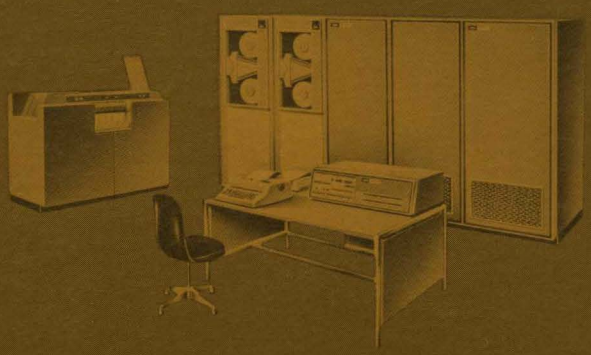
SP-209

THE RAYTHEON 520 SYSTEMS INTERFACE MANUAL



RAYTHEON COMPUTER

RAYTHEON

THE RAYTHEON 520 SYSTEMS INTERFACE MANUAL

November, 1965

RAYTHEON COMPUTER
2700 SOUTH FAIRVIEW ST., SANTA ANA, CALIFORNIA 92704

TABLE OF CONTENTS

	PAGE
520 System Input/Output Philosophy	1
Selection of An Input/Output Device	3
Data Transfer with An Input/Output Device	9
Glossary of Control Signals	11
The Multidevice Controller	15
Introduction	15
Theory of Operation	16
Priority Interrupt System	19
Logical Mechanization	21
Cable Requirements	22

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	520 Input/Output System Diagram	1
2	Communication Link between 520 and I/O Devices	2
3	Select Command Timing	5
4	Select Command Timing	6
5	Data Transfer Command Timing	8
6	Data Transfer Command Timing	10
7	Raytheon 520 Multidevice Controller	14
8	Data Transfer Timing in MDC	18
9	Cable, Multiconductor, 25 Twisted Pair, #24 AWG	22
10	System Expansion Interconnection	23
11	Typical Input/Output Communication Pair	23

520 SYSTEM INPUT/OUTPUT PHILOSOPHY

The Raytheon 520 has an input-output system which is modular by design. This gives the user the flexibility of specifying the exact configuration needed for his present requirements with ease of expansion when needed. The 520 achieves this flexibility by the use of a dual-bus structure (input and output) to which peripheral devices may be added at any time.

Communication with peripheral equipment is achieved by controlling their connection to the input/output bus. The connection of peripheral devices to the input/output bus is accomplished by the use of a device controller which is unique for each equipment type. (See Figure 1) Any of the controllers may be logically connected to or disconnected from the computer under program control.

A controller must contain the logic circuitry which enables it to perform the following functions:

1) Sample I/O bus control signals and determine when one of the devices connected to it is being addressed by the computer.
2) Connect the device to the bus for the purpose of receiving commands or transferring data.
3) Maintain the device in a previously commanded mode of operation until further commands are received or a predetermined condition occurs.
4) Accept and provide synchronizing signals required by the computer and the device.
5) Provide character or word buffering along with error detection.
6) Provide the necessary voltage levels and impedance matching.

Figure 2 is a block diagram of the communication link between the 520 and any one of its peripheral equipments. The communication link is controlled by the execution of an input/output command. The 520 has two input/output commands. They are the SELECT command (SEL) and the DATA TRANSFER command (DTR).

The 520 can select any I/O device with a SELECT command. The SELECT command will send to all controllers control signals and 24 bits of select

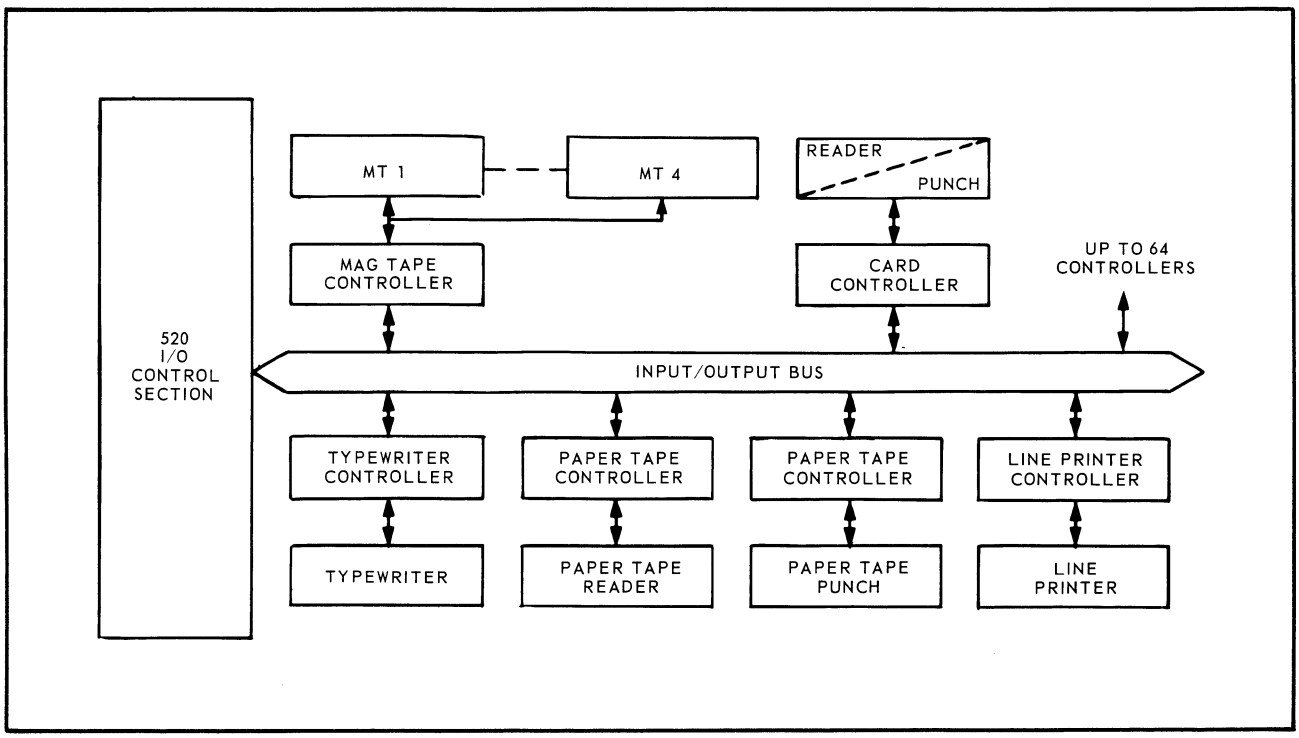


Figure 1. 520 Input/Output System Diagram

command word information over the output bus (Q bus). The select command word determines which I/O Controller will be activated and sets up in the selected controller all the desired conditions for the forthcoming transfer of data. A control signal will be sent from the controller to the 520 when the selection has been completed in the controller and results in the termination of the SELECT command (EKO). Along with the control signal to terminate the SELECT command, 24 bits of status information are sent to the 520 over the input bus (R bus). The SELECT command designates which register in the 520 is to receive the status information. Alternately, the SELECT command can request status information only.

Data transmission is accomplished by a DATA TRANSFER command (DTR). This command will specify a register in the 520 which will transmit or receive data or both. The type of data transfer (input, output, status, etc.) is determined by the action field of the DTR command which generates the appropriate control signal. When the selected I/O controller has determined that the data transfer operation has been completed, a control signal (EKO) will be sent to the 520 which will terminate the DTR command.

There are four channels provided for data transfer control signals between the 520 and the I/O device controllers. When a device controller is selected, it must be assigned one of the four channels which is not presently being used. Each channel can transmit an interrupt from the device controller to which it is assigned. This interrupt indicates to the computer that the device controller is ready to transmit data or to accept data. Each channel can send an enabling signal from the 520 to the device controller to which it is assigned. This enabling signal indicates to the controller that the 520 is starting a data transfer operation. These four channels allow up to four peripheral devices to be connected to the input/output bus at the same time. Since the 520 has the ability to connect and disconnect peripheral device controllers, it is possible for more than four devices to be operated simultaneously.

To coordinate the time-sharing of the input/output bus, a commutator has been mechanized in the 520. The commutator is a four-count stepping counter which counts at the the clock rate of the 520. Each count of the commutator scans one of the four interrupt channels. When an interrupt is present and the channel is enabled for automatic

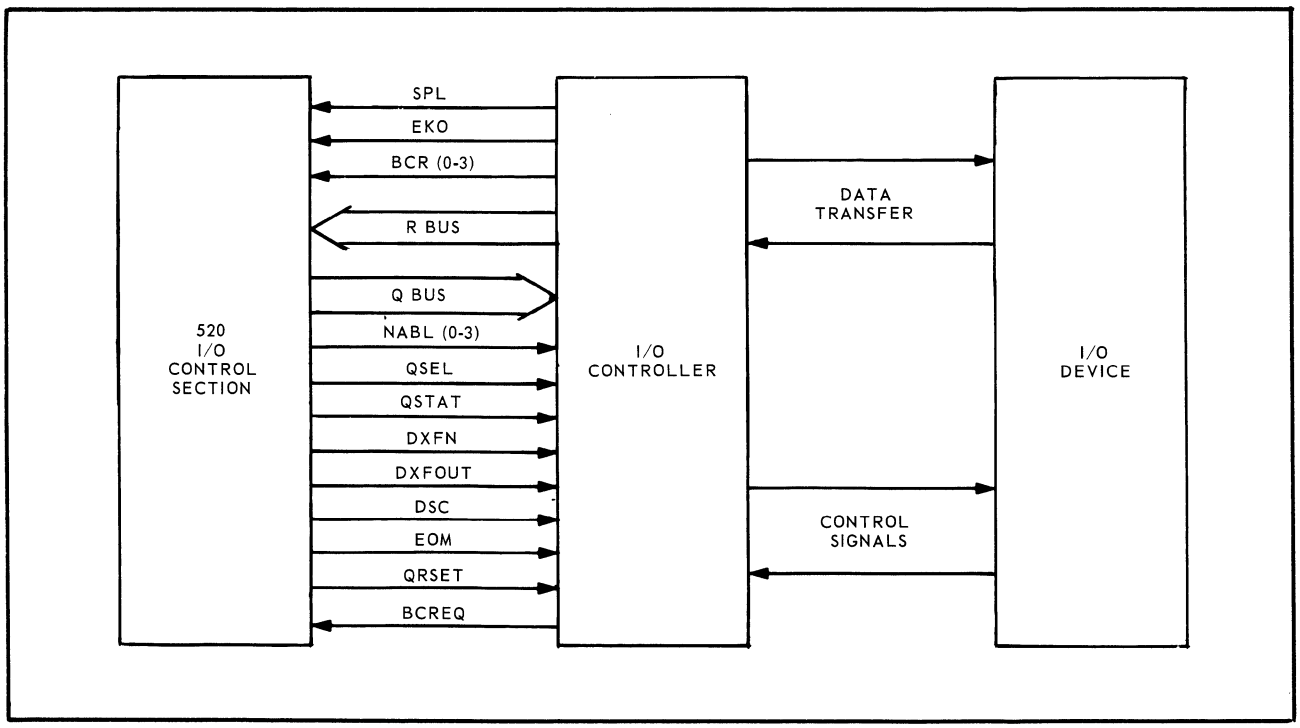


Figure 2. Communication Link Between 520 and I/O Devices

interrupt (each channel can be enabled or disabled under program control), the commutator will be locked on the channel of the interrupt. Upon completion of the instruction pair being executed, the computer will automatically access one of the first four locations of fast memory. The location accessed depends on the channel to which the commutator is locked (70000 for channel 0; 70001 for channel 1, etc.). This location in fast memory serves as an entrance to a subroutine which will handle this request for data transfer. In the event that fast memory is not part of the system configuration, a transfer to one of the same four locations, which are now assigned to main memory, will be effected.

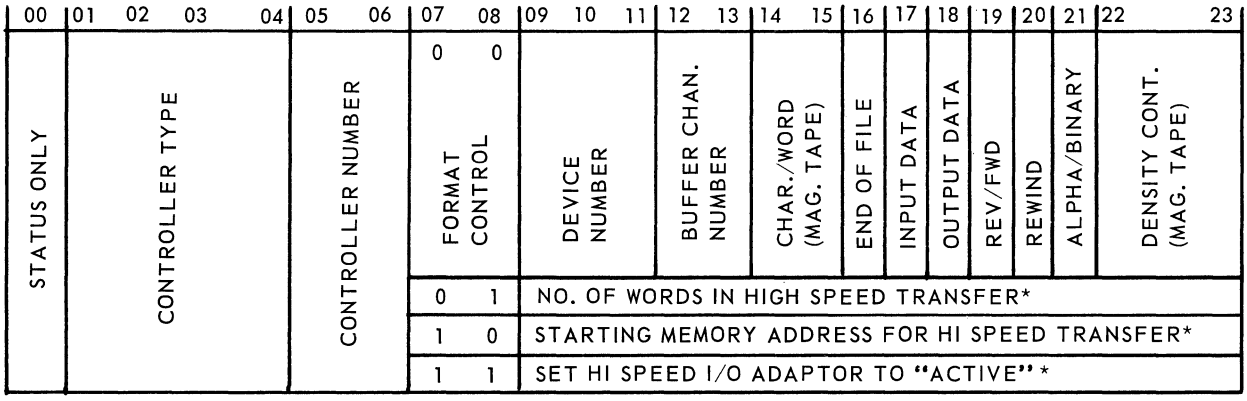
Once the commutator is locked on a channel, it will remain locked until released by program control or by the Reset button on the control console.

If automatic interrupt is not desired, all channels or any combination can be disabled under program control. The presence of an interrupt can then be determined by test commands under program control and action taken as written in the program.

In the execution of a SELECT or a DATA TRANSFER command, the circulation of the 520 system clock is inhibited until the selected I/O controller responds with a control signal (EKO) which indicates that the operation is complete. With the 520 system clock circulation inhibited, the computer is in a "wait" condition and no computation can be accomplished until EKO releases the 520 from this waiting state. Therefore, the time an I/O controller takes to process a peripheral command should be kept to a minimum (a typical 520 I/O controller takes two to three microseconds).

SELECTION OF AN INPUT/OUTPUT DEVICE

An input/output device is selected for data transmission by a SELECT command. This command sends to all controllers in the 520 I/O structure, control signals which indicate a selection is taking place and 24 bits of SELECT command information. The 24 bits of SELECT command information is gated onto the output bus (Q bus) and will activate only one controller. The field assignments of the SELECT command word are as follows:



*OPTIONAL FEATURE

STATUS FIELD (bit position 00)

The status field determines whether the I/O device addressed is being selected for data transmission or for status information only. (If bit 00 is a 1, select for status only – if bit 00 is a 0, select for data transmission). A request for status should not activate the controller except to give a status reply over the input bus (R bus). Selection for data transmission should place the controller on-line with the 520 and begin the function requested by the SELECT command.

CONTROLLER TYPE FIELD (bit positions 01 to 04)

A complete device address consists of the controller type, the controller number and the device number. The controller type field refers to the device type (paper tape reader, line printer, etc.). With this four-bit control field it is possible to select any one of 16 unique controller types. The controller type field assignment reserved for user-designed controllers is 1111.

CONTROLLER NUMBER FIELD (bit positions 05 and 06)

The controller number field allows up to four of like type controllers to be selected individually.

FORMAT DESIGNATION FIELD (bit positions 07 and 08)

This field is used in conjunction with an optional high-speed buffered input/output system and determines the interpretation to be placed on the remainder of the select command word. (Bits 09 to 23). This field will be 00 unless the optional high-speed buffered input/output system is incorporated into the system.

DEVICE NUMBER FIELD (bit positions 09 to 11)

The device number field allows a controller to select only one of up to eight unique devices of like type which may be connected to a controller.

CHANNEL ASSIGNMENT FIELD (bit positions 12 and 13)

There are four communication channels to which a device may be assigned. A controller must have a channel assignment before successful data transmission may occur. This determines the channel over which an interrupt will be sent to the 520 from a controller and the channel which will enable a controller for data transmission. The channel assignments are as follows:

Bit position		Channel Assignment
12	13	
0	0	Channel #0
0	1	Channel #1
1	0	Channel #2
1	1	Channel #3

OPERATION FIELD (bit positions 14 to 23)

The operation field determine all of the particulars of the forthcoming data transmission. For example, bit 21 determines whether the data transmission is binary or alphanumeric. (1 – alphanumeric, 0 – binary). [The field assignment chart gives a breakdown of the operation field.]

Timing diagrams, Figures 3 and 4, illustrate the timing considerations for a SELECT command. A sharply defined rise time and fall time indicate a function which is clocked whereas a sloping rise time and fall time indicate a non-clocked function.

When the 520 executes a SELECT command, the SELECT command word is immediately gated onto the output bus (Q bus). At Read/Write clock, which is 300 nanoseconds after the system clock, QSEL is generated and is an indication to all controllers that the information on the Q bus is a select command word and that a device selection is in process. At Data Ready clock, which is 450 nanoseconds after Read/Write clock, QSTAT is generated and will cause a status response sequence in the selected controller.

The SELECT command word is examined by all I/O controllers and decoded by only the desired controller. XCMD is the function in the I/O controller which decodes the SELECT command word and becomes active. XCMD remains in the active state until the SELECT command is terminated. With XCMD active and an I/O clock, the XCSET flip-flop is set. This serves to synchronize the 520 clock system with the I/O controller clock. At the next I/O clock pulse the XCLCH flip-flop is set and all the control flip-flops are set to the desired configuration, as indicated by the SELECT command word. XCLCH being in the set state indicates to the I/O controller that it is on-line with the 520 and able to communicate with the 520. (Sends interrupts when ready to transmit or receive data, processes DTR commands, etc.). XCLCH activates all of the control logic in the I/O controller which controls the device connected to the controller. XCLCH will remain in the set state

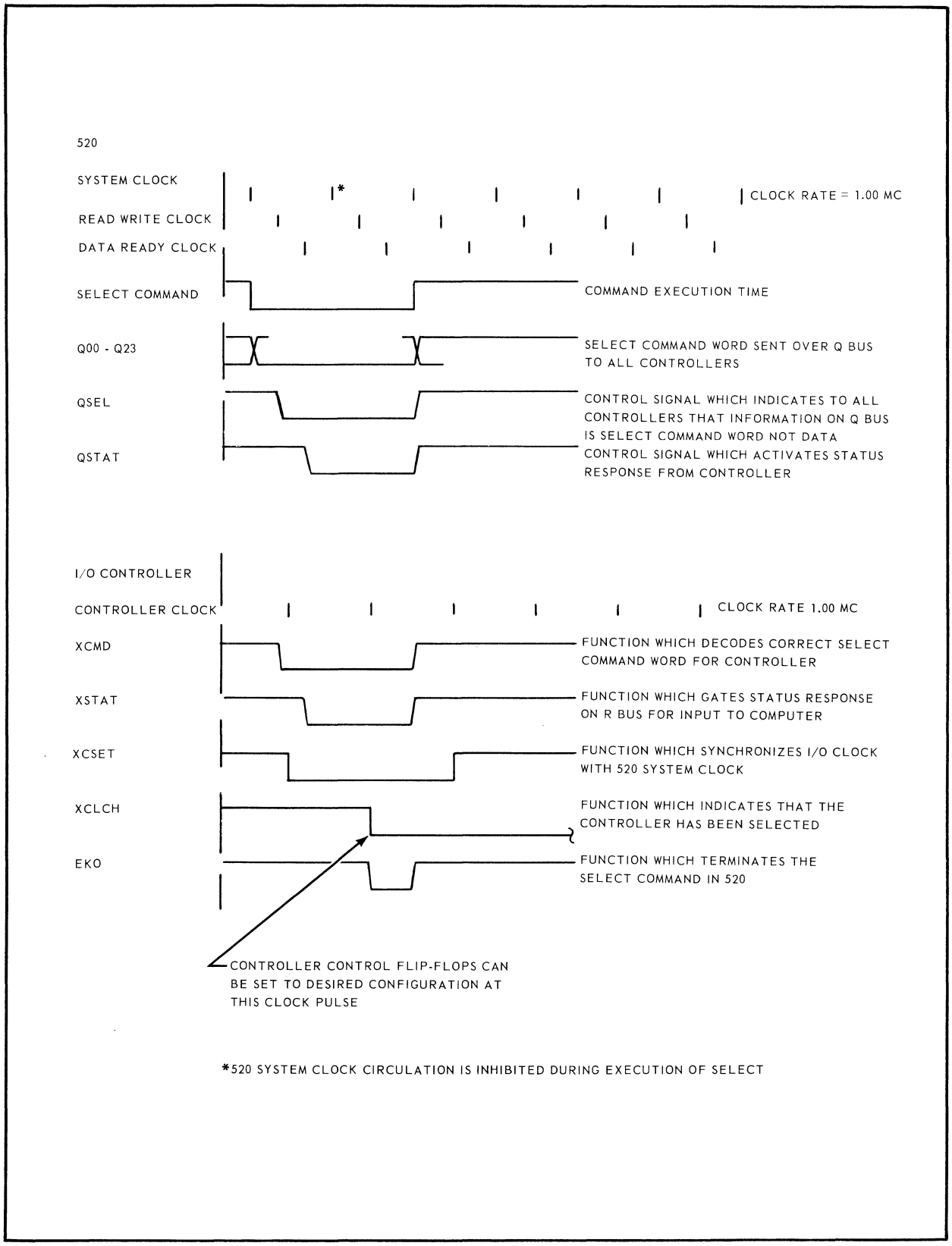


Figure 3. Select Command Timing. Controller Not Connected to 520 System

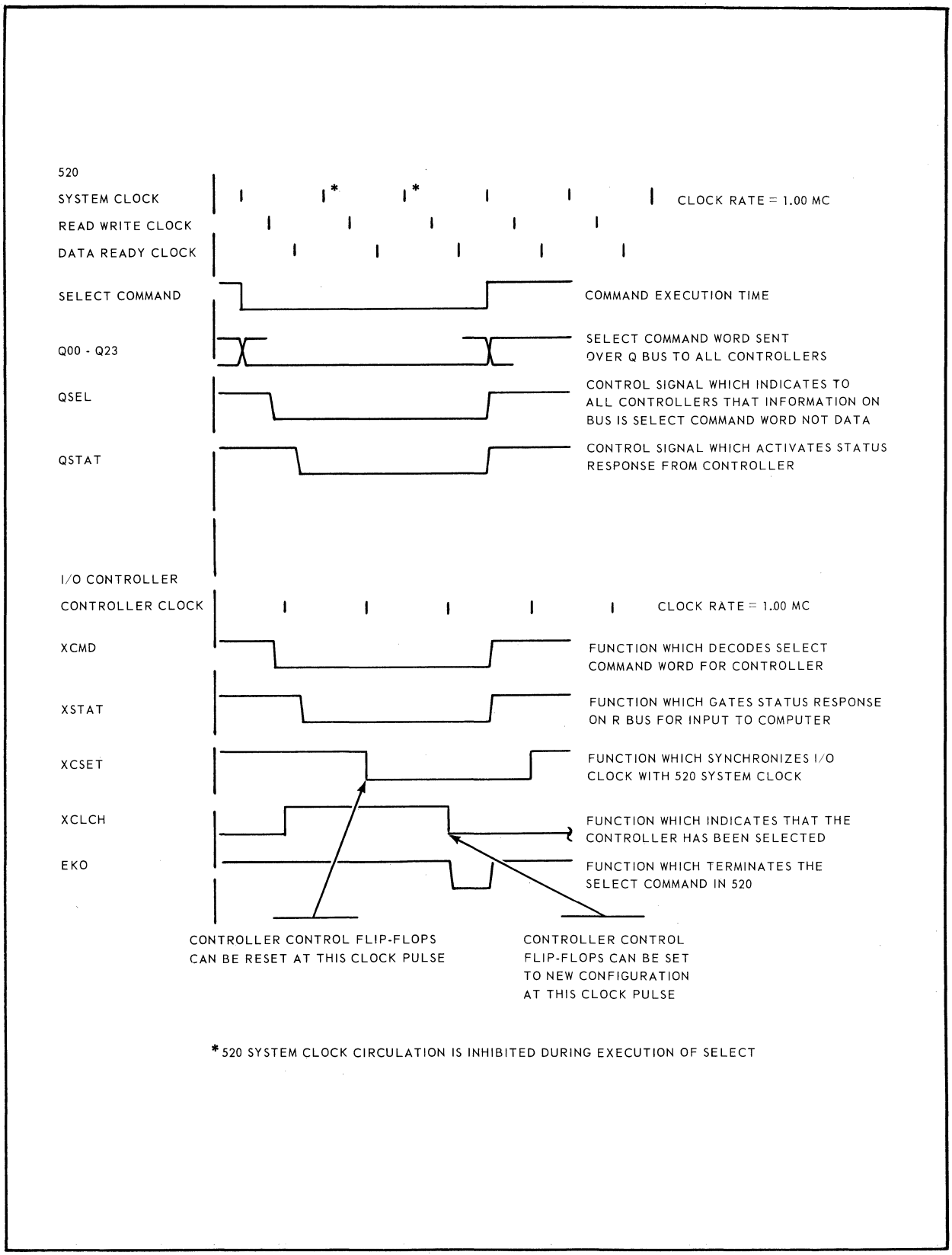


Figure 4. Select Command Timing. Controller Connected to 520 System by a Previous Select

until the I/O controller is disconnected by a DTR command, reselected while still on-line, or disconnected by the Reset button.

With XCMD, XCSET, and XCLCH all set, EKO becomes active and is sent to the 520. EKO results in the 520 system clock being allowed to circulate. The first 520 system clock which is circulated after EKO terminates the SELECT command and gates into the selected register in the 520 the status response word. This status response word was gated onto the input bus (R bus) by XSTAT during the select process in the I/O controller. The termination of the SELECT command causes the reset of QSEL, QSTAT, and the SELECT command word on the output bus (Q bus). XCMD goes false and at the next I/O controller clock XCSET is reset. XCLCH remains in the set state.

The control signal, EKO, should be terminated as soon as the SELECT command is terminated. This is a requirement because another peripheral command can not be executed until EKO is false.

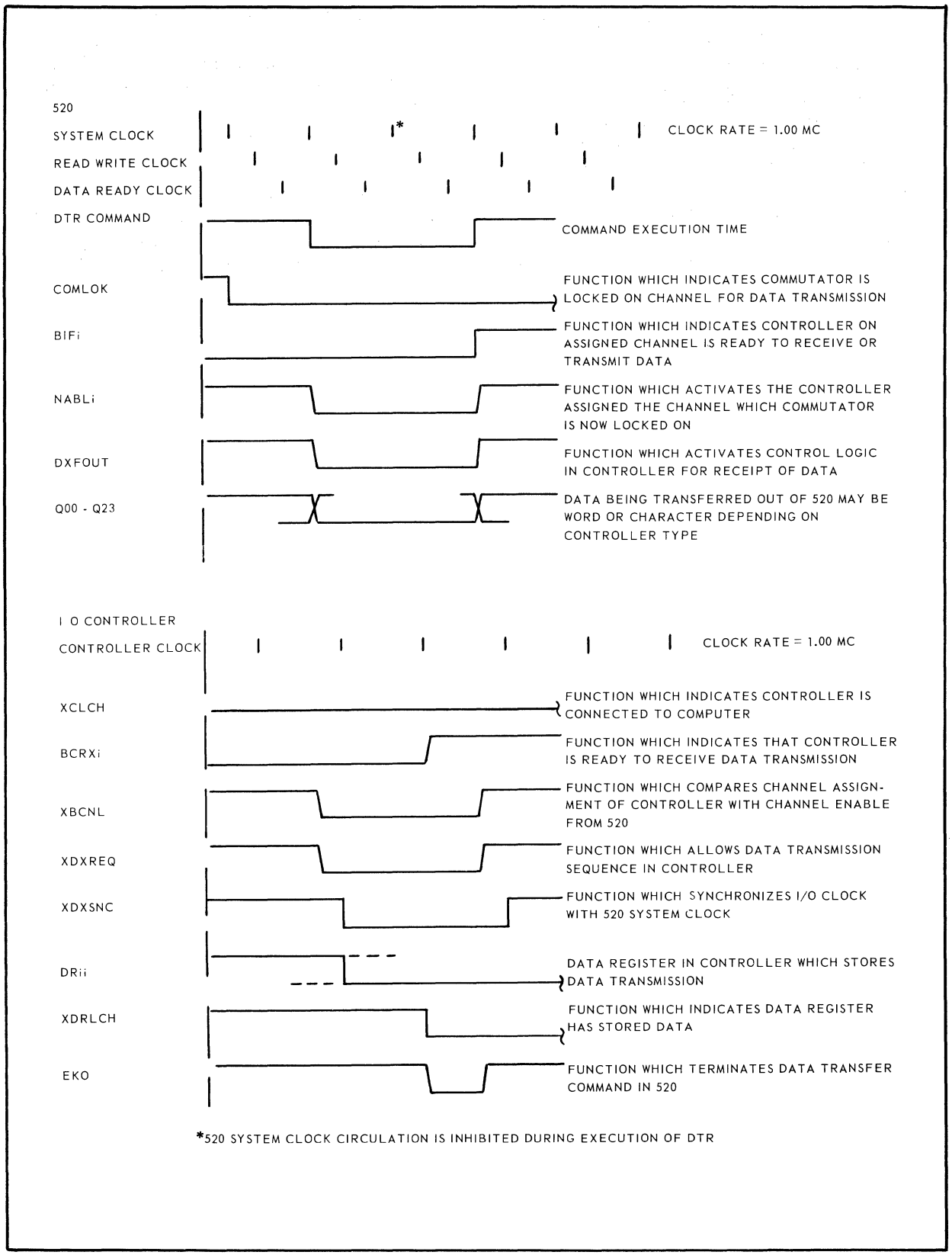


Figure 5. Data Transfer Command Timing. Data Transfer Out to 1/O Controller From 520

DATA TRANSFER WITH AN INPUT-OUTPUT DEVICE

A device controller must be selected by the 520 before data transmission can occur. The action field of the DATA TRANSFER command (R1 field) is three bits and specifies the operation to be performed. The table listed below gives a breakdown of the operations which can be specified by a DTR.

R1 FIELD 123	OPERATION	CONTROL SIGNAL
000	Disconnect Device	DSC
001	Data Transfer In	DXFN
010	Data Transfer Out	DXFOUT
011	Data Transfer In and Out	DXFN and DXFOUT
100	End of Message and No Data Transfer	EOM
101	Data Transfer In and End of Message	DXFN and EOM
110	Data Transfer Out and End of Message	DXFOUT and EOM
111	Request for Status	QSTAT

The DATA TRANSFER command will normally cause a skip condition in program execution. If the DATA TRANSFER command is used to disconnect a device a non-skip condition will occur. If in the execution of a DTR a special condition exists then a non-skip will also occur. This special condition can be defined by each controller type as to meaning. For example, in the Paper Tape Reader controller the special condition would exist when reading a frame of alphanumeric data if a parity error was detected. In the Card Read controller this special condition could define the end of card. The control signal generated by this special condition is SPL and it must only come true during the DTR which detects the special condition. SPL must come true before the control signal EKO which terminates the DTR.

A DTR can request status from a controller which is connected to the computer. This status can define a condition which exists at the device (power on) or a condition which occurred during transfer of data from the device to the controller (parity error when reading magnetic tape). This status response will be sent to the computer over the input bus (R bus) and can be interrogated under program control as to content.

A DTR can disconnect a device from the input-output bus. This operation will reset the XCLCH flip-flop and free the communication channel to which it was assigned for use by another device. The termination of a DTR for disconnect is accomplished by the computer in the main frame. Therefore, it is not necessary to have logic in the controller which would generate EKO for a DTR to disconnect. Once disconnected the controller must not be able to send an interrupt or transmit data. Any operation of the device, if in process, must be brought to an orderly termination.

End of Message (EOM) is a control signal which can be generated in conjunction with Data Transfer In or Data Transfer Out. This has significance for controllers which are involved with data transfer in block form. For example, data transmission to an 80 column line printer may be complete after the first 40 columns in a particular line of print. EOM is the control function which initiates a print cycle in the line printer.

The timing diagrams, Figures 5 and 6, illustrate the timing considerations for a DTR OUT and a DTR IN. A sharply defined rise time and fall time indicate a clocked function, whereas a sloping rise time and fall time indicate a non-clocked function.

The commutator must be locked to one of the four communications channels for the successful execution and termination of a DATA TRANSFER command. This is indicated in the main frame of the 520 by the COMLOK flip-flop being in the set state. COMLOK must be in the set state in order to generate the enable control signal (NABLi – i can be 0, 1, 2 or 3 depending on the channel the commutator is locked on) during the execution of the DTR. BIFi (i can be 0, 1, 2 or 3 depending on the channel which is sending the interrupt from the controller) is a flip-flop in the main frame of the 520 which gets set when an interrupt is present and remains in the set state as long as the interrupt is present. The 520 may respond to the interrupt automatically, in which case the commutator will lock to the channel of the interrupt by hardware, or under program test control which must lock the commutator to the proper channel by command execution before execution of a DTR command.

For a controller to communicate with the 520, the XCLCH flip-flop must have been set by a SELECT command, along with all the desired conditions of data transfer. The controller must generate various

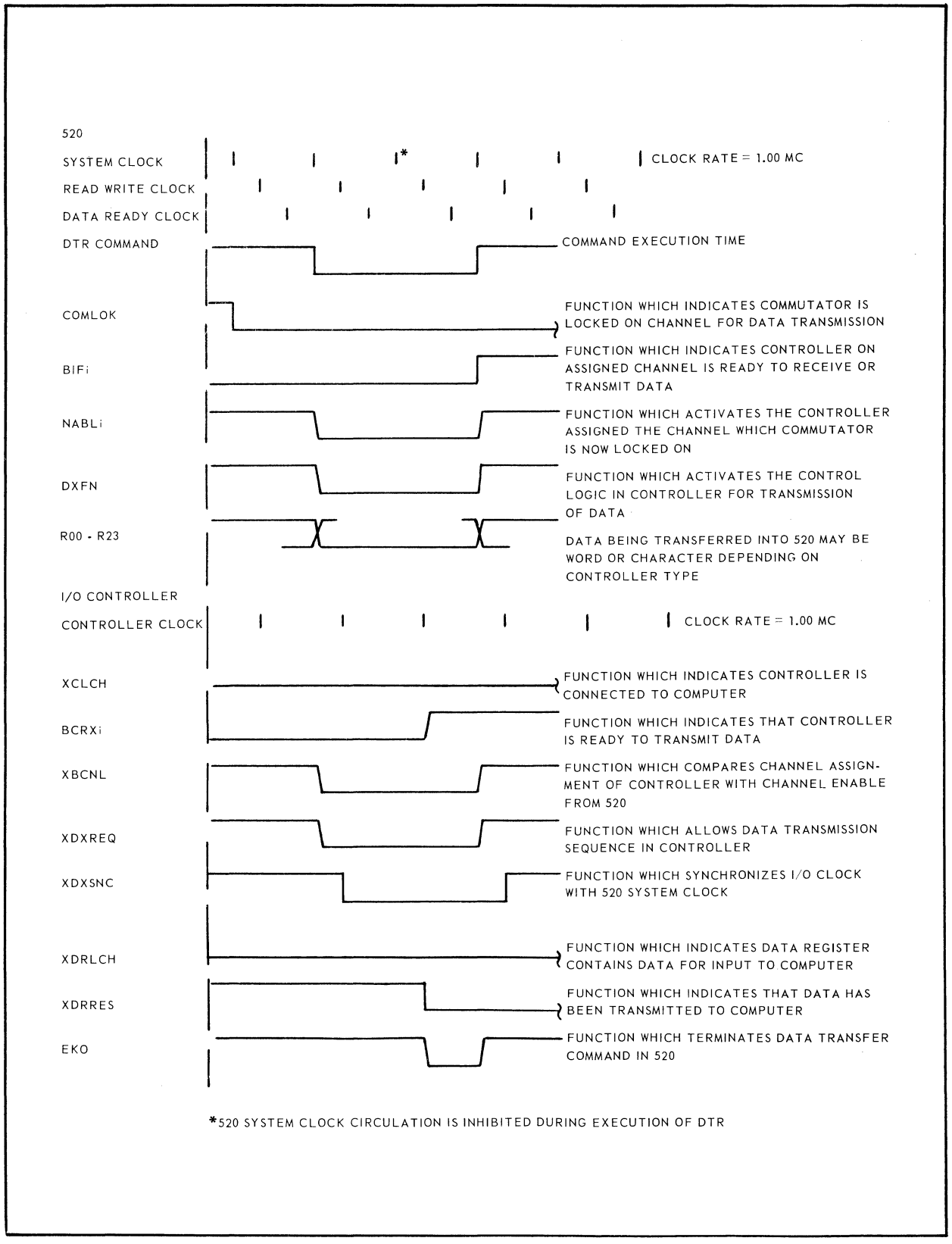


Figure 6. Data Transfer Command Timing. Data Transfer Into 520 From 1/O Controller

control functions to the device (e.g., motion control) to have data available for transmission to the 520 or to be able to accept data. When this situation occurs, the controller will issue an interrupt to the 520 (BCRXi – i can be 0, 1, 2, or 3, depending on the channel assigned to the controller). Depending on the device being controlled, the interrupt will last either until the data is processed or while the data is available.

When the DATA TRANSFER command is executed, the enable control signal (NABLi) and the selected action control signal or signals (DXFN, DXFOUT, EOM, DSC, or QSTAT) are sent to all controllers with no delay. NABLi determines which controller will transmit or receive data and is generated by every DTR. In the case of a DTR OUT, the data is gated onto the output bus (Q bus) at the same time the control signals are sent to the controllers.

NABLi is compared against the channel assigned the I/O controller by a SELECT command and, if they compare, XBCNL becomes active. XDXREQ, a second level function, comes active if XBCNL is active, if the controller is selected (XCLCH), and, if the controller is ready to handle data transfer. In the input mode, the contents of the I/O controller buffer are gated onto the input bus by XDXREQ. At the first I/O controller clock following XBCNL and XDXREQ becoming active, XDXSNC gets set. XDXSNC synchronizes the 520 clock system with the I/O controller clock and is in the set state only for the duration of the DTR command. Data transmission can be synchronized between the device – controller – computer by two flip-flops, XDRLCH and XDRRES. The truth tables below outline a typical application of these flip-flops to this synchronization for both input and output devices.

If the 520 executes a DTR command before the controller is ready to process data, the 520 will remain in a wait condition until the controller is ready. This is accomplished by XDXREQ not coming true until the controller is ready to process data.

INPUT DEVICE

FUNCTION		OPERATION
XDRLCH	XDRRES	
0	0	Controller is ready for input from device
0	1	Controller buffer can be cleared
1	0	Device has transferred data into controller buffer
1	1	Controller is transferring data into computer

OUTPUT DEVICE

FUNCTION		OPERATION
XDRLCH	XDRRES	
0	0	Controller is ready for input from computer
0	1	Controller buffer can be cleared
1	0	Computer has transferred data into controller buffer
1	1	Controller is outputting data to device

The DTR is terminated by a control signal, EKO, which is sent to the computer from the controller when the data transmission is complete. The control signal, EKO, should be terminated as soon as the DATA TRANSFER command is terminated to eliminate any delay in the execution of another DTR or SELECT command.

GLOSSARY

Computer to I/O Controller

DSC – Control signal generated by a DTR command which will disconnect the selected controller from the I/O bus.

DXFN – Control signal generated by a DTR command which indicates to the selected controller that a data transfer from the controller to the 520 is taking place.

DXFOUT – Control signal generated by a DTR command which indicates to the selected controller that a data transfer from the 520 to the controller is taking place.

EOM – Control signal generated by a DTR command which indicates the end of an input or output data transfer.

NABLi (0,1,2,3) – Control signal generated by a DTR command which indicates that the commutator is locked on a channel. This function will enable an I/O controller which is assigned the channel on which the commutator is locked.

QRSET – Control signal generated by the 520 power ON button which disconnects all I/O controllers from the I/O bus.

QSEL – Control signal generated by a SEL command which indicates to all I/O controllers that the information on Q bus is a SELECT command word,not data.

QSTAT – Control signal generated by a SEL or DTR command which causes a status response word to be sent to the 520 on the input bus (R bus).

Q00 to Q23 – The 24-bit output bus which is capable of sending either data or a SELECT command word to the selected controller.

I/O Controller to Computer

BCREQ – Control signal which indicates the presence of an interrupt on the special interrupt line. This special interrupt line is not able to automatically interrupt program execution and can only be determined to be active by execution of a test command.

BCRi (0,1,2,3) – Control signal which indicates the presence of an interrupt from an I/O controller. There are four interrupt lines from every I/O controller to the 520. The assignment to a particular interrupt line is made during the selection of the I/O controller. These interrupt lines can interrupt program execution automatically or can be tested depending on mode of operation.

EKO – Control signal which terminates either a DTR or an SEL command when the I/O controller concerned determines that the operation is complete.

R00 to R23 – The 24-bit input bus which is capable of sending data in either word or character format, or can send a status response word to the 520.

SPL – Control signal generated by the I/O controller during execution of a DTR command which indicates a special condition exists. This special condition, depending on the I/O controller can be a parity error, an end of record, an end of card, etc. SPL will cause the DTR command to non-skip which indicates to the program that a special condition occurred.

Internal to I/O Controller

XBCNL – Function which compares the NABLi control signal generated by a DTR command against the channel assignment flip-flops in every controller. If there is a comparison, the DTR will be processed in the I/O controller where the comparison is made.

XBCS0, 1 – In every I/O controller, there are two flip-flops which store the channel assignment of the I/O controller. These flip-flops determine which interrupt line will be used to signal an interrupt condition and will determine when a DTR command will be processed by the I/O controller.

XCLCH – Function which determines whether or not an I/O controller is on-line with the 520.

XCMD – Function which decodes the SELECT command word during SELECT command.

XCSET – Function whigh synchronizes the I/O controller clock with the 520 system clock during the SELECT command.

XDRLCH – Function which can synchronize the data transmission from the I/O controller to the 520 (or from the 520 to the I/O controller).

XDRRES – Function which can synchronize the data transmission from the I/O controller to the 520 (or from the 520 to the I/O controller).

XDXREQ – Function which allows a DTR command to be fully processed inside the selected controller. This takes into account whether or not the controller is on-line and ready to transmit data.

XDXSNC – Function which synchronizes the I/O controller clock with the 520 system clock during a DTR command.

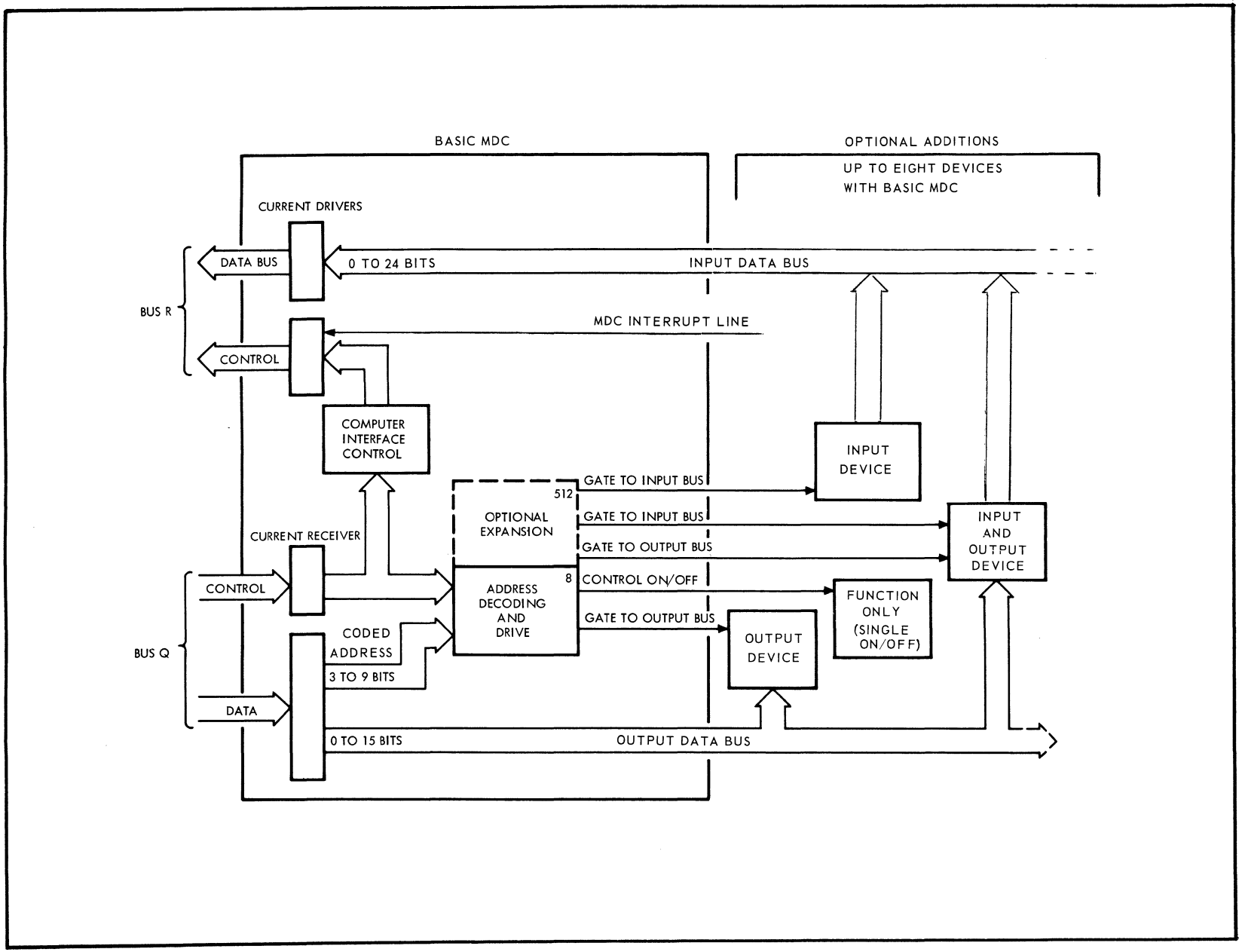


Figure 7. Raytheon 520 Multidevice Controller, Block Diagram

THE MULTIDEVICE CONTROLLER

INTRODUCTION

Although it is necessary to supply special purpose controllers for each type of device, this may not be necessary where several simple functions are required within one system. By combining several related functions within one controller, many advantages can be realized. By time-sharing certain interface requirements, such as input amplifiers, output line drivers, controller identification, and command circuitry, etc., significant cost savings and programming advantages are possible. Also, only one I/O channel is used for all combined functions.

Typical data systems interfaces are simple in nature, no complex timing or control signals being required, but usually communication with a relatively large number of devices is necessary, e.g., multiplexers, A/D converters, D/A converters, control registers, sense lines, etc. Also typical of data systems is the frequent need for change or expansion of the original system. The typical computer data systems supplied today are so special purpose in nature that any change requires significant engineering effort and factory support.

To meet the need of the computer data systems market, Raytheon Computer has designed a simplified and generalized computer interface system, the Multidevice Controller, which is intended to be equivalent to a "Real-Time Data and Control System Package." The design philosophy employed recognizes the fact that customer capability has increased in recent years to the point where simple extensions of a basic logical design may be implemented with standard digital modules by virtually all laboratories making use of data systems equipment.

The Multidevice Controller (MDC) utilizes standard Raytheon Computer digital modules, power supplies, and mounting hardware. The MDC may be supplied in a low cost minimum configuration which may be expanded as required in the field by plugging in additional digital modules and/or by installing standard pre-wired sub-assemblies. It is practical to implement almost any imaginable system requirement in this manner. The standardized hardware and techniques result in lower cost, simpler and easier to maintain systems. In addition, modifications require far less time and effort.

Virtually all computer systems communicate with data system devices via two I/O operations; one to address the device and set it up to accept or present data and a second I/O operation for the actual data transfer. Raytheon Computer experience indicates that because of the nature of data systems equipment, output data is rarely more than 12 - 15 bits, i.e., only a portion of one computer word, but that input data is frequently equal to a full computer word, e.g., reading sense lines in parallel. By utilizing an external decoder it is possible to output device identity and data in one word. Since device identification is always an output operation, this format also allows a full word length parallel input. This design philosophy, as implemented in the Raytheon Computer MDC, thus allows the simultaneous output of data and device address in a single I/O operation. This results in decreasing the I/O operations by one-half and either more efficient use of memory or elimination of time otherwise required to manipulate a word containing both address and data.

Basically, the MDC satisfies the basic I/O bus interface requirements and can output in parallel or accept in parallel 24 logic levels. The format may be varied so that either 15 or 16 bits are available for output data, the remaining corresponding 9 or 8 bits then being available for device addressing, i.e., 512 or 256 devices. As indicated in Figure 7, the address bits are output to a decoder. The function of the decoder is to identify and change the logic level on one out of up to 512 possible decoded lines.

Each device is either a data source or sink of one or more bits of logic levels presented in parallel. All output devices are connected to the output bus and all input devices are connected to the input bus at all times. The basic interface between the respective data bus and the device is a two input AND gate, i.e., the output of the AND gate will go true (change from zero volts to −10 volts) only when both inputs are true. One input of each AND gate is the device information from a given bit; the other input comes from the address decoder. Thus, when selected by the address decoder, only one device will be "gated" to a data bus.

As indicated in Figure 7, there are four general types of I/O devices: (1) output only, (2) input only, (3) combination of input and output, and (4) function only, i.e., single bit control (no need to connect to a data bus for a single bit). An example of a type 1 device is a D/A converter; a type 2 device, an array of 24 sense contacts addressed as a single device; type 3 device, a multiplexer, sample-and-hold amplifier, A/D converter subsystem; and a type 4 device, controlling an alarm function.

The 520 has the unique capability of unloading a register to an output device and simultaneously loading the same register from an input device with the same address in a single I/O operation. This is applicable to a type 3 device such as the analog subsystem. In this manner up to 15 bits of address data may be output to a multiplexer (one out of 32,000 analog channels) while data from the previous channel, which is stored in the A/D converter output register, is read into the computer. The entire operation can last from one to five μsec depending on the circuit modules used.

The basic MDC is supplied as two MC26 module cases which are pre-wired for decoding and driving 64 single-ended address lines but equipped with digital modules for eight address lines. Expansion to 64 lines is accomplished by simply plugging in additional digital modules. Beyond 64 address lines, an additional MC26 module case is added which is pre-wired for an additional 64 address lines but equipped with decoding and drive for 16 address lines. Additional module cases may be added to give a maximum of 256 or 512 address lines, depending upon a corresponding data output format of 16 or 15 bits.

In addition, standard options are available which are shipped as pre-wired and tested module cases and which are easily wired to the basic MDC. These options include: input assembly or output disassembly registers that allow packing one 24-bit, two 12-bit, three 8-bit, or four 6-bit data words into one computer word; interval timers (count down timer) of 15 or 24 bits with programmable input frequencies digital clocks (count up timer); analog timing subsystems for a multiplexer, sample and hold, and A/D converter; output control registers, six 15-bit power flip-flops per MC26 module case – 100 ma.; input register, three 24-bit registers per MC26 module case (sense switch or parallel load); and digital counters.

THEORY OF OPERATION

Basically, the Multidevice Controller provides a means for a number of devices (eight with the basic MDC), to communicate with the 520 and share the interface logic required. Functionally, the basic MDC must do the following:

1) Respond to a SELECT command by going on-line with the 520 with an interrupt channel assignment which is dependent on the SELECT command word. Once on-line, the MDC is in a passive state in that further communication is required from the 520 to address the device desired. This is accomplished with a DATA TRANSFER command.
2) Return a status response based on the optional devices added to the basic MDC. This status response may be requested by either a SELECT or a DATA TRANSFER command.
3) Generate an interrupt based on the optional devices added to the basic MDC.
4) Respond to a DATA TRANSFER command and perform address decoding to provide a control signal for the optional device addressed by the basic MDC. This addressing is accomplished by bits Q00–Q08 of the word being output by the DATA TRANSFER command. Q09 to Q23 can be treated as data output and routed to the device being addressed.
5) Respond to a DATA TRANSFER command for disconnect by going off-line and releasing the interrupt channel to which it was assigned.
6) Achieve flexibility of design in that the logical mechanization of the basic MDC can be easily tailored to fit the requirements of the optional devices to be added. This means that the logical description of the basic MDC which follows may be altered to better fit a system requirement.

Select Sequence

The Multidevice Controller is put on-line with the 520 by executing a SELECT command. Besides putting the MDC on-line, the SELECT command will assign the MDC one of the four interrupt channels and will send a status response to the designated register in the 520. The status response can be wired to fit the particular configuration of each MDC. The SELECT sequence in the MDC can be described as follows:

1) $MDCMD = \underbrace{Q00' \cdot QSEL}_{\text{Select for Data Transmission}} \cdot \underbrace{Q01 \cdot Q02' \cdot Q03' \cdot Q04'}_{\text{Controller Type MDC}} \cdot \underbrace{Q05' \cdot Q06'}_{\text{Controller \#0}}$

MDCMD is the function which decodes the SELECT command word on the Q bus and becomes active only during a SELECT of the MDC. Q00' indicates this is a select which will put the MDC on-line and QSEL indicates that the data on the Q bus is a SELECT command word.

2) s $MDSET = MDCMD \cdot MDLCH' + MDCNL \cdot DXFDLY' \cdot DXF$

r $MDSET = MDCMD' \cdot MDCNL' + MDBUSY' \cdot DXFDLY$

MDSET is the flip-flop which synchronizes the 520 clock system with the MDC clock system. MDSET performs a dual function in that it synchronizes the clock systems during a SELECT and also a DATA TRANSFER command. MDSET gets set at the first MDC clock pulse which occurs after MDCMD becomes active, if the MDC is off-line, and will remain in the set state until the first clock pulse after the SELECT command is terminated.

3) a. s MDLCH = MDCMD • MDSET • MDLCH'

r MDLCH = MDCMD • MDSET' + MDCNL • DSC + RESET

When both MDCMD and MDSET are true, MDLCH gets set at the next clock pulse. MDLCH, when set, indicates to the MDC that it is on-line with the 520 and able to communicate. MDLCH will get reset if a selection is in process and the MDC is already on-line, if a DATA TRANSFER command to disconnect is executed, or, if the Reset button is depressed. (Reset may be either the master reset in the 520 or the MDC reset). If MDLCH is in the reset state, then communication with the MDC for anything other than status is impossible.

b. s BCS0 = MDCMD • MDSET • MDLCH' • Q13

r BSC0 = MDCMD • MDSET • MDLCH' • Q13'

s BCS1 = MDCMD • MDSET • MDLCH' • Q12

r BCS1 = MDCMD • MDSET • MDLCH' • Q12'

At the clock pulse which sets MDLCH and puts the MDC on-line with the 520, the communication channel specified by the SELECT command word (Q12 and Q13) is set into the channel assignment flip-flops (BCS0 and BCS1).

c. EKO = MDCMD • MDSET • MDLCH + MDCMD' • MDSTAT

+ MDCNL • MDSET' • DXFDLY

At this point, with MDCMD, MDSET, and MDLCH all true, an EKO is generated by the MDC and sent to the 520. EKO results in the termination of the SELECT command. Since MDCMD depends on the signals generated by the SELECT command, the termination of the SELECT command results in EKO going false. It is essential for efficient 520 peripheral operation that EKO be terminated as soon as possible after the SELECT command is terminated.

With MDCMD false, and an MDC clock pulse, MDSET is reset. Therefore, at the conclusion of the MDC selection process MDLCH is true and both MDCMD and MDSET are false.

d. MDSTAT = (QSEL) • (Q01 • Q02' • Q03' • Q04') • (Q05' • Q06')

(QSEL): SELECT; (Q01 • Q02' • Q03' • Q04'): Controller Type MDC; (Q05' • Q06'): Controller #0

+ MDCNL • QSTAT • MDLCH

R23 = MDSTAT • [MDC Power On]

R22 to R00 = Wired to particular configuration desired

A SELECT command will always send a status response to the 520 over the R bus. The 520 system clock pulse which terminates the SELECT command will also strobe the status response on the R bus into the register designated by the SELECT command.

For a timing diagram of the MDC select process, refer to Figures 3 and 4 in the Section on "Selection of an Input/Output Device."

Once on-line with the 520, any further communications with the MDC is accomplished with DATA TRANSFER commands, except for a SELECT for status or a reselect to perhaps change the communication channel assignment.

Data Transfer Operation

1) MDC Interrupt Capability

The basic MDC has interrupt logic for four channels mechanized but not fully implemented. The final implementation is provided by the devices which are connected to the MDC. If the Priority Interrupt option is added to the MDC, then the final implementation of the interrupt mechanism will be dependent on this option. The basic MDC will provide the following interrupt logic:

BCR0 = MDLCH • BCS0' • BCS1' [A]

BCR1 = MDLCH • BCS0 • BCS1' [A]

BCR2 = MDLCH • BCS0' • BCS1 [A]

BCR3 = MDLCH • BCS0 • BCS1 [A]

The logic indicated by [A] will be supplied by the devices connected to the basic MDC.

2) MDC processing of a DATA TRANSFER command

a. MDCNL = MDLCH • BCS0' • BCS1' • NABL0
+ MDLCH • BCS0 • BCS1' • NABL1
+ MDLCH • BCS0' • BCS1 • NABL2
+ MDLCH • BCS0 • BCS1 • NABL3

MDCNL is a first-level function in the MDC which compares the Enable signal (NABL 0, 1, 2, or 3) generated by the DTR to the channel assignment stored in BCS0 and BCS1. If the comparison is made and the MDC is on-line (MDLCH), the DTR is then processed in the MDC.

b. s MDSET = MDCNL • DXFDLY' • DXF + MDCMD • MDLCH'
r MDSET = DXFDLY • MDBUSY' + MDCMD' • MDCNL'
DXF = DXFN + DXFOUT + EOM

MDSET is a flip-flop which synchronizes the 520 clock system with the MDC clock system. It is normally in the set state for two clock pulses during a DTR and can be used to strobe the output of the device decoder. If the MDC is busy (MDBUSY) and a DATA TRANSFER command is issued, then MDSET can not be reset until the busy condition no longer exists. This means that EKO can not be generated and the DTR will not be processed until the MDC is ready.

c. s DXFDLY = MDCNL • MDSET
r DXFDLY = MDCNL' • DXFDLY

DXFDLY is a flip-flop which essentially delays the generation of EKO for one MDC clock pulse. This allows the MDC sufficient time to fully process the DTR which includes address decoding as well as the transfer of data.

d. EKO = MDCNL • MDSET' • DXFDLY + MDCMD' • MDSTAT
+ MDCMD • MDSET • MDCLCH

EKO is generated when the DXFDLY flip-flop has reset MDSET during a DTR (MDCNL being active). The EKO will terminate the DTR command and this results in the reset of DXFDLY (MDCNL' • DXFDLY). The MDC is now ready to process another DATA TRANSFER command.

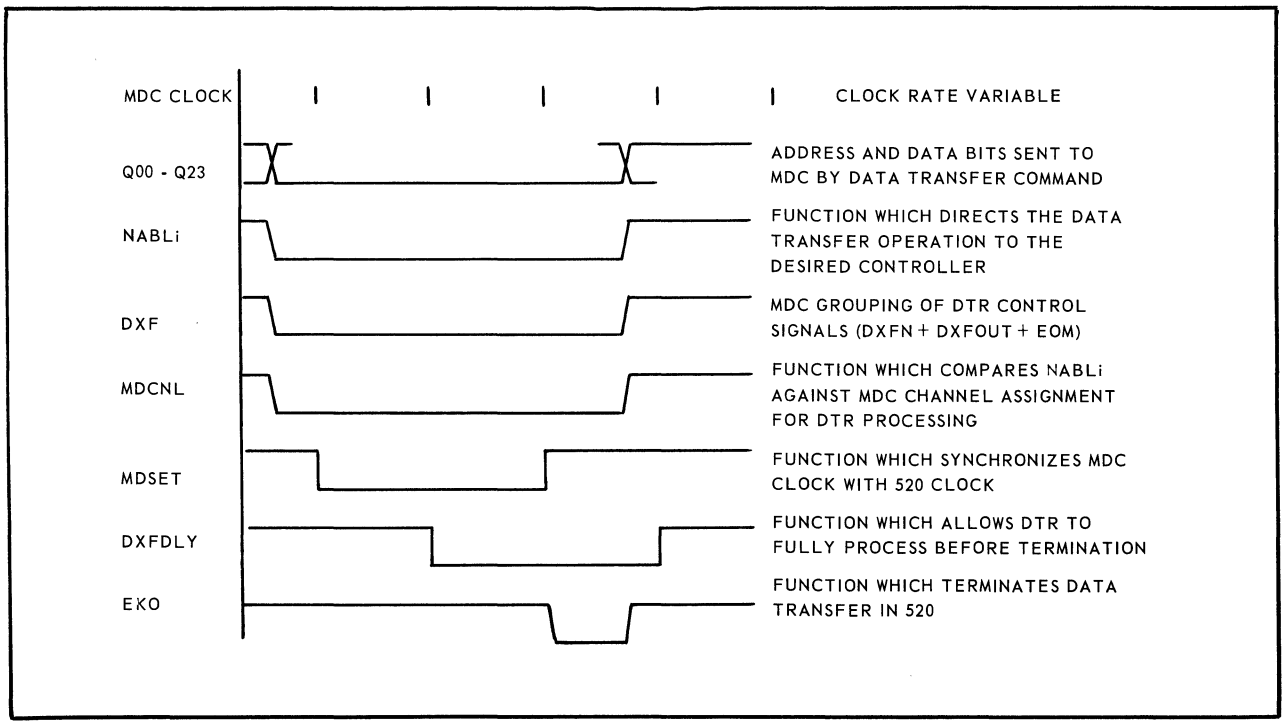


Figure 8. Data Transfer Timing in MDC

If a DTR for status is executed, MDSET and DXFDLY are not affected and the EKO is generated by MDCMD'·MDSTAT. The function MDSTAT gates the status response of the MDC onto the R bus. This is also true for the SELECT for status.

e. Since a DTR contains an address as well as data, the MDC must decode each DTR for the address of the device desired. This address decoding is accomplished with a decoding matrix module from Raytheon Computer's standard module line (GDM1). The GDM1 contains 16 four-input AND gates prewired to provide 4-bit binary to 16-line decimal code conversion. The basic MDC is wired for two GDM1's which can be gated together to provide 256 address control outputs. Only eight of the GDM1 outputs are buffered for use in the basic MDC. The GDM1 has a common input which enables the entire module. The logic function MDCNL · DXF is wired to this input for activation of the decoding matrix during a DTR.

Priority Interrupt System

One of the optional features available with the MDC is a priority interrupt system. There are two priority interrupt systems available with the MDC, a four level system and a two level system. In a systems requirement where expansion beyond 32 levels of priority is not anticipated the four level system should be used. This system is expandable in increments of 4 to 32 levels of priority interrupt. Where an expansion beyond 32 levels or an initial requirement of more than 32 levels is needed the two level system is a requirement. The two-level system is expandable in increments of 2 to 1024 levels of priority interrupt.

An interrupt generated from the priority interrupt system of the MDC will cause a departure from normal program sequence in the 520. At the completion of the instruction pair currently being executed, a transfer of program control will be effected to one of the first four locations in fast memory. The exact location depends on the communications channel assignment of the MDC. For example, if the MDC is assigned channel "0" then program control will be transferred to the instruction pair at 70000, the first location of fast memory. An MDC channel assignment "1" will result in a transfer to 70001, etc. In the event that fast memory is not part of the system configuration, a transfer to one of the same four locations, which are now assigned to main memory, will be effected.

This transfer of program control to one of the first four locations in fast memory is the entrance to a subroutine which will process the priority interrupt. Since the program location counter was not altered by this automatic transfer to one of the first four locations in fast memory, the instruction pair located there must preserve the address in the program location counter and affect a transfer to the unique memory location reserved for the interrupting level. The following program will illustrate this operation.

Location Counter	Instruction Pair	Remarks
07075	CPL A B ADL C A	Instruction pair being executed when interrupt comes active
07075 becomes 00042	STW O P DTR 3 P	Instruction pair which preserves "P" register (location counter) and transfers to unique location of interrupt (00042). This instruction pair comes from one of the first four locations in fast memory
00042		Subroutine to process interrupt
00045		
07076	STI D A BTR 6 0	Continuation of program which was interrupted. Location counter restored by subroutine and incremented by one when instruction pair brought into execution register

The unique memory locations reserved correspond to the level of the priority interrupt – level "0" goes to 00000, level "1" goes to 00001, etc. As an added option of the MDC, the memory locations reserved for priority interrupts can be assigned to any block in the 520 memory system. The size of the block depends on the number of priority levels as each level has a memory location reserved.

Once the 520 is processing a priority interrupt, it is not possible for a lower priority to gain program control. The fact that a lower priority interrupt exists will be stored and acted upon when it becomes the interrupt with the highest priority of those waiting to be processed. If, however, an interrupt occurs

which has a higher priority than the one in process, program control will be gained by the interrupt with the higher priority. This interrupt will be processed and, when completed, will return control to the interrupt subroutine of lower priority.

When an interrupt is being processed on a level and the same level has another interrupt, it will be ignored. Therefore, until the computer completes the subroutine for an interrupt on a level it will not respond to any interrupt or interrupts on the same level.

Also available with the MDC priority interrupt system is the ability to allow or disallow an interrupt on any level or combination of levels. Disarming an interrupt on a level does not even allow the presence of an interrupt to be stored by the priority interrupt system for later processing.

All of the communications between the 520 and the priority interrupt system are accomplished with the DATA TRANSFER command. The DATA TRANSFER command sends to the MDC a word which has an address and binary data. The address specifies the action desired in the priority interrupt system and the binary data specifies the particular portion of the priority interrupt system affected.

1. FOUR-LEVEL PRIORITY INTERRUPT SYSTEM

Each interrupt level has two flip-flops associated with it to indicate the present status of the interrupt level as illustrated below.

Request	Service	Action
0	0	No Interrupt present or level disabled
1	0	An Interrupt is present but not being processed
1	1	Level is sending an Interrupt to 520 for service
0	1	Interrupt is being serviced by 520

When an interrupt is received on a level, the REQUEST flip-flop is set if that level is not presently processing an interrupt. Once in the set state, the priority level control matrix examines all levels to see if a higher priority is being requested or serviced, or, if any level is now sending an interrupt to the 520. If none of these conditions exist, the SERVICE flip-flop is set and an interrupt is sent to the 520. The 520 responds to the interrupt with a DATA TRANSFER command which acknowledges the interrupt and causes the unique memory location associated with that level to be sent to the 520. This DATA TRANSFER command will reset the REQUEST flip-flop and terminate the interrupt from the MDC to the 520. Once the 520 has serviced the interrupt, a DATA TRANSFER command will be executed which will reset the SERVICE flip-flop and allow the level to process another interrupt.

2. TWO-LEVEL PRIORITY INTERRUPT SYSTEM

Each interrupt level has five possible states. They are Waiting, Requesting, Servicing, Completed, and Disarmed.

If a level is in the Waiting state, there is no interrupt on that level. When the level receives an interrupt, there will be an advance to the Requesting state. The requesting level is examined by the priority interrupt system and, if there is no higher level either requesting or being serviced, an interrupt from the priority interrupt system to the computer will be generated. At the same time the requesting level will generate a 10-bit address unique to the requesting level. If a higher priority interrupt is received before the computer acknowledges the present interrupt, the address will be changed to the memory location assigned the higher priority.

When the 520 responds to the interrupt and sends an acknowledgement, the address of the requesting level will be sent to the computer, the interrupt from the priority interrupt system to the 520 will be terminated, and the priority interrupt system will advance the acknowledged level from the Requesting state to the Servicing state. The 520 is now processing the interrupt.

When the 520 has completed the processing of a level interrupt, it will communicate with the priority interrupt system and indicate completion. This will advance the interrupting level from the Servicing state to the Completed state. If, at this time, the interrupt to the level is no longer present, the level will advance from the Completed state to the Waiting state.

The 520 can disarm the priority interrupt system on any level or combination of levels. The Disarmed state blocks any communications to or from the priority interrupt system on the affected level or levels. The priority interrupt system can be put in the Disarmed state from any of the other four states and will remain this way until released by the computer. Release from the Disarmed state will put the level or levels in the Waiting state.

3. QUICK RESPONSE – TOP PRIORITY INTERRUPT

For fast data acquisition, e.g., PCM data, through the MDC, a free-running mode is available. The 520 can go into free-running mode by setting the free-running mode flip-flop with a Set Toggle True command. The MDC must have a channel assignment of 0. When an automatic interrupt occurs on channel 0 the instruction pair from 70000 (the first location of fast memory) is gated into the execution register. This instruction pair will bring the data into the 520 from the MDC, store this data into main memory, and increment the store address by one. The free running mode locks the memory addressing for the next instruction pair to 70000. This instruction pair will be executed until the MDC indicates end of data transmission at which time program control returns to the interrupted program. This input loop in sustained operation brings data into the 520 and stores it in memory every three μsec (1 μsec memory) for 333 KC or every four μsec (2 μsec memory) for a 250 KC input word rate.

The program below will illustrate the free-running mode of operation.

LOCATION COUNTER	INSTRUCTION PAIR	REMARKS
	XXX	
A	XXX	
	STT	← Set up free-running mode
A + 1	XXX	
	XXX	← Top Priority Interrupt
A + 2	XXX	
	XXX	
A + 3	XXX	

After the execution of the instruction pair at A + 2 the automatic interrupt on channel 0 brings the instruction pair from 70000 into the execution register. The location counter remains at A + 3 and the instruction pair from 70000 gets executed until the end of the data transmission.

70000	1 μs	DTR 3 D	D ← PCM word
	1 μs/2 μs	STI L D	MEMORY ← D and Increment word count
	1 μs	INSTRUCTION PAIR ACCESS FROM 70000	
TOTAL	3 μs (1 μs memory)/4 μs (2 μs memory)		

This instruction pair will be executed until the MDC indicates end of data transmission with a skip condition. The skip condition with the DTR will cause program sequence to continue from A + 3.

LOGICAL MECHANIZATION

All 520 Input/Output Controllers are mechanized with standard Raytheon Computer digital circuit modules. (Refer to Raytheon Computer Manual SP-173A for module type, parameters, etc.) In general, SET, TRUE OR ACTIVE SIGNALS correspond to the –10 volt level; RESET, FALSE to the 0 volt level.

Communications between the 520 and all I/O controllers is accomplished with current mode signal transmission. The Current Mode Driver (#516899) will accept standard Raytheon 1.0 MC levels (0 Volts and –10 Volts) at the input and will convert these signals to current mode. This output current mode appears as a differential voltage and is used to drive the Current Mode Receiver (#516896). The Current Mode Receiver will convert this input to standard Raytheon 1.0 MC levels.

1. Current Mode Driver (4 stages per card)

Input

Frequency:	0 - 1.0 MC (up to 100 ft. of cable)
Voltage levels:	TRUE –9 to –12 Volts (–10 Volts nominal)
	FALSE 0 to –1 Volts (0 Volts nominal)
RISE time:	1.5 microseconds (maximum)
FALL time:	1.5 microseconds (maximum)

Output

Drive Capability:	1 Receiver (#516896) at 100 feet (up to 10 Receivers maximum)
Voltage levels:	TRUE 0.7 to 1.2 Volts
	FALSE 0 to 0.3 Volts

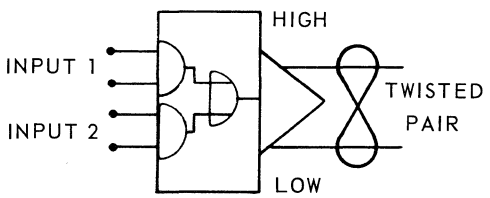
Power Requirements

–12 VDC with 183 milliamps maximum
+6 VDC with 12 milliamps maximum

Mechanical Configuration

3-3/4" H x 4-1/4" W x 1/16" T epoxy glass card, 35-pin Varicon

Symbolic Representation



2. Current Mode Receiver (4 stages per card)

Input

Frequency: 0 - 1.0 MC (up to 100 ft.of cable)

Voltage levels: TRUE 0.5 to 1.2 Volts (1.0 Volts nominal)
FALSE 0 to 0.1 Volts (0 Volts nominal)

Output

Drive Capability: 35 milliamps

Voltage levels: TRUE −9 to −12 Volts (−10 Volts nominal)
FALSE 0 to −1 Volts (0 Volts nominal)

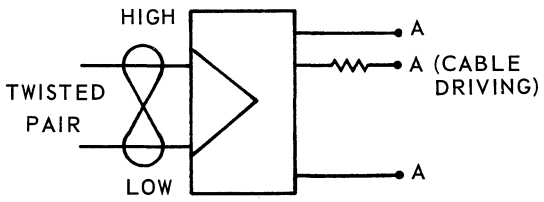
Power Requirements

−12 VDC with 300 milliamps maximum
+6 VDC with 45 milliamps maximum

Mechanical Configuration

3-3/4" H x 4-1/4" W x 1/16" T epoxy class card, 35-pin Varicon

Symbolic Representation



The TRUE side has two outputs. Either one or the other but not both may be used.

CABLE REQUIREMENTS

Each controller added to the 520 input/output system requires three cables. There are cables for input (R bus), output (Q bus), and control signals. Cables are 100 ohm twisted pair (25 pair) terminated by 120 ohms at both extreme ends. The 520 end of the cable is terminated in the 520 system (Input/Output Control Section). The last controller added to the input/output system will terminate the other end of the cable. A 120 ohm Cable Terminator plug is provided with the 520 system for this purpose (#517750).

All connectors in the 520 input/output system are Elco Varicon (#530501). All the plugs for connecting cables in the 520 input/output system are Elco Varicon (#530500).

Two connectors must normally be provided at the controller for each of the three cables, with twisted pairs from each connector to current mode drivers or receivers. Two connectors for each cable are necessary for input/output system expansion. This is illustrated in Figure 10.

A typical input/output communication pair is shown in Figure 11.

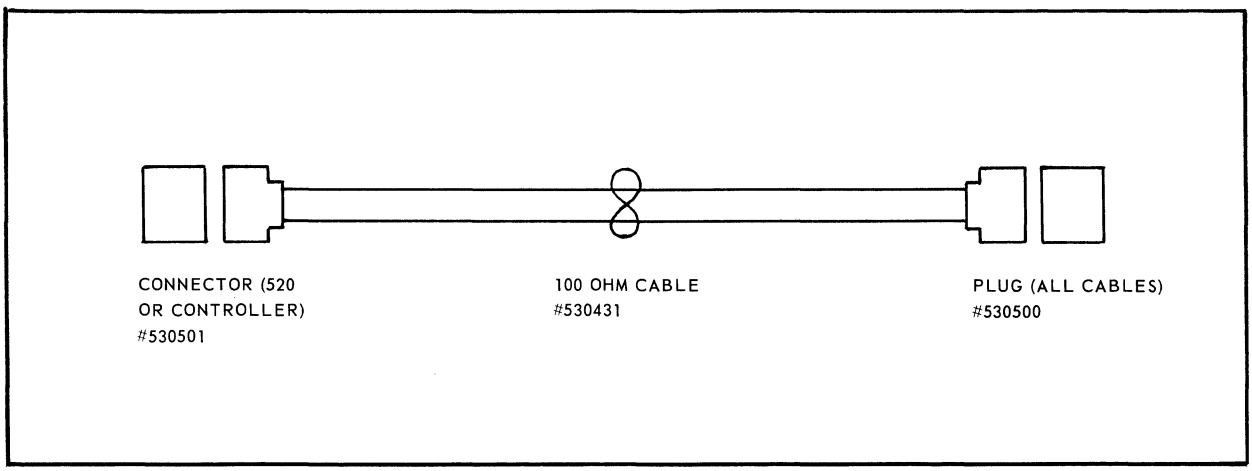


Figure 9. Cable, Multiconductor, 25 Twisted Pair, #24 AWG

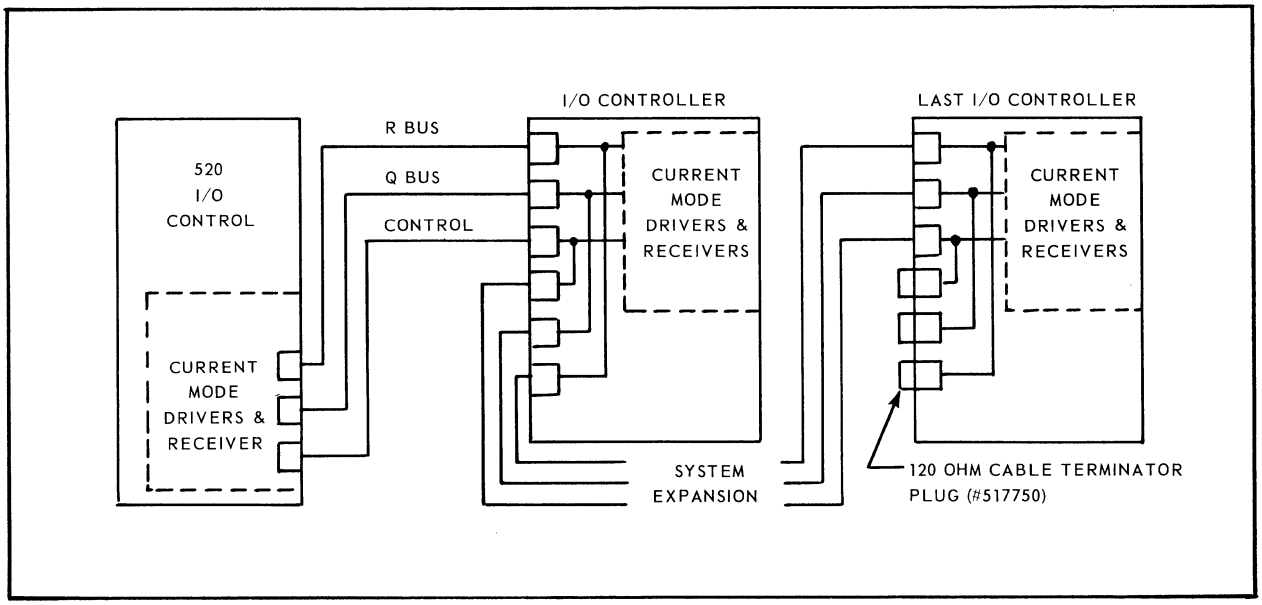


Figure 10. System Expansion Interconnection

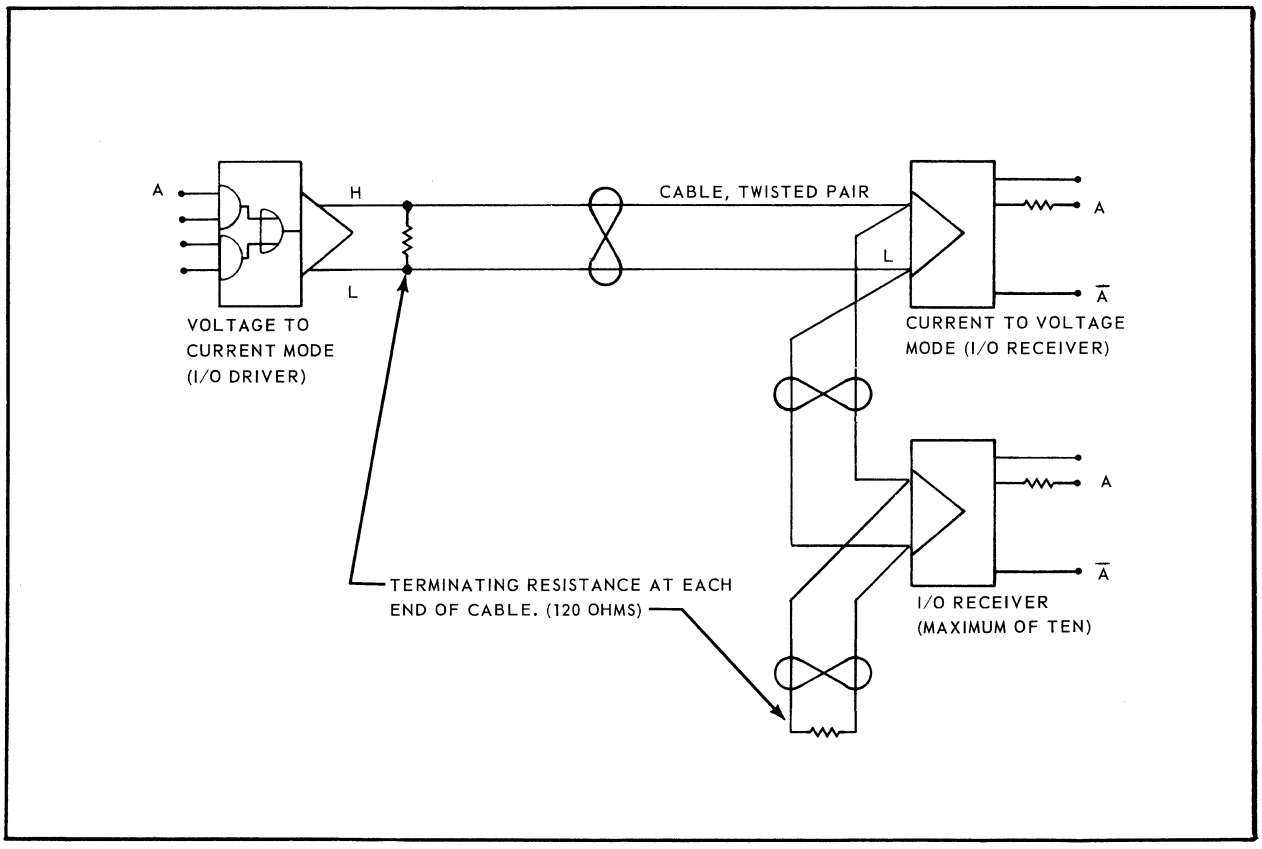


Figure 11. Typical Input/Output Communication Pair

TABLE 1

520 INPUT/OUTPUT

CONNECTOR ASSIGNMENTS

CONNECTOR NUMBER	FUNCTION
01J07	INPUT BUS - R BUS
01J08	INPUT BUS - R BUS
03J03	OUTPUT BUS - Q BUS
03J04	OUTPUT BUS - Q BUS
03J05	CONTROL SIGNALS (I/O)
03J06	CONTROL SIGNALS (I/O)

TABLE 2

PIN ASSIGNMENTS FOR INPUT BUS

PINS	SIGNAL	PINS	SIGNALS
A - B	R00	e - h	R12
C - D	R01	f - j	R13
E - F	R02	k - l	R14
J - K	R03	m - n	R15
L - M	R04	p - r	R16
N - P	R05	s - t	R17
R - S	R06	u - v	R18
U - V	R07	x - y	R19
W - X	R08	z - AA	R20
Y - Z	R09	BB - CC	R21
a - b	R10	DD - EE	R22
c - d	R11	HH - JJ	R23
		KK - LL	GND

NOTE: LEFT PINS ARE SIGNAL LOW, RIGHT PINS ARE SIGNAL HIGH, WHICH ARE TWISTED TOGETHER.

TABLE 3

PIN ASSIGNMENTS FOR OUTPUT BUS

PINS	SIGNAL	PINS	SIGNALS
A - B	Q00	e - h	Q12
C - D	Q01	f - j	Q13
E - F	Q02	k - l	Q14
J - K	Q03	m - n	Q15
L - M	Q04	p - r	Q16
N - P	Q05	s - t	Q17
R - S	Q06	u - v	Q18
U - V	Q07	x - y	Q19
W - X	Q08	z - AA	Q20
Y - Z	Q09	BB - CC	Q21
a - b	Q10	DD - EE	Q22
c - d	Q11	HH - JJ	Q23
		KK - LL	GND

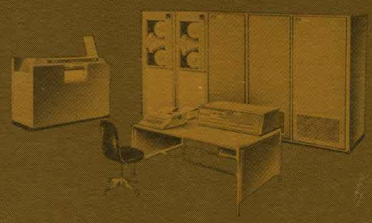
NOTE: LEFT PINS ARE SIGNAL LOW, RIGHT PINS ARE SIGNAL HIGH, WHICH ARE TWISTED TOGETHER.

TABLE 4

PIN ASSIGNMENTS FOR CONTROL SIGNALS

PINS	SIGNAL	PINS	SIGNALS
A - B	NABL0	e - h	INDICATOR
C - D	NABL1	f - j	INDICATOR
E - F	NABL2	k - l	R24
J - K	NABL3	m - n	BCREQ
L - M	QSTAT	p - r	EKO
N - P	QSEL	s - t	SPL
R - S	EOM	u - v	BCR0
U - V	DSC	x - y	BCR1
W - X	DXFN	z - AA	BCR2
Y - Z	DXFOUT	BB - CC	BCR3
a - b	Q24	DD - EE	GND
c - d	QRSET	HH - JJ	GND
		LL - KK	INDICATOR

NOTE: LEFT PINS ARE SIGNAL LOW, RIGHT PINS ARE SIGNAL HIGH, WHICH ARE TWISTED TOGETHER.



SALES OFFICES

Alabama

RAYTHEON COMPUTER
Holiday Office Center, Suite 47
Huntsville, Alabama 35801
Phone (205) 881-2844
TWX 510 579-2113

California

RAYTHEON COMPUTER
2700 South Fairview Street
Santa Ana, California 92704
Phone (714) 546-7160
(From Los Angeles 625-7645)
TWX 714 546-0444

Massachusetts

RAYTHEON COMPANY
Bedford Laboratory
P. O. Box 508
Bedford, Mass. 01730
Phone: (617) 274-7100, Ext. 643 and 644
TWX 617 274-6487

Texas

RAYTHEON COMPUTER
204 East Main
Arlington, Texas
Phone (817) CR 5-5361
TWX 817 274-3917

Houston, Texas
Phone (713) WA 3-1144

Washington, D. C.
Eastern Regional Office

RAYTHEON COMPUTER
4217 Wheeler Avenue
Alexandria, Va. 22304
Phone (703) 836-7616
TWX 703 931-4247

RAYTHEON

RAYTHEON COMPUTER

2700 SOUTH FAIRVIEW ST., SANTA ANA, CALIFORNIA 92704