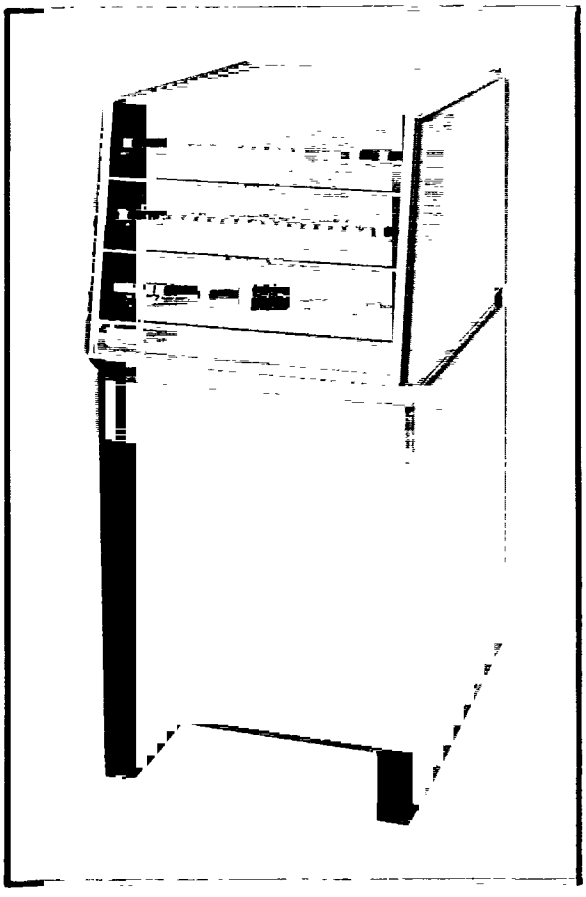


BIT 480 SYSTEMS

DESCRIPTION OF

INPUT – OUTPUT

BUSINESS INFORMATION TECHNOLOGY, INC

SERIES 480

Three Erie Drive, Natick, Massachusetts 01760, Telephone 617 235-8842

CONTENTS

1. INTRODUCTION TO BIT 480 INPUT/OUTPUT

1.0 GENERAL

The input/output system of the BIT 480 Processor consists of a number of data-channels each capable of operating in a Read-Write mode. The processor may have one, two, three or four such Read-Write channels. Certain channels are capable of executing data transfers while the processor is simultaneously computing. A maximum of sixty-four peripheral devices may be addressed.

1.1 INTERRUPT

All BIT 480 Processors come equipped with the necessary hardware to service program interrupts initiated from I/O devices. The number of interrupting devices permitted is not limited except by the rate at which interrupts are occurring and the length of time for a subroutine to service them. The basic BIT 480 Processor is equipped to recognize and service interrupts on as many as 64 peripheral devices. The machine can further issue commands to I/O devices, the effect of which is to prohibit further interrupts or enable interrupts on devices formerly prohibited.

Interrupt requests can be processed on either a priority basis or sequence basis, the choice being determined by the interrupt subroutine.

A complete description of the interrupt function of the BIT 480 and design procedure for incorporating interrupt capability into an I/O device is found in this manual in Section 3. The I/O Control Unit is that collection of hardware associated with a peripheral device which enables it to receive and execute commands from the computer. Control Units generally require some logic which is unique for their particular I/O device, and another set of logic which is more or less the same for any I/O control unit.

The information which is given in this manual is intended to convey the idea of what the basic ingredients of an interface are and how it is designed.

1.2 DATA CHANNELS

From a gross viewpoint, we may view the I/O system of the computer as illustrated in Figure 1.

This figure shows the computer as having a number of I/O devices, each of which has associated with it a control unit. It will be noted that the I/O devices and control units are divided into four groups. Within each group is a number of I/O devices and a common set of data transfer lines to the computer, called channel lines. The set of common data lines connects with some control logic at the computer. This complex of common data lines and control logic in the computer is called a DATA CHANNEL.

The computer may have a maximum of four distinct data channels. The data channel control logic within the computer consists of a collection of hardware which passes data into and out of the core memory of the computer. This set of hardware consists principally of an address register and a data register. The channel address register keeps track of where, in the core memory, the data is to be entered or extracted. The channel data register is used as temporary storage between the computer and the selected I/O device on the channel.

The system may have a maximum of four data channels. Three data channels permit simultaneous I/O and compute allowing execution of data transfers by the procedure of stealing memory cycles of the core memory only when necessary. This type of Input/Output is called "overlapped". The computer need not be equipped with an overlapped data channel. In this case, data transfers are executed without simultaneous computation on a channel called "Channel G". All BIT 480 systems are equipped with Channel G.

1.2.1 I/O LINES

In Figure 1A is shown a representation of

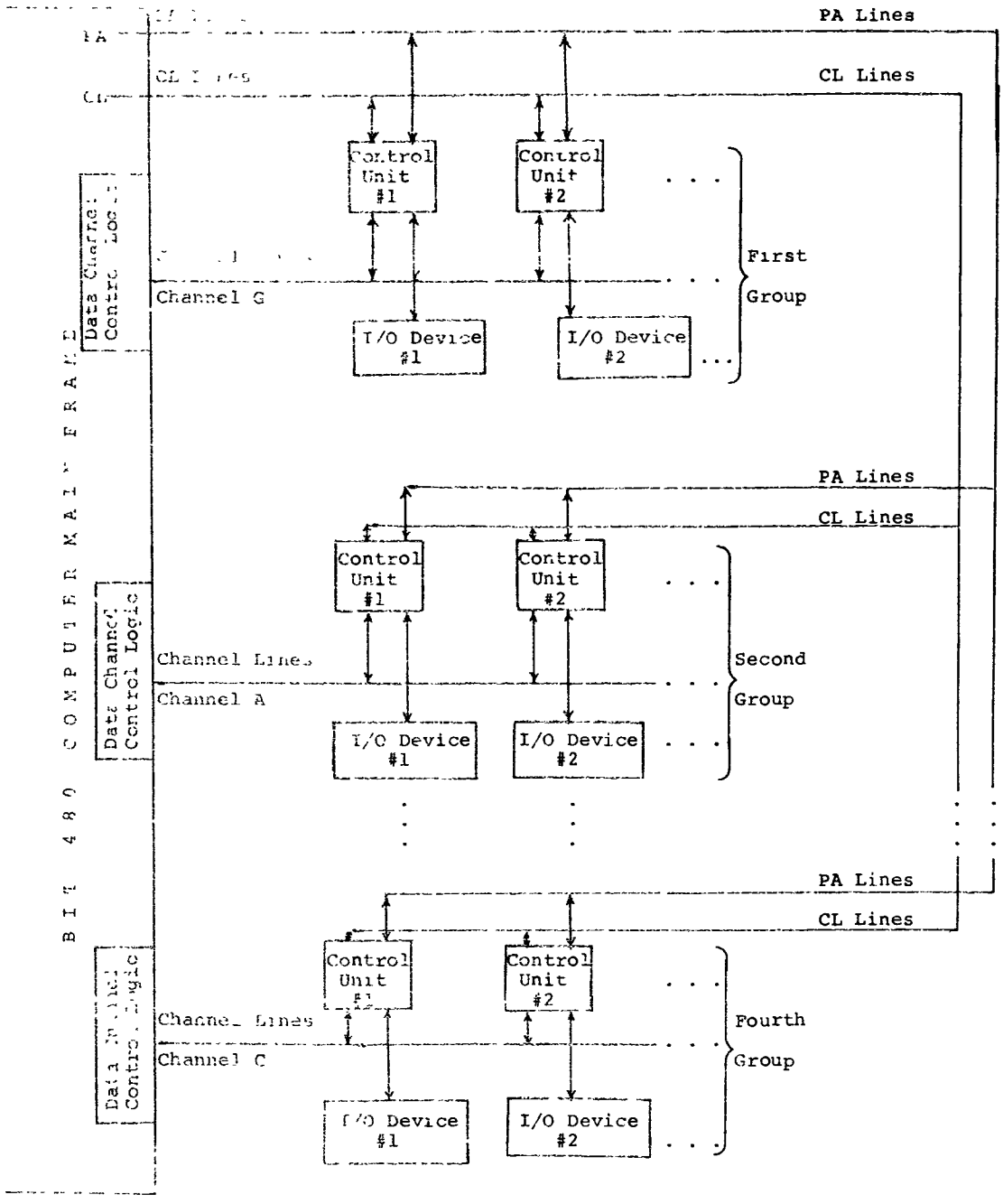


FIGURE 1 - BLOCK DIAGRAM OF I/O SYSTEM

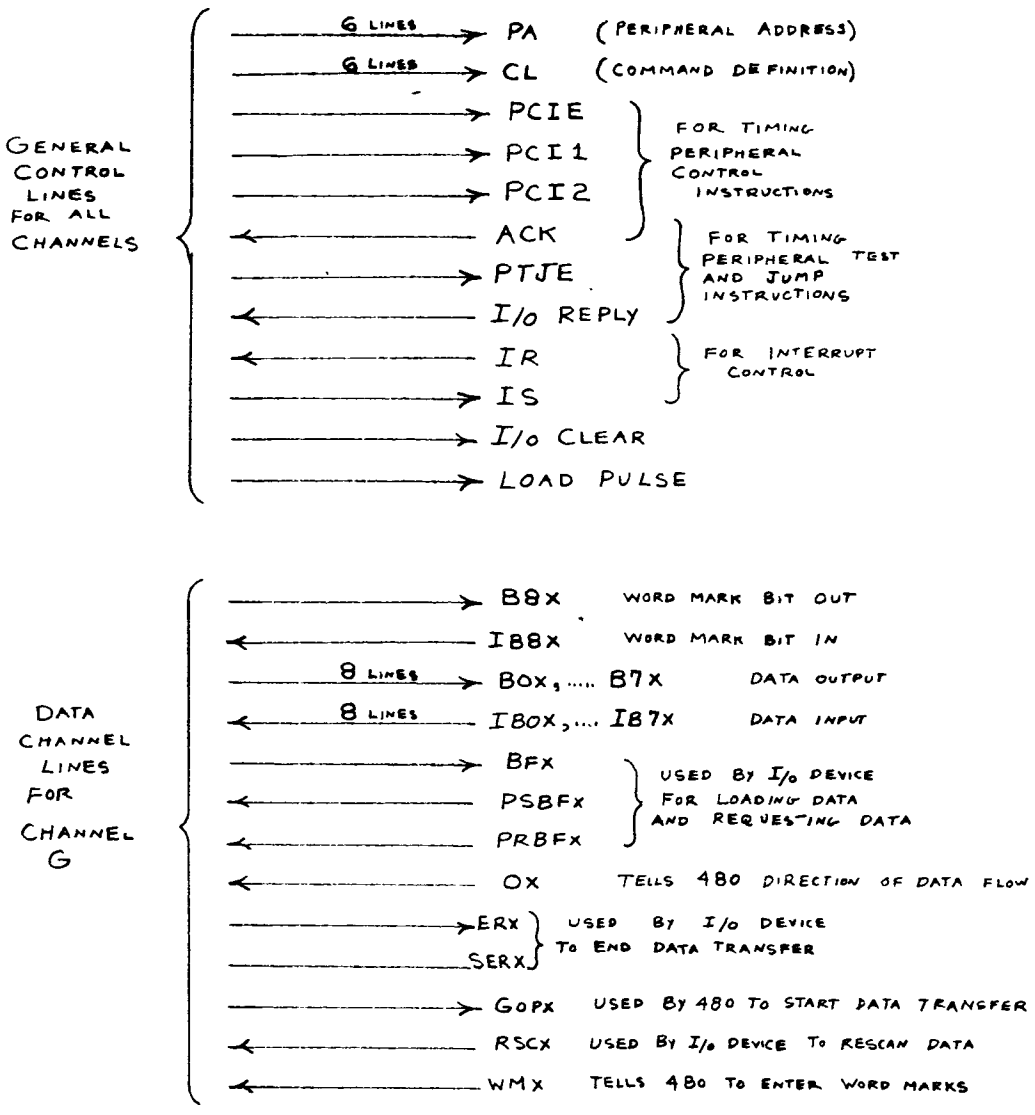


FIG 1A - I/O LINES FROM 480

input-output lines provided on a BIT 480 Processor having a single data channel. Briefly their meanings are as follows:

The PA lines are "Peripheral Address" lines

The CL lines define, to the addressed I/O device, what type of command is being given.

The functions PCIE, PCI1, PCI2 and ACK are timing functions supplied by the BIT 480 when a command is being given to an I/O device.

The functions ACK, PTJE and I/O REPLY are used when the BIT 480 is making a test on an I/O device.

The functions IR and IS are used for interrupt control. I/O CLEAR is a reset pulse which is activated by a button on the Manual Control Panel.

Load Pulse is generated from the Control Panel and is used for loading programs. The next set of 16 lines provides data output and input lines for a particular data channel.

The functions BFX, PSBFX and PRBFX are used by the I/O device for controlling the flow of data characters into and out of the computer.

The line OX from the I/O device specifies whether a data transfer is to be into or out of the BIT 480.

The functions ERX and SERX are used by the I/O device to end a data transfer.

The function GOPX from the BIT 480 commands an I/O device to begin a data transfer.

The input line RSCX enables the I/O device to cause repeat transmission from a record area in the BIT 480.

1.3 COMPUTER INSTRUCTIONS CONTROLLING I/O

There are three instructions which may be used by the computer to control the I/O system. They are:

Peripheral Control Instruction (PCI)

Peripheral Transfer Instruction (PTI)

Jump Instruction (J)

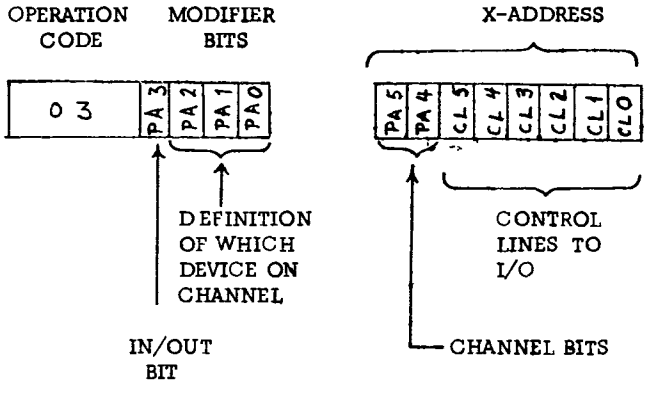


FIGURE 2 - PERIPHERAL CONTROL INSTRUCTION

1.3.1 PERIPHERAL CONTROL INSTRUCTION

The format of a PCI Instruction is shown in Figure 2. It is a two-byte instruction. The four modifier bits and the two high order bits of the X address constitutes a set of Peripheral Address lines:

PA_0, PA_1, PA_2, PA_3, PA_4, PA_5. The six low order bits of the low order character are a set of Control Lines: CL_0, CL_1, CL_2, CL_3, CL_4, CL_5.

The PA lines are used to select an I/O device. The CL lines are used to specify what the instruction is commanding to the selected device.

1.3.1.1 Addressing of I/O Devices

The PA bits have the following significance:

PA_5, PA_4:

These bits define one of four channels with which the BIT 480 may be equipped. Three data channels permit simultaneous I/O and compute, allowing the execution of data transfers by the procedure of stealing memory cycles of the core memory only when necessary. This type of input/output is called "overlapped". The computer may not come equipped with an overlapped data channel. In this case, data transfers are executed without simultaneous computation, in a channel called "Channel G". All BIT 480 Systems are equipped with Channel G. The PA_5 and PA_4 bits are assigned to data channels as follows:

PA_5	PA_4	CHANNEL
0	0	A
0	1	B
1	0	C
1	1	G (non overlapped)

PA_3:

This bit signifies to the I/O device whether a data transfer is to be IN or OUT of the computer. The significance of the bit is as follows:

$PA_3 = 1$ Means Transfer Out

$PA_3 = 0$ Means Transfer In

1.3.1.2 CL Lines

The CL Lines, define the operation to be performed on the I/O device which has been addressed by the PA Lines.

There are four different classes of PCI Instructions. These classes are defined by the high order CL bits as follows:

1.3.1.2.1 CLASS I: CL_5, CL_4 = 00

This class of PCI Instruction is not used for selecting an I/O device for data transfer or for test. It is used for initiating such functions as Backspace, Rewind, or Move to Top of Form. The four low order Control Line bits, CL_3, CL_2, CL_1, CL_0 are used to define the particular type of command for this class. The execution commences as soon as this type of PCI Instruction is ended.

The instruction ties up to the I/O device which is addressed. The "BUSY" flop within the I/O Control Unit is set during the execution of this command such that if the Control Unit is subsequently tested for a "BUSY" condition, a "yes" reply to the computer will result.

1.3.1.2.2 CLASS II: CL_5, CL_4 = 01

This class of PCI Instruction is used to select an I/O device for a data transfer to or from the computer. The peripheral device address is specified by the PA bits. This instruction does not initiate the data transfer. It only selects which I/O device on a channel can make the transfer. The selection of one device on the data channel disables all other devices on the same channel from making data transfers. This PCI Instruction causes the addressed I/O Control Unit to have its "DATA SELECT" flop set. Other I/O Control Units on the same data channel have their "DATA SELECT" flops reset. Selection of an I/O device by means of a PCI Instruction of this class does not cause the "BUSY" flop in its Control Unit to be set, since this instruction does not initiate a data transfer. The BUSY flop is set by the subsequent PTI Instruction which starts the transmission with a "GO" pulse to the Control Unit.

Bit CL_1 is used to determine whether the data being transferred is to include word marks.

If $CL_1 = 0$, the transmission will not contain word marks.

If $CL_1 = 1$, the transmission will contain word marks.

When the PCI Instruction specifies transmission with word marks, a control flop called WM is set in theControl Unit of the I/O device being selected. This enables the subsequent transmission to include word marks. If the transmission happens to be into the computer and with word marks, the consequence is that word marks in memory are erased during the transfer and new word marks are entered in accordance with the incoming data.

1.3.1.2.3 CLASS III: CL_5, CL_4 = 1 0

This class of PCI Instruction is used to select an I/O device for a test to be performed during a subsequent Jump Instruction. The peripheral address of the desired I/O device is specified by the PA bits. Selection of one I/O device for test causes all other I/O devices on all other channels to be de-selected. The selection of an I/O device causes its TEST flop to be set in its Control Unit, and all the TEST flops of all other Control Units to be reset. (The selection of an I/O device for test has no effect on its DATA SELECT flop. Only a CLASS II instruction can change the DATA SELECT flop.)

1.3.1.2.4 CLASS IV: CL_5, CL_4 = 1 1

This class of PCI Instruction is used for control of I/O devices which are designed with interrupt capability. Certain CL codes listed below, when occurring in a PCI Instruction have the following significance:

110000 - this is an "enable interrupt" command to the addressed I/O device. It causes the EI (enable interrupt) flop in the I/O control unit to be set. The I/O device can then get service from the BIT 480 Processor on any of its interrupt requests in the future.

110001 - this is a "disable interrupt" command to the addressed I/O device. It causes the EI flop in the addressed I/O device to be reset. The result is that the BIT 480 Processor will not process interrupt requests from this device.

1.3.1.3 Auxiliary Timing Functions

In addition to the PA lines and CL lines, four other timing functions are supplied for communication between the processor and the I/O device during the PCI Instruction. These four functions communicate with all I/O Control units. They are illustrated on the timing diagram of Figure 3.

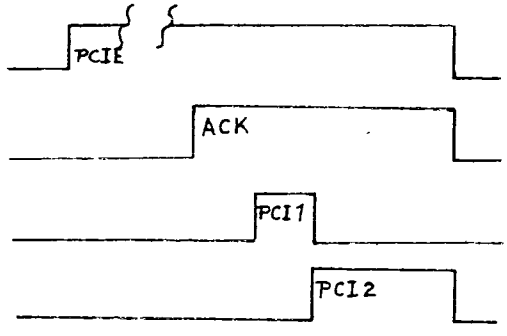


FIGURE 3 - AUXILIARY TIMING FUNCTIONS

The function called PCIE is sent out to all control units to indicate that a peripheral control instruction is being executed. This is necessitated by the fact that the CL and PA lines are changing each time

a new instruction is fetched. For most instructions, the PA and CL lines should be ignored. However, during PCI instructions, the CL and PA lines represent a command and an address respectively. The function PCIE, shown in Figure 3, is given during the execute time of the PCI Instruction. At this time, the CL and PA lines are guaranteed to be settled down and may be sampled by all I/O control units. The ACK Line is an acknowledge line, common to all I/O control units. Only the control unit being addressed may activate the ACK line.

Two functions, PCI_1 and PCI_2 are supplied by the computer to all I/O control units after the computer has received the ACK function.

These two pulses may be used by the addressed I/O control unit to clear and load its instruction register from the CL lines, or for other purposes.

The ACK line is necessitated by the fact that it is possible for an I/O device to receive a PCI Instruction with a new command before the I/O device has completed the execution of the previous command. In this case, it is vital that the computer be prevented from proceeding. The I/O control unit does this by failing to send an acknowledgement to the computer on the ACK line. This causes the computer to stay in the execution of the PCI instruction and not transmit the pulses PCI_1 and PCI_2. As soon as the I/O control unit completes the instruction in process and goes out of the busy state, it sends the acknowledge function (ACK) to the computer. The computer transmits the two pulses PCI_1 and PCI_2, and then ends the instruction.

1.3.2 JUMP INSTRUCTION

The format of the BIT 480 Jump Instruction is shown in Figure 4. It is a four byte instruction. The Jump Instruction is a peripheral test only if modifier bit m_3 is set.

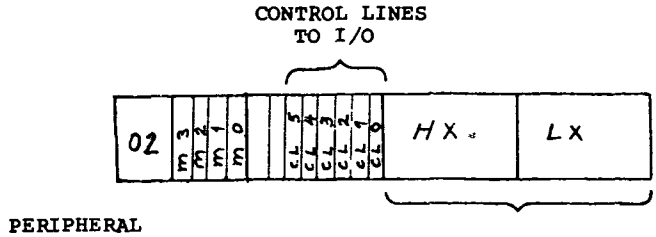


FIGURE 4 - JUMP INSTRUCTION

In the event m_3 is set, the peripheral test specified in bits CL_0, CL_1,....CL_5 is sent to all I/O devices. During a peripheral test, the following sequence of events occurs, as shown in the timing diagram of Figure 5.

An interrogate pulse, called PTJE is sent out from the computer. This informs all I/O devices that a peripheral test is in process, and that the Control Lines CL_0, CL_1.... CL_5 constitute a definition of what the test is.

One I/O device has been previously selected (as the one to be tested) by a PCI Instruction.

The selected I/O device examines the Control Lines and sends back to the computer two signals:

1. Acknowledgement of the Test
2. A signal called I/O Reply if the result of the test is a "yes" answer.

The I/O Reply line, if activated during a peripheral test, causes the program to jump to the address specified in the two bytes HX and LX in the Jump Instruction. Certain codes in the six low order bits of the modifier are reserved for specified tests as follows:

CODE	TEST
0 0 1 0 0 0	I/O Device Busy
0 0 0 0 0 1	Interrupt Requested
0 1 0 0 0 0	Channel Busy*

* This is of significance only when the machine is equipped with overlapped I/O.

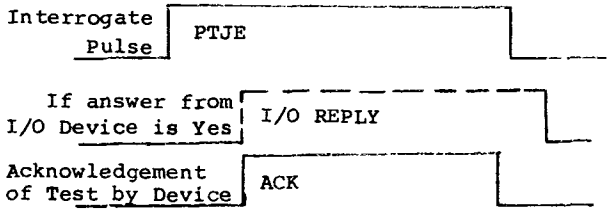


FIGURE 5 - TIMING DURING PERIPHERAL TEST

1.3.3 PERIPHERAL TRANSFER INSTRUCTION

The format of the Peripheral Transfer Instruction is shown in Figure 6. It is a two byte instruction.

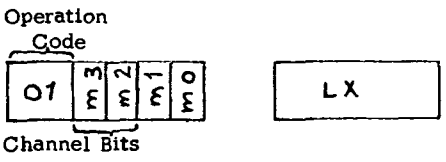


FIGURE 6 - PERIPHERAL TRANSFER INSTRUCTION

The LX address shown, together with the contents of the Page Register, constitutes the starting memory location for the transfer. The two high order m bits, m_3 and m_2, define on which channel the transfer is to take place:

m_3	m_2	CHANNEL
0	0	A
0	1	B
1	0	C
1	1	G (non overlapped)

If channel A, B, or C is indicated in the instruction, a "GO" pulse is transmitted out on the appropriate data channel. This GO pulse allows the I/O device (previously selected by a PCI Instruction) to begin the transfer in a cycle stealing mode. As soon as the GO pulse is delivered to the I/O device, the computer terminates the PTI Instruction and goes on to the next instruction.

If Channel G is specified in this instruction, a "GO" pulse is sent out to the selected device on the G Channel. However, the PTI instruction does not terminate until the data transfer is complete.

1.3.4 SUMMARY OF CONTROL LINES

Figure 7 lists all of the I/O lines discussed which the computer uses to communicate with I/O devices. These are lines which connect to all I/O devices as shown in Figure 1. They consist of the CL lines, the PA lines and other miscellaneous lines discussed in the description of the Peripheral Control Instruction, in the jump instruction, and in the Peripheral Test Instruction. They are collected in Figure 7 for easy future reference.

NAME	PURPOSE
CL_0 CL_1 CL_2 CL_3 CL_4 CL_5	(1) Definition of command during PCI (2) Definition of test during Jump
PA_0 PA_1 PA_2 PA_3 PA_4 PA_5	Defines the address of an I/O control unit during PCI
PCIE	Signifies that computer is executing PCI
PCI1	Control pulse during PCI
PCI2	Control pulse during PCI
ACK	Acknowledgement to computer during PCI or Jump
I/O REPLY	"Yes" answer during Jump
I/O CLEAR	Initial reset to all I/O devices
LOAD PULSE	Initiates loading of a program from manual control panel
IR IS	Interrupt Control Interrupt Control

FIGURE 7 - SUMMARY LIST OF CONTROL LINES

This set of lines represents all of the lines used to initiate commands and tests on I/O devices.

1.4 DATA CHANNEL LINES

Each data channel set of hardware within the computer has a set of signal functions which connect to all of the I/O control units associated with that channel. These functions are listed in Figure 8. Each function has a postscript letter X, symbolizing one of the four data channels. An equivalent set of functions is used for each of the four data channels. They have the postscript A, B, C, or G.

NAME	PURPOSE
ERX	End Record Flop (from Computer)
SERX	Set End Record (to Computer)
OX	Out (to Computer)
BFX	B-Full Flop (from Computer)
PSBFX	Peripheral Set B - Full (to Computer)
PRBFX	Peripheral Reset B - Full (to Computer)
BOX	Data bit 0 from Computer
B1X	" " 1 " "
B2X	" " 2 " "
B3X	" " 3 " "
B4X	" " 4 " "
B5X	" " 5 " "
B6X	" " 6 " "
B7X	" " 7 " "
B8X	Word Mark bit 8 from Computer
IBOX	Data bit 0 to Computer
IB1X	" " 1 " "
IB2X	" " 2 " "
IB3X	" " 3 " "
IB4X	" " 4 " "
IB5X	" " 5 " "
IB6X	" " 6 " "
IB7X	" " 7 " "
IB8X	Word Mark bit 8 to Computer
WMX	Transmit Word Marks
GOPX	Go Pulse (starts data transfer)
BUSY X	Selected Device Busy
RSCX	RESCAN from beginning of page

FIGURE 8 - DATA CHANNEL LINES

1.4.1 END RECORD

An "End-Record" flop is provided on each data channel. Its designation for Channel X is ERX. The output of this flop is transmitted to all associated I/O devices on the channel. The flop, when set, indicates that the channel is not engaged in any data transfer. The flop is then signifying a "not busy" condition.

1.4.2 SET END RECORD

A "set end record" line is provided to the I/O control units on the channel, designated as SERX in Figure 8. Only the selected I/O control unit which is involved in a data transfer may use this line. It is used by the I/O control unit to terminate the transmission.

It should be noted that only the processor can reset the end record flop, and only the selected I/O control unit can set it.

It requires a PTI Instruction by the computer to reset the end record flop. This starts the data transmission. The channel may then be said to be "busy". The transmission is stopped by the I/O control unit when it sets the end record flop. The channel is then said to be "not busy".

1.4.3 OUT

A line is provided which indicates the direction of data flow. It is signified in Figure 8 by the name OX. Only the selected I/O control unit on the channel may use this line. When OX is true, data will be transmitted out of the computer to the selected I/O device. When OX is not true, data will flow from the I/O device to the computer.

1.4.4 B-FULL

Each data channel has a flop, called the "B-Full Flop". This flop, when set, indicates that a valid data character resides in the B-Register of the channel. When the flop is reset, it indicates that the data character has been taken and the B Register is cleared. For the Channel functions listed in Figure 8, this flop is designated as BFX. A set function PSBFX and a reset function PRSBFX, are also provided on the channel. Only the selected I/O control unit can use these two lines to set and reset the B full flop.

1.4.5 DATA REGISTER LINES

As shown in Figure 8, there are data output lines corresponding to the bits of the B register, designated as:

B0X, B1X, B2X, B3X, B4X, B5X, B6X, B7X, B8X

In addition, there are data set lines for the B register:

IB0X, IB1X, IB2X, IB3X, IB4X, IB5X, IB6X, IB7X, IB8X

These lines enable the selected I/O device to load data into the B register. The line IB8X is used for entering word-marks during transmission to the computer.

During transmission out of the computer, word marks are always loaded into the B register flop B8X by the computer. They may be ignored or accepted by the selected I/O device depending upon whether the computer has called for transmission with word marks in the previous PCI Instruction.

1.4.6 WMX

A line called WMX communicates with all I/O control units on the data channel. Only the selected I/O controller may activate the line. If it is activated during transmissions INTO the computer, it causes word marks to be entered when B8X is set. If B8X is not set when a character is loaded into the B register, the effect is for a word mark to be erased in memory at the time of transfer between core and the B register.

1.4.7 GOPX

This line from the computer emanates to all I/O control units on the channel. It carries a "GO" pulse during a PTI Instruction. Only the selected I/O device will respond to this "GO" pulse. This pulse signifies that the address register of the channel is loaded with the proper starting memory location and the "End Record" flop is reset.

1.4.8 BUSY

This is a control line from the I/O devices to the Data Channel hardware. Only the I/O device which is selected for data transmission on the channel can activate this line. The purpose of this line is to enable the I/O device to stall the computer during a PTI Instruction if the I/O device is not yet through with a previous PTI Instruction. What occurs if this line is true is that the computer will not send a GO pulse until it recognizes the BUSY line as being inactive.

1.4.9 RSC

This stands for "Rescan Channel". This line is only used by the selected I/O device during a PTI Instruction. It enables the I/O device to reset the channel address register back to the beginning of page. This enables a card-punch control unit to rescan a card buffer area twelve times during punching of a card without the requirement of twelve instructions. It likewise can be used by a printer control unit to effect rescan of a print area in memory once for each character of the alphabet.

1.5 SIGNAL LEVELS

Signal levels to or from the computer are at one of two voltage levels:

"1" level $\geq$ 3.0 volts

"0" level = 0 volts

For each line from the computer, a maximum of two loads may be placed on it by an I/O control unit. Each load is assumed to consist of a diode gate with a 2.1K resistor to +4.5 volts. A maximum of six control units may be loaded onto a data channel. Repeater drivers are necessary on BIT 480 output lines in situations where more than 6 control units are to be driven.

Lines which are to communicate with the computer should be driven, at the I/O device by a Signetics SP659A Buffer/Driver, or equivalent. This circuit has a low impedance at both its logic levels.

It is necessary to "OR" some of the outputs from the control units together into one line (since for many of the lines, only the selected I/O unit may use it).

The technique for doing this is shown in Figure 11. A diode should be wired in series with the output of each buffer/driver. The termination is provided at the computer.

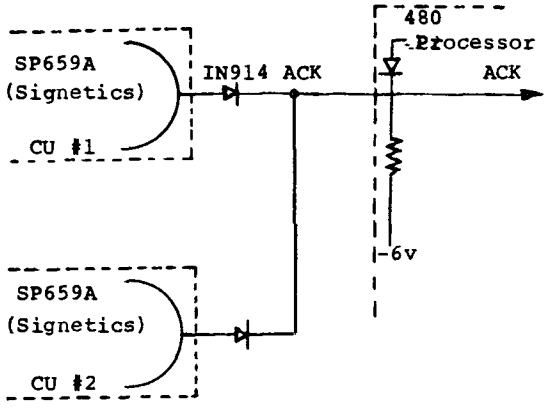


FIGURE 11 - PROCEDURE FOR "OR"ING CU LINES
(Illustrated for "ACK" Line)

1.6 INTERLOCKS

A number of features are provided in the computer which prevents certain types of programming errors in using peripheral instructions.

If an I/O control unit receives a PCI Instruction which would conflict with an operation already in process, then the I/O device, by not sending ACK, (Acknowledgement), will cause the computer to mark time until the present I/O operation is complete.

If the computer initiates a PTI Instruction on a data channel which is "busy" (End record flop is reset), the computer will mark time until the end record flop of the channel is set, signifying that the channel is no longer busy. The machine will then complete the execution of the PTI Instruction.

If, during a PTI Instruction, the computer finds the "device busy" line (Busy) active, it will stall until this line goes false before sending out a GO pulse on its channel.

In summary, either the "end record" flop reset or the BUSY line active will stall the computer during a PTI Instruction. The PTI will only transmit a GO pulse when both....

1. "End Record" flop set
2. BUSY inactive

1.7 TRAFFIC CONTROL

The traffic control system of the BIT 480 Processor consists principally of four flops within the main frame called TO, TFA, TFB and TFC, which collectively are called the Traffic Register.

The TO flop corresponds to the "compute" state. When in this state, the machine is running its program and there are no I/O data transfers taking place. The flops TFA, TFB, and TFC correspond to three peripheral states in which data transfers occur between the computer and program selected I/O devices. These three flops correspond to three distinct data channels, A, B, and C, which are overlapped. When the TFA flop is set, a data character is transferred on Channel A. A similar situation holds for the other two peripheral state flops.

No more than one of the four flops may be on at a given time. In other words, the computer is either in the TO state, or in one of the three traffic states. There is logic on each of the four flops which control what the next traffic state shall be.

In Figure 12 is shown a flow chart of the traffic control register. Note that, when in the TO state, the machine can jump to any of the other three states to process a "frame demand".

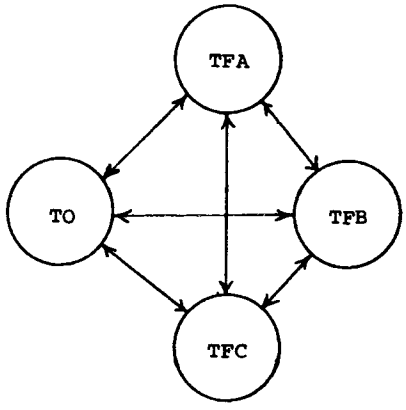


FIGURE 12 - FLOW CHART OF TRAFFIC REGISTER

1.7.1 TRAFFIC CONTROL RULES

1.7.1.1 If, when in the TO state, frame demands occur on more than one data channel simultaneously, then they are processed in the order A,B,C. For example, if the machine were in the TO state and the frame demands occurred on all three channels simultaneously, then the machine would jump first to the TFA state and process its demand. It would then jump successively to states TFB and TFC, processing the respective demand in each state.

1.7.1.2 The processor, when in one of its three peripheral states, does not return to the compute state as long as there is a frame demand to process. In other words, once in any of the three states A, B, and C, the computer will keep jumping back and forth between them until all demands are satisfied before returning to the TO state.

The decision of selecting which state to jump to is made at the end of processing of a frame demand. If, at this time, other frame demands exist, another peripheral traffic state will be entered. If, however, no frame demand exists at this time, the processor will jump back to the compute state (TO).

The return to the TO state can be made from any of the three I/O traffic states.

1.7.1.3 The processing of demands in the peripheral states proceeds in a clockwise direction.

For example, if frame demands are continuously being made on all three peripheral states, the order of processing will be A, B, C, A, B, C, A, B.....

1.7.1.4 In the processing of demands, only those states are entered in which a demand exists. For example, if states A and C are receiving continuous demands, the order of processing will be A, C, A, C,.....

1.7.1.5 The processor cannot remain in one peripheral traffic state for <u>two successive</u> frame demands if there is a demand on another channel.

For example, assume that the machine is processing a frame demand on Channel A (TFA flop set). Suppose that as soon as the demand is processed, another frame demand occurs on Channel A immediately before the decision is made selecting the next successive state. There are two possibilities for action here depending upon whether or not there simultaneously exists a demand from another channel:

a. If no demand exists on another channel, the machine will process the new demand on the A Channel. In other words, it will stay in the TFA state for another process time.

b. If, in the TFA state, there is a demand from another channel, in addition to a new immediate demand on the A Channel, the processor will jump to the <u>other</u> channel state and process the other demand before returning to the A state.

2. INTERRUPT

2.0 GENERAL

All BIT Processors are equipped with interrupt capability. By interrupt is meant the ability of an I/O device to derail the central processor from its main program into a subroutine. Typically, the subroutine effects a data transfer with the processor. Upon completion of the subroutine, the processor returns to the main program.

2.1 PRESERVATION OF MACHINE STATE FOR RETURN FROM SUBROUTINE

The central processor has two modes of operation:

Normal Mode - corresponds to execution of main program;

Interrupt Mode - corresponds to execution of interrupt subroutine.

A flip-flop within the computer, called the IS flop, defines which of these two modes is in operation. During Normal Mode of operation the IS flop is reset. The IS flop is in the set state throughout the duration of an interrupt subroutine.

Within the computer there are a number of flip-flops and registers which constitute the Normal Mode "configuration" of the machine. During the initiation of the interrupt subroutine, part or all of this "configuration" must be preserved for later return to Normal Mode so that the subroutine may be permitted to be completely general in its function. The "configuration" of the computer may be defined as consisting of the following internal elements:

GR flop, EQ flop - these two flops are conditionally set at the end of arithmetic operations indicating whether the result of the operation is greater than, or equal to zero. Upon initiation of the Interrupt Mode, the Normal Mode state of these flops is automatically preserved in backup flops, by means of hardware.

Accumulator - the Accumulator contents constitutes part of the configuration of the machine. This is not automatically preserved by hardware action during the entrance into the Interrupt Mode. However, it can be preserved by program action during the interrupt subroutine and restored before the end of the interrupt subroutine.

Page Register - the Page Register is an essential part of the configuration of the computer during the Normal Mode. Like the Accumulator, this is not saved by hardware action but may be preserved and restored during the interrupt subroutine by means of the Page Instruction.

Program Counter - the Program Counter, during Normal Mode, contains the memory address of the next Normal Mode Instruction. This register is automatically preserved during entrance to the Interrupt Mode by hardware action.

2.2 INITIATION OF THE INTERRUPT

During Normal Mode the computer tests for the existence of an interrupt request on a common I/O line called the PIR just before each instruction fetch. Upon detection of a request, the machine causes the program counter to be copied into the third and fourth address positions of the highest page of memory. This program counter address, which is stored, is arranged by the interrupt program to be the address portion of a "Return Jump from Interrupt" instruction which restores the machine to the Normal Mode, and resumes the main program. The diagram of Figure 13 illustrates this function.

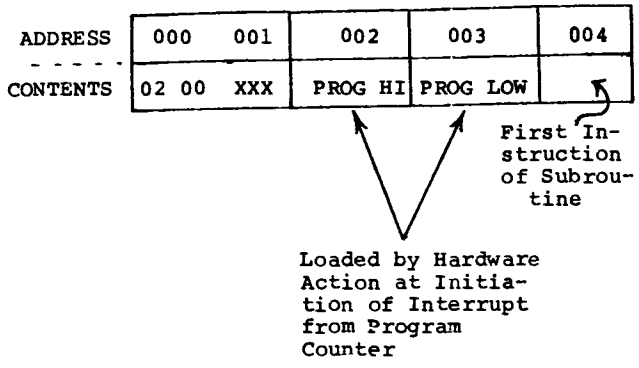


Figure 13

It will be noted from Figure 13 that following the "Return Jump from Interrupt" instruction, the next instruction is the first instruction of the interrupt subroutine. The hardware which causes the loading of the program counter contents into the highest page also arranges that the next instruction to be fetched is the first instruction of the interrupt subroutine. This instruction should be a Store Page Instruction, since the Page Register is generally referenced to a different part of memory.

Several other functions are performed by hardware during the storage of the program counter contents. The states of the GR and EQ flops are stored in back up flops. These states are restored at the end of the interrupt subroutine, during the Return Jump from Interrupt instruction. The IS flop (Interrupt State) is set during the storage of the program counter contents in the highest page of memory. This flop accomplishes two purposes: it prevents any further interrupt requests from being sensed during the execution of the interrupt subroutine, and its state is sent out to all I/O devices where it temporarily disconnects from the data channel lines any I/O device which has previously been selected for data transfer or peripheral test. This effectively frees all devices on the channel, during the IS state, for either tests or data transfers.

2.3 INTERRUPT SUBROUTINE

The interrupt subroutine must perform the function of locating the device which is interrupting, assuming that more than one I/O device has the interrupt capability.

The subroutine will make peripheral tests on successive I/O devices until it locates the first one requesting service. The manner of testing an I/O device is as follows:

First - a PCI instruction to address and select an I/O device for test.

Second - a peripheral test variety of the Jump instruction which effects a program jump if the I/O device has its Interrupt Request flop set.

The interrupt system can be programmed to be either a priority system or one in which devices are sequentially serviced. The choice is an option for the programmer.

Upon identifying the interrupting device, the subroutine executes service on it - typically a data transfer, although other peripheral operations are possible. The peripheral data transfer is accomplished by program as follows:

First - a PCI instruction which sets, within the I/O control unit, a flop called IDS (Interrupt Data Select). This flop is similar to the Data Select flop which is set during Normal Mode operation, however as mentioned previously, the IS flop temporarily has disengaged all Data Select flops on the channel.

Second - a PTI instruction causes execution of the data transfer, which may be either a single character, or a block.

When the peripheral operation is complete, the interrupt subroutine must then return the machine to Normal Mode Operation. This is performed as follows: a change page instruction restores the original page register contents which existed before the interrupt. The last instruction required is Return Jump from Interrupt, stored at the beginning of the highest page of memory, which restores the GR and EQ flops, turns off the IS flop, and causes a program jump back to the main program. If there are no other I/O devices requesting interrupt, the next instruction fetch will be for the main program. However, if at the end of the Return Jump from Interrupt, there are other devices requesting service, the machine will re-enter the IS state immediately.

2.4 INTERRUPT CONTROL LINES FROM BIT 480

All peripheral commands and data flow utilize the same set of I/O lines whether the computer is in the normal Mode or the Interrupt Mode.

A few additional signal functions are provided with all BIT 480 Processors for servicing of interrupts. These functions, which are listed below, communicate with all devices on the data channel.

IS - this function indicates to all I/O devices that the Central Processor is operating in the Interrupt Mode.

IR - this function, which means "Interrupt Request", signifies that some I/O device on the channel is requesting service.

3. CONTROL UNIT DESIGN

3.1 GENERAL

The connection of an input/output device to the BIT 480 Processor requires the design of a set of logic called a control unit. The control unit has two broad functions to perform, as follows:

1. The control unit must be capable of recognizing various commands issued by the computer. The control unit must recognize that it is the device that the computer is commanding and must store the given command. For some instructions, it will respond immediately. In other cases, it must wait to receive a "GO" pulse from the computer. The logic to perform these functions is essentially the same for all control units.

2. The control unit must have logic for operating its I/O device in several modes, to conform with the commands of the computer. The logic to do this will in general be different among various kinds of control units.

3.2 DESIGN of a NON-INTERRUPTING CONTROL UNIT

This section describes the essential logic which should be provided in an I/O control unit so that it can communicate with the BIT 480 Processor.

There are several cases presented in this section, corresponding to the various kinds of I/O situations to be implemented. The most common situation is one in which data is to be transferred to and from the computer. Another possibility might be the desire to perform a peripheral test on the I/O device. A third situation could be the execution of a Class I instruction (Section 1.3.1.2.1).

The logic diagrams that are presented in this section are divided into four parts:

1. Basic Dialogue Logic (all control units) Figure 14
2. Data Transfer Logic (optional) Figure 15, Figure 16
3. Peripheral Test Logic (optional) Figure 17
4. Class I Execute Logic (optional) Figure 18

Generally, it is necessary to have some subset of this logic in an I/O Control Unit. All control units should contain the Basic Dialogue Logic. In addition, the control unit may contain one or all of

Data Transfer Logic (Figure 15, Figure 16)

Peripheral Test Logic (Figure 17)

Class I Execute Logic (Figure 18)

Timing diagrams for the different types of BIT 480 peripheral operations is shown in Figures 19 and 20.

3.2.1 BASIC DIALOGUE

The logic for basic dialogue is shown in Figure 14. This shows the logic for decoding of a peripheral address and the logic for sending the ACK (Acknowledge) function back to the BIT 480 Processor when it receives a PCI instruction.

As shown in Figure 14, the PA (peripheral address) lines arriving from the BIT 480 are decoded to generate a function called PUA (peripheral unit addressed).

The logic diagram shows the PA lines going through a jumper card before being decoded. This is useful if it is desired to be able to modify the

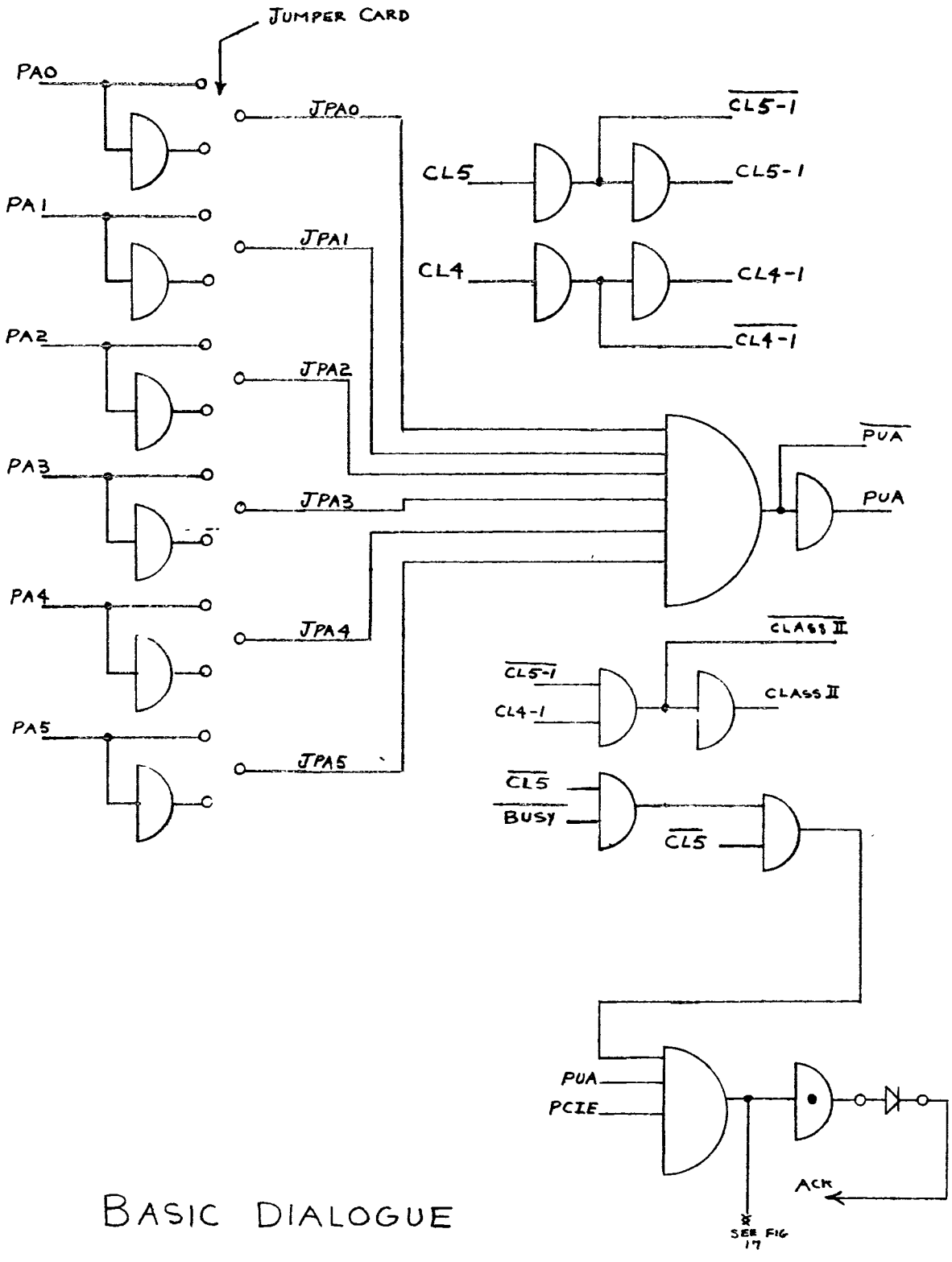


FIG 14 - STANDARDIZED CONTROL UNIT LOGIC - PAGE 1

address of an I/O device at a later time. Inverters are shown as an indication that they are necessary where negations of PA lines are required to generate PUA.

Figure 14 shows the generation of the ACK function. By means of this function the I/O device replies to the computer when it is given a PCI command. ACK is to be transmitted to the computer during PCI instructions as follows:

ACK = PCIE . PUA(BUSY[Class I + Class II] + Class III + Class IV)

In terms of the CL lines, this can be expressed as:

$$ACK = PCIE \cdot PUA\ (\overline{CL_5} \cdot BUSY + CL_5$$

3.2.2 DESIGNING FOR DATA TRANSFER

Figures 15 and 16 show the additional control unit logic which is necessary to process data transfers. It will be recalled that a PCI instruction selects an I/O device for a subsequent data transfer, and a PTI instruction executes the transfer. The I/O control unit requires two additional flops to process such commands.

3.2.2.1 Data Select Flop

A flop called DSF (Data Select Flop) in the control unit is used to remember that the device is selected during the PCI instruction for subsequent data transfer. This flop is either set or reset by a Class II PCI instruction.

Only one I/O device on the channel may have its DSF flop set. All other devices on the channel must have their DSF flops reset. The action of a Class II PCI instruction is to set one DSF flop in the selected device and to reset DSF flops in all other devices on the same data channel.

The set logic on the DSF flop in the chosen device is:

$$\text{Set DSF} = PCI_2 \cdot \text{Class II} \cdot \overline{IS} \cdot PUA$$

The reset logic on all other DSF flops of the channel in unselected control units is:

$$\text{Reset DSF} = PCI_1 \cdot \text{Class II} \cdot \overline{IS} \cdot \overline{PUA} \cdot \text{CHANNEL SELECTED}$$

It will be recalled that PCI_1 and PCI_2 are timing pulses supplied from the computer during PCI instructions. The function $\overline{IS}$, indicates that the computer is not in the interrupt mode.

3.2.2.2 Out Flop

Some I/O devices must be capable of both transmitting and receiving data from the BIT 480. In this case, a flop called OUT is required. It is set and reset with somewhat the same logic as the DSF flop. The meaning of the OUT flop being set is that data transmitted by PTI instructions is to flow "out" of the computer. Otherwise, if reset, the meaning is that data flows "to" the computer. When the OUT flop is used, one bit of the PA lines (PA_3) indicates that the OUT flop is to be set. The logic functions involved are:

$$\text{Reset OUT} = \text{Set DSF} \cdot PA_3$$

3.2.2.3 Busy Flop

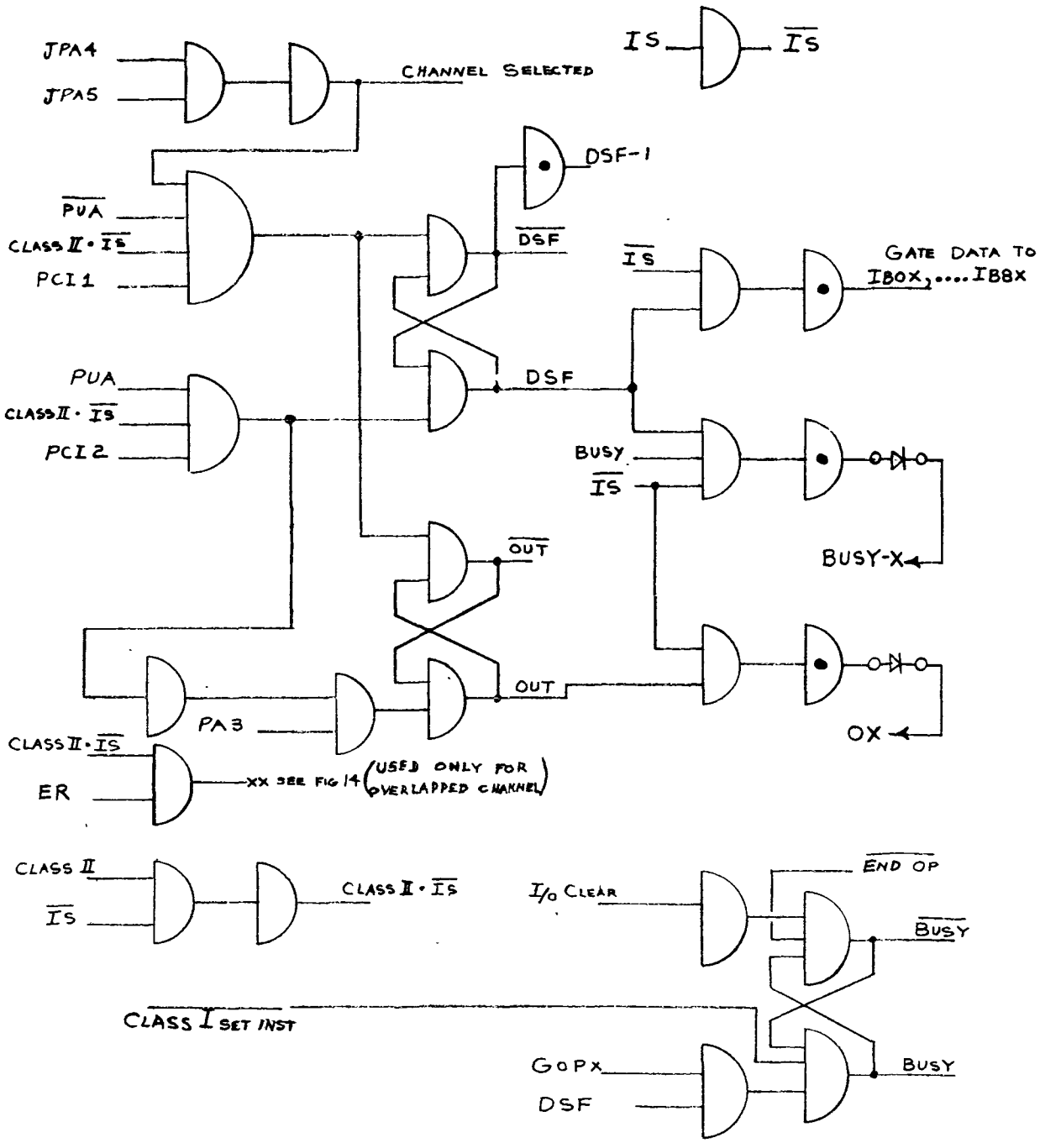
Figure 15 shows a flip-flop called BUSY. This flop is set by the GO pulse which occurs at the beginning of a PTI instruction if the device has been chosen for a data transfer (DSF flop set).

When the data transfer is complete, the BUSY flop is reset by a function called END OP (end of operation) which must be supplied by the I/O device. Also shown as a reset on the BUSY flop is a line called I/O CLEAR which is supplied by the computer. The I/O CLEAR function is generated when the master clear button is depressed on the BIT 480 control panel.

The set and reset functions for the BUSY flop are:

Set BUSY = GOPX · DSF

Reset BUSY = END OP + I/O CLEAR



DATA TRANSFER

FIG 15 - STANDARDIZED CONTROL UNIT LOGIC - PAGE 2

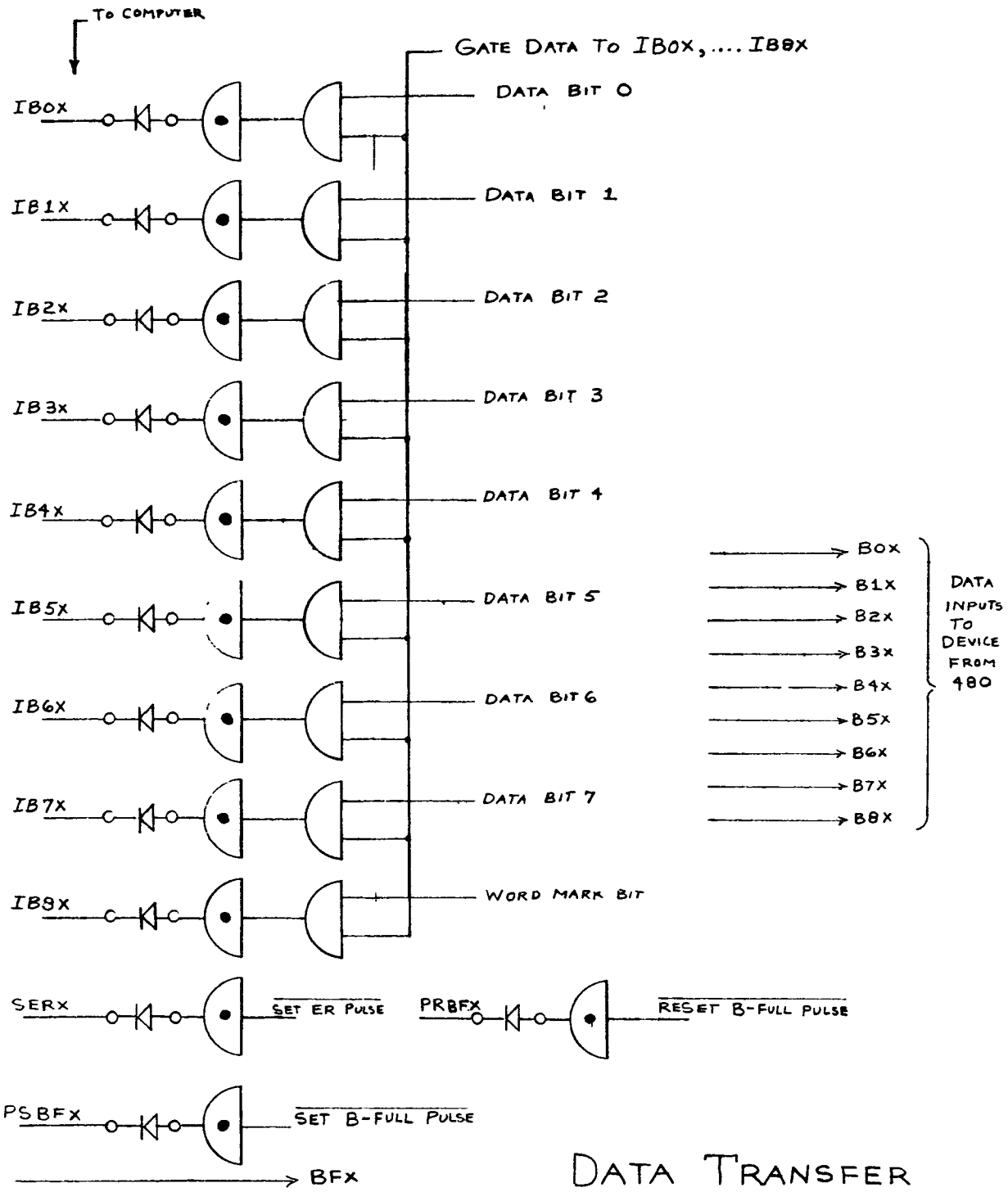
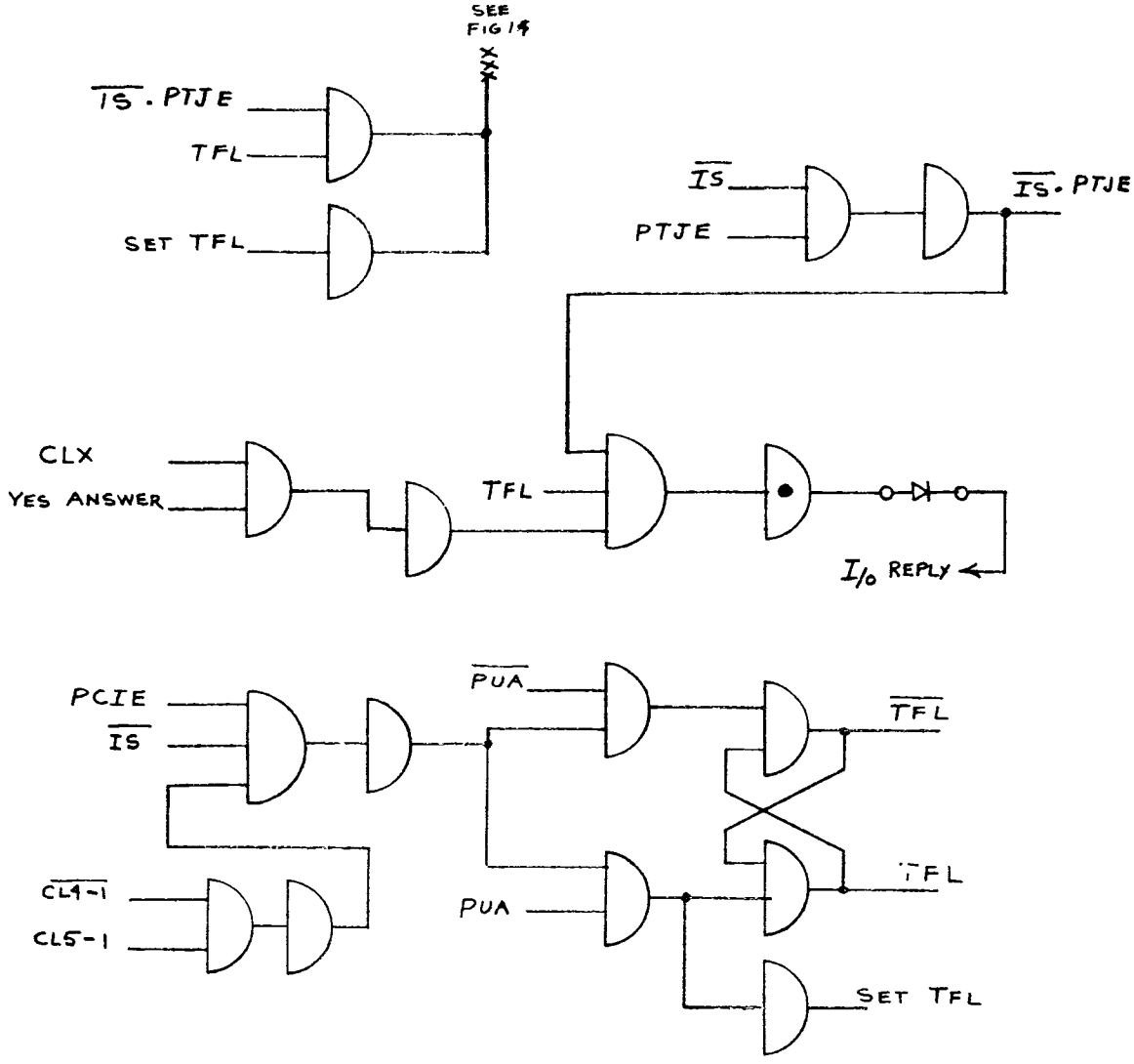
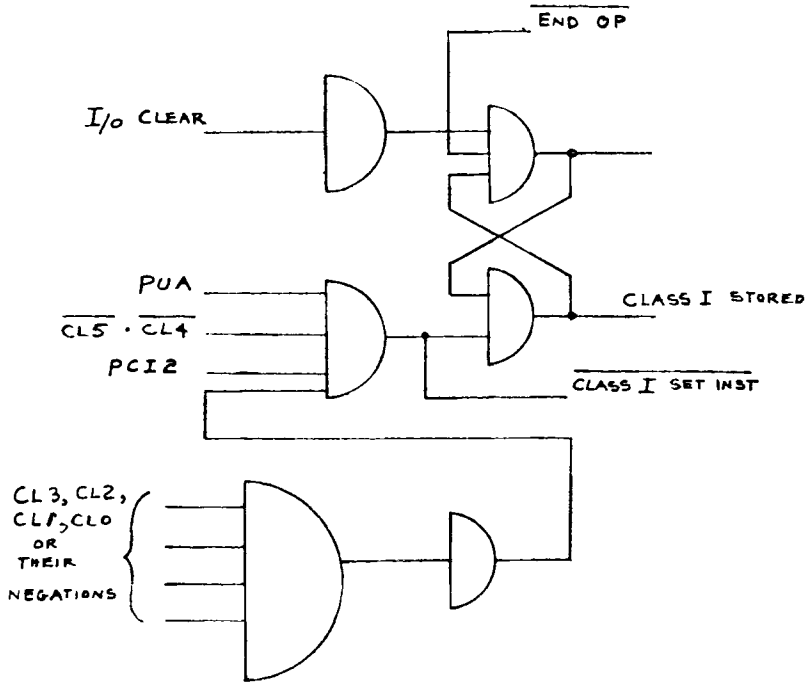


FIG 16 - STANDARDIZED CONTROL LOGIC - PAGE 3



PERIPHERAL TEST

FIG 17 - STANDARDIZED CONTROL UNIT LOGIC - PAGE 4



CLASS I

FIG 18 - STANDARDIZED CONTROL UNIT LOGIC - PAGE 5

3.2.2.4 Busy Line

Each data channel has a line called BUSY-X by means of which it informs the BIT 480 that a data transfer is in process. The generation of this function is shown in Figure 15. Its logical expression is:

$$\text{BUSY-X} = \text{DSF} \cdot \text{BUSY} \cdot \overline{\text{IS}}$$

The function BUSY is of course the BUSY flop of Figure 14. The negation of IS is used to disengage the line during an interrupt.

The main purpose of the BUSY-X line is to prevent the piling of successive PTI instructions by means of an interlock within the BIT 480, on a selected device if the device is not yet through with execution of a PTI. The BUSY-X line, if active, delays the commencement of a PTI instruction until BUSY-X goes off. Thus the BIT 480 stalls momentarily if it initiates a PTI instruction while the channel is still busy.

The line OX is the channel OUT line. It is generated from the OUT flop and $\overline{\text{IS}}$, so that this line is liberated for use by another device during an interrupt.

3.2.2.5 Data Lines to Computer

In the event that an I/O device is to transmit data to the computer, it must have access to the data set input lines to the 480, (IBOX,.....IB8X). These lines are shown in Figure 16. Some kind of gating is required before these outputs so that only the device whose DSF flop is set can have access to these lines. A master gating function, used in Figure 16, is generated in Figure 15 to gate data bits to the 480. Its expression is

$$\text{GATE DATA TO IBOX,... IB8X} = \text{DSF} \cdot \overline{\text{IS}}$$

The negation of the IS function is employed so as to free up the lines when another device on the channel interrupts.

Also shown in Figure 16 are output buffers for three functions which control entry and exit of data from the 480.

3.2.2.6 PSBFX

This function (peripheral set B-Full) is used to load data into the BIT 480. The data to be loaded, as shown in Figure 16, is impressed on lines IBOX,.....IB8X. A pulse PSBFX supplied from the I/O device causes the data to be strobed at the BIT 480 and loaded into it.

The BIT 480 replies to this action by making logically true the line BFX (B-FULL). The BIT 480 will keep BFX true until it is able to accept another frame of data, at which time it will cause BFX to go logically false. The timing diagram of Figure 19 illustrates this process.

When the last frame is to be loaded into the BIT 480, the I/O control unit should cause a pulse to be impressed on the SERX line ("Set End Record").

This pulse indicates to the BIT 480 that the last data frame is being received and permits the BIT 480 to end the PTI instruction.

3.2.2.7 PRBFX

This function ("Peripheral Reset B FULL") is used during data transfers from the BIT 480 to the I/O device.

It indicates to the BIT 480 that the I/O device has received the data. It is also interpreted by the 480 as a request for another data character.

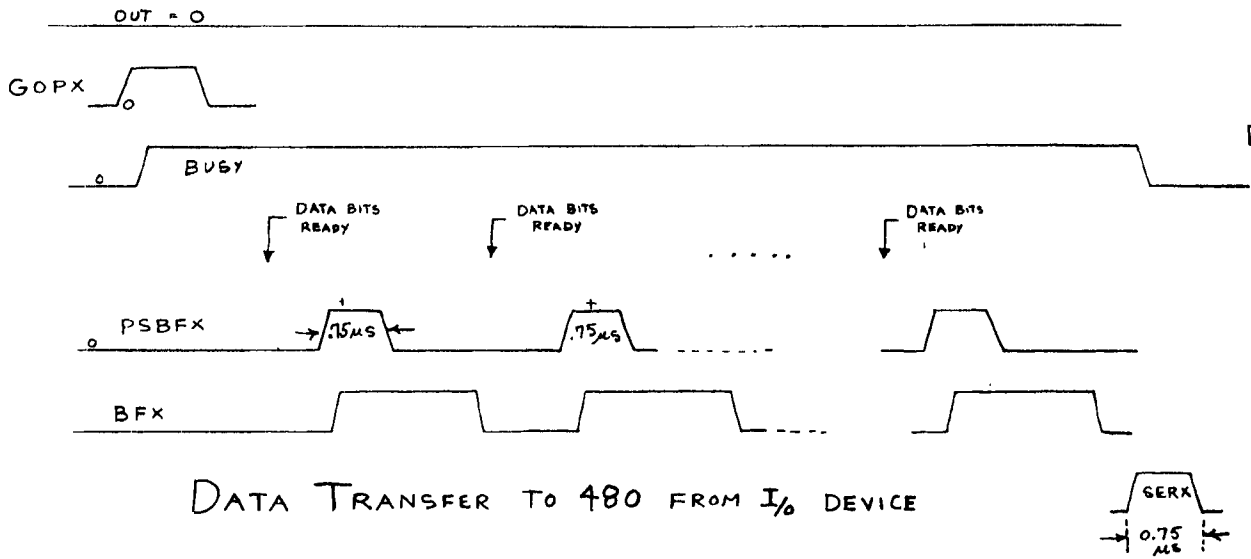
3.2.3 PTI TIMING DIAGRAM

Figure 19 shows timing diagrams which represent events as viewed by the I/O device which participates in a data transfer.

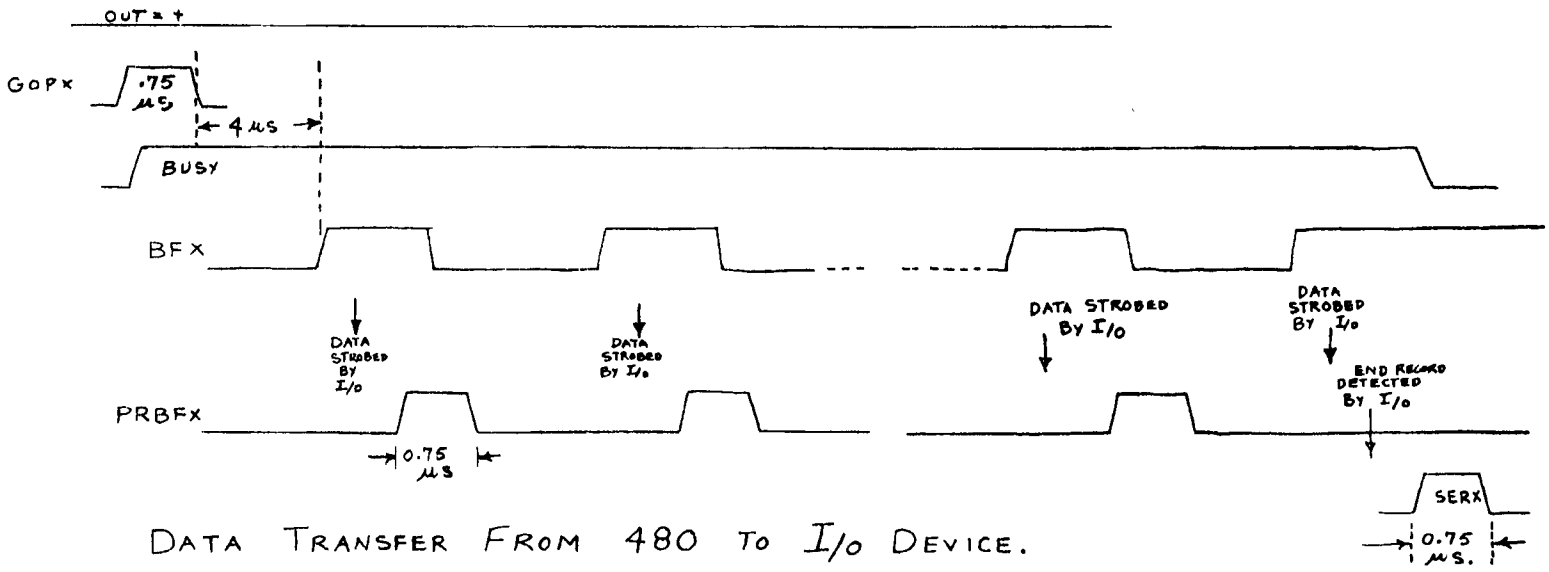
3.2.3.1 Out Transfers

In Figure 19 is shown the sequence of events which occur when data is transmitted out of the BIT 480 to a peripheral device. The action is commenced by a GO pulse from the BIT 480 which occurs early in a PTI instruction.

FIG 19
DATA TRANSFER
TIMING



DATA TRANSFER TO 480 FROM I/O DEVICE



DATA TRANSFER FROM 480 TO I/O DEVICE.

This sets the BUSY flop in the I/O device which has been selected by a PCI instruction. Approximately 4 microseconds after the trailing edge of the GO pulse, the first frame of data is presented by the BIT 480 on its data output lines to the I/O device. The I/O device recognizes the existence of data by the fact of the B-FULL flop, BFX, being set. The I/O device strobes the data character and calls for another character by resetting the B-FULL flop with the channel reset line PRBFX. This should be a pulse whose duration is 0.75 microseconds.

The resetting of the B FULL flop by the I/O device determines the data transfer rate since the BIT 480 will keep data on the data lines until BFX is reset.

As shown in the diagram, successive characters are called for by the I/O device until, on some frame, it detects the end of the record (either by word mark or some special character code). Upon detection of the last character, the I/O device instead of resetting BF, impresses a "set end record" pulse on the SERX channel line. This allows the computer to internally terminate the PTI instruction.

3.2.3.2 In Transfers

In Figure 19 is shown the sequence of events which occur during a data transfer from an I/O device to the BIT 480.

The action commences when the I/O device receives a GO pulse from the BIT 480 which signifies that it is executing a PTI instruction and is ready to receive data. The I/O device has previously had its data select flop set by a PCI instruction.

The I/O device may begin transmission to the BIT 480 any time after the trailing edge of the GO pulse. It transmits a frame of data by placing the data bits on the BIT 480 input data lines (IBOX,.....IB8X) and pulsing the PSBFX line. This causes the frame of data to be loaded into the BIT 480. As evidence of the loading, the BFX line from the BIT 480 becomes true. The BIT 480 then executes a memory cycle which stores the character. As soon as the BIT 480 has processed the loading, it resets the BF flop which indicates to the I/O device that it is ready to accept another frame.

The I/O device loads the next frame by pulsing the PSBFX line, and each time the computer signifies its readiness to receive new frames by turning off the BFX line.

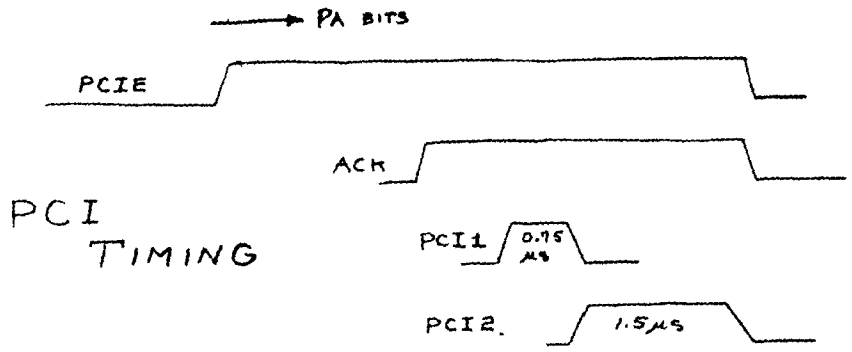
When the I/O device has detected the end of the record, it concludes the data transfer by applying a pulse to the SERX line. This tells the BIT 480 that no further data is to be transmitted and allows the BIT 480 to terminate the PTI instruction. This pulse may also be used by the I/O device to turn off its BUSY flop.

3.2.3.3 PCI and Test Timing

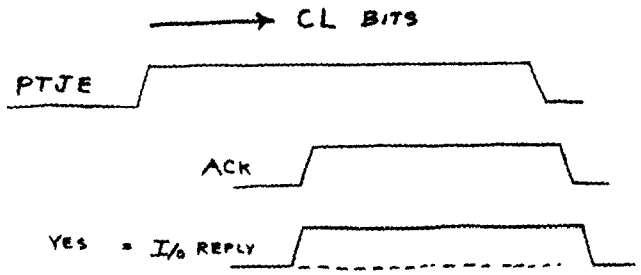
Figure 20 shows timing for two other situations of interest, the PCI instruction, and the Jump instruction which causes a peripheral test.

In the case of the PCI instruction, it will be noted that the BIT 480 sends out a function called PCIE. The address of the I/O device to receive the instruction is placed, by the BIT 480, on the PA lines. After receiving the acknowledge function from the I/O device (ACK), the BIT 480 sends out two pulses, PCI_1 and PCI_2 as shown in Figure 20.

In the case of a Peripheral Test, the BIT 480 sends out a function called PTJE, and a definition of the test on the CL lines. The selected I/O device activates the acknowledge line and supplies a "yes" reply on the I/O REPLY line.



PCI TIMING



PERIPHERAL TEST TIMING

FIGURE 20

3.3 DESIGN OF INTERRUPTING CONTROL UNIT

The "interrupting" I/O device requires essentially the same control unit logic as a non-interrupting device, the only difference being some minor modifications.

The interrupting control unit will require the BASIC DIALOGUE logic of Figure 14, regardless of the function to be performed. In addition, it will require logic to be described presently.

It will be recalled that the BIT 480 operates in either of two modes -

Normal Mode ($\overline{IS}$)

Interrupt Mode (IS)

An interrupting device will generally perform during the Interrupt Mode. A non-interrupting device will perform during Normal Mode.

In the general case, the interrupting device is required to participate in a data transfer with the BIT 480. It also must be capable of being "tested" by the BIT 480 (if there is more than one device which can interrupt), since it is by means of peripheral tests that the BIT 480 determines which device is requesting service.

3.3.1 INTERRUPT REQUEST FLOP

There is a common line emanating from the BIT 480 called "Interrupt Request" (IR). Any I/O device which requests interrupt service should make this IR line active. Activation of this line will cause the BIT 480 to go into the interrupt mode before the next instruction fetch.

To activate the IR line, the interrupting I/O device should have an interrupt request flop (IRF). This flop is set by the I/O device when it requests service. When service has been completed, the flop is to be reset.

set IRF = Req Service Pulse

reset IRF = Service Complete Pulse

The IRF flop is shown in Figure 21

3.3.2 INTERRUPT TEST FLOP

In the event that more than one device can interrupt, the BIT 480 determines which one is interrupting (during the interrupt mode) by means of Peripheral Test and Jump Instructions. In order to test a device to see if it is requesting service it is first necessary to select it for test by means of a Class III PCI Instruction and then to test it by means of a Jump Instruction.

The interrupt test flop (ITF) functions, during interrupt mode, in the same manner as the TFL flop of Figure 17, for a non-interrupting control unit. The interrupt test flop is shown in Figure 21. The set and reset functions on this flop are:

set ITF = PCIE·IS·CLASS III·PUA

reset ITF = PCIE·IS·CLASS III·$\overline{PUA}$

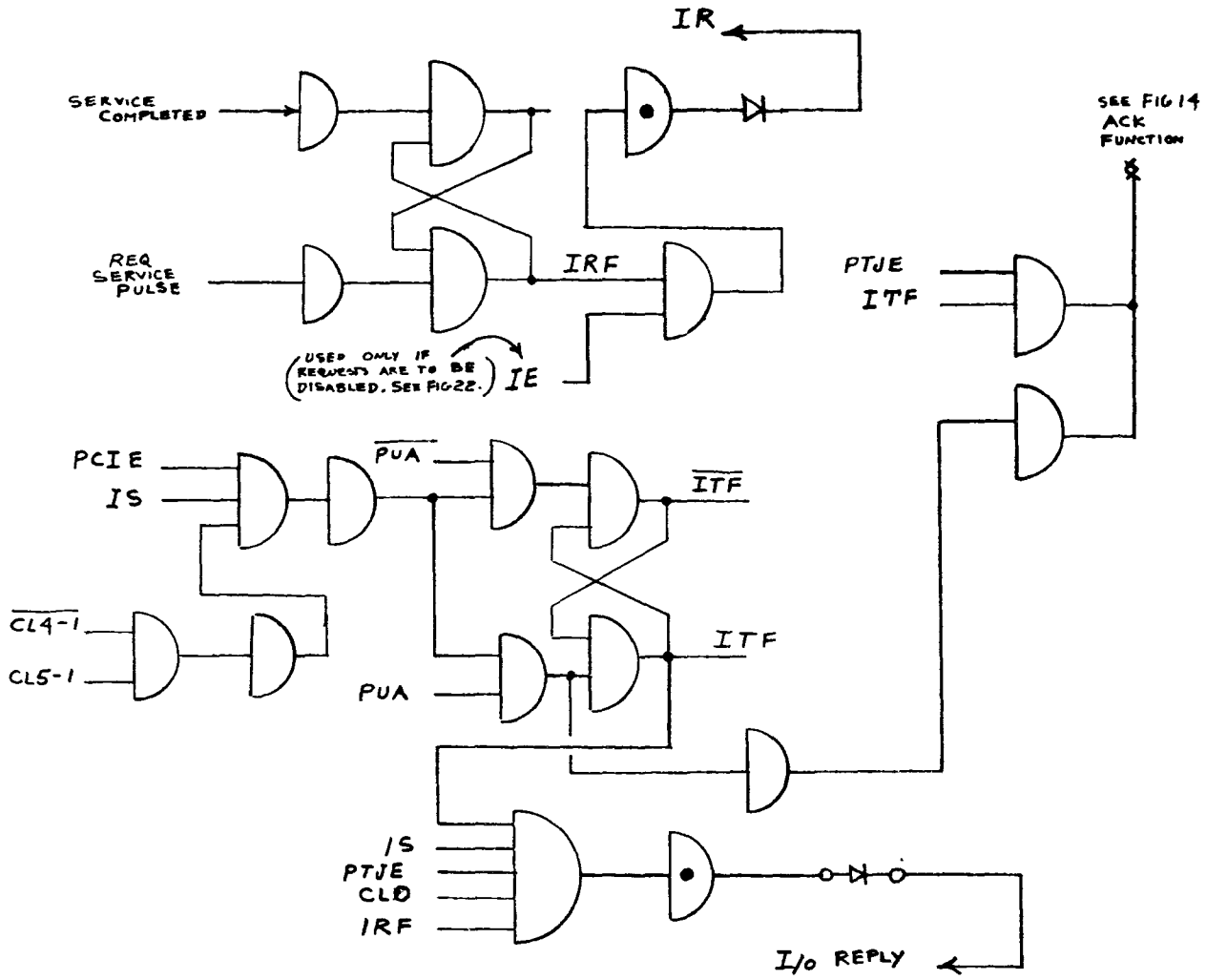
Note that the selection of the ITF flop can only occur during the interrupt state.

3.3.3 I/O REPLY FUNCTION

The I/O REPLY function is activated during a peripheral test in an interrupt subroutine if the interrupting device has its IRF flop set. In other words, the test which is to be performed, in Figure 21, is a test whether or not the IRF flop is set. The logic statement for I/O REPLY is:

I/O REPLY = PTJE·IS·CLO·IRF

(It will be recalled that in Section 1.3.2 the CLO bit defined the test as being for the presence of an Interrupt Request)



REQUEST SERVICE AND TEST

FIG 21 - INTERRUPT HARDWARE

3.3.4 LOGIC FOR DATA TRANSFER

The logic for executing a data transfer to or from the I/O device on an interrupt basis is essentially the same as is shown in Figure 15 and Figure 16, for the non-interrupting case.

Only one slight change need be made. On Figure 15, in every case where $\overline{IS}$ is used, it should be replaced by the function IS.

Timing for peripheral functions in the interrupt mode is the same as for the non-interrupt case (Figures 19 and 20).

3.3.5 INTERRUPT DISABLING

An additional control flop is required in the I/O control unit if it is desired to enable and disable interrupt requests from being serviced. This flop is called "interrupt enable" (IE) and is shown in Figure 22.

It is used to gate the IRF flop onto the IR bus. The IE flop is set and reset by Class IV PCI Instructions as described in Section 1.3.1.2.4.

The set and reset functions are:

set IE = PCI2·CLASS IV·PUA·$\overline{CLO}$

reset IE = PCI2·CLASS IV·PUA·CLO

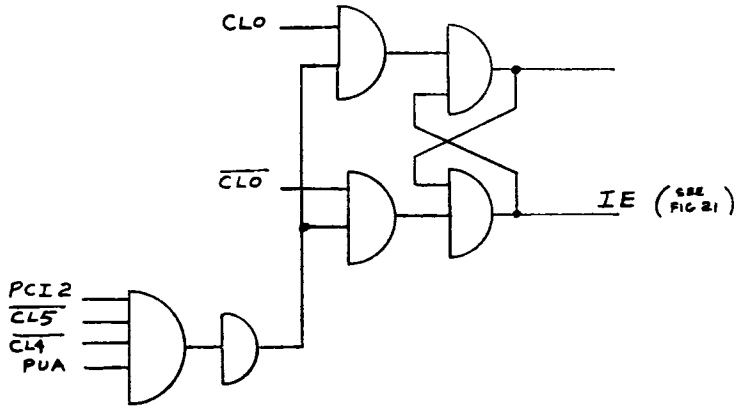


FIG 22 - INTERRUPT DISABLE

4. CABLE

4.1 I/O CABLE

The BIT 480 Processor comes equipped with an I/O cable for each data channel. In the simplest case, for a BIT 480 with just the standard G channel, one cable is provided. This cable terminates in two cable cards which are designed to insert into a pair of 44 pin connectors (.156" spacing) at the first I/O device to be interfaced.

In connecting to a number of I/O control units, the cable should be continued in the manner shown in Figure 23, by means of extension cables, so that output lines from the BIT 480 are common to all I/O devices on the same channel.

4.1.1 CABLE CARD PIN NUMBERS

The numbering of the pins on the cable card is illustrated in Figure 24. With the card oriented as in the figure with the large notch at the right, pin 1 is the pin at the right hand side of the card.

4.1.2 I/O SIGNALS

The following listing gives the pin location on the cable cards of the I/O functions described previously in this Manual.

Card No. 1:

Pin	Name	Pin	Name
1	B6X	23	IB6X
2	B7X	24	IB7X
3		25	
4	RSCX	26	
5	CL_5	27	
6	I/O CLEAR	28	
7	ACK	29	
8	PI	30	
9	IS	31	
10		32	
11		33	
12		34	
13		35	
14		36	
15		37	
16		38	
17		39	
18		40	
19		41	
20		42	
21	GROUND	43	GROUND
22	+4.5V	44	+4.5V

Card No. 2:

Pin	Name	Pin	Name
1	PA_0	23	BOX
2	PA_1	24	B1X
3	PA_2	25	B2X
4	PA_3	26	B3X
5	PA_4	27	B4X
6	PA_5	28	B5X
7	PCIE	29	B8X
8	PTJE	30	I/O REPLY
9	PCI_1	31	OX
10	PCI_2	32	WMX
11	LOAD	33	SERX
12	BUSY-X	34	PRBFX
13	ERX	35	PSBFX
14	GOPX	36	IBOX
15	BFX	37	IB1X
16	CL_0	38	IB2X
17	CL_1	39	IB3X
18	CL_2	40	IB4X
19	CL_3	41	IB5X
20	CL_4	42	IB8X
21	GROUND	43	GROUND
22	+4.5V	44	+4.5V

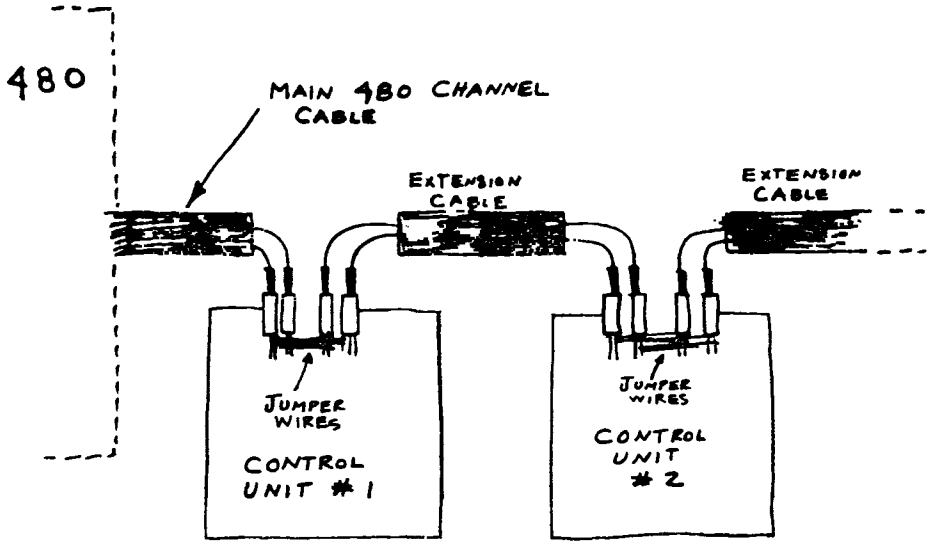


FIG 23 - CONNECTION BETWEEN 480 AND I/O DEVICES

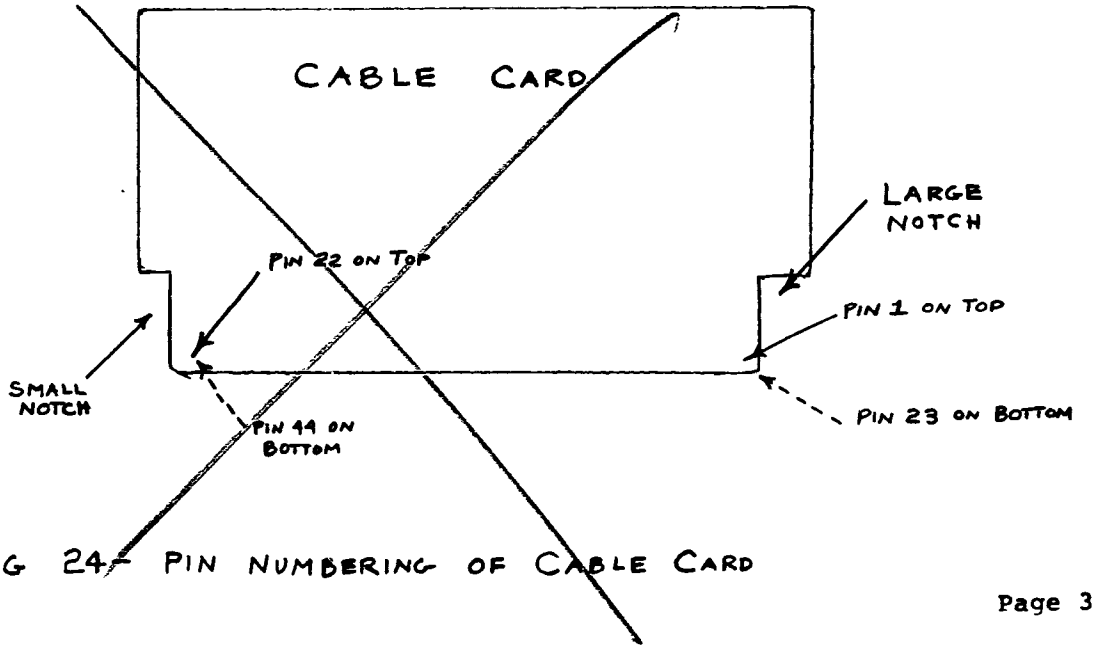


FIG 24 - PIN NUMBERING OF CABLE CARD

- SALES REPRESENTATIVES -

16 Offices

Eastern U.S.

New England

Instrument Dynamics, Inc.
—Wakefield Ind'l. Park, Wakefield, Mass. 01880 (617) 245-5100
—345 N. Main St., W. Hartford, Conn. 06117 (203) 233-5503

Mid-Atlantic (N.Y. to Va.)

Bartlett Associates, Inc.
—130 W. Lancaster Ave., Wayne, Pa. 19087 (215) 688-7325
—PO Box 72, 470 Mamaroneck Ave., White Plains, N.Y. 10605 (914) 949-6476
—7979 Old Georgetown Rd., Bethesda, Md. 20014 (301) 656-3061

Southeast (Va. to Fla. to La.)

Col-Ins-Co., Inc.
—PO Box 13189, 1313-44th St., Orlando, Fla. 32809 (305) 423-7615
—PO Box 1292, Bay St. Louis, Miss. 39520 (504) 833-1116
—PO Box 4246, Rm. 422, 2109 W. Clinton Bldg., 2109 W. Clinton Ave., Huntsville, Ala. 35802 (205) 539-1771
—PO Box 5597, Winston-Salem, N.C. 27103 (919) 765-3650
—PO Box 14077, 418A Armour Circle NE, Atlanta, Ga. 30324 (404) 873-5118

North Central (W. Pa., Ohio, Ind., Mich.)

Tower Engineering Co.
—2801 Far Hills Ave., Dayton, Ohio 45419 (513) 298-1491
—7016 Euclid Ave., Cleveland, Ohio 44103 (216) 881-3511

Mid-West (Minn., Wisc., Iowa, Ill., Mo., W. Ky.)

R. Edward Stemm, Inc.
—5681 W. Lake St., Chicago 44, Ill. (312) 379-2700
—2713 Lyndale Ave. So., Minneapolis 8, Minn. (612) 822-8404

Southwest (Tex., Okla., Ark.)

Arnold Barnes Co.
—PO Box 709, Richardson, Tex. 75080 (214) 235-4541
—Petroleum Ctr., Suite E106, San Antonio, Tex. 78213 (512) TA8-1323
—3230 Mercer St., Suite 108A, Houston, Tex. 77027 (713) NA2-3620

Or contact: BUSINESS INFORMATION TECHNOLOGY, INC.
3 Erie Drive, Natick, Mass. 01760
(617) 235-6842

4. CABLE

4.1 I/O CABLE

The BIT 480 Processor comes equipped with two Winchester Connectors (Part #MRA 50 S) for each data channel. On connecting to a number of I/O control units, the cable should be continued in the manner shown in Figure 23, by means of extension cables, so that output lines from the BIT 480 are common to all I/O devices on the same channel.

4.1.1 CONNECTOR PIN NUMBERS

The numbering of the pins on the Winchester Connectors is illustrated in Figure 24.

4.1.2 I/O SIGNALS

The following listing gives the pin location on the Winchester Connectors of the I/O functions described previously in this Manual.

CABLE #1 (J43)

pin	Name
A	PA0
B	PA1
D	PA2
E	PA3
F	PA4
J	GROUND
C	PA5
H	PCIE
M	PTJE
S	PCI1
W	PCI2
a	GROUND
K	LOAD
L	BUSY LINE
N	ERG
P	GO
R	BF
T	GROUND
U	CL0
V	CL1
X	CL2
Y	CL3
Z	CL4
b	GROUND
c	CL5
d	B0
f	B1
h	B2
j	B3
m	GROUND
e	B4
k	B5
r	B6
v	B7
z	B8
D_D	GROUND
p	
n	
s	
t	
u	
w	GROUND
x	
y	
A_A	
B_B	
C_C	
E_E	GROUND
H_H	
F_F	

ALL WIRES TWISTED PAIR
FIVE GROUNDS COMMON TO ONE PIN.

CABLE #2 (J44)

Pin	Name
A	I/O REPLY
B	OUT
D	SEVEN
E	PSER
F	PRBF
J	GROUND
C	PSBF
H	IB0
M	IB1
S	IB2
W	IB3
a	GROUND
K	IB4
L	IB5
N	IB6
P	IB7
R	IB8
T	GROUND
U	RSC
V	PCL
X	ACK
Y	IS
Z	PIR
b	GROUND
c	+4.5V
d	+4.5V
f	
h	
j	
m	GROUND
e	
k	
r	
v	
z	
D_D	GROUND
p	
n	
s	
t	
u	
w	GROUND
x	
y	
A_A	
B_B	
C_C	
E_E	GROUND
H_H	
F_F	

ALL WIRES TWISTED PAIR
FIVE GROUNDS COMMON TO ONE PIN.

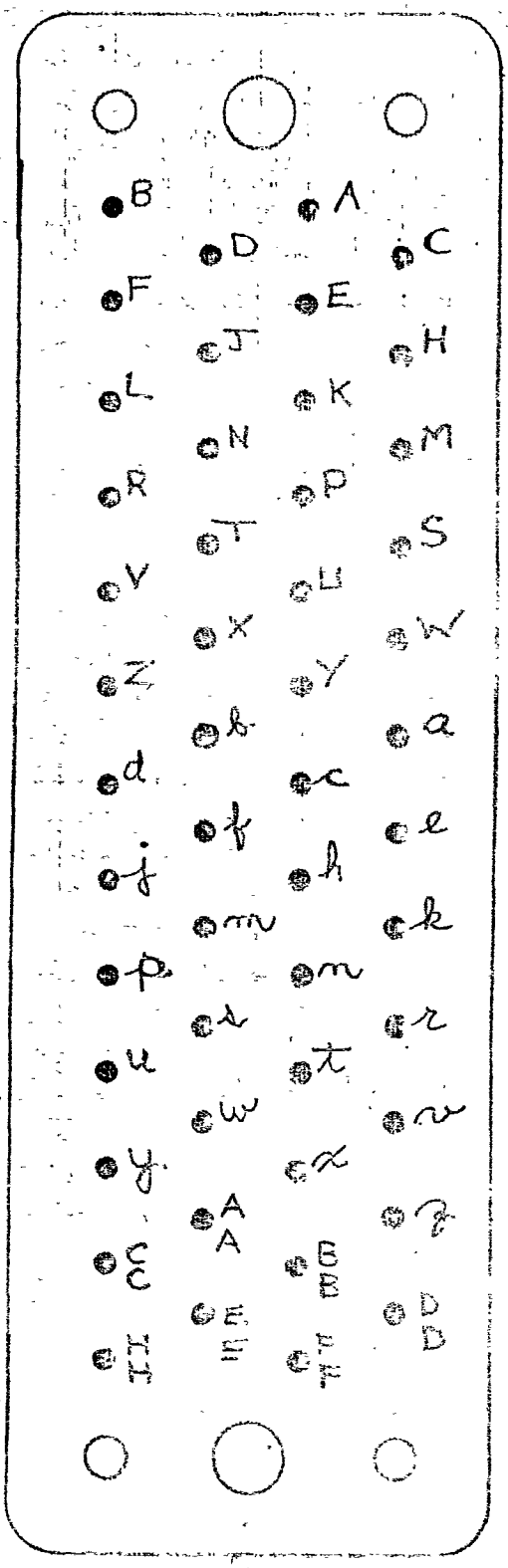


FIG. 24- PIN NUMBERING OF I/O CONNECTOR
WINCHESTER
MRA 50 S
WIRING SIDE