Development of l6-channel communication multiplexor for Hewlett Packard 2100 series computers
by Richard David Weaver

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE in Electrical Engineering
Montana State University

Abstract:
The subject of this thesis is the design, development and construction of a low-cost 16-channel communication multiplexor (MUX) used for interfacing from 1 to 16 teletypes, modems or other bit-serial devices to a single I/O channel on a Hewlett Packard 2100 series computer.

This thesis discusses the following items: First, reasons for the project and the problems to be solved are presented and discussed. Second, specifications of the hardware to be interfaced are presented, along with a detailed description of the specifications for the MUX. Third, the logic design used to realize the MUX is described. This section ends with a short summary of testing procedures used to verify correct operation of the logic. In the fourth section the physical and mechanical design and construction is presented. The thesis ends with an analysis and discussion of the overall performance and success of the project.

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Montana State University, I agree that the Library shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by my major professor, or, in his absence, by the Director of Libraries. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature Richard D. Weaver

Date May 23, 1974

DEVELOPMENT OF A 16-CHANNEL COMMUNICATION MULTIPLEXOR FOR HEWLETT PACKARD 2100 SERIES COMPUTERS

by

RICHARD DAVID WEAVER

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Paul E. Ehlrich
Head, Major Department

Kenneth E. Marcotte
Chairman, Examining Committee

Henry L. Parsons
Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

June, 1974

ACKNOWLEDGMENT

The development of a large digital system requires the resources and talent of many people. The author wishes to thank the people in the Electronics Research Laboratory at Montana State University for their help and the use of the shop, stockroom and photographic equipment.

The fabrication of printed circuit cards and sheet metal construction was done by Development Technology Incorporated of Bozeman, Montana.

The writer is deeply grateful to Western Telecomputing Corporation for financial support, facilities and equipment used in designing, building and testing the system.

R.D.W.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The subject of this thesis is the design, development and construction of a low-cost 16-channel communication multiplexor (MUX) used for interfacing from 1 to 16 teletypes, modems or other bit-serial devices to a single I/O channel on a Hewlett Packard 2100 series computer.

This thesis discusses the following items: First, reasons for the project and the problems to be solved are presented and discussed. Second, specifications of the hardware to be interfaced are presented, along with a detailed description of the specifications for the MUX. Third, the logic design used to realize the MUX is described. This section ends with a short summary of testing procedures used to verify correct operation of the logic. In the fourth section the physical and mechanical design and construction is presented. The thesis ends with an analysis and discussion of the overall performance and success of the project.

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

This thesis describes the specifications, design and construction of a 16-Channel Communication Multiplexor. The multiplexor (MUX) provides a way of interfacing from 1 to 16 bit serial (teletype) data channels to a Hewlett Packard (HP) 2100 series computer.

This thesis is organized in chapter form as described below. Chapter one provides the introduction and reasons for the project. Chapter two explains the computer interface, multiplexor and I/O device specifications. Chapter three describes the design and testing procedures used in developing the MUX. Chapter four describes the construction techniques and problems encountered in actually building a working model. Chapter five discusses the overall effectiveness of the project upon its completion. Relative cost, performance and reliability are also discussed.

1.2 BACKGROUND AND PROJECT INTRODUCTION

This project was started when it became necessary to connect 16 teletypes to an HP 2116B computer for use with Time-Share Basic. Also, a large number of serial data input channels were needed in data

acquisition systems that were being developed. There were two possible ways to do this interfacing before the MUX was constructed and both were expensive and not very satisfactory.

The first consisted of buying one teletype interface card for each channel to be interfaced. The cost per channel was high and each channel implemented required one input-output (I/O) slot in the computer. There are only 19 I/O slots total in the 2116B computer and two are used to interface a disc drive, one is required for a clock and three more for a high speed papertape reader, high speed papertape punch and console teletype. Therefore, only 13 slots were available for user teletype channels. Also the 2116B computer cannot supply enough current to that much I/O equipment so a power supply extender (also expensive) must be installed. Therefore, to interface only 13 user channels of Basic, required having 13 interface boards, a power supply extender and using every available I/O slot.

The other alternative was to buy another entire HP 2114 computer and the HP software multiplexor. This uses only one 16 parallel-bit board which brought in the 16 teletype channels, one on each of its 16 input bits. Then by sampling at a high frequency, the waveforms coming in on each line could be reconstructed by the computer. The whole I/O process completely tied up one computer making the cost for this solution very expensive. Also four 16 bits I/O cards were required for data transfer between the 2114 I/O computer and the 2116B

processor computer.

During the development of this project, HP replaced the older 2116, 2114 and 2115 series computers with the HP2100A computer. This computer, which they are still presently selling, has only 13 I/O slots total, leaving only 7 for time-share Basic users. With this computer the need for a multiplexor for time-share users was greater than ever.

The need for a multiplexing device can now be clearly seen. What is needed is some "black box" that will connect to 16 serial data channels (teletypes) and interface to the computer through only one I/O channel. This one interface to the computer, however, must not keep the computer so completely tied up in I/O that it cannot do other operations as was one of the problems with the HP software MUX.

CHAPTER II

THE SPECIFICATIONS OF THE MULTIPLEXOR

2.1 INTERFACE SPECIFICATIONS

The first step in designing the multiplexor is defining or stating in detail the specifications of the computer interface and the I/O equipment (teletype, modem or other bit-serial device). Figure 2.1 contains a block diagram of the overall system showing computer, MUX, and I/O device interconnections.

2.2 I/O COMPUTER INTERFACE SPECIFICATIONS

The computer interface selected is a standard HP 16 bit, parallel full-duplex I/O card. This gives two independent storage registers, one for 16 data output bits (from the computer) and one for 16 data input bits (to the computer). Also a "Device Command" signal is available that can be used to tell an external device to take some action. Another line, "Device Flag" can be used by the external device to signal to the computer.

The operation of the I/O interface board can then be summarized as follows: The computer can store 16 bits of information in the data output register with an OTA or OTB instruction. The contents don't change until the computer stores another 16 bits in the register. The "Device Flag" line is pulsed (for approximately 1 μsec),

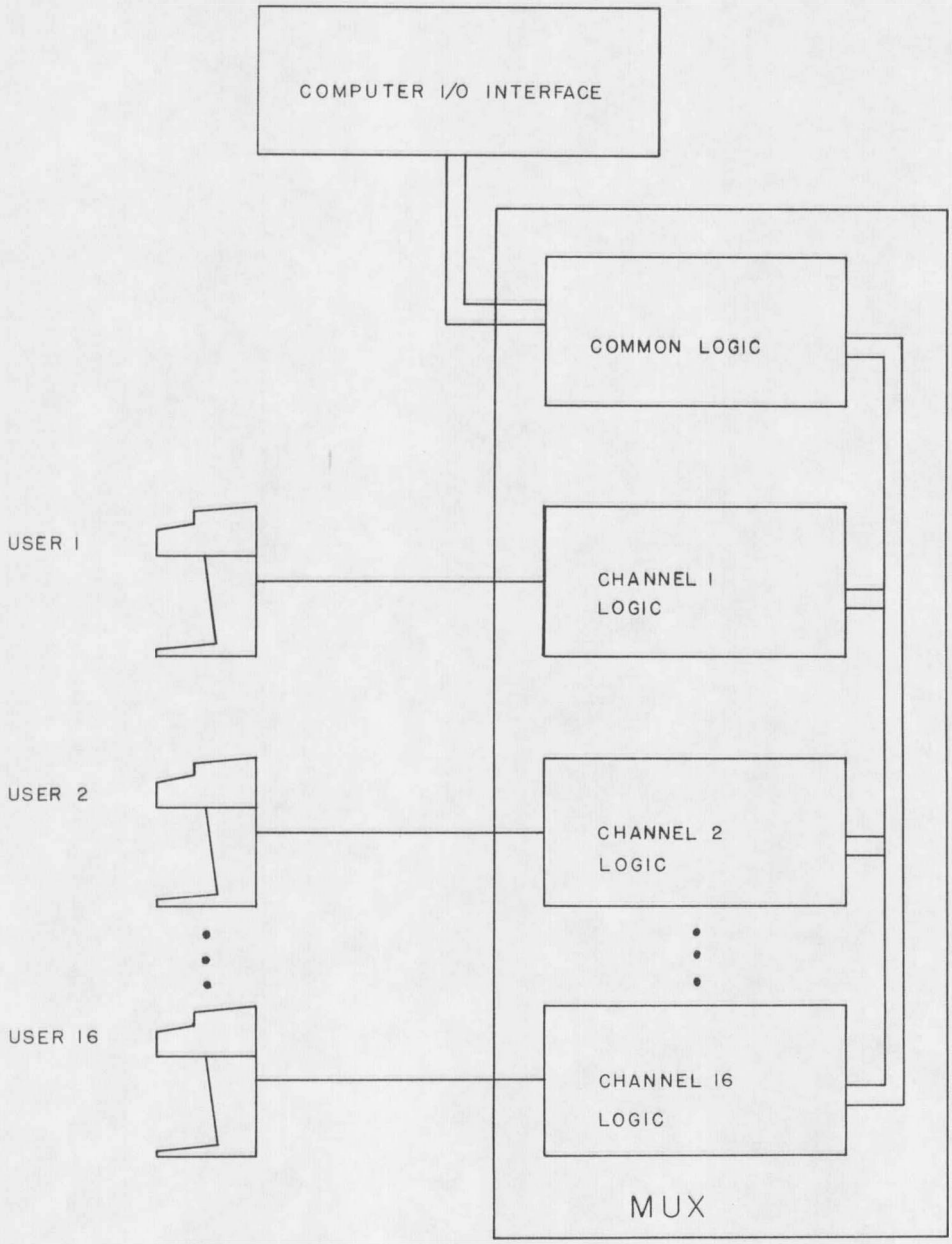


FIGURE 2.1 MUX SYSTEM BLOCK DIAGRAM

once every time a Set Control instruction is executed. Whatever data is present on the 16 data input lines coming from the external device will be loaded into the data input register when the "Device Flag" line is pulsed. This pulse also sets the I/O channel flag indicating to the computer that new data is available. The Computer can then bring in the data stored in the input register (with an LIA or LIB) and take whatever action it may desire, depending on the results of the input data word.

All signals described above are generated by standard (7400 series) TTL logic gates. This means a logic 0 = 0 to .8 volts, a logic 1 = 2 to 5 volts and voltage outside this range after transitions are undefined. The current that must be drawn from an input in the logic 0 state is 1.6 mA maximum and for a logic 1 state 10 μA must be supplied. The transition time from one logic state to the other is typically 5-50 ns. Each logic output is capable of driving at least 10 logic inputs for a fan-out of ten. The line drivers (SN7416 and SN7417) are capable of driving 20 gate inputs. The MUX was constructed using TTL logic for compatability with this computer interface.

2.3 SERIAL I/O DEVICE INTERFACE SPECIFICATIONS

At the user end of the MUX, 16 channels of serial data are to be interfaced. For each channel there is one line coming from a terminal and one going to the terminal over which the data is trans-

ferred. Figure 2.2 contains the timing diagram showing the structure of the serial data signal.

All data signals wait in the logic 1 or marking condition. When a character is to be sent, a start pulse of one bit duration is output in the logic 0 or space condition followed by the 8 information bits. The character is terminated by one or two (depending on the device) stop bits. Characters are transmitted asynchronously with the start and stop bits defining each character.

The MUX can also be interfaced directly to a modem for data transmission over telephone lines to remote users. In this case three control lines; ringing indicator, carrier detect and data terminal ready are necessary on each channel for control operations.

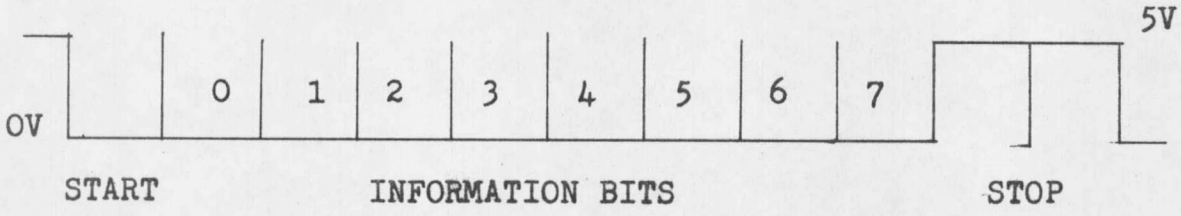
The ring indicator line is used to tell the computer when a channel is being called. The data terminal ready line is used to allow the modem to answer or once answered can terminate a call. The carrier detect tells the computer whether or not the customer has properly connected his modem for data transmission.

Therefore, for each channel two serial data lines, a ring indicator line, carrier detect line and data terminal ready are required.

Modems and many bit-serial I/O devices operate using EIA RS232 signal levels. These logic levels require a logic 1 to be in the range of +3V to +25V and a logic 0 to be in the range of -3V to -25V although 0V to -25V is often used for a logic 0. The TTL levels are

SERIAL DATA FORMAT

(GROUND TRUE TTL SIGNAL)



STANDARD BIT TIMES

110 BAUD-9.09 ms/BIT

300 BAUD-3.33 ms/BIT

FIGURE 2.2

also widely used so both signal types (EIA and TTL) are made available.

2.4 MULTIPLEXOR SPECIFICATIONS

The specification of the MUX can now proceed and must mesh effectively with the operation of the computer interface board and the data terminals and modems. The first step will define the meanings and operations associated with the 16 bit computer output word. Then the 16 bit computer data input word supplied by the MUX must be defined.

It is in these specifications that the overall efficiency of the MUX will be determined. If they are awkward the MUX will be hard to program and the computer will spend large amounts of time processing I/O. Remember the design is to be used with only one computer and 16 users all running time-share Basic and the time devoted to executing programs must be maximized. Also inefficient specifications can lead to more complex logic realizations for a given job and hence the overall product will cost more than necessary.

2.5 THE DATA OUTPUT WORD

The first step in the MUX specification is defining the computer output word. By outputting a series of these words the computer controls the entire operation of the MUX.

In Figure 2.3 it can be seen that all necessary data and information can be supplied to one MUX channel in one computer step of loading the output register of the I/O card. The "Device Command" signal is then pulsed to tell the MUX that the data is correct and can be accepted. Now from a programming point of view, the operation of outputting information to the MUX is a three-step process. First, the 16 bit data word is assembled in the computer. Second, it is stored on the I/O register card where the signals are actually transferred to the MUX. Then the Set Control instruction is executed which pulses "Device Command" causing the MUX to accept the new data.

2.6 "MODE" CONTROL

There are two types of output to a channel as illustrated in Figure 2.3. The first type sets up the "Mode" of the channel and is determined by bit 15 = 1 and bit 0 = 0.

Bits 8 through 11 determine the channel address in the following manner. If all four bits = 0 then channel 1 is accessed. Then by counting up in binary, each channel is selected in turn, according to its binary representation. With four bits, $2^4 = 16$ combinations are possible, which is the required number. As an example, an output of 0110 for bits 11 through 8 would select channel 7.

When setting the "mode", bits 1-7 are ignored by the channel. Bits 12 through 14 determine the "mode" or state of a channel and

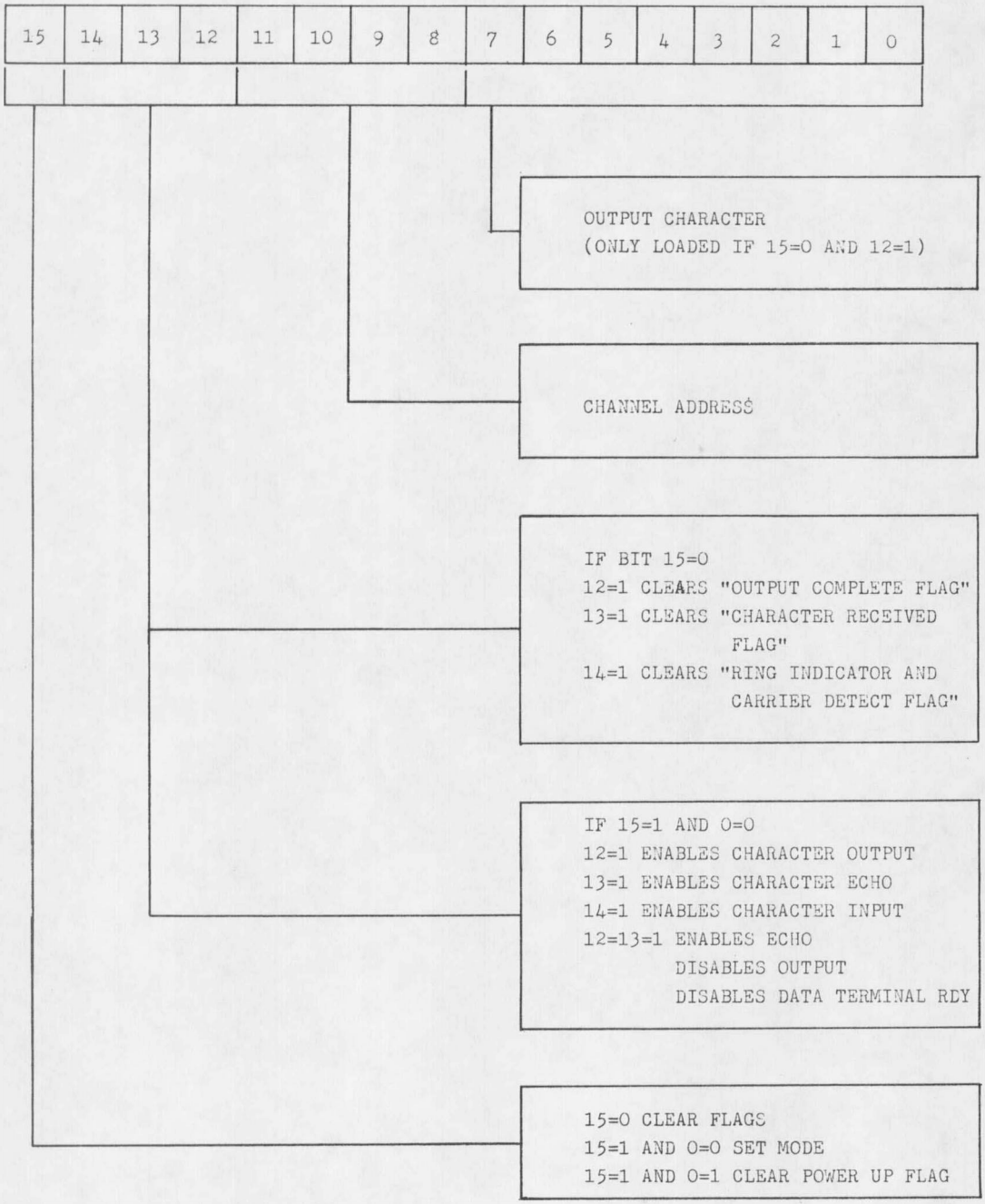


FIGURE 2.3 COMPUTER OUTPUT WORD BIT DEFINITION

must determine the following items; the state of data terminal ready, the enable or disable of input, the enable or disable of output and one other feature called echo. In the echo-on mode, data coming in on the serial data-in line is wrapped around and transmitted back out on the serial data-out line. In this case data should not simultaneously be output by the computer through the output register because an "oring" of the signals would result. The other possibility is echo-off. In this mode data coming in on the serial data-in line is kept completely separate from that being output by the computer on the serial data-out line. This independent transmitting and receiving of data is called Full-Duplex operation.

The modes are determined by the bit patterns as shown in Figure 2.3 and described below. Bit 12 = 1 enables output and bit 12 = 0 disables output. Bit 13 = 1 enables echo and bit 13 = 0 disables echo. Bit 14 = 1 enables input and bit 14 = 0 disables input. One combination that should not be allowed as discussed above is output enabled and echo on. For this case (bit 12 = 1, bit 13 = 1) echo is enabled output is disabled and data terminal ready is false. This mode would not normally be used while a user is accessing the system and provides the perfect case to terminate a call by setting data terminal ready false. All other bit combinations set data terminal ready true.

2.7 CLEARING FLAGS AND STARTING OUTPUT

The second type of output to a channel, also shown in Figure 2.3, is that of clearing flags and outputting characters. This is done by setting bit 15 = 0. There are three flags required which are set as follows: When input is complete the "character received" flag is set; when output is complete the "output complete" flag is set; and when the ringing indicator or carrier detect signals change state, the third flag is set. Remember that the data coming into or out of a channel to the I/O device or modem is in serial form. Therefore, each channel must perform a serial-to-parallel or parallel-to-serial bit conversion. This takes, by computer reference, a very long time and during a conversion the computer must be free to do other tasks and then be told when the various operations are complete.

To clear the flags the data word is assembled in the computer, output to the MUX through the I/O card and then a Set Control instruction is executed as described before. Bits 8 through 11 select the channel. Bit 14 = 1 clears the "ring indicator and carrier detect flag", bit 13 = 1 clears the "character received flag" and bit 12 = 1 clears the "output complete flag". When the "output complete flag" is cleared, the output shift register is loaded with bits 0-7 and a new serial output started <u>if</u> a valid output mode was set up ahead of time.

2.8 CLEARING THE POWER FAIL FLAG

If, for any reason, the MUX should have a power failure or power is initially turned on, the power fail flag will set. This causes the computer to load the input status word informing the computer of the power failure. To clear the flag an output word of 1XXXXXXXXXXXXXX1 (bit 15 and bit 0 = 1, all others = 1 or 0) is loaded followed by the usual Set Control instruction. After this operation the MUX is ready for normal use.

2.9 THE DATA INPUT WORD

The second part of the MUX specification defines the 16 bit computer input word. There is only one meaning attached to each bit of the input word and is diagrammed in Figure 2.4.

There is a design problem still to be solved that must determine which channel shall be allowed to input data if there should be a multiple request for service. But, assuming one has been selected, "Device Flag" can be pulsed by the MUX to load the data into the computer. The definition of bit meaning is given in Figure 2.4 and described below.

Bits 8-11 determine the channel requesting service as described in the last section. Bit 12 is used to indicate the completion of a data output (parallel-to-serial bit-conversion) and bit 13 the completion of a data input (serial-to-parallel) operation. If bit 13

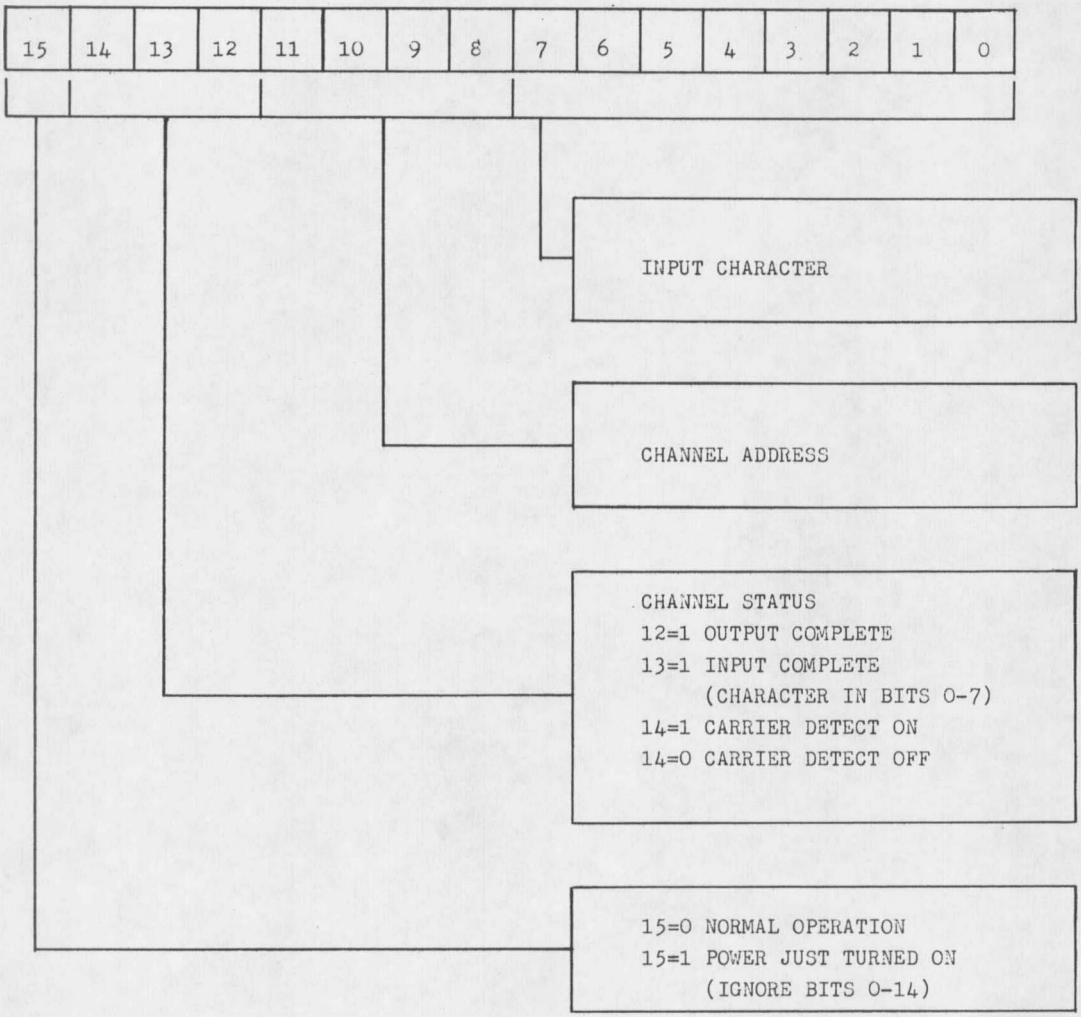


FIGURE 2.4 COMPUTER DATA INPUT WORD BIT DEFINITION

indicates input complete, the character is contained in bits 0-7. Once a flag is set it must be cleared by the computer as discussed in Section 2.7. This acknowledges that the information was received and enables other channels to request service.

The ring indicator and carrier detect functions are handled as follows. When either line changes state (0 to 1 or 1 to 0 transition), a flag is set requesting service. Bit 14 of the input word, however, contains only the state of carrier detect. Therefore, if a channel requests service and bits 12 and 13 are not set, the third flag set by ring indicator or carrier status must be assumed. If the state of carrier detect has changed, then this flag was set by the carrier detect signal. If, however, the carrier detect has not changed state, then the operation of the ring indicator line can be deduced.

If during any input operation bit 15 = 1, then the MUX has had a power failure. Upon return of the power, the power fail flag is set. This resets all channels to a start condition and then pulses "Device Command" to tell the computer of the new status. The computer must then respond by clearing the power fail flag as discussed under the output word (Section 2.8) which enables the MUX for normal operation.

2.10 SERIAL DATA CONVERSION

As discussed in Section 2.3 any data entering or leaving a

channel must be in the serial data format shown in Figure 2.2. The data transmitted or received by the computer I/O interface, however, is done in 16 parallel bit transfers as discussed in Sections 2.4 - 2.9. Each channel therefore requires an output shift register that can store the computer output character and shift the character out serially to the I/O device.

When entering serial data another shift register is necessary to shift in and store input characters. After converting the input character to parallel form, the computer can load the character into the I/O interface card as discussed in Section 2.9.

The character input and output functions are enabled, disabled and initiated as discussed in Sections 2.4 - 2.9.

2.11 PRIORITY INTERRUPT SYSTEM

Whenever a channel receives a character, completes a character output, or detects a change in carrier detect or ring indicator, the status information defined in Section 2.9 must be entered into the computer I/O interface for processing. In general, many channels will be asking to enter status simultaneously or before other channels have completed loading data.

To prevent this problem a priority scheme is incorporated into the logic that pulses the DF line preventing more than one channel from requesting service and loading data onto the input bus simul-

taneously. This logic works in the following manner: The first channel to request service sets a flag preventing any other channel from requesting service. If two or more channels set their request service flags simultaneously, a daisy-chained priority line disables all channels of lower priority. Then as each channel is processed the next lower priority channel is enabled to request service. Upon completion of processing all channels with request service flags set, the next channel (or channels) desiring service will be enabled and the process repeats.

2.12 SUMMARY

This concludes the specifications of the MUX. The specifications for the equipment to be interfaced have been presented and the control of the MUX was specified by giving each of the 16 output bits (from the computer) a special job. Next the encoding of the status of each channel of the MUX was described in terms of the 16-bit computer input word, and operation of serial data conversion was discussed, along with the request for service operations. The foundation has now been set upon which the actual design, discussed in the next chapter, can be implemented.

CHAPTER III

THE HARDWARE DESIGN OF THE MULTIPLEXOR

3.1 INTRODUCTION

The specifications of the last chapter provide the information for implementing the actual hardware logic design. Section 3.4 - 3.5 describe the final logic realizations used to implement those specifications. Section 3.6 describes how the design was tested and debugged.

3.2 LOGIC DIVISIONS

As shown in Figure 2.1 the logic circuits for the Multiplexor fall into two categories. The first are those logic functions common to all 16 channels. These include channel address encoding and decoding, signal buffering and other common signal decoding and encoding functions.

The second group provides the logic functions common only to one channel. These include the character input and output logic, mode latches and associated logic, priority and request service logic, flags, and buffering and interfacing logic for the user signals. This second group of logic is identical for each channel and is repeated once for each channel implemented.

3.3 SIGNAL DEFINITIONS

All signals will be assumed plus true, This means a logic 1 = 2V to 5V and a logic 0 = 0V to .8V for TTL logic. For EIA level signals a logic 1 = 3V to 25V and a logic 0 = -3V to -25V.

All signal names are from one to three characters long. If a signal is ground true a bar will be placed over the signal name. As an example, all 16 bits of the data output word coming from the computer interface board are ground true and have the following names:

$\overline{0}$ - (bit zero of output word; $\overline{0}$ = 0V to .8V when at a logic 1)

$\overline{1}$ - (bit 1 of output word)

$\overline{2}$ - (bit 2 of output word)

.
.
.

$\overline{15}$ - (bit 15 of output word)

Three other signal name examples found in Figure 3.1 are the clear flags, mode set and power fail clear signals, defined as follows:

$\overline{MS}$ - ("set mode" if $\overline{MS}$ = 0V to .8V)

$\overline{FC}$ - ("clear flags" if $\overline{FC}$ = 0V to .8V)

$\overline{PFC}$ - (clear power fail flag is $\overline{PFC}$ = 0V to .8V)

3.4 COMMON LOGIC FUNCTIONS

The common logic group, shown in Figure 3.1, handles the following operations: (1) The channel address is decoded from the computer output word and one of the 16 channels is enabled ($\overline{CS0}$-$\overline{CS15}$) for an operation. (2) Two signals, $\overline{MS}$ and $\overline{FC}$, are decoded for use by the channel logic for setting the mode of a channel and clearing flags on the channel. (3) A signal called PR is generated when power is first turned on. This signal resets certain portions of the logic on each channel board to a starting condition. Also when the power is first turned on, computer output bits $\overline{14}$, $\overline{13}$, and $\overline{12}$ are forced to a logic 1 to reset flags and modes in each of the channel logic groups. (4) When a channel requests service from the computer, device flag (DF) must be pulsed to set the flag on the computer interface card. This signal must also be pulsed when power is turned on to indicate to the computer that the MUX is ready for operation. (5) When power is first turned on, bit 15 of the computer data input word ($\overline{15I}$) must be made a logical 1. (6) The channel address must be encoded in the computer input word by the channel that is presently requesting service ($\overline{RS1}$-$\overline{RS15}$). (7) Signals ($\overline{0I}$-$\overline{15I}$) going to the computer interface must be buffered with cable line drives.

The implementation of these seven functions can be seen in Figure 3.1. On the left hand side of the drawing are the output line drivers and input line receivers on the I/O interface card

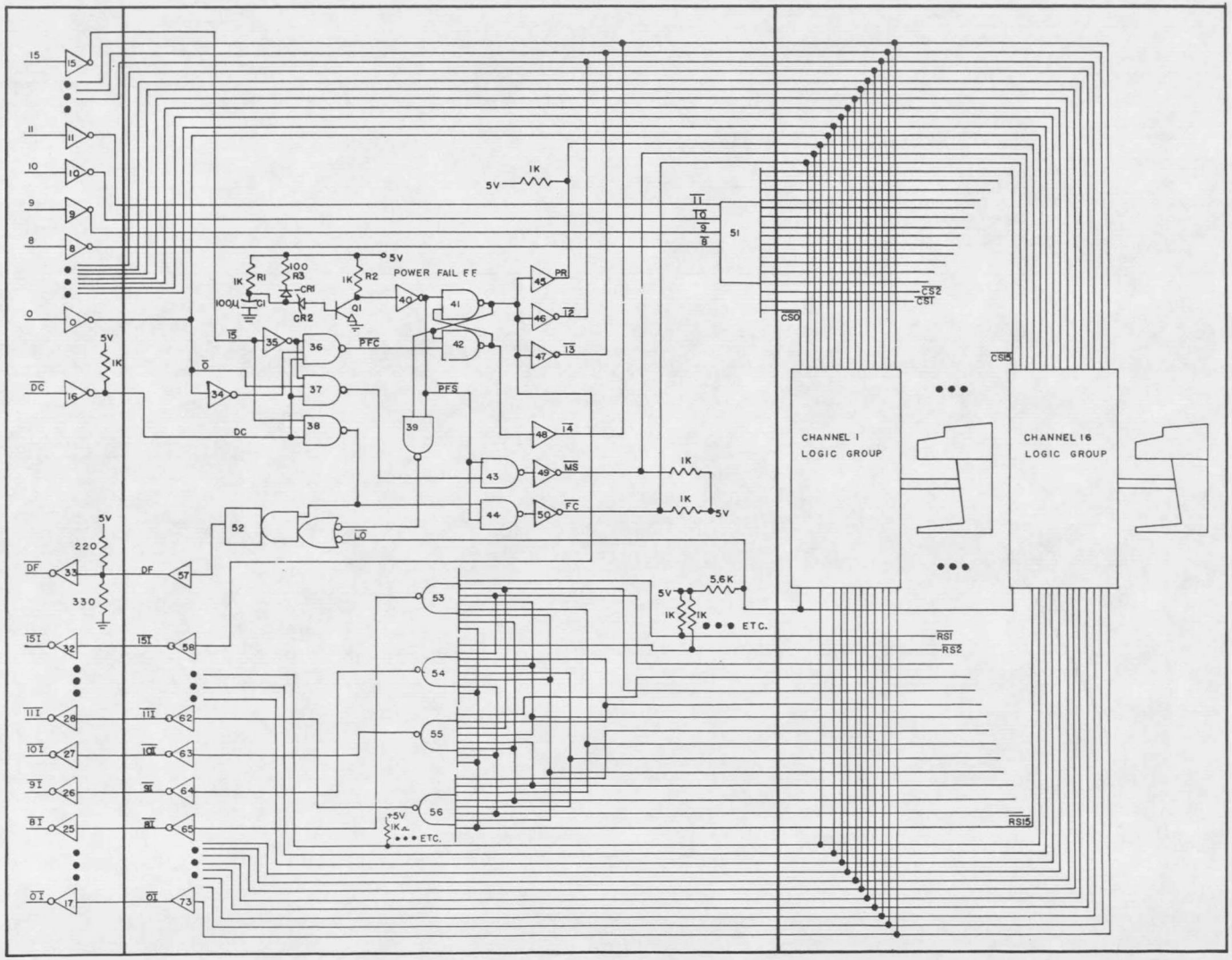


FIGURE 3.1 COMMON LOGIC FUNCTIONS

located in the computer. The center section shows the common logic functions of the multiplexor. The right hand side of the drawing illustrates the bussing to each of the sixteen channel logic groups. Next is a detailed description of the operation of the common logic using Figure 3.1 as a reference.

The output bits 8-11 from the computer (signals $\overline{8}$, $\overline{9}$, $\overline{10}$, $\overline{11}$) are decoded in IC51 and one of the 16 channel select lines ($\overline{CSXX}$) is enabled. This IC (SN74154) enables $\overline{CS0}$ if a binary address of 0000 is entered on lines $\overline{11}$, $\overline{10}$, $\overline{9}$, and $\overline{8}$. $\overline{CS1}$ is enabled if a binary address of 0001 is entered. This continues with each $\overline{CSXX}$ line selected in turn when the appropriate output address from bits $\overline{11}$, $\overline{10}$, $\overline{9}$, and $\overline{8}$ are entered.

The operation of the clear flag line ($\overline{FC}$) occurs when the device command (CD) is at a logical 1 (pulsed by a STC computer instruction) and output bit 15 (signal $\overline{15}$) is at a logic 0. Then gate 38 goes low giving a high level out of gate 44 which activates $\overline{FC}$ from line driver 50 (SN7416). Also, $\overline{FC}$ is activated when the power-fail flip-flop is being set (signal $\overline{PFS}$) by inverter 40. This causes the flags to be reset when power is initially applied.

In a similar manner the mode set signal ($\overline{MS}$) is activated when gate 37 goes low because output bit 15 ($\overline{15}$) is at a logical 1, output bit 0 is at a 0 ($\overline{0}$) and the device command signal (DC) is activated. This causes a high level from gate 43 which activates

$\overline{MS}$ from line drivers 49. Again the setting of the power-fail flag ($\overline{PFS}$) also activates the $\overline{MS}$ signal to reset the mode flip-flops to an initial starting condition when power is turned on.

The circuit to generate $\overline{PFS}$ operates as follows: When power is first turned on, resistor R1 and capacitor C1 provide a .1 second delay before charging C1 sufficiently to turn transistor Q1 "on" through the 3 volt zener diode CR2. CR1 and R3 are used to discharge C1 rapidly when the power is turned off. Hex inverter 40 provides correct polarity and squaring of the output from the collector of Q1 and sets the power-fail FF.

When set, the output of the power-fail FF is used to force $\overline{I4}$, $\overline{I3}$ and $\overline{I2}$ to a logical 1 through line drivers 46, 47 and 48 which are "wire ored" with the three outputs on the computer interface card. These three lines must be at a logical 1 to clear the three flags on each channel when the $\overline{FC}$ line is activated. Also this forces the mode of each channel into "input-on", "echo-on", "output-off", and data terminal ready false. The PR line is enabled by line driver 45 which resets certain portions of the channel logic to a starting condition.

Bit 15 (signal $\overline{I5I}$) of the computer input word is also set to a logical 1 through driver 58 to indicate that power has just been applied. Gate 39 causes one shot 52 (SN74121) to trigger (one microsecond pulse) at the end of the .1 sec power up delay. This sets

the flag on the I/O computer card which signals to the computer to load the data input word for processing.

The power-fail FF can then be cleared by signal $\overline{PFC}$, generated by gate 36 when signals DC, $\overline{15}$ and $\overline{0}$ are all at a logical 1. After the power-fail FF has been cleared the MUX is ready for operation.

When the MUX is ready to enter data into the computer, the $\overline{LO}$ signal is activated by the channel desiring service. This triggers one shot 52 which activates DF and sets the flag on the computer I/O card. The channel requesting service also activates an $\overline{RSXX}$ signal. $\overline{RS15}$ is activated by channel 16, $\overline{RS14}$ by channel 15, etc. ending with channel 2 because channel 1 doesn't activate a line. Gates 53, 54, 55 and 56 generate the appropriate binary address depending on which $\overline{RSXX}$ line is activated. By default if no $\overline{RSXX}$ line is activated binary address 0 (channel 1) is generated. The channel logic enters the appropriate data for input bits 0-7, and 12-14 (signals $\overline{0I}$ - $\overline{7I}$, $\overline{12I}$ - $\overline{14I}$). When the computer loads the input data the channel address, character and flags are entered by line drivers 57 through 73 which buffer the input signals and drive the cable going to the computer I/O interface board.

3.5 CHANNEL LOGIC

The channel logic (Figure 3.2) contains all logic needed to operate one specific channel. This logic uses the signals generated

by the computer I/O interface and the common logic group as previously discussed. It also generates the $\overline{LO}$, $\overline{RSXX}$ and data input bits 0-7 and 12-14.

The channel logic group must handle the following tasks: (1) Four mode flip-flops loaded by the $\overline{MS}$ and $\overline{CSXX}$ signals determine the "mode" or status of each channel. (2) All the user input and output signals are buffered with cable line receivers and cable line drivers. (3) The input data is loaded into an input buffer register with serial-to-parallel, bit-conversion logic. (4) The output data is transmitted with parallel-to-serial, bit-conversion logic. (5) When any operation is completed (character output, character input, or a change in ring indicator or carrier detect), the appropriate status flags are set. (6) The status flags are cleared by the $\overline{FC}$ and $\overline{CSXX}$ signals after servicing. (7) The $\overline{LO}$ and $\overline{RSXX}$ signals are activated when any flag gets set and no channels of higher priority are requesting service. (8) When a channel activates the $\overline{LO}$ and $\overline{RSXX}$ signals, data bits 0-7 and 12-14 (signals $\overline{0I}$-$\overline{7I}$, $\overline{12I}$-$\overline{14I}$) are loaded onto the data input bus. Refer to Figure 3.2 for the following detailed circuits description.

The channel "mode" is stored in flip flops 9-12 (SN7475). When the $\overline{CSXX}$ and $\overline{MS}$ signals are true, gate 3 goes high and loads $\overline{T2}$ (by inverter 7) into flip-flop 12, $\overline{T3}$ (by inverter 6) into flip-flop 11 and $\overline{T4}$ (by inverter 5) into flip-flop 10. Flip-flop 9 is

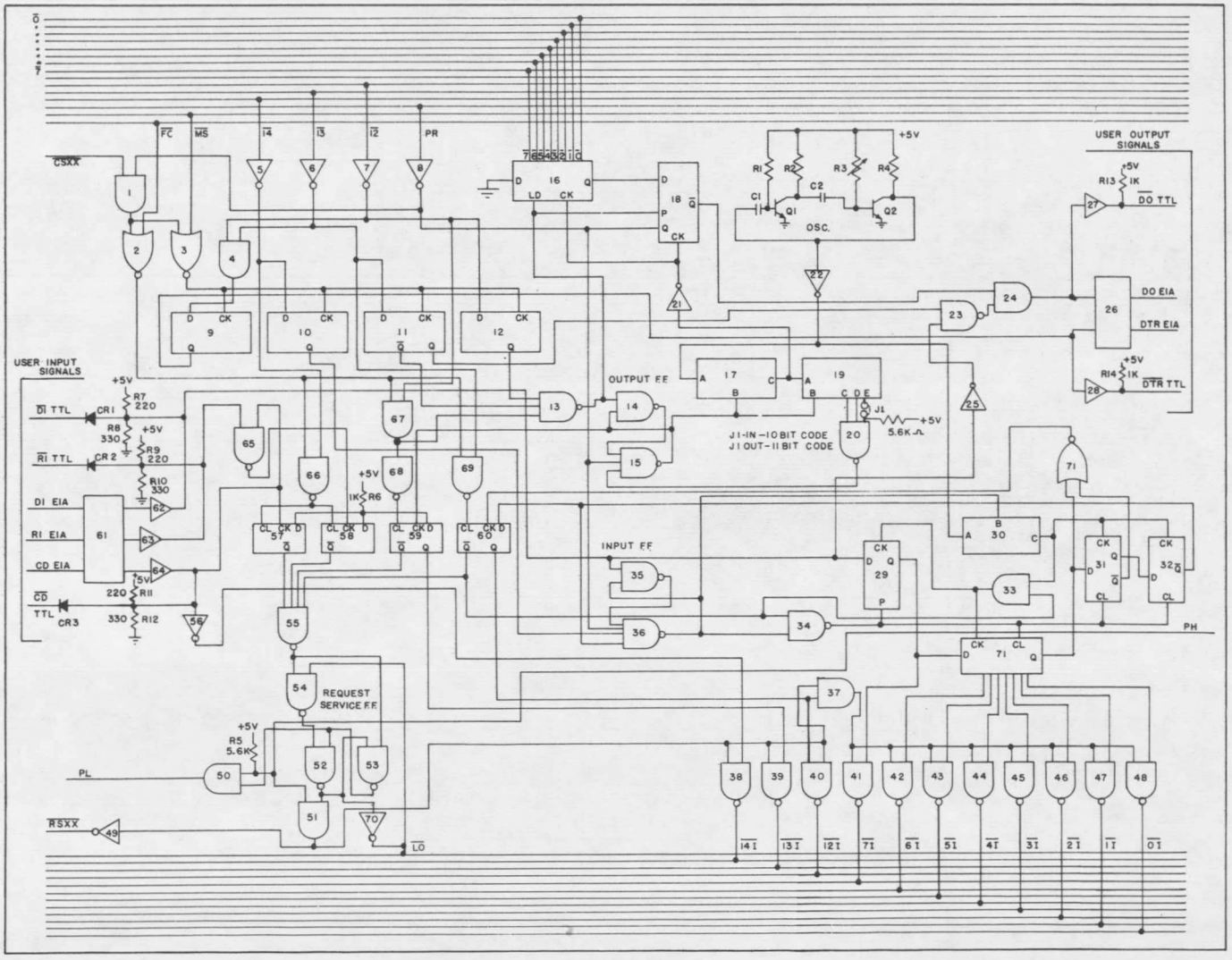


FIGURE 3.2 CHANNEL LOGIC

set false (Q output ground true) if $\overline{I3}$ and $\overline{I2}$ are both true (by gate 4), otherwise the output is set true. Flip-flop 12 enables or disables output, flip-flop 11 enables and disables echo, flip-flop 9 enables and the user signal "data terminal ready", and flip-flop 10 enables and disables input.

The mode flip-flops are also loaded when PR and $\overline{MS}$ are true. This loads the initial starting mode as discussed in Section 3.4 when power is first turned on.

User input data is entered on either the DI EIA or $\overline{DI}$ TTL line depending on signal levels. IC61 (SN75154) converts EIA levels to TTL and buffers 62-64 wire "or" tie the EIA inputs to the TTL inputs. Resistors R7-R12 are cable line impedance matching and bias networks. Diodes CR1 to CR3 provide input gate isolation by preventing input voltages greater than 5 volts on the gate side of the diodes.

The input data is then sent to gate 35, D on flip-flop 29 (all flip-flops are SN7474 except 9, 10, 11 and 12) and gate 23 through inverter 25. If echo is enabled the data also continues through gate 24 and out buffer 27 as $\overline{DO}$ TTL and out IC26 (SN75150) as DO EIA. For this case data in has been "echoed" back out on the data output line.

If input is enabled the input FF sets with the start pulse from the data input word. Initially the 8 bit shift register (SN74164) 71 is cleared along with flip-flops 31 and 32. Flip-flop 29, however,

has been preset to a one. The output of gate 36 goes ground true enabling the divider IC30 (SN7492). This IC divides (by 12) the oscillator output from inverter 22 which is entered into A if B is ground true. The output (C) is used to shift in the data bits through flip-flop 29 and shift register 71.

When the one originally set in flip-flop 29, and the start pulse reach flip-flop 31 and the last stage of IC71 respectively, gate 72 goes true setting the "character received" flag flip-flop 60. The $\overline{Q}$ output of flip-flop 60 prevents the character which is presently contained in flip-flop 29 and the lower 7 stages of IC71 from shifting by disabling the clock drive through gate 33. The flip-flops 31 and 32 are clocked once more however, and this shifts the original one of flip-flop 29 into flip-flop 32 which resets the input FF from output $32\overline{Q}$. After the "character received" flip-flop (60) is cleared flip-flops 31, 32 and IC71 are cleared by gate 34. Flip-flop 29 is preset to an initial one state and the logic is prepared for the next character. By clearing the input mode flip-flop 10 the input FF cannot be set and hence no input characters can be entered and input disabled. Also the PR signal, when true, clears the input FF through gate 8 thus resetting the input circuitry initially.

Characters are output by first setting the proper "mode" to "output on" and "echo off". Then when $\overline{CSXX}$ and $\overline{FC}$ are true the output of gate 2 goes high. Now if signal $\overline{T2}$ is at a one, gate 67 goes

high pulsing gate 13 low setting the output FF, loading output signals $\overline{0}$ to $\overline{7}$ into the data output register 16 (IC SN74165) and setting the start pulse into flip-flop 18. The divide by 12 circuit 17 (SN7492) is now enabled and clocks the character out through gates 24, 26 and 27. Counter 19 (also SN7492) resets the output FF through gate 20 after counting either 10 or 11 clock pulses thus terminating the parallel-to-serial conversion. A jumper option (J1) determines whether a 10 or 11 bit code is generated. As the output FF clears the "output complete" flag flip-flop 59 is set indicating the character has been transmitted. The output FF and flip-flop 18 are cleared when power is initially turned on by PR through gate 8, also. This resets the output logic to the initial starting condition.

When either the ring indicator (RI) or carrier detect (CD) change state, one of the flag flip-flops 57 or 58 sets through gates 64 or 65.

When any of the flip-flops 57-69 are set, channel information is ready to be entered into the computer. The information can only be entered if no other channels are presently loading their information. When any flip-flop 57-60 gets set, gate 55 goes true. Then if $\overline{LO}$ is <u>not</u> true (no other channel requesting service) the request service FF is set through gate 54. This causes $\overline{LO}$ to go true preventing any other channels from requesting service. If no other channel of higher

priority also wants service, PH is true and activates $\overline{RSXX}$ through gate 51 and inverter 49. Gate 51 also loads the status of the "character received flip-flop", "output complete FF" and the status of carrier detect through gates 38, 39 and 40 onto the data input bus bits $\overline{14I}$, $\overline{13I}$ and $\overline{12I}$. If the "character received" flip-flop is set, the character is loaded onto bus bits $\overline{0I}$-$\overline{7I}$ through gates 27 and 41-48.

When the request service FF is set PL goes false through gate 50. PL becomes PH of the next lower channel and prevents any channels of lower priority from requesting service until the present channel has been serviced and all flag flip-flops (57-60) cleared.

Flip-flops 57-60 are cleared by the signals $\overline{CSXX}$, $\overline{FC}$, and the appropriate output bit ($\overline{12}$, $\overline{13}$ or $\overline{14}$). $\overline{14}$ clears "change in RI or CD status flip-flops" (57-58), $\overline{13}$ clears the "character received flip-flop" (60) and $\overline{12}$ clears the "output complete FF" (59). Once these flip-flops are cleared, gate 55 goes false clearing the request service flip-flop and enabling the PL signal through gate 50.

3.6 TESTING OF THE MULTIPLEXOR

The initial testing of the logic was done by breadboarding small subgroups. These groups were then tested by entering input data with switches and monitoring outputs on an oscilloscope. The parallel-to-serial and serial-to-parallel converters were tested in

this way. Various oscillators were built up and tested also for use with these bit converters. Temperature tests were run to check the stability of the oscillators.

Next the priority logic was breadboarded and simulation interrupts were entered with switches. Again, the outputs were monitored with an oscilloscope.

The last circuits to be breadboarded and tested were the power-fail FF set ($\overline{PFS}$) generator and the flag clear and mode set common logic circuits.

The entire MUX was never totally tested until the prototype was constructed. This was built with printed circuit cards and interconnected with a hand wired backplane. The logic was partitioned quite differently on the original prototype, however. Those functions shown in Figure 3.2 were actually on three different printed circuit cards. One card contained the serial-to-parallel and parallel-to-serial bit-converters and the oscillator. Another board contained the flags, mode, interface and bussing logic. Still a third board contained all the priority logic common to 8 channels.

With the logic partitioned in this manner only eight channels could be implemented in one card file. Provisions were made to connect to another card file, however, to implement the other 8 channels.

The 8-channel MUX prototype was then connected to the computer and two simple programs were written to test the basic operation of

the MUX. After getting the initial bugs out of the system (cable, backplane and printed circuit wiring errors) the system was ready to be tested with time-share Basic. These drivers were written according to the specifications of Chapter 2 and several months were required to finish debugging the system.

Next a second 8-channel MUX was constructed having all the corrections of the first MUX and included a wire-wrap backplane. Later, the 16-channel MUX was built and included the combination of all channel logic on one printed circuit card. Included with this was some logic reduction and the final results are shown in Figure 3.2. The common logic could also be simplified and is shown in Figure 3.1.

3.7 SUMMARY

This concludes the description of the operation of the MUX. First, signal names, levels, currents and transition times were defined. Next the operation of the common and channel logic was described and finally testing proceedures were presented.

CHAPTER IV

THE MECHANICAL DESIGN OF THE MUX

4.1 INTRODUCTION

This chapter discusses the physical and mechanical construction of the MUX. Various card cages, printed circuit card sizes, power supplies, interconnecting cables and cable connectors were tried before the final configuration was decided upon. This chapter shows the evolution, problems encountered and considerations made during construction.

4.2 CARD CAGES

The first problem encountered was trying to find a suitable housing or cabinet for the logic. Printed circuit cards were decided upon to interconnect the logic and hence a cabinet capable of housing many (19 in the final configuration) cards was desired.

Cabinets and card cages come in standard heights and widths for mounting in relay racks and other enclosures. The most common width is 19" which was acceptable for this application. With a width of 19" an inside width of 17" is obtained. Printed circuit cards can be mounted 1/2" apart and hence approximately 32 cards can be installed in one card cage. This is more than enough space to house the MUX logic which requires only 16 channel logic boards, two common logic

boards and one plug board (for MUX connection to the computer).

The first card cage selected was a commercial version. The prototype was constructed using this card cage but extensive modification was required because the card guides were quite short and the maximum card width (without modification) was 4 1/2". Because of the unsatisfactory results obtained with the commercial unit, a custom card cage was designed and constructed using 1/8" sheet aluminum. This resulted in a much more satisfactory cabinet and is shown in Figure 4.1.

4.3 PRINTED CIRCUIT CARDS

At the present there are two common ways of interconnecting integrated circuits. The first and most reliable is with the use of printed circuit cards. The second is with a wire-wrap technique. The wire-wrap technique is somewhat cheaper for low volume work and is easiest to correct if an interconnection error is made. However, it is slower and more prone to errors once a printed circuit card is correctly laid out. Other problems include a large board spacing width and special tools and IC sockets required for construction.

Therefore, printed circuit cards were used for mounting the logic. Double sided circuits on 1/8" epoxy board were used with plate-through holes. All boards are interconnected with 44 pin edge connectors that plug onto gold plated fingers for high wear resistance.

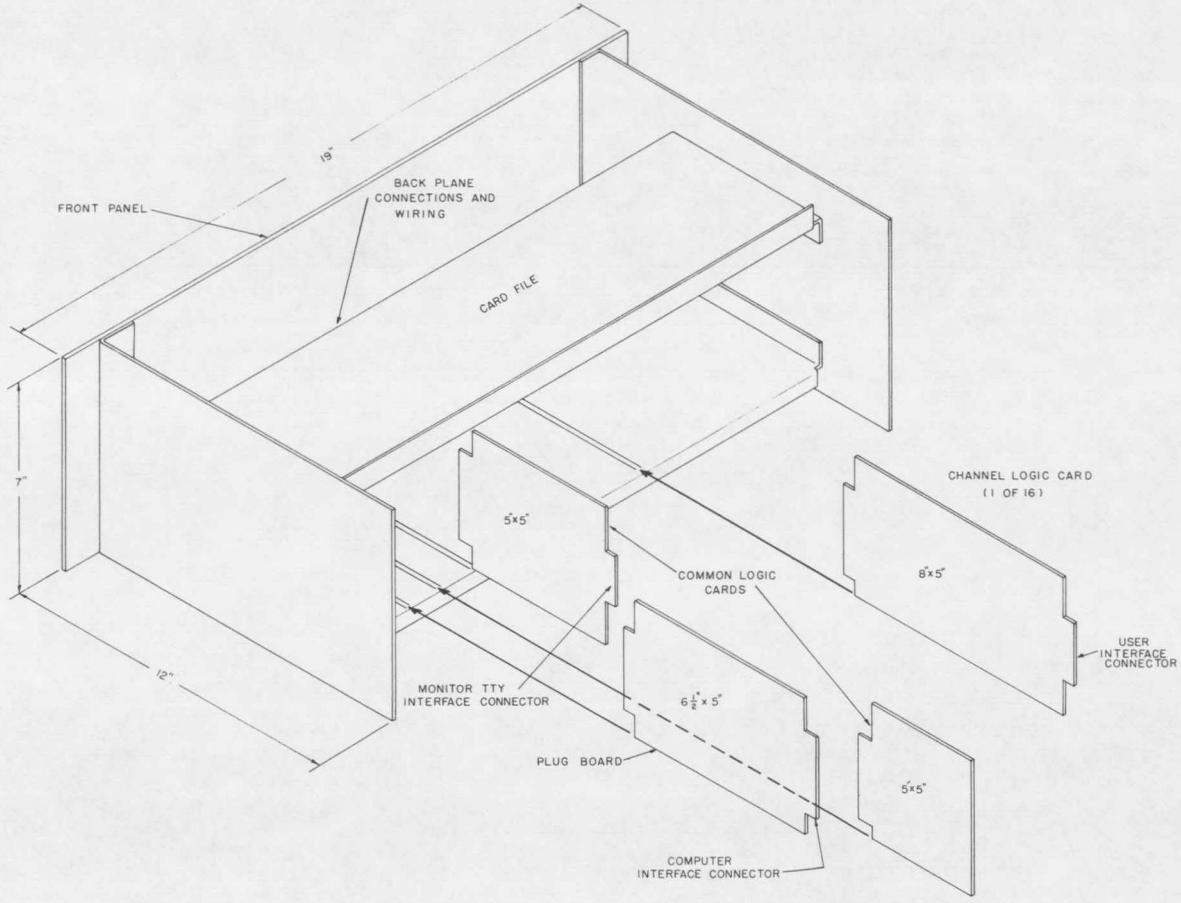


FIGURE 4.1 MUX PHYSICAL LAYOUT

These boards were laid out on mylar with black tape, twice actual size. These were then reduced on negatives to actual size for use in fabrication of the printed circuit cards. Figure 4.1 illustrates the printed circuit card sizes and locations.

4.4 CABLES, CONNECTORS AND BACKPLANE WIRING

Just as there are various techniques for interconnecting circuits on one board, there are also various techniques for interconnecting printed circuit cards and circuit modules.

The most convenient way to connect to a printed circuit card is with the use of an edge connector. Therefore, the printed circuit card I/O interface in the computer is connected to the plug card in the MUX through a cable connected to two edge connectors. The cable is a 68 conductor, twisted pair, low voltage cable twelve feet long. Twisted pairs were selected with one line carrying the signal and the other ground. This minimizes crosstalk and noise problems and provides enough conductors for the return ground currents.

The 16 users and the monitor teletype are similarly interfaced using edge connectors but because less signals are involved, shorter edge connectors were selected for this application. Also, because of the relatively slow signal switching speeds twisted pair cable is not required.

There are three common ways to interconnect back plane connectors.

The first is wire soldered to pin tabs, the second is wire wrap and the third is with a mother board printed circuit card. The prototype MUX was interconnected using the first technique of soldering wires between the appropriate connector tabs. The second MUX was built using the wire wrap technique but was not used again because it was slow and required special tools. The next three 16-channel MUX units were built using the first technique of soldered wire, but the bussing had been simplified enough to allow the use of bare bus wire insulated with teflon tubing.

4.5 POWER SUPPLY

The MUX requires three regulated power supply voltages of +5V at 4 amps, (for the logic) +12V at 1 amp and -12V at 1 amp (for the EIA interface IC's). Ripple and regulation must be less than 5% on these three power supply voltages for error free operation. A commercial power supply module meeting these voltage, current and ripple specifications was selected for this application.

This module was mounted on a 3 1/2" x 19" front panel and an on-off switch, indicator lamp and AC protection fuse installed. The MUX was then connected to the power supply with four 16 gage conductors connected to a terminal strip mounted on the power supply.

The entire MUX and power supply unit together occupy a total of 10 1/2" of front panel space in a standard 19" cabinet and re-

quires a maximum depth of 12".

4.6 SUMMARY

This concludes the mechanical description of the MUX. First, the cabinet and printed circuit card assemblies were described. Then, printed circuit and module interconnection systems were discussed, and finally power supply selection and mounting was presented.

CHAPTER V

FINAL EVALUATION, PERFORMANCE AND CONCLUSION

5.1 INTRODUCTION

The MUX has now been thoroughly tested after building and operating two prototypes and three production models. The system operation and performance can now be evaluated by discussing the following items: ease of construction, testing and trouble-shooting, reliability, relative cost and performance (as related to the specifications and goals set forth in Chapters 1 and 2).

5.2 CONSTRUCTION, TESTING AND TROUBLE-SHOOTING

The ease with which a product can be constructed and tested directly affects the cost of production and maintenance. The present MUX is constructed with 18 printed circuit cards which are quite easy and fast to build. The channel logic board, which is the most complex, contains only 26 integrated circuits, 8 capacitors, 3 diodes and 18 resistors. It can be built in less than two hours and computer checked in 5 minutes. The other two common logic boards are even faster and easier to build and check.

Since all the electronics are contained on printed circuit cards, trouble-shooting a MUX in the field consists of board swapping until the problem is solved. Also a computer program diagnostic can be

used which tests the operations of the computer interface board, common logic, and each individual channel board to help determine which printed circuit card is malfunctioning.

The two time consuming construction sections of the MUX at present are the hand wiring and checkout of the backplane and interconnecting cable to the computer. Both are slow and prone to mistakes, though once completed and checked, are quite reliable. Below is a list of typical times to construct and test various parts of the MUX:

Channel Board Construction and Test (one channel)	2 hours
Common Logic Board Construction and Test	4 hours
Card Cage Mechanical Assembly	2 hours
Card Cage Backplane Wiring	16 hours
Computer Interface Cable	8 hours
Total Construction Time	62 hours
Final Testing, Burn-in and Check-out	8 hours
Total	70 hours

5.3. COST

The cost of the 16-channel MUX is about half the cost of the two interfacing methods described in chapter one. This includes the cost of the computer interface board and power supply required by the MUX. Also, only as many channels as desired, need be implemented

in the MUX. If an eight channel system is needed only eight channel logic boards need to be installed thus cutting the cost from the 16-channel system substantially. By using this MUX, from 5 to 16 channels can be implemented more inexpensively than the alternate interfacing solutions discussed in chapter one. Additionally only one I/O slot in the computer is used, thus leaving the rest free for other interfacing tasks.

5.4 RELIABILITY AND PERFORMANCE

The MUX has proven to be a very reliable piece of equipment. There has not been one failure in a MUX once it has been placed in operation and the original prototype ran for one year. It was then replaced by a 16-channel MUX, and again there has been no down-time for this MUX either, during its first one year period of operation. With only two units in operation, however, true reliability is not very accurately determined. The reliability of the MUX is basically determined by the reliability of the integrated circuits. It has been the experience of the author that typically not more than one IC in 500 will fail in a 12 month period after initial burn-in.

Between 1% and 2% failure of IC's during initial testing and burn-in (first week) has been encountered. This is typical of semiconductor performance throughout the industry at the present, but with the use of a sufficient burn-in period and computer check-out,

most problems can be eliminated before putting the MUX in service.

No long term reliability figures have been determined, either, because of the short (2 year) operation time. The long term reliability of the MUX, however, will be determined by the life of the IC's.

5.5 CONCLUSION

The MUX has met every requirement and specification set forth in chapters one and two. It is a very easy device to program, very reliable and if problems do occur it is easy to service and repair.

The project was highly successful and because it was carried from original design to final testing and manufacturing, a great deal about various aspects of actual product engineering was learned.