SDS 940

THEORY OF OPERATION

Technical Manual

SDS 98 01 26A March 1967

SCIENTIFIC DATA SYSTEMS/1649 Seventeenth Street/Santa Monica, California/UP 1-0960

Printed in U.S.A.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES (Continued)

I. GENERAL DESCRIPTION

1.1 GENERAL

The SDS 940 Time-Sharing Computer is a general-purpose digital computer that operates at high internal speeds and uses a random access core memory. The SDS 940 Computer is an SDS 930 General Purpose Digital Computer that has been modified through the addition of special hardware features to produce an excellent computer for time-sharing applications.

From a hardware standpoint, the 940 Computer is a 930 Computer modified to permit time-sharing operation. The requirement that distinguishes the time-sharing atmosphere is many simultaneous users. The modifications enable the logic to preserve the privacy of each user. These modifications combine with the high speed of operation of the central processor to enable many users to use the computer capability simultaneously. The complete line of proven SDS peripheral equipment and all SDS system components are available with the 940 Computer to achieve unparalleled efficiency in handling the flow of information to and from the computer.

From a programming standpoint, the 940 Computer, being an extension of the 930 Computer, is fully compatible with all SDS 900 Series software. Because of this compatibility, the complete set of field-proven SDS software is available as any programs written for an SDS 900 Series computer can be run, without modification, on the 940 Computer. In addition, however, an extensive time-sharing software package is available for the 940 Computer that is not compatible with other SDS 900 Series Computers.

1.2 DOCUMENTATION

Since a single manual presentation of all the documentation concerning the 940 Computer is impractical, this manual provides an introduction to the 940 Computer and provides information for locating additional information that might be useful in operating and maintaining the 940 Computer.

The contents of this manual present the differences between the 940 Computer and the 930 Computer. This manual is primarily for use by field maintenance personnel with a knowledge of the operation and maintenance of the 930 Computer. Thus, Section I, General Description, provides overall pictures of the 940 Computer as it normally appears as well as an overall description and applicable characteristics. Section II, Operation and Programming, describes the modes of operation of the 940 Computer, and presents the operation codes. The logic designed especially for the 940 Computer is described in Section III, Theory of Operation. Installation and maintenance instructions are provided in Section IV insofar as applicable. Lists of reference drawings, standard equipment manuals, and module data sheets are provided in subsequent sections in a manner facilitating ease of reference.

1.3 PHYSICAL DESCRIPTION

The 940 Computer comprises a double-bay central processor cabinet, a single-bay power supply cabinet, a minimum of one single-bay input/output cabinet, and a control console. Figures 1-1 and 1-2 show the front and back of a 940 Computer, respectively.

A special swing-out door located behind the front cabinet doors in the two-bay central processor cabinet houses the memory logic chassis. The first 16,384-word core memory is behind the left front door, the second 16,384-word core memory is behind the right front door. Access to the back panel of the memory logic chassis is facilitated by the hinged memory doors. With the memory doors open, maintenance personnel can also gain access to the rear of the central processor logic modules. The central processor modules are loaded and removed from the rear of the cabinet behind the two central doors. Peripheral couplers are housed in the input/output cabinet as required. Additional cabinets can be added to house additional peripheral couplers. Eleven power supplies including power supplies for the two 16,384-word core memories are housed in the power supply cabinet. Two additional 16,384-word memories can be added and are housed in individual cabinets. The power supplies required by each memory are housed in the same cabinet as the memory. The additional memory cabinets are inserted between the central processor cabinets and the power supply cabinet.

Controls and indicators for manual operation and monitoring of the 940 Computer are panel mounted on a convenient table for ease of use as shown in Figure 1-3.

1.4 FEATURES

As an extension of the 930 Computer, the 940 Computer retains all the features of the 930 Computer and provides additional special features. Monitor and User modes of operation are available in addition to the 930 Operation Mode. A hardware-implemented memory map, system programmed operators, and non-stop operation protection are also included. Additional 940 Computer features are presented in Table 1-1.

1.5 INPUT/OUTPUT CAPABILITY

The 940 Computer has comprehensive input/output systems that complement the high internal processing speed and versatile instruction repertoire. These input/output systems have a scatter-write and a gather-read capability and can transmit data in word, character, or single-bit form to and from the computer at the speed of internal computation. While the input/output systems assume control of conditions imposed by various characteristics of a wide variety of devices, a high degree of control is left to the programmer. The computer configuration shown in Figure 1-4 contains four types of input and/or output.

(a) Parallel Input/Output System. Direct parallel input/output of 24-bit information either external or program controlled and sequenced, and up to 16,384 single-bit signals for output control or input test.

(b) Time Multiplexed Communication Channels. Buffered input/output of data words, each under direct program control.

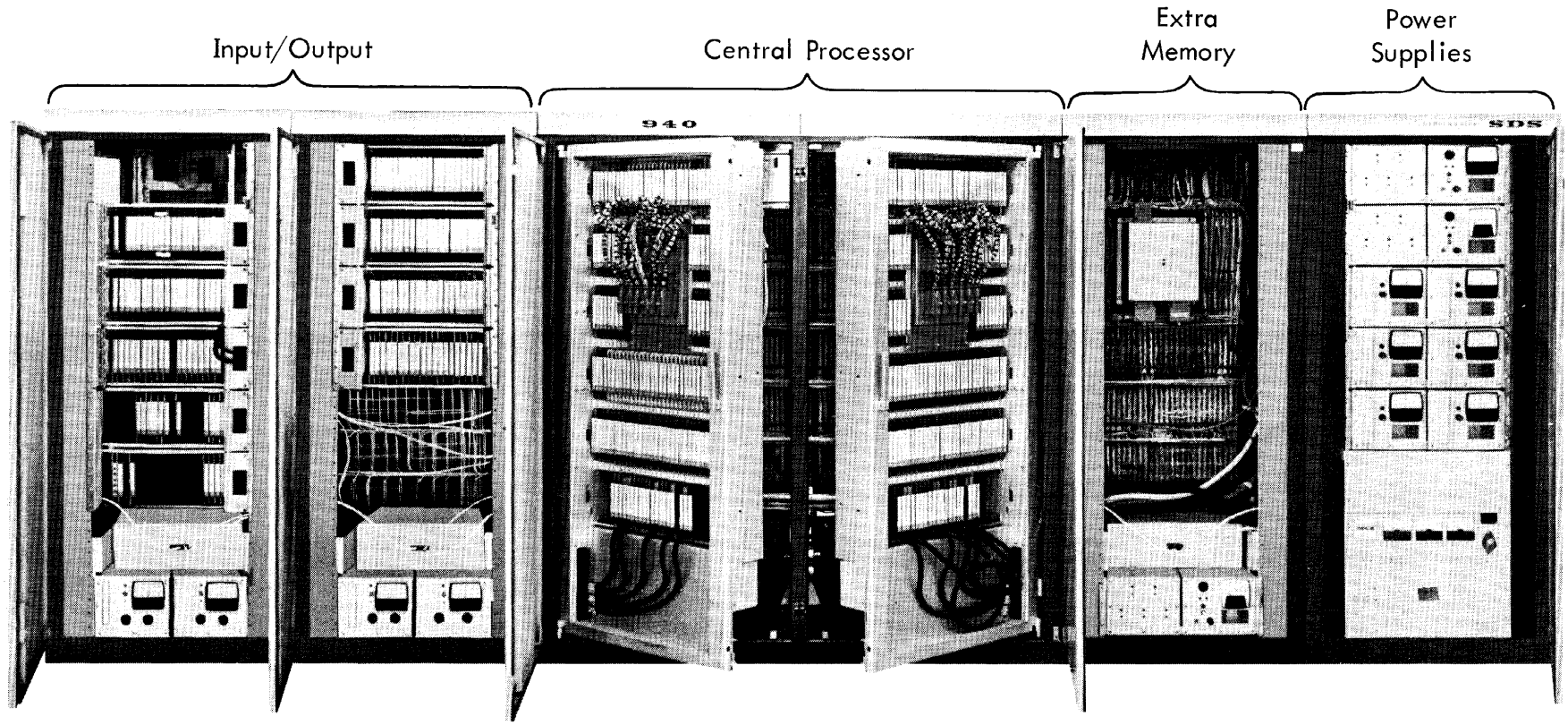


Figure 1-1. SDS 940 Computer, Front View

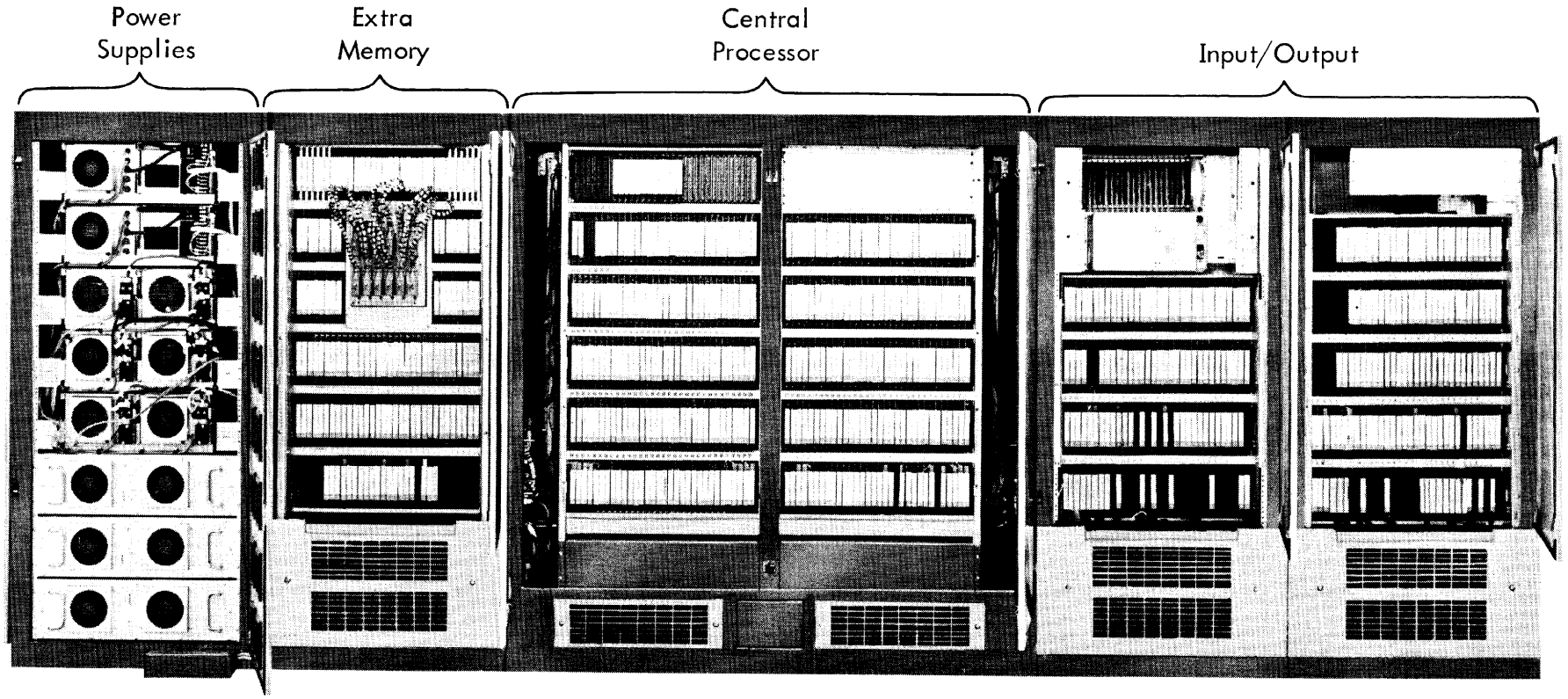


Figure 1-2. SDS 940 Computer, Rear View

Table 1-1. 940 Features

Feature	Definition
Word Length	24 information bits plus parity bit.
Arithmetic	Binary
Memory Size	16,384-word memory is required for the Time-Sharing Software system. Memory is expandable to 65,536 words in 16,384 word increments.
Access Time	A 0.7-microsecond access time results in a complete read/write cycle time of 1.75 microseconds.
Memory Protection	An optional feature enables the transfer of register contents to memory in the event of power failure.
Memory Interleaving	When two devices are accessing different memory modules, the operations proceed simultaneously.
Error Checking	All memory operations and input/output transmissions are parity checked.
Priority Interrupts	A completely automatic priority interrupt system provides up to 896 optional priority levels individually armed and disarmed under program control.
Input/Output	Six different input/output methods permit computer operation with diverse input/output devices. Extensive buffering allows numerous input/output operations simultaneous with one another and with computation. Magnetic tape and disc searching is accomplished independent of memory, thereby requiring no computer time.
Construction	Exclusive use of silicon semiconductors assure high reliability; the small component count yields very long Mean time between failure rates. Air conditioning is not required, since the central processor operating temperature range is 10° to 40°C.
Power Requirements	3 kva at 208 volts ± 10% (three-phase Y), 60 ± 0.5 cps.

(c) Priority Interrupt System. Permits immediate reaction to special external and internal conditions.

(d) Direct Input/Output System. Input/output of characters or words from memory, time-shared and multiplexed with computation or fully buffered and simultaneou with computation.

Each individual type of input/output is more fully described in the paragraphs that follow.

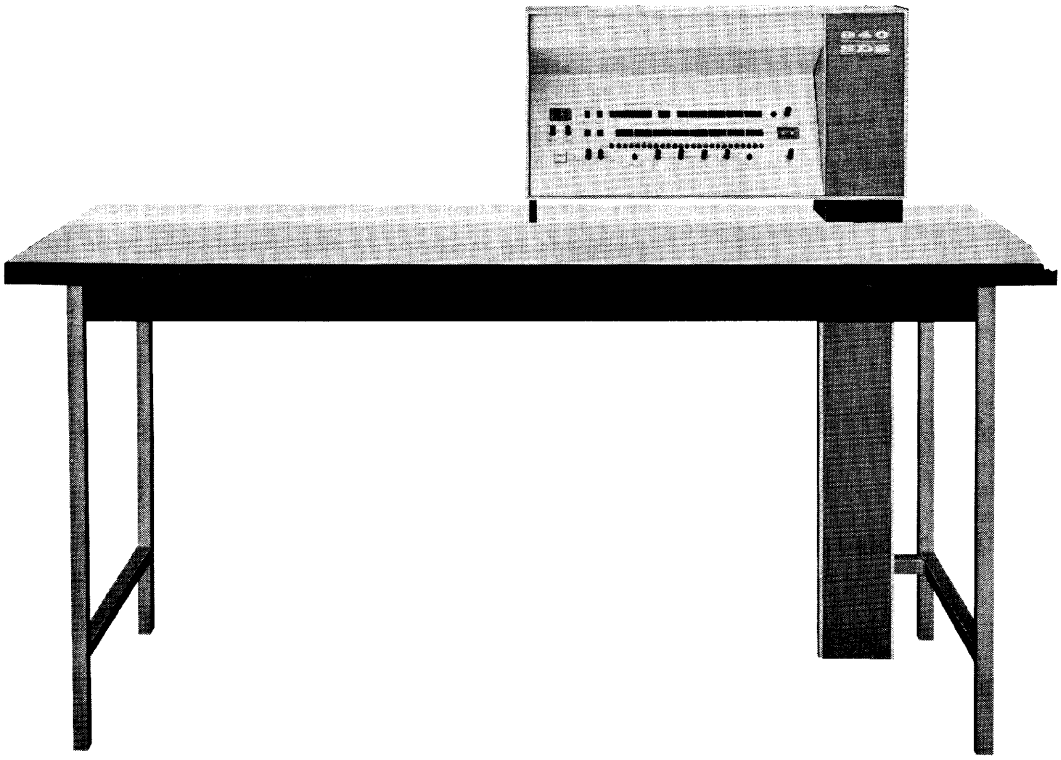


Figure 1-3. SDS 940 Computer Console

1.5.1 PARALLEL INPUT/OUTPUT SYSTEM. The parallel input/output system yields high-speed input/output of data or control and test signals with a transfer rate of one 24-bit word in 1.75 microseconds. This system facilitates operation with many types of asynchronous information under program control.

1.5.1.1 Word Parallel System. Under program control, any word in the memory can be presented in word-parallel format (24 bits) at a processor connector. Also under program control, any 24-bit data word sent to the processor connector can be stored in any memory location. The execution of either operation starts with the transmission of a signal to the external device signifying that the computer is ready for the transmission of data.

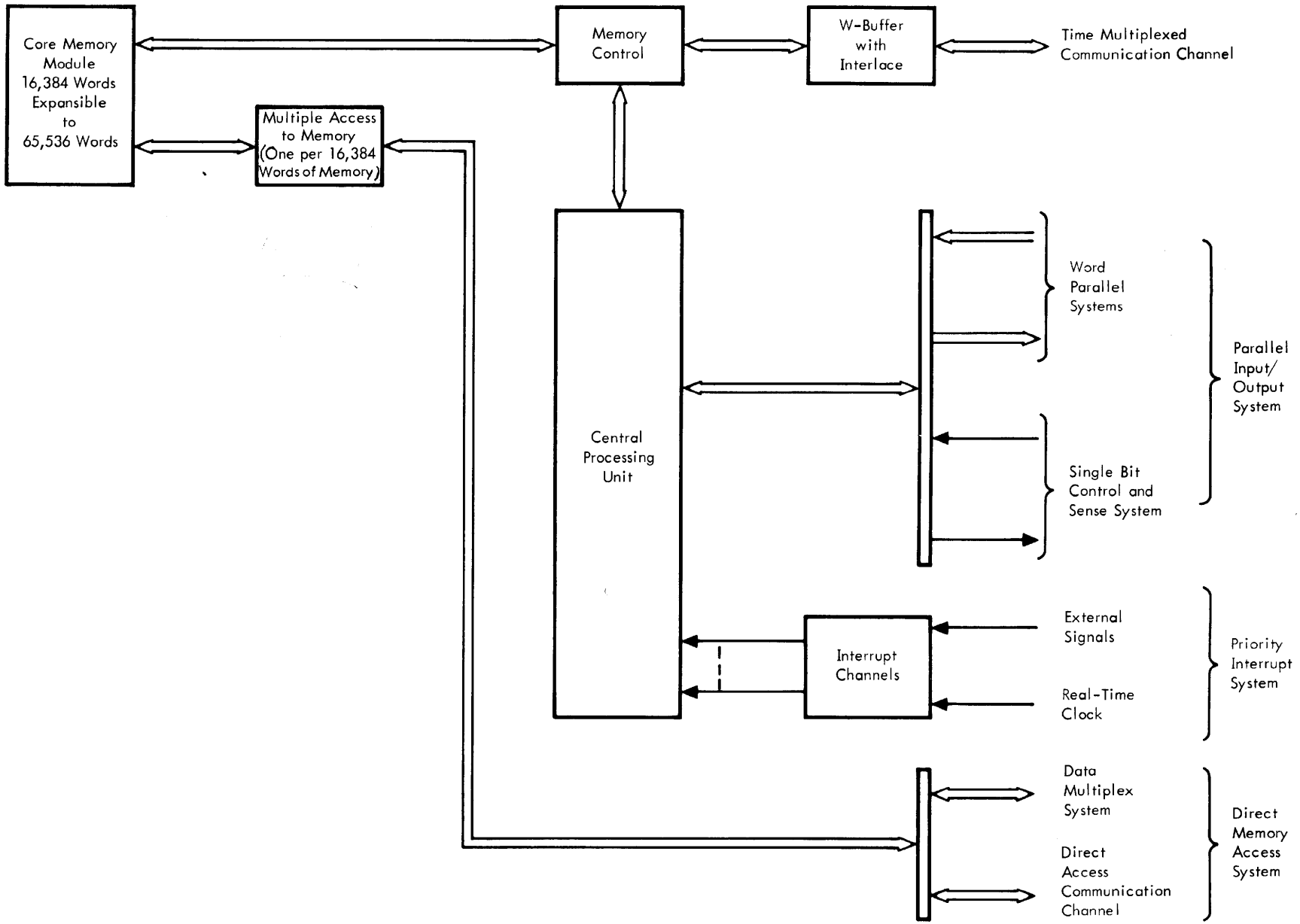


Figure 1-4. Basic SDS 940 Computer Configuration

During input operation, the computer is in a hold condition until the external device selected has presented the data accompanied by a ready signal. Only after the computer has detected the ready signal can the computer terminate input and continue the program. During output, the computer is in a hold condition until the external device signals the computer that the selected data has been received. Therefore, all input transmissions are close-looped.

To overcome the necessity of locking up the computer during input, the ready signal from the external device can be used to initiate an interrupt that causes the computer to branch to an input routine when data is ready for input. The computer is thus free to execute other programs until information appears at the parallel input connection to be received and stored. To avoid the output hold, the external device can use the interrupt system to signal the computer to output data only after the peripheral device is ready to receive data from the computer.

If a series of data words are to be transferred to or from sequential memory locations, the address portions of these instructions must be modified. Address modification is accomplished by index registers in conjunction with instructions for modifying the register, or by indirect addressing with the addition of instructions used for incrementing or decrementing specified memory locations.

1.5.1.2 Single-Bit Control and Sense System. The single-bit control and sense system issues instructions to all external devices and tests conditions in these external devices. When the computer sends an instruction for execution, a 1.4-microsecond pulse indicates to the device specified by the address portion of the instruction that the computer is requesting a specific operation. This signal is 0 volts for binary zero and +8 volts for a binary one.

When the computer sends an instruction requesting a test for a condition in an external device, a line goes true until that condition response is strobed to the computer. If the signal specified by the address portion of the test instruction is not true, the computer skips the next instruction in sequence. If the signal is true, the computer executes that next instruction in sequence. The computer can perform logical operations based on the presence or absence of the tested signals.

1.5.2 TIME-MULTIPLEXED COMMUNICATION CHANNELS. In addition to the word-parallel system, the 940 Computer includes one Time-Multiplexed Communication Channel (W) as standard equipment, as well as provision for three additional channels (Y, C, and D). These channels are capable of automatically controlling the flow of data to and from memory at rates up to 286,000 words per second. Independent channel operation only interferes with computer operation for data transfer between the Time-Multiplexed Communication Channel and memory.

Time-Multiplexed Communication Channels use central processor memory logic to facilitate the input and output of data words. The transfer of each word between the channel buffer and the memory requires two memory cycles. During this transfer time, computation is held up in the central processor. Priority for use of the TMCC input/output logic is in the order: Channel D, C, Y, and W. Any channel operating with interlace has priority over the central processor for memory access. Words can be transmitted between memory and peripheral devices under the

direct control of single instructions. A WIM instruction causes a word transmitted from a peripheral device to be taken from the channel buffer register (W) and placed into a specified memory location (M). A MIW instruction causes a word to be taken from a specified memory location (M) and placed into the channel buffer register (W) for output to the peripheral device currently connected to the channel. Both WIM and MIW instructions are preceded by instructions that set up the input or output operation. The transfer of data is done at the rate that either the WIM instruction for input or the MIW instruction for output can be executed which depends on the peripheral device connected to the channel.

1.5.3 DIRECT MEMORY ACCESS SYSTEM. A direct memory access system is also incorporated in the 940 Computer. This system uses a Multiple Access to Memory unit to provide a path to memory separate from that used by the central processor. Because of the separate access to memory, data transfer through the direct memory access system does not interfere with the central processor unless both are trying to use the same memory module at the same time.

The direct memory access system provides direct transmission between special-purpose peripheral devices and the computer core memory. Devices controlled and sequenced externally can present data and addresses to the direct access connections. Thus, input/output operations can be completely independent of the computer. The special input/output devices present an address plus various timing and control signals to the direct access connection. External data can be stored in any specified location or read from any location specified by the external unit.

Two methods of data handling depending on speed are used in conjunction with the direct memory access system. Individual sources are handled by the Direct Access Communication Channels. Multiple, relatively slow sources are handled by the Data Multiplexing System. Each of the data handling methods are described in the paragraphs that follow.

1.5.3.1 Direct Access Communication Channels. Four Direct Access Communication Channels can be attached to the direct memory access system. Although these channels operate in exactly the same manner as the Time-Multiplexed Communication Channels, they are faster and provide for a true input/output overlap with processing. These channels automatically control the flow of data to and from memory at rates equivalent to 572,000 words per second.

Each Direct Access Communication Channel has memory logic independent of the Central Processor. When memory access is needed to access or store a data word, computation is held up for one cycle if the access is in the same module that the central processor is addressing. If the module is not currently being addressed by the central processor, no time is lost and computation is unaffected. Thus, internal computation and direct access transmissions occur simultaneously and independently when the computer and the channel are accessing different memory modules.

Transmissions to and from core memory are under control of the Direct Access Communication Channel. At the beginning of each memory cycle, the control unit interrogates all channels to determine whether any channel requires a transfer to or from computer memory. Interrogations are performed in such a manner that each channel has priority on the basis of need. Channel control logic permits the transfer of only one word per memory cycle to and from the computer memory, regardless of the number of operating channels that are connected to the computer.

1.5.3.2 Data Multiplexing System. To transmit data directly to and from memory, a Data Multiplexing System has been developed. This system is of particular value in dealing with multiplexed sources of data each of which is slow relative to the maximum memory rate, or for systems which have very high data rates. The data multiplexing system uses the Multiple Access to Memory, which is completely under the control of attached units. The structure of the system permits the connection of unlimited Memory Interface Connections, and a Data Multiplex Channel that can operate with up to 64 Data Subchannels. Priorities can be individually assigned to each Data Subchannel or Memory Interface Connections.

Either internal or external interlace can be used. If internal interlace is used, two interlace words, memory location and word count, are stored in memory and accessed each time a given subchannel transmits information. If external interlace is used, the identical information is held in an external register. Maximum information transfer rate is one word every 5.25 microseconds with internal interlace; the transfer rate is increased to one word every 1.75 microseconds with external interlace.

Internal interlace allows a Data Subchannel to handle continuous data by alternately working from one memory area to another. Since the Data Subchannel can switch automatically from one interlace word to the other, the program is relieved of the necessity of making real-time responses to the zero-count condition.

The external interlace is attached to a Data Subchannel and comprises a 16-bit address register and an 8-bit count register. These registers are loaded automatically from the internal interlace memory locations when the Data Subchannel is activated. After each word is transmitted, the external interlace increments the address register and decrements the count register. When the count equals zero, the Data Subchannel can generate a program interrupt or notify the external device.

Two types of Data Subchannels are available. One Data Subchannel provides multiple input/output channels for standard SDS peripheral equipment and contains a character assembly/disassembly register that operates on either one or two 6-bit characters or directly on 12-bit words. The other Data Subchannel accepts only 24-bit information, has no internal storage, and operates on high-speed data. The transmitting and receiving devices provide the necessary storage.

1.5.4 PRIORITY INTERRUPT SYSTEM. For operation under real-time conditions, a priority interrupt system provides greater program control of input/output operations, aids in programming input/output and computer operations simultaneously, and permits immediate recognition and reaction to special external conditions. In addition to the interrupts provided with the input/output channels and power fail-safe option, the 940 Computer can have as many as 896 optional priority interrupts. The system can be enabled or disabled, and individual interrupts can be armed or disarmed as described in the following paragraphs.

Enable/disable functions operate on the entire interrupt system, except for the power fail-safe interrupts that are always enabled and armed. If one or more interrupts occur while the interrupt system is disabled, and the system is subsequently enabled, each of the interrupts is processed in accordance with its priority.

An optional arm/disarm feature permits any chosen interrupt to be individually armed or disarmed. The disarmed condition causes a priority interrupt to retain no record of the arrival of a stimulus, while the disabled condition records all stimuli for later processing.

Each armed interrupt channel can assume three states: inactive, when no interrupt has been received; waiting, with an interrupt received but not being processed; and active, when a received interrupt is being processed. The arm/disarm feature controls whether an interrupt can proceed from the inactive state to the waiting state.

The waiting condition exists

(a) If the computer is processing a higher priority interrupt,

(b) If the computer is executing an instruction when the interrupt stimulus arrives,

(c) If the interrupt system has been disabled.

In all cases, the waiting status of the channel is maintained until the interrupt can be processed.

While in the active state, each interrupt rejects new stimuli. Consequently, if a signal is presented to an interrupt while the computer is processing that interrupt, a jump does not occur. Since each interrupt has a defined purpose, the interrupt in progress should be handling the interrupt situations and no new action should be necessary until after the interrupt has been processed. Thus, the computer ignores the stimulus, regardless of its duration or frequency, until the active status is terminated.

If an interrupt of higher priority occurs while a given interrupt is being processed, that interrupt remains in the active state, but the program jumps to the higher-priority location. After processing the higher interrupt, the program returns to the initial interrupt, unless still another interrupt with higher priority is in waiting status.

Three sources of interrupt are used in the 940 Computer; External, Input/Output Channel, and Real-Time Clock. Each of these interrupt signal types is described in the paragraphs that follow.

1.5.4.1 External Interrupt. Each interrupt, when activated, sends the computer program to a unique memory location. The computer does not evaluate the external stimulus; rather, the computer proceeds automatically to the beginning instruction of a unique response subroutine designed for that particular interrupt. This type of interrupt is referred to as a subroutine priority interrupt to differentiate the operation from another interrupt known as a single instruction priority interrupt. In a single instruction priority interrupt, the memory location contains a single-instruction subroutine; after this instruction has been executed, the computer returns to the program unless a higher priority interrupt has been activated. Any interrupt for a particular system can be single-instruction or subroutine.

1.5.4.2 Input/Output Channel. The Time-Multiplexed Communication Channels and the Direct Access Communication Channels are capable of generating interrupts. Each Time-Multiplexed Communication Channel is normally capable of generating two interrupts for the computer while operating in the compatible mode. The end-of-word interrupt occurs at the end of each word transferred from the channel buffer to the peripheral device on output, or when the channel buffer is filled with data from the peripheral device on input. The end-of-transmission interrupt occurs when the peripheral device disconnects. In the extended mode four separate terminal functions can be programmed. Since these interrupts can be selectively armed and disarmed, the program specifies when and if these interrupts occur. The Direct Access Communication Channels always operate in the extended mode as just described. Refer to the applicable technical manuals for the details of the interrupt capabilities of the Time-Multiplexed Communication Channel and the Direct Access Communication Channel. Refer to the reference manual for details of programming in the compatible and extended modes.

1.5.4.3 Real-Time Clock. The Real-Time Clock (RTC) provides a flexible time-orientation system for the 940 Computer. Timing pulses are derived from the 60 Hertz computer power supply. These pulses are then used to produce a timing mark every 16.67 milliseconds (or, optionally, every 8.33 milliseconds). The RTC can also accept timing marks from a customer-designed source, thereby allowing time measurement to any required resolution for special applications. These timing marks are supplied at standard SDS logic levels to the RTC circuitry. The timing marks provide the computer by means of interrupt programming with either an elapsed-time counter or a continuously incrementing time counter depending upon customer needs.

1.6 INPUT/OUTPUT DEVICES

An extensive group of peripheral devices is available for use with the 940 Computer. Input/output devices communicate with the computer either directly or through buffers. The following paragraphs describe the two methods of input/output and the devices handled by both.

1.6.1 BUFFERED INPUT/OUTPUT DEVICES. The speed and flexibility of the 940 Computer buffered input/output ensures the efficient use of all types of peripheral equipment. Buffered peripheral devices include magnetic tape units, keyboard/printers, line printers, card readers and punches, paper tape readers and punches, and auxiliary memory discs as shown in Figure 1-5. Analog/digital conversion equipment and other special system equipment can be added by means of the Data Multiplex System. Since no additional buffering is required for these devices, expansion is quickly and easily performed in the field.

The various input/output devices that are available for the 940 Computer buffered input/output are presented in Table 1-2.

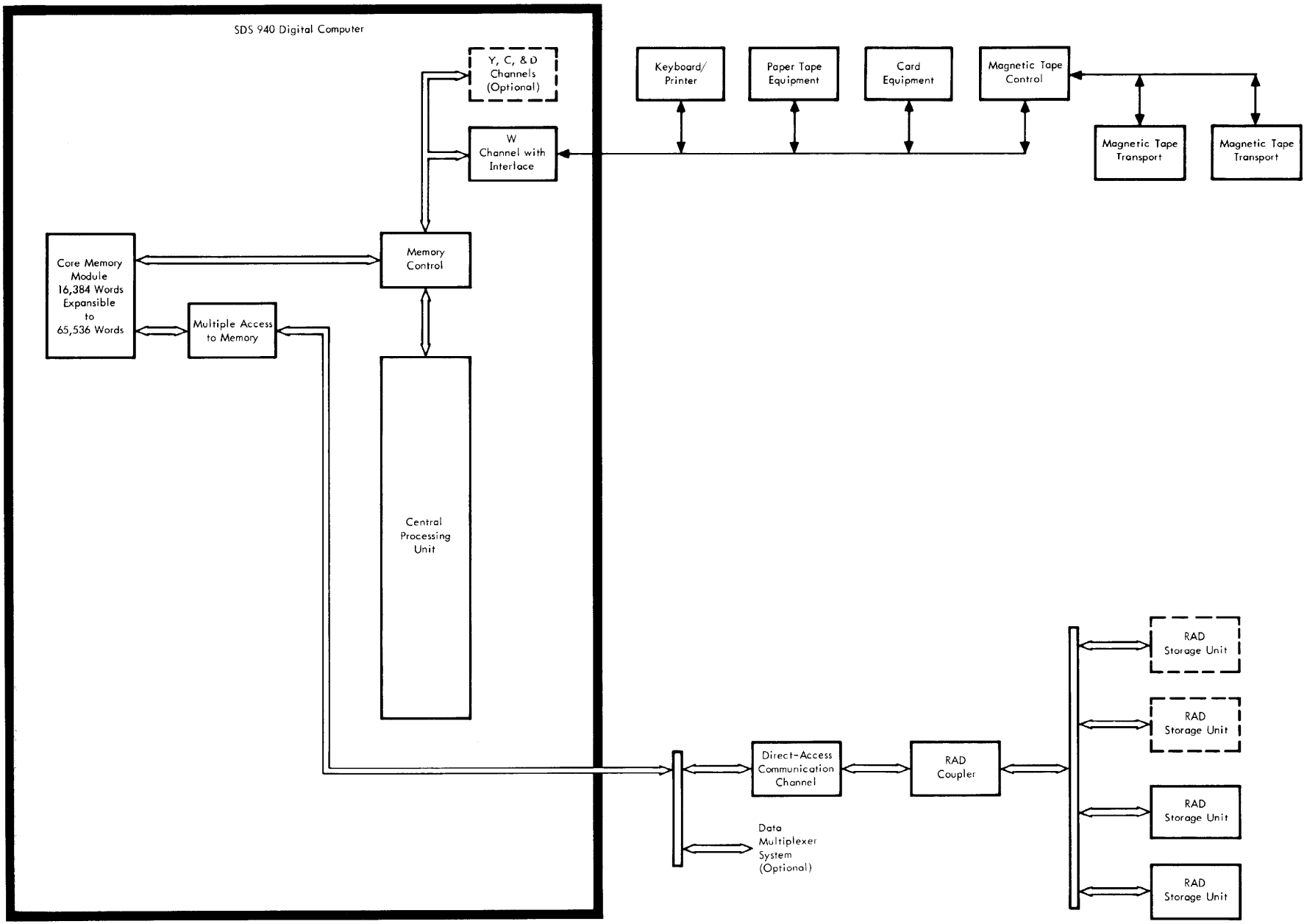


Figure 1-5. Buffered Input/Output

Table 1-2. Buffered Input/Output Equipment

Equipment	Features
Input/Output Typewriter	15 characters per second; 11- or 15-inch carriage
Keyboard/Printers and Low-Speed Paper Tape Equipment	10-characters-per-second Teletype input/output units can include 10-character-per-second tape punch and reader
High-Speed Paper Tape Equipment	300-character-per-second reader; 60-characters-per-second punch; 8-inch NAB reel spooler. Cart-mounting available
Magnetic Tape Systems	7 and 9 tracks; BCD or binary (all compatible with IBM 729 recording mode); 200, 556, and 800-bits per inch recording densities; 60, 75, or 120-inches per second tape velocities; 12,000 to 96,000 characters per second transfer rates
Magpak System	Low-cost, compact magnetic tape system; 4 independent information channels; 6 million character total storage capacity; 1,500 character per second transfer rate
Line Printers	140 to 2,400 line per minute speeds; 12 to 132 print positions, 10, 56 and 64 character sets
Card Equipment	400 and 800 cpm readers; 100 and 300 cpm punches
Mass Random Storage	524,288 to 8,388,608 character rapid-access devices; 8.3 to 67.1 million character mass memory disc files
Graph Plotter	Digital incremental drum plotter; 300 increments per second with 0.01-inch increments
Other Units	Oscilloscope display system

1.6.2 UNBUFFERED INPUT/OUTPUT DEVICES. Unbuffered input/output communications systems are based on the use of two basic units; the asynchronous communications controller and the full-duplex line group. This capability is shown in a typical 940 Configuration in Figure 1-6. The teletype communications system provided with the SDS 940 System permits the transfer of 11-unit, 8-level, 100-words-per-minute teletype information between the 940 Computer and standard Teletype Model 35 KSR or Model 35 ASR units. In addition, an automatic calling unit coupler can be used as described in the Asynchronous Communications Equipment Technical Manual (SDS Publication Number 980111A). This coupler also interfaces with the parallel output and parallel input lines from the computer and uses one level of priority interrupt.

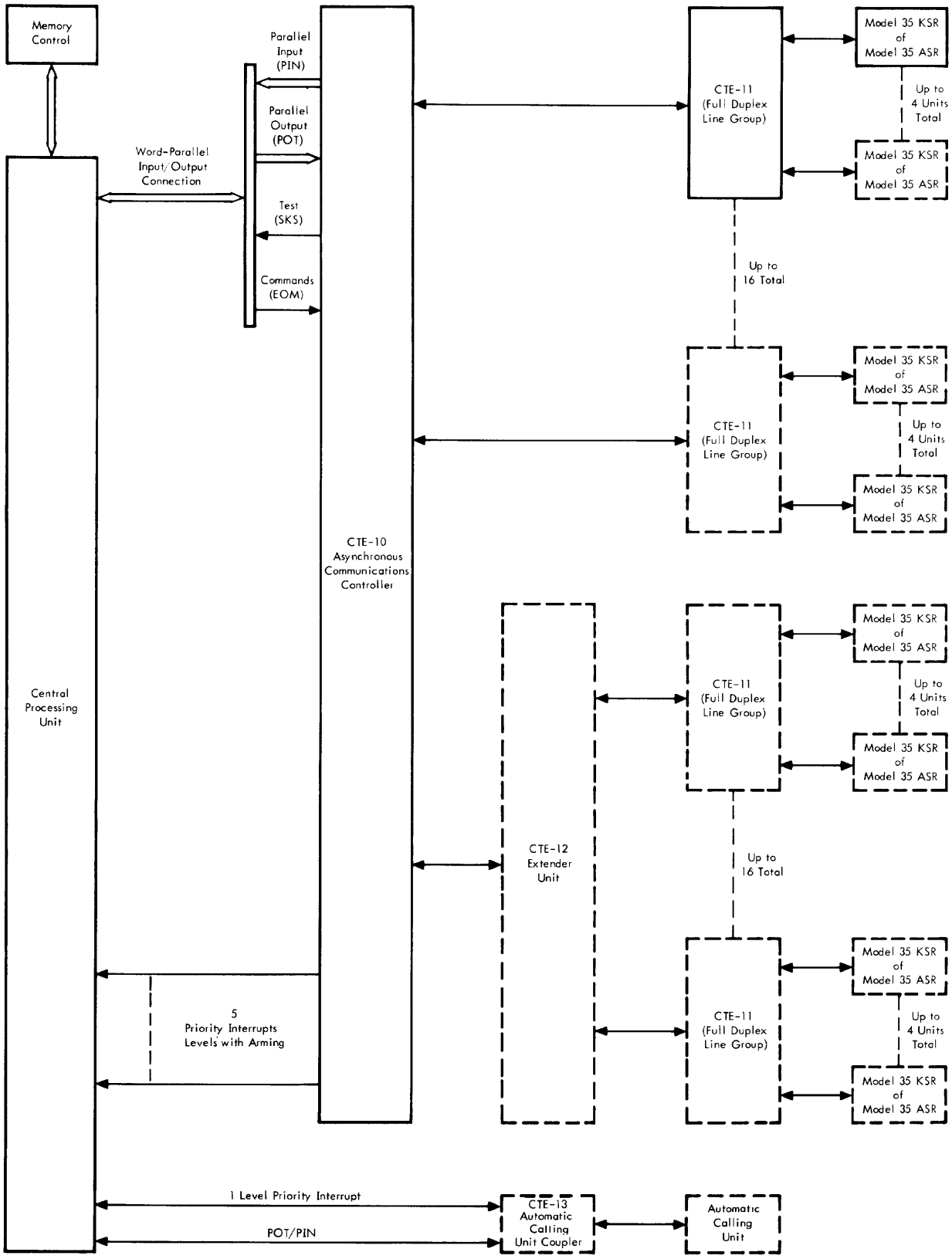


Figure 1-6. Unbuffered Input/Output

When operating with a full complement of 16 line groups, each controller provides full-duplex capability for 64 teletype lines. All lines can be simultaneously engaged in two-way transmission on a completely asynchronous basis. Multiple sets of such configurations can be tied together to form a communications system of any size. The interface of the communications system thus generated, operates through the word parallel system of the computer and uses four interrupt locations with the operation of a fifth location which is end-of-message interrupt.

The number of users simultaneously communicating with the system in synchronism establishes the operating limit. The computer must remove a character from any input buffer before information bits of the next character arrive. For 11-unit, 8-level code, this interval is 27 milliseconds. Assuming that this interval is treated as only 18 milliseconds, the processing time for a character is approximately 300 microseconds. Thus, 60 lines, all practically in exact synchronism, would constitute a worst-case condition beyond which the computer would lose information. Note that in the above example the actual processing time for the 60 characters from 60 separate lines is only 18 milliseconds out of every 100 milliseconds or 18 percent of the processor time, and that rate overruns could only occur if all 60 lines are simultaneously inputting in almost perfect synchronism. These considerations demonstrate that the communication processing limitations do not set the limit on the number of users who can simultaneously use the system. Even with many users, only a small percentage are simultaneously inputting; and they are probably not synchronized with each other. For example, message processing for 32 users is estimated to use about 5 percent of the available computer time.

II. OPERATION AND PROGRAMMING

2.1 GENERAL

In a time-sharing computer, as in any general purpose computer, programs contain the complete sequence of instructions necessary to perform functions or to solve problems. After the operator loads the required programs, sets up the computer, and starts operation, the operation is under control of the stored programs. In this respect, the 940 Computer is almost identical with the 930 Computer. The 940 Computer, however, offers another operation dimension. Any one of a group of remote users can execute instructions in the computer. The 940 Time-Sharing System contains additional hardware/software features that make the computer capability apparently available to each user while preserving the privacy of each user. However, the apparent computer used by the remote users is vastly different from the actual hardware configuration. The reason for this difference is presented in the following paragraphs.

Every multi-programming computer system must provide isolation between independently operating programs because such programs may overwrite each other with data, transfer control to each other, attempt to use the same input/output devices, halt, or otherwise hang up the computer. Isolation problems are usually solved by memory protection, input/output and illegal instruction detection, and system controlled input/output operation. Relocation of program areas and allocation of storage among the concurrent programs become a problem when main memory cannot contain all of the concurrent programs and programs must be moved to and from secondary storage. The mapping feature provides a method for relocation and allocation and also provides memory protection.

In order to insure isolation of user programs, two modes of operation are provided; a user's mode or a systems or monitor mode. The user mode restricts the user to a subset of the 930 Computer instructions. Because both operation modes require changes in programming conventions from the 930 Computer, a third mode called normal or 930 mode is used for normal operation. The normal mode is not used when operating in a time-sharing environment. In the time-sharing environment, a software system, the Executive/Monitor, has complete control over the system.

Only one user can be executing commands at any given instant in time. The Executive/Monitor program activates a user, allows execution of commands for some period, dismisses the user, and activates another user. A user's program runs in the user mode. An input/output operation request produces a mode change to the monitor mode to permit the Executive/Monitor to execute the input/output operation since all input/output operations are done in the monitor mode by the Executive/Monitor program. All interrupts are also processed in the monitor mode. An attempted illegal operation results in a trap condition which in turn results in a transition to monitor mode in which the Executive/Monitor program takes control to perform the necessary action.

Since the scope of this manual does not permit a description of the time-sharing software system, this section describes the instructions added to the 930 Computer that permit computer environment control by time-sharing software. Likewise, the detailed theory of operation section that follows this section only attempts to describe the hardware associated with these added features.

Very few changes have been made to the 930 Computer set of operation codes. With the exception of improved interrupt routine exit and alternate overflow indicator instructions, all instruction changes are associated with the new features; namely, mode changing instructions and the memory map instructions. Numerous logic changes to the 930 Computer exist at the hardware level, even though the changes are not discernible at the operational level. A detailed description of the logic changes is given in Section III of this manual.

Because the 940 Computer is an extension of the 930 Computer, this operation and programming section is limited to a description of mode changes, modes of operation peculiar to the 940 Computer, addressing changes, memory operation modes, available software, and the lists of operation codes and machine instructions.

2.2 CHANGING OPERATION MODES

Initial operation of the START button on the console control panel forces the machine into the normal mode. A program instruction is required to change from normal to monitor mode. No program or manual method can effect a direct transition from normal mode to user mode. Once the monitor mode is entered, the program controls the transition to the user mode. The transition from monitor mode to user mode is made by executing a branch instruction to a user mapped location (any branch instruction in which user relabeling is invoked). The transition from user to monitor mode is made by executing a System Programmed Operator (SYSPOP) instruction. Because no programmed controlled method for transition from monitor or user mode to the normal mode is available, operation of the START switch is required to return to the normal mode.

Two situations occurring in the computer logic will also cause mode transition. First, the occurrence of an interrupt or trap when in user mode will cause a transition to monitor mode. Second, following the execution of a single instruction interrupt routine, a transition to user mode will occur if the machine was in user mode at the time that the interrupt occurred.

2.3 MODES OF OPERATION

This section contains a detailed description of the 940 Computer modes of operation. As indicated previously, the modes are designated normal mode, monitor mode, and user mode. Each of the three operation modes are discussed individually in the paragraphs that follow.

2.3.1 NORMAL MODE. When the START switch is operated, the computer logic automatically enters the normal mode. In the normal mode of operation, the 940 Computer operations are almost identical to those of the 930 Computer. In the normal mode, the 940 Computer is compatible with the 930 Computer software.

The 940 Computer operation in the normal mode is different from the normal 930 Computer operation in four ways. None of these differences effect the operation of the standard 930 Computer software. The four differences include three new instructions and a word modification.

2.3.1.1 Interrupt Routine Return Instruction. A new instruction named Branch and Return from Interrupt Routine (designated by the mnemonic BRI) has been added and is operative in both the normal and monitor mode. The instruction utilizes operation code 11 that is normally undefined in the 930 Computer. The description of the execution of this instruction is presented in the discussion of the monitor mode in Paragraph 2.3.3.5.

2.3.1.2 Overflow Instructions. A new set of instructions to control the operation of the overflow indicator has also been added to the 940 Computer. The overflow indicator detects erroneous arithmetic operations that occur during the execution of a program. The instructions utilize operation code 22, a normally undefined operation code in the 930 Computer. The four new overflow instructions are defined as indicated in Table 2-1.

Table 2-1. Overflow Instructions

Mnemonic	Octal Representation	Full Name
OVT	0 22 00101	Overflow Indicator Test and Reset
REO	0 22 00010	Record Exponent Overflow
ROV	0 22 00001	Reset Overflow Indicator
OTO	0 22 00100	Overflow Indicator Test Only

The function of the first three of these instructions is identical to that of the instruction of the same name in the 930 Computer. In the 930 Computer, OVT (0 40 20001) is an SKS instruction, and REO (0 02 20010) and ROV (0 02 20001) are EOM instructions. In the 940 Computer, these instructions are still operative. Thus, two different methods are provided in the 940 Computer for invoking the execution of the same operation. The OTO instruction functions in the same fashion as the OVT instruction except that the state of the overflow indicator is not modified.

2.3.1.3 Mode Change Instruction. A new instruction has been added to provide a program controlled transition from the normal mode to the monitor mode. The new instruction uses the 02 operation code designated EOM (Energize Output M). The octal form of this new instruction is 0 02 22000.

2.3.1.4 Data Multiplex Channel Interlace Word. The format of the interlace word for the Data Multiplex Channel is different from that of the Data Multiplex Channel interlace word in the 930 Computer. The word count field of the new word contains only eight bits to permit a maximum record size of only 256 words. This word size enables the data address field to contain sixteen bits and permit addressing a 65,536-word memory.

2.3.2 MONITOR AND USER MODE, COMMON OPERATIONS. Several operations are available in both monitor and user modes, but not available in the normal mode. These common operations are described in the following paragraphs.

2.3.2.1 Hang-up Prevention. An Execute instruction (EXU), or a long chain of Execute instructions, is aborted in response to an interrupt request. The interruption assures that interrupt request acknowledgement is not excessively delayed because of a long chain or infinite loop of Execute instructions. This process is effected by terminating the Execute instruction and initiating the execution of a No Operation (NOP) instruction. The highest priority interrupt request is acknowledged, and the Program Counter contains the address of the aborted instruction. Incrementing of the Program Counter was inhibited during the No Operation (NOP) instruction. The normal interrupt routine exit will return to the interrupted instruction that will be restarted. This feature is effective only when mapping through the user map.

Similarly, execution of an instruction involving indirect addressing is interrupted when an interrupt request occurs during the indirect addressing phase of the instruction execution. Also, an interrupt request at the completion of an Increment Index and Branch Instruction (BRX) which calls for a jump is acknowledged. In this case, the Program Counter contains the location specified by the jump. This feature is effective only when mapping through the user map.

2.3.2.2 User Map. A memory mapping technique provides dynamic relocation of programs and fragmentation of memory. In the monitor mode the user map is enabled by individual instructions. The user map converts program addresses to memory addresses. For instance, addresses within the virtual machine in which the user's program is assuming operation are converted to actual physical core memory locations occupied by the user's program and data.

The memory is considered to be divided into 32 pages containing 2048 words each. When mapping occurs, memory is accessed under control of a 5-bit page number and an 11-bit address that specifies a location within the 2048-word page. The upper three bits of a program address constitute the page number. Mapping hardware replaces the user's page number with a physical page number, which may be different from time to time as the program is moved in and out of memory. The users program is not aware of the page structure of the memory. Thus, mapping hardware permits memory fragmentation by allowing user's storage to be located in noncontiguous blocks, that appear to the user and to the machine to be contiguous. The 14 bits of the address field permit only 16,384 words or eight pages to be directly addressable by any user at any one time.

2.3.2.3 Unauthorized Memory Access Protection. The user map provides protection against unauthorized memory accesses. Protection against any access to a 2048 word page of memory is effected by not placing the high order bits associated with the given page in any one of the eight mapping registers. To prevent a running program of fewer than eight pages from accessing a page outside its addressing range, those mapping registers associated with unused and therefore prohibited pages are loaded with an octal 40. If an attempt is made to reference a memory location having an address pointing to a relabeling register with this content, a trap to location 41_8 will result. The term trap is to be distinguished from the interrupt defined by SDS. The trap is a forced transfer to a fixed location; hence a trap routine is interruptible by any other interrupt or trap condition.

At the occurrence of the trap, the Program Counter contains the location of the offending instruction. If the instruction attempted a jump to an out-of-bounds location, the Program Counter contents depend on the status of the address. The notation L : INS A is used for the following information in which L equals the location of instruction, INS is the instruction, A equals the effective address of the instruction before mapping, and P equals the contents of the Program Counter.

L : BRM A	1.	A illegal, (P) = L
	2.	A legal but A + 1 illegal, (P) = A + 1
L : BRR A	1.	A illegal, (P) = L
	2.	(A) + 1 illegal, (P) = (A) + 1
L : POP		(P) = L
L : BRU A		(P) = A
L : BRX A		(P) = A
L : BRI A	1.	A illegal, (P) = L
	2.	(A) illegal, (P) = (A)

An intermediate level of memory protection is afforded by the flag bits. Reading and writing in any assigned block of memory is permitted if the associated flag equals zero. If the flag equals one, and the register contains something other than 40_8, the associated block is read-only. An attempt to store information in a read-only block results in a No Operation (NOP) and a trap to location 43_8. The Program Counter contains the information indicated previously when a jump to an out-of-bounds location was attempted.

2.3.2.4 Modified Unconditional Branch Instruction. The interrupt termination function of the Unconditional Branch Instruction (BRU) is inhibited in both the monitor and user mode.

2.3.2.5 Overflow Instructions. The group of overflow instructions using the operation code 22 are effective in all operation modes. These instructions are described in Paragraph 2.3.1.2.

2.3.3 MONITOR MODE. The several operations unique to monitor mode are described in the following paragraphs.

2.3.3.1 Exclusive Use of User Map. Mapping through the user map becomes effective when the sign bit of the instruction being executed is detected as a one. The machine will invoke the user map for the duration of this instruction. Thus, if the sign bit of a word fetched during indirect addressing is equal to one, all further memory references made by this instruction will be mapped through the user map.

2.3.3.2 Monitor Map. In the monitor mode, if the sign bit of the instruction is zero, the address is mapped through the monitor map. Monitor mapping is one-to-one for all addresses in the range 0 through 30000_8-1. Monitor program addresses in this range always result in equivalent core addresses in this range. For addresses in the range 30000_8 through 34000_8-1, monitor mapping is through the 5 bits contained in M6. For addresses in the range 34000_8 through 37777_8, monitor mapping is through the 5 bits contained in M7. The only access limitation imposed by the monitor map in that physical locations 0 through 3777_8 cannot be accessed using M6 or M7. A zero value in M6 or M7 is reserved to denote an out-of-bounds condition in the same manner as the 40_8 is used in the user map.

In the actual implementation, the sign bit of the instruction is inspected to determine whether or not to go through the user map. If this bit is a zero and the user map is not required, the three high order bits of the address are inspected to see if they contain the values 110 or 111. If they do not, the address is used without mapping. If they contain 110, the five bits of M6 are concatenated with the remaining low order eleven bits of the address to obtain the sixteen bit address of the actual location in memory. If they contain 111, the five bits of M7 are used in a similar fashion. The convention used overrides any use of memory extension register EM3 in the monitor mode. However, EM2 could be employed in this mode if desired.

2.3.3.3 Special Monitor Mode EOM Instructions. A special set of Energize Output M (EOM) instructions has been incorporated to enable entry to the monitor mode. In the monitor mode, a monitor program may exercise environment control. These special instructions in the octal configuration and functions are presented in Table 2-2.

Table 2-2. Monitor Mode EOM Instructions

Octal Configuration	Function
0 02 20400	Clear and select RL1 for loading
0 02 21000	Clear and select RL2 for loading
0 02 21400	Clear and select RL4 for loading
0 02 22000	Perform the transition from normal to monitor mode
0 02 22400	Enable the monitor-to-user transition trap

All of these EOM instructions may also be executed in the normal mode. The first three instructions in Table 2-2 must be followed by a Parallel Output (POT) instruction. The POT words have the structures shown in Figure 2-1.

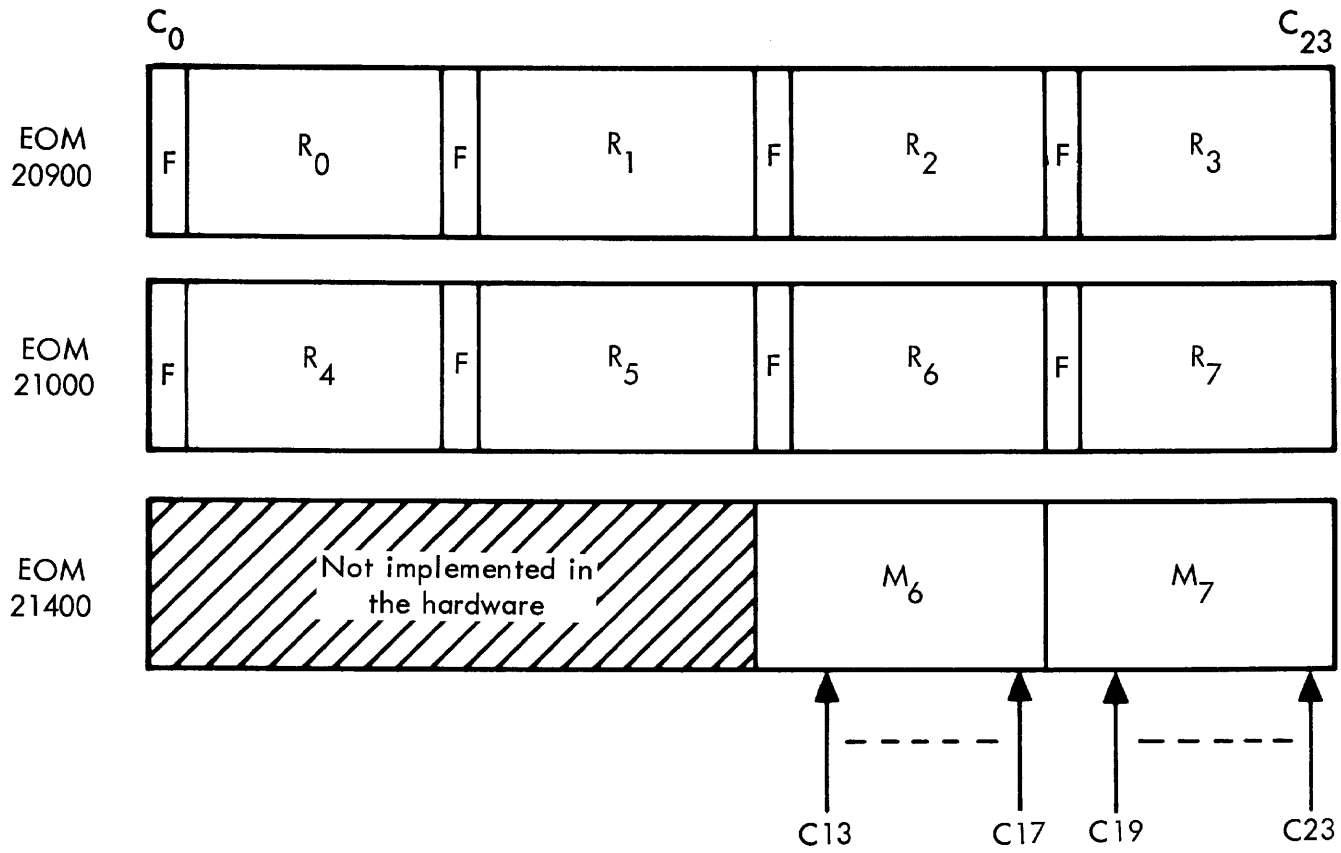


Figure 2-1. Parallel Output Map Word Configurations

2.3.3.4 Branch Instruction Return Address Changes. The return address associated with Mark Place and Branch (BRM), Return Branch (BRR), and Programmed Operation (POP) instructions has two changes.

(a) The overflow indicator is stored in bit two rather than in the sign bit.

(b) The sign bit is set to one to indicate that the return address is to be mapped through the user map. An example of setting the sign bit to a one is the occurrence of an interrupt when the machine is in user mode. In all other cases, the sign bit is set to zero.

2.3.3.5 Other Instruction Operation Changes. The operation of two other instructions has been changed. The Copy Effective Address into Index Register (EAX) instruction and the Branch and Return from Interrupt Routine (BRI) instructions have been changed as described in the following paragraphs.

In addition to performing all of its normal operations, in the monitor mode, the execution of a Copy Effective Address into Index Register (EAX) will also cause the sign bit to be set to zero if the effective address is not mapped through the user map and set to one if the effective address is so mapped.

The operation code 11 has been designated Branch and Return from Interrupt Routine (BRI) and functions in a manner similar to Return Branch (BRR) instruction, with the following exceptions:

(a) The return address is not incremented.

(b) The overflow indicator is first cleared and then set with the contents of bit 2 in the return address word. The BRR operation simply merges the two indicators.

(c) The highest currently active priority interrupt is terminated.

The BRI instruction is effective in both monitor mode and normal mode.

2.3.3.6 Monitor-to-User Transition Trap. A special monitor-to-user transition trap function has been incorporated. For a definition of the term trap, see Paragraph 2.3.2.3. This trap is enabled by a program operating in the monitor mode with an EOM described in Paragraph 2.3.4. The monitor-to-user transition trap is automatically disabled whenever the trap is entered. The trap is also disabled whenever any other trap occurs. If this trap is enabled, the program will trap to location 44_8 whenever a monitor-to-user mode transition is attempted.

2.3.4 USER MODE. The two most distinguishing characteristics of user mode are that memory mapping through the user map is always effective, and that several instructions, called privileged instructions, are not available to the user and will cause a trap if their execution is attempted. Another feature unique to the user mode is an extension of the standard programmed operator mechanism. These characteristics are described in the paragraphs that follow.

2.3.4.1 Privileged Instructions. The set of privileged instructions consists of all undefined order codes, Halt, all input/output instructions, all standard EOM instructions (operation code 02), and all standard sense instructions (operation code 40). An attempt to execute a privileged instruction while in user mode results in the execution of a No Operation (NOP) instruction and, subsequently, a trap to location 40_8. The program counter is not incremented during the execution of the NOP instruction. Consequently, the address stored by the Mark Place and Branch (BRM) instruction in location 40_8 is that of the offending instruction. A list of the privileged instructions are presented in Table 2-3.

2.3.4.2 System Programmed Operators. Input/output operations are performed by the Time-Sharing Monitor program. Many complex services are performed by the executive program. To provide a method of calling for these services as if they were individual instructions, the System Programmed operators feature enables these instructions to be called. The form of programmed operator instruction used with the time-sharing software is designated SYSPOP; therefore, call instructions for services available in the Executive and Monitor programs are made available to the user in the user mode. In the user mode, execution of a programmed operator instruction (POP) with C0 = 0 and C2 = 1 will cause a transfer to the user mapped location designated by C2 through C8 with the link word being stored in user mapped location 0. Also in the user mode, execution of a Programmed Operator instruction with C0 = 1 and C2 = 1 will cause a transfer to the absolute (non-mapped) location designated by C2 through C8 with the

link word being stored in absolute word 0. In addition, a transition from user mode to monitor mode will occur and the sign bit of the return address in absolute word 0 will be set indicating that this address is to be user mapped.

Table 2-3. Privileged Operation Codes

Code	Mnemonic
00	HLT
02	EOM
03	Undefined
04	Undefined
05	Undefined
06	EOD
07	Undefined
10	MIY
11	BRI
12	MIW
13	POT
15	Undefined
21	Undefined
24	Undefined
25	Undefined
26	Undefined
27	Undefined
30	YIM
31	Undefined
32	WIM
33	PIN
34	Undefined
40	SKS
42	Undefined
44	Undefined
45	Undefined
47	Undefined

2.4 ADDRESS CHANGES

The 940 Computer incorporates a number of addressing changes. Addressing in the normal mode is identical to that of the 930 Computer. The user map and monitor maps make possible the generation of 16-bit addresses and permits core memory sizes of up to 65,536 words in a 940 Computer System. Addressing in the 940 Computer is expanded to permit transmission of 16-bit addresses. The 940 Computer core memories accept and decode 16-bit addresses. Time-Multiplexed Communication Channels, Direct Access Communication Channels, the Data Multiplexing System, and all Multiple Access to Memory units and Memory Interface Connections accept 16-bit addresses and/or transmit 16-bit addresses to memory during their operation.

2.5 MEMORY INTERLEAVING

A memory interleaving feature is incorporated in the 940 Computer to permit more efficient system operation while performing input/output and compute overlap. The memory configuration of each particular system will determine the method of interleaving used.

The desired mode of memory operation is selected by switches on the memory cable plug modules.

The 940 Core memories may be operated as

(a) Non-interleaved

(b) Two (2) way interleaved

(c) Four (4) way interleaved

The 940 Core memories also include provisions for proper interaction with the Direct Access Communication Channel. This interaction permits the Central Processor to have access priority over the Direct Access Communication Channel to the maximum extent possible without causing data overruns.

2.6 SOFTWARE

In the design of a computer system, efficient man-machine communication is most important. In addition to the operator's console, which is especially designed for operator convenience, man-machine communication is facilitated at every programming level by a comprehensive package of automatic programming systems and aids.

A complete library of programming systems is supplied with the 940 Computer, incorporated under the Berkeley Time-Sharing Software System. The system is an integrated set of software that uses the latest concepts of interactive multi-programming. Maximum use is made of re-entrant processes and common routines. Being a generalized system, user operations are permitted in languages ranging from a machine-oriented assembly language through a FORTRAN compiler to a sophisticated list processor. The software provides maximum responsive service to the user and operating efficiency. The available time-sharing software is listed in Table 2-4.

Table 2-4. Time-Sharing Software

Program	Description
Time-Sharing Executive	Complete bookkeeping facilities for file storage and retrieval from secondary memory, usage accounting, and file security.
Time-Sharing Monitor	All input/output for user programs, scheduling of user program operations, program error processing, and program-to-program communication.
String Processing System	Subsystem that performs basic operations on strings of characters; reading and writing, hash-code string look-up, and string comparison.
Conversational Algebraic Language (CAL)	Relieves storage allocation burden for both programs and data and provides a problem-oriented language for conversational use.
Fortran II	Accepts symbolic source-language input created on-line by the Time-Sharing System Text Editor (QED) achieving an on-line compile-execute edit-compile cycle.
Time-Sharing System Text Editor (QED)	Enables creation and modification of symbolic text for any purpose.
Symbolic Macro-Assembler	A two-pass assembler with subprogram, literal, and macro facilities.
On-Line Debugging Package (DDT)	Permits user to examine, search, change, insert breakpoint instructions, and step-trace the program at the symbolic level. The command language is geared to rapid interactive operation by the on-line user.
Symbol-Manipulating Language (LISP)	Permits non-numeric application and logical analysis in a language closer to the user's problem language than the normal input form.
String Programming Language (SNOBOL)	Provides complete facilities for manipulation of strings of characters.
Assistance Program (HELP)	Provides on-line question answering service for use by the time-sharing Executive and other subsystems. Includes direct self-teaching facility in natural language.

2.7 SDS 940 COMPUTER OPERATION CODES

The operation code is that part of the instruction that defines the operation to be performed. The 48 operations performed by the 940 Computer are identified by their octal designation and their mnemonic in Table 2-5. Description of the operation, word formats, and timing of the operation codes is provided in the 940 Computer Reference Manual.

Table 2-5. 940 Operation Codes

Octal Designation	Mnemonic	Name
00	(HLT)	Halt
01	(BRU)	Branch to M
02	EOM	Energize Output M
03		
04		
05		
06	EOD	Energize Output D
07		
10	(MIY)	(M) to Y when ready
11	BRI	Branch and Return from Interrupt Routine
12	(MIW)	(M) to W when ready
13	POT	Parallel Output
14	(ETR)	Extract
15		
16	(MRG)	Merge
17	(EOR)	Exclusive OR
20	(NOP)	No Operation
21		
22		Overflow Instructions
23	(EXU)	Execute
24		
25		
26		

Table 2-5. 940 Operation Codes (cont)

Octal Designation	Mnemonic	Name
27		
30	(YIM)	Y to M when ready
31		
32	(WIM)	W to M when ready
33	PIN	Parallel Input
34		
35	STA	Store A
36	STB	Store B
37	STX	Store X
40	SKS	Skip if signal not set
41	BRX	Increment Index and Branch
42		
43	BRM	Mark Place and Branch
44		
45		
46		Blank Register Change Instructions
47		
50	(SKE)	Skip if (A) = (M)
51	BRR	Return Branch
52	(SKB)	No Skip if (B) (M) = 1
53	(SKN)	Skip if (M) negative
54	(SUB)	(A) - (M) to A
55	(ADD)	(A) + (M) to A
56	(SUC)	(A) - (M) - Carry to A
57	(ADC)	(A) + (M) + Carry to A
60	(SKR)	(M) - 1 to M, Skip if negative
61	(MIN)	(M) + 1 to M
62	(XMA)	Exchange (A) with (M)

Table 2-5. 940 Operation Codes (cont)

Octal Designation	Mnemonic	Name
63	(ADM)	(A) + (M) to (M)
64	(MUL)	Multiply
65	(DIV)	Divide
66		Blank Shift Right Instructions
67		Blank Shift Left Instructions
70	(SKM)	Skip if (A) = (M), B mask
71	LDX	Load X
72	(SKA)	No skip if (A) (M) = 1
73	(SKG)	Skip if (A) > (M)
74	SKD	Difference Exponents and Skip
75	LDB	Load B
76	LDA	Load A
77	EAX	Copy Effective Address into X

2.8 SDS 940 Mnemonic Instruction List

Assembly systems recognize mnemonics and perform the conversion to the proper octal configuration of the operation code. Table 2-6 is a list of the standard 940 Computer mnemonics.

Table 2-6. Instruction List

Mnemonic	Name	Timing
	Register Change	
CLA	Clear A	1
CLB	Clear B	1
CLR	Clear AB	1
CAB	Copy A into B	1
CBA	Copy B into A	1
XAB	Exchange A and B	1

Table 2-6. Instruction List (cont)

Mnemonic	Name	Timing
	Register Change (cont)	
BAC	Copy B into A, Clear B	1
ABC	Copy A into B, Clear A	1
CLX	Clear Index Register	1
CXA	Copy Index into A	1
CAX	Copy A into Index	1
XXA	Exchange Index and A	1
CBX	Copy B into Index	1
CXB	Copy Index into B	1
XXB	Exchange Index and B	1
STE	Store Exponent	1
LDE	Load Exponent	1
XEE	Exchange Exponents	1
CNA	Copy Negative into A	1
CLEAR	Clear A, B, and Index	1
	Memory Extension	
	Set Extension Register	1
	Extension Register Test	2, 3
	Overflow	
OVT	Overflow Indicator Test and Reset	1, 2
OVT	Overflow Indicator Test and Reset	1, 2
ROV	Reset Overflow Indicator	1
ROV	Reset Overflow Indicator	1
REO	Record Exponent Overflow	1
REO	Record Exponent Overflow	1
OTO	Overflow Indicator Test Only	1, 2

Table 2-6. Instruction List (cont)

Mnemonic	Name	Timing
	Interrupt	
EIR	Enable Interrupt System	1
DIR	Disable Interrupt System	1
IET	Interrupt Enabled Test	1, 2
IDT	Interrupt Disabled Test	1, 2
AIR	Arm Interrupts	1
	Channel Control[1]	
ALCW	Alert Channel W	1
DSCW	Disconnect Channel W	1
ASCW	Alert to Store Address in Channel W	1
TOPW	Terminate Output on Channel W	1
[1]Channels Y, C, D, E, F, G, and H are Coded 1 through 7 Respectively for Channel Control		
	Channel Test[2]	
CATW	Channel W Active Test; Skip if Channel Inactive	2, 3
CETW	Channel W Error Test; Skip if No Error	2, 3
CITW	Channel W Inter-Record Test	2, 3
CZTW	Channel W Zero Count Test; Skip if Count Equals Zero	2, 3
[2]Channels Y, C, D, E, F, G, and H are Coded 1 through 7 Respectively for Channel Test		
	Shift	
RSH	Right Shift AB	2-7
RCY	Right Cycle AB	2-7
LRSH	Logical Right Shift AB	2-7
LSH	Left Shift AB	1, 2
LCY	Left Cycle AB	1, 2
NOD	Normalize and Decrement Index	1, 2

Table 2-6. Instruction List (cont)

Mnemonic	Name	Timing
	Breakpoint Tests	
BPT1	Breakpoint No. 1 Test	1, 2
BPT2	Breakpoint No. 2 Test	1, 2
BPT3	Breakpoint No. 3 Test	1, 2
BPT4	Breakpoint No. 4 Test	1, 2
	Buffer Test	
BETW	W Buffer Error Test	1, 2
BETY	Y Buffer Error Test	1, 2
BRTW	W Buffer Ready Test	1, 2
BRTY	Y Buffer Ready Test	1, 2
	Monitor and User Map	
	Clear and Select RL1 for loading Clear and Select RL2 for loading Clear and Select RL4 for loading	
	Mode Transition	
	Perform transition to Monitor Mode Enable the Monitor-to-User Transition Trap	

III. THEORY OF OPERATION

3.1 GENERAL

This section presents a technical description of the differences between the SDS 940 Computer and the SDS 930 Computer. Computer organization is described briefly followed by a detailed description of the logic changes required to adapt the SDS 930 Computer to the time-sharing atmosphere. At the conclusion of this section, a glossary of logic terms and the equations peculiar to the SDS 940 Computer is presented.

Since this section describes the logic for the differences between the 940 Computer and the 930 Computer, only those illustrations and equations necessary for understanding the text are presented. The glossary and equations at the end of the section are for the difference between the 930 Computer and the 940 Computer. In the event that repairs are required, the reader will want to locate the complete logic schematic and the complete equations. For this reason, terminology is the same as that used in the complete schematics and wire lists. To assist the reader in understanding the logic symbols and equations used in this section as well as those used on the logic layout sheets and wire lists, the following paragraphs briefly describe the logic symbology used.

The equations used in the text conform to the following conventions. Logic inversion of a signal designator is denoted by a bar over the signal designation. For example, an inversion of RT would be $\overline{RT}$. The OR function is noted by a + sign between two signal designators. For example, A OR B is noted A + B. The AND function is indicated by a simple conjunction of signal designators or use of parenthesis between two signal designators. For example, A AND B are noted AB or (A)(B). Logic signal designators are introduced as required in the text.

3.2 MACHINE LANGUAGE

Communication within the computer is performed by manipulation of all or part of a 24-bit word. Each word can contain instructions or data. At the beginning of a machine instruction, the C-register contains an instruction. During a machine instruction, the C-register contains either data or the instruction. At the conclusion of the instruction, the C-register contains the next instruction to be performed.

3.2.1 INSTRUCTION WORD FORMAT. The instructions are in the form of 24-bit words. The 940 Computer instruction word contains 24-binary digits or bits, numbered from 0 through 23 as shown in Figure 3-1. Arithmetic is performed using the binary, two's complement number system. The sign bit can be considered to be integral with the data word. For simplicity of description, the 940 Computer instruction word is written in octal notation, where each numeral represents three binary digits. The function of the various parts of the instruction word are described in the following paragraphs.

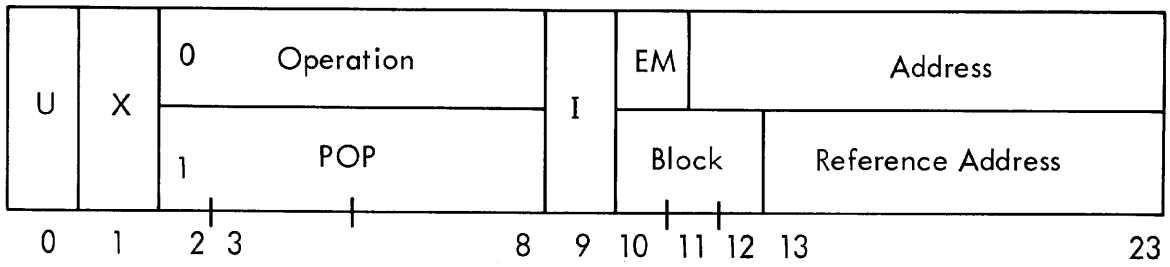


Figure 3-1. Instruction Word Format

3.2.1.1 Bit Position Zero. The function of this bit is completely different in the 930 mode and the 940 modes. The 930 Computer Reference Manual describes the use of bit position zero in the 930 mode. The use of this bit in the 940 modes is covered in Paragraph 2.3.4.2 of this manual.

3.2.1.2 Index Register Bit. A one in bit 1 causes the contents of bits 10 through 23 of the index register (X-register) to be added to the address portion of the instruction. The instruction in memory remains unchanged.

3.2.1.3 Programmed Operator Bit. The contents of bit position 2 (the programmed operator bit) determines the method of interpreting bits 3 through 8. If bit 2 contains a zero, bits 3 through 8 are decoded as a normal instruction. If bit 2 contains a one, bits 3 through 8 are used to determine a subroutine entrance address. If in the user mode, bit 0 is also used by the logic to determine the exact course of action.

3.2.1.4 Indirect Address Bit. A one in bit 9 causes the computer to interpret bits 10 through 23 (possibly modified by indexing) as the memory location where the effective address of the instruction may be found. A zero in bit 9 causes bits 10 through 23 (possibly modified by indexing) to be interpreted as the effective address of the instruction. Mapping may apply, beginning at any level.

3.2.1.5 Address Bits. Bits 10 through 23 normally determine the memory address referenced by the instruction code. When the address value in bit positions 10 through 23 is 8000 or greater, bit positions 10 and 11 select the appropriate memory extension register (EM2 or EM3) in the 930 mode. In the 940 modes, however, bit positions 10 through 12 constitute a virtual memory page number within the physical memory if mapping is involved.

3.2.2 DATA WORDS. Data words also contain 24 binary bits. However, two different word formats are used for data words as described in the following paragraphs.

3.2.2.1 Fixed Point Data Word. Fixed point data words contain 24 binary digits (eight octal digits), with the sign incorporated as the leading bit in the most significant octal digit. The fixed point data word format is shown in Figure 3-2.

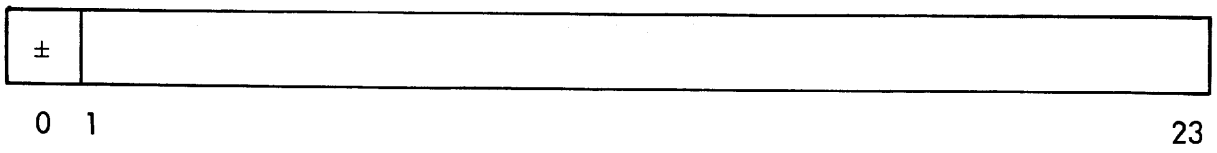


Figure 3-2. Fixed Point Data Word

3.2.2.2 Floating Point Data Words. Floating-point data words have the format shown in Figure 3-3 for double-precision operation. The fractional portion of a double-precision floating-point number is a 39-bit proper fraction, with the leading bit being the sign bit, and with the assumed binary point just to the left of the most significant magnitude bit. The floating-point exponent is a 9-bit integer, with its leading bit as the sign bit. Standard routines operate on both the fraction and exponent in two's complement form.

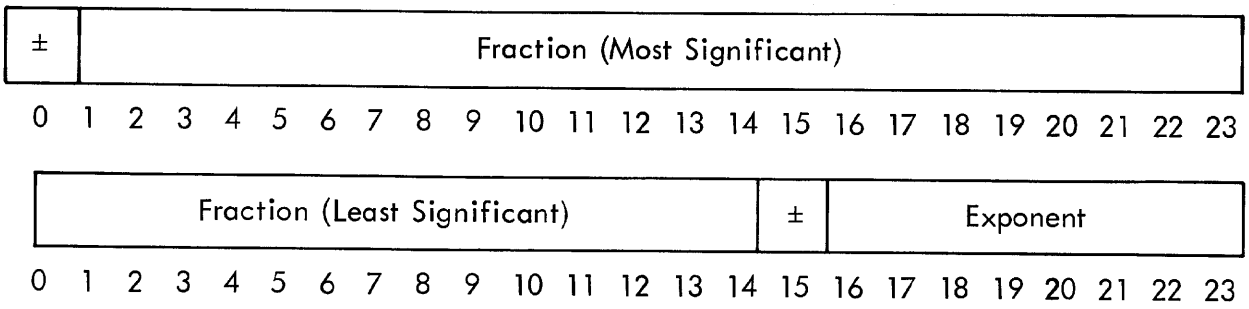


Figure 3-3. Floating Point Data Words

3.3 DESCRIPTION OF COMPUTER OPERATION

All operations begin with the instruction in a Central Transfer Register. From the Central Transfer Register, the fields of the instruction word are distributed as follows. The six bit instruction code field is transferred in parallel to a six flip-flop Instruction Code Storage Register where the contents will be used to control execution of the instruction. The instruction is octally shifted out of the Central Transfer Register through a Serial Octal Adder, and back into the Central Transfer Register. As the address field passes through the Serial Octal Adder, the address may be combined with the contents of an Index Register if the index flip-flop has been set by the index register bit 1. After returning to the Central Transfer Register from the Serial Octal Adder, the address field is transferred in parallel to a Memory Address Register and mapping may or may not be used. The memory is cycled. A Memory Data Register accepts the operand from memory and transfers the operand in parallel to the Central Transfer Register. At this point, the code of the operation to be performed is held in the Instruction Code Storage Register and the operand, upon which the operation is to be performed, is held in the Central Transfer Register.

During the execution of the instruction, the operand is octally shifted out of the Central Transfer Register and into the Input/Output Register, Accumulator, Extended Accumulator, or Index Register depending on the operation to be performed.

Subsequent operation is under control of a phase counter. Toward the end of each operation as defined by the contents of the Instruction Code Storage Register, the Instruction Address Counter is incremented. The Instruction Address Counter holds the 14-bit address of the next instruction to be executed. The address held in the Instruction Address Counter is transferred in parallel to the 14 flip-flops of the Memory Address Register that sets up the address lines in the core memory for access to the word position addressed. (The mapping feature may, or may not be used.) The contents of the desired word are then transferred in parallel with a parity bit to the 25 flip-flop Memory Data Register. During memory accessing, or storing, the Memory Address Register will hold the desired address, that may or may not be modified by the map, and the Memory Data Register will contain the operand or accept the operand from memory. After being received from memory, the 24-bit instruction in the Memory Data Register, is transferred in parallel to the Central Transfer Register. The instruction terminates with the next instruction in the Central Transfer Register and the unrelabeled address of that instruction in the Instruction Address Counter.

It should be clearly understood that all mapping occurs between the memory address register and memory if the mapping feature is to be invoked. A logical description of this is covered in Paragraphs 3.5.5 and 3.5.6 of this section.

3.4 LOGIC ORGANIZATION

The computer comprises a central processor and a memory. The central processor is capable of a certain number of operations and the memory is the storage device for the instructions and data that the central processor needs to perform the required operations. With the exception of mapping, the 940 Central Processor is organized essentially the same as a 940 Central Processor. The differences between the 940 Central Processor and the 930 Central Processor are found in this manual. The 940 Memory is essentially the same as a 930 Memory except for size and the added mapping feature. The core memory varies in size from 16, 384 words to 65, 536 words.

3.5 940 LOGIC

The logic required to adapt the 930 Computer to the time sharing environment includes: methods to enable certain programmable mode changes, special interrupt logic, logic to modify and add to existing instructions, logic for instructions required by the relabeling registers of the monitor and user maps, the logic of the monitor and the user maps, and memory logic changes. Each of these topics is described individually in the paragraphs that follow.

3.5.1 MODE CHANGES. Additional logic is required to enable changing the 940 Computer from one mode of operation to another under controlled circumstances. The mode of operation

is determined by the state of flip-flops Nm (Normal Mode) and Md (Mode). With the Nm flip-flop set, the computer is in the normal mode. The Nm flip-flop is set by the START switch on the control console.

$$sNm = St$$

The Nm flip-flop must be reset to enable the monitor or user mode of operation. The Nm flip-flop is reset at time T3 when an EOM instruction is generated through the user map.

$$rNm = EOM\ \overline{C10}\ C11\ C13\ \overline{C14}\ C15\ T3$$

With the Nm flip-flop reset, the mode of operation of the computer is determined by the mode flip-flop (Md). The computer is in the monitor mode if the Md flip-flop is set or the user mode if the Md flip-flop is reset.

The following paragraphs describe the logic required for the normal to monitor mode shift, the monitor to user mode shift, and the user to monitor mode shift, respectively. The only method for returning to the normal mode from the monitor or user mode is to set the Nm flip-flop with the START switch on the control panel as previously described.

3.5.1.1 Normal to Monitor Mode Change. The only mode transition possible from the normal mode is into the monitor mode because of the method of setting and resetting the mode flip-flops. When the Nm flip-flop resets, the Md flip-flop sets.

$$sMd = rNm + \cdots$$

Thus, if the Md flip-flop was reset when the Nm flip-flop was reset the Md flip-flop would set and the computer would be in the monitor mode. With Nm false and Md true, the Sm signal is set true and the computer is in the monitor mode.

$$Sm = \overline{Nm}\ Md$$

The monitor mode is sometimes referred to as the system mode.

3.5.1.2 Monitor to User Mode Change. The transition from monitor to user mode is made by resetting the Md flip-flop. With Md and Nm reset, the Um signal is true and the computer is in the user mode.

$$Um = \overline{Nm}\ \overline{Md}$$

The reset of Md is normally accomplished by using the signal MUT which indicates, when in the monitor mode, a jump to a relabeled address.

$$rMd = MUT + \cdots\cdots$$

The MUT signal is set during a Return Branch (BRR) instruction and a Branch and Return from Interrupt (BRI) instruction as follows.

$$MUT = Rb\ End\ \overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6 + \cdots\cdots$$

The Rb flip-flop is set if the current memory cycle is to be relabeled.

$$sRb = T3\ Rbs\ \overline{Int}$$

The Rbs term indicates that the upcoming memory cycle is to be relabeled through the user map.

During a Branch Unconditionally (BRU) instruction or an Increment Index and Branch (BRX) instruction, the MUT signal is set as follows.

$$MUT = Rb\ Ju\ \overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6 + \ldots$$

During a Mark Place and Branch (BRM) instruction, the MUT signal is set as follows.

$$MUT = Rb\ \phi 0\ TR\ MXC + \ldots$$

Resetting of the Md flip-flop can also be accomplished with the signal Usi Ib T4) generated by the occurrence of a single instruction interrupt while in the user mode.

$$rMd = (Usi\ Ib\ T4) + \ldots$$

This signal resets Md at the end of the single instruction interrupt to return to the user mode.

3.5.1.3 User to Monitor Mode Change. A transition from user mode to monitor mode will occur if any one of the following events occur while the computer is operating in the user mode.

(a) An interrupt

(b) A trap

(c) An SYSPOP

The logic required to return from user to monitor mode is the setting of the Md flip-flop. The Md flip-flop is set by UMT for an interrupt or a trap occurrence.

$$sMd = UMT + \ldots$$

Where

$$UMT = Um\ Int + Um\ TRAP\ T3 + Int\ Rbs\ T3$$

The (INT Rbs T3) term is active when an interrupt occurs during a transition from monitor to user mode. This occurrence will inhibit the transition until the interrupt is processed. The

interrupt return will be into the user mode because bit zero of the link word has been set to a one by Usi. The Usi flip-flop is set when a transition from user to monitor mode occurs or is soon to occur.

$$sUsi = UMT + \ldots$$

The output of Usi is used to copy the mode indicator into the zero bit position of the return link location. The output of Usi is set into the C0 bit position during the execution of the Mark Place and Branch (BRM) instruction in the interrupt or trap location. The overflow indicator is also stored into the C2 bit position at this time.

The Md flip-flop is also set when a SYSPOP is executed.

$$sMd = (T3\ Usi\ Go) + \ldots$$

In the execution of the SYSPOP, the Usi flip-flop is set by the following equation.

$$sUsi = (C0\ C2\ \phi 0\ T8\ \overline{Ia})\ Um + \ldots$$

3.5.2 TRAPS. The term trap means a forced transfer to a fixed location. Although a trap results in an operation similar to an interrupt, a trap is interruptible by an interrupt or another trap. There are four traps implemented in the 940 Computer as summarized in Table 3-1.

The trap locations will normally contain a BRM (43) instruction to the trap subroutine. The logic for each trap condition is described in the paragraphs that follow.

Table 3-1. Traps

Name	Description
Privileged Instruction	A trap to memory location 40_8 occurs if an attempt is made to execute one of the privileged instructions in the user mode.
Out-of-Bounds	A trap to memory location 41_8 occurs if an access to an out-of-bounds memory address is attempted. The condition results from selecting a relabeling register containing 40_8.
Read-Only	A trap to memory location 43_8 occurs when an attempt to write into a read only block of memory occurs.
Monitor to User	If enabled, trap to memory location 44_8 will occur when the monitor to user transition occurs. This trap is enabled by EOM 22400 and is disabled by the occurrence of any trap.

3.5.2.1 Privileged Instruction and Monitor to User Transition Traps. The conditions that result in a privileged instruction or monitor to user transition trap are detected at time T8 of Phase 0 and indicated by a true PiQ signal.

$$PiQ = Pid\ Go\ Um\ \overline{Ob}\ \overline{Ai}\ \overline{C2}\ \phi 0\ T8\ \overline{Ia}\ \text{(Privileged Instruction)}$$
$$+\ \overline{Ob}\ Go\ Um\ Tte\ \phi 0\ T8\ \overline{Ia}\ \text{(Transition Trap)}$$
$$+ \ldots$$

The Pid term is the privileged instruction decode and the Tte term is the output of a flip-flop that enables the transition trap.

The contents of the Instruction Code Storage Register (O-register) at time T8 of Phase 0 is 20_8 or a No Operation instruction (NOP) since the Oc term, reset signal clears the O-register to the 20_8 condition. The Oxc term which enables the transfer of data from the Central Transfer Register (C-register) to the O-register is inhibited by PiQ.

$$Oxc = (\phi 0\ T8\ \overline{Ia})\ \overline{C2}\ Go\ (\overline{Ob}\ \overline{Ai}\ \overline{PiQ})$$

Since the trap condition inhibits Oxc, the contents of the C-register cannot be transferred to the O-register and the NOP instruction in the O-register is executed.

During the execution of the NOP instruction, incrementing of the Instruction Address Counter (P-register) is inhibited. (See Paragraph 3.5.3.) The transfer from P-register to Memory Address Register (S-register) transfer is also inhibited.

$$Sxp = End\ \overline{TRAP}\ \overline{Int}\ T3\ Go + \ldots$$

At time T3 of the NOP instruction, the trap location is forced into the S-register as follows. The privileged instruction flip-flop is set by PiQ.

$$sPi = PiQ + \ldots$$

The TRAP term is generated by Pi.

$$TRAP = Pi + \ldots$$

The S-register is then set to the trap address as follows. If the privileged instruction trap occurs, the S9 flip-flop is set (40_8 memory address).

$$sS9 = TRAP\ T3 + \ldots$$

If the transition trap has occurred, the S9 flip-flop is set and the S12 flip-flop is also set (44_8 memory address).

$$sS12 = \overline{Ob}\ Pi\ Tte\ T3 + \ldots$$

The PiQ term also sets the phase counter to Phase 5.

$$sF1 = PiQ + \ldots$$

$$sF3 = PiQ + \ldots$$

At the time T8 of the following Phase 0, the BRM instruction in the trap location starts execution with the location of the attempted instruction in the P-register.

3.5.2.2 Out-of-Bounds Trap. The out-of-bounds trap condition is detected in either Phase 0 or during END at T2 time. The flip-flop Ob is set to indicate an attempted out-of-bounds memory access.

$$sOb = Oba + \ldots$$

Where

$$Oba = \Big[(SFM\ \overline{LS00A'}\ \overline{RDS}\ T2)(Ju\ O1\ \overline{O5}\ \overline{XW1})(\overline{\phi 0\ END})\ \overline{S3'}\ Go\Big]\ \overline{LSOA'}\ \overline{LS1A'}\ \overline{LS2A'}\ (SEL'6 + SEL'7 + REL)$$

At either time, when an out-of-bounds condition is detected, the O-register must be set to 20_8 in order to execute a forced NOP instruction.

The Oc reset signal clears the O-register to the 20_8 condition.

$$Oc = Tp\ (Ob + Ai) + \ldots$$

The NOP instruction in the O-register is then executed.

During execution of the NOP instruction, the P-register is not permitted to increment as described in Paragraph 3.5.3. At time T3 of the NOP instruction, the P-register to S-register transfer is also inhibited.

$$Sxp = End\ \overline{TRAP}\ \overline{Int}\ T3\ Go$$

Instead, the trap address is set into the S-register as follows. The TRAP term was generated by Ob as soon as the out-of-bounds trap was detected.

$$TRAP = Ob + \ldots$$

The S9 flip-flop is set at time T3 by TRAP.

$$sS9 = TRAP\ T3 + \ldots$$

The S14 flip-flop is also set.

$$sS14 = Ob\ T3 + \ldots$$

At time T8 of Phase 0 of the NOP instruction caused by an out-of-bounds trap condition, F1 and F3 are set to move the phase counter to Phase 5.

$$sF1 = T8\ (Ob + Ai) + \ldots$$

$$sF3 = T8\ (Ob + Ai) + \ldots$$

At time T8 of the following Phase 0, the BRM instruction in the trap location starts execution with the location of the attempted instruction in the P-register.

The out-of-bounds detection mechanism is inhibited for a BRX instruction that does not branch and for the superfluous memory access made by the EAX instruction since no problems are incurred which would effect the privacy of the memory location and the trap would not be an indication of a true memory violation.

3.5.2.3 Read-Only Trap. The read-only trap condition is detected in Phase 4 at T2 time. Flip-flops Ob and Pi are both set by STV.

$$Pi = STV + \ldots$$

$$Ob = STV + \ldots$$

The STV signal indicates an attempt to store information into a read-only memory location.

$$STV = \overline{OBA}\ SFM\ REL\ MXC$$

The phase counter is inhibited by Ob from advancing from Phase 4 to Phase 7 and is instead reset to Phase 0. In Phase 4, F1 is true, F2 and F3 are reset; Ob inhibits setting F2 and F3 and also resets F1.

$$rF1 = Tp\ Ob + \ldots$$

The Ob term also inhibits MXC, the signal normally sent to memory indicating new information is to be written into memory. The NOP operation code is forced into the O-register in the same manner as in the out-of-bounds condition previously described in Paragraph 3.5.2.2. The TRAP term was generated by Ob upon detection of a read-only condition.

$$TRAP = Ob + \ldots$$

The TRAP term is used to set the S9 flip-flop.

$$sS9 = TRAP\ T3 + \ldots$$

The S13 and S14 flip-flops are also set.

$$sS13 = Ob\ Pi\ T3 + \ldots$$

$$sS14 = Ob\ T3 + \ldots$$

The Pi flip-flop was set only to differentiate between the out-of-bounds and read-only conditions.

3.5.2.4 Read-Only or Out-of-Bounds - Additional Problems. A Mark Place and Branch instruction (BRM, Operation Code 43) with an effective address that is read-only or out-of-bounds presents some additional problems. Because the address portion of the C-register and the contents of the P-register are interchanged during Phase 0 before the trap condition is detected at time T2 of Phase 0, the C-register and P-register must be interchanged again during the forced NOP instruction. This operation is performed as follows. The illegal memory address is detected in the usual manner and either Ob or Ob and Pi are set. The Oc term is generated to set the O-register to a NOP operation code. However, O6 is also set.

$$sO6 = \overline{TSM}\ (JU\ \ O1\ \ O5)\ \ Ob\ \ Oc + \ldots$$

The O6 term is used in conjunction with Ob to generate Obr.

$$Obr = Ob\ \ O6$$

The Obr term is used to gate the data from the C-register into the P-register during the NOP instruction. The Cr3 term is also generated to shift the C-register.

$$Cr3 = \phi 5\ \ Ob\ \ O6\ (\overline{Ts}\ \ Q1) + \ldots$$

The Pr3 term is normally generated by a NOP instruction and needs no special implementation.

The contents of the registers are returned during the NOP and, at T3 time of Phase 5, the trap address is forced into the S-register as previously described in Paragraph 3.5.2.2.

The timing diagram in Figure 3-4 shows an attempted BRM instruction to an out-of-bounds location. The solid lines show what actually happens. The dotted lines show what would happen if MXC, Oxc, $\phi 6$, and sIa were not inhibited by Ob.

3.5.3 SPECIAL INTERRUPT FEATURE. The allow interrupt feature prevents a user from hanging up the machine and avoids excessive delays in processing interrupt requests. This feature is not used in the normal mode.

An allow interrupt flip-flop (Ai) is set at the T0 time of Phase 0 whenever an interrupt request is pending and the machine is determining an effective address through the indirect address mechanism.

$$sAi = \left[(En + \textcircled{En})\ \ Ir + Is\right]\ (To\ \phi 0\ \ \overline{Int}\ \ Rel)\ \ Ia + \ldots$$

The Ai flip-flop is also set at the T0 time of Phase 0 if an interrupt request is pending and the machine is performing an Execute (23) instruction or an Increment Index and Branch (41) instruction that has resulted in a jump.

$$sAi = \left[(En + \textcircled{En})\ \ Ir + Is\right]\ To\ \phi 0\ \ \overline{Int}\ \ Rel)\ (\overline{Eax}\ \ Ju + O2\ \overline{O3}\ \overline{O4}\ \ O5\ \ O6)\ \overline{O1} + \ldots$$

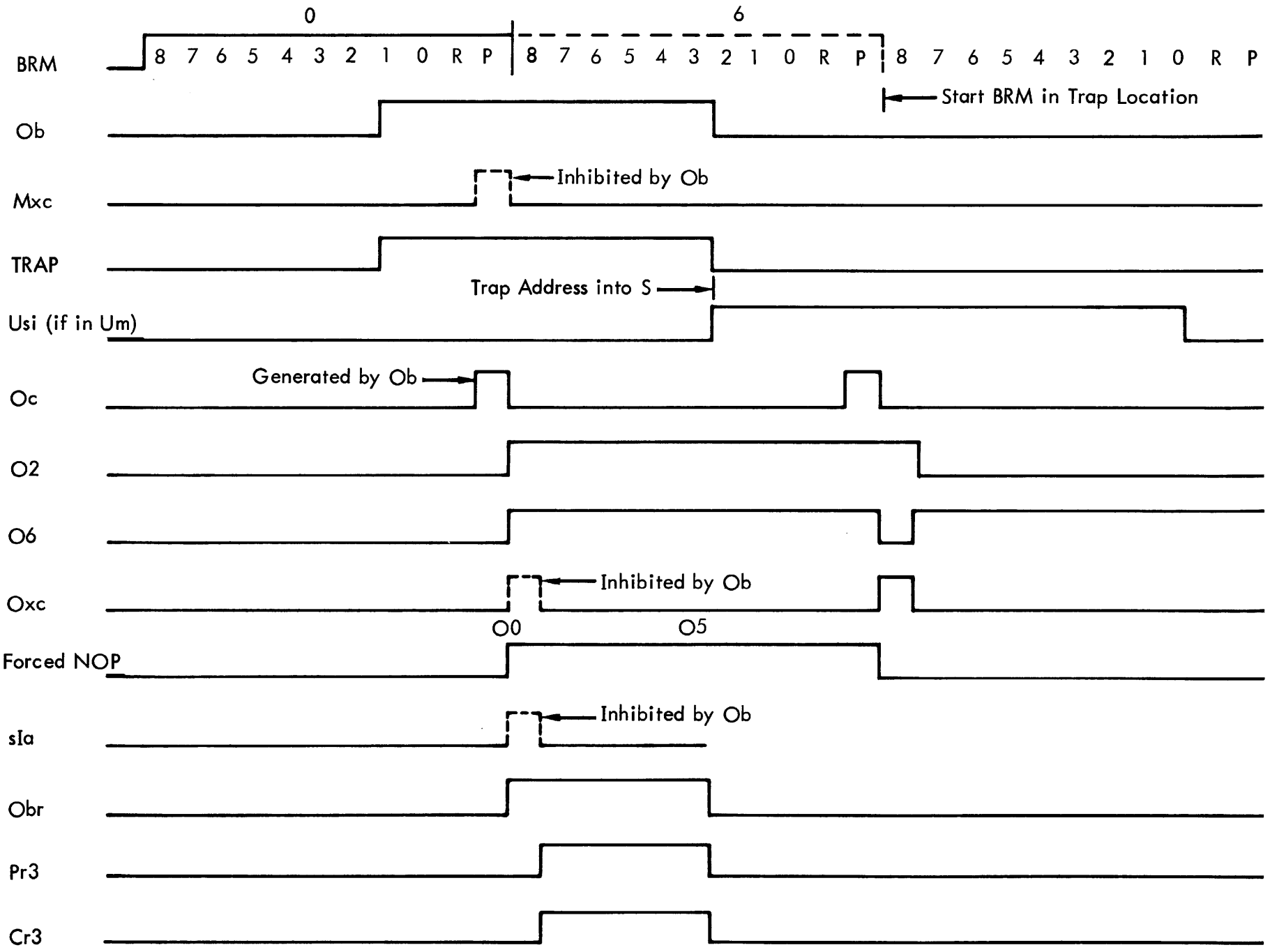


Figure 3-4. BRM to Out-of-Bounds Address Timing

With Ai true, the following operations occur.

(a) At the Tr time of the current Phase 0, the Indirect Address Flip-flop Ia is reset.

$$rIa = \overline{\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ}} + \ldots$$

(b) At the following Tp time, the Oc term is generated to reset the O-register to 20_8, the operation code for a No Operation (NOP) instruction.

$$Oc = Tp\ (Ob + Ai) + \ldots$$

(c) The A00, Ju, and EAX flip-flops are reset if set.

During execution of the forced NOP instruction, the Oxc term is inhibited and the set logic for Ia, EAX, Ju, and A00 is inhibited. Also, because Ia cannot set the P-register will not be incremented.

At T8 time of Phase 0 of the No Operation instruction, the phase counter will be set to Phase 5.

$$sF1 = T8\ (Ob + Ai) + \ldots$$
$$sF3 = T8\ (Ob + Ai) + \ldots$$

Since Phase 5 is an end phase, the highest priority interrupt request will be honored. Because the P-counter was not incremented, the interrupted instruction will begin execution again after the interrupt has been processed.

3.5.4 INSTRUCTION CHANGES. Two new instructions are implemented in the 940 Computer. These instructions use operation codes not used in the 930 Computer. The Branch and Return from Interrupt Routine uses operation code 11 and is assigned the mnemonic BRI. The four overflow instructions utilize the operation code 22 and are assigned the mnemonics OVT, REO, ROV, and OTO. The Copy Effective Address into Index instruction EAX has also been modified. The instruction changes are described in the following paragraphs.

3.5.4.1 Branch and Return from Interrupt Routine Instruction ($\phi 0 \rightarrow \phi 6$). The operation code 11 instruction (BRI) functions like a Return Branch (BRR) Instruction (operation code 51) with the following exceptions.

(a) The interrupt subroutine exit signal (Ib) is generated to clear the highest priority currently active interrupt channel.

$$Ib = \phi 6\ (\overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6)(\overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6\ \overline{Ts}) \quad \text{(BRI)}$$

$$+ \phi 0\ Ia\ Nm\ (\overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6\ \overline{Ts})\ \overline{O3} \quad \text{(BRU I)}$$

The redundancy in the equation occurs because previously formed terms were used for ease of implementation. Note than an indirect addressed unconditional branch instruction (BRU I) will clear an interrupt channel as described in normal mode only.

(b) The overflow flip-flop is cleared then set with the contents of bit two in the return address word in the monitor mode or with bit zero otherwise.

$$rOf = (\overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6)(\overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6)\ \phi6\ T8 + \ldots$$

$$sOf = \phi6\ O3\ (\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)\ \left[(Co + Sm)(C2 + \overline{Sm})\right]\ T7 + \ldots$$

(c) The return address word is not incremented because the serial adder carry flip-flop Cz set logic is inhibited by O1 false.

$$sCz = (Tr\phi0)\ \overline{O4}\ (\overline{O5}\ O6)\ O1 + \ldots$$

(d) To set Phase 6 for the BRI instruction, a new set term was added to F1.

$$sF1 = (\overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6)\ \left[\overline{Ob}\ (\phi0\ \overline{Ia}\ Tp)\right] + \ldots$$

3.5.4.2 New Overflow Instructions. The operation code 22 provides 4 overflow instructions in addition to the existing three 930 Computer overflow instructions. The address bits define the specific function to be performed. At the T8 time of Phase 0, the phase counter goes to Phase 5. In the Overflow Indicator Test and Reset (OVT) instruction, the skip flip-flop Sk is set at the Tr time of Phase 5.

$$sSk = \phi5\ O2\ O5\ \overline{Of}\ C17\ Tr + \ldots$$

The overflow flip-flop is reset at the same time.

$$rOf = \phi5\ O2\ O5\ C23\ Tr + \ldots$$

In the Record Exponent Overflow (REO) instruction, the overflow flip-flop is set at the T4 time of Phase 5.

$$sOf = \phi5\ O2\ O5\ C20\ \left[T4\ (Xn3 \oplus Xw1)\right] + \ldots$$

In the Reset Overflow Indicator (ROV) instruction, the overflow indicator is reset at the Tr time of Phase 5.

$$rOf = \phi5\ O2\ O5\ C23\ Tr + \ldots$$

In the Overflow Indicator Test Only (OTO) instruction, the skip flip-flop is set at the Tr time of Phase 5.

$$sSk = \phi5\ O2\ O5\ \overline{Of}\ C17\ Tr + \ldots$$

The three 930 Computer overflow instructions are still available without modification.

3.5.4.3 Modified EAX Instruction. The execution of a Copy Effective Address into Index (EAX) instruction has been modified to enable the address to be flagged if the address was derived through the user map. When an EAX instruction is executed in the monitor mode, Xo will be set to a one if mapping through the user map was invoked, and set to a zero otherwise. The control over Xo is accomplished as follows. The flip-flop Rbs is time-shared to extend the EAX signal through Tr time.

$$sRbs = \phi 0 \ T7 \ C0 \ Sm + T8 \ Sm \ C0 \ (\overline{O2} \ O3 \ \overline{O4} \ \overline{O5} \ O6) + \overline{Ju} \ EAX \ T3$$

$$rRbs = T3 \ \overline{(\overline{Ju} \ EAX \ T3)} + Tr$$

$$XIB = Sm \ Rbs \ Rb \ Tr + \ldots$$

3.5.5 USER MAP. In the User Mode, the user has access to a certain specified amount of memory. That memory is under program control. The logical method used to restrict the operator to a certain part of memory is to alter the addresses of all memory locations in a prescribed fashion for that user. The address modification is performed by a User Map. To accomplish address conversion, the user map manipulates the three most significant bits of the address field of the 940 Computer Memory address word. The 14-bit word permits user programs to directly address 16, 384 words of core memory. The mapping feature permits this 16, 384 words of memory to be divided into 8 blocks of 2048 words each, fragmented throughout the physical memory.

The user map comprises eight 6-bit quantities designated as R0, R1, ... R7 held in two 24-bit active circuit registers designated as RL1 and RL2. The process for converting a user's program address to a memory address through the user map is illustrated in Figure 3-5. The value, i, defined by the three high-order bits of a 14-bit program address, is used to select the correspondingly numbered quantity of the eight quantities, Ri. The low-order five bits of Ri then have appended to them the 11 low-order bits of the program address to form a 16-bit memory address. Address modification does not add any time to the instruction execution.

3.5.5.1 Select, Clear, and Load Relabeling Register 1. An EOM 20400 is required to select Relabeling Register 1. The EOM instruction must be followed by a POT instruction containing data for loading into the register. During the EOM instruction, RLC1 is set.

$$sRLC1 = EOM \ C10 \ \overline{C11} \ \overline{C13} \ C15 \ T3$$

The 24 flip-flops of RL1 are reset.

$$rRL1 = sRLC1 \ \overline{C14}$$

During the POT instruction, the RLC1 flip-flop is reset.

$$rRLC1 = POT \ T2 + St$$

A signal labeled RLS1 is formed to gate the data into the 24 flip-flops of Relabeling Register 1.

$$RLS1 = RLC1 \ \overline{RLC2} \ POT \ T3$$

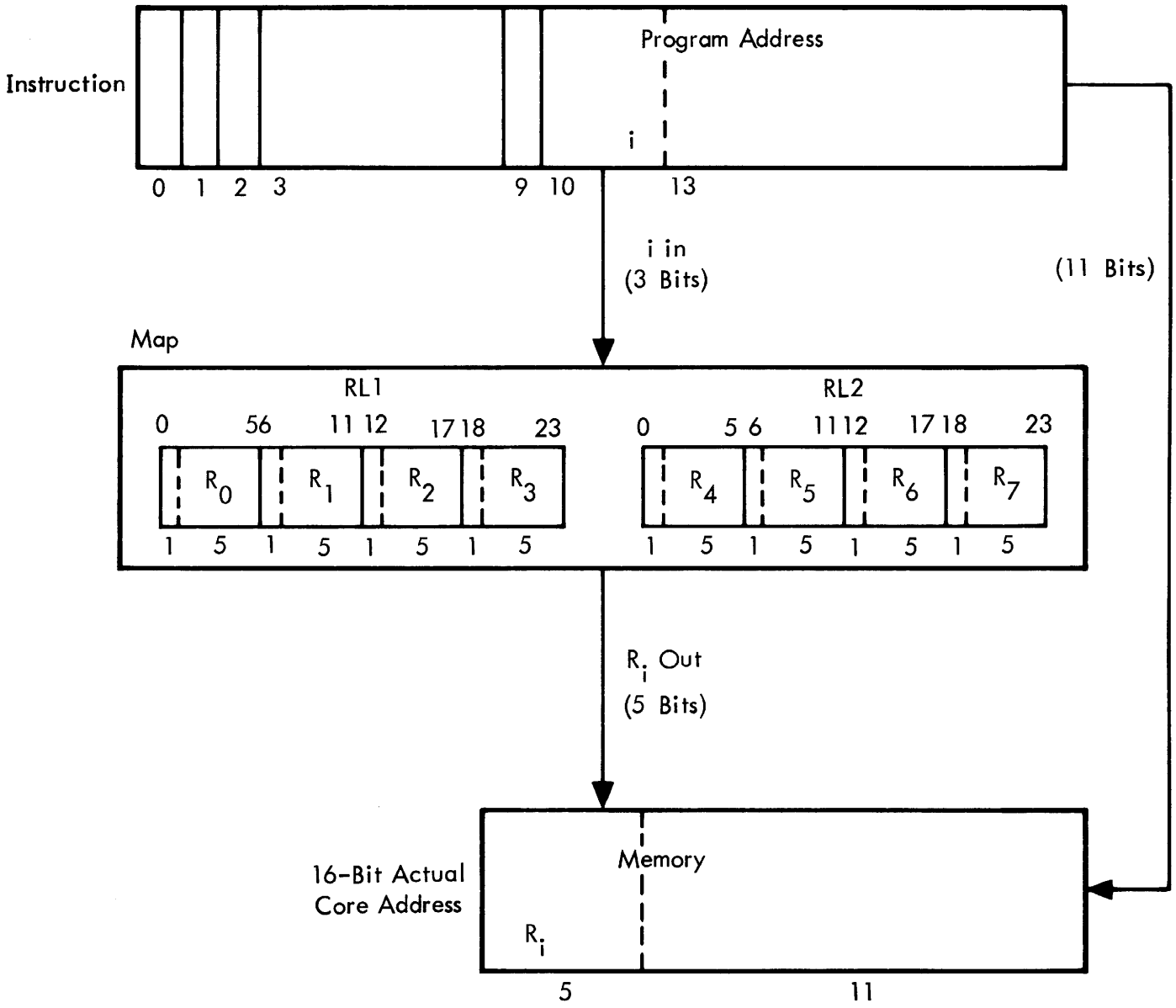


Figure 3-5. SDS 940 Computer Mapping Process

3.5.5.2 Select, Clear, and Load Relabeling Register 2. An EOM 21000 instruction is required to select Relabeling Register 2. The following POT instruction contains data for loading into the register. The RLC2 flip-flop is set by the EOM instruction.

$$sRLC2 = EOM\ \ C10\ \ \overline{C11}\ \ \overline{C13}\ \ C14\ \ T3$$

The 24 flip-flops of RL2 are reset.

$$rRL2 = sRLC2\ \ \overline{C15}$$

During the POT instruction, the RLC2 flip-flop is reset.

$$rRLC2 = POT\ T2 + St$$

The RLS2 signal is generated to gate the data into the 24 flip-flops of the Relabeling Register 2.

$$RLS2 = \overline{RLC1}\ \ RLC2\ \ POT\ \ T3$$

3.5.5.3 User Map Control Logic. Relabeling through the user map occurs for all memory accesses in the user mode, and may be used selectively in the monitor mode. The execution of any instruction in the monitor mode in which the sign bit is a one, causes address mapping through the user map to be used for the remainder of that instruction only. Mapping through the user map will also apply to any instruction during the determining of an effective address for which the sign bit in any intermediate address is a one. The following paragraphs describe logic applicable in the monitor mode.

If the address is to be relabeled in the monitor mode, a flip-flop designated Rbs (relabel next memory cycle through user map) will be set at either T8 or T7 time.

$$sRbs = \phi 0\ \ T7\ \ C0\ \ Sm + T8\ \ Sm\ \ C0\ (\overline{O2}\ \ O3\ \ \overline{O4}\ \ \overline{O5}\ \ O6) + \ldots$$

The Rbs flip-flop is reset at T3 time at the start of the memory cycle.

$$rRbs = T3\ \ \overline{(T3\ \ \overline{Ju}\ \ EAX)} + Tr$$

The Rbs term will qualify the set input of flip-flop Rb that enables monitor mode relabeling through the user map.

$$sRb = T3\ \ RbS\ \ \overline{INt}$$

The Rb flip-flop is not reset until the next T4 time.

$$rRb = END\ \ T4 + MUT$$

The presence of Rb will generate REL and cause relabeling through the user map.

$$Rel = \overline{Nm}\ \ (\overline{Md} + Rb)$$

The Rel signal is always true in the user mode.

In the user map relabeling register structures shown in Figure 3-5, both RL1 and RL2 are divided into four parts of six bits each. One of these eight parts will be selected for each access that is relabeled through the user map. The bit labeled F of the selected six bit quantity is used as a flag bit by the trap logic. The remaining five bits control the most significant five address lines to memory. One more address line was added to the memory in order to address the full 65,536-words of allowable memory in the 940 Computer. This added line is designated L00 and is the most significant bit of the address. The previously mentioned five bits control the five most significant address lines as presented in Table 3-2. The eleven least significant address bits go to memory in the normal fashion.

Table 3-2. Address Line Control

Bit Position	Address Line
H	L00
0	L0
1	L1
2	L2
3	L3

The proper six bit quantity of the Relabeling Registers is selected by decoding S1, S2, and S3 to generate one of eight select signals. The select signal gates the data from the proper relabeling flip-flops onto the address lines. See Paragraph 3.5.8.2 for the address line equations.

3.5.6 MONITOR MAP. The 940 Computer contains a partially implemented monitor map. The monitor map is laid out in a manner identical to that of the user map. The monitor map comprises eight mapping registers of six bits each, laid out in two 24-bit registers designated RL3 and RL4 as shown in Figure 3-6. Physically, no hardware is associated with the unshaded areas and the address falling within this range is used without mapping. Only the shaded portions of the two registers labeled M6 and M7 are implemented with flip-flops. The other six registers are phantom registers containing the values presented in Table 3-3.

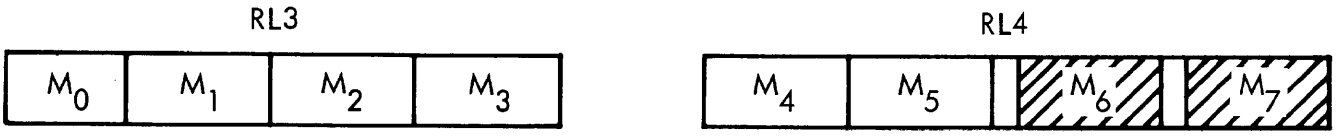


Figure 3-6. Monitor Map Register Structure

Table 3-3. Register Values

Register	Value	Register	Value
M0	0 00000	M3	0 00011
M1	0 00001	M4	0 00100
M2	0 00010	M5	0 00101

Registers M6 and M7 each contain only their five low order flip-flops. The high order bit position is permanently hardwired to contain a zero. Thus, in Figure 3-6, only the shaded areas, containing a total of ten bits, are implemented with flip-flops.

The following paragraphs describe the select, clear and load logic for the monitor map and the control logic, respectively.

3.5.6.1 Select, Clear, and Load Relabeling Register 4. An EOM 21400 is required to select Relabeling Register 4. When this EOM bit configuration occurs, select flip-flops RLC1 and RLC2 are both set.

The 10 flip-flops of RL4 are reset.

$$rRL4 = sRLC1\ \ sRLC2$$

The RLS4 signal is generated to gate data into the 10 flip-flops of Relabeling Register 4.

$$RLs4 = RLC1\ \ RLC2\ \ POT\ \ T3$$

3.5.6.2 Monitor Map Control Logic. Relabeling through this map occurs in the monitor mode for an address with an octal six (6) or seven (7) in the three most significant bits of the S-register. Two select signals are generated which select either the contents of M6 or M7 as the controlling register for the five most significant address lines to memory. These signals are:

$$SEL'6 = S1\ \ S2\ \ \overline{S3}\ \ \overline{Nm}\ \ Md\ \ \overline{Rb}$$

$$SEL'7 = S1\ \ S2\ \ S3\ \ \overline{Nm}\ \ Md$$

Although M6 and M7 do not have a flag bit associated with them, trap logic is such that an all zero quantity cannot be used in either of these registers. An all zero quantity will cause an out-of-bounds trap just as 40_8 will in the user relabeling registers. This limitation is the only access limitation imposed by the monitor map.

The user map has priority over the monitor map if a conflict exists. When in the monitor mode, bit zero is inspected to determine whether or not to go through the user map. If the zero bit contains a zero, then S1, S2 and S3 are inspected to determine whether or not the monitor map should be used.

3.5.7 MEMORY. The 940 Computer memory differs from the 930 Computer memory in three distinct ways.

(a) The 940 memory accepts and decodes 16 address bits.

(b) The 940 memory uses an extended priority scheme between the central processor and the direct access communication channels.

(c) The 940 memory permits two-way or four-way interleave features depending upon the memory configuration.

These differences are described individually in the paragraphs that follow.

3.5.7.1 Address Lines. The memory receives two additional address lines designated LS00 and LZ00. Signals on these lines are inverted in the memory in order to have both the true and false states available to the memory logic. The LS00 line carries the most significant bit of the memory address when the central processor is requesting a memory cycle. The LZ00 line transfers the contents of the JZ00 flip-flop to the memory. The flip-flop designated JZ00 in the central processor accepts and stores the most significant bit of the sixteen (16) bit direct access memory address. The JZ00 flip-flop functions in exactly the same manner as JZ0 through JZ14.

3.5.7.2 Memory Priority. If the central processor and a direct access communication channel address the same memory bank, the method of establishing priority insures that the direct access channel will not gain memory access unless the channel cannot wait until the next memory cycle. Thus, if the status of the direct access channel is such that data overrun will occur, then the channel will take priority over the central processor in the use of that memory cycle.

When a direct access channel makes a memory request, the status of a signal designated Wti, a function of the channel character count, is sent to the central processor. The Wti signal is inverted in the central processor. If the Wti signal was false, the inverted signal is used to set a flip-flop designated Zhp at the beginning of the memory cycle.

$$sZhp = \overline{Wti}\ \overline{Zhp}\ T3$$

The Zhp term is generated to indicate to the memory priority logic the urgency of the direct access channels request. The Zhp flip-flop is reset at the next T4 time.

$$rZhp = T4$$

Under ordinary circumstances, however, the Zhp term is false and priority is specified in the following manner.

The memory is instructed to perform a cycle on the address presented on the Ls lines by the Sa signal.

$$Sa = Sb\ (\overline{Zb\ Zhp}) = Sb\ (\overline{Zb} + \overline{Zhp})$$

The Sb term is an intermediate priority term indicating a memory cycle request from the S-register. The memory is instructed to perform a cycle on the address presented on the Lz lines by the Za signal.

$$Za = Zb\ (\overline{Sb} + Zhp)$$

Where Zb indicates a memory cycle request from the direct access communication channels. When a read only memory cycle request occurs from a direct access communication channel ($Za\ \overline{Mxz}$), the same priority situation occurs.

$$(Za\ \overline{Mxz}) = Zb\ (\overline{Sb} + Zhp)$$

When both Sb and Zb are true and Zhp has not been set, the term Crqm is generated.

$$Crqm = \overline{Zhp}\ Zb\ Sb$$

The Crqm term is sent back to the direct access communication channel through the central processor to indicate that the request for access was not granted. Receipt of Crqm inhibits interlace register counting and causes the same address to be requested again on the next memory cycle.

When the channel must have access, the $\overline{Wti}$ term is sent to the central processor. The Zhp flip-flop will be set at the beginning of the memory cycle and the direct access communication channel request will take priority over the central processor request for that memory cycle.

3.5.7.3 Interleaving. The interleave feature reduces the chances of memory conflict between the central processor and a direct access device. When used in conjunction with the extended priority scheme, this feature permits more efficient use of the memory by the central processor while simultaneously doing input/output on a direct access channel. Two-way and four-way interleave are possible depending on the memory configuration.

The two-way interleaving feature can be used with two 16,394 word memory banks. A two-way interleave may be explained by the following assumptions:

(a) A 15-bit address of the form L0, L1, . . . , L13, L14.

(b) Two 16,384 word memory banks, numbered 0 and 1.

(c) Address decoding within the memory banks such that if L0 = 0 the address falls within the range of bank 0, and if L0 = 1 the address falls within the range of bank 1.

For any series of sequential addresses, the least significant bit (L14) will toggle for every memory access. If L14 is switched with L0 in the memory, the physical memory location for any series of sequential addresses will alternate between bank 0 and 1. This method is used in the 940 Computer for a two-way interleave.

The four-way interleave feature requires four equal size memory banks. Because the 940 Computer must address up to 65, 536 words, the address word contains 16 bits in the form L00, L0, L1, . . ., L13, L14. In four-way interleaving, L13 and L14 are switched with L00 and L0 to define the memory bank containing the requested address. When the four-way interleave feature is used for a series of sequential central processor addresses, the physical data locations will be, bank 3, bank 2, bank 1, bank 0, bank 3, etc. This is due to the fact that the true side of L0 and the false side of L14 are physically switched in the hardware; L00 and L13 are switched in the same manner.

When the interleaving feature is used, the Lz address lines from direct access devices are treated in the same manner as the Ls address lines from the central processor.

The switching of L13 and L14 for L00 and L0 is accomplished on the central processor cable plug modules. Figure 3-7 is a schematic of the switching arrangement. The switches are arranged physically on the cable plug module from top to bottom as shown in Figure 3-7. The switch positions on P921 and P922 should always be the same. There are three possible interleave conditions.

(a) All switches in the up position; memory addressing is normal and no interleaving occurs.

(b) Two switches in down position selects two (2) way interleave.

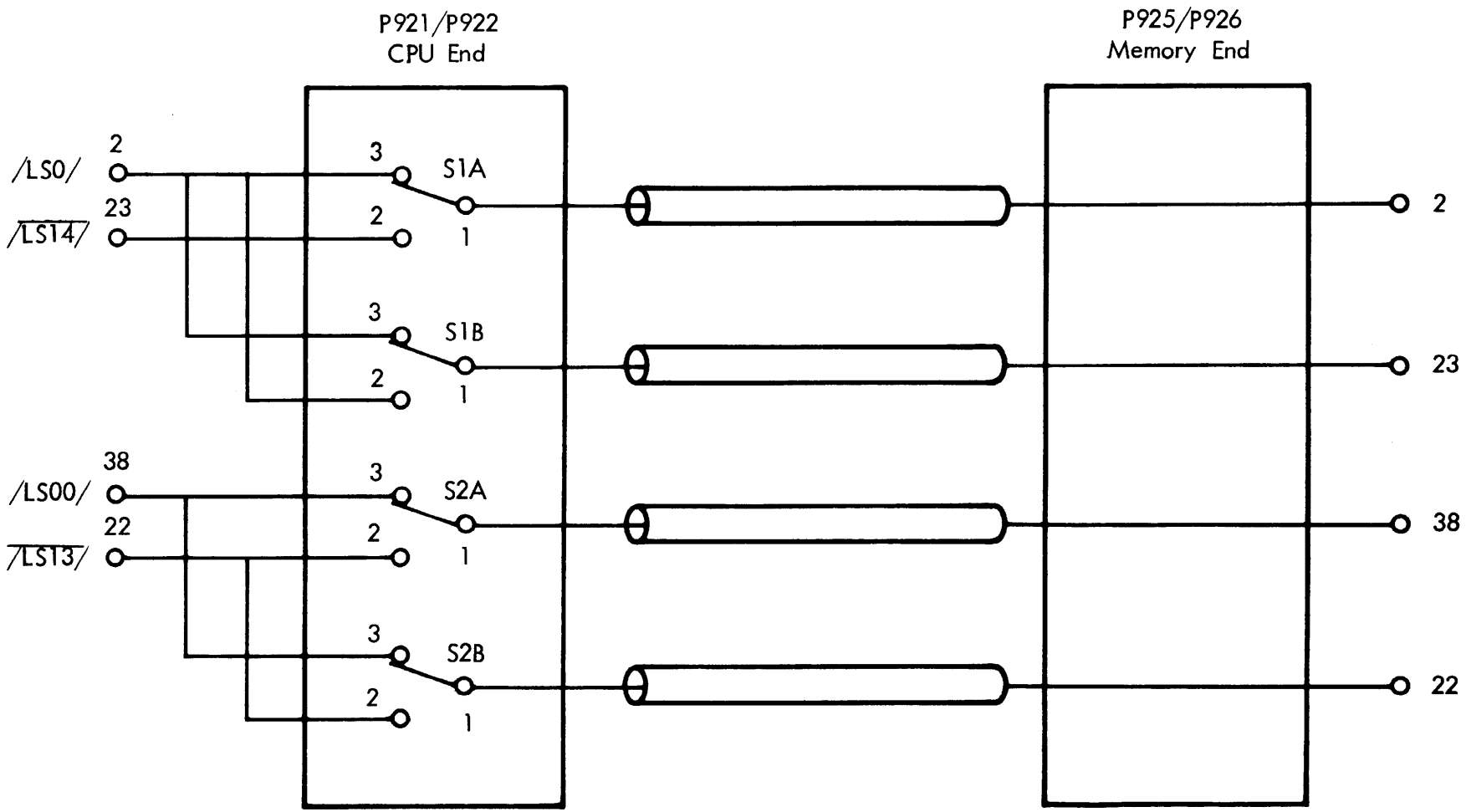
(c) All switches in the down position; selects four (4) way interleave.

Figure 3-8 shows a system with 3 memory banks using 2-way interleave. Figure 3-9 shows 4 memory banks using 4-way interleave.

3.5.7.4 Jumper Modules. Because of the extended addressing, a new set of jumper modules was designed for the 940 Computer. Tables 3-4 through 3-7 list these modules by number and give a complete description of each.

3.5.8 LOGIC TERMS. The Glossary of terms in this section describes only the new signals formed in the 940 Computer. The equations are for these new signals formed and for existing 930 signals where the logic equation was changed by the 940 modifications. For the complete 940 Computer equations see drawing number 126185.

3.5.8.1 Glossary. The glossary of logic terms presented in Table 3-8 include only those terms that are different from the 930 Computer. A complete glossary can be found in the drawing.



Notes:

a. Switches are shown in Up position.

b. P921 is in location 33F, 34F,
P922 is in location 35F, 36F.

c. P925 is in location 10F,
P926 is in location 13F.

Figure 3-7. 940 Computer Interleave Switch Arrangement

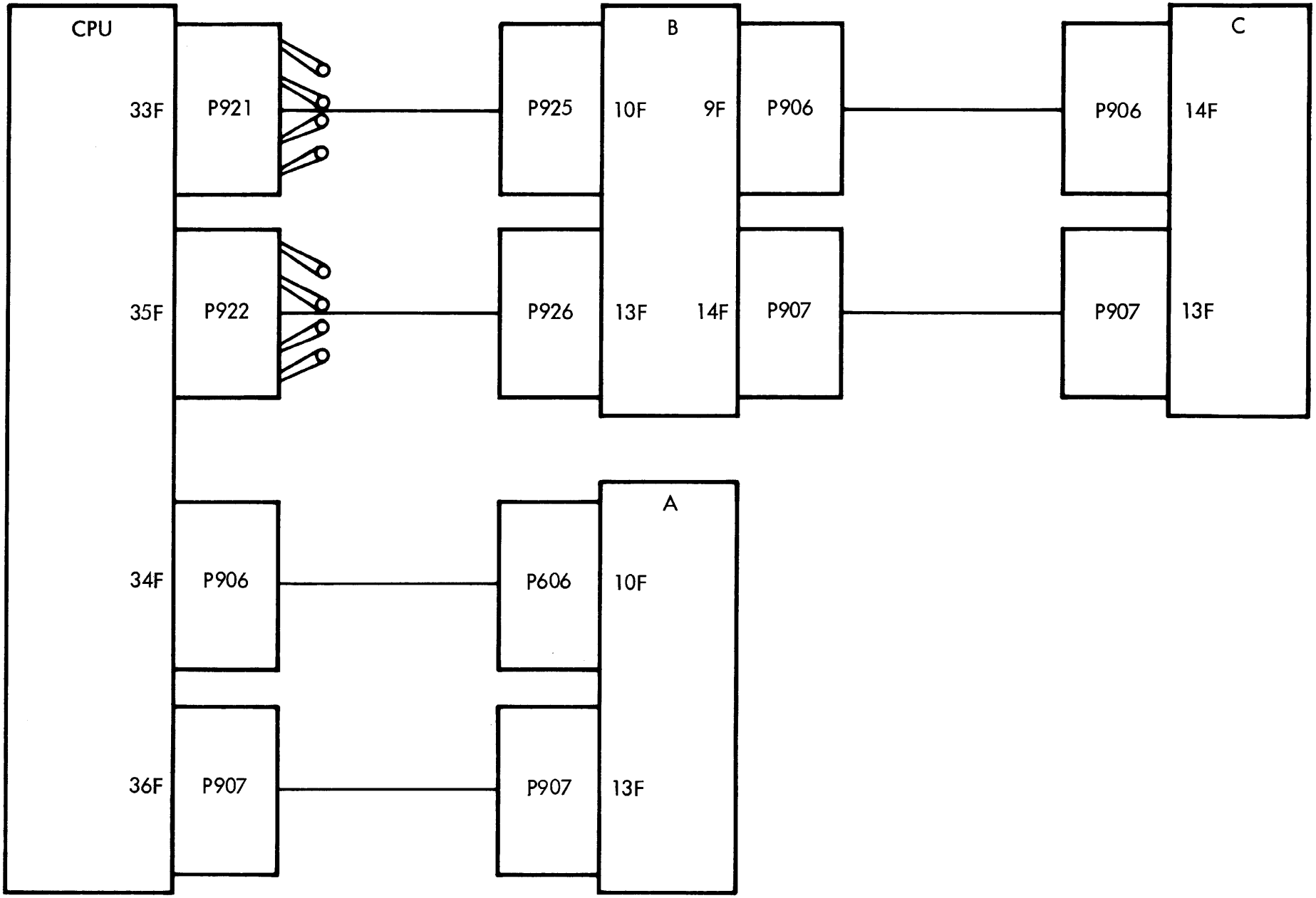


Figure 3-8. Two Way Interleave, Three Memory Banks

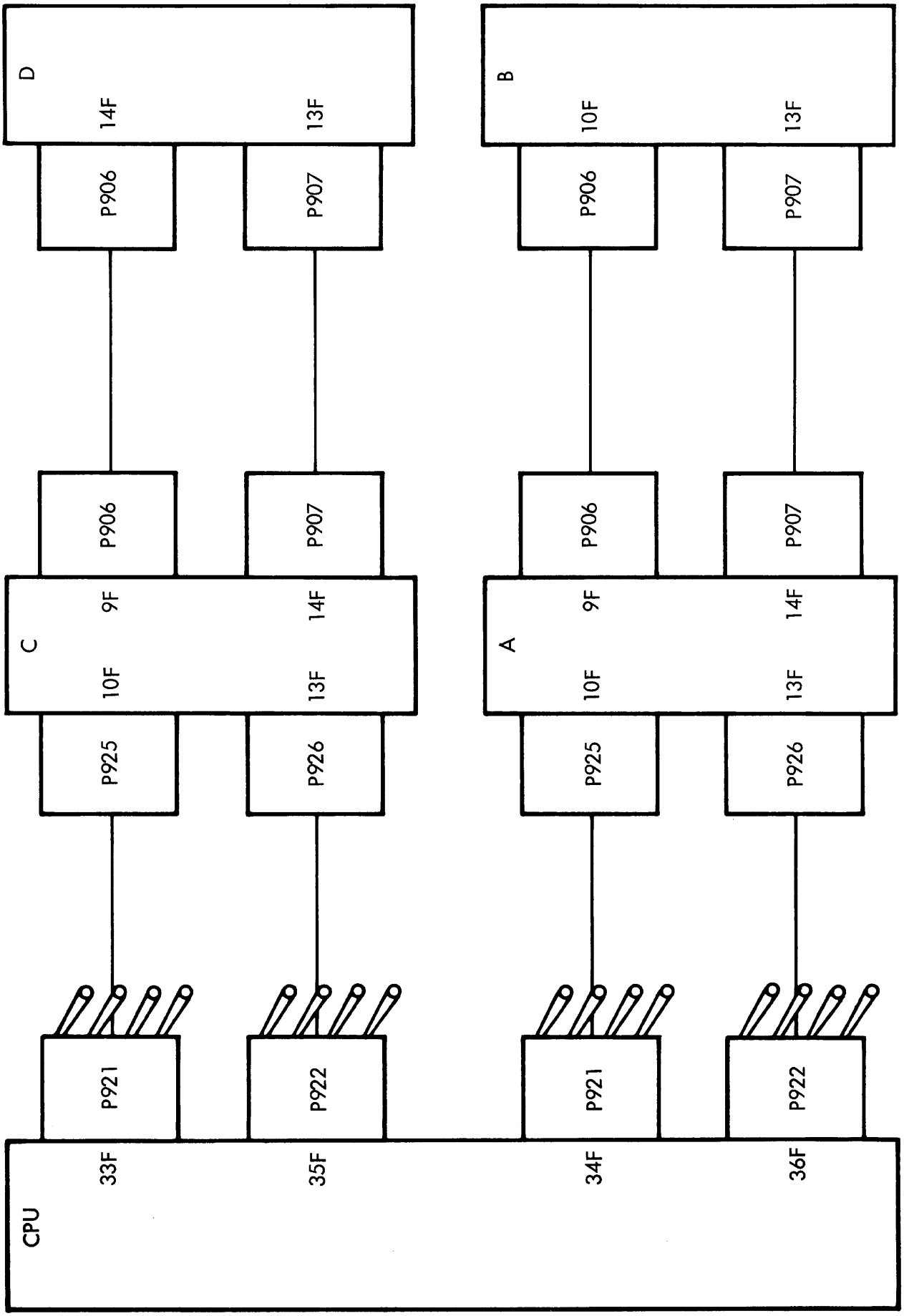


Figure 3-9. Four Way Interleave, Four Memory Banks

Table 3-4. Jumper Module ZB 65-60[1]

In		Out	
Signal	Point	Signal	Point
/$\overline{\text{LZ1}}$/	106	/$\overline{\text{LZ1K}}$/	131
/$\overline{\text{LZ2}}$/	107	/$\overline{\text{LZ2K}}$/	132
/$\overline{\text{LS1}}$/	118	/$\overline{\text{LS1K}}$/	129
/$\overline{\text{LS2}}$/	119	/$\overline{\text{LS2K}}$/	130
LZ0	143	/$\overline{\text{LZ0J}}$/	120
GND	220	/$\overline{\text{LZ1J}}$/	121
LZ00	142	/$\overline{\text{LZ2J}}$/	122
LS0	139	/$\overline{\text{LS0J}}$/	126
GND	228	/$\overline{\text{LS1J}}$/	127
LS00	138	/$\overline{\text{LS2J}}$/	128

[1]0 - 64K even-even, or 0 - 32K even, or 0 - 16K

Table 3-5. Jumper Module ZB 65-61[1]

In		Out	
Signal	Point	Signal	Point
/$\overline{\text{LZ0}}$/	105	/$\overline{\text{LZ0J}}$/	120
/$\overline{\text{LZ1}}$/	106	/$\overline{\text{LZ1K}}$/	131
/$\overline{\text{LZ2}}$/	107	/$\overline{\text{LZ2K}}$/	132
/$\overline{\text{LS0}}$/	117	/$\overline{\text{LS0J}}$/	126
/$\overline{\text{LS1}}$/	118	/$\overline{\text{LS1K}}$/	129
/$\overline{\text{LS2}}$/	119	/$\overline{\text{LS2K}}$/	130
GND	220	/$\overline{\text{LZ1J}}$/	121
LZ00	142	/$\overline{\text{LZ2J}}$/	122
GND	228	/$\overline{\text{LS1J}}$/	127
LS00	138	/$\overline{\text{LS2J}}$/	128

[1]0 - 64K even-odd, or 0 - 32K odd, or 16 - 32K

3.5.8.2 Equations. The logic equation presented in Table 3-9 include only those equations that are different from the 930 Computer. A complete list of the equations are included in the drawing.

Table 3-6. Jumper Module ZB 65-62[1]

In		Out	
Signal	Point	Signal	Point
/$\overline{\text{LZ1}}$/	106	/$\overline{\text{LZ1K}}$/	131
/$\overline{\text{LZ2}}$/	107	/$\overline{\text{LZ2K}}$/	132
/$\overline{\text{LS1}}$/	118	/$\overline{\text{LS1K}}$/	129
/$\overline{\text{LS2}}$/	119	/$\overline{\text{LS2K}}$/	130
LZ0	143	/$\overline{\text{LZ0J}}$/	120
GND	220	/$\overline{\text{LZ1J}}$/	121
/$\overline{\text{LZ00}}$/	141	/$\overline{\text{LZ2J}}$/	122
LS0	139	/$\overline{\text{LS0J}}$/	126
GND	228	/$\overline{\text{LS1J}}$/	127
/$\overline{\text{LS00}}$/	137	/$\overline{\text{LS2J}}$/	128

[1]0 - 64K odd-even, or 32 - 64K even, or 32 - 48K

Table 3-7. Jumper Module ZB 65-63[1]

In		Out	
Signal	Point	Signal	Point
/$\overline{\text{LZ0}}$/	105	/$\overline{\text{LZ0J}}$/	120
/$\overline{\text{LZ1}}$/	106	/$\overline{\text{LZ1K}}$/	131
/$\overline{\text{LZ2}}$/	107	/$\overline{\text{LZ2K}}$/	132
/$\overline{\text{LS0}}$/	117	/$\overline{\text{LS0J}}$/	126
/$\overline{\text{LS1}}$/	118	/$\overline{\text{LS1K}}$/	129
/$\overline{\text{LS2}}$/	119	/$\overline{\text{LS2K}}$/	130
GND	220	/$\overline{\text{LZ1J}}$/	121
/$\overline{\text{LZ00}}$/	141	/$\overline{\text{LZ2J}}$/	122
GND	228	/$\overline{\text{LS1J}}$/	127
/$\overline{\text{LS00}}$/	137	/$\overline{\text{LS2J}}$/	128

[1]0 - 64K odd-odd, or 32 - 64K odd, or 48 - 64K

Table 3-8. Glossary of Logic Terms

Logic Designation	Function
Ai	The flip-flop that allows a chain of indirect addressing or execute instructions to be interrupted. It also allows an interrupt to occur after a BRX (41) instruction that branches. This flip-flop does not function in the normal (930) mode.
CrQ	A signal from the C.P.U. to a direct access I/O channel to inform the channel that it was not granted memory access.
CrQM	The signal from the memory to make up CrQ in the C.P.U.
JZ00	The flip-flop that accepts and stores the most significant bit of the 16 bit address from a direct access I/O channel.
LS00	The cable driver that sends the most significant bit of the 16 bit computer address to memory.
LZ00	The cable driver that sends the most significant bit of the 16 bit direct access I/O channel address to memory.
M6H, M60, M61, M62, M63, M7H, M70, M71, M72, M73	The ten flip-flops of the monitor relabeling registers.
Md	One of the two flip-flops that define the mode of operation.
MUT	A signal generated when a monitor to user mode transition occurs due to a relabeled jump. It resets the flip-flops Md and Rb.
Nm	A flip-flop that is set in the normal (930) mode.
Ob	The out-of-bounds flip-flop, is also set for a read-only memory violation.
ObA	A signal indicating an attempted out-of-bounds memory access.

Table 3-8. Glossary of Logic Terms (Continued)

Logic Designation	Function
Obr	A signal used to gate the full adder outputs into the P-register during the forced NOP following an attempted BRM (43) to illegal memory.
Pi	The privileged instruction flip-flop. It is also set for a read only memory violation and a monitor to user transition trap.
Pid	The output of the privileged instruction decode networks.
PiQ	A signal generated during the attempted execution of a privileged instruction in the user mode or at the occurrence of a monitor to user transition trap. The signal is used to set Pi and force Phase 5.
Rb	The flip-flop that causes relabeling through the user map to be effective in the monitor mode.
Rbs	A flip-flop that indicates in the monitor mode the next memory cycle is to be relabeled through the user map. This flip-flop is also used to extend the EAX signal for an EAX (77) instruction.
Rel	A signal indicating the memory address is to be relabeled through the user map.
RL0F, RL0H, RL00, RL01, RL02, RL03 thru RL7F, RL7H, RL70, RL71, RL72, RL73	The 48 flip-flops of user relabeling registers.
RLC1	The select flip-flop for loading relabeling registers zero through three.
RLC2	The select flip-flop for loading relabeling registers four through seven.
RLS1	The signal that strobes the data into relabeling registers zero through three.
RLS2	The signal that strobes the data into relabeling registers four through seven.

Table 3-8. Glossary of Logic Terms (Continued)

Logic Designation	Function
RLs4	The signal that strobes the data into the monitor relabeling registers.
rRL1	The signal that clears relabeling registers zero through three.
rRL2	The signal that clears relabeling registers four through seven.
rRL4	The signal that clears the monitor relabeling registers.
Rtt	A ready signal used to set the Rf flip-flop which permits an I/O instruction to advance from Phase 2.
S3'	The fifth most significant memory address bit out of the memory map. It controls the L3 address line to memory.
SeL0 ------- SeL7	The signals that select the user relabeling register to be used for a memory access.
SeL'6, SeL'7	The signals that select the monitor relabeling register to be used for a memory access.
SFM	A signal that monitors the state of the flag bit in the selected relabeling register for a user mapped memory access.
SM	A signal indicating monitor mode of operation. This mode is sometimes referred to as Systems Mode.
STV	A signal indicating an attempted store into a read only memory location.
Tpf	A signal used to set the state of the phase counter to Phase 5.
TRAP	A signal that is true for all trap conditions.
Tte	The flip-flop that arms the monitor to user transition trap.
Um	A signal indicating user mode.
Umt	A signal indicating a user to monitor mode transition is to occur.

Table 3-8. Glossary of Logic Terms (Continued)

Logic Designation	Function
Usi	A flip-flop that is set when a user to monitor mode transition occurs or is soon to occur.
Wti	A signal from a direct access I/O channel indicating priority for a memory access.
ZhP	A flip-flop that indicates a direct access I/O channel is to have priority over the C.P.U. for memory access.

Table 3-9. 940 Logic Equations

$$
\begin{aligned}
sA00 &= Ex\ To\ \overline{A00}\ \textcircled{Ka} \\
&\quad + Ex\ To\ \overline{A00}\ \textcircled{Kx} \\
(64\text{-}67) &\quad + \phi 0\ T7\ A0 \\
(67) &\quad + AL2\ O5\ A1 \\
(40) &\quad + \phi 5\ T7\ C3\ \overline{C6}\ C11\ Go\ \overline{A00}\ (\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ}) \\
\\
rA00 &= Tp\ End\ Go \\
(40) &\quad + \phi 5\ T7\ (Go + Ht)\ A00 \\
(STS) &\quad + O1\ \overline{O2}\ Tr\ (Trq + Mit) \\
(67) &\quad + AL2\ O5\ \overline{A1} \\
(64) &\quad + (\phi 7\ \overline{O3}\ O4\ \overline{O5}\ \overline{O6})\ T7\ C0 \\
(23) &\quad + (Tp\ \overline{O1}\ \overline{O3}\ \overline{Ia}\ Go) \\
(20) &\quad + Ex\ To\ A00 \\
&\quad + Tp\ (Ai + Ob) \\
\\
sAi &= \left[(En + \textcircled{En})\ Ir + Is\right]\ (To\ \phi 0\ \overline{Int}\ Rel)\ Ia \\
(23) &\quad + \left[(En + \textcircled{En})\ Ir + Is\right]\ (To\ \phi 0\ \overline{Int}\ Rel)\ (\overline{Eax}\ Ju + O2\ \overline{O3}\ \overline{O4}\ O5\ O6)\ \overline{O1}
\end{aligned}
$$

Table 3-9. 940 Logic Equations (Continued)

$$rAi = T3$$

$$sC0 = Cr3 \{ \text{(same as 930 except for the following term)} + (Ju\ \overline{Ts})\ To\ (Usi + \overline{Sm}\ Of) + T1\ E3m0 + T2\ E2m0 + \ldots \}$$

$$rC0 = Cr3 \{ \overline{(Ju\ \overline{Ts})\ To\ (Usi + \overline{Sm}\ Of) + T1\ E3m0 + T2\ E2m0 + \ldots} \} + Cxm\ Tr + Cxi\ Q2 + Ck\ Cb$$

$$sC2 = Cr3 \{ \text{(same as 930 except for the following term)} + Ju\ \overline{Ts})\ (To\ Sm\ Of + T1\ E3m2 + T2\ E2m2) + \ldots \}$$

$$rC2 = Cr3 \{ \overline{(Ju\ \overline{Ts})\ (To\ Sm\ Of + T1\ E3m2 + T2\ E2m2) + \ldots} \} + Cxm\ Tr + Cxi\ Q2 + Ck\ C2$$

$$Cr3 = Tsr\ Q1$$

(ϕ0, ϕ1) $+ \overline{F1}\ \overline{F2}\ (\overline{Ts}\ Q1)$

(ϕ4, ϕ6) $+ F1\ \overline{F3}\ (\overline{Ts}\ Q1)$

(64, 65) $+ \phi7\ \overline{O5}\ (\overline{Ts}\ Q1)$

$+ \phi5\ Ob\ O6\ (\overline{Ts}\ Q1)$

$+ Ex\ (\overline{Ts}\ Q1)$

$+ St$

$+ (Kcr + Kmc)\ (\overline{Ts}\ Q1)$

$+ \overline{Go}\ \overline{Ix}\ Rf$

$$/\overline{CrQ}/ = /\overline{CrQM}/$$

Table 3-9. 940 Logic Equations (Continued)

sEax = $\left[(\phi 0\ T8\ \overline{Ia}\ Go)\ C2 + (\phi 0\ T8\ \overline{Ia}\ Go)\ C3\ C4\ C5\ C6\ C7\ C8\ \overline{C9} + (\phi 0\ T8\ \overline{C9})\ O1\ O2\ O3\ (O4\ O5\ O6)\right]\ (\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ})$

Eom = $(\phi 5\ \overline{O1}\ \overline{O2}\ O5\ \overline{Ts}\ (\overline{Q2}\ \overline{Q5})\ \overline{O4}) + Ix\ \overline{Go}\ \overline{Ht}\ (\overline{Q2}\ \overline{Q5})$ (Kg)

sF1 = $\overline{Ob}\ Tp\ (Eax + Sk + \phi 4)$

(14-17, 34-37, 54-57, 74-77) $+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ O3\ O4$

(41-43, 50-53, 60-63, 70-73) $+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ O1\ \overline{O4}$

(10-13, 30-33) $+ Tp\ \overline{Ob}\ (\overline{F1}\ \overline{F3}\ \overline{O1}\ O3\ \overline{O4}\ \overline{Ia})\ Rf$

$+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ \overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6$

$+ (\phi 0\ T8\ \overline{Ia})\ \overline{C2}\ \overline{C5}\ \overline{C8}\ (\overline{C3} + \overline{C4})$

$+ T8\ (Ob + Ai)$

$+ PiQ$

$+ T8\ \overline{Go}$

rF1 = $Tp\ End\ \overline{Sk}$

$+ Tp\ Ob$

sF2 = $\overline{Ob}\ Tp\ (Eax + Sk + \phi 4)$

(41-43, 50-57) $+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ O1\ \overline{O2}$

(1-5, 5-7) $+ \left[\overline{Ob}\ (\phi 0\ \overline{Tp}\ Ia)\right]\ O3\ (O1 + \overline{O2})$

(30, 32, 33) $+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ O3\ \overline{O4}\ \overline{Rf}$

(65) $+ \phi 1\ Tp$

(66, 67) $+ \phi 1\ O5\ \overline{Q2}$

(64) $+ \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\ O4\ \overline{O5}\ \overline{O6}$

Table 3-9. 940 Logic Equations (Continued)

$$
\begin{aligned}
rF2 &= Tp\ End\ \overline{Sk} \\
(30\text{-}33) &\quad + (\overline{F1}\ \overline{F3}\ \overline{O1}\ O3\ \overline{O4}\ \overline{Ia})\ Rf\ O2\ Tp \\
\\
sF3 &= \overline{Ob}\ Tp\ (Eax + Sk + \phi 4) \\
&\quad + (\phi 0\ T8\ \overline{Ia}\ \overline{C2})\ \overline{C5}\ \overline{C8}\ (\overline{C3} + \overline{C4}) \\
&\quad + T8\ \overline{Go} \\
(64, 65) &\quad + \left[\overline{Ob}\ (\phi 0\ Tp\ \overline{Ia})\right]\overline{O3}\ O4 \\
(66, 67) &\quad + (\phi 0\ \overline{Ia}\ Q4)\ \overline{O3}\ O4\ O5 \\
&\quad + T8\ (Ob + Ai) \\
&\quad + PiQ \\
\\
rF3 &= (Tp\ End\ \overline{Sk}) \\
\\
sIa &= \phi 0\ T8\ \overline{C2}\ C9\ \overline{Tpf} \\
&\quad + Tpf\ (\overline{Pi}\ \overline{Ai}\ \overline{Ob}\ \overline{PiQ})\ \textcircled{Kr} \\
&\quad + T8\ \phi 7\ Sk\ (Inr + \overline{Ij})\ \textcircled{Kr} \\
(\phi 4, \phi 6) &\quad + T8\ F1\ \overline{F3}\ \left[(Inr + \overline{Ij})\ (\overline{Rel} + \overline{Mxc} + \overline{Sfm})\right]\ \textcircled{Kr} \\
&\quad + T8\ \textcircled{Kmc} \\
\\
rIa &= \overline{(\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ})} \\
&\quad + (\overline{P12}\ \overline{P13}\ \overline{P14})\ Q2\ F1 \\
&\quad + Tr\ F1\ (\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ}) \\
&\quad + \phi 0\ T8\ \overline{C9}\ Ia \\
\\
Ib &= \phi 0\ Ia\ Nm\ (\overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6\ \overline{Ts})\ \overline{O3} \qquad (01) \\
&\quad + \phi 6\ (\overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6)\ (\overline{O1}\ \overline{O2}\ \overline{O4}\ \overline{O5}\ O6\ \overline{Ts}) \qquad (11) \\
&\quad + \overline{Ts}\ Ij\ End\ \overline{Inr} \\
\\
sInt &= \left[\overline{(\phi 5\ \overline{O1}\ O5\ \overline{Ts}\ Q1)}\ \overline{TRAP}\ End\ \overline{Ts}\right]\ \left[Is + Ir\ (En + \textcircled{En})\right]\ T4 \\
\\
rInt &= T7\ \overline{Sk}\ Go + St
\end{aligned}
$$

Table 3-9. 940 Logic Equations (Continued)

sJu = (ϕ0 T8 $\overline{Ia}$ Go C2 + ϕ0 T8 $\overline{Ia}$ Go $\overline{C4}$ $\overline{C5}$ C8 $\overline{C9}$ + ϕ0 T8 $\overline{C9}$ $\overline{O2}$ $\overline{O3}$) ($\overline{Ob}$ $\overline{Pi}$ $\overline{Ai}$ $\overline{PiQ}$)

rJu = Tp

sJz00 = T3 Iz00 (to Lz00 cable driver to memory)

rJz00 = T4

Ls00 = $\overline{Tsm}$ (Sel'6 M6H + Sel'7 M7H $\overline{Rb}$ + Sel 0 RL0H + Sel 1 RL1H + Sel 2 RL2H Rel + Sel 3 RL3H Rel + Sel 4 RL4H + Sel 5 RL5H + Sel 6 RL6H Rel + Sel 7 RL7H Rel) + Tsm Ir00

Ls0 = $\overline{Tsm}$ (Sel'6 M60 + Sel'7 M70 $\overline{Rb}$ + E2m0 $\overline{S2}$ S1 $\overline{Rel}$ + Nm S2 S1 $\overline{Rel}$ E3m0 + Sel 0 RL00 + Sel 1 RL10 + Sel 2 RL20 Rel + Sel 3 RL30 Rel + Sel 4 RL40 + Sel 5 RL50 + Sel 6 RL60 Rel + Sel 7 RL70 Rel) + Tsm Ir0

Ls1 = Tsm (Sel'6 M61 + Sel'7 M71 $\overline{Rb}$ + E2m1 $\overline{S2}$ S1 $\overline{Rel}$ + E3m1 Nm S2 S1 $\overline{Rel}$ + Sel 0 RL01 + Sel 1 RL11 + Sel 2 RL21 Rel + Sel 3 RL31 Rel + Sel 4 RL41 + Sel 5 RL51 + Sel 6 RL61 Rel + Sel 7 RL71 Rel) + Tsm Ir1

Ls2 = $\overline{Tsm}$ (Sel'6 M62 + Sel'7 M72 $\overline{Rb}$ + $\overline{Rel}$ $\overline{S1}$ S2 + E3m2 Nm S2 $\overline{Rel}$ + E2m2 S1 $\overline{Rel}$ $\overline{S2}$ + Sel 0 RL02 + Sel 1 RL12 + Sel 2 RL22 Rel + Sel 3 RL32 Rel + Sel 4 RL42 + Sel 5 RL52 + Sel 6 RL62 Rel + Sel 7 RL72 Rel) + Tsm Ir2

Ls3 = $\overline{Tsm}$ S3' + Tsm Ir3

sM6H = RLs4 C13

rM6H = rRL4

Table 3-9. 940 Logic Equations (Continued)

```
sM60    = RLs4 C14
rM60    = rRL4
  |       |
  |       |
  |       |
sM63    = RLs4 C17
rM63    = rRL4

sM7H    = RLs4 C19
rM7H    = rRL4

sM70    = RLs4 C20
rM70    = rRL4
  |       |
  |       |
  |       |
sM73    = RLs4 C23
rM73    = rRL4

sMd     = T3 Usi Go + Umt + rNm
rMd     = (Usi Ib T4) + Mut

sMgs    = T3 St (F1 F3 Tsm) (Bc23 φ5 Tsm) (φ0 Ia Q4 O1 O2 O3 Tsm)
rMgs    = T4

Mut     = (O2 O3 O4 O5 O6) End Rb                          (11, 51)
          + (O1 O2 O4 O5 O6 Ts) Ju Rb                      (01, 41)
          + (φ0 Tr) Mxc Rb                                 (43)

Mxc     = Ob [Tsm (Ju O1 O5 + φ4 Ob + Eax) + Tsm R9 + (Kmc)]

sNm     = St
rNm     = Eom C10 C11 C13 C14 C15 T3
```

Table 3-9. 940 Logic Equations (Continued)

$$sO6 = Oxc\ C8 + \left[\overline{Tsm}\ (Ju\ O1\ O5 + \phi 4\ \overline{Ob} + Eax) + Tsm\ \overline{R9}\right] Ob\ Oc$$

$$rO6 = Oc\left\{\overline{Ob} + \left[\overline{\overline{Tsm}\ (Ju\ O1\ O5 + \phi 4\ \overline{Ob} + Eax) + Tsm\ \overline{R9}}\right]\right\}$$

$$sOb = Oba + Stv$$

$$rOb = Go\ T3 + St$$

$$Oba = \left\{\left[(Sfm\ \overline{Ls00A'}\ \overline{Rbs}\ T2)\ \overline{(Ju\ O1\ \overline{O5}\ \overline{Xw1})}\ \overline{(\phi 0\ \overline{End})}\ \overline{S3'}\ Go\right]\left[\overline{Ls0A'}\ \overline{Ls1A'}\ \overline{Ls2A'}\ (Sel'6 + Sel'7 + Rel)\right]\right\}$$

$$Obr = (Ob\ O6)$$

$$Oc = Tp\ End\ \overline{Sk} + Tp\ \overline{O1}\ \overline{O3}\ \overline{Ia}\ Go + Tp\ (Ob + Ai)$$

$$sPi = Stv + PiQ$$

$$rPi = T3\ Go + St$$

$$Pid = C6\ \overline{C5}\ \overline{C3} + \overline{C3}\ C5\ \overline{C6} + \overline{C4}\ \overline{C5}\ \overline{C6}\ \overline{C8} + \overline{C3}\ \overline{C4}\ C7\ \overline{C5} + \overline{C3}\ \overline{C4}\ \overline{C7}\ C8\ C5 + \overline{C3}\ C4\ \overline{C7}\ C8\ \overline{C5} + \overline{C3}\ C4\ \overline{C7}\ \overline{C8}\ C5 + \overline{C4}\ \overline{C5}\ \overline{C7}\ \overline{C8} + \overline{C4}\ \overline{C5}\ C6\ C8$$

$$PiQ = Pid\ (Go\ Um)\ (\overline{Ob}\ \overline{Ai})\ \overline{C2}\ (\phi 0\ T8\ \overline{Ia}) + \overline{Ob}\ (Go\ Um)\ (\phi 0\ T8\ \overline{Ia})\ Tte$$

Table 3-9. 940 Logic Equations (Continued)

$$sRb = T3\ Rbs\ \overline{Int}$$

$$rRb = T4\ End + Mut$$

$$sRbs = \phi 0\ T7\ C0\ Sm + T8\ Sm\ C0\ (\overline{O2}\ O3\ \overline{O4}\ \overline{O5}\ O6) + T3\ \overline{Ju}\ Eax$$

$$rRbs = \overline{T3\ (T3\ \overline{Ju}\ Eax)} + Tr$$

$$sRc = (\phi 0\ T8\ \overline{Ia}\ Go)\ \overline{C2}\ C3\ \overline{C4}\ \overline{C5}\ C6\ (\overline{Ob}\ \overline{Pi}\ \overline{Ai}\ \overline{PiQ})$$

$$rRc = Tr + C17\ T5$$

$$Rel = \overline{Nm}\ (\overline{Md} + Rb)$$

$$sRf = Q2\ \overline{F3}\ F2\ O6\ Rtt + (\text{Same as 930}) \qquad (13, 33)$$

$$rRf = Tp\ \overline{\phi 1}\ \overline{(\overline{Go}\ Ht)} + (\phi 7\ O5\ T7)\ \overline{S3} + Kf2\ Ix + (St) \qquad (67)$$

$$Rtt = Rt + RLC1 + RLC2$$

Table 3-9. 940 Logic Equations (Continued)

User relabeling register flip-flops:

sRL0F = RLS1 C0
sRL0H = RLS1 C1
sRL00 = RLS1 C2
| |
| |
sRL03 = RLS1 C5

sRL1F = RLS1 C6
sRL1H = RLS1 C7
sRL10 = RLS1 C8
| |
| |
sRL13 = RLS1 C11

sRL2F = RLS1 C12
sRL2H = RLS1 C13
sRL20 = RLS1 C14
| |
| |
sRL23 = RLS1 C17

sRL3F = RLS1 C18
sRL3H = RLS1 C19
sRL30 = RLS1 C20
| |
| |
sRL33 = RLS1 C23

rRL0F thru RL33 = rRL1
rRL1 = sRLC1 $\overline{C14}$

sRLC1 = EOM C10 $\overline{C11}$ $\overline{C13}$ C15 T3
rRLC1 = POT T2 + St

RLS1 = RLC1 $\overline{RLC2}$ POT T3

sRL4F = RLS2 C0
sRL4H = RLS2 C1
sRL40 = RLS2 C2
| |
| |
sRL43 = RLS2 C5

sRL5F = RLS2 C6
sRL5H = RLS2 C7
sRL50 = RLS2 C8
| |
| |
sRL53 = RLS2 C11

sRL6F = RLS2 C12
sRL6H = RLS2 C13
sRL60 = RLS2 C14
| |
| |
sRL63 = RLS2 C17

sRL7F = RLS2 C18
sRL7H = RLS2 C19
sRL70 = RLS2 C20
| |
| |
sRL73 = RLS2 C23

rRL4F thru RL73 = rRL2
rRL2 = sRLC2 $\overline{C15}$

sRLC2 = EOM C10 $\overline{C11}$ $\overline{C13}$ C14 T3
rRLC2 = POT T2 + St

RLS2 = RLC2 $\overline{RLC1}$ POT T3

Table 3-9. 940 Logic Equations (Continued)

Set and reset signals for monitor relabeling register flip-flops:

$$RLs4 = RLC1\ RLC2\ POT\ T3$$

$$rRL4 = sRLC1\ sRLC2$$

$$S3' = \Big[\overline{(SeL'6\ \overline{M63})}\ \overline{(SeL'7\ \overline{M73}\ \overline{Rb})}\ \overline{(Sm\ \overline{S3}\ \overline{S2}\ \overline{Rb})}\ \overline{(Sm\ \overline{S1}\ \overline{S3}\ \overline{Rb})}\ \overline{(Nm\ \overline{S3})}$$
$$\overline{(SeL\ 0\ \overline{RL03})}\ \overline{(SeL\ 1\ \overline{RL13})}\ \overline{(SeL\ 2\ \overline{RL23}\ ReL)}\ \overline{(SeL\ 3\ \overline{RL33}\ ReL)}$$
$$\overline{(SeL\ 4\ \overline{RL43})}\ \overline{(SeL\ 5\ \overline{RL53})}\ \overline{(SeL\ 6\ \overline{RL63}\ ReL)}\ \overline{(SeL\ 7\ \overline{RL73}\ ReL)}\Big]$$

$$sS9 = Sxc\ C6 + Sxp\ P6 + Sxn\ N9 + (66, 67)\ Sx48 + TRAP\ T3$$

$$rS9 = Sc + (66, 67)\ Sd2\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}$$

$$sS12 = Sxc\ C9 + Sxp\ P9 + Sxn\ N12 + (66, 67)\ Sd2\ \overline{S12}\ \overline{S13} + \overline{Ob}\ Pi\ Tte\ T3$$

$$rS12 = (Sc + Sx48) + (66, 67)\ Sd2\ S12\ \overline{S13}$$

$$sS13 = Sxc\ C10 + Sxp\ P10 + Sxn\ N13 + (66, 67)\ Sd2\ \overline{S13} + T3\ Ob\ Pi$$

Table 3-9. 940 Logic Equations (Continued)

$$rS13 = (Sc + Sx48)$$
$$(66, 67) \quad + Sd2\ S13$$

$$sS14 = Sxc\ C11$$
$$+ Sxp\ P11$$
$$+ Sxn\ N14$$
$$(66) \quad + (\phi 3\ O5\ \overline{O6}\ \overline{Sk})\ \overline{S14}$$
$$(67) \quad + (\phi 3\ O5\ O6\ \overline{Sk})\ C5\ (A1\ \overline{A2} + \overline{A1}\ A2)$$
$$+ Ob\ T3$$

$$rS14 = (Sc + Sx48)$$
$$(66) \quad + (\phi 3\ O5\ \overline{O6}\ \overline{Sk})\ S14$$
$$(67) \quad + \left[(\phi 3\ O5\ O6\ \overline{Sk})\ C5\ \overline{(A1\ \overline{A2} + \overline{A1}\ A2)}\ (A2\ \overline{A3} + \overline{A2}\ A3)\right.$$
$$\left.\overline{(\overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13})}\right]$$
$$(67) \quad + (\phi 1\ O5\ T2)\ C2\ (A0\ \overline{A1} + \overline{A0}\ A1)$$

$$SeL\ 0 = \overline{S1}\ \overline{S2}\ \overline{S3}\ ReL$$

$$SeL\ 1 = \overline{S1}\ \overline{S2}\ S3\ ReL$$

$$SeL\ 2 = \overline{S1}\ S2\ \overline{S3}$$

$$SeL\ 3 = \overline{S1}\ S2\ S3$$

$$SeL\ 4 = S1\ \overline{S2}\ \overline{S3}\ ReL$$

$$SeL\ 5 = S1\ \overline{S2}\ S3\ ReL$$

$$SeL\ 6 = S1\ S2\ \overline{S3}$$

$$SeL'6 = S1\ S2\ \overline{S3}\ \overline{Nm}\ Md\ \overline{Rb}$$

$$SeL\ 7 = S1\ S2\ S3$$

$$SeL'7 = S1\ S2\ S3\ \overline{Nm}\ Md$$

Table 3-9. 940 Logic Equations (Continued)

SFM = $\left[\overline{(SeL\ 0\ \overline{RL0F})}\ \overline{(SeL1\ \overline{RL1F})}\ \overline{(SeL\ 2\ \overline{RL2F})}\ \overline{(SeL\ 3\ \overline{RL3F})}\ \overline{(SeL\ 4\ \overline{RL4F})}\right.$
$\left.\overline{(SeL\ 5\ \overline{RL5F})}\ \overline{(SeL\ 6\ \overline{RL6F}\ ReL)}\ \overline{(SeL\ 7\ \overline{RL7F}\ ReL)}\right]$

sSk = $(\phi 5\ O2\ O5)\ \overline{Of}\ C17\ Tr$ (22)
+ (same as 930)

rSk = Ob Tr
+ (same as 930)

Sm = $\overline{Nm}$ Md

Stv = $\overline{Oba}$ Sfm ReL Mxc T2

Sxp = End $\overline{TRAP}$ $\overline{Int}$ T3 Go
+ T3 $\overline{Int}$ Ju $\overline{Eax}$ Go
+ (Kmc) $\overline{Go}$ T3

Tpf = $(\phi 0\ T8\ \overline{Ia})\ \overline{C2}\ \overline{C5}\ \overline{C8}\ (\overline{C3} + \overline{C4})$
+ PiQ
+ T8 (Ob + Ai)

TRAP = Ob + Pi

sTte = T3 EOM C10 $\overline{C11}$ C13 $\overline{C14}$ C15

rTte = T3 TRAP (Um Go) + St T4

Um = $\overline{Nm}$ $\overline{Md}$

UMT = Int T3 Rbs
+ Um T3 TRAP
+ Int Um

Table 3-9. 940 Logic Equations (Continued)

$sUsi = UmT + Um\ C2\ (\phi 0\ T8\ \overline{Ia})\ C0$

$rUsi = (Ju\ \overline{Ts})\ \overline{Int}\ T0 + Usi\ Ib\ T4 + St$

X register delay element:

$Xid = \left[(O1\ \overline{O2}\ O3\ O4\ \phi 6)\ Tr\right] Cz + \left[\overline{(O1\ \overline{O2}\ O3\ O4\ \phi 6)\ Tr}\right] Xw1\ \overline{(Sm\ Rbs\ Tr)} + Sm\ Rbs\ Rb\ Tr$

$\overline{Xz1}$ = the inverse of:

$(Ob + \phi 0 + \phi 7\ O5 + \phi 6\ \overline{O2}\ \overline{O4} + \phi 4\ \overline{O5}\ O6)\ Ix\ Xn1$
(67) (41, 51, 53) (61)

+ (same as 930)

$Xz1$ = the inverse of:

$(Ob + \phi 0 + \phi 7\ O5 + \phi 6\ \overline{O2}\ \overline{O4} + \phi 4\ \overline{O5}\ O6)\ (\overline{Ix} + \overline{Xn1})$
(67) (41, 51, 53) (61)

Similarly Xz2, Xz3

$sZhp = \overline{Wti}\ \overline{Zhp}\ T3$

$rZhp = T4$

IV. INSTALLATION AND MAINTENANCE

4.1 GENERAL

Since the 940 Computer is a 930 Computer modified for adaptation to the time-sharing software, installation and maintenance of the 940 Computer is similar in all respects to installation and maintenance of the 930 Computer. With the exception of the memory stack temperature adjustment, which has been modified, basic installation and maintenance information is contained in the 930 Computer Technical Manual (SDS Publication Number 900066C). The contents of this section is limited to information not contained in the 930 Computer manual.

4.2 GENERAL INSTALLATION INFORMATION

Since detailed installation information is presented in the 930 Computer Technical Manual, the following paragraphs present general installation information regarding the differences between the 930 Computer and the 940 Computer and the installation procedure required for adding a fourth memory bank.

The 940 Computer is always installed by SDS personnel and details of installation are not generally required by the user. The slight differences between the 930 Computer and the 940 Computer regarding sizes are presented in Figure 4-1. Although only one input/output cabinet is shown, additional cabinets may be required to house the input/output hardware. Each additional memory (two maximum) is housed in an individual cabinet along with power supplies. All cabinets are 62.50 inches high and set 0.94 inches off the floor. The Control Console installation is presented in SDS Drawing Number 107946. Various power and air conditioning variations between the 940 Computer and the 930 Computer are noted in Figure 4-2. Additional installation information can be found in Section IV of the 930 Computer Technical Manual.

4.3 MAINTENANCE

In the general sense, maintenance means to keep the computer and memory functioning, or to restore the computer and memory to a functional condition after malfunction. The following paragraphs present general maintenance information to assist in locating the necessary information as well as specific information on the relabeling registers that have been added to the 940 Computer.

4.3.1 GENERAL MAINTENANCE INFORMATION. Any discussion of maintenance hinges on a thorough understanding of the computer. Between the 930 Computer Technical Manual and the contents of this manual, maintenance personnel are presented with all available technical data. Where programming knowledge is required, no adequate framework exists to provide maintenance personnel with the necessary information. Extensive programming information is available in the form of reference manuals, programming manuals, etc. The maintenance personnel are recommended to read carefully the description of the logic symbology used in the 940 Computer documentation as presented at the beginning of Section III.

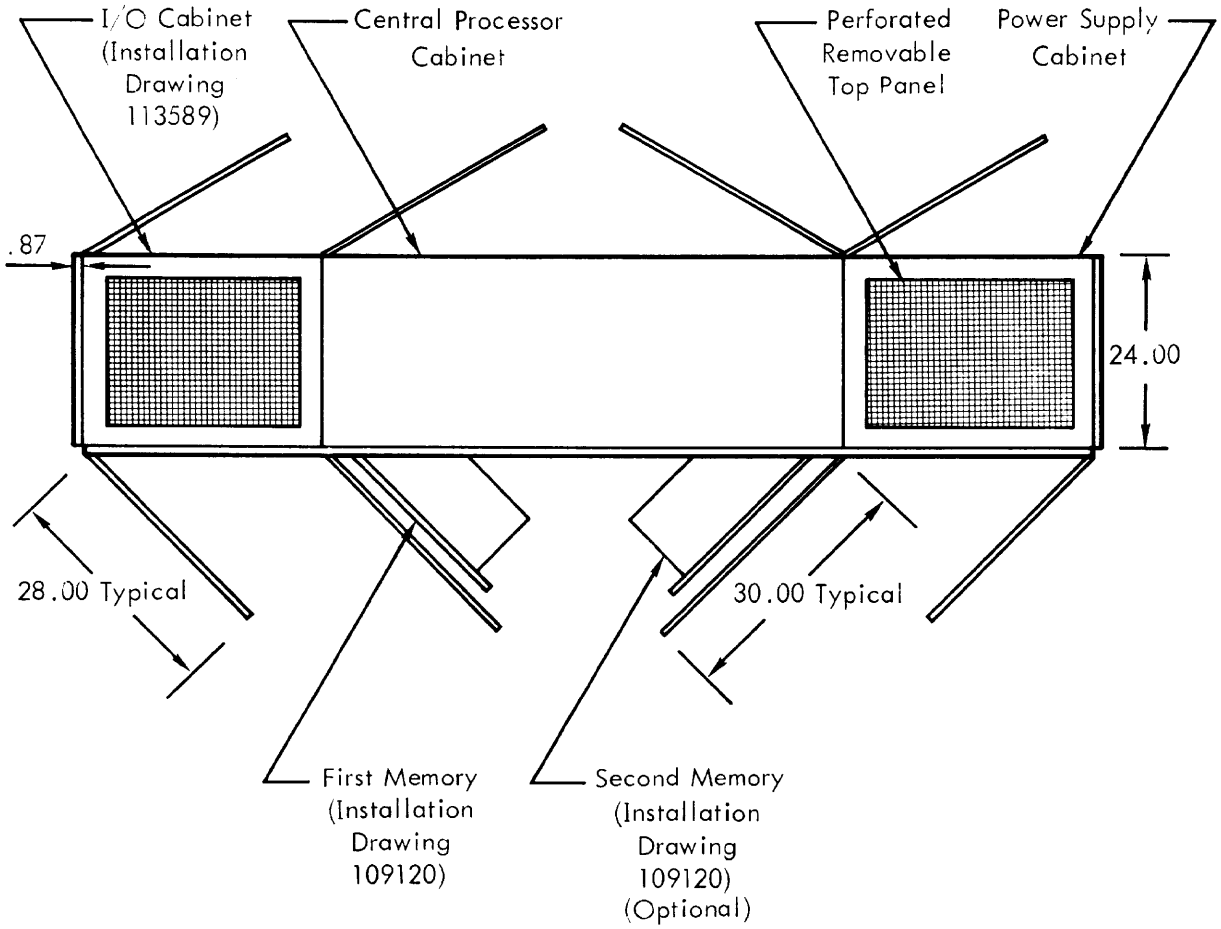


Figure 4-1. 940 Computer Size Information

4.3.2 RELABELING REGISTERS. Three relabeling registers are used in the 940 Computer. Relabeling Registers 1 and 2 make up the User Map. Relabeling Register 4 is the Monitor Map. There is no Relabeling Register 3.

Relabeling Register 1 comprises 24 flip-flops designated RL0F, RL0H, RL00 through RL03; RL1F, RL1H, RL10 through RL13; RL2F, RL2H, RL20 through RL23; and RL3F, RL3H, RL30 through RL33. Relabeling Register 2 comprises 24 flip-flops designated RL4F, RL4H, RL40 through RL43; RL5F, RL5H, RL50 through RL53; RL6F, RL6H, RL60 through RL63; and RL7F, RL7H, RL70 through RL73. The relation between the C-register output lines and the Relabeling Register contents are presented in Table 4-1. The C-register column shows the C-register bit position that is gated into the flip-flop in the RL column during the POT instruction that loads the register. The output pin of the flip-flop is listed under the POINT column. The registers are selected, cleared, and loaded by an EOM/POT sequence. Two select flip-flops (RLC1 and RLC2) enable the data to be gated into the proper register. Relabeling Register 4 comprises 10 flip-flops designated M6H, M60 through M63; and M7H, M70 through M73. The relation between the C-register output line and the relabeling register contents are presented in Table 4-2. The Relabeling Register is also selected, cleared and loaded by means of an EOM/POT sequence.

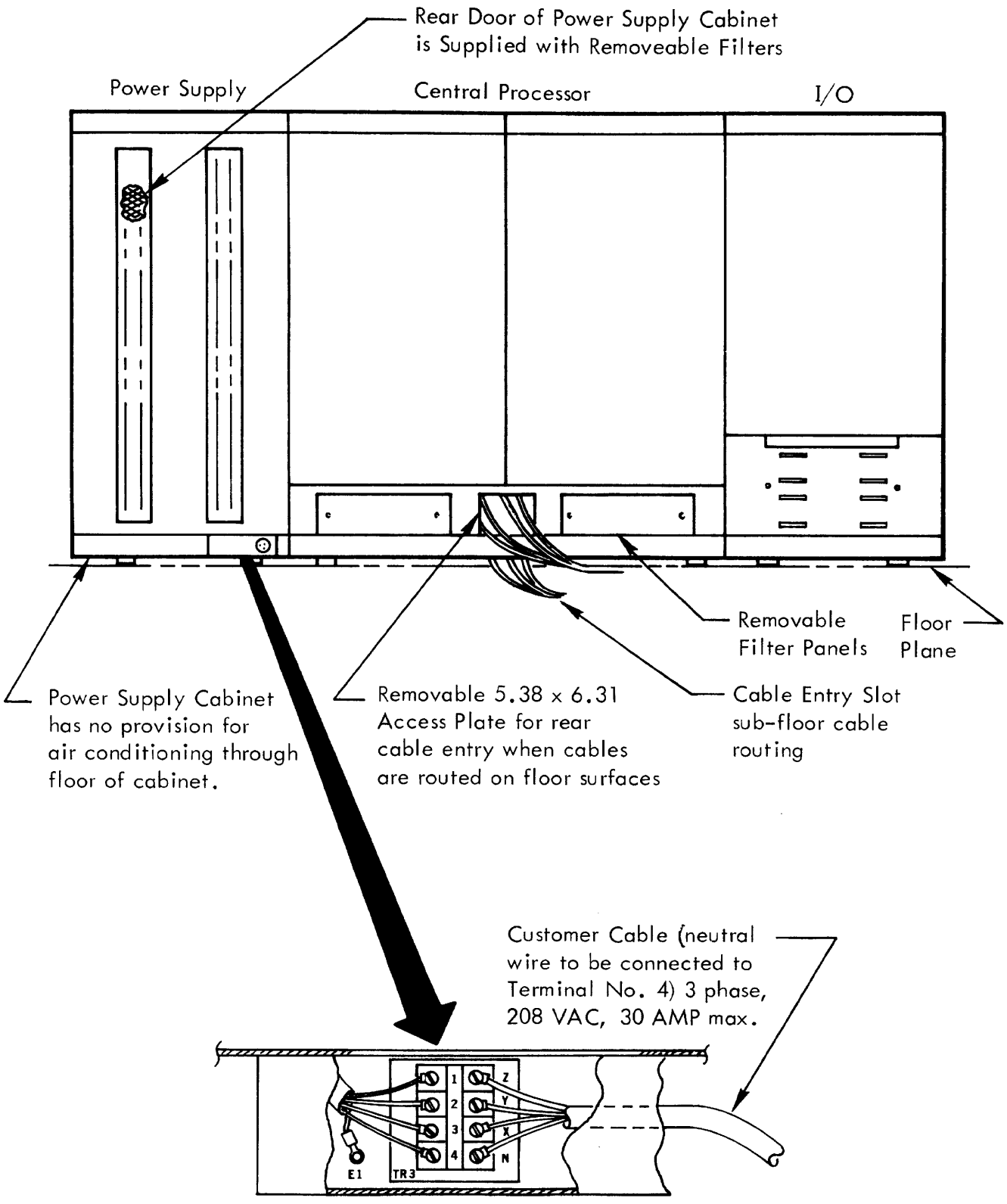


Figure 4-2. 940 Computer Power and Air Conditioning Information

Table 4-1. Relabeling Register 1 and 2 Information

User Relabeling Registers

C-Register Output Lines	RL1	Point	RL2	Point
0	$\overline{\text{RL0F}}$	49A16	$\overline{\text{RL4F}}$	48A16
1	RL0H	46A24	RL4H	47A24
2	RL00	50A2	RL40	51A2
3	RL01	50A23	RL41	51A23
4	RL02	50A24	RL42	51A24
5	$\overline{\text{RL03}}$	49A8	$\overline{\text{RL43}}$	48A8
6	$\overline{\text{RL1F}}$	49A32	$\overline{\text{RL5F}}$	48A32
7	RL1H	53A2	RL5H	52A2
8	RL10	18F2	RL50	19F2
9	RL11	18F23	RL51	19F23
10	RL12	18F24	RL52	19F24
11	$\overline{\text{RL13}}$	20F8	$\overline{\text{RL53}}$	21F8
12	$\overline{\text{RL2F}}$	46A8	$\overline{\text{RL6F}}$	47A8
13	RL2H	53A23	RL6H	52A23
14	RL20	20F24	RL60	21F23
15	RL21	20F24	RL61	21F24
16	RL22	22F2	RL62	23F2
17	$\overline{\text{RL23}}$	22F16	$\overline{\text{RL63}}$	23F16
18	$\overline{\text{RL3F}}$	46A16	$\overline{\text{RL7F}}$	47A16
19	RL3H	53A24	RL7H	52A24
20	RL30	22F24	RL70	23F24
21	RL31	24F2	RL71	26F2
22	RL32	24F23	RL72	26F23
23	$\overline{\text{RL33}}$	24F32	$\overline{\text{RL73}}$	26F32

Table 4-2. Relabeling Register 4 Information

Monitor Relabeling Register		
C-Register Output Lines	RL4	Point
13	M6H	54A23
14	M60	54A24
15	M61	54A2
16	M62	55A23
17	$\overline{M63}$	55A8
19	M7H	55A24
20	M70	56A23
21	M71	56A24
22	M72	56A2
23	$\overline{M73}$	57A16

4.3.3 ADDITIONAL MAINTENANCE INFORMATION. Additional information regarding drawings that might be useful in maintaining the 940 Computer will be found in Section V of this manual. Additional 940 Documentation is listed in Section VI. A complete module list for both the computer and memory is included in Section VII.

V. DRAWINGS

5.1 GENERAL

An extensive drawing package is supplied with the 940 Computer. In this section, drawings are listed in four categories: drawings lists, assembly drawings, logic layouts and wire lists, and cable assembly drawings according to system components.

5.2 940 COMPUTER

The drawing list for the 940 Computer is presented in Drawing Number 126182. The assembly drawings are presented in Table 5-1. Logic layouts and wire lists for the 940 Computer are presented in Table 5-2. Table 5-3 contains cable assembly drawings.

Table 5-1. Computer Assembly Drawings

Drawing Number	Title
125145	Fourth Memory Installation Procedure
126142	940 Computer Assembly
126174	Central Processor Assembly
126175	Computer Chassis Assembly
126176	Main Frame Assembly
126184	Installation Drawing
126186	Test, Procedure, Acceptance Specification
126187	6 Chassis (RH) Assembly
126188	Module Guide Assembly
126218	Specification, Test Procedure

Table 5-2. Computer Logic Layouts and Wire Lists

Drawing Number	Title
126177	List, Wire-Computer Main Frame
126178	List, Wire Index
126179	List, Wire-Main Frame Power
126183	Diagram, Logic
126185	Logic Equations

Table 5-3. Computer Cable Assembly Drawings

Drawing Number	Title
106336	Assembly, Cable Plug Module
106337	Assembly, Cable Plug Module
106338	Assembly, Cable Plug Module
107203	Assembly, Cable Plug Module
107317	Assembly, Cable Plug Module

5.3 MEMORY

Drawing lists for the memory are presented in Table 5-4. Memory assembly drawings are listed in Table 5-5. Table 5-6 contains the logic layout and wire list drawing numbers for the basic 16,384-word memory. Cable assembly drawings for the memory are listed in Table 5-7.

Table 5-4. Memory Drawing Lists

Drawing Number	Title
150466	Drawing List, Basic 16,384-Word Memory
150524	Drawing List, Jumper Module (16K)
150526	Drawing List, Jumper Module (16K)
150528	Drawing List, Jumper Module (16K)
150530	Drawing List, Jumper Module (16K)

Table 5-5. Memory Assembly Drawings

Drawing Number	Title
150462	Assembly, 16,384-Word Memory
150463	Assembly, Computer Memory Chassis
150464	Assembly, Memory Chassis
150467	Assembly, 16K Memory and Modules
150473	Chart, Module Location, Computer Memory Chassis

Table 5-6. Memory Logic Layouts and Wire Lists

Drawing Number	Title
126196	Diagram, Logic - Memory
150465	List, Wire-Memory Chassis
150474	Equations, 16K Memory Logic

Table 5-7. Memory Cable Assembly Drawings

Drawing Number	Title
107202	List, Wire, Cable Plug Module
107639	Assembly, P.W., Cable Plug Module
107640	Assembly, P.W., Cable Plug Module
107641	Assembly, P.W., Cable Plug Module
126189	List, Wire, Cable Plug Module (Switch)
126194	Schematic Cable Plug Module (Switch)
126211	Assembly, P.W., Cable Plug Module (Switch)
126213	Assembly, P.W., Cable Plug Module (Switch)

5.4 OPTIONS

The basic 940 Computer requires additional hardware to perform the time-sharing operations. The following paragraphs present the drawings supplied for these options.

5.4.1 MULTIPLE ACCESS TO MEMORY. The drawing list for the Multiple Access to Memory (MAM) is presented in drawing number 150472. Table 5-8 contains numbers of drawings supplied for the MAM.

Table 5-8. MAM Drawings

Drawing Number	Title
126212	Assembly, P.W., Cable Plug Module (P222-P226)
150469	Multiple Access to Memory Assembly
150470	Multiple Access to Memory Assembly
150471	Multiple Access to Memory Assembly

5.4.2 DIRECT ACCESS COMMUNICATION CHANNEL. The drawing list for the Direct Access Communication Channel (DACC) is presented in drawing number 126204. Table 5-9 contains numbers of drawings supplied for the DACC.

Table 5-9. DACC Drawings

Drawing Number	Title
126198	Installation Drawing, Input/Output Buffer (DACC)
126199	Assembly, Input/Output Buffer (DACC)
126200	Assembly, Input/Output Chassis (DACC)
126202	Module Location Chart, Input/Output Buffer (DACC)
126203	Logic Diagram, Input/Output Buffer (DACC)
126205	Equations, DACC Logic
126201	Wire List, Input/Output Buffer (DACC)
126208	Wire List Index, Input/Output Buffer (DACC)

5.4.3 TIME MULTIPLEXED COMMUNICATION CHANNEL. The drawing list for the Time Multiplexed Communication Channel (TMCC) is presented in drawing number 126210. Table 5-10 contains numbers of drawings supplied for the TMCC.

Table 5-10. TMCC Drawings

Drawing Number	Title
126207	Assembly, Input/Output Chassis (TMCC)
150475	Assembly, Input/Output Buffer (TMCC)
126209	Logic Diagram, Input/Output Buffer (TMCC)
126181	Equations, Input/Output Buffer (TMCC)
150476	Wire List, Input/Output Buffer (TMCC)

5.4.4 WY INPUT/OUTPUT BUFFER. The drawing list for the WY Input/Output Buffer is presented in drawing number 150606. Table 5-11 contains numbers of drawings supplied for the WY Buffer.

Table 5-11. WY Buffer Drawings

Drawing Number	Title
105477	Assembly, WY Input/Output Chassis (TMCC)
150605	Assembly, WY Input/Output Buffer (TMCC)

5.4.5 ZB65 JUMPER MODULE. Table 5-12 contains numbers of drawings supplied for the ZB65 Jumper Module.

Table 5-12. ZB65 Jumper Module Drawings

Drawing Number	Title
150479	Assembly, P.W., Jumper Module (16K)
150481	Assembly, P.W., Jumper Module (16K)
150482	Assembly, P.W., Jumper Module (16K)
150480	Assembly, P.W., Jumper Module (16K)

VI. STANDARD EQUIPMENT MANUALS

6.1 GENERAL

The data package supplied with the system includes available documentation on the computer and the peripheral devices used. For convenience, these documents are divided into two lists in the paragraphs that follow.

Table 6-1. 940 Computer Documentation

Publication Number	Title
64-03-13	930 Computer Brochure
900064	930 Reference Manual
64-70-04	SDS Programming Systems
900066	930 Theory of Operation
900561	Interface Manual
900636	CPU and Memory Logic Equations
900685	92200/93200 TMCC Theory of Operation
900648	930 Computer Installation Data Sheet
900689	Memory Trouble-Shooting Guide
900677	Printed Circuit Board, Component Replacement Procedure
64-03-14	925/930/9300 - Input/Output
900592	930 CPU Logic Diagrams
900620	930 Memory Logic Diagrams
900557	930 TMCC I/O Logic Diagrams
900608	930 Basic Interrupt Logic Diagrams
900619	General Reference Drawings
64-65-05	Spare Parts List
900036	Suppliers Code Index
900665	Semiconductor Cross-Reference List
900623	Module Reference Data
900707	PX18 Power Supply
900708	PX19 Power Supply

Table 6-1. 940 Computer Documentation (cont)

Publication Number	Title
900709	PX20/21 Power Supply
900710	PX22 Power Supply
900711	PX23 Power Supply
900713	PX25 Power Supply
900001	PX14/15 Power Supply
900865	Memory Trouble-Shooting Manual

VII. MODULE DATA SHEETS

7.1 GENERAL

Detailed information concerning the individual module cards within this system is provided on the individual Module Data Sheets. Table 7-1 lists logic modules for the 940 Computer. Table 7-2 lists logic modules for the memory.

Table 7-1. Computer Logic Modules

Publication Number	Number	Title
900259A	AB51	Cable Driver
900260A	AB52	Receiver Inverter
900261A	AB53	Receiver Inverter Buffer
900262	AB55	Cable Driver #2
900263	CB50	Clock Generator
900276	*DB50	Shift Register
900291	*FB50	DC Flip-Flop
900257A	FB51	Dual Flip-Flop
900341A	FB52	Triple Flip-Flop
900256A	FB54	NAND Flip-Flop
900748A	IB52	NAND Module
900265A	IB56	NAND #2
900344A	IB57	BAND NAND
900345A	IB58	NAND #3
900347A	NB50	Interface +8 to +4
900305A	*NB53	Indicator Interface
900306	*NB57	Interface Module
900271	*ZB50	Termination Module (+4)
900349	*ZB52	Cable Termination Module
900385	*ZB68	Termination Module

* Not available at time of printing

Table 7-2. Memory Logic Modules

Publication Number	Number	Title
900260A	AB52	Receiver Inverter
900262	AB55	Cable Driver #2
900269A	AB56	Z-Driver
900291	*FB50	DC Flip-Flop
900284A	HB 53-2	Sense Amplifier Shield
900748A	IB52	NAND Module
900265A	IB56	NAND #2
900344A	IB57	BAND NAND
900284A	IB59	NAND #4
	**OB50-2	One-Shot Multivibrator
900267A	*QB53	Sink Switch
900268A	*QB54	Driver Switch
900271	*ZB50	Termination Module +4
900349	*ZB52	Cable Termination Module
900274	*ZB63	ZY Resistor Module
900273	*ZB64	Z-Resistor Module
	**ZB65-60	Jumper Module (16K)
	**ZB65-61	Jumper Module (16K)
	**ZB65-62	Jumper Module (16K)
	**ZB65-63	Jumper Module (16K)

* Not available at time of printing

**No Data Sheet available