SERVICE MANUAL

SINGER MODEL 1501
INTELLIGENT TERMINAL

SINGER
BUSINESS MACHINES

SERVICE MANUAL

SINGER MODEL 1501

INTELLIGENT TERMINAL

23-0563-300
EDITION B
APRIL 1976
CHANGE DECEMBER 1976

SINGER
BUSINESS MACHINES

PRINTED IN U S A.

Edition Letter	Suffix	Description
None	00	Original Printing - April 1975
B	01	Reprinted - April 1976
B	02	First Revision - December 1976

LIST OF EFFECTIVE PAGES

The total number of pages in this publication is 238, consisting of the following:

Page No.	Issue
*Title	December 1976
*Revision Record	December 1976
*A	December 1976
*i thru ii	December 1976
iii	Original
*iv thru vii/viii	December 1976
1-1	Original
*1-2 thru 1-8	December 1976
*2-1	December 1976
2-2 thru 2-3	Original
*2-4	December 1976
2-5	Original
*2-6	December 1976
2-7 thru 2-22	Original
*2-23	December 1976
2-24	Original
*2-25	December 1976
2-26 thru 2-31	Original
*2-32	December 1976
2-33 thru 2-34	Original
*2-35	December 1976
2-36 thru 2-40	Original
*2-41 thru 2-42	December 1976
*2-42A thru 2-42D	December 1976
2-43 thru 2-46	Original
*2-47 thru 2-48	December 1976
2-49 thru 2-53	Original
*2-54	December 1976
2-55 thru 2-57	Original
*2-58 thru 2-59	December 1976
2-60 thru 2-78	Original
*2-79	December 1976
2-80 thru 2-106	Original
*2-107	December 1976
2-108 thru 2-170	Original
*2-171	December 1976
2-172 thru 2-175	Original
*2-176	December 1976
3-1	Original
*3-2	December 1976
3-3 thru 3-19	Original
*3-20 thru 3-30	December 1976
4-1 thru 4-3/4	Original
5-1 thru 5-12	Original

*This asterisk indicates pages changed, added, or deleted by the current change. Insert the latest changed pages; destroy superseded pages.

SECTION PAGE

CONTENTS (Continued)

CONTENTS (Continued)

SECTION PAGE

CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

TABLES

1-1. PURPOSE AND USE OF THE INTELLIGENT TERMINAL.

The Singer Model 1501 Intelligent Terminal is a desk-top unit that has a wide range of data processing applications, such as inventory control, payroll processing, sales analysis, and purchasing. These various functions can be accomplished by the use of different pre-recorded program tapes.

The Model 1501, shown in figure 1-1, consists of a mini-computer, keyboard, tape drives, and a display screen. Within the unit are the processor, memory, and input-output controller which comprise the computer. These components are printed circuit boards. Two cartridge tape drives are mounted on the upper right section of the unit. Magnetic tape cartridges loaded on these drives provide programs and store data for the computer.

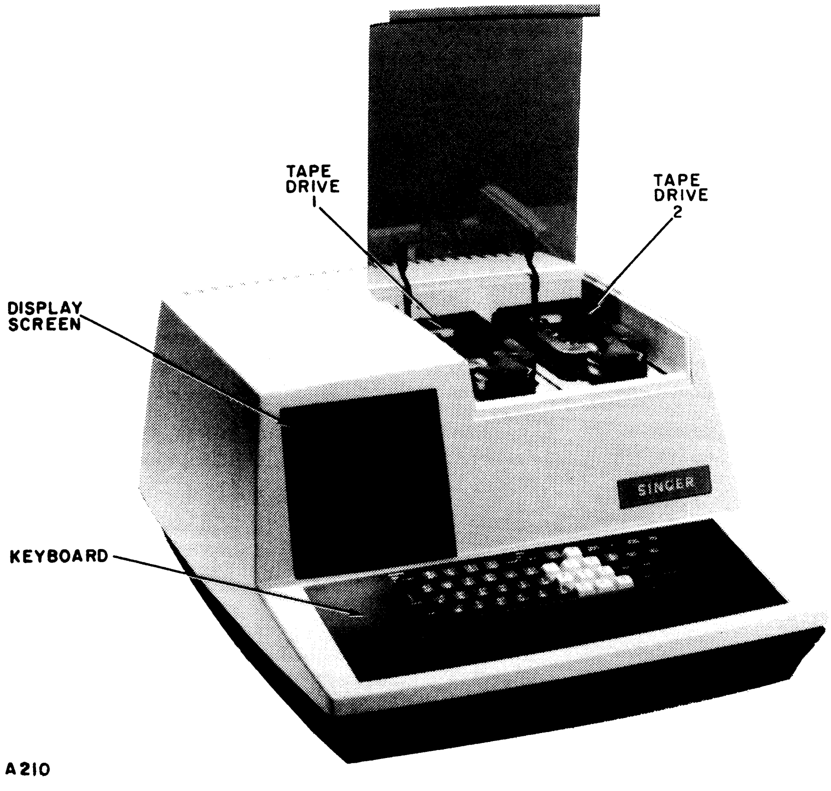


Figure 1-1. Model 1501 Intelligent Terminal

A CRT and keyboard are housed in the terminal. These components allow the operator to perform the data processing for which the terminal is intended. The keyboard allows the operator to enter data and control operation of the computer, while the CRT displays data, questions, and instructions to assist the operator.

The terminal can also communicate with other terminals and peripheral equipment, such as tape units and printers, over a serial input-output channel operated by the I/O controller. With the addition of a communications adapter board to the main chassis and an external MODEM (modulator-demodulator), the terminal also has the capability to transmit and receive over telephone lines.

A well beneath the display screen houses the program load and system reset switches. After power is applied to the terminal, the first program is loaded into the processor's memory from a tape drive cartridge. This is accomplished when the operator presses forward the program load switch. This loads the program stored on tape into the processor memory. From this point on, the operation of the processor is under control of that program.

1-2. MODELS AND OPTIONS AVAILABLE.

There are three basic models of the 1501. They are: (1) the standard 1501; (2) the 1501-FF, which includes a transaction counter; and (3) the 1501-CL, which does not include tape drives and is intended to be operated in the clustered mode with other terminals. The suffix "CL" means clustered.

There are several optional features that can be added to the terminal to increase its capabilities. These options and the basic Intelligent Terminal part numbers are listed in table 1-1. Where the suffix "XXX" appears in the table, it indicates that the item is available in several different variations in which such things as power requirements, keyboard keytops, and memory capacity differ. Of course, other options may have been added or part numbers changed since this manual was published so only the Illustrated Parts Manual and the sales representative's catalog should be used for selecting and ordering options.

Table 1-1. Models and Options Available

Model	Part Number
1501 Intelligent Terminal	001-003200-XXX
1501-FF Intelligent Terminal	001-002994-XXX
1501-CL Intelligent Terminal	001-001000-XXX
1534A Asynchronous Communications Adapter	003-001907-XXX
1535A Binary Synchronous Communications Adapter	003-001907-039
1530-1, 13-Key Numeric Pad	003-003233-XXX
Booster Transformer Option for 100 VAC Input	003-003350-006
*Dual SIO Option	003-002830-008
1533 Dual Drive	003-001574-XXX

*Must be included with 1501-CL

1-3. LOCATION OF MAJOR COMPONENTS.

The location of the tape drives, CRT, and keyboard assembly are shown in figure 1-1. However, most of the electronic components are mounted inside the terminal, and the location of these components is shown in figure 1-2.

1-4. CONTROLS AND INDICATORS.

The Model 1501 Intelligent Terminal has relatively few operator controls aside from the keyboard and pad. All of these controls are described in the Operator Instructions Manual; however, those of interest to the field service engineer are summarized in the following paragraphs.

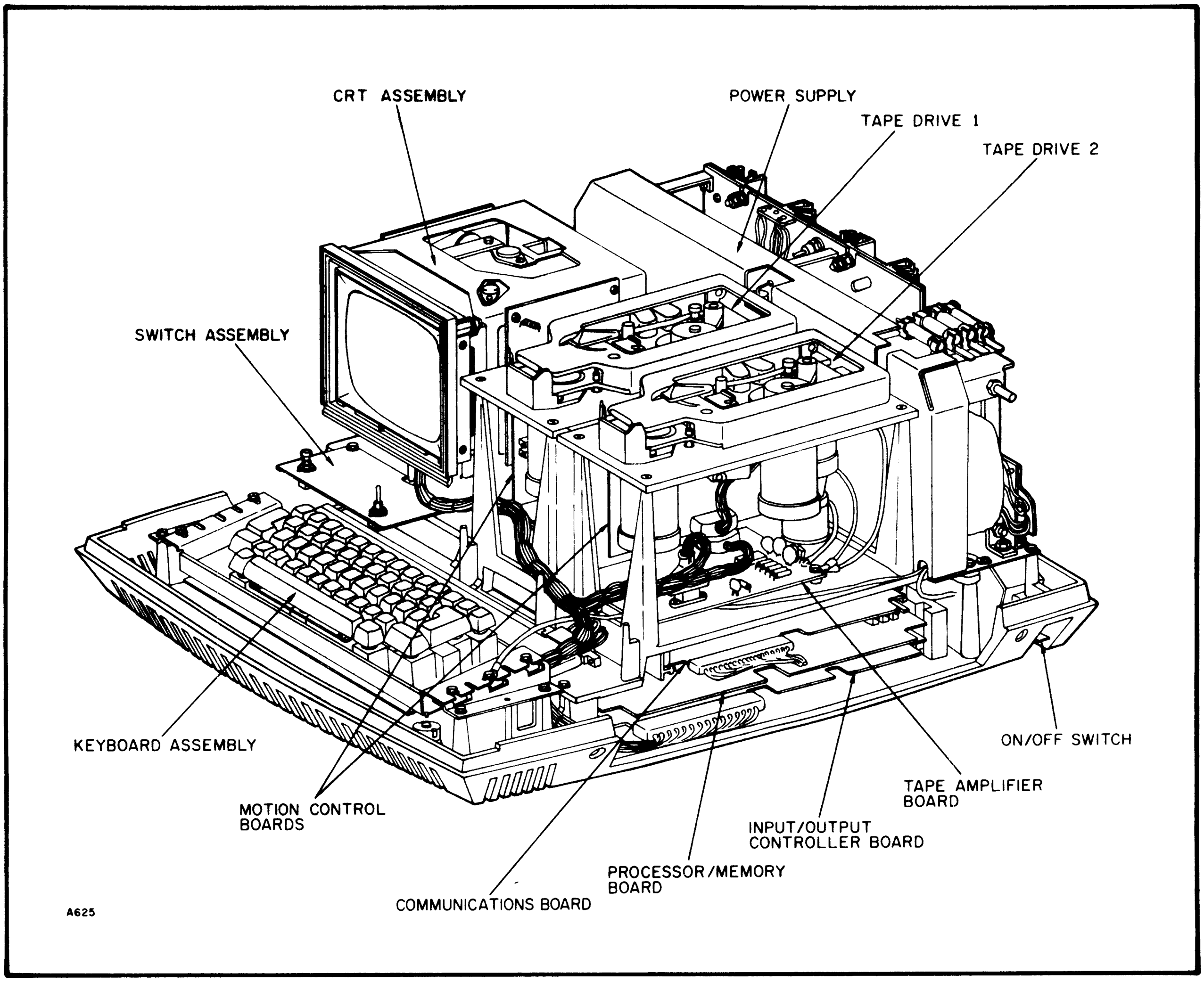


Figure 1-2. Location of Major Components

Following are the three controls used to apply power and start the terminal:

a. The POWER ON/OFF switch, located on the right rear bottom of the side panel.

b. The PROGRAM LOAD switch, located in the well beneath the display screen. This switch, when pushed to the PROGRAM LOAD position, causes the program to be loaded into the processor memory. Except for the 1501-CL, the source of the program is the tape cartridge on tape drive 2. In the 1501-CL, the program is available from another terminal over the serial input-output channel. This switch has three positions; PROGRAM LOAD, off, and program interrupt. The interrupt position, which is not labeled, causes the processor program to jump to a specific memory location (page 03, location 000) to perform an interrupt servicing routine.

c. The SYSTEM RESET switch is located in the well beneath the display screen. This switch causes the processor program to jump to a specific memory location (page 02, location 000) and start the program located there. The action the processor takes when the SYSTEM RESET switch is used depends upon the program currently stored in the memory.

Both the normal operating programs and the diagnostic programs are stored on tape cartridges. Located on the outer case of the cartridge is a two-position hole for the write pin. If the write pin is inserted into the position on top of the cartridge, the tape cannot be written on. If the write pin is inserted into the position on the bottom of the cartridge, the tape may be written on; this is called the "write-enable" position.

Loading the cartridge is a simple procedure that is fully explained in the Operator Instructions Manual. Once the cartridge is placed in the loader, the loader must be pushed down and the locking lever moved to the lock position. Of course, this procedure is reversed in order to unload a tape.

Normally, the processor rewinds the tape before instructions are given to remove the cartridge. If, however, the tape is to be removed before it has been rewound by the processor, it is necessary to depress the red rewind pushbutton on the tape drive. Once this button has been depressed, the tape will be rewound and can be unloaded.

1-5. CAPABILITIES.

The capability of each terminal depends upon the options selected. In the following paragraphs, the capabilities of the standard components are described first. Following this description, the options available are discussed.

Display and Keyboard.

The terminal houses a 5-inch CRT on which a visual display is created. Up to 8 rows of 32 characters each can be shown. Each character is formed on a 5 x 7 dot matrix and is selected by the processor. In the standard character set, 64 different characters are available, including the alphabet, numerals, and punctuation marks. However, the keytops can be replaced with any character set, and the processor program can create any character set on the display. The keyboard consists of a 52-key alphanumeric keyboard and an optional 13-key keypad. A speaker is housed in the terminal to provide audible cues to the operator.

Processor.

Operation of the terminal is controlled by a computer housed within the terminal. This computer consists of a general-purpose processor, a memory, and I/O controller. The processor is a stored program, general-purpose machine, with repertoire of 39 instructions. The format of all instructions and data in the processor is organized into bytes (8 bits). Instruction execution time ranges from three to six microseconds.

Memory.

Memory capacity is a maximum of 16,384 bytes in 4096-byte increments. The memory is random access with read/write capability.

I/O Controller.

An I/O controller is provided to handle data transfer between most I/O devices and the processor. This unit operates the cartridge tape drives, the CRT and keyboard assembly, and a serial input-output channel. This channel can be cabled to as many as 64 terminals or peripheral equipment units. Cable length is a maximum of 1500 feet and cable impedance is 93 ohms.

Cartridge Tape Units.

Except for the 1501-CL version, two cartridge tape units are standard in the 1501. Each holds one cartridge like that shown in figure 1-3. The data is recorded serially on a single channel tape at a density of 1600 bits per inch. Capacity of each cartridge is 900 records of 143 bytes each. Reading and writing speed is 10 inches per second, and the tape is driven at 40 inches per second during search and rewind operations.

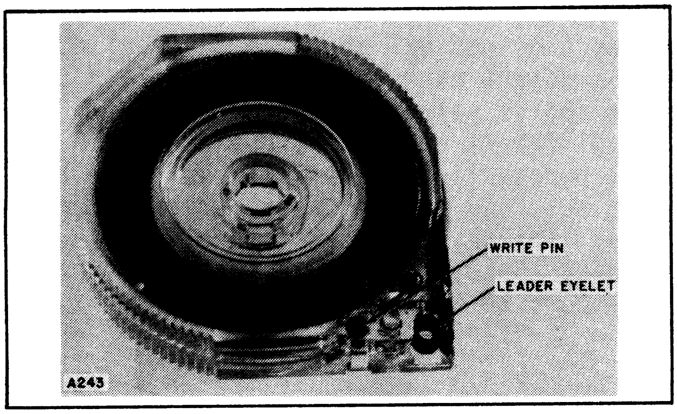


Figure 1-3. Tape Cartridge

Dual Serial Input-Output (SIO).

The standard terminal can communicate with other Series 1500 equipment over one SIO channel. The addition of a dual SIO board provides two SIO channels.

Communications Adapters.

The communications adapters are single boards that are added to the 1501 to allow the terminal to communicate with a MODEM and subsequently with telephone lines. Either the Model 1534 Asynchronous Communications Adapter or the Model 1535 Binary Synchronous Communications Adapter can be used; however, the terminal can accommodate only one such board. This option is either factory or field installable.

1-6. WEIGHT AND DIMENSIONS.

The Model 1501 Intelligent Terminal weighs approximately 64 pounds (29 kg). Its dimensions are:

Height - 10" (25 cm)
Width - 18.5" (47 cm)
Depth - 24" (60 cm)

1-7. POWER REQUIREMENTS.

The Model 1501 Intelligent Terminal is available in versions that operate on 100, 115, or 230-volt AC supplies. However, the older models with the Cogar power supply can operate only on 115 or 230-volt inputs. These models require the Booster Transformer Option if only a 100-volt input is available.

Voltage	100V RMS nominal, 50-60 Hz, Single Phase
Range	90V - 110V RMS
Voltage	115V RMS nominal, 50-60 Hz, Single Phase
Range	104V - 129V RMS
Voltage	230V RMS nominal, 50-60 Hz, Single Phase
Range	208V - 258V RMS

1-8. ENVIRONMENTAL REQUIREMENTS.

The Model 1501 Intelligent Terminal is designed to operate in a normal office environment. Listed below are the temperature and humidity limitations for both normal operation and transportation and storage.

Ambient Temperature

Operating:	+60°F to 90°F (15.6°C to 32°C)
Storage:	-25°F to +135°F (-32°C to 57°C)

Relative Humidity

Operating:	20 to 80 percent
Storage:	5 to 95 percent

2-1. GENERAL.

The purpose and organization of the Model 1501 Intelligent Terminal were briefly discussed in Section 1 of this manual. This section begins with a review of that material in order to introduce the detailed descriptions that follow.

The Model 1501 is a small computer that includes a processor, a memory, and an input-output controller. These components are printed circuit boards housed within the terminal. Two tape drives, which are operated by the computer, are mounted on the top right of the terminal. Magnetic tape cartridges loaded on these drives provide programs for the computer and store data produced by the computer.

A CRT and keyboard, housed in the terminal, allow the operator to perform the data processing for which the terminal is intended. The keyboard allows the operator to enter data and control operation of the computer, while the CRT displays data, questions, and instructions to assist the operator.

The terminal can also communicate with other terminals and peripheral equipment, such as tape units and printers, over a serial input-output channel operated by the I/O controller. With the addition of a communications adapter board and an external MODEM (modulator-demodulator), the terminal also has the capability to transmit and receive over telephone lines.

The processor, memory, and I/O controller make up a small, stored-program, general-purpose computer. After power is applied to the terminal, the first program is loaded into the processor's memory from the tape cartridge on the second tape drive. This is accomplished when the operator pushes the PROGRAM LOAD switch, which causes the processor to perform a short loading program stored in a read-only memory on the processor board. This program loads the program stored on the tape into the processor memory. From this point on, the operation of the processor is under control of that program. The 1501-CL model does not have tape drives, however, and it acquires programs over the serial input-output channel.

The processor has 41 different instructions in its instruction repertoire, and there are many input-output instructions that are executed by the I/O controller. Since the data flow between units depends upon how the programmer uses these instructions, the following description discusses the basic flow of data that can be accomplished rather than how data is processed by any specific program. Refer to the block diagram in figure 2-1 for the following discussion.

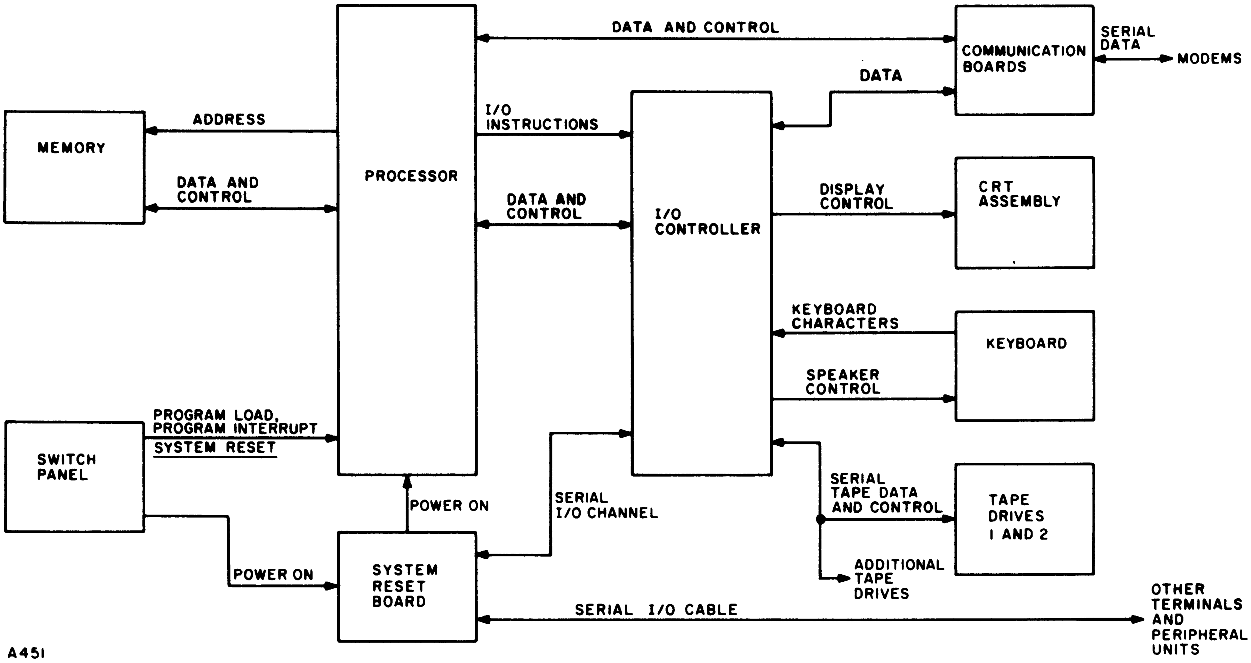


Figure 2-1. Model 1501 Intelligent Terminal Block Diagram

Organization and Use of the Memory.

The memory in the 1501 Intelligent Terminal is expandable from a minimum of 4,096 bytes to 16,384 bytes in 4,096-byte increments. The memory is housed on the processor board.

The processor controls access to the memory through an address net and access control circuits. Either the processor or the I/O controller can supply an address to the address net; however, access to the memory is coordinated with the basic processor timing cycles, and the controller is allowed to gain access to memory only at specific points in the processor cycle.

Data to be stored in memory is routed to a memory input multiplexer, and is gated through the multiplexer under control of the processor. When data is read from memory, it is available through the memory output multiplexer. There, it is available to the processor and the I/O controller.

Exchange of Data With the I/O Controller.

The I/O controller is a printed circuit board mounted in the terminal. Its purpose is to control the keyboard, the CRT, the tape drives, and the serial I/O channel. It coordinates the exchange of data between these units and the memory.

The processor initiates operation of the I/O controller when it decodes an instruction read from memory and finds it is an I/O instruction. (All I/O instructions have an operation code that begins with 17.) Sensing that the current instruction is an I/O instruction, the processor gates the instruction to the I/O controller for execution. There, it is decoded further to determine which device is selected and what action is required. From this point on, the I/O controller takes the action necessary to carry out the instruction independently of the processor except when data must be exchanged with the memory.

Functions performed by the I/O controller are:

a. Produce the CRT display.

b. Control the keyboard and accept keyboard inputs.

c. Control the tape drives, including reading and writing.

d. Control the exchange of data through the communication adapters.

e. Control operation of the serial I/O channel.

Display Generation.

The standard display unit is a 5-inch CRT that is operated by circuits in the I/O controller. The display area on the CRT consists of 8 rows of 32 character positions. To initiate the display on a page of data, the processor sends an I/O instruction to the I/O controller. This instruction selects the CRT circuits and specifies the memory location of the data to be displayed. Since each page on the display consists of 256 characters, one "page" of memory, which consists of 256 bytes, is required to produce the display. Once the processor transfers the page number to the I/O controller, the controller requires no additional processor inputs to produce the display.

The display is produced by a scanning process similar to that in a television receiver, except that the fast scan is vertical and the slow scan is horizontal. Each character is composed of columns of dots. As the scan approaches each character position, the controller gains access to the selected memory page and reads the code for the character to be displayed in that specific position. Based on that code, the controller again gains access to memory and this time reads the dot pattern required to construct the selected character. The dot pattern then causes the CRT beam to be unblanked at the appropriate position to produce dots that form the selected character. The only signals routed to the CRT assembly by the controller are the unblanking signals and the triggers required to begin and end the vertical and horizontal scans.

Once a page number has been selected for display, the I/O controller continues to read that page and produce the display. Of course, the processor can store new data in the selected page and this causes the display to change on the screen.

Keyboard Inputs.

Each time that the processor is ready to read the keyboard input, it branches to a program routine required to accept the keyboard input. It can either wait until the operator depresses a key or, if the keyboard input is not ready, it can proceed with the program and return later to read the keyboard.

An encoder built into the keyboard converts each key output into a 6-bit character code corresponding to the physical location of the key. Three control signals (alpha, numeric, and control) accompany the keycode. In the I/O controller, the control signals are converted to a 2-bit code to indicate whether the keyboard character is the control key and whether the keyboard is in the "lower case" or "upper case" mode. The 2-bit control code is then combined with the 6-bit character code to form a byte in the accumulator in the controller. At this point, the processor can examine and act upon the keyboard input.

Operation of the Cartridge Tape Units.

Except for the 1501-CL model, two cartridge tape drives are mounted on the top of the terminal. A cartridge holds 100 feet of narrow (0.15 inch wide) magnetic tape that records one channel of digital data in serial form. The processor controls these tape units by issuing I/O instructions to the I/O controller.

Circuits required to operate two tape drives are housed on the I/O controller board. Optional drives are available in a Model 1533 Dual Drive. A tape multiplex board in the Model 1533 contains circuits that are duplicates of some of the tape control circuits on the I/O controller board. Up to three Model 1533 Dual Drives can be used with each 1501.

Tape operation is initiated by I/O instructions sent from the processor to the I/O controller. These instructions first select the drive to be used, and then start the tape motion required. Read or write instructions then establish the mode.

When reading, serial data from the tape is moved into the I/O controller and converted to parallel form. The controller then notifies the processor when a byte is available, and the processor accepts the byte by issuing another I/O instruction. When writing, the processor transfers the byte to be recorded to the I/O controller. In turn, the controller converts the data to serial form, encodes it in the form required for recording, and shifts it serially to the tape unit.

Operation of the Communications Adapters.

Available for installation in the communication slot of the terminal are communication adapter boards that allow the terminal to communicate with other terminals or similar equipment over long distances via telephone or teletype lines. The circuits on these boards are basically serial-to-parallel and parallel-to-serial converters and timing synchronizers. Connected to each communications board is an external MODEM that transmits and receives over the long distance lines.

The processor initiates the communication by sending an I/O instruction to the I/O controller that selects the communication channel for operation. The I/O controller then activates the adapter board. To coordinate the operation, several timing and control signals are exchanged directly between the communication adapter and the processor; however, the actual data transfer takes place between the I/O controller and the adapter boards.

Serial Input-Output Channel.

All external peripheral equipment operates on the serial I/O channel. The channel is a single coaxial cable terminated at both ends. Up to 64 units can communicate with one another over this link, with the Intelligent Terminal acting as master and the peripheral units, such as printers and tape units, acting as slaves.

The processor initiates the operation by sending an I/O instruction to the I/O controller. In turn, the controller begins transmission on the line, addresses the selected unit (each unit has an address selection switch or plug), and sends commands as required to execute the instruction from the processor. The selected peripheral unit responds by returning its address, to verify that the link is established, and the control signals necessary to accomplish the exchange between it and the Intelligent Terminal.

It should be noted that all information sent over the serial I/O channel is in serial, phase-encoded form. The I/O controller accomplishes the synchronization required to establish the link. It also converts data from parallel to serial for transmission and serial to parallel for reception.

2-2. PROCESSOR.

The processor is a single board that is the second board from the bottom of the chassis. It is a small, stored program, general purpose computer with a repertoire of 41 instructions. This number does not include the input-output instructions, which are executed by the I/O controller.

A block diagram of the processor is shown in figure 2-2. Those familiar with the organization of computers will recognize most of the basic elements, which are:

a. An operation register and decoder. These circuits hold and decode each instruction read from memory.

b. A memory data register, which holds data read from memory.

c. A B-register, which holds one of the operands involved in many of the operations.

d. An adder, which performs the actual arithmetic or logical operation.

e. A memory address register, which holds the address of a memory location to be used.

Other elements not shown on the block diagram but which normally are used in general purpose computers are:

a. Index registers, which hold a value used to modify a memory address. This processor has index registers but they are memory locations rather than hardware registers. Seven memory locations in each memory section are reserved to be used as index registers.

b. An instruction address register, which holds the address of the memory location in which the next instruction to be executed can be found. Again, memory locations substitute for hardware registers. Thirty-two memory locations are reserved to hold instruction addresses. A complex scheme in which a counter, called the stack pointer, designates the memory location to be read to acquire the address of the next instruction to be executed is used in this processor. This scheme is explained in detail later in this description.

c. An accumulator, which normally holds one of the operands involved in an arithmetic operation and also holds the result when the operation is completed. This processor has an accumulator that performs the aforementioned functions; however, it also serves as a primary input-output register for the processor, and for this reason it is located on the I/O controller board in order to simplify the I/O wiring.

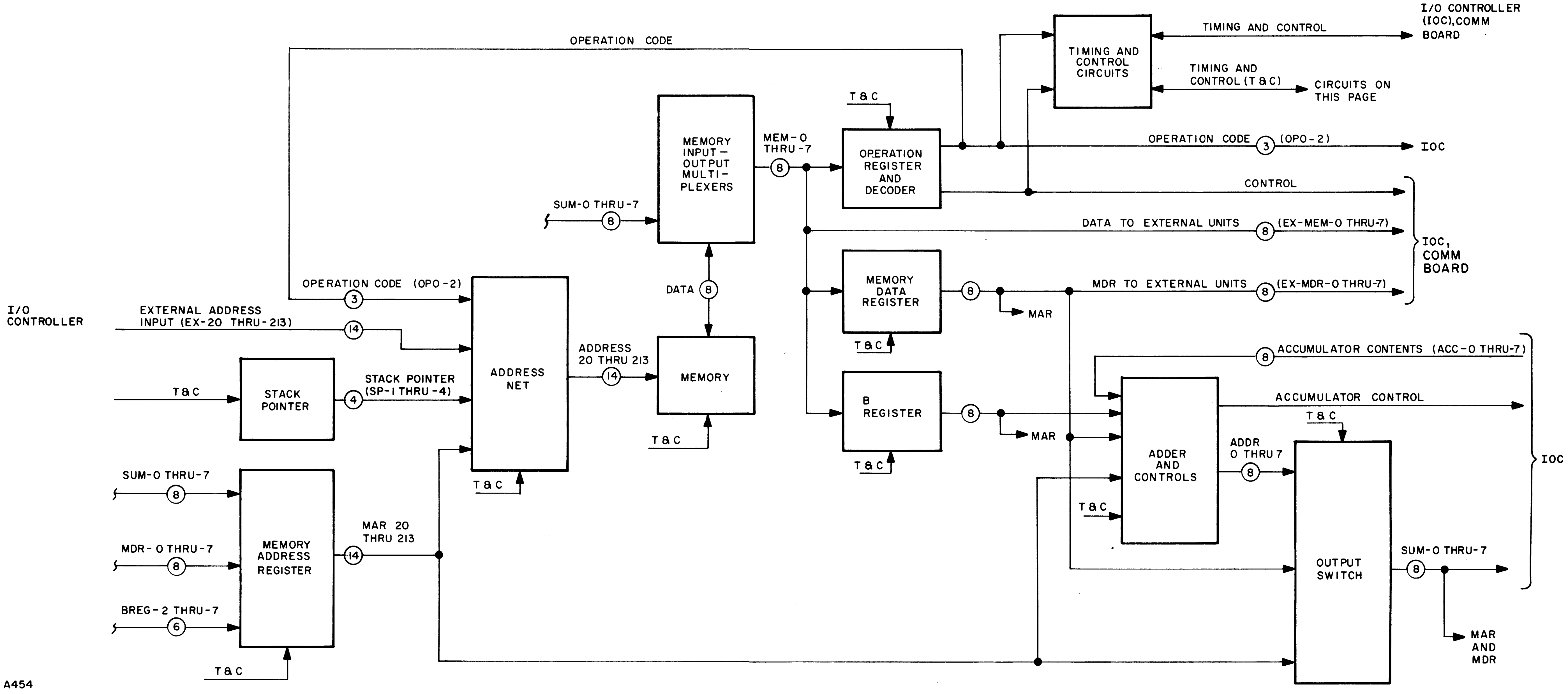


Figure 2-2. Processor Block Diagram

Word Formats.

The processor is organized to handle both instructions and data in 8-bit bytes. The bit positions in each byte are labeled 0 through 7, in order by the power of two that the position represents.

A typical instruction format is shown in figure 2-3. Note that each instruction is made up of two parts: (1) the IWR, instruction word-right, and (2) the IWL, instruction word-left. Memory addresses are also made up of two bytes, left and right, while data is handled as a single byte.

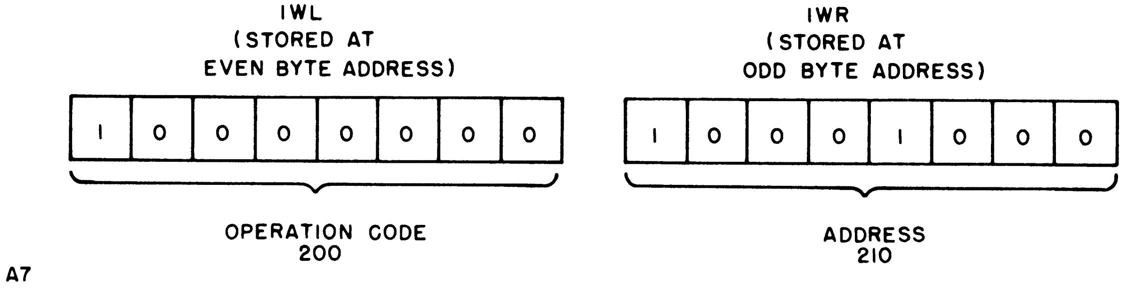


Figure 2-3. Typical Instruction Format

Memory Addresses.

Because the operation of the processor is very closely related to the memory, a brief description of the memory addressing scheme is presented here before operation of the processor circuits is discussed. The memory in the Model 1501 is solid-state with a maximum capacity of 16,384 bytes, and this memory is organized into groups of 256 bytes each; each group is called a "page". Shown in figure 2-4 are the 77 (octal) pages that make up a 16,384-byte memory.

An address of 14 bits is required to address a memory of 16,384 locations. Since the processor is organized to handle 8-bit bytes, this requires two bytes, as shown in figure 2-5. Note that the section bits and level bits together are called the page number; they choose a 256-byte page. The eight low-order bits are called "byte" number; they select one of the 256 bytes in the page. For example, the address format to select byte 320 (octal) of page 16 (called P16-320) would appear as shown in figure 2-6. This memory location can hold half of another memory address, half of an instruction, or a data byte.

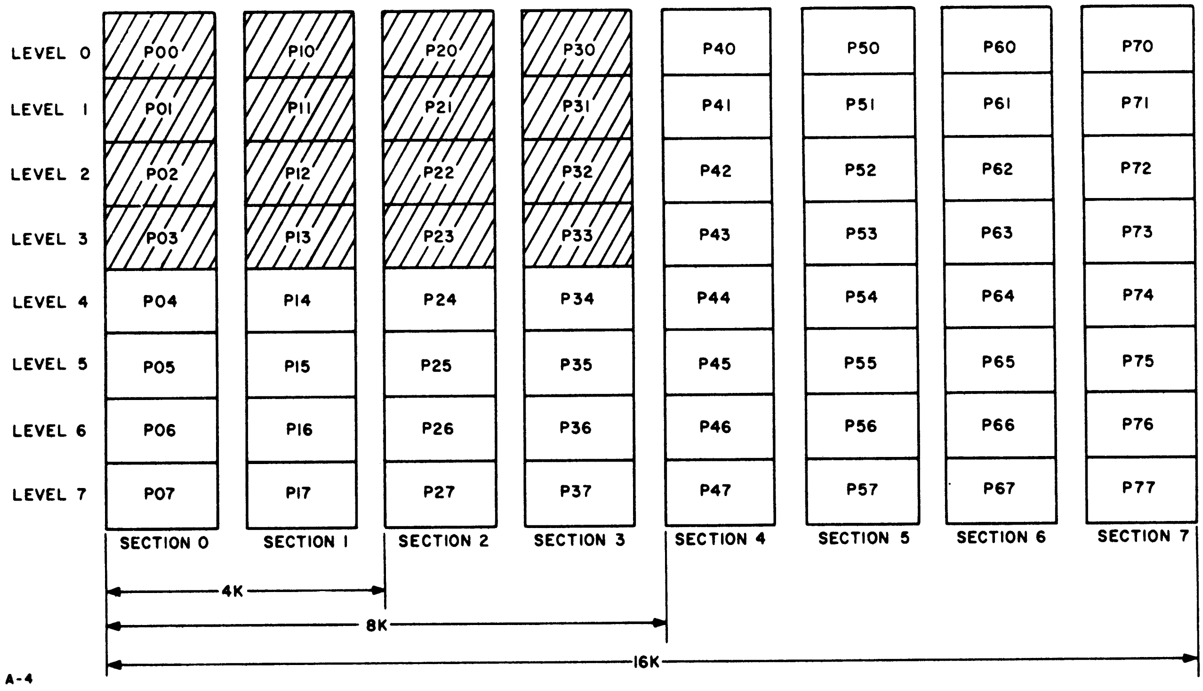


Figure 2-4. Organization of Memory

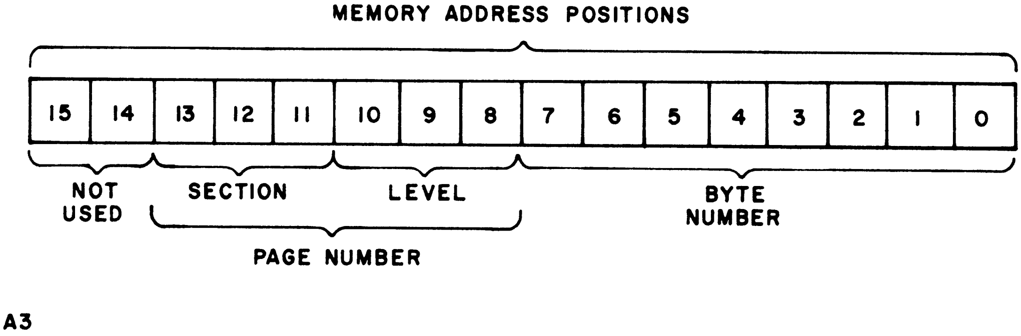


Figure 2-5. Memory Address Format

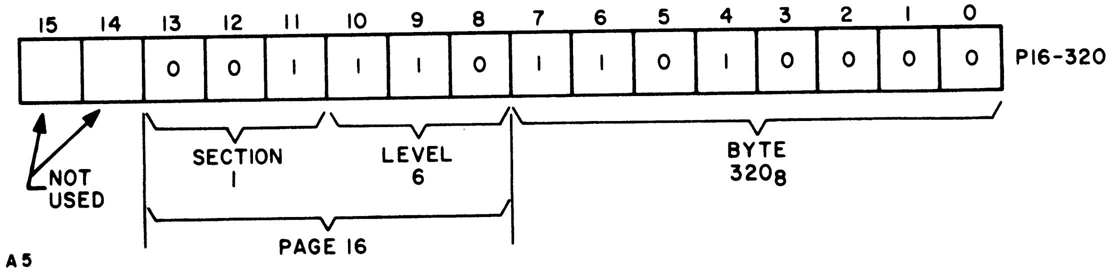


Figure 2-6. Sample Address

Instruction Addressing.

Operation of the processor is controlled by a program made up of instructions. This program is stored in memory, from which it is read and executed one instruction at a time. The address of the instruction currently being executed is held in two memory locations, called stack pointer addresses. There are 32 of these locations, which means that the addresses of 16 different instructions can be held in the "stack".

The contents of two stack pointer addresses are shown in figure 2-7. Note that the first location holds the byte number and that the second location holds the page number of the memory address in which an instruction is stored. In order to execute that instruction, the processor must read the contents of a pair of stack pointer addresses. (The setting of the stack pointer determines which pair of addresses is used.) Once the processor has read the instruction address from the memory location specified by the stack pointer, it then uses this address to read the instruction itself from memory.

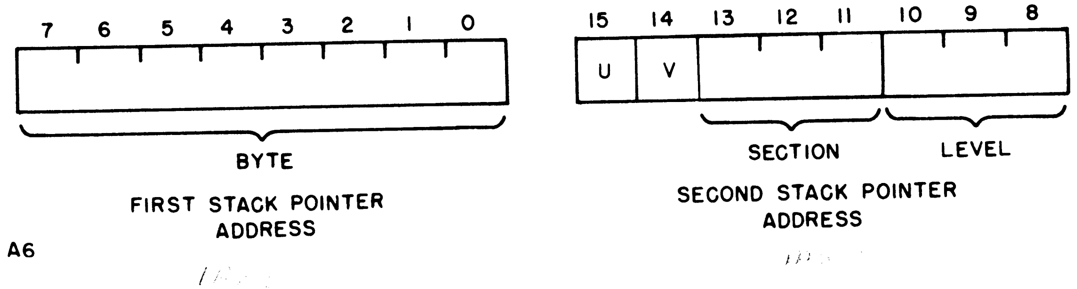


Figure 2-7. Contents of Stack Pointer Addresses

Instructions and their corresponding addresses are written in the following format:

Address of Instruction	Instruction
P03-100	200-377
P03-102	260-001
P03-104	230-007

The first instruction above (200-377), load 377 into the accumulator, is located at octal location 100 of page 3. P03-100 is the address of the IWL, the 200 byte. The 377 byte (the IWR) is located at the address P03-101. These instructions would be stored in page 3 of memory as shown in figure 2-8.

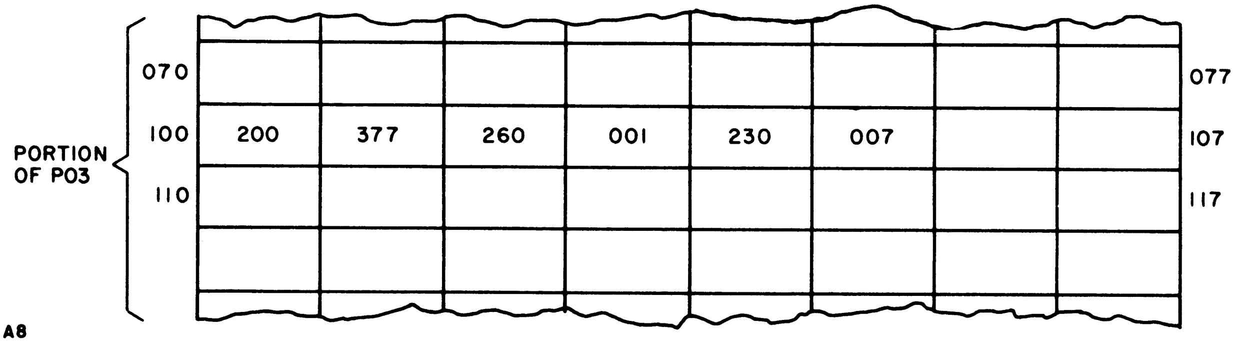


Figure 2-8. Typical Location of Instructions in Memory

Use of the Stack Pointer.

As shown in the preceding discussion, the address of the current instruction consists of two bytes. It is called the instruction address word left and right (IAWL and IAWR). The IAW in an active program is held in two specific memory locations, initially P00-040 and P00-041. These two locations are the first two stack pointer addresses.

When instructions stored in sequential addresses are executed, the IAW is read from P00-040 and 041, incremented by two during execution of the instruction, and returned to P00-040 and 041. However, there are many cases in which an instruction transfers to a non-sequential memory location to acquire the next instruction. Non-sequential execution is encountered in programs that contain "Test" and "Branch" instructions. These instructions cause the program to jump to a non-sequential address if a certain condition is met.

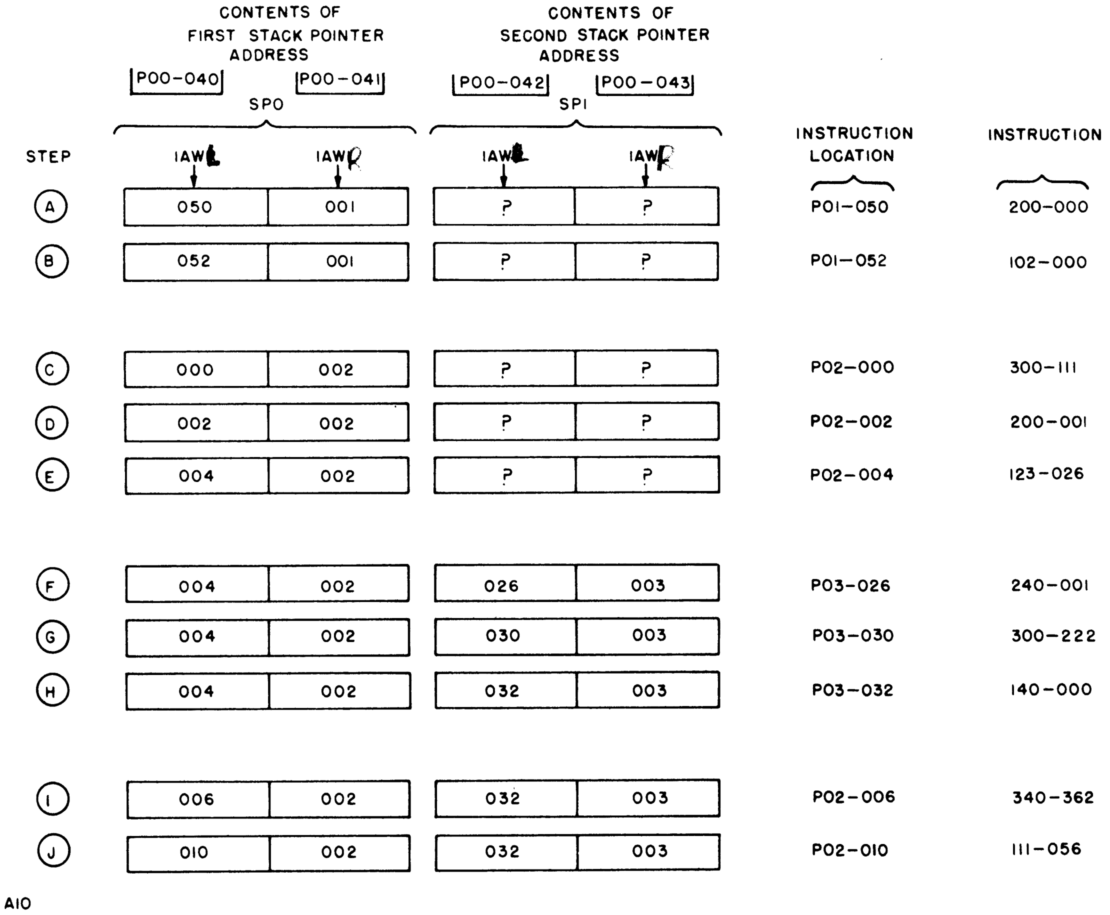
Figure 2-9 illustrates the principles discussed above. Steps A and B illustrate the execution of instructions stored in sequential addresses. Step A is instruction 200-000 (Load Accumulator with 000), located at address P01-050. This instruction address is being held in the first stack pointer address P00-040 and P00-041. (The IAWR is in P00-040; the IAWL is in P00-041.)

While the Load Accumulator instruction is being executed, the IAW~~R~~L (050) is incremented by two and inserted back into P00-040. Now the address of the current instruction is P01-052. This is shown in Step B.

The instruction stored in address P01-052 is a branch instruction. When branch instructions are executed, they put the new branch address into both stack pointer addresses (P00-40 and P00-41), as shown in Step C. Note that branch instruction 102-000 produces address P02-000 in the stack pointer address locations. This means that the instruction held in P02-000 is executed next.

Steps D and E again illustrate sequential instruction addressing; however step E is a stack-and-branch instruction. In addition to putting a new IAW into the stack pointer addresses, this instruction steps the stack pointer counter to cause the counter to "point" to the second pair of stack pointer addresses. This changes the location of the stack pointer addresses to P00-042 and 043. The old stack pointer addresses (SP0 in the figure) are not changed and they are left containing the address of the stack and branch instruction executed in step E. The purpose of storing the address of the instruction at which the branch took place is to allow the program to return to this point later. The new, and now the current, stack pointer addresses (SP1 in the figure) contain the new "branch to" address.

Steps F, G, and H illustrate sequential execution using the new stack pointer addresses (SP1) to hold the instruction address words. Step H is an exit instruction. This instruction does not contain any "branch to" address. It simply steps the stack pointer counter down one, returning to stack pointer addresses P00-040 and 041 to get instruction address words. The IAW retrieved is 004-002. This is incremented by two to obtain instruction address P02-006, in step I, and thus the program has



	CONTENTS OF FIRST STACK POINTER ADDRESS		CONTENTS OF SECOND STACK POINTER ADDRESS			
	P00-040	P00-041	P00-042	P00-043		
	SP0		SP1			
STEP	IAWL	IAWR	IAWL	IAWR	INSTRUCTION LOCATION	INSTRUCTION
A	050	001	?	?	P01-050	200-000
B	052	001	?	?	P01-052	102-000
C	000	002	?	?	P02-000	300-111
D	002	002	?	?	P02-002	200-001
E	004	002	?	?	P02-004	123-026
F	004	002	026	003	P03-026	240-001
G	004	002	030	003	P03-030	300-222
H	004	002	032	003	P03-032	140-000
I	006	002	032	003	P02-006	340-362
J	010	002	032	003	P02-010	111-056

Figure 2-9. Use of the Stack Pointer

returned to the point at which the branch took place (the next instruction address after the instruction in step E). The stack pointer addresses continue to be P00-040 and 041 unless another stack-and-branch instruction is executed, and the contents of the SP1 locations do not change.

As previously mentioned, the purpose of the stack-and-branch, then exit, instructions is to preserve the "come from" address and allow the program to return to this address by executing the exit instruction. This allows the sharing of subroutines by different areas of the main program. A maximum of 16 consecutive stack and branch operations are possible before returning to the original starting point. The IAW's are stored in stack pointer addresses P00-040 through P00-077.

Data Addressing.

Data bytes can be addressed by an instruction in any one of three different addressing modes: immediate, direct, or indexed. In the immediate addressing mode, the IWR is used as the data byte, and no further memory access is required by the current instruction. Immediate addressing provides a means of entering a constant into the accumulator by specifying the constant in the IWR of the instruction in which it is used. For example, instruction 200-311 causes an octal 311 to be loaded into the accumulator.

In the direct addressing mode, the IWR is used as the address of the data byte. Since only the eight bits of the IWR are available for an address, only 256 (377 octal) selections are possible. Essentially, a byte within a page can be selected in the direct addressing mode but the page itself cannot be chosen.

In this mode, the page number is established by two conditions. First, the section of memory (0 through 7, the most significant digit of the page number) can be selected by a Set Memory Section instruction previously executed. Second, the level of memory (0 through 7, the least significant digit of the page number) is not supplied. Therefore, it is effectively zero. Thus, instructions executed in the direct addressing mode can select addresses in pages 00, 10, 20, 30, 40, 50, 60, and 70, with the most significant digit of the page number having been previously established. For example, instruction 210-240 causes the contents of P00-240 to be loaded into the accumulator if memory section 0 is selected.

The indexed addressing mode provides a method of addressing data stored anywhere within memory. An indexed address is composed of a byte address contained in a specified index register in memory plus a page address contained in the IWR. The high-order six bits of the IWR (2-7) specify the page within memory and the index register specifies the location within that page. Bits 0-2 of the IWL select the index register to be used.

An example of an instruction executed in the indexed address mode is 213-050, the format of which is shown in figure 2-10. This instruction causes the contents of the location in P12 specified by the contents index register 3 to be loaded into the accumulator. If index register 3 contained 145 (octal), for example, the contents of P12-145 would be loaded into the accumulator. The two low-order bits of the IWR determine what changes the instruction makes in the contents of the selected index register, as follows:

00 - Leave unchanged

10 - Increment by 1

11 - Decrement by 1

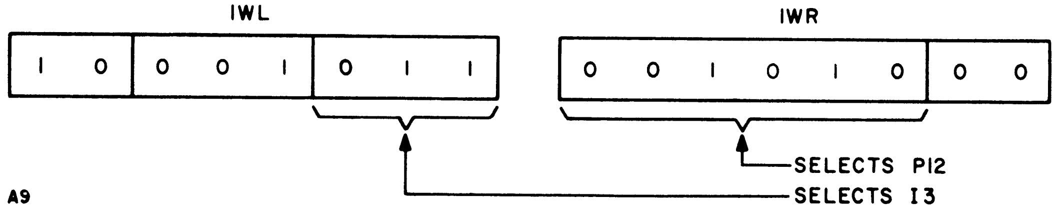


Figure 2-10. Format of an Instruction Using the Indexed Address Mode

Processor Timing.

The basic timing unit for the processor is the machine cycle, which is one microsecond long. Figure 2-11 is a timing diagram in which the signals generated during each machine cycle are shown. A 6 MHz oscillator is the source of the timing signals. A counter and count decoding gates then use the oscillator output to produce the rest of the timing signals shown in the figure.

Instruction Cycles.

The next timing division in the processor is the instruction cycle, which is the time required to acquire an instruction and execute it. Depending upon the type of instruction, an instruction cycle is made up of three to six machine cycles. Thus, an instruction may require between three and six microseconds.

As stated above, a maximum of six machine cycles are required to perform an instruction. These cycles are designated I (instruction) or E (execution) as follows: I1, I2, I3, I4, E2, and E3.

Instructions can be divided into four general classes:

Class 0, Jump Instructions

Class 1, Branch and I/O Instructions

Class 2, Transfer and Arithmetic Instructions

Class 3, Boolean and Compare Instructions

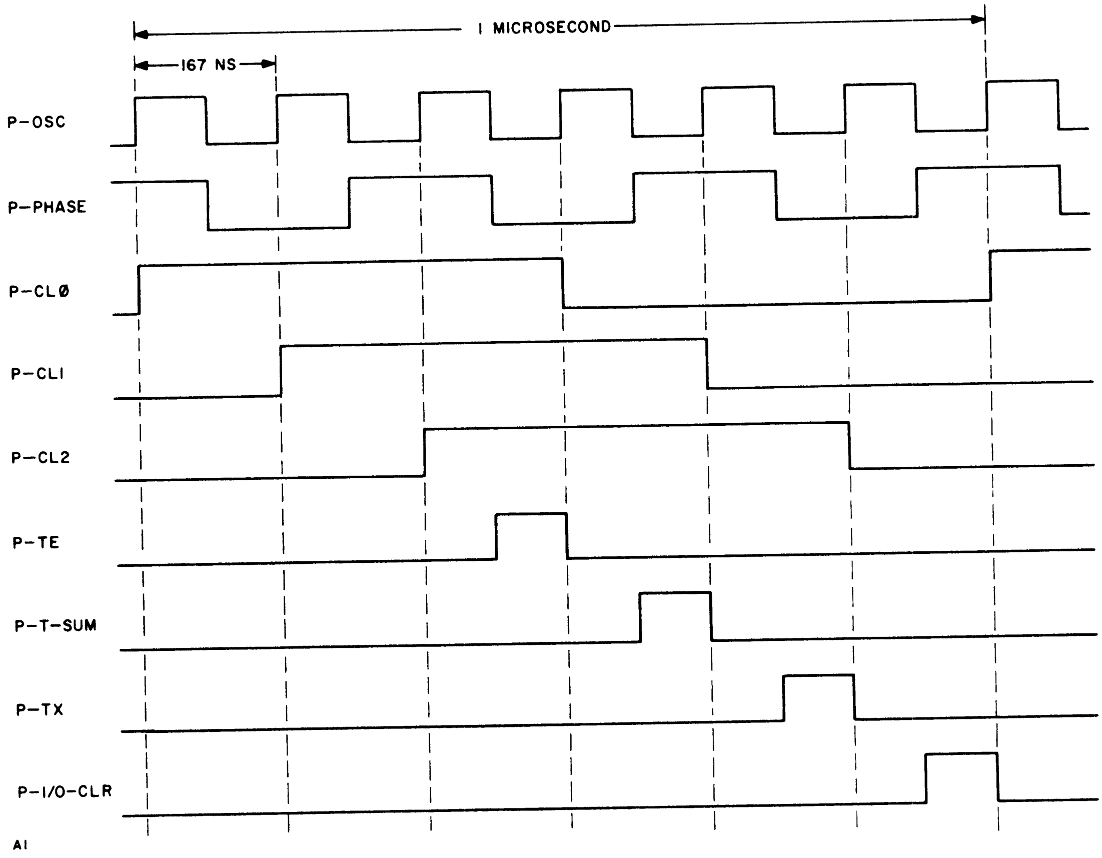


Figure 2-11. Machine Cycle Timing Diagram

The class to which an instruction belongs is indicated by the two most significant bits of the left byte of the instruction word, which are 0, 1, 2, or 3, corresponding to the class of instructions. Instruction cycles I1 through I4 are common to all instruction classes, while execution cycles E2 and E3 operate only during Class 2 and Class 3 instructions. Table 2-1 shows the cycles required for each class of instruction.

The number of machine cycles in an instruction depends upon the instruction and the conditions that exist when the instruction is executed. Since the IWL, which contains the operation code, is read during cycle I3, the following discussion defines the starting cycle as I3. Figure 2-12 shows the functions performed during each cycle.

At the beginning of cycle I3, the address of the instruction to be executed has been formed in the memory address register and is applied to the address net. The least significant position of the address net is controlled directly by the I3 and I4 signals. During I3, it is held at zero to read an even address and during I4 it is switched to a one to read the odd address.

Table 2-1. Cycles Required by Each Instruction

IWL	TYPE	REQUIRED CYCLES
0XX 10X 11X 12X 13X 14X 15X 16X 17X	Neither Direct, Indexed, Nor Immediate Address	I_1, I_2, I_3, I_4
20X 22X 24X 26X 30X 32X 34X 36X	Immediate Address	I_1, I_2, I_3, I_4, E_3
21X 23X 25X 27X 31X 33X 35X 37X	Direct or Indexed Address	I_1, I_2, I_3, I_4, E_2, E_3

During the first half of I3, the IWL is read. This byte is then loaded into the operation register, the B register, and the memory data register. At this point, the operation code is decoded and instruction execution can begin.

During the second half of I3 the I/O controller gains access to memory. The I/O controller reads either character codes or dot patterns for the CRT display.

I1	1ST HALF	1. Using the stack pointer as an address, read the IAWL. 2. Load the IAWL in the MDR
	2ND HALF	1. Add 2 to IAWL (if no branch) 2. Restore the IAWL 3. Load MAR with IAWL or branch address
I2	1ST HALF	1. Using the stack pointer as an address, read the IAWR. 2. Load the IAWR in the MDR
	2ND HALF	1. Add carry to IAWR (if carried from IAWL) 2. Restore the IAWR 3. Load MAR with IAWR or branch address
I3	1ST HALF	1. Using the MAR as an address, read the IWL 2. Load the IWL into the operation register, B register, and MDR.
	2ND HALF	1. Using the I/O controller address, read the CRT character or dot pattern code. 2. Transfer character code or dot pattern to I/O controller.

A489 (Sheet 1)

Figure 2-12. Functions Performed During Each Instruction Cycle (Sheet 1 of 2)

I4	1ST HALF	1. Using the MAR as an address, read the IWR 2. Load the IWR in the MDR and/or B register
	2ND HALF	1. Using the I/O controller address, read the character code or dot pattern for the CRT character position. 2. Transfer the character code or dot pattern to the I/O controller.
E2	1ST HALF	1. If the indexed address mode is selected, use the index register number and section number as an address, read the contents of an index register. Load the contents of the index register into the MDR. 2. If the direct address mode is selected, no memory access is required.
	2ND HALF	1. If the indexed mode is selected, modify the contents of the index register and restore the contents of the index register. Load MAR with composite address from the B register and the MDR. 2. If the direct address mode is selected, load the MAR with the IWR without another memory access cycle.
E3	1ST HALF	1. If necessary, read the contents of the location specified by MAR. 2. Load the contents into MDR.
	2ND HALF	1. Execute instruction 2. If necessary, store the results in the location specified by MAR.

A489 (Sheet 2)

Figure 2-12. Functions Performed During Each Instruction Cycle (Sheet 2 of 2)

At the beginning of cycle I4, the processor again uses the contents of the memory address register to select a memory location and read the contents. As previously mentioned, the I4 signal applied to the address net switches the low-order position to a one; thus, the IWR is read from an odd address. The IWR is loaded into the MDR, replacing the previously loaded IWL. Thus, at the end of cycle I4, the IWL is in the operation register while the IWR is in the memory data register.

During the second half of cycle I4, the I/O controller again addresses memory. Again, it reads either the character code or the dot pattern required to produce the characters on the CRT display.

The next step taken depends upon the class of instruction. Class 0 and Class 1 instructions require only cycles I3, I4, I1, and I2 to perform their function, whereas Class 2 and Class 3 instructions also require cycles E2 and E3.

Assume that the current instruction is a Class 2 or 3 instruction. Once this instruction is decoded, the processor determines from its operation code whether it is to be executed in the immediate, direct, or indexed addressing mode. Operation in the indexed mode is discussed first.

When the indexed address mode is selected, the index register number specified in the IWL and the section number held in the section register are gated through the address net to select an index register and read its contents during cycle E2. The data read is loaded into the memory data register. During the second half of cycle E2, the contents of the MDR are modified as specified by the two index register control bits. The modified number is then written back into the same index register.

At the end of cycle E2, the contents of the MDR (contents of the index register) are transferred to the low-order positions of the MAR, and bits 2-7 of the B register (the page number portion of an address) are loaded in the high-order positions of the MAR. Together, these portions form a composite address of the memory location to be used by this instruction.

If the direct address mode is selected, the memory access cycle during the first half of cycle E2 is not required. The MDR still holds the IWR and, during the second half of E2, the IWR is transferred to the MAR.

For class 2 or 3 instructions operating in the immediate address mode, cycle E2 is skipped. Cycle E3 begins immediately after cycle I4.

If a memory access cycle is required during cycle E3 of the Class 2 and 3 instructions, the MAR contents are used to address memory as required to execute the instruction. Note that some of the Class 2 and 3 instructions do not require further memory access for their execution.

After an instruction (regardless of class) is executed, cycle I1 must be performed to acquire the next instruction. During cycle I1, the contents of the stack pointer are gated through the address net. This reads the IAWL from memory.

The action taken during the second half of I1 depends upon whether or not the instruction is to execute a branch. Assuming that no branch is to be taken, the IAWL is increased by two, to specify the address of the current instruction, and returned to the stack pointer address from which it was read. The modified IAWL is then loaded into the low-order positions of the MAR to become part of the address from which the next instruction will be read.

Still assuming that no branch is to be taken, the processor must determine if cycle I2 is necessary. There are four different conditions that cause cycle I2 to be performed. These conditions are:

a. The take branch flip-flop is set. This flip-flop records the results of branch instructions executed during I3 and I4, and it determines whether or not a branch is to be taken.

b. The instruction is in the 15X class.

c. The current instruction used either the direct or indexed address mode. This means that the page number held in the high-order section of the MAR was changed and, therefore, it is necessary to read the IAWR to reacquire the page number.

d. A carry resulted when two was added to the IAWL in cycle I1. If a carry resulted, the IAWR has been stepped into the next page and, therefore, the IAWR must be read and modified.

If cycle I2 is not required, the processor moves to cycle I3 and reads the IWL from the location specified by the contents of the MAR. This begins a new instruction.

If, however, cycle I2 is performed, the following action takes place: The IAWR is read from the odd stack pointer address. If a carry must be added, this is done and the updated IAWR is restored to the location from which it was read. The IAWR is also loaded into the high-order positions of the MAR and used to read the next instruction.

The next case to be considered is what must be done if the current instruction produces a branch condition. In this case, both cycles I1 and I2 must be performed, but the action taken during these cycles depends upon which instruction is being executed. (Detailed descriptions of each instruction follow.) Basically, an instruction address is formed to substitute for the IAWL and IAWR. This new address is loaded into the MAR and the next instruction read from the new address during I3 and I4.

Processor Instruction Set.

As mentioned earlier in this section, the processor is a stored program, general-purpose computer. The program that it executes is made up of instructions stored in the memory. An instruction is read from memory, decoded, and executed; then, the next instruction is read from memory and the process repeated.

Although a program may be made up of several thousand instructions, it is simply a combination of the 41 basic instructions that the processor can execute directly and the I/O instructions that are carried out by the I/O controller. The 41 basic instructions are described below, and their formats are shown in figure 2-13.

TLJ - Test Literal and Jump

The accumulator is compared to the literal, and the result is indicated in the condition register. If they are equal, a jump forward (+) or a jump backward (-) up to 15 instruction locations is performed. If, however, they are not equal, the next sequential instruction is executed. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

TMJ - Test Mask and Jump

The IWR is used as a mask. When any of the accumulator bits selected by a one in the mask are zero, the condition register is set to unequal. When the selected bits are all ones, the condition register is set to equal. When a mask bit is zero, the corresponding accumulator bit is ignored. If the condition register is set to equal, a jump forward (+) or jump backward (-) up to 15 instruction locations is performed. If the condition register is not set to equal, the next sequential instruction is executed. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

TLX - Test Literal and Exit

The accumulator is compared to the literal, and the result is indicated in the condition register. If they are not equal, the next sequential instruction is performed. If they are equal, a conditional exit is performed to complete the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding level, which contains the address of the last stack and branch instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is the instruction address where processing continues. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

INSTRUCTION	FORMAT																OUTPUTS OF OPERATION CODE DECODER
	IWL								IWR								
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
TLJ - Test Literal and Jump (0XX)	0	0	0	JUMP COUNT				*	LITERAL								BR-OFFSET-OP
*0-Jump Forward, 1-Jump Backward																	
TMJ - Test Mask and Jump (0XX)	0	0	1	JUMP COUNT				*	MASK								BR-OFFSET-OP
TLX - Test Literal and Exit (000)	0	0	0	0	0	0	0	0	LITERAL								BR-OFFSET-OP
TMX - Test Mask and Exit (040)	0	0	1	0	0	0	0	0	MASK								BR-OFFSET-OP
BRU - Branch Unconditional (10X)	0	1	0	0	0	PAGE			ADDRESS							0	BCD-UNC
BRE - Branch on Equal (10X)	0	1	0	0	0	PAGE			ADDRESS							1	BCD-UNC
BRH - Branch on High (11X)	0	1	0	0	1	PAGE			ADDRESS							0	BCD
BRL - Branch on Low (11X)	0	1	0	0	1	PAGE			ADDRESS							1	BCD
SBU - Stack and Branch Unconditional (12X)	0	1	0	1	0	PAGE			ADDRESS							0	STK-UNC
SBE - Stack and Branch on Equal (12X)	0	1	0	1	0	PAGE			ADDRESS							1	STK-UNC
SBH - Stack and Branch on High (13X)	0	1	0	1	1	PAGE			ADDRESS							0	STK
SBL - Stack and Branch on Low (13X)	0	1	0	1	1	PAGE			ADDRESS							1	STK
EXB - Exit and Branch (16X)	0	1	1	1	0	PAGE			ADDRESS							0	EXIT-BRANCH

A490 (Sheet 1)

Figure 2-13. Processor Instruction Set (Sheet 1 of 5)

INSTRUCTION	FORMAT																OUTPUTS OF OPERATION CODE DECODER
	IWL								IWR								
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
EXU - Exit Unconditional (140)	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	EXIT
SMS - Set Memory Section (150)	0	1	1	0	1	0	0	0	0	0	SECTION			0	0	0	15-X-OP, SECTION-OP
SMC - Set Memory Control (151)	0	1	1	0	1	0	0	1	Y	Y	0	0	0	0	0	0	15-X-OP, 151-OP
SSC - Set Memory Section and Control (152) YY = 0-Reset U&V, 1-Set V & Reset U, 2-Set U & Reset V, 3-Set U&V	0	1	1	0	1	0	1	0	Y	Y	SECTION			0	0	0	15-X-OP, 152-OP
SAC - Set Arithmetic Condition (153)	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	15-X-OP, 153-OP
LSW - Load Sense Switches (154) (or) EMP - Enable Memory Parity	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	15-X-OP, SENSE-1
LPS - Load Processor Status (155)	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	15-X-OP, 155-OP
DPI - Disable Processor Interrupt (156)	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	15-X-OP, 156-OP
EPI - Enable Processor Interrupt (156)	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	15-X-OP, 156-OP
CPI - Clear Processor Interrupt (156)	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	15-X-OP, 156-OP
TRM - Load Terminal Identification (157)	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	15-X-OP, 157-OP
LDA - Load Accumulator (200) (Immediate Address)	1	0	0	0	0	0	0	0	LITERAL								LOAD-OP, DIRECT
LDA - Load Accumulator (210) (Direct Address)	1	0	0	0	1	0	0	0	ADDRESS								LOAD-OP, INDIRECT

A646 (Sheet 2)

Figure 2-13. Processor Instruction Set (Sheet 2 of 5)

INSTRUCTION	FORMAT																OUTPUTS OF OPERATION CODE DECODER
	IWL								IWR								
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
LDA - Load Accumulator (21X) (Indexed Address)	1	0	0	0	1	IR			PAGE ADDRESS						Y	Y	LOAD-OP, INDIRECT
YY - Index Register Control																	
LDX - Load Index Register (20X)	1	0	0	0	0	IR			LITERAL								LOAD-OP, DIRECT
LIA - Load Instruction Address (22X)	1	0	0	1	0	IR			LITERAL								STORE-OP, DIRECT
STA - Store Accumulator (230) (Direct Address)	1	0	0	1	1	0	0	0	ADDRESS								STORE-OP, INDIRECT
STA - Store Accumulator (23X) (Indexed Address)	1	0	0	1	1	IR			PAGE ADDRESS						Y	Y	STORE-OP, INDIRECT
ADA - Add to Accumulator (240) (Immediate Address)	1	0	1	0	0	0	0	0	LITERAL								DIRECT
ADA - Add to Accumulator (250) (Direct Address)	1	0	1	0	1	0	0	0	ADDRESS								INDIRECT
ADA - Add to Accumulator (25X) (Indexed Address)	1	0	1	0	1	IR			PAGE ADDRESS						Y	Y	INDIRECT
ADX - Add to Index Register (24X)	1	0	1	0	0	IR			LITERAL								DIRECT
SUA - Subtract from Accumulator (260) (Immediate Address)	1	0	1	1	0	0	0	0	LITERAL								SUB-OP, DIRECT

A490 (Sheet 3)

Figure 2-13. Processor Instruction Set (Sheet 3 of 5)

INSTRUCTION	FORMAT																OUTPUTS OF OPERATION CODE DECODER
	IWL								IWR								
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
SUA - Subtract from Accumulator (270). . . . . . (Direct Address)	1	0	1	1	1	0	0	0	ADDRESS								. . SUB-OP, INDIRECT
SUA - Subtract from Accumulator (27X). . . . . . (Indexed Address)	1	0	1	1	1	IR			PAGE ADDRESS						Y	Y	. . SUB-OP, INDIRECT
SUX - Subtract from Index Register (26X)	1	0	1	1	0	IR			LITERAL								. . SUB-OP, DIRECT
ANA - Logical AND to Accumulator (300) . . . . . (Immediate Address)	1	1	0	0	0	0	0	0	LITERAL								. . AND-OP, DIRECT
ANA - Logical AND to Accumulator (310) . . . . . (Direct Address)	1	1	0	0	1	0	0	0	ADDRESS								. . AND-OP, INDIRECT
ANA - Logical AND to Accumulator (31X) . . . . . (Indexed Address)	1	1	0	0	1	IR			PAGE ADDRESS						Y	Y	. . AND-OP, INDIRECT
SAN - Shift and Logical AND to Accumulator (30X)	1	1	0	0	0	SHIFT COUNT			LITERAL								. . AND-OP, DIRECT
ERA - Exclusive OR to Accumulator (320). . . . . (Immediate Address)	1	1	0	1	0	0	0	0	LITERAL								. . EX-OR-OP, DIRECT
ERA - Exclusive OR to Accumulator (330). . . . . (Direct Address)	1	1	0	1	1	0	0	0	ADDRESS								. . EX-OR-OP, INDIRECT
ERA - Exclusive OR to Accumulator (33X). . . . . (Indexed Address)	1	1	0	1	1	IR			PAGE ADDRESS						Y	Y	. . EX-OR-OP, INDIRECT

A490 (Sheet 4)

Figure 2-13. Processor Instruction Set (Sheet 4 of 5)

	FORMAT																
	IWL								IWR								OUTPUTS OF
																	OPERATION CODE
INSTRUCTION	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	DECODER
SER - Shift and Exclusive OR to Accumulator (32X)	1	1	0	1	0	SHIFT COUNT			LITERAL								EX-OR-OP, DIRECT
IRA - Inclusive OR to Accumulator (360) (Immediate Address)	1	1	1	1	0	0	0	0	LITERAL								OR-OP, DIRECT
IRA - Inclusive OR to Accumulator (370) (Direct Address)	1	1	1	1	1	0	0	0	ADDRESS								OR-OP, INDIRECT
IRA - Inclusive OR to Accumulator (37X) (Indexed Address)	1	1	1	1	1	IR			PAGE ADDRESS						Y	Y	OR-OP, INDIRECT
SIR - Shift and Inclusive OR to Accumulator (36X)	1	1	1	1	0	SHIFT COUNT			LITERAL								OR-OP, DIRECT
CPA - Compare Accumulator (340) (Immediate Address)	1	1	1	0	0	0	0	0	LITERAL								COMP-OP, DIRECT
CPA - Compare Accumulator (350) (Direct Address)	1	1	1	0	1	0	0	0	ADDRESS								COMP-OP, INDIRECT
CPA - Compare Accumulator (35X) (Indexed Address)	1	1	1	0	1	IR			PAGE ADDRESS						Y	Y	COMP-OP, INDIRECT
CPX - Compare Index Register (34X)	1	1	1	0	0	IR			LITERAL								COMP-OP, DIRECT

A490 (Sheet 5)

Figure 2-13. Processor Instruction Set (Sheet 5 of 5)

TMX - Test Mask and Exit

The IWR is used as an eight-bit mask. When any of the accumulator bits selected by a one in the mask are zero, the condition register is set to unequal. When the selected bits are all ones, the condition register is set to equal. When the mask bit is zero, the corresponding accumulator bit is ignored. If the condition register is set to equal, a conditional exit is performed to complete the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding level, which contains the address of the last stack and branch instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is instruction address where processing continues. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

BRU - Branch Unconditional

An unconditional branch is performed to the address specified. The branch address can be any location within the current section of memory. (Timing: 4 microseconds.)

BRE - Branch on Equal

A branch to the address specified is performed when the condition register, set by a previous instruction, is found to be set to equal. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

BRH - Branch on High

A branch to the address specified is performed when the condition register, which has been set by a previous instruction, is found to be set to high. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

BRL - Branch on Low

A branch is performed when the condition register, set by a previous instruction, is found to be set to low. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

SBU - Stack and Branch Unconditional

A stack and branch operation is performed regardless of the setting of the condition register. The current instruction address is stored and the stack pointer is incremented to the next level. (Timing: 3 microseconds.)

SBE - Stack and Branch on Equal

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to equal. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

SBH - Stack and Branch on High

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to high. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

SBL - Stack and Branch on Low

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to low. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

EXB - Exit and Branch

The exit and branch instruction branches to the address specified and decrements the stack pointer to the preceding level. The address specified in the operand is the instruction address within the current section where processing continues. (Timing: 4 microseconds.)

EXU - Exit Unconditional

This form of exit performs the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding stack level, which contains the address of the last stack and branch instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is the instruction address where processing continues. (Timing: 4 microseconds.)

SMS - Set Memory Section

The set memory section instruction provides a means of transferring from the current section of memory to another section. A branch instruction (branch, stack and branch, or exit and branch) preceded by an SMS instruction will transfer to the address defined by the branch address and the section specified in the Set Memory Section instruction operand. (Timing: 4 microseconds.)

SMC - Set Memory Control

This instruction controls the state of the U and V bits. A description of how the U and V bits affect memory addressing appears later in the processor description.

SSC - Set Memory Section and Control

This instruction performs both the set memory section operation of SMS and the set memory control operation of SMC. (Timing: 4 microseconds.)

SAC - Set Arithmetic Condition

This instruction forces the condition register to a condition that depends on the state of bits 4 and 5 of the accumulator, at the time of the SAC instruction, as follows:

ACCUMULATOR BITS		CONDITION FORCED
5	4	
1	0	Equal
0	1	High
0	0	Low
1	1	Equal

(Timing: 4 microseconds)

LSW - Load Sense Switches

This instruction loads the state of 8 toggle switches into the accumulator. This instruction is available only for those units with the sense switch option. (Timing: 4 microseconds.)

EMP - Enable Memory Parity

This instruction enables the parity checking circuits in the memory and clears any previous parity error. This instruction is used only with the 16,384-byte memory that has parity checking capability. (Timing: 4 microseconds.)

LPS - Load Processor Status

This instruction loads a processor status word into the accumulator. (Timing: 4 microseconds.)

DPI - Disable Processor Interrupt

This instruction inhibits the automatic stack and branch that occurs upon receipt of an interrupt. It does not disable the interrupt. If the Disable Processor Interrupt instruction is executed and the interrupt is subsequently activated, the automatic stack and branch operation will occur only after the execution of an EPI instruction. (Timing: 4 microseconds.)

EPI - Enable Processor Interrupt

This instruction enables the automatic stack and branch operation that occurs upon receipt of an interrupt. It must be executed once for each interrupt.

CPI - Clear Processor Interrupt

An interrupt overflow condition occurs when more than two interrupts have been activated before the execution of an Enable Processor Interrupt (EPI) instruction. This condition can be tested by the Load Processor Status (LPS) instruction. A Clear Interrupt instruction clears the interrupt overflow indicator. (Timing: 4 microseconds.)

TRM - Load Terminal Identification

Loads the accumulator with the identification assigned to this terminal by the program interlocking plug. (Timing: 4 microseconds.)

LDA - Load Accumulator

Loads the accumulator with the value specified by the immediate, the direct, or the indexed address. (Timing: 4 microseconds when immediate address form is used, otherwise, 6 microseconds.)

LDX - Load Index Register

Loads the specified index register with the IWR. (Timing: 4 microseconds.)

LIA - Load Instruction Address

Transfers the 8 least significant bits of the current instruction address to the specified index register. If the IWR is zero, the section and page portions of the current instruction address are transferred to the accumulator. If the IWR is not zero, the IWR is transferred to the accumulator. (Timing: 4 microseconds.)

STA - Store Accumulator

Stores the contents of the accumulator in the address specified by the direct or indexed address. (Timing: 6 microseconds.)

ADA - Add to Accumulator

Adds to the accumulator the value specified by the immediate, the direct, or the indexed address. In the event of an overflow condition, the overflow character is lost. (Timing: 4 microseconds when immediate address form is used, otherwise, 6 microseconds.)

ADX - Add to Index Register

Adds the IWR to the index register specified. In the event of any overflow condition, the overflow character is lost. (Timing: 4 microseconds.)

SUA - Subtract from Accumulator

Subtracts from the accumulator the value specified by the immediate, the direct, or the indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SUX - Subtract from Index Register

Subtracts the IWR from the index register specified. (Timing: 4 microseconds.)

ANA - Logical AND to Accumulator

Logically ANDs with the accumulator the value specified by the immediate direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise 6 microseconds.)

SAN - Shift and Logical AND to Accumulator

This form of the logical AND instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the logical AND of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

ERA - Exclusive OR to Accumulator

Performs an Exclusive OR of the accumulator and the value specified by the immediate, direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SER - Shift and Exclusive OR to Accumulator

This form of the Exclusive OR instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the Exclusive OR of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

IRA - Inclusive OR to Accumulator

Logically ORs the accumulator and the value specified by the immediate, direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise 6 microseconds.)

SIR - Shift and Inclusive OR to Accumulator

This form of the inclusive OR instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the inclusive OR of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

CPA - Compare Accumulator

Compares the contents of the accumulator to the value specified by the immediate, direct, or indexed address. The condition register value is changed to reflect the high, low, or equal result of the comparison. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

CPX - Compare Index Register

Compares the contents of the index register specified to the IWR. The condition register value is changed to reflect the high, low, or equal result of the comparison. (Timing: 4 microseconds.)

Operation Register and Decoder.

The operation register is an 8-bit latch shown on sheet 7 of the processor logic diagram. During instruction cycle I3, it is loaded with the output of the memory output multiplexer, which is the left half of an instruction word (IWL). This word is held in the operation register until the next instruction is loaded during the next instruction cycle I3.

The contents of the operation register are designated OP0 through OP7, and are distributed to several circuits. The primary use of the operation code is to specify the instruction to be executed, and this is accomplished as shown in tables 2-2 through 2-4.

Table 2-2. Operation of Class 1 Decoder B15

OP CODE BIT					ACTIVE B15 PIN	ACTIVE SIGNAL	INSTRUCTION TYPE
7	6	5	4	3			
0	1	0	0	0	1	BCD-UNC	BRANCH UNCONDITIONAL
0	1	0	0	1	2	BCD	BRANCH
0	1	0	1	0	3	STK-UNC	STACK & BRANCH UNCONDITIONAL
0	1	0	1	1	4	STK	STACK & BRANCH
0	1	1	0	0	5	EXIT	EXIT
0	1	1	0	1	6	N-15-X-OP	15X OP CODE
0	1	1	1	0	7	N-EXIT-BRANCH	EXIT & BRANCH UNCONDITIONAL
0	1	1	1	1	9	N-I/O-OP	I/O OP CODE

Table 2-3. Operation of Class 1 and 15X Decoder B16

15-X-OP SIGNAL	OP CODE BIT			ACTIVE B16 PIN	ACTIVE SIGNAL	INSTRUCTION TYPE
	2	1	0			
LOW	0	0	0	1	N-SECTION-OP	SET MEMORY SECTION
	0	0	1	2	N-151-OP	SET MEMORY CONTROL
	0	1	0	3	N-152-OP	SET MEMORY SECTION & CONTROL
	0	1	1	4	N-153-OP	SET CONDITION REGISTER
	1	0	0	5	N-SENSE-1	LOAD SENSE SWITCH
	1	0	1	6	N-155-OP	LOAD PROCESSOR STATUS
	1	1	0	7	N-156-OP	INTERRUPT CONTROL
	1	1	1	9	N-157-OP	LOAD TERMINAL ID

Table 2-4. Operation of Class 2 and 3 Decoder B14

OP CODE BIT				ACTIVE B14 PIN	ACTIVE SIGNAL	INSTRUCTION TYPE
7	6	5	4			
1	0	0	0	1	N-LOAD-OP	LOAD
1	0	0	1	2	N-STORE-OP	STORE
1	0	1	1	4	N-SUB-OP	SUBTRACT
1	1	0	0	5	N-AND-OP	LOGICAL AND
1	1	0	1	6	N-EX-OR-OP	EXCLUSIVE OR
1	1	1	0	7	N-COMP-OP	COMPARE
1	1	1	1	9	N-OR-OP	INCLUSIVE OR

The operation decoder is shown at the left side of sheet 3 of the processor logic diagram. Its purpose is to decode the left half of the instruction word (IWL) and activate control lines to indicate which instruction is held in the operation register. Tables 2-2 through 2-4 list Class 1 through 3 instructions and show which control lines are activated by each instruction.

As described earlier in this section, the two high order bits (OP7 and OP6) indicate the class of instruction (0, 1, 2, or 3). For the four class 0 instructions, both OP7 and OP6 are zeros. This causes the operation decoder to activate the BR-OFFSET-OP line.

Class 1, 2, and 3 instructions require that additional bits of the operation code be decoded to further define the instruction. In addition, some Class 2 and 3 instructions may use any one of three addressing modes (immediate, direct, and indexed). This choice is made by OP3, which is a zero for the immediate addressing mode and a one for the other two modes. By sensing bits OP7 and OP3, the operation decoder can determine that the instruction is class 2 and 3 and what addressing mode is to be used. For the immediate addressing mode, the decoder activates the DIRECT line. The other two modes, in which bit OP3 is a one, cause the decoder to activate the INDIRECT line.

Cycle Counters.

The cycle counters are shown on sheet 1 of the processor logic diagram. Their purpose is to originate the I1, I2, I3, I4, E2, and E3 signals required to execute each instruction and allow the I/O controller to gain memory access. The cycles required for each instruction and the order in which they occur were described earlier in this section.

There are two cycle counters, one that produces the I1 through I4 signals and another that produces the E2 and E3 signals. Although they are called counters, these units are each 4-bit shift registers. In the I-cycle counter, the I1 through I4 signals are each produced by the output of one of the stages in the shift register, with I1 coming from the least significant stage and I4 from the most significant stage. In the E-cycle counter, the E2 and E3 signals are produced by the two center stages.

Although the shift registers are simple, a group of complex control circuits is required to control them. These control circuits are discussed in the following paragraphs. First to be described is the way in which both counters are cleared.

When power is first applied, the processor clear signal is driven low. This remains low until the program loading sequence is started. The processor clear signal holds both counters cleared until program loading begins.

Next, each counter requires a positive-going transition of its clock signal to be loaded or shifted. Both counters are clocked by the same signal. During the initial cycle, the first clock signal is formed by the output of the I-START flip-flop and an output of the oscillator. After that, each I/O-CLR-GATE (I/O clear gated by the stop signal) signal and the oscillator output combine to produce the clock signal. This means that the clock signal is formed at the end of each machine cycle.

The remaining control signal applied to the shift registers is the load/shift line. When this signal is low, the shift registers can be loaded with the parallel input; however, a clock pulse is required to load. When the load/shift signal is high, the clock pulse can shift the contents of the registers.

The load/shift line for the E-cycle counter is simply the I4 signal from the I-cycle counter. Thus, when cycle I4 is reached, the E-cycle counter can be loaded, and when the I4 cycle ends the E-cycle counter can be shifted.

Control of the load/shift line to the I-cycle counter is more complex. Examine first the four parallel inputs to the counter. They are:

Stage	Input
D(I4)	Fixed 0 (grounded)
C(I3)	Controlled
B(I2)	Fixed 0 (grounded)
A(I1)	Controlled

Based on the above, it can be seen that when the load signal is present, the I1 and I3 stages depend upon controlled inputs while the I2 and I4 stages are loaded with zeros.

The inputs to the E-cycle counter are as follows:

Stage	Input
D	Controlled by OP7
C(E3)	Controlled by direct address mode
B(E2)	Controlled by indirect address mode
A	Fixed 0 (grounded)

In addition to the parallel inputs, each shift register has a serial input to its least significant stage that is entered when the register is shifted. At the E-cycle counter, this input is grounded, so zeros are entered. At the I-cycle counter, this input, which would generate the I1 signal, is controlled.

When the program load sequence is started, the initial conditions load a 0001 into the I-cycle counter; this produces the I1 signal. The load/shift line then switches to shift and the one in stage A moves to stage B during the I/O clear signal at the end of cycle I1. Cycle I2 is then performed. At the end of I1, I3, and I4, the register is shifted again.

As mentioned in the discussion of instruction cycles, the purpose of the I2 cycle is to acquire and modify the IAWR. The conditions that make cycle I2 unnecessary were described earlier. If they exist, the load/shift line is switched to load at the end of I1. This places a zero in stage A (I1), places a zero in stage B (I2), and loads a one in stage C (I3), effectively skipping I2. The shift mode is then resumed and, at the end of I3, the marker bit is shifted to stage D, the I4 position, and I4 is performed.

As previously mentioned, the I4 signal controls the load/shift line to the E-cycle counter. At the end of I4, the E-cycle counter can be loaded. If the instruction is Class 0 or 1, no E-cycles are required. Bit OP7 from the operation register is sensed to determine this. If the current instruction is in Class 0 or 1, the E-cycle counter is loaded with all zeros and a one is entered at the serial input of the I-cycle counter, restarting the I-cycle counter at I1.

If the current instruction is in class 2 or 3, the value loaded in the E-cycle counter depends upon whether the direct or indirect address mode is required. For the direct mode (an instruction using immediate addressing), a one is placed in stage C, the E3 stage. For the indirect mode, which means that the direct address mode or the indexed address mode is selected, a one is placed in stage B, the E2 stage. At the end of each machine cycle, the E-cycle counter is shifted. When the end of cycle E3 is reached, a one is entered in the serial input of the I-cycle counter, returning to cycle I1.

The STOP flip-flop shown on the logic diagram is used only during single cycle test with the maintenance console. It blocks certain timing signals and enables the memory refresh operation during the single cycle mode. Each time the EXT-ADV (external advance) signal is applied to the control circuits, the STOP flip-flop allows one machine cycle to be performed.

Initial Conditions.

When power is first applied, the output of the POWER ON switch is routed to several circuits on sheet 1 of the processor logic diagram. There, it clears the program load flip-flop and drives the PROC-CLR and GEN-CLEAR signals low. These two clear signals reset many circuits, and hold them in the reset condition. Included are the clock circuits and the cycle counters, so the processor is inactive even though power is applied.

The next step taken is the operation of the PROGRAM LOAD switch. This sets the program load flip-flop and the reset flip-flop. When the switch is released, the reset flip-flop is cleared. This triggers a one-shot and a delay network. The processor and general clear signals then move to a high and remain there. This allows the circuits in the processor to operate.

The I-START flip-flop is set and, in turn, it sets the INITIAL CYCLE flip-flop. These flip-flops start the cycle counter at I1, as described earlier. In the meantime, the program load signal has selected tape drive 2, started the tape, and begun reading. The program load signal also cleared the stack pointer, thus establishing the first location in the address stack as the source of the IAW.

During cycle I1, the following events take place:

a. At the address net, the stack pointer addresses location 040.

b. The contents of location 040 are read, but the initial cycle signal prevents their transfer to the B inputs of the adder. (Normally, the IAWL is gated to the adder during I1.)

c. The initial cycle signal also prevents the IAWL from being incremented by 2 in the adder, as it normally is.

d. The result is that an IAWL of 001 is formed at the adder output, transferred to the MAR, and also stored in location 040. The least significant bit being a one has no effect because it is not used at the MAR.

During cycle I2, the following events take place:

a. At the address net, the stack pointer addresses location 041.

b. The contents of 041 are read but the initial cycle signal prevents their transfer to the adder. Normally the IAWR goes to the adder at this time.

c. The initial cycle signal forces a 001 into the IAWR at adder input B. Since there is no carry from cycle I1, the 001 appears unchanged at the adder output.

d. At T-SUM time, the IAWR of 001 is loaded into the high-order position of the MAR. This makes the composite instruction address P01-000.

e. The IAWR is also stored in location 041 during this cycle.

Ever since the PROGRAM LOAD switch was depressed, the ROM load flip-flop has been set. Now, the cycle counter moves the I3 and allows the ROM to be addressed by the output of the address net. (The program stored in the ROM is discussed in the ROM description.) Simultaneously, the memory output multiplexer is switched to accept the ROM output rather than the memory output.

The following events take place during I3:

a. At the address net, address P01-000 is accepted from the MAR. This selects the ROM and causes the first instruction to be read out.

b. The first IWL (201) is transferred to the operation register, B register, and MDR at TE time.

c. The initial conditions flip-flop is reset, eliminating all special conditions. (The I-START flip-flop was reset at TX time of the first machine cycle.)

The cycle counter then moves to I4 in the normal manner, and the following events take place during I4:

a. At the address net, the MAR addresses the IWR in the ROM.

b. The IWR (030) of the first instruction is read from the ROM and operated on.

At this point, the remaining seven instructions of the ROM program are read and executed normally. They read the first record from the tape in drive 2 and branch to the first instruction read in. This instruction steps the stack pointer, which eliminates the SP-0 signal and clears ROM load flip-flop, ending the loading sequence.

Tape Loader ROM.

After power is applied to the processor, the first step is to load the first program. In most models of the 1501, this program is stored in the first record on the tape cartridge mounted on tape drive 2, and it must be read into memory and performed in order to start processor operation. How this is accomplished was described in the initial conditions description.

The PROGRAM LOAD switch sets the program load flip-flop which, in turn, produces the PRG-LOAD signal that is distributed to the ROM circuits on sheet 7. The ROM holds an eight-instruction program. This program reads 128 bytes from tape drive 2, stops the tape, and then branches to the first instruction read from the tape. From this point, the program read from the tape controls the processor.

The program stored in the ROM is shown in table 2-5. The instruction addresses are held in the instruction address stack locations 40/41, and the processor runs through the normal instruction cycles for each instruction read from the ROM. Thus, the instruction addresses are stepped in the normal way.

The four low-order positions (20 through 23) of the address net are applied to the ROM and, during I3 and I4, the instructions are read from the ROM and applied to the memory output multiplexer. The fact that the ROM load flip-flop is set allows the multiplexer to accept the ROM output rather than the memory output. In this manner, the instructions read from the ROM are substituted for the normal memory output.

Table 2-5. Tape Loader ROM Program

INSTRUCTION LOCATION	INSTRUCTION	DESCRIPTION
P01-000	201-030	LDX-030, LOAD IX1 WITH 030
P01-002	170-007	TRANSFER BYTE, UNCONDITIONAL
P01-004	231-002	STA, STORE BYTE IN PAGE 00-030, INCREMENT IX1
P01-006	341-230	CPX, COMPARE IX1 TO LITERAL 230
P01-010	111-003	BRL, BRANCH TO P01-002 IF RECORD NOT READ
P01-012	170-016	READ STATUS TO RESET READ
P01-014	170-005	STOP TAPE
P01-016	100-030	BRU, BRANCH TO PAGE 00-030

The first instruction in the ROM program loads index register 1 with an octal 30. The second instruction is a Transfer Byte instruction that is sent to the I/O controller for execution. Since the right byte is 007 rather than 207, the processor stalls until the byte is transferred into the accumulator.

When the byte is available in the accumulator, the third instruction (231-002) stores it in location P00-030 and increments index register by 1, bringing it to 31. The fourth instruction compares the contents of index register 1 to the literal 230. If the index register has not reached 230 yet, the next instruction branches back to continue reading.

When 128 bytes have been read, the index register equals 230 and the branch is no longer performed. The reset read and stop tape instructions then terminate the reading operation, and the last instruction branches to P00-030 to acquire the next instruction; this being the first instruction loaded from the tape.

SIO Channel Loader ROM.

There are no tape drives in the 1501-CL from which a program can be read, therefore, the 1501-CL must acquire its programs from another terminal over the serial I/O channel. The equipment changes required to do this are very simple, as follows:

a. The standard ROM with the tape loader program is replaced by a 16-pin socket.

b. An SIO channel loader ROM, mounted on a small circuit board, is then connected to the new socket by a ribbon cable. (The logic diagram number of this board is 4842.)

c. Another address bit (24) is wired to the ROM in order to address 16 additional locations.

The new ROM holds a program that communicates with the remote terminal over the SIO channel, acquires a program, and branches to the first instruction of that program. From that point on, the program read from the remote terminal controls operation of the 1501-CL.

The remote terminal is either a Model 1503 or Model 1501-40 operating with several 1501-CL models in the "clustered" mode. Refer to the System Description manuals for further explanation of this mode of operation.

Program loading is started in the same way as previously described for the tape loader ROM. This causes the 15-instruction program held in the SIO loader ROM to be executed. As will be shown in the following paragraphs, this program reads a one-page program (256 bytes) from the remote terminal and then branches to an instruction in the newly acquired program. The program read from the remote terminal is actually a bootstrap loader program whose function is to acquire operating programs for the 1501-CL.

It should be noted at this point that the 1501-CL is equipped with the dual SIO channel option so it has two serial I/O channels to operate. The remote terminal from which the 1501-CL acquires programs is connected to channel 2, and peripheral equipment operated by the 1501-CL is connected to channel 1. In order to understand the operation of the SIO channel loader program, it may be necessary to read the description of the SIO channel circuits given later in this manual.

The SIO channel loader program shown in table 2-5A begins with the assumption that SIO channel 1 could be in operation with a peripheral unit when the PROGRAM LOAD switch is depressed and this operation is to be terminated in an orderly manner. This is the purpose of the first four instructions, and they operate as follows:

a. The first instruction (200-010) loads the accumulator with 010, which will be used as a control byte sent to the peripheral unit active on SIO channel 1.

b. 175-004, the second instruction, is the Stop Transmit instruction. This terminates communications on the SIO channel after the peripheral unit has returned its status word.

c. 175-003 is the Transmit Control Byte instruction. This causes the control byte, which is 010 placed in the accumulator by the first instruction, to be sent to the active peripheral unit. There, the control byte is interpreted as a send status command.

d. The peripheral unit then returns its status word and the fourth instruction, 175-001, Transfer Byte, Unconditional, transfers the status word to the accumulator. The status word is not used; the purpose of this instruction is simply to allow communications to be terminated. If the channel was not active, a dummy signal is returned in response to this instruction and the same effect is achieved.

The next five instructions then prepare the 1501-CL circuits to operate in the slave mode and wait for contact from the remote terminal, which acts as a master. This is accomplished as follows:

a. Instruction 157-000, Load Terminal ID, loads the address of this 1501-CL from the address plug into the accumulator.

b. Instruction 175-007, Set Device Address, moves the address from the accumulator to the address register in the SIO channel circuits.

c. Next, instruction 200-102 places code 102 in the accumulator. When this is read by the master terminal, it is interpreted as a request for the bootstrap load.

d. Instruction 175-011, Set Slave Mode, then places the 1501-CL SIO channel circuits in the slave mode and transfers the code 102 from the accumulator to the input data register in the SIO circuits.

e. The last instruction in this group 173-014, Switch to Channel 2, disconnects 1501-CL from channel 1 and switches it to channel 2. This allows the master terminal to contact the 1501-CL.

Table 2-5A. SIO Channel Loader ROM Program

INSTRUCTION LOCATION	INSTRUCTION	DESCRIPTION
P01-000	200-010	Load accumulator with control byte 010
P01-002	175-004	Stop transmit (after status is received)
P01-004	175-003	Transmit control byte
P01-006	175-001	Transfer byte to accumulator (accept status)
P01-010	157-000	Load terminal ID into accumulator
P01-012	175-007	Set device address
P01-014	200-102	Load accumulator with code 102
P01-016	175-011	Set slave mode
P01-020	173-014	Switch to channel 2
P01-022	201-000	Load index register 1 with 000
P01-024	175-001	Transfer byte to accumulator
P01-026	231-012	Store accumulator, step index register 1
P01-030	341-000	Compare index register 1 to 000
P01-032	111-024	Branch to P01-024 until program is read
P01-034	122-356	Stack and branch to P02-356 for first step

The purpose of the remaining instructions is to accept 256 bytes from the master terminal, store them in memory, and then branch to the first instruction. This is accomplished as follows:

a. Instruction 201-000 loads index register 1 with 000.

b. Next, instruction 175-001, Transfer Byte, Unconditional, is executed. This instruction transfers a byte from the SIO channel circuits to the accumulator, and it stalls the processor until the byte is ready. Thus, the processor remains at this instruction until a byte has been received from the master terminal.

c. After a byte is received, the processor continues. The next instruction 231-012, Store Accumulator, transfers the first byte to location 000 in page 02 in memory and steps index register 1 by one.

d. The next instruction, 341-000, Compare Index Register, compares the current contents of index register 1 to 000. Since the index register was stepped to 001 when the first byte was read, it will not be 000 again until 256 (377 octal) bytes, plus one, have been read. Each time this instruction finds that the index register is not 000, it sets the condition register to "high". When the index register is 000, the condition register is set to "equal".

e. Instruction 111-024, Branch on High, examines the state of the condition register. As long as the condition is "high", the BRH instruction returns to location 024 in the ROM to obtain the next instruction. This is the Transfer Byte instruction. After the entire program is read from the master terminal, index register 1 will have been returned to zero and the branch no longer takes place.

f. The last instruction is 122-356, Stack and Branch Unconditional to page 02, location 356 (octal). This is the first instruction to be executed in the program acquired from the master terminal.

Interrupt Controls.

The processor is provided with two methods of interrupting normal instruction execution: one is the program interrupt position of the PROGRAM LOAD switch on the console; the other is an external interrupt signal input for use by external devices. The circuits that accept the interrupt signals are shown on sheet 2 of the processor logic diagram, and a timing diagram of their operation appears in figure 2-14.

When the switch is pushed, an interrupt signal from the switch sets the program interrupt flip-flop (136A). (The flip-flop is cleared when the switch is released.) This triggers the interrupt timing one-shot (182). An external interrupt signal also triggers the one-shot. In turn, the interrupt timing one-shot sets the interrupt store flip-flop to record the fact that one interrupt signal has occurred.

At TX time of the I3 cycle, the output of the interrupt store flip-flop is sampled and the interrupt sync flip-flop (179) is set. This synchronizes the interrupt signal with the processor timing. The output of the interrupt sync flip-flop clears the interrupt store flip-flop. If a second interrupt signal occurs before the interrupt store flip-flop is cleared, the interrupt overflow flip-flop is set. The output of this flip-flop becomes bit 6 in the processor status word.

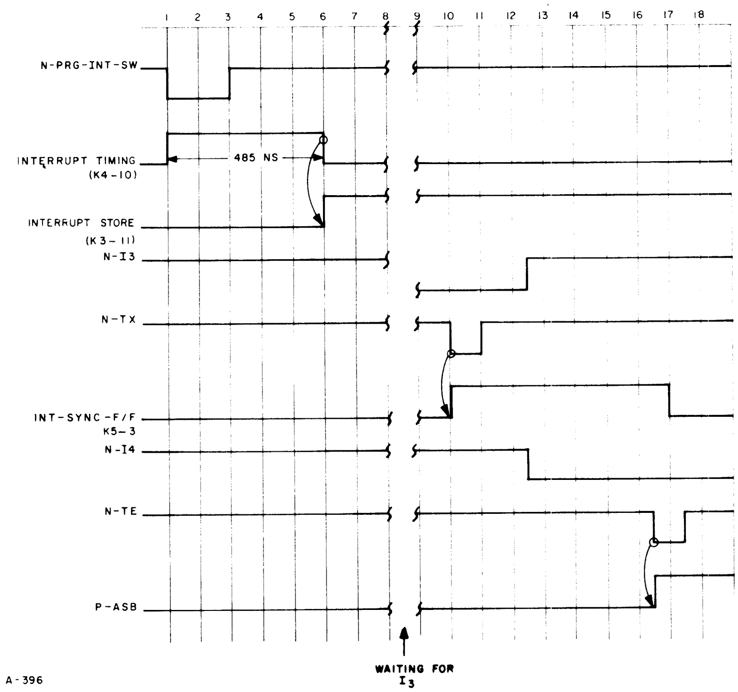


Figure 2-14. Interrupt Controls Timing Diagram

If no inhibiting conditions exist, the output of the interrupt sync flip-flop causes the automatic stack and branch flip-flop (170) to set at TE time of the I4 cycle. The ASB (automatic stack and branch) signal causes program control to be transferred to location 000 of page P03, which contains the first instruction of the interrupt service routine.

Four conditions can inhibit the stack and branch operation by preventing the ASB flip-flop from being set. They are:

a. The tape is moving, as indicated by the RUN-F/F signal.

b. A keyboard character is available to be transferred to the accumulator. The KBD-CHAR-READY signal indicates this.

c. A Disable Interrupt instruction has been executed, as indicated by the INT-INHIBIT signal.

d. The USE-SECTION-F/F signal is present. This prevents an interrupt from taking place until the IAWR has been restored to the stack pointer address and thus prevents the page portion from being lost.

After the ASB flip-flop is set, it gates the I4 signal to produce the INT-TIME-CNTL (interrupt time control) signal. This signal is routed to the cycle counters on sheet 1, where it prevents cycles E2 and E3 and forces the cycle counter to cycle I1. The ASB signal also performs several other functions, as follows:

a. It sets the INITIAL-CYCLE-F/F on sheet 1, and this produces an address of 000 at the adder output just as it does during the program loading discussed earlier in this section. This produces an IAWL of 000 during cycle I1.

b. At the stack pointer on sheet 3, the ASB signal forces an unconditional stack operation to step the stack pointer. Also on sheet 3, the ASB signal clears the TK-BR-F/F to eliminate the control of the address by this flip-flop.

c. At the adder input switch controls on sheet 6, the ASB signal enters a threc in the IAWR during cycle I2. This produces the page 03 address.

d. At the U and V bit circuits on sheet 8, the ASB signal disables the UV gate. This produces U and V bits of zero when the IAW is stored in the new stack pointer address and prevents the previous U/V bits from affecting the interrupt servicing routine.

To summarize the ASB operation:

a. The stack pointer is stepped up by one.

b. During I1 and I2, an IAW of P03-000 is forced through the adder, stored in the new stack pointer addresses, and loaded into the MAR.

c. During I3 and I4 the next instruction is read from P03-000. This is the first instruction of the interrupt servicing routine.

Branch Controls.

The key element in the branch controls is the TK-BR (take branch) flip-flop, which is shown on sheet 3 of the processor logic diagram. This flip-flop is initially cleared by the processor clear signal, so it begins in the cleared state. It is also cleared at I3, which begins each new instruction, and by the automatic stack and branch operation initiated by interrupt signals.

The decision as to whether the take branch flip-flop is to be set is made at TX-time of cycle I4 of the current instruction. Two gates are sampled at this time to make the branch decision.

The first gate is enabled by the four class 0 instructions TLJ, TMJ, TLX, and TMX. In each case, if the accumulator and literal are equal, the ACC=MDR signal allows the take branch flip-flop to be set.

The second gate monitors the results of the branch, stack and branch, and exit and branch instructions. If the instruction determines that a branch should be taken, the take branch flip-flop is set. Of course, the stack and exit instructions also control the stack pointer.

The take branch signal is then distributed to several places:

a. At the cycle counters on sheet 1, it controls whether or not the next cycle I2 is required. If a branch is to be taken, I2 is required.

b. At the adder controls on sheet 4, it establishes the conditions necessary to introduce the branch address.

c. At the adder input switch on sheet 6, it blocks the addition of two to the IAW.

d. At the address controls on sheet 8, it allows the USE SECTION and USE PAGE flip-flops to be set if the other required conditions are met.

Stall and Skip Conditions.

In order to exchange data with the peripheral equipment operated by the I/O controller, the processor sends a Transfer Byte instruction to the controller. This instruction can be executed in either of two modes: stall or skip. If the IWR is 007, the instruction is executed in the stall mode; 207 in the IWR chooses the skip mode.

The Transfer Byte instruction is decoded and executed in the I/O controller. If the peripheral unit is not ready, the controller returns the stall signal to the processor. The action taken by the processor then depends upon the instruction mode selected, as follows:

a. If the stall mode is selected, the processor remains at the current instruction until the peripheral unit is ready.

b. If the skip mode is selected, the processor skips the next instruction and continues with the program.

The processor circuits that accomplish this are the adder input switch controls, shown in the lower left corner of sheet 6 of the processor logic diagram, and the skip flip-flop shown in the upper right corner of sheet 1. The I/O STALL signal from the I/O controller is applied to the adder input switch controls. There, it prevents the adder from adding two to the IAWL during cycle I1. This means that the same address is returned to the current stack pointer location and the processor executes the Transfer Byte instruction over and over until the I/O controller removes the stall signal. Note that there are two other external stall signals available to the adder input switch controls but they are not used.

The adder input switch controls also forward the STALL-COND signal to the skip flip-flop. During cycle I4, the STALL-COND signal and bit MDR-7 are tested. MDR-7 is the high-order bit of IWR of the Transfer Byte instruction. If this bit is a one, the IWR is 207-Skip on Busy. A zero in this position means that the IWR is 007-Stall on Busy. Assuming that MDR-7 is a one, the skip flip-flop is set. This sends the SKIP signal to the adder input switch controls. Note that the skip flip-flop is cleared at the next I3, which begins the next instruction. (A HS-SKIP signal is also available but is not used.)

During I1, the cycle in which the IAWL is being modified, the SKIP signal applied to the adder input switch controls causes the IAWL to be increased by four instead of the usual two. This causes the processor to skip the next instruction and proceed.

Stack Pointer.

The stack pointer, which is shown on sheet 3 of the processor logic diagram, is a 4-bit counter that can be incremented or decremented under program control. It produces a 4-bit address, SP 1-4, that is applied to address net positions 1-4, respectively. During machine cycles I1 and I2, the stack pointer output addresses memory to acquire next instruction address.

Address net bit position 0 is a zero during cycle I1 and one during cycle I2; position 5 is a 1 during both of these cycles. This causes the stack pointer to address even locations 40 through 76 (octal) during I1 and odd locations 41 through 77 during I2. These 32 locations, 40 through 77 (octal), constitute the instruction address stack.

The purpose of the instruction address stack is to provide a means of returning to the point at which the program branched to a subroutine. When the processor is started, the stack pointer begins at zero. This addresses locations 40 and 41 to acquire the IAWL and IAWR, respectively. Each time that an instruction is executed, the IAW read from these locations is stepped by two and returned to memory. Thus, when the contents of locations 40 and 41 are read to acquire the IAWL and IAWR for the next instruction, the IAWL and IAWR address the next instruction in sequential order. If a branch instruction is executed, the address normally returned to the stack pointer locations is replaced by the "branch to" address, and instructions are again executed in sequence, starting at the "branch to" address.

There are, however, eight instructions that change the stack pointer addresses to new locations. They are:

a. TLX - Test Literal and Exit

b. TMX - Test Mask and Exit

c. SBU - Stack and Branch Unconditional

d. SBE - Stack and Branch on Equal

e. SBH - Stack and Branch on High

f. SBL - Stack and Branch on Low

g. EXB - Exit and Branch

h. EXU - Exit Unconditional

The instructions can increase or decrease the stack pointer so as to cause the stack pointer tσ "point" to different locations from which the IAWL and IAWR are to be read. The stack instructions step the counter up, and the exit instructions step the counter down.

The stepping action always takes place at TX time of cycle I4. One gate (162) forms the step up signal, and another (161) forms the step down signal. The conditions required for a step up are:

a. STACK-OP (stack operation) which indicates that one of the four stack instructions or an automatic stack and branch (ASB) operation is being executed.

b. The condition being tested by the conditional stack instructions (SBE, SBL, SBH) is met.

The conditions required for a step down condition are:

a. One of the exit instructions is being executed.

b. The condition being tested by the conditional exit instructions TLX and TMX is met.

Use of the U and V Bits.

Two control bits, "U" and "V", determine which memory page and section that the direct address, the indexed address, or the branch address affects. These two control bits are stored in latches on sheet 8 of the processor logic diagram. These are established by the SMC and SSC instructions. When either of these instructions is executed, bits 6 and 7 of the IWR are stored in the latches to become the V and U bits, respectively.

Each time that the IAWR is restored to a stack pointer address during cycle I2, the U and V bits occupy positions 7 and 6, respectively, of the IAWR. (Refer to figure 2-7 for the format of the contents of the stack pointer addresses.) When the IAWR is read during the next I2 cycle, the U and V bits are loaded into their respective latches on sheet 8 and used to control the next instruction.

One function of the U bit is to determine which section of memory is to be used in indexed address instructions. As mentioned in the overall processor description, addresses 1 through 7 in each memory section are reserved for use as index registers. Whenever the U bit is a zero, the index registers in section 0 are to be used. Whenever the U bit is set to a one, the index registers to be used are those in the same section as the current instruction.

The second function of the U-bit is to determine which level and section the direct address affects. When the U-bit is a zero, the IWR (address) of all instructions executed in the direct address mode applies to level 0 of section 0. Conversely, when the U-bit is a one, the direct address applies to level 0 of the same section as the current instruction.

To summarize the function of the U-bit:

a. When it is a zero, the section 0 index registers are to be used and direct addresses apply to level 0 of section 0 (page 00).

b. When it is a one, the index registers in the same section as the instruction are to be used, and the direct addresses apply to level 0 of the same section as the instruction (page 00, 10, 20, etc.).

The V-bit performs a similar function for branching operations. If the V-bit is a one, any branch, stack-and-branch or exit-and-branch instruction with page 0 specified in the branch address causes the branch to occur within the current section and page. If any page other than 0 is specified in the branch address, the V-bit control is inactive and a normal branch occurs. For example, assume that the V bit is a one and a branch instruction located in page 5 specified a branch to page 0 location AAA. The resulting branch will be, to page 5, location AAA. If the branch address specified was page 6 location BBB, then the resulting branch will be to page 6, location BBB. If the branch instruction is located in any page other than page 0 and a branch to page 0 is desired, the V bit must left at a zero.

To summarize the use of the V-bit:

a. If it is zero, a branch instruction located in any page other than page 0 can branch to page 0.

b. If it is a one, a branch instruction with page 0 specified in the branch address will branch within the same section and page as the instruction.

Memory Address Register.

The memory address register (MAR) is a 16-bit register shown on sheet 9 of the processor logic diagram. Its purpose is to hold a memory address that the processor applies to the address net and subsequently to memory. The MAR is divided into two 8-bit sections, a low-order section and a high-order section. Since only 14 bits are required for the memory address, the two high-order bit positions of the 16-bit register are not used.

The address loaded into the MAR originates in the memory data register (positions 0-7), the B register (positions 2-7), or the output switch (positions sum 0-7). Selection of the address souce is accomplished by a set of gates and multiplexers. Since the input to the low-order (0-7) section of the MAR differs considerably from the input to the high-order (8-13) section, they are discussed separately in the following paragraphs.

An eight-position, 2-to-1 multiplexer preceeds the low-order section of the MAR. Applied to this multiplexer are bits 0-7 from the MDR and sum positions 0-7 from the output switch. Only during instruction cycle E2, when the MDR holds the contents of an index register or the direct address, is the MDR 0-7 input allowed to reach the MAR. During all other instruction cycles, the sum 0-7 lines from the output switch reach the low-order section of the MAR.

The input to the high-order section of the MAR is controlled by two sets of gates. The first set of six gates receives B register positions 2-7, and the second set receives the sum 0-5 lines from the output switch. During instruction cycle I2, the sum 0-5 positions are gated to the high-order section of the MAR. This brings the modified page portion of an instruction address into the MAR.

During instruction cycle E2, the INDEX-0 (index register number is 0) signal is checked and if it indicates that an index register has been selected by the current instruction, bits 2-7 from the B register are gated to the high-order section of the MAR. This is the page part of a composite address. The other part (the byte number) is read from an index register.

To summarize the inputs to the MAR:

a. The adder output positions sum 0-7 are gated to the low-order portion in all cases except cycle E2. During E2, the output of the MDR, which holds the direct address or the contents of an index register, is gated to positions 0-7 of the MAR.

b. The adder output positions sum 0-5 are gated to the high-order portion during cycle I2. In this case the data on these lines is a modified page portion of an instruction address. During cycle E2 of instructions using the indexed address mode, the page portion of a composite address reaches the MAR from the B register. These are bits 2-7 of the IWR.

The next action to be taken is to load the MAR with the input selected by the multiplexers and gates. Again, the low-order and high-order sections are handled differently; therefore, they are discussed separately in the following paragraphs.

All loading is controlled by the T-SUM signal, which is generated once each machine cycle. This signal is applied to both the low and high-order sections of the MAR. Thus, loading can take place during the second half of each machine cycle if the other conditions are met.

The conditions required to load the low-order section are: (1) instruction cycle I1 and T-SUM or (2) instruction cycle E2 and T-SUM. During cycle I1, the adder output, which is the modified byte part of an instruction address, is gated into the MAR. During instruction cycle E2, the MDR output, which is the direct address or the contents of an index register, is loaded into the MAR.

The conditions required to load the high-order section are: (1) instruction cycle I2 and T-SUM or (2) instruction cycle E2 and T-SUM. During cycle I2, the sum 0-5 positions are available to be loaded. This is the modified page part of an instruction address. During cycle E2, the B register positions 2-7 hold the page portion of the composite address. The composite address being made up of this address and the contents of an index register.

The MAR outputs are labeled according to their weight, 20 for 2^0, the low-order bit, through 213 for 2^{13}, the high-order bit. At the address net, the MAR output is gated through bit positions 1-13 during the first half of I3 and I4. Bit 0 is a zero during I3 and one during I4, thereby addressing the two successive bytes of the instruction word. If direct or indexed addressing is used, the MAR is also gated through bit positions 0-13 during E3.

Address Net.

The address net is a set of 14 gates shown on sheet 9 of the processor logic diagram. Their purpose is to select a 14-bit address and apply it to memory. The net also provides the same address at plug D of the processor card for use with an external memory. Four bits from the address net (0-3) are also routed to the ROM in order to read the loading program.

A summary of the inputs and outputs of the address net appears in figure 2-15. At the top of the figure the 14-bit address output of the net is shown. These positions are numbered 20 through 213 on the logic diagram, with 20 being the LSB and 213 the MSB. Note that positions 0-7 select the byte locations within a page and positions 8-13 select the page number.

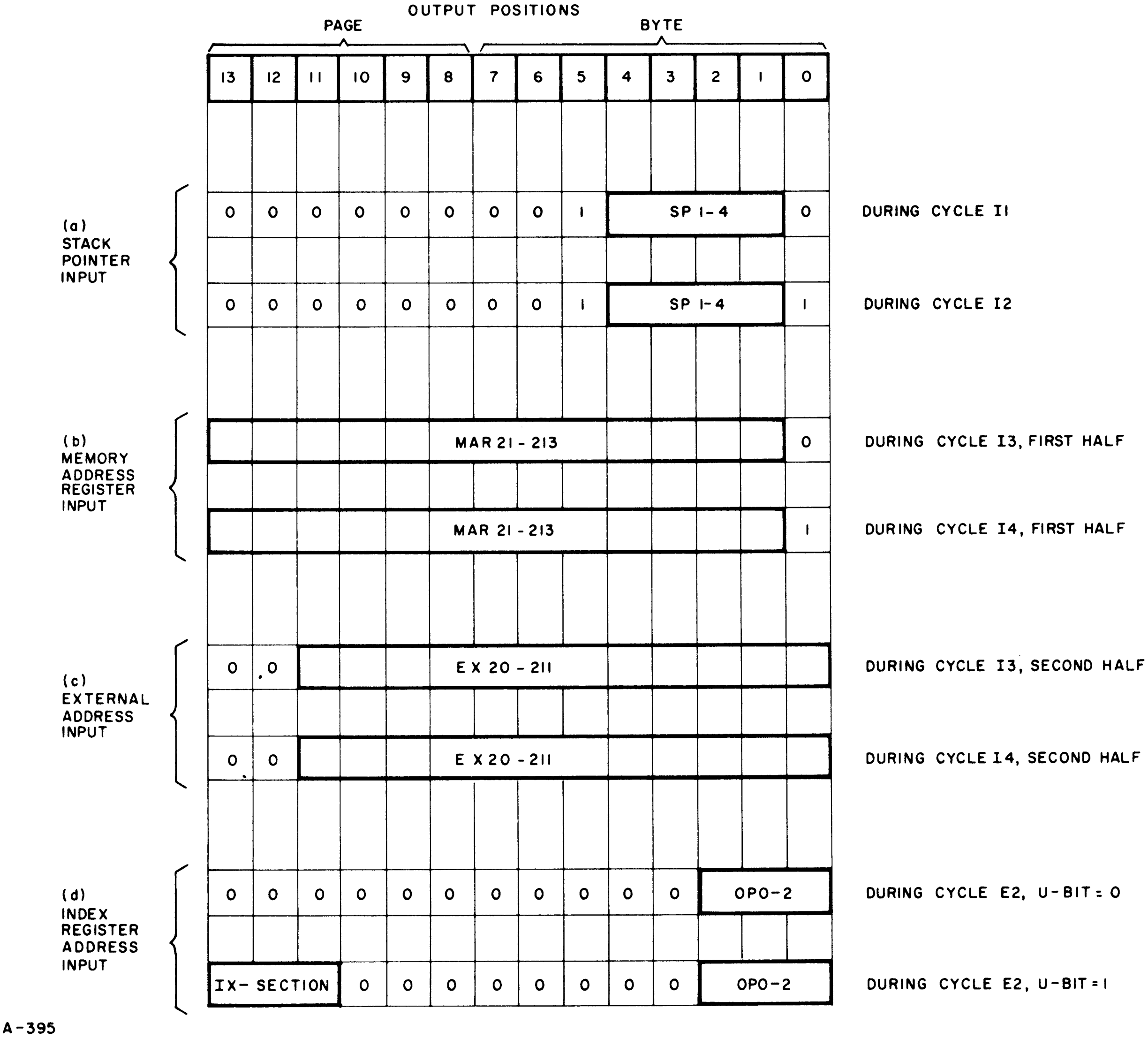
There are four basic sources of the address gated through the net to memory. They are the stack pointer, the memory address register, an external input address from the I/O controller, and the IWL that selects an index register. The conditions under which each of these inputs are accepted are described in the following paragraphs.

Inputs from the Stack Pointer

Stack pointer output bits SP-1 through SP-4 are routed to four of the five low-order positions of the net. During both cycles I1 and I2, the stack pointer contents are accepted if no external entry is present.

The four-bit output of the stack pointer occupies positions 1-4 of the net, and bit 5 is a fixed one to establish the lowest address as 40. Bit 0 is directly controlled by the I2 signal. During I1, bit 0 is a zero, causing the address to be even; during I2, bit 0 is a one, causing the address to be odd.

During the first half of cycle I1 and cycle I2, the stack pointer input selects memory locations from which instruction addresses are read. During the second half of each of these cycles, the stack pointer input selects the memory location in which an updated instruction address is to be stored. This places the instruction address in the location specified by the stack pointer.



OUTPUT POSITIONS

	PAGE						BYTE								
	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
(a) STACK POINTER INPUT	0	0	0	0	0	0	0	0	1	SP 1-4				0	DURING CYCLE I1
	0	0	0	0	0	0	0	0	1	SP 1-4				1	DURING CYCLE I2
(b) MEMORY ADDRESS REGISTER INPUT	MAR 21 - 213													0	DURING CYCLE I3, FIRST HALF
	MAR 21 - 213													1	DURING CYCLE I4, FIRST HALF
(c) EXTERNAL ADDRESS INPUT	0	0	EX 20 - 211												DURING CYCLE I3, SECOND HALF
	0	0	EX 20 - 211												DURING CYCLE I4, SECOND HALF
(d) INDEX REGISTER ADDRESS INPUT	0	0	0	0	0	0	0	0	0	0	0	OPO-2			DURING CYCLE E2, U-BIT = 0
	IX- SECTION			0	0	0	0	0	0	0	0	OPO-2			DURING CYCLE E2, U-BIT = 1

A-395

Figure 2-15. Output of Memory Address Net

Note that the lowest stack pointer address that can be used is 40 and the highest is 77. These locations are always assigned to be used for stack pointer addresses.

Inputs From the Memory Address Register

A description of how an address is formed in the memory address register was presented earlier in this section. Once this address is formed, it is available to the address net.

During instruction cycles I3 and I4, the objective is to read the instruction from memory. Note that the address was acquired during cycles I1 and I2 and is held in the memory address register. The CL-0 signal allows the MAR inputs to be accepted during the first half of I3 to read the left half of the instruction and during the first half of I4 to read the right half of the instruction. Again, the low order bit is not accepted from the address source; it is left at zero during I3 and switched directly to a one during I4. This operation is shown in figure 2-15(b). It should be noted that the STOP signal prevents the MAR output from being accepted.

External Address Inputs

The second half of cycle I3 and I4 are devoted to reading memory to refresh the CRT display. If an external access request is present, the external address is accepted during the second half of these cycles. The CL-0 signal switches the address net to accept external input address lines EX-20 through EX-211 during this time. Bits 212 and 213 are left at zero. This is shown in figure 2-15(c).

Selection of an Index Register

During instruction cycle E2 of instructions operating in the indexed address mode, the objective is to acquire the contents of an index register. The index register to be addressed is specified by bits 0-2 of the IWL, which are held in the operation register and applied to positions 0-2 of the address net. These addresses may be in any section, and a control bit (U) discussed earlier determines whether the section is 0 or another. During E2, the index register number is gated through positions 0-2 of the net, and if the U bit is set, the section number from the index section register is gated through positions 211-213. This is shown in figure 2-15(d).

Selection of a Data Address

During instruction cycle E3, the purpose of the address net is to select a memory address to acquire data to be operated on or store a result. The source of the address depends upon whether direct or indexed addressing is used.

If indexed addressing is used, a composite address made up of the IWR and the contents of an index register (read during cycle E2) is formed in MAR. This is gated through the address net during E3 to read or store data in memory.

If direct addressing is used, the IWR specifies the byte number within a page. This is held in the MAR. The section number is supplied by the section register. Since the level number is not supplied, the level is effectively zero. Thus, in the direct addressing mode, the address net accepts an address from the MAR and section register to select an address in page 00, 10, 20, 30, 40, 50, 60, or 70.

Memory Data Register.

The memory data register is an 8-bit latch shown on sheet 7 of the processor logic diagram. Its purpose is to hold information read from memory until other processor circuits can accept and act on that information.

During the first half of each machine cycle, information is read from memory and appears at the input of the MDR. Timing signal TE, which occurs midway in each machine cycle, then loads this information in the MDR, as follows:

a. During cycle I1, the IAWL is loaded in the MDR if the BR-DIRECT (branch direct) signal is not active. Thus, for all instructions in which a branch is not to be taken, the MDR holds the IAWL read from memory.

b. During cycle I2, the IAWR is loaded in the MDR.

c. During cycles I3 and I4, the IWL and IWR, respectively, are loaded in the MDR.

d. During cycle E2, the contents of an index register are loaded in the MDR provided that an index register is specified by operation code positions OP 0-2. If no index register is specified, the MDR is left unchanged during cycle E2.

e. During cycle E3, the contents of a memory location are loaded into the MDR for any instruction that requires that data be read from memory for its execution.

B Register.

The B register is an 8-bit latch shown on sheet 7 of the processor logic diagram. It receives the output of the memory output multiplexer and is loaded at TE time of each machine cycle in which the loading conditions are met.

The TE signal occurs at the end of the first half of each machine cycle. This signal is combined with the I3 and I4 signals so as to load the B register at the end of the first half of I3 and again at the end of the first half of I4.

During the first half of cycle I3, the processor reads the IWL from memory. This is loaded into the B register. Thus, the B register holds the IWL at the end of cycle I3.

During the first half of cycle I4, the processor reads the IWR from memory. If the current instruction is a Class 2 or 3 instruction, as indicated by bit OP7 of the operation code being a one, the TE signal loads the IWR into the B register. Note that in Class 2 and 3 instructions the IWR is always a literal or an address.

The contents of the B register are routed to four destinations, as follows:

a. Positions 0-7 are routed to the adder input switch. If the IWR held in the B register is a literal, it is gated to the adder. Note also that positions 1-4 of the IWL hold a jump count that is gated to the adder.

b. Positions 2-7 are routed to the MAR. If the IWR held in the B register is an address and the current instruction is being operated in the indexed address mode, these positions are the page portion of a composite address.

c. Positions 0 and 1 are routed to the adder controls. In the IWR of instructions operating in the indexed address mode, these positions determine whether the contents of the index register are to be incremented, decremented, or left unchanged.

d. Positions 1-4 are the jump count in the IWL of the TLJ and TMJ instructions. They are routed to a jump count decoder in the branch control circuits.

Adder.

The adder appears on two sheets; sheet 5 shows the high-order bits (4-7) and sheet 6 shows the low-order bits (0-3). The adder is made up of two model 74181 arithmetic-logic units (ALU's) which can perform either logical or arithmetic functions, depending upon the mode selected and the function chosen within that mode. Adder controls on sheet 4 select both the mode and function based upon the instruction being executed.

Two eight-bit quantities can be applied to the adder to be acted upon. The selection is made by the adder input switch, shown on sheets 5 and 6, based upon the instruction being executed. The adder input controls also determine whether or not a carry-in signal is to be applied to the adder.

Based upon the inputs mentioned above, the adder forms an 8-bit quantity at its output. Also produced by the adder are the carry-out signal and the A-EQ-B (A equals B) signal.

Adder Controls.

The adder control circuits are shown on sheet 4 of the processor logic diagram. Their purpose is to select the adder mode, select the adder function, and produce gating signals to control the adder input switch.

The adder mode is selected by the A-MODE signal produced by the adder controls. When the arithmetic mode is selected, carries between adder stages are enabled. Conversely, when the logic mode is selected, carries are disabled.

The function that the adder is to perform is selected by a 4-bit function code (AFUN-SO through S3) developed by the adder controls. Based upon the current instruction and cycle number, the adder controls produce a specific function code and either activate the carry-in line or leave it inactive.

Gating signals developed by the adder controls also depend upon the cycle number and instruction being performed. The following gating signals can be produced:

a. MDR-to-ADD B, which gates MDR 0-7 to the B-input of the adder.

b. MDRO-to-ADDA, MDR-12-to-ADDA, MDR-345-to-ADDA, and MDR-67-to-ADDA. These signals can gate selected parts of the MDR to the A-input of the adder.

c. GACC-to-ADDA, which gates the accumulator contents to the A-input of the adder.

d. BO-to-ADDB, B12-to-ADDB, B34-to-ADDB, and B567-to-ADDB. These signals gate selected parts of the B register to the B-input of the adder.

e. EXECUTE-TBO (execute take branch operation).

Output Switch.

The output switch is a set of gates that appear on sheets 5 and 6 of the processor logic diagram. Their purpose is to accept one of four inputs and gate them to the SUM 0-7 lines. In turn the SUM 0-7 lines are routed to four locations:

a. The memory input multiplexer for the purpose of being stored in memory.

b. The MAR for the purpose of becoming part of a memory address.

c. The accumulator on the I/O controller board, where they are loaded into the accumulator.

d. The "D" edge connector on the processor board, where they are available for external use.

As previously mentioned, the output switch receives four inputs:

a. Memory data register (MDR) 0-7. These inputs are gated through the switch by the LITERAL signal.

b. Memory address register (MAR) positions 28-213 and two fixed ones (which replace the unused address bits 214 and 215). The inputs are gated through the switch by the MAR-HIGH signal.

c. MAR positions 20-27. The MAR-LOW signal gates these bits through the switch.

d. Adder output positions 0-7, which are gated through the switch by the SUM-GATE signal.

The next step is to discuss the control circuits that originate the gating signals. They are shown on sheet 5, and are described in the following paragraphs.

Normally, the outputs of the adder are gated through the output switch by the SUM-GATE signal. In addition, the ACC-LD (load accumulator) signal is produced during E3 to load the adder output into the accumulator.

The LITERAL, MAR-LOW, and MAR-HIGH signals are produced under specific conditions that also inhibit the SUM-GATE signal, as follows:

a. During cycle I4 of either a Load Processor Status instruction (155-OP) or a load program key operation (157-OP), the P-LITERAL signal is activated. The load accumulator signal (P-ACC-LD), is also produced. The effect is that the accumulator is loaded with either the processor status word or program key bits that appear in the MDR.

b. Both the MAR inputs and the MDR inputs are gated through the output switch when the Load Instruction Address instruction (22X-LLL) is executed. First, during I4, one of two possible events can happen, depending upon the value of the IWR (LLL). If LLL is 000, the N-0-L (zero literal) signal is low, and signal P-MAR HIGH goes high to gate the MAR high bits into the accumulator via the P-SUM lines. These bits comprise the level and section (page) bits of the IAW (address of the current instruction). If LLL is not 000 (N-0-L is high), the P-LITERAL signal goes high. This gates the MDR bits (i.e., the IWR) into the accumulator. As expected, either case generates the P-ACC-LD (accumulator load) signal.

c. Secondly, during E3, the index register specified by the X in an operation code of 22X is loaded with the low order bits of the IAW, which are held in the MAR. The signal P-MAR-LOW goes high because there is a 22X instruction active during E3. This gates MAR-0 through MAR-7 bits into memory but does not produce a P-ACC-LD pulse.

Memory Input and Output Multiplexers.

These multiplexers are shown on sheet 7 of the processor diagram. The purpose of the input multiplexer is to choose either the information on the SUM-0 through SUM 7 lines from the output switch or the information on a second set of input lines to be stored in memory. The second group of lines is an external input (EX-BIT-0 through -7) and is not used.

The memory input multiplexer is made up of two, four position, 2-to-1 multiplexers. Enabling or disabling the multiplexers is the write signal from the control circuits on sheet 2. This signal is an active low whenever information is to be stored. This occurs when the IAW is restored, when the index register contents are to be stored, when an external input is to be stored, and when any instruction requiring data to be stored in memory is executed.

The selection line to the input multiplexers is the MEM-WRITE-SW signal from an external unit. Since this line is not used, the input multiplexers are always set to accept the input from the SUM-0 through -7 lines.

The memory output multiplexer is also shown on sheet 7 of the processor logic diagram. Two sets of eight lines each are routed to this multiplexer. The first is the memory output. The second set of lines carries one of the following: (1) the output of the program-loading ROM, (2) external data EX-BIT-0 through 7, (3) a status word (consisting of SP1-4 and four status bits), or (4) an 8-bit word produced by the program interlocking plug.

The enabling line to the memory output multiplexer is always enabled. The selection of whether the memory output or the second group of data is accepted is made by a group of control circuits in the lower left corner of sheet 7.

First to be discussed is what information is present on the second group of lines to the output multiplexer. This information originates in three places:

a. The status word/program interlock multiplexer on the left side of sheet 7. The program interlock code passes through this multiplexer unless instruction 155-Load Processor Status is being executed. The selected output then reaches a set of gates. These gates are enabled by the KEY-GATE signal, which is present only during instruction 155 or 157-Load Terminal Identification. Therefore, the output of the status word/program interlock multiplexer is available to the memory output multiplexer only during instructions 155 and 157.

b. The program-loading ROM on the left side of sheet 7. This output is present only during the program loading operation.

c. External input bits EX-BIT-0 through -7, which are not used.

One of three signals can switch the memory output multiplexer to accept the data on the second group of lines, as follows:

a. The KEY-GATE signal, which is active during the 155 and 157 instructions mentioned above.

b. The program loading signal, which allows the program from the ROM to be read.

c. The MEM-WRITE-SW signal, which is not used.

Program Interlock.

The program interlock is an 8-bit plug that occupies position K1 on the processor board. The purpose of this plug is to assign a unique number to the processor in which it is installed.

From the program interlock plug, the 8-bit code is routed to a 2-to-1 multiplexer. This code passes through the multiplexer at all times except when instruction 155, Load Status, gates the status word through the multiplexer and blocks the program interlock code.

The output of the multiplexer is routed to a set of eight gates. Either the instruction 155 or 157, Load Terminal Identification, enables the gates and places the code on the memory I/O lines where it is available to the memory I/O multiplexers.

The purpose of the program interlock code is to allow a program to distinguish this processor from all others. For example a program may allow only specific processors to have access to payroll records. In this case, the program would read the program interlock code to determine if this processor was to be granted access to those records.

Execution of Class 0 Instructions.

Earlier in the description of the processor, each instruction was briefly described and its format shown. The discussion in the following paragraphs outlines the action taken by the processor for each instruction in class 0. Refer to the formats shown in figure 2-13 as required.

The following instructions make up Class 0:

a. TLJ - Test Literal and Jump - 0XX

b. TMJ - Test Mask and Jump - 0XX

c. TLX - Test Literal and Exit - 000

d. TMX - Test Mask and Exit - 040

Provided that the condition for the jump is met, the TLJ and TMJ instructions allow forward or backward jumps over a number of instructions. The jump count, or "offset", is specified by bits 1-4 of the IWL. Since four bits are used to determine the jump count, up to 15 instructions can be jumped. The jump is taken if the value in the accumulator is equal to the literal or mask value given in IWR.

In the TLJ instruction, the byte in the accumulator must be identical to the IWR in order to jump. A test mask operation (TMJ) requires that the byte in the accumulator have "one" bits in the same bit positions as the IWR in order to jump. Note that bit 5 of the IWL differentiates between test-with-literal (TLJ) and test-with-mask (TMJ).

The TLX and TMX instructions (the "exit" versions) differ from the "jump" instructions in that the jump count is always zero in the exit instructions. If the tested condition is met, the TLX or TMX instruction decrements the stack pointer to the level that contains the address of the last branch and stack instruction executed. This address is then read and incremented by two to acquire the address of the next instruction to be executed.

During the class 0 instructions, the processor cycles in the sequence I3-I4-I1-I2. The following action takes place during each of the aforementioned cycles:

a. During cycle I3, the IWL is read from memory and loaded into the operation register, the B register, and the memory data register (MDR). The instruction is then decoded.

b. During cycle I4 the IWR is read from memory and loaded into the MDR, replacing the IWL. Next, the contents of the MDR are gated to adder input B and the accumulator contents are gated to the adder input A. For the TLJ and TLX instructions, the adder is set up for A minus B minus one. If the result equals zero, the condition register is set to "equal" and the take branch flip-flop is set.

To test the contents of the accumulator with a mask in the TMJ and TMX instructions, the adder is set up for $\overline{A} \cdot B$ by the operation decoder. If the accumulator contents have "one" bits in the same positions as the mask, the condition is said to be "equal" and, the branch is taken.

In the case of the TLX and TMX instructions, the jump count is zero, which indicates that the instruction is an exit instruction rather than a jump. This causes the stack pointer to be decremented by one to achieve the exit from this routine. The take branch flip-flop is cleared when the stack pointer is decremented.

c. During cycle I1, the action taken depends upon whether the take branch flip-flop is set and whether the instruction is a jump or an exit, as follows.

(1) Assume first that the instruction is a jump and the take branch flip-flop was not set. In this case, the IAWL is read from the current stack pointer address, updated by two and returned to the same address. Then the updated IAWL is transferred to the MAR to become part of the instruction address. This is the normal process when instructions are executed in sequence.

(2) The second possible case is that the instruction is an exit instruction and that the exit conditions were not met. In this case, the next instruction in sequence is executed, as in case (1) above.

(3) Assume next that the instruction is a jump and that the take branch flip-flop is set. In this case, the next step is to accomplish the jump by modifying the instruction address. The B register output, which contains the jump count in positions 1-4, is gated to the adder. During cycle I1, the IAWL is read from memory and held in the MDR. The MDR contents are then gated to the A-input of the adder, thus bringing the instruction address to the adder.

At this point, both the jump count and the instruction address are available at the adder. If bit 0 of the IWL is low, specifying a jump forward, the adder adds the IAWL to the jump count; if bit 0 of the IWL is high, specifying a jump backward, the adder subtracts the jump count from the IAWL. The modified IAWL is then returned to the stack pointer address and is transferred to the MAR to become part of the address of the next instruction.

If the jump transfers control beyond a page boundary, a carry is produced when the IAWL is modified. During I2, the IAWR is read from memory and incremented to update to the new page number.

(4) The last condition that can result in the execution of a class 0 instruction is that the instruction is an exit instruction and the exit condition was met. In this case, the stack pointer was decremented during cycle I4, and the take branch flip-flop was left in the cleared condition. Under these conditions, the IAWL is read from the new stack pointer address and sequential instruction execution resumes at that point.

d. Cycle I2 may not be required by class 0 instructions. If it is not, the cycle counter skips to cycle I3 and reads the next instruction from the address in the MAR. However, cycle I2 is required under three conditions:

(1) The jump or exit conditions were not met but the update of the IAWL produced a carry. In this case, the IAWR is read, stepped by one, loaded into the MAR, and also restored in the current stack pointer location. This is the normal sequence when no branch takes place.

(2) The exit condition was met and when the IAWL from the new stack pointer location was updated it produced a carry. In this case the same action is taken as in (1) above except that the IAWR is returned to the new stack pointer address.

(3) The jump condition was met and when the jump count modified the current IAWL, it produced a carry. In this case, the IAWR is handled the same as (1) above.

Execution of Class 1 Instructions.

The instructions in class 1 include the branch, stack and branch, exit, memory control, and interrupt control groups. In the following paragraphs, the instructions are discussed in groups, because of the similarities between instructions within each group. Branch instructions are discussed first. Refer to the instruction formats in figure 2-13 as required during the following descriptions.

Branch instructions BRU, BRH, and BRL specify the address of the next instruction to be executed, thereby allowing the processor to deviate from the normal sequential execution of the instructions. This address replaces the current address in the current stack pointer location.

Stack-and-Branch instructions SBU, SBE, SBH, and SBL are usually used to branch to subroutines. In these instructions, the branch address is stored in the next higher-level position in the instruction address stack. This preserves the address at which the stack-and-branch instruction was encountered in the current stack pointer location and thereby provides a link back to the main program.

Exit instructions EXB and EXU decrement the stack pointer in order to return to the point in the program at which the last stack and branch instruction was performed. Once the stack pointer is decremented, the EXB and EXU instructions take different action. The EXU instruction leaves the current contents of the new stack pointer location unchanged, so the next instruction address is read from this location. The EXB instruction, however, replaces the current contents of the new stack pointer location with the branch address. Thus, when the next instruction address is read, it will be the branch address.

In all branch instructions, the IWL, read during cycle I3, contains the level number portion of the page number in positions 0 through 2. (How the section portion is acquired is discussed later.) The IWR, read from memory during I4, gives the branch location within the page. Since a branch, if taken, is always to an even location in memory, the least significant bit of the IWR is not needed for addressing and can be used to specify the branch condition (unconditional or on equal; on high or on low).

An unconditional branch sets the TK-BR-F/F (take branch flip-flop) during cycle I4. A conditional branch does so only if the condition register was set to meet the condition by a previous compare operation. A stack-and-branch operation also increments the stack pointer counter, which directs the processor to a higher level in the instruction address stack.

During cycle I1, the TK-BR-F/F prevents the normal transfer of the IAWL to the MDR. Thus, the MDR still holds the IWR of the current instruction. This is the byte location portion of the branch address. It is gated through the A inputs of the adder and into the MAR. In this manner, the byte portion of the branch address is substituted for the IAWL normally read during cycle I1.

During cycle I2, the IAWR is read and loaded in the MDR. Position 3-5 of the MDR, which hold the section number, are gated to the corresponding B-inputs of the adder and the B register positions 0-2, which hold bits 0-2 of the IWL (the level part of the page number), are gated A-inputs. The adder outputs then yield the combination of the current section and the new level. This combination is stored in the instruction address stack level indicated by the stack pointer. It is also loaded into the high-order half of the MAR so that upon termination of this cycle the MAR contains the complete branch address.

Note that the action described in the preceeding paragraph automatically limits branching to within the current section because it uses the current section number from the IAWR. Thus, a stack-and-branch or branch operation is restricted to one section. For out-of-section branching, the instruction must be preceeded by a Set Memory Section instruction. This causes the section number set in the section register to replace the section portion of the IAWR and thus allows a branch to any section.

An Exit Unconditional instruction decrements the stack pointer by one during TX-time of I4. At this point, the preceding level instruction address stack location still contains the address of the Stack-and-Branch instruction that caused the stack pointer to be incremented last. The MAR is updated with this address (after 2 has been added to the IAWL) during I1 and I2.

The Exit-and-Branch instructions combine the functions of the Exit Unconditional instruction and the Branch Unconditional instruction. After the stack pointer is decremented, both the MAR and the new instruction address stack location are updated with the branch address specified, as described earlier.

Next to be discussed among the Class 1 instructions are the memory control instructions SMS, SMC, and SSC. The SMS and SMC instructions are related to out-of-section branching and the use of index registers in sections other than section 0. Note that the SSC instruction, Set Memory Section and Control, simply combines the functions of the SMS and SMC.

Since branch instructions themselves do not specify the section within the branch address, a Set Memory Section instruction must precede a branch instruction if the branch is meant to transfer control to a different section. During the SMS instruction, the processor cycles in the sequence I3, I4, I1, I2. The IWL is read from memory during I3 and loaded into the operation register, the B register, and the MDR. As a result, the operation decoder produces the signals SECTION-OP and 15X-OP. The IWR, which is read from memory and loaded into the MDR during I4, gives the section portion of a branch address. The corresponding MDR outputs (3-5) are connected to the section register, and at TX-time they are loaded into the section register.

The section register outputs (SECTION 0, 1, 2) are gated to adder inputs A3 through A5 by the signal P-SECTION-GATE. The inverted signal (N-SECTION-GATE) prevents the MDR-3 through 5 from reaching the adder by forcing enable signal P-MDR-345-ADDA low. As a result, when the address is formed in the MAR and the instruction address word-right (IAWR) rewritten into the instruction address stack, the section number in the section register replaces the current section number.

The Set Memory Control instruction controls the setting of the U and V bits. (Refer to the description of the use of the U and V bits earlier in this section as required to review the function of these bits.) Note in figure 2-13 that positions 6 and 7 of the IWR of the SMC instruction are the set and reset controls for the U and V bits. At T-SUM time of cycle I4 of the SMC and SSC instructions, bits 6 and 7 of the MDR are transferred into the U/V register.

The SAC instruction sets the condition register to an equal, high, or low condition, depending upon the state of bits 4 and 5 of the accumulator at the time the SAC instruction is executed. It is usually used to restore the condition register after returning to the main program from a subroutine. After this instruction is read from memory during cycle I3, it produces the 153-OP signal at the operation decoder, which gates accumulator outputs 4 and 5 to the condition register inputs. During I4, the condition register is set in accordance with the accumulator bits. (The instruction description given earlier in this section defines the relationship between the accumulator bits and the setting of the condition register.)

The LSW instruction is used only if the machine is equipped with the eight optional sense switches. In response to this instruction, the operation decoder produces the SENSE-1 signal and sends it to the I/O controller. The state of the sense switches is then loaded into the accumulator. In machines with the memory board that employs parity checking, the operation code 154 is the code for the Enable Parity instruction.

The LPS instruction loads the status of the processor into the accumulator. This instruction produces the signal P-155-OP, which, in turn, causes the signal N-KEY-GATE to become active during I4. The 155-OP signal gates the status word through the status word multiplexers, and the KEY-GATE signal gates the processor status word through the memory output multiplexer from where it is loaded into the MDR. Next, the MDR output is gated through the output switch by P-LITERAL signal, which is generated by the N-KEY-GATE signal. Also generated by the N-KEY-GATE signal is the accumulator load signal P-ACC-LD, which loads the status byte into the accumulator at TX-time of cycle I4.

The processor status byte is organized as follows:

a. Bits 0 through 3 are stack pointer bits P-SP-1 through P-SP-4, respectively.

b. Bit 4 and 5 are the P-ACC>MDR and P-ACC=MDR signals from the condition register, respectively.

c. Bit 6 is the interrupt overflow indicator (P-INT-OVFL) condition.

d. Bit 7 is the program interrupt switch (P-PRG-INT-SW) position.

Note that bits 4 and 5 of the status word are the same positions of the accumulator that are used during the SAC instruction to set the condition register. Thus, the program can execute instructions to read and store the processor status word before starting a subroutine. After the subroutine is completed, the status word can be read from memory into the accumulator. A SAC instruction will then restore the condition register to the value it held before the subroutine was entered.

Also included in Class 1 are the three instructions dealing with processor interrupts: Disable Processor Interrupt, Enable Processor Interrupt, and Clear Processor Interrupt. Refer to the description of the use of interrupts earlier in this section as required while reading the following instruction description.

Notice in figure 2-13 that all three instructions are identical except for bits 0 and 1 of the IWR. All three instructions produce the 156-OP signal at the output of the operation decoder. During TX-time of I4, this signal examines the state of bits 0 and 1 of the MDR (the IWR) and produces the following action:

a. Disable Processor Interrupt (MDR 0 and 1 are 00) - set the LOCKOUT flip-flop. The LOCKOUT-F/F prevents the AUTO-STK & BR-F/F from setting in response to an interrupt.

b. Enable Processor Interrupt (MDR0 is set and MDR1 is clear) - clear the LOCKOUT-F/F.

c. Clear Processor Interrupt (MDR0 is clear and MDR1 is set) - generate a signal that resets the INT-OVRFL-F/F at TX-time of I4.

Execution of Class 2 Instructions.

The Class 2 instruction set includes data transfer and arithmetic instructions LDA, LDX, LIA, STA, ADA, ADX, SUA, and SUX. (The format of all class 2 instructions is shown in figure 2-13.) Note that immediate, direct, or indexed addressing modes can be used with the LDA, STA, ADA, and SUA instructions.

The three addressing modes were discussed earlier in this section; however, a brief review is presented below:

a. In immediate addressing, the right half of the instruction (IWR) is the data to be used by the instruction. When bit 3 of the IWL is a zero, it indicates that immediate addressing is to be used. This activates the DIRECT signal so that class 2 instructions operating in the immediate address mode cause the processor to sequence through the cycles I3, I4, E3, I1, and I2.

b. The direct address mode requires that the instruction acquire data from or that it store data in the memory location specified by the address held in the right half of the instruction (IWR). The IWR is recognized as a direct address if bit 3 of the IWL is a one and no index register is specified by bits 0-2 of the IWL. Class 2 instructions operating in the direct address mode activate the INDIRECT signal. This causes cycles I3, I4, E2, E3, I1, and I2 to be used by Class 2 instructions operating in the direct address mode. The IWL is read during I3, the IWR during I4, and the IWR is used to address memory during E3.

c. The indexed addressing mode provides a method of addressing data stored anywhere within memory. An indexed address is composed of a byte address contained in a specified index register plus a page number contained in the IWR. The high-order six bits of the IWR specify the page within memory and the index register specifies the location within that page.

Since any page can be addressed by six bits, two bits of the IWR (0 and 1) are used to indicate whether the value in the index register must be incremented, decremented, or left unchanged when the instruction is executed, as follows:

IWR-1	IWR-0	Indicates
0	0	Leave index register unchanged
1	0	Increment index register
1	1	Decrement index register

In the indexed address mode, bit OP3 again activates the INDIRECT signal to force the processor to go through instruction cycles I3, I4, E2, E3, I1, and I2. After the IWL and the IWR have been read from memory during I3 and I4, respectively, the index register specified in the three low order bits of the IWL is addressed and its contents loaded into the MDR during E2.

The index register contents and the IWR (still held in the B register) then provide the full memory address. These values are loaded into the MAR; the MDR contents are loaded into the low order half and B register bits 2-7 into the high-order half.

In addition, the contents of the index register, held in the MDR, are returned to memory either incremented, decremented, or unchanged. For that purpose, the MDR output is gated to adder input A. Normally the adder is set up for A plus B. With nothing applied to input B, the value at input A would appear at the output, which is the case if the register value is to remain unchanged. If the value is to be incremented by one, a carry is forced into the low-order half of the adder. If the index register is to be decremented, the adder is set up for A minus B minus 1. Note that the two low order bits in the B register (the IWR) indicate the action to be taken.

Now that the addressing modes have been reviewed, the instructions in class 2 are discussed. No mention of the addressing modes is made in the following discussion unless an instruction is limited in the modes that it can use. The load instructions are discussed first.

There are three load instructions in class 2: LDA - Load Accumulator, LDX - Load Index Register, and LIA - Load Instruction Address. The Load Accumulator instruction is discussed first. At TE-time of I3, the IWL read from memory is loaded into the operation register, the B register, and the MDR. After the address mode is selected and the required action taken, the value to be loaded into the accumulator is held in the MDR. The MDR contents are gated through the B inputs of the adder during E3 by the signal P-MDR-ADDB. In addition, the signal P-ACC-LD is sent to the I/O controller during E3 to load the adder output into the accumulator.

The Load Index Register instruction uses only the immediate addressing mode. When this instruction is read from memory and decoded, the N-INDEX-0 signal goes inactive and P-LOGIC-DIRECT signal becomes active. At E3, the latter signal gates operation register lines 0 through 2 to the address net to select one of seven index registers. At the same time, the B register (which contains the literal) is gated into the adder and selected to reach to the memory input multiplexer by P-SUM-GATE signal. Then the N-MEM-WRITE-SW signal loads the literal into the index register specified during the second half of cycle E3.

The LIA (Load Instruction Address) instruction is operated only in the immediate address mode. Its purpose is to place the IAWL in an index register and the IAWR or the literal in the accumulator. During I4, one of two things can happen, depending upon the value of the IWR. If the IWR (the literal) is 000, the N-0-L (zero literal) signal is low, and signal P-MAR HIGH goes high to gate the MAR high-order bits into the accumulator via the P-SUM lines of the output switch. These bits comprise the page and section bits of the IAW (address of the current instruction). If the literal is not 000 (N-0-L is high), the P-LITERAL signal goes high. This gates the MDR bits (the IWR) into the accumulator. As expected, either case generates the P-ACC-LD (accumulator load) signal.

Secondly, during E3, the index register specified by the X in an operation code of 22X is loaded with the low order bits of the IAW, which are held in the MAR. The signal P-MAR-LOW goes high at the output switch because there is a 22X instruction active during E3. This gates MAR-0 through MAR-7 bits onto the sum 0-7 lines and into memory.

The Store Accumulator instruction can be operated in either the direct or indexed address mode. Once the address is formed in the MAR, the processor moves to cycle E3. During E3, the signal P-GACC-ADDA gates the accumulator contents through the adder. This byte then appears on the sum 0-7 lines to the memory input multiplexer and is stored in the location specified by the MAR.

The add and subtract instructions are discussed next. There are four of these instructions in class 2:

a. ADA - Add to Accumulator

b. ADX - Add to Index Register

c. SUA - Subtract from Accumulator

d. SUX - Subtract from Index Register

In the add to and subtract from accumulator instructions, the current accumulator contents are connected to the adder input A. The MDR is connected to input B. A value read from memory into the MDR is subtracted from or added to the value in the accumulator and the result is loaded into the accumulator at TX-time of cycle E3.

The ADX and SUX instruction operate only in the immediate address mode. The contents of the index register are sent to the adder A inputs (via the MDR) and the B Register contents (literal) are sent to the adder B inputs to be added or subtracted during E3. The result is loaded into the index register during the write memory cycle of E3.

Execution of Class 3 Instructions.

The class 3 instruction set includes the Boolean, shift, and compare functions. The immediate, direct, or indexed addressing modes can be used as indicated in the instruction formats shown in figure 2-13. No mention of addressing mode is made in the following description unless a specific instruction is limited in the modes available.

The Boolean instructions are ANA, ERA, and IRA. At TE-time of I3, the IWL is read from memory and loaded into the operation register, the B register, and the MDR. At TE-time of I4, the IWR is loaded into the MDR and the B register. If this is a literal, it is used directly; otherwise, a memory address is formed and the contents of that location are read and loaded into the MDR. Thus, regardless of addressing mode, the quantity in the MDR is to be used in a Boolean function with the quantity in the accumulator.

With no shift count specified in bits 0-2 of the operation register, the MDR contents are gated to the B inputs of the adder during E3 by the signal P-MDR-ADDB. The accumulator contents are gated to the adder A inputs by the signal P-GACC-ADDA. The selected Boolean function is then performed. Following execution, the adder sends the result back to the accumulator.

The shift instructions are each combined with a Boolean function but can only be operated in the immediate address mode. The IWL and IWR are acquired normally during I3 and I4, respectively. The shift count in bits 0-2 of the IWL causes the accumulator contents to be shifted the required number of positions during I3 and I4. Then, during E3, the Boolean function specified is performed in the same manner as described above.

There are two compare instructions: CPA, Compare Accumulator, and CPX, Compare Index Register. The index register is compared to the literal in the IWR: however, the accumulator can be compared to a literal or the contents of a memory location.

For the Compare Accumulator instruction, the contents of the MDR are gated to adder B inputs and the accumulator contents to adder A inputs. During E3, the adder is set up for the A minus B minus 1 operation. If the contents of the MDR and the accumulator are equal, the adder yields zero at its outputs. The signal P-A-EQ-B then becomes high. At TSUM-time, this signal sets the condition register and the signal P-ACC-MDR goes high.

If inputs A and B are not equal, the signal P-C-OUT (the carry output of the high-order part of the adder) indicates whether the value in the accumulator is greater or less than the value in the MDR. If the P-C-OUT signal is high, A (accumulator) is greater than or equal to B; if the P-C-OUT signal is low, A (accumulator) is less than B. At T-SUM time, the results are entered in the condition register.

When a Compare Index Register instruction is executed, an index register contents are to be compared with a literal. The B register (loaded with the literal during I4) is connected to adder inputs B, and the MDR output (loaded with the contents of the selected index register during E3) is gated to adder inputs A. The comparison is made as described above and the condition register is set according to the result of the comparison.

2-3. 8,192-BYTE MEMORY.

The earlier models of the 1501 have a memory board with a capacity of 8,192 bytes. This memory board consists of the memory array of integrated circuits ("chips") which each store 1024 bits, as well as supporting circuits such as the address decoder, output register, etc.

The memory board is mounted on the processor board and plugs into a socket connected to the processor circuits. A block diagram of the memory appears in figure 2-16.

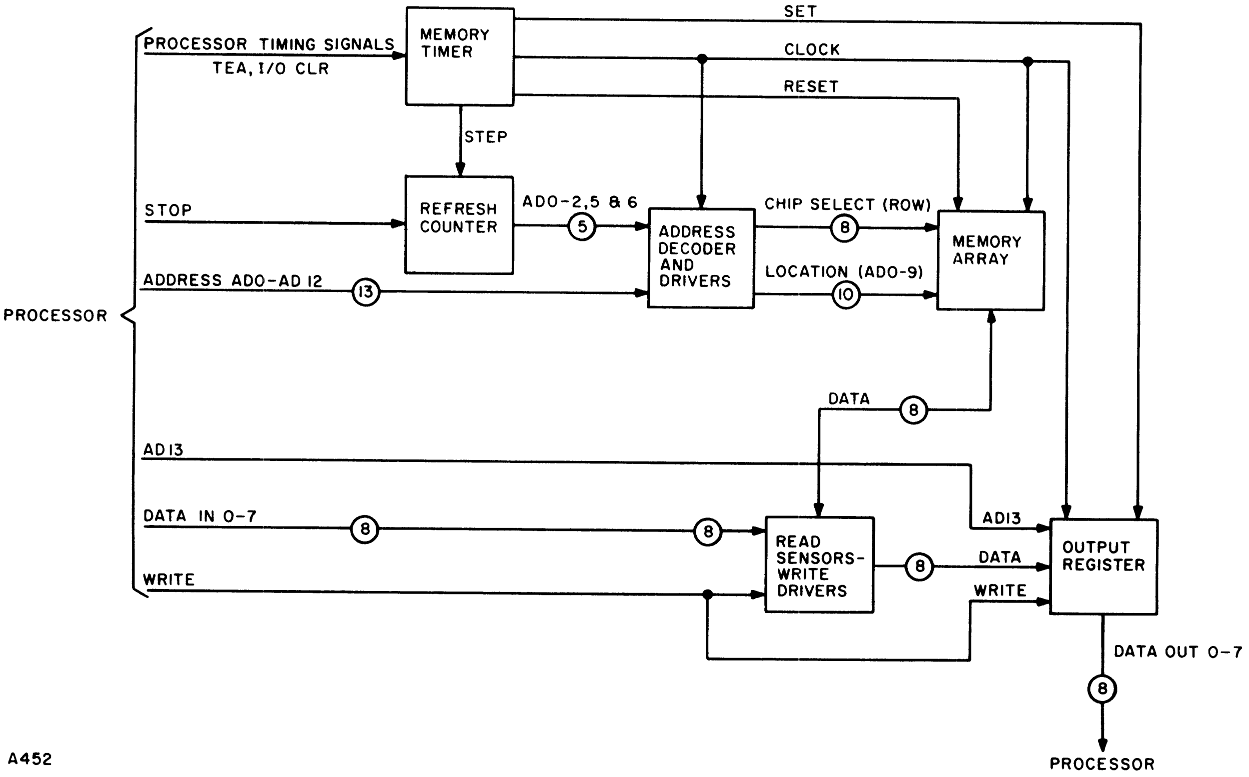


Figure 2-16. 8,192-Byte Memory Block Diagram

Selection of the Memory Address.

The organization of the 16-bit memory address available at the output of the processor address net is shown in figure 2-17. Bits 13, 14, and 15 are not used, which reduces the address available to 13 bits (AD-0 through AD-12). Bits AD-0 through AD-12 are sufficient to address the 8,192 locations on the board. The bit AD-13 input to the memory board is wired so that the board is always selected.

Memory Board.

A memory board is shown in logic diagram 2244. A total of 64 chips arranged in an 8-by-8 matrix provide the storage capacity of 8,192 bytes. The layout of the matrix and the addressing scheme used is shown in figure 2-18.

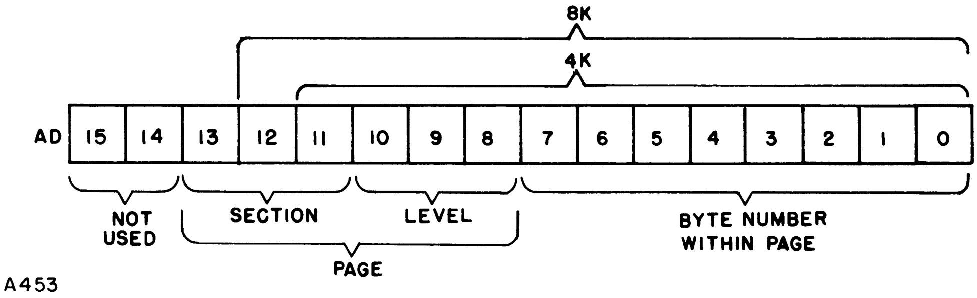


Figure 2-17. Address Applied to 8,192-Byte Memory

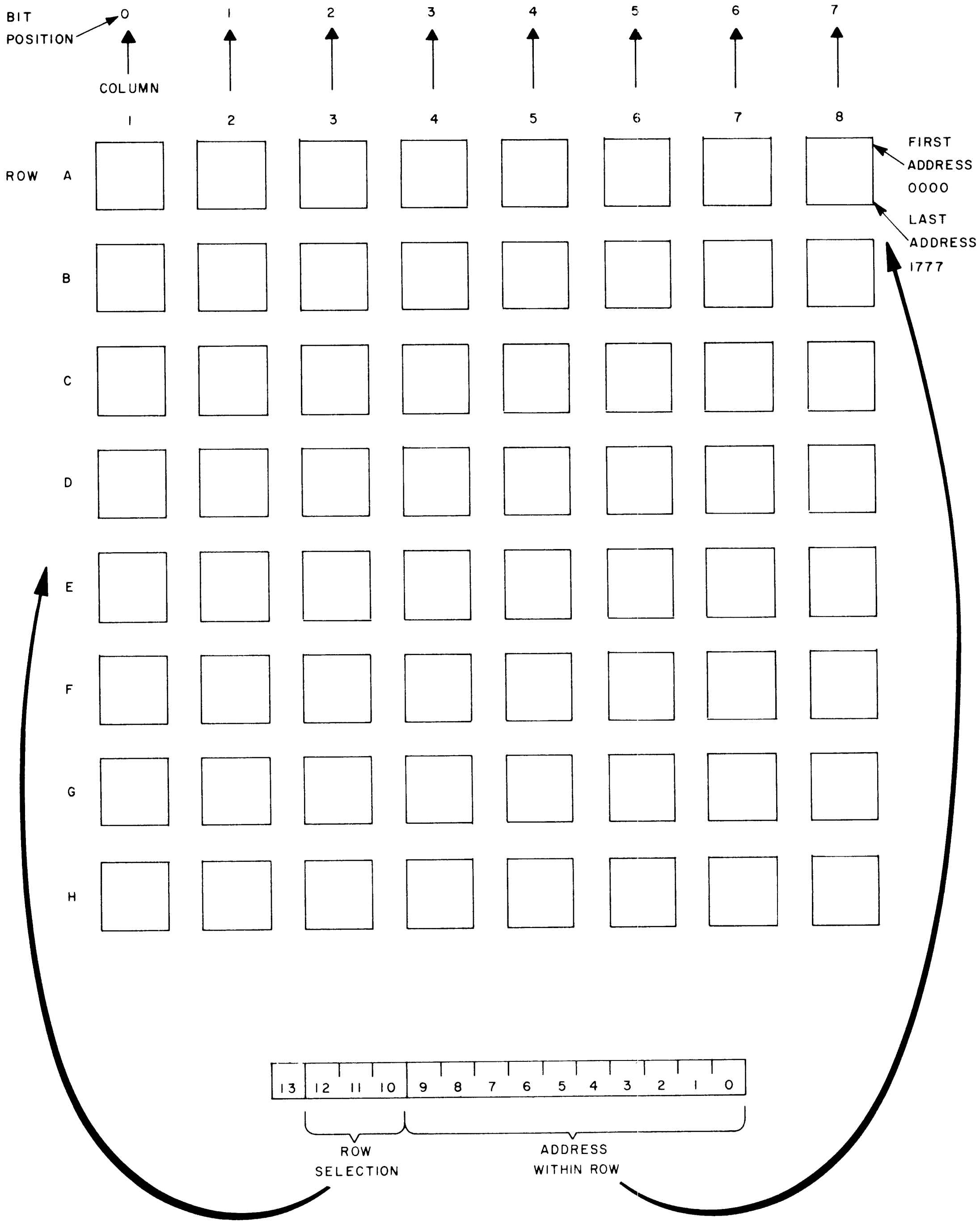
Note that each chip stores 1024 bits and therefore, has 1024 addresses. The chips are arranged in columns so that the outputs of all 8 chips in column 1 are connected in parallel to provide bit 0 of the output, all the chips in column 2 provide bit 1, etc. Address bits AD-10 through AD-12 are decoded to select a row; then, bits AD-0 through AD-9 select one of the 1024 addresses in that row.

Data to be entered into memory is routed to a bank of eight write driver-read sensor circuits. When the WRITE signal is active, the data is stored in the selected memory address.

Data read from memory is gated through the read sensors when the write signal is inactive. The data is then routed to a memory output register. Whenever the WRITE signal is low, the output register can be loaded with the byte available from the read sensors. The SET signal from the timer loads the output register.

Even though the byte is present in the output register, it is not gated to the common DATA-OUT lines until the AD-13 line selects this memory board. Since the AD-13 line is wired so as to always enable the board, the clock signal from the timer provides the gating signal necessary to apply the data to the output lines.

The characteristics of the memory chips used are such that the stored data must be periodically refreshed. Because of the way in which the array is organized, only five address lines (A0, A1, A2, A5, and A6) need to be activated to refresh the memory. These addresses are generated by a five-stage refresh counter and gated to the memory, replacing AD0, 1, 2, 5, and 6 from the address net whenever the N-STOP signal from the processor is low. This occurs only when the processor is being operated in the single cycle mode. During all other times, memory access by the processor and I/O controller effectively accomplish the refreshing required.



A449

Figure 2-18. Organization of the 8,192-Byte Memory Array

Memory Timing.

Timing of the memory operations is controlled by a group of four one-shot multivibrators. A timing diagram of the inputs and outputs of the timer are shown in figure 2-19, and the circuits are shown in logic diagram 2244.

The memory timer is synchronized with a machine cycle of the processor. Each machine cycle is one-microsecond in duration, and several timing pulses are produced by the processor during each machine cycle. (Refer to the processor timing description in this section if a review is necessary.)

Two of the timing pulses produced by the processor during each machine cycle are TEA, which occurs midway in the cycle, and I/O clear, which occurs at the end of the cycle. These are the two signals that start the memory timer. Thus, the memory timer runs through its sequence twice during each machine cycle, once during the first half and once during the second half. This means that the processor can gain access to memory twice during each machine cycle.

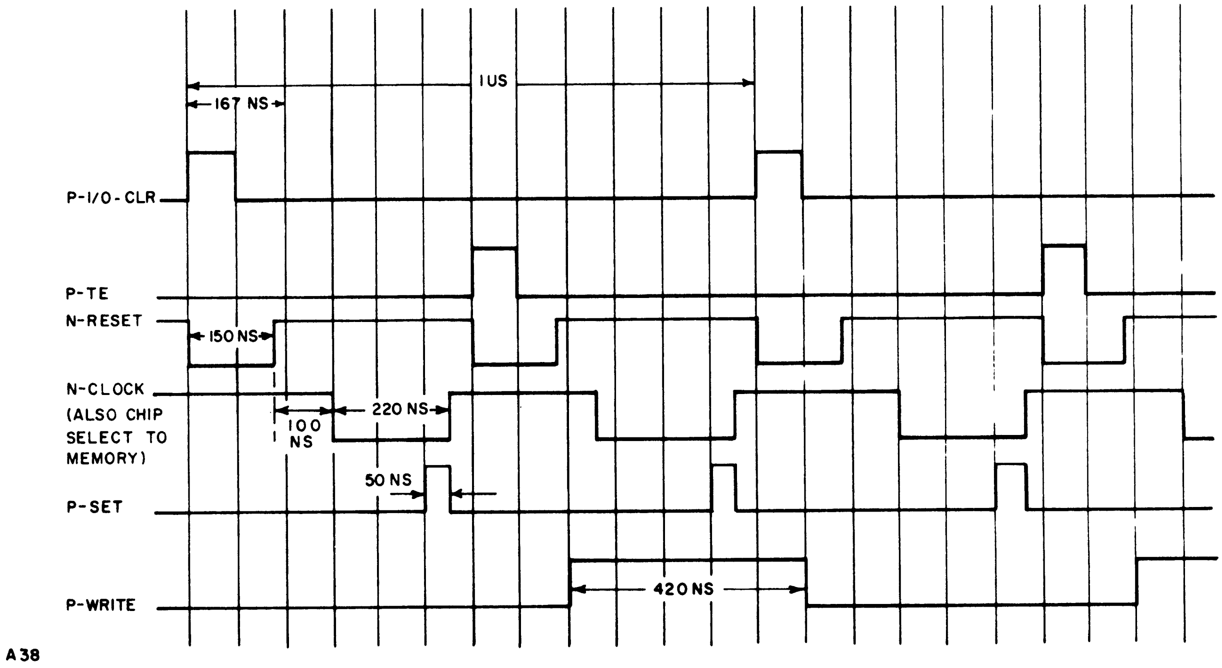


Figure 2-19. 8,192-Byte Memory Timing Diagram

The I/O clear signal at the end of a machine cycle initiates the memory timer sequence required for the processor to gain access to memory during the first half of the next machine cycle. Then, the TEA signal triggers the sequence required for memory access during the second half of the machine cycle.

The first signal produced by the memory timer is labeled P-RESET. This signal, which is not shown on the timing diagram, is a short pulse formed by both the TEA and I/O CLR signals. It steps the refresh counter to the next address and triggers the reset one-shot. In turn, the reset one-shot forms the 150-nanosecond reset signal and applies it to memory. When low, this signal prepares the memory for operation.

The clock, chip select, and set signals are also formed by the memory timer. When the reset signal ends (returns to a high level), the clock and chip select signals allow the memory to decode the address applied to it and perform the read operation. The set signal then transfers the byte read from memory into the output register.

The preceding description illustrated that two reading operations can be conducted during each machine cycle. Since the readout of memory is nondestructive, it is not necessary to conduct a writing operation. However, there are some instructions that require information to be stored in memory, and the writing is controlled by the write signal from the processor. This signal is activated only during the second half of a machine cycle that requires information to be stored. It switches the write drivers/read sensors to the write mode, and when the clock and chip select signals to the memory are active, the DATA-IN to be stored is written in the selected address. Because the write signal is active, no change can be made in the output register; thus the SET signal has no effect during writing operations.

2-4. 16,384-BYTE MEMORY WITH PARITY.

Currently used in the Model 1501 is a memory board with a maximum capacity of 16,384 bytes and parity checking capability. This board is mounted on the processor board and plugs into a socket connected to the processor circuits. A block diagram of the memory appears in figure 2-20, and the logic diagram number is 3079.

The memory array is made up of integrated circuits ("chips") that each store 4096 bits (4096 by 1). These circuits are mounted on a printed circuit board along with the supporting circuits such as an address decoder, output register, etc.

Selection of the Memory Address.

A brief review of the memory addressing scheme used in the 1501 is required before the memory is described. Shown in figure 2-21 is the organization of the 16-bit memory address available at the output of the processor address net. Bits 14 and 15 are not used, which reduces the address available 14 bits. These bits are sufficient to address the 16,384 locations.

Organization of the Memory Array.

As previously mentioned, each chip used in the memory array stores 4096 bits and has 4096 addresses. The chips are arranged in 9 columns and 4 rows as shown in figure 2-22. In each column, data inputs and outputs of each chip are connected in parallel with one another so that each column is related to one bit position. Column 2, for example, holds bit 2 for all 16,384 addresses.

Address bits 12 and 13 are decoded to select one of the four rows. This is accomplished by elements 107 and 112 on sheet 1 of logic diagram 3079. A chip select signal is then applied to all nine chips in the selected row.

Six address bits (those not involved in memory refreshing operations) are applied directly to all four rows. Six other address bits and a chip enable signal, however, are applied to only the selected row. In the row receiving the chip enable signal, the 12 address bits combine to select one of the 4096 addresses.

Data Input.

Data to be stored in the memory is supplied by the processor on the DATA-IN-0 through -7 lines. Each bit is then routed to all chips in one column. For example, bit 1 is routed to all chips in column 1, bit 2 is routed to all chips in column 2, etc. When the address and write signals are activated, the data is stored in the selected address.

The input data is also applied to the parity generator, logic element 125 on sheet 2 of the logic diagram. In turn, the parity generator supplies either the 1 or 0 required to provide odd parity for the nine bits stored. The parity bit is routed to all chips in the parity bit column, and stored at the same time and in the same address as the data.

Data Output.

Data read from the memory is gated into the output register, which is made up of elements 119, 122, and 123 shown on sheet 1 of the logic diagram. An output of the timing circuits gates the data into the register, but the transfer is allowed only during a read operation. Either a refresh operation or a write operation blocks the transfer.

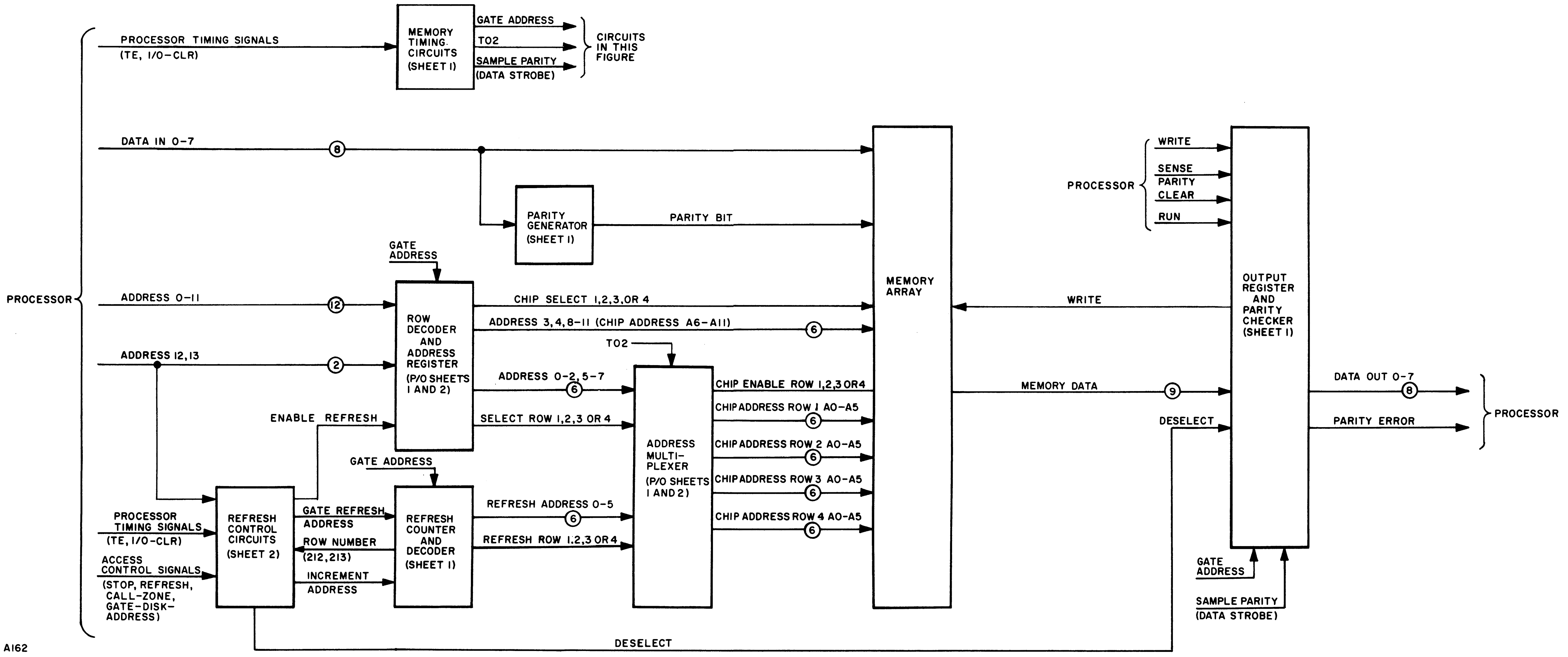


Figure 2-20. 16,384-Byte Memory Block Diagram

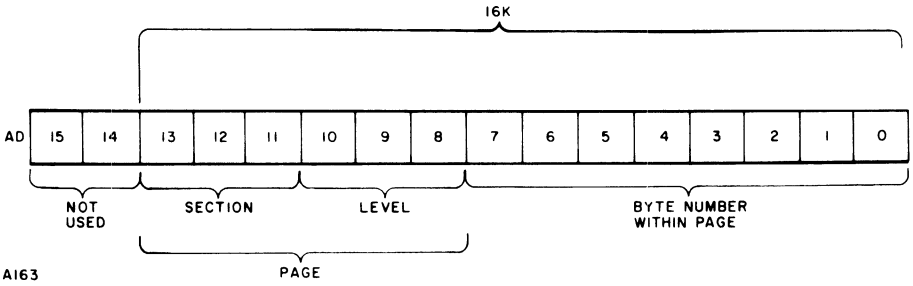


Figure 2-21. Organization of 16,384-Byte Memory Address

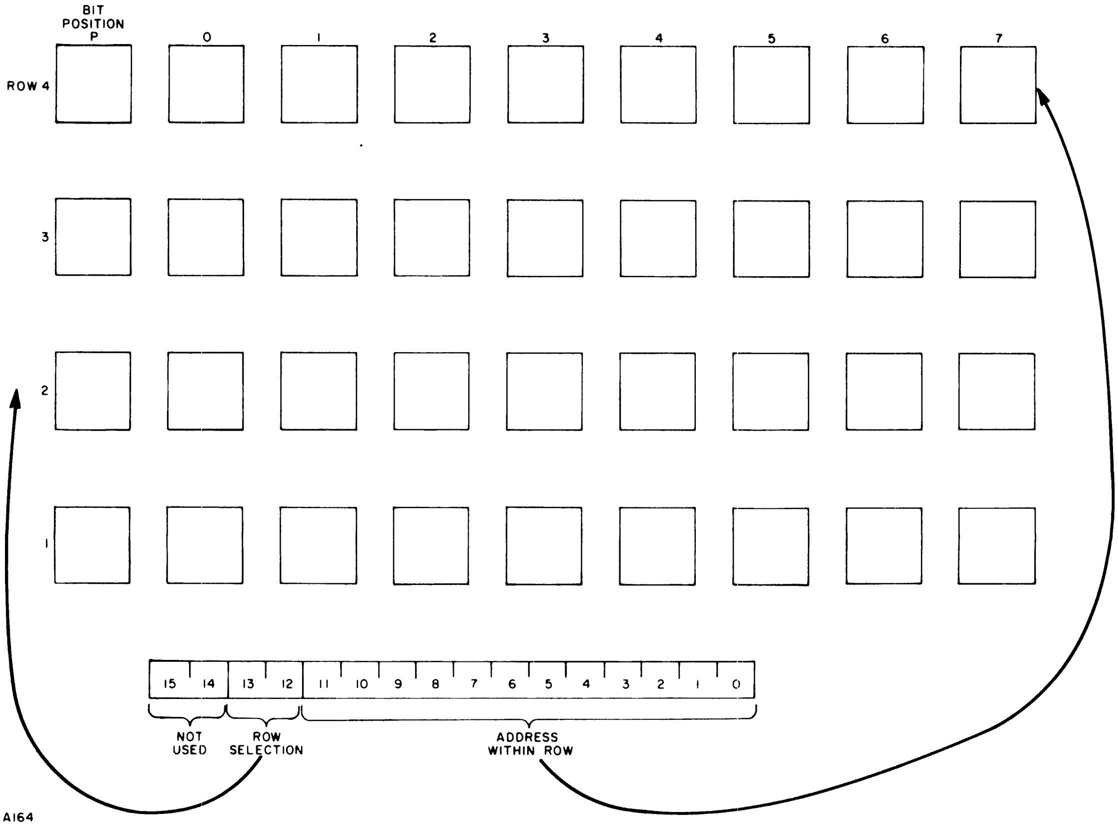


Figure 2-22. Organization of the 16,384-Byte Memory Array

After the data has been entered into the output register, the contents are applied to the parity checker (element 124) along with the parity bit for that data byte. If the total number of ones in the 8-bit byte plus the parity bit is odd, no output is emitted from the parity checker and the data transfer takes place normally. Detection of an error, however, sets the parity error flip-flop (element 130), which sends a parity error signal to the processor.

Address Multiplexers.

Gating of the address to the memory chips is somewhat complicated by the need to refresh the memory by cycling through specific memory addresses. Address bits 3, 4, 8, 9, 10, and 11 are not involved in the cycling process. Therefore, they are loaded into an address register and applied to the memory array by the GATE-ADD signal. The address register is shown on sheet 2 of the logic diagram.

Address bits 12 and 13 are decoded to select a row. They supply the chip select signal and the chip enable signal to the selected row. Note that the refresh counter can also produce address bits 12 and 13 to select a row.

The remaining bits of the address (0, 1, 2, 5, 6, and 7), however, can originate either at the refresh counter or at the input from the processor. The circuits that select one or the other of these inputs are the address multiplexers.

Memory Timing.

Timing signals to control memory operations are developed by a timing chain (elements 105, 106, 113, and 117) shown in the upper left section of sheet 1 of the logic diagram. This chain of one-shot multivibrators produces several of the timing signals shown in figure 2-23. The signals TE, I/O-CLR, and WRITE are provided by the processor.

An internal oscillator that starts when power is applied generates the initial timing pulses. They are followed by the TE and I/O-CLR signals, which are generated by the processor during each machine cycle and synchronize the memory with the processor.

The TE and I/O-CLR signals are combined to produce the trigger signal (TRIG) that starts the timing chain. Each one-shot produces a timing signal and turns on the next element in the chain. The function of each timing signal is as follows:

a. GATE-ADD (gate address) gates both the chip select signal and address bits 0-2 and 5-7 to the selected row; address bits 3, 4, and 8-11 to all rows.

b. T02 gates the chip enable signal to the selected row.

c. DATA-STROBE, which is developed by element 117 and is also labeled SAMPLE-PARITY, gates data read from memory into the output register and sets the parity error flip-flop if an error exists.

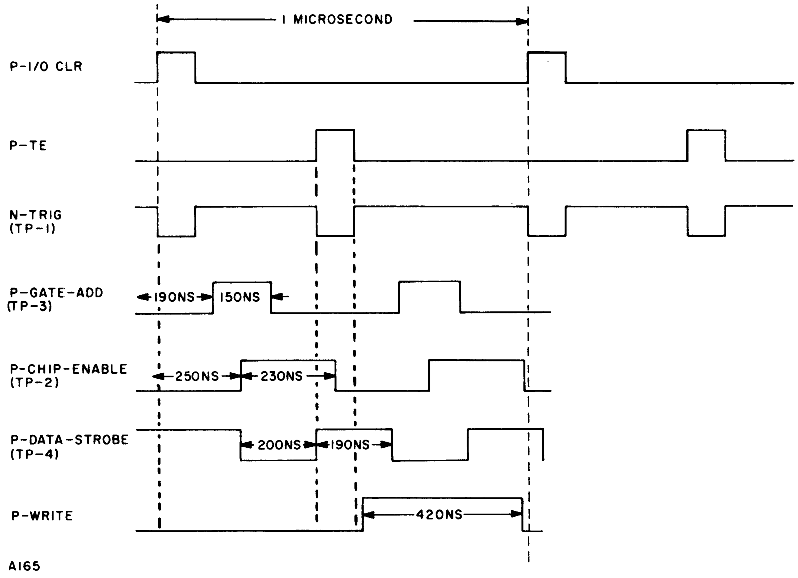


Figure 2-23. 16,384-Byte Memory Timing Diagram

Refresh Counter and Decoder.

The characteristics of the memory chips used are such that stored data must be periodically refreshed. This is accomplished by cycling through the six low-order address lines on the memory chip. Since the maximum refresh time is 2 milliseconds, each address must be activated at least once within this interval.

Shown on sheet 1 is a refresh address counter (elements 103 and 108), a refresh row decoder (element 109) and a refresh control flip-flop (element 107A). The six low-order positions of the counter are applied to the address multiplexers. There, they can substitute for the address (0-2 and 5-7) supplied by the processor.

The two high-order stages of the refresh address counter select the row to which the 6-bit refresh address is applied. A decoder monitors these two stages and activates a.signal (RFSH-ROW-1, -2, -3, or -4) depending upon the contents of the counter. This signal switches one of the address multiplexers to substitute the refresh address for the processor address input and thus refresh the chips in that row.

In turn, the decoder is controlled by the refresh control flip-flop. If refreshing is to be accomplished, the flip-flop is cleared by the GATE-ADD signal. This allows the decoder to operate as described above. If refreshing is to be disabled, the control flip-flop is set by the GATE-ADD signal. This disables the decoder and thus prevents the refresh address from reaching the memory.

Also involved in the refresh operation is the EBLE-RFSH signal from the refresh control circuits. When this signal is a high, it allows the row decoder (which decodes processor address bits 12 and 13) to operate normally. When the BLE-RFSH signal is in its active low state, it disables the row decoder and drives all four outputs high. When the GATE-ADD pulse occurs, the decoder output is transferred to the chip select register. The effect is to set all four stages of the register and thus supply a chip-select signal to all four rows.

Once a row is selected, the refresh counter steps from 0 to 64. When the count of 64 is reached, the next row is selected and the address returns to zero. The method of refreshing a row at a time has the advantage of using less power to drive the memory board.

Refresh Control Circuits.

The circuits that control refreshing are shown at the upper left section of sheet 1 of the logic diagram. The refresh control circuits examine the signals that indicate the following four conditions:

a. The second half of cycle I3 or I4 is present, which is indicated by the refresh signal from the processor. Refreshing is accomplished during the second half of the I3 and I4 portions of each instruction cycle. The signal that steps the refresh counter is I/O-CLR from the processor timing cycle. This is the normal refresh method.

b. The processor is operating in the single cycle mode, which is indicated by the STOP signal. Refreshing is performed continuously in this mode.

c. The disk is gaining access to memory.

d. The CRT control circuits are gaining access to memory.

After sensing these conditions, the refresh control circuits take the action required to gate the proper address to the memory and step the refresh address counter.

One output of the control circuits is the EBLE-RFSH (enable refresh) signal. Its function is to activate all four chip select signals and to inhibit output data transfer and parity checking during refreshing. If the disk or CRT is gaining access to memory, the output data transfer, parity checking, and normal row address decoding are enabled.

The actual refreshing is accomplished by the GATE-RF-SH-ADD (gate refresh address) signal from the refresh control circuits, which gates the refresh addresses to the memory. If the row number (address bits 12 and 13) being refreshed is the same as the row being addressed by the CRT, the refresh address is blocked and the CRT row address is applied to the memory. If the two row addresses are not equal, the memory refreshes.

The third signal produced by the refresh control circuits is INC-ADD (increment address), which steps the refresh counter by the one address on every I/O-CLR signal. If the CRT is using the same row address, the refresh signal is suppressed and the INC-ADD signal also inhibited by the refresh control circuits. The circuit blocking the INC-ADD signal is cleared on the TE timing pulse after the refresh signal from the processor indicates that CRT access is complete. Note that some circuits shown refer to the "disk". These are used only when this memory board is used in terminals that have disk units.

In summary: the important point is that if the same row is not selected by the CRT and the refresh address, refreshing is performed normally.

Parity Errors.

It was previously mentioned that detection of a parity error sent the parity error signal to the processor. No action is taken in the memory circuits, however, until the RUN flip-flop, which indicates that a cartridge tape unit is in operation, is cleared. (This ensures that system lock-up does not occur while a tape is in motion.) Then, the LOCK-SYS (lock system) signal is produced. In turn, the lock system signal disables the data output register. The register outputs go high, except for bit 7, which is held low by the lock system signal. This produces the character "177" on the data output lines. Sensing this, the processor provides a special display (full dot pattern) on the CRT. When the processor executes the EMP-Enable Memory Parity instruction the next time, the parity error flip-flop is cleared.

Internal Oscillator.

An internal oscillator is shown on sheet 2 of the memory logic diagram. This oscillator starts when power is first applied, and its function is to trigger the memory timing circuits. In turn, the memory timing circuits produce timing signal T02, which is necessary to apply the +5V output of the power supply to the memory circuits.

The general clear signal allows the oscillator output to reach the timing circuits until the PROGRAM LOAD switch is operated. Then, the clear signal blocks the oscillator output and processor timing signals TE and I/O clear trigger the memory timing circuits.

2-5. INPUT-OUTPUT CONTROLLER.

The input-output controller is a single board located beneath the processor board in the base of the 1501. Its purpose is to coordinate the exchange of information between the processor and the input-output devices. A block diagram of the I/O controller is shown in figure 2-24.

The accumulator, which is functionally part of the processor, is physically located on the I/O controller board. The purpose of this is to simplify the exchange of data between the accumulator and the I/O devices.

There are five input-output devices with which the I/O controller exchanges data and commands. They are:

a. Cartridge tape units. The I/O controller operates two cartridge tape units directly. This includes reading, writing, and control of tape motion.

b. CRT display. The I/O controller produces the display for the CRT by sending the sweep triggers and unblanking signals to the CRT assembly.

c. Keyboard. The I/O controller accepts the inputs from the keyboard and operates the speaker that is used to give operator cues.

d. Serial I/O channel. The serial I/O channel mentioned earlier in this section is operated by circuits in the I/O controller. This channel allows the intelligent terminal to communicate with other terminals or peripheral devices such as printers, tape units, etc.

e. Communications adapters. The communication adapters can exchange data with the accumulator. The I/O controller sends data and commands to the adapter and accepts the adapter output.

Input-Output Instructions.

Whenever the operation decoder in the processor finds that the operation code is 17x, it activates the I/O operation line (I/O-OP) to indicate that the instruction must be decoded and executed by the I/O controller. In turn, the I/O controller examines the three low-order bits (OP 0, 1, and 2) of the operation code to determine which I/O device is to be used.

The IWR is also available to the I/O controller. This byte determines the action to be taken, and it is routed to the I/O controller over the EX-MDR lines from the processor.

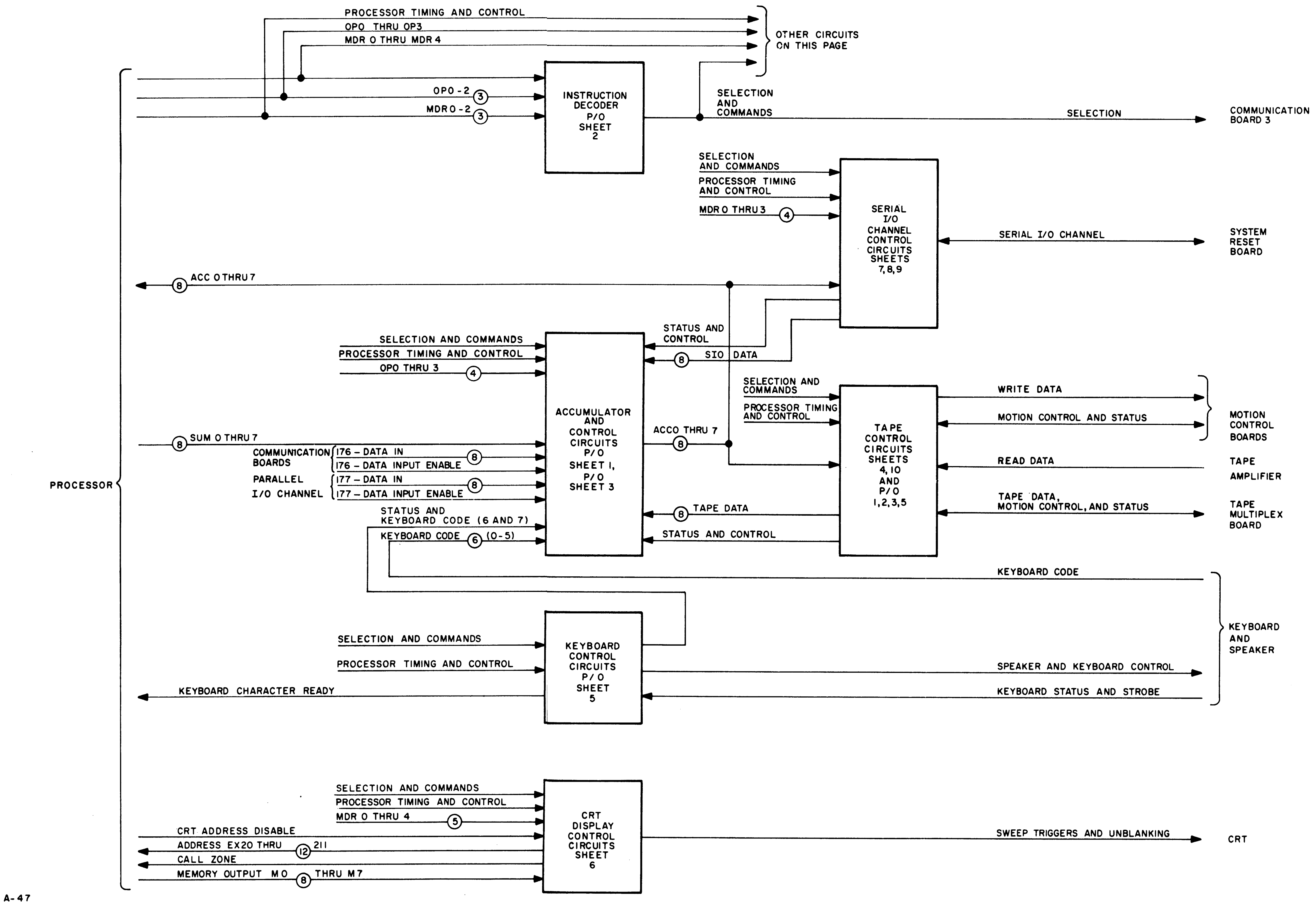


Figure 2-24. I/O Controller Block Diagram

Figure 2-25 shows the format of an input-output instruction. Note that the low-order bits of operation code choose the input-output device to be used, and the IWR chooses the action to be taken. However, if the serial input-output channel is selected, an address byte transmitted on the serial I/O channel specifies the unit to be operated and the control byte transmitted to the selected unit determines that action to be taken.

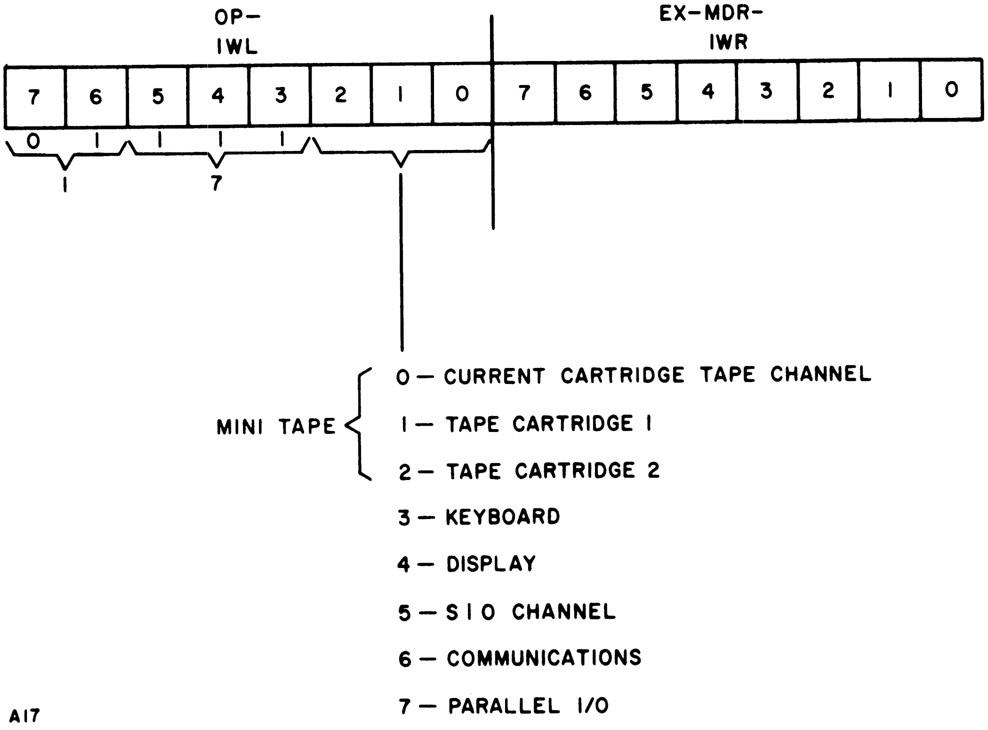


Figure 2-25. Format of an I/O Instruction

Input-Output Instruction Decoder.

The I/O instruction decoder is shown in the upper left corner of sheet 2 of the I/O controller logic diagram. Its purpose is to decode bits OP0-OP2 of the IWL of the I/O instruction in order to select the I/O device to be operated, then decode MDR 0-5 of the IWR to determine what action is to be taken.

When the first section of the decoder receives the I/O-OP (input-output operation) signal from the processor, it decodes IWL bits OP-0, OP-1, and

OP-2 into the signals listed below during the first half of cycle I4. A flip-flop set at the beginning of the I4 cycle maintains the decoder output during I4.

OP0-OP2 Code	I/O Device Selected	Signal
0	Cartridge tape, current tape channel	N-CURRENT-TAPE
1	Tape Unit 1	N-TAPE-A-OP
2	Tape Unit 2	N-TAPE-B-OP
3	Keyboard	N-KEYBOARD
4	Visual Display	N-CRT
5	Serial I/O channel	N-I/O
6	Communications channel	N-176-INT
7	Parallel I/O channel	N-OP-177

If bit OP2 is a zero, indicating that the code is between 0 and 3 and that a tape unit or the keyboard is selected, the second section of the decoder is activated during the second half of I4. This section examines bits MDR 0-5 (the IWR) and activates a tape or keyboard commands as follows:

MDR0- MDR5 Code	Command	Signal
00	Start tape forward, slow, with erase	N-FNS-WE
01	Start tape forward, slow, without erase	N-FNS-WOE
02	Start tape forward, fast	N-FHS
03	Start tape reverse, slow	N-RNS
04	Start tape reverse, fast	N-RHS
05	Stop tape	N-STOP-TAPE
06	Unlock keyboard	N-KEYBOARD-UNLOCK
07	Transfer byte	N-TRAN-BYTE

MDRO- MDR5 Code	Command	Signal
10	Set tape write mode	N-WRITE-SET
11	Set tape read mode	N-READ-SET
12	Rewind tape	N-SET-RWD
13	Enable keyboard click	N-KEYBOARD-CLICK
14	Lock keyboard	N-KEYBOARD-LOCK
16	Read status word	N-READ-STATUS

It should be noted that the device selection signals are held at the decoder output during the entire cycle of I4, while the commands are active only during the second half of cycle I4. The effect of each of these signals is discussed in the description of the circuits to which they are related.

Accumulator.

The accumulator is shown on sheet 1 of the I/O controller logic diagram, and the accumulator shift control circuits are shown in the upper right corner of sheet 3. The accumulator serves two purposes:

a. It holds one of the operands involved in many of the processor instructions and, after the instruction is executed, the accumulator holds the results of the operation.

b. It holds the data being transferred during input-output operations.

Also shown on sheet 1 is a tape shift register, in which data sent to the tape is converted to serial and data from the tape is converted to parallel, and a tape data buffer register. These circuits are not considered part of the accumulator for the purposes of this discussion. They are covered in the tape read-write circuits description.

The accumulator itself is an 8-bit shift register connected so that it can accept an 8-bit parallel input, provide an 8-bit parallel output, and perform circular right shifts (bit 7 toward bit 0, with bit 0 re-entering bit 7). The shifting is controlled by shift pulses developed by the shift control circuits.

Data from seven sources can be loaded into the accumulator. This data is applied to a multiplexer that preceeds the accumulator, and input selection is accomplished by a group of circuits shown on the lower left corner of sheet 1. These circuits develop a 3-bit selection code for the input multiplexer; they also develop a load signal that switches the accumulator to the parallel load mode.

Inputs to the selection circuits originate in the circuits that are using the accumulator for data transfer. For example, the keyboard control circuits provide the KBRD-IN (keyboard in) signal when they are reading a character from the keyboard. This establishes the selection code to gate KBO-KB7 through the multiplexer and into the accumulator. Listed below are all inputs to the multiplexer, the code required to accept them, and the signal that produces that code.

Input	Selection Code			Controlling Signals
	S2	S1	SO	
Fixed zeros	0	0	0	Absence of any of the signals below.
Adder (Sum 0-7)	0	0	1	N-INH-ACC-LD, and N-COMP-OP, and P-ACC-LD
Keyboard (KBO-7)	0	1	0	N-KBRD-IN
Tape Register (Tape 0-7)	0	1	1	N-TAPE-IN
SIO (I/O 0-7)	1	0	0	N-I/O-RD
PIO (177-DI-0 thru 7)	1	0	1	N-177-DI-ENABLE
Status word	1	1	0	N-READ-STATUS
Communication adapter (176-DI-0 thru 7)	1	1	1	N-176-DI-ENABLE

Notice in the table above that three signals must be present simultaneously in order to load the sum from the adder. They are: (1) the accumulator load signal from the processor, (2) the not-inhibit-accumulator-load signal from the processor, (3) the not compare operation signal from the processor. The compare operation signal is present only during the CPA (Compare Accumulator) and CPX (Compare Index Register) instructions. Thus, during all other instructions, the not compare operation signal allows the accumulator to be loaded with the adder outputs whenever the processor generates the accumulator load signal and does not inhibit accumulator loading.

Three Class 3 instructions require that the accumulator contents be shifted.

They are:

a. SAN - Shift and Logical AND to Accumulator

b. SER - Shift and Exclusive OR to the Accumulator

c. SIR - Shift and Inclusive OR to the Accumulator

In each case, the instruction performs a circular right shift of the accumulator by the number of positions specified in the shift count (OPO through OP2) before the logical operation is carried out. The circuits that accomplish the shift are shown on sheet 3, and they are described in the following paragraphs.

The operation code for all instructions is decoded in the processor, and the processor sends the LOGIC-OP (logical operation AND, OR, Exclusive OR) signal to the I/O controller. This signal, which indicates that the current instruction is ANA, SAN, ERA, SER, IRA, or SIR, is applied to the shift control circuits. However, the shift control circuits must then distinguish between the instructions requiring shifts (SAN, SER, SIR) and those that do not.

Refer to the instruction formats shown earlier in this section. The shift instructions all have a zero in bit OP3. (Note that the immediate address mode of ANA, ERA, and IRA also have a zero in bit 3 but it has no effect for reasons shown later in this discussion). Thus, the logical operation signal and a zero in bit OP3 allow the shift control circuits to produce shift pulses.

The number of shift pulses is controlled by the shift count in bits OPO-OP2. This is always zero in the immediate address mode of ANA, ERA, and IRA and thus no shifts are produced by these instructions even though the logical operation signal is present and OP3 is zero.

Figure 2-26 illustrates how the shift pulses are produced. In cycle I3, the instruction is loaded into the operation register and the inputs to the shift control circuits become active. Timing signals are ANDed with the shift count so as to produce the number of shifts selected by the shift count. (A maximum of seven is shown in the figure.) Three shifts can be accomplished during cycle I3.

During cycle I4, the processor acquires the IWR from memory. Simultaneously, the shift control circuits continue to combine the shift count and timing signals to shift the accumulator. (A maximum of four shifts can be accomplished during I4.) By the beginning of the E2 cycle, shifting has been completed and the logical operation called for in the instruction can be executed.

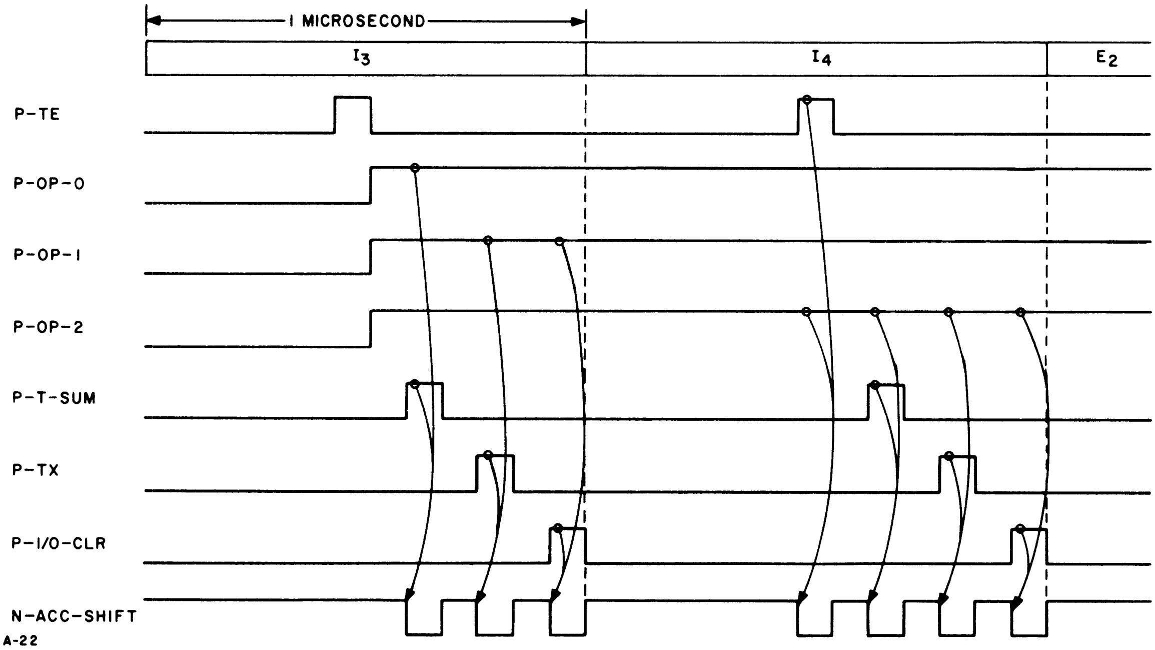


Figure 2-26. Shift Control Circuits Timing Diagram

One addtional function of the shift control circuits is to produce the ACC-SHIFT signal when the accumulator is being loaded in parallel. This is necessary because the ACC-SHIFT signal is the clock pulse to the accumulator and, even though the ACC-LD signal has switched the accumulator to the load mode, the clock input must be activated to accomplish the loading. The TX timing signal and ACC-LD signal combine to produce the clock pulse for loading.

When the Read Status instruction is given, the accumulator is loaded with status conditions from several circuits. The format of the status word is shown in figure 2-27.

Keyboard Control Circuits.

The keyboard control circuits are shown on sheet 5 of the input-output controller logic diagram. Their purpose is to coordinate the operation of the keyboard with the processor; the use of the keyboard input depends upon the program being executed by the processor.

The keyboard includes an encoder, and when a key on the keyboard is pressed, the encoder produces a 6-bit code corresponding to the physical location of the key. This code is applied directly to the accumulator, where it is accepted when the keyboard control circuits generate the

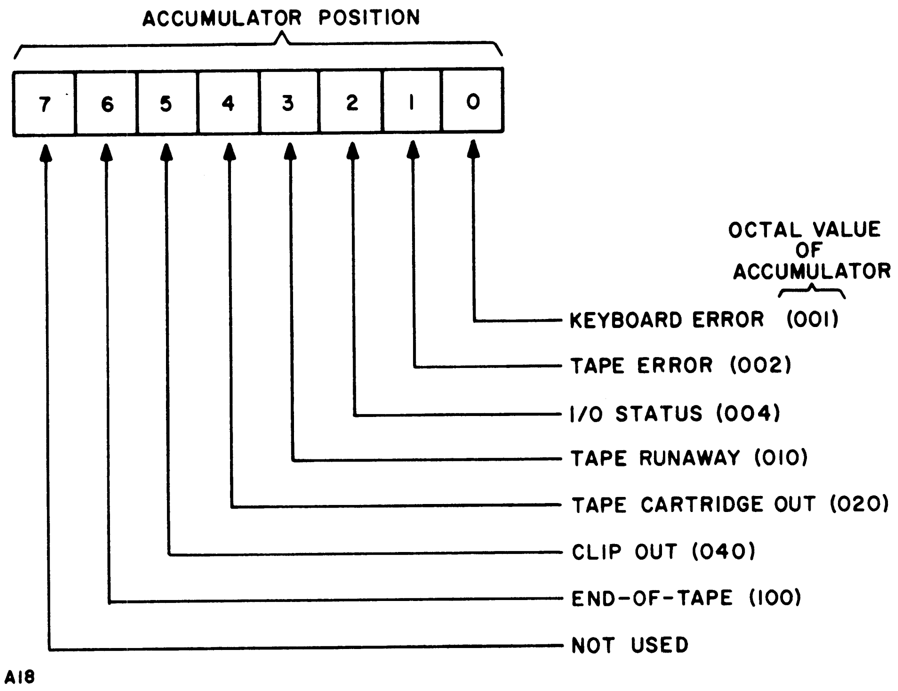


Figure 2-27. Format of I/O Status Word

proper control signal. However, the keyboard also sends four control signals to the keyboard control circuits. They are:

a. STROBE, which indicates that a keyboard character is ready.

b. NUM-SH (number shift), which indicates that the keyboard has been shifted to the "upper case" mode and that the keyboard output represents the upper character on the key.

c. ALPHA-SH (alphabetic shift), which indicates that the keyboard is in the "lower case" mode in which the keyboard output represents the lower character on the key.

d. CONT'L (control key), which indicates that the control key has been depressed.

When the strobe signal is applied to the keyboard control circuits, it causes two things to happen:

a. It allows a decoder to examine the number shift, alphabetic shift, and control key lines and form a 2-bit code to indicate which of these lines is active. This 2-bit code (KB-6 and KB-7) is then routed to positions 6 and 7 of the accumulator.

b. It sets the character ready flip-flop, which forwards the KYBD-CHAR-RDY (keyboard character ready) signal to the processor.

It should be noted that the KYBD-CHAR-RDY signal indicates that a character is ready to be loaded into the accumulator but its function in the processor circuits is to prevent an interrupt from taking place until the keyboard input is read. Processor action is required in order to load the accumulator. Each time the processor is to read the keyboard input, it branches to a program routine required to accept the keyboard input. One of the instructions in this routine is 173-007, select keyboard and transfer byte-unconditional. When the I/O controller decodes this instruction, it sends the KEYBOARD and TRAN-BYTE (transfer byte) signals to the keyboard control circuits. In turn, the keyboard control circuits produce the KBRD-IN (keyboard in) signal, which is applied to the accumulator. There, it activates a 3-bit selection code to gate KB0-KB7 through the accumulator input multiplexer and loads it into the accumulator. The processor then accepts the character and continues its program.

If a transfer byte instruction has been executed, the character ready flip-flop is cleared at TE time of cycle I3. If a second key is depressed before the first has been accepted and the character ready flip-flop cleared, the strobe signal accompanying the second character sets the keyboard error flip-flop. The status of this flip-flop is available as one bit of the status word.

The preceeding discussion assumed that the keyboard had a character ready when the processor read the keyboard. However, it is also possible that processor is expecting a keyboard input and causes the I/O controller to execute a transfer byte instruction before the key has been pressed and the strobe signal generated. In this case, the I/O stall flip-flop is set at TE time of cycle I3. In turn, the I/O stall signal is sent to the processor.

The processor's response to the stall signal depends upon whether the transfer byte instruction was 173-007, transfer byte-unconditional, or 173-207, transfer byte-conditional (skip on busy). If the former instruction is active, the stall signal prevents the instruction address from being advanced, thus holding the processor at the current instruction until the strobe signal is received and the I/O stall flip-flop is cleared. If the latter instruction (173-207) was executed, however, the I/O stall signal causes the processor to skip one instruction and proceed. In this case, the processor must execute another transfer byte instruction to acquire the keyboard character, and the time at which this instruction is executed is determined by the program.

CRT Display Control Circuits.

The CRT display control circuits are shown on sheet 6 of the I/O controller logic diagram. Their purpose is to produce the display on the CRT, therefore, a brief review of the display and the CRT characteristics is presented before the display control circuits are described.

The CRT display is designed to display 256 characters in 8 rows of 32 characters each. Each character is formed by lighted dots within a 5-space-wide by 7-space-high matrix. There are 9 blank spaces between character rows and 3 blank spaces between character columns. This is shown in figure 2-28.

Only three signals are forwarded to the CRT assembly by the display control circuits.

They are:

a. CRT-UNBLANK, which allows the beam to be displayed.

b. F-TRIG (fast trigger), which triggers the vertical scan.

c. S-TRIG (slow trigger), which triggers the horizontal scan.

It should be noted here that the CRT is a standard 5-inch television tube with the scanning pattern rotated 90 degrees. In this application, the fast scan is vertical and the slow scan is horizontal. In other words, the beam scans a column and then moves to the next column until all columns have been scanned: then, the process is repeated.

The characters are formed by unblanking the beam when a dot is to be formed and blanking it at all other times. The information that is used to control the unblanking originates in dot patterns stored in memory.

Each character location on the screen has an associated memory location that holds a code specifying the character that is to be displayed at that location. The code is made up of 6 bits that are capable of selecting any one of 64 different characters. When a specific character location is reached, the display control circuits use the code stored for that location to address the dot pattern stored in memory.

The dot patterns stored in memory are shown in the standard software memory map (drawing 001804-004 in Volume II of the Service Manual). Refer to that drawing. Note that the 64 different characters that can be displayed are numbered in octal 00 through 77 left to right across the top of the page. The character code stored for a specific location on the screen selects a group of five memory addresses, which are shown vertically below each character code.

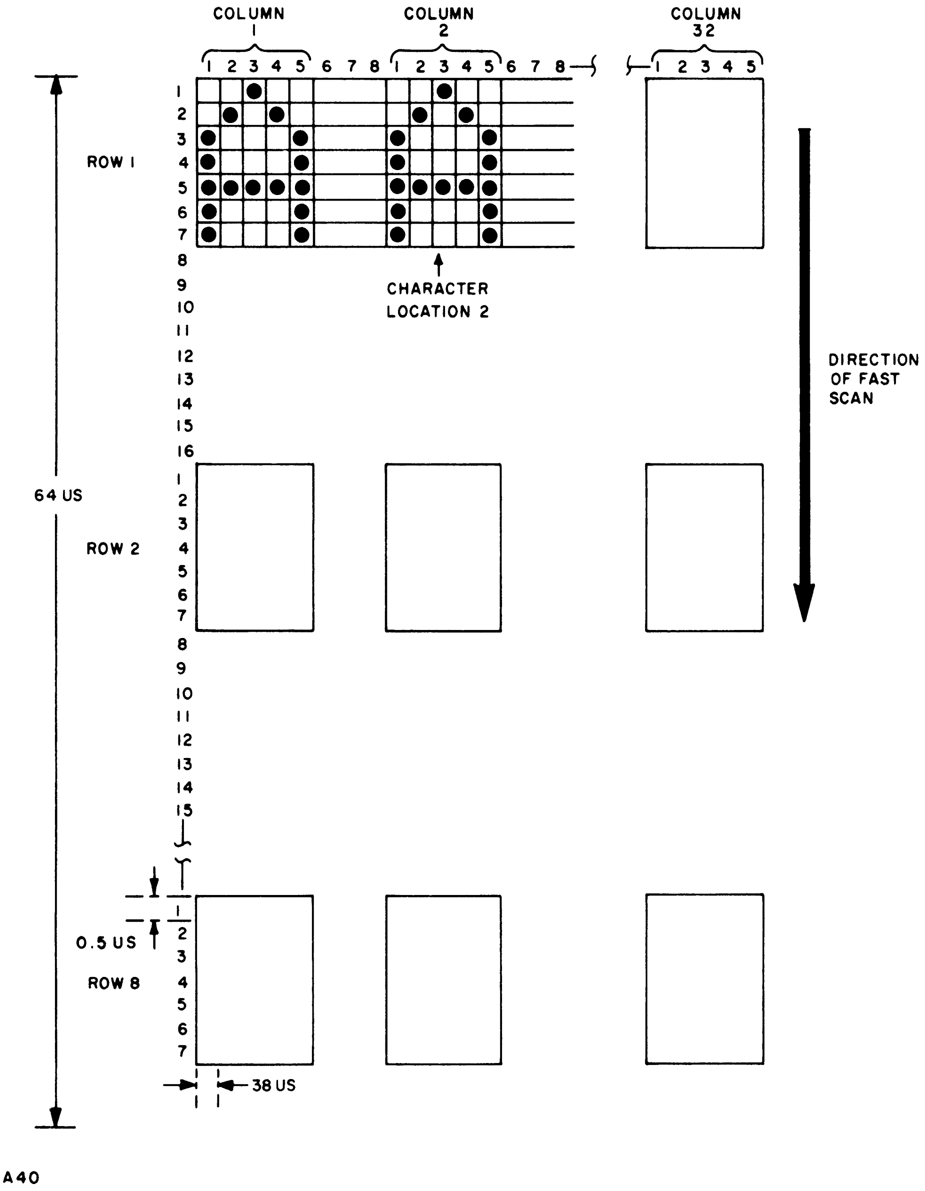


Figure 2-28. Format of Display Area

During the first column of the first character position, the dot pattern stored in the first memory location is read out and shifted serially to the CRT. This causes unblanking of dots in the proper positions to form the first column of a character. Next, the scan proceeds to the first column of the first character in the second row. This process continues until the end of the first vertical scan line is reached. At this point, the beam is moved to the next column to be scanned and the process of reading the contents of memory locations and shifting them serially to the CRT is repeated. Since there are five vertical lines in every character, five memory locations are required to store each dot pattern. Therefore, 320 memory locations are required to store the dot patterns for the 64-character set.

The CRT display circuits can only gain access to memory when they are permitted to do so by the processor at I3 and I4 times. While a dot pattern line of a given character is being displayed, the display control circuits must read out the required dot pattern line for the next lower character on the screen.

The following is an example of the sequence of events resulting in the display of a character at a specified location on the CRT: Assume the next character location to be displayed on the CRT is location 2 shown in figure 2-25, and also assume that the processor has stored the code for an "A" in character code table location 2. Although the character code can be read during either I3 or I4 and the dot pattern can be read during either I3 or I4, the following example uses I3 for the character code and I4 for the dot pattern. During I3, the CRT display control circuits address table location 2. The code for the character A is read from memory and loaded into a buffer register. This code is used to address memory during I4, and the memory location containing the dot pattern for column 1 of the A is acquired and stored in another buffer register. The dot pattern is then shifted serially out of a shift register and transmitted to the CRT as an unblanking signal that turns the beam on and off.

The next step in this description is to review the input-output instructions that apply to the CRT display control circuits. When the IWL of an input-output instruction is 174, the "4" selects the CRT display control circuits for operation. The IWR on the EX-MDR 0-7 lines is then decoded to determine what action is required. The format of the IWR is shown in figure 2-29.

Note that bits 1, 5, 6, and 7 select the page in memory containing the codes for the characters to be displayed; the "S" bits selecting the section and the "L" bits selecting the level within the section. Since there are only two "S" bits and only two "L" bits, the pages that can be selected are:

a. 00, 01, 02, and 03

b. 10, 11, 12, and 13

c. 20, 21, 22, and 23

d. 30, 31, 32, and 33

Each page selected consists of 256 memory locations, one location for each of the 256 character positions on the screen. Stored in each of these 256 locations is the 6-bit code that chooses the character to be displayed.

Bit 0 is the mode selection bit. When it is a zero, the full 8-line display is created. When it is a one, it limits the display to four lines, thus saving memory space. Bit 2 is the disable CRT command. It prevents any display from appearing. Bit 3 is the underscore bit; its effect is to place the underscore line under characters.

Bit 4 is the interleave bit in the 8-line display mode and the half-page, or "zone" bit, in the 4-line display mode. The function of this bit is explained in greater detail below.

There are four basic modes of operation for the display circuits:

a. Normal mode -- 8 rows in sequence

b. Interleave mode -- 8 rows interleaved

c. Four top rows only

d. Four bottom rows only

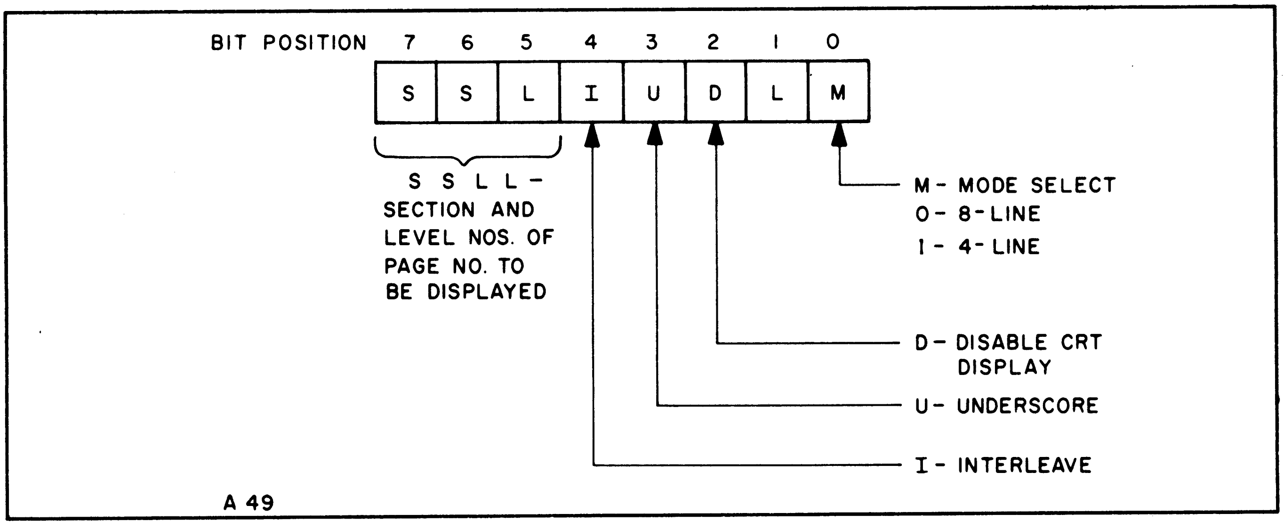


Figure 2-29. Format of IWR for CRT Instructions

In the normal display mode, a page of 256 characters is displayed in a continuous fashion, from memory locations 000 through 377, (octal) of the selected page, as shown in figure 2-30. In this case, the IWR of the I/O instruction is as follows:

a. Bit 0 (Mode) is a zero, indicating the full 8 lines are to be displayed.

b. Bit 4 (interleave) is a zero, indicating no interleaving.

If bit 4, the interleave bit, is a one, the 8-line interleaved mode is selected. In the interleave mode, the page of 256 characters is displayed in the sequence shown in figure 2-31.

If bit 0, the mode selection bit, is a one, only the top or bottom four rows are displayed. They are read from the memory locations shown for the normal mode, however, bit 4, the zone bit chooses either the top four rows (if bit 4 is a zero) or the bottom four rows (if bit 4 is a one).

Command Register

The command register is shown in the upper left corner of the display control circuits logic diagram (sheet 6 of the logic diagram). Its purpose is to hold bits 0 through 4 of the IWR when the CRT is selected by the I/O instruction.

The I/O controller's operation decoder examines OPO-2 of the I/O instruction during the I4 cycle and, if the CRT is selected, it activates the CRT signal to the command register. The TX signal then loads bits EX-MDR-0 through 4 into the command register.

Simultaneously, the CRT-ZONE-SELECT signal is sent to the processor where it loads MDR 5 and 6 into the zone register and MDR 7 into the zone E flip-flop. The effect of this is to activate the ZONE-SELECT C, D, and E lines sent to the memory address net based upon bits 5, 6, and 7 of the current I/O instruction IWR. Bit 1 of the instruction is the remaining page address selection bit. It is routed from the command register to position EX-28 of the address output circuits and subsequently to the address net.

The remaining bits in the instruction are distributed as follows:

a. Bit 0 either activates the 4L (four-line) command or leaves it inactive.

b. Bit 2 (the disable bit) sets the disable flip-flop if the CRT is to be disabled. This removes the CRT-EN (CRT enable) signal that is present unless the disable command is issued.

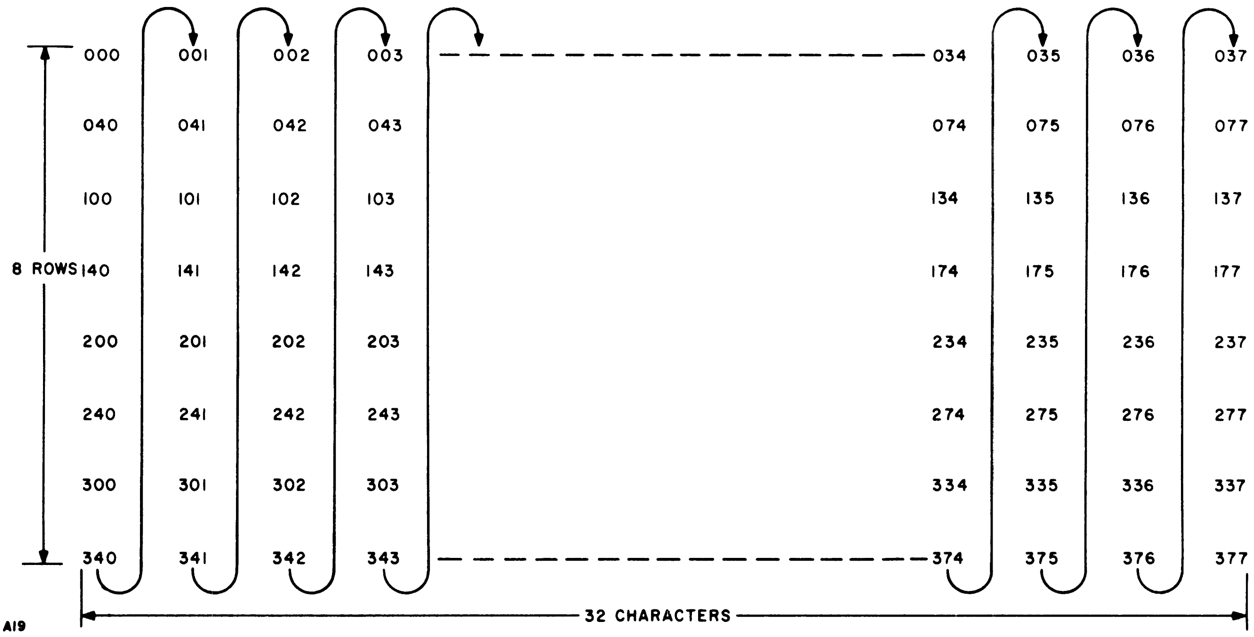


Figure 2-30. Display Sequence in Normal Mode

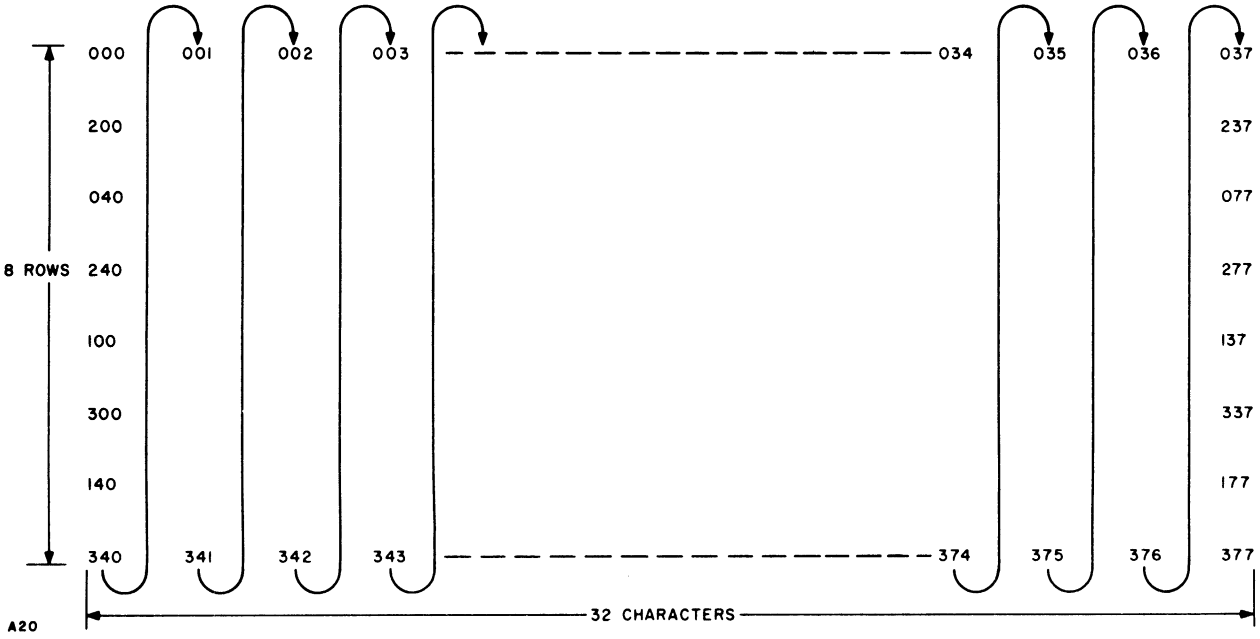


Figure 2-31. Display Sequence in Interleaved Mode

c. Bit 3 (underscore) is applied to position EX-29 of the address output circuits and to a gate where it is ANDed with position M6 of the character buffer. If M6 is a zero, an underscore latch is set at the proper time. Use of the underscore latch output is discussed later in this description.

d. Bit 4 (interleave or zone select) produces the EP (enable page) signal to the CRT-V output circuit. This unblanks the CRT at the proper time to produce either the interleaved or half-page display. Bit 4 is also ANDed with the mode bit (bit 0) to produce the INTL (interleave) signal if the interleave mode is selected.

Clock Circuit

The basic timing of the operation of the CRT display control circuits is established by clock pulses that occur at a 2MHz rate. These clock pulses are derived from three processor timing signals MEM-CLK, TE, and I/O-CLR. They combine as shown in figure 2-32 to produce the 2 MHz clock pulses. In turn, these pulses are distributed throughout the CRT display control circuits. Note that these pulses are distributed even though the CRT is disabled. The enable signal simply prevents the unblanking line from becoming active and prevents the CALL-ZONE signal from reaching memory.

Display Timing

Before the function of the display counter can be described in detail, the timing requirements of the CRT scan must be discussed. As previously mentioned, the characters are produced by a vertical scan of the CRT beam, and the beam is unblanked during specific time periods to form dots which, in turn, form characters.

Figure 2-28 shows a grid pattern that represents the CRT positions in which characters are formed. Note that there are seven time periods in each column of a character, and that there are nine time periods between rows of characters. Thus, the vertical space alloted to a character and the blank space is 16 time periods. Each time period is 0.5 microseconds; therefore, 8 microseconds are used for each character and the separation area. There are eight rows of characters, making the total time for one vertical scan 64 microseconds.

After 64 microseconds, the vertical scan ends and the beam returns to the top of the next column to be scanned. Twelve microseconds are allowed for the retrace and preparation for the next scan. This means that the vertical scan is triggered every 76 microseconds (64 for the scan and 12 for the retrace). A timing diagram showing development of the vertical timing is shown in figure 2-33.

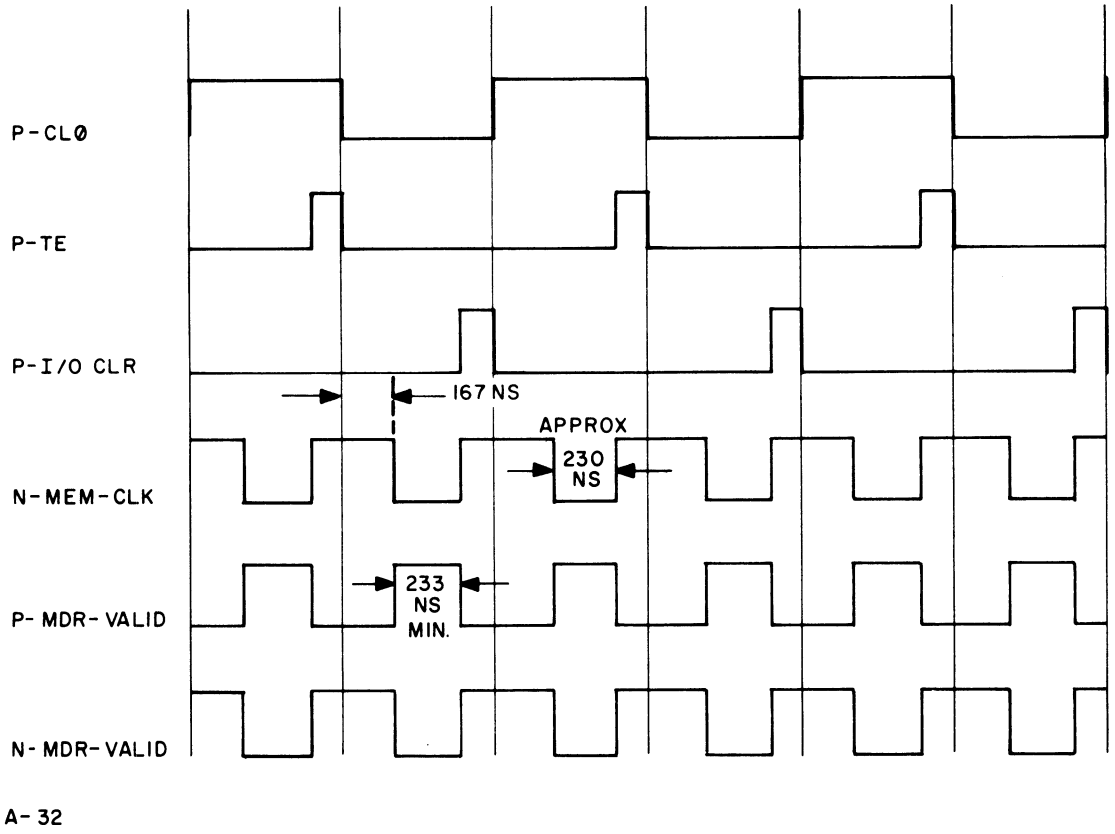


Figure 2-32. 2MHz Clock Timing Diagram

The F-TRIG (fast trigger) signal that is sent to the CRT is formed by a flip-flop (F18), which is operated by stages D and Q of the display counter. The 2MHz clock signal steps the display counter, and stage Q is active for 64 microseconds. The combination of the stage D and stage Q outputs form the fast trigger signal. Horizontal scanning is controlled by the S-TRIG (slow trigger) signal, and operates much more slowly than the vertical scan.

It should be noted at this point that the vertical lines are not scanned in consecutive order 1, 2, 3, 4, 5 etc., but are scanned in an interlaced pattern. First the odd lines are scanned, then the even lines. Therefore there are actually two "fields", an odd field and an even field, similar to the scan in a television receiver. One complete horizontal scan is made in which all the odd vertical lines appear, then another horizontal scan is made in which all the even vertical scan lines appear.

Stage G of the display counter indicates whether the odd or even field is being scanned. The outputs of the stages of the display counter involved in producing the slow trigger are shown in figure 2-34. Notice that the

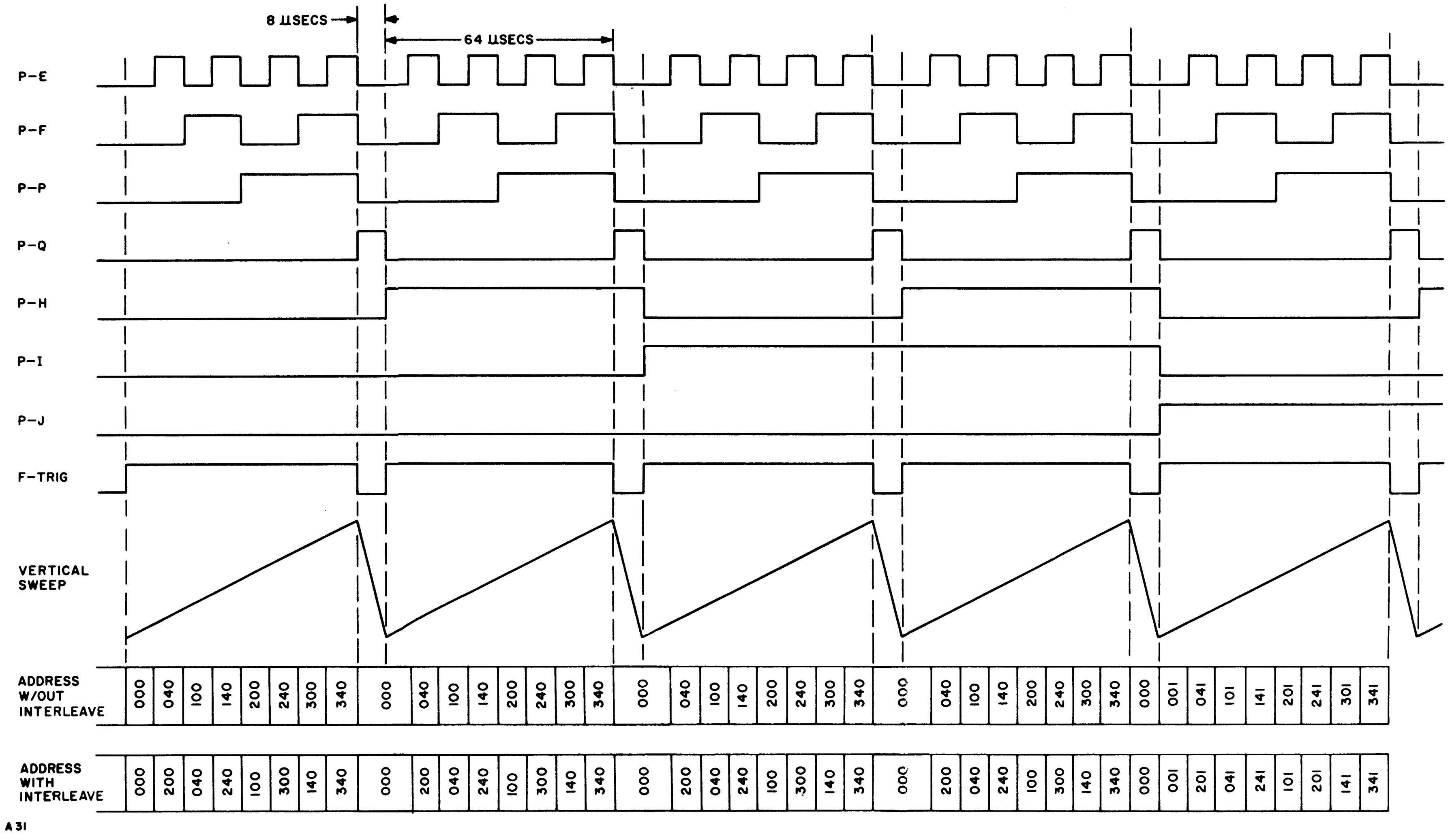


Figure 2-33. Vertical Sweep Timing Diagram

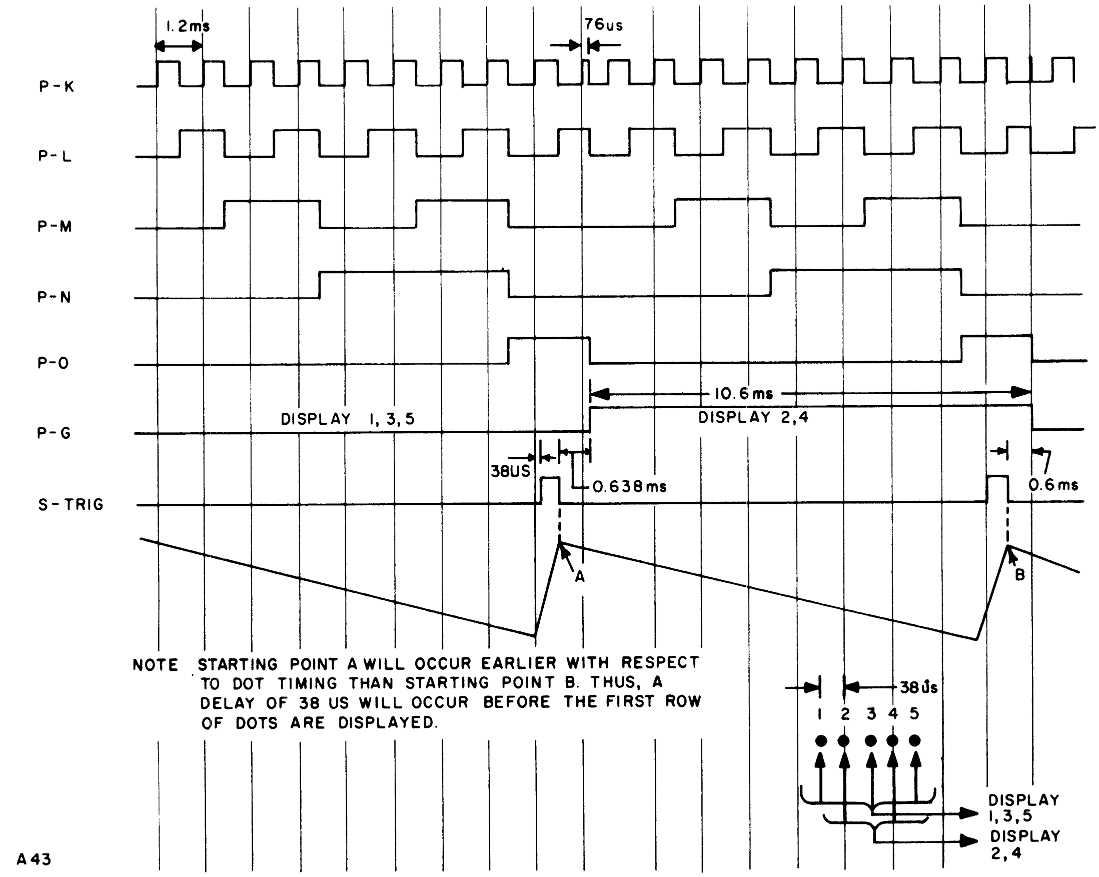

1.2ms
76us
P-K
P-L
P-M
P-N
P-O
P-G
DISPLAY 1,3,5
10.6ms
DISPLAY 2,4
38US
0.638ms
0.6ms
S-TRIG
A
B
NOTE STARTING POINT A WILL OCCUR EARLIER WITH RESPECT TO DOT TIMING THAN STARTING POINT B. THUS, A DELAY OF 38 US WILL OCCUR BEFORE THE FIRST ROW OF DOTS ARE DISPLAYED.
38us
1 2 3 4 5
DISPLAY 1,3,5
DISPLAY 2,4
A43

Figure 2-34. Horizontal Sweep Timing Diagram

duration of the horizontal sweep is either 11.476 milliseconds, if the odd field is being scanned, or 11.4 milliseconds, if the even field is being scanned. The additional 76 microseconds allows the first vertical scan line to take place in the odd field. (There are actually 305 vertical scan lines, one extra in the odd field.)

Memory Addressing

Prior to the display of each character, the character code for that character must be read from memory. Then, the character code is used to address the dot patterns. Figure 2-35 shows the sources of the address applied to memory. Readout of the character code is discussed first.

When the I/O instruction that selects a display for the CRT is first received, four of the bits select the page to be displayed. MDR 5, 6, and 7 are held in the zone register in the processor. When the CRT display circuits send the CALL-ZONE signal to the processor, these bits address memory. MDR 1 is routed to memory over the EX-28 line. In combination, these bits select the page that holds the 256 character codes that are to be displayed. The addresses within the page versus character position on

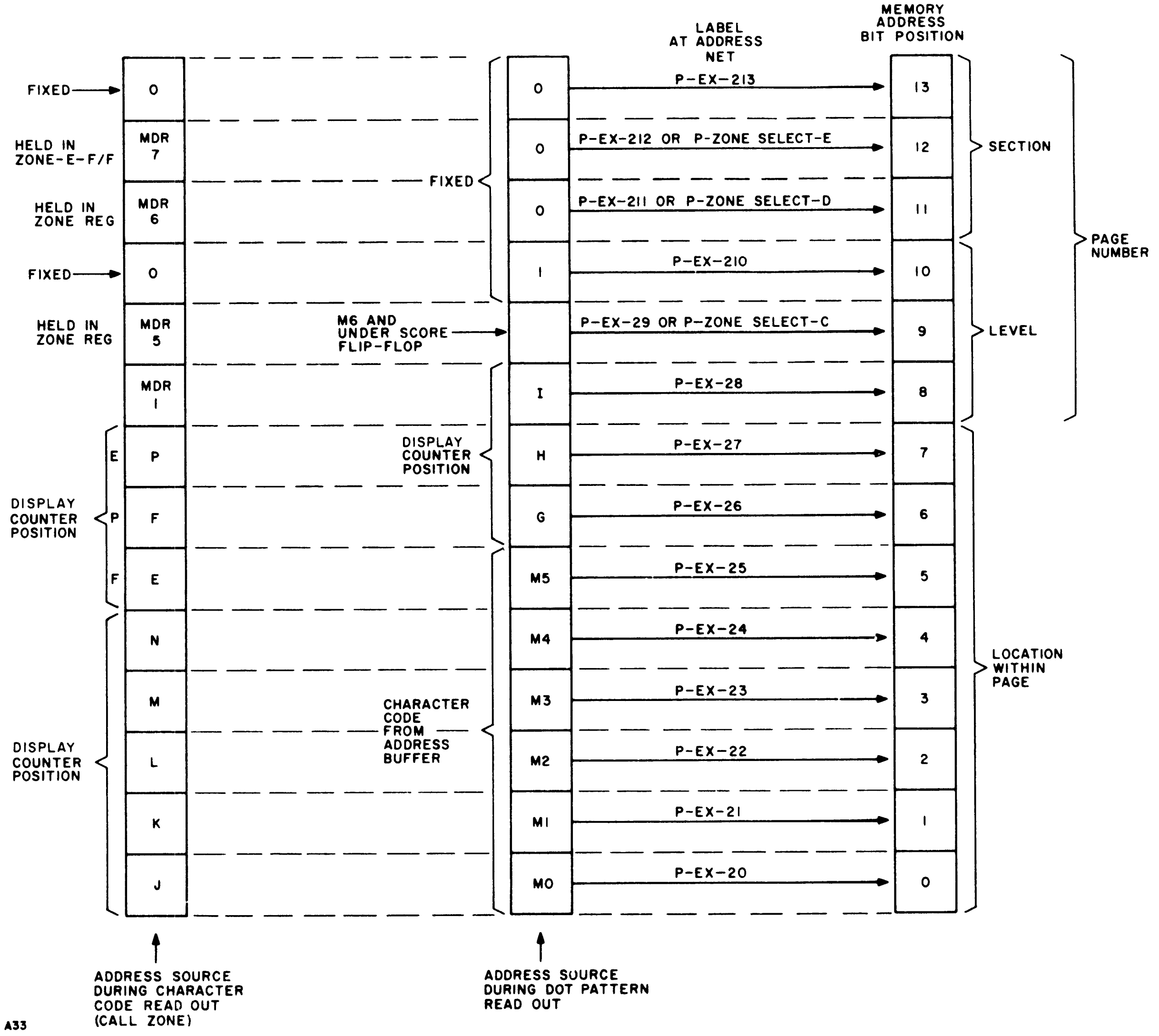


Figure 2-35. Addresses Applied to Memory

the screen are shown in figure 2-30. Note that they run consecutively, horizontally.

Assume first that the eight rows of characters are to be displayed in the normal mode. In this case, stages E, F, and P of the display counter (which are the vertical scan stages) supply the three high-order bits of the address within the selected page. As the vertical scan progresses from the top character in the first column to the bottom character in the first column, these stages are stepped by the clock and, in effect, count the row number. When the output of these stages are gated to the memory address lines the effect is to skip the memory addresses down the first column shown in figure 2-30.

The remaining five digits of the memory address are supplied by stages J, K, L, M, and N of the display counter. These are the horizontal scan stages, and they count horizontal time, effectively character columns. So, these stages count the columns, left to right, as shown in figure 2-30. When these character column bits combine with the character row bits supplied by the vertical scan stages, they produce the series of memory addresses 000, 040, 100, 140, 200, 240, 300, 340, 001, 041, etc. shown in figure 2-27. Thus, as the scan progresses, the memory location holding the character code for that screen location is addressed in the sequence that the character appears physically on the screen, top to bottom, left to right.

When the interleave mode is selected by bit 4 of the I/O instruction, the address of the memory location that corresponds to a specific physical location on the screen is changed by moving the stage E output from the least significant position of the row number to the least significant position, and by moving stages P and F each to the next least significant position. The counter continues to count in the same order as the scan progresses from the top to the bottom of the display but the memory locations addressed change to the order shown in figure 2-31. (Note that no change is made in the horizontal counter outputs.)

After the character code for a specific character position is acquired from memory and loaded into the address buffer, the dot patterns required to form the character must be read from memory. Figure 2-35 shows the source of memory addresses used during read out of the dot pattern.

First, it should be noted that the dot patterns are always located in the same positions in memory, all of page 04 and the first 77 addresses in page 05. This is shown in the standard memory map on drawing 1804.

As shown in figure 2-35, the four most significant bits of the page address supplied to memory while dot patterns are being read are 0001. This establishes the page as 04 or greater. The two remaining bits of the page number address are determined as follows:

a. The low-order bit (28) is controlled by stage I of the display counter. "I" is one of the horizontal count stages, and "I" switches the page number from 04 to 05 when the fifth column of a character is reached.

b. Bit 29 is controlled by bit 6 of the character code (M6) and the underscore flip-flop. If bit 6 is a one and the underscore command has not been given all addresses are shifted to pages 06 and 07 to acquire dot patterns for an alternate character set.

Positions M0-M5 shown in figure 2-35 are the six bits of the character code acquired from memory and held in the address buffer, and they choose one of the columns of dot patterns shown in the memory map. The horizontal count in stages G, H, and I of the display counter specifies which column of the character is being produced and addresses the dot pattern for the appropriate column.

When the vertical scan reaches the next lower character position, a new character code is loaded in the addresses buffer and the dot pattern for the same column is read from memory. After one vertical scan, the horizontal count moves up to read the dot patterns for the next column of each of the eight characters.

As long as bit M6 is a zero, the characters do not have an underscore line. However, if bit M6 is a one and the underscore command is present, an extra one bit is loaded directly into the most significant position of the shift register. The result of this is to place an extra dot in the column, as shown in figure 2-36.

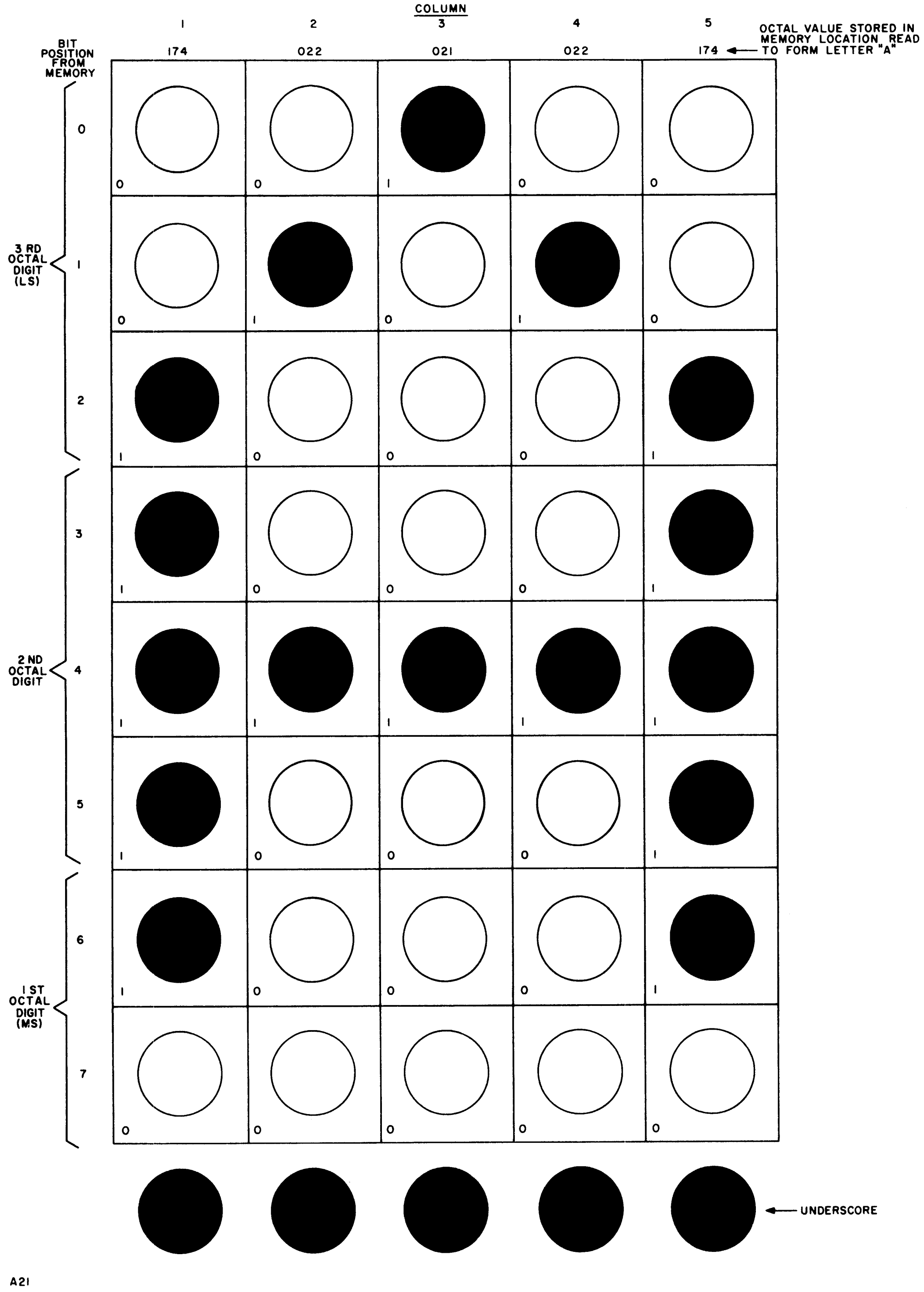
The next point that must be considered in the readout of dot patterns is that the odd vertical lines are scanned during the odd field and the even vertical lines are scanned during the even field. This requires that the dot patterns for columns 1, 3, and 5 of a character be read during the odd field and columns 2 and 4 during the even field. The signal that indicates whether the odd or even field is being scanned is stage G of the counter.

Stage G determines whether the most significant digit of the dot pattern address is odd or even. When the first, third and fifth columns of a character are being scanned, stage G places a zero in bit 26 to make the most significant digit of the dot pattern even (0XX and 2XX of page 04 and 0XX of page 05). When the second and fourth columns of a character are being scanned, stage G places a one in bit 26 to make the most significant digit of the dot pattern odd (1XX and 3XX of page 04).

As previously mentioned, bits 27 and 28 of the address are controlled by counter stages H and I, which indicate which of the five character columns are being scanned. These two bits combine with Stage G (the odd-even field stage) to read the dot patterns in the following sequence:

```
                         ,character code bits (M0-M5)
                        /
                       ⏞
                     ⎧0XX, Page 04
     odd ————————————⎨2XX, Page 04
     field           ⎩0XX, Page 05

     even ___________⎧1XX, Page 04
     field‾‾‾‾‾‾‾‾‾‾‾⎩3XX, Page 04
```



A21

Figure 2-36. Formation of the Character A

Character Synchronization Counter

The character synchronization counter (CSC) is shown in the lower left corner of sheet 6 of the I/O controller logic diagram. Its purpose is to coordinate the readout of character codes and dot patterns from memory with the vertical scan of the CRT display. A timing diagram showing the relationship of processor instruction cycles to the character synchronization counter appears in figure 2-37.

Since the display counter controls the CRT scan, its output also triggers the CSC. The output of stage D of the display counter is the key timing waveform. Its cycle time is 8 microseconds, which is the time from the beginning of one character to the beginning of the next character in each vertical scan. The stage D signal and the output of stage A (a 1-microsecond cycle) are combined in the start flip-flop (B18) so as to form a 0.5 microsecond ST (start trigger) signal each time the stage D waveform makes a negative-going transition, which is every eight microseconds. In turn, the start trigger performs three functions:

a. It loads the current dot pattern into the dot pattern shift register.

b. It clocks the underscore flip-flop to record the underscore status of the current character.

c. It resets the first stage of the CSC to allow the character synchronization cycle to begin.

At the same time that the stage D signal forms the start trigger, it also sets the second stage of the CSC. Note that the D signal is gated to the second stage of the CSC by the stage Q output of the refresh counter. Since the Q signal is active for the 64 microseconds required for the vertical scan of eight characters, it allows the D signal to reach the CSC during the trace time and blocks it during the 12-microsecond retrace period. Thus, the second stage of the CSC indicates that a vertical scan is taking place.

The next step is to acquire the character code from memory, and this must be coordinated with cycle I3 or I4. During I3 or I4, the third stage of the CSC is set, and the output of the third stage performs the following:

a. It sends the CALL-ZONE signal to the processor. This causes the processor to read the contents of the memory address selected by the display control circuits. The character code address is gated to memory because stage 4 of the CSC is not yet set.

b. It resets stage 2 so that the next operation of the CSC must be started by the start (ST) signal.

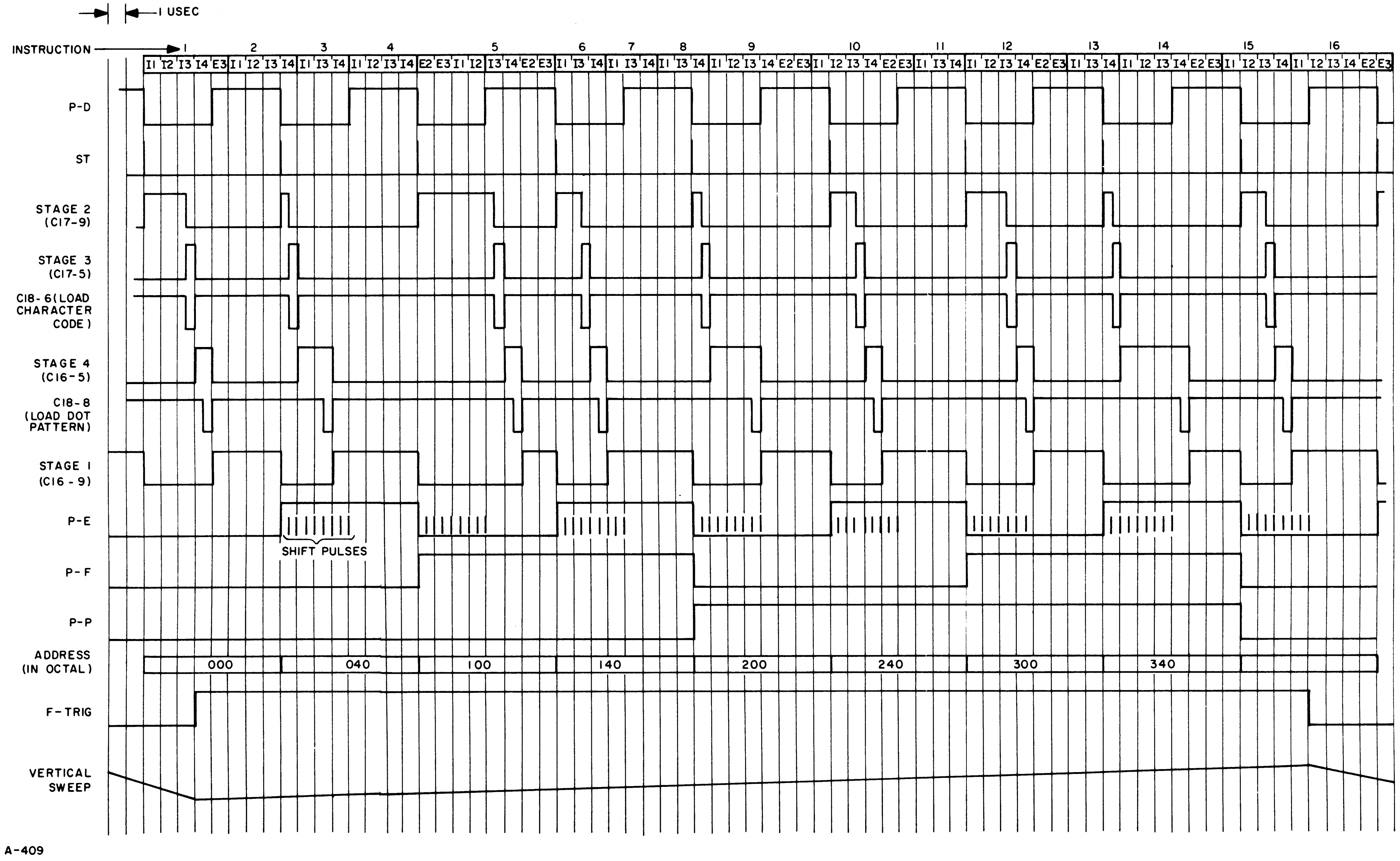

1 USEC
INSTRUCTION
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
I1 I2 I3 I4 E3 I1 I2 I3 I4 I1 I3 I4 I1 I2 I3 I4 E2 E3 I1 I2 I3 I4 E2 E3 I1 I3 I4 I1 I3 I4 I1 I3 I4 I1 I2 I3 I4 E2 E3 I1 I2 I3 I4 E2 E3 I1 I3 I4 I1 I2 I3 I4 E2 E3 I1 I3 I4 I1 I2 I3 I4 E2 E3 I1 I2 I3 I4 I1 I2 I3 I4 E2 E3
P-D
ST
STAGE 2 (C17-9)
STAGE 3 (C17-5)
C18-6 (LOAD CHARACTER CODE)
STAGE 4 (C16-5)
C18-8 (LOAD DOT PATTERN)
STAGE 1 (C16-9)
P-E
SHIFT PULSES
P-F
P-P
ADDRESS (IN OCTAL)
000
040
100
140
200
240
300
340
F-TRIG
VERTICAL SWEEP
A-409

Figure 2-37. Character Synchronization Timing Diagram

When I3 or I4 ends, stage 4 is set. This indicates that the character code has been acquired from memory. The end of I3 or I4 loads the character code on the M0-M7 lines into the address buffer register and deactivates the CALL-ZONE signal. At this point, the character code for the next character position is held in the address buffer register. When stage 4 was set, it reset stage 3. So, after a character code has been read, only stage 4 of the CSC remains set.

The next instruction cycle I3 or I4 then occurs, and the dot pattern address specified by the character code held in the address buffer register is read from memory. At the end of this cycle, the dot pattern is loaded into the dot pattern buffer and stage 1 of the CSC is set. (It remains set until the next start trigger is produced.)

To summarize the operation of the CSC:

a. It is started every 8 microseconds during the 64-microsecond vertical trace.

b. A character code is read during I3 or I4 and loaded into the address buffer register.

c. A dot pattern is read during I3 or I4 and loaded into the dot pattern buffer. (It is transferred to the dot pattern shift register by the next start trigger.)

d. While the new character is being acquired, the old dot pattern is being shifted to the CRT by 2MHz shift pulses.

Tape Control Circuits.

Up to eight cartridge tape drives can be operated by the Model 1501 Intelligent Terminal; two are standard and six are optional. The optional drives are added in pairs in the Model 1533 Dual Drive, which is a separate unit. The tape itself is very narrow (0.15 inches wide), and it records only one channel of serial data. A small cartridge holds 100 feet of this tape. The cartridge is installed in a simple tape drive. A description of the tape drive appears later in this section.

Recording Method

Before the tape control circuits can be discussed, the recording method and format must be described. The data is recorded in the Manchester Code, which is also called phase encoding. In this method, a signal transition always occurs in the center of a bit period regardless of whether the bit is a one or a zero. The direction of the transition indicates the bit value; a positive-going transition indicates a one and a negative-going transition indicates a zero. Since a signal transition always takes place in the center of a bit period, the timing signals required to process the data when it is read can be developed directly from the signal itself. This allows considerable variation in tape speed with no effect on the data.

Tape Format

Data recorded on the tape is organized into records, each record consisting of 143 bytes, as follows:

a. A 3-byte preamble. Two all-zero bytes followed by a byte with a one in bit position 7 and zeros in positions 0-6.

b. An 8-byte header.

c. 128 data bytes.

d. A CRC (cyclic redundancy character) byte, which is calculated by the processor and is used for error checking.

e. A 3-byte postamble. A byte with a one in position 0 and zeros in positions 1-7, followed by two all zero bytes.

The format of each record is shown in figure 2-38. Note that the tape can be read in either direction. When the tape is read forward, the preamble produces 23 zeros followed by a one; when the tape is read backward, the postamble produces 23 zeros followed by a one. This is the synchronization pattern required by the reading circuits in order to locate the beginning of a record and start reading.

Tape Commands

Input-output instructions are sent from the processor to the I/O controller to control the tape units. In turn, the I/O controller decodes the instructions and sends commands to the tape motion control circuits to manipulate the tape. The input-output instructions related to tape operation and the action that each instruction produces is described in the following paragraphs.

Figure 2-25 illustrates the I/O instructions. Note that the two low-order bits of the IWL select tape unit and that the IWR determines the action to be taken. Each instruction is described below; the number given before the instruction title is the octal value of the IWR for that instruction.

a. 000-Start Tape Forward at Normal Speed with Erase. This instruction is normally given at the beginning of a writing sequence. It starts the tape moving forward at 10 inches per second and erases it until a write instruction is given. Approximately 30 milliseconds are required for the tape to reach a stable operating speed of 10 ips.

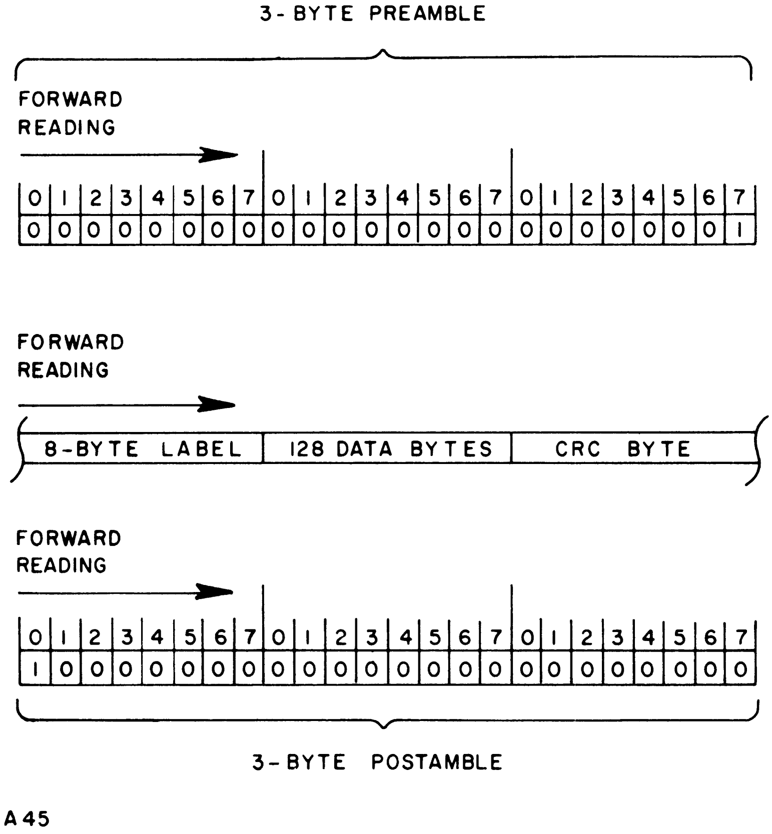


Figure 2-38. Format of Cartridge Tape Record

b. 001-Start Tape Forward at Normal Speed Without Erase. This instruction performs the same function as the preceeding instruction except that the tape is not erased. It is generally given at the beginning of a read sequence.

c. 002-Start Tape Forward at High Speed. This instruction starts the tape moving forward at 40 inches per second.

d. 003-Start Tape in Reverse at Normal Speed. This instruction starts the tape moving in reverse at 10 ips. Erasing is prevented whenever the tape is moving in reverse. If the clip reaches the home position, indicating that the tape is rewound, the tape is stopped.

e. 004-Start Tape in Reverse at High Speed. This instruction performs the same function as the preceeding instruction except that the tape is moved at 40 ips.

f. 005-Stop Tape. This instruction stops the tape.

g. 007 or 207-Transfer Byte. This instruction controls the transfer of characters to and from the tape. The transfer is controlled by a busy (not ready) condition within the tape control circuits, and the instruction can be executed in two modes: (1) stall if busy (007), and (2) skip if busy (207). In the stall-if-busy mode, the program stops at the transfer byte instruction until a tape sprocket signal is generated, indicating that a byte has been written or read. In the skip-if-busy mode, the program automatically skips the next instruction.

h. 010-Set Write Mode. This instruction starts the write operation and begins the timing sequence that controls the writing frequency, loads and shifts the tape buffer, and generates the sprocket signal.

i. 011-Set Read Mode. This instruction activates the tape read circuits. It is executed only once in a normal tape read sequence, and setting of the read condition resets the write condition.

j. 012-Rewind. This instruction sets a rewind flip-flop for the specified tape. This flip-flop is reset only by the clip-in signal, which indicates that the tape is fully rewound. This permits each tape to rewind independently of operations going on in other tape units.

Tape Unit Selection

The circuits that select the tape unit to be used are shown on sheet 2 of the I/O controller logic diagram. Their purpose is to choose one of four possible pairs, and then select one of the two tapes in the pair.

A series of gates and three latches determine which pair of tape units is selected. When power is first applied or whenever the general clear signal is given, these latches are set so that pair 1, the pair of units that is standard in the 1501, is selected. This causes the PAIR-1 signal to be distributed within the tape control circuits.

The pair selected can only be changed by a Read Status I/O instruction that specifies a tape other than the current tape. In this case, the Read Status instruction causes the tape selection circuits to examine positions 3-5 of the IWR (MDR 3-5) and reset the latches based upon the tape pair selected by these bits.

A tape select flip-flop chooses the tape unit within a pair. When the PROGRAM LOAD switch is activated, this flip-flop is switched to the second tape. Thus, when the unit is started, the first program is loaded from the second tape in pair 1.

The state of the tape select flip-flop is then controlled by bits OPO, 1, and 2 of the I/O instructions. When both bits are 0, the flip-flop is left unchanged; thus, the current tape remains selected. A one in bit 0 selects tape A (the first tape) and a one in bit 1 selects tape B by switching the tape select flip-flop to the proper state. Note that a switch on the I/O controller board can reverse the normal selection.

Tape Multiplex Circuits

Shown on sheet 10 of the I/O controller logic diagram is a group of circuits called the tape multiplex circuits. The purpose of these circuits is to control the exchange of data and motion control signals between the I/O controller and the first pair of cartridge tape units. The additional tape units, which are housed in the Model 1533, include a tape multiplexer board that performs the same general function as the tape multiplex circuits shown on sheet 10 do for the first pair.

The tape multiplex circuits are divided into three groups. The top group on the sheet gates data and control signals to the tape unit A; the second group of circuits performs the same function for tape unit B. Shown at the bottom of the sheet is the third group of circuits mentioned above. The purpose of this group is to gate signals from both tape units onto one set of lines to be processed in the I/O controller.

The first group of circuits is enabled when pair 1 is selected and the TAPE A signal is active; the second group is enabled by the combination of the pair 1 signal and an inactive TAPE A signal, which indicates that tape B has been chosen. Since both groups of circuits are identical, only the tape A circuits are discussed.

One of the two control flip-flops in these circuits is the ERASE-A-F/F. When the start-tape-forward-at-normal-speed-with-erase instruction is executed, the SET-ERASE signal is gated to set the erase flip-flop. The gating signal is produced by the fact that pair 1 and tape A have been selected. The erase flip-flop is cleared by the reverse signal or the general clear signal.

The output of the erase flip-flop is combined in two NAND gate write drivers (103 and 106) with the data to be written on the tape. Since there is no data to be written until the write flip-flop is set to gate the data to these circuits, the output of the write drivers is held so that they drive in one direction without change, effectively erasing the tape. When the data is available, it alternates the driver output as described in the discussion of recording. The erase flip-flop is cleared by the general clear signal and any operation that runs the tape in reverse.

A transistor monitors the write line to the tape unit. Whenever an erase or a write operation is begun and the writing circuit is broken, this transistor supplies a high to a 2-input gate and produces the set write fault (SWF) signal. This signal is gated by the write signal to clear the RUN flip-flop and set the RUNAWAY flip-flop.

The next control flip-flop to be discussed is the rewind flip-flop. This flip-flop is set by the Rewind instruction when tape A of pair 1 is selected. It is cleared either by the general clear signal or the CLIP-IN-A signal, which indicates that tape has fully rewound and the clip has returned to the home position.

Whenever the rewind flip-flop is set, it produces three commands to tape unit A. They are: REV-A (reverse), THS-A (tape high speed), and RUN-A. These commands can also be activated by signals from the control circuits on sheet 2; however, the rewind flip-flop produces them directly. This allows the control circuits to send a rewind command for one tape unit and permits that tape unit to accomplish the rewind operation while the control circuits deal with another tape unit.

The three commands mentioned above (REV-A, THS-A, and RUN-A) and a fourth command TCA (tape control A) can be produced by signals from the control circuits, as follows:

a. REV-A is produced whenever the forward-reverse flip-flop is set to reverse and tape A and pair 1 are selected.

b. THS-A is produced directly from the high-speed flip-flop output.

c. RUN-A and TCA are produced when the RUN flip-flop is set and tape A of pair 1 is selected.

The circuits shown at the bottom of sheet 10 process the following signals:

a. CART-IN (cartridge in). One of these signals, which indicates that the tape cartridge is in place, is available from each tape unit. The pair 1 selected and the tape A or B signal gates the CART-IN signal from the active tape to the single CART-OUT line to the accumulator on sheet 1, where it becomes bit 4 of the status word.

b. CLIP-IN. This signal is handled in the same way as the CART-IN signal mentioned above. It indicates that the tape clip is in the home (fully rewound) position. A CLIP-OUT signal is gated to the accumulator to become bit 5 of the status word and to the control circuits where it clears the RUN flip-flop if the tape is operating in reverse and the clip reaches the home position.

c. EOT (end of tape). An end-of-tape marker is placed approximately 10 inches from the end of the tape supply. This signal is gated from the active tape to the accumulator where it becomes bit 6 of the status word.

d. DATA-Q and PULSED ENERGY. These signals are the result of sensing data on the active tape, and they originate in the tape amplifier. Their use is discussed in the description of the read circuits. Here in the tape multiplex circuit they are simply accepted and gated to the read circuits whenever pair 1 is selected.

Write Clock

Timing of the data written on the tape is developed by two interconnected shift registers shown in the lower left corner of sheet 3 of the I/O controller logic diagram. The TX signal from the processor, which occurs once every microsecond, clocks one of the registers. This register is connected so that the output of its most significant stage is an 8-microsecond squarewave. Thus, the first register devides by eight. This is used as the clock input to the second register, which is also connected to divide by eight. Thus, the output of the most significant stage of the second register is a 64-microsecond squarewave. Note that the write clock runs continuously whether or not the output is used.

Writing on the Tape

After the instruction to move the tape is given, a short delay is introduced by the processor program to allow the tape to reach its operating speed of 10 ips. Then the Set Write instruction is executed; this sets the write flip-flop (110/111). The output of the write flip-flop gates the output of the write clock to step the bit counter (141) and to shift the tape data shift register shown on sheet 1. This starts the byte held in the shift register moving serially into the phase encoder shown in the lower center of sheet 3.

The phase encoder is made up of a flip-flop (166) and two NAND gates (164 and 165). Applied to one input of each NAND gate is the write clock signal. Applied to the second inputs of these NAND gates is the data bit (WR-DATA) to be encoded, positive (P-WR-DATA) at one NAND gate and inverted (N-WR-DATA) at the other.

Timing diagram 2-39 shows the write clock signal (D13-6), the P-T-SHIFT signal (which shifts the data serially to the phase encoder), the bit periods and their contents, and the WR-DATA signals that this bit configuration produces. When WR-DATA signals and the write clock are applied to the two NAND gates in the phase encoder, they produce the outputs shown in the timing diagram. To encode a zero, NAND gate 164 supplies a direct set signal to the phase encoder flip-flop 166; to encode a one, NAND gate 165 supplies a direct reset signal to the phase encoder flip-flop. The effect of these signals on the phase encoder output (D10-9, P-DATA-IN) is shown in figure 2-39.

Two other signals are applied to the phase encoder flip-flop:

a. The P-WR-DATA signal is applied to the D-input of the flip-flop. As shown in the timing diagram, this signal is high when a one is being encoded and is low when a zero is being encoded.

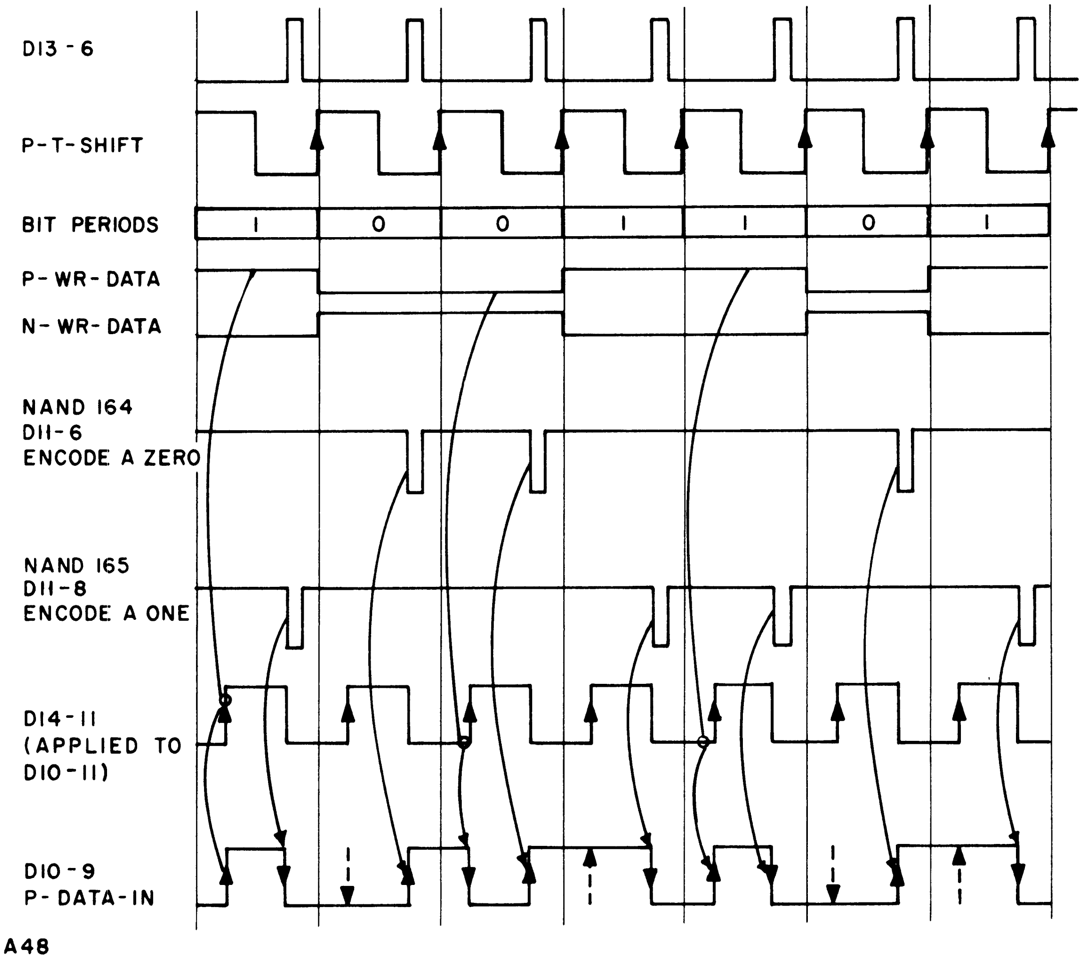


Figure 2-39. Write Operation Timing Diagram

b. The write clock signal from D14-11 is applied to the clock input of the phase encoder flip-flop. On every positive going transition, this signal transfers the value of the P-WR-DATA signal to the phase encoder flip-flop.

The effect of the clocked input on the phase encoder is shown in the timing diagram. Note that it produces the transition in the P-DATA-IN signal when two bits of the same value (1 or 0) are transmitted in succession. When the bit value changes from one bit period to the next, the transition in the P-DATA-IN signal is caused by the direct set and reset inputs.

The bit counter on sheet 3 counts the signals that shift the tape shift register and, when eight bits have been counted, it generates the TP-SPR (tape sprocket) signal. This transfers the next byte to be written from the accumulator to the tape shift register, and it in turn is shifted through the phase encoder.

Writing continues as long as the write flip-flop remains set. The WRITE-F/F signal gates the write clock signal to step the bit counter and to produce the shift pulses for the tape shift register. It also gates the data from the phase encoder to the tape.

The way in which writing is terminated depends upon the program. However, there are three ways in which the write flip-flop can be cleared: (1) general clear (N-GEN-CLEAR), (2) read flip-flop is set, or (3) tape is switched from forward to reverse.

Reading from the Tape

A brief review of the recording technique and format is required before the reading process is described. As discussed earlier in this section, data is recorded on the cartridge tapes in phase-encoded form, and in order to distinguish between the significant and non-significant transitions a start pattern is required to allow the reading circuits to establish synchronization with the data. When the tape is read in the forward direction, the start pattern is called the preamble.

The preamble, which consists of 23 zeros followed by a one, performs three functions:

a. It allows the reading circuits to synchronize reading with the tape flux changes.

b. It allows the reading circuits to distinguish between a data record and tape noise.

c. It allows the reading circuits to locate the first bit of data on the tape.

If the tape is read while moving in reverse, the postamble appears in the same form as the preamble, 23 zeros followed by a one. However, the primary purpose of the postamble is to allow proper reading of the last bit in the record when the tape is being read forward. If the postamble were not used, the last bit of the record would take on undesirable characteristics due to the lack of flux changes following the last bit. Essentially, the postamble terminates the record so that the last bit of valid data can be recovered properly.

In the tape amplifier, the data read from the tape is routed through two separate paths. The first path is the energy detector circuit, the purpose of which is to determine the presence of a signal on the tape. The signal from this circuit is labeled PULSED-ENERGY. The second path is a data amplifier and its output is labeled DATA-Q. The reason for routing the signal read from tape through two circuit paths is discussed below.

In phase encoding, the information is carried by the transitions of the waveform, rather than by the amplitude. The signal read from the tape is amplified and clipped and is therefore relatively immune to amplitude variations. At this high gain, the background noise in the absence of tape signals is sufficient to make it difficult to determine when legitimate data is present. To solve this problem, a separate, limited-gain path is provided after the second stage of an amplification. This channel feeds an energy detector which responds to signals of pre-determined amplitude and duration: greater than 30 percent of nominal signal amplitude and repeating within 250 microseconds for at least 1 millisecond. The circuits in the tape amplifier determine that the amplitude requirements are met, while the tape reading circuits in the I/O controller determine that the pulsed energy signal meets the timing requirements. Only when both the amplitude and timing requirements are met are the reading circuits allowed to recognize the DATA-Q signal.

Reading is ordinarily done with the tape moving forward. In this case, the processor sends the Start-Forward-at-Normal-Speed-Without-Erase command to the I/O controller, and the motion control circuits start the tape moving forward at 10 ips. If the Start-Tape-Reverse-at-Normal-Speed command is given, the motion control circuits move the tape in reverse at 10 ips. Erasing is automatically prevented when the tape is moving backward.

After the tape reaches operating speed and the Start Read command is given to set the read flip-flop, the next step is to synchronize the operation of the reading circuits with the data on the tape. This is accomplished by the circuits on sheet 4 of the I/O controller logic diagram and is shown in the timing diagram in figure 2-40.

The first circuit involved is the PULSED-ENERGY one-shot (101, F12-9). This one-shot is initially cleared by the general clear signal, and it remains cleared until the tape amplifier detects the presence of data on the tape. This is the beginning of a record to be read, and the pulsed energy signal is shown on the first line of the timing diagram.

Each negative-going transition triggers the pulsed energy one-shot. While the first 23 zeros in the preamble are being read, this occurs every 64 microseconds (assuming that the tape speed is the nominal 10 ips). Thus, the output of the PULSED ENERGY one-shot switches and remains in the active state as long as data is not interrupted. This is shown on the third line of the timing diagram.

The next one-shot is the ENERGY one-shot (102, F12-7). Before data on the tape is detected by the PULSED ENERGY one-shot, 102 is triggered every microsecond by the TX signal. This keeps the ENERGY one-shot on

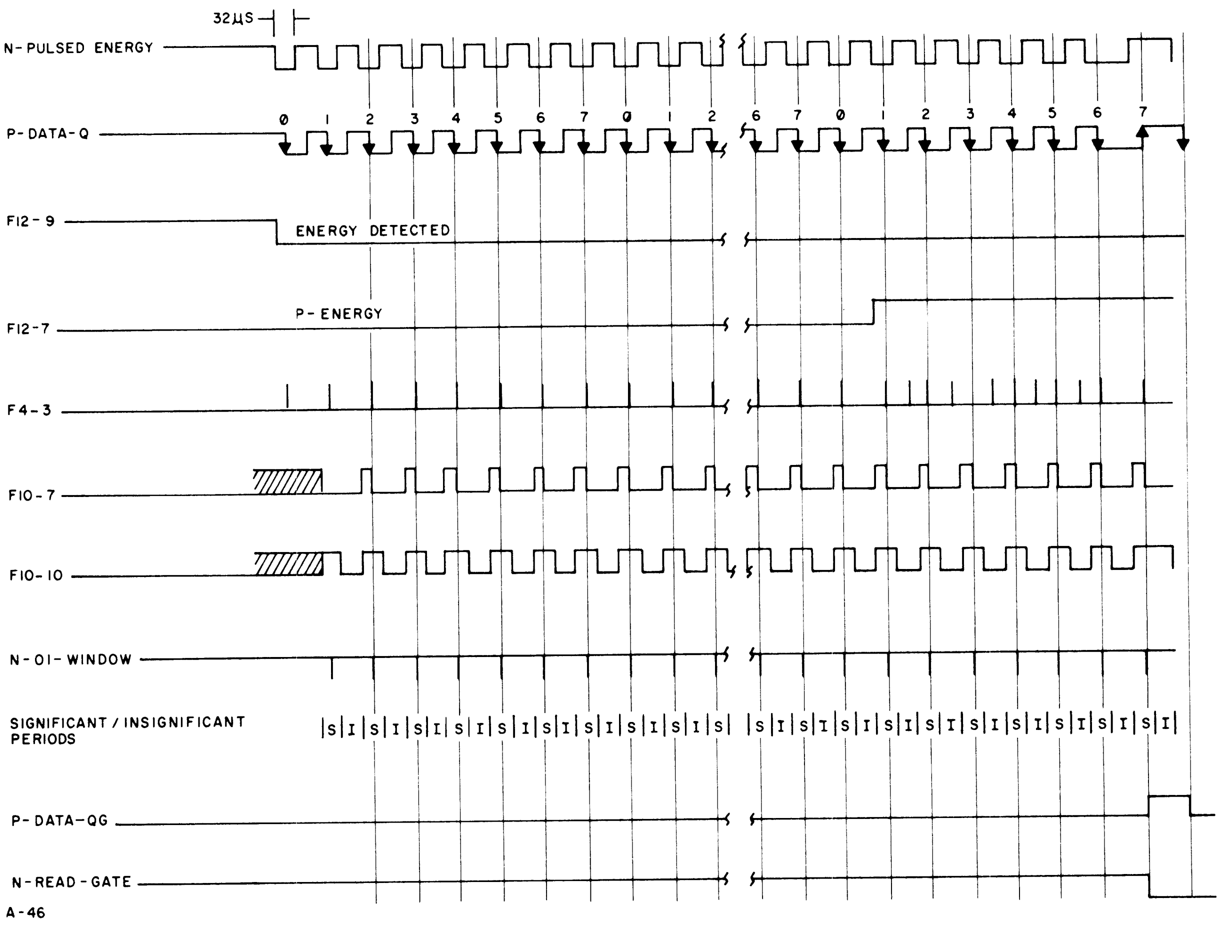


Figure 2-40. Read Operation Timing Diagram

continuously. After the PULSED ENERGY one-shot detects the pulsed energy signal, its output also triggers the ENERGY one-shot by applying a steady low to the active low input. This allows the delay to expire, and the output at F12-7 makes the transition shown in figure 2-40. This transition sets ENERGY flip-flop 105.

The data from the tape (DATA-Q) is routed through two gates to the transition detectors. Both the P-TF (tape forward) and N-TF (tape reverse) signals are used as gating signals, thus the tape data can reach the detectors during either forward or reverse tape motion.

The transition detectors are one-shot multivibrators arranged so that a positive-going transition triggers the ones detector and negative-going transition triggers the zeros detector. When the transition occurs, the detector output switches. The output of the transition detectors are ORed in gate 121 so that either transition shifts the output of gate 121 from a low to a high.

At this point, the effect of the ENERGY flip-flop 105 on the transition detectors is discussed. When the read flip-flop is not set, the READ-F/F signal holds the ENERGY flip-flop 105 in the reset state. This holds a low on the reset input of the ones transition detector (122) and prevents ones from triggering the one-shot. When the read flip-flop is set, ENERGY flip-flop 105 is no longer held off and it can be set when energy is detected by one-shots 101 and 102 discussed earlier.

When the tape amplifier detects the beginning of a record, the pulsed energy signal, shown in line 1 of the timing diagram, is present. This triggers the pulsed energy one-shot 101 and retriggers it regularly, thus switching pin 9 to a low and keeping it there unless a gap is detected.

The signal at F12-9 is applied to pin 5 of ENERGY one-shot 102. The negative-going transition triggers this one-shot, but since there are no more transitions, the delay expires and the output of 102 (F12-7) switches. This is shown in line 4 of the timing diagram. Thus, once the beginning of a record has been detected, 102 is reset and remains reset.

When 102 is first reset, the positive-going transition provides the clock pulse required to set ENERGY flip-flop 105. The P-ENERGY F/F signal thus goes high and removes the low that held the ones transition detector off.

Before energy was detected, F12-9 and F11-6 were high, producing a low at F4-11 and a high at F3-10. This gates the write clock through gate 113. This signal passes through gate 114 and triggers the tracking window one-shots 127 and 128, forming the tracking window in synchronism with the signal (the write clock) that was used to write the tape.

When energy is detected, this path is blocked and the zero transitions are allowed to trigger the tracking window one-shots until the energy flip-flop is set. Then, the path through gate 118 is blocked.

Gate 120 now becomes the path to trigger the tracking window one-shots. If either a one or a zero transition occurs within the window, it retriggers the tracking window one-shots. If no transition occurs within the window, the phase error flip-flop (132) is set and the tape error F/F signal is produced at gate 133A.

Each time the signal N-01 WINDOW signal is produced, it causes the P-T-SHIFT signal to occur. This shifts the incoming serial data through the tape shift register. A NAND gate monitors the high and low order bits of the register. When this gate senses the end of the third byte of the preamble, it produces the SET-READ-GATE signal. This sets the TAPE-READ-GATE-F/F on sheet 4, which was cleared when the read operation began, and produces the read gate signal.

When the read gate signal is active, it allows the N-01-WINDOW signal that produces shift pulses to also step the bit counter. Every 8 bits, this counter produces a byte complete signal. In turn, this signal transfers the byte in the tape shift register to the tape data buffer and produces the TP-SPR (tape sprocket) signal to notify the processor that a byte is available.

Tape I/O Stall Circuits

The keyboard stall circuits were discussed earlier in this section, and the tape I/O stall circuits operate in a similar manner. The primary difference is that the tape I/O stall circuits operate when the processor is writing on the tape as well as reading from it. Of course, the keyboard stall circuits operate only when the processor is reading the keyboard input.

Shown on sheet 5 of the I/O controller logic diagram are the circuits that generate the I/O stall signal sent to the processor. The action that the processor takes on the stall signal depends upon whether the Transfer Byte instruction is 007, Transfer Byte-Unconditional, or 207, Transfer Byte-Skip on Busy. In the former, the processor remains at the current instruction until the byte is transferred. In the latter, the processor skips one instruction and proceeds. Another Transfer Byte instruction must then be executed to transfer the byte. The timing of when this instruction occurs is determined by the program.

The circuits that generate the I/O stall signal for tape data transfer are shown in the center of sheet 5, and the I/O stall flip-flop is shown at the right. Note that the I/O stall flip-flop can be set by any one of three sources (keyboard, tape, or serial I/O channel).

The TX signal from the processor samples NAND gate 135. If a Transfer Byte instruction is being executed and a tape unit has been selected, the condition of flip-flop 137 determines whether or not the I/O stall flip-flop will be set.

The TP-SPR (tape sprocket) signal indicates that a tape byte has been processed. In a reading operation, this indicates that a byte is ready for the processor; in the writing operation, it indicates that the tape can accept the next byte from the processor. Applied to the clock input flip-flop 123, the tape sprocket signal sets this flip-flop to record the fact that a byte has been processed.

NAND gate 124, which monitors the state of flip-flop 123, is checked at TE of cycle I3. If flip-flop 123 was set, this NAND gate sets flip-flop 137. Then, at TX time, the I/O stall flip-flop cannot be set. If, however, the tape sprocket signal had not set flip-flop 123, flip-flop 137 is still cleared at TX time and the I/O stall flip-flop is set.

Assuming that flip-flop 137 was set, the I/O clear pulse, which follows TX, enables NAND gate 136. This clears both flip-flops, 123 and 137, returning the two flip-flops to their original condition in preparation for the next tape sprocket signal and Transfer Byte instruction.

Assume next that a tape sprocket signal had not set 123 and consequently 137 was also left in the cleared condition. In this case, the I/O stall flip-flop was set and the processor reacted as described earlier. The tape sprocket then arrives. It sets 123, and then 137 is set. Either the current Transfer Byte instruction (if the processor stopped) or the next Transfer Byte instruction (if the processor skipped) transfers the byte and clears both flip-flops. Note also that the P-W-OR-R (write or read) signal clears both flip-flops when it goes low at the end of a write or read operation.

The I/O stall flip-flop is cleared by the general clear signal and at the beginning of every I3 cycle. I3 is the cycle during which an instruction is read from memory. Clearing the stall flip-flop at the beginning of this cycle has the effect of removing the conditions that were used in the previous instruction.

Serial I/O Channel Circuits.

Sheets 7, 8, and 9 of the I/O controller logic diagram make up the serial I/O channel control circuits. These circuits transfer data between the Intelligent Terminal and any of the other devices attached to the SIO coaxial cable. Up to 64 units can be attached to the cable.

Information on the SIO line is in phase-encoded form, and the data is self-synchronizing in that the receiver regenerates the timing signal from the phase-encoded data. During the phase-encoded communication, information is transmitted as energy level changes (transitions) along the coaxial cable. Positive-going transitions signify one-bits; negative-going transitions signify zero-bits. Obviously, intermediate transitions must take place between successive bits. The intermediate transitions are termed "insignificant" transitions, whereas information-bearing transitions are termed "significant" transitions. Receiver circuits differentiate between the significant and insignificant transitions. (Refer to figure 2-41 for a diagram showing a typical data transmission.)

The intelligent terminals connected to the I/O cable may be operated in either of two modes, master or slave. Only a unit operating in the master mode can initiate transmission. Selection of master or slave modes of operation is a function of the processor program.

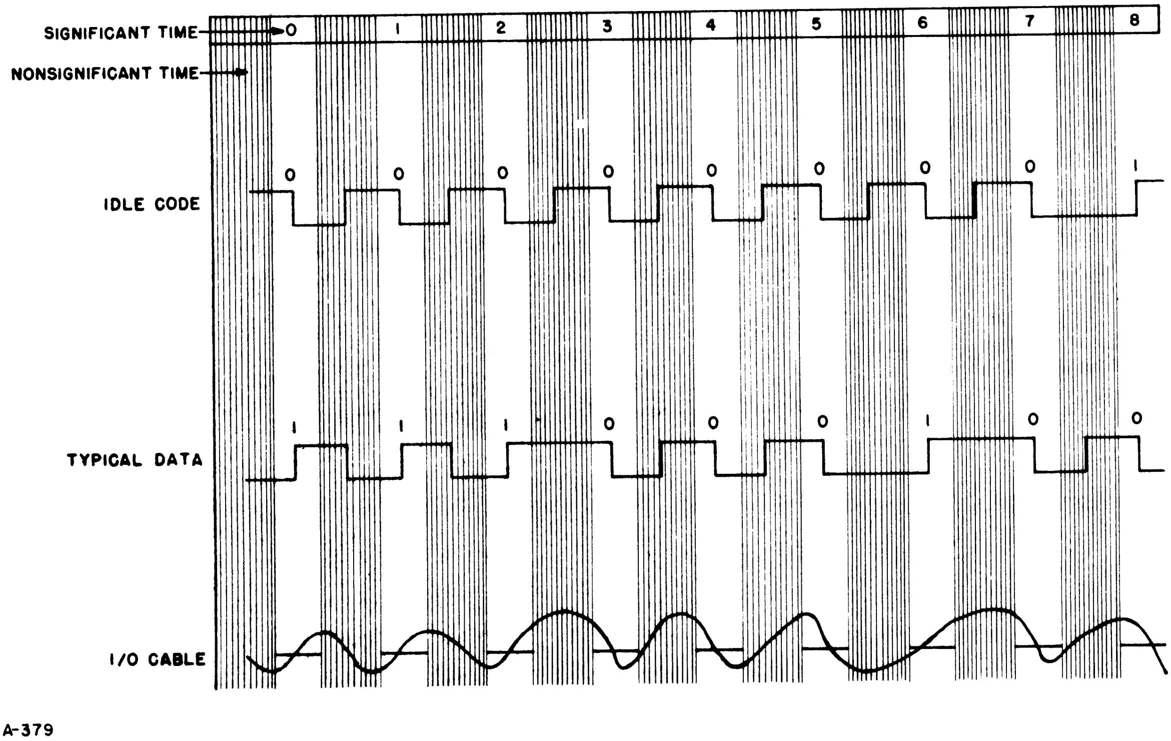
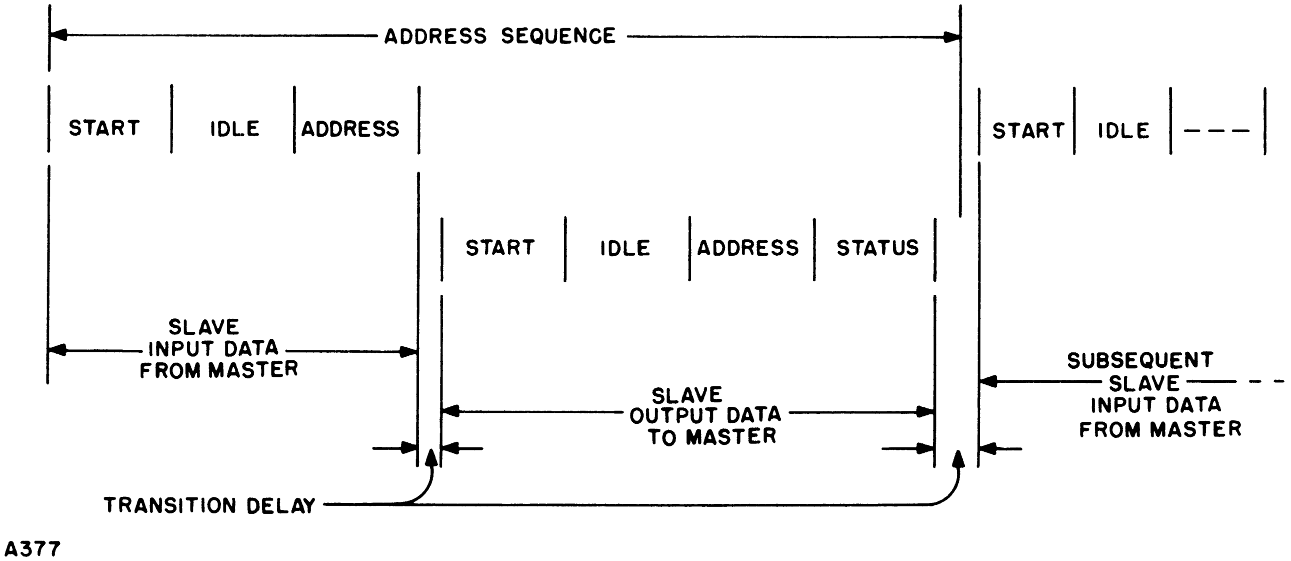


Figure 2-41. Phase Encoded Signals

Since the serial I/O channel is a "party line" with the possibility of several intelligent terminals and a number of input/output devices being simultaneously connected to the line, it is necessary for each device to be identified by a unique address. Communication over the SIO cable takes place between a master device and a slave device. An address sequence, which is shown in figure 2-42 is used to establish communication between these two units. The address sequence consists of an initial transmission of start and idle codes by the terminal operating in the master mode. This synchronizes all receivers connected to the line. The idle codes are followed by the transmission of the address of the slave device with which communication is to be established.

The slave device that recognizes its address is set to an active state. Once in an active state, the cessation of transmission by the master causes the slave to transmit the start and idle code, echo its address, and send its status. The transmission of a status byte by a slave device automatically causes the slave transmission to stop.


ADDRESS SEQUENCE
START
IDLE
ADDRESS
START
IDLE

START
IDLE
ADDRESS
STATUS
SLAVE INPUT DATA FROM MASTER
SLAVE OUTPUT DATA TO MASTER
SUBSEQUENT SLAVE INPUT DATA FROM MASTER
TRANSITION DELAY
A377

Figure 2-42. Address Sequence

As the address echo is received by the master, it is compared to the address that was transmitted to ensure that communication with the proper device has been established. The status byte is transferred to the master terminal's accumulator for use by the program. The cessation of transmission by the slave causes the master to resume transmission, starting with start and idle codes in preparation for the transmission of commands and/or data.

A block diagram of the SIO control circuits is shown in figure 2-43. The extra width lines represent the main signal paths, and a brief description of the data flow along these paths is presented in the following paragraphs.

The I/O data is transferred on the serial I/O line. This line has a common connection with both the receive and transmit circuits on the system reset PCB.

The I/O data signal received is applied to the receiver amplifier where it is amplified and squared so that it remains phase-encoded but now has sharp positive- and negative-going transitions. With cable losses restored, the output signal now operates between the logic levels of 0 and +5 volts. The phase decoder detects the transitions, both positive- and negative-going, to establish timing, and counts these transitions to establish synchronization with the sending unit and to produce logic levels corresponding to the transitions. The output of the phase decoder

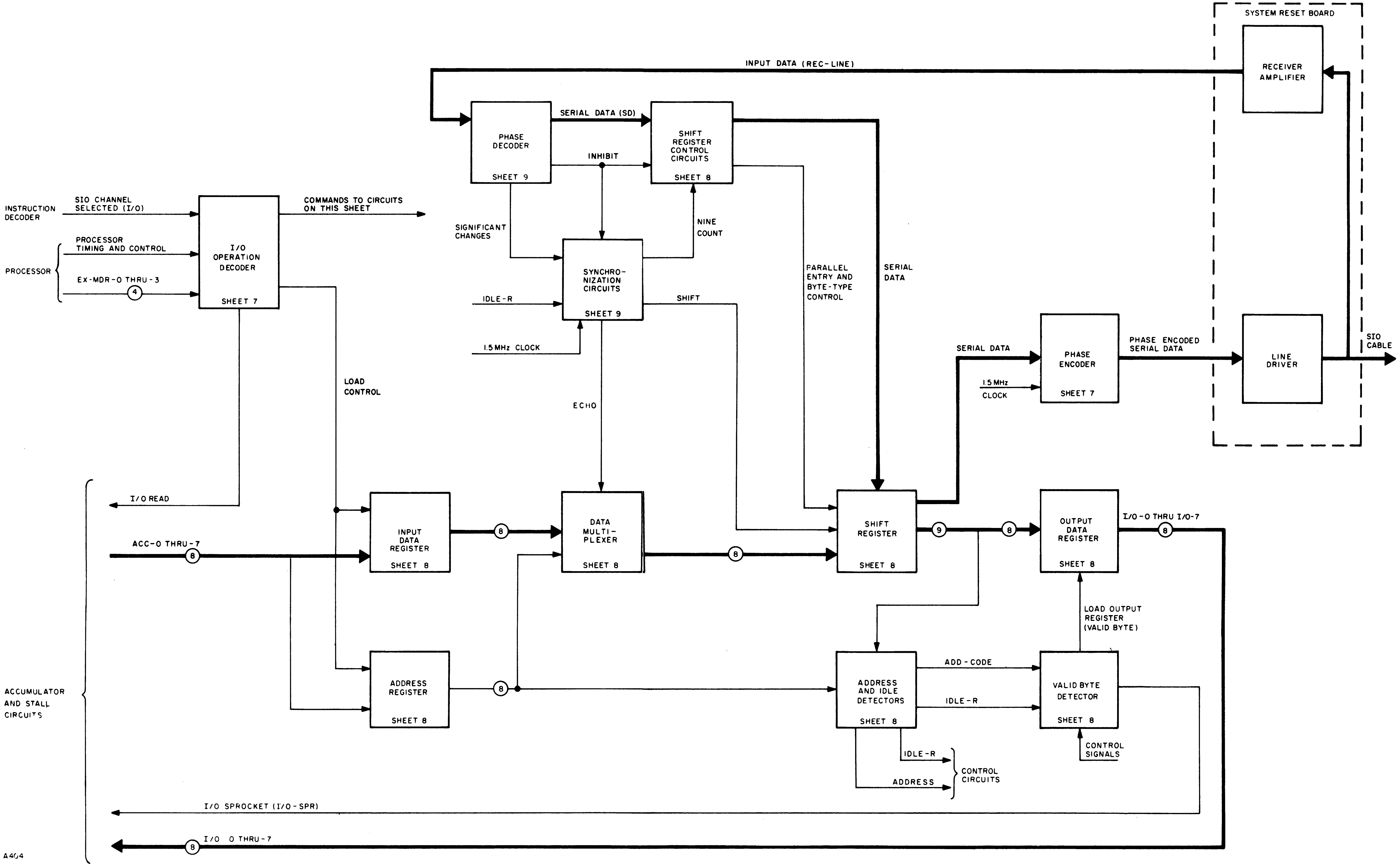


Figure 2-43. SIO Control Circuits Block Diagram

is serially shifted into the I/O shift register. Data that is received and is to be transferred to the accumulator is gated into the output register from the I/O shift register after it is determined that a complete byte is ready.

When data is to be transmitted, the appropriate data is entered into the input multiplexer. There are also times when the address is to be transmitted and therefore must be entered into the input multiplexer. The sync circuit determines which of the inputs is gated, in parallel, into the I/O shift register. The sync circuit also controls the serial shift of the input data from the I/O shift register into the phase encoder. This circuit converts the standard logic levels into phase-encoded data that is applied to the line driver on the system reset PCB. This driver provides the signal level required to drive up to 1000 feet of the I/O cable.

Transmission Techniques

Communication between an Intelligent Terminal and a peripheral unit is accomplished by a group of alternating transmission sequences. Figure 2-44 shows an exchange between an Intelligent Terminal and a Model 1511 magnetic tape unit which, of course, is operating in the slave mode. The number of transmissions in a group depend on the operation. The significant points of this communication techniques are:

a. One of the devices is always transmitting except for a transition time of approximately one microsecond.

b. A device transmits idle bytes when not transmitting an information byte.

c. A device stops transmitting at the end of its sequence.

d. A device starts transmitting as a result of detecting that the other device has stopped transmitting.

Organization of Data on the SIO Line

Six types of bytes can be transmitted over the SIO channel. They are: data bytes, command bytes, address bytes, status bytes, start bytes, and an idle byte used for synchronization. Each byte transmitted is supplemented by a hardware-generated ninth bit to distinguish between data and the other types of information. The ninth bit is zero for start and data bytes, and is a one for all other types of bytes.

The content and use of each type of byte is as follows:

a. Start Byte. The start byte is used together with an idle byte for synchronization and timing. It is always the first byte of a transmission sequence, and is not used in any other way. The start byte contains all zeros.

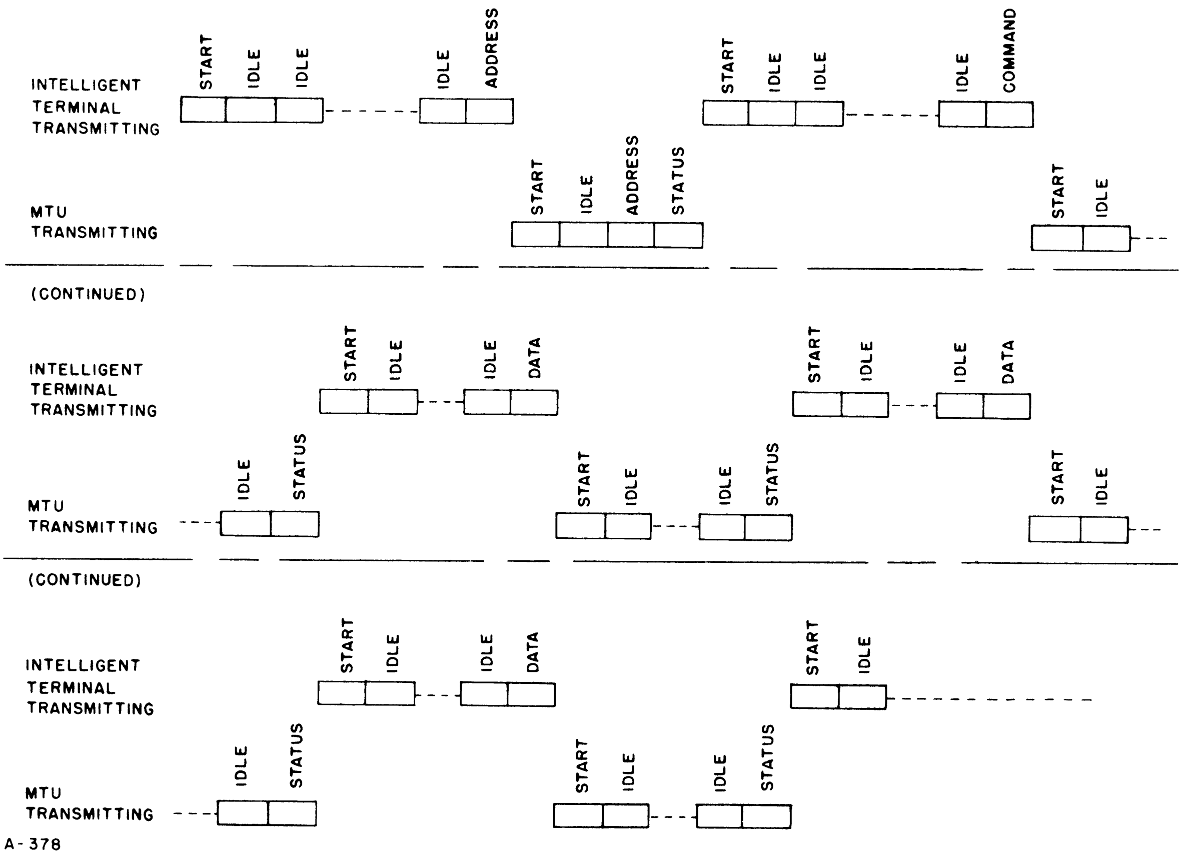


Figure 2-44. Typical Exchange Between Terminal (Master) and Magnetic Tape Unit (Slave)

b. Idle Byte. The idle byte is used for synchonization and timing. It is used together with a start byte at the beginning of a transmission sequence and it is used after that to maintain the synchronization between the master and slave units when there is no other data being transmitted. An idle byte has the form shown in figure 2-45.

c. Status Byte. A status byte has a one in position eight. It is differentiated from an address byte (which also has a one in position eight) by the remaining bits in the byte. The status byte has the form shown in figure 2-46 and indicates any of several different conditions, one for each bit. It is always transmitted by the slave as the last byte of the transmission sequence.

d. Address Byte. An address byte has ones in bit positions 7 and 8, and a zero in bit position 6. The remaining 6 bits specify an address. Thus, up to 64 addressable devices may be placed on a single line. An address byte has the form shown in figure 2-47.

BIT NUMBER	8 (MSB)	7	6	5	4	3	2	1	0
CODE	1	0	0	0	0	0	0	0	0

A-385

Figure 2-45. Idle Byte Format

BIT NUMBER	8	7	6	5	4	3	2	1	0
	1	0	1	STATUS					

A-407

Figure 2-46. Status Byte Format

BIT NUMBER	8 (MSB)	7	6	5	4	3	2	1	0
CODE	1	1	0	ADDRESS					

A-406

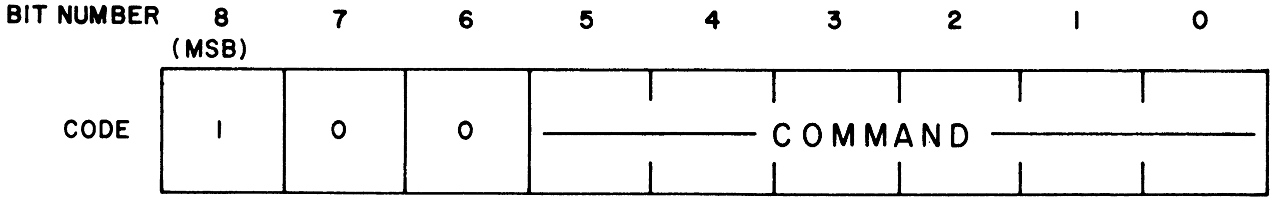
Figure 2-47. Address Byte Format

e. Command Byte. A command byte is characterized by a one in bit 8 and zeros in bits 7 and 6. The remaining six bits specify a particular command. A command byte has the form shown in figure 2-48.

f. Data Byte. The data byte is characterized by a zero in bit position 8. The remaining bits carry the data. A data byte has the form shown in figure 2-49.

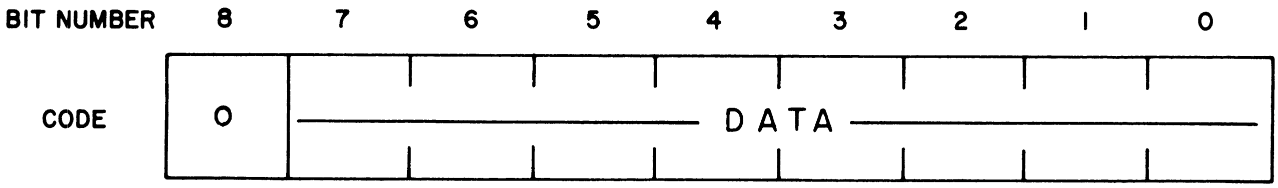
SIO Channel Instructions

The Intelligent Terminal controls the SIO channel with I/O instructions in the 175 family. These instructions are listed in table 2-6. When the processor decodes an instruction in the 17X family, it sends an I/O operation signal to the I/O controller. In turn, the controller decodes the low-order bits, "5" in this case, to determine which I/O device is selected. When the SIO channel is selected by the "5" in these bits, the I/O controller sends a selection signal to the SIO control circuits. This allows



A-381

Figure 2-48. Command Byte Format



A-380

Figure 2-49. Data Byte Format

Table 2-6. SIO Channel Instructions

Octal Instruction Code	Instruction
175-000	Start Transmit*
175-001	Transfer Byte, Unconditional
175-201	Transfer Byte, Skip On Busy
175-002	Transmit Data Byte
175-003	Transmit Control Byte
175-004	Stop Transmit*
175-006	Inhibit Line*
175-007	Set Device Address
175-010	Set Master Mode
175-011	Set Slave Mode

*Executed by Master Only

the SIO control circuits to decode the right half of the instruction and take whatever action is required. The action taken for each of the ten SIO instructions is as follows:

a. The Start Transmit instruction causes an Intelligent Terminal operating in the master mode to initiate transmission. This instruction also releases the clamp if an Inhibit Line instruction has been previously executed.

b. The Transfer Byte instruction is used to accept a byte into the accumulator. It performs several functions, depending upon which part of an SIO communication is in progress when the instruction is executed. Normally, the Transfer Byte instruction causes the processor to stall if the SIO control circuits have not sent a sprocket signal indicating that a byte is available.

If the Intelligent Terminal is operating in the master mode and transmission is in progress during which at least one Transmit Data Byte or Transmit Control Byte instruction has been executed, the Transfer Byte instruction causes the transmission to stop, and the active slave should respond. If the terminal is in the slave mode, the Transfer

Byte instruction has no effect on the transmission, but causes the processor to stall until a sprocket signal is received. If the SIO channel is not active but the terminal is in the master mode, the Transfer Byte instruction causes the generation of a dummy sprocket signal. In either case, once a sprocket has been sent, the Transfer Byte instruction causes the contents of the SIO output data register to be transferred to the accumulator, and the processor leaves the stall condition, allowing further instructions to be executed.

The skip-on-busy feature is used with the Transfer Byte instruction to avoid the stall condition anytime the programmer knows that the reply from the slave will not occur for a considerable period of time. Since the Transfer Byte instruction is used to initiate the handover of transmission from the master to the slave, there is always an interval of about 20 microseconds between the time the Transfer Byte instruction is first executed and when a sprocket signal may occur. The occurrance of additional sprocket signals is dependent upon the particular program or devices being utilized.

c. A Transmit Data Byte instruction causes the contents of the accumulator to be transferred to the input data register and transmitted on the I/O channel.

d. A Transmit Control Byte performs the same function as the Transmit Data Byte instruction, but the SIO circuits distinguish between the two instruction types by coding the ninth bit differently.

e. A Stop Transmit instruction is used by the master to terminate a communication exchange after the next transmission of status by the slave. Stop Transmit may be executed at any time during the final transmission of the master.

f. When an Inhibit Line instruction is executed by a master, the coaxial transmission line is driven to a high level and held at that state until a Start Transmit instruction is executed. The Inhibit Line instruction also causes the accumulator contents to be transferred to the input data register, although it is not normally used for this purpose. After execution of the Inhibit Line instruction, every SIO station connected to the coaxial cable is cleared by timing circuits that expire due to lack of further transmission line activity. A period of at least 15 microseconds must be allowed to elapse between the Inhibit Line instruction and the Start Transmit instruction.

g. The Set Device Address instruction transfers the accumulator contents to the address register and input data register of the SIO channel circuits, and it should be executed before the Set Slave instruction is used. Both slave and master must have the address of the slave stored in its address register.

h. The Set Master instruction conditions an Intelligent Terminal to function as the controlling station in an SIO exchange. Only the station that is in master mode can execute the Inhibit Line, Start Transmit, and Stop Transmit instructions.

i. The Set Slave instruction inhibits SIO circuits from responding if Inhibit Line, Start Transmit, or Stop Transmit instructions are executed, and conditions the station to operate as a slave device. Execution of this instruction also transfers the accumulator contents to the input data register of the SIO circuits.

Sequence of Operation in the Master Mode

When an Intelligent Terminal is to be operated as the master unit for SIO communication, it must be programmed to execute the address sequence to control the communication exchange. The instructions that must be executed are as follows:

a. The Set Master Mode instruction is executed first. Following this, the Inhibit Line instruction may be executed. If the Inhibit Line instruction is used, a period of at least 15 microseconds must elapse between the Inhibit Line instruction and the Start Transmit instruction.

b. The next instruction to be executed is the Start Transmit instruction. At this point, it is necessary to set the address register to the address of the device with which communications are to be established. Therefore, the next instruction would be a Load Accumulator instruction (not an I/O instruction) during which the accumulator is loaded with the desired address. This address has the octal form 2XX, where the 2 forms the SIO address byte code and the XX represents the 6 address bits. Once the address is loaded into the accumulator, a Set Device Address instruction is executed, followed by a Transmit Control Byte instruction.

c. The next instruction to be executed is the Transfer Byte instruction. At this point, the program stalls while waiting for the status of the slave device to be transferred into the accumulator. Once this status is received, additional program steps depend on the device with which communication is taking place and on the status condition that has been received.

d. If the status received at the end of the address sequence indicates that the slave device is in a ready condition, the normal procedure is for the master unit to transmit a command to the slave in order to set up a data transfer. Data may be transferred in either a burst mode or in a single character mode of communication. In the burst mode, the sending unit transmits continuously with the data interspersed between idle codes. In using the burst mode of transmission, the characteristics of the devices involved must be considered so that the ability of the receiving unit to assimilate the data is not exceeded. In a single character mode of data transfer, each character is acknowledged by a transmission from the receiving unit. This mode of data transfer is normally used when the time of arrival of each character is critical. In all cases, the transmission of a status byte by a slave will cause its transmission to terminate.

Sequence of Operation in the Slave Mode

When power is turned on, the SIO circuits are set to the slave mode by the general clear signal. This assures that a station will not interfere with SIO activities. In order to employ a device as a slave, it is first necessary to set the address and data registers to a condition that will allow the SIO section to automatically respond to a transmission of the proper address from a master station. Preparation for the address sequence is accomplished by the execution of the following instructions:

a. First, a Load Accumulator instruction with the address that this unit will recognize and respond to is executed. Second, a Set Device Address instruction is executed. This instruction loads the address into the address register and the input data register. A Load Accumulator with status instruction followed by a Set Slave Mode instruction are executed next. The Set Slave Mode instruction ensures that the unit is in the slave condition and transfers the status word from the accumulator to input data register. This status word will be transmitted to the master during the address sequence.

b. If the slave device is ready for SIO communication, the next instruction is a Transfer Byte, which will stall the processor until it is addressed by the master. If the slave is not ready, it places a not-ready status word in the input data register and continues to execute instructions.

c. Since the address sequence for the slave is a strictly SIO function and does not involve the processor, a slave with a not-ready status set in its input data register may be addressed many times without interference with instructions being executed by the processor.

d. A station operating as a slave will normally be stalled in a Transfer Byte instruction waiting for commands from the master. The slave must be programmed to terminate each of its transmission by transmitting a status byte, using the Transmit Control Byte instruction.

Input Data Register

The input data register, which is also shown on sheet 8, is used to transfer accumulator data to the SIO circuits. It accepts the contents of the accumulator when P-LOAD-DATA signal goes high. Instructions that generate P-LOAD-DATA signal are: (1) Set Slave, (2) Transmit Data Byte, (3) Transmit Control Byte, (4) Set Device Address, and (5) Inhibit Line.

Address Register and Detector

An address sequence is used to place a master device, such as an Intelligent Terminal, in communication with a particular slave device, such as a specific tape unit. The start pattern and synchronization operation described previously, synchronized every slave device on the SIO line but did not establish a communication link with a specific slave device. This requires the transmission and acknowledgement of the address for the specific device. The master transmits the desired address and the slave device that recognizes this address as its own, transmits its address back to the master device. The slave immediately follows the address byte with a status byte and then stops transmitting. When the master device receives the same address it transmitted, it knows the communication link is complete and it knows the status of that slave device. What communication follows is controlled by the master device.

For all other slave devices which do not detect their own address, they do not become active and are unable to transmit after detecting the end of reception. They continue to receive the data on the SIO line and load it into the I/O shift register. This data in the register is meaningless because the device is inactive and the data cannot be transferred out of the shift register. A timing diagram of the circuits and signals involved in the address sequence is shown in figure 2-50.

In both a master and slave unit, the address register, which is shown on sheet 8 of the I/O controller logic diagram, is used to store the device address of the slave unit involved in a communication exchange. The address register accepts the contents of the accumulator when a Set Device Address instruction causes the P-LOAD-ADDRESS signal to go high. The master transmits the device address to all slave units in the system. The slaves compare this address to the address held in their address register and the proper slave echoes the address back to the master.

The address detector, which is also shown on sheet 8, performs two functions. First, with gates 135 and 136, it senses the coding of 1-1-0 in an address byte's three high-order bit positions. When this code is detected in the shift register, the N-ADD-CODE signal is generated at gate 136. This signal disables gate 148 to prevent an N-BYTE signal from occurring.

The second function of the address detector is to compare the contents of the shift register with the address held in the address register. The gates that make this comparison generate the P-ADDRESS signal when the addresses match. In a slave, the P-ADDRESS signal sets the active flip-flop on sheet 7, effectively placing the slave on-line to the master and ready to begin transmitting as soon as the master stops. In the master, the P-ADDRESS signal sets the ADDRESS REC-flip-flop in response to a valid address echo from the slave.

Data Multiplexer

The data multiplexer, which is made up of elements 108 and 114 on sheet 8 of the I/O controller logic diagram, selects data from one of two sources for input to the shift register. The address register is selected when P-ECHO signal is high (i.e., when a slave is sending its address back to a master). The input data register is selected as a source at all other times.

Shift Register

The shift register, which is shown on sheet 8, includes two 4-bit shift registers (110 and 113) and a D-flip-flop (109), which is arranged to function as the ninth bit. Three signals control the shift register:

a. P-SHIFT steps serial data (P-SD) from the receiver circuits or parallel data from the data multiplexer through the shift register.

b. N-PARALLEL-ENTRY enables eight bits of data to be entered from the data multiplexer during transmit operations.

c. N-CLEAR resets the shift register.

During reception, the P-SD signal is clocked into the D-flip-flop of the shift register by the P-SHIFT signal and shifted into the J-K inputs of the shift register elements. When a byte has been assembled in the shift register, the N-BYTE signal loads the shift register contents into the output data register.

During transmission, the shift register accepts parallel data from the data multiplexer. The byte transmitted can be an address, a control byte, or a data byte. The ninth bit is a zero for a data byte, a one for all other bytes. Information to be transmitted is sent serially to the phase encoder.

The ninth bit element (109) is set by the P-SHIFT signal if the N-IND-BIT signal from the indicator bit latch (124) is active, signifying non-data transmission. This latch is set when a Transmit Data Byte instruction is executed, thus the N-IND-BIT signal is high for non-data transmissions.

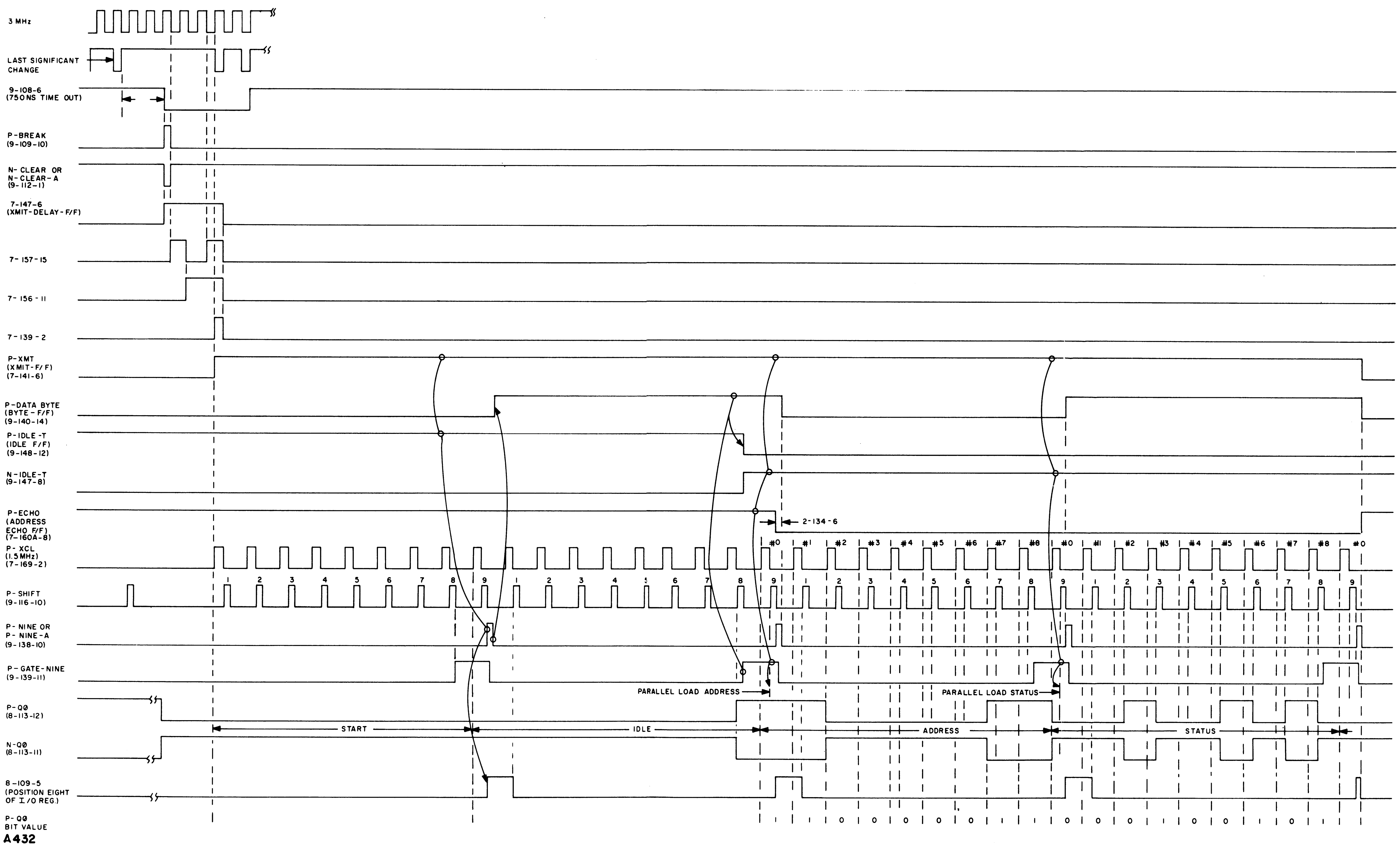


Figure 2-50. Address Sequence Timing Diagram

The shift register also generates the idle code (octal 400) required for transmission of information. The eight zeros are the shift register contents after an information byte has been shifted out for transmission. The ninth-bit one is inserted by presetting the ninth bit flip-flop (109) at P-NINE time.

Shift Register Control Circuits

These circuits, which are shown on sheet 7, are a group of gates that perform the following functions:

a. Enable the decoded received data (P-SD) to be applied to the serial input (position eight) of the I/O shift register for serial shift into the register.

b. Determine if and when there will be a parallel transfer from the input multiplexer into the I/O shift register.

c. Enable specific conditions to determine the byte-type bit (position eight) for transmitted data.

d. Determine when the data to be transmitted is serially shifted out of the register, under control of the shift pulses, and into the phase encoder.

Before receiving any data, the I/O shift register has been cleared and contains all zeros. When the condition levels indicate the unit is not transmitting, the P-SD signal is inverted twice and applied to the serial input of the I/O shift register. The P-SHIFT pulses from the sync circuit shift the data into the register.

In the transmit mode, the shift register control circuits control entry of the byte-type bit into position eight of the register and controls the time for parallel load into, and serial shift out of, the register. This is done for each of the six different types of bytes the processor can supply as input data. The six bytes and corresponding type bits are:

Type of Byte	Type Bit (Position 8)
Start	0
Idle	1
Status	1
Address	1
Command	1
Data	0

The action taken by the control circuits for each type of byte is as follows.

a. Start Byte - At the beginning of operation or after the last data was received, the clear signal cleared the I/O shift register. Thus, it contains all zeros, including position 8. During transmission of the start byte, nothing is done to change this condition. As a result, the start byte consists of nine zeros.

b. Idle Byte - As the start byte is shifted out, the idle byte is being shifted into the I/O shift register and the XMIT-F/F and the IDLE-F/F are set. This means that the N-XMIT-(A) and N-IDLE-T signals applied to the shift register control circuits are both low and the serial data input at position eight of the shift register is low. With each P-SHIFT pulse, the register is loaded with another zero until eight zeros are shifted in. (The N-IDLE-T low level also prevents a parallel transfer of data from the input multiplexer into the shift register.) The ninth shift pulse completes shifting out the start byte and causes the nine counter in the sync circuit to make the P-NINE-A signal active. This sets a one in position eight of the register. With the next shift pulse, which is count number one of the next bit time, the serial output of the idle byte begins.

c. Status, Address, and Command Bytes - During the transmission of these bytes, the shift register is loaded in parallel from the input multiplexer. First, the serial data input is arranged so that all zeros are shifted into the register. Each shift pulse is counted. Between counts eight and nine of the nine counter in the sync circuit, the P-GATE-NINE signal becomes active. The N-PARALLEL-ENTRY signal then goes low, enabling the parallel entry for positions zero through seven. On the next P-SHIFT pulse, the contents of the input multiplexer are transferred into positions zero through seven of the shift register. When the N-PARALLEL-ENTRY signal became active, the N-IND-BIT was gated to the serial data input to position eight. Since this bit is a one for all except data bytes, a one is placed in position 8 by the same P-SHIFT signal that transferred the parallel input to the register.

d. Data Byte - The shift register control circuit produces the parallel transfer of data from the input multiplexer into the shift register in the same manner as for status, address, and command bytes. In the case of data bytes, however, the IND-BIT is a zero, and thus a zero is entered in position 8.

Idle Detector

The idle detector, which is shown on sheet 8, monitors all nine bits of the shift register. When a pattern of eight zeros and a one in the ninth bit (the idle code) is sensed, the idle detector generates the N-IDLE-R (idle received) and the P-IDLE-R signals. In turn, the N-IDLE-R signal prevents gate 148 from generating the N-BYTE signal thereby preventing the output data register from accepting an input from the shift register and disabling the I/O sprocket one-shot. The processor, therefore, ignores the idle code. During reception, the P-IDLE-R signal sets the sync-flip-flop (element 135 on sheet 9), which enables the nine counter (133 on sheet 9). This indicates that synchronization with the start pattern has been achieved.

Output Data Register

The output data register shown on sheet 8 receives data from the parallel outputs of the shift register and makes it available to the accumulator. Data available at its inputs is sensed when the N-BYTE signal is active. The N-BYTE signal also triggers the I/O sprocket one-shot to signal the processor that a byte is available at the output data register.

Since the output data register must function for both control bytes and status bytes, in addition to data bytes, bit 6 (N-REGISTER-6) and bit 7 (N-REGISTER-7) must pass certain tests before reaching the accumulator. I/O bit 6 is gated to the accumulator only if a status byte or a data byte is sensed. I/O bit 7 is used to indicate I/O errors, and is combined at gate 143 with the N-ERROR signal before reaching the accumulator.

Output Data Control Circuits

Command, status, and data bytes must be transferred from the I/O shift register to the accumulator. The content of the shift register is a valid command, status, or data byte if all the signals applied to NAND gate 148 on sheet 8 are high. This requires the following conditions:

a. Shift register content is not a start byte, idle byte, or address.

b. All nine bits have been shifted into the register.

c. No receiver error has been detected.

d. The unit is not transmitting.

e. The unit is active - a specific communication has been established. (The unit is either a master or a slave that has been addressed.)

When these conditions are met, a valid byte signal is applied to the output register and the eight bits in the I/O shift register are transferred to the output register. The control bit in position 8 is used differently. When the valid byte signal (BYTE) is active, it transfers the control bit to the I/O-STATUS flip-flop (144 on sheet 8). This activates the I/O-STATUS signal to the accumulator, and sets position 2 of the status word. A one in this position indicates a command or status byte and a zero indicates a data byte.

SIO Sprocket and Stall Circuits

The purpose of these circuits is to coordinate the transfer of data between the processor and the SIO channel. There are three groups of circuits involved:

a. The I/O stall circuits, which are shown at the bottom of sheet 5 of the I/O controller logic diagram.

b. The I/O sprocket one-shot on the right side of sheet 8.

c. The portion of the I/O instruction decoder (on sheet 7) that decodes the Transfer Byte instruction.

The signal that indicates that a valid command, data, or status byte has been received is the P-BYTE signal on sheet 8. When this signal is active, it transfers the current byte to the output register and triggers the I/O sprocket one-shot. In turn, this one shot sends the I/O-SPR signal to sheet 5. There, the I/O-SPR signal sets the I/O sprocket flip-flop 127 to record the fact that a byte has been received. If this flip-flop is set at TE-time of cycle I3 of a Transfer Byte instruction, flip-flop 132 is set to prevent a stall. The byte available is then accepted into the accumulator and the processor proceeds. If, however, the I/O-SPR signal has not set the I/O sprocket flip-flop when a Transfer Byte instruction is received, the I/O-STALL flip-flop is set and the I/O-STALL signal is sent to the processor. In response, the processor remains at the current Transfer Byte instruction until the I/O-SPR signal is received. This clears the stall flip-flop, the byte available is transferred to the accumulator, and the processor proceeds to the next instruction.

I/O Operation Decoder

The I/O operation decoder is shown on sheet 7 of the I/O controller logic diagram. It decodes the four low-order bits of the memory data register (P-EX-MDR-0 through P-EX-MDR-3) and activates one of nine control lines to the SIO control circuits each time that a 175-XXX instruction is sensed by the instruction decoder on sheet 2 of the I/O controller logic diagram. The control line activated by each of the SIO instructions is as follows:

a. The Start Transmit instruction activates the START-XMIT line if the terminal is in the master mode. This clears the LINE-CLAMP flip-flop, sets the ACTIVE flip-flop, sets the XMIT flip-flop, sets the START flip-flop, and triggers the LINE BUSY one-shot.

b. The Transfer Byte instruction activates the REC-BYTE line in order to accept an input byte. This signal clears the I/O sprocket flip-flop on sheet 5 and activates the I/O RD (I/O read) line. At the accumulator, the I/O RD signal loads the byte received into the accumulator. On sheet 5, the inverted I/O read signal allows the stall signal to be produced if the I/O sprocket signal has not occurred. On sheet 8, the P-I/O RD signal generates a dummy sprocket signal if the terminal is in the master mode and is not active.

c. The Transmit Data Byte instruction activates the XMIT-DATA-BYTE line and the Transmit Control Byte instruction activates the XMIT-CONTROL-BYTE line. Either of these signals activates the XMT-BYTE line. This sets the BYTE-XMITED flip-flop and on sheet 9 sets the BYTE-READY flip-flop.

d. The Stop Transmit instruction is effective only in the master mode. It sets the STOP-XMIT flip-flop and thus prevents the XMIT flip-flop from being set.

e. The Inhibit Line instruction sets the LINE CLAMP flip-flop, which is cleared by a general clear signal or the Start Transmit instruction.

f. The Set Address instruction activates the LOAD ADDRESS line, which transfers the contents of the accumulator to the address register.

g. The Set Master instruction sets the MASTER flip-flop, and the Set Slave instruction clears it.

Another part of the instruction decoder is the INDICATOR BIT LATCH shown on sheet 7. When the Transmit Data Byte instruction is decoded, the "2" in the low-order positions (MDR 0 and 1) sets the indicator bit latch. This places a zero in the ninth bit of data bytes and leaves it a one for all other types of bytes.

Five instructions (Transmit Data Byte, Transmit Control Byte, Set Device Address, Set Slave, and Inhibit Line) activate the P-LOAD-DATA signal from the operation decoder. This signal transfers the contents of the accumulator into the input data register.

Synchronization Circuits

The SIO signal changes (also called crossings) establish the timing relationship between communicating devices. Synchronization is knowing which bit is which; that is, being able to identify each bit in each position, for each byte. The sync circuit, most of which is shown on sheet 9, establishes the synchronization and produces signals for the following functions:

a. Shift serial data into and out of the I/O shift register.

b. Recognize byte length.

c. Determine time of generating the byte-type bit in position eight of the I/O shift register.

The start pattern is a series of 17 zeros followed by a single one. This is actually a start byte followed by an idle byte. Without exception, this start pattern begins each legitimate transmission. This is so even when a communicating device automatically turns around to start transmitting after detecting the end of reception. The next byte after the start pattern may be an idle byte, address byte, or any other transmission.

To make sure the synchronization process begins with a complete character of zeros, the first bit received by the unit sets the INHIBIT one-shot (110) in the phase decoder on sheet 9. This prevents detection of changes corresponding to ones for about 8 usec, or 12 bit periods. This phase decoder must then operate to maintain a LINE BUSY status with only zero inputs. If a one was received in place of a zero, the P-BREAK signal would be activated but would not start the turn-around to a transmit condition because the STOP-F/F has not been reset. Instead, the next zero change would retrigger the N-INHIBIT one-shot and this, once again, prevents detection of one changes for 8 usec.

The active N-INHIBIT signal also disables the P-SYNC flip-flop (135) and the nine counter. The input zeros are being loaded into the I/O shift register but the data is not being used. When the N-INHIBIT signal becomes inactive, 12 zeros have been shifted into the I/O shift register and the system is in the middle of the idle byte. Since the complete idle byte, with the one in the most significant bit position, is not yet in the I/O shift register, P-IDLE-R signal remains inactive and there is still no sync and no nine count.

When the one bit of the idle code is shifted into the I/O shift register, the P-IDLE-R signal becomes active. The trailing edge of the same shift pulse that entered the idle code one-bit causes the SYNC flip-flop to be set. This enables the nine counter (139) tc start counting. The first count of the nine counter will correspond with the first bit following the one of the idle code. The nine counter circuit outputs are used at various points to indicate the reception of a complete byte. This is also true when the unit is transmitting idle bytes. Figure 2-51 shows the nine counter circuit timing.

The sync circuit and the portions of the phase decoder with which it interfaces work the same for both the receive and transmit modes except that the shift signals and the nine-counter operation are based on the transmit clock (P-XCL) instead of the changes in the input signal. Also, the data output (P-SD) of the phase decoder is disabled by the effects of the P-XMT level. The P-SYNC signal remains active as long as there is no break in the data and no corresponding clear signal is generated.

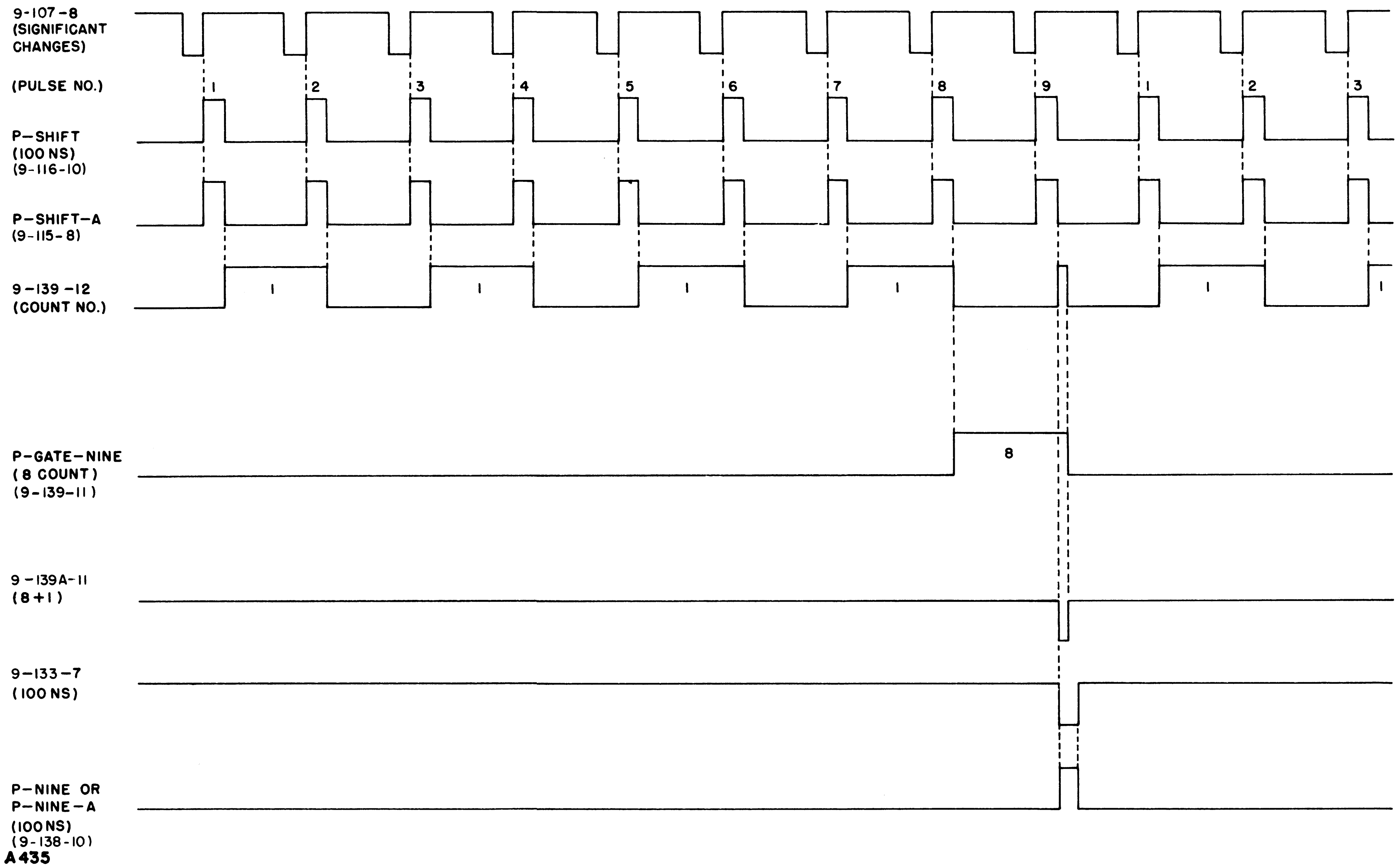


Figure 2-51. Nine-Counter Timing Diagram

Phase Decoder

The receiver portion of the serial I/O circuits detects transmission line activity or absence of activity, distinguishes between significant and insignificant crossings on the transmission line, converts line levels to data, and provides an indication of a "hit" (erroneous change of state) on the line. Significant crossings arrive from the transmitter at a clock rate of 1.5 MHz, and the 666-2/3 ns separation of significant crossings is used as a "known" value by the receiver in achieving synchronization.

The functions outlined above are performed by the phase decoder, which is shown on sheet 9 of the I/O controller logic diagram. The main input to the phase decoder is the serial, phase-encoded data (P-REC-LINE) from the receiver amplifier. The main output is the serial binary signal (P-SD) supplied to the shift register. A timing diagram of phase decoder operation is shown in figure 2-52.

The phase decoder detects the significant data changes and performs the following functions:

a. Generates the timing signals used to sample the received data at significant times for each bit position. The timing signals are also used to produce the shift pulses required when the unit is receiving.

b. Detects when changes on the line are abnormal and indicates an error condition.

c. Detects when reception stops (as indicated by no more changes).

A string of 17 zeros (start byte followed by an idle byte) is always the first data to arrive from the transmitter. Insignificant crossings occur with these, and could be interpreted as data "ones", but the phase decoder initially ignores them, accepting only crossings that can represent data "zeros". It begins repeating a 400-ns delay each time a zero crossing is detected, and then samples the line state after each delay period. If any insignificant crossing were being sensed, it would occur during the 400-ns delay and be ignored.

Distinguishing between significant and insignificant changes in the signal on the SIO line is based on the following points:

a. One bit period is 667 ns.

b. Significant changes occur approximately 444 ns after the start of the bit period.

c. A 170-ns signal is generated for each level change of the input data. The trailing edge of the 170-ns signal starts a 400-ns delay that is used as a gate signal. Since this signal ends about 570 ns after the previous significant level change in the input data, the gate is conditioned to pass a change for about 100 ns before the next scheduled significant change should take place. Since it is unlikely any other changes would occur during this 100-ns period, these other changes are inhibited and are referred to as insignificant changes.

The first circuit in the phase decoder is the energy detector. The P-REC-LINE signal from the receiver amplifier is inverted and applied to this circuit, which consists of two 170-ns one-shots that generate pulses to indicate line crossings. One (element 105) triggers on positive-going crossings, and the other (element 102) triggers on negative-going crossings. A 170-ns P-ENERGY pulse appears at gate 104 for each crossing, whether significant or insignificant. If the input included a short-duration spike, or other signal with a duration of less than 170 ns, both one-shots would be on at the same time. This would normally set the ERROR-F/F, which is described later.

The next circuit to be discussed is the significant crossing gate, which is made up of an adjustable 400 ns one-shot (106) and gate 107. The one-shot is triggered on the trailing edge of the P-ENERGY pulse from gate 104. Its inverted output ($\overline{Q}$) then inhibits gate 107 for 400 ns, so insignificant P-ENERGY pulses cannot affect the break detector and shift one-shot. When the significant crossing gate has ended, the next pulse to pass gate 104 will be significant. It will re-initiate the break detector and trigger the shift one-shot to generate a shift pulse for the shift register and nine counter. The first significant pulse will also cause the serial data gate to sample the state of the line at the end of its 400 ns-period. Re-triggering of the significant crossing gate by insignificant crossings during the 400 ns period is inhibited by its Q output. The Q output is returned to the one-shot input, and acts as an inhibit for any additional trigger until its cycle is complete.

The data is separated at the serial data gate (element 111), which samples the inverted line state at the end of the significant crossing gate period. Line state is available at its "D" input (pin 12) and the $\overline{Q}$ output of the significant crossing gate one-shot serves as its clock at pin 11. At sampling time, the output of inverter (101) is at a high level for a "one" or low for a "zero". The $\overline{Q}$ output of the serial data gate (P-SD) is gated into the ninth bit of the shift register. Simultaneously, the shift one-shot (116) forms the shift pulse applied to the register.

The purpose of the break detector is to detect the end of transmissions. The two one-shots of the break detector generate the P-BREAK signal if transmission line activity is interrupted longer than 750 ns. If the first one-shot (element 108) does not receive a trigger from a significant crossing pulse, it times out and the trailing edge of its Q output triggers the second one-shot (element 109). The P-BREAK signal triggers a station into transmission automatically after another station stops transmitting to it. Either the P-BREAK or N-LINE-BUSY signal can generate N-CLEAR and N-CLEAR-A signals, which clear various portions of I/O circuits.

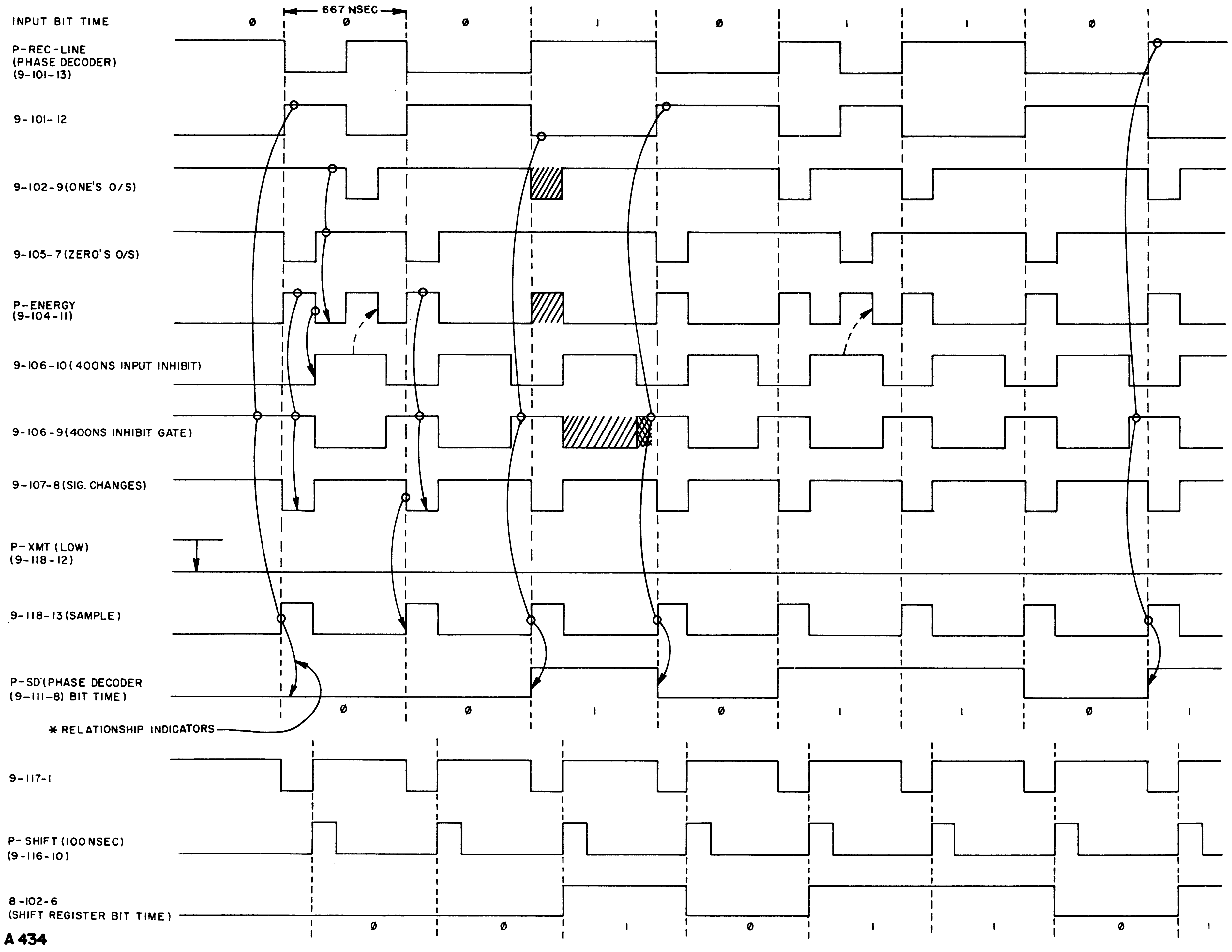


Figure 2-52. Phase Decoder Timing Diagram

The ones inhibit circuit (element 110) is an 8-us one-shot that fires one time for each reception interval. The first detected line crossing reaches the break detector before the significant crossing gate can be activated. As the break detector activates, its $\overline{Q}$ output (element 108, pin 7) triggers the ones inhibit one-shot. Normally, the break detector does not time out during reception, so the ones inhibit circuit is not triggered again. The N-INHIBIT signal from the ones inhibit circuit disables "ones" energy detector one-shot 102 for 8 us (12 bit positions), during which time the receiver synchronizes to the start pattern.

The next circuit to be discussed is the line busy one-shot (119). The purpose of this 10-us one-shot is to sense when communication has ended between a master and slave unit. The P-ENERGY signal triggers line busy one-shot 119 to indicate the presence of transmission line activity. It should be noted that the receiver is capable of detecting line activity generated by its own I/O transmissions. This allows the P-ENERGY signal to keep the line busy one-shot firing. When the communications end, the line busy one-shot times out. This produces the same clear signals as the break one-shot, but, in addition, it also clears several control flip-flops which make it necessary to re-establish the communication link.

The circuits that detect errors on the line were mentioned briefly earlier. They consist of an error detector and an error flip-flop. The receive error detector is a single 4-input NAND gate (element 103 on sheet 9). After timing synchronization occurs (idle code detected), it continuously samples the outputs of the energy detector one-shots. Since crossings, both significant and insignificant, are at a given clock rate, the 170 ns energy detector outputs should not overlap. An overlap indicates an erroneously line activity, as might result from interference, that causes the receive error detector to set the error flip-flop (element 158 on sheet 7). When the N-ERROR signal is active, data is not transferred out of the I/O shift register to the output register. The N-ERROR signal also resets the LINE-BUSY one-shot and then keeps the positive-going changes from continuing to re-trigger the circuit. This circuit then times out in about 10 us, producing a clear signal.

Phase Encoder

The transmission of data requires the data be converted from standard binary levels to a phase-encoded form. The data is then placed on the SIO line with enough drive to be received by any unit on the line. The following paragraphs describe details of how data is converted for transmission.

The circuits involved in data transmission are the clock and the phase encoder on the I/O controller board and the line driver on the system reset board. The clock and phase encoder are shown on sheet 7 of the I/O controller board logic diagram, and the system reset board is shown in drawing 1092.

The phase encoder uses the 1.5 MHz outputs of the clock circuit, the complemented serial outputs of the I/O shift register, and the P-XMIT-A signal as inputs. These signals are applied to a group of NAND gates to produce signals that control the flip-flop to produce the desired phase-encoded signal.

The following are basic requirements of the phase encoder:

a. Produce a signal level change at the proper time for each bit.

b. A one bit requires a positive-going change and a zero bit requires a negative-going change.

c. When the bit value does not change from one position to the next, a change called an insignificant change must occur so the level can again change in the same direction as it did for the previous bit.

The clock signals are summed to divide one bit period into four parts. These one-quarter-bit duration signals are selectively NANDed with the complemented serial (P-QØ and N-QØ) outputs of the I/O shift register.

The gate outputs are applied to the phase encoder flip-flop. The inputs and gates are selected to set and reset the flip-flop in the desired manner. The output (P-XMIT-LINE) of the phase encoder is supplied to a driver on the system reset board to a transistor buffer. This transistor supplies the I/O-CABLE signal.

A timing diagram showing the operation of the phase encoder for a specific byte appears in figure 2-53. The byte being encoded appears across the top line of the diagram.

When the operation decoder decodes a Start Transmit instruction, the N-START-XMIT signal triggers the LINE BUSY one-shot (119 on sheet 9), resets the START flip-flop (143 on sheet 9), and sets the XMIT-F/F (141 on sheet 7). With the circuit now enabled to begin transmitting, the start pattern, idle code, and device address must be sent to all slaves on the SIO channel. The line busy one-shot generates the N-CLEAR-A signal at element 113 on sheet 9, which sets the START PATTERN flip-flop (153 on sheet 9) and the IDLE flip-flop (148 on sheet 9). The clear signal also resets the shift register. The start pattern flip-flop enables the circuits to shift the cleared contents of the shift register to the phase encoder.

The transmit encoding clock, element 172 on sheet 7, generates timing pulse P-XCL for the I/O circuits during transmission. A divide-by-two flip-flop (174 on sheet 7) receives a 6-MHz clock input (N-OSC) and produces a 3MHz clock to the transmit encoding clock. The outputs at gates 168 and 169 are pulse trains of 1.5 MHz. These are, in turn, gated with shift register serial data (P-QO and N-QO) to set and reset the

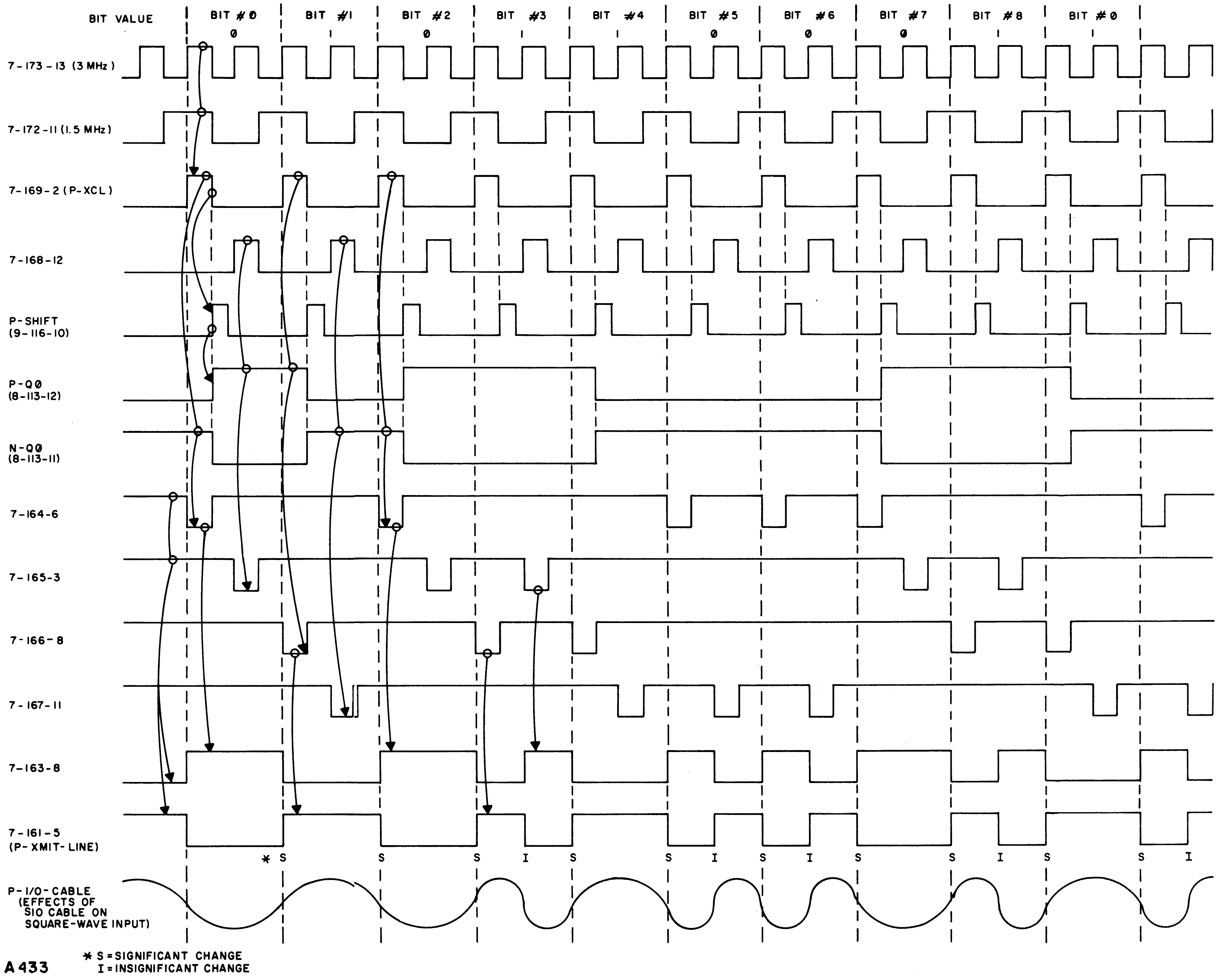


Figure 2-53. Phase Encoder Timing Diagram

XMT-ENCODER-F/F, as required for line transmission. The XMT-ENCODER-F/F, elements 162 and 163 on sheet 7, drives the transmission line via driver 161. (Note that when an Inhibit Line instruction is decoded at the operation decoder, the N-CLAMP signal resets the XMT-ENCODER-F/F and the transmission line is held at a +5V level.)

Each shift pulse (P-SHIFT-A) generated by the shift one-shot increments a shift pulse counter (element 139 on sheet 9). When this counter has counted to nine, it triggers the nine counter one-shot (element 133 on sheet 9), which resets the start-pattern flip-flop and presets the shift register ninth bit flip-flop in preparation for transmission of the idle code. When the start byte has been sent, the shift register contains octal 400 -- the pattern of the idle code. Successive shift pulses transmit this pattern to the SIO line.

When the shift pulse counter generates the P-GATE-NINE signal at the end of the idle code, the contents of the input data register are loaded into the shift register. This data is the device address of the slave with which communication is required.

2-6. CARTRIDGE TAPE UNITS.

Standard in the Model 1501 Intelligent Terminal are two mechanically independent tape transport mechanisms. Mounted at each transport is a motion control board. This board provides drive signals to the transport's three motors and receives signals from the tachometer, mounted at the capstan shaft, as well as various sense switches that indicate the status of the mechanism.

Located at the read head on each transport is a tape preamplifier. Signals from each of the preamplifiers are routed to a dual tape amplifier board, which serves both tape units.

Tape Transport Drive and Interlock Components.

First to be discussed are the tape drive mechanism and the interlock components, which are shown in figure 2-54. A leader from the take-up reel passes over the drive capstan, over the read and write heads, through an end-of-tape sensor, and terminates at the leader clip. This is shown with the leader clip in the home position. When in this position, the clip depresses the actuator arm which, in turn, depresses the clip-in switch (S3) plunger and also rotates the tape-out interlock to the released position. With switch S3 closed, a signal is furnished to the motion control board where it disables rewinding and provides a clip-in status signal to the I/O controller. This is discussed in more detail in the motion control section.

With the tape-out interlock in the released position, it is possible to lift the loading arm (not shown) to the load position. With a cartridge placed in the loading arm, it is lowered into position and the lock arm is rotated to secure the loader.

The cartridge case then depresses the cartridge-in switch plunger (S5), and rotation of the lock arm closes S1. With these switches closed, a signal to the motion control board enables the drive circuits and provides a cartridge-in status signal to the I/O controller.

If it is desired to write data onto the tape, a small plastic pin is inserted into the cartridge. This write pin then depresses a plunger, closing S2. With S2 closed, as long as the manual rewind switch (S4) is not depressed, a connection exists between the motion control board and the write head.

The tape contained on the cartridge is equipped with an eyelet at the beginning of the tape. When the cartridge is placed into position, the eyelet passes over one side of the leader clip. The other side of the clip is attached to the leader. Thus automatic tape threading can be achieved.

Tape is driven forward or in reverse by the capstan. Power is applied to motor M3 and speed is regulated by closing a feedback loop with tachometer M4. In the forward direction, tape is wound on the take-up reel by motor M2. In the reverse direction, tape is wound on the supply reel by motor M1. When tape is driven in the forward direction, the leader clip leaves the home position, dropping the clip-in signal and rotating the tape out interlock, which latches the loader arm.

The end of the tape attached to the supply reel contains a reflective surface which actuates the end-of-tape photo sensor. This provides a signal to the motion control board and also to the I/O controller.

If the processor fails to rewind the tape at the time that the tape is to be removed, then the leader clip being out of home position will prevent opening of the loader arm. In this case, depressing the rewind switch (S4) causes the tape to fully rewind. Since this switch sets a flip-flop, it is not necessary to hold the switch closed. When the clip enters its home position, the rewind function is cleared.

Tape Motion Control Board.

The tape motion control board, which is shown in diagram 1035, accepts control commands from the I/O controller or the multiplex board and provides drive voltages to the tape transport. There are five modes of tape motion, as follows:

a. Forward at normal speed (10 inches per second).

b. Forward at high speed (40 inches per second).

c. Reverse at normal speed.

d. Reverse at high speed.

e. Stop.

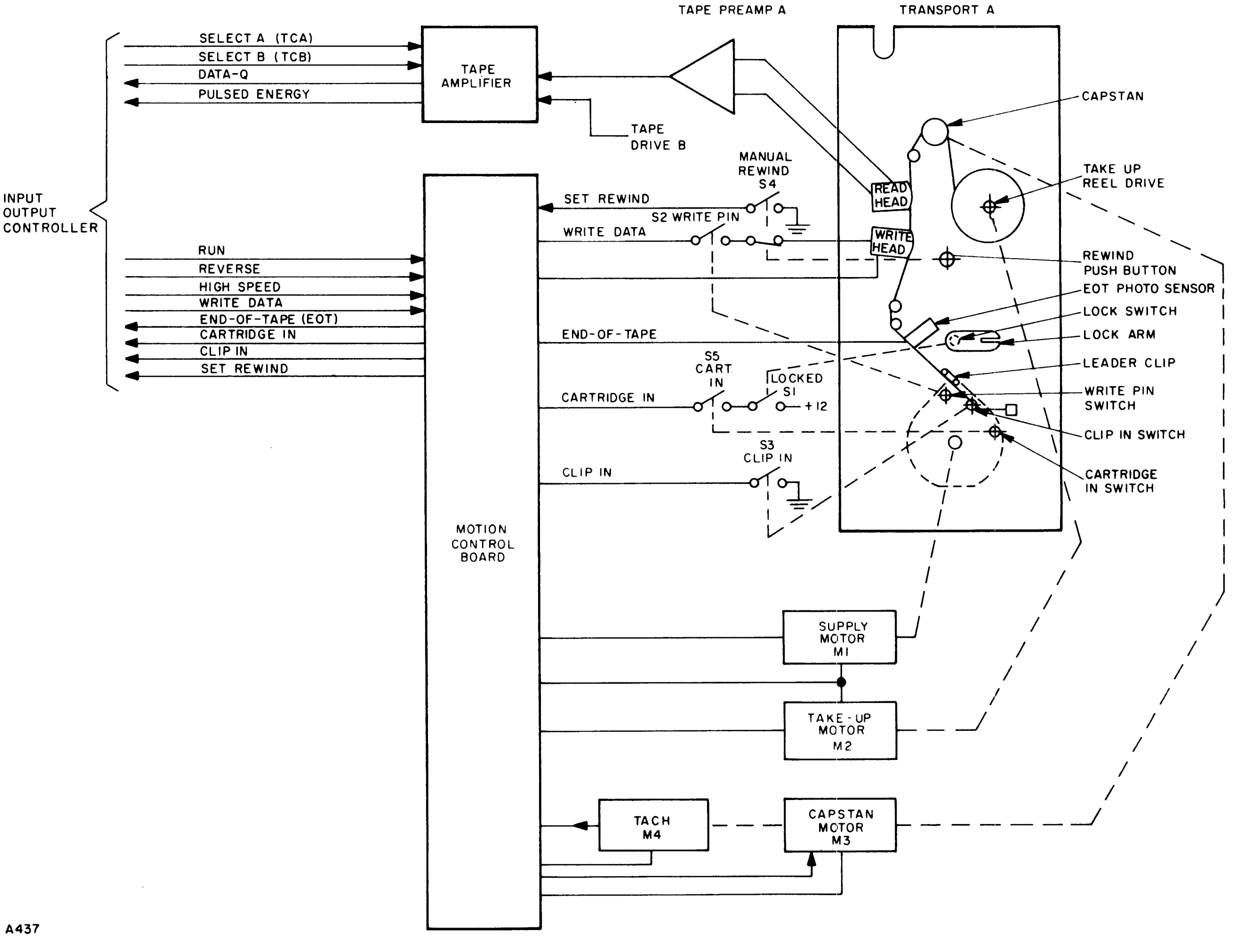


Figure 2-54. Cartridge Tape Unit Block Diagram

Command Decoding

Command decoding circuits are shown at the left of the motion control board diagram. The output of gate 102 represents reverse, and the output of gate 106 represents forward. The absence of both active states from these gates is the stopped condition of the transport.

To enable either gate, a high must be placed on gate 102, pin 10, and gate 106, pin 12. Thus, the outputs of gates 101, 108, and 115 must be high, and they depend on the following conditions:

a. Cartridge In and Locked. When the cartridge-in switch and the locked switch are both closed, +12 volts through diode CR20 generates a high at the input of gate 114. Therefore, the output of gate 115 is high.

b. Clip-In. If the clip-in switch is closed, a low is placed on gate 109 input, resulting in a high on pin 4 of gate 108. If a reverse command is issued, the output of gate 105 is high. Thus gate 108 output is low, which disables gates 102 and 106. This prevents driving in reverse if the tape is already rewound with the leader clip in the home position. If the clip is not at home or if the reverse command is not issued, then the output of gate 108 is high.

c. Run and Manual Rewind. The final condition necessary to enable the gates is to have either a run command or have the manual rewind switch depressed.

d. Direction Control. If the enabling conditions to allow tape motion are satisfied, then the selection of the tape direction is controlled by the state of the reverse command.

Capstan Drive Amplifiers

To the right of the decoding circuits on the logic diagram are shown the analog circuits used to drive the capstan motor. Amplifiers 103, 107, 111, and 116 are operated in the open loop mode (no feedback resistor) and serve as level shifters to generate ±12-volt level outputs from the standard logic levels. Amplifier 117 establishes a reference voltage against which the tachometer output is compared. Comparison takes place at amplifier 118 whose output is buffered by power drivers Q12 and Q13.

Direction of tape motion is established in the following manner: The voltage divider formed by R8 and R10 establishes +2 volts at the non-inverting inputs of amplifiers 103, 107, and 116, and at the inverting input of 111. If gate 102 is at the active low level, the output of amplifier 103 is high, which switches Q1 on. This connects the negative voltage, adjusted by R24, to the reference amplifier 117 input. Likewise, if gate 106 is at its active low level, the output of amplifier 107 is high, which switches Q2 on, connecting the positive voltage, adjusted by R19, to the reference amplifier input.

If the THS (tape high speed) command is active, the output of amplifier 116 is at -12 volts, causing Q4 to be in the off state. This causes one end of R28 to be disconnected from the summing node at amplifier 117, pin 3. The voltage switched by the direction switches is applied to the voltage divider formed by R26 and R27.

If the THS command is inactive, the output of amplifier 116 will be +12 volts, causing Q4 to conduct, which clamps one end of R28 to ground. The voltage switched by the direction switches is applied to a voltage divider which now has R28 in parallel with R27. The attenuation is now four times greater, and the output reference voltage from amplifier 117 is one-fourth as much as it was with the THS signal active.

The reference voltage from amplifier 117 appears at the non-inverting input of amplifier 118. The reference voltage has five possible magnitudes that correspond to the five speeds and directions of the tape travel previously discussed. These are:

a. Positive, low magnitude: forward at normal speed.

b. Positive, high magnitude: forward at high speed.

c. Negative, low magnitude: reverse at normal speed.

d. Negative, high magnitude: reverse at high speed.

e. Zero: stopped.

For example, assume a positive voltage at amplifier 118 input with the capstan motor stopped. The tachometer is connected to the inverting input of amplifier 118. The output of the amplifier is therefore the difference between the input reference voltage and the feedback voltage from the tachometer. At rest, the tachometer voltage is zero, and with the assumed positive reference voltage, amplifier 118 produces a positive output. This is applied to the bases of Q12 and Q13. These two transistors simply buffer the amplifier voltage. Therefore, a positive voltage is applied to the capstan motor, which then accelerates in the forward direction. As the tachometer voltage rises, the difference between it and the reference voltage diminishes, which reduces the voltage applied to the capstan motor.

Finally, when the tachometer voltage nears the value of the reference voltage, there is just enough voltage applied to the motor to maintain the proper speed, and the system is in equilibrium. The servo amplifier regulates the capstan motor speed to within ±2 percent of the specified values of 10 and 40 inches per second.

The operational amplifiers discussed make use of the +12 and -12 volt supplies, and correct circuit operation depends upon these voltages being of correct magnitude. When power is first applied to the supplies, the supply output voltages will not reach operating values at the same time. To prevent tape motion during this time, transistors Q11 and Q14 clamp the servo output to zero volts until both +12 and -12 volts are present.

If the +12 volts is present without the -12 volts, then Q11 is turned on, clamping the output to zero. Likewise, if the -12 volts is present without the +12 volts, Q14 is on, clamping the output to zero volts. If both voltages are present, both transistors are biased to zero volts and, therefore, are not conducting.

The next step is to discuss the motion-inhibit circuit. If the cartridge-in switch or the locked switch is open, then gate 112, pin 6, is high. This causes the output of amplifier 111 to be high. In turn, this high causes Q3 to conduct, placing R28 in the reference amplifier circuit and thus producing the lower value of reference voltage. Also, Q10 conducts and shorts out R32 to reduce the gain of amplifier 118 to unity. This ensures that no drive voltage will be placed on the capstan motor until the cartridge is in and locked.

Supply and Take-Up Motor Drive

As soon as the cartridge is in and locked, the motion control board takes action to maintain tape tension. This is accomplished as follows: With the cartridge in and locked, +12 volts is applied through CR15 to the supply and take-up motor switching circuits. Standby current is applied to the supply motor through R47. Also, R50 turns on Q9, providing standby current through R48 to the take-up motor. The standby current maintains some tape tension even when the tape is in the stopped condition.

The next step involved is to start the tape moving and apply full tape tension. Although the tape used is very light and the mass of the tape reels is low, some acceleration time must be allowed to bring the tape up to operating speed. (The same applies to deceleration time when the tape is to be stopped.) Shown in figure 2-55 are the motion control waveforms involved in starting, running, and stopping the tape.

Upon receiving a start condition, gate 101, pin 3, goes high. This enables capstan drive signals and brings gate 112, pin 6, low. Resistor R11 discharges C10, making the output of amplifier 111 -12 volts. This turns off Q3 and Q10, restoring normal operation to the capstan drive circuits.

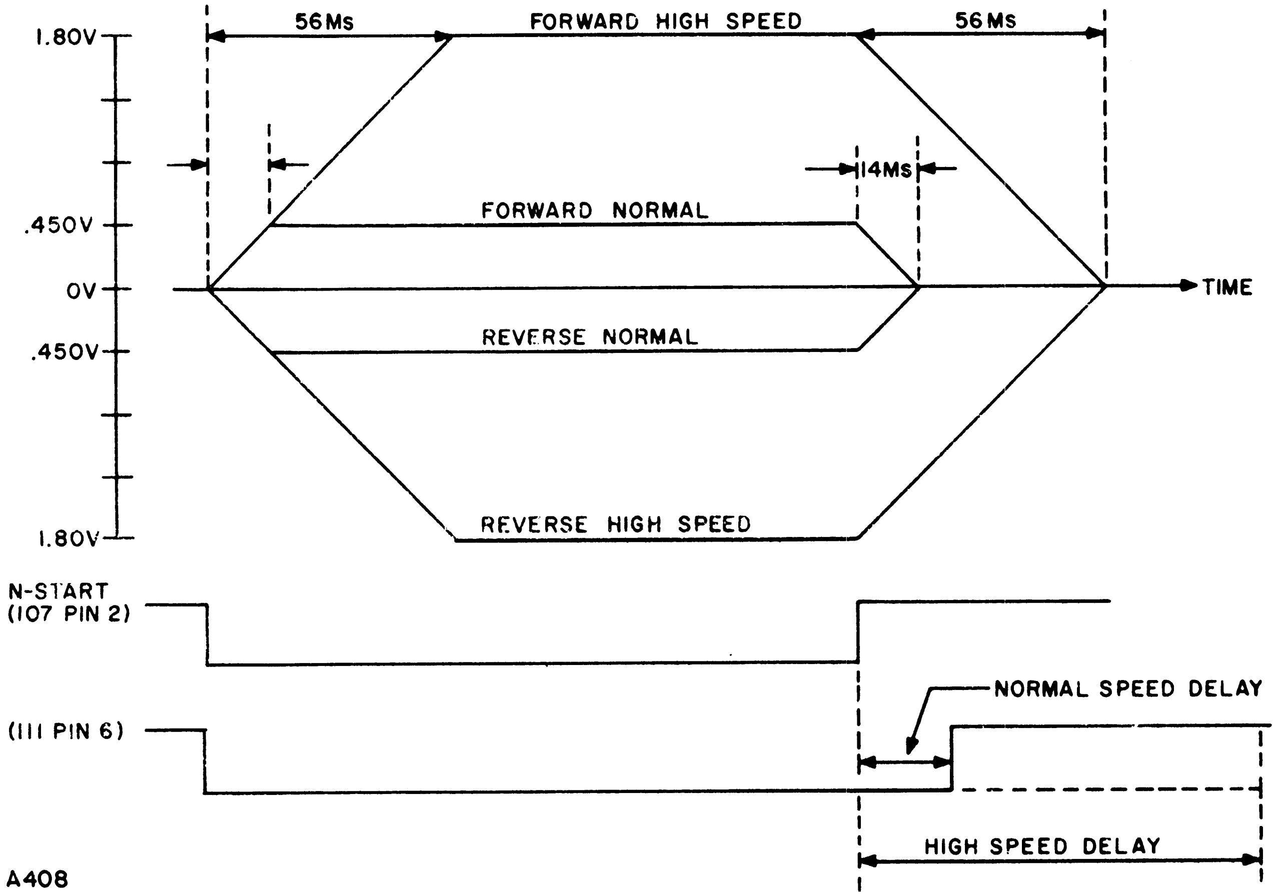


Figure 2-55. Motion Control Timing Diagram

The discharge of C10 produces the start-up ramp when the capstan begins to turn. Until C10 is discharged, the supply and take-up motors are still operating on standby current and maintaining reduced tape tension. Then, transistor Q6 is turned on, turning on Q7. This places -12 volts on the take-up motor. With Q7 on, Q8 is turned on, placing +12 volts on the supply motor. Both motors now produce full torque. Tape motion is determined by the capstan motor, however. If driving forward, the capstan motor overcomes the torque produced by the supply motor and the take-up motor winds the tape onto the take-up reel. If driving in reverse, the torque of the take-up motor is overcome and the supply motor rewinds the tape.

When the tape is to be stopped, the capstan must be stopped and the supply and take-up motors returned to the standby condition. At normal speed, tape motion continues for 14 milliseconds; however, the supply and take-up motors maintain full tension for 56 milliseconds.

When gate 101, pin 3, goes low, gate 112 no longer holds C10 low and C10 begins to charge through R13 and, in some cases, R12. If the THS command is inactive, gate 110 is off and C10 is charged by the parallel combination of R12 and R13. If the THS command is active, then gate 110 is low and CR5 is reverse-biased, allowing only R13 to provide the charging current to C10. Thus, the delay period is extended when in the high speed mode. When C10 reaches the bias level of +2 volts, the output of amplifier 111 returns to +12 volts and switches off the supply and take-up motor voltage.

End-of-Tape Circuits

When the end-of-tape photo detector is illuminated by the reflective strip attached to the end of the tape, the conduction current drives Q16 on. This latches the EOT flip-flop, gates 119 and 120, which turns off Q15. When rewind occurs, Q16 turns off. However, the flip-flop remains set until reset by the closing of the 'clip-in' switch.

2-7. TAPE AMPLIFIERS.

This description covers both the tape preamplifiers, which are an integral part of the read head assembly in each tape drive, and the tape amplifier board. The tape amplifier board is a dual-input unit that receives inputs from the preamplifiers in both tape drives in a pair. However, the tape drive selection signals TCA and TCB are applied to the amplifier to select one input and block the other. Thus, the tape amplifier processes the data read from only one tape drive.

The read preamplifier circuit is shown on schematic diagram 0321-208 and in figure 2-56. Several components located on the read amplifier board are included in figure 2-56 for clarity. Transistors Q1 and Q2 operate as a differential amplifier with their bases biased at ground. Gain stability is provided by emitter resistors R3 and R4. Emitter current is supplied through RA. Load resistors RB and RC provide differential outputs which are AC-coupled to amplifier 101. Each side of the transistor pair provides a gain of 25, for a total differential gain of 50.

A brief review of the recording technique and format is required before a description of the tape amplifier is given. Each tape cartridge contains one hundred feet of .15-inch-wide magnetic tape. Data may be written onto this tape when the tape is transported in the forward direction at 10 inches per second. The tape can be read in either the forward or reverse direction. Although actual data is read only at 10 inches per second, records may be counted while the tape is moving at 40 inches per second. This allows high speed searching for a particular record. Data is recorded on the tape in a single track. For this reason, the recording method must allow both data and timing information to be recorded simultaneously.

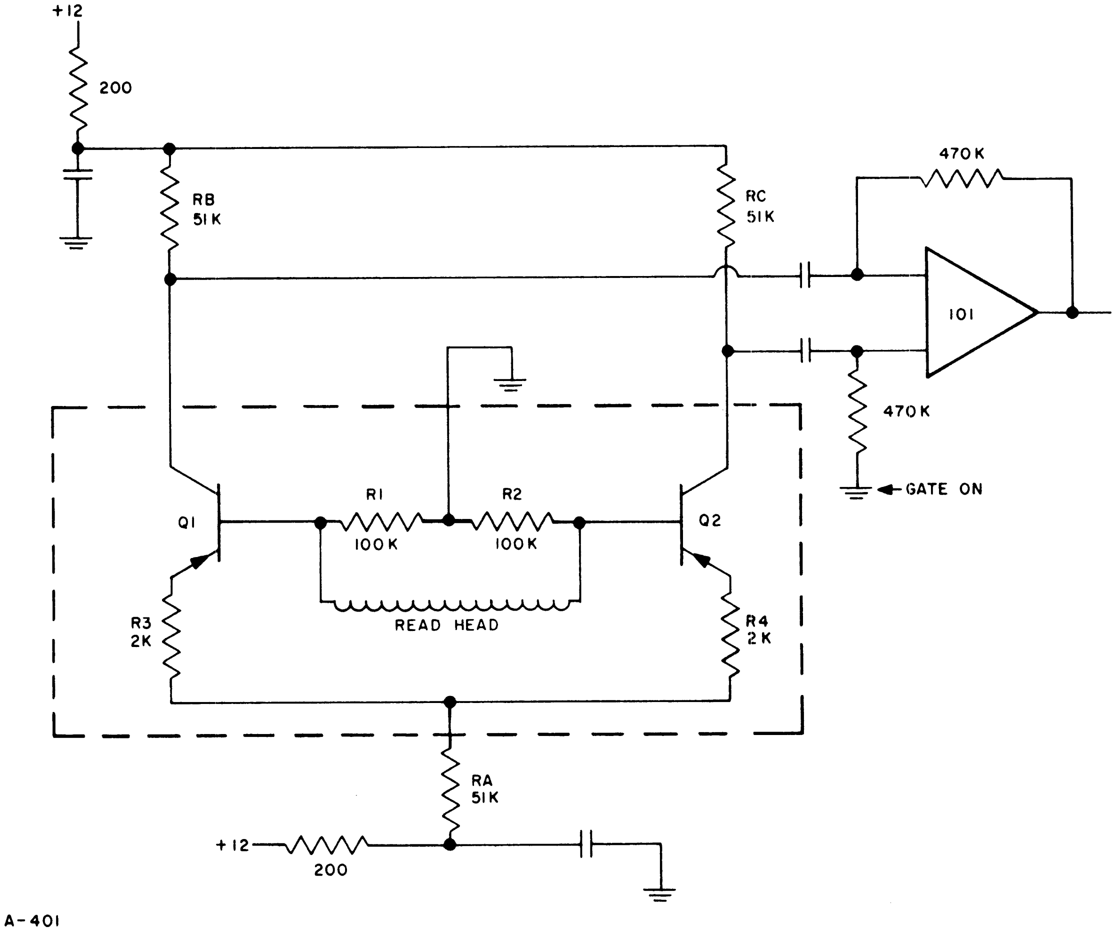


Figure 2-56. Read Preamplifier Simplified Schematic

Recording Method.

To record both data and timing information in a single track, the Manchester Code is utilized. This is a form of phase encoding in which data is represented by signal transitions; a positive-going transition being a one, and a negative-going transition a zero. Obviously, in order to provide the correct transition for a particular data bit, it may or may not be necessary to "precondition" the signal level depending upon the state of the previous data bit. Any "conditioning" transitions are termed insignificant transitions. Actual data transitions are significant transitions. Therefore, at least one and sometimes two transitions occur each bit period, which provides the necessary clocking information. For maximum insensitivity to tape speed variations, the insignificant transitions are positioned midway between the significant transitions.

Record Format.

Data is organized onto the tape in groups of eight-bit bytes. Such a group represents a record. Because there is a possibility of two transitions per bit where only one is significant, it is necessary for the system to synchronize to the data as it is read. Once synchronized, a "locked-on" condition occurs which allows only the significant transitions to be recognized as data.

The format of each record is under the control of the processor. A typical record format consists of 143 bytes organized as follows:

a. Preamble: 3 bytes. Two all zero bytes followed by a byte with a one in bit position 7 and all zeros in positions 0-6.

b. Label: 8 bytes.

c. Data: 128 data bytes.

d. CRC Byte: (cyclic redundancy character). This byte is calculated by the processor and is used for error checking.

e. Postamble: 3 bytes. A byte with a one in position 0 and zeros in positions 1-7, followed by two all zero bytes.

The preamble, which consists of 23 zeros followed by a one, performs three functions:

a. It allows the reading circuits to synchronize reading with the tape flux changes.

b. It allows the reading circuits to distinguish between a data record and tape noise.

c. It allows the reading circuits to locate the first bit of data on the tape.

If the tape is read while moving in reverse, the postamble appears in the same form as the preamble, 23 zeros followed by a one. However, the primary purpose of the postamble is to allow proper reading of the last bit in the record when the tape is being read forward. If the postamble were not used, the last bit of the record would take on undesirable characteristics due to the lack of flux changes following the last bit. Essentially, the postamble terminates the record so that the last bit of valid data can be recovered properly.

Interrecord Gap.

When a record is to be written onto the tape, the transport is started and a delay introduced before writing occurs. This causes an erased gap of about 1.1 inches to be created on the tape prior to the first bit of the preamble. After the record is completed, the first bit written in the record is positioned just before the read head. Therefore, it is possible to perform read-after-write checking without stopping or backing up the tape. The interrecord gap also makes possible the counting of records with the tape moving at 40 inches per second.

Selection of Tape Channel.

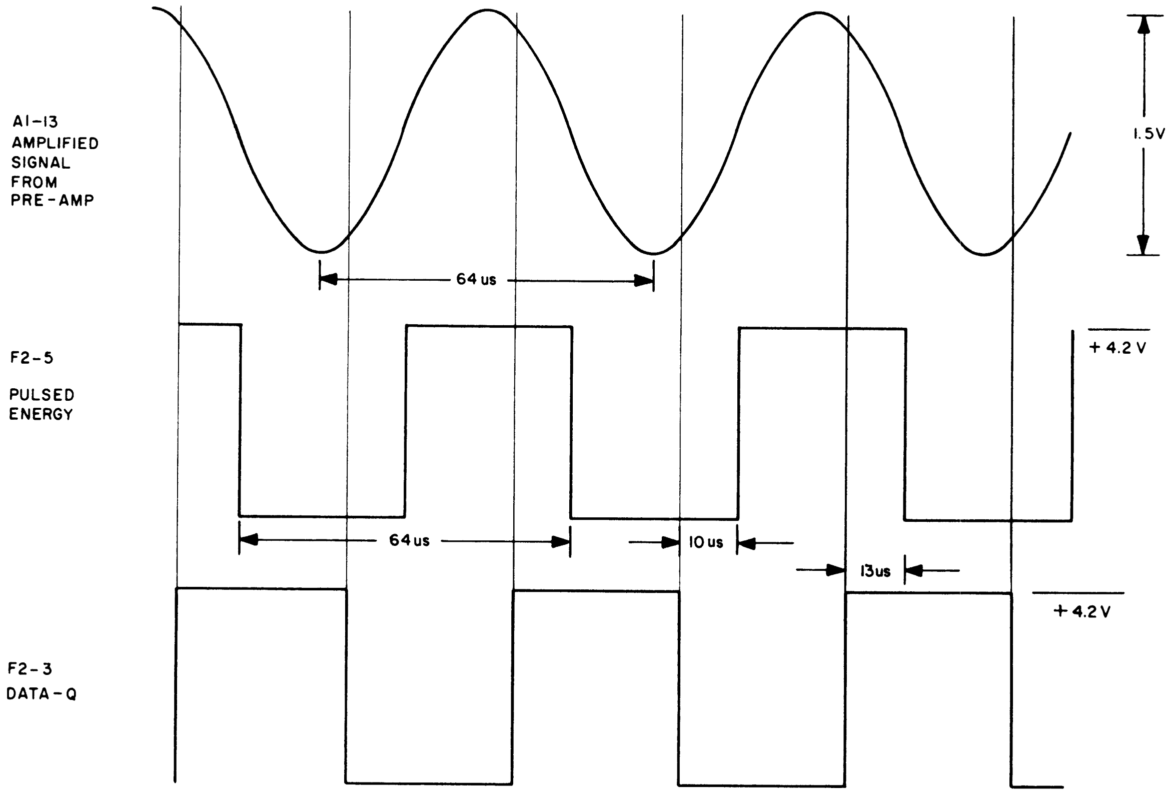
The tape amplifier, which is shown in logic diagram 0321-174, is a two-channel unit. Connected to the first input is the signal read from tape unit A. Note that a preamplifier (0321-208) is an integral part of each read head assembly and that it is the preamplifier output that is connected through a shielded cable to the tape amplifier. The second input to the tape amplifier is the signal read from tape unit B.

Selection signals TCA and TCB from the I/O controller determine whether the tape amplifier processes the signal from tape unit A or B. The selected signal is then routed through two different paths.

Energy Detector and Data-Q Paths.

The first path is the energy detector circuit, the purpose of which is to determine the presence of a signal on the tape. The signal from this circuit is labeled PULSED-ENERGY and is shown in figure 2-57. The second path is a data amplifier, and its output is labeled DATA-Q. The reason for routing the signal read from tape through two circuit paths is discussed below.

In phase encoding, the information is carried by the transitions of the waveform, rather than by the amplitude. The signal read from the tape is amplified and clipped and is therefore relatively immune to amplitude variations. At this high gain, the background noise in the absence of tape signals is sufficient to make it difficult to determine when legitimate data is present. To solve this problem, a separate, limited-gain path is provided after the second stage of an amplification. This channel feeds an energy detector which responds to signals of pre-determined amplitude and duration: greater than 30 percent of nominal signal amplitude and repeating within 250 microseconds for at least 1 millisecond. The circuits in the tape amplifier determine that the amplitude requirements are met, while the tape reading circuits in the I/O controller determine that the pulsed energy signal meets the timing requirements. Only when both the amplitude and timing requirements are met are the reading circuits allowed to recognize the DATA-Q signal.



CONDITIONS:
A402 TAPE DECK B IN READ MODE WITH AN ALL ONES TAPE

Figure 2-57. Tape Amplifier Signals

The output of each preamplifier drives an input stage on the amplifier board, which is shown in schematic diagram 000321-174. Two signals, N-TCA and N-TCB, provide selection signals to amplifiers 101A and 103, respectively. If N-TCA is inactive, then gate 102 is off and amplifier 101A, pin 5, is pulled high by R7. This drives the amplifier into saturation and any signal present on capacitors C3 and C4 cannot pass through the amplifier.

If N-TCA is active, then R8 is clamped to ground, allowing amplifier 101A to operate in its normal linear mode. In this condition, input signals are amplified by about 10 times and capacitively coupled to amplifiers 101B and 106. These two amplifiers represent the two separate signal paths that produce DATA-Q and pulsed energy output signals.

The signal N-TCB operates in the same manner with amplifier 103. Thus, either amplifier may be selected to produce the data and pulsed energy output signals. It will be assumed, for the following discussion, that channel A is enabled.

For the DATA-Q channel, the output signal from amplifier 101A is applied to R10, the input to the operational amplifier 101B. This produces an inverted output with a gain of five, which is AC-coupled to R16, the input to amplifier 108.

Amplifier 108 produces a gain of 10 and its output is AC coupled to amplifier 109. This amplifier has positive feedback through R25 and C24. This forms a Schmitt trigger, which produces sharp edge transitions. The output of amplifier 109 drives logic gate 110 whose output represents the DATA-Q signal to the I/O controller.

For the pulsed energy channel, the output of amplifier 101A is applied to amplifier 106, which produces a gain of 10. The output is AC-coupled to threshold detector 107, where the reference threshold is set by R39 and R40 to 0.75 volts. This represents approximately one-third of the signal amplitude at the detector input. The output of amplifier 107 drives gate 111 whose output is the pulsed energy signal sent to the I/O controller.

Timing diagram 2-58 shows the inputs and outputs of energy detector stages 106 and 107. The signal read from the tape unit and amplified is shown in the line labeled A1-13. The signal labeled B1-13 shows the signal from the first stage of the energy detector. In the energy detector, the triggering (threshold) level of operational amplifier D1 is set at one-third of the nominal level of the signal expected during reading. Integration of noise pulses is provided to help prevent them from being recognized as data.

The output at D1-9 is also shown in the timing diagram. This signal is converted to standard logic levels and routed to the reading circuits in the I/O controller. The basic difference between this and the data signal is that the pulsed energy signal is produced by a lower level of signal. When the pulsed energy signal is present, the reading circuits in the I/O controller are allowed to detect the significant crossings in the data signal

2-8. SYSTEM RESET BOARD.

The system reset board is shown on circuit diagram 1092. This small board is divided into two separate circuits, which perform the following functions:

a. The reset section accepts an input from the POWER ON switch. When AC power is first applied to the terminal, this section forwards a N-POWER-ON-SW signal to the processor. In response, the processor sends a general clear signal to the major circuits in the terminal and clears itself with the processor clear signal.

b. The input-output section acts as an input/output amplifier for serial I/O data on the SIO coaxial cable. Serial I/O data is transmitted to and from the I/O controller.

Reset Section.

When the system power switch is turned on, the systems reset power circuit generates the N-PWR-ON-SW signal for 1.3 seconds. The trailing edge of this signal provides a general clear signal to the system. Figure 2-59 is a timing diagram of this operation.

The N-POWER-ON-SW signal is generated when the AC power switch on the front panel of the terminal is pressed to the ON position causing +12 volts to be applied to the systems reset board. The +12 volts is applied to a charging circuit that causes a relay to energize after a short period. This allows time for all DC power to be applied. At the end of this time, the relay on the systems reset board energizes, opening the normally closed contacts and removing the ground from the N-POWER-ON-SW line.

This signal is fed to the processor. There, it causes the processor to generate the general clear and processor clear signals. The general clear signal is distributed to the I/O controller and the communication adapter board slot, whereas the processor clear signal is distributed to circuits within the processor itself. A discussion of the functions performed by these signals appears in the processor description.

Input-Output Section.

Before starting to transmit serial I/O data, the SIO cable must be cleared for approximately 20 usec. The Inhibit Line I/O instruction executed by the I/O controller in a master terminal performs this function by generating the P-CLAMP-LINE signal, which drives the P-XMT-LINE signal (and the SIO cable) to a +5V level for this length of time. This clamp is left on long enough to cause all slave devices to become inactive because they do not detect any data on the line for at least 10 microseconds. This action is normally taken immediately preceding an initial transmission from a master to a slave. A Start Transmit instruction resets P-CLAMP-LINE, thereby freeing the cable to accept transmission.

The receiver amplifier is shown on the lower-right portion of diagram 1092. It consists mainly of a buffer, a differential amplifier, and a driver. A reference level of about 1.5 volts is applied to the differential amplifier (A1-4) which keeps the output of the amplifier low (about 0 volts) when the input signal is below the reference level. This keeps most noise on the SIO line from triggering the circuit. When the signal level exceeds the reference, the output of the amplifier goes high (about 4 volts) very quickly. As the signal again drops below the reference, the amplifier output goes low very quickly. The output from the driver stage supplies the P-REC-LINE signal to the I/O controller.

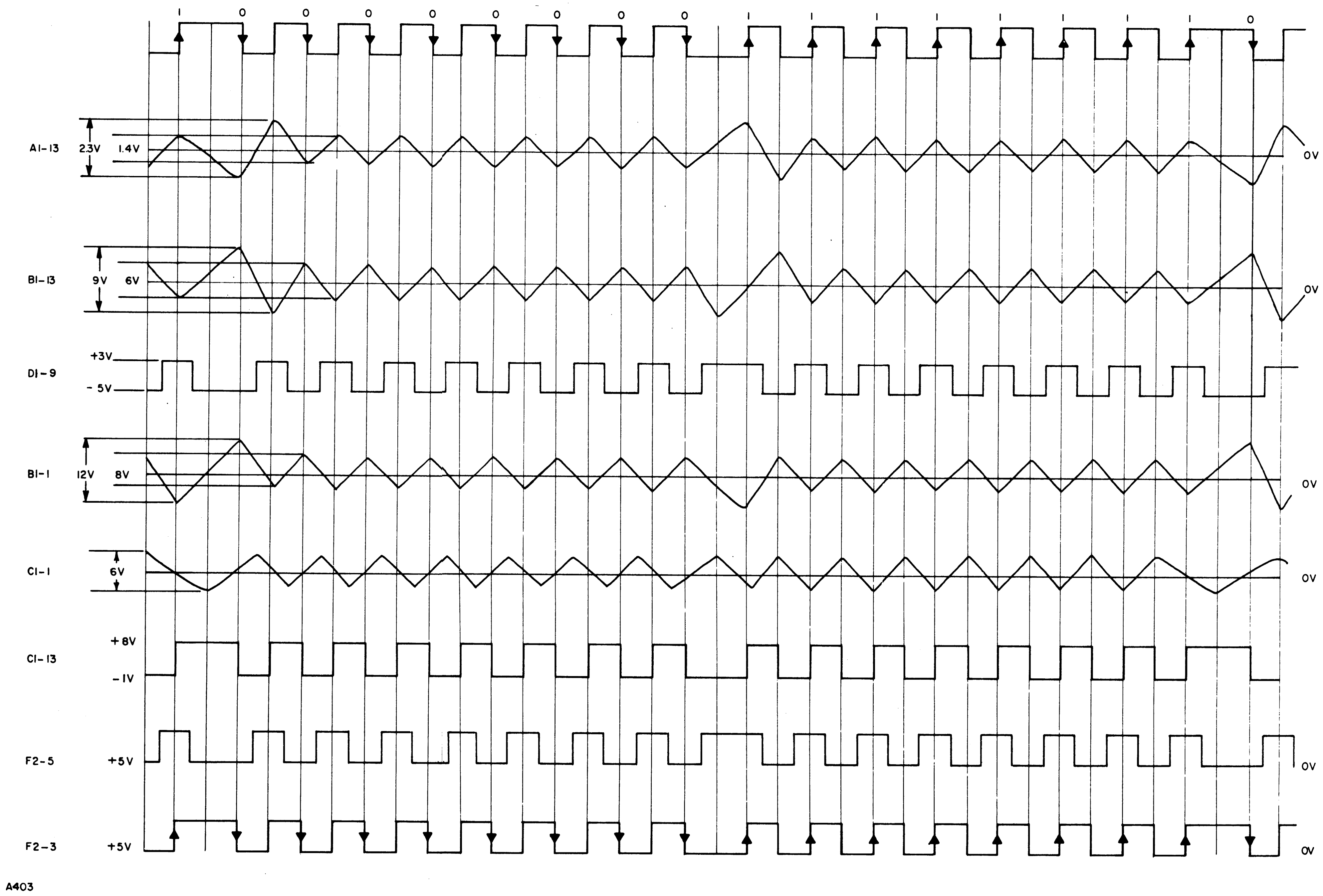


Figure 2-58. Tape Amplifier Timing Diagram

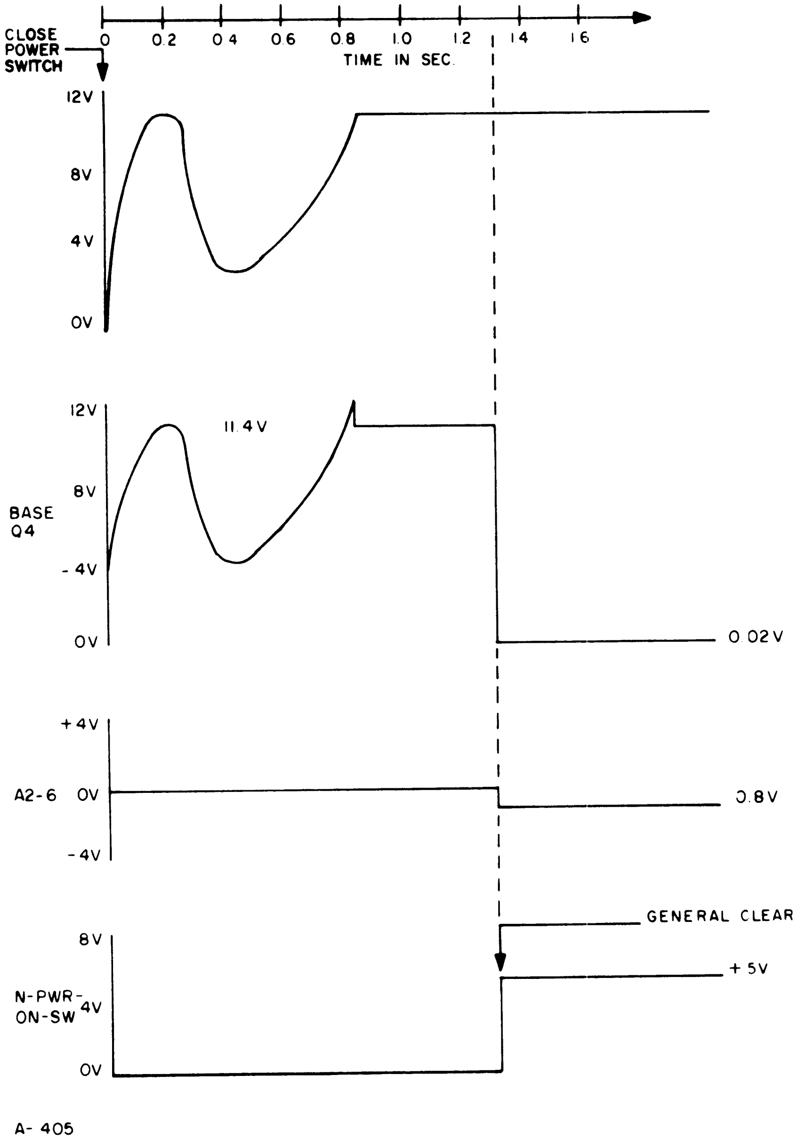


Figure 2-59. Power-On-Clear Timing Diagram

In summary, the positive-going change of the phase-encoded signal produces a high voltage level and the negative-going change produces a low voltage level. These changes are made to produce standard binary logic signals by the phase decoder which distinguishes between the significant changes of the phase-encoded data and the insignificant changes.

Dual Input-Output Section.

One of the options available for the Model 1501 Intelligent Terminal is the Dual I/O System Reset Board. This option allows the terminal to operate two SIO cables rather than the single cable that is standard. However, there is only one set of SIO circuits in the I/O controller and these circuits must be switched between the two SIO cables.

A schematic of the dual installation is shown in diagram 2799. Note that two boards are involved. The first is very similar to the standard board; the major difference being a set of connections to place the second board in parallel with the first.

The lines to the SIO circuits in the I/O controller are connected in parallel to both boards. Then each board is connected to its own I/O cable. Note, however, that only the first board performs the power-on reset functions.

Two input-output instructions are used to switch between channels 1 and 2. These instructions are 173-006-Unlock Keyboard and 173-014-Lock Keyboard. They control the status of the lock flip-flop in the I/O controller and supply the KBRD-LOCK-SOL signal to the dual I/O system reset board. When power is first applied, the status of this signal is such that I/O channel 1 is selected. The lock instruction is then executed to switch to channel 2 and the unlock instruction is executed to switch back.

2-9. COUNTER FOR THE MODEL 1501-FF.

The Model 1501-FF differs from the standard Model 1501 in that it has a five-digit transaction counter located on the rear of the unit. The transaction counter is incremented by issuance of instruction 173-014, Keyboard Lock or Switch to Channel 2.

SECTION 3

MAINTENANCE

3-1. GENERAL.

This section provides maintenance instructions to support field maintenance of the Model 1501 Intelligent Terminal. In general, this information includes overall test procedures, removal and replacement procedures, adjustment procedures, and preventive maintenance instructions necessary for maintenance to the modular (subassembly) level. A module is a complete printed circuit board, a complete tape drive, a complete power supply, etc.

Maintenance of these terminals is done in the field by replacement of entire modules. Conditions may arise whereby it is expedient to attempt minor repair of modules in the field. This should only be attempted where a high degree of certainty exists as to the success of the attempt and it is to everyone's advantage. Only modules that are whole and do not exhibit missing components will be accepted for repair at a module repair station. In the event any field troubleshooting of modules has been attempted with unsuccessful results, dismantled modules must be reassembled before they can be accepted for repair at a repair station.

3-2. LOCATION OF MAJOR COMPONENTS.

Shown in figure 3-1 is the location of the major components in the 1501 that are field replaceable. Access to components is gained by removing the top cover of the unit.

3-3. TOOLS, TEST EQUIPMENT, AND MATERIALS REQUIRED.

The equipment required for maintenance consists of several common tools and measuring devices that are commercially available, and a few special tools and diagnostic test tapes. The following is a list of items necessary for modular maintenance and adjustments:

a. Digital voltmeter

b. 5/16-inch nutdriver

c. 1/4-inch nutdriver

d. 1/4-inch flat blade screwdriver

e. 1/8-inch flat blade screwdriver

f. #0 Phillips screwdriver

g. #1 Phillips screwdriver

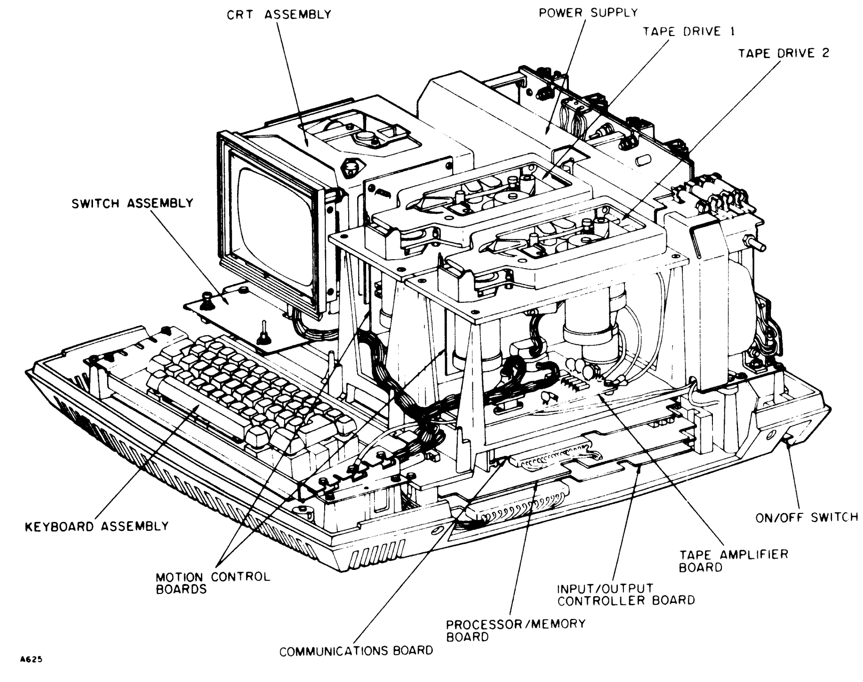


Figure 3-1. Location of Model 1501 Major Components

h. 6-inch long nose pliers

i. Miniature potentiometer adjustment screwdriver

j. 3/32-inch plastic or nylon screwdriver (for CRT adjustments)

k. Capstan Puller Tool (Tool No. T19221)

l. Cleaning kit (SBM No. 9012173-55)

m. Loctite Quick Set 404

n. Diagnostic test tapes

o. Series 1500 Field Engineering Diagnostic Manual 23-0676-100

3-4. PREVENTIVE MAINTENANCE AND CLEANING.

The preventive maintenance schedule appears in table 3-1. Note that it includes cleaning, inspection, checks, and adjustments.

General cleaning is the responsibility of the operator. This includes the usual dusting or vacuuming of the exterior. In addition, the operator should clean the tape drives regularly. The procedure for cleaning and inspecting the drives appears in the Operator Instructions Manual (Publication No. 22-6504-01).

Both the CRT face and the implosion screen must be kept clean. CRT cleaning is a simple procedure that can be easily accomplished whenever the CRT cover has been removed. The dust and dirt that accumulates between the plastic implosion screen and CRT face can be removed with a clean cloth.

Table 3-1. Preventive Maintenance Schedule

Task	Weekly	Every 3 mos.	Yearly	During Each Service Call
Check for worn tape leader		X		X
Clean tape path	X*			X
Check and (if necessary) adjust power supply		X		X
Check and (if necessary) adjust tape drive speed		X		X
Check Capstan			X Replace if worn or defective	X
Remove dust from inside the CRT implosion screen and face.			X	X

*The operator may find it necessary to clean more often, depending on use of system and the environment. Daily cleaning is not unusual in heavy daily use.

Whenever the top cover is removed, any excessive accumulation of dirt on the interior components should be vacuumed away. Excessive accumulation of dust and dirt in electronic circuits acts as an insulating blanket and, by preventing efficient heat dissipation, may cause overheating and component breakdown. At the same time the interior is being cleaned, a careful and thorough visual inspection for signs of dirt, wear, cracks, binds, loose connections or loose hardware should be made. Extreme care must be exercised in the handling of boards, cables, and connections so as not to induce any equipment malfunctions due to handling. Inspection of the interior should be done only by service personnel familiar with the equipment.

The tape drive components should be inspected as called for in the PM schedule. Use the inspection and cleaning procedures provided in the Operator Instructions Manual or in the cleaning kit provided with each terminal. The procedure for checking and adjusting tape speed is included in the adjustment procedures in this section. The PM schedule also requires that the power supply outputs be checked and adjusted regularly.

3-5. TEST PROCEDURES.

Many failures require the use of diagnostic programs to isolate the malfunction. The programs available and the procedures for running them appear in the Field Engineering Diagnostic Manual, Publication Number 23-0676-100. Refer to that manual as required.

The first checks to be made, however, are those that either confirm or eliminate some obvious or common problems, such as physical damage, missing components, or operator error. Listed below are several checks to be made. It is expected that through experience the field engineer will develop his own list of items that he checks before running diagnostic programs or making voltage measurements.

First, inspect the unit for obvious physical damage, such as:

a. Damaged line cord

b. Dented enclosure

c. Circuit boards that could have been loosened by a hard impact.

d. Jammed keyboard keys. Inspect for materials binding the key, and if the key sticks, remove the keytop and lubricate the switch with Freon.

e. Dirty or worn parts in the tape drives.

After preventive maintenance or a repair has been performed and a malfunction exists, check the following:

a. Missing or improperly installed leader on the tape drive.

b. Circuit board missing, incorrectly installed, or improperly seated.

c. Cable or connector not installed or improperly seated.

d. Blown fuses not replaced after repair was made.

Occasionally, an operator error can appear to be an equipment malfunction. This usually involves the tape drive, so the following should be checked:

a. Cartridge had eyelet slightly out of cartridge when it was loaded.

b. Write enable pin is not placed properly.

In order to further assist the field engineer in isolating failures quickly, tables of symptoms versus probable cause have been prepared. These tables (3-2 through 3-6) appear on the following pages. Match the symptoms observed with the left column of the table, then make the checks shown. If the failure is associated with circuits that can be checked by program, run the appropriate diagnostic program to further isolate the fault before replacing components.

In general, the tables are organized so there is one table for each point at which the failure symptoms are observed. All the CRT failure symptoms are grouped together, for example.

Table 3-2. Serial I/O Failure Symptoms

Symptom	Possible Cause	Action to be Taken
No data output or no data input	1. I/O Controller Board	Test and replace
	2. Terminator or I/O cable	Check for cable and proper termination in place
	3. System Reset Board	Replace
No data exchange with only one peripheral unit	Address switch	Check address switch or plug and verify that peripheral unit is ready

Table 3-3. Small CRT Failure Symptoms

Symptom	Possible Cause	Action to be Taken
Display is blank	1. I/O Controller Board 2. CRT Module 3. Cable Connector 4. Processor/Memory Boards	Test and replace Replace Check Test and replace as necessary
Characters are incorrect	1. I/O Controller Board 2. Processor/Memory Boards	Test and replace Test and replace
Display lacks clarity	1. CRT Module 2. I/O Controller Board	Adjust or replace if necessary Test and replace
Display narrows from top to bottom	1. CRT Module 2. Loose Connector 3. Power Supply	Replace Check Check and replace
Top row compressed	1. I/O Controller Board 2. CRT Module	Test and replace Replace
Display not centered	CRT Module	Replace
Erroneous dot pattern for one character	Memory Board	Test and replace
CRT jitters	Power Supply	Test and replace
CRT jitters while a tape drive is in motion	1. Tape Drive 2. Power Supply	Test and replace Test and replace

Table 3-4. Power Failure Symptoms

Symptom	Possible Cause	Action to be Taken
Cooling fan not rotating and no lights on tape drives	1. AC power on/off switch not on or switch defective	Turn switch on and check switch output
	2. AC line cord loose	Check line cord
	3. AC line fuse blown	Check and replace fuse
Line fuse blows continually	1. Wrong size fuse	Check fuse rating
	2. Wiring defect	Locate wiring defect
	3. Power supply	Replace power supply
Cooling fan rotating too slowly	1. System wired for 220V	Notify sales representative
	2. 220V fan installed in 110V system	Replace with 110V fan
Burning odor	1. Power supply burned component	Measure outputs. If outputs are okay, isolate source of odor and replace
	2. Other subassemblies burned component	Isolate to a particular subassembly and replace

Table 3-5. Keyboard Failure Symptoms

Symptom	Possible Cause	Action to be Taken
Key sticks	1. Keyboard or keypad	Check for material binding the key, or remove key top and clean switch with Freon.
	2. Keyboard alignment	Re-align keyboard
Key is not accepted	1. Keyboard or keypad	Test and replace
	2. I/O Controller Board	Test and replace
Intermittent characters from keyboard	Keyboard	Check for short circuits or replace
No keys accepted	1. I/O Controller Board	Test and replace
	2. Keyboard	Test and replace
	3. Loose connector	Check
Group of keys malfunction	1. Keyboard	Test and replace
	2. I/O Controller Board	Test and replace
Wrong character from keyboard	1. Keyboard	Test and replace
	2. I/O Controller Board	Test and replace

Table 3-6. Tape Drive Failure Symptoms

Symptom	Possible Cause	Action to be Taken
Program does not load from drive 2	1. Tape cartridge installed improperly	Check tape loading procedure
	2. PROGRAM LOAD switch defective	Retry PROGRAM LOAD switch
	3. Undetermined	Change tape drive selection switch position on I/O controller board. If in proper position (up), place in drive 1 (down) position, mount program tape on other drive, and retry program load. If program does not load, restore switch to up position.

Table 3-6. Tape Drive Failure Symptoms (Continued)

Symptom	Possible Cause	Action to be Taken
Program does not load from drive 2 (continued)	4. Tape Amplifier Board	Replace
	5. Tape Drive 2	Check, clean, adjust, and replace as necessary.
	6. Processor Board	Replace
Program does not load from either drive	1. Processor/Memory Boards	Test and replace
	2. I/O Controller Board	Test and replace
	3. Power Supply	Check power supply voltages. If not satisfactory, adjust or replace.
	4. Tape Amplifier Board	Replace
	5. PROGRAM LOAD Switch	Replace
	6. System Reset Board	Replace
Intermittent program loading	1. Processor/Memory Boards	Test and replace
	2. I/O Controller Board	Test and replace
	3. Tape Drive	Check, clean, adjust, and replace as necessary.
	4. Tape Amplifier Board	Replace
System does not read or write on selected drive	1. Tape Drive	Check, clean, adjust, and replace as necessary.
	2. I/O Controller Board	Check connections. Test and replace.
Read and/or write errors	1. Dirty read head	Clean tape drive
	2. Dirty write head	
	3. Speeds off	Adjust speeds

Table 3-6. Tape Drive Failure Symptoms (Continued)

Symptom	Possible Cause	Action to be Taken
Read retries on good records	1. Read head	Clean or, if necessary, replace
	2. Speeds off	Adjust speed
	3. Power Supply	Measure outputs, adjust, or replace
	4. Tape Amplifier Board	Replace
Writes unreadable records (Write Errors)	1. Dirty write head	Clean write head
	2. Speeds off	Adjust speed
	3. Tape Deck	Replace
	4. Memory Board	Test and replace
Tape runaway	1. Broken leader 2. Broken tape	} Inspect tape drive and replace tape or leader.
Gives tape runaway status	1. I/O Controller Board	Test and replace
	2. Tape Deck	Inspect and replace
Noisy or erratic motion	Tape Drive	Inspect and replace
Worn Capstan (visible groove)	Capstan	Replace
Leader has trouble returning to "home" position	1. Dirty Capstan	Clean Capstan
	2. Faulty Leader	Replace Leader
	3. Clip-in switch	Inspect and replace tape drive
Slack in take-up leader when lock arm is closed with a cartridge in	1. Power Supply	Check power supply outputs. Replace if necessary.
	2. Tape drive lock arm switch/activator	Inspect and replace tape drive
	3. Loose plug at motion control board	Check connection

Table 3-6. Tape Drive Failure Symptoms (Continued)

Symptom	Possible Cause	Action to be Taken
EOT sensor not lighted on any drive	Power Supply	Check power supply output voltages per power supply adjustment procedure. If not satisfactory, replace power supply.
Tape drive EOT sensor not lighted on only one drive	1. Tape Drive 2. Loose cable connection	Check and replace tape drive Check connectors
EOT not sensed	1. EOT Sensor 2. Tape Drive	Replace Check and replace
One drive will not rewind	1. Tape drive 2. I/O Controller Board	Check and replace tape drive Test and replace
Neither drive will rewind, or neither drive will more forward	1. Power Supply 2. I/O Controller Board	Check outputs. If necessary, replace Test and replace
Drive will not run at slow speed (only high speed is possible)	1. Tape drive tachometer bad 2. Broken wire from tachometer	Replace tape drive Repair
Drive speed wrong or not constant	Tape drive tachometer	Adjust tape speed. Replace tape drive.
Tape drives rotate too slowly	1. Tape Drive 2. Power Supply	Adjust tape speed Check outputs. If not satisfactory, adjust or replace.

3-6. REMOVAL AND REPLACEMENT PROCEDURES.

Included in the following paragraphs are the removal and replacement procedures for the components normally replaced during field maintenance. Figure 3-1 shows the component locations for the 1501 terminal. In all cases, turn off power to the unit before performing a removal and replacement procedure.

Processor/Memory Board Removal and Replacement.

Proceed as follows to remove and replace the processor board and the memory board mounted on it:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. The processor and memory boards are accessible on the right side of the machine, the processor board is the second board from the bottom of the chassis.

d. To remove the processor board, remove the board locks (located on each side of the board). Grasp board firmly and pull straight out. DO NOT bend board while removing.

e. To remove the memory board, loosen the two memory board retainer screws until the retainers can be rotated away from the board. Pull the board up slightly, then out of the memory connector, taking care not to bend the board excessively.

f. Install the new memory in the processor board, component side up. Put the memory board retainers back in the lock position.

g. To re-insert processor board, place board in board guides with memory facing up and towards back panel connector. Slide the board until it hits against back panel connector. Now place both thumbs on edge of processor board, at the same time grasping cover cabinet. Push firmly with your thumbs until a very loud click is heard. The processor board is now seated in back panel connector.

h. After board is inserted in machine, secure all boards by replacing the board locks.

i. Reassemble by following step b then a.

I/O Controller and Communications Boards Removal and Replacement.

Perform the following steps to remove and replace either the I/O controller or communications board:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. The boards are accessible on the right side of the machine. Refer to figure 3-1 for location of boards.

d. Remove connector from the board.

e. For the I/O controller, remove the board locks (located on each side of the board). The communications board has no board locks. Grasp board firmly and pull straight out. DO NOT bend board up when removing.

f. To re-insert, place board in board guides with component side facing up. Slide board until it hits against back panel connector. Now place both thumbs on edge of board, at the same time grasping the lower cabinet, push firmly with your thumbs until a very loud click is heard. The board is now seated in the back panel connector. Make sure the drive select switch on the I/O controller board is positioned toward the operator.

g. If removed earlier, replace the board locks.

h. Reassemble by following steps d through a in reverse order.

Keyboard Removal and Replacement.

Perform the following steps to remove and replace the keyboard:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Using 1/4-inch nutdriver, remove the two mounting screws on the left side of the keyboard and loosen the three mounting screws on the right side.

d. Withdraw the keyboard several inches and take the connector off the upper right corner of the keyboard assembly. The keyboard is now free and can be withdrawn from the machine.

e. Before replacing the keyboard, be certain the PC board will not be shorted to either the aluminized case or the PROGRAM LOAD switch by applying insulating tape as required.

f. Reassemble by following these procedure steps in reverse order. When reassembling, the keyboard may not be properly centered with respect to the opening in the cabinet top. It may be necessary to remove the cabinet top after reassembly and shift the position of the keyboard.

CRT Visual Display Removal and Replacement.

Perform the following steps to remove and replace the CRT visual display module:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Using the Phillips screwdriver, remove the ground wire from the power supply chassis.

d. Using the 1/4-inch nutdriver, remove the four mounting screws.

e. Carefully lift the CRT assembly several inches and disconnect the connectors of the cable harness. The assembly is now free to be removed.

f. To replace the CRT display unit, follow this procedure starting with step e, working in reverse order.

System Reset Board Removal and Replacement.

Perform the following steps to remove and replace the system reset board:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. The reset board is located under the lower left corner of the power supply. Remove the hex-head screws holding down the BNC bracket.

d. Remove the two reset board mounting screws and the connector plug.

e. Remove the BNC bracket and the reset board as a unit from the system by loosening the screws holding the bracket and the reset board to the base.

f. Reassemble by following these steps in reverse order.

Cartridge Tape Drive Removal and Replacement.

Perform the following steps to remove and replace the cartridge tape drive unit:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. With a small Phillips screwdriver, loosen both read/write head assembly retaining screws. (See figure 3-2.) Push the pre-amp retainer bracket away from the pre-amp just far enough to remove the pre-amp from the read head. Remove the pre-amp cable from the clamp.

d. Remove the mounting screws from tape drive. Lift the tape drive several inches. You will notice that some green ground wires are secured to the tape drive by a screw on the bottom of the drive; loosen this screw and remove the wires.

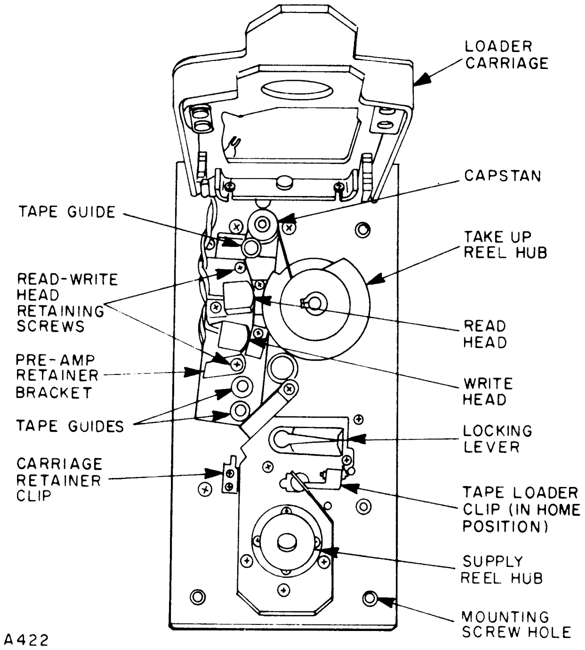


Figure 3-2. Cartridge Tape Drive

e. Next, remove cable connector (plug J1, figure 3-3) from bottom center of the motion control board affixed to the tape drive. The tape drive is now free to be withdrawn from the system.

f. To replace the tape drive, follow this procedure starting with step e, and working in reverse order.

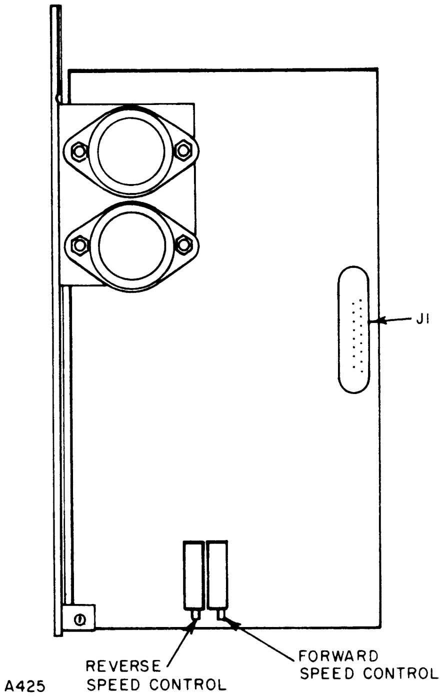


Figure 3-3. Location of Tape Speed Adjustments on the Motion Control Board

Tape Amplifier Board Removal and Replacement.

Perform the following steps to remove and replace the tape amplifier board:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. The tape amplifier board is located directly beneath tape drive #2. To gain access to the tape amplifier board, remove tape drive #2.

d. After tape drive #2 is removed, remove the pre-amp from tape drive #1.

e. Disconnect cable from connector J3.

f. Using the 1/4-inch nutdriver, loosen and remove the four corner mounting screws. A ground lead will be secured by one of the screws and will be removed with the screw. Four insulating washers may be present between the tape amplifier board and the cabinet.

g. Reassembly: When replacing the part, the four insulating washers, if required, should be aligned above the screw holes. It is very important that the pre-amplifier assembly cables be crossed. (Left pre-amp to drive 2; right to drive 1.) Reassemble by following these procedure steps in the reverse order.

Capstan Removal and Replacement.

A tape capstan can be removed by applying the Capstan Puller Tool (T-19221) to the capstan and turning the tool's screw until the part has been lifted free of the drive. Figure 3-4 shows the capstan being removed.

Proper orientation of the new capstan should be determined as the motor shaft is tapered and the capstan will only fit one way. The new capstan should seat so that the motor shaft is not farther than 1/16-inch above or below the top surface of the capstan.

The new capstan (Singer Part No. 9009295-91) should seat with thumb pressure after applying Lok-tite Quick Set 404 to the shaft.

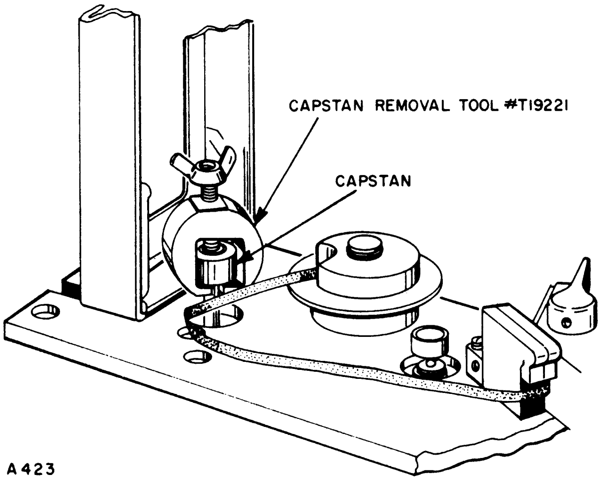


Figure 3-4. Capstan Removal

Wired Module Removal and Replacement.

Perform the following steps to remove and replace the 1501 wired module:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Remove the board locks.

d. Remove the processor board, the I/O controller board and the communications option board, if applicable.

e. Now loosen and remove the two left power supply mounting screws using the 1/4-inch nutdriver. Also remove the keyboards.

f. Loosen the three rack mounting screws half-way.

g. Using the long nose pliers, remove all the wires that attach the wired module to the DC terminal block, the reset board, and the program load/ reset switch assembly.

h. Remove the four wired module mounting screws.

i. While lifting the left side of the rack 1/2 inch, remove the wired module.

j. To replace the wired module, follow steps h through a in reverse order.

NOTE

> Exercise care in rewiring the module; incorrect wiring could damage system components. The program load wires should be connected first followed by the reset wires, and then the DC power wires.

Cooling Fan Removal and Replacement.

Perform the following steps to remove and replace the 1501 cooling fan:

a. Turn the AC power switch to off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Using the 1/4-inch nutdriver, remove the two mounting screws from the top corners of the fan frame.

d. Move the fan frame away from the system far enough so that the two AC input wires can be removed, using the long nose pliers, from the fan terminals.

e. The fan can now be completely withdrawn from the system.

f. Before the replacement fan is secured to the system, check the air flow direction. Air flow must be toward the back of the system (its direction is illustrated by a small arrow on the side of the fan frame). Then follow this procedure starting with step d, working in reverse order. Care should be taken in tightening down the fan mounting screws, if they are over-tightened the frame of the fan can be distorted and the fan blade will hit the frame.

Power Supply Removal and Replacement.

Perform the following steps to remove and replace the 1501 power supply assembly:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Remove the pair of green ground wires from each side of the power supply using a Phillips screwdriver.

d. Remove the four mounting screws using the 1/4-inch nutdriver.

e. Disconnect the DC voltage connector from the power supply.

f. Tilt power supply and disconnect the AC input wires. Extract the power supply.

g. To install a new power supply, attach the AC wires to pins 2 and 5* on the AC terminal block of the power supply.

h. Check voltages.

i. Perform steps e through a in reverse order.

*Terminal jumpers are different for different AC voltages.

Voltage	Jumper
90-110	1&2, 5&6
104-129	2&3, 4&5
208-258	3&4.

SIO Channel Bootstrap Loader Board Removal and Replacement.

This procedure applies to only the 1501-CL models. Perform the following steps to remove and replace the bootstrap loader board:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Grasp the lower side edges of the cabinet cover and spread the bottom outward slightly; then lift cover straight up.

c. The bootstrap loader board is attached to the "D" connector of the processor board. Remove it as follows:

   (1) Remove the ribbon cable from socket A1 of the bootstrap loader board.

   (2) Remove the bootstrap loader board from the processor board by lightly pulling it off the "D" connector.

d. To install or replace the bootstrap loader board, proceed as follows:

   (1) Ensure that the pigtail wire found between sockets A3 and A4 of the bootstrap loader board is inserted into the attached connector plug at location 21.

   (2) Place the plug (with the bootstrap loader board attached) on the "D" edge connector of the processor board.

   (3) Route the ribbon cable between the vacant G5 location of the processor board and location A1 of the bootstrap loader board.

e. Replace the cabinet top cover. Tighten the four screws.

3-7. ADJUSTMENT PROCEDURES.

During troubleshooting of a failure or during a preventive maintenance inspection, it may be necessary to make some equipment adjustments. Subsequent paragraphs explain the circumstances under which adjustments to the CRT visual display, speaker volume, tape speed, and power supply voltage levels should be made.

CRT Display Adjustments.

The CRT contains six different controls (see figure 3-5) that can be used to adjust the display. The CONTRAST, BRIGHTNESS, and FOCUS controls effect the presentation of the characters. VERTICAL SIZE effects the spacing between characters. VERTICAL LINE effects the side-to-side positioning of the display. Rotating the yoke makes the display level.

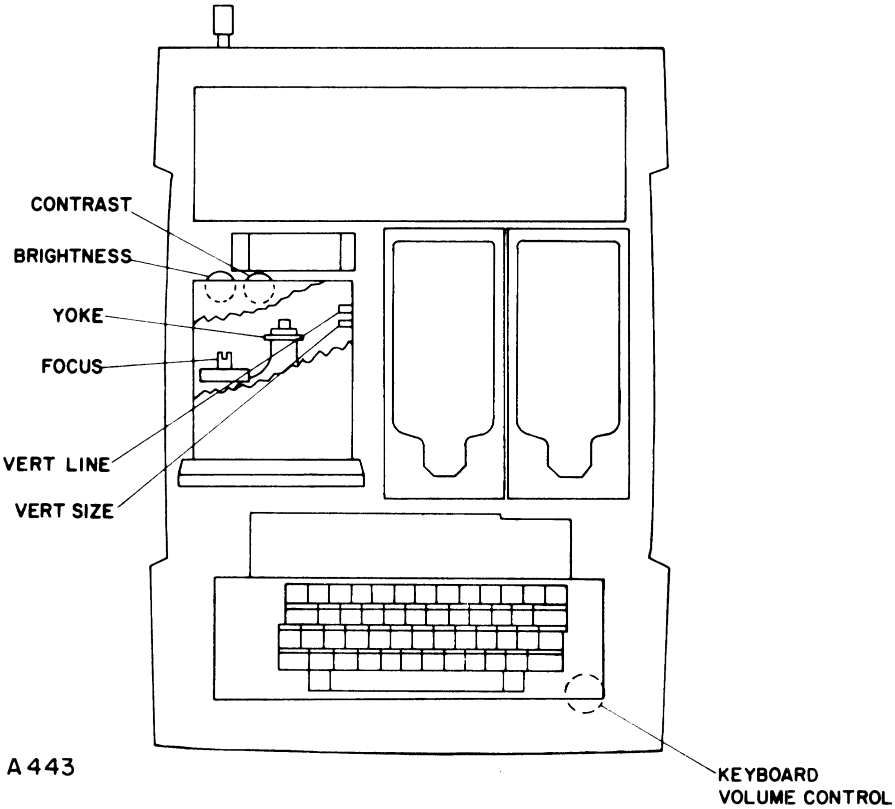


Figure 3-5. CRT/Speaker Adjustments

The CONTRAST and BRIGHTNESS controls are fingertip-adjustable; FOCUS is a screwdriver adjustment; VERTICAL SIZE and VERTICAL LINE are adjusted with a plastic tuning wand. The following paragraph outlines a detailed procedure to be followed.

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Insert line cord into an AC outlet and turn power switch on.

d. Obtain a CRT display by running the CRT Display Test (20009).

CAUTION

> Hazardous voltages exist in certain portions of the display circuitry. Personal contact with exposed circuits should be avoided.

e. Adjust the BRIGHTNESS and CONTRAST controls on the rear of the CRT module.

f. To adjust VERTICAL SIZE, VERTICAL LINE or the YOKE, turn power off and remove the four screws that fasten the CRT to the chassis.

g. Loosen the yoke screw enough so that the yoke can be rotated.

h. Turn power switch on and reload the CRT Display Test.

i. Turn the yoke with an insulated tool until the display is level.

j. Use the plastic wand to adjust VERTICAL SIZE and/or VERTICAL LINE, if required.

k. Turn power off, retighten the yoke screw and replace the CRT unit.

l. Retest display for quality. If satisfactory, turn the power off.

m. Carefully replace the cabinet top straight down and insert the screw fasteners.

Speaker Volume Adjustment.

The speaker volume may be adjusted to the level desired by the operator. One program available for use in making this adjustment is the Keyboard Test (any program that uses a keyboard beep may be used). Load the program into the 1501 and follow the instructions on the CRT until the tone begins to pulsate.

a. While the speaker is sounding, slide the front of the terminal three inches past the edge of the table on which it is resting such that the speaker volume control is visible on the underside of the cabinet (see figure 3-5).

b. Using a #1 Phillips screwdriver, adjust this control to achieve the desired volume level. Turn clockwise for increased volume, counterclockwise for decreased volume.

c. Slide the terminal back to its original location on the table.

Tape Speed Adjustment.

Tape drive speeds are adjustable and must be checked and/or adjusted when a tape drive is replaced. The Automatic System Test Program 20003 is employed for this purpose. The Field Engineering Diagnostic Manual describes use of the automatic system test.

The following procedure is to be used for tape speed adjustments:

Materials required:

1 - medium-sized screwdriver, 1/8-inch

1 - Automatic Systems Test Tape

a. Load the Automatic Systems Test Tape and note the forward and reverse speeds.

b. If the tape speed needs adjustment, turn the AC power switch off.

c. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

d. Turn system on.

e. Locate the motion control board on the left side of the tape drive. Refer to figure 3-3 for the location of pots that govern the forward and reverse speeds.

f. To alter the speeds using the 1/8-inch screwdriver, turn the adjustable resistors as follows:

Forward Speed (Lower Pot, R19)

Faster - Clockwise
Slower - Counterclockwise

Reverse Speed (Upper Pot, R24)

Faster - Counterclockwise
Slower - Clockwise

g. Since the forward and reverse speeds are interdependent, it may be necessary to readjust the speeds until both the forward and reverse fall within the range displayed.

h. If proper speeds cannot be obtained, remove tape drive for subsequent troubleshooting.

Power Supply Adjustment.

Perform the following steps to adjust the power supply:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Turn the system on and allow 15 minutes of warm-up before making any adjustments.

d. All DC voltage measurements will be made at the DC terminal block. Refer to figure 3-6 for location of adjustments. The voltage levels should be as follows:

T.B. Ref.	Wire Color	Setting	Maximum Ripple (Peak-to-Peak)
+12	Red	+12.0V	100mv
-12	White/Red	-12.0V	60mv
+ 5	Orange	+ 5.0V	50mv
+10	Yellow	+19.0V	200mv
- 7	White/Yellow	+21.5V	200mv

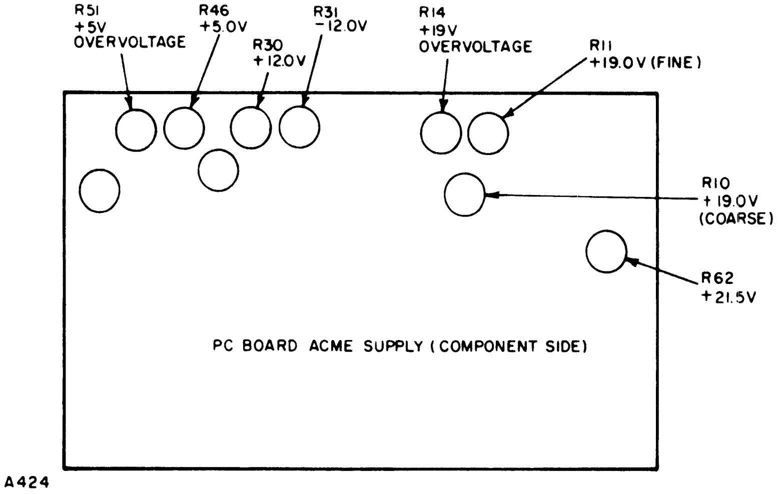


Figure 3-6. Location of Adjustment Potentiometers

e. Set R11 (the +19.0V fine adjustment) in the midpoint of its swing. Adjust R10 (the +19.0V course adjustment) until the voltage is approximately +19.0V at the DC terminal block. Adjust R11 until the voltage is exactly +19.0V.

NOTE

If +19.0V cannot be obtained, it may be necessary to adjust the overvoltage pot R14. A clockwise adjustment of R14 will increase the voltage required to activate the overvoltage circuit.

f. Adjust R46 to obtain +5.0V at the DC terminal block.

NOTE

If +5.0V cannot be obtained, it may be necessary to adjust the overvoltage pot R51. A clockwise adjustment of R51 will increase the voltage required to activate the overvoltage circuit.

g. Adjust R62 to obtain +21.5V at the DC terminal block.

NOTE

> If +21.5V cannot be obtained, it may be necessary to readjust the +19.0V level. Increase the output voltage of the +19.0V course adjustment R10. Decrease the fine adjustment pot R11 to reduce the voltage to the correct value. This should give the needed voltage to obtain +21.5V with R62.

h. Adjust R30 to obtain +12.0V at the DC terminal block.

i. Adjust R31 to obtain -12.0V at the DC terminal block.

j. Turn the AC power switch off and carefully replace the cabinet and insert the screw fasteners.

Power Supply Overvoltage Adjustment.

The +5V overvoltage circuit is required to be activated between the ranges of 5.6V minimum and 6.2V maximum. Potentiometer R51 is used to adjust the circuit to activate within the specified range.

The +19V has a similar overvoltage circuit. It may be adjusted using the same procedure as the +5V circuit. The range for activation of this circuit is 21V minimum to 22V maximum. R14 adjusts the overvoltage level for the +19V.

Proceed as follows to adjust the +5V overvoltage circuit:

a. Remove the two orange wires from the DC terminal block. These are the two +5V wires coming from the power supply. This is to adjust the +5V overvoltage circuit without damaging the circuit boards.

b. Turn R51 clockwise as far as it will go. This will set the overvoltage at its upper limit.

c. Using a DC voltmeter, monitor the +5V while adjusting R46. Adjust R46 so that the +5V level falls between 5.6V and 6.2V.

d. Turn R51 counterclockwise until the overvoltage circuit activates and the +5V is shut down.

e. Turn AC power off.

f. Turn R46 back to approximately its original position.

g. Place the two orange wires back into the DC terminal block.

h. Turn power on and readjust the +5V.

Proceed as follows to adjust the +19V overvoltage circuit:

a. Remove the yellow and white/yellow wires from the DC terminal block. These are the +19V and +21.5V wires from the power supply. Removal allows adjustment of the +19V overvoltage circuit without damage to the memory board.

b. Turn R14 clockwise as far as it will go. This will set the overvoltage at its upper limit.

c. Using a DC voltmeter, monitor the +19V (yellow wire) while adjusting R10 (coarse) and R11 (fine); the +19V voltage level pots, so that the +19V level falls between 21V minimum, and 22V maximum.

d. Turn R14 counterclockwise until the overvoltage circuit activates and the +19V is shut down.

e. Turn AC power off.

f. Turn R10 and R11 back to approximately their original positions.

g. Place the yellow and white/yellow wires back into the DC terminal block.

h. Turn power on, and readjust the +19V and +21.5V.

Power Supply Adjustment (Acme P/N PS-1-61449).

Note that this procedure applies only to the Acme Power Supply with part number PS-1-61449 which is used in the newer units. Perform the following steps to adjust the power supply:

a. Turn the AC power switch off.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up while grasping its bottom side edges and pulling slightly outward.

c. Turn the system on and allow 15 minutes of warm-up before making any adjustments.

d. All DC voltage measurements will be made at the DC terminal block. Refer to figure 3-7 for location of adjustments. The voltage levels should be as follows:

T.B. Ref.	Wire Color	Setting	Maximum Ripple (Peak-to-Peak)
+12	Red	+12.0V	100mv
-12	White/Red	-12.0V	60mv
+ 5	Orange	+ 5.0V	50mv
+10	Yellow	+19.0V	200mv
- 7	White/Yellow	+21.5V	200mv

NOTE

Only the voltage and overvoltage potentiometers should be adjusted; all other adjustments will be checked and set when the supply is returned for repair.

e. Adjust R7 to obtain +19V at the terminal block. If +19.0V cannot be obtained, it may be necessary to adjust the overvoltage potentiometer R8. Refer to the +19V overvoltage adjustment procedure below.

f. Adjust R20 to obtain +5.0V at the terminal block. If +5.0V cannot be obtained, it may be necessary to adjust the overvoltage potentiometer R21. Refer to the +5V overvoltage adjustment procedure below.

g. Adjust R28 to obtain +21.5V at the terminal block. Note that adjustment of the +19V will affect the 21.5V output.

h. Adjust R12 to obtain +12.0V at the terminal block.

i. Adjust R16 to obtain -12.0V at the terminal block.

j. Turn the AC power switch off and carefully replace the cabinet top and insert the fasteners.

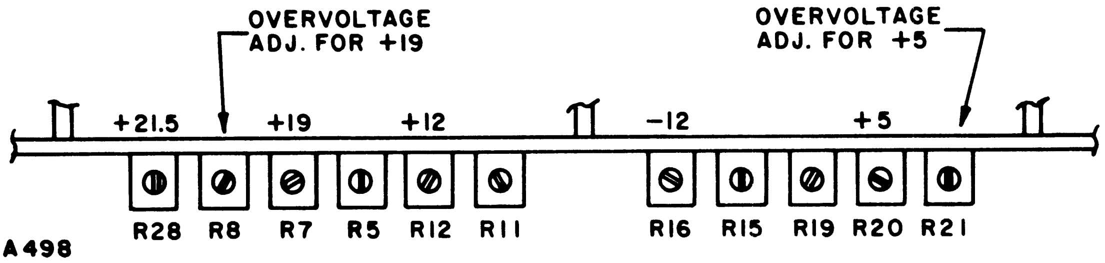


Figure 3-7. Location of Adjustments for Power Supply PS-1-61449

Power Supply Overvoltage Adjustment (Acme P/N PS-1-61449).

Note that this procedure applies to only the Acme Power Supply with part number PS-1-61449. The +5V overvoltage circuit is required to be activated between the ranges of 5.6V minimum and 6.2V maximum. Potentiometer R21 is used to adjust the overvoltage circuit to activate within the specified range.

The +19V has a similar overvoltage circuit. It can be adjusted using the same procedure as the +5V circuit. The range for activation of this circuit is 21V minimum to 22V maximum. R8 adjusts the overvoltage level for the +19V.

Proceed as follows to adjust the +5V overvoltage circuit:

a. Remove the two orange wires from the DC terminal block. These are the two +5V wires coming from the power supply. This allows adjustment of the +5V overvoltage circuit without damaging the circuit boards.

b. Turn R21 clockwise as far as it will go. This will set the overvoltage at its upper limit.

c. Using a DC voltmeter, monitor the +5V while adjusting R20. Adjust R20 so that the +5V level falls between 5.6V and 6.2V.

d. Turn R21 counterclockwise until the overvoltage circuit activates and the +5V is shut down.

e. Turn R20 counterclockwise as far as it will go.

f. Replace the two orange wires in the DC terminal block.

g. Turn R20 clockwise to obtain +5V at the DC terminal block.

Proceed as follows to adjust the +19V overvoltage circuit:

a. Remove the yellow and white/yellow wires from the DC terminal block. These are the +19V and +21.5V wires from the power supply. Removal allows adjustment of the +19V overvoltage circuit without damage to the memory board.

b. Turn R8 clockwise as far as it will go. This will set the overvoltage at its upper limit.

c. Using a DC voltmeter, monitor the +19V while adjusting R7 (the +19V voltage level potentiometer) so that the +19V level falls between 21V minimum and 22V maximum.

d. Turn R8 counterclockwise until the overvoltage circuit activates and the +19V is shut down.

e. Turn R7 counterclockwise as far as it will go.

f. Replace yellow and white/yellow wires in DC terminal block.

g. Turn R7 clockwise to obtain +19V at the DC terminal block.

h. Check the +21.5V output and adjust R28 as necessary.

SECTION 4

INSTALLATION

4-1. GENERAL.

This section of the service manual provides the instructions required to unpack, install, and checkout the Model 1501 Intelligent Terminal.

Complete installation and checkout instructions are also packed with the equipment. Either those instructions or this section of the manual can be used to install the terminal. It is expected that the procedure packed with the terminal will be discarded after being used. Therefore, the procedure in this section is available if the equipment must be repacked for shipment to another location and re-installed.

4-2. TOOLS AND EQUIPMENT REQUIRED.

Unpacking, installation, and checkout can be completed easily with a minimum of tools. Listed below are the tools needed to unpack and set up the equipment and the special test tapes needed for system checkout:

a. 6-inch screwdriver

b. 12-inch screwdriver

c. Phillips screwdrivers

d. 7/16-inch box wrench

e. Nutdriver set

f. Digital voltmeter

g. Series 1500 Field Engineering Diagnostic Manual

h. Automatic Systems Test Tape (20003)

i. Memory Test IV Test Tape (20131)

j. Series 1500 Systems Test Library (30022) or
Series 1501A Systems Test Library (30024)

k. If applicable:

Series 1501-FF Test Library (30011)
Communications Test Library (30010)
Series 1500 I/O Test Library (30015)

4-3. UNPACKING PROCEDURE.

The Model 1501 Intelligent Terminal is shipped in one carton, as shown in figure 4-1. Unpack the equipment as follows:

a. Examine the shipping carton for signs of damage and record any such damage.

NOTE

Retain packing material as required to repack and move the unit.

b. Verify that the box contains the parts listed:

(1) 1501 Intelligent Terminal

(2) Power cord

(3) Cleaning kit

(4) Installation instructions

(5) Installation report

(6) Cartridge labels

c. Remove plastic bag containing power cord.

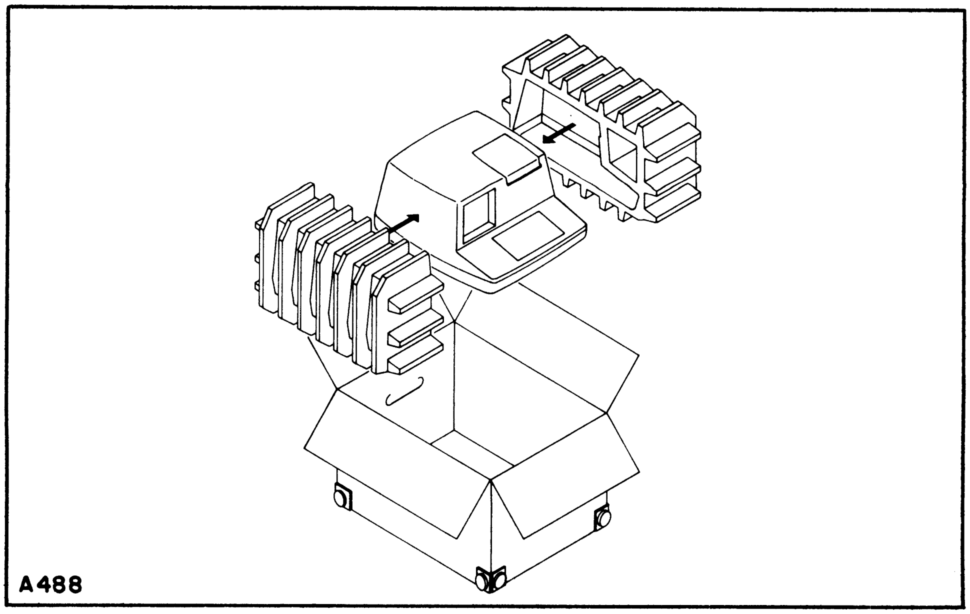


Figure 4-1. Model 1501 Intelligent Terminal Packing

4-4. INSTALLATION PROCEDURE.

Proceed as follows to install the Model 1501 Intelligent Terminal:

a. Remove the 1501 Intelligent Terminal from its shipping container and set the terminal on a table.

b. Remove the four cabinet top screw fasteners. Close the tape drive access door and lift the cabinet top straight up, while grasping its bottom side edges and pulling slightly outward.

c. Remove the two board locks. Pull the processor and other printed circuit boards out and reseat them. Reinstall the board locks.

d. If a communications board is to be installed, unpack it and install it and the communications cable in the 1501 at this time.

e. If a communications board and cable is installed in step d, remove the semicircular popout cover on the lower right side of the unit. The communications cable runs through this cutout.

f. Remove the power cord from the packing bag and connect the female end to the socket in the rear of the unit.

g. Insuring that the 1501 terminal power switch is in the OFF position, connect the power cable to the power source.

h. Switch on the power to the 1501.

4-5. SYSTEMS VERIFICATION.

The following steps verify the correct installation and checks the system components:

a. Run the Automatic Systems Test on both tape drives.

b. Run two passes of Memory Tester IV with parity check.

c. Run the applicable Series 1500 Systems Test Library (30022 or 30024).

d. If communications is installed, run the applicable communications test from the Communications Test Library (30010).

e. For Model 1501-FF run the Series 1501-FF Test Library (30011) in conjunction with the Series 1500 Test Library (30022).

f. For any serial I/O units, run the applicable tests on the Series 1500 Serial I/O Test Library.

SECTION 5

REFERENCE DATA

This section provides three types of information: (1) an explanation of how to use the logic diagrams provided in Volume II, (2) a glossary of terms in which the meaning of signals labeled on the logic diagrams is given appears in table 5-1, and (3) an interconnection diagram of the Intelligent Terminal is shown in figure 5-1.

In general, each printed circuit board in the Intelligent Terminal has a separate logic diagram. This diagram appears in Volume II of the Service Manual, and is preceeded by a board layout drawing in which the physical location of each circuit is shown.

The logic diagrams use symbols that are commonly used by the manufacturers of most electronic equipment, however, there are some points that require explanation. They are:

a. Signals entering or leaving a board terminate in a circle. The letter in the circle indicates which board connector (A, B, C, or D) is used, and the number in the circle is the pin number within that connector.

b. Interconnections between boards are made by either cables or wired modules. It is necessary to locate the cable or wired module wiring list in order to trace signals. The interconnection diagram shown in this section gives the number of the cable or wired module.

c. An "N" or a "P" prefix is added to each signal label. This indicates whether the active state of the signal is low (N) or high (P).

d. Each logic symbol is labeled with a logic element number, "127" for example, to identify that circuit. Also included in each symbol is the location of the circuit on the board (F7, for example) and the manufacturer's model number, such as 7400.

e. When a signal line is broken, either to go to another sheet, to go to another circuit on the same sheet, or to go to another board, the origin or destination of that signal is labeled, as follows:

   (1) *146, for example, indicates that the signal is connected to another circuit on the same sheet. In this case, logic element 146.

   (2) 2-106, for example, indicates that the signal is connected to logic element 106 on sheet 2 of this logic diagram.

   (3) If the signal begins or ends at a circle, that signal is connected to another board, and it is necessary to use a cable diagram or wired module diagram to determine which board and which pin.

The signal glossary is intended to be used to find the meaning of a signal name when the abbreviation is insufficient. It is organized in alphabetical order, with the prefix "N" or "P" left out. The signal meaning given in the glossary indicates what condition the signal represents.

Table 5-1. Signal Glossary

Signal	Meaning
A-EQ-B	A-input equals B-input (from adder)
A, B, C through G	Stages of display counter
A-MODE	Adder mode selection line
ACC-0 through ACC-7	Accumulator position output lines
ACC=MDR, ACC>MDR, and ACC<MDR	Accumulator equals, is greater than, is less than, respectively, the contents of the MDR.
ACC-LD	Accumulator load
AD0 through AD13	Address-0 through -13 (memory address lines)
ADDR-ENBL	Address enable
AFUN-S0 through S3	Adder function selection lines
ALPHA-SH	Lower case (alphabet) signal from keyboard
ASB	Automatic stack and branch
B_X_-ADDB.	Gate B register position _X_ to adder input B
BR-OFFSET-OP	Branch offset operation (class 0 instruction)

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
BR-DIRECT	Branch direct
C-FORWARD	Carry forward
C-OUT	Carry out (from adder)
CALL-ZONE	Character code memory access signal during CRT operations
CART-IN-A, B	Cartridge-in signals from cartridge tape drives
CART-OUT	Cartridge out status line
CH-7.	Channel 7. 177 instruction selected the parallel input-output channel
CHAR COUNT CTL	Character count control
CL-0, -1, -2	Processor clock signals
CLAMP-LINE	Puts +5V on serial I/O line
CLIP-IN-A, B	Cartridge tape clip in home position signals from drives A and B, respectively
CLIP-OUT-D	Cartridge clip out of home position status line
CLR-TP-BUFF	Clear tape buffer
COMM	Communications
CONT'L.	Control function key signal from keyboard
COMP-OP	Compare operation

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
COMP-EXIT	Comparison exit
CRT	CRT selected
CRT-ADDR-DISABLE	CRT address disable
CRT-EN	CRT enable
CRT-ZONE-SELECT	Timing pulse that loads the zone register when the CRT has been selected.
CRT-UNBLANK	Unblanks CRT
CTL-BL	CRT blanking control
DATA-IN	Data to write head for tape write operation
DATA-IN-0 through 7	Data to memory
DATA-OUT-0 through 7	Memory output data lines
DATA-Q and QG	Serial data from cartridge tape drive
EOT	End-of-tape status line
EOT-A, B	Cartridge tape drive end-of-tape signals
EX-ACC < MDR and EX-ACC > MDR	External output of condition register
EXT-ADV	External advance
EX-BIT-0 through 7	External data to memory input
EX-CH-7	External channel 7. 177 instruction selected the parallel input-output channel
EX-CLK-EDGE	External output of clock edge

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
EX-CL-0 through CL-2	External output of processor clocks CL-0 through CL-2
EXECUTE-TBO	Execute take branch operation
EX-E2 and E3	External output of execution cycles 2 and 3
EX-I/O-CLR	External I/O clear signal
EX-I1 through I4	External output of cycles I1 through I4
EX-I01	External output, cycle I01
EX-MEM-0 through 7	External memory output lines
EX-MDR-0 through 7	External output of memory data register
EX-SP-1 through 4	External output of stack pointer 1 through 4
EXT-ENTRY	External entry
EXTERNAL 156 OP	External output of 156 operation signal
EX-T-SUM	External output of timing pulse T-SUM
EX-TE	External output of timing pulse TE
EXTERNAL-INTRPT	External interrupt signal
EXU-COND	Execute condition
EXT-WRITE	External write
EX 20-213	External memory address bits 20-213
F-TRIG	Fast trigger. Causes CRT retrace.
FHS	Forward, high speed

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
FNS-WE.	Forward, normal speed, with erase
FNS-WOE	Forward, normal speed, without erase
GACC-ADDA	Gate accumulator to adder A input
GEN-CLR	General clear
HS-SKIP	High-speed skip signal from external device
I-102	I cycle 1 or 2
I-CYCLE	Initial cycle
I1 through I4	Instruction cycles
IDLE-T, R	Idle code transmitted or received, respectively
INDEX-0	No index register specified
INH-ACC-LD.	Inhibit accumulator load
INT-INHIBIT	Interrupt inhibit
INH-SET	Inhibit set
INT-OVRFL	Interrupt overflow
INT-TIME-CNTL	Interrupt time control
INTL.	Interleave
INST-RESET.	Instruction reset
I/O	Input-output. When used in I/O controller, indicates that SIO channel is selected by I/O instruction.
I/O-CLR	Timing pulse I/O clear
I/O-CLR-GATE.	I/O clear pulse gated by stop flip-flop

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
I/O-OP	I/O instruction. Causes I/O controller to decode instruction.
I/O-RD	I/O read
I/O-STALL	Stall signal to processor
IO1	Instruction cycle IO1 (cycle before I1)
IX-SECTION-211 through 213	Index register section bits
KB-0 through -7	Character byte from keyboard
KB-ERROR-F/F	Keyboard error flip-flop
KBD-CHAR-READY	Keyboard character ready
KBRD-IN	Keyboard input present
LOGIC-OP	Logical instruction (Exclusive OR, AND, or OR).
LATCH 6 to 9	Latch (high state 6 to state 9)
MAR	Memory address register
MAR-HIGH, -LOW	Gate high-order or low-order bits from MAR
MO through M7	Memory output data
MDRX-ADDB	Gate MDR position X to adder input B
MEM-CLOCK	Memory clock
MEM-WRITE-SW	External memory write switch. Enables memory to accept external input data.
MDR 0 through 5	Memory data register output
NUM-SH	Upper case (numbers) function key from keyboard
OP-0 through OP-7	Operation register lines

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
OP-177	I/O instruction 177 is present
OSC.	Processor 6-MHz oscillator
PL-SW.	Program load switch
PRG-INT-SW	Program interrupt switch
PAIR-1 through 4	Tape drive pair select signals
PULSED-ENERGY.	Energy detected from tape drives
PRG-LOAD	Program load. Program load switch depressed with power on.
PROC-CLR	Processor clear
PWR-ON-SW.	Reset pulse initiated by closing of AC power switch
PRG-CLR.	Program clear
QØ	Serial data from stage 0 of shift register
REC-LINE	Serial I/O receive line
RNS.	Reverse, normal speed
RST-SW	System reset switch
RUN-D.	Tape run command
REV-A, B	Cartridge tape drive reverse commands
RUN-A, B	Cartridge tape drive run signals
RUN-F/F.	Tape run flip-flop
SECTION-BR	Section branch
SP-0 through -4.	Stack pointer positions 0 through 4
SET-ERASE.	Set erase mode in cartridge tape drives

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
SET-RWND.	Tape rewind command
STROBE.	Keyboard data strobe. Causes N-KYBD-CHAR-RDY signal
SUM-0 through 7	Adder output lines
STALL-COND.	Stall condition
SENSE-1	Load sense switch instruction
SWF	Set write fault
SWFG.	Set write fault-gated
ST.	Start or start trigger
SET-RWND-A, B	Cartridge tape drive rewind signals
STOP.	Puts processor and memory into an idle mode
S-TRIG.	Slow trigger. Retrace control for CRT.
SYG-CY.	Single cycle
TBL-A, -B	Tape buffer load A, B
TCA, B.	Cartridge tape drive A and B select signals, respectively
TE.	Timing pulse TE
TF.	Tape forward command
THS-A, B.	Tape high-speed (cartridge tape signals)
TK-BR-F/F	Take branch flip-flop
TAPE-A.	Cartridge tape drive A select signal
TAPE-ST-WRITE	Tape start write timing signal
TRAN-BYTE	Transfer byte

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
T-SUM.	Timing pulse T-SUM
TX	Timing pulse TX
TP-SPR	Tape sprocket
U.	U-bit
UV-GATE.	U and V bits gate
V.	V-bit
WRITE.	Write signal to memory
WRITE -A, B.	Write signals to cartridge tape drives
W.	Write
W-OR-R	Write or read
WRITE-CHK.	Write check
WRITE/CLK.	Write clock
WRITE-DATA	Write data (serial data to tape drive)
WRITE-MEM.	Write memory
XCL.	Transmit clock
X-STALL.	External stall signal
XMT.	Transmit
XMIT-LINE.	Serial I/O transmit line
01-WINDOW.	Zero or one window
1's F/F.	Ones flip-flop
176-DI-0 through 7	Communication channel data
176-DI	Enable communication channel data

Table 5-1. Signal Glossary (Continued)

Signal	Meaning
177-DI-0 through 7	Parallel I/O data
177-DI-ENABLE.	Enable parallel I/O (channel 7) data
20 through 213	Memory address net lines
176-INT.	Instruction to communications interface
Ø-L.	Zero literal
22X-I4	Cycle I4 of instruction 22X (LIA)

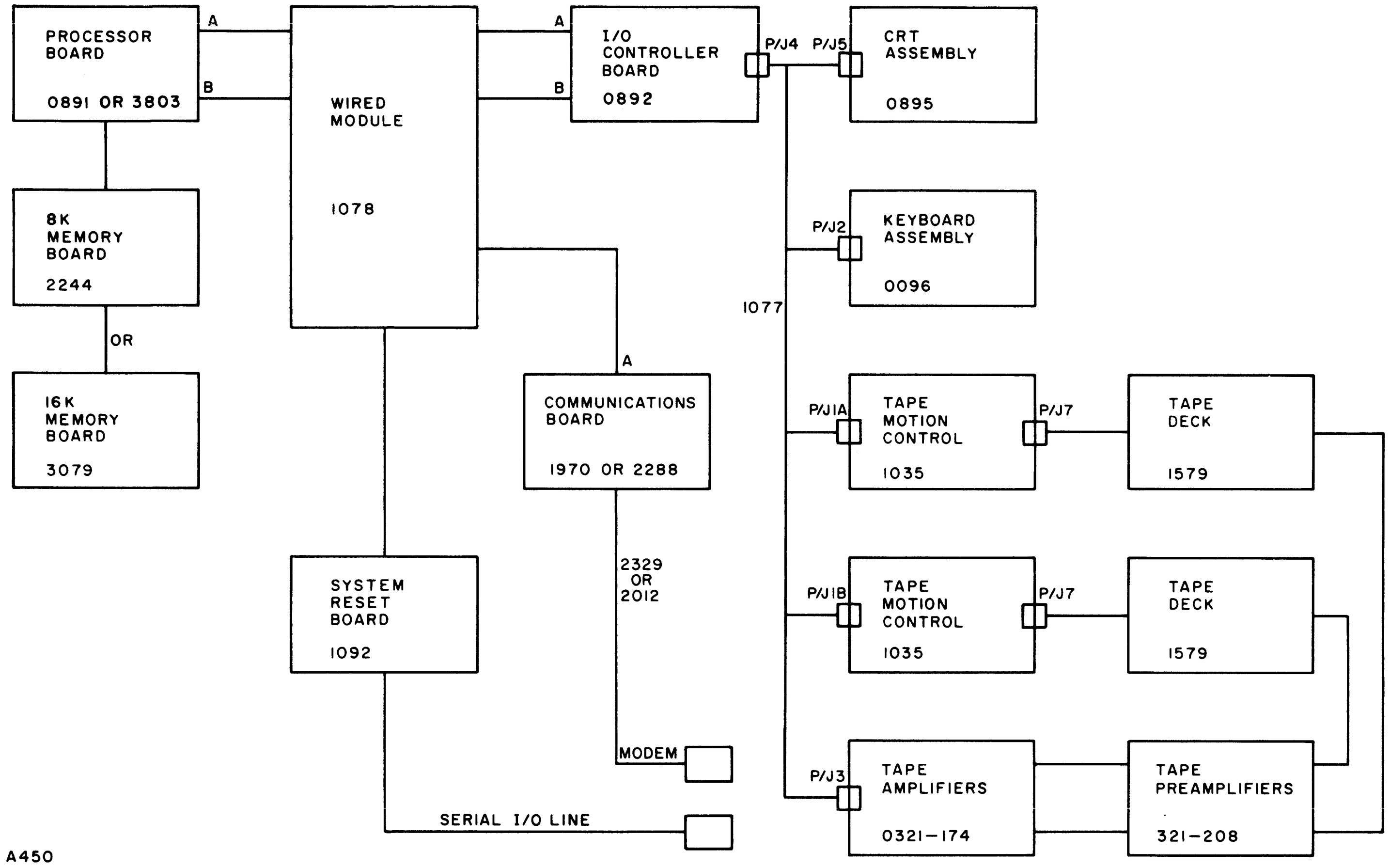


Figure 5-1. Model 1501 Intelligent Terminal Interconnection Diagram

SINGER
BUSINESS MACHINES