PRELIMINARY!

SOLOMON PROJECT TECHNICAL
MEMORANDUM NO. 29

SOLOMON II Design Reference Manual

29 November 1963

WESTINGHOUSE DEFENSE AND SPACE CENTER
Defense and Space Systems Operations
Baltimore, Maryland

TABLE OF CONTENTS

1. GENERAL DESCRIPTION OF SOLOMON II

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

INTRODUCTION

This technical memorandum is to serve as a design guide for the development of the SOLOMON II computer. This manual is to be the sole reference for the SOLOMON machine designers. If in any case it is found that the contents of this document is incorrect, or possibly invalid, it should be recorded and the SOLOMON Project Engineer should be informed immediately. In the event changes are absolutely necessary such changes will be recorded and the holders of this document will be supplied with the changes as well as periodic additions.

This document is to be internally distributed and is highly company proprietary and therefore should be treated as such.

R. C. McReynolds
SOLOMON Project Engineer

1. GENERAL DESCRIPTION OF SOLOMON II

Figure 1-1 is a general block diagram of the SOLOMON II Parallel Network Computing System. The basic elements of the system are the Processing Element (PE) Network, the PE control unit, the PE Program Memory, and the Input-Output Unit. As an optional feature to SOLOMON II, a General Purpose (GP) computing subsystem can be added in a modular fashion. The PE subsystem and the GP subsystem as shown in figure 1-2, can communicate by means of common memories and an interrupt feature which allows the PE subsystem to automatically interrupt the GP subsystem.

The Input-Output (I/O) Subsystem can be quite flexible. It is a common subsystem to both the PE and the GP subsystems as shown in figures 1-1 and 1-2; however, if a particular application requires, there can be a unique I/O subsystem for each the PE and GP subsystem.

1.1 PROCESSING ELEMENTS NETWORK

1.1.1 General

The heart of the SOLOMON II Computing System consists of many identical interconnected Processing Elements. The Processing Elements are each provided with computational ability to process data simultaneously and are controlled by a single program executed by a common centralized control unit.

The Processing Element Network consists of 256 n PE's, each possessing complete arithmetic capabilities and having its own associated memory. Figure 1-3 is a functional block diagram of a single PE and its associated memory.

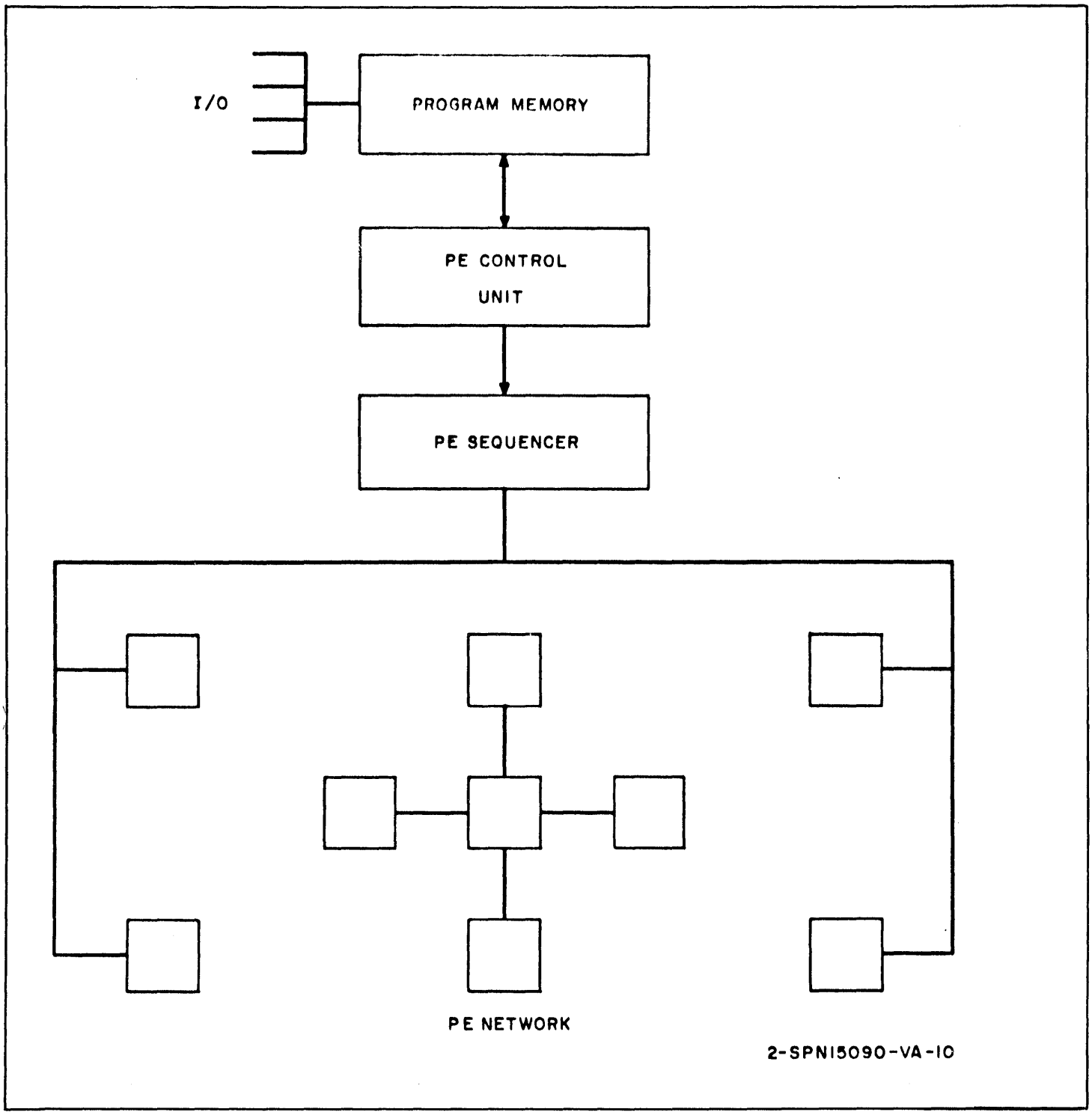


Figure 1-1. General Block Diagram of SOLOMON II

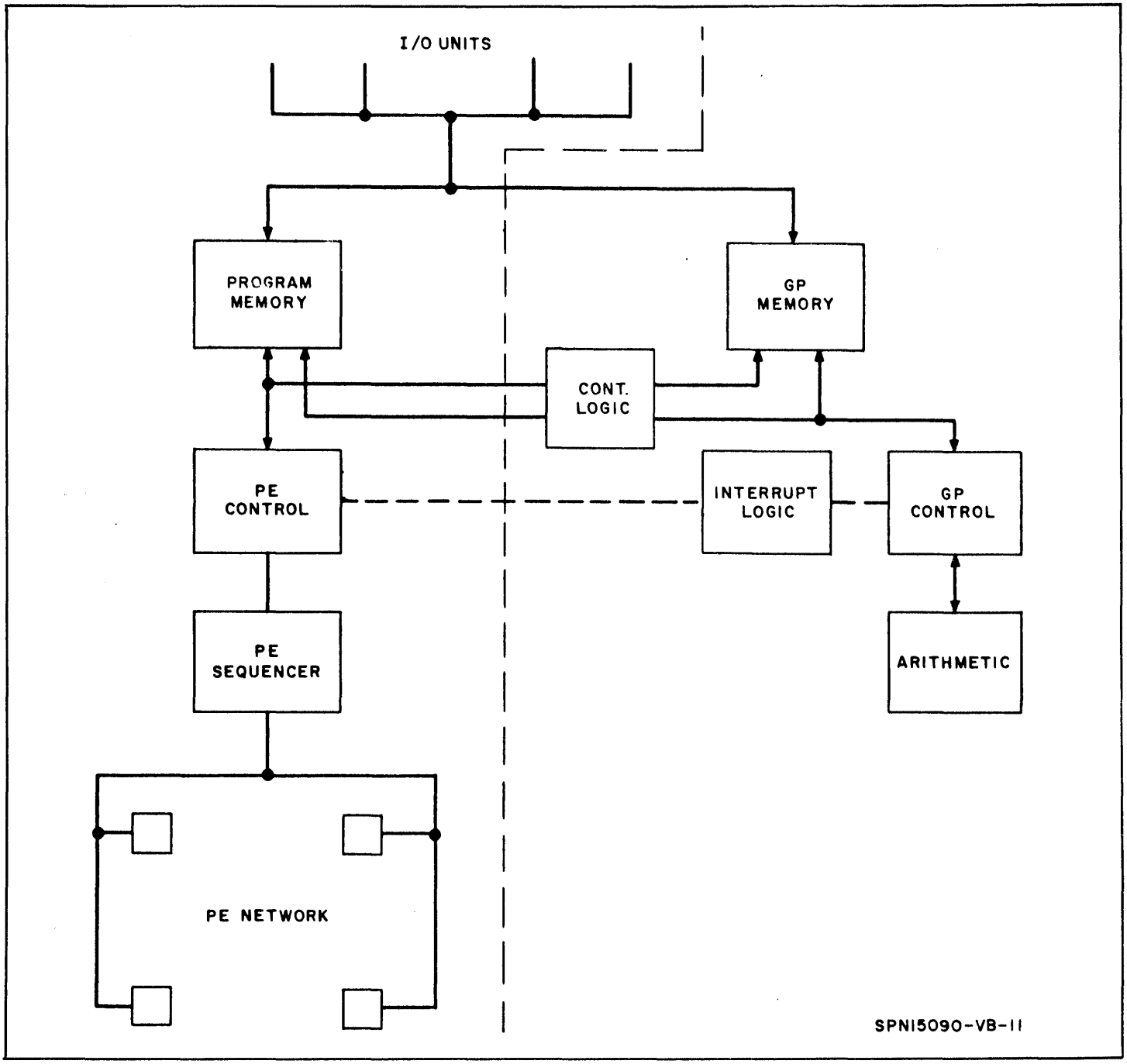


Figure 1-2. Block Diagram of SOLOMON II With Optional GP Subsystem

Each PE has associated with it one memory frame and two addressable registers (designated P and Q). During a given operation, 20-bit words from the PE memory and/or accumulator are serially transferred into the PE arithmetic logic. As the

execution of the operation takes place in the arithmetic logic, the result as it is generated is transferred to the P or Q Registers.

In addition to performing internal operations, each PE has the capability to communicate with its nearest neighbors (designated north, south, east, and west) within the PE Network; that is, a given PE operation can be accomplished between an operand contained in the P Register of any base PE with an operand contained in memory of any of its four nearest neighbors. In addition to the normal routing function between a PE and its neighbors, each PE has a fifth neighbor which is common to all PE's in the network. This fifth neighbor is called the Broadcast Input. This input will allow

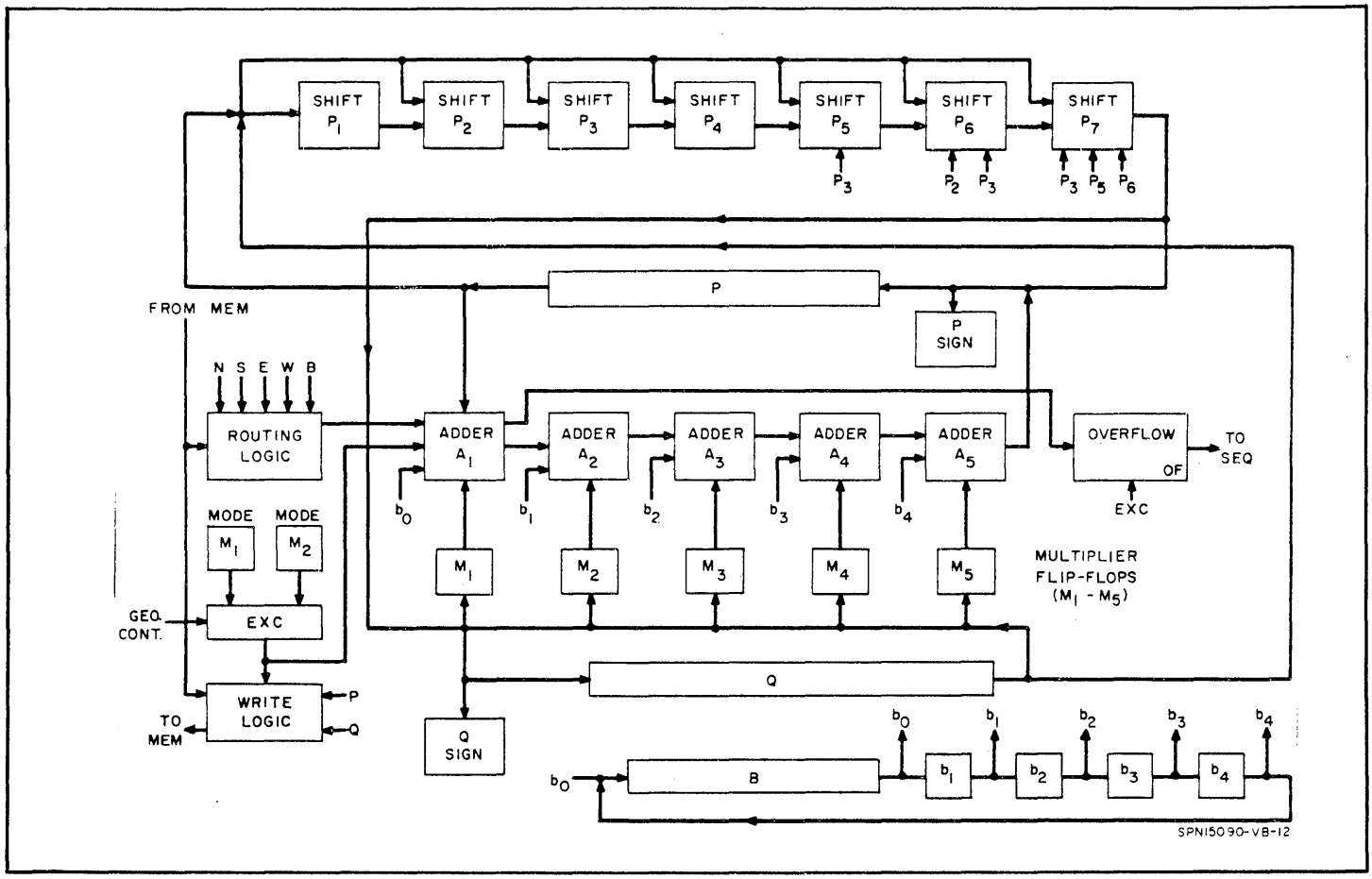


Figure 1-3. Processing Element Block Diagram

an operand which is common to all or a large number of PE's to be accessible to each without being actually stored in each PE. When such an operand is needed in a calculation, the operand can be transferred from the program memory into the Broadcast Register. The contents of this register are then serially fed to all PE's as a normal operand and the calculation is simultaneously performed by all specified PE's. An additional feature available to the programmer through the routing logic is the ability to select an operand from the L-buffer memory. The function performed here is to route up to 32 operands located in the L-buffer memory to up to 32 rows or columns in the network; that is, in the case of routing the L-buffer contents to PE columns, the first word in the L-buffer (20 bits) is routed to all PE's in the first column, the second word in the L-buffer is routed to all PE's in the second column, and the 32d word in the L-buffer is routed to all PE's in the 32d column.

The geometry of the PE Network is illustrated by first considering the network as a 32 x 32 plane. Let N(i, j) represent the PE in the ith row and jth column in the plane (see figure 1-4). By connecting PE N(i, 31) to PE n(i, 0) for all i = 0, ... 31, the network is formed into a cylinder. Subsequently connecting, in a similar manner, each N(0, j) to each N(31, j), a second cylinder is formed. If both sets of connections are specified simultaneously, the result is that the network is formed into a torus. In addition, an option to the programmer is to select the simple planar geometry.

The PE Network is designed to simultaneously execute a common instruction in all PE's. However, control is provided within each PE to give it the capability of not executing a given instruction. The technique used to control the execution of instructions within the individual PE's is called mode control. Basically, each PE can be in

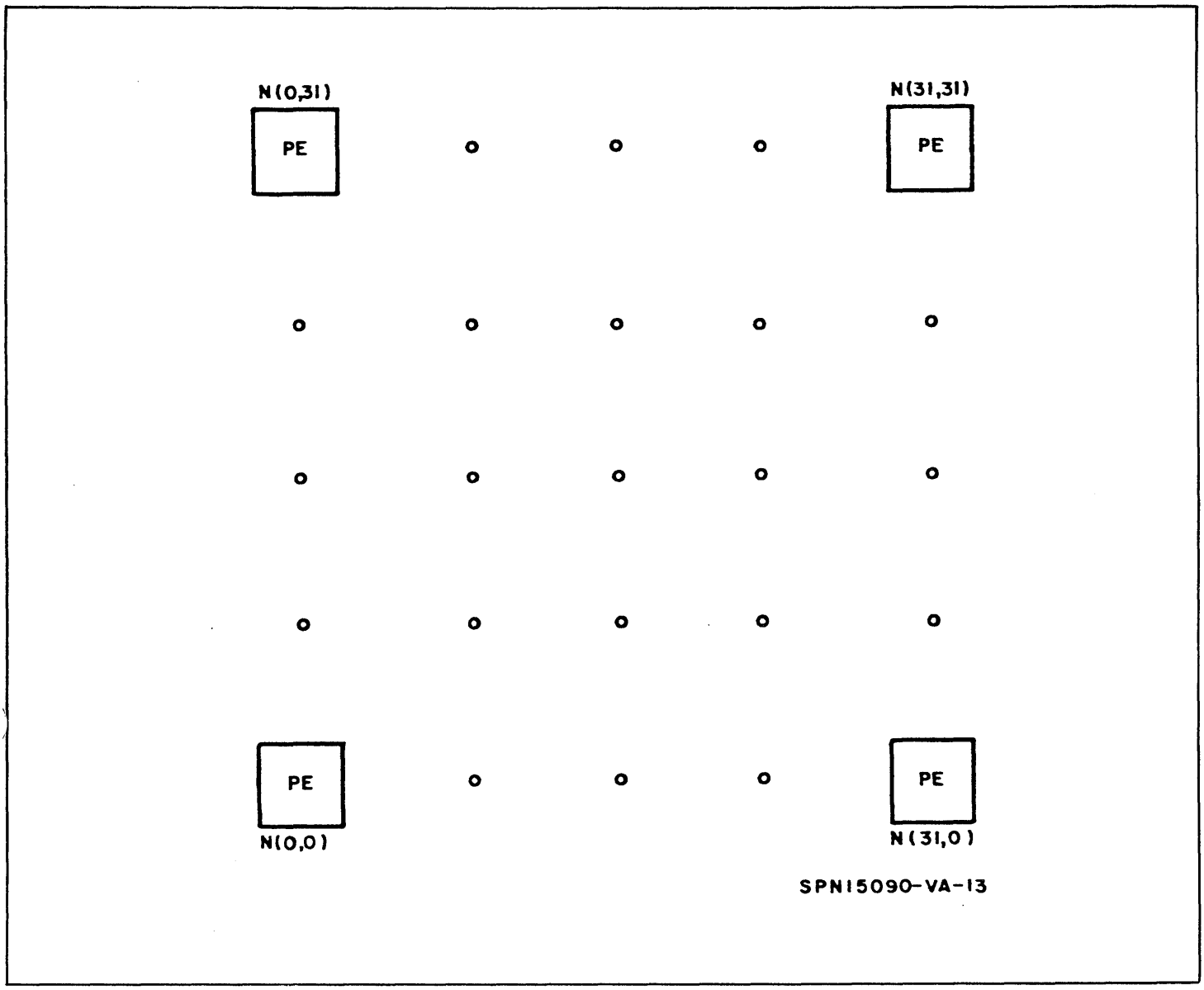


Figure 1-4. Identification of PE's Within the Network

any one of four mode states. Each PE instruction specifies the particular mode states in which PE action is required. Any combination of the four states can be specified. For example, if a given instruction specifies addition for PE's in modes 1 and 3, then those PE's in the modes 1 and 3 perform the addition, those in the modes 0 and 2 do

not. A full complement of mode control instructions is provided which allow for mode sensing, changing mode states, and loading and storing of mode states in the PE data memory. This latter capability enables the programmer to effectively have an unlimited number of mode states available.

In addition to the concept of mode control imposed over the individual PE's, a technique termed geometric control is available. Geometric control is a technique for controlling the PE Network in a purely geometrical manner (see figure 1-5). Two 32-bit registers called the Row Select Register and the Column Select Register are functionally placed along the PE Network where each bit in the register corresponds to a given row or column in the PE Network. The combination of geometric control and mode control is known as execute control. A geometric option register, set before the PE instructions are executed, specifies either, both, or neither of these registers to control the operation of the PE Network. In addition to requiring that the mode state(s) be as specified in the instruction, the specified geometric control registers are monitored and the execution of a given instruction will be permitted or not, on an individual PE basis, based on a rule that both geometric and mode control must be in accord with the conditions specified by the programmer. The Row Select Register and the Column Select Register are addressable by the programmer and hence can be altered in any desired manner.

A necessary requirement for most applications is the capability to obtain a mode map of the PE Network at various points in a program. To facilitate this requirement, an instruction is available which allows the programmer to command each PE in the specified mode to set a corresponding bit in the Geometric Control Registers (destroying the original contents).

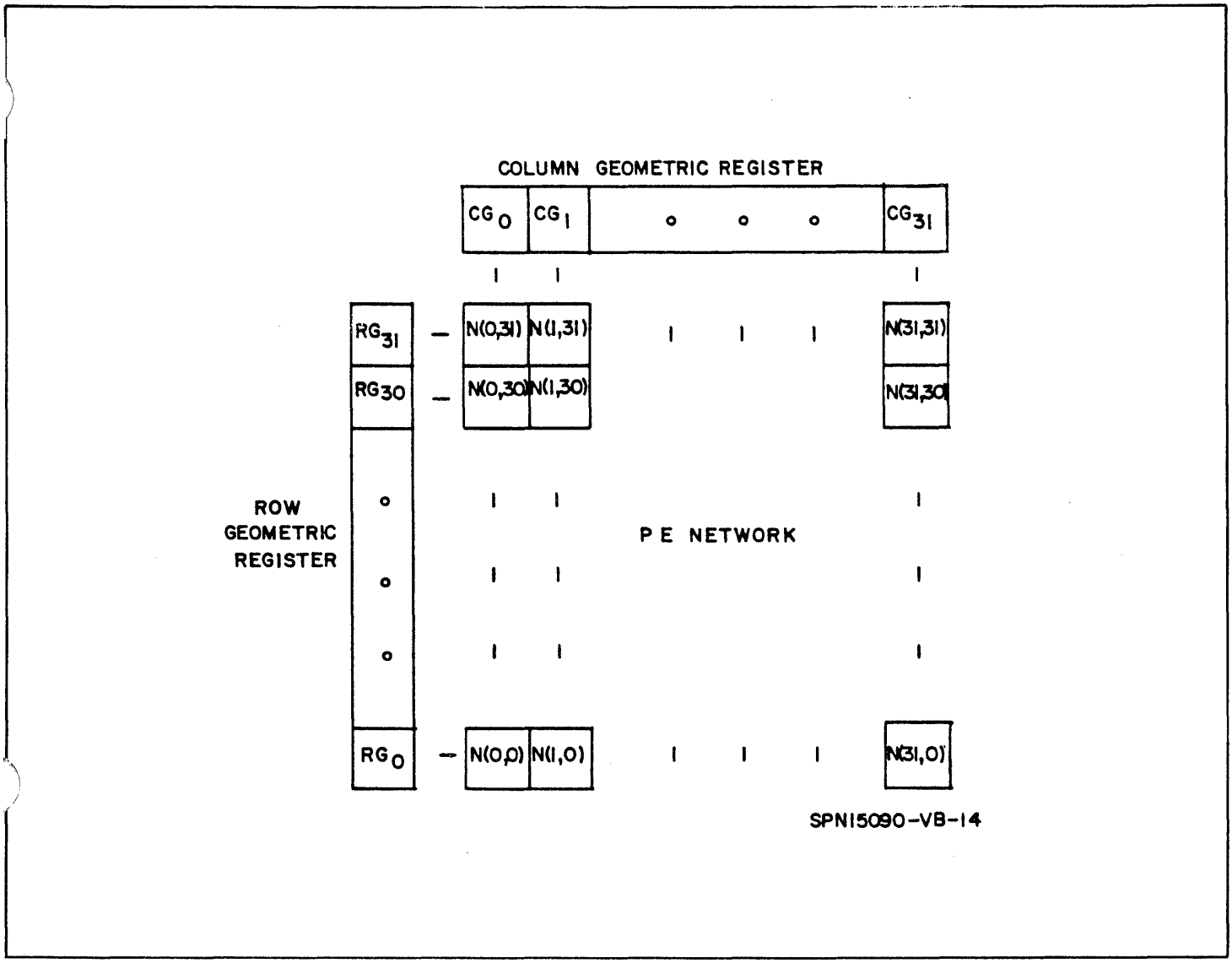


Figure 1-5. Geometric Control Registers

The transfer of data to and from the PE Network is accomplished by a 32- by 20-bit flip-flop memory called the L-buffer memory which communicates between the PE's in a selected row or column of the PE Network. The information from PE memory is gated into the PE logic, where, under execute control, it is directed to the L-buffer. When going from the L-buffer to PE memory, the information also enters the PE at the write logic level where, under execute control, it is directed to memory.

1.1.2 Detail PE Information

The Processing Element Logic contains two accessible registers known as the P and Q Registers. The P Register acts as the accumulator for the processing element instructions and, although accessible, the Q Register may perform only the limited number of functions as specified in the instruction description. The processing element contains one other effective register (known as the ICAND Register) which is not program addressable. This register holds the multiplicand and the divider in the multiply and divide instructions, respectively.

The PE Registers (P, Q, and, ICAND) are implemented with glass delay lines and associated logic. The bit rate of the delay line loops is 10 megacycles. This bit matches the 10-megacycle input available from the 10 cascaded 1-microsecond core memories. The basic word cycle is 2.2 microseconds with 2.0 microseconds for a switching blank.

To provide a thorough understanding of the operation of the PE's several PE instructions are discussed in detail.

All simple arithmetic and logic instructions use the A_1 Adder, shown in figure 1-3, to perform their operations on the selected operands. The result is generally stored in the P Register. The five adders, illustrated in figure 1-3, are used to provide a high speed multiply instruction.

All simple arithmetic and logic instructions use the Al Adder, shown in figure 1-3, to perform their operations on the selected operands. The result is generally stored in the P Register. The five adders, illustrated in figure 1-3, are used to provide a high speed multiply instruction. The multiply instruction is the only instruction which

uses the five adders in the PE. Figure 1-6 shows the sequence of operations which are performed in the adders for the multiply instruction. (The example indicates 10 bit words, however, it contains all the necessary sequences to illustrate the multiply of 20 bit words). The following paragraphs discuss the operation of the PE in multiply and shift instructions as examples of use of its internal logical organization.

1.1.3 The PE Multiply Operation

Functionally, 5 multiplier bits are read into the five multiplier flip-flops (M_1 to M_5) from the R-Register in the 5 bit times required to obtain bits from the delay line. The multiplier flip-flops (M_1 - M_5) control their respective adders (A_1 - A_5). A "one" causes the adder to add the multiplicand, and a zero causes a zero to be added to the partial product. The remaining mutliplier bits are read back into the R Register. In the meantime, the memory has been addressed and the multiplicand is routed to the PE's where it is both added and stored in the B delay line loop.

The multiply scheme which is used may be best understood by referring to figure 1-6. The top figures represent separate bit times in a multiply of 2-10 bit numbers. In the spaces below are the operations performed for each of the five adders at the bit time specified. Note that at bit time 1 zero is added to the first multiplicand bit (b_1) in adder 1, and the other adders perform no operation (N.O.). At bit time 2, zero is added to the second multiplicand bit (b_2) in adder 1 while zero is added to the sum output of adder 1 (S_1) in adder 2 and the other adders perform no operation. At the third bit time 0 is added to b_3, the first multiplicand bit b_1 is added to the output of adder 1 (S_1) in the second adder, and adder 3 adds zero to the sum output of adder 2 (S_2). The other 2 adders do nothing. Figure 1-6 indicates the complete

BIT TIMES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Adder 1	$0 + b_1$	$0 + b_2$	$0 + b_3$	$0 + b_4$	$0 + b_5$	$0 + b_6$	$0 + b_7$	$0 + b_8$	$0 + b_9$	$0 + b_{10}$	$0 + b_{10}$	$0 + b_{10}$	$0 + b_{10}$	$0 + b_{10}$	N.O.	N.O.	N.O.	$P_{6_A} + b_1$	$P_{7_A} + b_2$	$P_{8_A} + b_3$	$P_{9_A} + b_4$	$P_{10_A} + b_5$	$P_{11_A} + b_6$	$P_{12_A} + b_7$	$P_{13_A} + b_8$	$P_{14_A} + b_9$	$P_{14_A} + b_{10}$	$P_{14_A} + b_{10}$	$P_{14_A} + b_{10}$	$P_{14_A} + b_{10}$	$P_{14_A} + b_{10}$	N.O.	N.O.	N.O.	N.O.
Adder 2	N.O.	$S_1 + 0$	$S_1 + b_1$	$S_1 + b_2$	$S_1 + b_3$	$S_1 + b_4$	$S_1 + b_5$	$S_1 + b_6$	$S_1 + b_7$	$S_1 + b_8$	$S_1 + b_9$	$S_1 + b_{10}$	$S_1 + b_{10}$	$S_1 + b_{10}$	$S_1 + b_{10}$	N.O.	N.O.	N.O.	$S_1 + 0$	$S_1 + b_1$	$S_1 + b_2$	$S_1 + b_3$	$S_1 + b_4$	$S_1 + b_5$	$S_1 + b_6$	$S_1 + b_7$	$S_1 + b_8$	$S_1 + b_9$	$S_1 + b_{10}$	$S_1 + b_{10}$	$S_1 + b_{10}$	N.O.	N.O.	N.O.
Adder 3	N.O.	N.O.	$S_2 + 0$	$S_2 + 0$	$S_2 + b_1$	$S_2 + b_2$	$S_2 + b_3$	$S_2 + b_4$	$S_2 + b_5$	$S_2 + b_6$	$S_2 + b_7$	$S_2 + b_8$	$S_2 + b_9$	$S_2 + b_{10}$	$S_2 + b_{10}$	$S_2 + b_{10}$	N.O.	N.O.	N.O.	$S_2 + 0$	$S_2 + 0$	$S_2 + b_1$	$S_2 + b_2$	$S_2 + b_3$	$S_2 + b_4$	$S_2 + b_5$	$S_2 + b_6$	$S_2 + b_7$	$S_2 + b_8$	$S_2 + b_9$	$S_2 + b_{10}$	$S_2 + b_{10}$	$S_2 + b_{10}$	N.O.	N.O.
Adder 4	N.O.	N.O.	N.O.	$S_3 + 0$	$S_3 + 0$	$S_3 + 0$	$S_3 + b_1$	$S_3 + b_2$	$S_3 + b_3$	$S_3 + b_4$	$S_3 + b_5$	$S_3 + b_6$	$S_3 + b_7$	$S_3 + b_8$	$S_3 + b_9$	$S_3 + b_{10}$	$S_3 + b_{10}$	N.O.	N.O.	N.O	$S_3 + 0$	$S_3 + 0$	$S_3 + 0$	$S_3 + b_1$	$S_3 + b_2$	$S_3 + b_3$	$S_3 + b_4$	$S_3 + b_5$	$S_3 + b_6$	$S_3 + b_7$	$S_3 + b_8$	$S_3 + b_9$	$S_3 + b_{10}$	$S_3 + b_{10}$	N.O.
Adder 5	N.O.	N.O.	N.O.	N.O.	$S_4 + 0$	$S_4 + 0$	$S_4 + 0$	$S_4 + 0$	$S_4 + b_1$	$S_4 + b_2$	$S_4 + b_3$	$S_4 + b_4$	$S_4 + b_5$	$S_4 + b_6$	$S_4 + b_7$	$S_4 + b_8$	$S_4 + b_9$	$S_4 + b_{10}$	N.O.	N.O.	N.O.	$S_4 + 0$	$S_4 + 0$	$S_4 + 0$	$S_4 + 0$	$S_4 - b_1$	$S_4 - b_2$	$S_4 - b_3$	$S_4 - b_4$	$S_4 - b_5$	$S_4 - b_6$	$S_4 - b_7$	$S_4 - b_8$	$S_4' - b_9$	$S_4 - b_{10}$
	-	-	-	-	P_1	P_2	P_3	P_4	P_5	P_{6_A}	P_{7_A}	P_{8_A}	P_{9_A}	P_{10_A}	P_{11_A}	P_{12_A}	P_{13_A}	P_{14_A}	-	-	-	P_{6_B}	P_{7_B}	P_{8_B}	P_{9_B}	P_{10_B}	P_{11_B}	P_{12_B}	P_{13_B}	P_{14_B}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}

Figure 1-6. PE Multiple Example *

*b_n - Multiplicand bit n

S_n - Output of adder n

N.O. - No Operation

P - Partial product bit n

repetitive pattern for the multiply operation for 10 bit words. It is important to note that the sign bit must be repeated at certain times as is shown for adder one in bit times 10 to 14 and that the last bit set in the multiplier flip flops is the sign bit. This bit is used for the correction cycle of the multiply and produces a subtraction of the multiplicand from the partial product. This is shown in adder 5 bit times 26 to 35.

The first 5 partial product bits formed in each 5 bit pass replace the five bits of the multiplier which are used in that pass [for example (P_1 to P_5) in bit times 5 to 10]. At the end of the operation the 19 least significant bits are located in R and the most significant in A. There are 4 cycles during which the 5 multiplier bits are recognized and thus partial product bits are formed. The total execution time for this multiply implementation is 11.2 microseconds per P.E.

Figure 1-7 represents the actual multiply operations which are performed by the 5 adders. Here the columns represent bit order and are shown as a person would manually perform the additions.

1.2 PE SHIFT OPERATIONS

All of the PE shift instructions are performed using various combinations of delay elements whose outputs are labeled P_n on the logic diagram and block diagrams. The number of delay elements and therefore the total delay is determined by the shift length. Execution times vary from 2.2 microseconds for a single length left shift to 6.6 microseconds for a double length right shift. Because of their uniqueness each shift instruction is discussed subsequently.

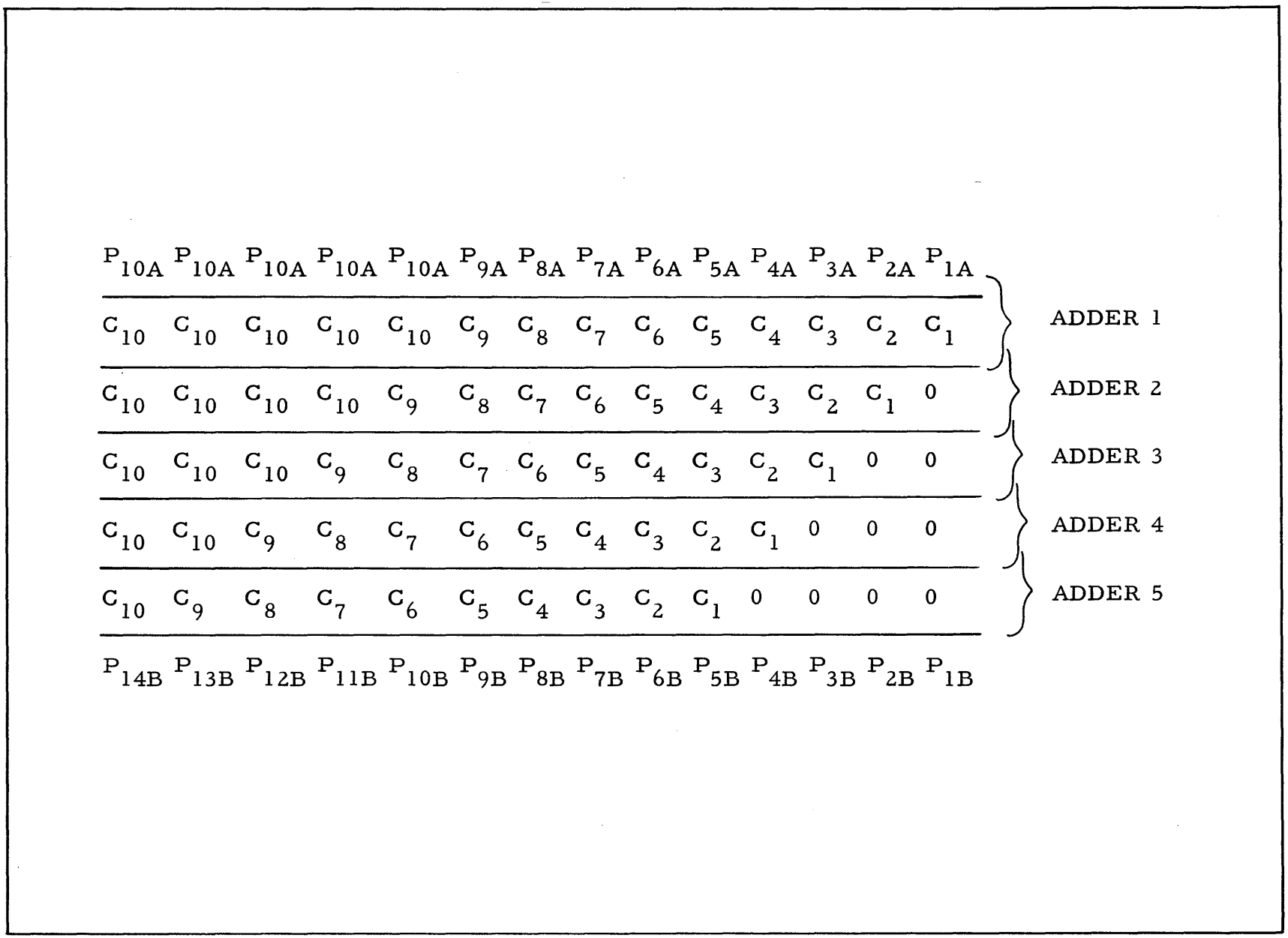


Figure 1-7. 5 Multiplier Bits 10 Bit Word Multiply Example

1.2.1 Shift Left-Double Length

The total time for this instruction is 4.4 microseconds, independent of the magnitude of the shift length. Initially, the A register must contain the most significant bits and the R Register the least significant bits.

Bits 1 through 20-(SL + 1) (Sl = shift length) are gated into a total delay of SL bit-times. At the time of 20-SL, an additional 1-bit delay is added to the original delay in order to properly place the bits in the new most significant word.

1.2.2 Shift Right-Double Length

The instruction begins by inhibiting the first SL bits of the R register. The next 20-(Sl + 1) bits are inserted into a delay of 20-SL bit times which includes the 1 bit delay used in the double length left shift. By gating the 1 bit delay in and out at the appropriate times the shift is completed. During the last SL bit times the sign may be propagated or not, depending upon the programmer's option. The execution time is 6.6 microseconds.

1.2.3 Shift Left-Single Length

The instruction is performed by gating the nineteen magnitude bits into a delay of SL bit times. The last SL bits excluding the sign bit are compared with the sign bit if an algebraic shift is being performed. The execution time for this instruction is 2.2 microseconds.

1.2.4 Shift Right-Single Length

The first SL bits of the specified register are inhibited and the following 20-SL bits are inserted into a delay of 20-SL bit times. The 20-SL delay includes the 1 bit delay mentioned earlier. At 2.3 microseconds the 1 bit delay is gated out making the total

delay equal to 20 - (SL + 1). The sign is propagated during the last SL bit times if specified by the programmer. The execution time for this instruction is 4.4 microseconds.

1.2.5 Shift Left-Circular

This instruction is restricted to the R register. The execution time is 4.4 microseconds.

SL bits as specified by the programmer are shifted through a delay of zero (0) bits. The resulting twenty bits are then gated through a delay of 20-SL bit times. In order to end on an integral multiple of 2.2 microseconds, the word is gated through a delay of 2.0 microseconds.

1.3 PE MEMORIES

Each of the 1024 processing elements in the SOLOMON II System require access to 40,960 serial memory bits at a 10-megacycle repetition rate. Since 1024 separate serial memories would be impractical to build, the memory is actually designed as several conventional word-select units.

The basic memory module is a ferrite core, 4096-word, 256-bits-per-word unit with a read-write cycle time of 1.0 microsecond. This speed is obtained by using a linear select, partial switching mode of operation utilizing 30- x 18-mil cores. By driving 4 of the basic modules at one time, 1024 simultaneous outputs are obtained, 1 output going to each PE. The 10-megacycle repetition rate is obtained by staggering the outputs of 10 1-megacycle memories. Thus the complete PE memory is made up of 40 basic memory modules: groups of 4 in series to obtain 1024 simultaneous outputs by 10 in parallel to obtain a 10-megacycle repetition rate.

Extremely close timing is not required of the individual memory units, see Figure 1-8. Each of the ten units in parallel has its output taken to a separate holding register. The data is strobed out of each register in sequence by a one megacycle clock pulse, each clock pulse displaced in time by 100 nanoseconds from the previous one, onto a common data line for transmitting the 10 megacycle data flow to the PE. With this type of timing control the ten parallel memories do not have to remain exactly in phase with their phase of a ten-phase clock.

The timing control for the data flow from the PE to one of the ten parallel memories for writing into the memories is similar to that described above. Again, very stringent timing requirements are not imposed on the individual memory units.

1.4 BASIC MEMORY MODULE

A block diagram of one of the basic modules of the PE memory is shown in figure 1-9.

A 12-flip-flop address register holds the information for selecting 1 of 4096 words. A 64 by 64 selection matrix is used so 6 of the flip-flops select 1 of the 64 drivers while the other 6 flip-flops select 1 of 64 switches. The simultaneous operation of one switch and one driver permits selection of a word.

The load sharing matrix (LSM) transformer is utilized as the driving element for obtaining the high power outputs required of the full read and partial write pulses. A 12-input, 8-output LSM is used with each output multiplexed in 8 places through the use of center-tapped transformers and saturated switches. Twelve constant current drivers are used to energize the LSM transformers.

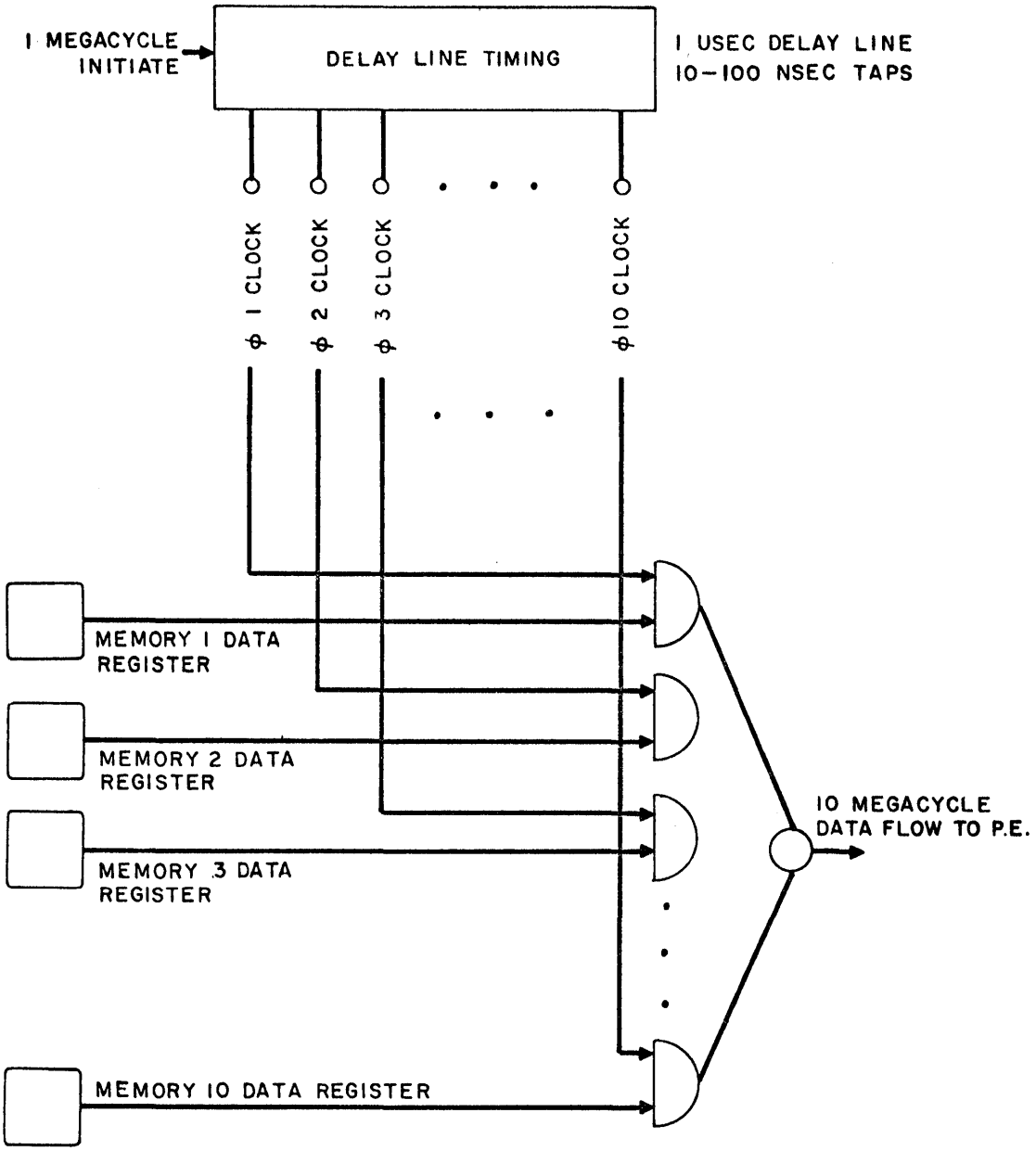


Figure 1-8. Timing - Memory Output to PE

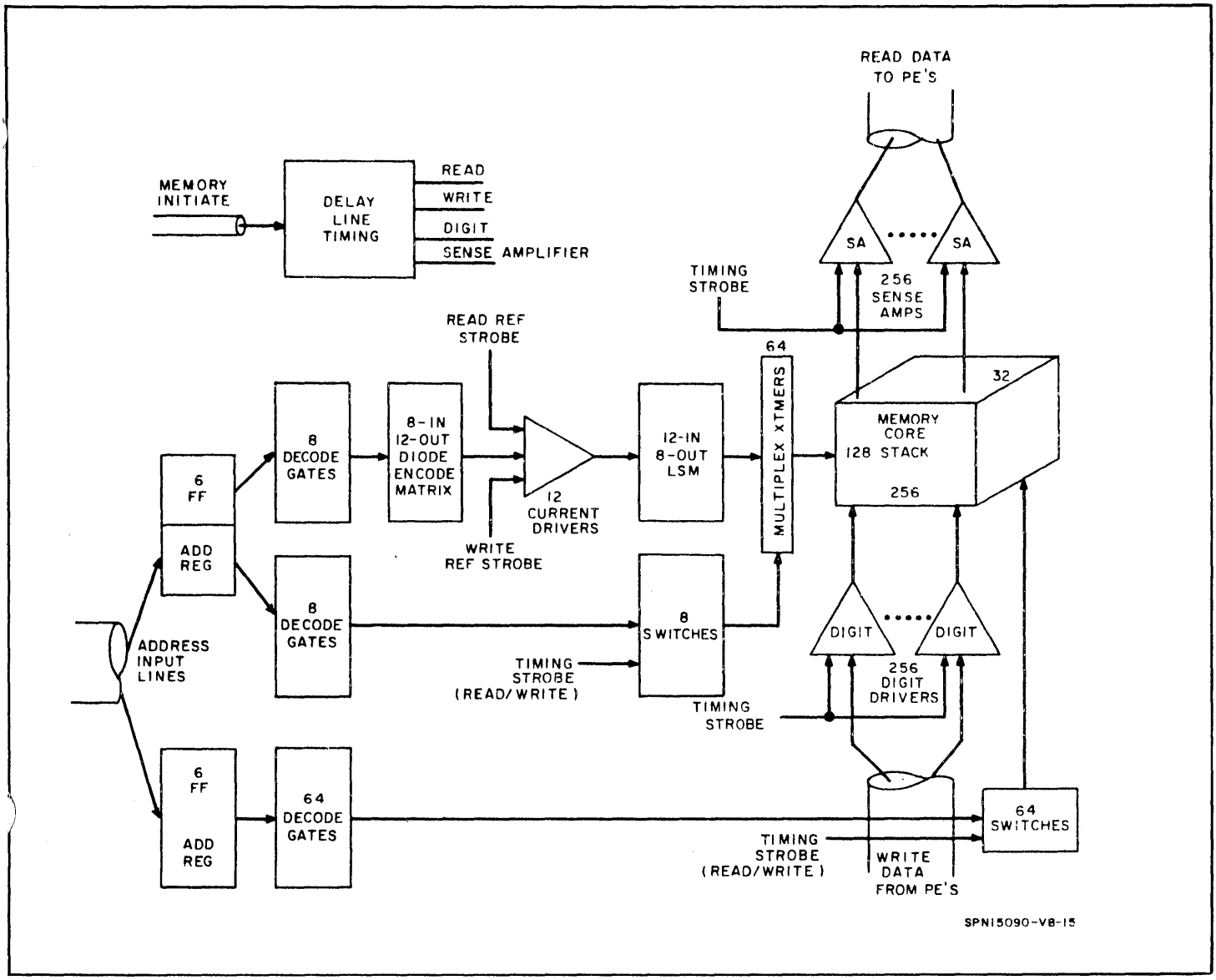


Figure 1-9. Block Diagram SOLOMON PE Memory

The word address is actually divided into three sections and decoded through the use of stroke gates. The output of 1 3-flip-flop section is then encoded by a diode matrix and used to turn on the current drivers which energize the LSM transformers. The output of another 3-flip-flop section is used to control the saturated switches for multiplexing the LSM outputs. The output of the 6-flip-flop section directly controls the switches along 1 axis of the word selection matrix.

A full read pulse switches a large amount of flux in all cores which represent ones and a very small amount in all cores which represent zeros. The core outputs are then amplified in a sense amplifier and strobed during the time a readout is expected to prevent write noise from appearing as signal in the final output which is transmitted to the PE.

Coincidence of a digit pulse with the partial write pulse permits writing a one into all bits of the word in which the digit pulses are present.

1.5 THE NETWORK SEQUENCER

The PE Sequencer (see figure 1-10) provides the PE's with the control signals to execute the PE instruction. When the Network Control Unit (NCU) sends an instruction for the PE Network it is held in the PE Sequencer Instruction Register. At the same time a start signal is issued to the Sequencer logic. This signal starts the various counters which control the PE memory and PE logic signals. The instruction control counter keeps track of where in time the command is and when decoded with the instruction determines what signals should be sent to the PE's. The clock and a pulse shaper then produce the proper timing for these signals and a set of 32 drivers/signal are then required to drive the nearly 140 control signals to the PE's. The 10-mc clock must also be driven to the PE's for control of the more than 40 end-set flip-flops in the PE.

The Sequencer also contains address control for the 10 memory banks used for the processing elements. Each memory bank has an address register associated with it. These are controlled from two counters, one controlling the memory bank and one

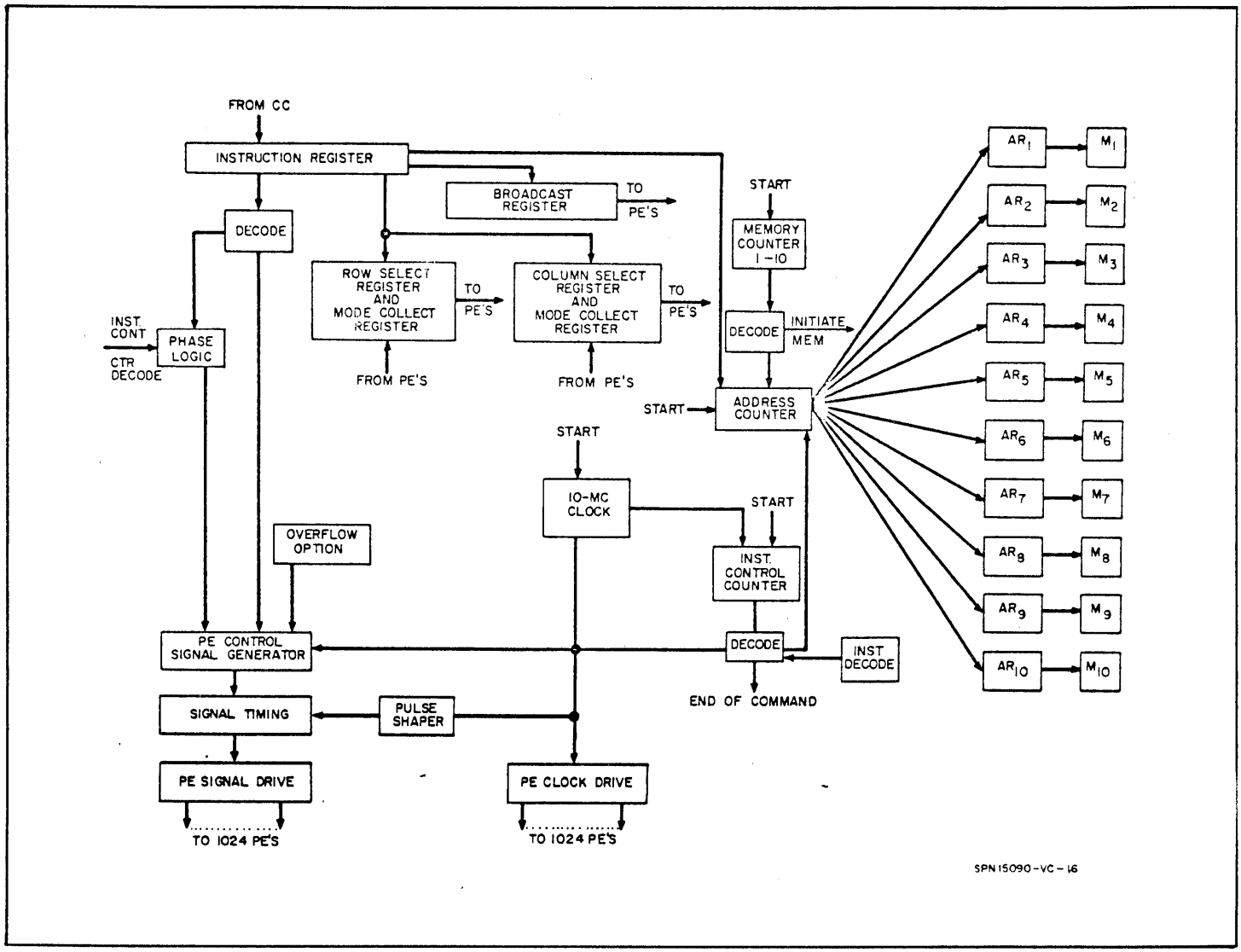


Figure 1-10. Network Sequencer

controlling the actual address of each memory bank. These are tied together in such a manner as to provide the correct address for the PE's.

In the case when the PE Network is busy the NCU Control waits for an end-of-command signal from the Sequencer. This signal is timed such that it occurs early enough to allow partial decoding of the instruction. Once the PE Network has finished an instruction, only 200 nanoseconds (the switching blank) are available before the next command. If this predecoding did not exist, the command setup time would be greater than the 200 nanoseconds switching blank and thus a delay of an additional 2.2

microseconds would be incurred before the PE bits would be in the proper position for command execution.

Since the Instruction Register is double banked, it may be used not only for the instruction execution but simultaneously for input of information to the Broadcast Register and Geometric Control Registers. These registers are also capable of double storage to permit simultaneous loading and use of information.

1.6 CONTROL UNITS

Within the computing system there are two control units, the control unit for the General Purpose Unit (GPU) and a control unit for the Network Processor, the Network Control Unit (NCU). The function of each of the units is to decode and route operands and instructions to other units within their respective systems. Note that the entire GP subsystem is optional to the customer, hence the GPU need not be in the system.

Each control unit has associated with it a 40-bit 32 K word memory with the capability of expanding to 64 K words. Although each memory is independent and is associated with one control unit, it is possible, through priority logic, for a control unit to access either memory bank.

The organization of each of the control units within their system allows for the units to obtain an instruction, decode, index if necessary, route the IR data, or obtain and route memory operands, and immediately obtain a new instruction without having to wait for the addressed unit to finish its instruction. This feature allows the GPU to deliver an instruction along with an operand to the Arithmetic Unit and while the Arithmetic Unit is performing the arithmetic operation the control unit can initiate a new instruction such as a control or I/O instruction.

The NCU in the same manner may initiate the PE Sequencer, then perform other control instructions, returning to the Sequencer as a new PE Network instruction is decoded.

If either control unit should address a unit which is busy, it decodes, indexes, and fetches operands as the instruction demands and stores these in its own registers until the busy unit is free. It then delivers the data to the unit and proceeds as usual.

Although the control units are identical in organization and operation, the NCU is more flexible than the GPU due to its unique requirements. The NCU has several instructions which are not implemented in the GPU.

1.6.1 Network Control Unit (NCU)

The NCU has the main function of decoding and routing operations and operand addresses to the PE Sequencer unit. The NCU also has the ability to initiate input-output commands as well as the ability to modify its instructions and operate on data for Sequencer operations.

The NCU is organized as shown in figure 1-11. The Instruction Register (IR) is 40 bits and doubles as an Operand Register for data manipulation. The Instruction Save Register is 8 bits in length and holds parameter constants when the IR is used as an Operand Register. The operation or command decoding is accomplished through flip-flops. These flip-flops act as the IR Save Register and facilitate the unit busy logic.

The control unit contains seven Index Registers addressable through the instruction word. These registers allow modification of the instruction word's M-field, by use of the 20-bit, high speed, parallel adder.

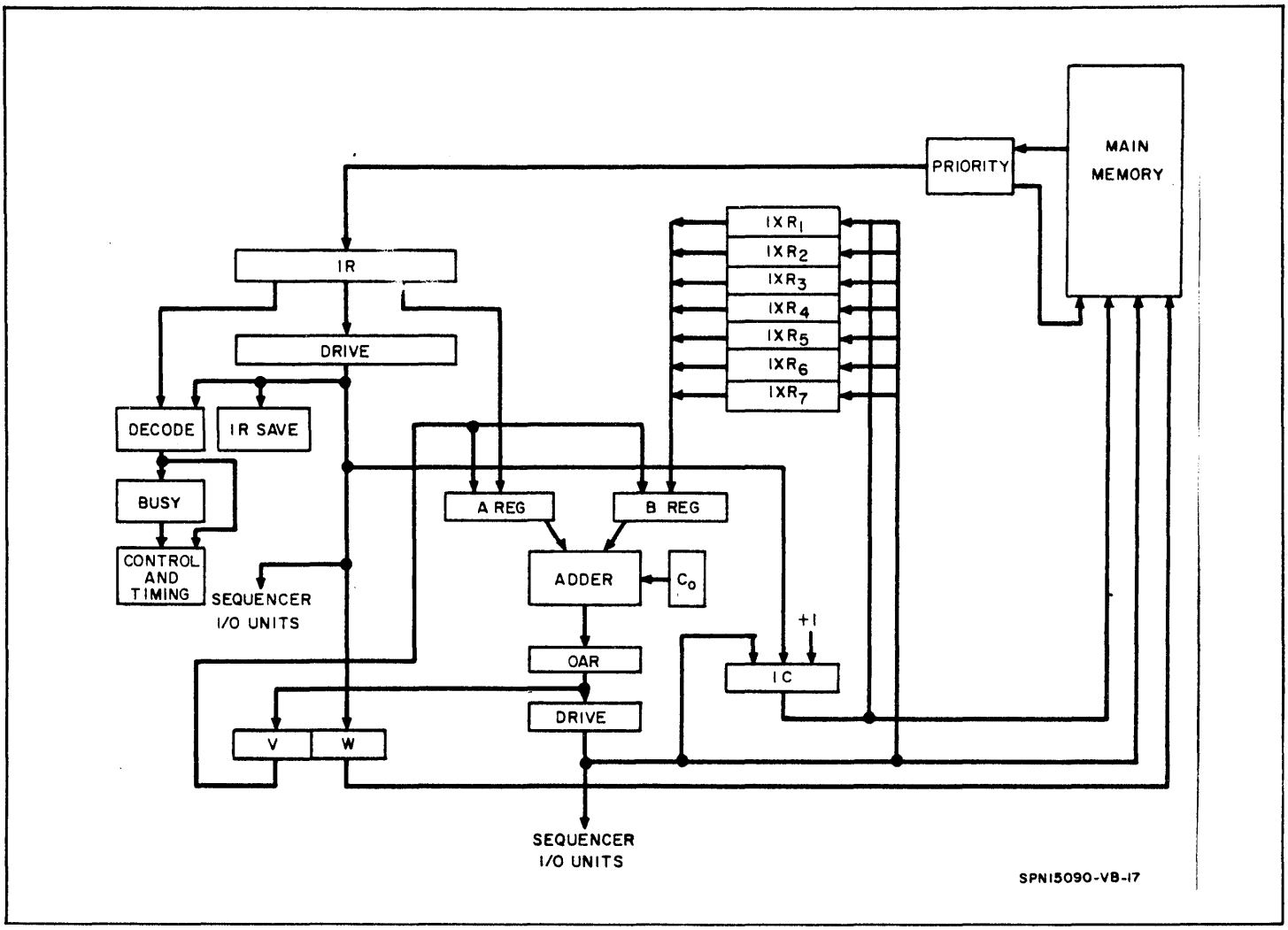


Figure 1-11. Network Control Unit

The index adder adds the contents of the B gates, which selects the desired index value, and the A Register, which may contain the IR M-field or the contents of the accumulator. The add result may be placed in the Operand Address Register (OAR) which is used both as an Operand Address Register and as a Buffer Register for writing into the Index Register. The add result may also be placed into the 20 least significant bits of the W Register. The low order 20 bits of the W Register also are used as the accumulator, V, for the NCU.

The W Register is 40 bits in length and provides the ability to perform boolean and shift operations on any memory location.

The instruction counter is 16 bits and has the ability to be set in parallel from the OAR or M-field of the IR, in the case of jump instructions or to be incremented by one for normal instruction sequencing.

1.6.2 General Purpose Control Unit (GPU)

The organization of the GPU is shown in figure 1-12. This unit initiates the arithmetic unit, up to 32 I/O channels, and has the ability to perform control instructions internally in its own logic.

The unit contains an Instruction Register (IR) which is 40 bits in length. The IR is also used for holding operands which are to go to the arithmetic unit. The IR acts as a buffer to allow the control unit to make arithmetic operand fetches although the arithmetic unit may be busy, thus in many cases increasing the speed of arithmetic operations by saving operand fetch times.

All instruction decoding is accomplished with flip-flops which, acting together with 16-bit Instruction Save Register, preserve instruction information when the IR acts as temporary operand storage.

The GPU has in its organization a high speed parallel adder which is used for indexing address instruction fields and in the execution of control instructions. The adder inputs originate from the 20-bit A and B gates. The A register may contain the IR address field or its complement, while the B gates may contain information from one of the 7 16-bit index registers or the 20-bit V register. The result of all index adder operations are placed in the 20-bit Operand Address Register (OAR).

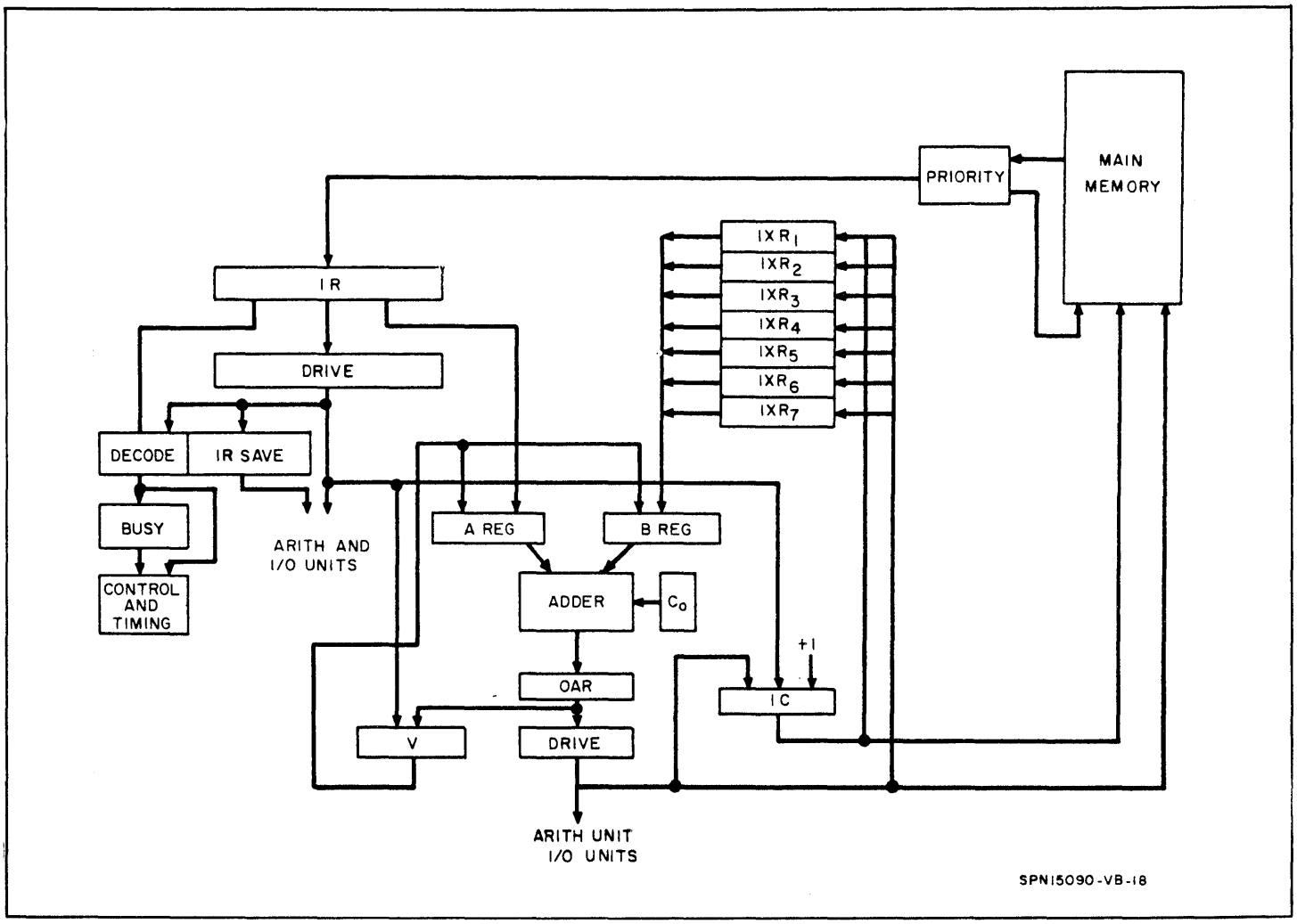


Figure 1-12. General Purpose Control Unit

This register is used for addressing all operand memory fetches. It serves also as a buffer register when data is written into the index or V register with control instructions.

The instruction counter is used for sequencing the instruction addresses and may be set, under jump conditions, in parallel from the OAR or M field of the IR.

1.7 ARITHMETIC UNIT

1.7.1 General Description

The Arithmetic Unit is the calculation unit of the GP subsystem. It is a high speed 40-bit sequential system with the normal General Purpose type of operations.

The Arithmetic Unit, figures 1-13 and 1-14 contain three 42-bit shifting registers, A, R, and I. Each register has an upper and lower bank, designated U and L, respectively. Shifting is accomplished by transferring data between the upper and lower banks of the registers, using the shift paths shown in figure 1-11. All results of arithmetic operations are put in the accumulator register A and the remainder register R. The contents of the I register are used as the multiplicand for multiply operations and the divisor in divide operations.

For floating point operations, the adder, and registers are separated between the 9th and 10th bit from the right. The nine right-most bits are used for exponent arithmetic. The remaining 33 bits are used for the fraction and sign.

1.7.2 The Adder

The adder has a level of Boolean function logic, three levels of simultaneous carry logic, and a level of sum logic. It performs a 42-bit parallel addition in 275 nanoseconds.

1.7.3 Arithmetic Instruction

Operations in the Arithmetic Unit are initiated by the GPU by sending an operand and an operation field. The operation field includes the operation code, parameter bits, shift field, and Boolean operation field. When the operation field and operand, if required, are received from the GPU, the arithmetic unit begins to execute the command and sends a busy signal to the GPU. This signal is maintained until the arithmetic operation is completed.

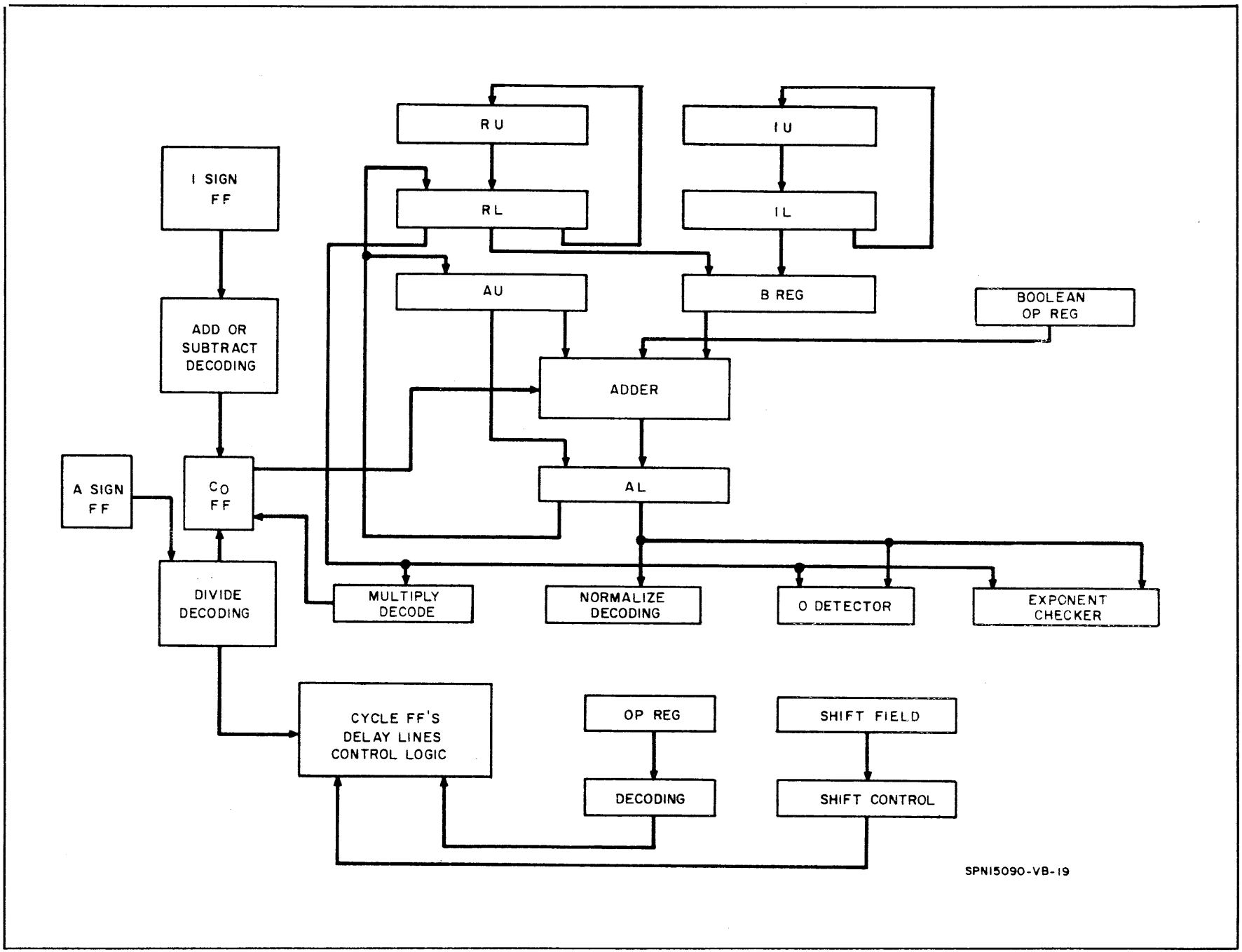


Figure 1-13. Arithmetic Unit

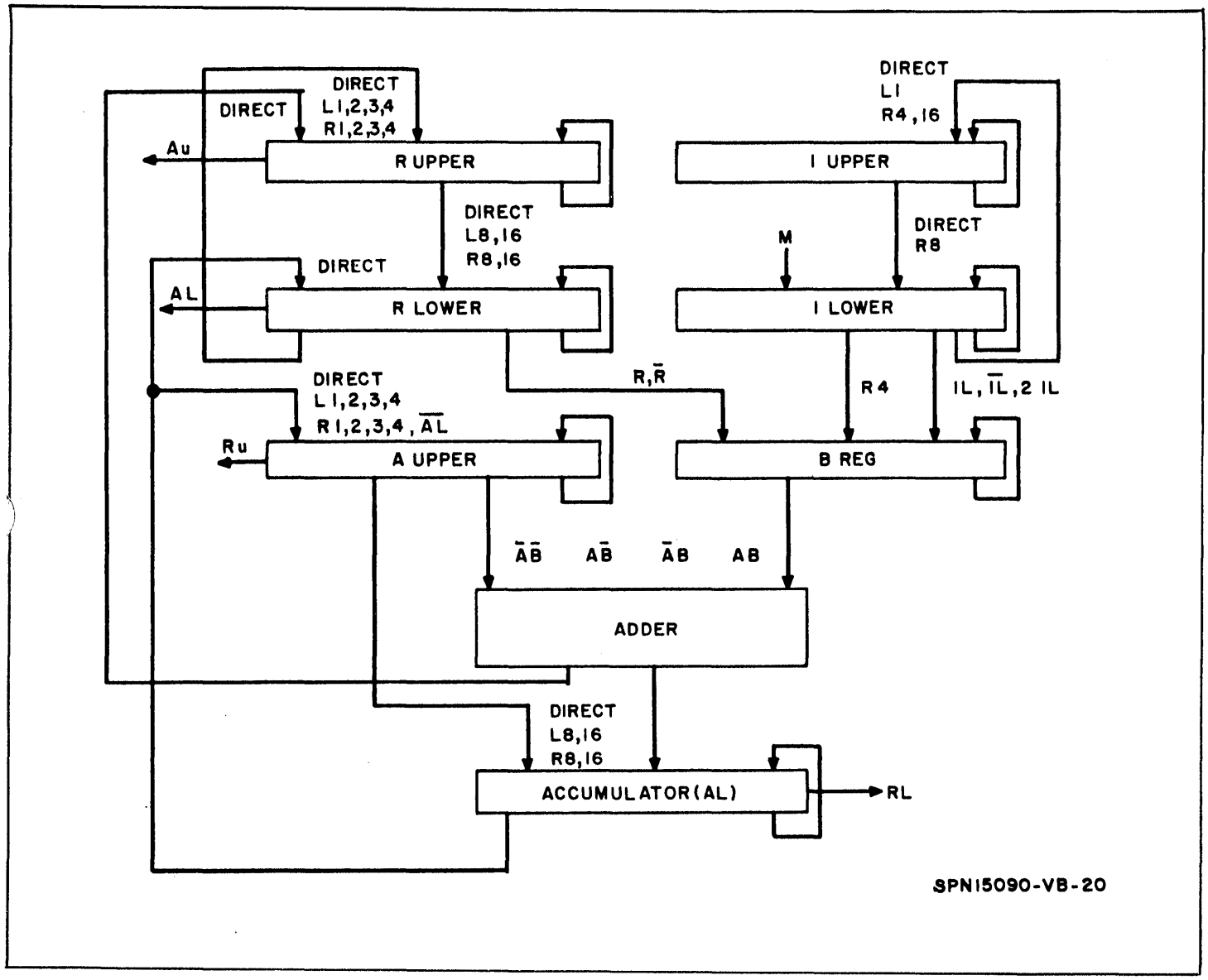


Figure 1-14. Arithmetic Unit

a. **Example: Fixed Point Multiply**

1) Initial Cycle:

The pulse which gates the information into the operation register enters the initial cycle delay line. The first available pulse sets the decoding flip-flops which makes known the type of operation to be executed. The necessary logic is enabled. When a multiply is called for the contents of the A lower register are transferred to the R lower register and serves as the multiplier. If the multiplicand (X operand) has been loaded into the I lower register, a multiply cycle is entered and the outputs of the initial cycle delay line are no longer needed. If the multiplicand has not been loaded the multiply cycle is not entered until the multiplicand is loaded into IL.

2) Multiply Cycle:

A pulse from the initial cycle delay line enters the cyclic delay line to initiate the multiply cycle. The multiply cycle consists of decoding, adding and shift operations.

To minimize the number of multiply cycles, the four least significant bits of the multiplier (contents of the R lower register) are decoded simultaneously. The results of the decoding decide whether the multiplicand (contents of the I lower register), twice the multiplicand, the two's complement of the multiplicand or zero is to be added to the existing partial product. After this addition, the multiplier (RL) is shifted one, two, three or four bits, decided by the previous decoding, and a new set of four bits are decoded to determine the next cycle operation. These cyclic operations are continued until 39 bits have been shifted at which time a final add is done without shifting the partial product. The most significant part of the product is located in the accumulator and the least significant part in the R lower register. The RL register

is shifted right three bits, destroying the original sign of the multiplier, and the three sign bits of the product are inserted into the R lower register. The multiply operation is completed.

The multiply operation just described produces a double-length product. When a single length product is desired a round-off operation is performed during the first cycle. This is accomplished by inserting a "one" in the AU register (in the 39 bit position) and adding this word to the multiplicand. The normal multiply cycles follow.

To multiply two 40 bit words, an average of 17 multiply cycles are executed. Each multiply cycle is 450 nanoseconds.

b. Example: Floating Point Add

1) Initial Cycle:

When the X strobe pulse is received at IL, the floating point add initial cycle timing is started. Exponent (X) and exponent of (AL) are compared to determine whether exponent (AL) is equal to exponent (X), one more than exponent (X), or one less than exponent (X). A subtraction is started with a transfer of the contents of bits 9 through 1 of IL, designated as (IL9 - IL1), to the B register, (AL9 - AL1) to AU9 - AU1 and RL9 - RL1, and the generation of a C_o. The exponent difference is inserted in AL9 - AL1, and is decoded to determine whether the magnitude of the difference is less than 8. The results of the exponent comparison are used to determine the following operations:

a) When the exponents are equal, the add cycle is initiated and (RL9 - RL1) is the exponent of the result.

b) When exponent (AL is one greater than exponent (X), (IL42 - IL10) is shifted four right and the add cycle initiated. (RL9 - RL1) is the exponent of the result.

c) When the exponent (AL) is one less than exponent of (X), (AL74 - AL10) is shifted four right and the add cycle initiated. (IL9 - IL1) is the exponent of the result.

d) When exponent (AL) minus exponent (X) is less than 8 and positive, (IL42 - IL10) is shifted right 4, 8, 12, or 16 bit positions per shift cycle. One, two, three or four is subtracted from the exponent difference for each shift. The shifting continues until the exponent difference is zero. The add cycle is initiated at the end of the last shift cycle. (RL9 - RL1) is the exponent of the result.

e) When exponent (AL) minus exponent (X) is less than 8 and negative, (RL42 - RL10) is shifted right 4, 8, 12, or 16 bit positions per shift cycle. One, two, three, or four is subtracted from the exponent difference for each shift. The shifting continues until the exponent difference is zero. The add cycle is initiated at the end of the last shift cycle, (IL9 - IL1) is the exponent of the result.

f) When exponent (AL) minus the exponent (X) is more than 8 and positive (RL9 - RL1) is the exponent of the result. (RL9 - RL1) is transferred to AL9 - AL1.

g) When exponent (AL) minus exponent (X) is more than 8 and negative, (AL42 - L10) is set to zero and the add cycle is initiated. (IL9 - IL1) is the exponent of the result.

2) Add Cycle:

The floating point add cycle transfers (AL42 - AL10) and (IL42 - IL10) or

(IL42 - IL10) + C_o to the adder as determined by the operation code (add or subtract).

The sum is inserted into AL. If bits AL42 through AL37 are all ones or all zeros a normalize cycle is started. When (AL42) is not equal to (AL41) or (AL40), (AL42 - AL10) is shifted right four places and a one is added to (AL9 - AL1). A carry out from the ninth stage of the adder causes (AL) to be set to all ones and an overflow flip-flop is set.

1.8 INPUT-OUTPUT (I/O) SUBSYSTEM

The SOLOMON II Computer Input-Output (I/O) Subsystem (figure 1-15) provides a maximum of 32 data transfer channels or trunks which handle the buffering and control of all data transfers to or from the computer memory banks and various selected input-output devices. The basic, typical machine employs 5 channels, with 3 standard input-output devices, while the 27 remaining channels are left open and available for use as needs arise. The input-output devices and associated channel controls proposed herein are optional depending on the specific requirements of the user.

Channels 1 and 2 form a channel pair and are associated with up to 16 magnetic tape handlers/high speed printers. The channel pair, in addition to containing the necessary logic required for controlling tape operations, may optionally contain logic for controlling high speed printer operations. Therefore, magnetic tape handlers are interchangeable with high speed printers (plus buffering) from a hardware standpoint.

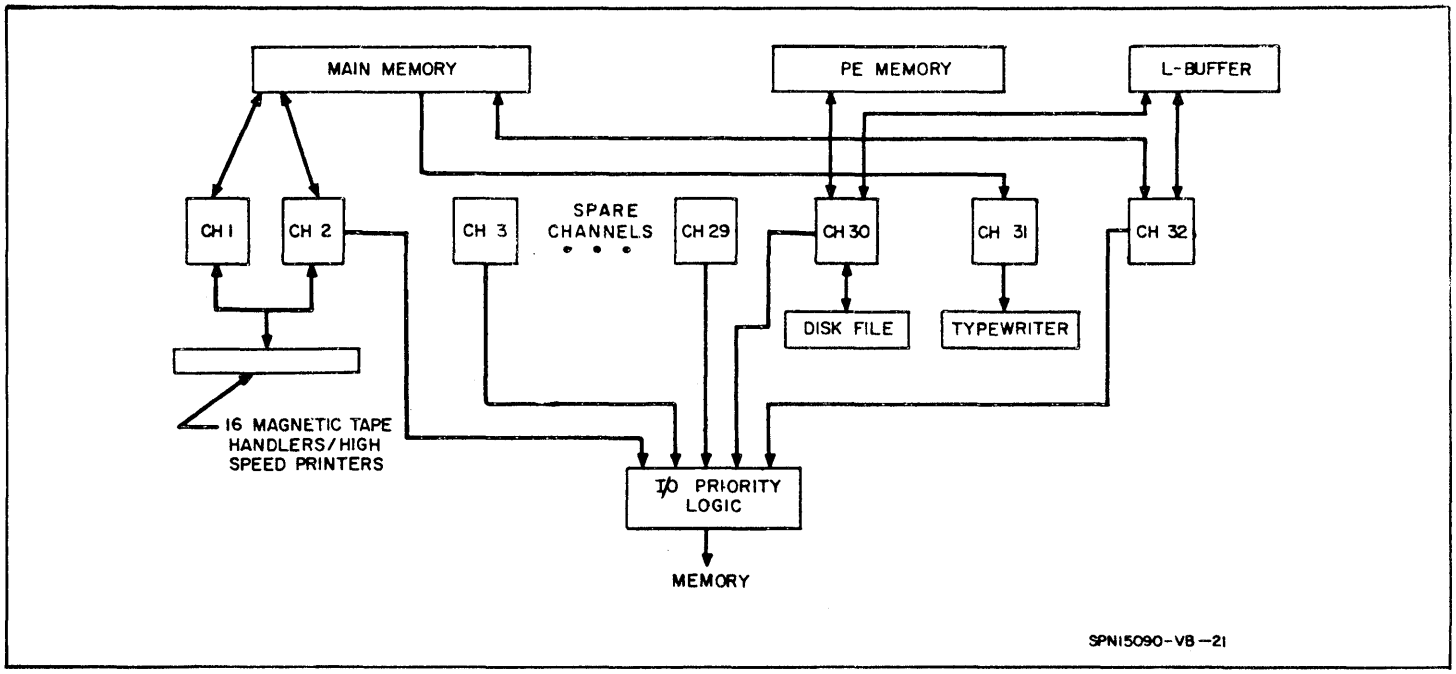


Figure 1-15. Input-Output Data Channels

From a programming standpoint, it is necessary to know whether a unit operating under control of the channel pair is a magnetic tape handler or a high speed printer. Any combination of printers and tape handlers is possible if the sum total of units operating under control of a channel pair is less than or equal to 16. If system requirements dictate the use of more than 16 tape handlers/printers, 2 additional channels (1 channel pair) must be employed for each quantity of 16 units. Other channel pairs can be selected from the 27 spare channels of the proposed 32-channel I/O subsystem. The channel pair control logic is responsible for the selection of specified units, the buffering of data flowing to or from main memory, and the provision of control signals required by specified units throughout an operation. Channel pair operation can be initiated through either the NCU or GPU Control Units.

Channel 30 controls data flow between the PE memory and a high speed, high capacity disk file. The channel includes the disk address verification logic and buffering required in transferring data words to and from the PE memory at a 10-megacycle rate. Channel 30 operation can be initiated through the NCU or GPU.

A typewriter, capable of typing out the 6 characters contained in 1 memory word (40 bits) per command, is controlled by channel 31. The typewriter channel is unidirectional in that typing into memory is disallowed. Channel 31 operation can be initiated through either the NCU or GPU Control Units.

Channel 32 controls the block transfer of data between the main memory and the L-buffer. While instructions which select this channel do not result in the transfer of data to any media external to the computer, the operation is analogous and therefore, most conveniently handled by assigning operation control to an I/O data channel. Channel 32 operations are initiated through either the NCU or GPU.

The I/O subsystem has associated with it priority logic which is responsible for determining which of a group of simultaneously requesting channels may gain access to the memory for a given memory cycle. The I/O priority logic transfers its decision to the main memory priority logic where a final decision on priority with regard to subsystems is made. The order of channel priority is arbitrary and may be easily changed by the modification of patch-board wiring.

Channel busy conditions are sent to the NCU and GPU to prevent a channel from receiving an instruction while carrying out a function specified by a previous instruction. Channel busy is determined at the NCU or GPU for a given instruction specifying a channel operation. Therefore, no instructions violating a channel busy condition are sent from the NCU or GPU to the selected channel. The channel busy logic is updated by the individual channel controls when the busy status of any channel changes.

The I/O subsystem is modular in design resulting in a high degree of feasibility in expanding the proposed system. Channels listed as spares for the proposed system exist only in that provisions are included for addressing them.

At present a single I/O subsystem is proposed for SOLOMON II; however, as previously mentioned, it is possible to have a separate I/O subsystem for both the PE and GP subsystems.

1.8.1 Channels 1 and 2 - Magnetic Tape Transports and Line Printers

A pair of channels or data paths is provided for each group of 16 tape handlers. One or more of the tape units may be replaced with a high speed printer (plus printer control) which may receive data for printout through either channel.

As the 16 tape units are common to both channels, the requirements for channel busy are coordinated through busy logic which is also common to both channels. All commands when issued through either specified channel tie-up the channel control logic for a given interval of time. This time interval is determined for example by the entire time of data flow for a read instruction or by the time required to select a unit for rewind (not the rewind time) in the case of a rewind instruction. The individual channels send busy lines to the NCU and GPU. In addition to the channel busy lines, a single busy line from each unit is also sent to the NCU and GPU to be combined with the channel busy lines to form an execute instruction signal. If the instruction in question violates either the unit busy or channel busy conditions, the operation codes and memory addresses are not sent to the channel and the instruction is not executed until a later time.

Both channels contain the logic involved in the control of the tape transports for all operations. In addition to control logic, the channels each have a data buffer register and storage registers which hold the operation code, memory address, number of blocks, density mode, and number of words to be transferred. Logic common to both channels includes the unit busy flip-flops and the End of File indicator flip-flops.

The channels are capable of supplying characters to the tape units at 2 rates: 120 and 83.4 kc. (If a tape speed of 150 inches per second is assumed, the record rates dictate tape character densities of 800 and 556 characters per inch, respectively). Characters may be read from tape at 3 rates: 120, 83.4, and 30 kc (200 characters per inch). At the 120-kc character rate, 1 word (6 characters) is transferred to or from main memory every 50 microseconds (8.33 microseconds per character).

The 8.33-microsecond character time is adequate to request and gain access to the computer main memory through the I/O priority logic. Therefore data flow is continuous through the channel data buffer registers during a read or write operation. The least significant 4 bits of the 40-bit memory words are ignored in the transfer. A block diagram of a tape transport and printer channel (one-half of a channel pair) is shown in figure 1-16.

The channel control logic for channels 1 and 2 is capable of carrying out the following instructions:

a. Write memory range on tape

b. Read tape into memory range

c. Read N records into memory range

d. Backspace N records

e. Backspace N files

f. Jump on End of File

g. Search for End of File

h. Rewind

i. Rewind and Unload and

j. Write End of File.

1.8.2 Channel 30 - Disk File Modules

Channel 30 controls data transfers between the PE memory and a high speed, high capacity disk file. The channel logic controls read and write operations, locates and verifies specified disk addresses, and detects errors in data transfer. A block diagram illustrating the disk file control is shown in figure 1-17.

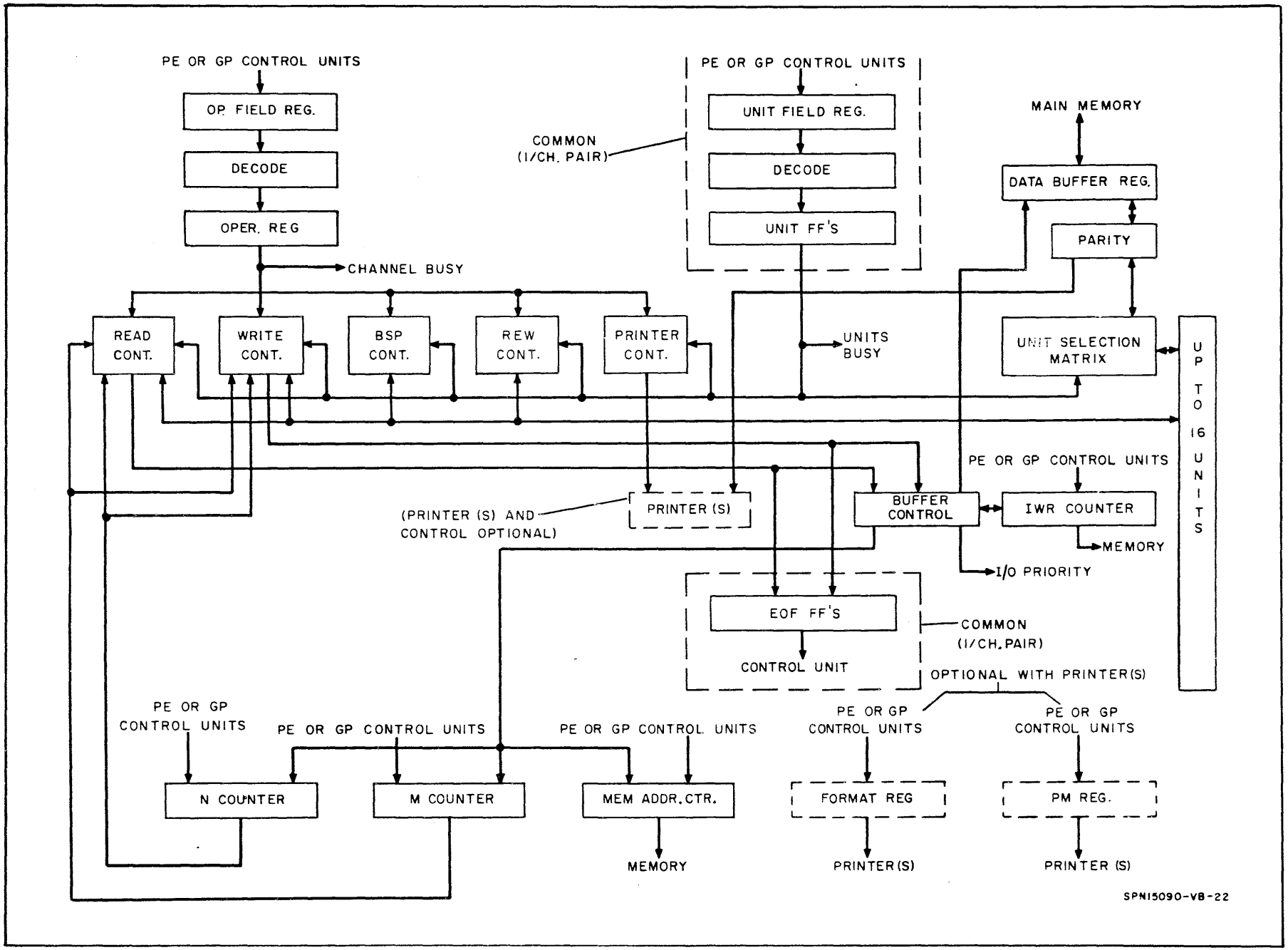


Figure 1-16. Magnetic Tape Transport/Line Printer Channel

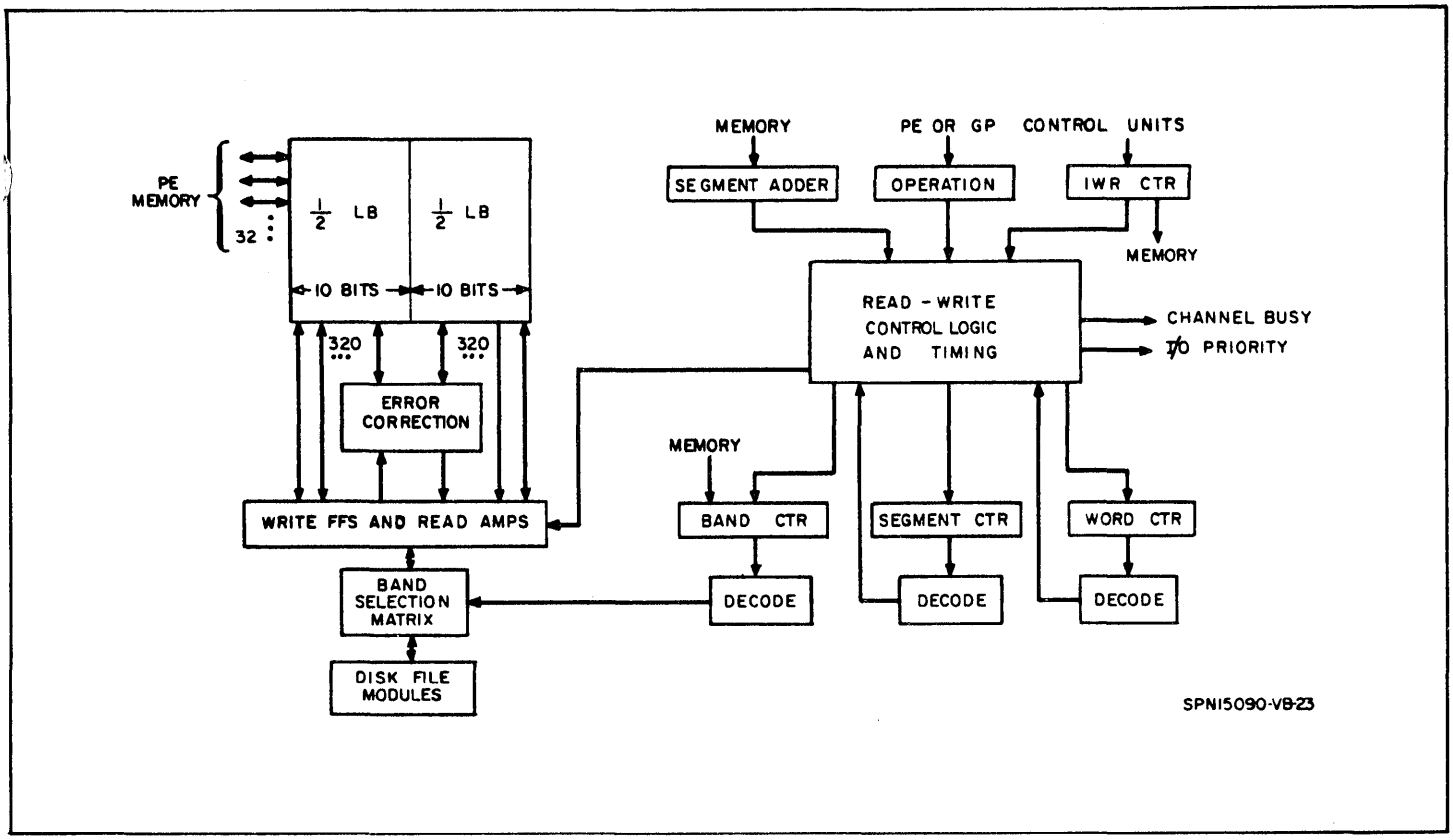


Figure 1-17. Disk File Channel

Table 1-1 lists the significant details of the disk file module organization.

TABLE 1-1

DISK FILE MODULE CHARACTERISTICS

Organization	Operation Frequency (mc)
4 disks per module	Zone 1: 1.008
160 information tracks per band	Zone 2: 1.344
2 bands per zone	Zone 3: 1.446
3 zones per module	
960 information tracks per module	
64 segments per disk side	

TABLE 1-1 (Continued)

Bits Per Track	Bits Per Segment
Zone 1: 40,320	Zone 1: 630
Zone 2: 53,760	Zone 2: 840
Zone 3: 57,856	Zone 3: 904

48,619,520 bits per module (total capacity-information bits)

Data transfers between the PE memory and disk file are routed through the L-buffer. The number of 20-bit PE words readout from (or into) each row or column is controlled by the PE Sequencer. Selection of PE's which supply or receive data is achieved through geometric and mode control. A continuous flow of data between the disk file and L-buffer is achieved by dividing the L-buffer in half. Serial transfers occur alternatively between the PE memory and L-buffer halves. The serial transfer time (1.0 microsecond) between the L-buffer and the PE memory is faster than the minimum parallel transfer time (1.38 microseconds) between the L-buffer and the disk file. Therefore, as the disk file has a continuously available source of (or sink for) data, no time is wasted through loss of a disk revolution. The PE memory is able to supply or absorb data at a greater rate than the disk file. PE memory operation therefore must periodically cease and wait for the disk file to fill or empty one-half of the L-buffer.

Sixteen L-buffer half words (160 bits) are transferred in parallel between the disk file and the L-buffer per disk file cycle. Therefore, two disk cycles are required to

empty or fill one half of the L-buffer. Disk cycle times vary according to which zone of the disk is being used (see table 1-1).

The operation code (read or write) and the instruction word address are delivered to the channel read-write logic from the NCU or GPU. The instruction word is stored in the IWR counter. A request is developed in the I/O priority to gain access to the memory location specified by the IWR counter. The instruction word contains the starting segment address and band selection fields which are stored upon arrival in the corresponding channel registers. Operation differs slightly depending on whether data is being recorded on or read from the disk. For the former case, one half of the L-buffer is loaded from the PE memory. The specified segment address is located on the disk and recording begins. In the latter case, the specified segment is immediately located and data are read from the disk in parallel into one half of the L-buffer. When one half of the L-buffer is saturated, PE memory cycles are initiated and the serial transfer begins. Disk address verification and the checking and single-error correction of data are achieved in the channel. At the termination of each transfer, the IWR counter is checked to determine if another instruction word specifying another transfer is to be obtained. If no more instruction words are specified, the channel is cleared to receive another operation from the NCU or GPU.

1.8.3 Channel 31 - Typewriter

The typewriter channel control is illustrated in block diagram form in figure 1-18. As one typewriter channel instruction exists, involving the output of only one word from memory, the channel operation register consists of a single flip-flop which has the double function of providing a busy indication to the NCU and GPU plus providing an

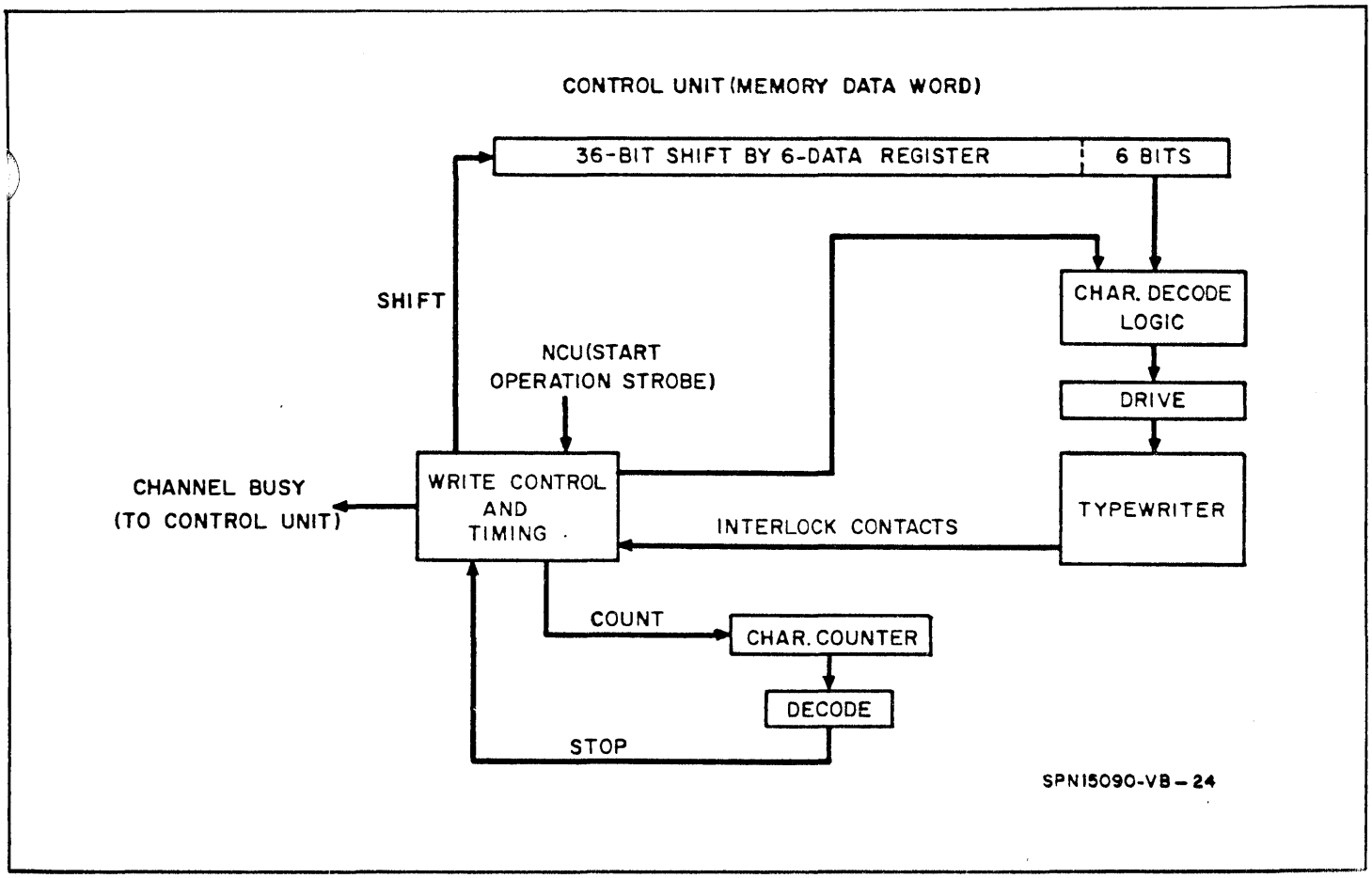


Figure 1-18. Typewriter Channel Control

enabling level to other logic in the channel. As only one memory word is transferred, priority and memory access are achieved before the command is sent to the channel control.

Channel operation is initiated by the start operation strobe from the GPU or NCU. The start operation pulse enters the channel timing unit which provides channel control pulses throughout the operation. Upon receiving the start operation pulse, the channel timing unit generates control pulses which prepare the channel for operation (i.e., load data buffer register, clear character counter, etc.). When the data buffer register is loaded (36-bits of 1 memory word), the character decoding gates are strobed. The amplified decoded outputs operate the character magnets and function

solenoids of the typewriter. When the typewriter begins to type, one of the interlock contacts (tabulate, character, carriage return, or space interlocks) opens to prevent another character selection before the existing type-out is completed. The character is typed and the interlock contact closes, enabling a pulse to enter the timing unit. Upon receiving this signal, the timing unit generates control pulses which simultaneously increment the character counter and shift the data buffer register six bits to the right. When the second character has been shifted to the least significant character position in the data buffer register, the character decode gates are strobed to enable type-out of the second character. The operation proceeds until six characters have been typed. The states of the character counter are decoded to determine the number of characters which have been typed. When the counter reaches the sixth state, and the interlock contact closes, indicating that the last character of the word has been typed, a signal is generated which resets the operation flip-flop, thereby removing the channel busy indication and terminating the operation.

1.8.4 Channel 32 - L-Buffer Transfer Channel

Data transfers between the main computer memory and the PE array memory are routed through the L-buffer. The PE Sequencer controls the serial transfer between the L-buffer and the PE array memory while parallel transfers between the L-buffer and main memory are controlled by Channel 32. The entire transfer is ultimately controlled by the NCU and GPU which send control instructions to both the sequencer (for PE - L-buffer transfers) and channel 32 (for main memory - L-buffer transfers).

A functional block diagram of channel 32 is shown in figure 1-19. The L-buffer read-write control contains the logic which produces all timing pulses throughout an operation. Initially the computer control unit sends a start operation pulse to the

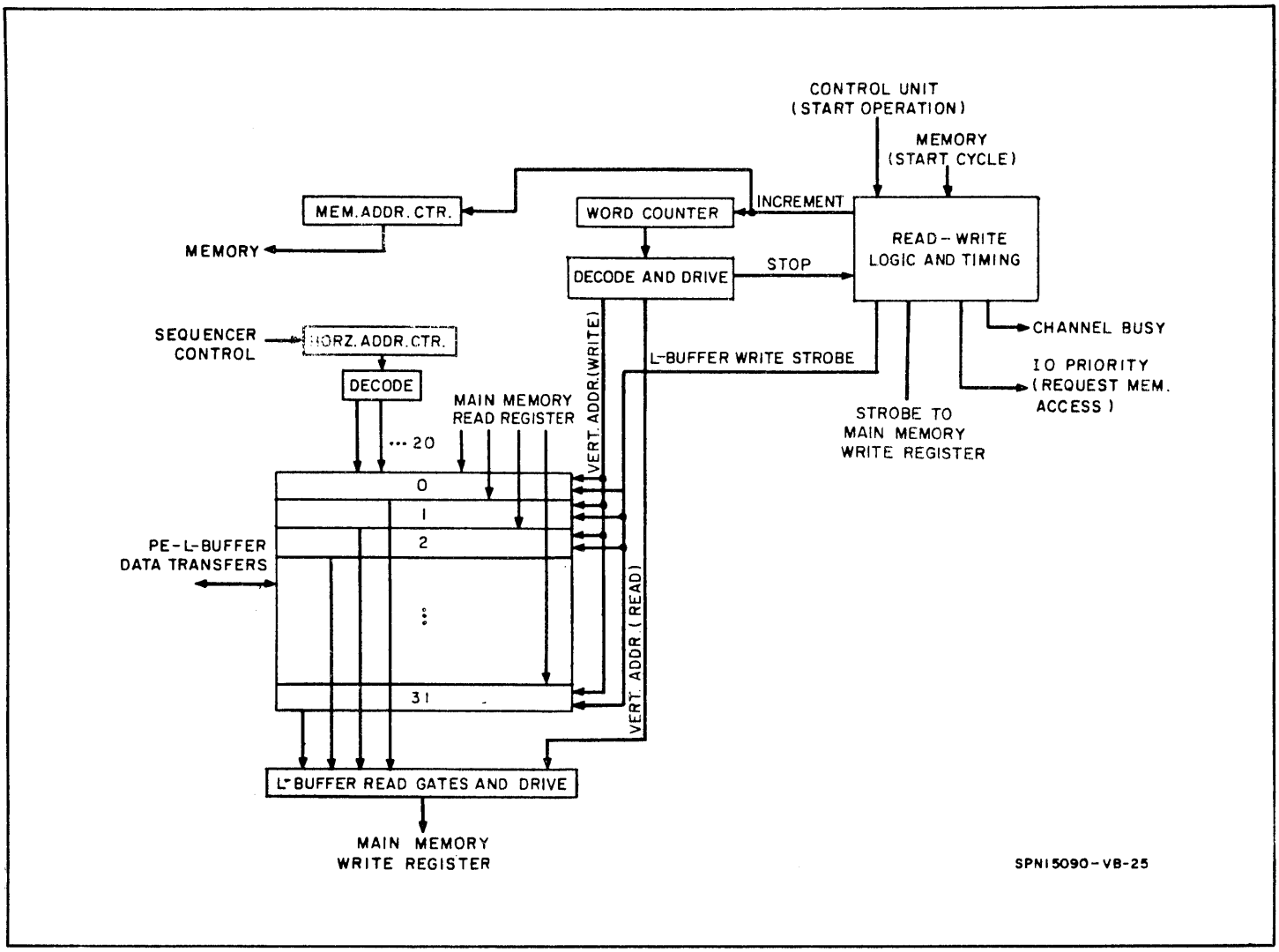


Figure 1-19. L-Buffer Transfer Channel

channel read-write control logic. From the original start pulse the read-write logic develops pulses which clear the channel for operation, sets the busy flip-flop, and enables the initial memory address and operation code to enter the proper channel storage registers. The initial memory address and operation code are sent to the channel from the NCU or GPU along with the start operation pulse. After completing the channel initialization process, the read-write logic sends a signal to the I/O priority logic which develops a request to gain access to the memory location specified by the contents of the channel memory address register. From this point channel

control differs depending upon whether data is being transferred from the main memory to the L-buffer (L-buffer write) or from the L-buffer to the main memory (L-buffer read).

a. L-buffer Write

The word counter is cleared in the initialization process and the word counter decode logic enables the outputs from 160-bit main memory read register to bear on the flip-flop inputs of the first 8-word positions of the L-buffer (word positions 0 - 7). As the L-buffer word length is 20 bits, 4 memory cycles (8 L-buffer words per cycle) are required to completely fill the 32-word L-buffer.

When the request for access to the specified memory location has been granted, a signal is sent to the read-write control logic. The read-write logic initiates a delay of proper length to assure that the memory word has entered the main memory read register. When the delay is completed, an L-buffer write strobe is sent from the read-write logic to the L-buffer, thereby setting the contents into the first 8-word positions of the L-buffer. The word counter and memory address counters are incremented by one and a request for access to the second memory address is developed in the I/O priority logic. The second state of the word counter is decoded and enables the data flow path from the memory read register to the second group of 8 L-buffer word positions. The procedure is repeated until 32 L-buffer words (4 memory words) have been transferred. The terminating condition is sensed by the read-write logic through the word counter decode logic. Upon sensing the terminating conditions, the read-write logic resets the busy flip-flop, thereby ending the operation.

b. L-Buffer Read

The channel is initialized by the start operation pulse in the same manner as described above for L-buffer write operations. The outputs of the L-buffer flip-flops from the first 8 word positions are collected in the L-buffer read gates. The word counter decode logic enables the outputs from the correct 8 word positions to flow to the main memory write register. When access to the specified memory location is granted, the read-write logic sets the main memory write register. The word counter and memory address counter are then incremented by one and a request for access to the location specified by the incremented memory address counter is developed in the I/O priority logic. When the read-write logic senses that 32 L-buffer words have been transferred, the channel busy condition is removed and the operation is terminated.

The main memory cycle time is 1 microsecond. As 8 L-buffer words are transferred per main memory cycle, the time required to transfer 32 L-buffer words is 4 microseconds. The sequencer can transfer 32 20-bit words between the L-buffer and PE memory in 2.2 microseconds. Therefore, the overall transfer time for 1 block of data is 4 + 2.2 = 6.2 microseconds. As 640 bits are transferred per block, the bit transfer rate is $640/6.2 \times 10^{-6} = 103.2 \times 10^{6}$ bits per second.

1.9 INSTRUCTION DESCRIPTIONS

In the detailed instruction descriptions, the letters which appear in parentheses represent, in the mnemonic code, the option whose description they follow. The same letter may represent more than one option in a code, in which case its sequential appearance in the code matches the sequence of its definitions.

Tables 6-2, 6-3, and 6-4 depict the operations and their mnemonic in condensed form. For the convenience of the reader, these tables appear later in this section adjacent to the applicable data.

1.10 NCU AND GPC INSTRUCTIONS

1.10.1 Transfers With Memory

A transfer (T) of the operand in the memory location specified by M (M) to W (W), B (B), CG (CG), RG (RG), SL (SL), or MO (MØ) may be made. Data transfers are made between like numbered bit positions. Data are transferred from the first named register to the last named register.

A transfer from W (W) to the memory location (M) may be made.

The acceptable mnemonic codes are:

TMW	TMRG	TMMØ
TMB	TMCG	TWM
TMSL		

1.10.2 Transfers to W

A transfer (T) of the contents of CG (CG), RG (RG), or SS (SS) to W (W) may be made. An operand containing all zeros (Z) or all ones (Ø) may also be transferred. A preclear of W is provided.

The acceptable mnemonic codes are:

TCGW	TZW
TRGW	TØW
TSSW	

1.10.3 Transfers to V

The transfer (T) of an operand containing all zeros (Z) or all ones (Ø) to V (V) may be made.

The acceptable mnemonic codes are:

TZV TØV

On transfers to V with the immediate address option, the M Field is placed into the V register and the option to clear or not clear the remaining bits of W is available.

1.10.4 Transfers With Index Registers

A transfer (T) of the value contained in the M Field of an instruction (I), the value in M_{20} -M_{39} of the memory location specified by M (M), or the value in V (V) to the index register (Y) specified by the Y Field of the instruction may be made.

The two's complement (C) of a value may also be transferred.

A transfer from an index register (X) to V (V) may be made.

Index registers may contain fewer than 20 bits. The missing bits should be considered to be on the high order (left) end, and zero.

These instructions are indexable.

The acceptable mnemonic codes are:

TIY	TIY	TMY	TMY	TVY	TVY	TXV
TIYC	TIYC	TMYO	TMYC	TVYC	TVYC	TXVC

1.10.5 Arithmetic With Index Registers

An addition (A) or subtraction (S) of the value contained in the M Field of the instruction (I), or the value in M_{20} - M_{39} of the memory location specified by M (M) to or from the value in the index register (X) specified by X may be made.

The result is stored in the index register (Y) given by the Y Field or into the V register.

These instructions are not indexable.

The acceptable mnemonic codes are:

AIXY	AIXV	AMXY
SIXY	SIXV	SMXY

1.10.6 Arithmetic With V

An addition (A) or subtraction (S) of the value contained in the M Field of the instruction (I), or the value in M_{20} -M_{39} of the memory location specified by M (M) to or from the value in V (V) may be made.

The values are considered to be signed, and overflow occurs if $C_o \neq C_1$.* If overflow detection is specified (Ø), the Overflow Indicator is reset to zero initially and set to one if overflow occurs.

The result is stored in V.

The acceptable mnemonic codes are:

AIV	AMV	SIV	SMV
AIVØ	AMVØ	SIVØ	SMVØ

All immediate address options are not indexable.

1.10.7 Change of Sign in V

The sign of the value in V (V) may be set (S) opposite (Ø).

If V contains 10 ... 0, overflow occurs. If overflow detection is specified (Ø), the Overflow indicator is reset to zero initially and set to one if overflow occurs.

The acceptable mnemonic codes are:

CVS CVSØ

*$C_o \neq C_1$: the carry from position 1 to position zero, C_1, is not equal to the carry lost from position zero.

1.10.8 Shifts

A shift (S) of the operand in W (W) or the value in V (V) may be made to the left (L) or right (R). The operand in W may be shifted circularly (C).

Shifts in W are logical (L), (unsigned), and shifts in V are algebraic (A), (signed).

Overflow occurs on left algebraic shifts when a shift into V_1 is made while $V_1 \neq V_o$. If overflow detection is specified (Ø), the Overflow Indicator is reset to zero initially and set to one if overflow occurs.

The number of places to shift is specified by M modulo 32. Each type of shift is defined below by a diagram of a one place shift. Shifts of more than one place are repetitions of the one place shift.

SWLL

SWRL

SWLC

SVLA
SVLAØ

SVRA

1.10.9 Boolean Operations

With the operand in W (W) or V (V) as the first operand, a Boolean operation is performed.

The operation may be AND (A), Negated Coimplication (NC), Negated Implication (NI), Exclusive OR (EØ), OR (Ø) Negated OR (NØ), Equivalence (EQ), Implication (I), Coimplication (C), and Negated AND (NA).

The second operand is either the value contained in the M Field of the instruction (I), or the contents of the memory location specified by M (M).

The operation, NOT (N), may be performed on the operand in W (W), V (V), M Field (I), or memory contents (M) with storage in W (MW) or V (MV).

In all cases the result is stored in W or V.

The results of the four possible bit pairings are tabulated in table 6-1.

The acceptable mnemonic codes are:

WAM	VAI
WNCM	VNCI
WNIM	VNII
WEØM	VEØI
WØM	VØI
WNØM	VNØI
WEQM	VEQI
WIM	VII
WCM	VCI
WNAM	VNAI
NW	NV
NMW	NIV

The ability to perform boolean functions on the index registers is also provided. All sixteen boolean functions are available between (X) and (V) or (M) or (I).

Indexing is not available on Immediate address options.

1.10.10 Jumps

A jump (J) causes a new sequence of instructions to be established, beginning with the instruction in the memory location specified by M.

The jump may occur unconditionally or only if some condition is met. If a condition is specified and it is not met, the normal sequence is maintained. When a new

sequence is established, Index Register 7 may be set (S) to the location of the next instruction in the normal sequence.

The acceptable mnemonic codes for the unconditional, nonindex modifying jumps are:

J	JS

1.10.11 Jumps Which Modify Index Registers

An unconditional jump occurs after adding (A) or subtracting (S), the value in the V Register from the index register (X) specified by X.

These instructions are not indexable and the setting of Index Register 7 with the next sequential location is not allowed.

The acceptable mnemonic codes are:

JAX	JSX

1.10.12 Jump On Count Equal to Zero

The instruction increments (I) or decrements (D) the value in the next register (X) by one, places the result into the index register (Y), and jumps to the memory location given by M when the value (X) ± 1 is equal to zero.

These instructions are not indexable.

The acceptable mnemonic codes are:

JIZ	JDZ	JSIZ	JSDZ
JINZ	JINZ	JSINZ	JSDNZ

1.10.13 Jumps Which Test Index Registers

The status of the index register (X) specified by X may provide the condition for jumping.

The index register may be tested for being zero (Z), or nonzero (NZ) or for being equal to (E), not equal to (NE), greater than or equal to (GE), or less than (L) the value in V.

These instructions are not indexable.

The acceptable mnemonic codes are:

JXZ	JSXZ	JXGE	JSXGE
JXNZ	JSXNZ	JXL	JSXL
JXE	JSXE		
JXNE	JSXNE		

1.10.14 Jumps Which Test W or V

The status of W (W), V(V), the high order position of V (HV), or the low order position of V (LV) may provide the condition for jumping.

The V and W Registers may be tested for being zero (Z) or nonzero (NZ).

The bit positions may be tested for being zero (Z) or one (Ø).

The acceptable mnemonic codes are:

JWZ	JSWZ	JHVZ	JSHVZ
JWNZ	JSWNZ	JHVØ	JSHVØ
JVZ	JSVZ	JLVZ	JSLVZ
JVNZ	JSVNZ	JLVØ	JSLVØ

1.10.15 Jumps on Overflow

The status of the Overflow Indicator may provide the condition for jumping.

The Indicator may be tested for overflow (Ø), (one), or no overflow (NØ), (zero). A reset (R) to zero of the indicator may occur after the test.

The acceptable mnemonic codes are:

JO	JSØ	JØR	JSØR
JNO	JSNØ		

1.10.16 Jumps on I Indicator Status

The I Indicator (I) may be tested for zero (Z) or one (Ø) as the condition for jumping.

The acceptable mnemonic codes are:

JIZ	JSIZ
JIO	JSIØ

1.10.17 Jumps on Network Status

The PE Network (N) may be interrogated to determine if there are some active (A) PE's or no active (I) PE's. This instruction uses the MD Field as in PE instructions.

The acceptable mnemonic codes are:

JNA	JSNA
JNI	JSNI

1.10.18 Indicator Sets on Index Comparisons

The I Indicator (I) may be set to one (S) if the index register (X) specified by X is equal to (E), not equal to (NE), less than (L), or greater than or equal to (GE) the value in the M Field of the instruction. A preliminary reset (R) to zero of the indicator is allowed.

This instruction is not indexable. The mnemonic codes are as follows:

ISXE	ISXNE	ISXL	ISXGE
ISXER	ISXNER	ISXLR	ISXGER

1.10.19 Indicator Sets on Boolean Comparisons

The I Indicator (I) is set to one (S) if the result of a Boolean operation (B) is zero (Z) or not zero (NZ). The Boolean operation performed is specified by the MD Field of the instruction and may be any of the boolean operations which involve W and M, V and I, X and V, X and W, X and I. The result of the Boolean operation is not

saved; the register contents are not modified. A preliminary reset (R) to zero of the indicator is allowed. The mnemonic codes are as follows:

ISBZ ISBNZ
ISBZR ISBNZR

1.10.20 Halt and Jump

The Control Unit will halt (H). Upon depression of the START Button by the operator, an unconditional jump (J) or jump with set (JS) occurs.

The acceptable mnemonic codes are:

HJ HJS

1.10.21 No Operation

No operation (NØP) is performed and the next sequential instruction is obtained.

The mnemonic code is:

NØP

1.10.22 Input-Output

An Input or Output (IØ) operation is attempted. The details of these operations will be given later.

The mnemonic code is:

IØ

1.11 PE INSTRUCTIONS

1.11.1 Transfers

A transfer (T) of the operand in P (P) or Q (Q), or an operand of all zeros (Z), to the PE Network memory (M) may be made.

A transfer of an operand of all zeros (Z) to P (P) or Q (Q) may be made.

A transfer of the operand specified by the Routing Field (R) to P (P) or Q (Q) may be made.

An exchange (E) of operands between P (P) and Q (Q) may be made.

The ecceptable mnemonic codes are:

TPM	TZP	TRP	EPQ
TQM	TZQ	TRQ	
TZM			

1.11.2 Transfers With the L-Buffer

A transfer (T) of the operands in the L-Buffer (L) to the Network Memory location specified by M (N) may be made under geometric control. See paragraph 3.2.6.

If Row Geometric Control is specified, the L-Buffer operands are transferred to active PE's in corresponding columns (L0, L1, ..., L31 to C0, C1, ..., C31). If Column Geometric Control is specified, transfer is to the corresponding rows (L0, L1, ..., L31 to R0, R1, ..., R31).

If neither Row nor Column Geometric Control is specified, no transfer occurs. If both are specified, the result is undefined.

A transfer from the PE Network Memory location specified by R (R) to the L-Buffer (L) may also be made under geometric control. If more than one operand is transferred to the same L-Buffer location, the logical OR of the operands is stored. If no operand is transferred to an L-Buffer location, the location is set to zero.

The acceptable mnemonic codes are:

TLN	TRL

1.11.3 Addition and Subtraction With P

An addition (A) or subtraction (S) of the operand in P (P) to or from the operand specified by R (R) may be performed. The result is stored in P.

The absolute value (A) of the operand in P may be used.

If overflow detection is specified (Ø) and overflow occurs, MS may be set to the value specified by MV.

The use of the Carry Indicator is specified (C), the Carry Indicator is not set initially; otherwise it is set as required. In either case, the Carry Indicator is set to the value of C_1 which occurs during the operation. When use of the Carry Indicator is specified, the operand from P is an extended operand (if the operand is complemented, it is one's complemented).

The acceptable mnemonic codes are:

AP	SP
APØ	SPØ
APC	SPC
APCØ	SPCØ
APA	SPA
APAØ	SPAØ
APAC	SPAC
APACØ	SPACØ

1.11.4 Addition and Subtraction With R

An addition (A) or subtraction (S) of the operand specified by R (R) to or from zero (Z), or from the operand in P, may be performed. The result is stored in P.

If overflow detection is specified (Ø), and overflow occurs, MS is set to the value specified by MV.

If use of the Carry Indicator is specified (C), the Carry Indicator is not set initially; otherwise it is set as required. In either case the Carry Indicator is set to the value of C_1 during addition. When use of the Carry Indicator is specified, the operand specified by R is an extended operand.

The acceptable mnemonic codes are:

ARZ	SRZ	SR
ARZC	SRZØ	SRØ
ARZCØ	SRZC	SRC
	SRZCØ	SRCØ

1.11.5 Multiplication

Multiplication (M) of the operand in Q (Q) by the operand specified by R may be performed, and the result is stored in P and Q.

The most significant half of the product is stored in P and the least significant half in Q. The sign of the product appears in both P_0 and Q_0.

The acceptable mnemonic codes are:

MQ	MQØ

1.11.6 Division

Division (D) is performed with the operand in P (P) as the most significant half of a double length dividend, and the operand specified by R as the divisor. The least significant half of the dividend may be zero or the operand in Q (Q).

The quotient is stored in P and the remainder is stored in Q.

The original value of Q_0 does not effect the result.

The absolute value of the operand in P is larger than the absolute value of the operand specified by R, or if the operands are algebraically equal, the quotient will not be valid.

The acceptable mnemonic codes are:

DP	DPQ
DPØ	DPQØ

1.11.7 Sign Operations

The sign of the operand in P (P) may be set (S) positive (P), negative (N), or opposite (Ø).

Overflow occurs if P contains 10 ... 0 and the sign is set positive or opposite. If overflow detection is specified (Ø) and overflow occurs, MS is set to the value specified by MV.

The acceptable mnemonic codes are:

PSP	PSPØ
PSN	PSØØ
PSØ	

1.11.8 Shifts

A shift (S) of the single length operand in P (P) or the double length operand in P and Q (PQ) is made to the left (L) or right (R). The shift may be logical (L) or algebraic, (A); that is, unsigned or signed.

The operand in Q (Q) may be shifted circularly (C).

Overflow, which may only be detected on left algebraic shifts, occurs when a shift into P_1 is made while $P_1 \neq P_0$. If overflow detection is specified (Ø) and overflow occurs, MS may be set to the value specified by MV.

The number of places to shift is specified by M modulo 21. Each type of shift is defined below by a diagram of a one-place shift. Shifts of more than one place are repetitions of the One-place shift.

SPLL

SPRL

SPLA
SPLAØ

SPRA

SPQLL

SPQRL

SPQLA
SPQLAØ

SPQRA

SQC

1.11.9 Boolean Operations

With the operand in P (P) as the first operand, a Boolean AND (A), OR (Ø), or Exclusive OR (EØ) is performed. The operand specified by R (R) is the second operand.

The operation NOT (N) may be performed on the operand in P (P) or the operand specified by R (R).

The result is stored in P in all cases.

The acceptable mnemonic codes are:

PAR	NP
PØR	NR
PEØR	

1.11.10 Mode Sets From MV

The MS Register (M) may be set (S) to the value in MV. According to the options to set either, both or neither of the MS Register flip-flops.

The set may be unconditional; that is dependent only on the instruction (I).

The set may be conditional on the result of a comparison of the operand in P (P), or the absolute value (A) of that operand, with the operand specified by R. The operand in P may be tested for being equal to (E), not equal to (NE), greater than (G),

greater than or equal to (GE), less than (L), less than or equal to (LE), the R operand.

The set may also be conditional on Q_{19} (LQ) being zero (Z) or one (Ø).

The acceptable mnemonic codes are:

MSI	MSPL	MSPAL
MSE	MSPGE	MSPAGE
MSNE	MSPG	MSLQZ
MSPAE	MSPLE	MSLQØ
MSPANE		

1.11.11 Miscellaneous Sets

The MS Register (M) may be set (S) from Q_{18}, Q_{19} (Q).

The Carry Indicator (C) may be set (S) from Q_{19} (Q).

Bits Q_{18}, Q_{19} (Q) may be set (S) from the MS Register (M). Bit Q_{19} (Q) may be set (S) from the Carry Indicator (C). None of these sets change other bits in Q.

The Geometric Control Registers (G) may be set (S) from PE Network (N) conditions. Each bit position of the CG Register which corresponds to a column containing at least one active PE is set to one. Each bit position of the RG Register which corresponds to a row containing at least one active PE is set to one. Other bit positions of the CG and RG Registers are set to zero. Normal geometric control is available with this set.

The acceptable mnemonic codes are:

MSQ	QSM	GSN
CSQ	QSC	

1.12 CIRCUITS

1.12.1 Stroke Gates

The stroke gate microelectronic device is the basic circuit for the SOLOMON-II system. The quantity of gates used in the system dictates that the individual gate must have extremely high reliability under the worst operating conditions. Saturated logic circuitry of the stroke gate configuration offers the advantages of: decreased power dissipation, circuit simplicity, high noise rejection, operation under wider variations in component parameters and supply voltage, as compared to other forms of logic circuitry. These characteristics provide the extremely high reliability demanded by the system.

The SOLOMON-II stroke gates's logic function (Figure 1-20) may be given as:

$$D = a \cdot b$$

$$C = \overline{a} + \overline{b}$$

The diodes A and B consitute one AND gate. The logic levels of the gate (0 and 1) correspond to voltage levels of .3 and 6 volts, respectively.

With the input (A and B) high (+6 volts), the diodes of the gate will be back-biased and current will flow from E_1 through R_1, D_1' and the D_2's to E_2, and the base of the transistor. The majority of this current will flow into the base of T. Saturating the transistor, the collector voltage of T_1 will drop to approximately 0.3 volts. With point C at approximately .3 volts, diode D_1 in the second stage will conduct; the transistor in the second stage will cut off, because of the voltage drop across the three D_2's will drive the transistor T_2 base negative. If one input of the AND gate is low (.3 volts) the above circuit operation will reverse, with the transistor in the first

stage cutting off and the transistor in the 2nd stage saturating, thus both transistors in stage 1 and 2 switch between the cut off and saturated modes. In order to start switching of the gate, the input has to exceed approximately 2.4 volts; which is a desirable level for a computer circuit. Point E in Figure 1-20 is for the purpose of (OR)ing together additional AND gates.

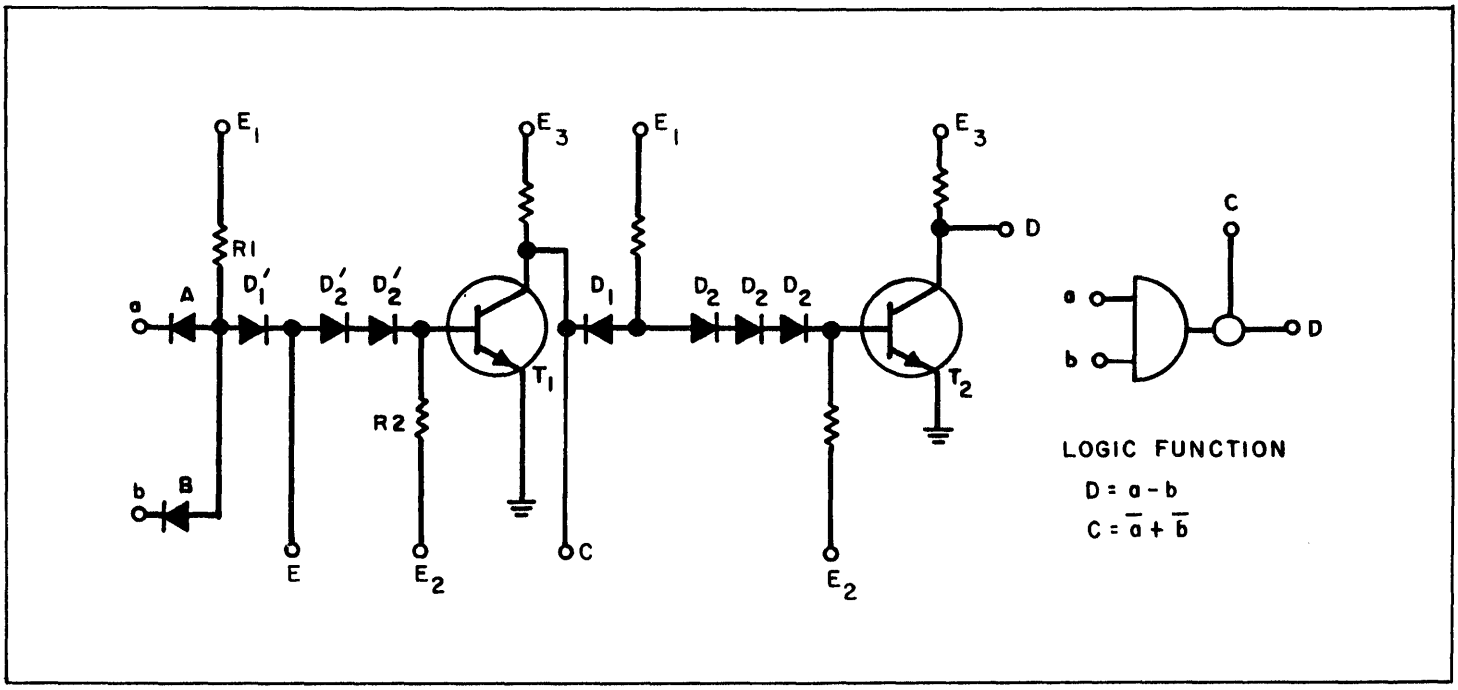


Figure 1-20. SOLOMON Stroke Gate

1.12.2 The Integrated Logic Circuits

The microelectronic or molecular logic circuits designed for the SOLOMON system consist of: diode AND gates, a two stage driver and an inverting amplifier, i.e., dual strokes. The circuits (Med 1 - Med 11) are shown in (Figures 1-22 - 1-20) and are illustrated logically in Figure 1-21. In Med 1 and Med 2 (Figures 1-20, 1-21) the AND gates are connected to the first stage of the driver to perform the OR function, with the two-stage driver supplying the direct and complement function of the input; while in

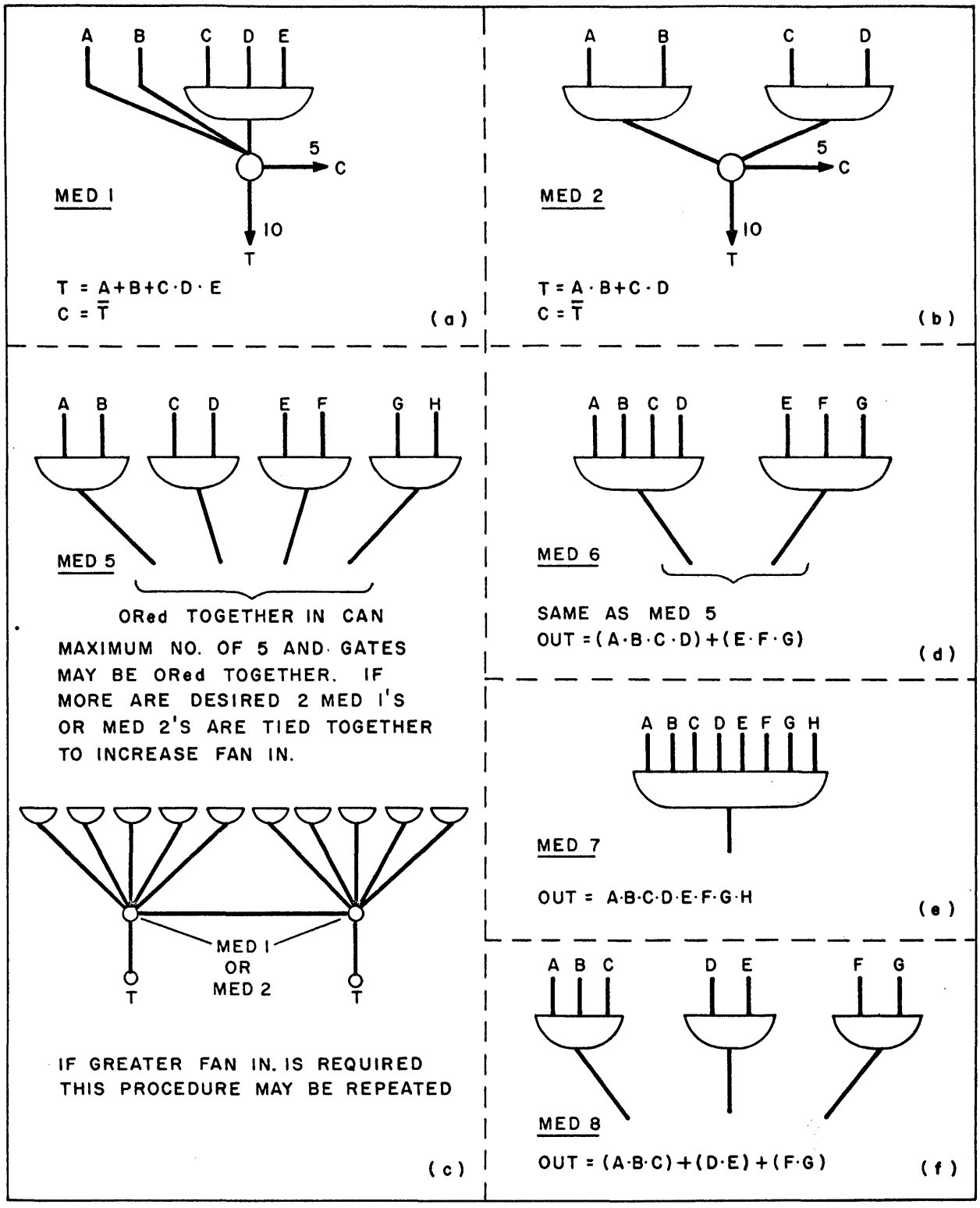


Figure 1-21. Molecular Logic Modules (Sheet 1 of 2)

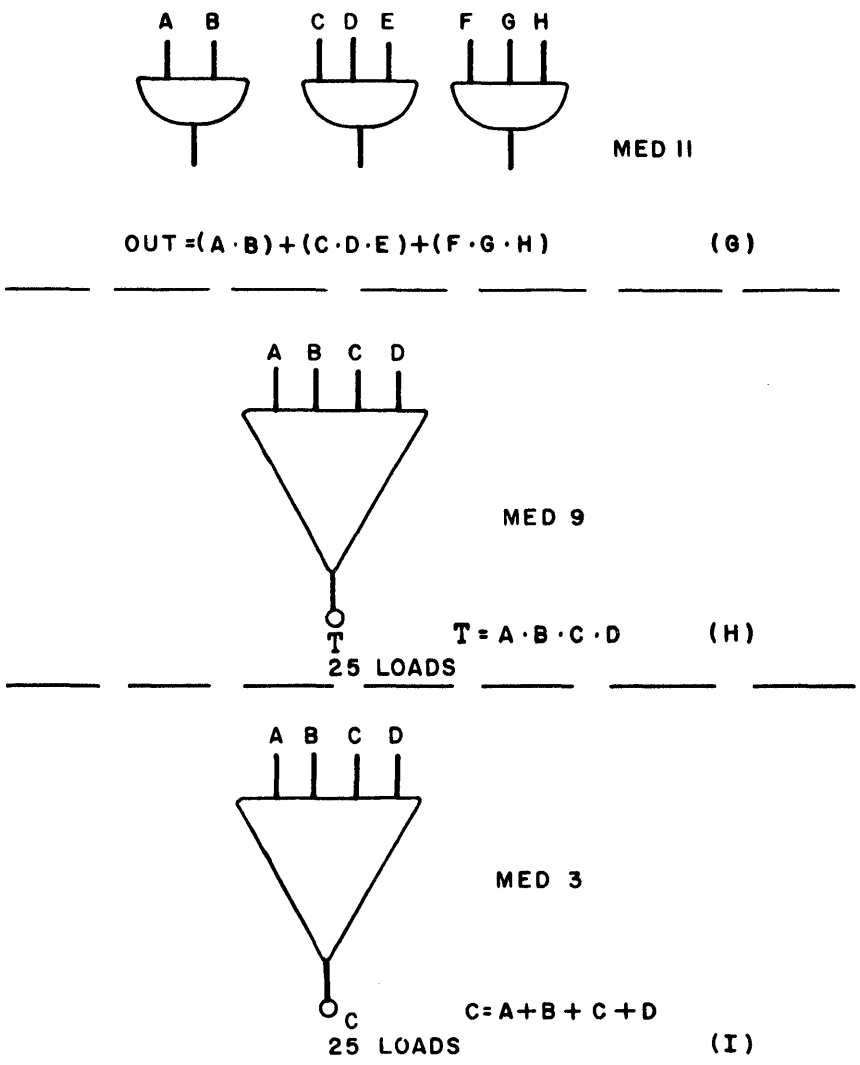


Figure 1-21. Molecular Logic Modules (Sheet 2 of 2)

Med 9 only the direct function is supplied as an output (see Figure 1-21H, 1-24). The complete logic circuitry is implemented with 10 fundamental integrated circuits packaged in 10-Pin TO-5 cans (Med 1 through Med 11), which in turn are mounted on printed circuit boards. The integrated circuits will consist of silicon and/or thin film single element chips and/or multiple element chips arranged and interconnected to perform the circuit function.

The drivers (Med 1, 2 and 9) utilize a very low storage transistor in the first stage. This allows high speed, although a high speed recovery diode blocks removal of stored carriers from the transistor. The propagation delay time through the driver is less than 25 nanoseconds (turn-on-time and turn-off-time of the driver are both less than 25 nanoseconds). However, with full loading of the drive and with associated wiring capacitance in actual use, the driver is expected to have a propagation delay time of less than 50 nanoseconds. The voltage and resistor values for the drivers were chosen to optimize the switching efficiency and to with stand ±10 percent changes in voltages and resistors. In the worst case with full fan out loading, the circuits were optimized so that the transistors require a minimum beta. The input loading of 2.65 M.A. (1 unit load nominal case) was chosen as a sufficient current level to charge wiring and stray capacitance, while keeping the power dissipation at a reasonable level.

Figures 1-21 and 1-25 illustrate the AND gates. The number of inputs on an AND gate and the number of AND gates in a 10-Pin TO-5 can package were chosen such that a minimum number of TO-5 cans would be required to implement the logic circuitry. Each AND gate has an input loading of 2.65 M.A.. As many as 5 AND gates may be "OR" ed together at nodal point N on the drivers. If it is desired to OR more gates

together, Med 1 or Med 2 may be connected as shown in Figure 1-21C. The tolerance on the AND gate voltage and resistor value is ±10 percent. The diodes are all high speed fast recovery type 1N905 or equivalent.

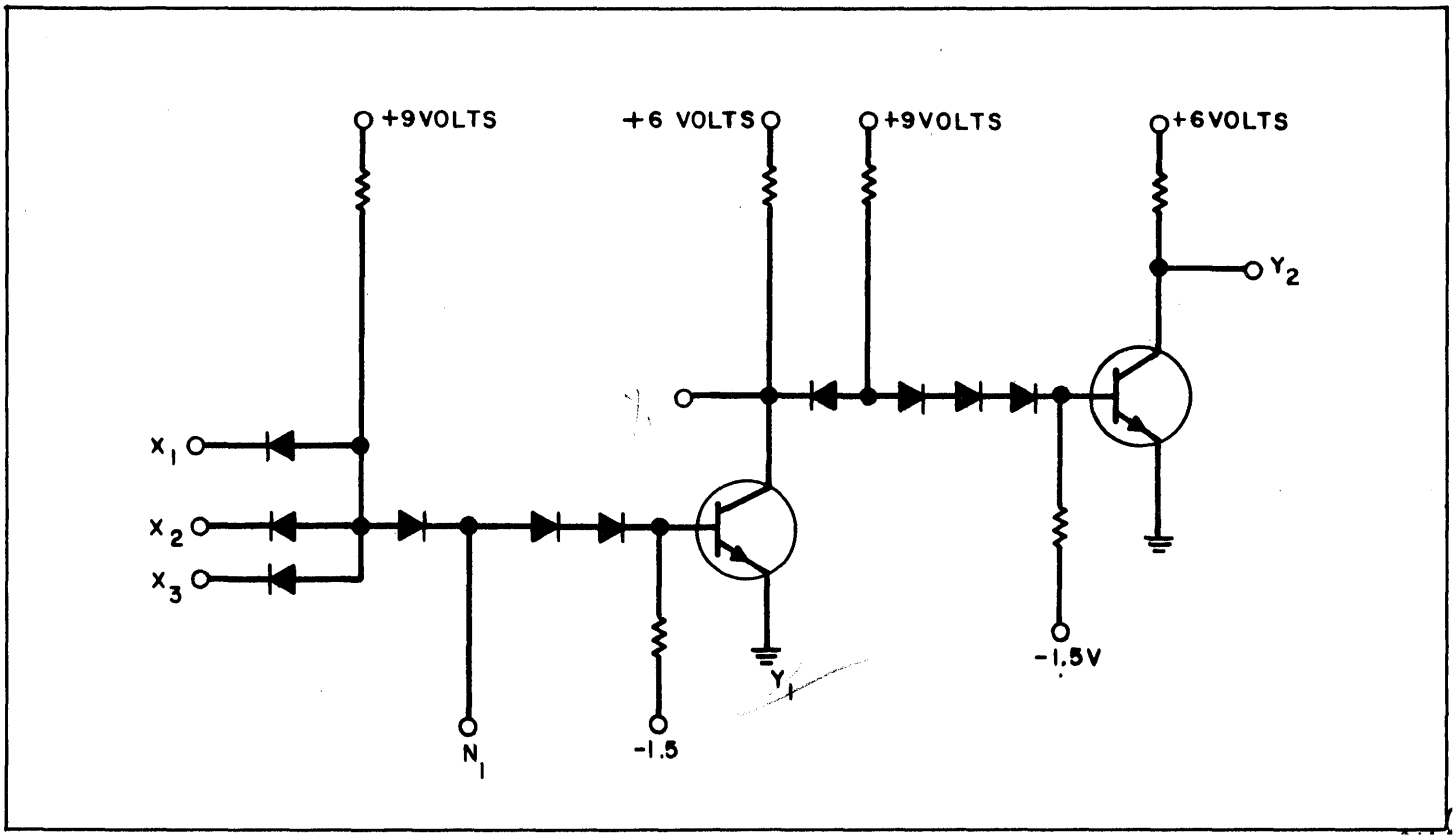


Figure 1-22. Med 1

Specifications

Logic Function: $Y_1 = \overline{(X_1 \cdot X_2 \cdot X_3) + N_1}$

$Y_2 = (X_1 \cdot X_2 \cdot X_3) + N_1$

N_1 is a nodal point where diode AND gates may be connected where they are "OR"ed with diode gate X_1, X_2, X_3 (Med 5, 6, 7, 8, 11).

Input Loading: 1 load each

Maximum Circuit Delay: .025 microseconds

Output Drive: Y_1 = 5 loads

Y_2 = 10 loads

Temperature Range: 0° C to 125° C

Maximum Power Requirement Per Pack:

	Logic 1	Logic 0
+6	1.45 M.A.	4.15 M.A.
+9	6.3 M.A.	6.03 M.A.
-1.5	.433 M.A.	.433 M.A.
P diss Max	66.05 M.W.	79.82 M.W.

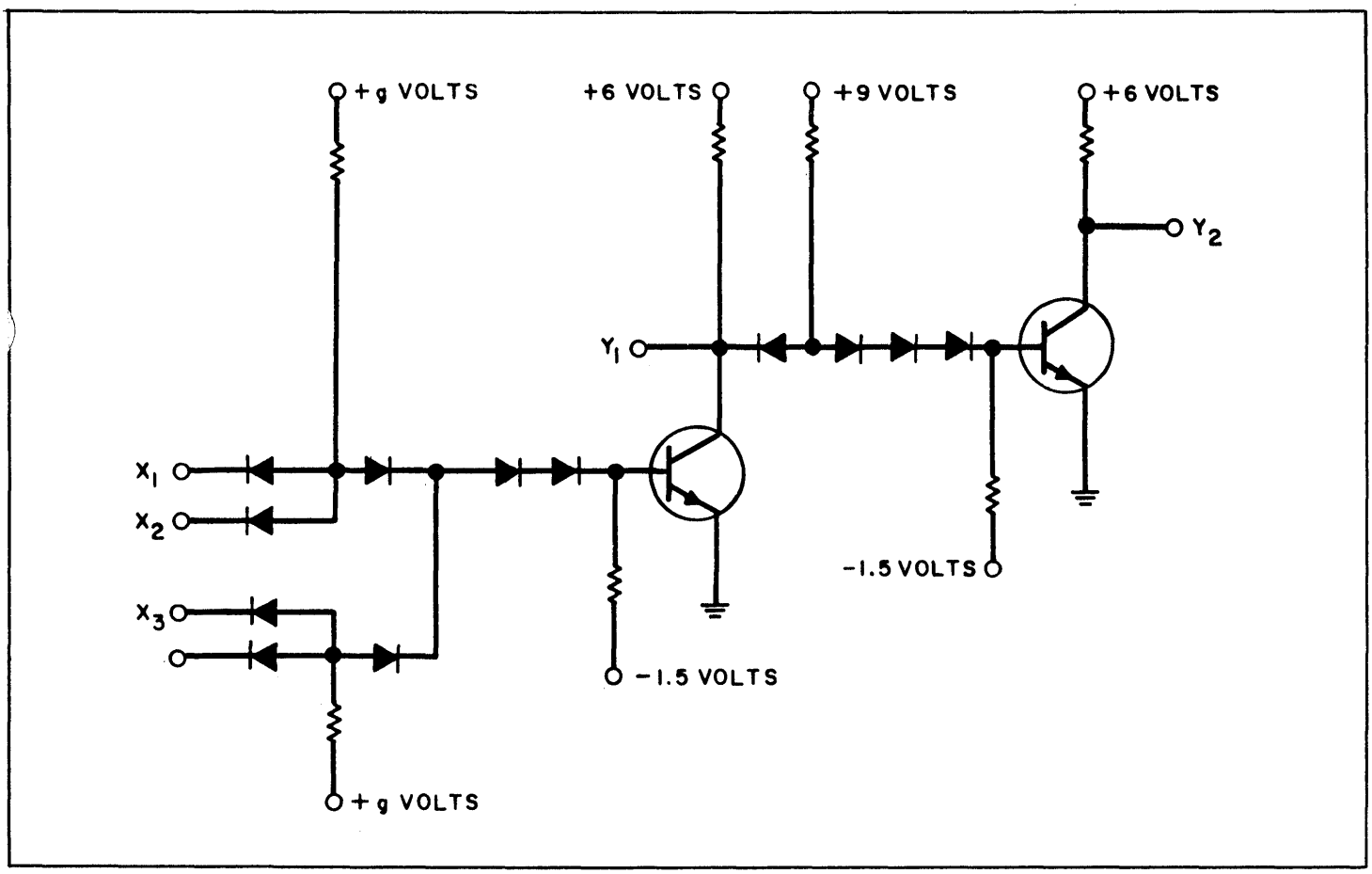


Figure 1-23. Med 2

Specifications

Logic Function: $Y_1 = (X_1 \bullet X_2) + (X_3 \bullet X_4)$
$Y_2 = (X_1 \bullet X_2) + (X_3 \bullet X_4)$

Input Loading: 1 load each

Maximum Circuit Delay: 025 microseconds

Output Drive: Y_1 = 5 loads

Y_2 = 10 loads

Maximum Power Requirement Per Pak:

	Logic 1	Logic 0
+9 -	8.5 M.A.	8.76
+6 -	1.04 M.A.	4.15
-1.5 -	.866 M.A.	.866
P diss Max (M.W.)	84.04 M.W.	104.3

Temperature Range: (0°C to 125°C)

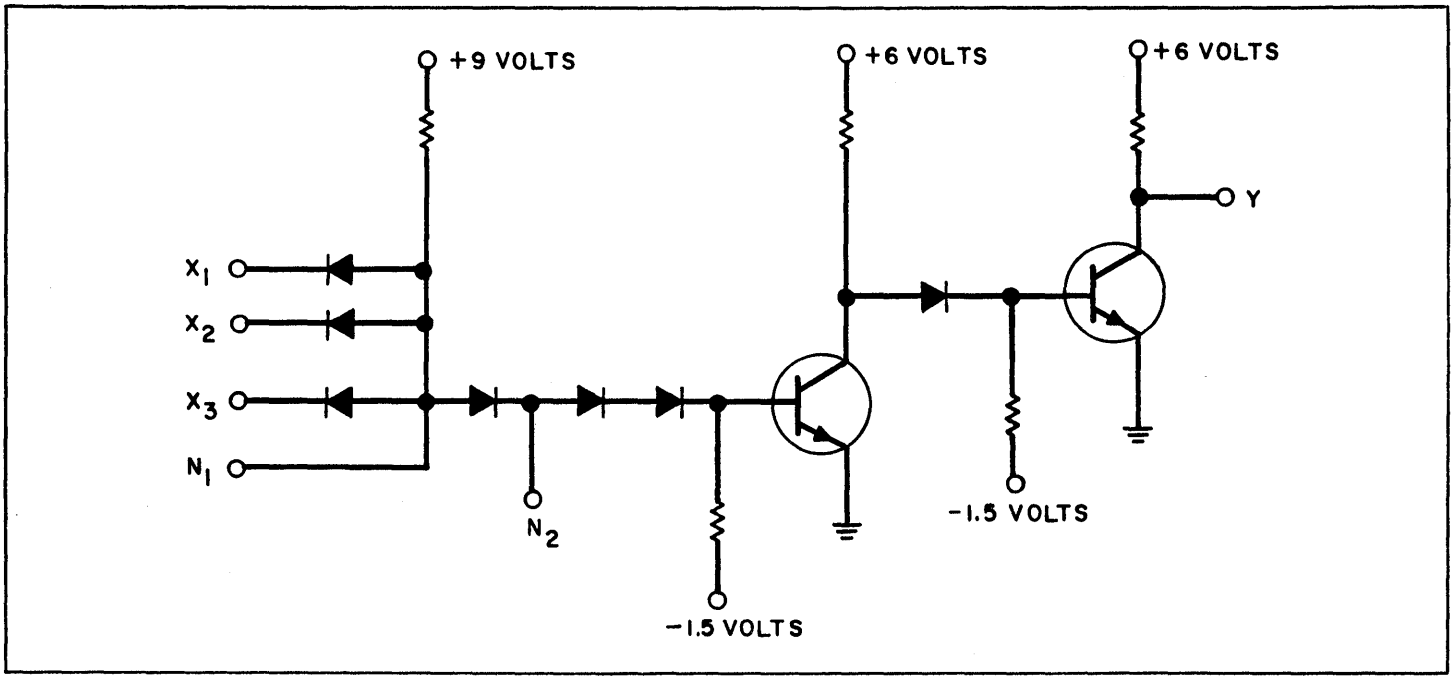


Figure 1-24. Med 9

Specifications

Logic Function: $Y = X_1 \bullet X_2 \bullet X_3 \bullet X_N$

Input Loading: 1 Load each

Output Drive: 25 Loads

Operating Temperature Range: (0° - 125° C)

Maximum Circuit Delay: .020 microseconds

N_1 is a nodal point where external diodes may be connected to enlarge the AND gate.

N_2 is a nodal point where external AND gates may be connected together where they are "OR"ed with diode gate (X_1, X_2 and X_3)

(Med 5, 6, 7, 8, 11)

Maximum Power Requirement Per Pack:

	Logic 1	Logic 0
+6	11.6 M.A.	15.6 M.A.
+9	2.2 M.A.	2.73 M.A.
-1.5	.50 M.A.	.43 M.A.
P diss Max	90.1 M.W.	118.8 M.W.

Figure 1-26 is the inverter driver. The nominal input loading is 3.4 loads or 9 milliamperes and the noise rejection is 2.4 volts. The output of the driver has a fan-out capability of 25. The transistor used in the driver is a gold doped expitaxial planar device capable of switching large currents with rapid speeds. The three slow recovery diodes in series with the base increase speed by providing a leakage path for the holes in the transistor during the turn off time of the driver.

The reliability of the integrated circuit package is not only achieved by worst case circuit design procedure but also by the establishment of rigid quality controls over

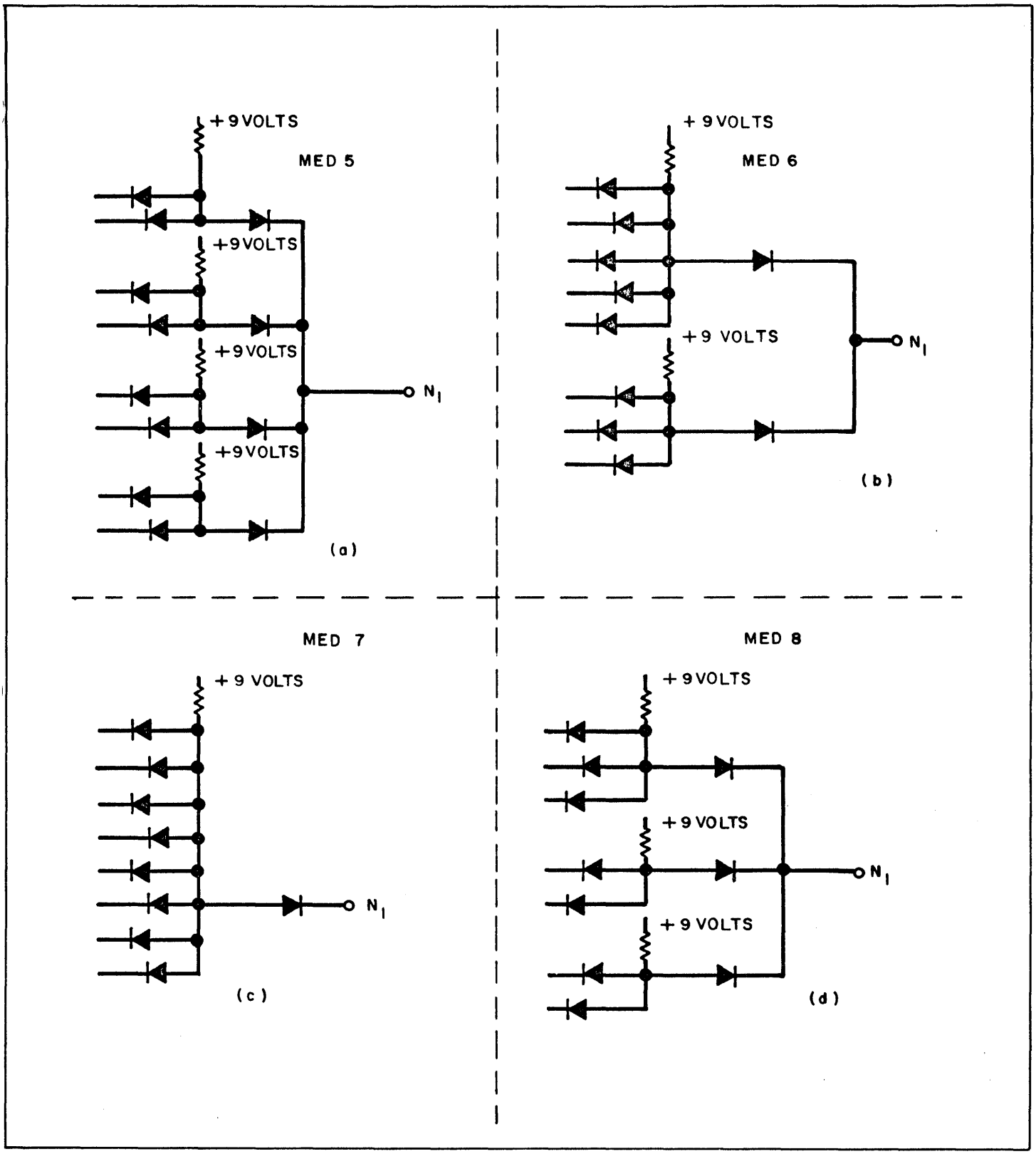


Figure 1-25. (Sheet 1 of 2)

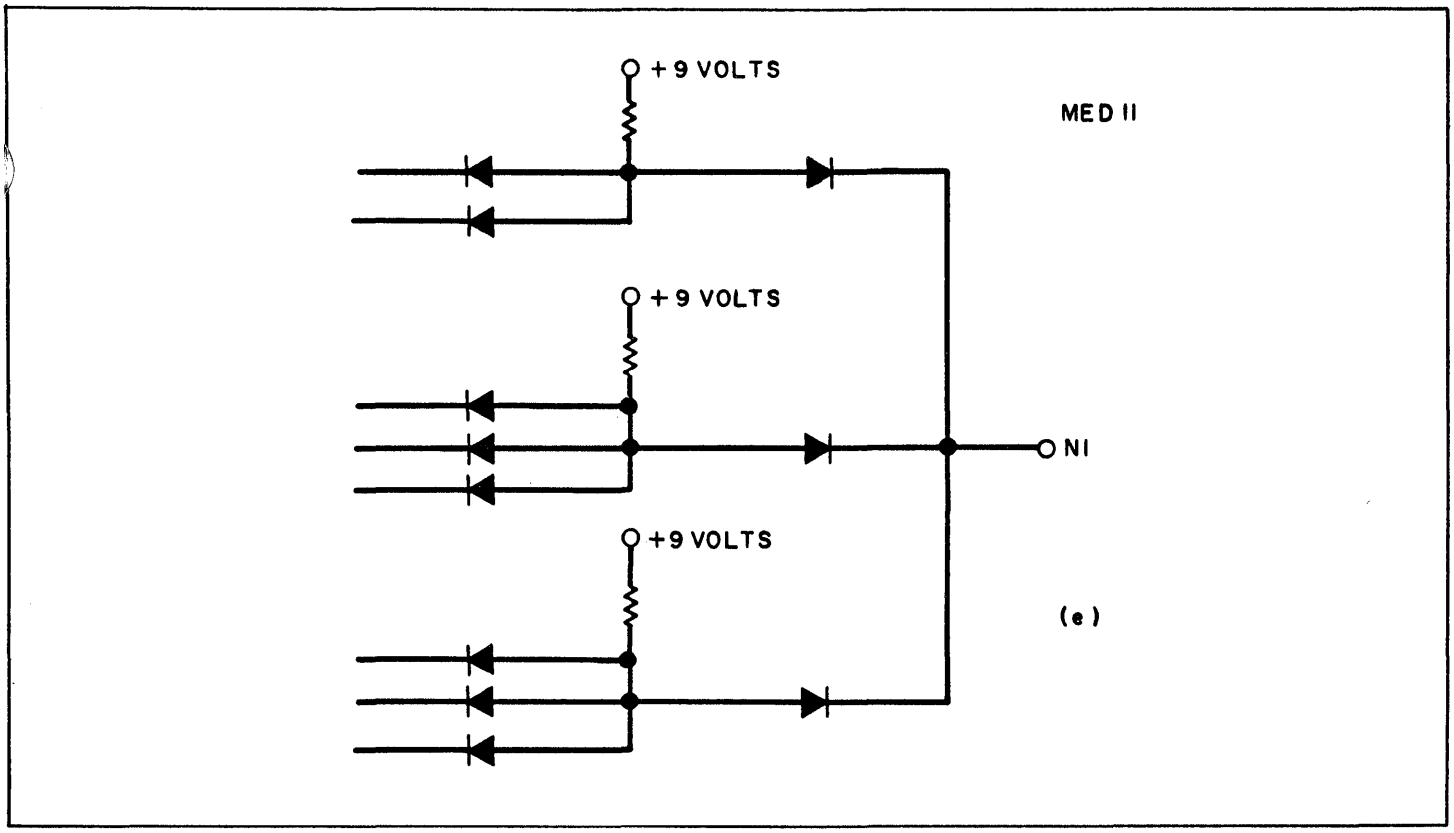


Figure 1-25. (Sheet 2 of 2)

Specifications

Logic Function: AND, OR

N_1 is a nodal point which may be connected to the corresponding point N_1 on Med 1 or Med 9.

Maximum Power Requirement Per Pak: (+9 volts)

	Med 5	Med 6	Med 7	Med 8
Logic 1	8.8 M.A.	4.4 M.A.	2.2 M.A.	6.6 M.A.
Logic 0	10.8 M.A.	5.4	2.7	8.1 M.A.
Logic 1	79 M.A.	39.6 M.W.	19.8 M.W.	59.3 M.W.
Logic 0	97 M.W.	48.6	24.3	73 M.W.

Operating Temperature Range: 0°C to 125°C

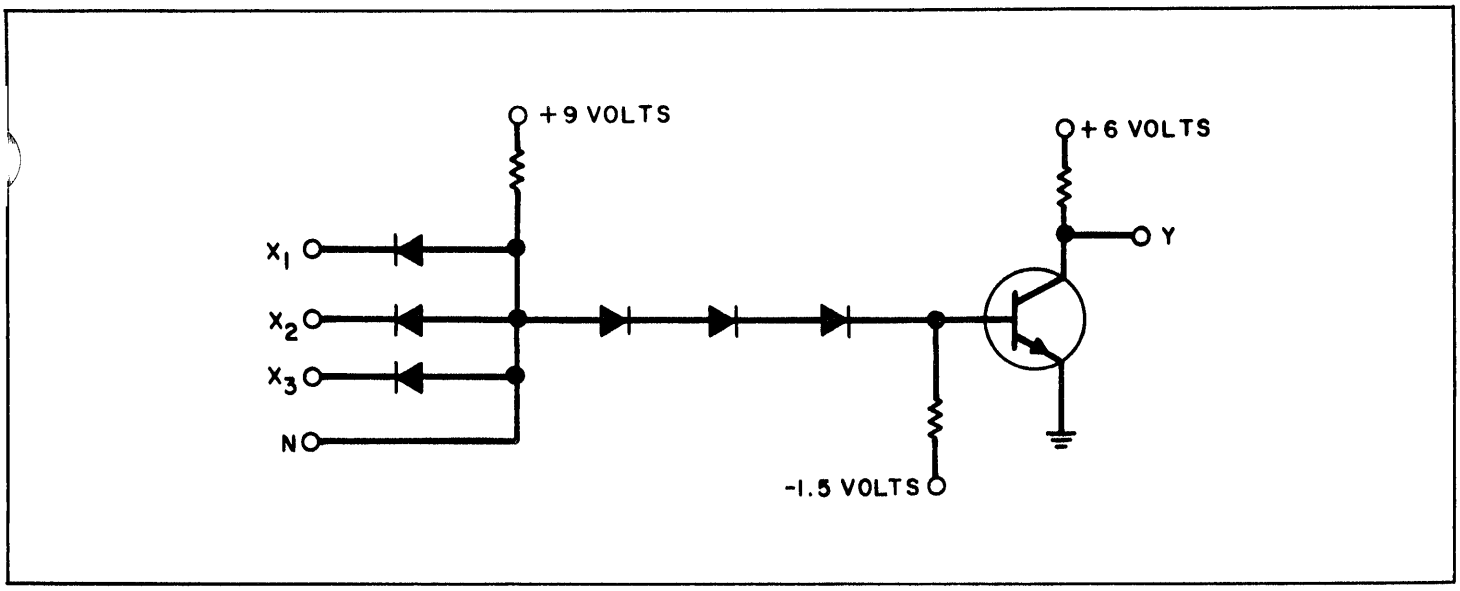


Figure 1-26. Med 3

Specifications

Input Loading: 3.4 Loads

Output Drive: 25 Loads

Logic Function: $Y = \overline{X_1 \bullet X_2 \bullet X_3}$

Maximum Circuit Delay: .020 microseconds

N is a nodal point where external diodes may be connected to expand the number of inputs to the AND gate.

Maximum Power Requirement Per Pak:

	Logic 0	Logic 1
+6	8 M.A.	6.44 M.A.
+9	9.11 M.A.	6.88 M.A.
-1.5	.08 M.A.	.37 M.A.
P diss Max.	82.11 M.W.	101.11 M.W.

Operating Temperature: (0° to 125°C)

the manufacture of each device. Final static and dynamic electrical tests combined with rigid mechanical tests insure a high degree of reliability for each package.

Worst case circuit design was achieved by assuming worst case tolerances on circuit parameters (±10%) and supply voltages in satisfying the requirement (Bubsat - icsat) (1) where: is the ratio of collector current to base current, ibsat, is the base current flowing into the base of a saturated transistor and icsat is the collector current flowing into the collector of a saturated transistor. Therefore, the design procedure consists of assuming worst case tolerances on circuit parameters and supply voltages, which in turn effect ibsat and icsat, and then satisfying equation (1) for a minimum fixed beta.

Consideration in the worst case design is also given to the effect of parameter and supply voltage variations on the turn-on and turn-off times of the circuit in satisfying the requirement that both the turn-on and turn-off time be less than 25 nanoseconds. Worst case design also considered the junction temperature of each molecular device. The junction temperature of a semiconductor device is given by: T_J = (ambient temp.) + (Thermal resistance) X (Power dissipated in device). The power dissipated in each device was calculated (worst case tolerances in parameters, supply voltages and on thermal resistivities) and, in turn the junction temperature was computed. With the transistors in the devices operating in the saturated and cut off modes with low collector to emitter saturation voltages, and low collector leakage current together with the backbiasing current furnished by the -1.5 volt supply stable operation is ensured with extremely low probability of circuit failure due to temperature and ageing.

The final static and dynamic electrical tests consist of: direct measurement of turn-on and turn-off times under worst case conditions of loading, peak inverse voltage

tests, noise rejection tests, life tests, threshold voltage tests and forward current conductance tests. The mechanical tests of these devices consist of: visual-mechanical (prior to encapsulation of the device), temperature cycling, Joy bomb, acceleration, moisture, vibration fatigue, salt atmosphere, head fatigue, shock and thermal tests of the can.

The turn-on and turn-off times of the drivers (Med 1, 2, 9) are reassured through a similar procedure. A .3 microsecond pulse with a 1 Mc repetition rate is applied to an input diode of the device. The rise and fall times of this pulse shall be 3 nanoseconds/volts or less. The output and input waveshapes are observed simultaneously on a dual channel oscilloscope (see Figure 1-27). The time measurements of Med 3 are made using the same procedure, only the output waveshape is inverted from the input. This procedure is repeated with and without loads. The turn-on and turn-off time must be below 20 nanoseconds for Med 3, 20 nanoseconds for Med 9, and 25 nanoseconds for Med 1 and Med 2. The noise rejection and threshold voltage tests consist of applying a 0 to 1.5 volt and 0 to 2.7 volt rectangular pulse to the packs (Med 1, 2, 3, 9) and noting the output voltage for each pulse shape. The output must not respond to the 0 to 1.5 volt pulse. These tests should be made under full fan-out loading conditions. The operating life test consists of connecting the packs as shown in Figure without the scope. A ring oscillator connection may be used in place of this test. The forward current conductance tests are made by measuring the forward conduction current through the input diodes under static voltage conditions. For packs (Med 1, 2, 9, 3) these currents must fall between certain minimum and maximum

WESTINGHOUSE ELECTRIC CORPORATION

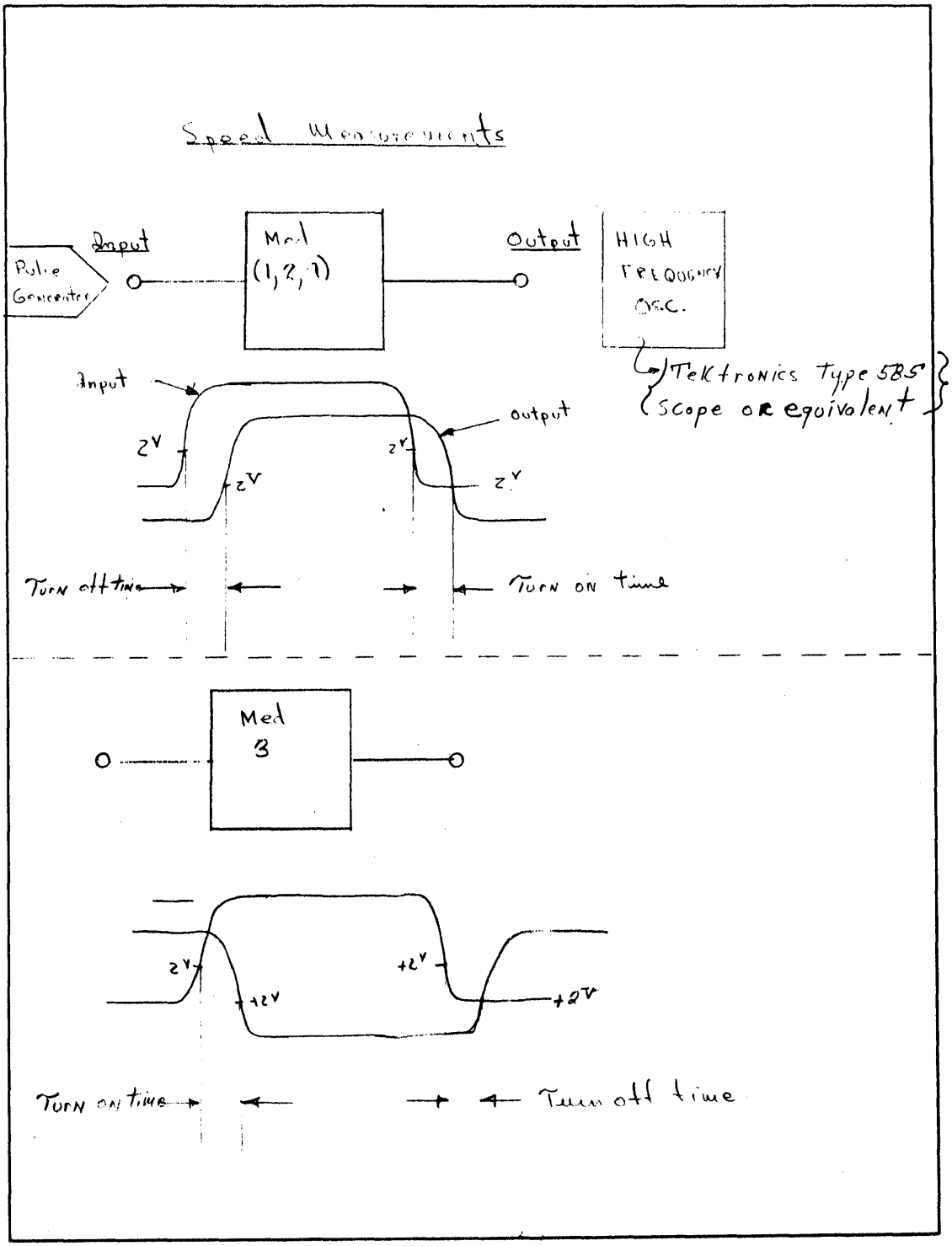


Figure 1-27 Speed Measurements

values. The reverse current tests are made by applying a reverse bias voltage to the input diodes and measuring the reverse current of the diodes. These currents should have a certain maximum value. The peak inverse voltage tests consist of backbiasing collector to emitter junctions with a nominal voltage below breakdown and noting the results.

The above electrical tests coupled with rigid mechanical tests and worst case design procedures ensure a reliable device.

1.12.3 Other Standard Paks

The Standard Timer is a circuit package that is used in the central control, sequencer, arithmetic and input/output units of the computer. The timer is a delay device used to synchronize timing signals throughout the computer. It consists of: pulse shaper, delay line driver, delay line with taps, delay line threshold circuits and drivers. (See Figure 1-28).

The pulse shaper and delay line driver shape the incoming pulse to a specified pulse width, provide a length guard of 90 percent of the length of the line with an error detector, and drive the delay line. The delay line is a lumped constant L-C, tapped, low impedance line with high pulse fidelity characteristics. The threshold circuits present a high input impedance for the delay line, while at the same time providing additional pulse shaping for the pulse in the line. The drivers provide the necessary fan-out capabilities to distribute the signals throughout the system.

The entire time acts as a delay device. The output pulses are being delayed at various times from the input pulse by selecting the desired tap. Fidelity of the pulse is maintained by shaping the pulse before it enters the delay line and shaping

WESTINGHOUSE ELECTRIC CORPORATION

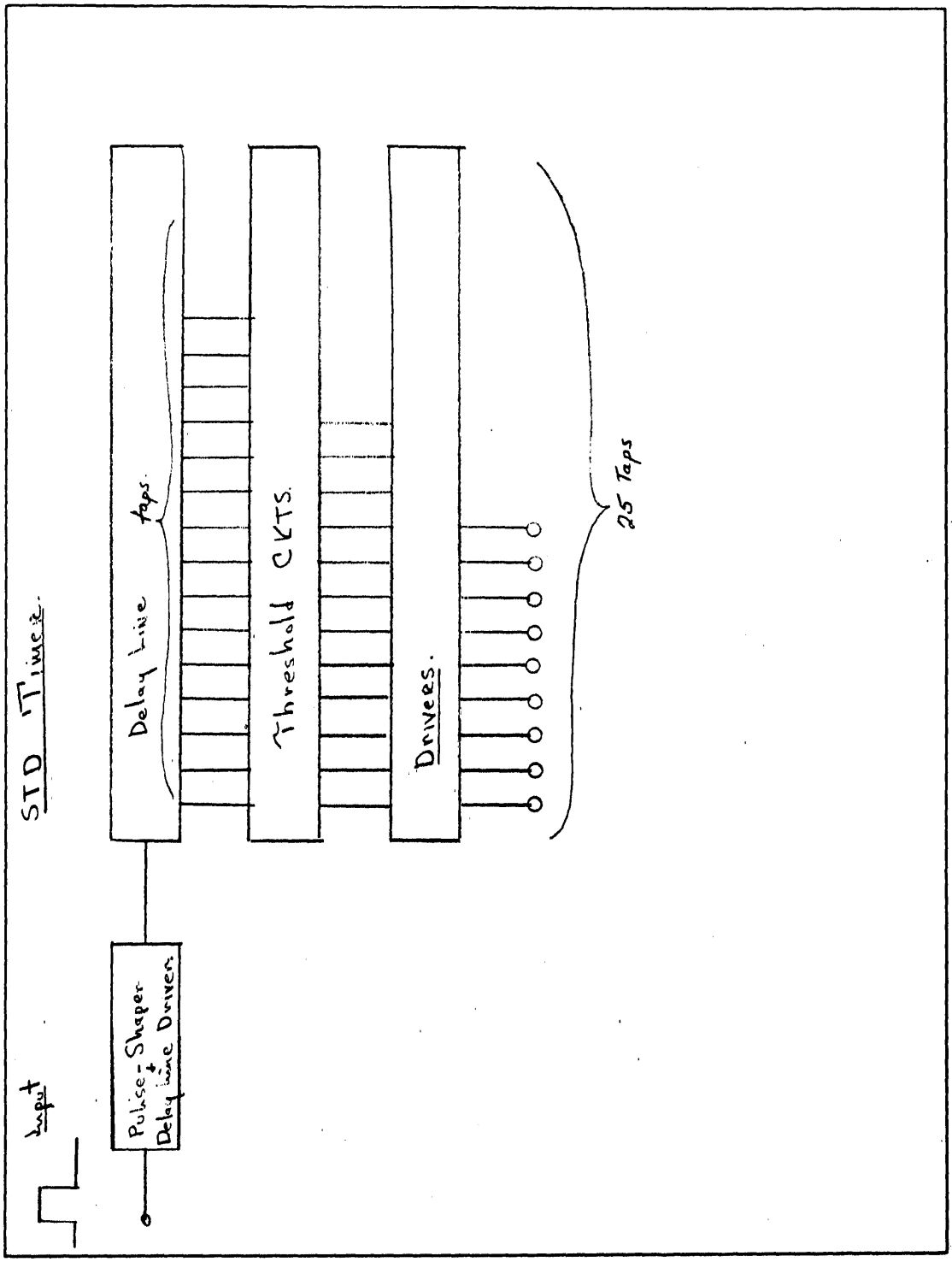


Figure 1-28 STD Timer

it at the output tape of the delay line. A length guard is provided such that two pulses cannot be in the line during 90% of the delay time of the line.

The Standard Timer will be packages using: molecular packs, standard components (resistors, diodes, transistors) and a lumped constant delay line. The components will be mounted on printed circuit boards and the circuits will in turn connect to the tapped delay line.

The pulse shaper and delay line driver shape the incoming pulse to a pulse width of .10 microseconds ±.005 microseconds. (See Figure 1-29). The incoming pulse (1) sets a (0) in the delay line and will remain until No. 1 flip-flop is set into the zero state from tap 4. This provides the initial shaping of the input pulse into the delay line. Med #3 provides the necessary drive for the delay line. Flip-flops 2, 3, and 4 provide a length guard of 90 percent or if the delay line is 1 microsecond and the repetition rate on the input pulse is 2 megacycles, the neon indicator will indicate an error. Two pulses will not appear in the line during 90 percent of the delay time of the line. This is accomplished by setting the flip-flop 1 AND gate of the input to (0) and setting flip-flop 2 AND gate to (1) through flip-flop 4. This effectively sets flip-flop 2 so it will set when another input pulse is received in less than 90% of the delay time of the line, and if flip-flop 2 is set to (1) the neon indicator will light. This lock-up circuit guards against the possibility of an initial error being locked up in a circulating system without an error being indicated.

The delay line for the timer is a 1.20 microsecond line with taps every .02 microseconds. The impedance of the line is low (100 ohms) to minimize the effects of capacitive loading (wire) on the line. Temperature stability of the line is

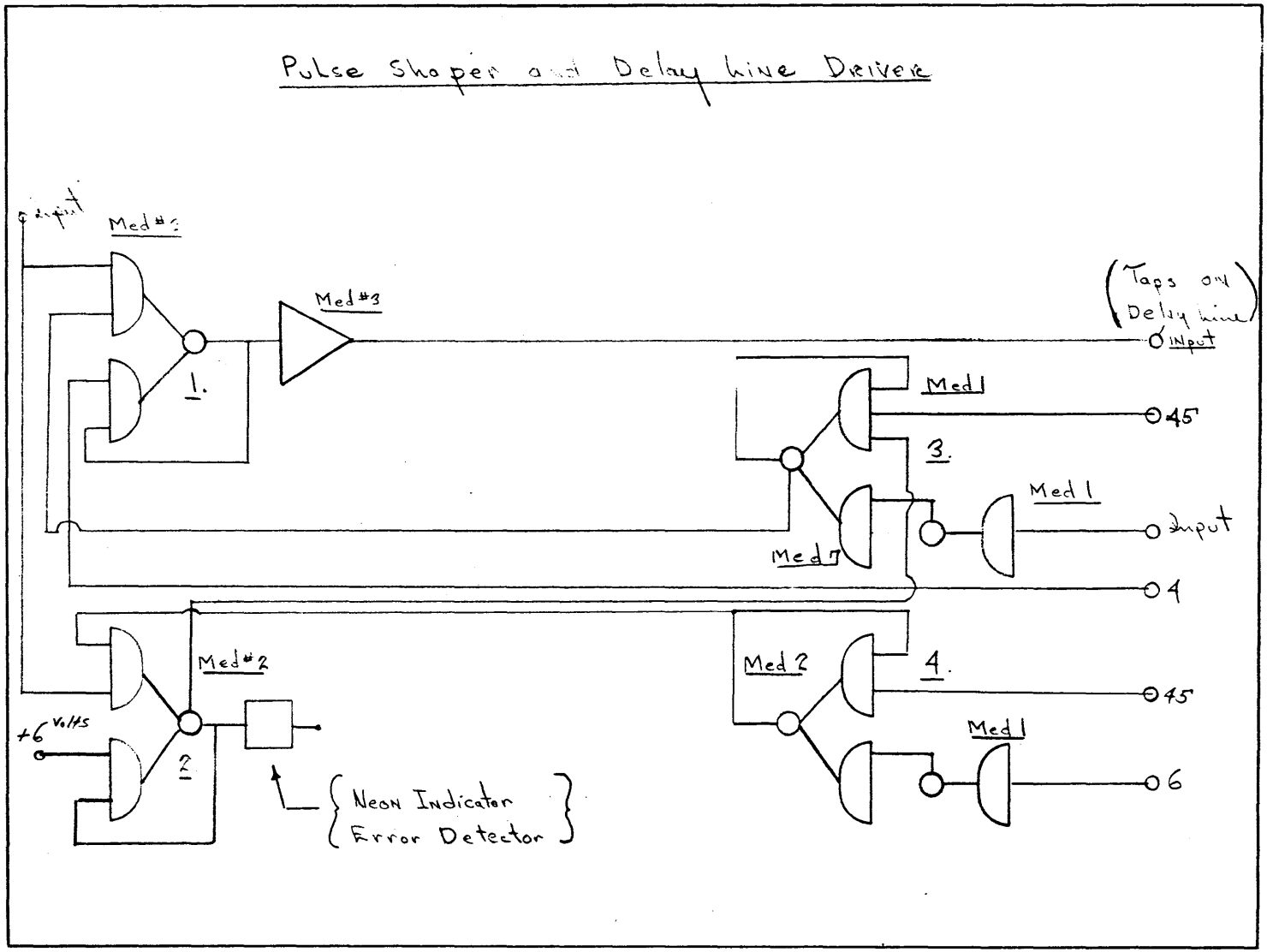


Figure 1-29 Pulse Shaper and Delay Line Driver

maintained at a .05 nanoseconds/microseconds/° C about 25° C. The line maintains a high fidelity for a pulse in the line by minimizing pulse distortion (defined as the largest peak amplitude of all spurious responses either in the positive or negative direction relative to the pulse amplitude.

The threshold circuit on the delay line taps presents a high impedance to the line, while at the same time performing a NOT/OR function on the pulses in the line (see Figure 1-30). A lumped constant delay line has the characteristics of decreasing the pulse width of an incoming pulse as it travels down the line. The pulse shaper and threshold circuits maintain the fidelity of the pulse by first increasing the pulse width at the input to the line to .120 microseconds, using the threshold gates to limit the pulse width at the initial taps and connecting the bases of the transistors together at the end of the line, if it is necessary. If either base of the circuit is high, the common emitter point will be high, backbiasing diode D_1 and saturating transistor T_3, dropping the output to 0. If both bases are low, diode D_1 will conduct and T_3 will cut off, raising the output to a logic (1). A variable capacitance (C_1) and an inductance are provided in order to adjust the delay of the line within ±.005 microseconds. Appropriate drive for these signals is provided after the threshold circuit to distribute both the true function and the complement function throughout the system.

1.12.4 Memory Circuits

Detailed descriptions of the individual memory circuits are presented in the following subsections.

WESTINGHOUSE ELECTRIC CORPORATION

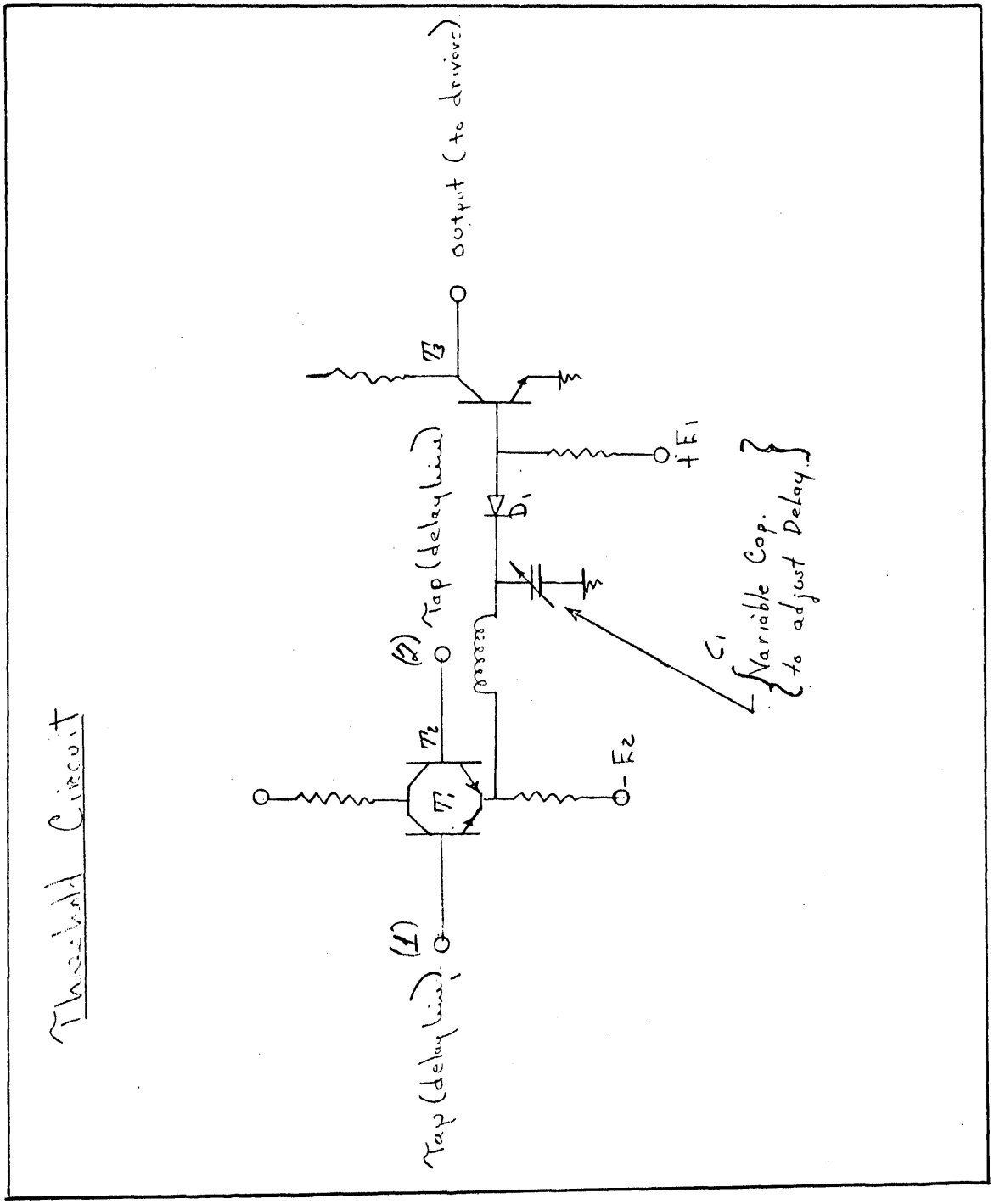


Figure 1-30 Threshold Circuit

1.12.5 Load Sharing Matrix Transformer & Diode Encode Matrix

The load sharing matrix* is a group of transformers (Figure 1-31). Their primary drive windings are connected in a particular logical pattern which, when driven by the drivers, performs the functions of paralleling the drivers, switching the outputs, matching impedance, and providing bipolar output currents from unipolar drivers. As a result of the particular pattern in which the primaries are connected and the combination in which the drivers are turned on, the net sum of all the drive winding ampere turns is zero on all output transformers except one, in which the drive ampere turns are additive. The polarity of a given output transformer can be reversed by energizing the complementary combination of drivers. The input power to the selected output transformer is effectively the total combined output power of the selected drivers, thus high energy drive pulses can be conveniently generated from drivers that are lightly stressed.

A diode encode matrix (Figure 1-32) is used to select the various combinations of drivers to be energized. The pattern of the selection diode matrix is similar to the pattern of transformer interconnections (Figure 1-31). In comparing the two figures it can be seen that a diode replaces each winding that has a dot at the top; in windings with dots at the bottom, no diode connection is used.

1.12.6 LSM Driver

This driver (Figure 1-33) is a three-stage feedback amplifier in which the reference resistance can be varied in order to change the output current. The circuit has

*A Class of Optimal Noiseless Load Sharing Matrix Switches. R. T. Chein, IBM Journal, p. 415, October 1960.

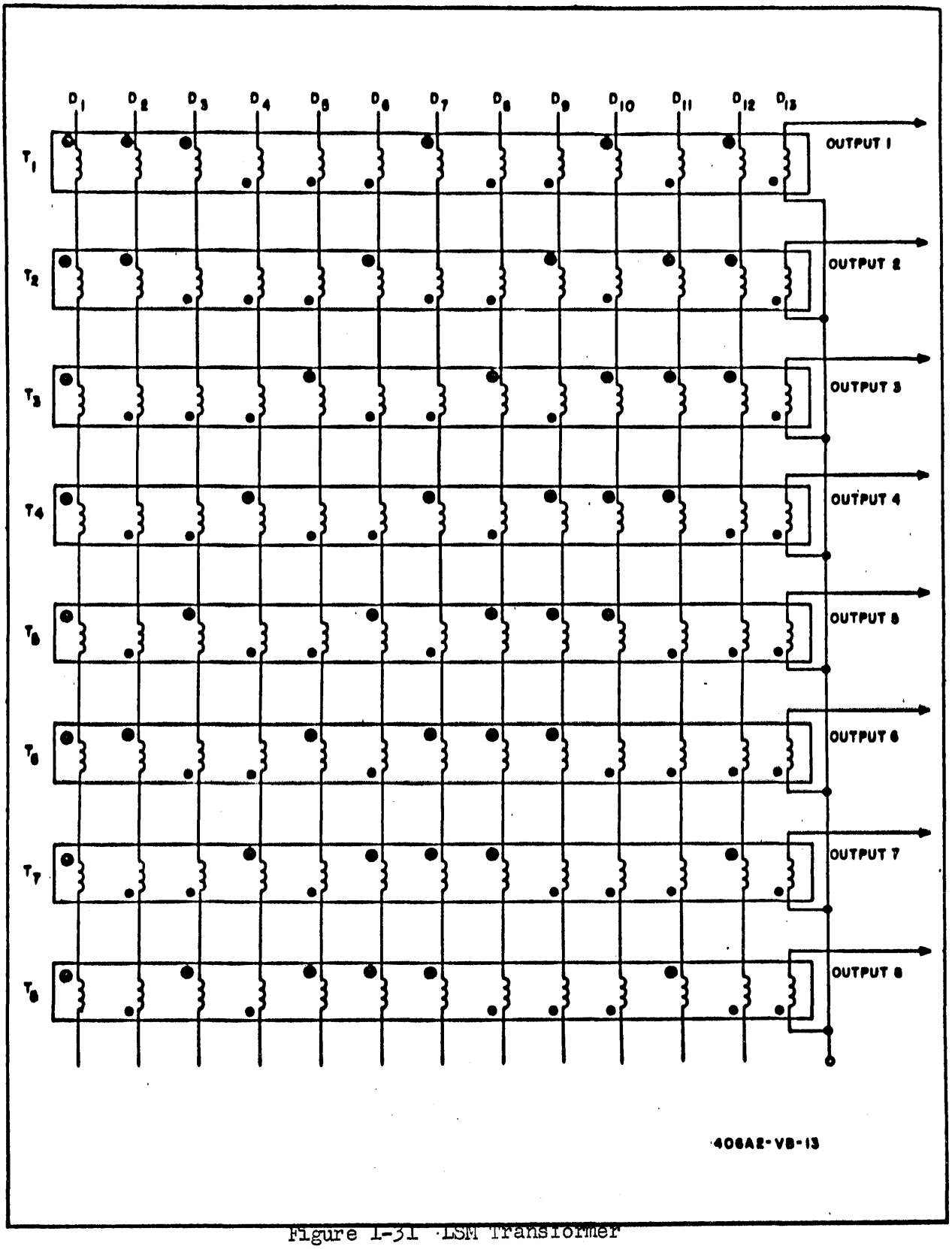


Figure 1-31 LSM Transformer

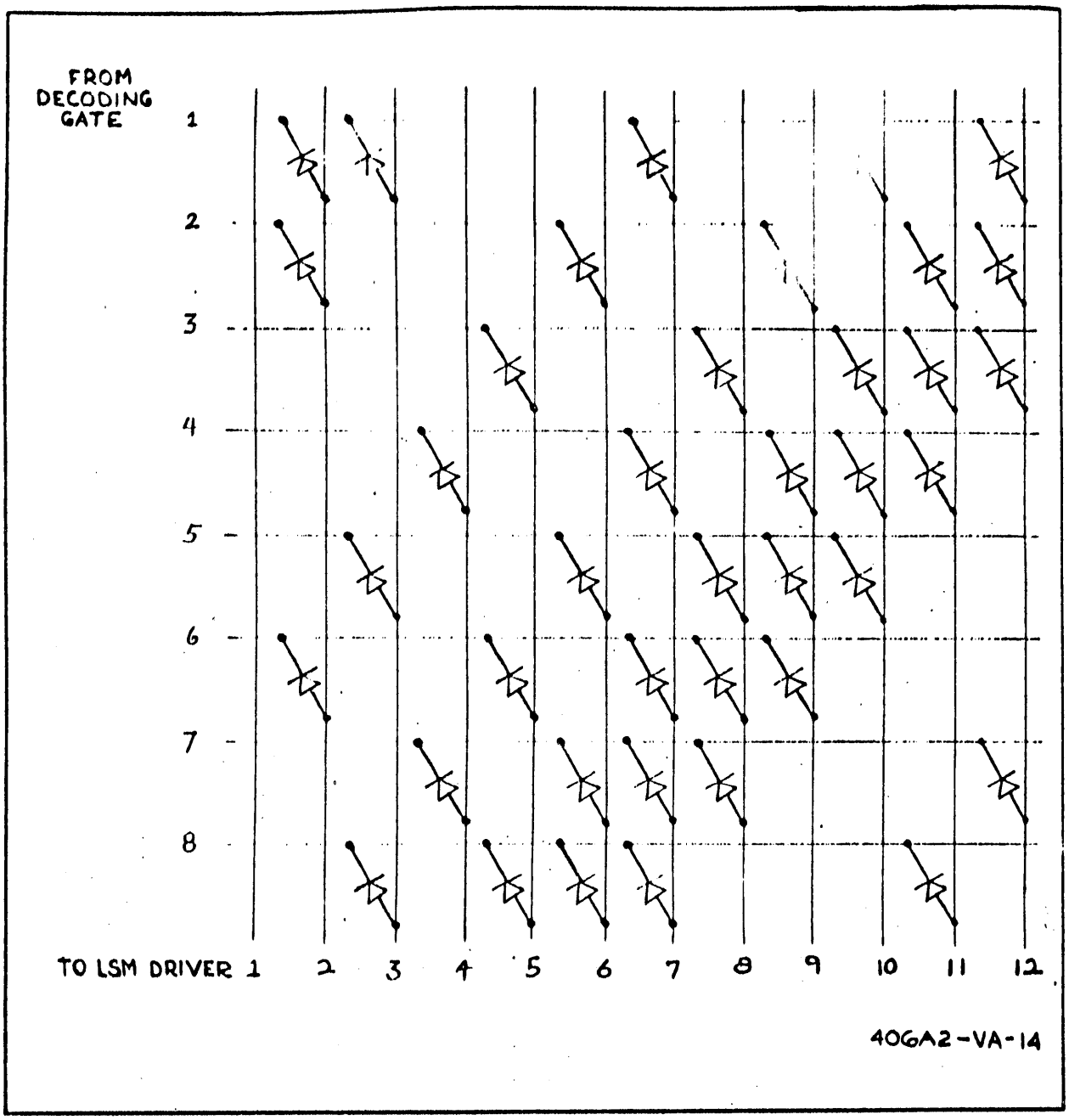


Figure 1-32 Diode Encode Matrix

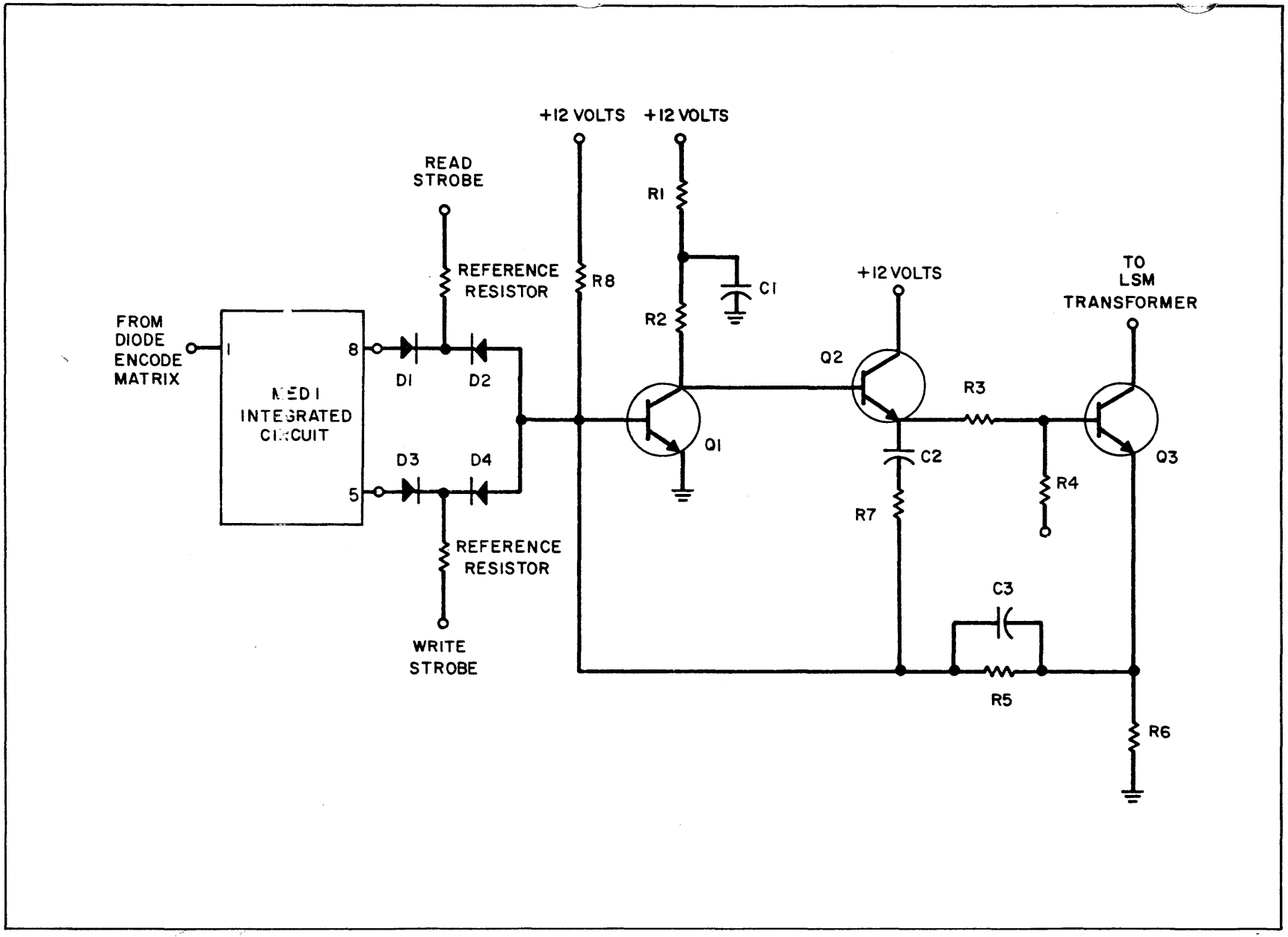


Figure 1-33. LSM Driver

low power dissipation in the off condition and is characterized by very fast turn-on and turn-off times.

The combination of capacitor C_5 and resistor R_{20} is a high frequency attenuation network, necessary for dynamic stability. Capacitor C_4, together with R_{19} shapes the overall frequency response of the amplifier to control the rise and fall times. The amplifier is preceded by a stroke gate, an inverter, two AND gates and a diode OR gate for decoding the read and write addresses.

1.12.7 Sense Amplifier

The sense amplifier consists of two differential stages followed by a transformer-coupled thresholding stage and a strobed output stage (Figure 1-34). The circuit is designed to provide a unipolar output with an input signal of 50 mv of either polarity.

The 150 ohm resistors terminate the sense line and references it to ground. Constant dc emitter currents are supplied to Q_1 and Q_2 by the emitter constant current source Q_3. The stage limits at any input greater than 130 millivolts. A resistor is placed across the collectors to provide both gain stability and gain control.

The second stage is also limiting and is much like the first one. It operates at a higher quiescent current level and is returned to a higher collector supply to provide a greater signal swing. The emitters are tied together through two back-to-back 6.8 microfarad capacitors both for gain and for compensating any unbalance in transistor base-to-emitter voltage characteristics.

The third stage rectifies the bipolar signals and establishes the threshold voltage level at the transformer center tap. Since an output of the thresholding stage can be

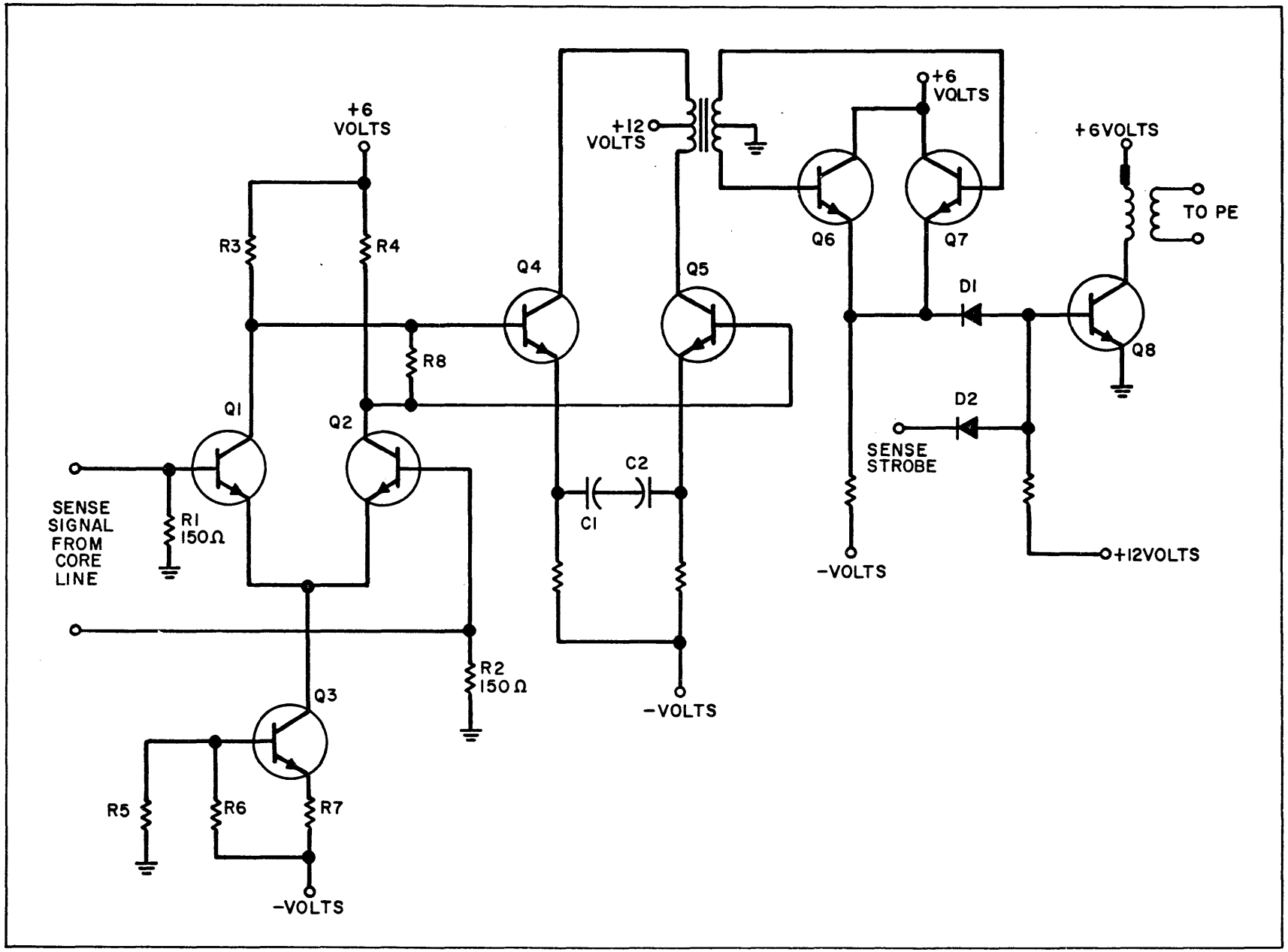


Figure 1-34. Sense Amplifier

the result of noise as well as from a core output, the output stage is strobed when a readout is expected.

1.12.8 Digit Driver

This circuit (Figure 1-35) consists essentially of a stroke gate first stage which is transformer-coupled to the power stage.

Transistor Q_1 acts as a switch which connects a voltage source across the digit winding and its terminating resistance. The digit line is split in two and each side is terminated in its characteristic impedance. The MED 3 integrated circuit is the switch which provides a large base current to Q_1 to turn it on and off quickly. R_3 and C_1 control the rise and fall time of the output current.

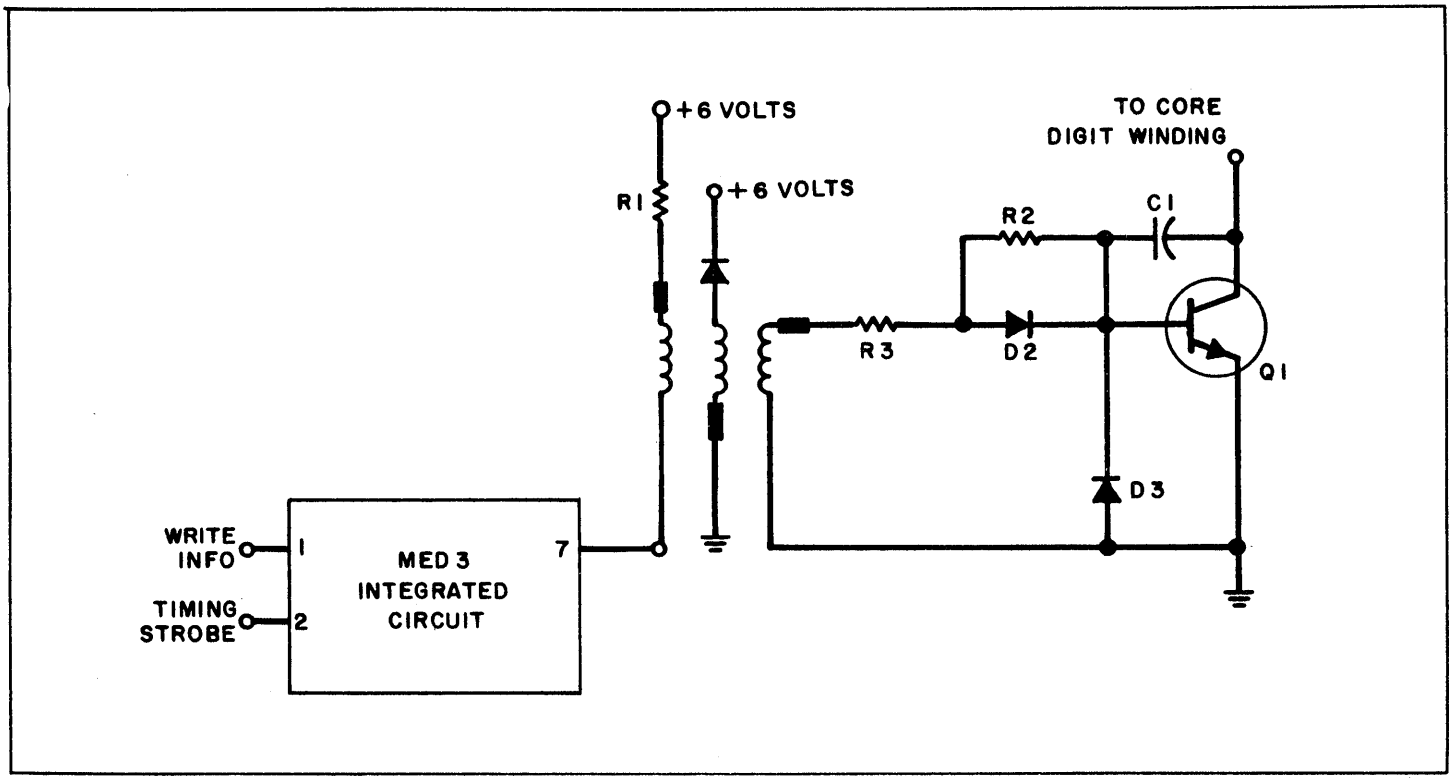


Figure 1-35. Digit Driver

1.12.9 Saturated Switch

This three-stage saturating circuit (Figure 1-36) is similar in operation to that of the digit driver. Transistor Q_2 acts as a switch which, in conjunction with its collector load, provides a large base current to Q_3 to turn it on and off quickly. This circuit uses diode coupling rather than transformer coupling because of the duty factor considerations.

1.12.10 Central Memory

The Central Memory contains all program instructions and the operands for the sequential programs. It consists of 16,384 words of 40 bits each. A 1.0 microsecond read-write cycle time is obtained by using a 30 x 18 mil ferrite core in a linear select, partial switching mode of operation.

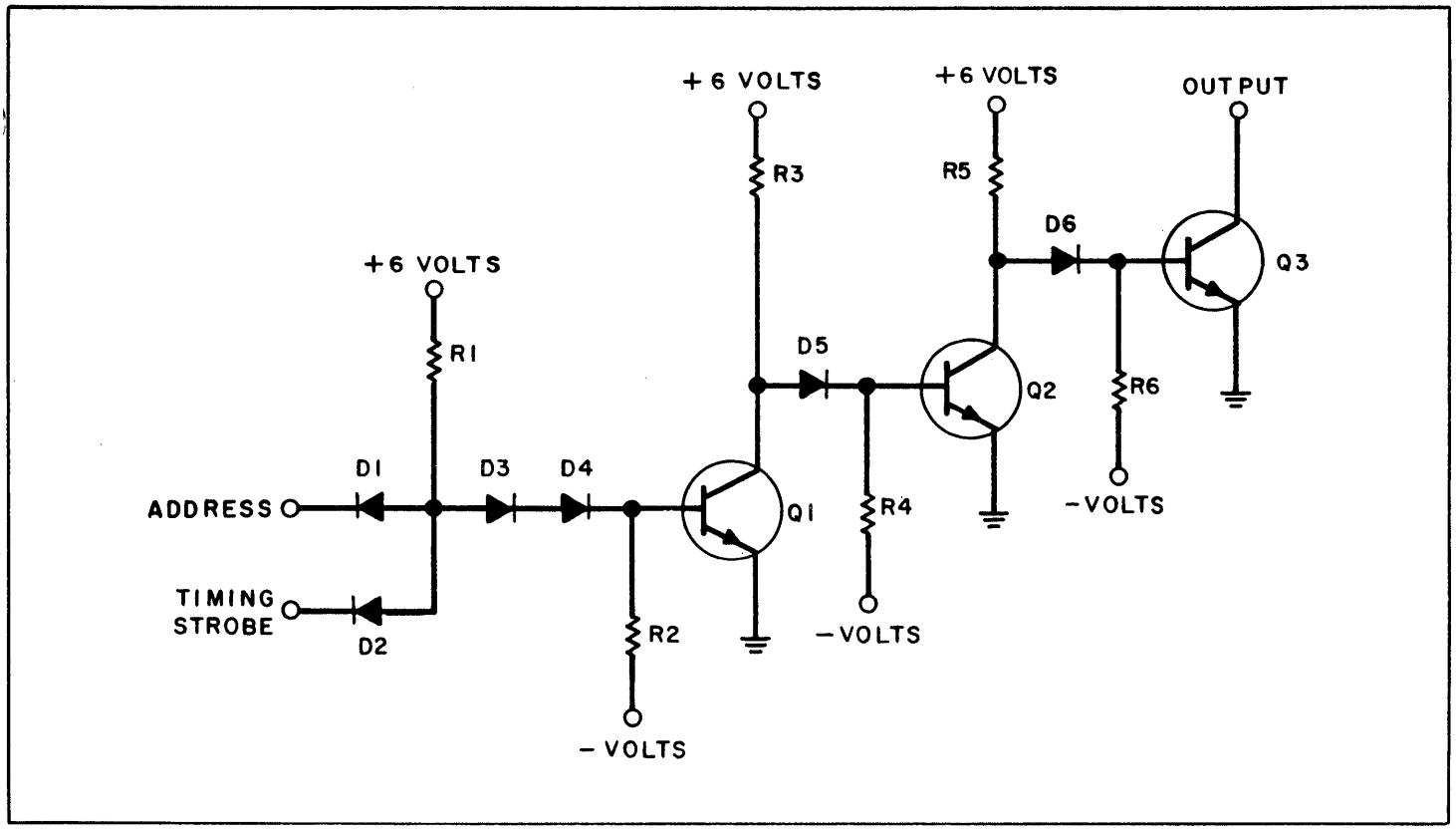


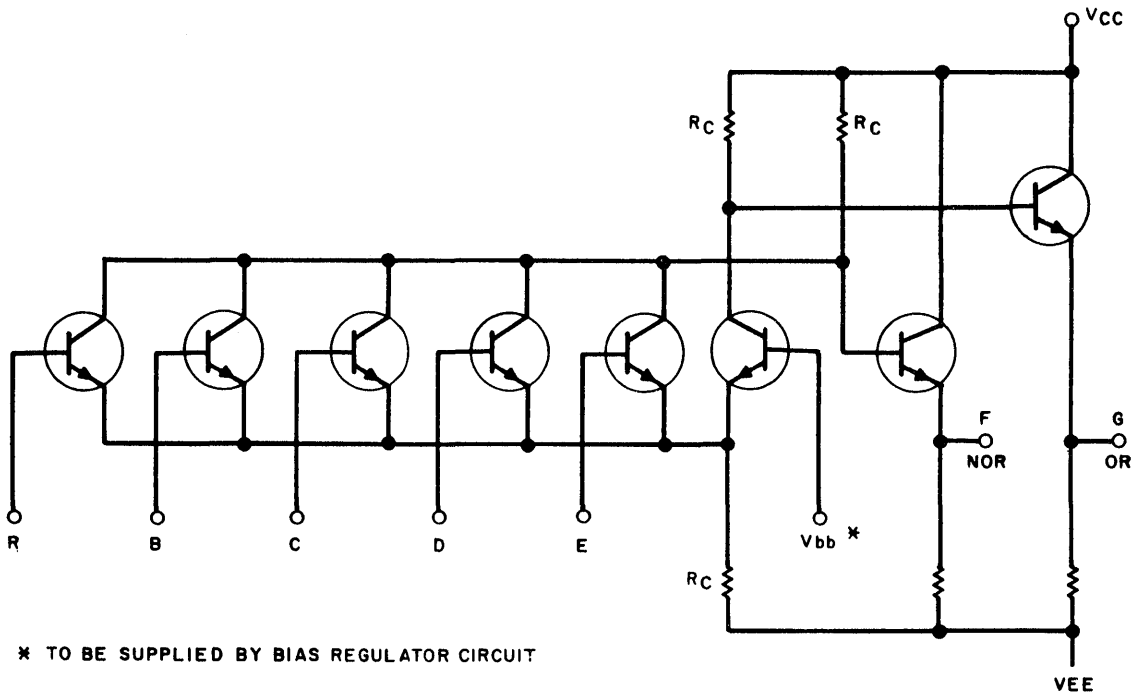
Figure 1-36. Saturated Switch

The memory is actually built as a 4096 word, 160 bits/word unit. Thus four words are read during one read portion of a cycle with the desired word gated to the data output lines. In addition to the savings in access circuitry obtained by using this type of memory organization, the rate of data flow between the central memory and L-buffer is substantially increased. The L-buffer operated at a 10 megacycle repetition rate and contains 32 words of 20 bits each. Thus during one read cycle of the central memory the L-buffer could load in all 160 bits readout of the central memory.

The general operation of the central memory and its circuitry is identical to that of the PE memory module.

1.12.11 High Speed Circuitry

In areas where extremely high speed is desirable, current mode logic switching techniques are used. High speed operation, excellent d.c. stability, and good noise immunity are among the advantages offered by current-mode switches. Basically, the current-mode switch provides nonsaturated transistor operation. This mode of operation eliminates one of the most speed limiting characteristics associated with the conventional saturated switch, namely-storage time. The elimination of storage time, as in current-mode logic permits significant improvement in switching speeds. The basic current-mode logic gate is shown in Figure 1-37. It consists essentially of an NPN current-mode switch, with emitter-followers coupling the signal from the collectors of the gate to the output. The current switch, in turn, contains six transistors, five of which have their bases connected to the respective inputs. The base of the sixth transistor is connected to a fixed bias such that this transistor is conducting when no signal is supplied to any of the inputs. The circuit has two



emitter follower coupled outputs; one taken from the collector of the fixed bias transistor and the other from the common junction of the collectors of the five input transistors. The emitter follower serves a two-fold purpose in this gate. First, it serves as a D.C. translator so that the D.C. level of the outputs of the gate is compatible with the level requirements at the various inputs. Secondly, since the emitter follower has a low output impedance it provides a large fan out capability.

If a positive signal representing a logical "1" is applied to any of the inputs, the collector of the fixed bias transistor goes positive and the output from the commonly

connected collector output goes negative. The two outputs, therefore, perform the logical operations "OR" and "NOR", respectively. If, on the other hand, a negative signal had been used to represent a logical "1", the basic gate would perform the logical operations "AND" and "NAND".

The fixed bias, V_B, for the gate is supplied by a regulator circuit which as the name implies, compensates for drift in D.C. output level due to variations in temperature and power supply voltage.

A complete set of nonsaturate current mode logic circuits including the five input gate discussed above are available in integrated form, packaged in TO-5 cans, or can be constructed with discrete components.

Several characteristics particularly worthy of attention are as follows:

1) Six nanosecond typical propagation delay at average gate power dissipation of only 35 mW,
2) High noise immunity - 50% of logic swing,
3) High fan-in - up to 25,
4) High fan-out - up to 25,
5) Constant loading of power supply, thereby minimizing "internally generated" noise.

The fan-in, fan-out capabilities of this type of logic circuitry is inherently large because of the low impedance level of the common emitter mode and because of the high input impedance and low output impedance characteristics. The logical capability is high because it performs the "OR/NOR" functions simultaneously in addition to the large fan-in, fan-out capability. These circuits are extremely fast; i.e., the

propagation delay is very small, since most of the logical decisions are performed at the low impedance level of the common emitter mode, the signal paths are essentially through emitter followers and grounded base stages, and since the transistors in these circuits never become saturated.

This kind of circuitry will have good noise immunity because the input impedance of these circuits is high and only a small amount of power transmitted from the driving circuit to receiving gates. Because the input is low, cross-talk between adjacent interconnections is minimized. Also, noise signals induced on power supply lines are minimized since the current demand of these circuits is constant and independent of state.

Finally, it is well to mention disadvantages where special precaution is in order. A noise immunity of 50% of the logic swing is a noise signal in the millivolt region, since the logic swing is only about one volt. Another disadvantage of current-mode logic is the need for many transistors; this can make such circuits economically unattractive.

The molecular stroke gate device is the basic circuit for the SOLOMON system. The quantity of gates used in the system dictates that the individual gates must have extremely high reliability under the worst operating conditions. The saturated logic circuitry of the stroke gate configuration offers the advantages of: decreased power dissipation, circuit simplicity, high noise rejection, operation under wider variations in component parameters and supply voltages, as compared to other forms of logic circuitry. These characteristics provide the extremely high reliability demanded by the system.

Other standard circuit pacs are required in relatively low quantity in various portions (i.e., memory drivers and amplifiers, etc.) of the system. Each of these circuits is designed and tested to ensure the same high reliability expected from the molecular circuits.

In areas where extremely high speed is desirable, current-mode logic switching techniques are used. High speed operation, excellent stability, and good noise immunity are among the advantages offered by current-mode logic. Basically, the current-mode logic provides nonsaturated transistor operation. These high speed elements are molecularized stroke elements. The circuits are used primarily in the PE Sequencer design, in small quantities, to ensure successful operation in this critical area of the system.

1.13 PACKAGING

The various parts of SOLOMON-II are housed in modular racks which are approximately 1 foot thick, 4 feet wide, and 6 feet high. Each of the functional groups occupy one or more modular units. The functional groups are purposely separated into modular racks so that, should damage occur to a rack, only that section need be taken out of the system. The overall system; 1024 PE's, NCU, GPU, Arithmetic Unit, and I/O System, is made up of 17 modular rack units, plus the consoles. Sixteen of the racks are distributed in a pentagonal configuration as shown in figure 1-38. Four sides of the pentagon are made up of bays of 4-rack modules, each hinged to the main frame so as to open for maintenance in book fashion. Space for interconnecting wiring is provided within the main frame.

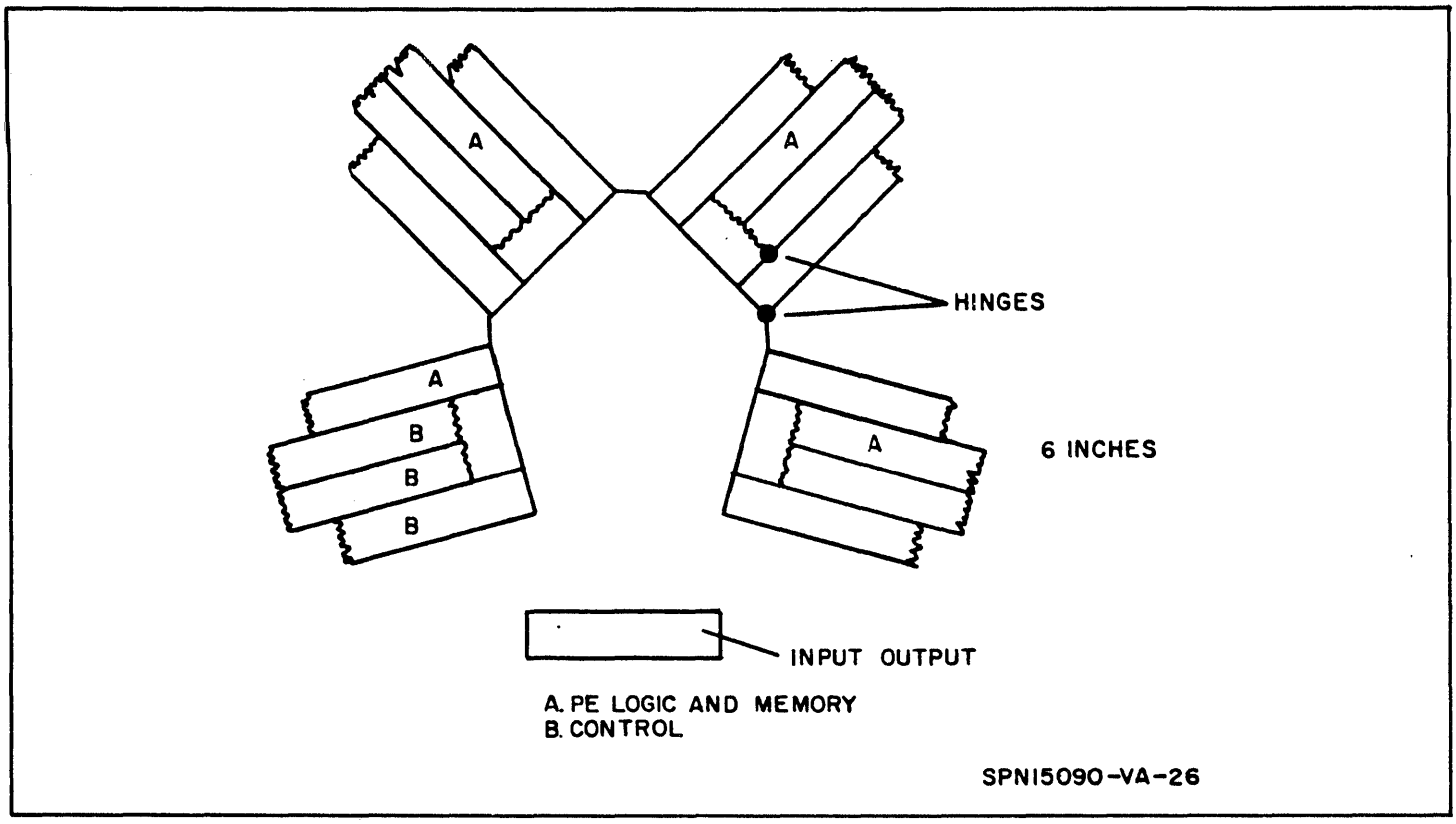


Figure 1-38. SOLOMON II Packaging Configuration

As computer speeds have increased, phase shifts and delays introduced by the interconnecting wiring have approached the same order of magnitude as the system parameters. In an ultra high-speed system, such as SOLOMON II, where an extended structural configuration would result in abnormally long delays, a "cross" or "star" configuration is chosen.

Thirteen of the rack modules are given over to the PE Network consisting of 4 256-PE Memory Banks plus logic and controls for the entire PE Network. Each of the four memory banks requires two rack modules, and the logic occupies five more racks. The PE Control and Sequencer Units share a single rack as do the Central Control and Arithmetic Units. The total Central Memory, including both

PE Program and General Purpose, is housed in a single rack. The last rack is given over to five Input-Output Channels. The consoles are separate desk-type units.

2. COOLING AND POWER SUPPLIES

Each of the racks contains its own cooling system and power supply to permit the racks to be moved without recourse to elaborate cabling and ducting. Forced air convection cooling is used with a single blower in each rack to move the air in two non-sealed recirculating loops.

Air is blown vertically past the printed wiring cards and returned through the power cabling duct. A separate horizontal loop at higher temperatures is provided for the power equipment. A finned-tube liquid air heat exchanger provides the primary sink for the rack.

The power dissipation of the system excluding the peripheral input-output devices is 16 kw. All power supplies have their high stress elements derated by very large factors and have built-in fault isolation and redundant paths.

3. CABLING

Inter-rack wiring within a bay of four is carried in tape cable which permits rotation of a rack on its hinges. Wiring between bays is carried in conventional cable. Rack motion is permitted by positioning the cable in approximate concentricity with the line of hinge centers so that the cable is twisted axially as the rack is rotated.