SOLOMON PROJECT TECHNICAL MEMORANDUM NO. 25

SOLOMON II Physical Characteristics

4 November 1963

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

This technical memorandum is published solely for information and use by project personnel and is not intended for external distribution. The material contained herein is PROPRIETARY.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION OF SOLOMON II

Paragraph Page

1.1 Processing Elements Network 1-1

1.1.1 General 1-1

1.1.2 PE Memories 1-9

1.2 The Network Sequencer 1-11

1.3 Control Units 1-13

1.3.1 Network Control Unit (NCU) 1-14

1.3.2 General Purpose Control Unit (GPU) 1-15

1.4 Arithmetic Unit 1-17

1.4.1 General Description 1-17

1.4.2 The Adder 1-17

1.4.3 Arithmetic Instruction 1-17

1.5 Input-Output (I/O) Subsystem 1-19

1.5.1 Channels 1 and 2 - Magnetic Tape Transports and Line Printers 1-22

1.5.2 Channel 30 - Disk File Modules 1-25

1.5.3 Channel 31 - Typewriter 1-29

1.5.4 Channel 32 - L-Buffer Transfer Channel 1-30

1.6 SOLOMON Circuits 1-33

1.7 Packaging 1-34

2. COOLING AND POWER SUPPLIES

Cooling and Power Supplies 2-1

3. CABLING

Cabling 3-1

LIST OF ILLUSTRATIONS

Figure		Page
1-1	General Block Diagram of SOLOMON II	1-2
1-2	Block Diagram of SOLOMON II With Optional GP Subsystem	1-3
1-3	Processing Element Block Diagram	1-4
1-4	Identification of PE's Within the Network	1-6
1-5	Geometric Control Registers	1-8
1-6	Block Diagram SOLOMON PE Memory	1-10
1-7	Network Sequencer	1-12
1-8	Network Control Unit	1-15
1-9	General Purpose Control Unit	1-16
1-10	Arithmetic Unit	1-18
1-11	Arithmetic Unit	1-19
1-12	Input-Output Data Channels	1-20
1-13	Magnetic Tape Transport/Line Printer Channel	1-24
1-14	Disk File Channel	1-25
1-15	Typewriter Channel Control	1-29
1-16	L-Buffer Transfer Channel	1-31
1-17	SOLOMON II Packaging Configuration	1-35

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.

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

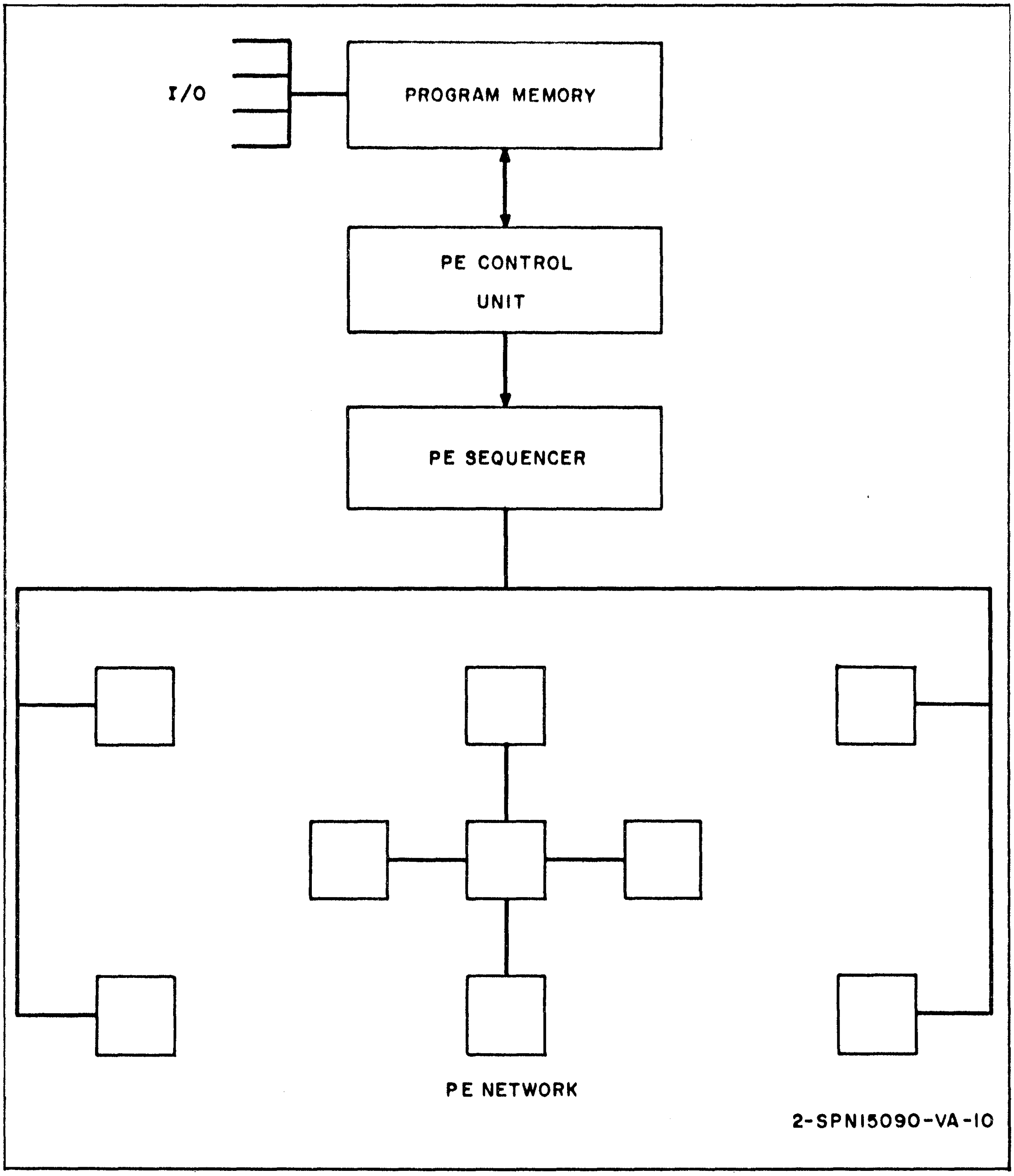

Figure 1-1. General Block Diagram of SOLOMON II

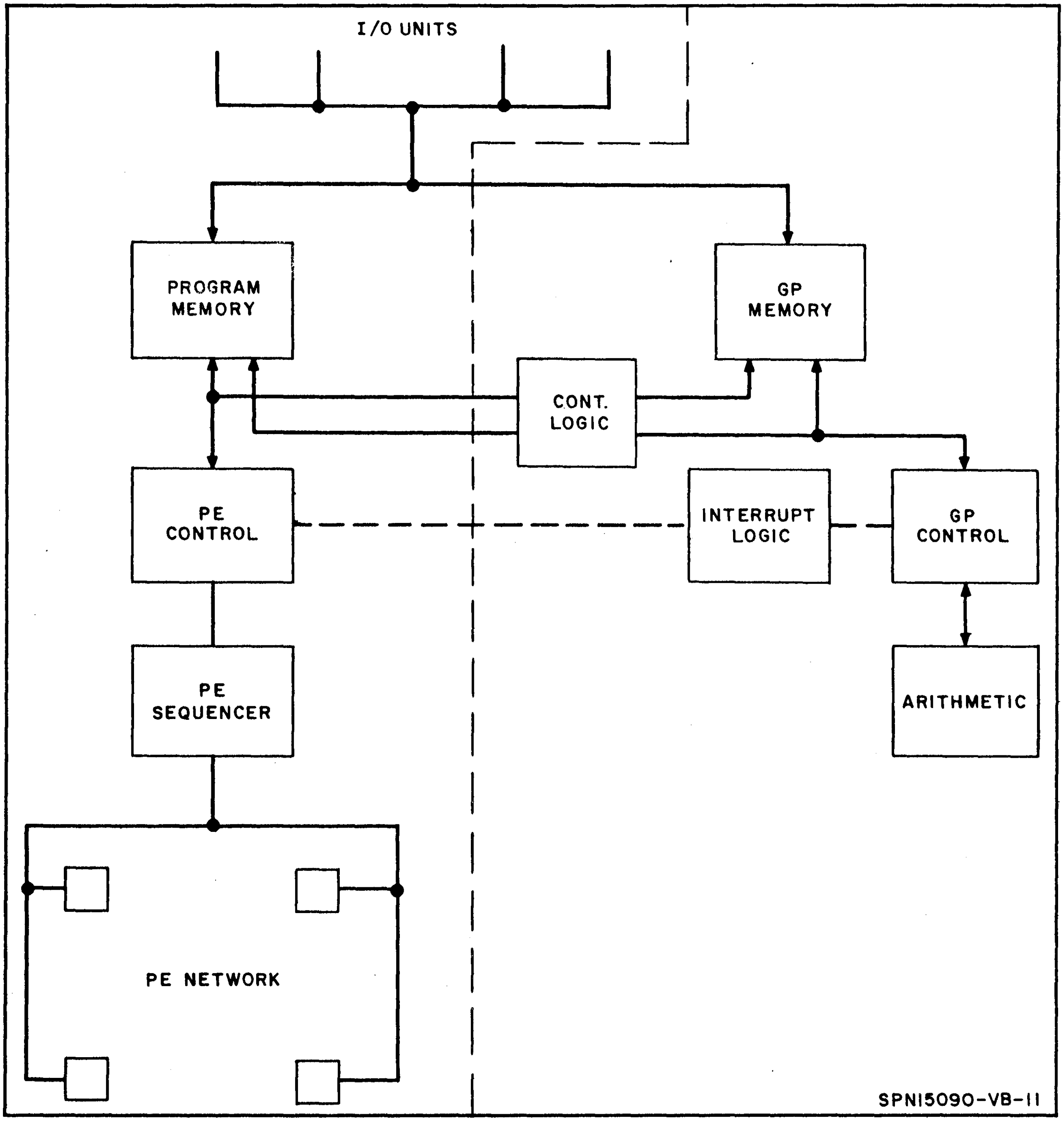

Figure 1-2. Block Diagram of SOLOMON II With Optional GP Subsystem

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.

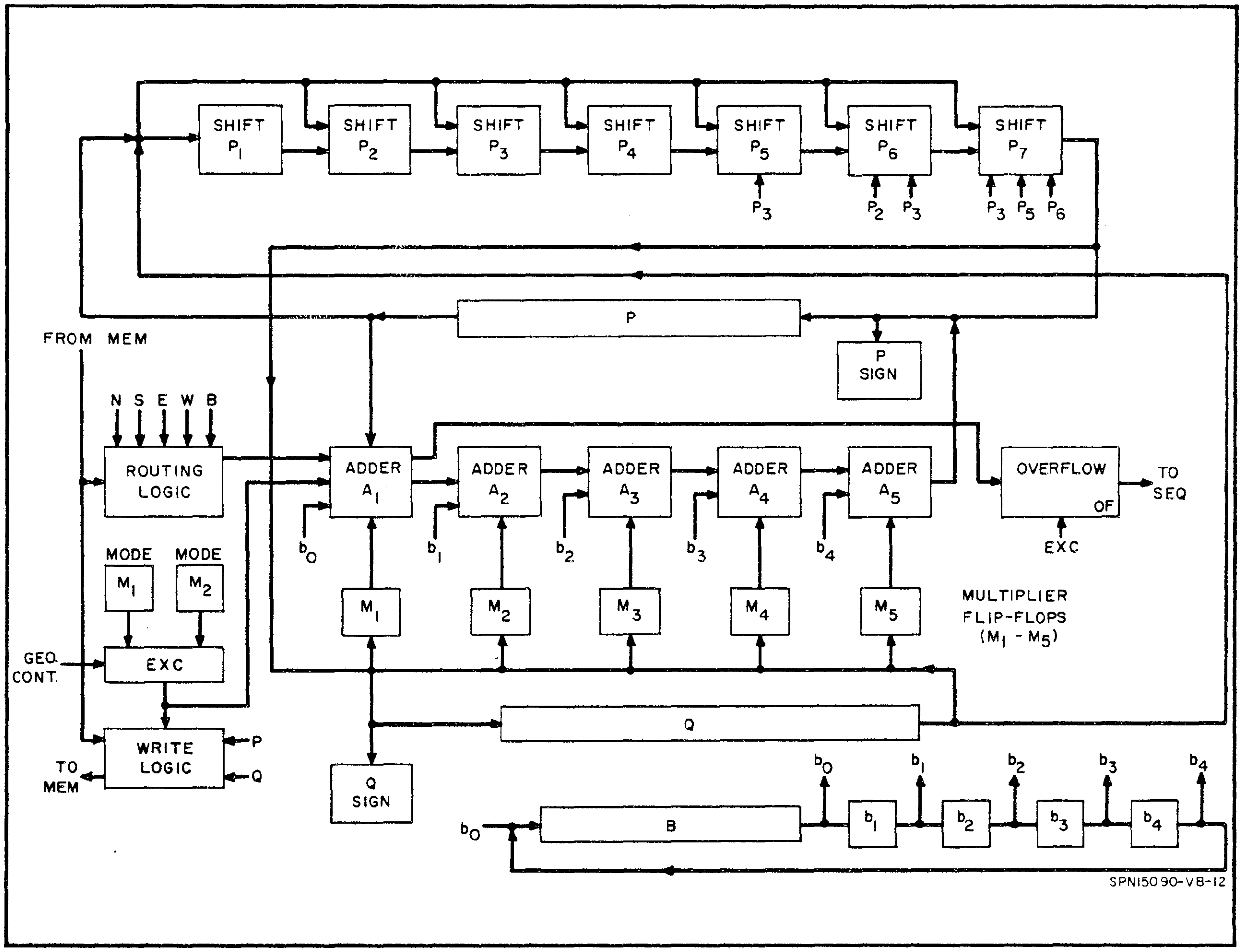

Figure 1-3. Processing Element Block Diagram

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 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 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

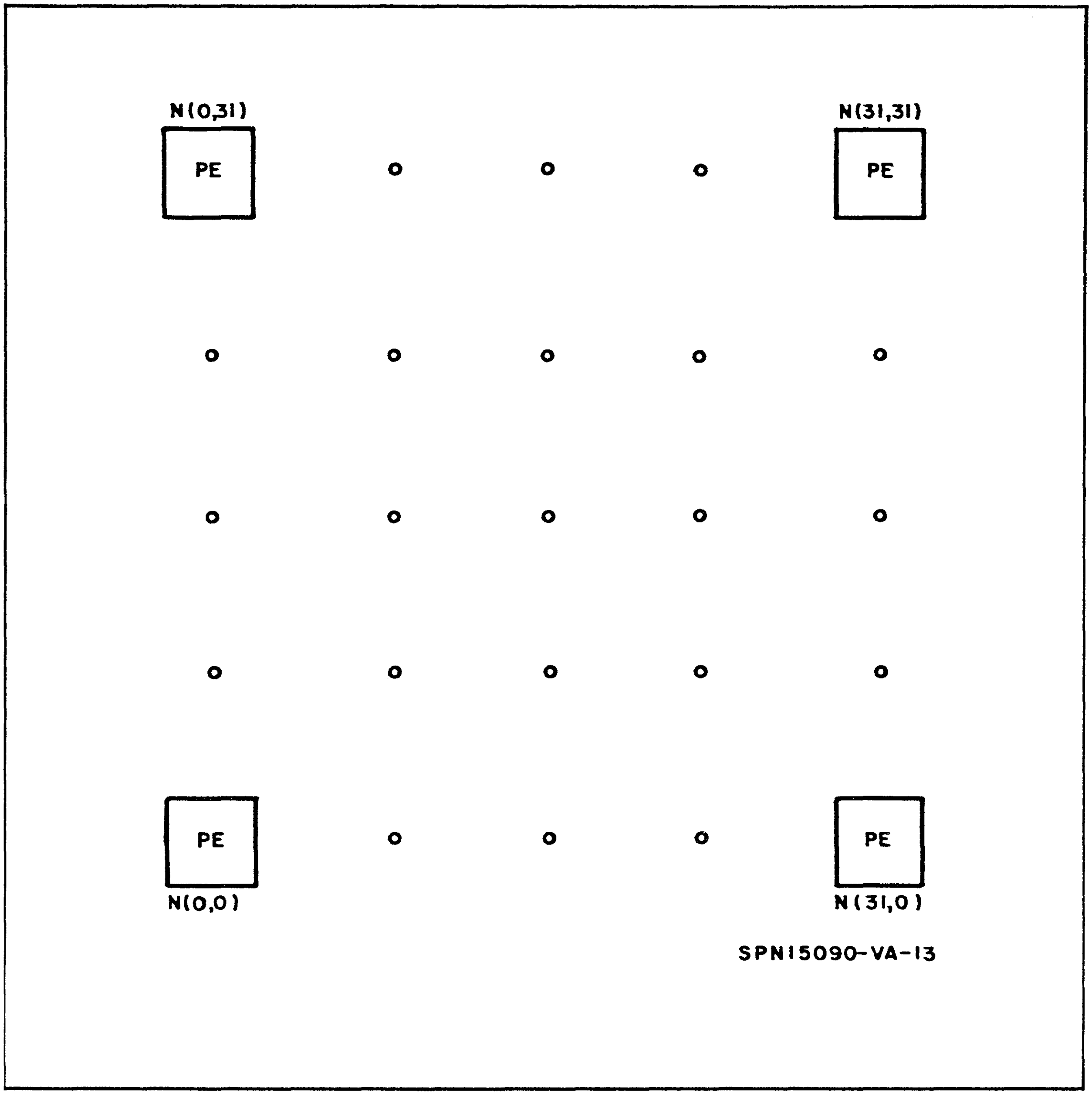

Figure 1-4. Identification of PE's Within the Network

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).

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.

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

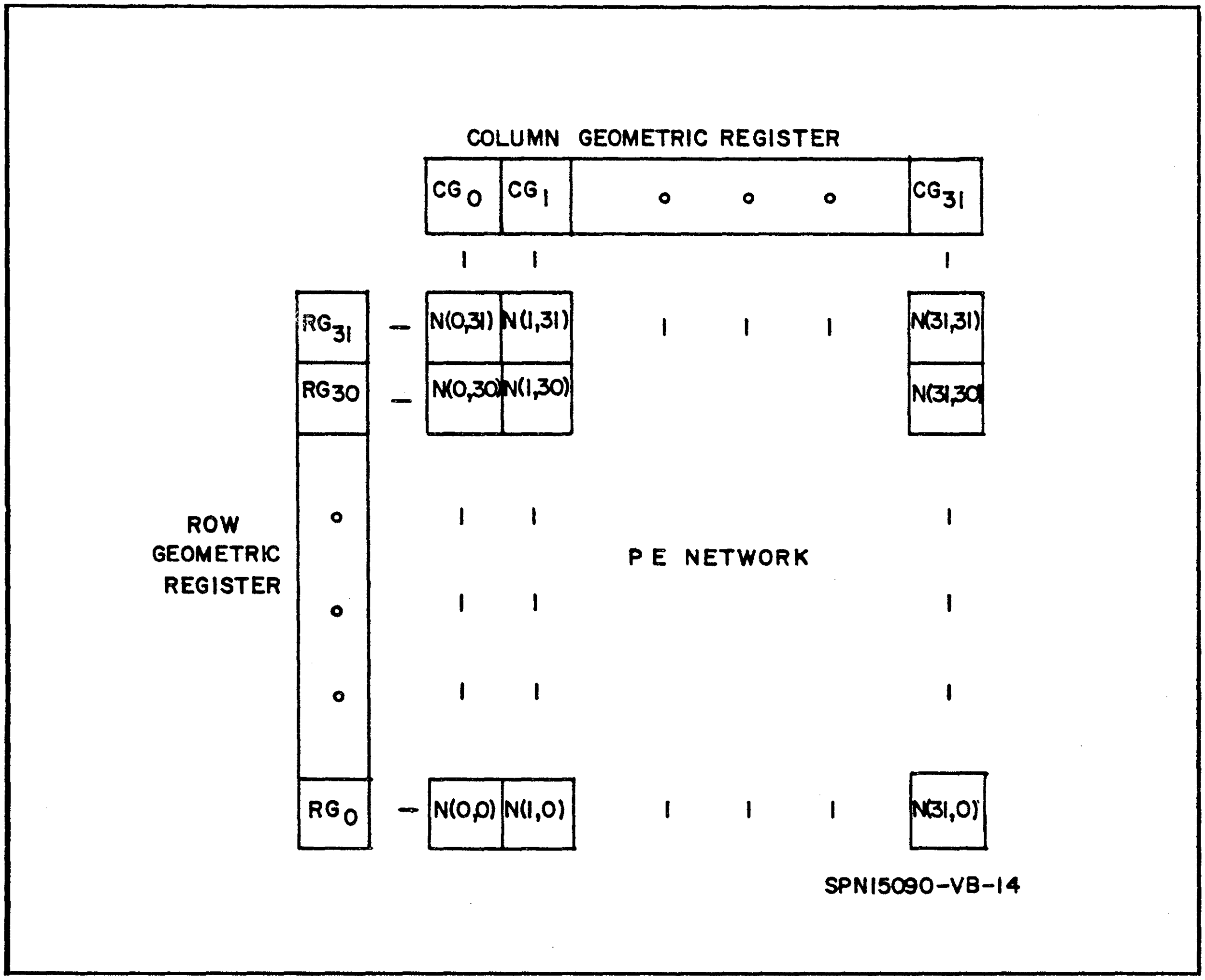

Figure 1-5. Geometric Control Registers

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 rate 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 information and 0.2 microseconds for a switching blank.

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.

1.1.2 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 101-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.

A block diagram of one of the basic modules of the PE memory is shown in figure 1-6.

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

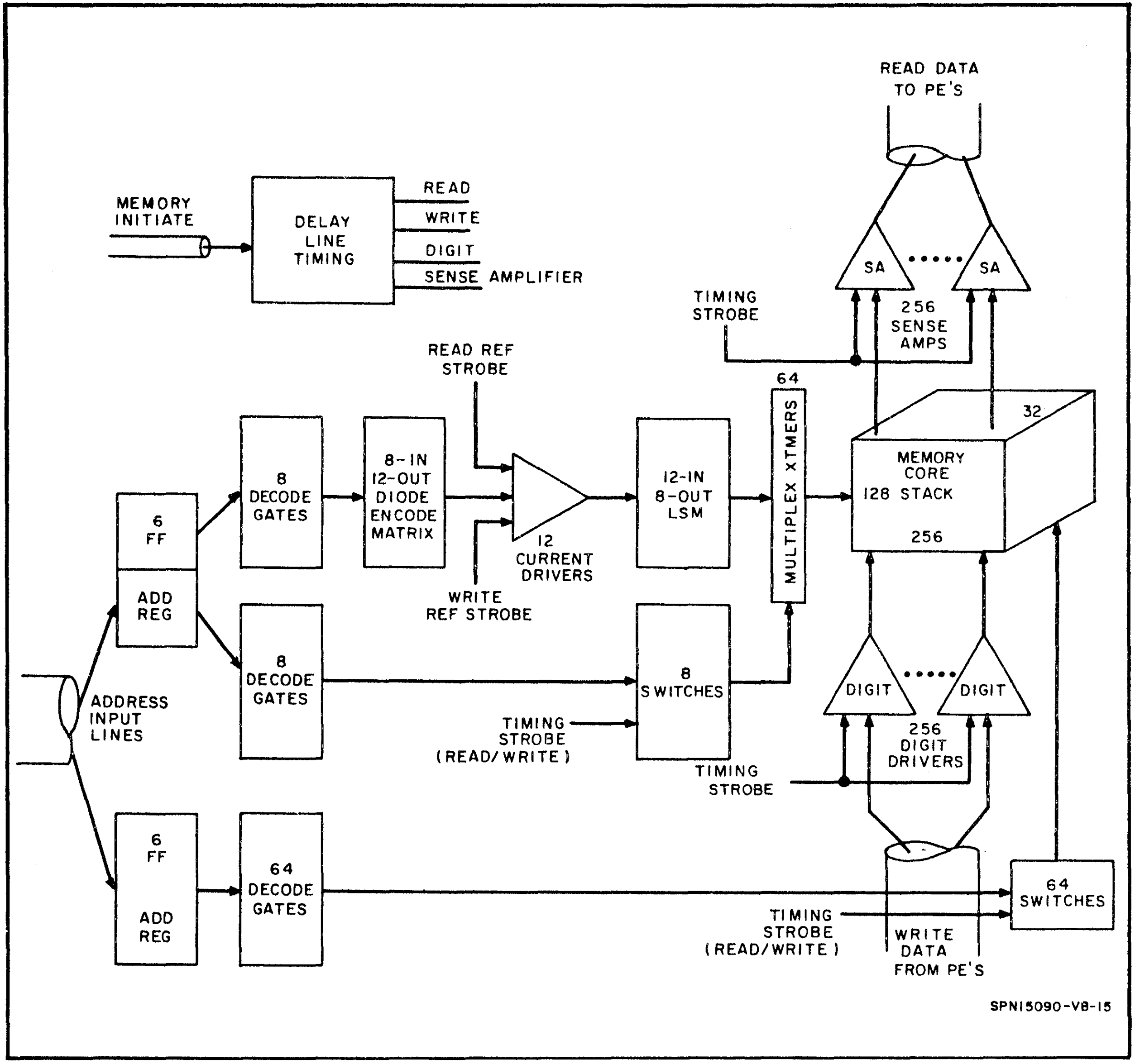

Figure 1-6. Block Diagram SOLOMON PE Memory

partial write pulses. A 12-input, 8-output LSM is used with each output multiplexed to 8 places through the use of center-tapped transformers and saturated switches. Twelve constant current drivers are used to energize the LSM transformers.

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.

1.2 THE NETWORK SEQUENCER

The PE Sequencer (see figure 1-7) 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 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.

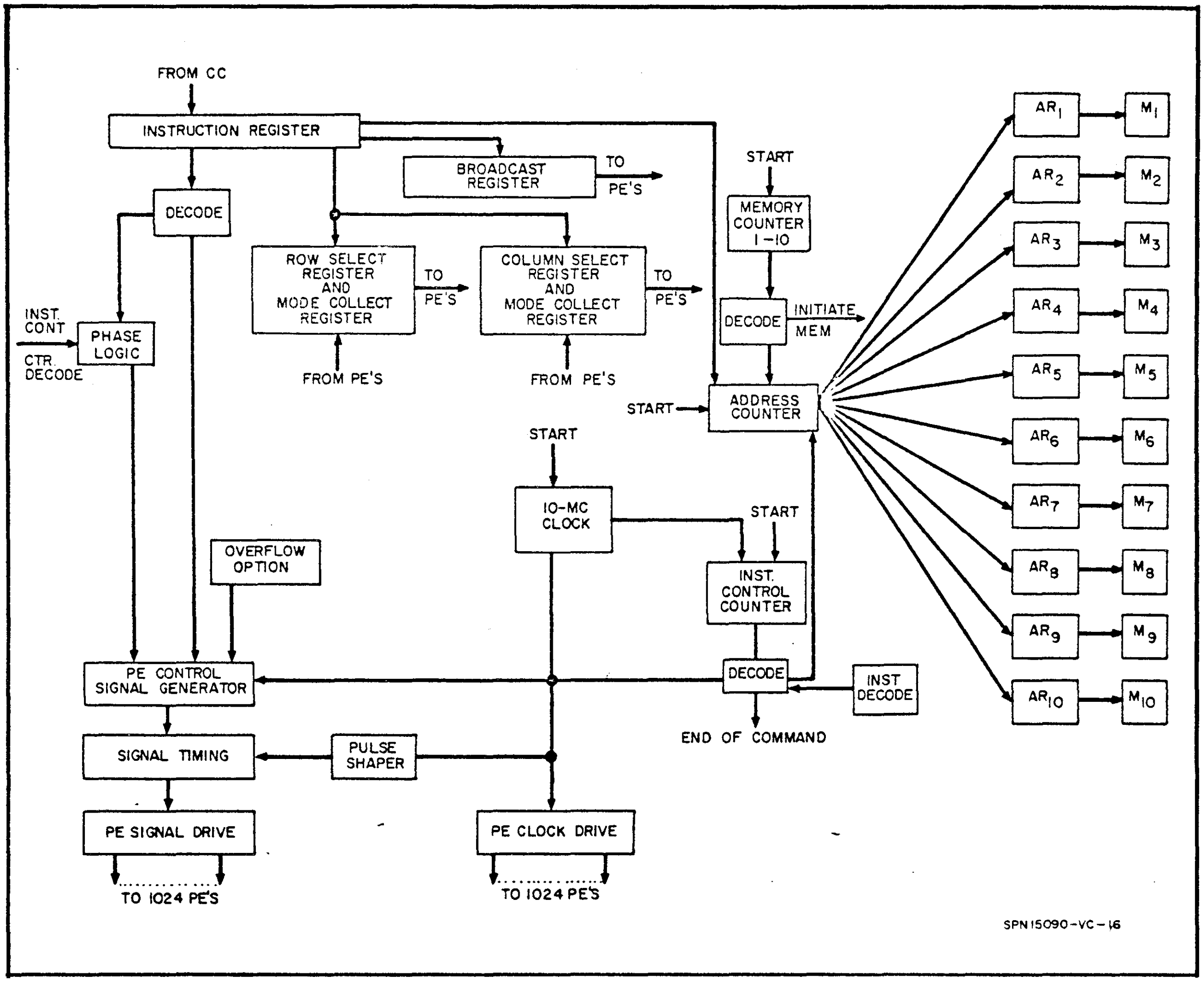

Figure 1-7. Network Sequencer

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. 3 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.

1.3.1 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 1-8. 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.

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.

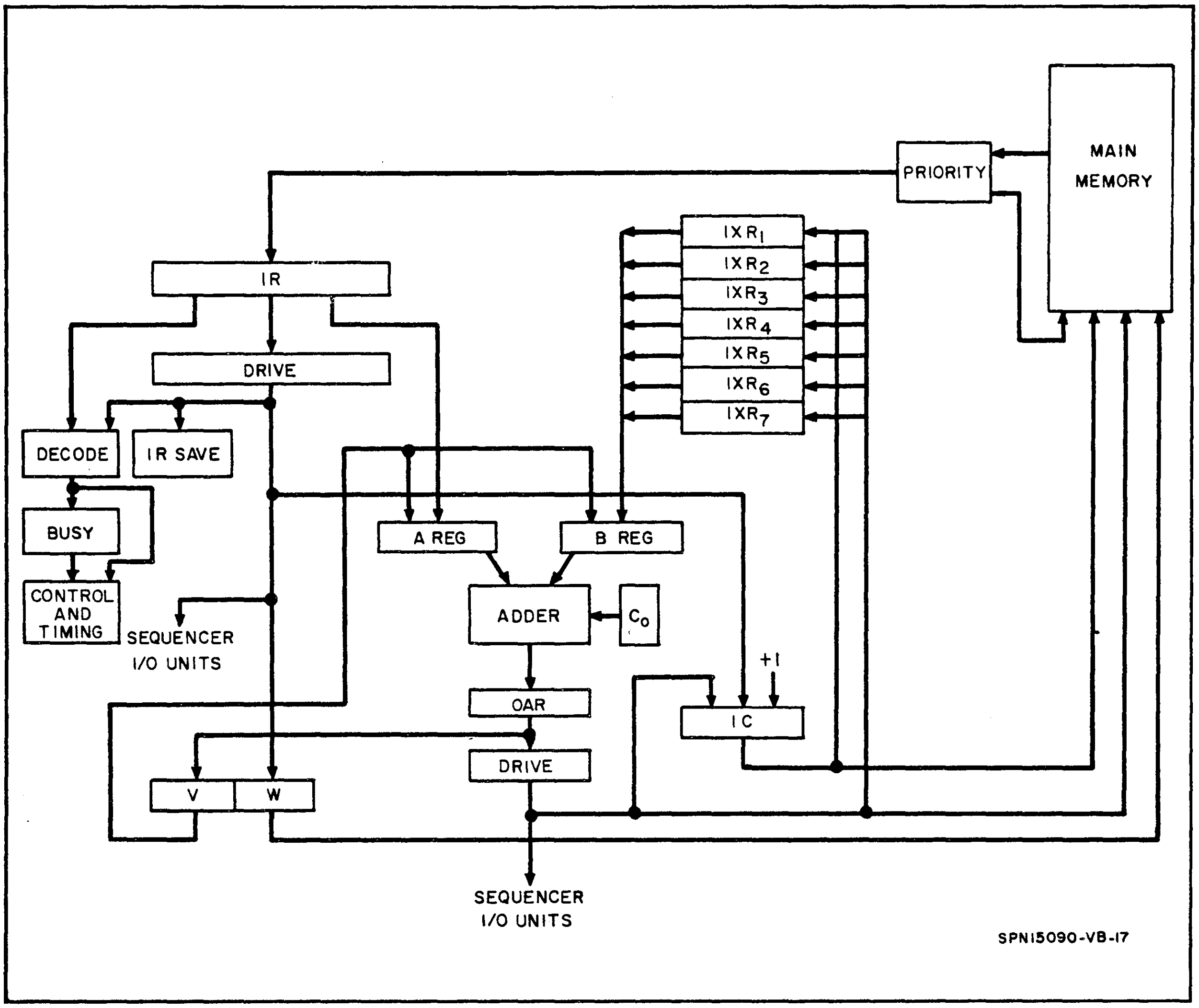

Figure 1-8. Network Control Unit

1.3.2 General Purpose Control Unit (GPU)

The organization of the GPU is shown in figure 1-9. 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.

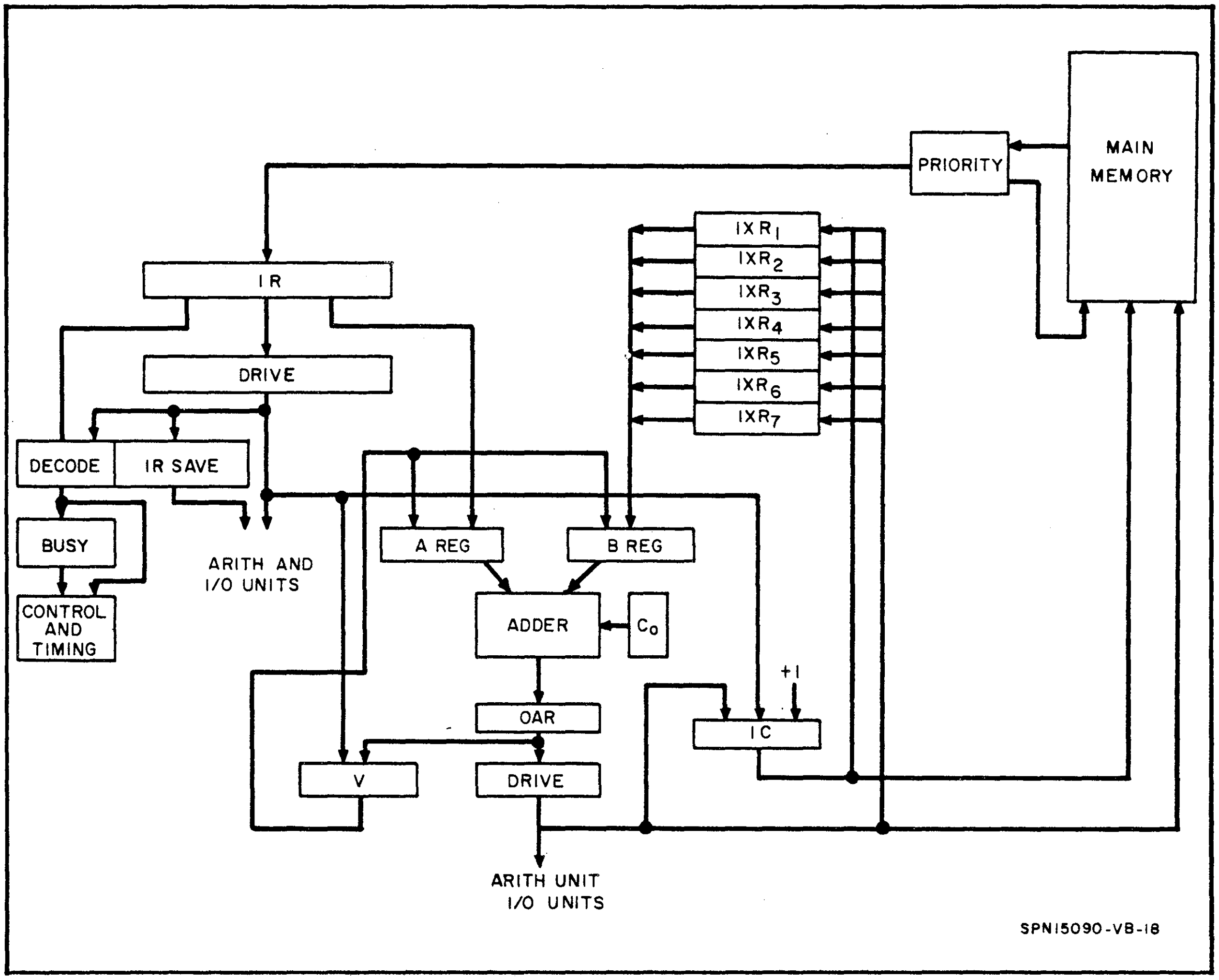

Figure 1-9. General Purpose Control Unit

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).

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.

1.4 ARITHMETIC UNIT

1.4.1 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, figure 1-10 contains 3 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.

1.4.2 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.

1.4.3 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

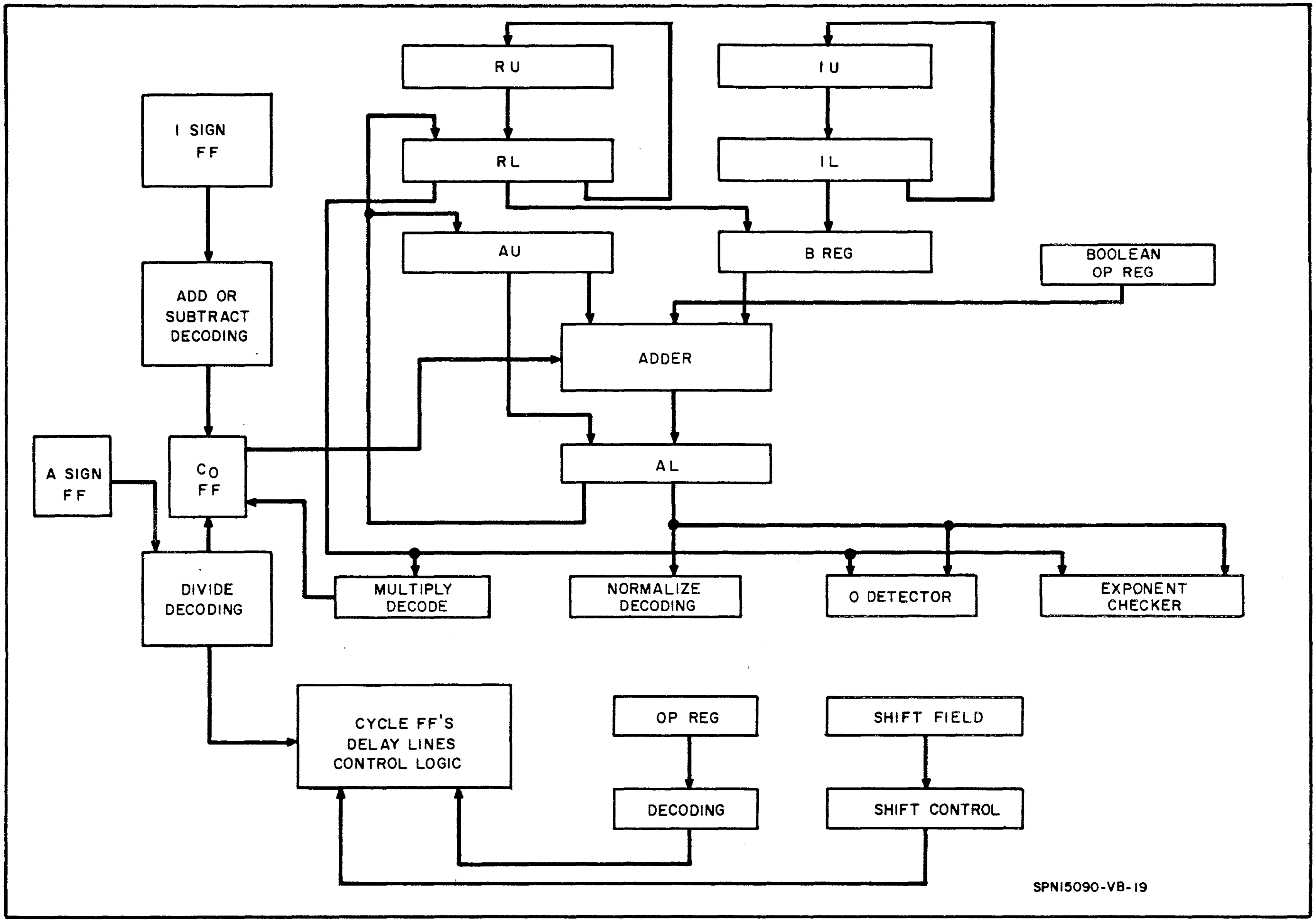

Figure 1-10. Arithmetic Unit

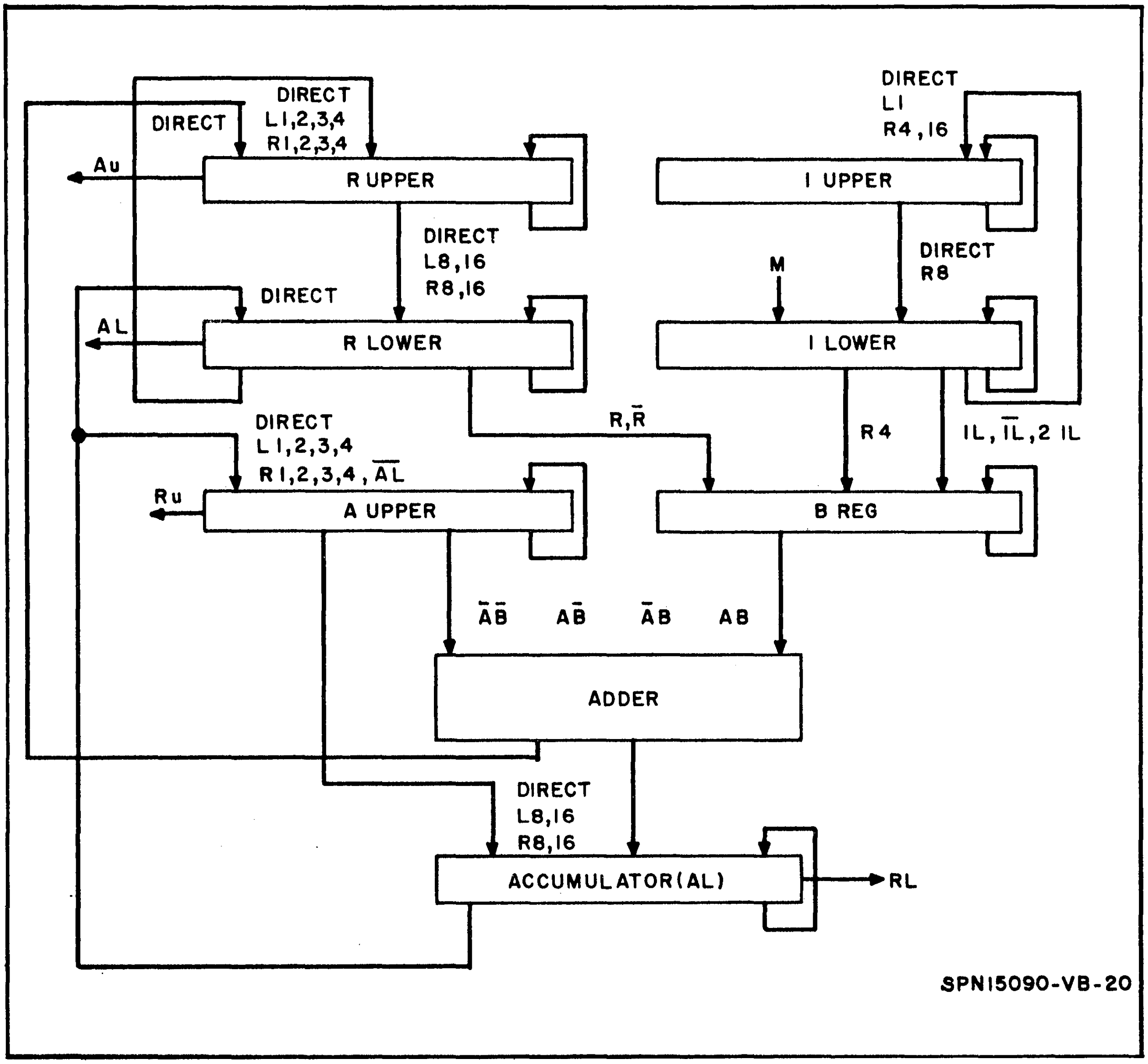

Figure 1-11. Arithmetic Unit

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.

1.5 INPUT-OUTPUT (I/O) SUBSYSTEM

The SOLOMON II Computer Input-Output (I/O) Subsystem (figure 1-12) provides a maximum of 32 data transfer channels or trunks which handle the

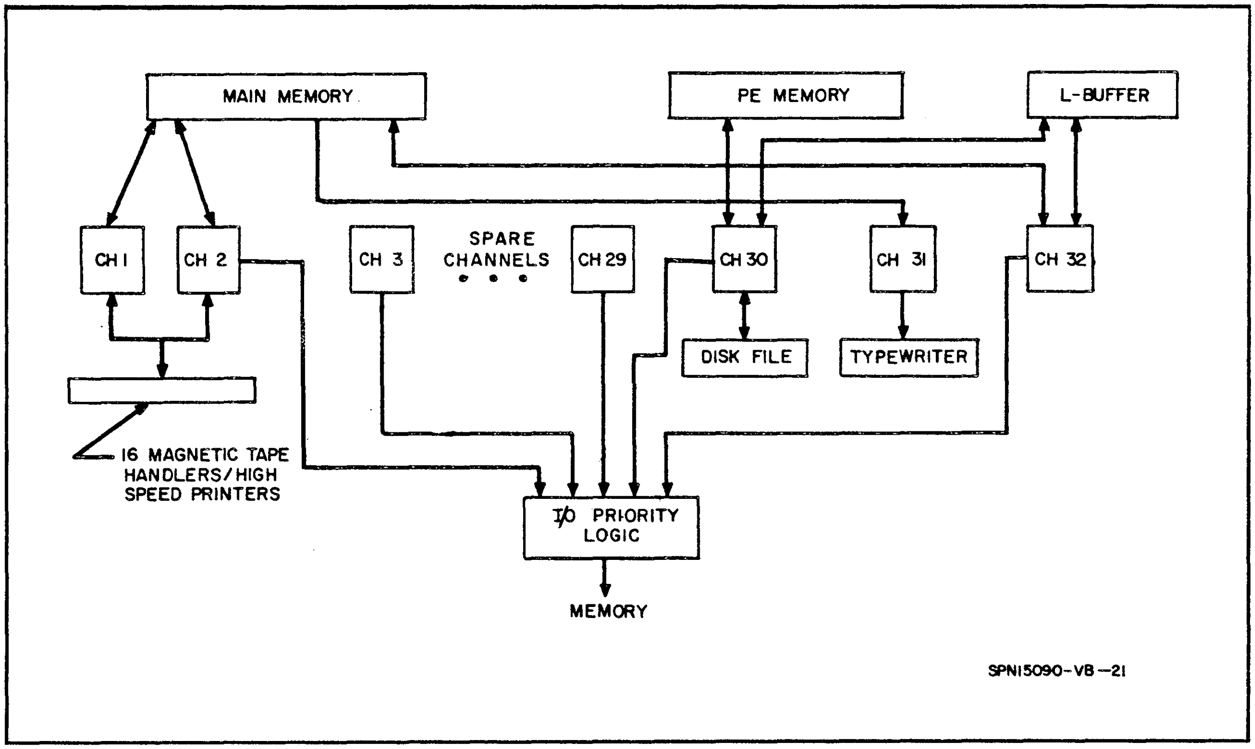

Figure 1-12. Input-Output Data Channels

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. 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 patchboard 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.

1.5.1 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 1-13.

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.

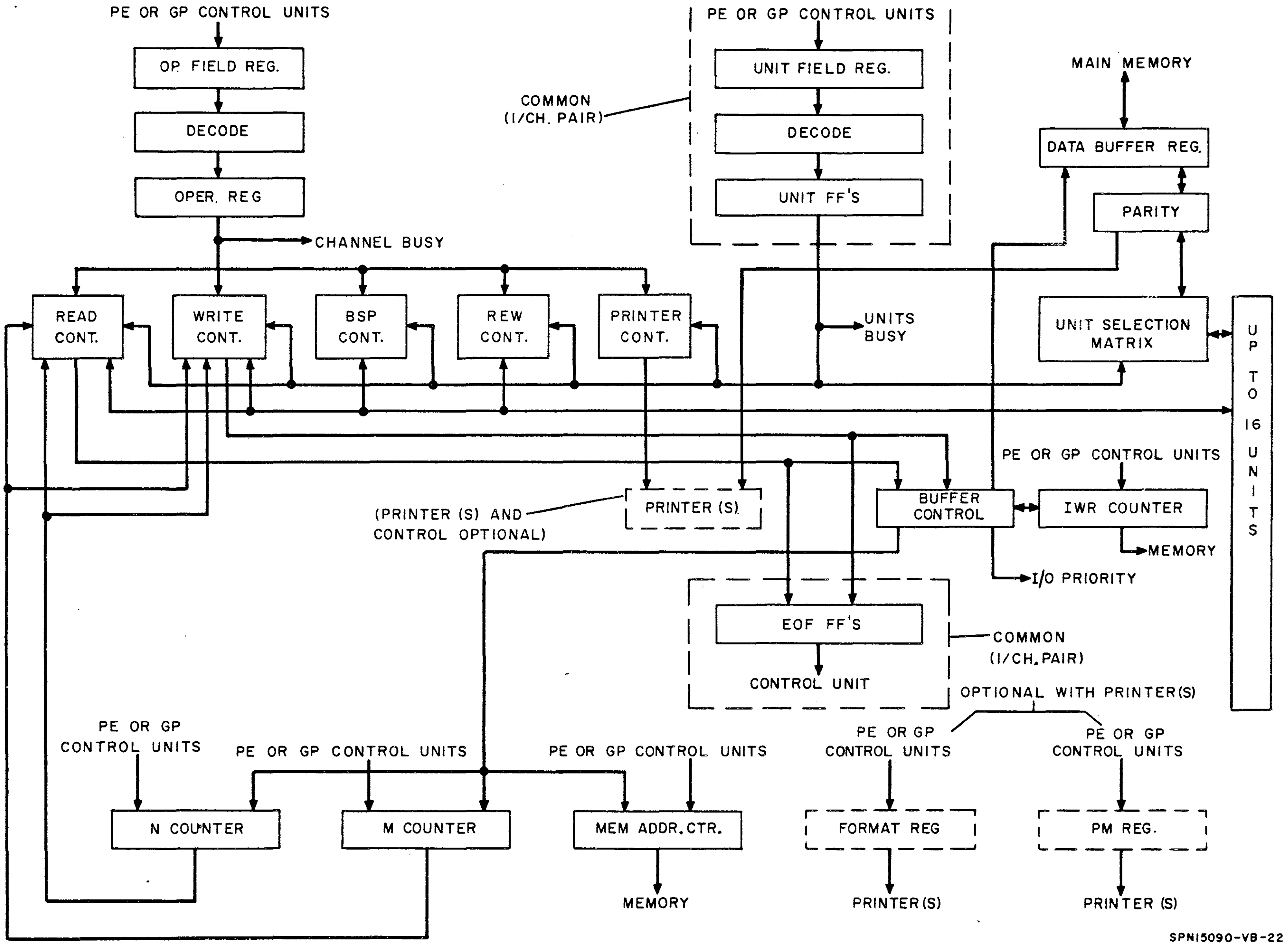

Figure 1-13. Magnetic Tape Transport/Line Printer Channel

1.5.2 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 1-14.

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

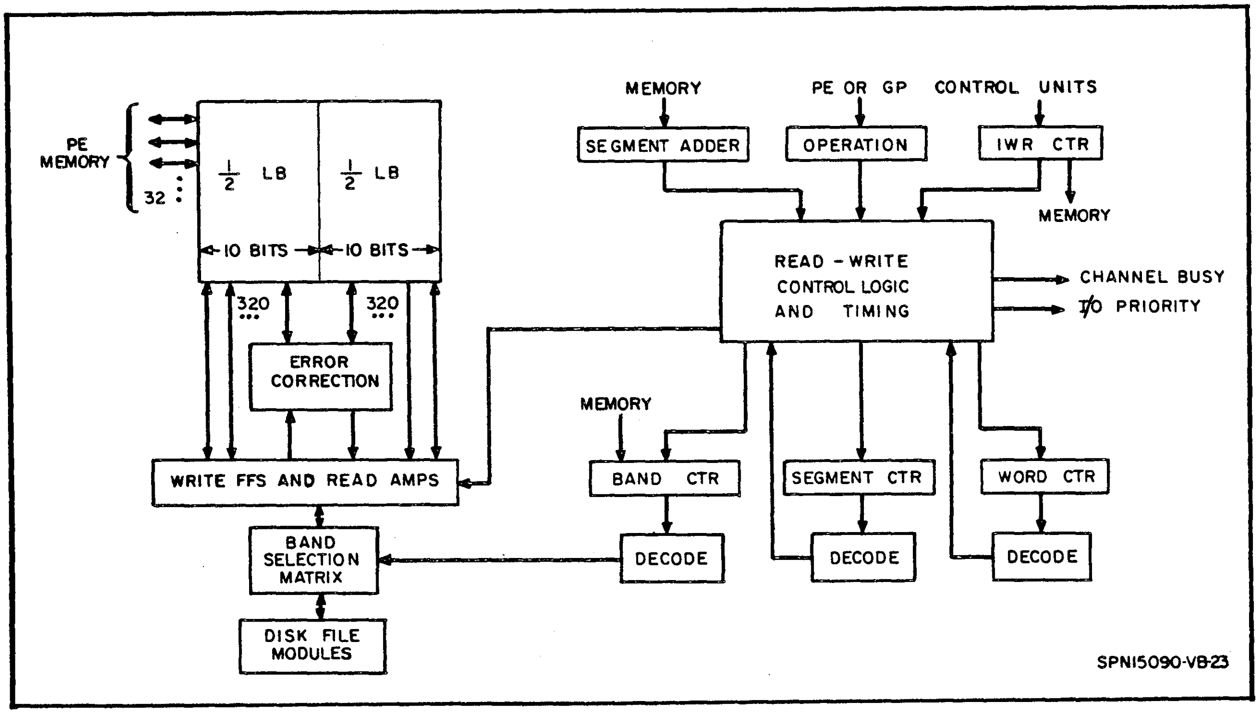

Figure 1-14. Disk File Channel

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)

Organization	Operation Frequency (mc)
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.

1.5.3 Channel 31 - Typewriter

The typewriter channel control is illustrated in block diagram form in figure 1-15. 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 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.

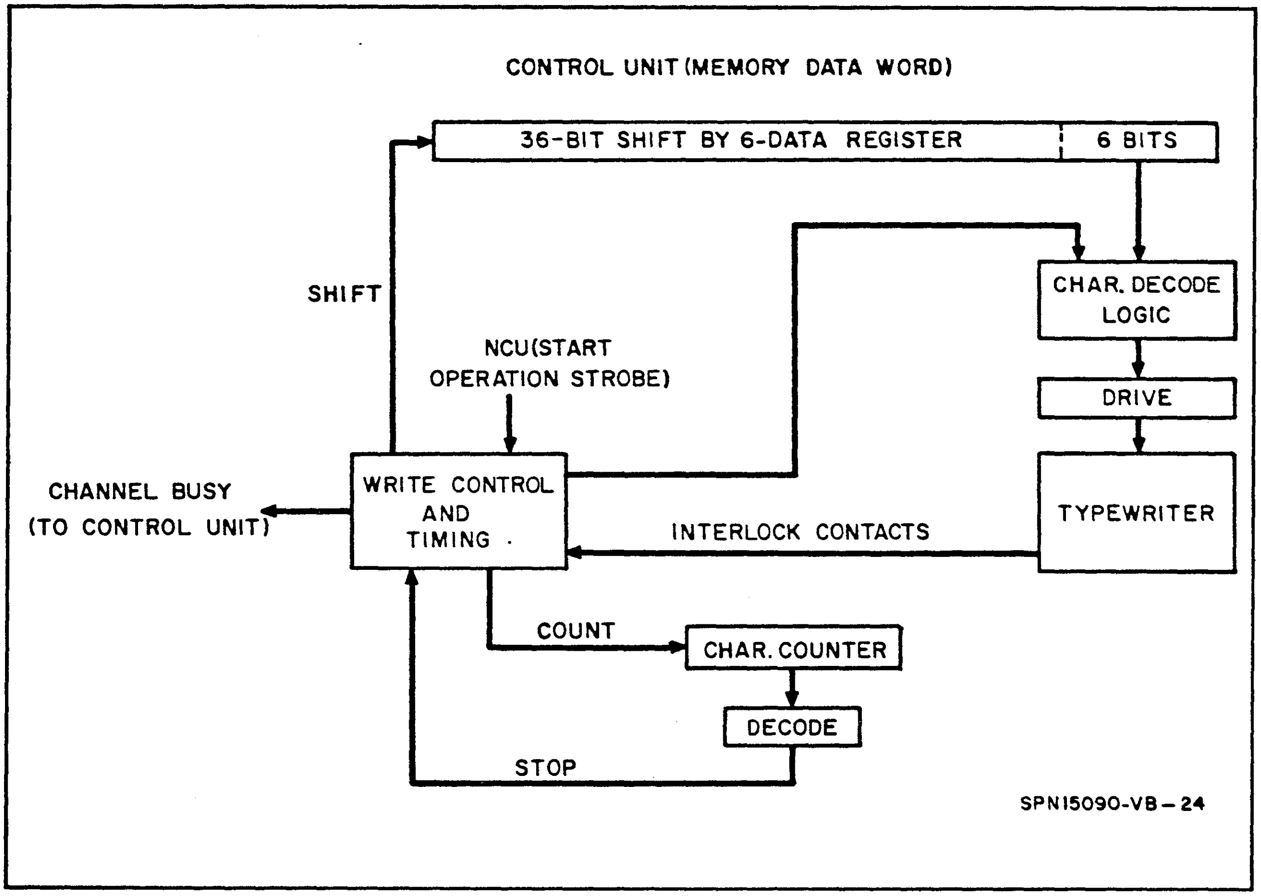

Figure 1-15. Typewriter Channel Control

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.

1.5.4 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 1-16. 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

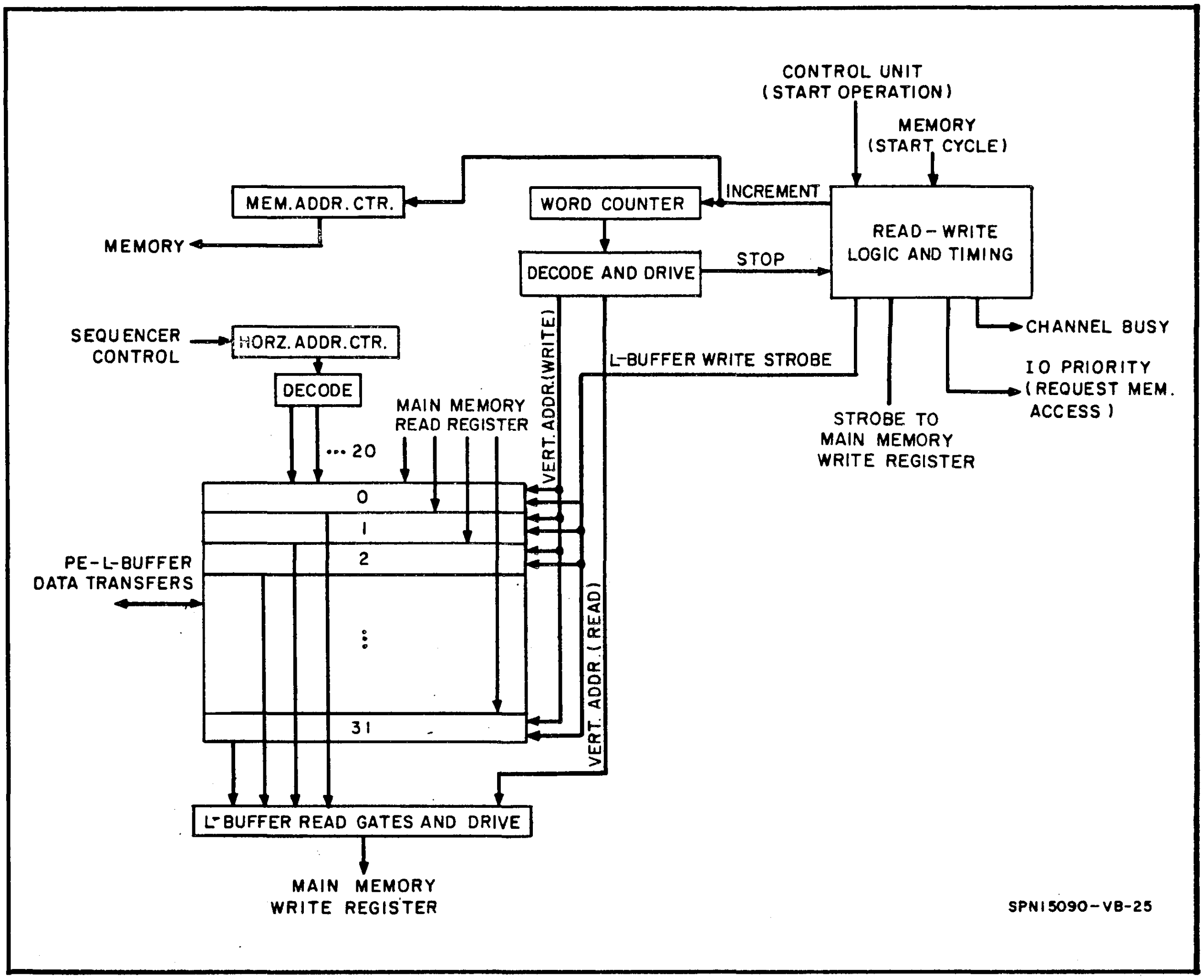

Figure 1-16. L-Buffer Transfer Channel

operation pulse to the 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.

1.6 SOLOMON CIRCUITS

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.

1.7 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 1-17. 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.

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.

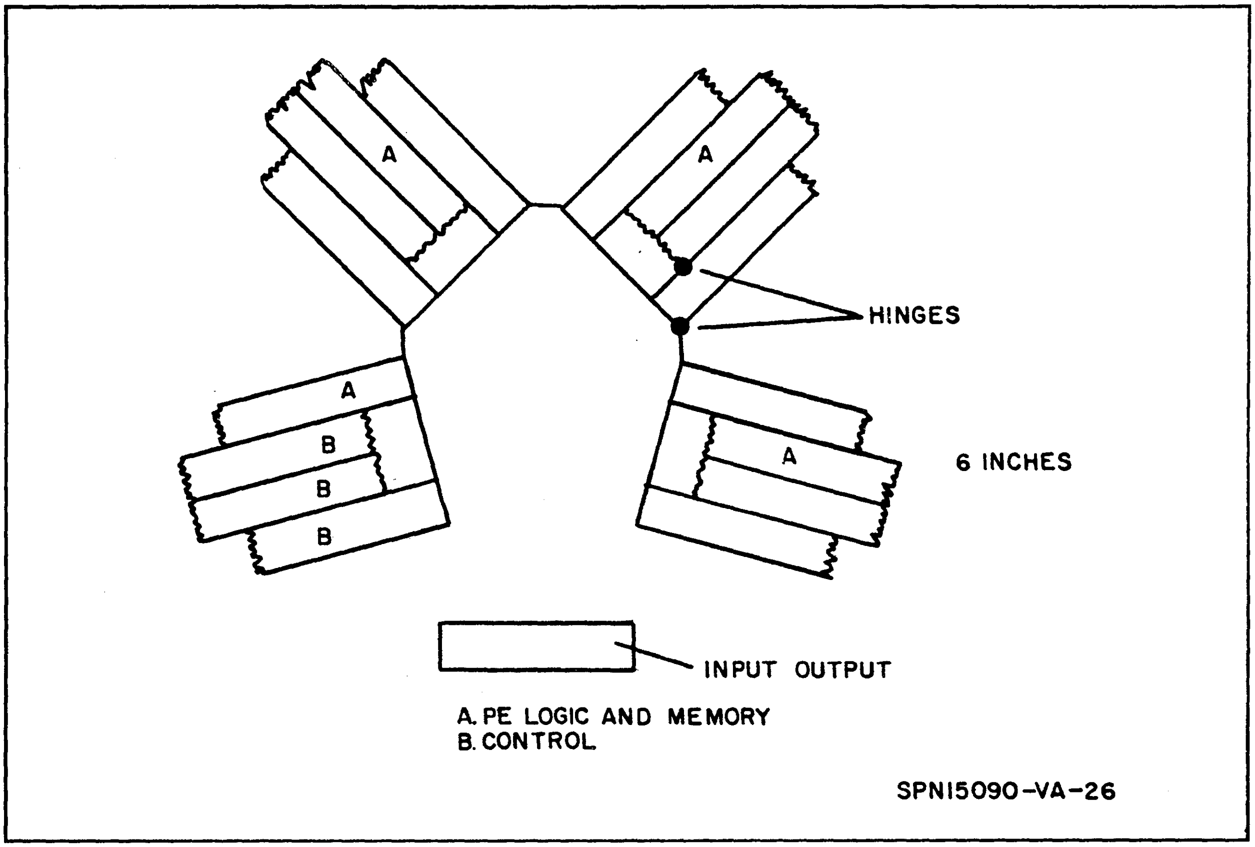

Figure 1-17. SOLOMON II Packaging Configuration

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 nonsealed 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.