Systems Engineering Laboratories

Design Manual
Input/Output Interface
SEL 810B Computer

Design Manual
Input/Output Interface
SEL 810B Computer

August, 1968

310-095117-000
Price: $4.00

6901 West Sunrise Blvd., Fort Lauderdale, Florida 33310
Area Code 305 587-2900

LIST OF EFFECTIVE PAGES

The total number of pages is 105, as follows:

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

LIST OF ILLUSTRATIONS

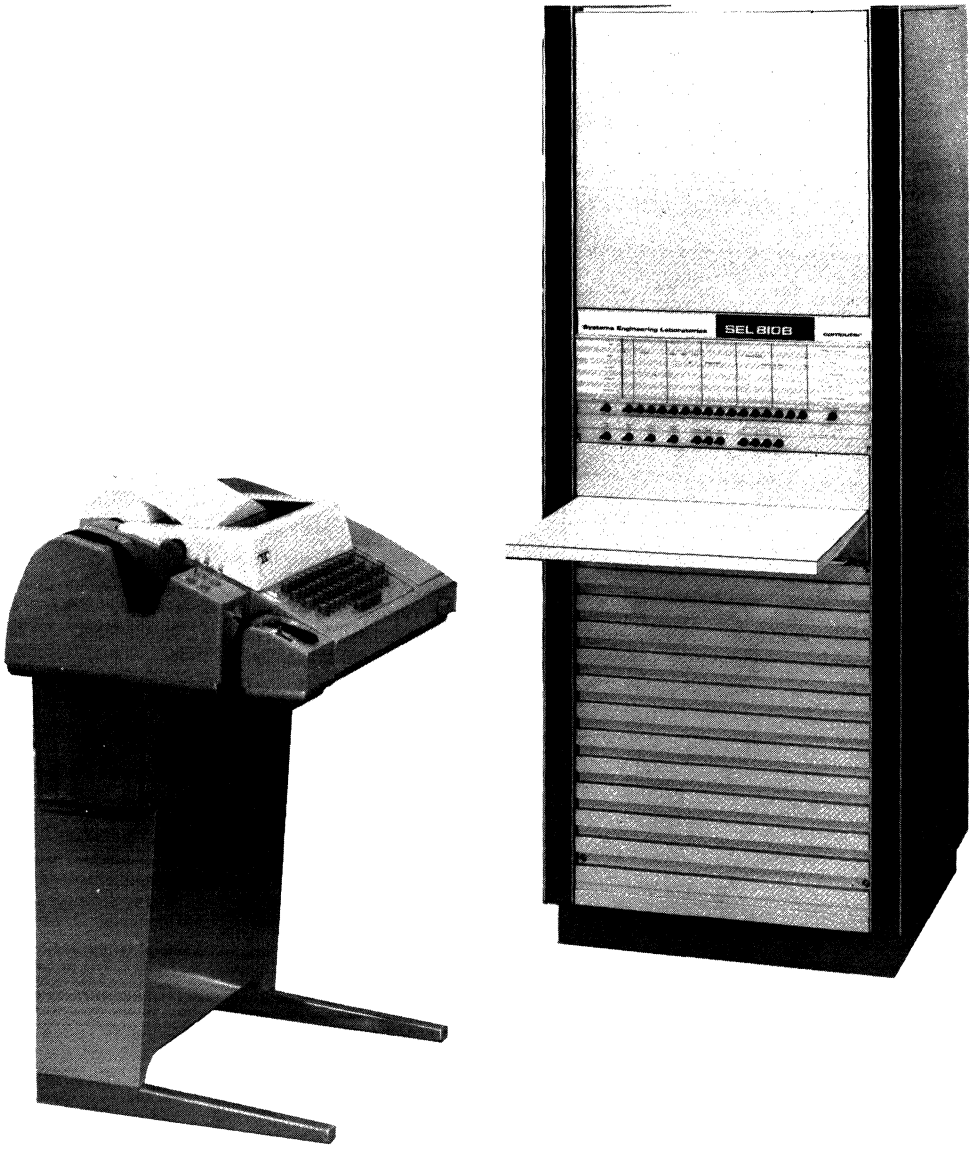
LIST OF ILLUSTRATIONS (CONT'D)

LIST OF TABLES

LIST OF RELATED PUBLICATIONS

The following publications contain information not included in this manual but necessary for a complete understanding of the SEL 810B Computer System.

Publication Title	Publication No.
SEL 810B Computer Drawings Manual	95116
SEL 810B Computer Reference Manual	95118
SEL 810B General Purpose Computer Operating Instructions	95119
SEL 810B Computer Technical Manual	95019
SEL Series 81-711/81-712 Typewriter Systems Technical Manual	95007
SEL Series 81-711/81-712 Typewriter Systems Drawings Manual	95070



SEL 810B General Purpose Computer (Minimum System Configuration)

SECTION I
GENERAL DESCRIPTION

INTRODUCTION

The SEL 810B Computer illustrated in the frontispiece is a high-speed, parallel operated, 16-bit binary machine. The full memory cycle time of 750 nanoseconds and a highly flexible input/output structure enables this computer to be suited for real time data collection, processing, and control applications.

SEL 810B COMPUTER CHARACTERISTICS

The SEL 810B Computer characteristics are:

- Manual control panel
- Silicon, monolithic, integrated logic circuits
- Sixteen-bits word length plus one parity bit
- 8, 192 word memory
- Full cycle time of 750 nanoseconds
- Memory parity bit with parity generator 1 checker
- Fully parallel operation
- Computation time including access and indexing

Add, Subtract	1.50 microseconds
Multiply	4.50 microseconds
Divide	8.25 microseconds

- Double length accumulator
- Index register (B-Accumulator)
- I/O structure capable of handling 64 peripheral device controllers
- Two separate levels of priority interrupt
- Sixteen sense switches
- Switch addressable program halt
- Power fail safe
- ASR-33 typewriter with paper tape reader and punch mounted on stand beside the computer
- Computer size: 24 inches wide, 62 inches high, 30 inches deep (45 inches including optional desk top)
- Typewriter size: 22 inches wide, 35 inches high, 18 inches deep
- Temperature environment, 50° to 95°F

SEL 810B COMPUTER OPTIONS

The SEL 810B Computer options are:

- Maximum of eight block transfer control (BTC) units, capable of transferring up to 1, 330, 000 words per second
- Computer graphics processor (CGP) units (maximum 6)
- Additional index register
- Memory expandable to 32, 767 (32 K) word locations
- 60-Hz clock
- Program protect and instruction trap feature for guarding blocks of memory against modification, and for preventing execution of privileged instruction
- Up to 98 individual levels of priority interrupts
- Variable base register, increases direct addressing capability
- ASR-35 console typewriter in place of ASR-33
- I/O parity checker and generator
- Instruction trap; can prevent execution of privileged instructions
- Stall alarm
- Auto start

SEL 810B COMPUTER STANDARD SOFTWARE

The SEL 810B Computer standard software characteristics are:

- Full ASA FORTRAN compiler; operates in 8 K memory
- FORTRAN library
- ASSEMBLER relocatable object format; macro capability, and extensive set of pseudo-operations
- Compiler/Assembler loader
- Utility routines such as, debugging aids, I/O handlers, and tape editor
- Maintenance routines; complete set for computer and peripheral devices

SEL 810B COMPUTER ASSOCIATED PERIPHERAL DEVICES

The SEL 810B Computer associated peripheral devices are:

- Card reader; 400 cards/minute
- Card punch; 100 cards/minute
- Paper tape reader (photoelectric); 300 characters/second
- Paper tape punch; 110 character/second
- Magnetic tape control unit; handles up to eight tape units

- Magnetic tape units - 45, 75, 120, 150 inches/second; 200, 556, 800 characters/ inch; seven and nine track
- Movable head disc file; 1.5 million word storage, 150 milliseconds maximum track access time (track 00 to track 99)
- Fixed head disc files; up to 909K, 16-bit word storage; 8.3 milliseconds average access time
- Typewriters; ASR-33, KSR-33, ASR-35, KSR-35, RO-33, RO-35 10 characters/ second.
- Line printers; 300, 600, 1000 lines/minute 120 columns/line
- Incremental plotters; 12-inch chart width (300 steps/second) and 31-inch chart width (200 steps/second)
- CRT display 16 inches, character generator, function switches, and light pen
- Interval Timer
- Interface subsystem components
- Multiplexers, low-level and high-level, solid state, and relay switching
- Sample and hold units
- Analog-to-digital converter, up to 15 bits binary; word rates to 50K words/second
- Custom interface
- Digital-to-analog converter, up to 12 bits binary

SEL 810B COMPUTER ORGANIZATION

The SEL 810B Computer is formed by four major units. These sections consist of:

- Input/Output
- Memory
- Control
- Arithmetic

Refer to the block diagram on figure 1-1. The combined control and arithmetic units are often referred to as the mainframe section.

INPUT/OUTPUT UNIT

The input/output structure, generally referred to as the input/output processor (I/OP)transfers data

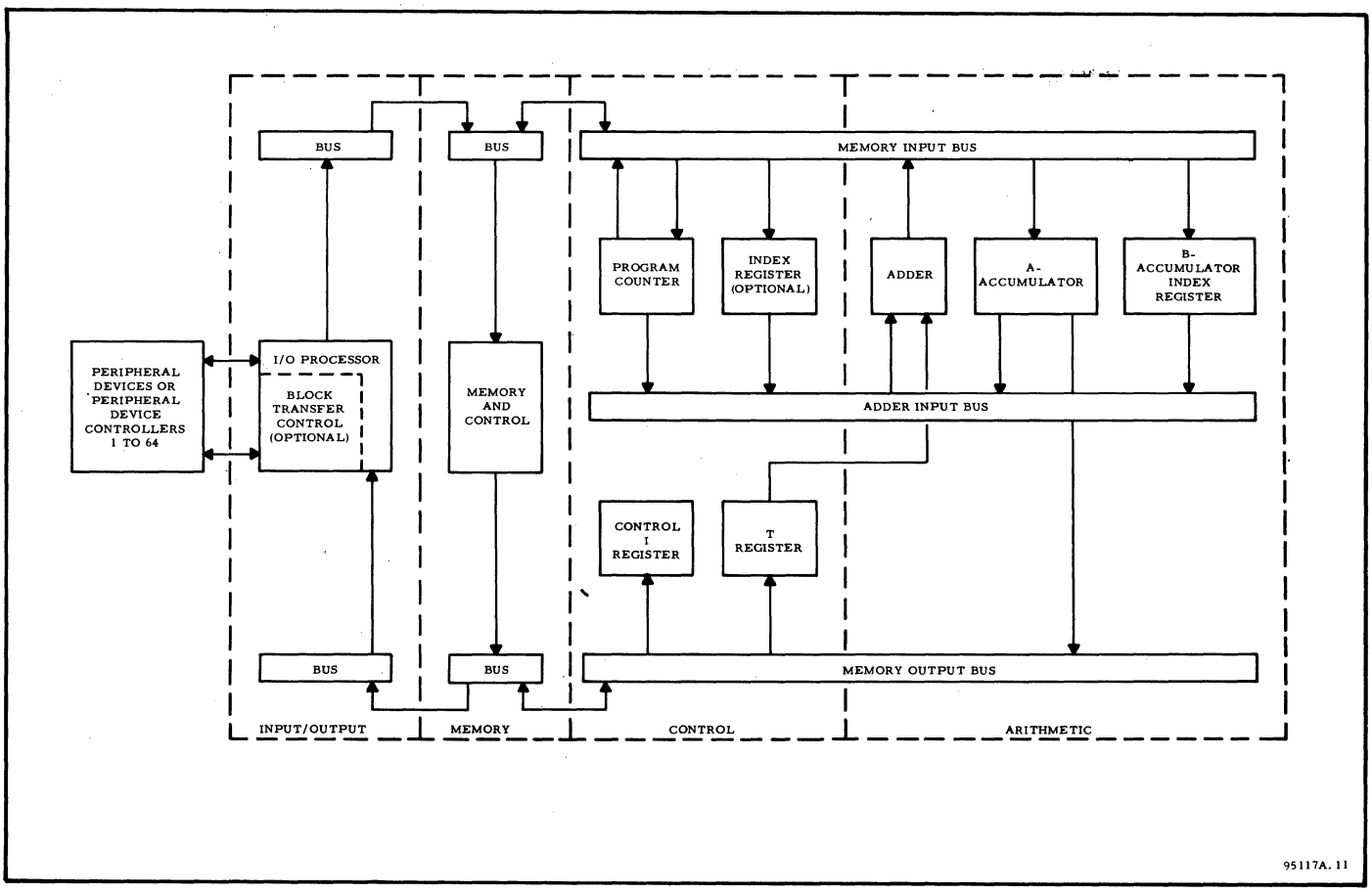


Figure 1-1. SEL 810B Computer Block Diagram

words, commands, and status reports between the computer and peripheral devices. The I/OP is capable of communicating with up to 64 peripheral device controllers. Device controllers such as magnetic tape control units, may communicate with several peripheral devices. Therefore the number of individual peripheral devices that the computer can communicate with is highly expandable. Refer to figure 1-2.

Data transfer instructions are provided which enable word transfers directly between computer memory and peripheral devices as well as between the A-Accumulator and peripheral devices. In addition, external device command and test instructions are provided.

The I/O instruction set is particularily powerful because each instruction causes several functions to be performed. First, execution of each I/O instruction causes a device to be connected to the computer. The device (unit) number is contained in each I/O instruction. Second, an automatic test is made of the device which determines if the device can execute the instruction. Third, the data or command transfer is made if the device is ready. Fourth, the device is disconnected. If the device is not ready when tested, the computer will either wait until the device is ready and then transfer or it will disconnect the device and advance the program counter to a reject location. A wait flag is provided in each I/O instruction, (except the test input TEU) to enable the programmer to specify the wait or skip mode of execution. The normal time required to perform the complete connect, test, transfer, and disconnect operation is only four machine cycles.

In addition to the basic I/O structure, up to eight fully buffered Block Transfer Control (BTC) channels or six computer Graphics Processors (CGP) may be added to the computer.

A BTC transfers blocks of data between the computer and associated peripheral devices at rates up to 1, 330, 000 words per second. One machine cycle is stolen from the computer per word transferred.

Transfers are made under hardware control, with block length varying from a single word to 32, 767 words. An automatic re-initialization feature allows chaining of blocks.

A priority interrupt system is provided enabling the computer to have up to 98 individual levels of priority interrupts. Interrrupts can be selectively enabled and disabled under program control. A unique memory location is assigned to each level.

An ASR-33 Typewriter, with paper tape reader, and punch, is supplied with the computer. This complete unit is referred to as the Computer Console Input/Output Station, or Console I/O Station. The typewriter may be operated either off-line, or on-line with the computer. When operating on-line, the input (reader and keyboard) and output (printer and punch) operate independently. For example: this permits a paper tape to be read in and a separate set of characters to be printed at the same time. The paper tape reader is under complete control of the program when operating on-line; the reader is started and stopped by a command from the computer. Operating speed is 20 characters per word for the reader and 10

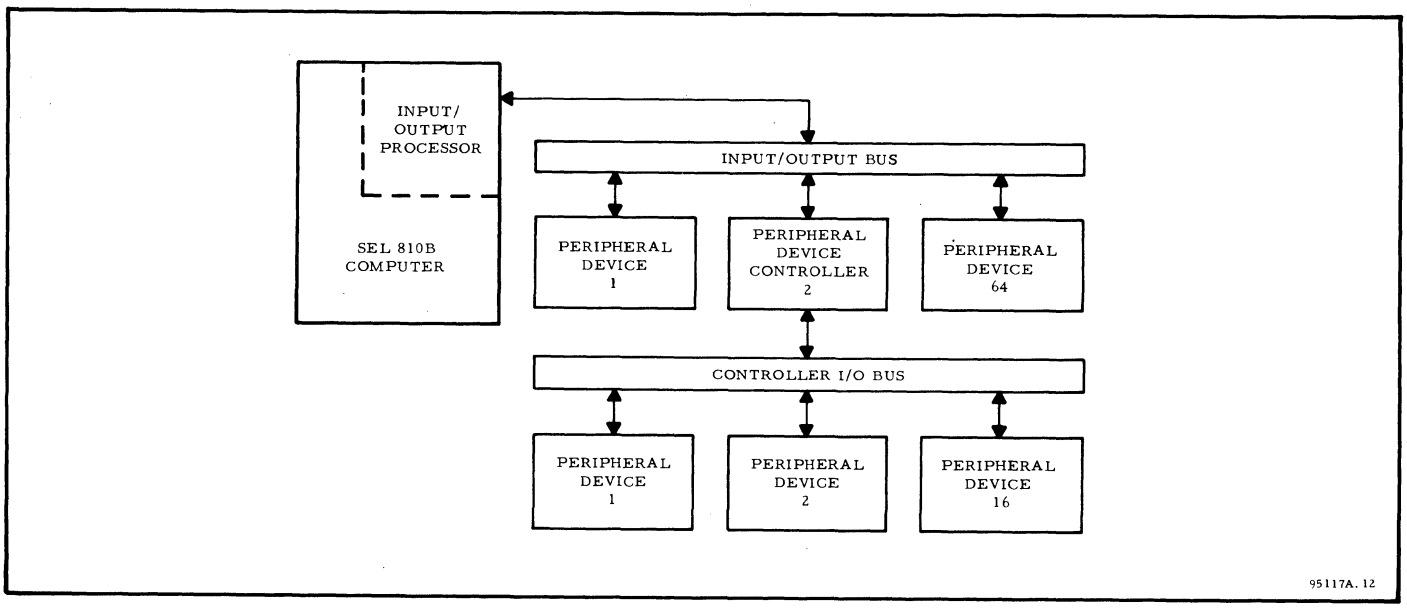


Figure 1-2. Connection of Peripheral Devices to the Computer

characters per second for the keyboard, printer and punch.

The ASR-33 typewriter may be replaced by ASR-35 upon request. The optional Computer Graphics Processor (CGP) is similar to the BTC except that the CGP examines each word as it comes from memory and either interprets the word as data, or as an instruction.

MEMORY UNIT

The memory unit stores the instruction words that define the operation of the computer (program) and the data words on which the computer operates.

The memory unit is composed of one, two, three, or four separate modules. Each module has 8,192 addressable storage locations. Each location consists of one 16-bit data or instruction word plus a parity bit. The total number of storage locations can range from 8,192 provided by the basic 8K module to 32,767 available with four 8K modules.

Individual modules are composed of these four elements:

- 8K x 17-Bit Magnetic Core Memory
- 13-Bit Memory Address Register
- 17-Bit Data Register
- Self-Contained Timing and Control

Instruction words and data words are loaded into specific addresses prior to the program execution. Loading may be performed manually through the panel controls or automatically from peripheral units through the use of the supplied loader program. Each input word is transferred to the memory data register and the accompanying storage address is transferred to the memory address register. When both registers have been loaded, a write command is issued by the program control unit and the 17-bits in the memory data register are written into the 17 magnetic cores addressed by the memory address register.

CONTROL UNIT

The control unit calls up and decodes the instruction word, then issues the commands to operate the computer.

The control unit contains a 15-bit binary program counter capable of directly addressing 32,767 memory locations. This counter supplies the addresses of the instruction words from which the computer operates. The counter is initially set to the address minus one of the first instruction of a program when the computer is started. It is then automatically advanced by each instruction until a Halt, Branch, or Conditional Skip instruction is read from memory. The Halt instruction stops the computer while the Branch instructions change the contents of the program counter to the operand address contained in the instruction. The Skip instructions cause the program counter to be advanced by either one or two locations, depending on the value of the Skip condition specified by the instructions.

The instruction words are read from memory into the instruction register and automatically restored in memory. The binary digits forming the instruction word are then applied to the operation control circuits. The unique codes assigned to each instruction are then decoded and used to provide timing and gating signals to the remainder of the machine. The signals from switches on the control console are also connected to the operation control circuits. External priority interrupts will cause the control circuits to switch the program counter to programs designed to process the external demand.

The memory cycle during which instruction words are read and decoded is referred to as the Instruction Cycle. Some instructions, called memory reference instructions, contain a memory address which specifies the location of an operand which is to be operated on by the computer. For these instructions, one or more additional memory cycles, called Execution Cycles, are required. During the instruction cycle, the memory address in the T-Register is supplied in part by the operand address contained in the instruction word and by the program counter. The operand is read from memory and operated upon according to signals provided by the operation code. Most memory reference instructions are accessed and executed in a total of two cycles. However, instructions such as multiply and divide require more than one execution cycle.

Many instruction words require no operand from memory and are executed completely within the instruction cycle. Others, while requiring no operand from memory, do require one or more execution cycles for completion. Chief among this latter group are the shift instructions. For these instructions, a group of bits within the instruction word defines the number of shifts to be performed while the operation code of the word defines the type of shifting to be done. Other instructions, notably the input/output control instructions, are composed of two instruction words; one defining the type of operation and the unit and the other

defining the actual operand or the operand memory location. The words forming these input/output instructions are automatically unloaded from memory in the proper sequence.

ARITHMETIC UNIT

The arithmetic unit directed by the control unit performs computations with the data words from memory. The data word formats are shown on figure 1-3.

The arithmetic unit consists of a 16-bit adder and several accessory storage registers. Two of these registers, the A-Accumulator and the B-Accumulator, may be loaded and unloaded by program control. The A-Accumulator is the primary arithmetic register and derives its name from its function of accumulating results of the arithmetic operations. Because only one word may be taken from the memory and input/output units by each instruction, the second operand in add and subtract operations must be loaded in a register prior to the add and subtract instructions. The A-Accumulator fulfills this function and also provides temporary storage for the result of the arithmetic operation. The B-Accumulator holds the multiplier during multiply operations and stores the least significant bits of the product. In addition to these strictly arithmetic functions, the two accumulators provide a convenient storage area for rearranging data words through shifting and logical operations.

A third register connected to the adder is the T-Register which holds the operand unloaded from the memory. This 16-bit register plus the 16-bit A- and B-Accumulators supply inputs to the 16-bit binary adder. When an add instruction is performed, the data words are simply added according to the rules of binary two's complement arithmetic.

The basic data format of the SEL 810B Computer is a 16-bit binary, single-precision fixed point word. This format contains the sign bit in bit position 0, with bit position 1 holding the most significant data bit and bit position 15 holding the least-significant bit. Two's complement representation is used for negative numbers. This format is defined as an integer with an imaginary binary point located to the right of bit position 15. The SEL 810A set of library integer subroutines assumes this representation. The programmer may, of course, scale single-precision words in any desired manner and utilize the extensive shift and test instruction repertoire to maintain the binary point location.

The SEL 810B Computer also accomodates double-precision data words of 30 bits plus sign through the use of the extended (B) accumulator. Each double-precision data word is normally stored in two adjacent memory locations with the most significant half stored in the first (lower) address. The product generated by a single-precision multiply is located in the A- and B-Accumulators in this format. The dividend is assumed to be in this double-precision format prior to the execution of the divide instruction.

Three floating point data formats are utilized by the SEL 810B Computer library. The single-precision floating point format consists of two words. The first word contains the sign and 15 most significant bits of the fractional mantissa; the second word contains the six least significant mantissa bits and the signed eight-bit exponent. The words are stored in adjacent memory locations with the first word located in the lower memory address. Both the mantissa and the exponents carry separate signs so that the mantissa may be positive or negative independent of the sign of the exponent. Two's complement representation is used for negative numbers.

A double-precision floating point format consisting of three memory words is provided for use with the set of double-precision floating point library subroutines. The third floating point data format (complex floating point data) is provided for the set of FORTRAN IV subroutines dealing with complex numbers.

The arithmetic unit includes two single bit registers which are addressable by the program. The first of these is the overflow latch which can be set during addition, subtraction, and division operations. The overflow for an add or subtract occurs when the result exceeds the accumulator capacity; a divide overflow occurs if the divisor is equal to, or smaller than, the dividend.

This latter overflow is due to the fact that the machine treats all divide arguments as double-precision numbers by scaling the single-precision divisor by 2^{15}. If the dividend is larger than the scaled divisor, the quotient will necessarily be a number greater than 2^{15}. Such a number exceeds the capacity of the 15-bit A-Accumulator in which the quotient is to be stored and this produces a false divide.

The overflow latch lights the overflow indicator on the control console and remains set until tested, and reset, by an SOF (skip no overflow) instruction. Because the latch remains set until tested, such a test should be made immediately following an arithmetic process when an overflow condition could result. This prevents the possiblity of a second overflow being undetected by the already set latch.

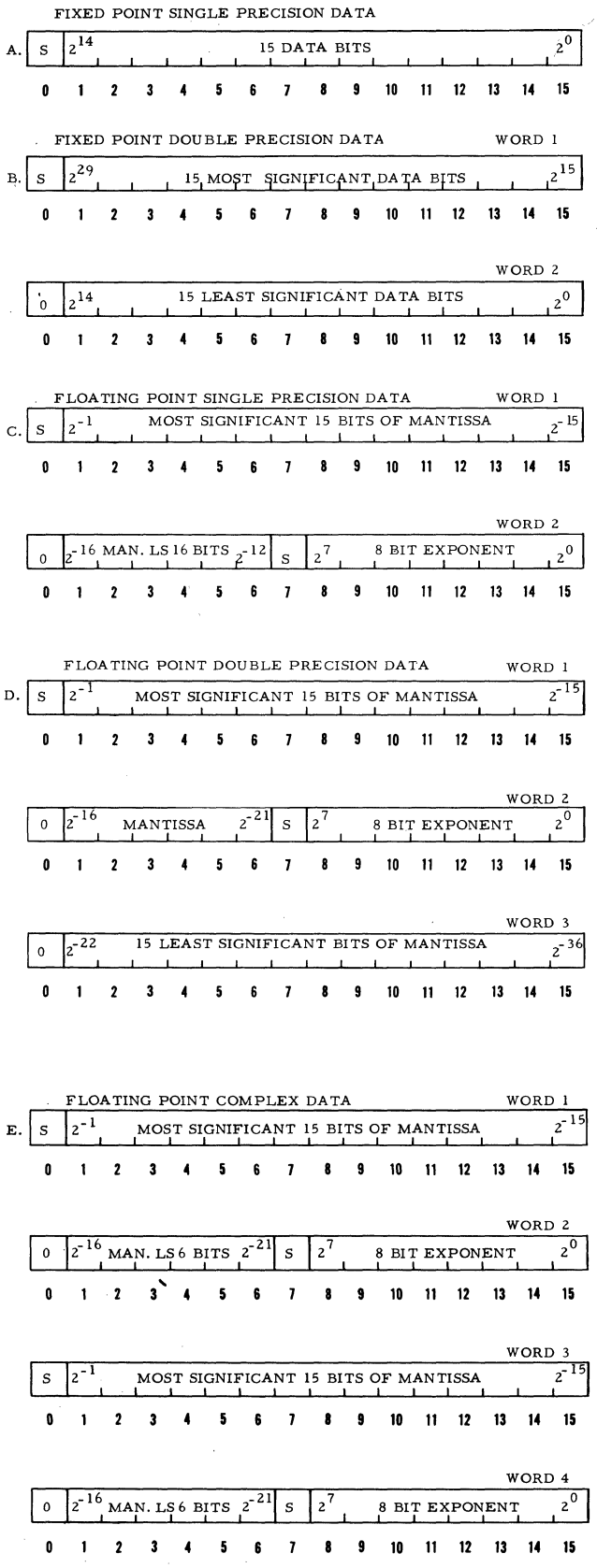


Figure 1-3. 810B Computer Word Formats

The overflow latch may be set with an OVS (set overflow) instruction.

The second addressable arithmetic latch is the carry latch which connects to the least significant bit of the parallel adder. This latch is set in the regular arithmetic processes to produce a two's complement number (one's complement of the number plus one). The latch is used in the addition and subtraction of double-precision numbers formed in the A- and B-Accumulators. The least significant words of the double-precision numbers are processed and stored in the B-Accumulator. If a carry or borrow is generated, it will cause the sign of the B-Accumulator to change. A CSB (copy sign of B) instruction is used to set the carry latch to the state of the B-Accumulator sign bit and then reset the B sign bit to ZERO (as required in the double-precision format). If the operation is addition, the True output of the carry latch is added together with the most significant word; if a subtract operation is in process, the False output of the carry which is added to the most significant word (effectively subtracting the borrow).

The CSB instruction should be followed immediately by the AMA or SMA instruction which operates on the most significant half of the double-precision operand, since the carry latch is cleared at the end of the execution of all instructions except CSB.

COMPUTER DATA FLOW

Most data transfers within the SEL 810B Computer moves all 16 bits of a computer word at once. This type of transfer is termed parallel as opposed to serial transfers where 16 separate bit transfers would be required to transfer a computer word. The exceptions to this general rule are the lateral shifts within and between the accumulators, transfers of flag bits between registers, and transfers of six or eight bit characters to and from peripheral devices.

Each major data patch (bus) consists of 16 (17 if parity and program protect is included) parallel signal lines between the source and destination registers or devices. Multiple-input OR gates are used on those buses which transfer data from more than one source. AND gates permit specific registers or units to be selected as the source and destination of the data. The memory modules contain input and output AND circuits, address registers, and data registers.

An instruction word read from a memory module is brought into that module's data register, gated through the 17-bit OR circuit to the memory output bus (MOB). From the MOB, it is then gated into the Instruction and T-Registers. If the instruction is a memory reference instruction, the address portion of the word (least significant bits) is applied to the nine low-order adder circuits. If a MAP bit is contained in the instruction, the six most significant bits of the program counter are applied to bits one through six of the adder. If the index flag is set in the instruction, the 16-bits (including sign) of the B-Accumulator or Index Register (depending on previous setting of the index pointer latch), are gated onto the adder input bus to be summed with the address.

If either or both the MAP and Index Flags are absent, the gates from the source register(s) remain closed. The output address of the adder is applied through 13 AND gates and input OR circuits to the memory addressing control (the module is selected from the two high-order address bits) and to the memory modules where it is gated into the selected module's address register.

A data word read from memory is applied through to the memory output bus and loaded in the T-Register. The data word may then be applied through the adder to the input AND gates of the A- or B-Accumulators. If an arithmetic operation is to be performed, the outputs of either the A- or B-Accumulators may be AND gated onto the adder input bus to be summed with the memory word. The result (or the unchanged data word, depending on the instruction) may be AND gated into either the A- or B-Accumulators.

When a data word is to be transferred from an accumulator register to memory, the AND gates of the desired source are opened to the adder input bus. The output of the adder is then AND and OR gated to the selected memory module and the parity generator. The word is entered into the module's data register along with the parity bit.

Branch instructions change the instruction addresses in the program register. The path for these changes are from the T-Register through the adder into the program register. These instructions contain the low-order, nine-bit address with the instruction word itself. To this may be appended (if the MAP flag is present) the upper six bits of the program register. In this case, those six bits are AND gated into bit positions 1 through 6 of the adder, and the remaining low-order nine bits are gated from the T-Register to the adder.

SECTION II
INPUT/OUTPUT ORGANIZATION

INTRODUCTION

The SEL 810B Computer is operated by a series of instruction words stored in memory. These instructions, each specifying an operation, are successively read from memory locations addressed by the program counter. Normally, the program counter is advanced one count after each instruction, and sequentially accesses the instruction word in the next memory address. The program counter may be preset to an address by a Branch instruction, but will continue its sequential upcount from the new address.

An instruction word is formed by 16 bits, each of which performs a particular function, defining the operation to be performed, addressing a memory location, defining the number of shifts, etc. The function of a particular bit will vary in different types of instructions. For example, in some words, bit 14 forms part of a memory address; in others, bit 14 forms part of the operation code. The function of the bits depends on the instruction word type defined by the four-bit operation code located in bits 0-3 of the first word of each instruction.

There are three major types of instruction words used by the SEL 810B Computer:

- Memory Reference Instructions - instructions containing memory addresses.
- Augmented Instructions - instructions containing additional code bits in lieu of memory addresses.
- Input/Output Instructions - instructions consisting of one or two separate words forming one instructions. Input/Output instructions contain augmenting code bits and, depending on the manner in which they are executed, may or may not contain memory addresses.

The input/output instruction group is discussed in this manual. For information pertaining to any other instruction, refer to the SEL 810B General Purpose Computer Reference Manual in the List of Related Publications.

INPUT / OUTPUT INSTRUCTIONS

GENERAL

Six instructions are provided to perform data input/output and external device control. Two instructions, A input (AIP) and A Output (AOP), are provided to enable words or characters to be transferred between the A-Accumulator and peripheral devices. These instructions provide a convenient character assembly/dis-assembly capability. Each of these instructions occupies a single memory location. The two instructions, Memory Input (MIP) and Memory Output (MOP), enable words or characters to be transferred directly between specified memory locations and peripheral devices. The instruction Command External Unit (CEU) enables all system devices connected to the computer to be controlled by the program. The CEU instruction is used to initiate Block Transfer Control (BTC) operations as well as to control computer peripheral devices and special system units. The Test External Unit (TEU) instruction is provided to enable system units to be tested by the computer. The test result causes the instruction following the TEU to be either executed or skipped. Two sequential memory locations are required to store the MIP, MOP, CEU, and TEU instructions, with the second word called for automatically by the control unit.

Input/Output instructions are all augmented instructions containing operation codes of 13_8 or 17_8 and all, except AIP and AOP, are formed by two memory words. The first word contains the operation code, three to five augment code bits and Indirect, MAP and Wait flags. The single-word AIP and AOP instructions contain the same data plus a Merge flag in the AIP instruction.

An illustration of the single and two word instruction format is shown on figure 2-1 and the analogy table 2-1. The augmenting code bits are located in bit positions 9, 10, and 11, with bit 11 having the prime significance as the Wait bit. The Wait bit feature is contained in all input/output instructions except the TEU instruction.

EXECUTION SEQUENCE

The basic, automatic execution sequence for all I/O instructions consists of three steps:

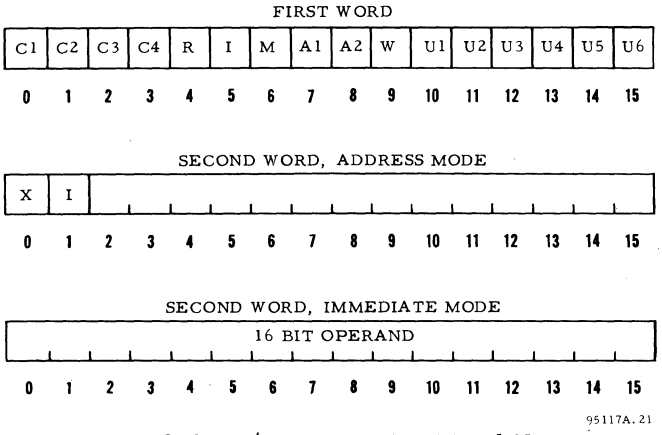


Figure 2-1. I/O Instruction Word Format

Table 2-1. I/O Instruction Word Analogy

Symbol	Definition	Instruction
C	Operation Code	All I/O Instructions
R	Character Merge Flag	AIP
I	Indirect Address Flag	MIP, MOP, CEU, TEU
M	Map Bit	MIP, MOP, CEU, TEU
A	Augmented Command Code	All I/O Instructions
W	Wait Flag	AIP, MIP, AOP, MOP, CEU
U	Unit (Device) Number	All I/O Instructions
X	Index Flag	MIP, MOP, CEU, TEU

a. Connect the device specified by the instruction to the I/O bus.

b. Execute the transfer directly between the device and the A-Accumulator or memory.

c. Disconnect the device from the I/O bus.

Three very significant features of this execution sequence are:

- The device is always specified by the I/O instruction.
- The device is always connected to and disconnected from the computer by the execution of the instruction.
- Data transfers are always made directly between the specified device and the computer with no intermediate buffering.

The result of these three features is that the computer I/O structure is always available for use without testing. It is nevery busy, except during the times that I/O instructions are being executed. In addition, no device selection instructions are required, since each I/O instruction causes the device specified by the instruction to be selected for transfer.

EXECUTION MODE

Data or command word transfer instructions may be executed in either of two modes - Wait Mode or Skip Mode.

Wait Mode

When the instruction is programmed in the Wait Mode (Bit 9 = One) the SEL 810B Computer program counter will halt at the current location until the addressed peripheral device signifies that it is ready to execute the command or transfer data. The SEL 810B Computer tests for the Ready signal each machine cycle, and then executes the transfer during the first cycle following the recognition of the Ready signal. After the transfer, the device is disconnected and the next instruction in sequence is executed. The specific meaning of the Ready signal is defined in each I/O instruction descrption.

Skip Mode

If the instruction is programmed in the Skip Mode (Bit 9 = Zero), the SEL 810B Computer tests the Ready status of the peripheral device once only. If the device is ready, the transfer is executed. The program register is then advanced by two, which causes the next instruction to be skipped. If the device indicates Not Ready, the device is disconnected and the program register is advanced by one. This conditional skip feature enables all I/O instructions (except TEU) to perform the following sequence of operations:

a. Connect the device.

b. Test for ready.

c. Transfer data if ready.

d. Disconnect the device.

A flow chart showing the execution of the AIP and AOP instructions is shown in figure 2-2. As shown in the flow chart, the state of the Wait Flag determines whether the instruction is executed in the Wait or Skip Mode. The MIP, MOP, and CEU

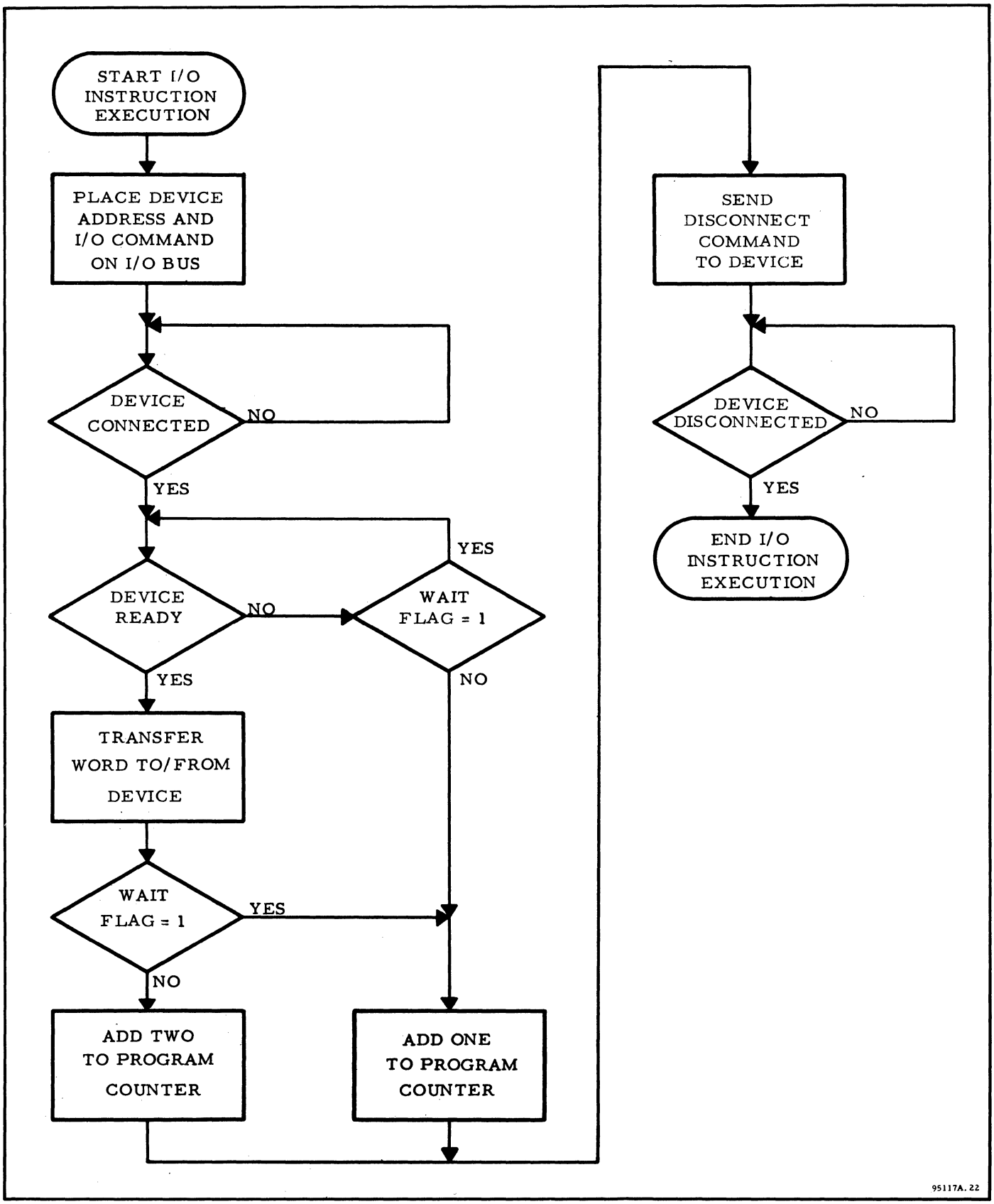


Figure 2-2. AIP/AOP Instruction Execution Flow Chart

instructions are executed in the same manner, except that the program counter is advanced by one before the transfer is made in order to obtain the operand address.

Execution of the TEU instruction requires no ready test. An on-line unit is always ready to be tested. The test word is always transferred to the device and a Test Return signal is tested. The result of the test is a conditional skip of the next instruction.

ADDRESSING SELECT

The two-word I/O instructions (MIP, MOP, CEU, and TEU) provides two selectable operand addressing modes - Immediate Mode and Address Mode. The addressing mode is specified in the first instruction word by the value of the Indirect Address Flag (I) bit. If I is a one, the Address Mode is executed; when I is a zero, the Immediate Mode is executed.

Immediate Mode

In this mode, the second instruction word is treated as the operand. In executing MOP, CEU, and TEU instructions, the contents of the second instruction location are transferred to the specified device. MIP execution consists of transferring a word or character from a specified device into the second instruction location.

Address Mode

In this mode, the second instruction word is interpreted as the operand address. The indirect address format is used in the second instruction word. Therefore, indexing and indirect chaining may be used in addressing the operand.

I/O INSTRUCTION FORMAT

I/O INSTRUCTION LISTING

The following paragraphs describe the formats for the listed input/output instructions:

- CEU (Command External Unit)
- TEU (Test External Unit)
- AOP (Accumulator Word Output to Peripheral)
- AIP (Accumulator Word Input from Peripheral)
- MOP (Memory Word Output to Peripheral)
- MIP (Memory Word Input from Peripheral)

COMMAND EXTERNAL UNIT

Code Selection

The following octal operation codes are provided for using the CEU instruction as demands require: (Refer to figure 2-3)

1300 = CEU, Immediate - Skip Mode
1301 = CEU, Immediate - Wait Mode
1320 = CEU, Address - Skip Mode
1321 = CEU, Address - Wait Mode
1330 = CEU, Address - Map - Skip Mode
1331 = CEU, Address - Map - Wait Mode

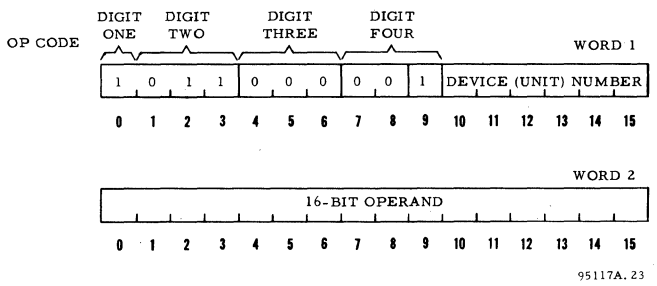


Figure 2-3. CEU Word Format

Description

Transfers the command second word (up to 16-bits) contained in the specified memory location to the peripheral specified in the device (unit) number.

Operand Address Modes (Word 1, Bit 5)

I = 0; Second Word Immediate Mode

I = 1; Second Word Address Mode

NOTE

To reference a 15-bit address, M is required (bit 6 = 1) to append the most significant 15th bit to the 15-bit address.

Execution Modes (Word 1, Bit 9)

W = 0; Skip Mode

W = 1; Wait Mode

Transfer Criterion

A device (unit) answers Ready to a CEU test if the device can immediately start execution of any new function command.

NOTE

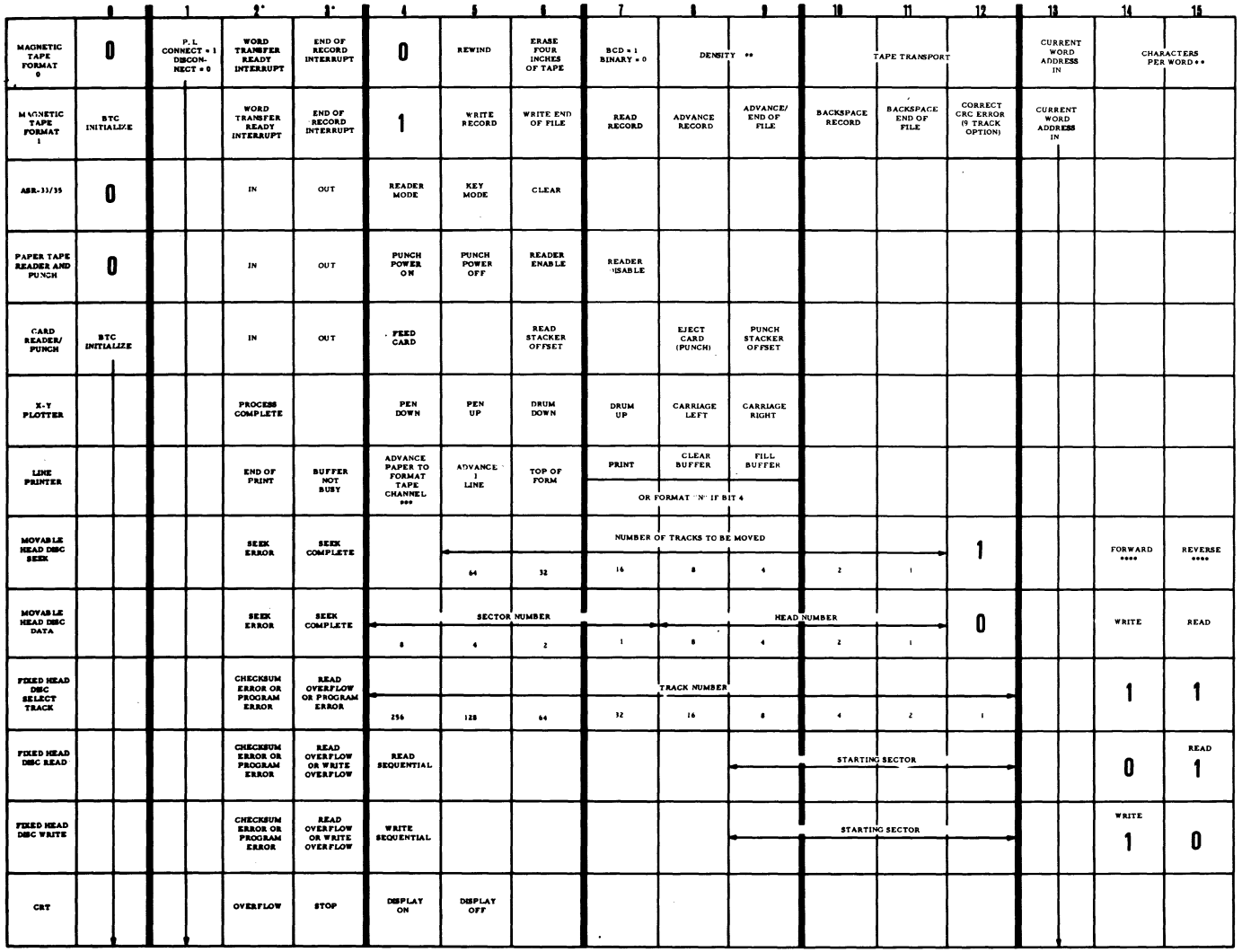
The bits in most device command codes are microprogrammed. Thus, either one or several function commands may be transferred to a device by execution of a single CEU instruction.

Timing

The execution time required to complete the CEU instruction is four cycles and wait.

CEU Second Word

The CEU second word format is shown in figure 2-4 which illustrates the 16-bit configuration applicable to the Systems Engineering Laboratories



	0	1	2*	3*	4	5	6	7	8	9	10	11	12	13	14	15
MAGNETIC TAPE FORMAT 0	0	P. I. CONNECT = 1 DISCONNECT = 0	WORD TRANSFER READY INTERRUPT	END OF RECORD INTERRUPT	0	REWIND	ERASE FOUR INCHES OF TAPE	BCD = 1 BINARY = 0	DENSITY **		TAPE TRANSPORT			CURRENT WORD ADDRESS IN	CHARACTERS PER WORD**	
MAGNETIC TAPE FORMAT 1	BTC INITIALIZE		WORD TRANSFER READY INTERRUPT	END OF RECORD INTERRUPT	1	WRITE RECORD	WRITE END OF FILE	READ RECORD	ADVANCE RECORD	ADVANCE/ END OF FILE	BACKSPACE RECORD	BACKSPACE END OF FILE	CORRECT CRC ERROR (9 TRACK OPTION)	CURRENT WORD ADDRESS IN		
ASR-33/35	0		IN	OUT	READER MODE	KEY MODE	CLEAR									
PAPER TAPE READER AND PUNCH	0		IN	OUT	PUNCH POWER ON	PUNCH POWER OFF	READER ENABLE	READER DISABLE								
CARD READER/ PUNCH	BTC INITIALIZE		IN	OUT	FEED CARD		READ STACKER OFFSET		EJECT CARD (PUNCH)	PUNCH STACKER OFFSET						
X-Y PLOTTER			PROCESS COMPLETE		PEN DOWN	PEN UP	DRUM DOWN	DRUM UP	CARRIAGE LEFT	CARRIAGE RIGHT						
LINE PRINTER			END OF PRINT	BUFFER NOT BUSY	ADVANCE PAPER TO FORMAT TAPE CHANNEL ***	ADVANCE 1 LINE	TOP OF FORM	PRINT	CLEAR BUFFER	FILL BUFFER						
								OR FORMAT "N" IF BIT 4								
MOVABLE HEAD DISC SEEK			SEEK ERROR	SEEK COMPLETE				NUMBER OF TRACKS TO BE MOVED					1		FORWARD ****	REVERSE ****
						64	32	16	8	4	2	1				
MOVABLE HEAD DISC DATA			SEEK ERROR	SEEK COMPLETE	SECTOR NUMBER				HEAD NUMBER				0		WRITE	READ
					8	4	2	1	8	4	2	1				
FIXED HEAD DISC SELECT TRACK			CHECKSUM ERROR OR PROGRAM ERROR	READ OVERFLOW OR PROGRAM ERROR				TRACK NUMBER							1	1
					256	128	64	32	16	8	4	2	1			
FIXED HEAD DISC READ			CHECKSUM ERROR OR PROGRAM ERROR	READ OVERFLOW OR WRITE OVERFLOW	READ SEQUENTIAL					STARTING SECTOR					0	READ 1
FIXED HEAD DISC WRITE			CHECKSUM ERROR OR PROGRAM ERROR	READ OVERFLOW OR WRITE OVERFLOW	WRITE SEQUENTIAL					STARTING SECTOR					WRITE 1	0
CRT			OVERFLOW	STOP	DISPLAY ON	DISPLAY OFF										

*INTERRUPT LEVELS:
BIT 2 = GROUP 1, LEVEL 1
BIT 3 = GROUP 1, LEVEL 2

**MAGNETIC TAPE DENSITY:

BITS 8 9	DENSITY	BITS 14 15	CHARACTERS PER WORD
0 0	200 BPI	0 1	1
0 1	556 BPI	1 0	2
1 0	800 BPI	1 1	3
		0 0	4

***WHEN A ONE IS PRESENT IN BIT POSITION 4, ADVANCE TO THE FORMAT TAPE CHANNEL NUMBER (EXPRESSED IN OCTAL) REPRESENTED BY THE BITS PRESENT IN POSITIONS 7, 8 AND 9.

****TO SEEK TRACK 00 BOTH BITS MUST BE ZERO.

95117A-24

Figure 2-4. CEU Second Word Format

Standard Peripheral Devices. This word format may not apply to any Systems Engineering Laboratories Special Data Systems where use of the bit position designation is dependent on the system design.

TEST EXTERNAL UNIT

Code Selection

The following octal operation Codes are provided for using the TEU instruction as demands require: (Refer to figure 2-5).

1302 = TEU, Immediate Mode
1322 = TEU, Address Mode
1332 = TEU, Address, Map Mode

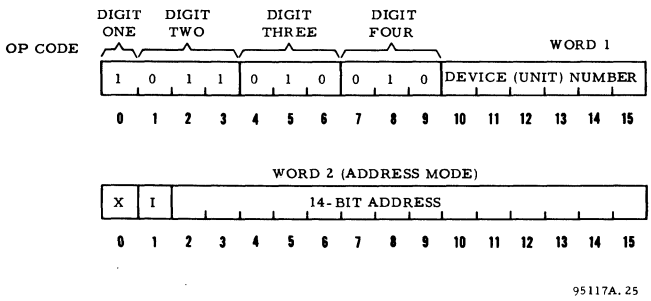


Figure 2-5. TEU Word Format

Description

Transfers the test code (up to 16 bits) contained in the specified memory location to the peripheral device specified in the device (unit) number. A return signal from the device is then tested, and the program register is advanced accordingly. If the return signal indicates a Ready or Go condition, the next instruction in sequence is skipped. A return signal indicating a Not Ready or abnormal condition causes the next instruction to be executed.

Operand Address Mode (Word 1, Bit 5)

I = 0; Second Word Immediate Mode

I = 1; Second Word Address Mode

NOTE

To reference a 15-bit address, M is required (bit 6 = 1) to append the most significant 15th bit to the 14-bit address.

Execution Modes

This instruction is always executed in the same mode. An on-line device is always Ready to accept a test dode. Therefore, the code is always transferred and the return is always tested. The Wait Flag (W) is not used.

Timing

The execution time required to complete a TEU instruction is four cycles.

TEU Second Word

The CEU second word format is shown in figure 2-6 which illustrates the 16-bit configuration applicable to the Systems Engineering Laboratories Standard Peripheral Devices. This word format may not apply to any Systems Engineering Laboratories Special Data Systems where usage of the bit position designation is dependent on the system design.

ACCUMULATOR WORD OUTPUT TO PERIPHERAL

Code Selection

The following octal operation codes are provided for using the AOP instruction as required: (Refer to figure 2-7)

1700 = AOP, Skip Mode
1701 = AOP, Wait Mode

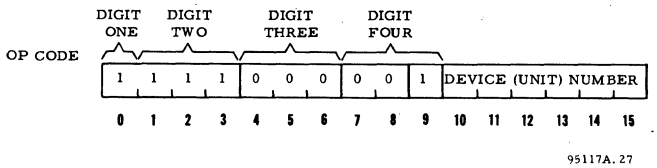


Figure 2-7. AOP Word Format

Description

Transfers a word from the A-Accumulator to device (unit) number. Character-oriented devices accept only bits A_0 - A_7.

Execution Modes (Word 1, Bit 9)

W = 0; Skip Mode
W = 1; Wait Mode

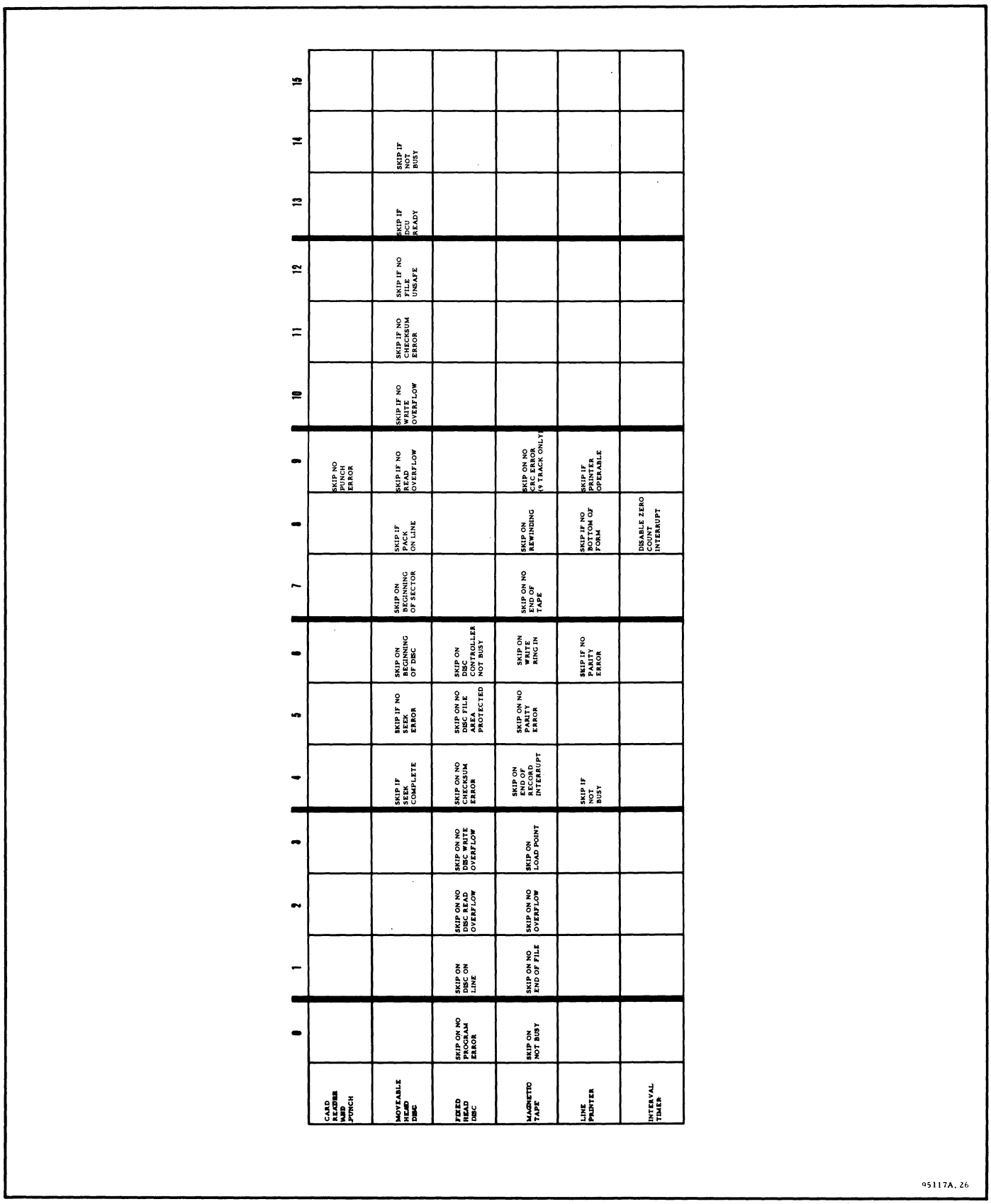


Figure 2-6. TEU Second Word Format

Transfer Criterion

A device (unit) answers Ready to an AOP test if the device can immediately receive a new word or character.

Timing

The execution time required to complete the AOP instruction is four cycles and wait.

ACCUMULATOR WORD INPUT FROM PERIPHERAL

Code Selection

The following octal operation codes are provided for using the AIP instruction as demands require: (Refer to figure 2-8).

1702 = AIP, Skip Mode
1703 = AIP, Wait Mode
1742 = AIP, Merge, Skip Mode
1743 = AIP, Merge, Wait Mode

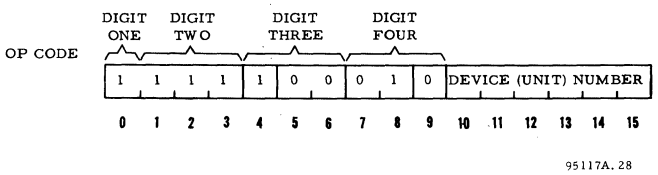


Figure 2-8. AIP Word Format

Description

Transfers a word or character from the device (unit) number into the A-Accumulator. Character-oriented devices transfer characters into bits A_8 - A_{15}.

Two methods of loading data into the accumulator are provided for the AIP, through use of the Merge bit (bit 4) in the instruction word. When bit 4 is a ZERO, the entire contents of the accumulator will be destroyed as the input word is loaded. When bit 4 is a ONE, the input word is added to the contents standing in the accumulator.

When the inputs are eight-bit words, the accumulator can be loaded with two words by programming a left shift eight position instruction before loading the second eight-bit word.

Execution Modes (Word 1, Bit 9)

W = 0; Skip Mode

W = 1; Wait Mode

Transfer Criterion

A device (unit) answers Ready to an AIP test if the device has a word or character ready for immediate transfer.

Timing

The execution time required to complete the AIP instruction is four cycles and wait.

MEMORY WORD OUTPUT TO PERIPHERAL

Code Selection

The following octal operation codes are provided for using the MOP instruction as demands require: (Refer to figure 2-9).

1704 = MOP, Immediate, Skip Mode
1705 = MOP, Immediate, Wait Mode
1724 = MOP, Address, Skip Mode
1725 = MOP, Address, Wait Mode
1734 = MOP, Address, Map, Skip Mode
1735 = MOP, Address, Map, Wait Mode

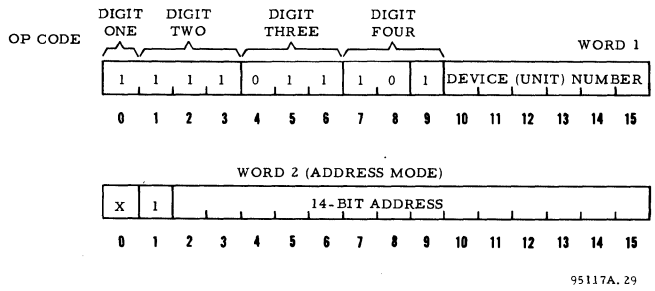


Figure 2-9. MOP Word Format

Description

Transfers a word from the specified memory location to the device (unit) number. Character-oriented devices accept only the seven most significant bits of the specified memory location.

Operand Address Modes (Word 1, Bit 5)

I = 0; Second Word Immediate Mode

I = 1; Second Word Address Mode

NOTE

To reference a 15-bit address, M is required (bit 6 = 1) to append the most significant 15th bit to the 14-bit address.

Execution Modes (Word 1, Bit 9)

W = 0; Skip Mode

W = 1; Wait Mode

Transfer Criterion

A device (unit) answers Ready to an MOP test if the device can immediately receive a new word or character.

Timing

The execution time required to complete a MOP instruction is four cycles and wait.

MEMORY WORD INPUT FROM PERIPHERAL

Code Selection

The following octal operation codes are provided for using the MIP instruction as demands require: (Refer to figure 2-10).

1706 = MIP, Immediate, Skip Mode
1707 = MIP, Immediate, Wait Mode
1726 = MIP, Address, Skip Mode
1727 = MIP, Address, Wait Mode
1736 = MIP, Address, Map, Skip Mode
1737 = MIP, Address, Map, Wait Mode

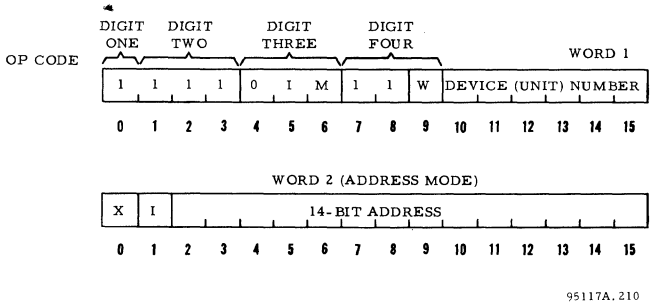


Figure 2-10. MIP Word Format

Description

Transfers a word or character from device (unit) number to the specified memory location. Character-oriented devices transfer characters into the seven least significant bits of the specified memory location. Bits 0 through 8 are reset to zero.

Operand Address Modes (Word 1, Bit 5)

I = 0; Second Word Immediate Mode
I = 1; Second Word Address Mode

NOTE

To reference a 15-bit address, M is required (bit 6 = 1) to append the most significant 15th bit to the 14-bit address.

Execution Modes (Word 1, Bit 9)

W = 0; Skip Mode
W = 1; Wait Mode

Transfer Criterion

A device (unit) answers Ready to an MIP test if the device has a word or character ready for immediate transfer.

Timing

The execution time required to complete a MIP instruction is four cycles and wait.

DATA TERMINAL

GENERAL

To ensure uniform timing in data communications between the computer mainframe and standard or non-standard peripheral units, Systems Engineering Laboratories has designed a standard data terminal. This data terminal provides a uniform interface between the mainframe and various types of input/output devices. The data terminal consists of two functional cards plus other assorted circuits. One functional card contains a patchable NOR matrix which is used to provide standard level output signals when the one (of 64) unit code for which it is wired is decoded. The second card contains all of the circuits necessary to answer and return standard timing signals that occur for the various input/output instructions. (Refer to figure 2-11).

I/O STANDARD TIMING SIGNALS

The I/O control signals associated with the data terminal provide a standard interface between the SEL 810B Computer input/output logic and peripheral devices. The I/O control signals lists in figure 2-12 are described in the following paragraphs.

Data Transfer Instruction (310)

This signal signifies to the data terminal that an AIP, AOP, MIP, or MOP instruction is being executed. The signal is present for the duration of the instruction.

Test Instruction (311)

This signal signifies that a test external unit (TEU) instruction is being executed and is present for the

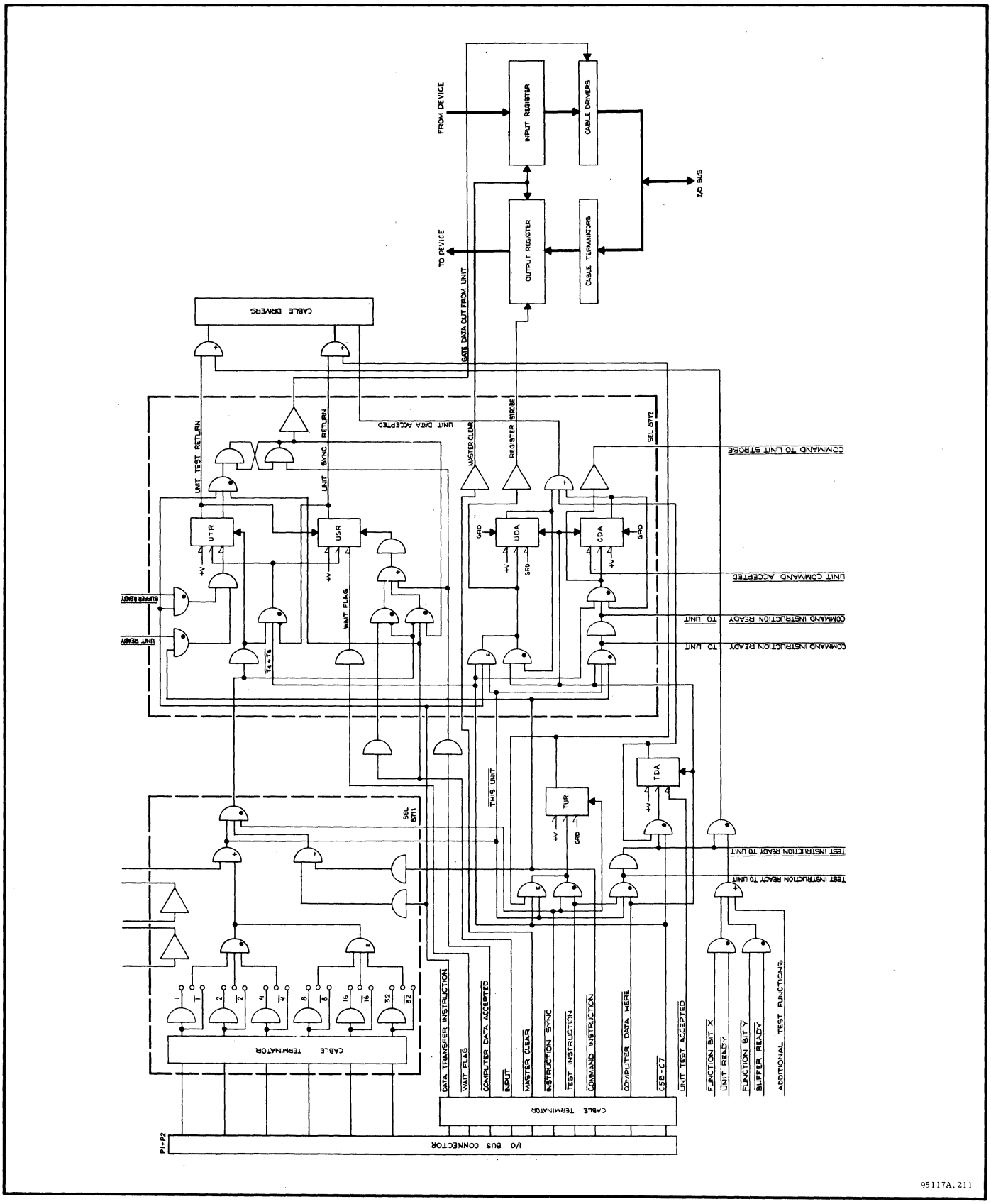


Figure 2-11. SEL Data Terminal Logic Diagram

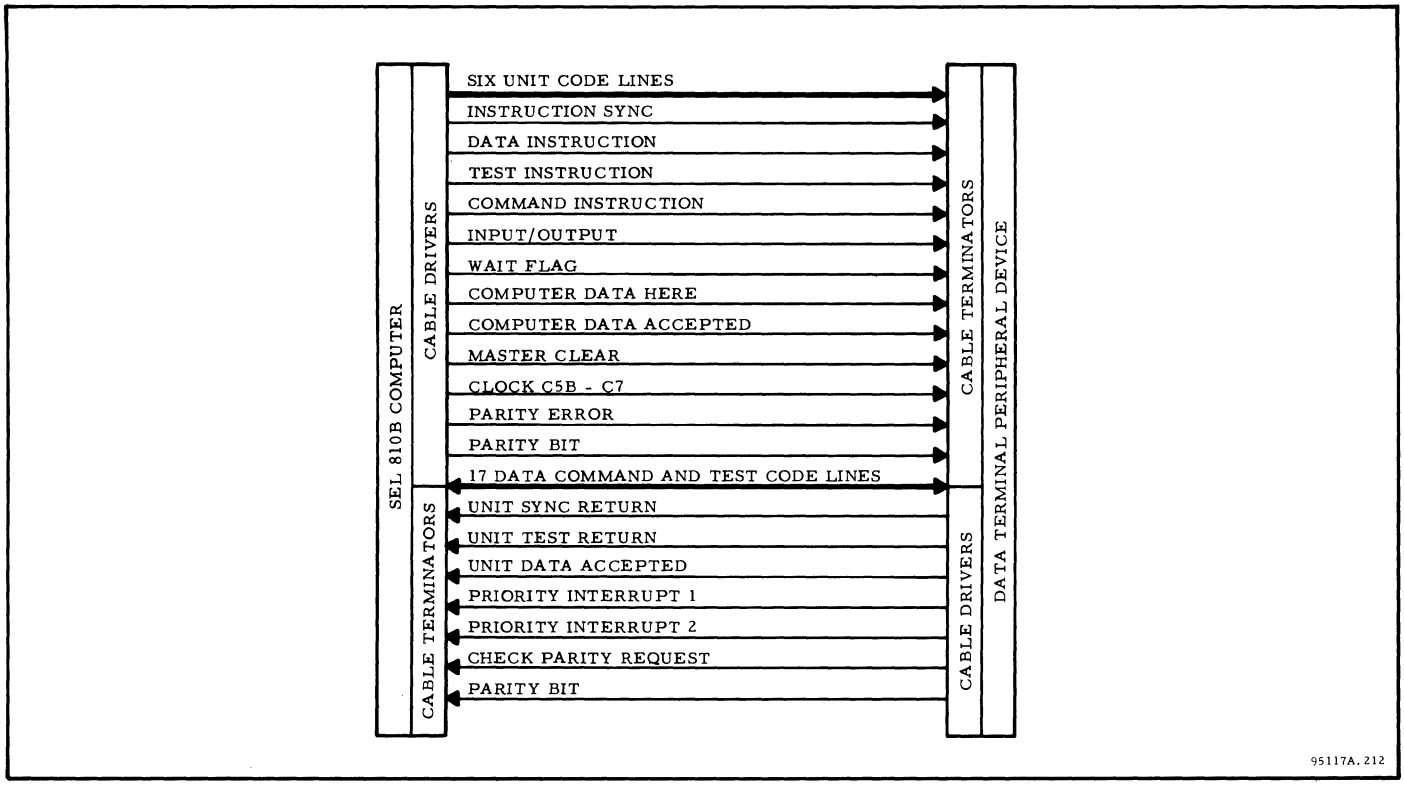


Figure 2-12. Standard Timing Signals

duration of the instruction. The second word of the instruction contains the test code.

Command Instruction (312)

This signal signifies that a Command External Unit (CEU) instruction is being executed and is present for the duration of the instruction. The second word of the instruction contains the function (operation) bits.

Instruction Sync (313)

The instruction sync is used to allow the device to interrogate its device (unit) number lines, and in turn answer via the unit sync return line. The instruction sync signal is present until the device answers.

Input/Output (509)

This signal signifies the direction of transfer in conjunction with a Data Transfer Instruction. The signal is present for the duration of the instruction. A one indicates input; a zero indicates output.

Wait Flag (510)

This signal signifies that the instruction being executed contains the computer wait flag bit. The wait flag causes the device to inhibit answering via the unit sync return until it is ready to perform the desired instruction. The wait flag is not used during a Test Instruction. The flag is present for the duration of the instruction.

Computer Data Here (512)

This signal signifies that the computer has recognized the unit sync return line and the test return - if applicable - and has proceeded into the execution cycle of the instruction. The Computer Data Here is used to notify the device that the data bits are present on the I/O bus and can be interrogated or loaded into the output register. The signal remains until the device answers on its Unit Data Accepted line.

Computer Data Accepted (513)

This signal signifies that the computer has accepted data from the unit as the result of an AIP, MIP instruction. The signal presence is controlled by the mainframe logic.

Unit Test Return (710)

This signal signifies the status of the device after receipt of the Instruction Sync from the computer. If the device is capable of performing the required

instruction, the Test Return line is enabled thus allowing the computer program a skip of the next instruction. The Test Return signal remains valid until the Instruction Sync signal is removed.

Unit Sync Return (711)

This signal signifies that the device has recognized the Instruction Sync signal. The signal remains until the computer removes the Instruction Sync signal.

Unit Data Accepted (712)

This signal signifies that the device has accepted the data word present on the I/O bus. In the case of the test instruction it notifies the computer that the test return line is valid and can be interrogated. The signal remains until the Computer Data Here signal is removed.

Master Clear (709)

The master clear line is activated by the Clear switch on the computer control panel and also by the ICB condition when power is turned on. In the case of the Clear switch, the line is a dc level and remains activated as long as the switch is depressed. The ICB condition enables the Master Clear line for approximately one second.

C5B - C7 (511)

The clock line is formed by generating a 125-nanosecond pulse from the master computer clock beginning at time C5B and extending through time C7.

Parity Error to Unit (505)

This signal signifies that the word transferred to the computer contained a parity error. This line is valid only if the equipment contains the optional I/O parity check capability.

Parity Request from Unit (506)

This line also operates in conjunction with the optional I/O parity check and is enabled during any input data transfer instruction. The line notifies the computer to perform a parity check on the word being transferred.

Parity Bit to Unit (507)

All data words presented to the I/O Bus include a parity bit to allow the device to perform a parity check if desired.

Parity Bit to Unit (508)

If the I/O parity option is included, the device must provide a parity bit for any input data transfer in order that the computer can perform a parity check on the word.

SIGNAL SEQUENCE

The Instruction Sync signal from the computer is accompanied by one of the following:

- Data Instruction for MOP, MIP, AOP, and AIP instructions
- Test instruction for the TEU instruction
- Command instruction for the CEU instruction

The input signal is high for the AIP or MIP instruction or low for the CEU, TEU, AOP, or MOP instruction. The six device (unit) number lines are always present to select a particular unit. The Computer Data Here signal is present for CEU, TEU, MOP, and AOP instructions. The Computer Data Accepted signal is generated near the conclusion of the MIP and AIP instructions. The C5B-C7 clock pulses are the only computer timing pulses used by the data terminal. The Parity Error signal is generated when an input parity error has occurred and the Check Parity Request signal came from the I/O device.

The Unit Sync Return is generated by the Instruction Sync signal from the computer immediately when no wait flag is present, and after either the device is ready or the buffer is ready depending whether this is a Command or Data Transfer Instruction if the wait flag is present in the original instruction. The Unit Test Return signal is generated by the Instruction Sync signal if the device is ready for a Command Instruction or the buffer is ready for a Data Transfer instruction. The Unit Data Accepted signal is generated as a result of an Output (from the computer) Data Transfer which can occur with CEU, MOP, or AOP instructions. The Priority Interrupt signals can occur if the standard priority interrupt are enabled and the interrupt condition is present.

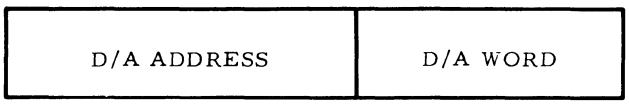
DATA TERMINAL SHARING

The Data Terminal is described in the previous sections normally operates with one peripheral device utilizing one unit number. In the case of special-purpose units it is possible to share the unit sync logic and utilize more than one unit number. The particular application usually dictates the most optimum logic configuration, or in some cases where it is required to have more than 64 units connected to the computer I/O structure, it

is necessary for many units to share one unit number. Also, each of the I/O cables are only capable of driving 16 loads where each cable terminator presents one load; a condition sometimes requiring sharing of a Data Terminal to conserve circuits.

Figure 2-13 is a block diagram of three units sharing one Data Terminal where it is used for command functions to the individual units. It can be seen that each of the various functions (command instruction ready to unit, command instruction strobe to unit, unit ready, command accepted, etc.) must be gated against their respective unit decode logic in order to ensure that only the addressed unit is commanded. This same approach can be extended as shown in figure 2-14. This block diagram shows an example where three output registers, each having separate unit numbers, are sharing a command Data Terminal. The MOP/AOP instructions are used to output data to the registers and the only additional gating required is to steer the register strobe to the proper unit. The unit ready is also gated with the device (unit) number to enable the skip feature of the instructions to be executed by the computer.

Various schemes can be devised to conserve unit numbers. The most applicable uses of Data Terminal sharing are in the case where a large number of input or output registers (displays, for example) must be connected to the computer I/O bus. If the registers do not utilize the full computer word, addressing can be done in the spare bits of the computer word such that only one unit number is used. For example, if 50, eight-bit digital-to-analog converters were to be connected to a Data Terminal, the individual converter address can be coded in the extra bits of the output word.



95117A.214

Due to the large number of possible uses, it is not possible to set down any hard or fast rules that apply to Data Terminal sharing. The examples and block diagrams presented are for guidelines and merely illustrate possible applications.

STANDARD PRIORITY INTERRUPTS

Each pair of standard priority interrupt circuits are formed on one functional circuit card. A logic diagram of this card is shown in figure 2-15.

The Computer Data Here signal, used to transfer the function bits of the second word of the CEU instruction, is also used to disable or enable either or both of these two standard priority interrupts. The presence of a 1 in function bit 1 of the second word of the CEU instruction causes either or both interrupts to be enabled. The presence of a 0 in function bit 1 of the second word causes either or both standard interrupts to be disabled. The selection of the interrupt to be enabled or disabled is determined by the presence of 1's in function bits 2 and 3. If function bit 2 is a 1, the input interrupt will be either enabled or disabled depending on the state of function bit 1. If function bit 3 is a 1, the output interrupt will be enabled or disabled depending on the state of function bit 1.

The enable/disable logic utilizes a function of circuit design of the micrologic flip-flop. If both the J and K inputs to the flip-flop are at + v, any strobe trigger will be ignored. If, however, + v exists on the set input and 0 v on the reset steering input, the flip-flop will be set by any trigger strobe that is applied.

When either interrupt channels is enabled, the NOR latch connected to the reset steering input goes to 0 v so that the flip-flop may be set. When the channel is reset, this same NOR latch is reset so that + v is applied to the reset steering input thus making the flip-flops impervious to any trigger strobe. The Computer Data Here signal produces a Command To Unit signal which is applied to four AND circuits so that either the input or output interrupt or both may be either enabled or disabled by this command. Note that it is not possible to enable one interrupt while disabling the other; however, the arrangement does allow these interrupts to be selectively enabled or disabled by the same command that causes the device to perform some operation.

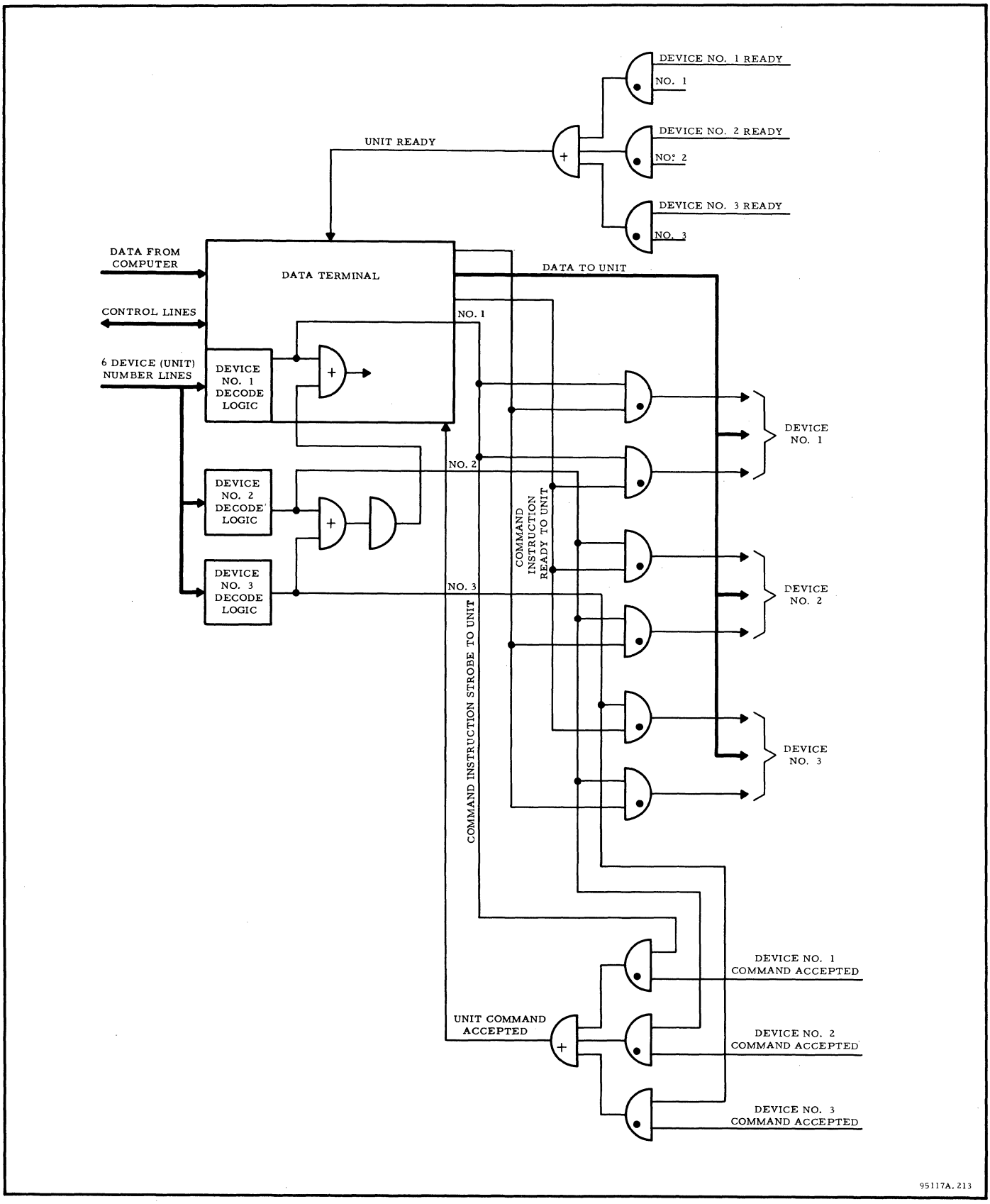


Figure 2-13. Shared Data Terminal Command Sharing

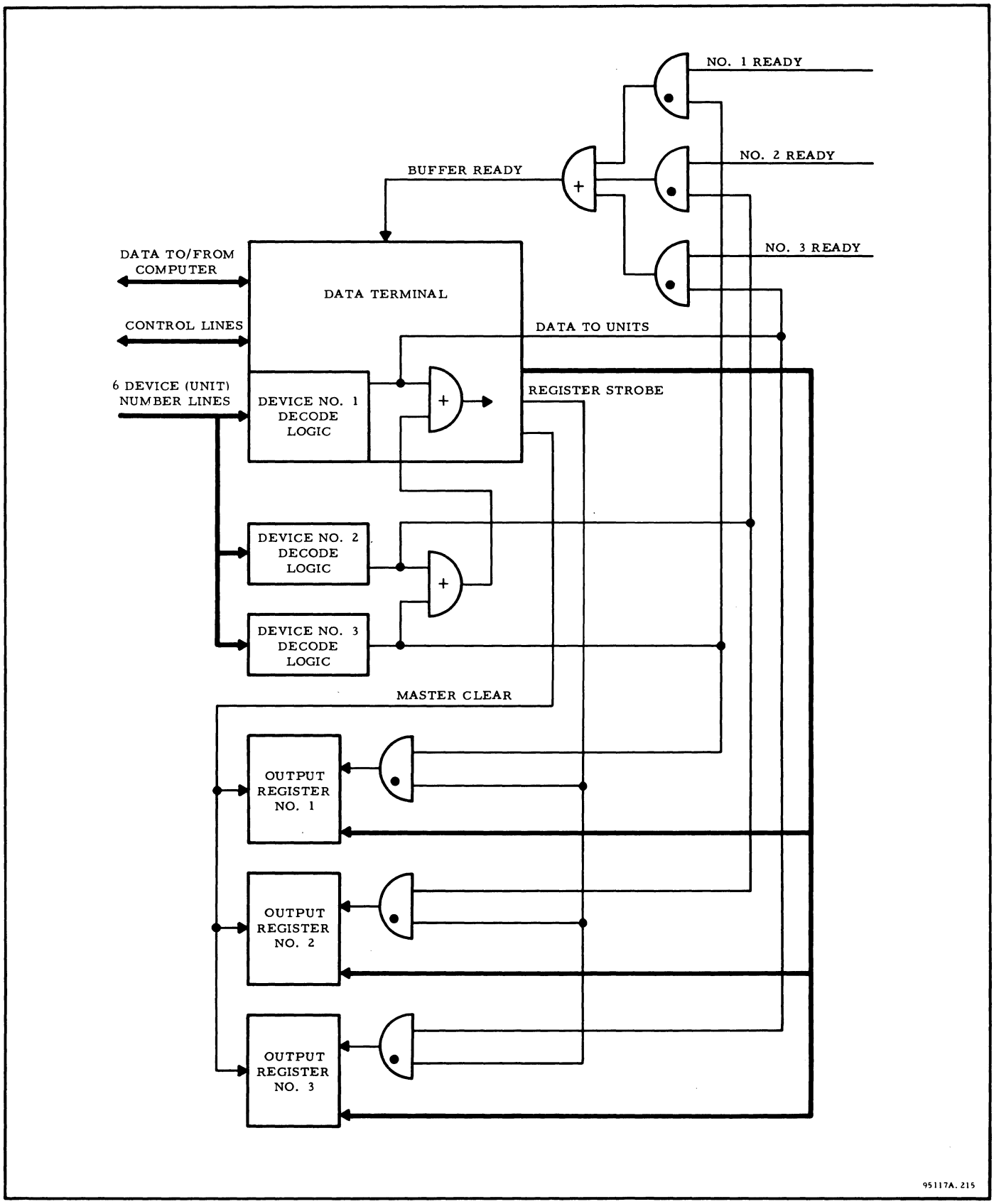


Figure 2-14. Shared Data Terminal - Register Sharing

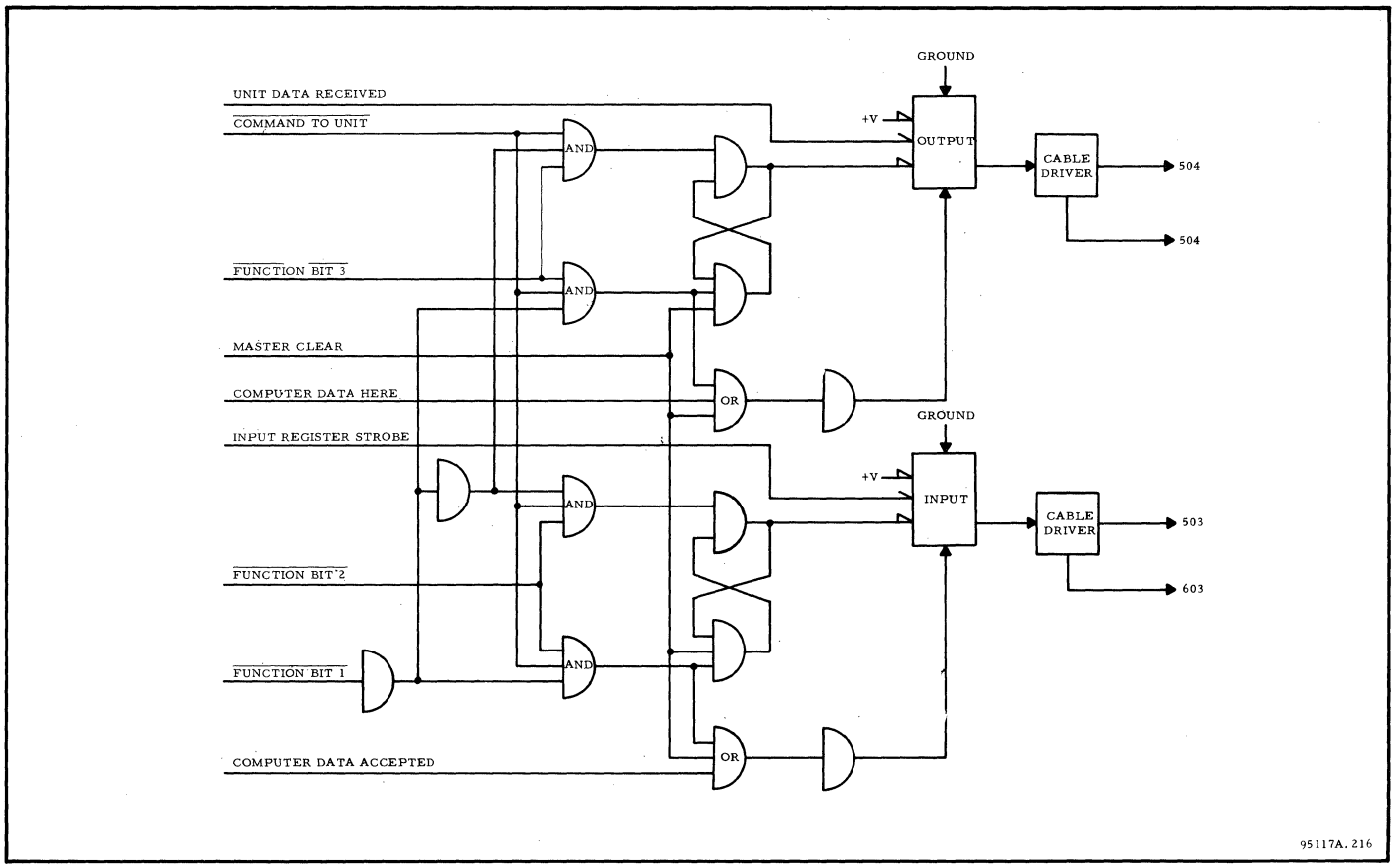


Figure 2-15. Standard I/O Interrupt Logic Diagram

SECTION III
INPUT/OUTPUT THEORY

GENERAL

The SEL 810B Computer Input/Output Processor provides a positive synchronization control for data flow between the computer and peripheral devices. It can synchronize data transfer between a peripheral device and either Memory or the A-Accumulator. The data path for each word or character transferred is controlled by the program which can execute any of the input/output instructions.

The I/O Bus connects all peripheral units to the I/O Processor in a daisy-chained manner, as shown in figure 3-1. The I/O Bus contains 16 data lines, (plus parity option), six device lines, and numerous control lines.

The 16 data lines provide two-way communication paths. All data, CEU command words, and TEU test words are transferred over these lines. Word-oriented units such as acquisition subsystems contain a full set of 16 cable drivers and terminators for the data lines. Character-oriented devices having character assembly buffers such as magnetic tape control units also contain a full set of cable drivers and terminators. Character-oriented devices having character buffers such as paper tape punches and readers contain only eight to 10 cable drivers and/or terminators. In this case, data commands and test codes are always received from the computer on the eight lines corresponding to computer bit positions 0-7. Some devices also receive commands from bits eight to 15. Single characters are always transferred to the computer on the data lines corresponding to bit positions 8-15. Characters having less than eight bits are right-justified in the eight-bit field. The data lines connected to each peripheral unit are defined in Section II.

The six device number lines connected to each device permit up to 64 individual devices to be addressed by the computer.

The control signal lines listed in table 3-1 are used to enable I/O instructions to be executed in the following basic sequence (TEU differs):

a. The computer initiates execution by sending out the device (unit) number contained in the instruction. The computer also sends out the Instruction Sync and Instruction Command (Data, Command, Test, Input/Output) signals.

b. The addressed device responds by sending the Unit Sync Return and Unit Test Return signals to the computer.

c. After recognizing the Unit Sync Return signal, the computer tests the Unit Test Return signal for the device status (Ready to execute command or Not Ready).

d. If the device indicates Ready, the data transfer is made. The Data Here and Data Accepted signals synchronize the transfer. For computer input transfer, the device Ready signal also indicates Data Here.

e. After the transfer is completed, the computer tests the control lines from the device to ensure that they have returned to the Off level. The next instruction is started in the following machine cycle.

The normal executed time for each immediate mode I/O instruction is four machine cycles. In addition, presence of the Wait Flag in an I/O instruction delays completion of instruction execution until the device indicates Ready. (The operation of the Wait Flag is described in Section II.

The execution sequence is similar for all instructions except TEU. When a TEU instruction is executed, no Ready test is made before transfer of the test word. Transfer is made following recognition of the Unit Sync Return signal. The Test Return line is tested after the test word has been transferred to the device. The return signal is a particular device status gated on the Test Return line by the value of the test word transferred to the device.

I/O INSTRUCTION WORD FLOW

Typical connections used to perform input and output data word transfers are shown in figure 3-2. Each connecting cable is also linked to a plug to which another device may be connected in parallel with the first. This daisy-chaining of devices (units) permits all devices to appear identical from the mainframe.

The CEU, TEU, and MOP instructions call the second word containing the command, test, and data bits from memory. The second word is

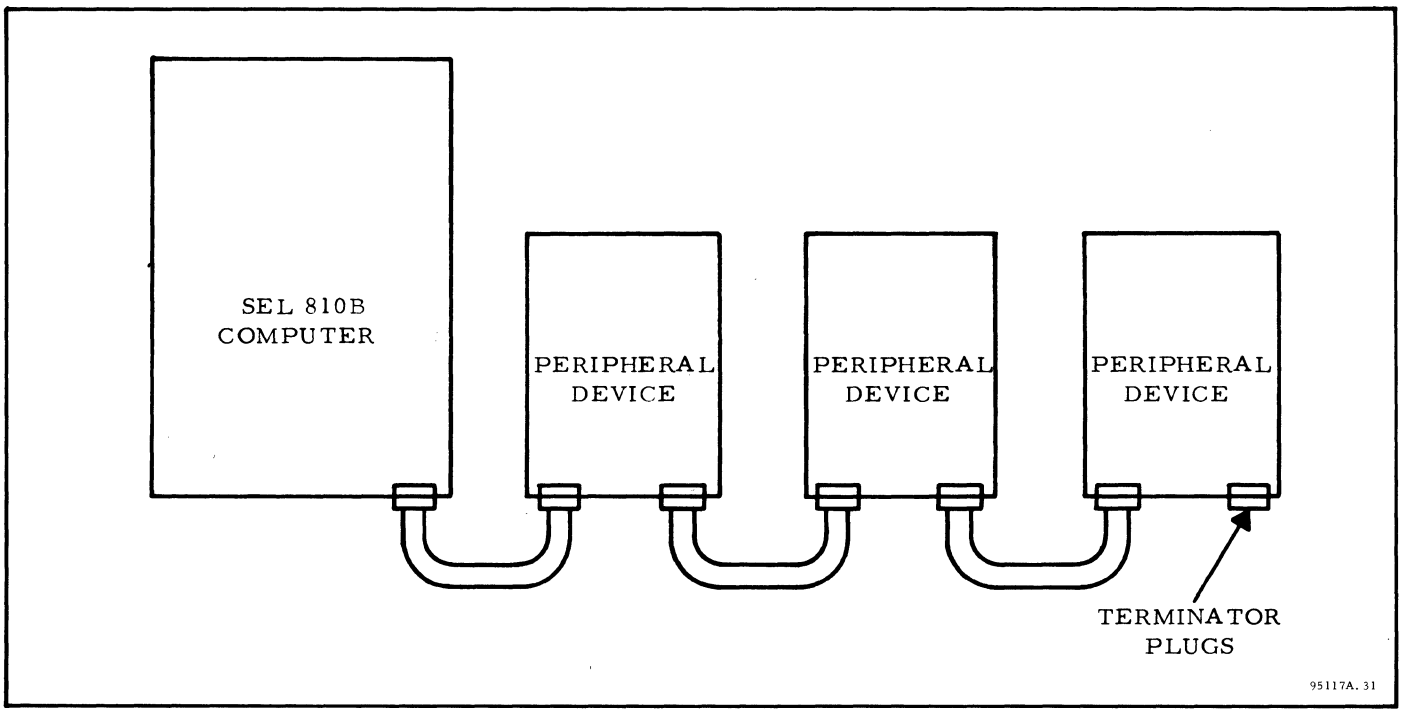


Figure 3-1. Peripheral Device Bus Connections

Table 3-1. I/O Control Signals

Signal	Computer Commands
Instruction Sync (from Computer)	AIP, AOP, MIP, MOP, CEU, TEU
Data Instruction (from Computer)	AIP, AOP, MIP, MOP
Command Instruction (from Computer)	CEU
Test Instruction (from Computer)	TEU
Input/Output (from Computer)	AIP, AOP, MIP, MOP
Wait Flag (from Computer)	AIP, AOP, MIP, MOP, CEU
Unit Test Return (from Device)	AIP, AOP, MIP, MOP, CEU, TEU
Unit Sync Return (from Device)	AIP, AOP, MIP, MOP, CEU, TEU
Computer Data Here (from Computer)	AOP, MOP, CEU, TEU

Table 3-1. I/O Control Signals (Cont'd)

Signal	Computer Commands
Computer Data Accepted (from Computer)	AIP, MIP
Unit Data Accepted (from Device)	AOP, MOP, CEU, TEU
Computer Clock C5B-C7 (from Computer)	All I/O Instructions
Master Clear (from Computer)	All I/O Instructions

read from memory and gated to the I/O cable through cable driver circuits.

The control signals necessary to transfer the memory word and to select the proper device are supplied from the computer I/O control circuits. The memory word and control signals are supplied to all devices in parallel. Only the one unit device selected by the unique device (unit) number will be able to accept the memory word and use the supplied control signals.

The function and test bits of the CEU and TEU second words are gated by control signals to set control latches or to jointly test for various unit

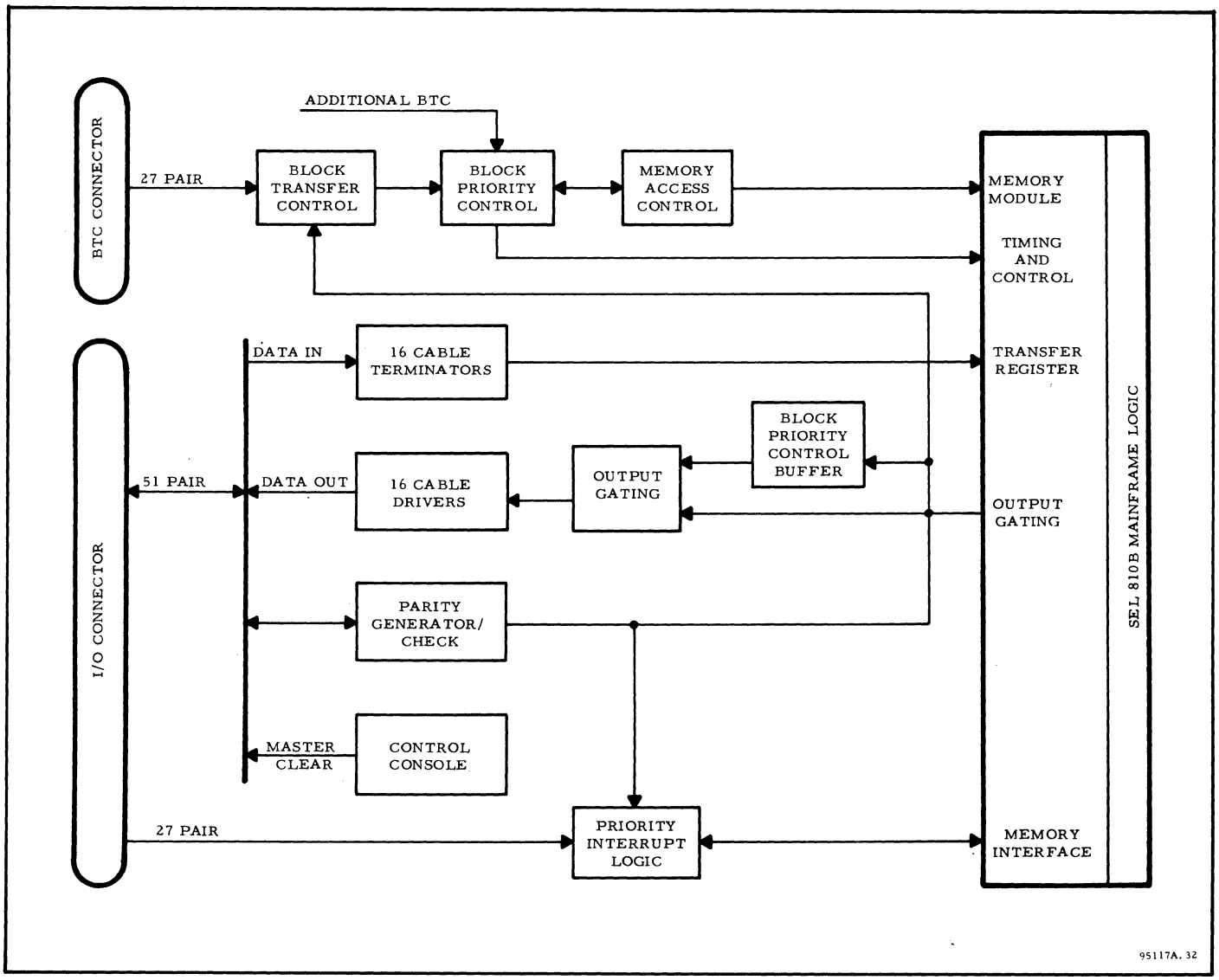


Figure 3-2. Computer I/O Interface Diagram

conditions. The data word supplied by the MOP instruction is to be loaded into an output data register. The Data Terminal returns signals to the mainframe upon acceptance of the computer, test or data bits. The TEU instruction also causes a high or low level to be generated on a command sense return line to indicate the status of the condition tested.

The AOP instruction causes the data word in the A-Accumulator to be transferred to an output data register in the selected device. The path data is from the A-Accumulator through the adder, the memory output OR gates, and the I/O cable drivers to an output data register.

The MIP instruction gates the input word from the device to the memory data register. The data patch followed by the input word is from the device input data register through the device cable drivers, the data in cable terminators, the memory input AND gates, and the memory input OR gates to the data register of the selected memory module.

The AIP instruction which inputs a data word into the A-Accumulator, follows the same path up to the memory input OR gates. Instead of being gated to a memory module, the data word is applied through the memory output OR gates. The word is then gated through the intrinsic AND circuits into the T-Register. Later in the execution cycle, the word is transferred from the T-Register through the adder to the A-Accumulator.

The two standard I/O priority interrupts provide specific memory addresses to the mainframe when

they occur. These addresses are forced through hardware circuits onto the memory output bus and then to the T-Register. At the same time, a hardware Store Place and Branch (SPB) Indirect instruction code is forced onto the memory output bus and then to the Instruction Register. Each memory location addressed by the interrupt is reserved for the starting address of the priority interrupt servicing program for that specific interrupt.

The block transfer control utilizes specific memory locations to hold the block count and starting memory address. The data patch for these memory words is from the Data Register in the selected memory module, through the memory output OR gates, the block transfer control buffers into the block transfer control. The return path is for the current memory address which can be stored in memory, if requested by the program in a CEU instruction.

COMPUTER TIMING

The computer accesses and executed instructions during specified intervals determined by the timing generator in the computer. The intervals are machine cycles named the instruction (I) cycle and the execution (E) cycles. The E cycles are subscripted one and two to differentiate between them.

The I, E_1, and E_2 cycles are produced by a group of mainframe circuits termed the control cycle generator. The generator advances from I cycle to E_1 cycle and back to I cycle for direct memory reference instruction. It advances into the E_2 cycle for the increment memory and skip (IMS) and compare memory and the A-Accumulator (CMA) instruction.

Each I and E cycle is composed of 15 shorter time pulse units designated T times, for example, T1, T2, T3, etc. These pulses are gated to produce supplementary timing pulses to perform the various functions required to decode and execute an instruction. The T time pulses have a time duration of 50 nanoseconds and are divided in two, half-time periods, for example, T1A, T1B, T2A, T2B, etc., each having a time period of 25 nanoseconds. (Refer to figure 3-3.) The C5B-C7 time pulse signal used for peripheral device interface timing has a time duration of 125 nanoseconds.

Every instruction is obtained from memory and decoded during an I cycle. The I cycle will be followed by one or more E cycles or an I cycle, depending on the timing required by the particular instruction decoded. At the completion of the last cycle required by an instruction, the I cycle for the next instruction is generated, thus an I cycle may be followed by an E_1 cycle, an E_1 and E_2 cycle, or an I cycle.

The sequence of the timing signals for all instructions revolves around the rear or write memory sequences. The T1 timing pulse occurs when a memory word has been read from the memory and is available for use by the control and arithmetic units. The computer word is then generally held in active registers (that is, instruction, transfer, and accumulator registers) until the next T1 time when it may be replaced by the next memory word.

The memory requires approximately 350 nanoseconds to obtain a word from the addressed magnetic cores. In order for this word to be available by T1 time, the memory read command is given at T7 time of the preceeding cycle. The memory output word is then present on the memory output bus from the beginning of T5 through the end of T7. During this period, the memory word is applied to the steering inputs of the Instruction Register and T-Register flip-flops and to the cable driver circuits, source of the I/O output bus. The memory word is also held in the Memory Data Register until replaced by the next word from memory, at the next T5 time.

During I cycles, the memory word is treated as an instruction word and is loaded into both the

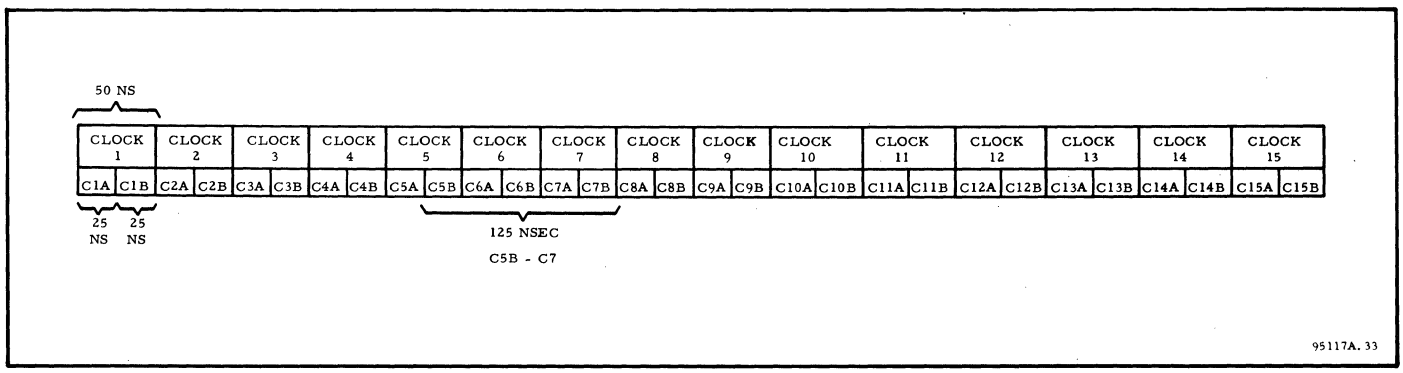


Figure 3-3. Clock Time Pulses

Instruction and T-Registers at T1 time. By T3 time, the operation code has been decoded. If it is a memory reference instruction, the T-Register supplies the nine least significant bits of the operand address. If the instruction is not a memory reference instruction, the contents of the T-Register supplies the nine least significant bits of the operand address. If the instruction is not a memory reference instruction, the contents of the T-Register are ignored. The T-Register is used for data transfers during the execution of many augmented instructions.

A 1 in the indirect flag (bit 5) position of a memory reference instruction word does not affect the base or indexed address in any manner, but does force a memory read command to the memory. At the same time, the address output of the adder is gated to the Memory Address Register and the timing control is forced to remain in the I cycle (termed the Indirect I Cycle). The word subsequently read from memory and put in the T-Register by the next T1 timing signal is treated as a 14-bit address with the address bits located in the 14 low-order positions of the register. The two high-order bits of this word are loaded into bit positions four and five of the Instruction Register replacing the previous contents of those positions. The two bits are the Index and Indirect flags respectively.

The address bits in the T-Register are gated to the 14 low-order positions of the adder. If the MAP flag is a 1, the one most significant bit (bit 1) of the Program Register is gated to the bit 1 position of the adder. If the MAP flag is a 0, a 0 is gated to the bit 1 position of the adder. At the same time, the new index flag is tested and, if a 1, the contents of the B-Accumulator are summed with the base address. If the new indirect flag is a 1, the address is gated to the Memory Address Register, a memory read command is generated, and the computer is again held in the I cycle.

If the new or the original Indirect flag is a 0, the computer timing is allowed to advance to the E_1 cycle. The address output of the adder, either direct or indirect, is gated to the Memory Address Register. Either a memory read or a memory write signal may be generated depending on the operation code contained in the current instruction.

For those instructions which are to read a word from memory, the word will be loaded into the T-Register at T1 time. From the T-Register, the word may then be gated to an arithmetic register, or to the Program Register depending on the operation code in the Instruction Register. When words are to be written into the memory, they may come from an Accumulator Register, the program register, or an input/output device. Regardless of the source, the word must be gated to the Memory Data Register during the period T13 through T6. This timing is necessary due to the memory timing that starts the write operation at T14 time. The period from T7, when the write command was given, until T13 time is used to erase the previous word from the addressed memory location.

COMPUTER I/O TIMING

The first word (and only word for AIP and AOP instructions) of the I/O instructions is loaded into the Instruction Register during the I cycle. This word remains in the instruction register for the duration of the instruction execution period. Bit positions 10 through 15 of the instruction register hold the device (unit) number bits during the entire instruction period. The outputs of the six device (unit) number flip-flops are connected through cable drivers to the computer I/O Bus. The device specified by the unique unit number is the only device connected to the data and command buses during the current instruction.

Most augmented instructions other than I/O instructions require only the I cycle for unloading the instruction and executing it. In the case of shift instructions, where four lateral shifts of the accumulator are performed during each cycle, the control cycle generator is held in the current I cycle until the specified number of shifts are completed.

During I/O instructions, the control cycle generator is held in the I cycle for at least one additional memory cycle time. This is to allow the mainframe to test the device to determine that the unit is ready to receive a command of data. This one-cycle delay is essentially a propagation delay so that time is allowed for the device to respond to the test. If the device is located a long distance (that is, several hundred feet or more) away from the mainframe the propagation delay of the query and response signals may be significant.

The return signal consists of two parts: (1) a return synchronizing pulse that causes the control cycle generator to advance to the E_1 cycle; (2) a return test signal that is either high to indicate the device is ready, or low to indicate that the device is not ready. If the instruction is programmed for the Skip mode, a high test return signal causes the completion of the instruction during the E_1 cycle and advances the program register twice to skip the next sequential instruction. If the test return is low, the E_1 cycle occurs, but no signals are generated to execute the instruction; and the program register is advanced once to execute the next sequential instruction.

In the Wait mode of operation, the second I cycle is repeated indefinitely until the selected device is ready. Both the return synchronizing pulse and the high test return are generated when the ready condition occurs. The control cycle generator then advances the E_1 cycle, the instruction execution signals are generated, and the program register is advanced once to call the next sequential instruction.

The presence of a 1 in the bit 5 (Indirect flag) position of the Instruction Register forces the control cycle generator into an indirect I cycle. During this cycle, (the second I cycle specified in the preceding test) another memory read is accomplished. This second word (stored in indirect address format) is taken from the memory location following the first word of the instruction. The 14-bit address portion of the word is then loaded into the 14 least significant bit positions of the T-Register. The Indirect and Index flags possible contained in bit positions 0 and 1 of the address word are loaded into bit positions 4 and 5 of the Instruction Register.

The Indirect flag contained in the address word replaces the Indirect flag contained in the first word and is stored in the Instruction Register. If the new flag is a 0, the computer now waits until the return synchronizing signal is received from the device. If the new flag is a 1, another Indirect I cycle is accomplished and the Indirect flag in that address word replaces the previous flag in the Instruction Register. The Indirect I cycles are repeated until a 0 is loaded into bit 5 of the Instruction Register with each direct address specifying the memory location of the next address. If the return synchronizing signal occurs during one of these Indirect I cycles, it is stored until a 0 finally appears in bit 5 of the Instruction Register.

When the final indirect address has been called into the T-Register and the return synchronizing signal is received, the address held in the T-Register specifies the source or destination memory cell for the data, command, or test word used in the E_1 cycle.

For command, test, and output data tranfer instructions, this address specifies the source. In the Immediate Address mode, the source is always the second word of the instruction or the A-Accumulator. The execution of the CEU, TEU, and MOP instruction requires that the command, test, or data word be read from memory and be placed on the I/O bus. This is accomplished by reading the memory output word into the Memory Data Register during the E_1 cycle (when the unit is ready) and gating the outputs of this register to the I/O bus. During an AOP instruction, the outputs of the A-Accumulator are gated through the adder and memory output OR gates to the I/O bus.

The input data transfer instructions, MIP and AIP, require a slightly different timing arrangement during the E_1 cycle. The MIP instructions requires that the data word supplied by the unit be present at the inputs of the Memory Data Register during the T14 through T1 times. This is because the memory write cycle clears the addressed cores and the data register to 0's during the first half of the memory cycle from T7 through T13 times. The memory timing circuits then generate a Load Data Register signal at T4 time to load the new word into the Data Register. This word is then subsequently copied into the addressed magnetic cores.

The AIP instruction requires that the input word be on the memory output bus by T1 time when it is loaded into the T-Register. The contents of the T-Register are then gated through the adder to the inputs of the A-Accumulator. The word is finally loaded into the A-Accumulator at T7B time.

I/O INSTRUCTION TIMING AND DESCRIPTION

GENERAL THEORY

COMMAND EXTERNAL UNIT (CEU)

Function

The Command External Unit Instruction is used to prepare an external device for some subsequent operation. The instruction may:

a. Set the mode of operation of the device.

b. Set the direction of operation of the device.

c. Select a sub-unit of the addressed device for operation.

d. Enable or disable the standard input and/or output priority interrupts.

e. Initialize a block transfer operation.

f. Extract the current memory address of a block transfer operation.

g. Set the number of characters per word for a character assembly buffer.

The CEU instruction is a two-word instruction which is stored in two sequential memory locations. The instruction may operate in either the Wait or Skip modes of operation and in either Immediate

or Address modes. The first word of this instruction sets the modes of operation by flags contained in bit positions five and nine.

Bit nine contains the Wait Flag, which if set to a 1, causes the device's Data Terminal logic; not to answer Unit Sync Return until the device is Ready, therefore causing the computer to wait. The presence of a 0 in bit 9 of the first word will cause the computer to operate in a Skip mode. The Skip mode causes the computer to access the next sequential instruction following the CEU instruction if the device is not ready, but to skip the next sequential instruction and obtain the subsequent instruction if the device is ready. When the computer is in a wait cycle, it cannot be interrupted by any priority interrupt except for the optional power fail safe/restore or stall alarm interrupts. If however, the Skip mode is used, the computer may be programmed through use of a BRU instruction to cycle until the device is ready before proceeding. This allows the servicing of priority interrupts that might occur between the CEU instruction and the subsequent BRU instruction.

The Indirect flag, located in bit five position of the first word of the CEU instruction, defines whether the instruction will operate in the Immediate mode or the Address mode. If the indirect flag is a 0 (Immediate mode), the second word of the Instruction becomes the operand. The address mode, indicated by a 1 in bit position five of the first word of the instruction, utilizes the indirect address word format. The second word of the instruction becomes an indirect address of 14 bits plus an Index and Indirect flag. The Indirect flag (bit 1) may be used for indirect chaining, and the index flag (bit 0) for an index operation. If the MAP bit (bit 6) of the first word of the instruction is a 1, the most significant bit of the Program Register is appended to the 14 bit address. The inclusion of the MAP bit modifier allows the operand to be called anywhere in the entire 32 K of memory. If the MAP bit contains a 0, the 15th or most significant bit of the address will always be zero. The final word obtained from memory at the end of an indirect chain is identical in format to the second word called from memory in the Immediate mode.

This second word may contain up to 16 function bits. These function bits specify:

a. In bit 0, BTC initialize;

b. Bit 1, priority interrupt enable/disable;

c. In bits 2 and 3, the two standard I/O interrupts to be either enabled or disabled depending on the state of bit 1;

d. In bits 4 through 9, function code bits which specify the operation, mode, etc., to which the device is to respond;

e. In bits 10 through 12, the tape device (unit) number if the addressed device is a tape control device;

f. In bit 13, a command to load the current address of a block transfer control unit a specified memory location;

g. In bits 14 and 15 the number of characters into which a computer word is to be disassembled for output transfer if the addressed device contains a character assembly buffer.

Peripheral - Computer Timing Relationships

The basic computer timing, described previously consists of a continuous series of 750-nanosecond cycles. This cycle time is the time required for the memory to perform a complete read/write cycle. The cycle during which the instruction is unloaded from memory is referred to as the I (instruction) cycle; the memory cycles during which data is unloaded from or loaded into memory are referred to as E (execution) cycles.

Each CEU instruction consists of at least two I cycles and one E cycle. The first I cycle is the cycle during which the first word of the instruction is unloaded from memory. The second I cycle is the time during which the device is tested to see whether it is ready to accept a command from the computer. The second I cycle may be repeated indefinitely if the instruction is used in the Wait mode. Additional I cycles are also required if the instruction is used in the Address mode requiring more than one indirect cycles. The first indirect cycle is performed while the device is being tested. When the second word containing the data (function code bits, etc.) is unloaded from memory at the end of an indirect chain or immediately if the Immediate mode is used, and the device has responded that it is ready for a transfer of instructions from the computer, then the computer is allowed to go into an execution cycle referred to as the E_1 cycle.

Execution Sequence

The basic instruction, used in the Skip and Immediate Address modes, requires four complete memory cycle times. Refer to the timing diagram, figure 3-4, flow chart, figure 3-5, and terminal signal flow, figure 3-6. Note that during the first I cycle the command instruction is shown as being available from the beginning of T3 time of the first I cycle continuously until the end of

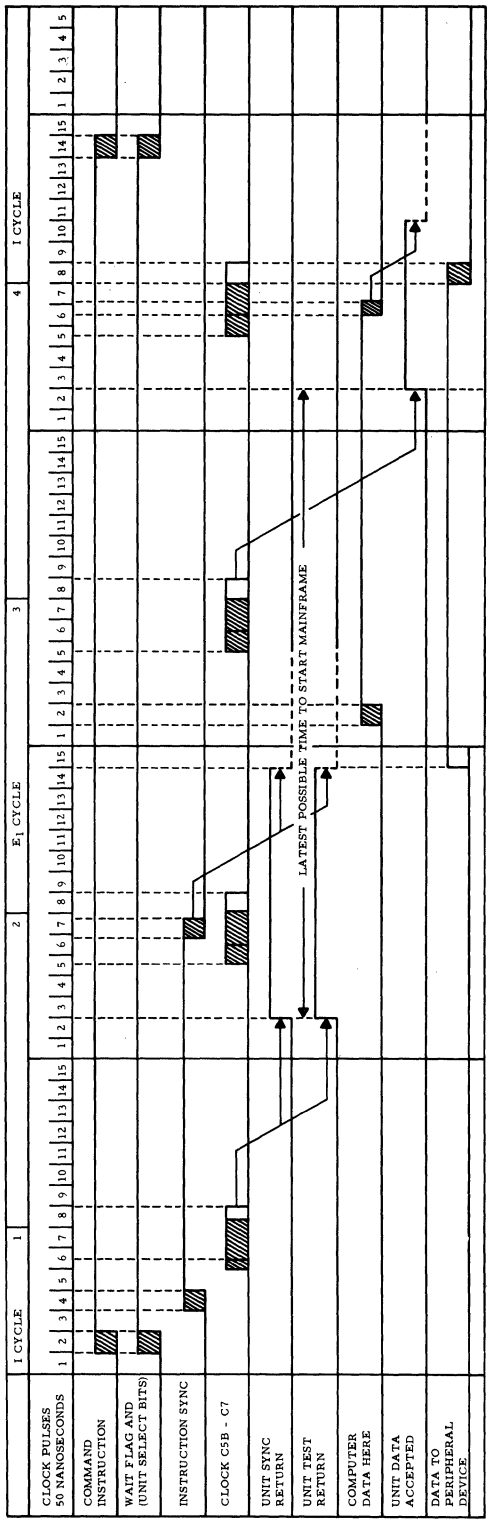


Figure 3-4. CEU Instruction Timing Diagram (At Computer Mainframe)

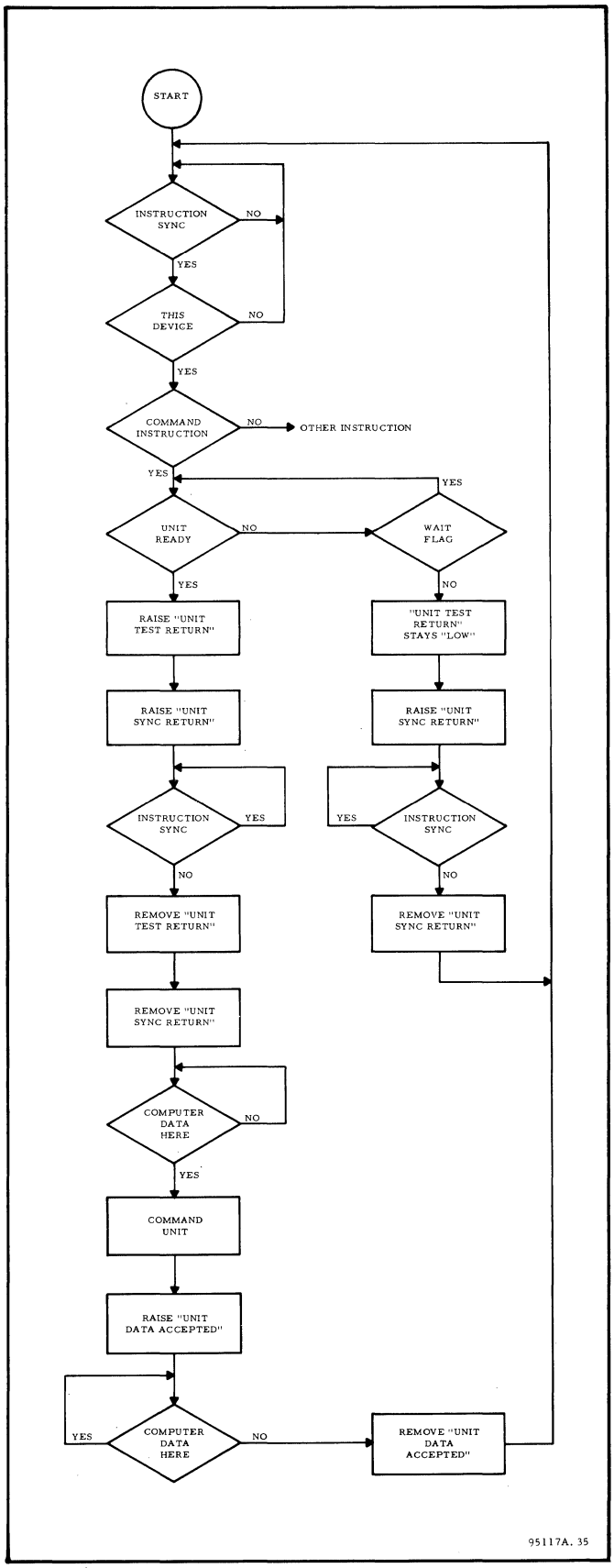


Figure 3-5. CEU Flow Chart

T13 time of the I cycle of the next subsequent instruction. During this entire period the CEU instruction remains in the Instruction Register in the mainframe and thus provides CEU enables to both the mainframe circuits and to the selected device. The Instruction Register also holds the wait flag and the device (unit) number for this period.

At T5 time of the first I cycle, the computer circuits produce an Instruction Sync level which is gated to the selected device circuits by the device (unit) number. If the device is ready, the Instruction Sync signal gates the clock pulse to set a latch in the peripheral device which provides simultaneous Unit Test Return and Unit Sync Return signals back to the mainframe. The Instruction Sync signal which initiated this action in the device also set a latch in the mainframe which holds the mainframe to the I cycle. When the Unit Sync Return signal comes back to the mainframe it resets this latch and allows the computer to finish the I cycle and advance into the E_1 cycle during which the second word is unloaded from memory. The Unit Test Return signal is used to perform the skip test. If the Unit Test Return signal is received, the program register will be advanced twice so as to skip the next subsequent instruction. If the Unit Test Return signal does not come back, no Computer Data Here signal will be generated, and the Program Register will be advanced once and will call the next sequential instruction.

The Unit Test Return signal also provides a Computer Data Here signal from the computer to the device coincident with the time that the second word of the instruction is available on the I/O bus. The Computer Data Here signal sets a latch in the device which provides a signal back to the computer that states Unit Data Accepted. The philosophical point is that if the device was ready to return the Unit Sync Return and Unit Test Return signals, then it obviously will be ready to accept the data from the second word when strobed by the Computer Data Here signal.

The entire sequence of the CEU instruction, as it pertains to the device, is shown in the CEU flow diagram. The first signal produced by the mainframe to the device is the Instruction Sync level. When that signal occurs, and the device is selected by the device (unit) number, it will produce the Unit Test Return level at the next clock pulse, if the device is ready, which in turn produces the Unit Sync Return level. Note that if the device is not ready when the Instruction Sync level is generated by the mainframe, a test is made to determine if the wait flag is present in the original instruction. If the wait flag is present, the computer will continually test for the device ready

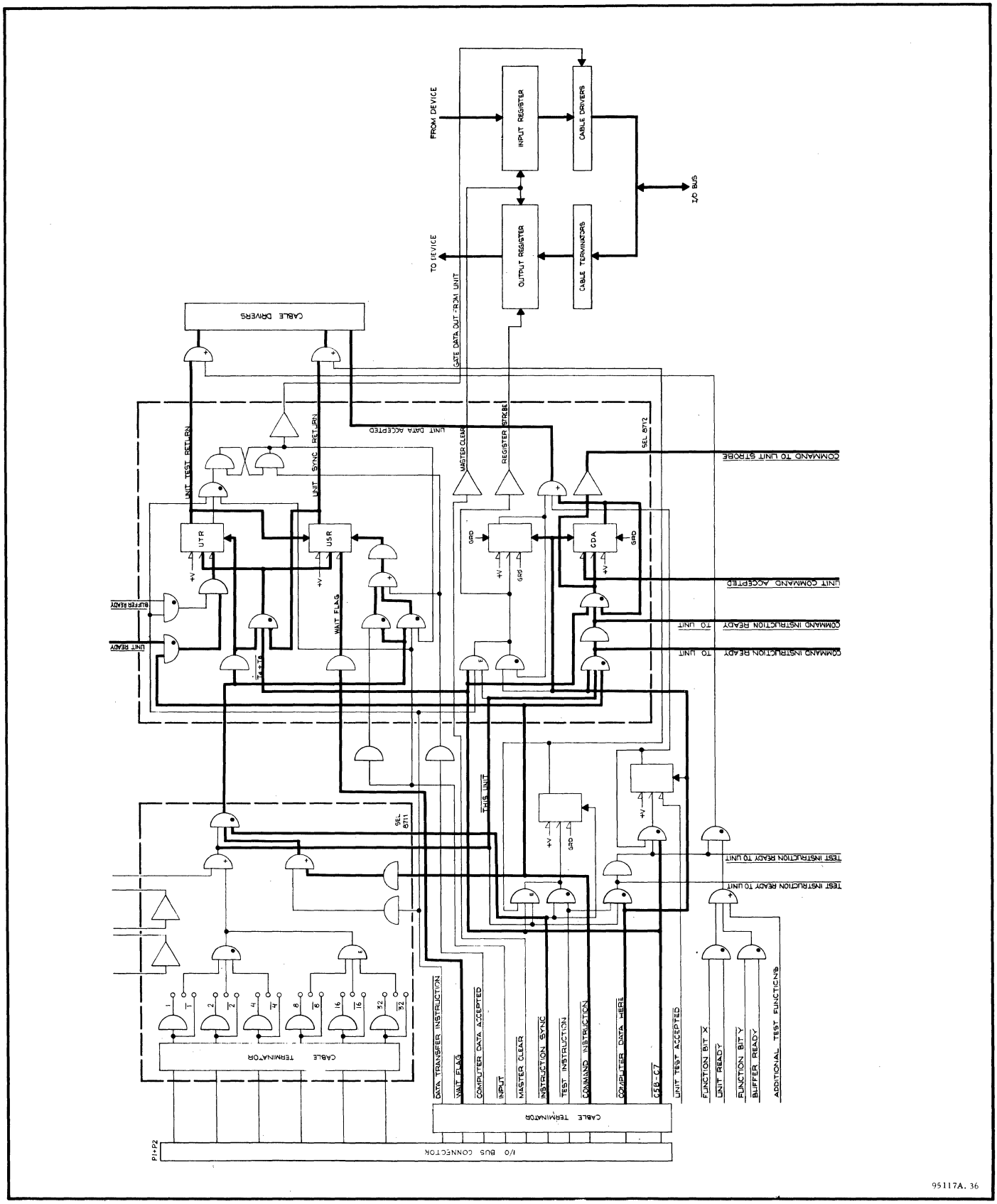


Figure 3-6. Data Terminal CEU Signal Flow

condition, and the mainframe will remain in the I cycle. If the wait flag is not present, then the Unit Test Return signal remains low, but the Unit Sync Return signal is raised.

The removal of the Instruction Sync level provides for removing the Unit Sync signal. If the device is ready, both the Unit Test Return and Unit Sync Return levels are produced. Again, the removal of the Instruction Sync level provides the means of resetting or removing the Unit Test Return and Unit Sync Return signals. The subsequent occurrence of the Computer Data Here signal will command the device to perform its function by gating the function bits into the proper circuits in the peripheral device. At the same time, it will raise the Unit Data Accepted line. The Computer Data Here signal remains for a long enough period for the unit to answer Data Accepted. The removal of Computer Data Here causes the device to remove the Unit Data Accepted signal.

Detailed Operation

The presence of the CEU instruction in the Instruction Register causes the Command Instruction line at the output of the cable terminator to go to zero volts. This zero volt level is applied through an inverter and an OR gate on the functional Unit Select card. The output of the OR gate is ANDed with the zero volt unit select level from the upper OR gate.

The Command Instruction zero volt level is also applied to partially enable two other AND gates; one shown on the drawing at the top of functional card 8712 and the other at the bottom of that card. The first gate is fully enabled when the Unit Ready is at zero volts. The output of the fully enabled AND gate is applied through an OR gate to disable the reset steering input of the UTR flip-flop, thus effectively enabling the set steering input of that flip-flop. The other AND gate partially enabled by the Command Instruction level is shown at the bottom of the drawing of the 8712 functional card and is input to a gate which is completely enabled by This Unit signal produced by the unit select card and the Computer Data Here signal.

The first timing signal to be sent to the data terminal for the CEU instruction is the Instruction Sync signal. This line goes to zero volts at approximately T5 time and causes the output of the AND gate on the device select card to go to + v. This 1 level is applied through an inverter to remove the dc reset signal from the UTR flip-flop and to partially enable an AND gate queried by the clock pulse. The output of the device select AND gate is also applied through two AND gates on an OR gate to remove the dc reset from the USR flip-flop.

The next clock pulse to query the AND gate enabled by the inverter output will be gated through to query both the UTR and USR flip-flops. This gated timing pulse will set the UTR flip-flop only if the device is ready due to the steering arrangement from the Unit Ready and Command Instruction AND gate connected to the reset steering input. The same timing pulse however will set the USR flip-flop to produce a Unit Sync Return signal if the instruction is programmed for the Skip mode. Note that the Wait Flag level will be at + v if the wait flag was not in the word format, the CEU instruction. This + v level is inverted and applied as a zero-volt level to the reset steering input of the USR flip-flop, thus effectively enabling the set steering input of that flip-flop. If the wait flag was present in the CEU instruction, then the output of the inverter and the input to the reset steering input will be + v. The presence of + v on both the set and reset steering inputs to the flip-flop will make its state impervious to any trigger strobe. Therefore, the only way that the Unit Sync Return signal can be produced is by the dc set input to the USR flip-flop. This will occur only if the UTR flip-flop is set. This will occur only when the device becomes ready and the reset steering input of the UTR flip-flop is disabled. The steering inputs of the UTR flip-flop are constantly tested by the clock pulse. Therefore, when the device UTR flip-flop is set, it produces a Unit Test Return signal which in turn immediately produces the Unit Sync Return signal by setting the USR flip-flop.

The generator of the Unit Test Return signal, when in the Skip mode, will cause the Program Register to advance twice after calling out the second word of the CEU instruction. If the Unit Test Return signal is not produced, that is, if the device test return signal remains low, then the program register will advance to the next instruction following the second word of the CEU instruction. The Unit Sync Return signal, always produced by this instruction, will release the I/O hold latch in the mainframe to allow the computer to read the second word of the CEU instruction from memory. When this word is available on the I/O bus and on the inputs to the device, the computer causes the Computer Data Here line to go to zero volts if the device test return line was not true. The zero volt level removes the dc set input to the CDA flip-flop and the dc reset signal from the UDA flip-flop. The same zero volt level also completes the enabled to the bottom AND gate on the Systems Engineering Laboratories 8712 card thus producing a + v Command Instruction Ready to Unit signal and through an inverter a zero volt Command Instruction Ready to Unit signal. The zero volt reset output of the CDA flip-flop combines with the Command Instruction Ready to Unit to partially

enable an AND gate on the trigger input line of the CDA flip-flop. The next clock timing pulse will be gated through this AND gate to query the trigger input and to produce a Command To Unit Strobe signal through an inverting buffer. The CDA flip-flop will be reset by the first clock pulse to occur following the Unit Command Accepted signal going to zero volts, and thus disabling the set steering input of that flip-flop. The Unit Command Accepted signal will go to zero volts as a result of the Command Instruction Ready to Unit signal and the subsequent use of this signal to strobe the function bits, etc., of the second word of the CEU instruction into the proper registers, latches, etc. The time between the generation of the Command Instruction Ready to Unit signal and the changing of the Unit Command Accepted level from + v to zero volts is entirely a function of the particular device and its electronic and circuit configuration. When the CDA flip-flop is reset, the reset output goes to + v and provides a zero volt level from the OR gate directly above the flip-flop. This OR gate then produces a Unit Data Accepted signal back to the mainframe to reset the mainframe control circuits to their initial configuration.

TEST EXTERNAL UNIT (TEU)

Function

The Test External Unit instruction is used to test an external device for some particular function. The instruction may test for: (1) device busy; (2) parity error; (3) bottom of form (for printer); (4) device inoperative; (5) not busy; (6) end-of-file, end-of-record, load point, write ring in, etc., (7) specific functions for specific equipment as designated by the designer of that equipment.

The TEU is a two-word instruction which is stored in two sequential memory locations. The instruction operates only in the Skip mode and is always presumed ready for a test. If the result of the test is true, then the Program Register in the mainframe is advanced twice following the unloading of the second word of the TEU instruction. If the tested-for condition is absent, then the Program Register is advanced to the next sequential instruction following the second word of the TEU instruction. This instruction does not contain the wait flag option.

The TEU instruction may be used in either the Immediate or the Address mode. The programmed mode is determined by the presence of a 1 or 0 in the Indirect flag position, bit 5, in the first word of the TEU instruction. If the Indirect flag position contains a 0 (Immediate mode), the second word from memory will contain the operand. The address mode utilizes the indirect address word format to contain the direct address. This format provides the direct address plus an Indirect flag bit and an Index flag bit. The address portion contains the 14 least significant bits of the memory address. The second most significant bit (bit 1) will be a 0 if bit 6 of the first word of the TEU instruction (the MAP bit) is a 0. If the MAP bit contains a 1, the most significant bit of the program register becomes the most significant bit of the address. Inasmuch as the most significant bit of the Program Register may be either a 0 or a 1, the inclusion of the MAP bit modifier allows the programmer to operate anywhere in the entire 32 K memory. The indirect address format is used because it allows the utilization of the Indirect flag for chaining and the Index flag for accessing a table of data words.

The final word obtained from memory at the end of an indirect chain is identical in format to the second word called from memory in the Immediate mode. The operand contains 16 bits which are coded to specify the function to be tested.

Peripheral - Computer Timing Relationships

The basic computer timing, described previously, consists of a continuous series of 750-nanosecond cycles. This cycle time is the time required for the memory to perform a complete read/write cycle; the cycle during which the instruction is unloaded from memory is referred to as the I (instruction) cycle; the memory cycles during which data is unloaded from or loaded into memory are described as E(Execution) cycles.

Each TEU instruction consists of the same cycle time as a CEU instruction. The first I cycle is a cycle during which the first word of the instruction is unloaded from memory. During this I cycle and the second I cycle, the device is prepared for the subsequent function bits used in the test instruction. Additional I cycles are also required if the instruction is used in the Address mode requiring more than one indirect cycle to obtain the second word from memory. When the second word containing the data (function code bits) is unloaded from memory at the end of the indirect chain or immediately if the Immediate mode is used, the computer is allowed to go into an execution cycle referred to as the E_1 cycle.

Execution Sequence

The basic instruction, if used in the Immediate mode, requires four complete memory cycle times. Refer to the timing diagram, figure 3-7, flow chart, figure 3-8, and terminal signal flow, figure 3-9. Note that during the first I cycle the command instruction is shown as being available

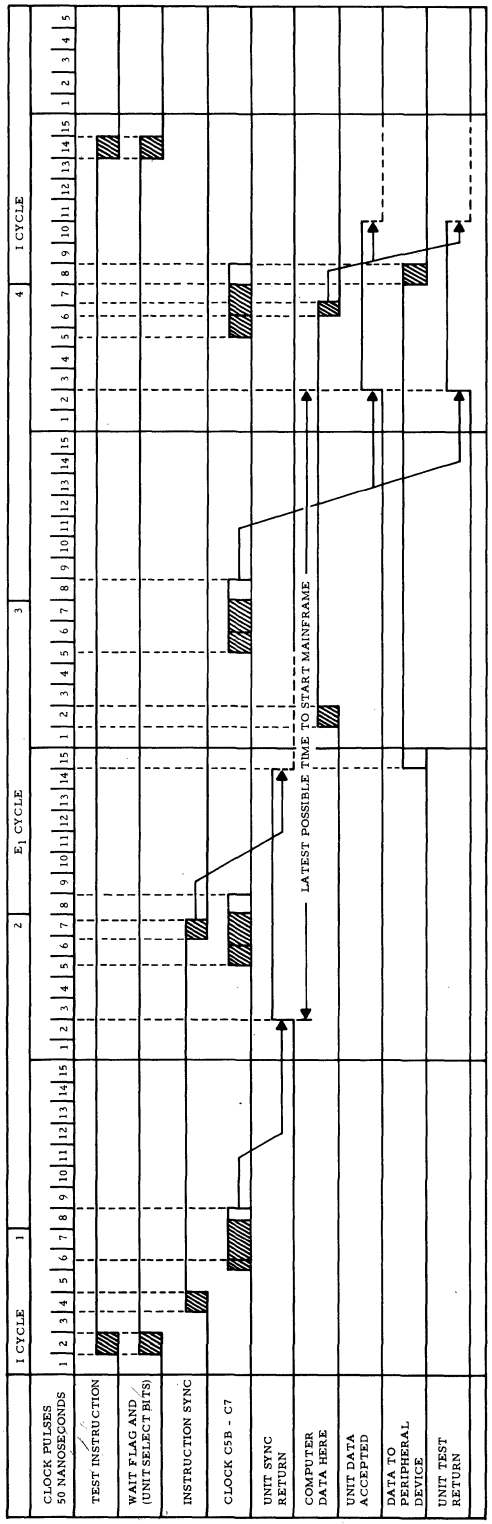


Figure 3-7. TEU Instruction Timing Diagram (At Computer Mainframe)

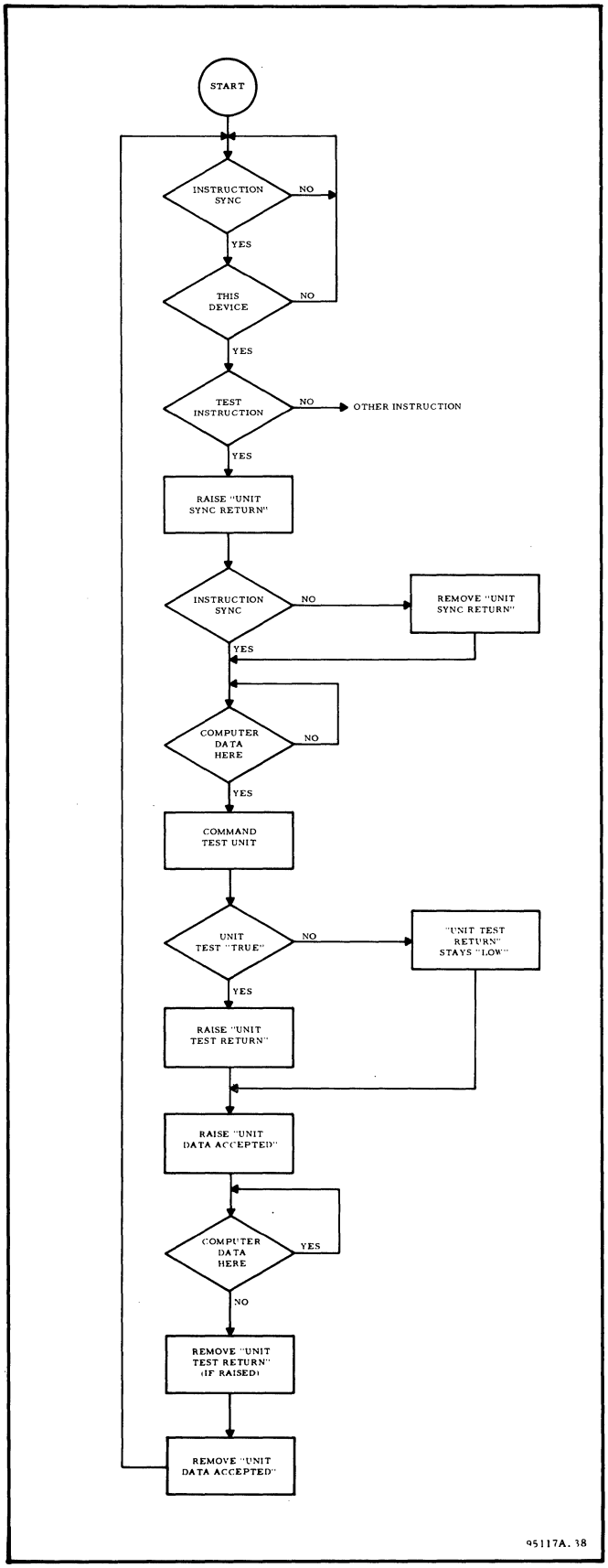


Figure 3-8. TEU Flow Chart

from the beginning of T3 time of the first I cycle continuously until the end of T13 time of the next subsequent instruction. During this entire period, the TEU instruction remains in the Instruction Register in the mainframe, and thus provides TEU enables to both the mainframe; and, through the mainframe circuits to the selected unit. The Instruction Register also holds the device (unit) number for this period.

At T5 time of the first I cycle, the computer circuits produce an Instruction Sync level which is gated to the selected device circuits by the device (unit) number. If the device is connected, the Instruction Sync signal gates a clock pulse to set a latch in the peripheral device which provides Unit Sync Return signal back to the mainframe. The Instruction Sync signal which initiated this action in the device also sets a latch in the mainframe which holds the mainframe to the I cycle. When the Unit Sync Return signal comes back to the mainframe it resets this latch and allows the computer to advance into the E_1 cycle during which the second word is unloaded from memory.

The computer provides a Computer Data Here signal to the device coincident with the time that the second word of the instruction is available on the I/O bus. The Computer Data Here signal causes the device to test the function specified by the second word, and raise the Unit Test Return signal if the test was true. The device raises the Unit Data Accepted signal to signify to the computer that the device test line is ready to be tested. If the Unit Test Return is true, the Program Register will be advanced twice so as to skip the next subsequent instruction. If the Unit Test Return signal is not received by the computer, the next sequential instruction will be executed. The computer removes the Computer Data Here signal after the Unit Test Return signal has been tested, which causes the device to remove both Unit Test Return and Unit Data Accepted signals.

The input sequence of the TEU instruction as it pertains to the device is shown in the TEU flow diagram. The first signal produced by the mainframe to the unit is the Instruction Sync level. When that signal occurs, and the device is selected by the device (unit) number, it will produce a Unit Sync Return level.

The next signal produced by the mainframe is the Computer Data Here signal. When that signal arrives it is gated to test the various function bits as compared to the functions which they specify. If the tested-for condition is true, then the Unit Test Return signal is produced and in turn produces the Unit Data Accepted signal. The Computer Data Here signal also removes the

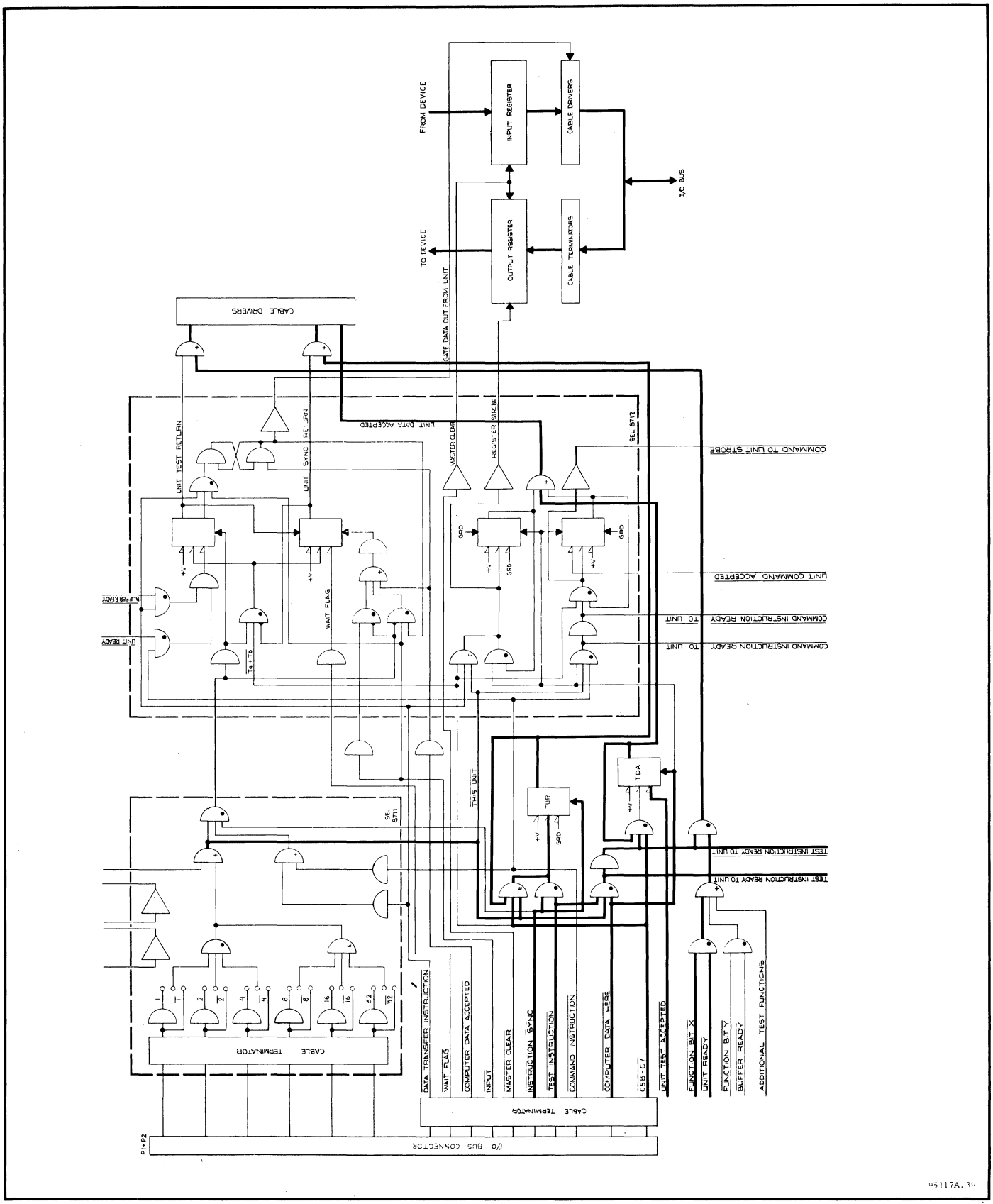


Figure 3-9. Data Terminal TEU Signal Flow

Unit Test Return and the Unit Data Accepted levels.

Detailed Operation

The presence of the TEU instruction in the Instruction Register causes the Test Instruction signal to go to zero volts. This zero volt level partially enables two AND circuits. One of these is composed of a two-input AND gate connected with a three-input expander and providing the trigger input to the UTR flip-flop. This AND gate is completely enabled when the UTR flip-flop is reset and the device is selected and the Instruction Sync signal goes to + 0 volts. The occurrence of the first computer-produced timing signal, the Instruction Sync level gates the next clock pulse through the AND circuit to set the UTR flip-flop. The resulting + v level from the set output of this flip-flop inhibits the trigger input AND gate and also provides a Unit Sync Return signal back to the mainframe.

The next signal produced by the mainframe is the Computer Data Here which goes to zero volts as when the second word of the TEU instruction becomes available on the I/O bus. This level AND's with the Test Instruction zero volt level and This Unit zero volt level produced by the device select card to provide a Test Instruction Ready to Unit and through an inverter a Test Instruction Ready to Unit signal. The Computer Data Here signal also removes the dc reset signal from the TDA flip-flop. The Test Instruction Ready to Unit signal gates the next clock pulse as a trigger strobe to the TDA flip-flop. The Test Instruction Ready to Unit also partially enables another AND gate which will be fully enabled if the function bit and the tested for condition are found to coincide. If this condition is found to be true, then the two inputs to the AND gate go to zero volts and the output is of the AND gate is applied through an OR gate as a Unit Test Return level. Sometime following this signal, the Unit Test Accepted signal goes to zero volts to allow the next clock pulse to trigger the TDA flip-flop and set it. The resulting + v output of the TDA flip-flop is applied through the NOR gate as a Unit Data Accepted signal which goes back to the mainframe and returns the Computer Data Here signal to + v. When this occurs, the TDA flip-flop is dc reset and the device and the Data Terminal as well as the mainframe returns to their initial configurations.

OUTPUT DATA TRANSFER INSTRUCTIONS (MOP AND AOP)

Function

There are two output data transfer instructions available on the SEL 810B Computer. These are the Accumulator Out to Peripheral (AOP) and the Memory Out to Peripheral (MOP) instructions. The AOP instruction is a single-word instruction which transfers data from the A-Accumulator to the selected device. The MOP instruction transfers data from a specified memory cell to the peripheral device. The MOP and the AOP instructions require a minimum of three memory cycle times to be executed. Both instructions may operate in either the Skip or the Wait mode of operation; but only the MOP instruction may be operated in the Immediate or Address mode.

In addition to the function code, the AOP instruction word contains a wait flag in bit position 9 and the device (unit) number to which the data is to be transferred in bit positions 10 through 15. If the wait flag position contains a 1, this causes the computer to operate in Wait mode. In this mode, the device will not answer with Unit Sync Return until it is ready to accept the data; therefore, causing the computer to wait. The presence of the 0 in the wait flag position will cause the computer to operate in a Skip mode. The Skip mode causes the computer to access the next sequential instruction following the AOP instruction if the device is not ready; but to skip the next sequential instruction and obtain the subsequent instruction if the device is ready. When the computer is in a wait cycle, it cannot be interrupted by any priority interrupt except for the optional power fail safe/restore or stall alarm interrupts. If however, the Skip mode is used, the computer may be programmed through use of a BRU instruction to cycle until the device is ready before proceeding. This allows the servicing of priority interrupts that might occur between the AOP instruction and the BRU instruction.

The MOP is a two-word instruction which is stored in two sequential memory locations. The instruction may operate in either the Wait or Skip modes and either Immediate or Address modes of operation. The first word of this instruction sets the modes of operation by flags contained in bits positions 5 and 9. Bit 9 contains the wait flag which is set to a 1 to cause the computer to operate in the Wait mode. This mode will cause the device not to answer Unit Sync Return until it is ready to receive the data word, causing the computer to wait. The presence of a 0 in the wait flag position of the first word will cause the computer to operate in the Skip mode. The Skip mode causes the computer to execute the next sequential instruction following the second word of the MOP instruction if the device is not ready; but to skip the next sequential instruction and obtain the subsequent instruction if the device is ready. When the computer is in a wait cycle it cannot be interrupted by any priority interrupt except for the

optional power fail safe/restore or stall alarm interrupts. If the Skip mode is used, the computer may be programmed through the use of a BRU instruction to cycle until the device is ready before proceeding. This allows servicing of priority interrupts that might occur between the MOP instruction and the BRU instruction.

The Indirect flag located in bit position 5 of the first word of the MOP instruction defines whether the instruction will operate in the Immediate mode or the Address mode. If the Indirect flag position contains a 0, the second word unloaded from memory becomes the operand. The Address mode (bit 5 contains a 1) utilizes the indirect address word format to contain the direct address. This format provides a direct address plus Indirect and Index flags. The address portion contains the 14 least significant bits of the memory address. The 15th bit will be a 0 if bit 6 of the first word, the MAP bit, is a 0. If the MAP bit contains a 1, the most significant bit of the program register becomes the most significant bit of the direct address. Inasmuch as the most significant bit of the program register may be either a 0 or 1, inclusion of the MAP bit modifier allows the programmer to operate anywhere in the entire 32K memory. The indirect address format is used because it allows the utilization of the Indirect flag for c haining the Index flag for accessing tables of words.

The word obtained from memory at the end of an indirect chain is identical in format to the second word called from memory in the Immediate mode. This second word contains the data word that is to be transferred to the peripheral device.

Peripheral - Computer Timing Relationships (AOP)

The basic computer timing consists of a continuous series of 750-nanosecond cycles. This cycle time is a time required for a memory to perform a complete read/write cycle. The cycle during which the instruction is unloaded from memory is referred to as the I (instruction) cycle.

Each AOP instruction consists of at least two I cycles. The first I cycle is a cycle during which the AOP instruction word is unloaded from memory. The second I cycle is the time during which the device is tested to see whether it is ready to accept a data transfer from the computer. The second I cycle may be repeated indefinitely if the instruction is used in the Wait mode. When the device has responded that it is ready for a transfer of data from the computer then the computer is allowed to go into E1 cycle during which the data from the A-Accumulator is transferred to the device.

Peripheral - Computer Timing Relationships (MOP)

Each MOP instruction consists of at least two I cycles and one E cycle. The first I cycle is a cycle during which the first word of the instruction is unloaded from memory. The second I cycle is a time which the device is tested to see whether it is ready to accept a data transfer from the computer. The second I cycle may be repeated indefinitely if the instruction is operated in the Wait mode. Additional I cycles are also required if the instruction is used in the Address mode requiring more than one indirect cycles. When the second word containing the data is unloaded from memory at the end of an indirect chain, or immediately if the Immediate mode is used, and the device has responded that it is ready for a transfer of data from the computer, then the computer is allowed to go into an execution cycle referred to as E_1 cycle.

AOP/MOP Execution Sequence

The execution sequence, insofar as a ready peripheral device is concerned, is identical for both the AOP and MOP instructions. The basic instructions used in the Skip and Immediate Address modes requires a minimum of three complete memory cycle times. Refer to the timing diagram in figure 3-10, the flow chart in figure 3-11, and the terminal signal flow figure 3-12. Note that during the first I cycle the Data Transfer Instruction enable is shown as being available from the beginning of T3 time of the first I cycle continuously until the end of T13 time of the I cycle of the next subsequent instruction. During this entire period, the MOP or AOP instruction remains in the Instruction Register in the mainframe, and thus provides the output transfer enables to both mainframe, and through mainframe circuits to the selected unit. The Instruction Register also holds the wait flag and the device (unit) number for this period.

At T5 time of the first I cycle, the computer circuits produce an Instruction Sync level which is gated through the selected device circuits by the device (unit) number. If the device is ready, the Instruction Sync signal gates a clock pulse to set a latch in the peripheral device which provides simultaneously a Unit Test Return and Unit Sync Return back to the mainframe. The Instruction Sync signal which initiated this action in the device also sets a latch in the mainframe which holds the mainframe to the I cycle. When the Unit Sync Return signal comes back to the mainframe, it resets this latch and allows the computer to advance into the E_1 cycle during which time the second word is unloaded from memory (or transferred from the A-Accumulator). The

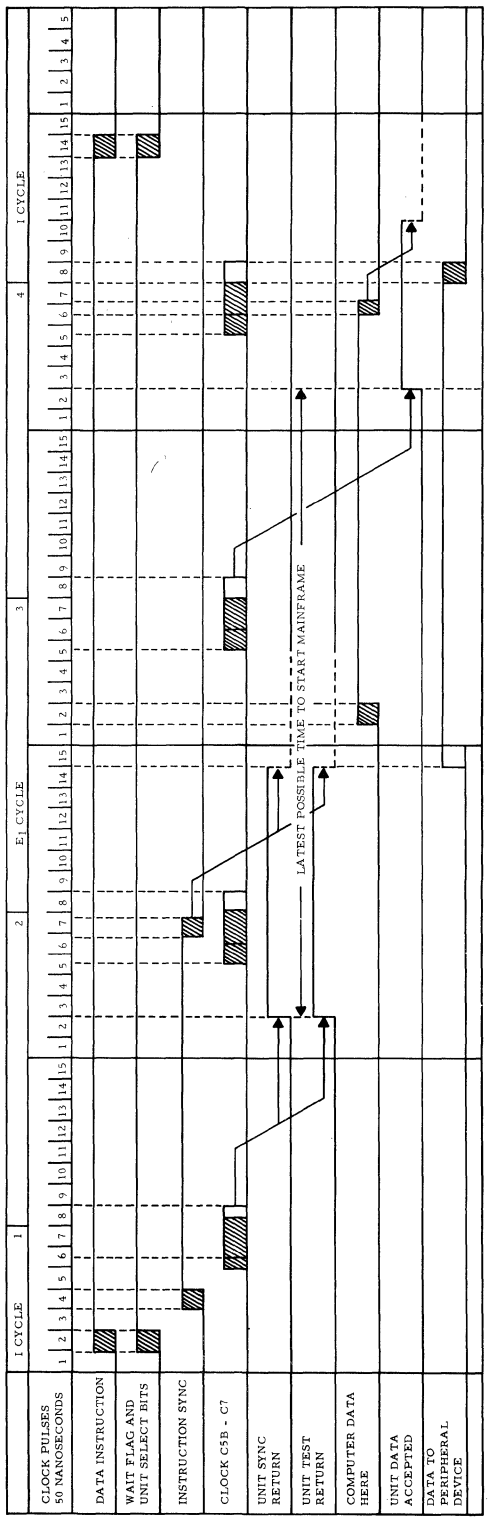


Figure 3-10. AOP/MOP Instruction Timing Diagram (At Computer Mainframe)

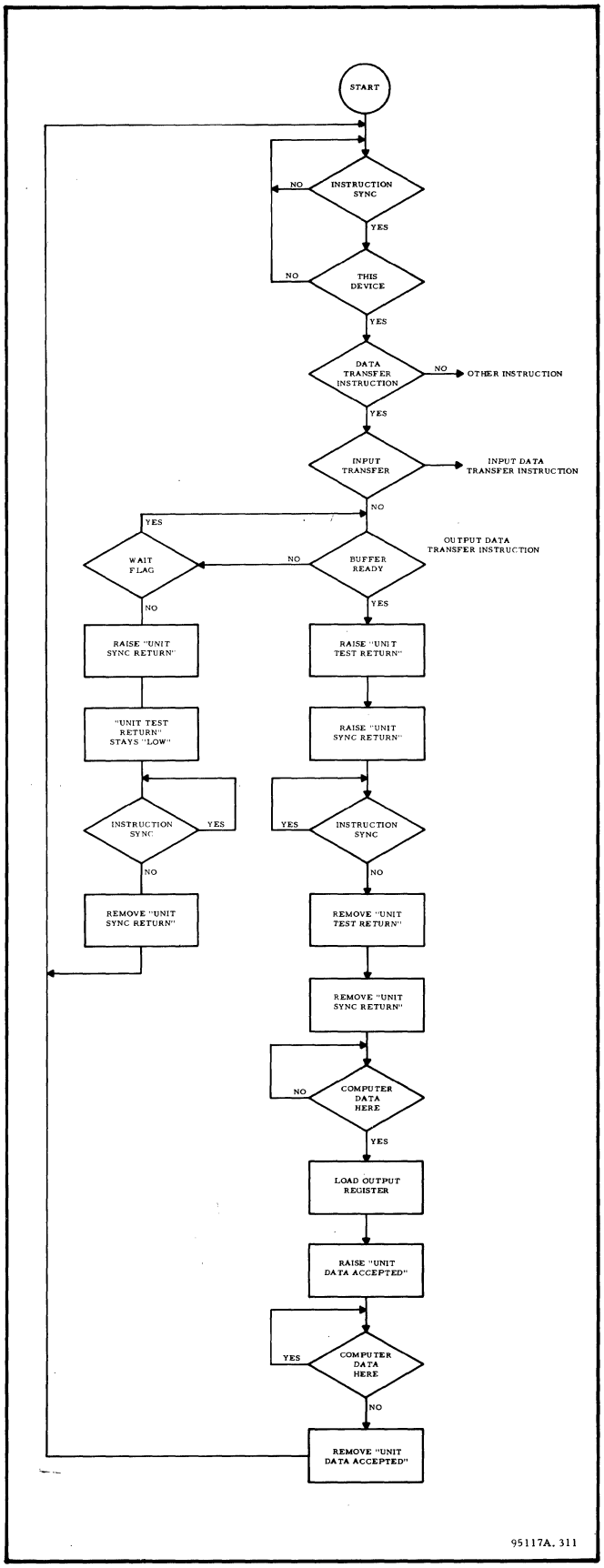


Figure 3-11. AOP/MOP Flow Chart

Unit Test Return signal is used to perform the skip test. If the Unit Test Return signal is received, the program register will be advanced twice so as to skip the next subsequent instruction. If the Unit Test Return signal does not come back, then the program register will be advanced once and call the next sequential instruction. The Unit Test Return signal also provides a Computer Data Here signal from the computer to the unit coincident with the time that the data word provided by the instruction is available on the I/O bus. The Computer Data Here signal sets a latch which provides a signal back to the computer termed Unit Data Accepted. The Unit Data Accepted signal is produced at the same time that an Output Register Strobe signal is generated in the peripheral device. The Output Register Strobe loads the data word provided by the memory or the A-Accumulator into the Output Register of the device.

Output transfer is controlled by AOP and MOP instructions. The first signal produced by the mainframe to the device is the Instruction Sync level. When that signal occurs, if the output buffer is ready for a new word to be strobed in and, if the device is selected by the device (unit) number, the device will produce a Unit Test Return signal. Note that if the buffer is not ready for a transfer to be performed when the Instruction Sync level is generated by the mainframe, a test is made to determine if the wait flag is present in the original instruction. If the wait flag is present, the device will continually test for the Buffer Ready condition and the mainframe will remain in the I cycle. If the wait flag is not present, then the Unit Test Return signal remains low, but the Unit Sync Return signal is returned.

The removal of the Instruction Sync level, also provides for the removing of the Unit Sync Return signal. If the buffer is ready, both the Unit Test Return and the Unit Sync Return levels are produced. Again, the removal of the Instruction Sync level provides the means of resetting or removing the Unit Test Return and Unit Sync Return signals. The subsequent occurrences of the Computer Data Here signal will provide the signal to load the output register and produce the Unit Data Accepted signal. The Computer Data Here signal then remains until the computer has received the Unit Data Accepted signal. The removal of computer data here resets the Unit Data Accepted flip-flop.

AOP, MOP Detailed Operation

The presence of either an AOP or a MOP instruction in the Instruction Register will cause the Data Transfer Instruction level to go to zero volts. This zero volt level is applied through an inverter

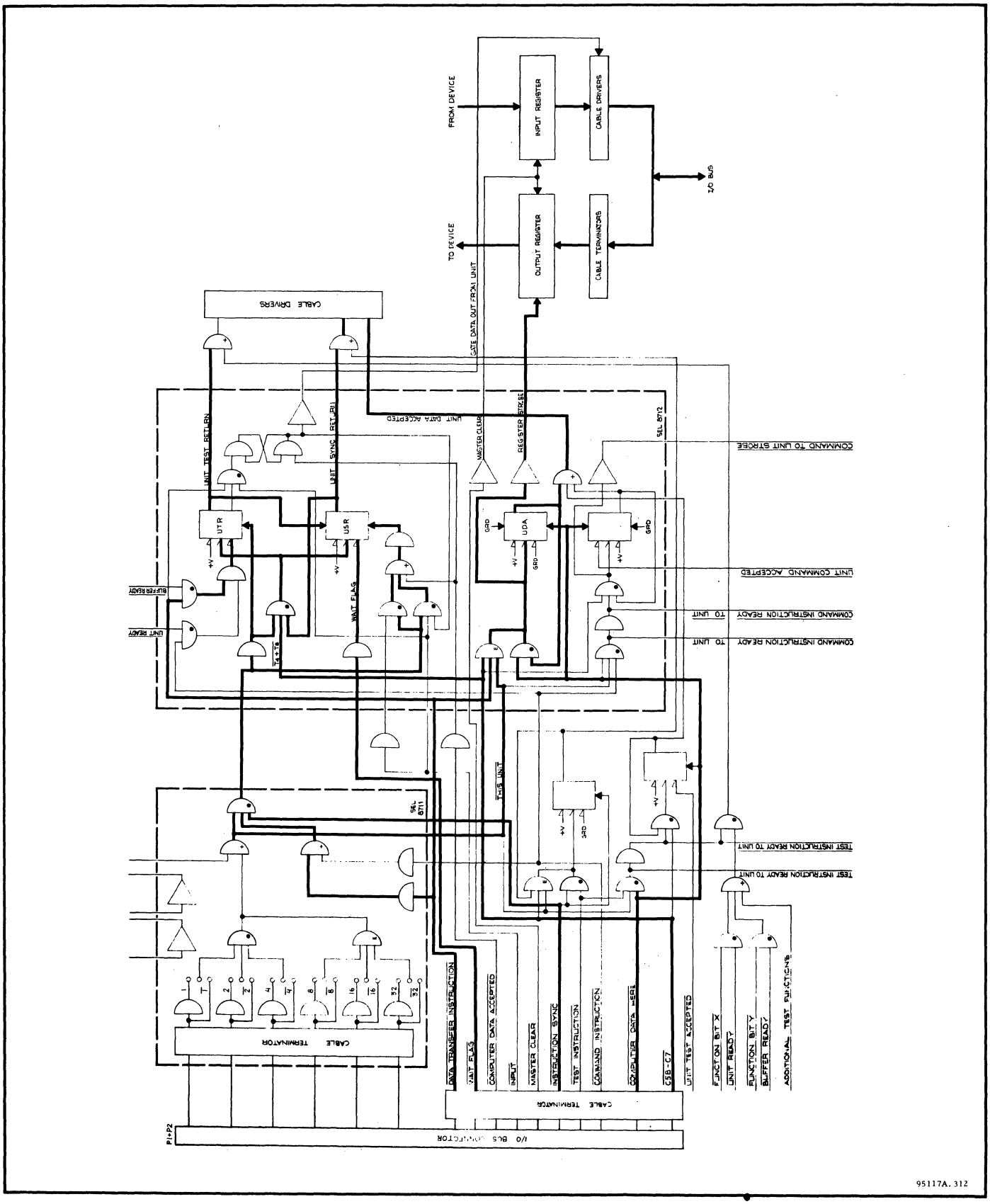


Figure 3-12. Data Terminal AOP/MOP Signal Flow

and an OR gate to enable the output AND gate on the device select function card. The Data Transfer Instruction zero volt level also enables an AND gate shown at the top of SEL 8712 card drawing. The second input to this AND gate will be at zero volts when the buffer is ready. The output of this AND gate is applied through an OR gate to disable the reset steering input of the UTR flip-flop when the buffer is ready. The disabling of the reset steering input effectively enables the set steering input. The Data Transfer Instruction zero volt level also partially enables a four-input AND gate which provides the triggering input to the UDA flip-flop. The other enabling inputs to the four-input AND gate are This Unit zero volt level and the normally zero volt set output of the UDA flip-flop itself.

The first signal produced by the computer to the device is the Instruction Sync signal which goes to zero volts at the output of the cable terminator. This zero volt level completes the enables to the device select AND gate. The output of this circuit removes the dc reset to the UTR flip-flop through an inverter and enables an AND gate which connects to the trigger inputs of both the UTR and USR flip-flops. This gate allows a clock pulse to query the trigger input of both flip-flops. If the transfer instruction contains a wait flag, the reset input of the USR flip-flop will be at + v. The set steering input is permanently held at + v so the clock pulse trigger strobes have no affect. The UTR flip-flop will be set only when the Buffer Ready level goes to zero volts to effectively enable the set steering input of that flip-flop. When the flip-flop is set, the Unit Test Return signal will be produced and sent to the computer. At the same time, the signal sets the USR flip-flop to provide the Unit Sync Return signal. In the Skip mode the USR flip-flop will be set by the first clock pulse to be gated by the Instruction Sync and Data Transfer Instruction enables. The Unit Test Return line will reflect the condition of buffer ready or not ready.

The next timing signal to be gated to the device by the mainframe is the Computer Data Here signal if the device is ready. The zero volt level completes the enable on the four-input AND gate controlling the trigger input of the UDA flip-flop. The resulting output to the AND gate will set the UDA flip-flop to produce a Unit Data Accepted signal and will also provide a strobe to load the output data word into the output register. The return of the Computer Data Here signal to + v at the end of the AOP or MOP instruction dc resets the UDA flip-flop to remove the Unit Data Accepted signal.

INPUT DATA TRANSFER INSTRUCTIONS (MIP AND AIP)

Function

There are two input data transfer instructions available on the SEL 810A Computer. These are the Accumulator In From Peripheral (AIP) and the Memory In From Peripheral (MIP) instructions. The AIP instruction is a single-word instruction which transfers data from the selected device to the A-Accumulator. The MIP instruction transfers data from the selected device to a specified memory cell. The MIP and AIP instructions require a minimum of three memory cycle times to be executed. Both instructions may operate in either the Skip or the Wait modes, but only the MIP instructions may be operated with the Immediate or Address mode.

In addition to the function code, the AIP instruction word contains a wait flag in bit position 9 and the device unit number to which the data is to be transferred in bit positions 10 through 15. If the wait flag position contains a 1, this causes the computer to operate in Wait mode. In this mode, the device will not answer Unit Sync Return until it is ready to transfer data. The presence of the 0 in the wait flag position will cause the computer to operate in a Skip mode. The Skip mode causes the computer to access the next sequential instruction following the AIP instruction if the device is not ready; but to skip the next sequential instruction and obtain the subsequent instruction if the device is ready. When the computer is in a wait cycle, it cannot be interrupted by any priority interrupt except for the optional power fail safe/restore or stall alarm interrupts. If however, the Skip mode is used, the computer may be programmed through use of a BRU instruction to cycle until the device is ready before proceeding. This allows the servicing of priority interrupts that might occur between the AIP instruction and the BRU instruction.

The AIP instruction format also includes a merge flag in bit position 4 for programming convenience. If this flag is a 0, the input data word is loaded into all 16-bit positions of the A-Accumulator. If this word is an eight-bit character, the character is loaded into bit positions 8 through 15 and zeros are loaded into bit positions 0 through 7. If the merge flag is a 1, only the eight least significant bits (a character) are loaded into bit positions 8 through 15 and the contents of bit positions 0 through 7 (a prior input character) are undisturbed.

NOTE

If bits 8 through 15 contain data when the merge operation occurs, the new character will be ORed with the previous data.

The MIP instruction is a two-word instruction which is stored in two sequential memory locations. The instruction may operate in either the Wait or

Skip modes and either Immediate or Address modes of operation. The first word of this instruction sets the modes of operation by flags contained in bit positions 5 and 9. Bit 9 contains the wait flag which is set to a 1 to cause the computer to operate in the Wait mode. This mode will cause the device not to answer Unit Sync Return until it is ready to transfer data. The presence of a 0 in the wait flag position of the first word will cause the computer to operate in the Skip mode. The Skip mode causes the computer to exit to the next sequential instruction following the second word of the MOP instruction if the device is not ready; but to skip the next sequential instruction and obtain the subsequent instruction if the device is ready. When the computer is in a wait cycle it cannot be interrupted by any priority interrupt except for the optional power fail safe/restore or stall alarm interrupts. If the Skip mode is used, the computer may be programmed through the use of a BRU instruction to cycle until the device is ready before proceeding. This allows servicing of priority interrupts that might occur between the MIP instruction and the BRU instruction.

The Indirect flag located in bit position 5 of the first word of the MIP instruction defines whether the instruction will operate in the Immediate mode or the Address mode. If the Indirect flag position contains a 0, the second word becomes the location in which the data word is to be loaded. The Address mode (the Indirect flag is a 1) utilizes the indirect address word format to contain the direct address. This format provides a direct address plus indirect and index flags. The address portion contains the 14 least significant bits of the memory address. The 15th bit will be a zero if bit 6 of the first word, the MAP bit, is a 0. If the MAP bit contains a 1, the most significant bit of the program register becomes the most significant bit of the direct address. Inasmuch as the most significant bit of the Program Register may be either a 0 or 1, inclusion of the MAP bit modifier allows the programmer to operate anywhere in the entire 32K of memory. The indirect address format is used because it allows the utilization of the Indirect flag for chaining and the Index flag for accessing tables of words.

Peripheral - Computer Timing Relationships (AIP)

The basic computer timing consists of a continuous series of 750-nanosecond cycles. This cycle is a time required for a memory to perform a complete read/write cycle. The cycle during which the instruction is unloaded from memory is referred to as the I (instruction) cycle.

Each AIP instruction consists of at least two I cycles and one E cycle. The first I cycle is a cycle during which the AIP instruction word is unloaded from memory. The second I cycle is the time during which the device is tested to see whether it is ready to provide a data transfer to the computer. The second I cycle may be repeated indefinitely if the instruction is used in the Wait mode. When the device has responded that it is ready to provide a transfer of data to the computer, then the computer is allowed to go into the E1 cycle during which the data from the device is transferred to the A-Accumulator.

Peripheral - Computer Timing Relationships (MIP)

Each MIP instruction consists of at least two I cycles and one E cycle. The first I cycle is a cycle during which the first word of the instruction is unloaded from memory. The second I cycle is a time which the device is tested to see whether it is ready to provide a data transfer to the computer. The second I cycle may be repeated indefinitely if the instruction is operated in the Wait mode. Additional I cycles are also required if the instruction is used in the address mode requiring more than one indirect cycle. When the address of the word to receive the data is read from memory at the end of an indirect chain or immediately if the Immediate mode is used, and the device has responded that it is ready to transfer data to the computer, then the computer is allowed to go into an execution cycle referred to as E_1 cycle.

AIP/MIP Execution Sequence

The execution sequence, insofar as a ready peripheral device is concerned, is identical for both the AIP and MIP instructions. The basic instructions used in the Skip and Immediate modes requires a minimum of three complete memory cycle times. Refer to the timing diagram on figure 3-13, the flow chart on figure 3-14, and the terminal signal flow on figure 3-15. Note that during the first I cycle the Data Transfer Instruction enable is shown as being available from the beginning of T3 time of the first I cycle continuously until the end of T13 time of the I cycle of the next subsequent instruction. During this entire period, the MIP or AIP instruction remains in the Instruction Register in the mainframe, and thus provides the output transfer enables to both mainframe and, through mainframe circuits, to the selected device. The Instruction Register also holds the wait flag and the device (unit) number for this period.

At T5 time of the first I cycle, the computer circuits produce an Instruction Sync level which is gated through the selected device circuits by the unit code. If the device is ready, the Instruction Sync signal gates a clock pulse to set a latch in the peripheral device which provides simultaneously

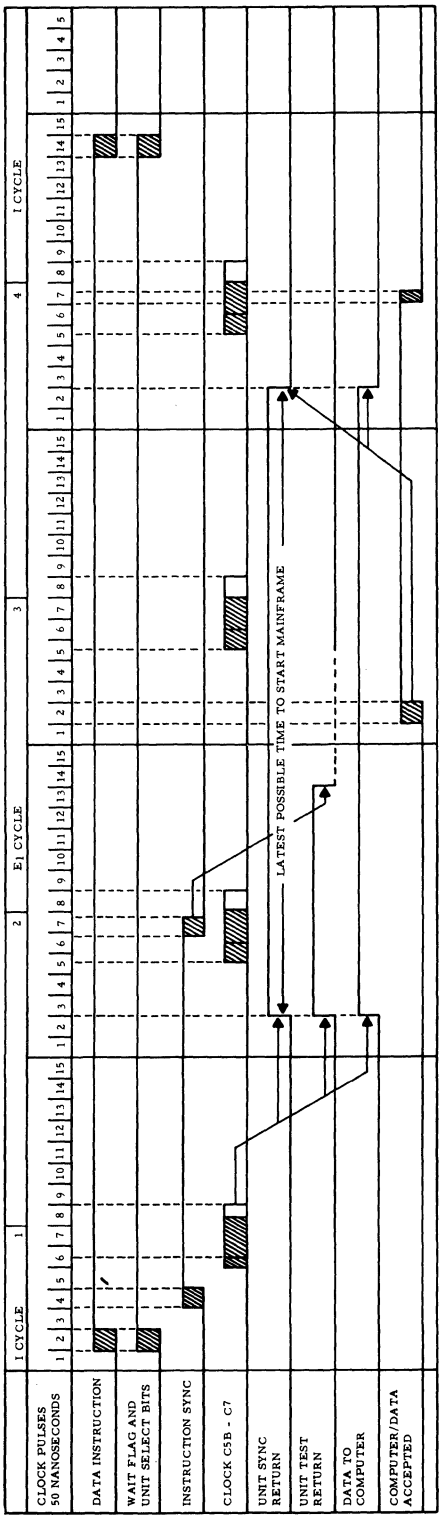


Figure 3-13. AIP/MIP Instruction Timing Diagram (At Computer Mainframe)

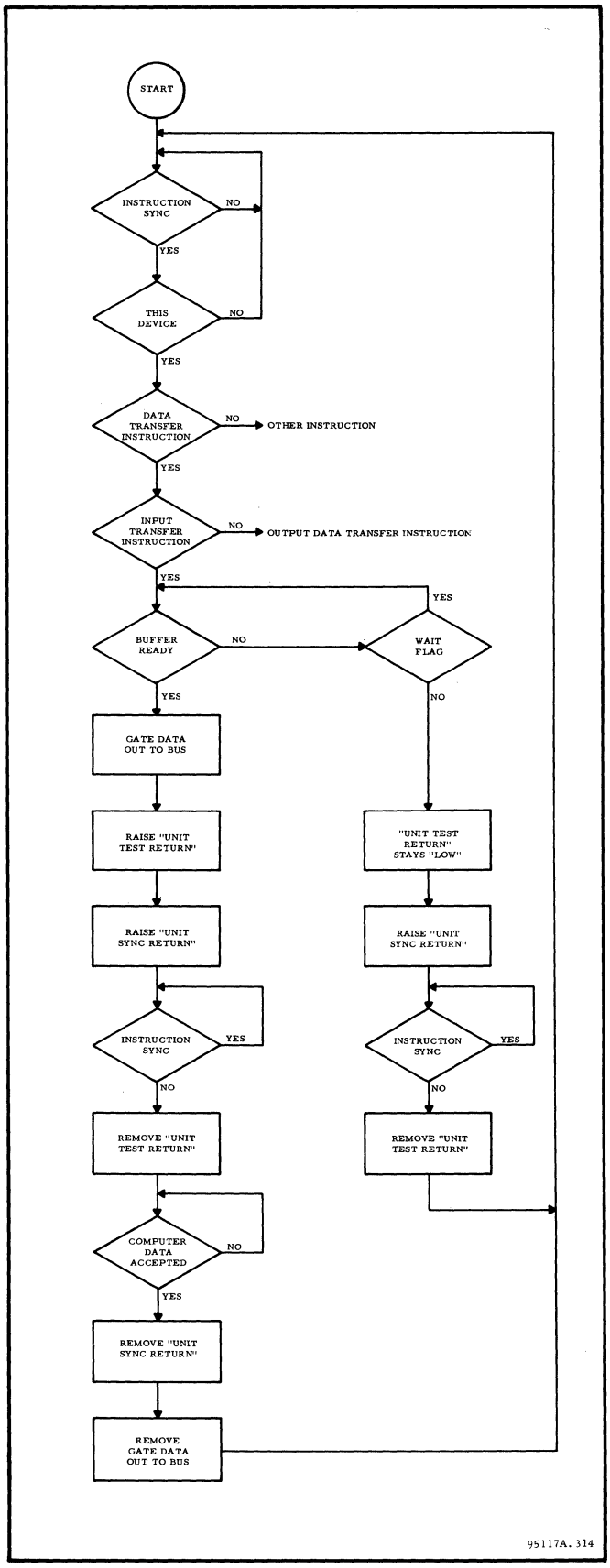


Figure 3-14. AIP/MIP Flow Chart

a Unit Test Return and a Unit Sync Return back to the mainframe. The Instruction Sync signal which initiated this action in the device also sets a latch in the mainframe which holds the mainframe to the I cycle. When the Unit Sync Return signal comes back to the mainframe, it resets this latch and allows the computer to advance into the E_1 cycle during which the second word is loaded into memory (or transferred to the A-Accumulator). The Unit Test Return signal is used to perform the skip test. If the Unit Test Return signal is received, the program register will be advanced twice so as to skip the next subsequent instruction. If the Unit Test Return signal does not come back, then the Program Register will be advanced once and call the next sequential instruction. The Unit Test Return signal also provides a Gate Data Out To Computer signal to the Input Register in the device. The computer then provides a Computer Data Accepted signal that resets the Gate Data Out To Computer and Unit Sync Return levels.

Input transfer is controlled by AIP and MIP instructions, the first signal produced by the mainframe to the device is the Instruction Sync level. When that signal occurs and the device is ready for a new word to be strobed in, and if the device is selected by the device (unit) number, the device will produce a Unit Test Return signal. Note that if the buffer is not ready for a transfer to be performed when the Instruction Sync level is generated by the mainframe, a test is made to determine if the wait flag is present in the original instruction. If the wait flag is present, the device will continually test for the Buffer Ready condition and the mainframe will remain in the I cycle. If the wait flag is not present, then the Unit Test Return signal remains low, but the Unit Sync Return signal is returned.

The removal of the Instruction Sync level, provides for the removing of the Unit Sync Return. If the buffer is ready, both the Unit Test Return and the Unit Sync Return levels are produced. Again, the removal of the Instruction Sync level provides the means of resetting or removing the Unit Test Return and Unit Sync Return signals. The subsequent occurrence of the Computer Data Accepted signal will provide the means to gate the input word to the computer. The Computer Data Accepted signal removes the Unit Sync Return and the Gate Data Out To Bus signals.

AIP, MIP Detailed Operation

The presence of either a AIP or an MIP instruction in the instruction register will cause the Data Transfer Instruction and Input levels to go to zero volts. The Data Transfer Instruction zero volt level is applied through an inverter and an OR gate

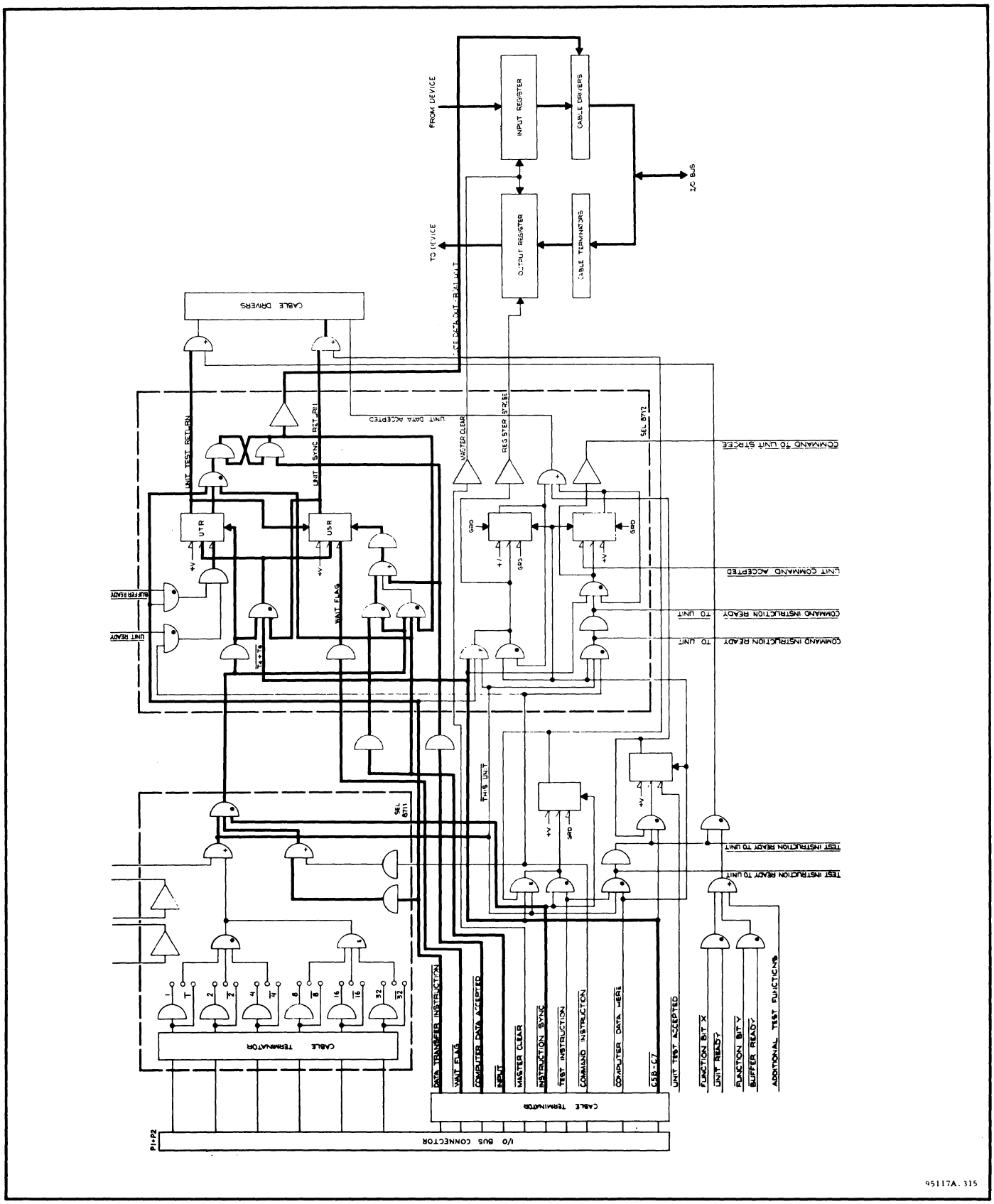


Figure 3-15. Data Terminal AIP/MIP Signal Flow

to enable the output AND gate on the unit select functional card. The Data Transfer Instruction zero volt level also enables an AND gate shown at the top of SEL 8712 card drawing. The second input to this AND gate will be at zero volts when the buffer is ready. The output of this AND gate is applied through an OR gate to disable the reset steering input of the UTR flip-flop when the buffer is ready. The disabling of the reset steering input effectively enables the set steering input.

The Input zero volt level enables an AND gate on the output of the UTR flip-flop to set a NOR latch when the UTR flip-flop is set. The latch output provides the Gate Data Out To Computer signal to the Input Register. The first signal produced by the computer to the device is the Instruction Sync signal which goes to zero volts. This zero volt level completes the enables to the device select AND gate. The output of this circuit removes the dc reset to the UTR flip-flop through an inverter and enables an AND gate which connects to the trigger inputs of both the UTR and USR flip-flops. This gate allows the clock pulse to query the trigger input of both flip-flops. If the transfer instruction contains a wait flag, the reset input of the USR flip-flop will be at + v. The set steering input is permanently held at + v so the clock pulse strobe has no affect. The UTR flip-flop will be set only when the Buffer Ready level goes to zero volts to effectively enable the set steering input of that flip-flop. When the flip-flop is set, the Unit Test Return signal will be produced and sent to the computer. At the same time, the signal sets the USR flip-flop to provide the Unit Sync Return signal. In the Skip mode the USR flip-flop will be set by the first clock pulse to be gated by the Instruction Sync and Data Transfer Instruction enables. The Unit Test Return line will reflect the condition of buffer ready or not ready.

The next timing signal to be gated to the device by the mainframe is the Computer Data Accepted signal which goes to zero volts if the unit was ready. This zero volt level is inverted to reset the USR flip-flop and the Gate Data Out From Unit NOR latch.

I/O PARITY

The I/O parity option provides a parity check on data transferred between the computer and peripheral units. A priority interrupt is normally used in conjunction with the I/O parity in order to process the parity error.

All data transferred to a peripheral device contains an additional parity bit allowing the device to perform a parity check; however, the standard devices for the computer do not have parity check capability unless they are purchased as such. In order to utilize the I/O parity check option, the device must be prepared to check parity on an output data transfer instruction during the time the Computer Data Here is present. If a parity error is detected during this interval, a priority interrupt in the device is set.

The computer uses the memory parity generator to perform a parity check on input data from a peripheral device. During an input data transfer instruction, the device must enable the parity request line to the computer. This line should be enabled during the gate data out time. The gate data out latch can be connected directly to the cable driver that provides the parity request. The device also must contain a parity generator to provide a parity bit to the computer. During the time that the data word from the peripheral device is being loaded into either the computer memory or the A-Accumulator, a parity check is performed. If an error is recognized during an input data transfer, the error line to the peripheral device is enabled and in turn the device can enable its priority interrupt so that the error can be processed.

Figure 3-16 illustrates the logic and signal lines required to have I/O parity capability. The lines which interface the peripheral device and the computer refer to the I/O bus which is described in detail in Section V.

I/O CABLE SIGNAL DELAYS

Input/Output cable signal delays are dependent on cable length. The following timing charts illustrate four cases that exist with a 10-foot cable length.

a. Figure 3-17 illustrates the computer executing an input instruction with best signal delay case existing.

NOTE

The term best case indicates minimum signal time delay at a specified length.

b. Figure 3-18 illustrates the computer executing an input instruction with worst signal delay case existing.

NOTE

The term worst case indicates maximum signal time delay at a specified length.

c. Figure 3-19 illustrates the computer executing an output instruction with best signal delay case existing.

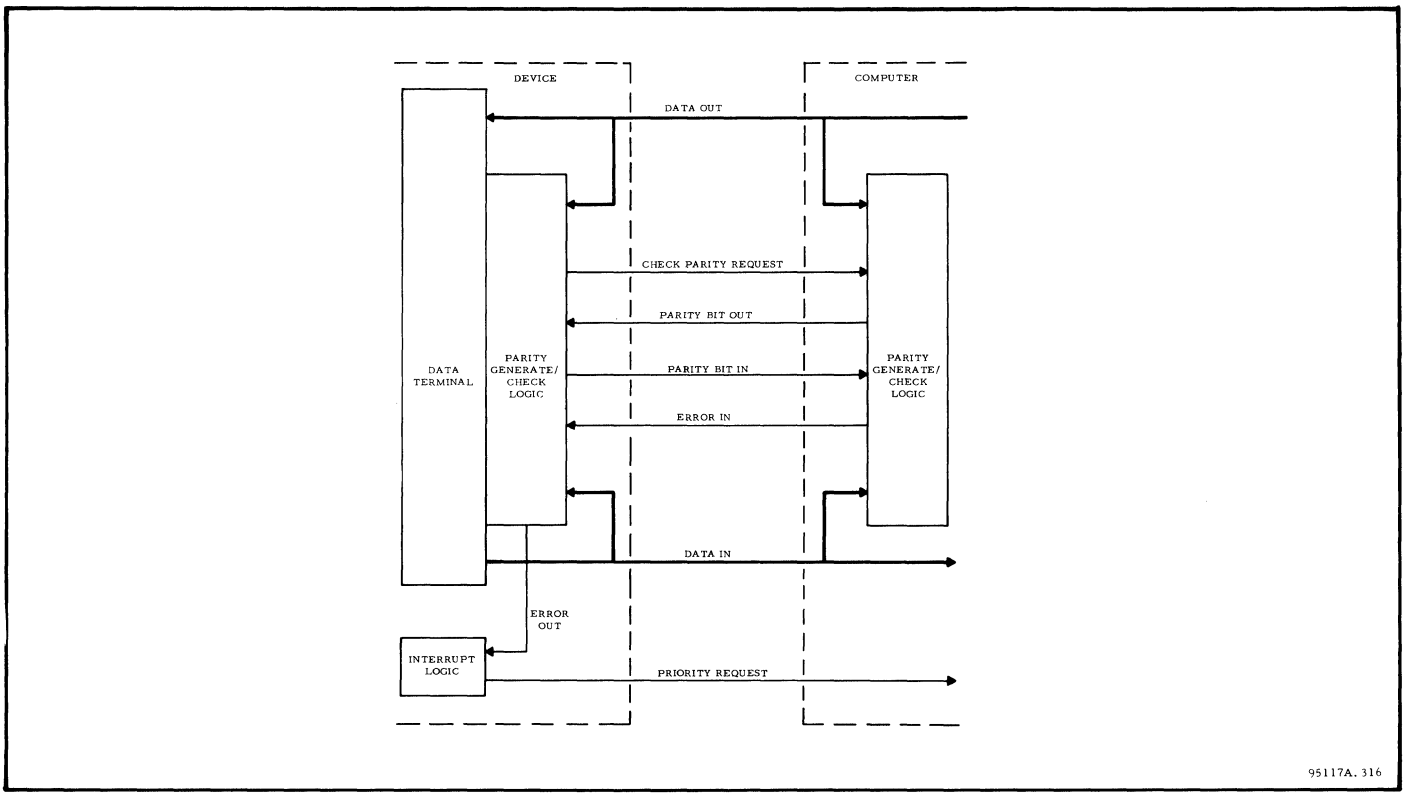


Figure 3-16. I/O Parity Block Diagram

d. Figure 3-20 illustrates the computer executing an output instruction with worst signal delay case existing.

DELAY FORMULA EXPRESSION

The four cases of 10-foot cable length mentioned may be expanded to any other selected cable length. The associated cable signal timing delay may be derived by the following formula:

$$\text{SIGNAL DELAY} = VD \left(\frac{L - 10}{10}\right)$$

where:

L = is the cable length desired

VD = is the value of delay of the specified signal as shown in the 10-foot timing charts. The 10-foot timing charts delays are in reference to the mainframe timing charts.

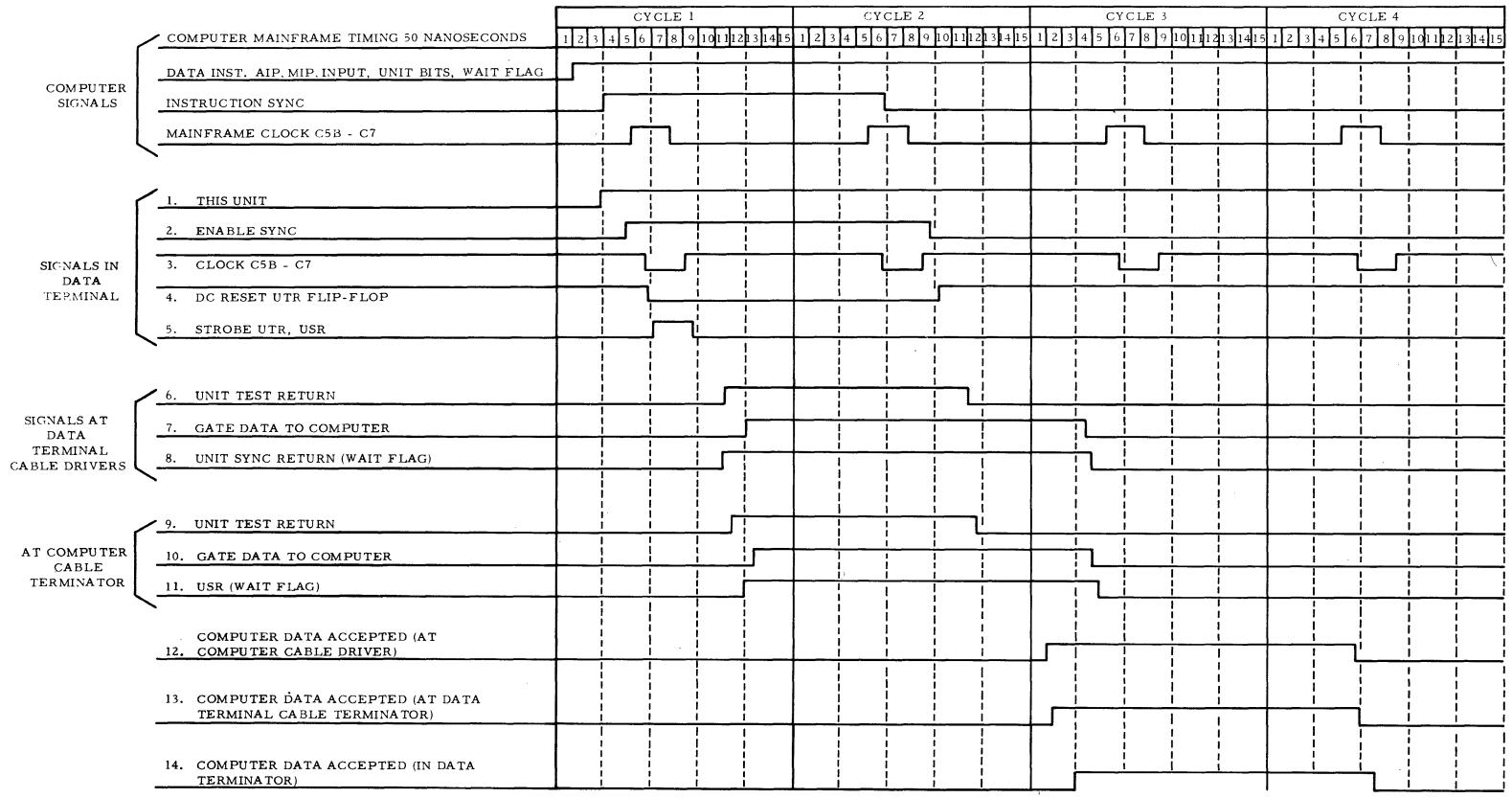


Figure 3-17. Input Data Instruction Timing Diagram (Best Case 10-Foot Cable)

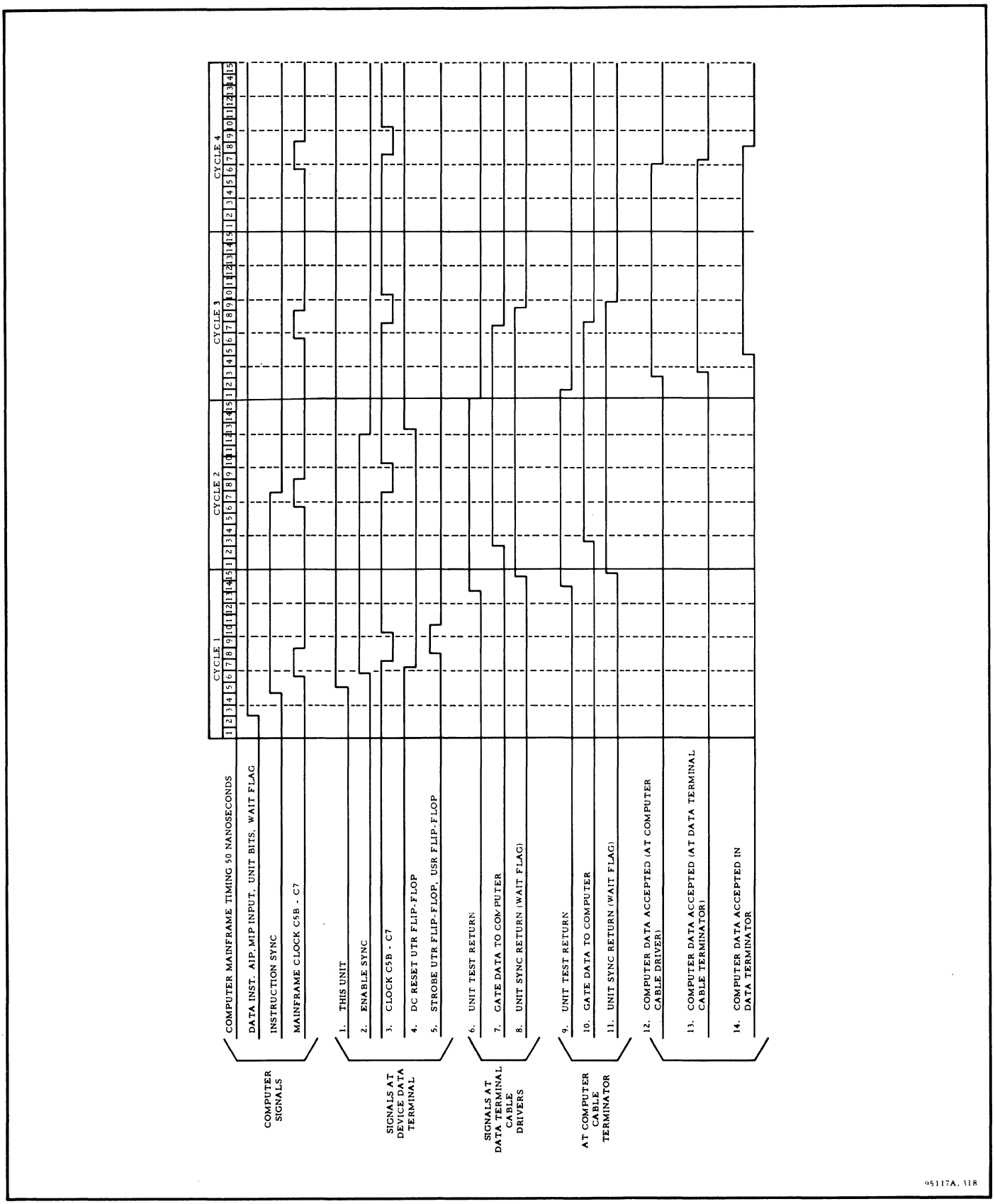


Figure 3-18. Input Data Instruction Timing Diagram (Worst Case 10-Foot Cable)

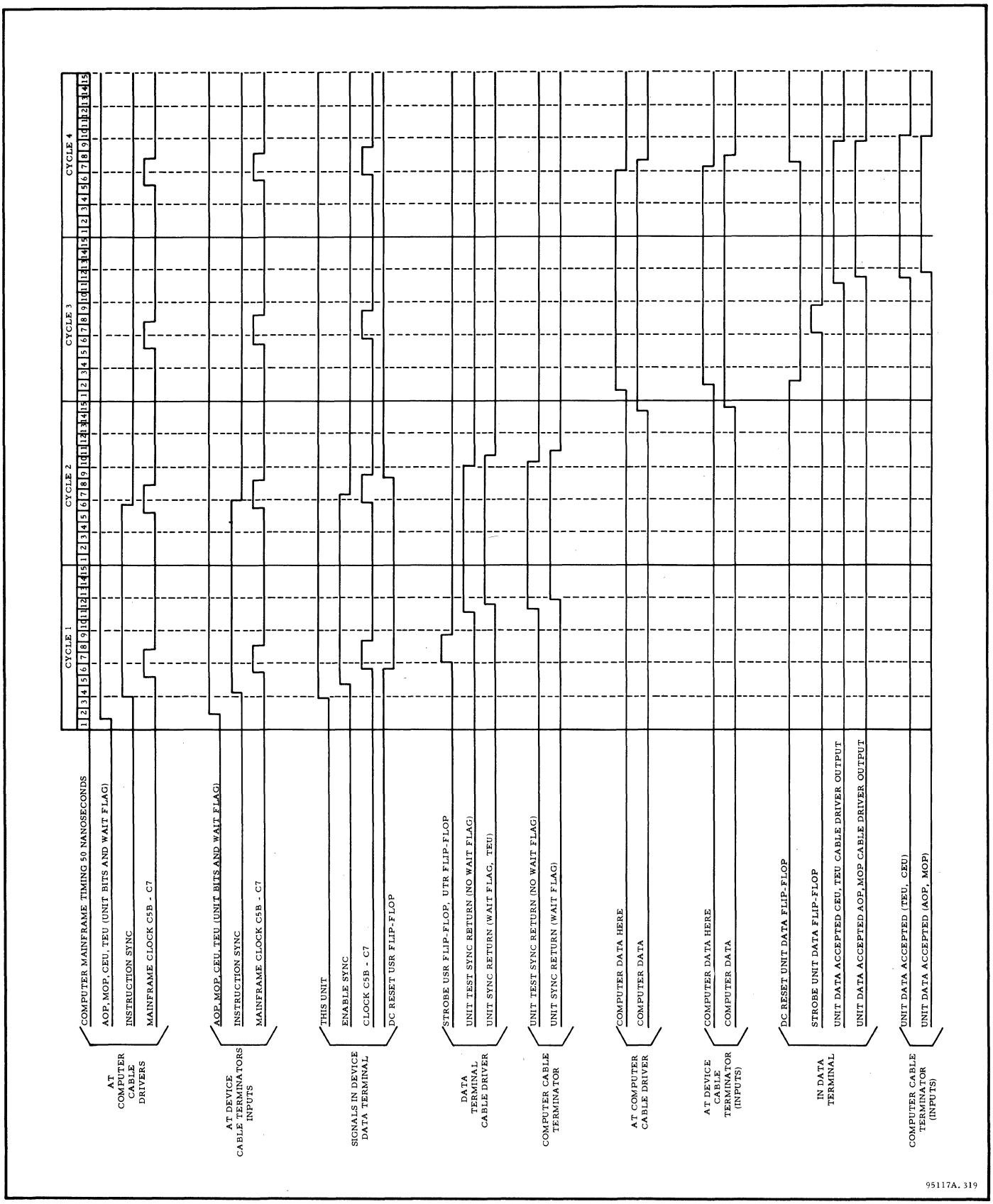


Figure 3-19. Output Data Instruction Timing Diagram (Best Case 10-Foot Cable)

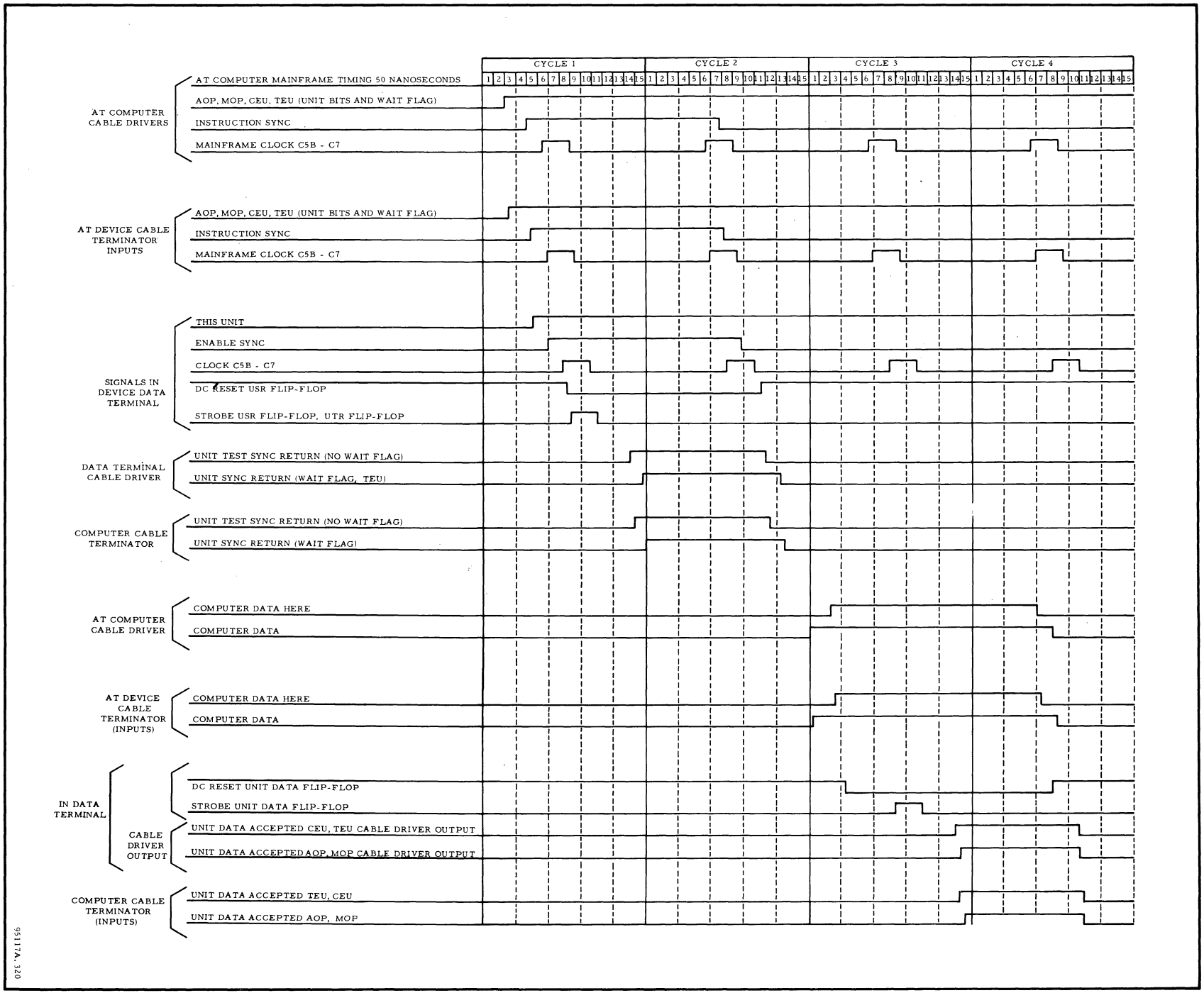


Figure 3-20. Output Data Instruction Timing Diagram (Worst Case 10-Foot Cable)

SECTION IV
BLOCK TRANSFER CONTROL AND PRIORITY INTERRUPT

BLOCK TRANSFER CONTROL

GENERAL

The Block Transfer Control Unit (BTC) is an optional computer input/output control unit which enables fully-buffered transfer of data between peripheral devices and computer memory. Figure 4-1 illustrates the data and control paths involved. The figure shows two peripheral devices connected to one BTC and a third peripheral device connected to a second BTC. These devices, as previously described, may communicate with the computer through execution of any of the I/O instructions. In addition, they may transfer data under BTC control, rather than under single-word program control. Although five devices are illustrated, a maximum of 64 devices, or device controllers, may be connected to the computer I/OP.

The salient features of this unit are listed below:

- Bits per Transfer — Full Computer Word
- Maximum Words per Block — 32,768
- Maximum Transfer Rate — 1,330,000 words per second
- Memory Cycles Stolen per Transfer — one
- Block Transfer Reinitialization — Automatic
- Maximum Number of BTC's per Computer (optional) — Eight
- Maximum Number of CGP's per computer (Optional) — Six
- Maximum Number of Peripheral Units per BTC — 16

OPERATION

A block diagram of the Block Transfer Control Unit and Block Priority Control is shown in figure 4-2. The BTC contains two binary counters plus transfer initialization and synchronization logic. One of the counters stores the first word address (FWA) and the second stores the word count (WC). FWA defines the storage location for each word transferred to/from memory, and WC defines the number of words to be transferred. The initial values for FWA and WC are obtained from two fixed locations in computer memory by the BTC each time a new block transfer is initiated (See table 4-1 for memory location assignments). Each time a word is transferred between memory and the selected peripheral device FWA is incremented and WC is decremented. The block transfer is completed, and an interrupt is generated when WC = 0. After a block transfer is completed, the BTC automatically initiates a new block transfer by obtaining a new initial set of FWA (CWA) and WC values from the dedicated memory locations. The block transfer sequence is ended by placing a terminate code in the WC word. The terminate code is a ONE in bit 0 (sign bit).

Table 4-1. BTC Memory Assignments

Octal Memory Location	BTC Assignment
1060	BTC 1 First Word Address (FWA)
1061	BTC 1 Word Count (WC)
1062	BTC 2 FWA
1063	BTC 2 WC
1064	BTC 3 FWA
1065	BTC 3 WC
1066	BTC 4 FWA
1067	BTC 4 WC
1070	BTC 5 FWA
1071	BTC 5 WC
1072	BTC 6 FWA
1073	BTC 6 WC
1074	BTC 7 FWA
1075	BTC 7 WC
1076	BTC 8 FWA
1077	BTC 8 WC

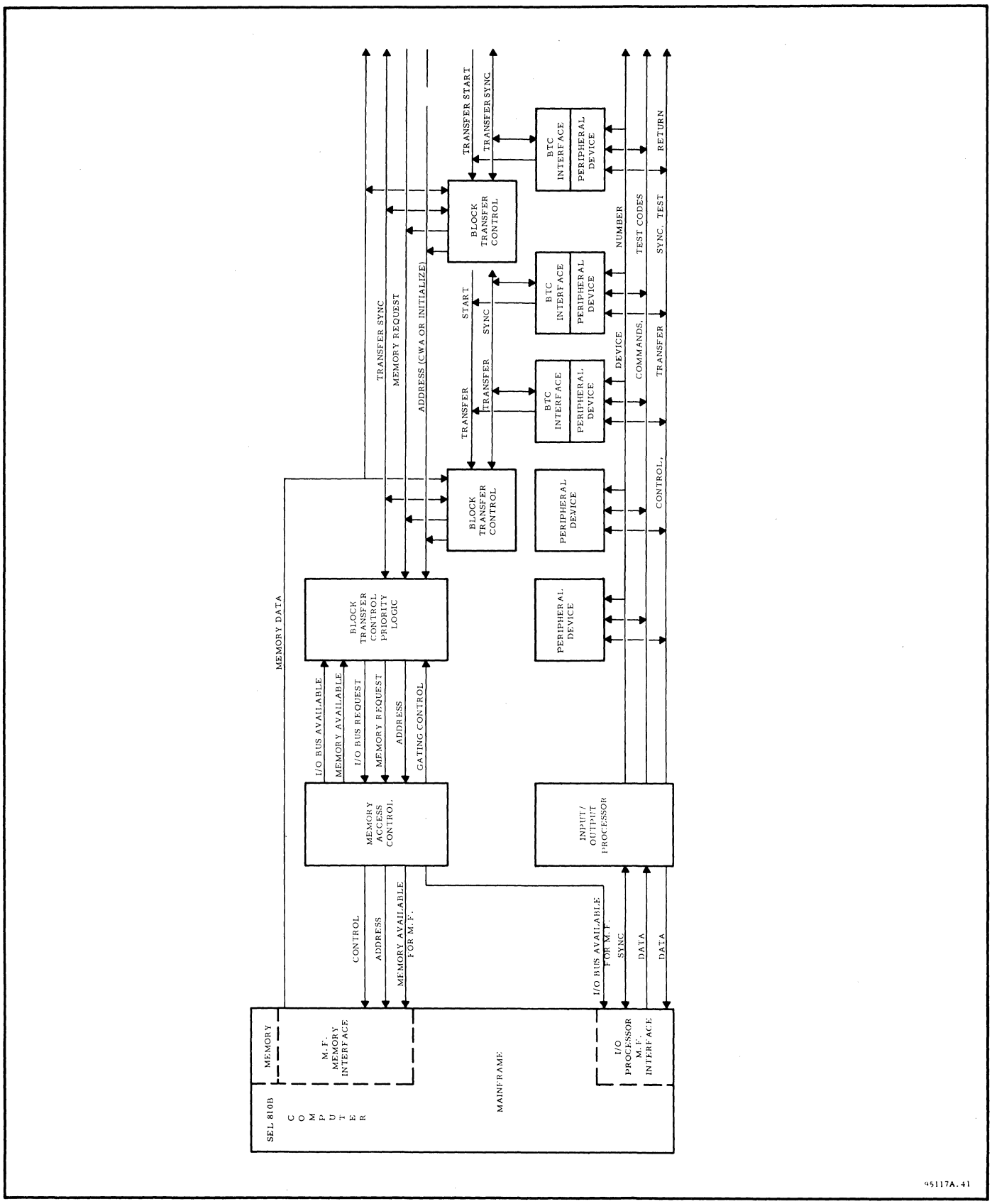


Figure 4-1. Input/Output Configuration and Computer Interface

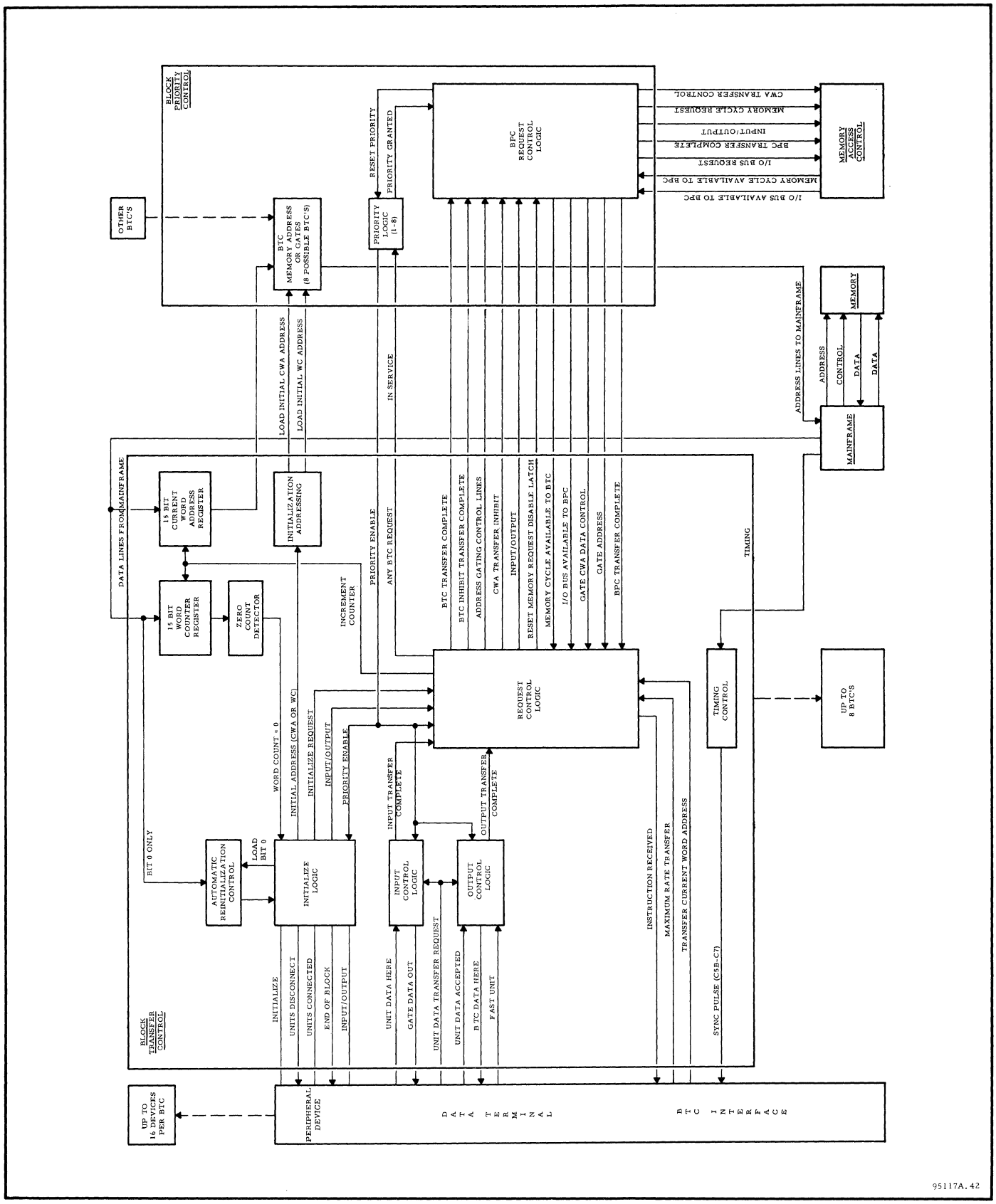


Figure 4-2. Block Transfer Control and Priority Block Diagram

NOTE

The initial value of the Current Word Address (CWA) is identified as the First Word Address (FWA). This allows the BTC starting address to be defined and distinguished from any other (CWA).

The CWA value may be transferred from the BTC to the dedicated memory location by the execution of a CEU command to the device currently operating with the BTC by inserting a ONE in bit 13 of the command code. The presence of this bit causes the device logic to generate the signal Transfer Current Word Address to the BTC. If the BTC currently has a memory cycle request in to the Block Priority Control, the transfer CWA command is stored by the BTC and the CEU instruction is allowed to terminate in order not to tie up the computer. Immediately following the memory cycle for the BTC, the CWA is transferred to the dedicated memory location.

INITIALIZATION AND DATA FLOW

The Block Transfer Control (BTC) is initialized through the peripheral device to/from which the data block transfer is made. Execution of the proper Command External Unit (CEU) instruction causes the device specified by the instruction to send an Initialize signal to the BTC to which it is cabled. In many peripheral devices, this instruction also causes the device to initiate action to transmit/accept data.

The flow chart shown on figure 4-3 illustrates a diagrammatical sequence of the initialization process, while figure 4-4 shows the BTC timing diagram for a ten-foot I/O bus cable (worst case).

The basic BTC signal flow is sequenced in the following manner:

a. The command function bits sent to the peripheral device cause that device to raise the signal Initialize to the BTC.

b. The function bits also enable the device to raise the Input/Output line depending on direction of data transfer.

c. The signal Units Disconnected is sent to all devices causing them to disconnect from the BTC.

d. The BTC recognizes that the device is disconnected by the signal Unit Connected from the device.

e. The BTC then removes the signal Units Disconnected. The device logic generates the signal Unit Data Accepted indicating the command second word has been processed; and sends this signal to the mainframe control unit which terminates the I/O instruction and removes the function bits.

f. The device recognizes the removal of the function bits and generates the signal Unit Connected to the BTC.

g. The BTC then receives the signal Data Transfer Request from the device logic.

h. The device receives either one of two signals along with a data word transfer; BTC Data Here To Unit if the device is inputting data; Gate Data Out if the device is outputting data.

i. The device sends either one of two signals with each data word transfer. Unit Data Here is sent to the BTC if the device is inputting data; and Unit Data Accepted is sent to the BTC if the device is outputting data.

j. When the value of the Word Count is decremented to zero and termination reached, the BTC sends out the signals End of Block and Units Disconnect.

The signal Fast Unit is utilized by some high-speed peripheral devices during output transfer from the computer. This signal is generated in addition to those discussed previously, and is sent to the computer in parallel with the Unit Connected signal. The signal Fast Unit enables the computer to output data to the device while ignoring the removal of the device response signals, Data Transfer Request and Unit Data Accepted. This signal also permits the BTC to proceed and generates the data transfer complete sequence permitting termination of data transfer and reinitialization of the control unit for future transfer requests.

Figure 4-5 illustrates a BTC Data Input transfer flow timing chart for a ten-foot I/O Bus Cable (worst case). Figure 4-6 shows a BTC Data Output transfer flow timing chart for a ten-foot I/O Bus Cable (worst case).

When the BTC receives the Initialize signal from the device, it requests a memory cycle through the Block Priority Control (BPC). The BTC also generates the assigned memory address of the First Word Address (FWA). When the memory cycle is granted, the FWA value is transferred from the memory to the CWA (FWA) Register in the BTC. It should be noted that initial value of the (CWA) is identified as the (FWA). This allows the BTC

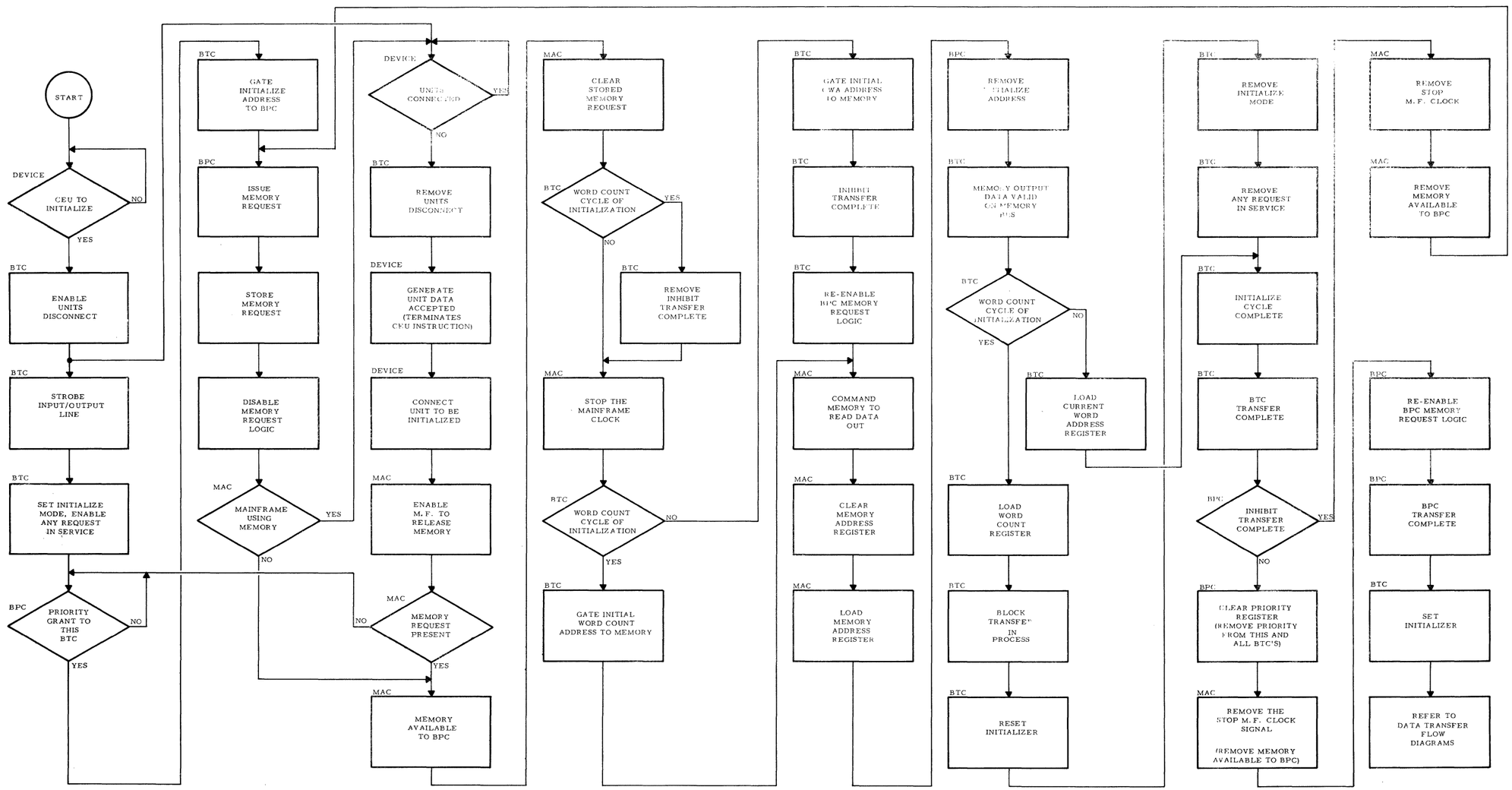


Figure 4-3. Block Transfer Control Initialization Flow Chart

95117A.43

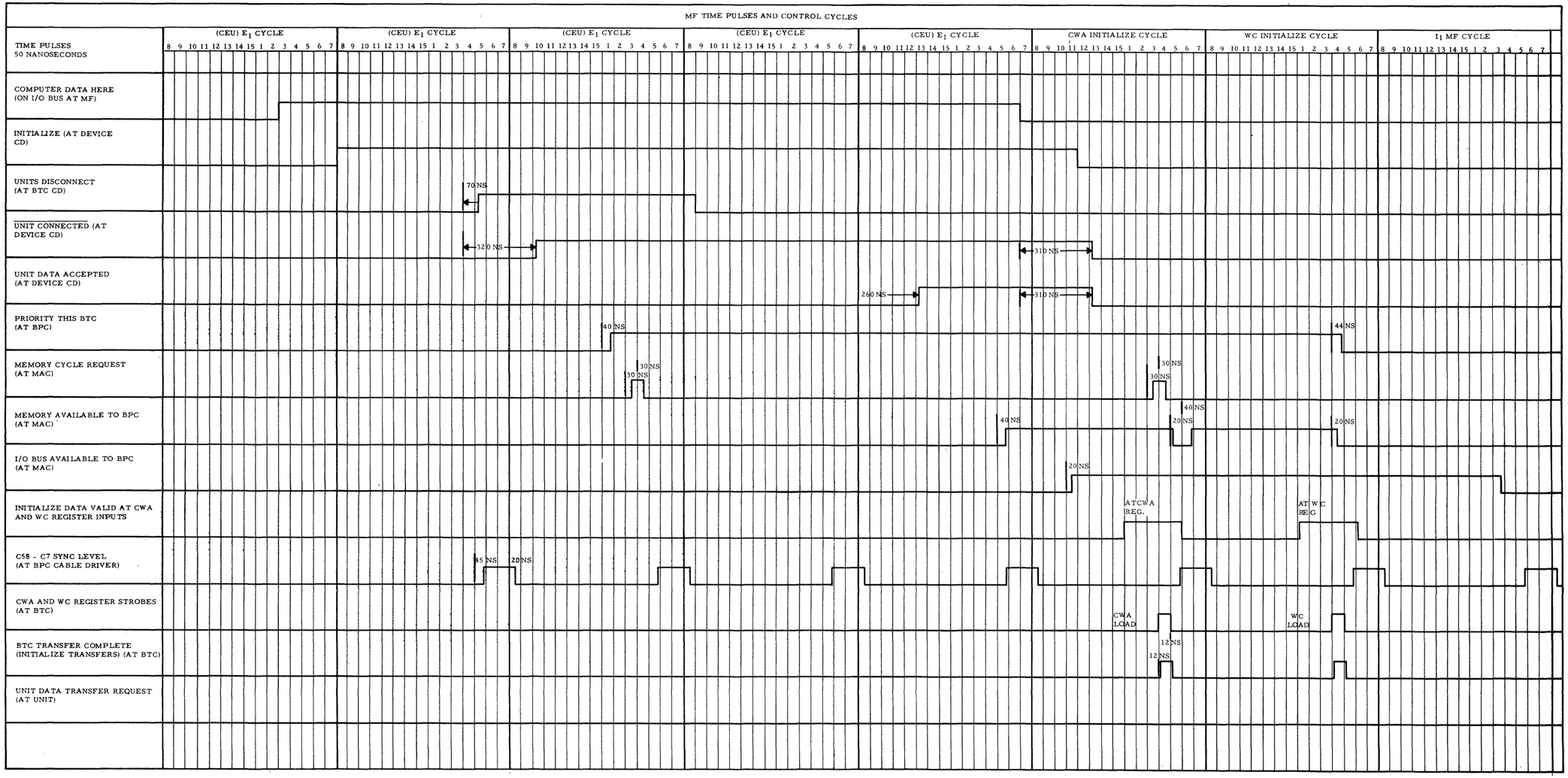


Figure 4-4. BTC Initialization Timing I/O Bus Cable (Ten Foot Worst-Case)

95117A.44

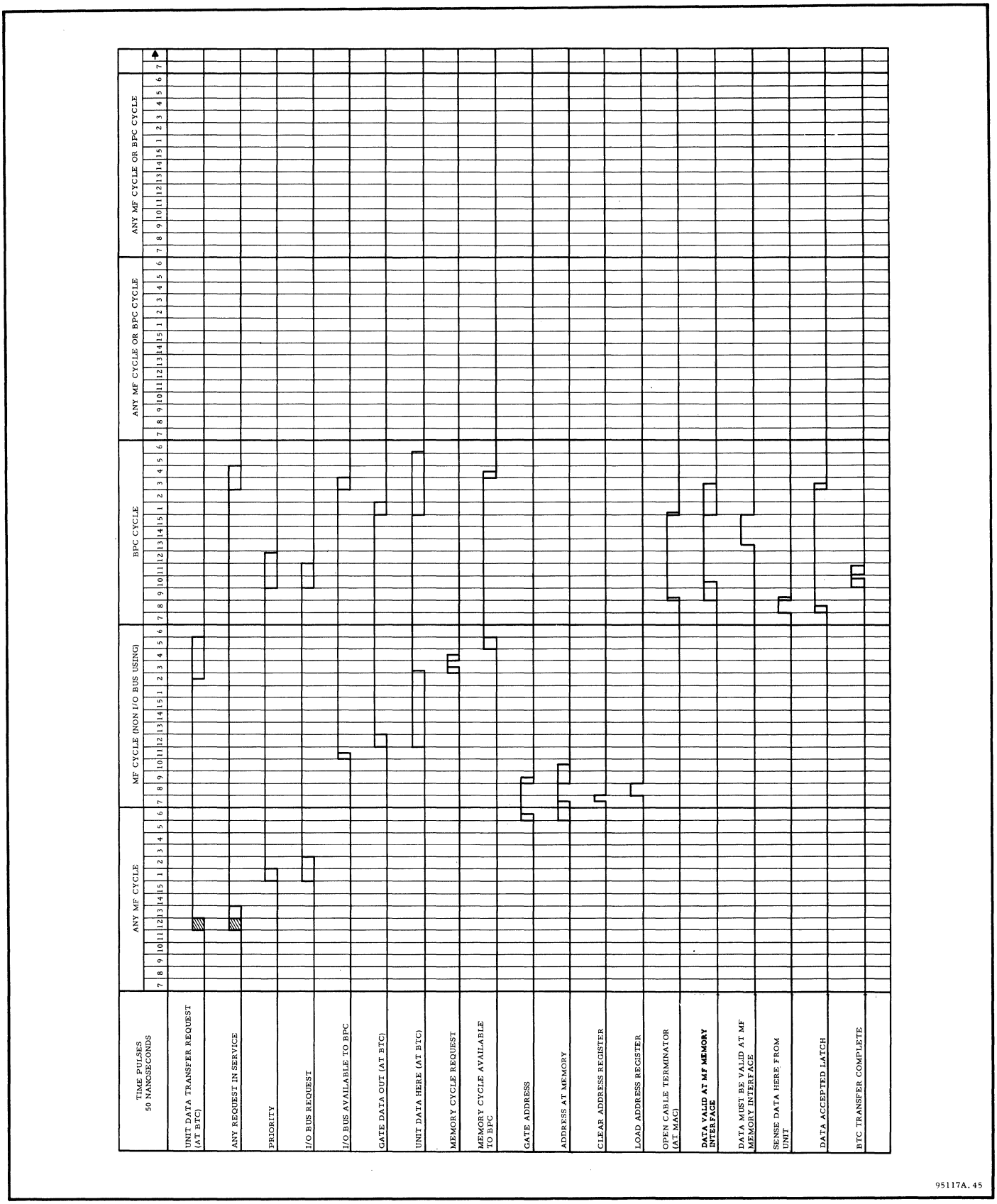


Figure 4-5. BTC Data Input Transfer Flow Timing Chart (ten foot I/O Cable)

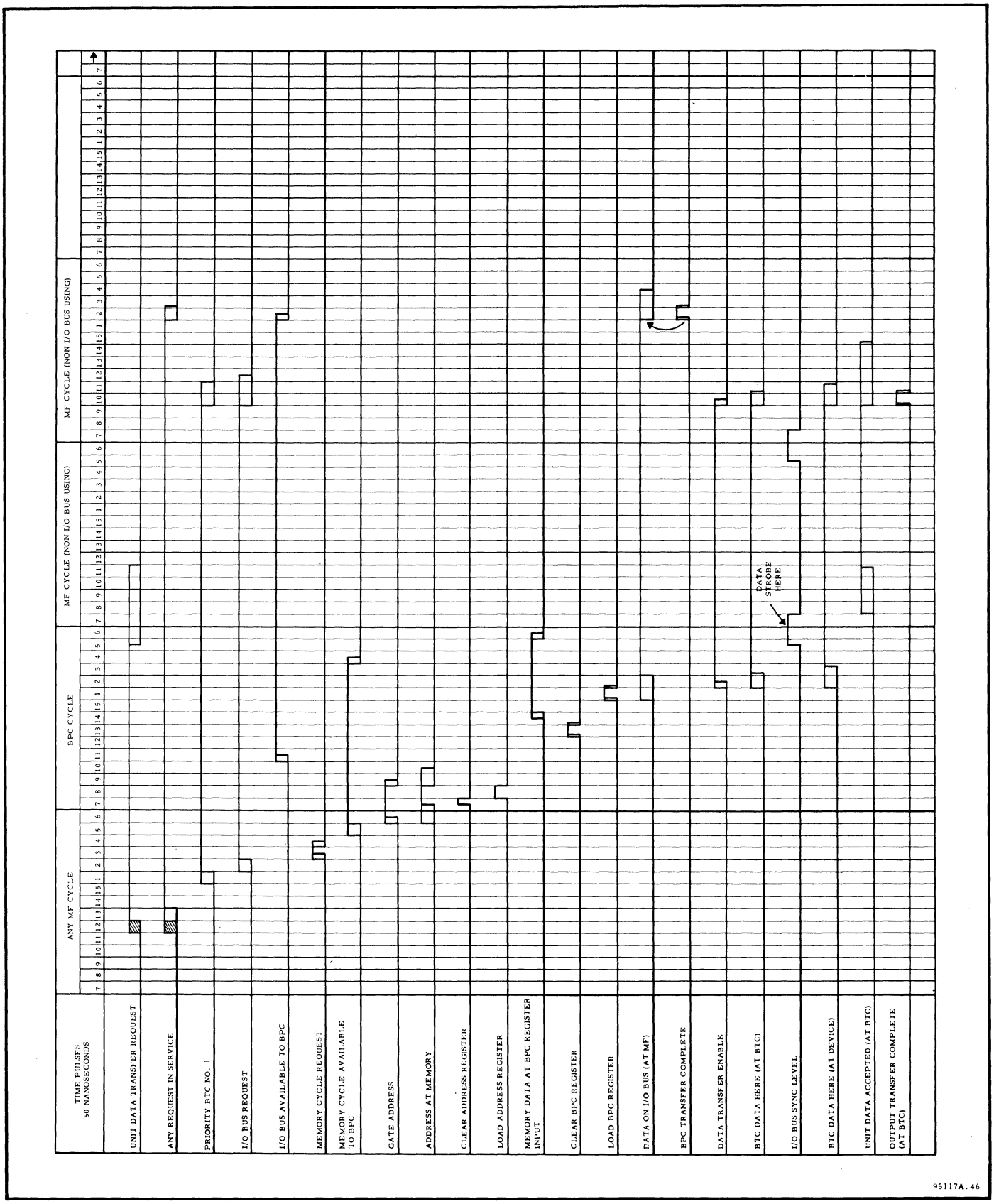


Figure 4-6. BTC Data Output Transfer Flow Timing Chart (Ten-Foot I/O Cable)

starting address to be defined and distinguished from any other (CWA).

A request for a second cycle is then made by the BTC and the address of the memory location containing WC is placed on the address lines by the BTC. When the second cycle occurs, WC is transferred from memory to the WC Register in the BTC. The terminate bit (bit 0) contained in the WC word is also tested and a latch is set if the terminate bit is a ONE, which signifies that no more block transfers are to be made after completion of the one being initialized. The maximum time for the entire initialization is six cycles for the CEU execution, plus two cycles for the CWA and WC transfers, which occur immediately following CEU execution.

After BTC initialization, words are transferred between the selected peripheral device and memory over the Unit I/O Bus under the joint control of the BTC, the Block Priority Control (BPC), and the Memory Access Control (MAC). All word transfers are initiated by the device which sends a Data Transfer Request to the BTC. The Data Transfer Request causes the BTC to request a memory cycle through the BPC to the MAC. When it is determined that a memory cycle can be granted, a Memory Available signal is sent to the BTC. The BTC, in turn, sends a signal to the peripheral device which causes it to connect to the Unit I/O data lines, execute the data transfer, and then disconnect from the data lines. During the data word transfer, the CWA value is incremented and the WC value is decremented in the BTC. All words are transferred by repetition of this cycle, which is always initiated by the peripheral device. The Block Transfer Control operations described in this section are illustrated by the flowchart on figure 4-7.

When the value of WC is decremented to zero, the block transfer is terminated. If the terminate latch in the BTC has not been set by the terminate bit in the last WC word acquired from memory, a new block transfer is automatically initiated by the BTC. Reinitialization consists of acquiring new CWA and WD values from the memory locations assigned to the BTC. After reinitialization, an interrupt is generated which signifies that the transfer of the last block is completed and a new block transfer is initialized. The interrupt processing routine can then store in the dedicated locations the CWA and WC values for the next block transfer anytime prior to the completion of the current block transfer. This reinitialization technique reduces the problem of reinitializating block transfer under program control between the times of occurrence of two successive words in a continuous data stream.

If the terminate latch in the BTC has been set by the terminate bit in the last WC word acquired from memory, an interrupt is generated when the value of WC is decremented to zero and no new transfer is initialized by the BTC. In addition, the Data Transfer Request line from the peripheral device is disconnected until a new Initialize signal is received. Hence, the BTC disconnects from the peripheral device.

BLOCK PRIORITY CONTROL

GENERAL

The Block Priority Control (BPC) processes memory cycle requests from the individual BTC's and allocates them on a priority basis. Each BTC is assigned a particular priority. The BPC priority logic is similar to that of the priority interrupt logic in the computer, ensuring that higher priority BTC's are always serviced before a lower priority BTC. All BTC's may request a memory cycle simultaneously but the highest priority BTC will be granted the next memory cycle. Figure 4-7 illustrates the flow chart that includes the Memory Access Control (MAC) sequence of processing BPC requests. The MAC block diagram on figure 4-8 illustrates the logic arrangement.

The BPC operates in conjunction with the mainframe MAC logic. The BPC has priority over the mainframe requests so that the mainframe is locked out during the time a word transfer involving a BTC takes place. There are certain instructions and certain cycles of other instructions where the BPC is inhibited from stealing cycles from the computer. All of the I/O instructions (MIP, MOP, AIP, AOP, TEU, CEU) inhibit the BPC. The last cycle of the DIV instruction and the E_1 cycle of IMS also inhibits the BPC from having access to memory.

A succession of I/O instructions, however, does not inhibit cycle stealing by the BTC's since the BPC may request memory in between any sequence of instructions. The maximum time that the highest priority BTC would have to wait for a memory cycle would depend on an I/O instruction. All I/O instructions require a minimum of four cycles execute time between the computer and the peripheral device; but depending on the I/O cable length, this time may be increased. The Wait flag also causes the cycle to be extended.

The maximum collective transfer rate for a BTC (or group of BTC's) is 1,330,000 words per second. The use of the Maximum Rate Transfer signal from the peripheral device to the BTC is required in order to attain a transfer rate of 1.33 MHz for any one BTC; otherwise, a rate of up to 443 kHz is realized. The maximum rate transfer line is

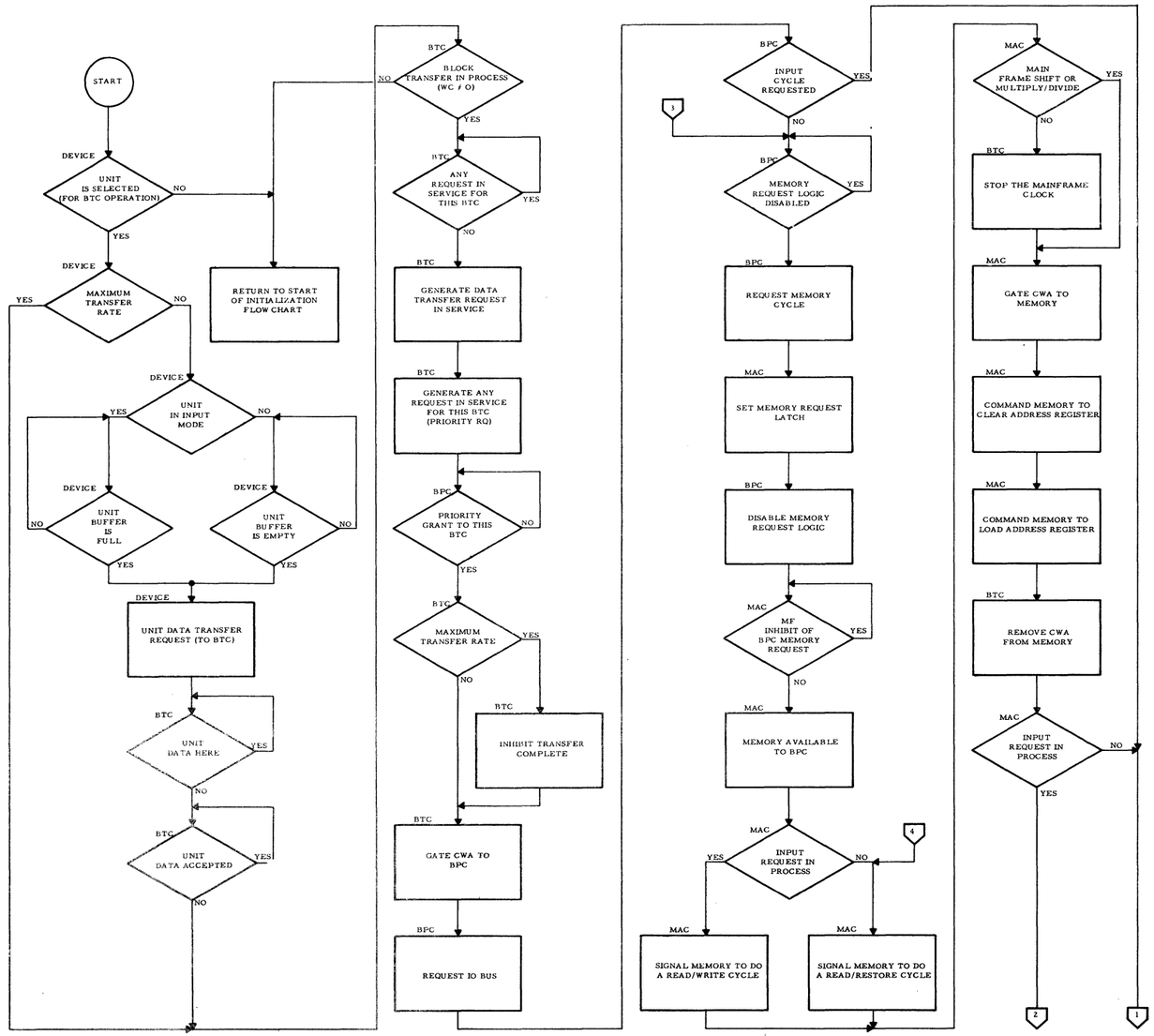


Figure 4-7. Block Transfer Control Data Transfer Flow Chart, Sheet 1 of 3

95117A.47-1

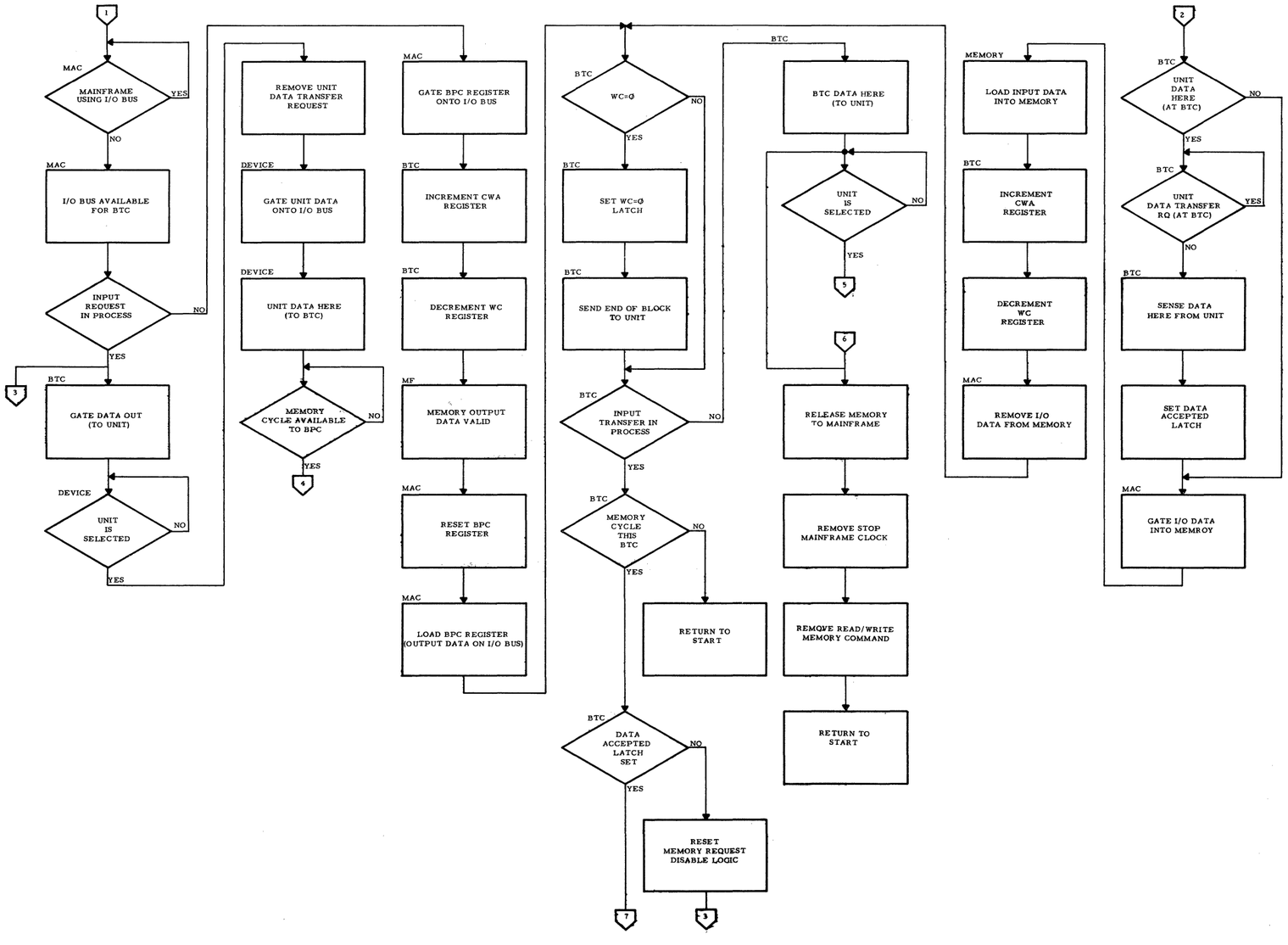


Figure 4-7. Block Transfer Control Data Transfer Flow Chart, Sheet 2 of 3

95117A.47-2

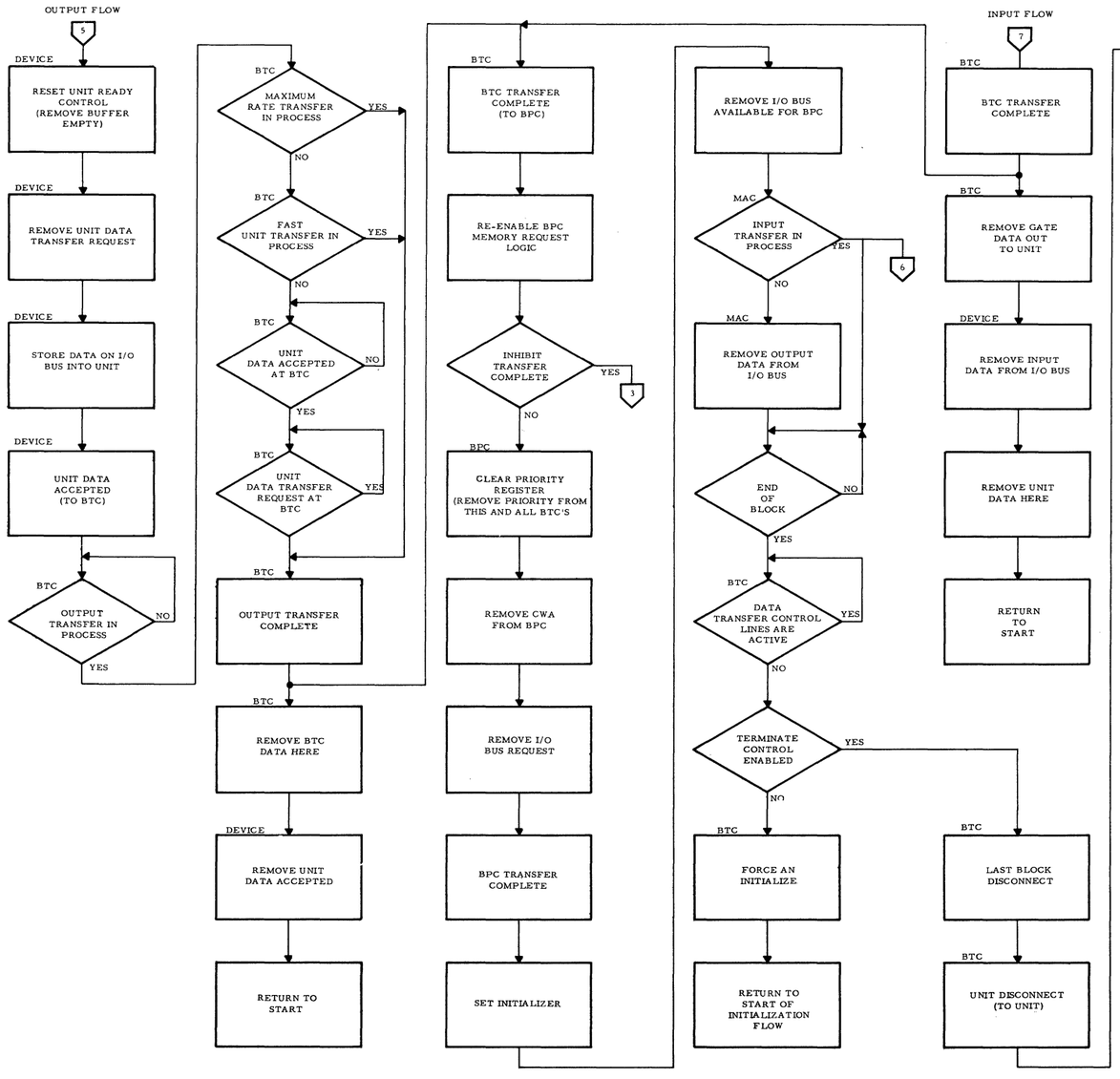


Figure 4-7. Block Transfer Control Data Transfer Flow Chart, Sheet 3 of 3

95117A.47-3

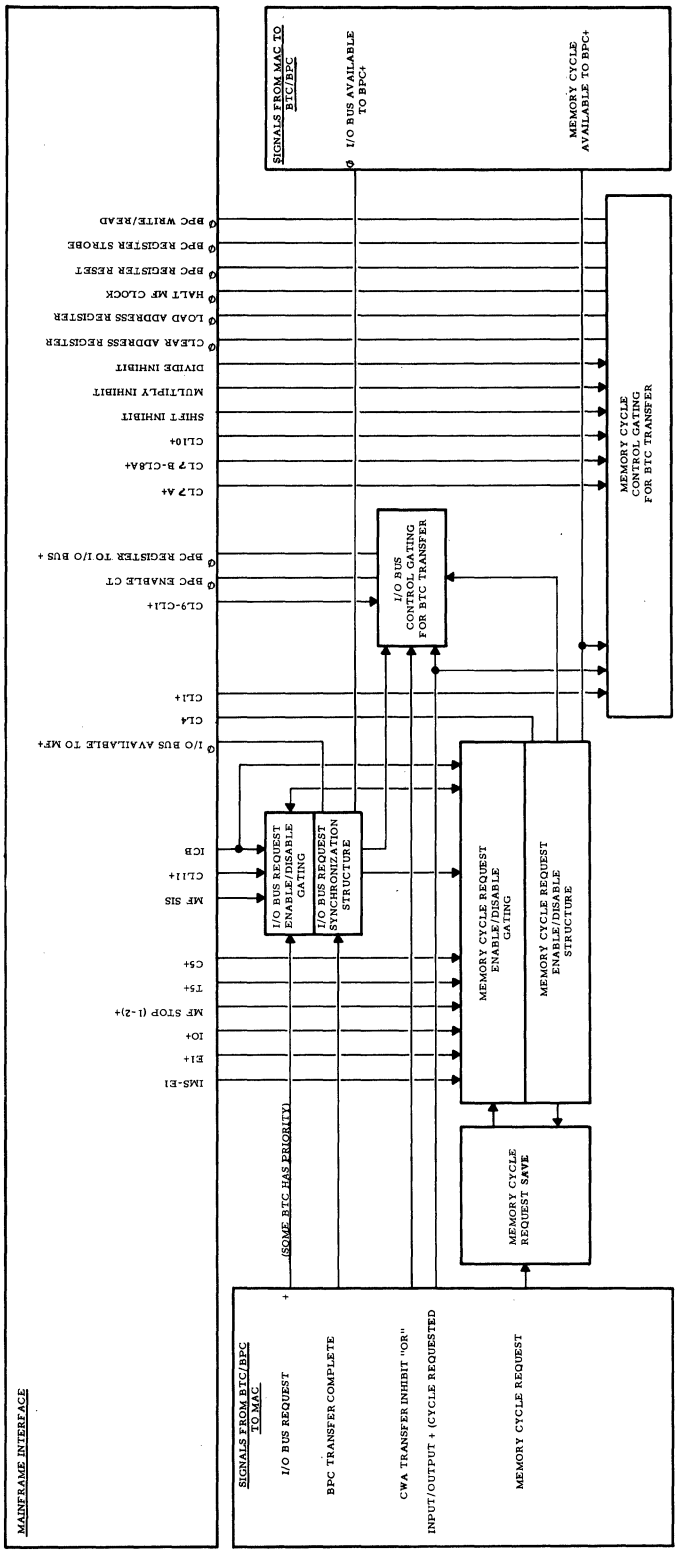


Figure 4-8. Memory Access Control Block Diagram

controlled by the peripheral device. This signal must be enabled during the first Data Transfer Request to the BTC, and remain enabled as long as the device desires to continue at a maximum rate.

Upon completion of the block, the device must disable this control line in case another device operating on the same BTC should desire to connect for data transfer.

Enabling the Maximum Rate Transfer signal from the peripheral device during a computer output mode permits the BTC to transfer data at the rate of one word per one memory cycle. The computer ignores the removal of the device signals Data Transfer Request and Unit Data Accepted, decrementing the word count and incrementing the current word address.

When the computer is performing an Input Data Transfer mode, a single word transfer per memory cycle is executed. Depending upon the peripheral device response time and I/O cable length delay, a single word transfer may require one or more computer memory cycles. The computer checks for the Unit Data Here signal accompaning a word transfer during each memory cycle. Should this latter signal not appear, the BTC requests a second cycle, freezing the Word Count (WC) and Current Word Address (CWA) during the first cycle.

When the Unit Data Here signal is received, the WC is decremented and CWA is incremented. Coincidentally, the computer mainframe is inhibited from ever obtaining a cycle as long as a BTC is operating at this maximum rate transfer. Another point of importance is that the BTC must be set up to terminate at the finish of the particular block; otherwise, the computer is locked out since there is no way for the program to modify the dedicated memory location containing the block length.

A problem may arise when operating at the maximum rate transfer. If the device through some failure or programming error does not input the required number of words to the BTC, the computer remains hungup since the BPC is requesting memory, awaiting the last data word to be transferred. These latter possibilities are all failure conditions or programming errors.

COMPUTER GRAPHICS PROCESSOR

The optional Computer Graphics Processor (CGP) Model 84-235MP, is a high-speed data transferring control unit designed to satisfy the specialized needs of the Systems Engineering Laboratories Computer Graphics Systems. The control unit is similar to the SEL 810B optional BTC with the exception of its specialized operating characteristics and added control functions. A BTC, when outputting data, is unmindful of the nature of this data. However, the CGP examines each word as it comes from memory and either interprets the word as data and sends it to the Computer Graphics System or as an instruction and takes appropriate action.

The instructions allow the CGP to operate on its address counters, thereby freeing the SEL 810B Computer from much of the control unit servicing, and allowing it more time to operate on the buffer areas of the system. This feature allows the use of subroutines to generate frequently used patterns.

The CGP is used in conjunction with the SEL 816A Computer Graphics System to provide the most efficient method of transferring data from the SEL 810B Computer to the display unit. Using the CGP minimizes the amount of computer memory and transfer time required to support the display. The CGP also provides a high degree of flexibility in display format generation since the CGP contains the capability of executing the following instructions: (1) Branch Unconditionally, (2) Store, Place and Branch, and (3) Stop. The first two instructions, which have the same execution capabilities as the corresponding computer instructions, enable the contents of non-contigous memory areas to be transferred automatically to the display. This capability enables display programs to be organized to provide maximum usage of closed subroutines that are stored in memory a single time and used as often as required in a given display format. The stop command enables the display unit to automatically control the refresh rate, and maintain a fixed rate regardless of the amount of data being displayed.

The operating characteristics and interface specifications of the CGP are the same as those for an SEL 810B - BTC with the following exceptions and specifications:

a. One CGP replaces one BTC. Block Transfer Control Units and Computer Graphics Processors are installed in desending order of priority with BTC number one having the highest priority, and BTC or CGP number eight having the lowest. When BTC's and CGP's are mixed, all BTC's have the highest priority and the CGP's the lowest. If only CGP's are used, the CGP assigned priority number three has the highest priority (total of six).

b. Block transfer length is controlled by instruction words and not a word count (WC) register. Therefore, a CGP has no WC register and only requires one cycle of initialization (for the current word address).

c. The CGP transfers data (or instructions) from the SEL 810B memory out to the CGP or CGS or both; no data input transfers are ever performed.

d. The CGP has no automatic reinitialization control.

e. The CGP operates on a fast unit basis only, and therefore does not use a maximum transfer rate facility.

f. The CGP examines the four most significant bits (bit 0 - bit 3) of all words being transferred to the CGS. If a 0011 code is detected in bits 0 through 3 respectively, the word being transferred is interpreted as an instruction and further analysis of the instruction word is commanded. This analysis entails examination of the instruction words three least significant bits (bit 13 - bit 15), and will determine what operation is instructed. The types of instructions and bit coding is shown in table 4-2.

g. Instruction word execution procedure for codes listed in table 4-2 are as follows:

1. Stop - The Stop command is a single-word instruction which is used to terminate a block transfer. The Stop instruction is transmitted to the CGS (as well as the CGP) where it is decoded as such and initiates a priority interrupt request. Upon execution of the Stop, further CGS requests for data transfers are ignored and the transfer CWA request is the only transfer that can be performed until the CGP is reinitialized by the program.

2. BRU - The BRU instruction is a two-word instruction (more if indirect addressing is used) which is used to unconditionally branch to a specified location in memory. The BRU instruction word is transmitted to the CGS (as well as the CGP) where it is optionally used as a control word to govern further CGS operation. The CGP handles this control word by executing the following sequence:

(a) Detect and decode BRU instruction.

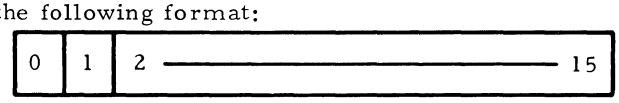
(b) Complete data transfer to CGS

(c) Request transfer of BRU address (or indirect address) from the memory location specified by the CWA register. This address data is not transmitted to the CGS nor is any future address data if indirect addressing is used.

(d) The actual BRU address is loaded into the CGP's CWA register when it becomes available. The CWA (obtained from memory) is not incremented during this cycle.

(e) The transfer completes and control is returned to the CGS. It is important to note that the CGS operates asynchroniously to the CGP once an instruction word starts execution; however, a request for data transfer or CWA transfer is ignored by the CGP while it is in process of servicing an instruction word.

(f) The second word (address word or indirect address word) of the BRU instruction uses the following format:



Bit 0	Bit 1	Bit 2 - Bit 15
MSB of address or indirect address	Indirect bit 0 = actual addr. 1 = indirect addr.	Second MSB through the LSB of the address or indirect address

3. SPB - The SPB instruction is a two-word instruction (more if indirect addressing is used) which is used to unconditionally branch to a specified location in memory while retaining the address of the next consecutive word in memory following the address word of the SPB instruction. The SPB instruction word is transmitted to the CGS (as well as the CGP) where it is optionally used as a control word to govern further CGS operation.

Table 4-2. Instruction and Bit Coding

Instruction Word	Indicated Action	Words per Instruction
0 0 1 1 x x x x x x x x x 0 0 1	Branch Unconditionally (BRU)	2
0 0 1 1 x x x x x x x x x 0 1 0	Store, Place, and Branch (SPB)	2
0 0 1 1 x x x x x x x x x 1 0 0	Stop the Block Transfer (Stop)	1

Note: When the 0011 instruction code appears in a word, one and only one of the command bits (bit 13 - bit 15) must be asserted (a 1).

The CGP handles this control word by executing the following sequence:

(a) Detect and decode SPB instruction.

(b) Complete data transfer to CGS.

(c) Request transfer of SPB address (or indirect address) from the memory location specified by the CWA register. This address data is not transmitted to the CGS nor is any future address data if indirect addressing is used.

(d) The CWA register is incremented and stored in a special register (SPB) after it has been used to access the address word of the SPB instruction.

(e) The actual SPB address is loaded into the CGP's CWA register when it becomes available. This address (now contained in the CWA register) is not incremented during the cycle.

(f) The CGP now requests a Store, Place cycle which transfers the contents of the SPB register into the memory address specified by the CWA register.

(g) Complete Store Place cycle (including CWA register increment) and return control to the CGS.

(h) The rules concerning CGS asynchronious transfer requests during instruction word execution and address word formatting are the same as spelled out for the BRU instruction.

BLOCK TRANSFER CONTROL INTERFACE

The BTC interface logic is an intergral part of the data terminal used with a unit operating with a BTC. The units data terminal connects both to the standard I/O Bus and to the Block Transfer Control. Figure 4-9 illustrates the resistor/transistor logic (RTL) block diagram required to provide the BTC interface. Signal lines applied through cable terminators and cable drivers interface with the BTC through the BTC cable. All other signals are derived from either the data terminal or the unit control logic. The diagram is representative of a general purpose interface, and depending on the nature of the peripheral unit, could become much more complex.

The Data Transfer Request line is the primary signal to the BTC used to initiate data transfer. The line is enabled by one of two gates depending on the direction of transfer. When the BTC is operating in the input mode, the ready condition is determined by the fact that the input buffer register has been loaded. Conversely, when the BTC is operating in the output mode, the ready condition is determined by the fact that the output register has been unloaded and ready to receive another data word. In conjunction with the data transfer request line are the Unit Data Accepted and Unit Data Here lines to the BTC. The labels applied to these lines are self-explanatory.

PRIORITY INTERRUPT

GENERAL

The SEL 810B Computer can have up to 98 individual levels of priority interrupts. Ninety-six of these levels can be selectively enabled and disabled under program control. Two levels are supplied with the basic computer. Additional levels are available in groups of 12 each, except that the first group contains 10 optional levels.

Assignment of interrupts is highly flexible. Internal signals such as Overflow and Memory Parity can be connected to interrupt levels. BTC End of Block signals, and external signals from peripheral units and custom system components are connected to the levels which best fit the operation of each system.

Two special interrupts, Power Fail Safe/Restore and Stall Alarm, are optionally available. These levels, when present, are always enabled and carry the highest priorities. Interrupt signals at these levels override all other computer functions, including Halt, I/O Wait, and indirect address chaining.

A unique location in memory is assigned to each interrupt level. These locations are assigned in MAP 1 to keep the entire MAP 0 available for program usage. Location $1,002_8$ is assigned to the highest priority level under program control, location 1003_8 to the second highest level, etc. Table 4-2 shows the assignment of interrupt locations as well as BTC locations.

When an interrupt signal is recognized by the mainframe, a wired-in instruction SPB*L (Store, Place, and Branch Indirect) is executed, where L is the address of the memory location assigned to the interrupt level. By storing the starting location of the interrupt processing routine in L, a linkage is provided to any point in memory. Since the address of the instruction to have been executed in the interrupted program is stored in the interrupt routine entry point by the SPB instruction, a means for returning to the point of interrupt is provided.

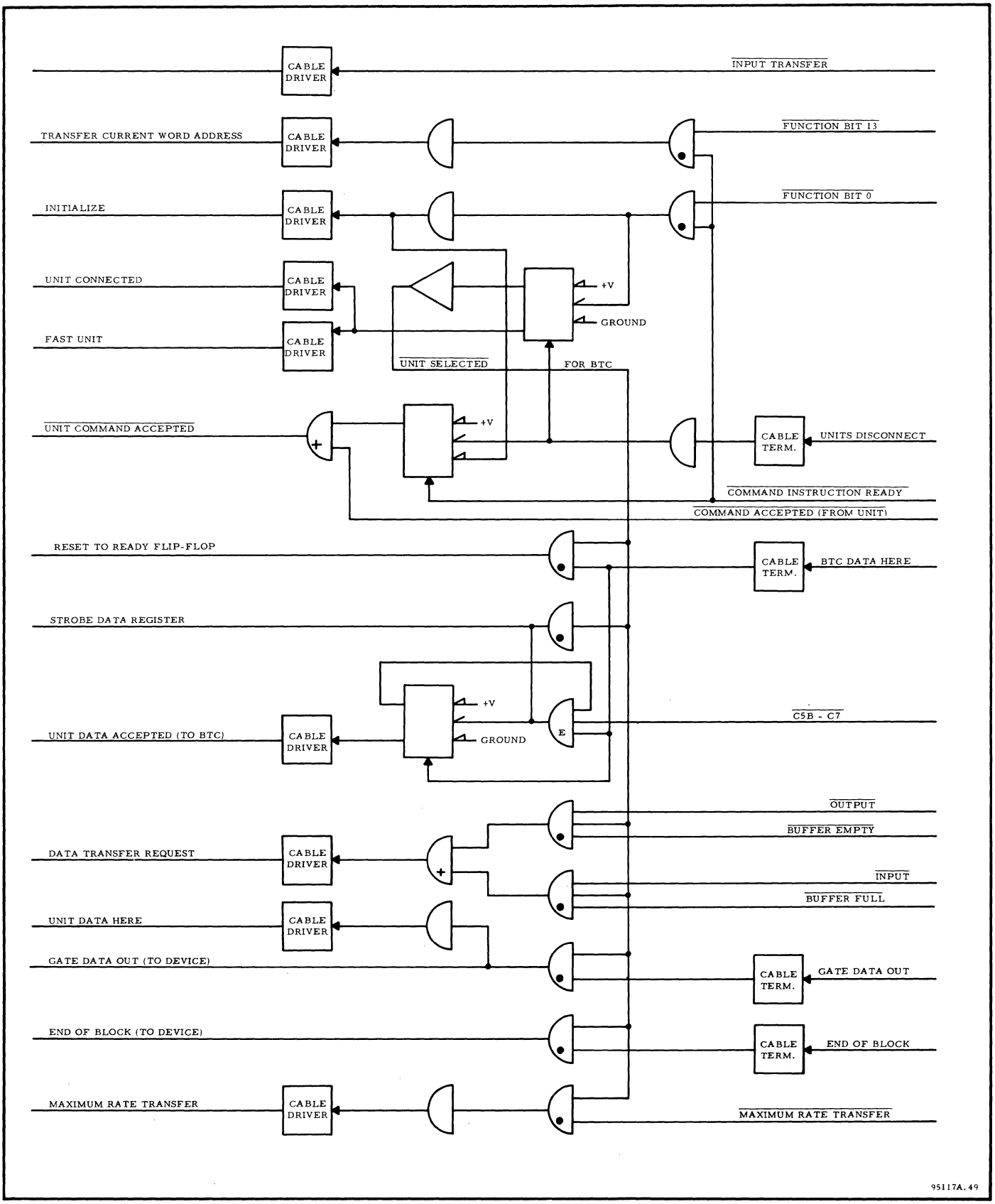


Figure 4-9. Block Transfer Control Interface RTL Logic Diagram

The mainframe may be interrupted by a particular interrupt level provided that the level has been previously enabled, and no higher level interrupt is active.

If a higher level interrupt is avtive when an interrupt signal occurs, the interrupt will be stored until the completion of execution of the higher level interrupt processing routine. The lower level routine will then be initiated. It will continue until completed or until interrupted by a higher level interrupt signal. In this case, the lower level routine will be completed after completion of the higher level routine. Program control will then be returned to the original point of interrupt. The priority logic enables any number of interrupt levels to be requested at the same time. Routine execution is always performed in the order of priority of the requested interrupts.

DETAILED LOGIC DESCRIPTION

The detailed logic circuits for one priority interrupt level are shown in figure 4-10. Two priority interrupt levels are contained on each 83252 priority interrupt circuit card. A priority interrupt system consists of up to 96 levels, input control gates, an output OR gate, and a unitary-to-binary converter.

The flip-flop is designated as the active flip-flop; the latch associated with input pins 12 and 16 is designated as the request latch, and the latch associated with input pins 22, 26 and 72 is designated as the enable latch. When the INTERRUPT switch on the control panel is positioned to ON, a Hold At Reset signal is removed from the active and enable latches. When a PIE instruction is performed, a strobe pulse from the timing and control logic is gated to query the AND gates associated with the enable latches in each circuit. If the level has been selected, the AND gate provides a signal to set the enable latch. A level select gate performs a similar function to reset the latch when a PID instruction is performed. The set output of the latch enables or inhibits a four-input AND gate.

The second Interrupt Source line connects the external signal to the priority interrupt logic. This signal is gated to set the request latch by the timing and control logic. If an interrupt signal has been generated, the two-input AND gate provides a signal to set the request latch. If the enable latch has been set and a higher priority interrupt routine is not being performed, the four-input AND gate provides an interrupt request to the unitary-to-binary converter and through the output OR gate to the timing and control logic. When the priority interrupt request is honored, a signal is received from the timing and control logic to strobe the unitary-to-binary converter to generate a Store, Place and Branch instruction which is placed on the memory output bus. The output of the request latch is also presented through an AND gate and OR gate to inhibit the output of the request latches in the lower level priority interrupt channels.

If the request is honored by the mainframe, the timing and control logic provides a signal which permits the active control flip-flop to set the active flip-flop. The positive output of the active flip-flop terminates the priority interrupt request at the four-input AND gate and inhibits the lower level priority interrupt channels.

Upon completion of the priority interrupt routine, a reset active flip-flop command is received from the control logic to query the AND gate associated with pin 46. If the active flip-flop of a higher level circuit has not been set, indicating that a higher priority interrupt level has interrupted this lower level interrupt's program, the AND gate provides a signal to reset the active and request latches. A detailed timing diagram and flow chart of the interrupt process is shown in figure 4-11.

STANDARD INTERRUPT CONNECTIONS

Two levels of interrupts are supplied with the basic computer. The priority levels of these interrupts are Group 0, Levels 11 and 12. The levels of the standard interrupts are placed at the bottom of Group 0 to make optional levels available of both higher and lower priorities.

The two standard levels are connected through the I/O cable to all peripheral units, as required. Interrupt signals in each unit are connected to one or the other of the two levels under program control.

Connection of a unit interrupt to a standard priority level is performed by execution of the CEU instruction. The format of the second word of the CEU instruction for the peripheral units contain three bits used to connect and disconnect the standard unit interrupts. The interpretation of the three bits is given in table 4-3.

ENABLING/DISABLING

Interrupt levels can be selectively enabled and disabled, one group (up to 12 levels) at a time, by the two instructions:

PIE PRIORITY INTERRUPT ENABLE 1306

PID PRIORITY INTERRUPT DISABLE 130601

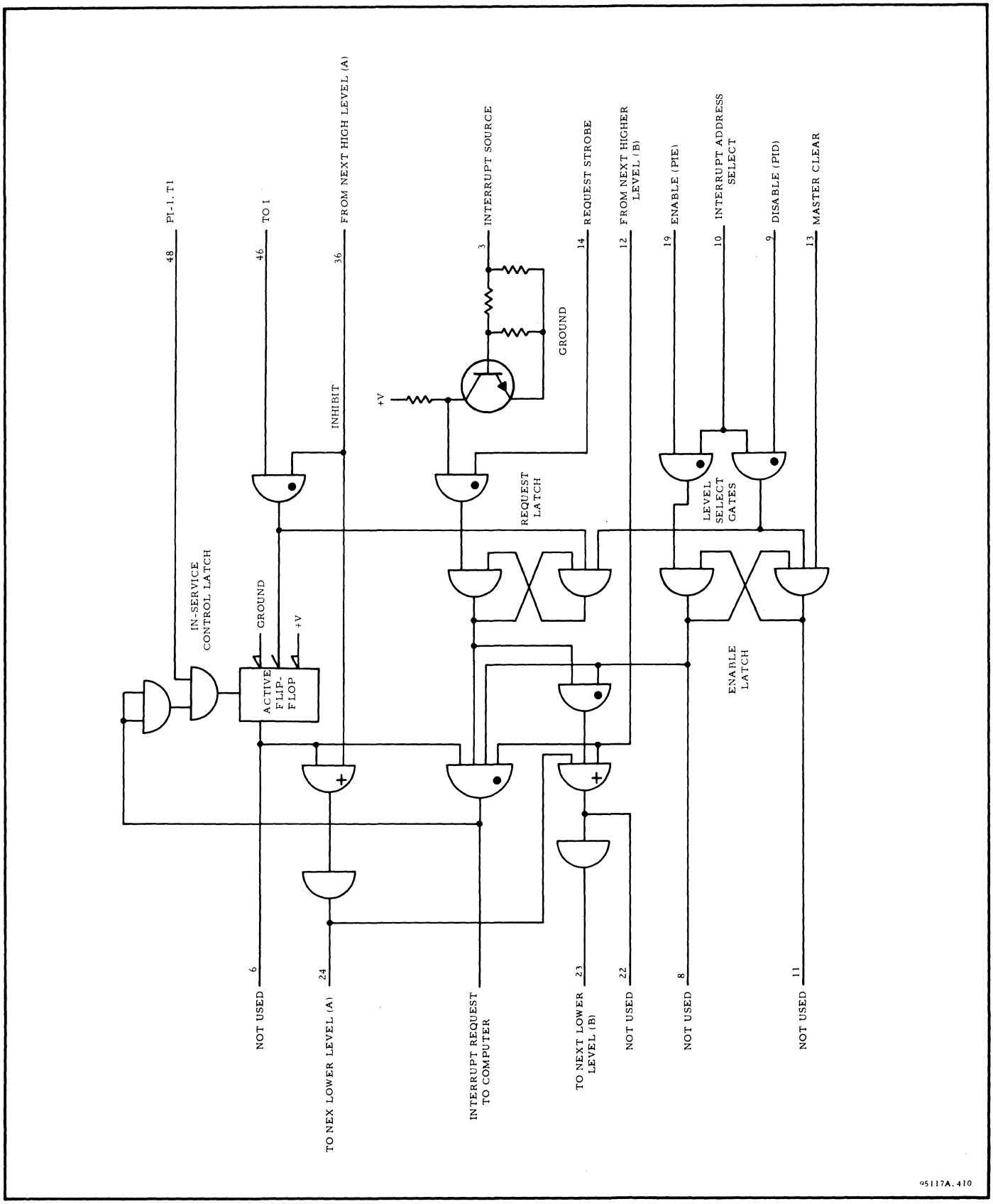
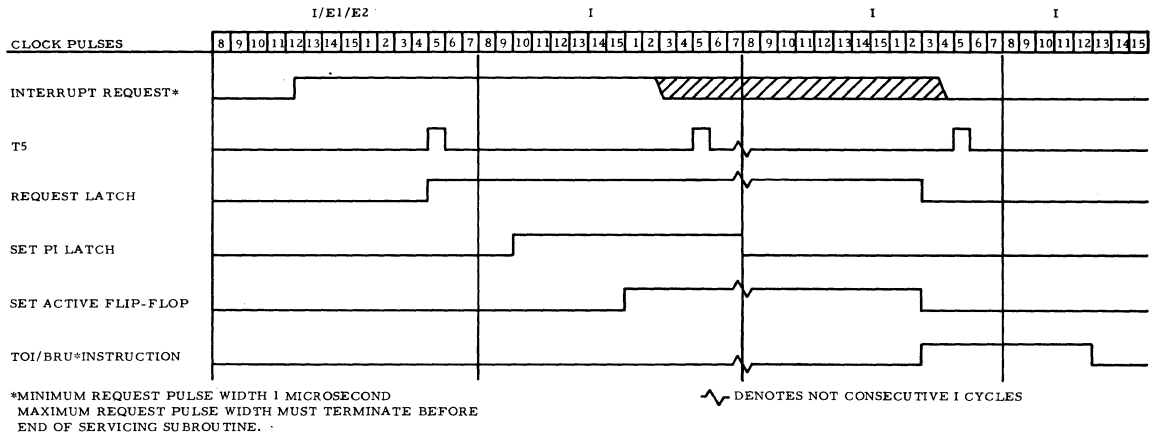
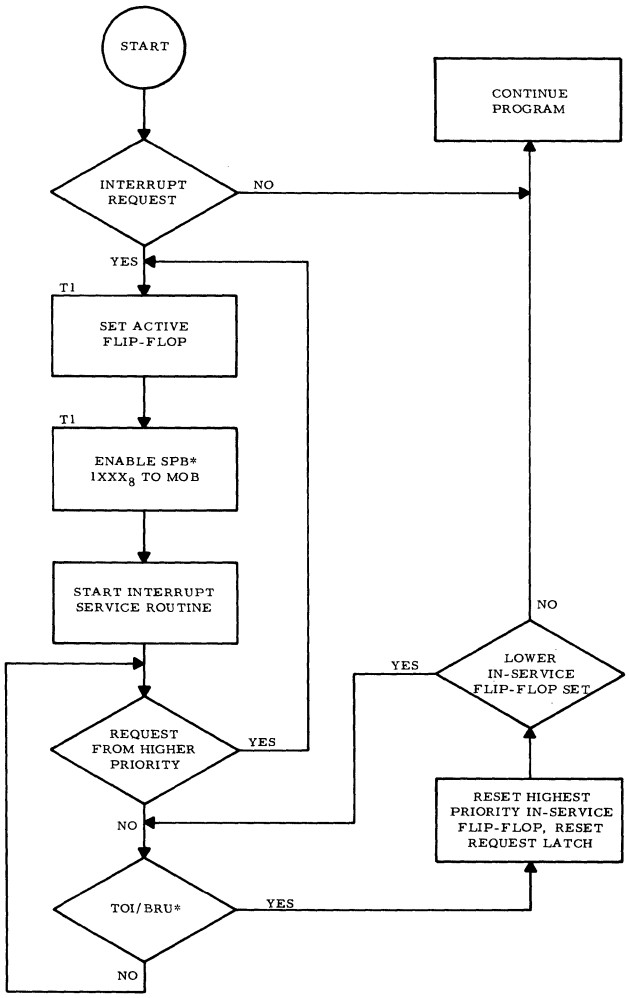


Figure 4-10. Priority Interrupt Logic Diagram (83252)





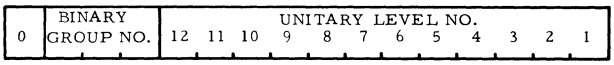
95117A.411

Figure 4-11. Priority Interrupt Timing Diagram

Table 4-3. Standard Interrupt CEU Bit Functions

Bit	Function
1 = ONE	Connect Levels Designated in Bits 2 and 3
2 = ONE	Connect/Disconnect Level 11, Group 0
3 = ONE	Connect/Disconnect Level 12, Group 0
1 = ZERO	Disconnect Levels Designated in Bits 2 and 3
2 = ZERO	Leave Level 11, Group 0 in Present State
3 = ZERO	Leave Level 12, Group 0 in Present State

The second word of these two-word instructions has a three-bit group field and a 12-bit level field which designate the group and the one to 12 levels within the group to be affected by the instruction. The second word bit assignment is shown below:



95117A.412

The group field is binarily-coded, group 000 being the highest priority group. The level field is unitarily-coded, a ONE in bit 15 is signifying the highest priority level within a group (level 1). An instruction which will cause the five highest levels in group TWO to be enabled is written in assembly language as:

```
PIE
DATA                '20037
```

Execution of this instruction leaves the seven lower levels within group TWO (if present) unaffected. They remain either enabled or disabled.

SECTION V
CONNECTOR DESCRIPTION

GENERAL

This section contains illustrations and tables which provide information required to interface external equipment with the SEL 810B Computer, including peripheral units, block transfer controls, and priority interrupts. The conventions of fabrication as applied to connector keying and signal definition are stated and will be used as a standard.

The I/O connector panel assembly (a representation is shown in figure 5-1) is located on the lower right of the computer cabinet. All equipment external to the main computer cabinet is connected through this panel or a similar panel in cabinet 2. The configuration shown will accommodate 32 peripheral units (J10 and J11), five Block Transfer Controls (J5 - J9), and 46 individual priority interrupt levels (J1 - J4).

If the system does not require in excess of 16 peripheral units, only J11 is provided on the I/O connector panel (1R1C). If additional jacks are required due to more than 16 units, wiring information is provided upon request. J11 is also referred to as the computer I/O bus, since it is common to all peripheral units supplying both data and control functions.

Figure 5-2 is a functional diagram showing a typical connection of a peripheral unit to the computer I/O bus. All units are wired with two connectors in parallel in order to facilitate additional peripheral units as desired. The end of the cable, that is, the last unit in the chain, is terminated with a terminator plug consisting of 100-ohm resistors across each twisted pair. Without this terminator plug, reflections will be set up on the open transmission line which will cause malfunctions.

Table 5-1 and figure 5-3 define the signal lines contained in the 51-pair cable, the keying associated with the cable connectors, and the fixed connectors which are mounted in the unit and in the computer I/O connector assembly. The numbers specifying keying hardware are listed in table 5-2, which is a list of the connector hardware and the associated AMP part number. Table 5-1 wiring information applies only to J11.

Table 5-3 and figure 5-4 define the signal lines and connector keying applying to the first Block Transfer Control. The connectors defined as J5 - J9 are used to connect units operating in conjunction with BTC. Up to 16 units can be connected to one BTC; thus similar chaining of units as in the I/O bus is required. Each unit is connected to the computer I/O bus regardless of whether it operates on a BTC, since data communication and command functions are performed via the I/O bus.

Table 5-4 and 5-5 define the priority interrupts, their associated connector, group number, and assigned octal memory location. The two standard interrupts are shown in the table which are the last two interrupts in group. Each interrupt group contains 12 priority interrupts. The first two interrupts are assigned to Power Fail Safe and Stall Alarm.

These two interrupt levels have the highest priority, but are not controllable in respect to enabling or disabling under program control. The highest interrupt available to the programmer then becomes interrupt level I, which is in group 0. Priority decreases as group and interrupt numbers increase.

Figure 5-5 shows functionally the connection of a priority interrupt and also defines the connector keying. The external signal necessary to ensure recognition by the computer is also shown.

Table 5-1. I/O Bus Connector

SIGNAL WIRE			GROUND WIRE	
Pin No.	Wire Color	Function	Pin No.	Wire Color
101	Brown	Data or Function Bit 0	201	Black
102	Red	Data or Function Bit 1	202	Black
103	Orange	Data or Function Bit 2	203	Black

Table 5-1. I/O Bus Connector (Cont'd)

SIGNAL WIRE			GROUND WIRE	
Pin No.	Wire Color	Function	Pin No.	Wire Color
104	Yellow	Data or Function Bit 3	204	Black
105	Green	Data or Function Bit 4	205	Black
106	Blue	Data or Function Bit 5	206	Black
107	Violet	Data or Function Bit 6	207	Black
108	Gray	Data or Function Bit 7	208	Black
109	White	Unit Select Bit 1 from Computer	209	Black
110	Brown	Unit Select Bit 2 from Computer	210	Blk-Wht
111	Red	Unit Select Bit 4 from Computer	211	Blk-Wht
112	Orange	Unit Select Bit 8 from Computer	212	Blk-Wht
113	Yellow	Unit Select Bit 16 from Computer	213	Blk-Wht
301	Brown	Data Bit 8	401	White
302	Red	Data Bit 9	402	White
303	Orange	Data Bit 10	403	White
304	Yellow	Data Bit 11	404	White
305	Green	Data Bit 12	405	White
306	Blue	Data Bit 13	406	White
307	Violet	Data Bit 14	407	White
308	Gray	Data Bit 15	408	White
309	Green	Unit Select Bit 32 from Computer	409	Blk-Wht
310	Blue	Transfer Instruction (from Computer)	410	Blk-Wht
311	Violet	Test Instruction (from Computer)	411	Blk-Wht
312	Brown	Command Instruction (from Computer)	412	Blue
313	Red	Instruction Sync (from Computer)	413	Blue
501	Brown		601	Gray
502	Red		602	Gray
503	Orange	Input Priority Interrupt (to Computer)	603	Gray
504	Yellow	Output Priority Interrupt (to Computer)	604	Gray
505	Green	Parity Error (from Computer)	605	Gray
506	Blue	Parity Check Request (to Computer)	606	Gray
507	Violet	Output Parity Bit (from Computer)	607	Gray
508	Orange	Input Parity Bit (to Computer)	608	Blue
509	Yellow	Input/Output (from Computer)	609	Blue
510	Green	Wait Bit (from Computer)	610	Blue
511	Brown	Clock (CL5B-CL7) from Computer	611	Green
512	Red	Computer Data Here (from Computer)	612	Green
513	Orange	Computer Data Accepted (from Computer)	613	Green

Table 5-1. I/O Bus Connector (Cont'd)

SIGNAL WIRE			GROUND WIRE	
Pin No.	Wire Color	Function	Pin No.	Wire Color
701	Brown		801	Violet
702	Red		802	Violet
703	Orange		803	Violet
704	Yellow		804	Violet
705	Green		805	Violet
706	Blue		806	Violet
707	Yellow		807	Green
708	Brown		808	Yellow
709	Red	Master Clear (from Computer)	809	Yellow
710	Brown	Unit Test Return (to Computer)	810	Orange
711	Red	Unit Sync Return (to Computer)	811	Orange
712	Brown	Unit Data Accepted (to Computer)	812	Red
713		Shield	813	

Table 5-2. I/O Cable and Connector Parts List

	Part Description	AMP Part Number
(1)	104-Pin Contact Male Block	201036-1
(2)	104-Pin Contact Female Block	201037-1
(3)	Corner Guide Socket for 104-Pin Contact Block	201047-7
(4)	Corner Guide Pin for 104-Pin Contact Block	201046-1
(5)	Fixed Jack Screw, Male	200873-8
(6)	Fixed Jack Screw, Female	200872-8
(7)	Turnable Jack Screw, Male	200871-1
(8)	Turnable Jack Screw, Female	200867-1
(9)	No. 30-Pin (No. 32-30 Wire)	201555-1
(10)	No. 30-Socket (No. 32-30 Wire)	201554-1
(11)	50-Pin Contact Male Block	200276-2
(12)	50-Pin Contact Female Block	200277-2
(13)	Corner Guide Socket for 50 Contact Block	200835-5
(14)	Corner Guide Pin for 50 Contact Block	200833-5
(15)	104-Pin Hood	201364-4
(16)	104-Pin Shield Assembly	201131-1
(17)	Die Set for No. 30-Pin and Socket	69158
(18)	Hand Tool for No. 30-Pin and Socket	45099
(19)	Extraction Tool for all Contacts	305183
(20)	Insertion Tool for all Contacts	380431-2

Table 5-2. I/O Cable and Connector Parts List (Cont'd)

Part Description	AMP Part Number
I/O Cables - Constructed of No. 30 gauge, color-coded wires (19 strands/wire). Cables are available with 51 or 24 twisted pairs of wires. Wire bundle is wrapped in Foil-Aluminum-Mylar shield and covered with vinyl jacket. Shield is provided with No. 22 gauge, stranded drain wire. Standard lengths in stock: 10 feet and 30 feet. (Manufacturer - Columbia Wire and Cable Company)	

Table 5-3. Block Transfer Control Connector

SIGNAL WIRE			GROUND WIRE	
Pin No.	Wire Color	Function	Pin No.	Wire Color
101	Brown	Initialize from Computer	201	Black
102	Red	Unit Connected	202	Black
103	Orange	Input/Output	203	Black
104	Yellow	Unit Data Here (to Computer)	204	Black
105	Green	Unit Data Accepted (to Computer)	205	Black
106	Blue	Unit Data Transfer Request	206	Black
107	Violet	Transfer Current Address	207	Black
108	Gray	Maximum Transfer Rate	208	Black
109	White	Clock Sync (C5B-C7)	209	Black
110	Brown	Fast Unit (to Computer)	210	Blk-Wht
111	Red		211	Blk-Wht
112	Orange		212	Blk-Wht
			213	Blk-Wht
301	Brown	BTC Data Here (from Computer)	401	White
302	Red	Gate Data Out	402	White
303	Orange	Unit Disconnect	403	White
304	Yellow	Instruction Received	404	White
305	Green	End of Block	405	White
306	Blue		406	White
307	Violet		407	White
308	Gray		408	White
309	Green		409	Blk-Wht
310	Blue		410	Blk-Wht
311	Violet		411	Blk-Wht
312	Brown		412	Blue
			413	Blue

Table 5-4. Priority Interrupt Assignments - SEL 810B

Group	Level	Jack	Signal Pin	Ground Pin	Octal Memory Location	Group	Level	Jack	Signal Pin	Ground Pin	Octal Memory Location
--*	--	N/A	N/A	N/A	1000	2	8	J3	108	208	1051
--**	--	N/A	N/A	N/A	1001	2	9	J3	109	209	1052
0	1	J1	101	201	1002	2	10	J3	110	210	1053
0	2	J1	102	202	1003	2	11	J3	111	211	1054
0	3	J1	103	203	1004	2	12	J3	112	212	1055
0	4	J1	104	204	1005	3	1	J4	101	201	1102
0	5	J1	105	205	1006	3	2	J4	102	202	1103
0	6	J1	106	206	1007	3	3	J4	103	203	1104
0	7	J1	107	207	1010	3	4	J4	104	204	1105
0	8	J1	108	208	1011	3	5	J4	105	205	1106
0	9	J1	109	209	1012	3	6	J4	106	206	1107
0	10	J1	110	210	1013	3	7	J4	107	207	1110
0	11	J1	111	211	1014	3	8	J4	108	208	1111
0	12	J1	112	212	1015	3	9	J4	109	209	1112
1	1	J2	101	201	1022	3	10	J4	110	210	1113
1	2	J2	102	202	1023	3	11	J4	111	211	1114
1	3	J2	103	203	1024	3	12	J4	112	212	1115
1	4	J2	104	204	1025	4	1	J4	101	201	1122
1	5	J2	105	205	1026	4	2	J4	102	202	1123
1	6	J2	106	206	1027	4,	3	J4	103	203	1124
1	7	J2	107	207	1030	4,	4	J4	104	204	1125
1	8	J2	108	208	1031	4,	5	J4	105	205	1126
1	9	J2	109	209	1032	4,	6	J4	106	206	1127
1	10	J2	110	210	1033	4,	7	J4	107	207	1130
1	11	J2	111	211	1034	4,	8	J4	108	208	1131
1	12	J2	112	212	1035	4,	9	J4	109	209	1132
2	1	J3	101	201	1042	4,	10	J4	110	210	1133
2	2	J3	102	202	1043	4,	11	J4	111	211	1134
2	3	J3	103	203	1044	4,	12	J4	112	212	1135
2	4	J3	104	204	1045	5,	1	J5	101	201	1142
2	5	J3	105	205	1046	5,	2	J5	102	202	1143
2	6	J3	106	206	1047	5,	3	J5	103	203	1144
2	7	J3	107	207	1050	5,	4	J5	104	204	1145

* Power Fail Safe (Optional)
** Stall Alarm (Optional)

Table 5-4. Priority Interrupt Assignments - SEL 810B (Cont'd)

Group	Level	Jack	Signal Pin	Ground Pin	Octal Memory Location	Group	Level	Jack	Signal Pin	Ground Pin	Octal Memory Location
5,	5	J5	105	205	1146	6,	9	J6	109	209	1172
5,	6	J5	106	206	1147	6,	10	J6	110	210	1173
5,	7	J5	107	207	1150	6,	11	J6	111	211	1174
5,	8	J5	108	208	1151	6,	12	J6	112	212	1175
5,	9	J5	109	209	1152	7,	1	J7	101	201	1202
5,	10	J5	110	210	1153	7,	2	J7	102	202	1203
5,	11	J5	111	211	1154	7,	3	J7	103	203	1204
5,	12	J5	112	212	1155	7,	4	J7	104	204	1205
6,	1	J6	101	201	1162	7,	5	J7	105	205	1206
6,	2	J6	102	202	1163	7,	6	J7	106	206	1207
6,	3	J6	103	203	1164	7,	7	J7	107	207	1210
6,	4	J6	104	204	1165	7,	8	J7	108	208	1211
6,	5	J6	105	205	1166	7,	9	J7	109	209	1212
6,	6	J6	106	206	1167	7,	10	J7	110	210	1213
6,	7	J6	107	207	1170	7,	11	J7	111	211	1214
6,	8	J6	108	208	1171	7,	12	J7	112	212	1215

Table 5-5. Interrupt Connector

SIGNAL WIRE			GROUND WIRE	
Pin No.	Wire Color	Interrupt Level	Pin No.	Wire Color
101	Brown	1	201	Black
102	Red	2	202	Black
103	Orange	3	203	Black
104	Yellow	4	204	Black
105	Green	5	205	Black
106	Blue	6	206	Black
107	Violet	7	207	Black
108	Gray	8	208	Black
109	White	9	209	Black
110	Brown	10	210	Blk-Wht
111	Red	11	211	Blk-Wht
112	Orange	12	212	Blk-Wht
			213	Blk-Wht

All Connectors

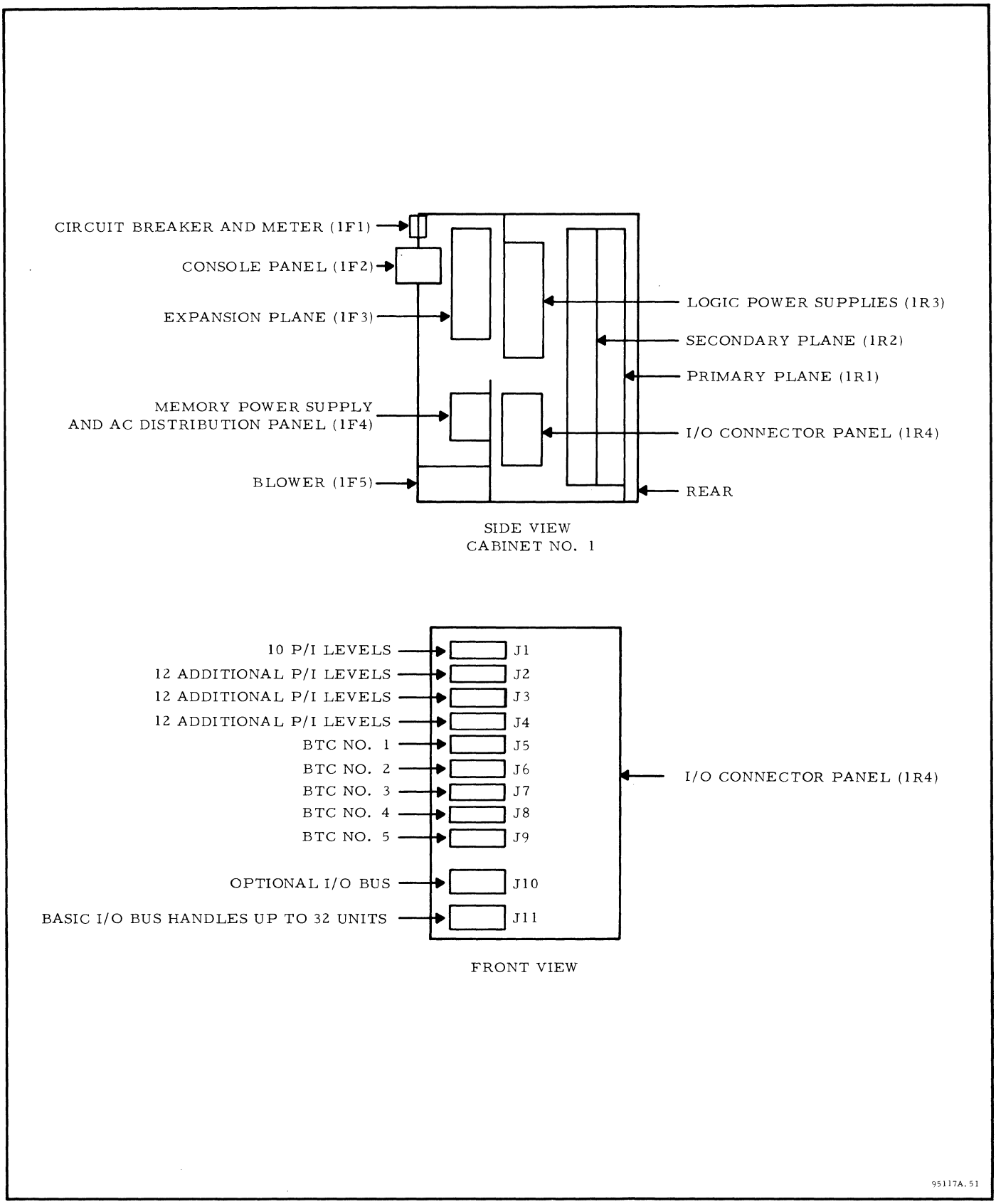


Figure 5-1. I/O Connector Panel

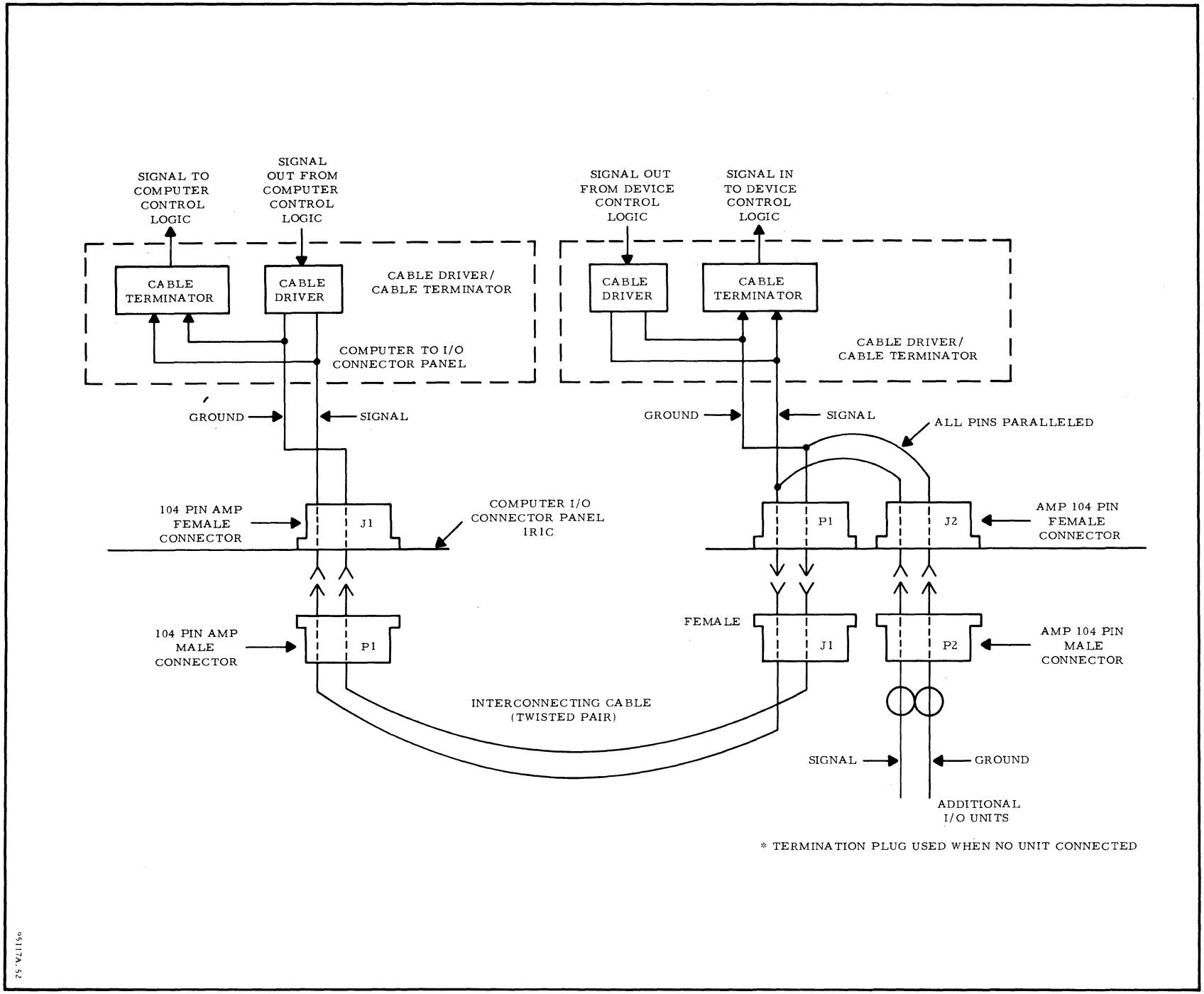


Figure 5-2. Typical I/O Unit Cabling

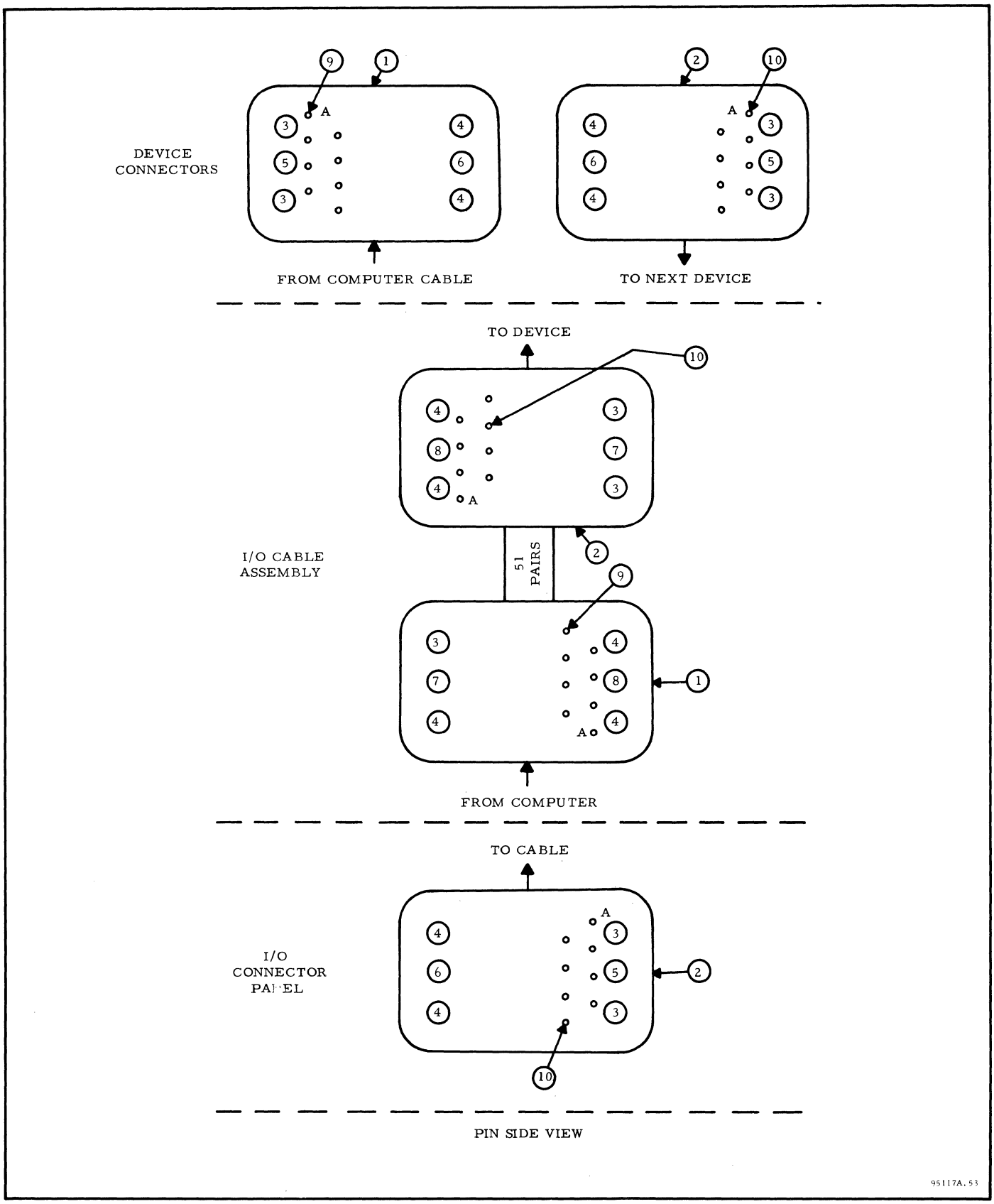


Figure 5-3. Connector Conventions I/O Bus

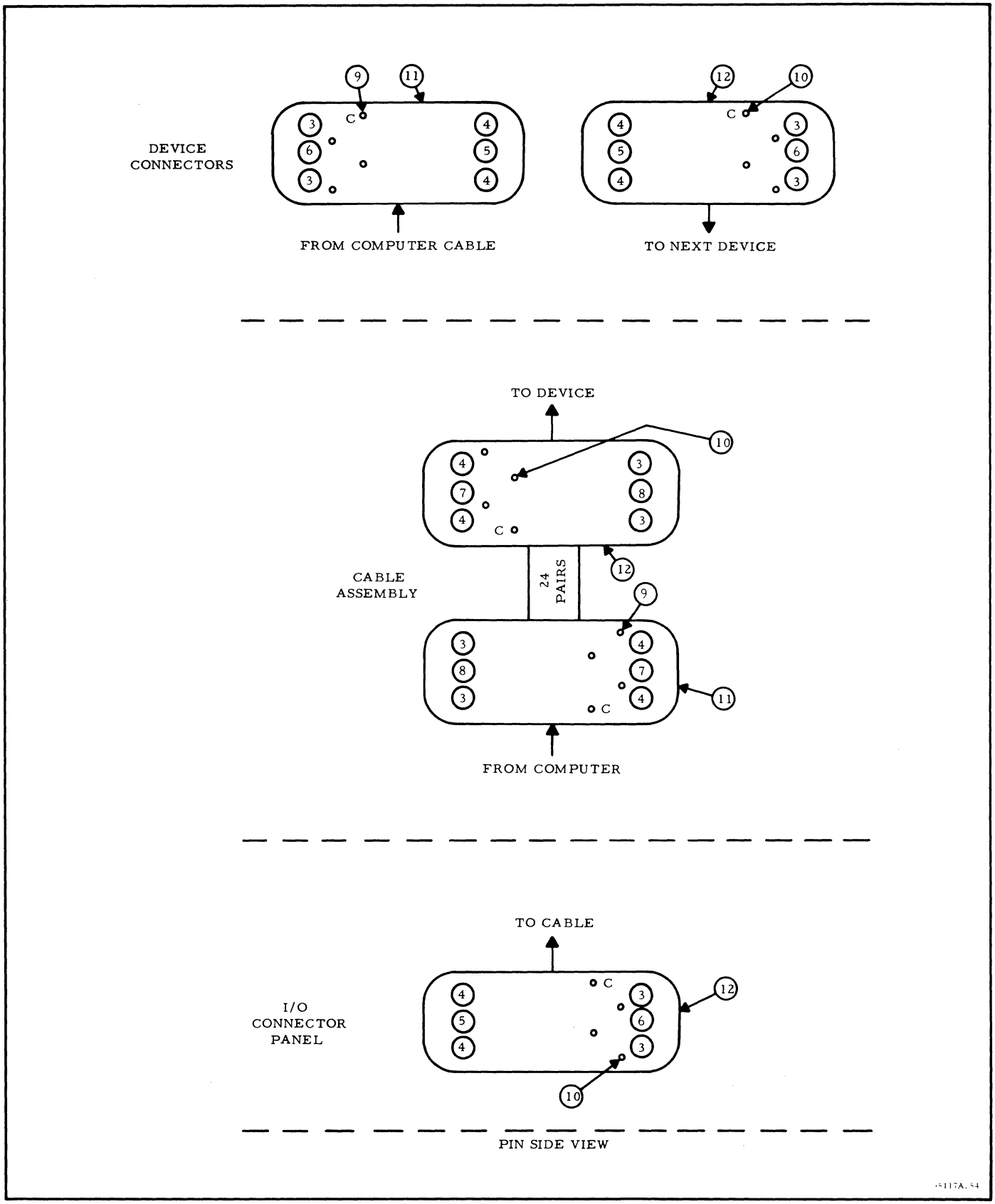


Figure 5-4. Connector Conventions Block Transfer Control

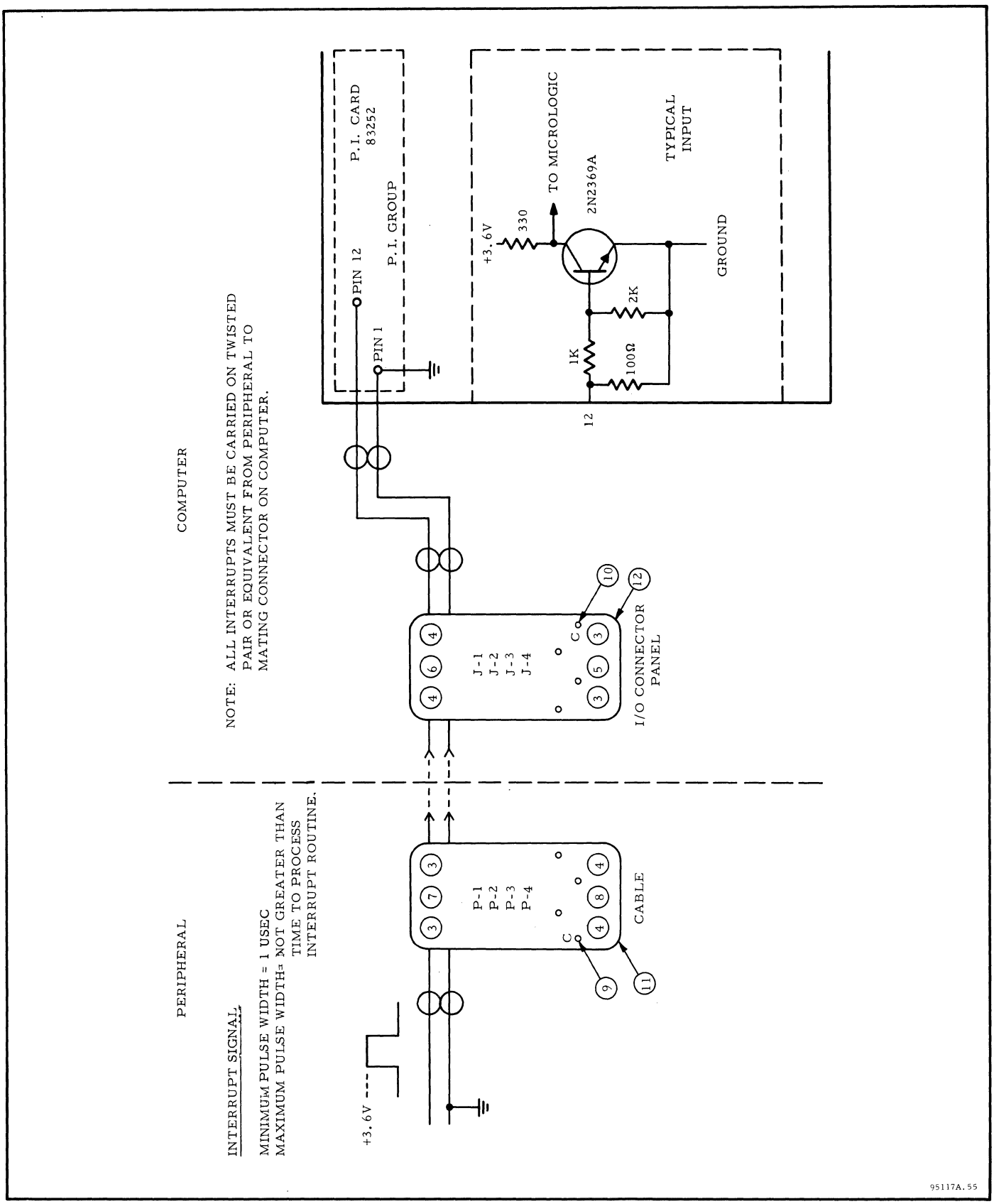


Figure 5-5. Typical Priority Interrupt Connection SEL 810B