53 amperes at +4.8 V and 3.56 amperes at -2.0 V.

Since the above numbers represent worst-case PEM power requirements, it can be concluded that a power dissipation of about 400 watts per PEM is equivalent to 3 milliwatts per memory bit (400 watts ÷ 131072 bits).

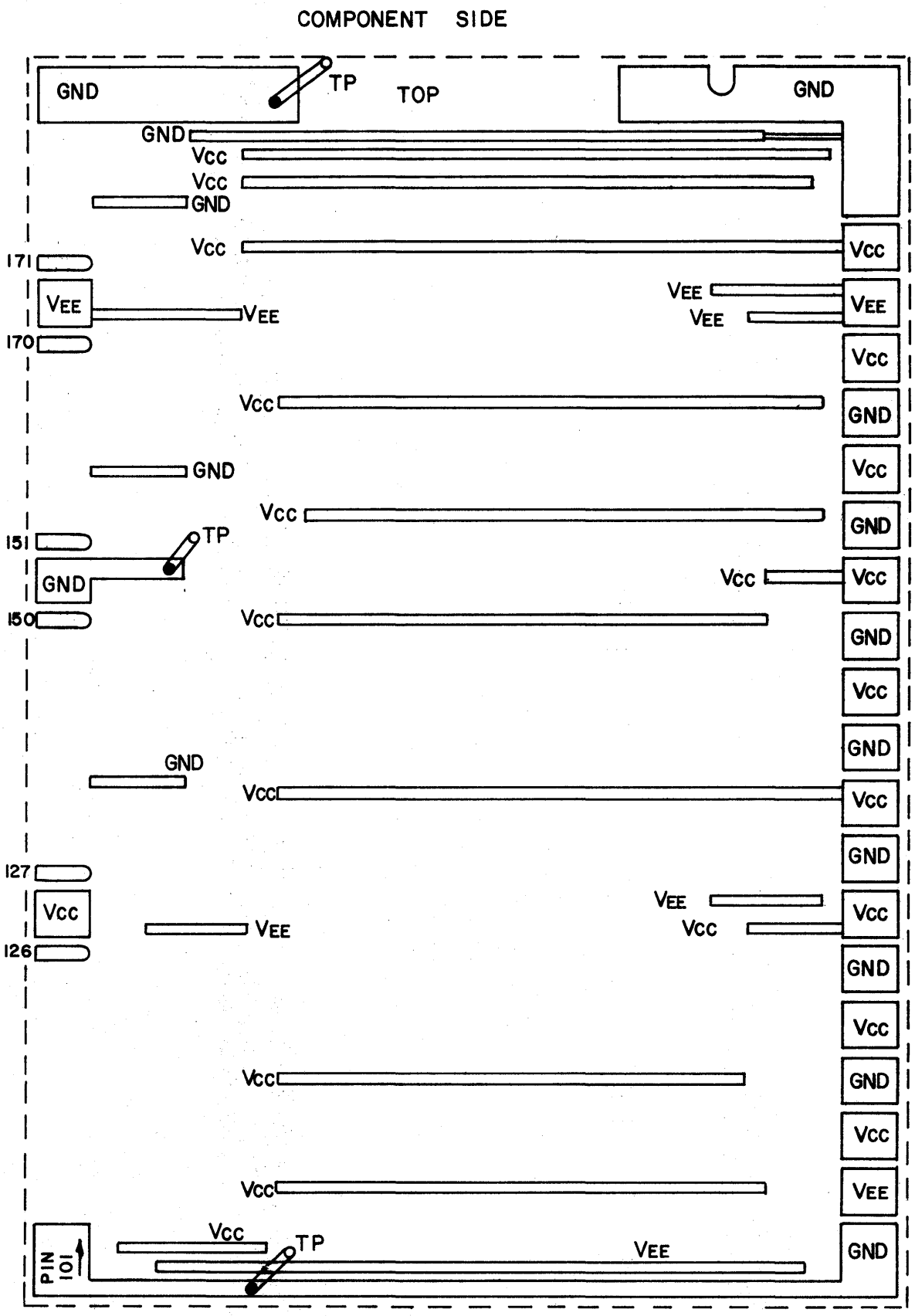


Figure 4-1. Memory Board Power Bus Configuration

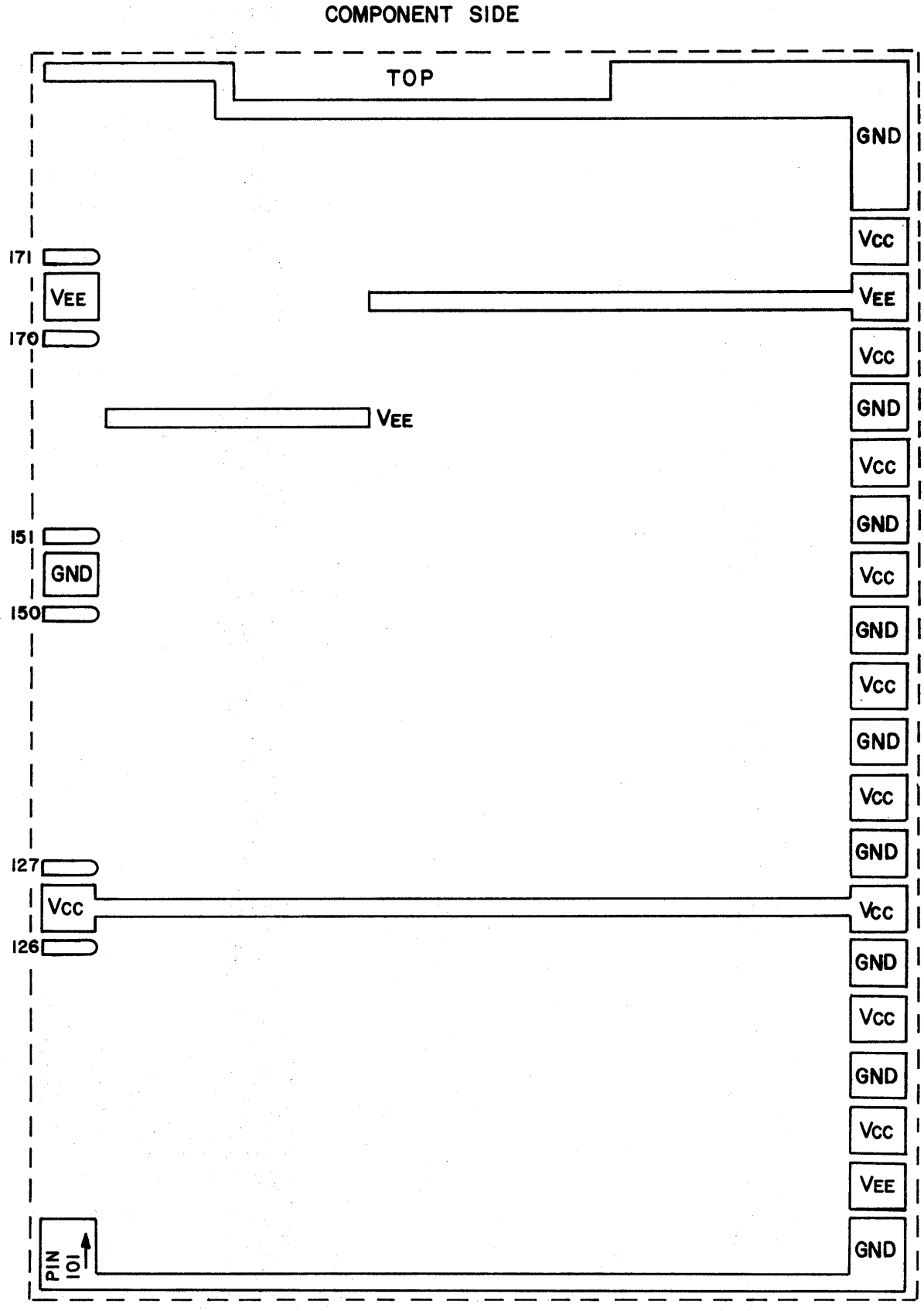


Figure 4-2. Control Board Power Bus Configuration

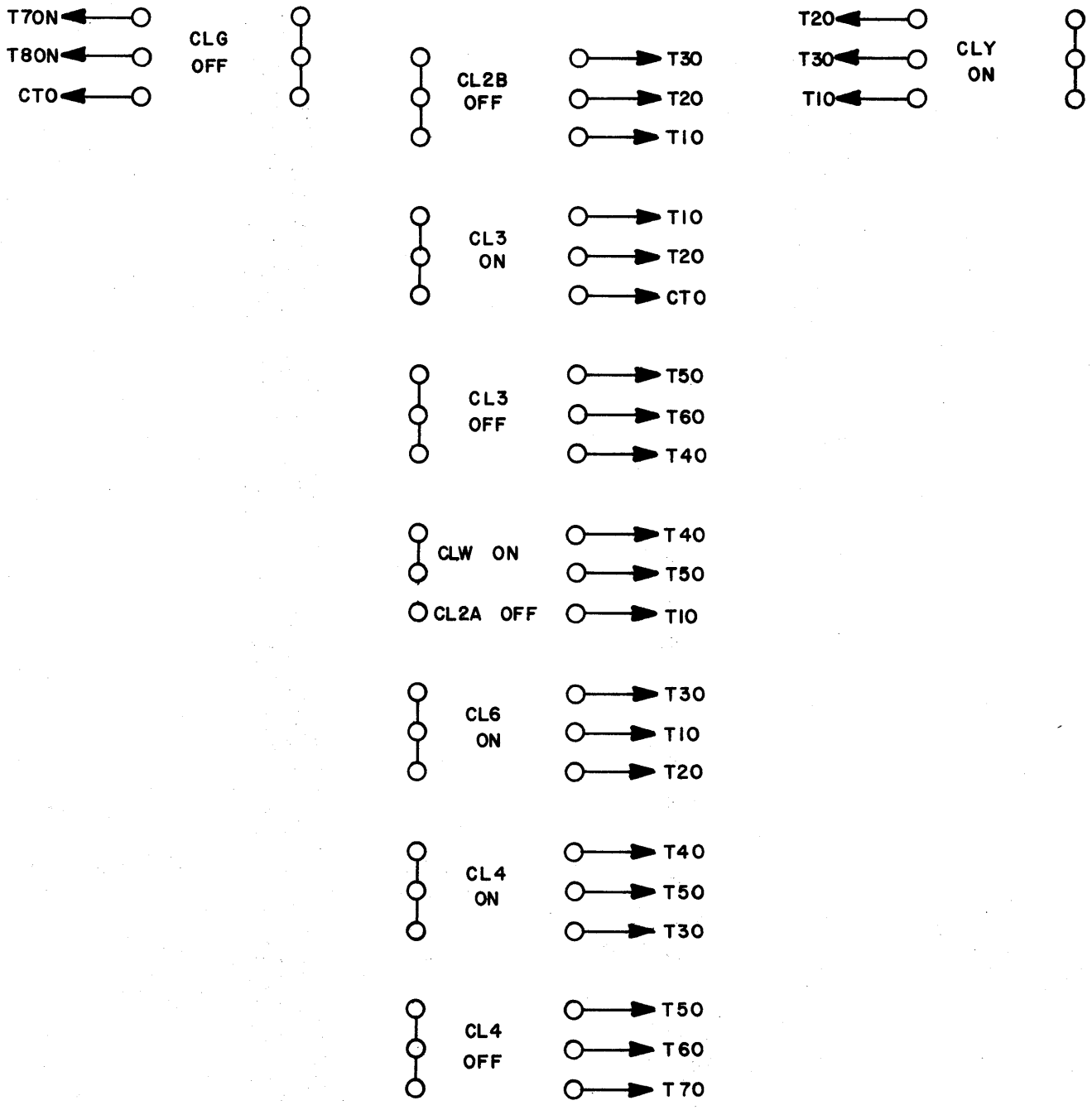


Figure 4-3. Control Board Adjustable Timing Taps

ACKNOWLEDGMENTS

The author wishes to express his thanks to Dr. Wendell Sander, Brooks Cowart, and Theodore Laliotis of Fairchild for their explanatory remarks on the operation of the MμL 4100 device. The cooperation of Robert Davison of Burroughs is greatly appreciated. The author also wishes to express his gratitude to his colleague Morris Knapp at Automation Technology for his sincere encouragement and administrative assistance, which helped to bring this manuscript into reality. Finally, thanks are extended to Mrs. Mildred Pape and Mrs. Jo Ann Kuehl for their invaluable help during the typing of the manuscript.

REFERENCES

[1] Rice, R., Sander, W. B., and Greene, F. S., "Design Considerations Leading to ILLIAC IV LSI PEM," *IEEE Journal of Solid-State Circuits*, October 1970.

[2] Greene, F. S., and Sander, W. B., "Address Selection by Combinatorial Decoding of Semiconductor Memory Arrays," *IEEE Journal of Solid-State Circuits*, Vol. SC-4, No. 5, October 1969.

[3] Greene, F. S., "ILLIAC IV PEM Engineering Design Report," Fairchild Semiconductor, Report No. ER0048, Rev. A.

[4] Economidis, T., "Principles of Operation of the ILLIAC IV Memory Logic Unit," NASA/Ames Research Center, IAC Doc. No. PO-I1200-0000-A, February 15, 1973.