NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

FIBER-OPTIC IMPLEMENTATION OF
MIL STD-1553
A SERIAL BUS PROTOCOL

by

Robert Stanley Wester

March 1987

Advisor: J.P. Powers

Approved for public release; distribution is unlimited.

T233786

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION	1b RESTRICTIVE MARKINGS
UNCLASSIFIED	
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	

4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION	6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION
Naval Postgraduate School	62	Naval Postgraduate School

6c ADDRESS (City, State, and ZIP Code)	7b ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943-5000	Monterey, CA 93943-5000

8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
FIBER-OPTIC IMPLEMENTATION OF MIL STD-1553: A SERIAL BUS PROTOCOL

12 PERSONAL AUTHOR(S)
WESTER, Robert Stanley

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1987 March	108

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	MIL STD-1553; Serial Data Bus Protocol; Fiber-Optics; Micro Computer System Design; Loop Interconnect Topology

19 ABSTRACT (Continue on reverse if necessary and identify by block number)
This thesis explores the design and implementation of a Fiber Optic Link for use in MIL STD-1553 environments. The discussion includes specific hardware and software designs to demonstrate a basic fiber-optic implementation of th standard. These designs are presented in sufficient detail to allow reconstruction with a minimum of effort. Results such as Built-In-Test performance and Maximum Data Rates are included. The design and associated fiber optic link provide a good prototype to be used for further research involving fiber optic solutions for serial data link Local Area Networks.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Prof J. F. Powers	(408)646-2679	62Po

DD FORM 1473, 84 MAR
83 APR edition may be used until exhausted
All other editions are obsolete

Fiber-Optic Implementation of MIL STD-1553:
A Serial Bus Protocol

by

Robert Stanley Wester
Captain, United States Air Force
B.A.E.E., Park College, MO., 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

ABSTRACT

This thesis explores the design and implementation of a fiber optic link for use in MIL STD-1553 environments. The discussion includes specific hardware and software designs to demonstrate a basic fiber-optic implementation of the standard. These designs are presented in sufficient detail to allow reconstruction with a minimum of effort. Results such as Built-In-Test performance and maximum data rates are included. The design and associated fiber optic link provide a good prototype to be used for further research involving fiber optic solutions for serial data link Local Area Networks.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The subject of this thesis is the application of Fiber-Optics communications to a serial bus protocol known as Military Standard 1553 (Mil Std-1553)[Ref. 1]. Mil Std-1553 is a prime candidate for such an evolutionary step. It is widely used in weapon systems ranging from the F-16 fighter aircraft to the DIVAD gun system [Ref. 2]. The protocol has been around since 1973 and has currently been used on such weapon systems as the B-1B bomber and the Trident II submarine. Some unique benefits of the standard will be discussed in the next chapter. A twisted shielded copper wire pair has been used to date for Mil Std-1553. It has been fabricated in such a way as to improve its noise immunity and has been coupled at each end via a matched transformer. Radio Frequency Interference and Electromagnetic Interference (RFI and EMI) have been issues, especially the Electromagnetic Pulse (EMP). The data rate has been limited to one megabit per second to increase data integrity.

Fiber optics can be successfully used with Mil Std-1553 to virtually eliminate RFI and EMI problems and the data rate would be limited to the processing devices and not the communications channel itself. Thus the data rate could extend to fifty megabits per second or higher. Many other benefits of fiber optics will be discussed later.

The thrust of this effort is to demonstrate the implementation of Mil Std-1553 using fiber optics. This implementation is relatively basic but could be used as a model for further study. A fair amount of flexibility

has been included along with an intelligent front end (8748 microcomputer) to allow for any of a number of general applications. The design as it stands could be used on such applications as a Local Area Network (LAN) between engineering workstations, a plant control communications link, or a digital data acquisition system on board an airborne test bed. The system design will be discussed in sufficient detail to allow reconstruction with minimal effort.

II. MIL STD-1553

A. CHARACTERISTICS OF MIL STD-1553

The protocol used in Mil Std-1553 is based on a Manchester II serially encoded BiPhase data stream. This encoding scheme was devised to allow for regeneration of the data clock directly from the data stream. The clock can be derived as the first harmonic of the serial data. Each encoded bit is formed in such a way as to contain a positive or negative going transition at the mid-bit position (see Figure 1). By detecting these transitions, the clock can be regenerated without a-priori knowledge of when the data stream started.

B. COMMAND AND DATA WORD FORMATS

As per Mil Std-1553, there are three types of word formats used. The Command and Status word formats are shown in Figure 2 [Ref 2]. The Data word format is also shown in Figure 2. As can be seen, each type of word has a unique type of synchronization pattern. This is referred to as Improper Manchester Sync. The sync pattern consists of a three bit period in which one and one-half bit periods are in one state and one and one-half bit periods are in the complementary state. This allows for easy discrimination between the "normal" serial data and the synchronization pattern. A free running clock is started at the beginning transition (whether positive or negative), and at each clock period the data stream is sampled. If the sampled stream indicates a highor low state for more than one bit period, an "improper Manchester" state is generated which is used to signal the beginning of a frame of data.
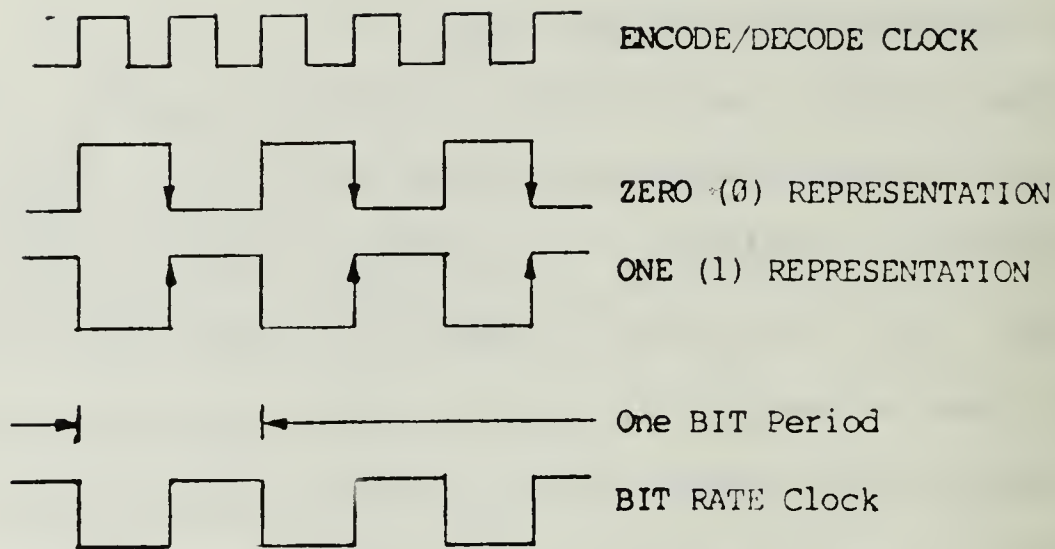
Figure 1.   Manchester II BiPhase Encoding Scheme

Bit Periods	1 2 3	4 5 6 7 8	9	10 11 12 13 14	15 16 17 18 19	20
COMMAND WORD	SYNC	TERM ADDR	T/R	SubAddr	Word Count	P
DATA WORD	SYNC	←———————— 16-Bit DATA ————————→				P

Figure 2. Mil Std-1553   Command and Data Word [Ref. 1]

12

The Terminal address field represents the "from" terminal identification and the Sub-address field represents the "to" terminal identification. The "T/R" bit is used to pass a "token" in a token passing ring network. The Word Count field indicates the number of data words that are going to be sent with each Command word. The parity bit field is used to represent the "Odd" parity of the previous sixteen bits in the word. As can be seen in Figure 2, there can be 2^5 or 32 terminals addressed as well as 32 data words in each data block. Mil Std-1553 allows for several types of configurations. Ring networks can be established using the "T/R" token and Star networks can be implemented using the addressing fields.

Each Data word contains sixteen bits of binary data and is framed by the Data Sync and is error checked by the "Odd" parity bit in conjunction with the Manchester encoding scheme. The data is encoded and decoded in a "Non-Return to Zero" (NRZ) format. It can represent anything from ASCII characters to signed magnitude numbers.

C. MIL STD-1553 COMMUNICATIONS CHANNEL

As mentioned earlier, a twisted shielded coaxial cable has been used to date for the communications channel. The channel is excited using a transformer-coupled differential line driver/receiver pair as seen in Figure 3a. The transformer is a special type of "capacitively" excited device to provide for increased noise reduction and is "tuned" to the 1 MHz clock frequency. The driver/receiver pair excites the transformer with a 15 volt differential signal. Mil Std-1553 dictates very precise

13
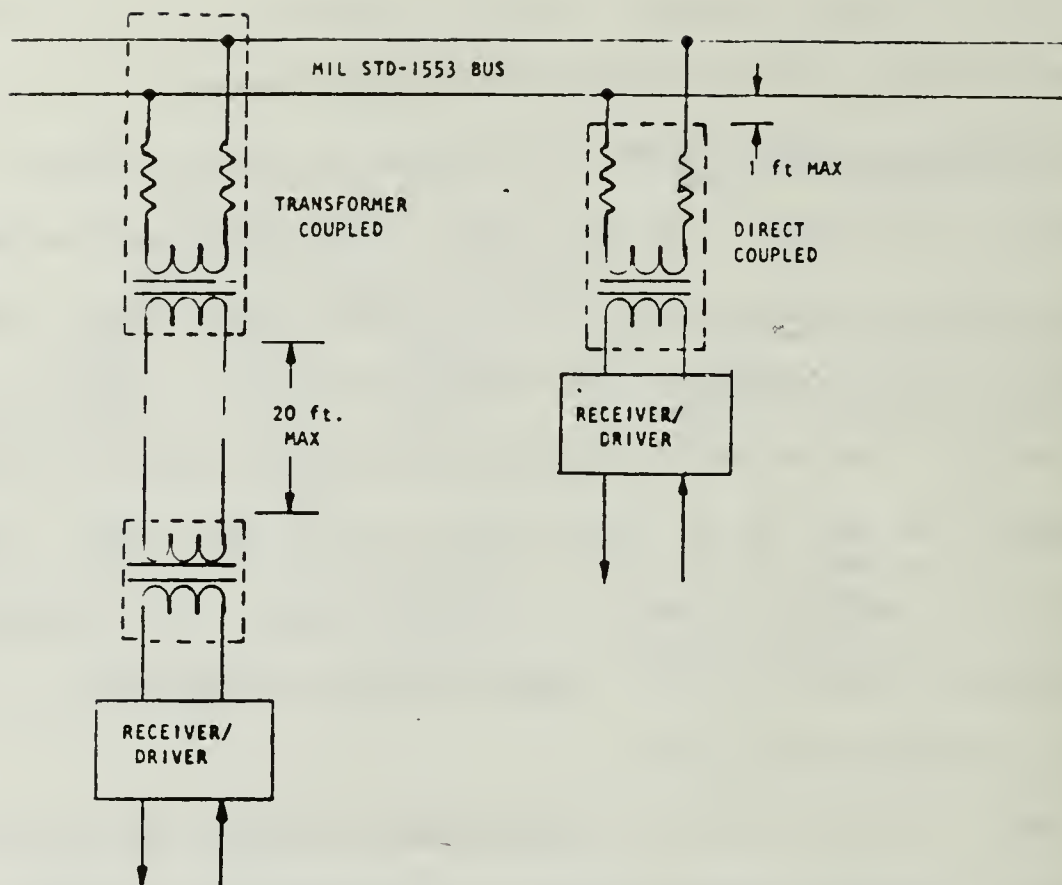
Figure 3a.   Mil Std-1553 Transformer Coupled Coaxial Cable for Serial

Bus [Ref.2]
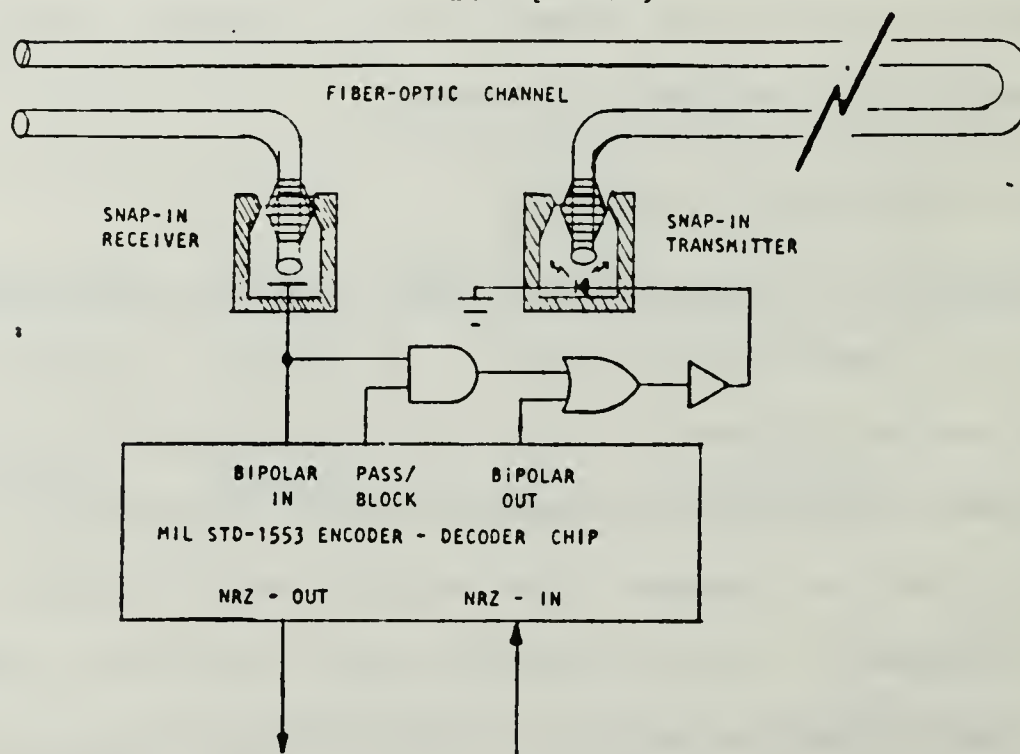
Figure 3b.   Fiber Optic Implementation of Mil Std-1553

rise and fall times for this excitation in an attempt again to reduce the noise interference. The standard also specifies a particular "BNC" connector to allowfor the twisted shielded pair alignment. A very attractive alternative to this scheme would be, of course, fiber optics. Also included in Figure 3b is an "equivalent" fiber optic scheme which is greatly simplified and eliminates noise interference by optically isolating each communicating device. In addition, the bandwidth can exceed 1 MHz by factors of tens.

D. A TYPICAL MIL STD-1553 ARCHITECTURE (F-16)

One particular application of the standard is the F-16 Digital Avionics System [Ref. 6]. A simplified block is shown in Figure 4. The Fire Control Navigation Computer (FCNP) acts as the Bus Controller in this star-type network. All intersystem communications are managed by the FCNP. A typical scenario might be for the Heads Up Display (HUD) to query the Inertial Navigation System (INS) for the present position. The HUD would request this data block from the FCNP. The FCNP would honor this request by directing the INS to ship this data block to the HUD. The INS would then ship this block directly to the HUD. In this scheme, each subsystem acknowledges receipt of a frame with a status word indicating proper/improper receipt of the frame, including the proper number of words received.
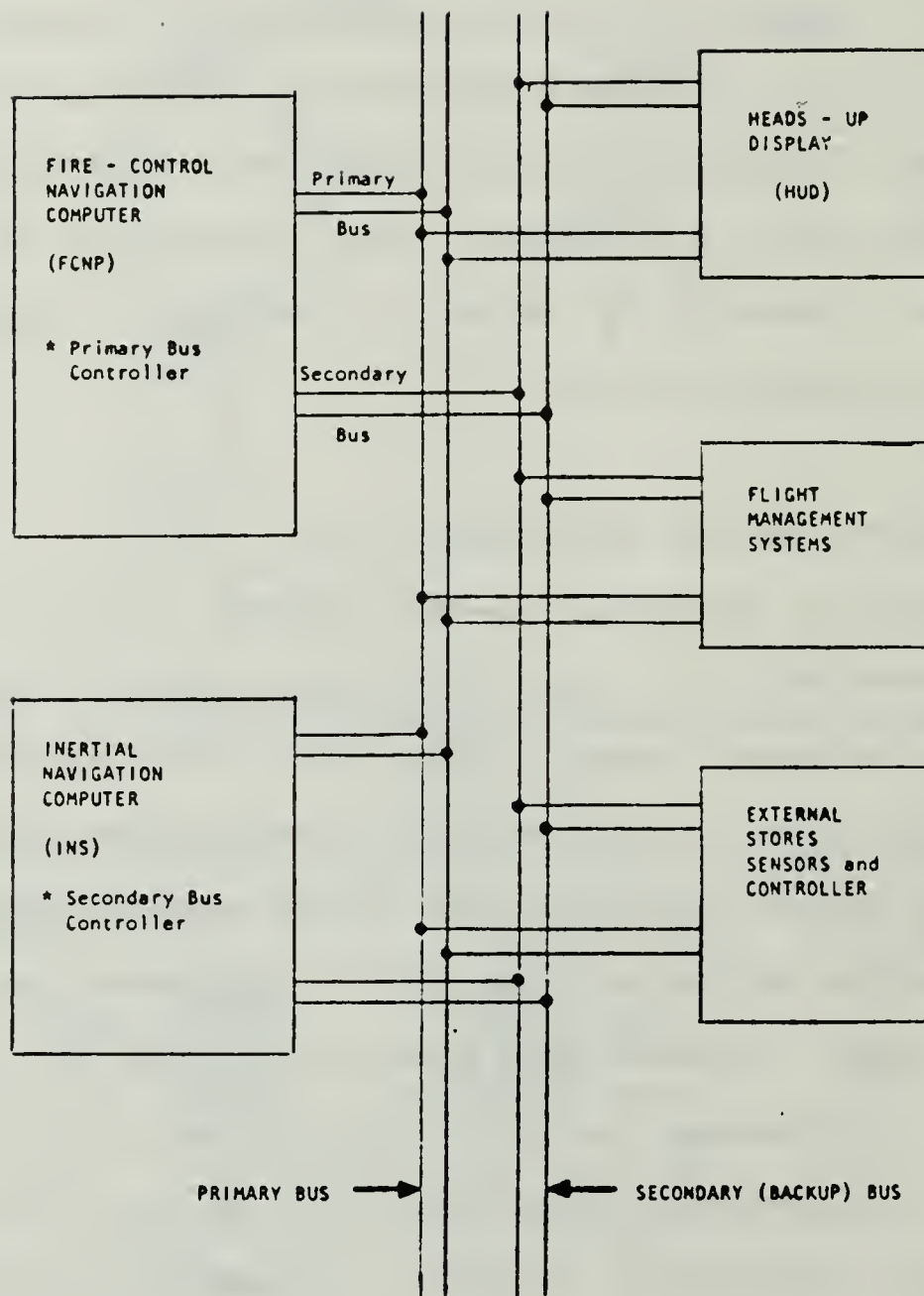
Figure 4. A Typical Mil Std-1553 Application -

The F-16 Aircraft Avionics System [Ref. 6]

E. THE HD-15530 MANCHESTER ENCODER/DECODER CHIP

In the late 1970s, Mil Std-1553 was implemented using approximately 30 small and medium scale integrated circuit chips. With the advent of Large Scale Integrated circuit (LSI) technology, the standard is increasingly being compacted into one or two LSI chips. With hundreds of thousands of transistors residing on a silicon wafer, the complexity of the standard can easily reside on one chip. In the early 1980s, several LSI implementations of the standard were being marketed. The HD15530 chip from Harris Semiconductor is one such implementation. It was incorporated in the design for this effort due to its availability. There are, in fact, many superior chips currently available [Ref. 1].

The HD15330 chip can functionally be broken down into two independent sections [Ref. 3]. The Encoder and the Decoder can operate completely independent of each other. They can be clocked at differing rates and enabled/disabled separately. The Encoder is shown in Figure 5. It accepts Serial NRZ data at the "Send Clock" rate and generates either bipolar one/zero Manchester data or unipolar data. It can generate either a Command/Status Sync or a Data Sync depending on the signal level at the "Command/Data" pin when the encoder is enabled. It requires a "Send Clock" at twice the desired data rate.

The "Decoder Clock" requires a clock at twelve times the data rate. Its internal circuitry contains a divide-by-six counter to conveniently derive the "Send Clock". The Decoder, as shown in Figure 6, accepts either unipolar or bipolar Manchester data with associated Sync and produces serial NRZ data with a Command or Data Sync indication. It also

17
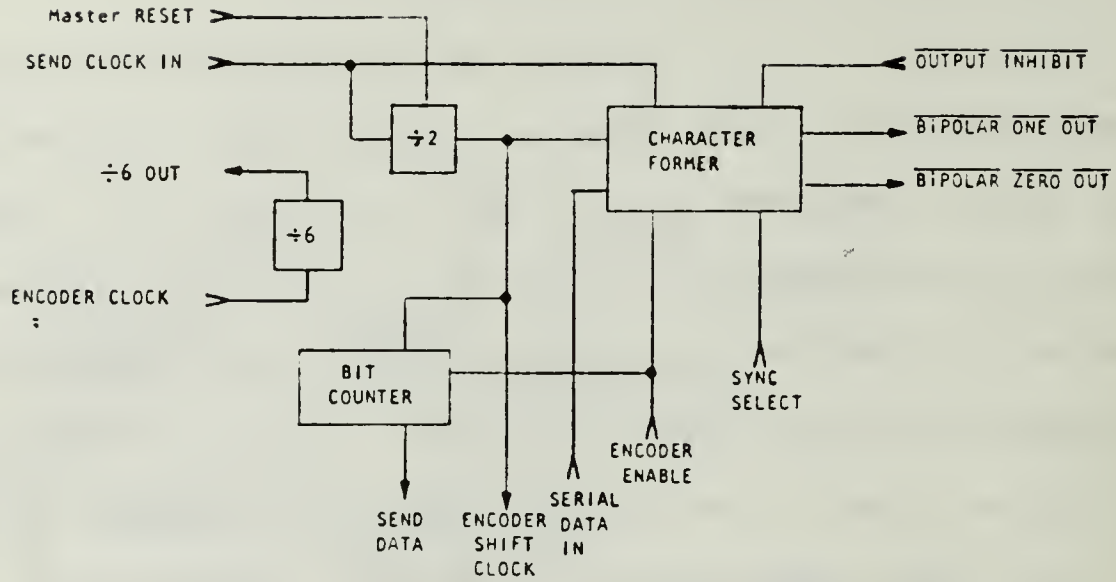
Figure 5. Harris HD-15530 Encoder [Ref. 3]
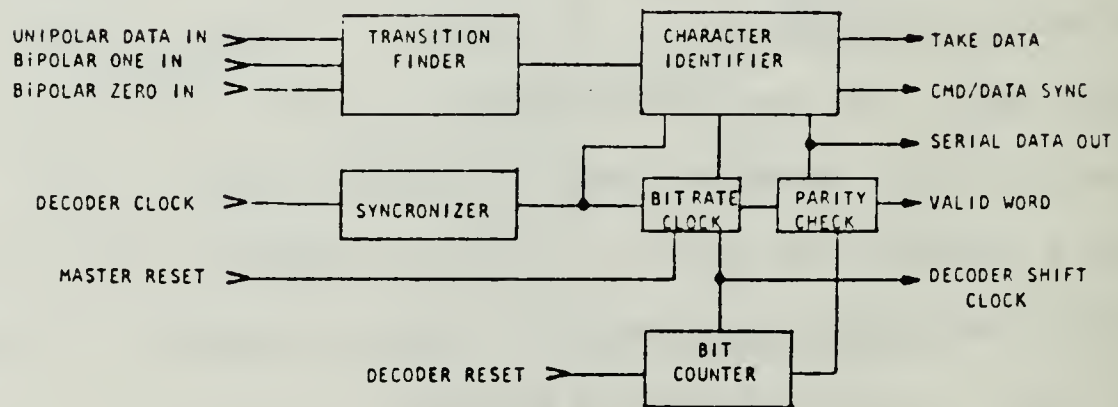
Figure 6. Harris HD-15530 Decoder [Ref. 3]

provides for proper Mil Std-1553 decoding and detection via an internal bit rate clock, synchronizer, and parity checker. If the received word is proper in all respects, a "Valid Word" signal is generated. The serial NRZ data is clocked out at the "Decoder Shift Clock" rate.

The HD15530 chip can operate at a maximum data rate of 1.25 MHz. At this rate, the maximum "Encoder Clock" frequency is 15 MHz. The device is a CMOS chip and is compatible with standard TTL chip families. It uses a standard +5 VDC power source and with internal circuitry requires only one single phase clock. Both the Decoder and Encoder can be reset with "Master Reset".

III. SYSTEM DESIGN SUMMARY

A. SYSTEM OVERVIEW

The intent of this effort was to demonstrate the feasibility of using fiber optics in many Mil Std-1553 architectures. Flexibility was incorporated in the design. For the purpose of initial verification and check-out, a "built in test" capability was incorporated.

Functionally, there are five modes of operation. Two are for testing/demonstration purposes and two are for parallel data input operation. The last is for serial data input operation.

To accomplish this generality and provide for ease of interfacing, an intelligent "front end" was designed. Availability of parts and development tools was a primary criterion.

B. INTELLIGENT FRONT END CONTROL (μC8748)

In order to keep the chip count and development time to a minimum, a programmable controller was selected. The Intel 8748 microcomputer chip was readily available and provides most of the requisite facilities. It is not the optimum solution (see Conclusions and Recommendations) but allowed completion of this effort with reasonable resources.

A block diagram of the μC8748 is given in Figure 7 [Ref. 4]. As can be seen, it contains an Arithmetic Logic Unit (ALU), up to sixteen working registers in addition to an accumulator, various flags (C, AC, Z, F0, and F1), 64 bytes of Random Access Memory (RAM), and 1 kilobyte of Erasable Read Only Memory (EPROM) for Program Storage. It can handle one external interrupt and contains an internal programmable timer/counter

20

which can also interrupt. It contains internal circuitry to facilitate single step operation as an aid to program debugging.
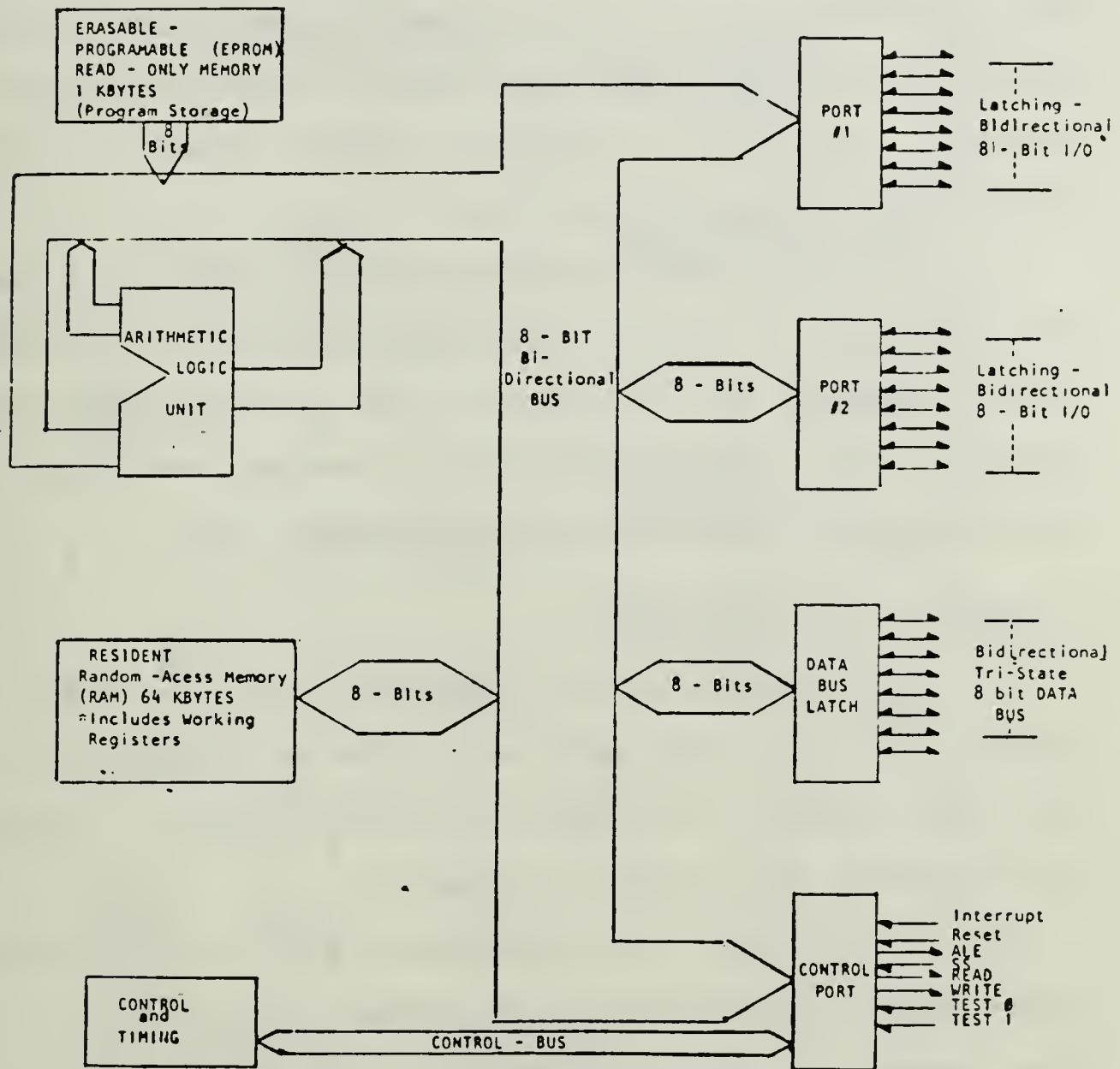
Figure 7.   Block Diagram of the ⨍C8748 (Ref. 4)

There are two latching 8-bit Bidirectional Ports and one 8-bit Bidirectional Data Bus. Two Testable input pins are available as event sensors. The ⨍C8748 has a power-down mode which can be utilized in

isolation type-scenarios. The chip can operate up to 6 MHz per memory cycle or be single stepped as mentioned above.

The instruction set is relatively flexible and efficient with 70% of the instructions requiring only one processor cycle (2.5 to 5.0 μSec). There are data transfer instructions which can move data internally or externally. Accumulator operations include Arithmetic, Logical, Exchange, and bit manipulation functions.

In general, the μC8748 satisfies all requirements of a general purpose computer. As such, it meets the requisites for this effort. Program development was tedious due to lack of user-friendly software development tools. An assembler, in-circuit emulator, and linker/loader would have reduced development time significantly.

C. SYSTEM BLOCK DIAGRAM SUMMARY

The major components of the system design are the μC8748, the HD15530, control logic, data paths and the fiber optic interface. A high level block diagram of the design can be seen in Figure 8. A detailed circuit schematic can be studied in Appendix A.

The μC8748 provides data flow buffering and control. The HD15530 performs the Mil Std-1553 encoding and decoding. Control logic provides proper strobing and detection functions for the various ports and the HD15530 chip. The data paths provide either 8/16-bit parallel or serial data channels between the input/output interface, the μC8748, and the HD15530 chip. Fiber optic drivers and receivers convert the Mil Std-1553 encoded data to light that couples into the fiber optic snap-in link.
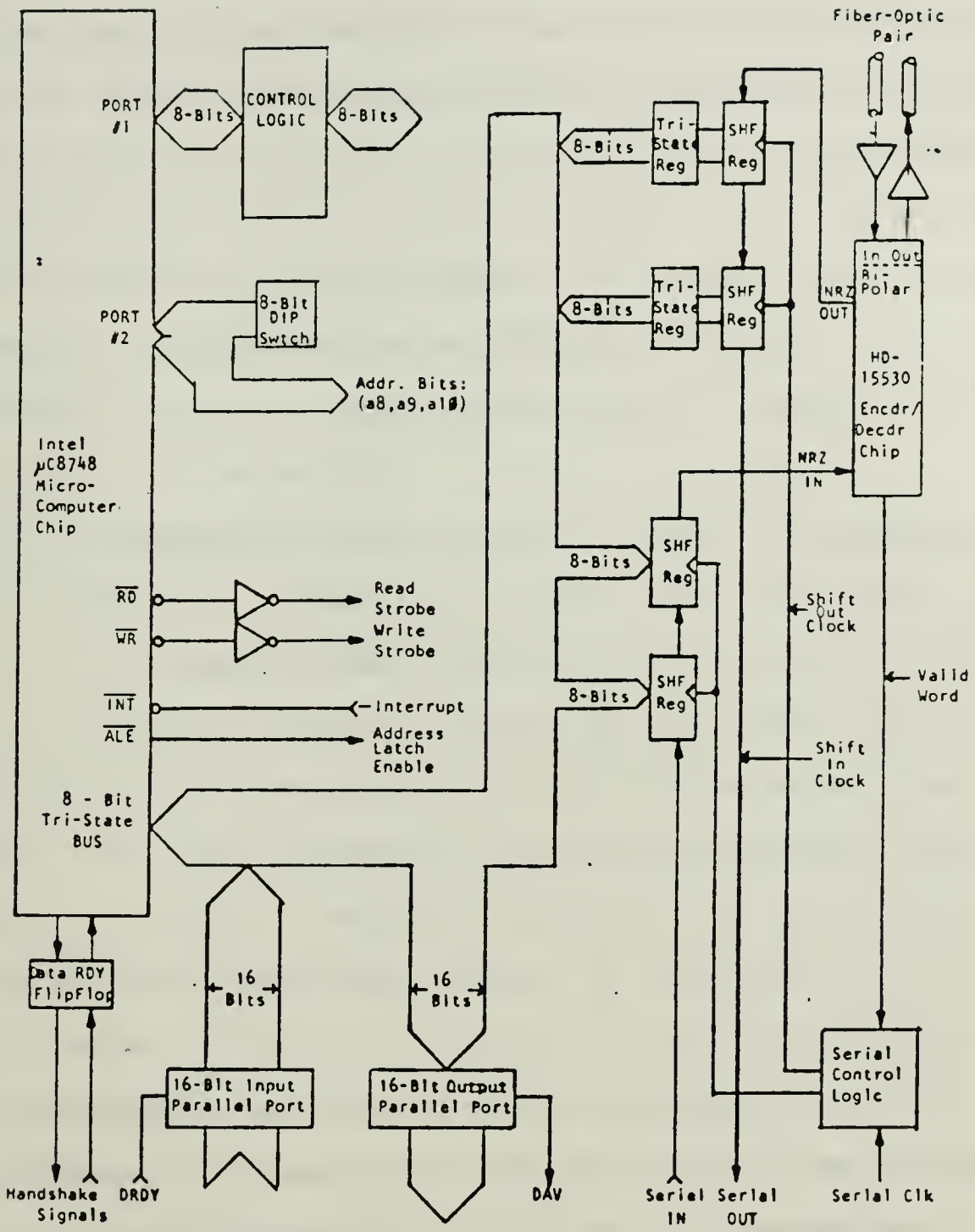
Figure 8.   Block Diagram of Mil Std-1553 Card Design

23

There are five modes of operation as mentioned earlier. These modes of operation are provided by the software internal in the μC8748 EPROM. Each of the modes can be selected via a Dual In-line Package (DIP) switch located on the board. The Mode Selection Switch settings are indicated in Table 1.

TABLE 1

MODE SELECTION DIP SWITCH SETTINGS

S_8	S_7	S_6	S_5	MODE
0	0	0	0	TEST Pattern
0	0	0	1	Parallel Standard
0	0	1	0	Invalid
0	0	1	1	Parallel Bypass
0	1	0	0	Invalid
0	1	0	1	Serial
0	1	1	0	Invalid
0	1	1	1	Echo
1	X	X	0	Built-In-Test (BIT)

Testing and Demonstration can be accomplished using the Test Pattern generation mode on one board and the ECHO mode on a second board. In this configuration, a test pattern of alternating ones and zeros is transmitted across the fiber optic link to a second board. The second board in turn retransmits (echos) the pattern but swaps the addresses in the Command Word to allow receipt by the originating (test pattern generator) board. In this way, using an oscilloscope and the "Single

24

Step and Display" plug-in card (to be discussed later), program debugging and hardware check-out can be performed. Additionally, the drive current potentiometer for the fiber optic driver can be adjusted to the optimum level to ensure accurate reception while maintaining minimum drive current on the driver to increase the life of the fiber optic driver LED.

There are two modes of operation for parallel data transfers. One is the Parallel Standard mode in which the parallel data is buffered internally in the μC8748. Up to 12 hex or 18 decimal 16-bit words can be buffered in this way to allow slower devices to interface to the fiber optic link.

The second parallel mode bypasses the μC8748 by transferring the 16-bit parallel data directly to and from the I/O registers for the HD15530 chip. This is a much faster mode because the delay is related directly to the HD15530 encoding time and not the μC8748 buffering time. The μC8748 controls the transfer by simple handshaking techniques.

The Serial mode allows serial Non-Return-to-Zero (NRZ) data to be clocked in directly to and from the HD15530 chip I/O registers at an independent rate. The μC8748 controls the transfer by sensing when 16 bits of serial data have been clocked in or out and then enabling the appropriate ports.

In either of the parallel modes and in the serial mode, the number of words to be transferred should reside on the least significant half (LSH) of the parallel port prior to enabling the μC8748 for transfer initiation.

25

IV. DETAILED SYSTEM DESIGN

The detailed discussion of the hardware circuitry, software routines, and interfacing circuitry will be presented here. The intent is to explain the peculiarities of the design that are not inherently obvious.

As mentioned in System Design Summary, the system design can be segregated into three areas: hardware circuit design, software routine development, and interface logic. These categories will be expanded upon now.

A. SPECIFIC CIRCUIT DESIGNS

The hardware circuitry used in the design is predominantly Transistor-Transistor Logic (TTL) with the exception of the μC8748, the HD15530, and the fiber optic drivers and receivers. All devices are either TTL or TTL-compatible. Both the μC8748 and HD15530 chips were discussed earlier. They will be included here only in the context of how they were used to implement the system design.

The hardware circuitry that requires specific description are:

1. Control Signal Generation Logic
2. Data Paths for the parallel modes of operation
3. Interrupt generation logic for the receive function
4. Terminal Address Selection circuitry
5. Serial Input/Output logic
6. Fiber optic driver/receiver circuitry

Timing diagrams for some of the above will be included in the discussions to clarify their operation.

1. Control Signal Generation

The primary function of the Control Logic is to manage the operation of the serial and parallel ports in conjunction with the HD15530 encoder/decoder chip. A schematic of this circuit is given in Figure 9. As can be seen from Figure 9, Port 1 on the μC8748 provides excitation for the control logic. Each bit on this port initiates a specific function and Table 2 provides a definition for them.

TABLE 2

μC8748 PORT 1 AND PORT 2 SIGNAL DEFINITIONS

Port 1 Bit No.	Definition	Port 2 Bit No.	Definition
0	LSH Input Port Control	0	Address Bit A
1	MSH Input Port Control	1	Address Bit A
2	LSH Shift Req Latch	2	Address Bit A
3	MSH Shift Req Latch	3	Mode Switch 8
4	Encoder Enable/Disable	4	Mode Switch 7
5	CMD/DATA SYNC Select	5	Mode Switch 6
6	Serial Gate Control	6	Mode Switch 5
7	Flip Flop control	7	Ser/Par RDY Strobe

There are four classes of functions that will be discussed. In particular, control of the 16-bit parallel input port, the parallel-to-serial shift register latches for the HD15530 chip, the enabling and sync pattern selection function, and control of the Serial Input/Output mode data stream are managed by Port 1 of the μC8748.

27

Parallel data input latching is accomplished by Port 1 bits zero

and one. The software performs a logical "OR" function to turn these
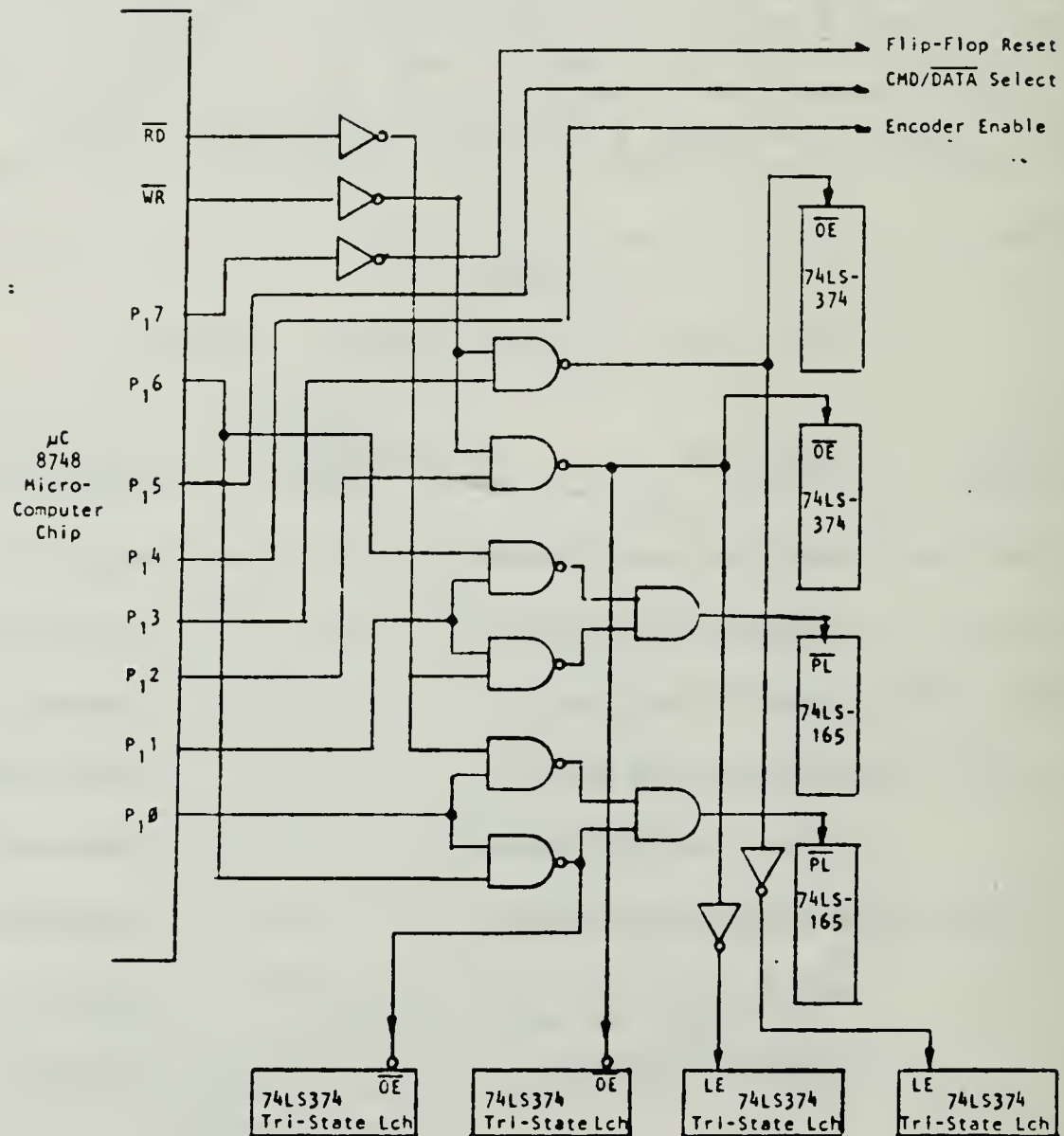
Figure 9. Control Logic

bits "on" or "off". It then executes an "Input to Accumulator from the

Bus" (INS A,BUS) to generate a "read" pulse. This pulse is "ANDed" with

Port 1 bits 0 or 1 to perform the latching function. This sequence of events loads the parallel data iteratively into the internal /ᴿC8748 buffer area for the Parallel Standard mode. In the Parallel Bypass mode, Port 1 bits 0 and 1 and bit 6 are set to one to simultaneously read and latch the parallel data from the parallel port directly to the parallel-to-serial shift register latches for input into the HD15530 chip.

The complement of this sequence is performed when data are received by the HD15530 chip and output to the parallel port. Port 1 bits 2 and 3 in conjunction with a "read" pulse are used to perform this function. Command/Data Sync selection and Encoder enable functions are also accomplished via Port 1. Bit 5 on the port is used to select the desired Sync type and Bit 4 enables the Encoder on the HD15530.

Timing diagrams are given in Figure 10 for all the functions above. Note that Port 1 is "Strobed" on and off by the instruction sequence "ANL P1,#00; ORL P1 desired bit; ANL P1,#00".

It is critical to clear the Port with an "ANL P1,00" after each sequence to prevent two devices from trying to write simultaneously. In the serial mode, data are transferred in and out of the system exclusive of the /ᴿC8748 Bus. Handshaking and control functions are, however, performed by the /ᴿC8748. Specifically, Port 1 bit 6 enables or disables the Serial Control gates to pass or block the serial data stream. Port 1 bit 7 is used to detect when 16 bits of serial data have been shifted in or out.
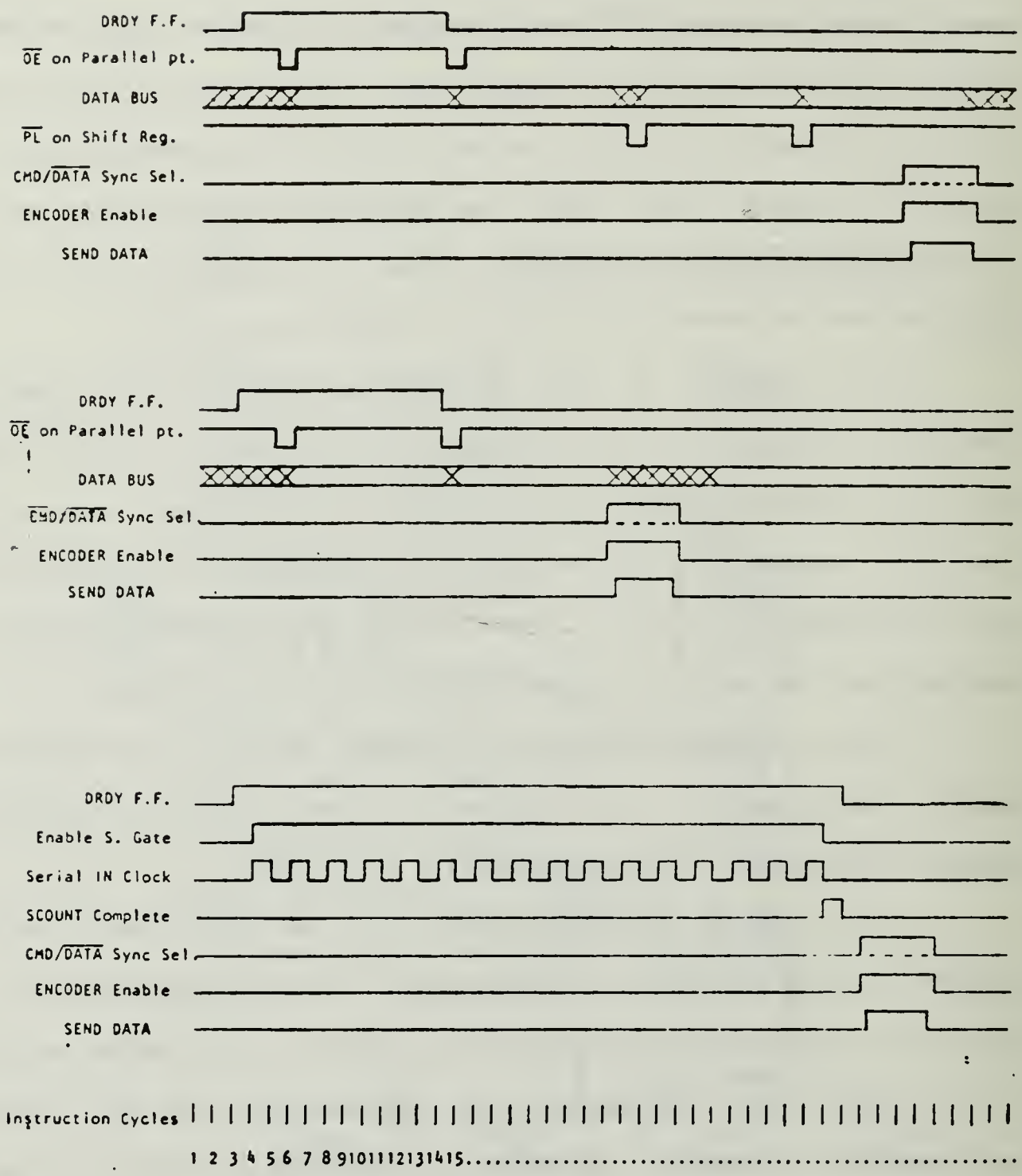
Figure 10. Timing Diagrams for Parallel and Serial Modes of Operation

The TØ and Tl test inputs on the /ᴜC8748 are used to sense the state of the HD1553Ø chip. TØ monitors the "Encoder Complete" state and Tl detects when a word has been received and decoded by the Decoder. On Port 2 bit 7 both the Parallel Input and Serial Input Strobes are monitored. By a series of Rotate Left thru Carry (RLC) and Jump if Not Carry (JNC) instructions, this bit is tested for ready.

Port 2 is also used to decode the desired mode of operation by reading the Switch positions on bits 6, 5, 4, and 3. Table 1 defines this relationship.

2. Data Paths for Parallel Modes of Operation

There are two modes of operation for the Parallel Input/Output functions. Figure 11 indicates the data path for the Parallel Standard mode and Figure 12 defines the data path for the Parallel Bypass mode. For both modes of operation, the Word Count must appear on the LSH of the parallel port along with the Subaddress field for each first (command) word. This information is read into the /ᴜC8748 and used to manage the data transfer sequence.

For the Parallel Standard mode, all data is buffered in the /ᴜC8748 on input and transferred out, in correct order, for the transmission sequence. For this mode of operation, the word count must not exceed 12 hex or 18 decimal or data will be lost. For the Parallel Bypass mode, only the command word (first word) is read into the /ᴜC8748. All data words are transferred directly to the input Shift Register for the HD1553Ø chip. For this reason, data transfers are performed faster

31
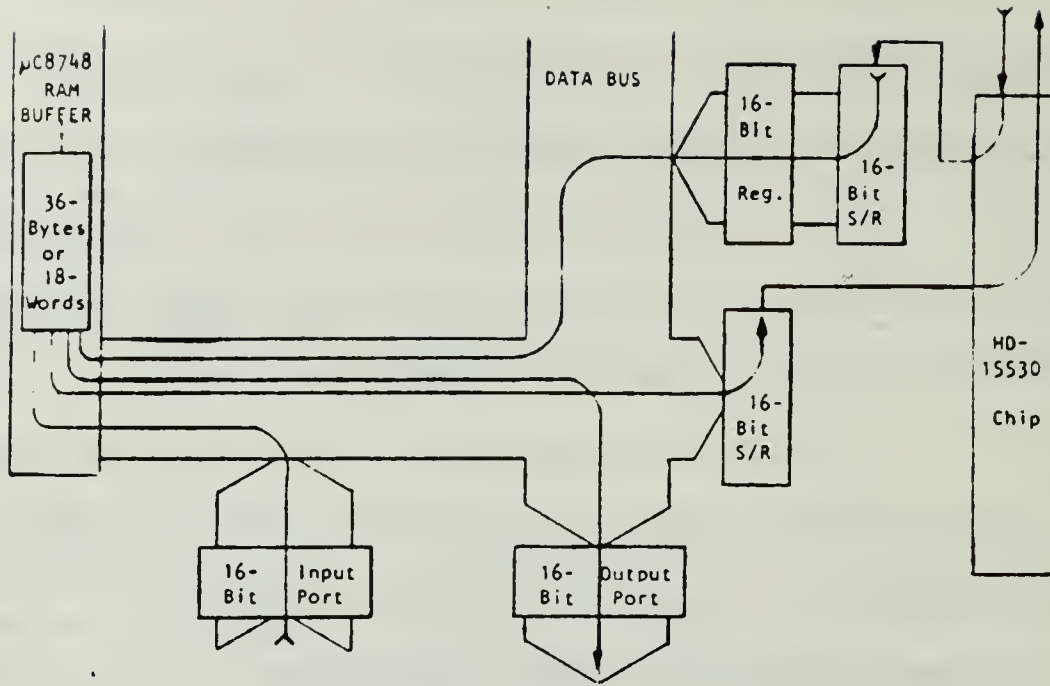
Figure 11.   Data Path for Parallel Standard Mode
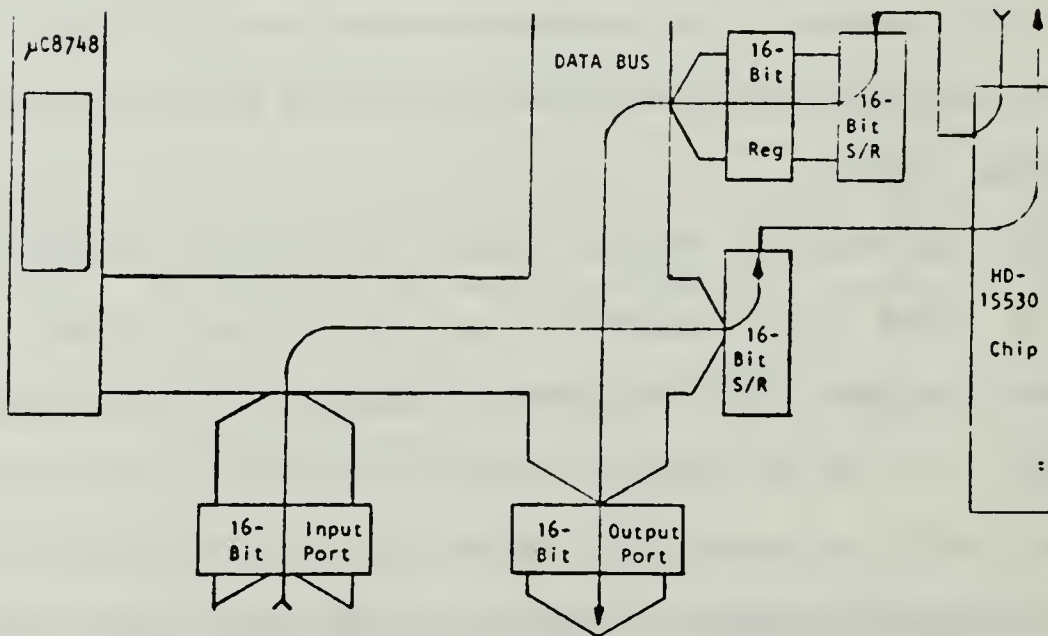
Figure 12.   Data Path for Parallel ByPass Mode

and the word count could extend to FF hex or 256 decimal. In this implementation however, the word count is limited to 20 hex or 32 decimal words to conform to Mil Std-1553.

3. Interrupt generation logic for the receive function

When a command word is received with the proper terminal address, an interrupt is generated. Special circuitry was designed to perform this function, alleviating the μC8748 from this task. Figure 13 indicates how the correct terminal address is detected. A comparator (74LS85) is used to compare the terminal address bits of a received command word with the switch settings on the upper 4 of the DIP switch. It is only enabled during receipt of a command word and therefore eliminates detection of an improper terminal address during a data word reception. The "A=B" signal triggers an interrupt on the μC8748. The interrupt remains cleared by application of the Command/Data NOT signal. Figure 13 shows this relationship.

4. Terminal Address Selection Circuitry

Figure 13 also shows this circuitry. The resistors are used to "pull-up" the input pins on Port 2 of the μC8748. The "ON" side of these switches are grounded. Thus, to select a specific terminal address, the complement of the address is manually selected.

5. Serial Input/Output Logic

This circuit is indicated in Figure 14. Two Serial Control gates are used to pass or block the input or output stream. Port 1 bit 6 controls this gate. The "Exclusive OR" gates (74LS86) allow clocking of the serial data either by the HD15530 chip (in all modes) or by the Serial Input/Output operation; the μC8748 controls who clocks data.

33
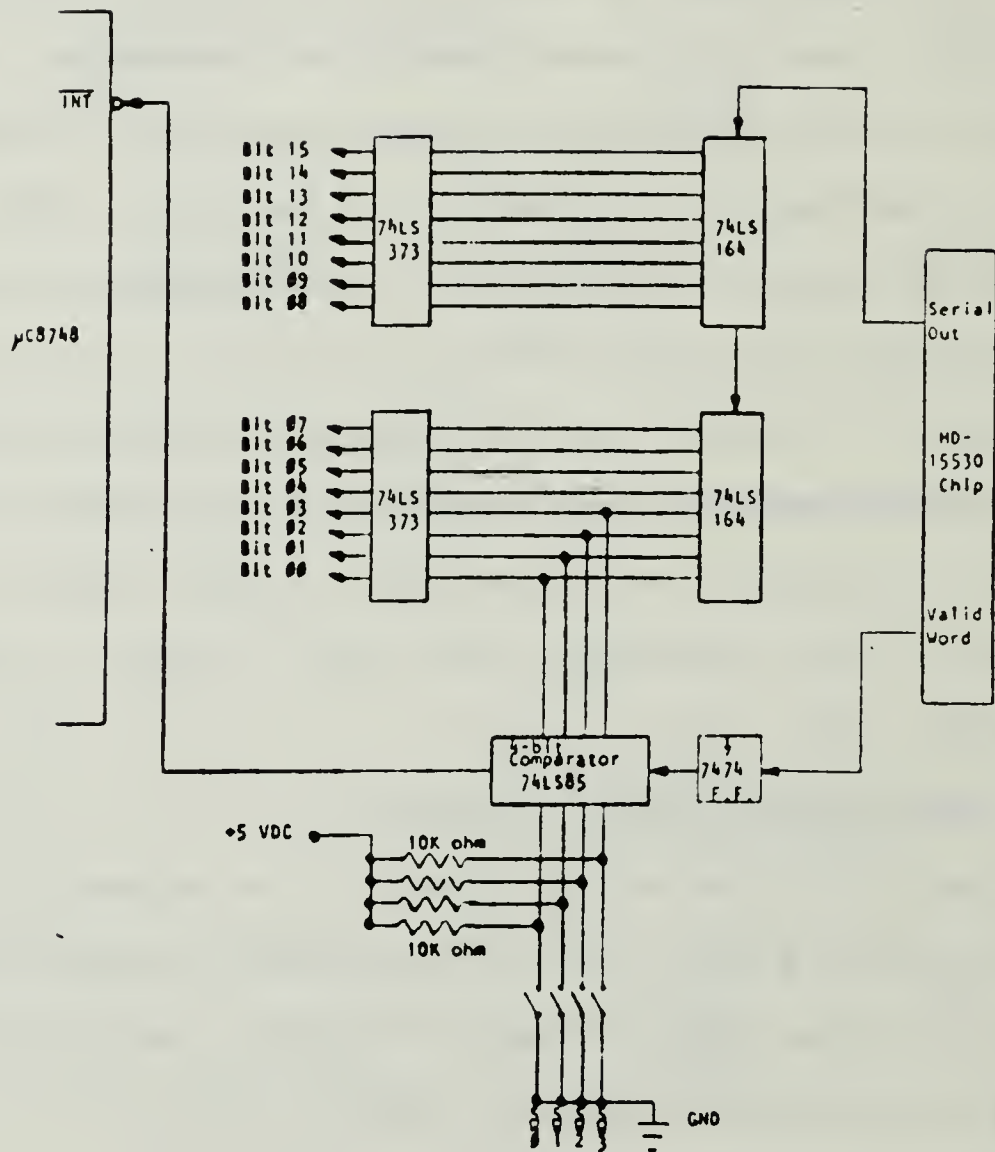
Figure 13. Terminal Address Selection and Interrupt Generation Circuits
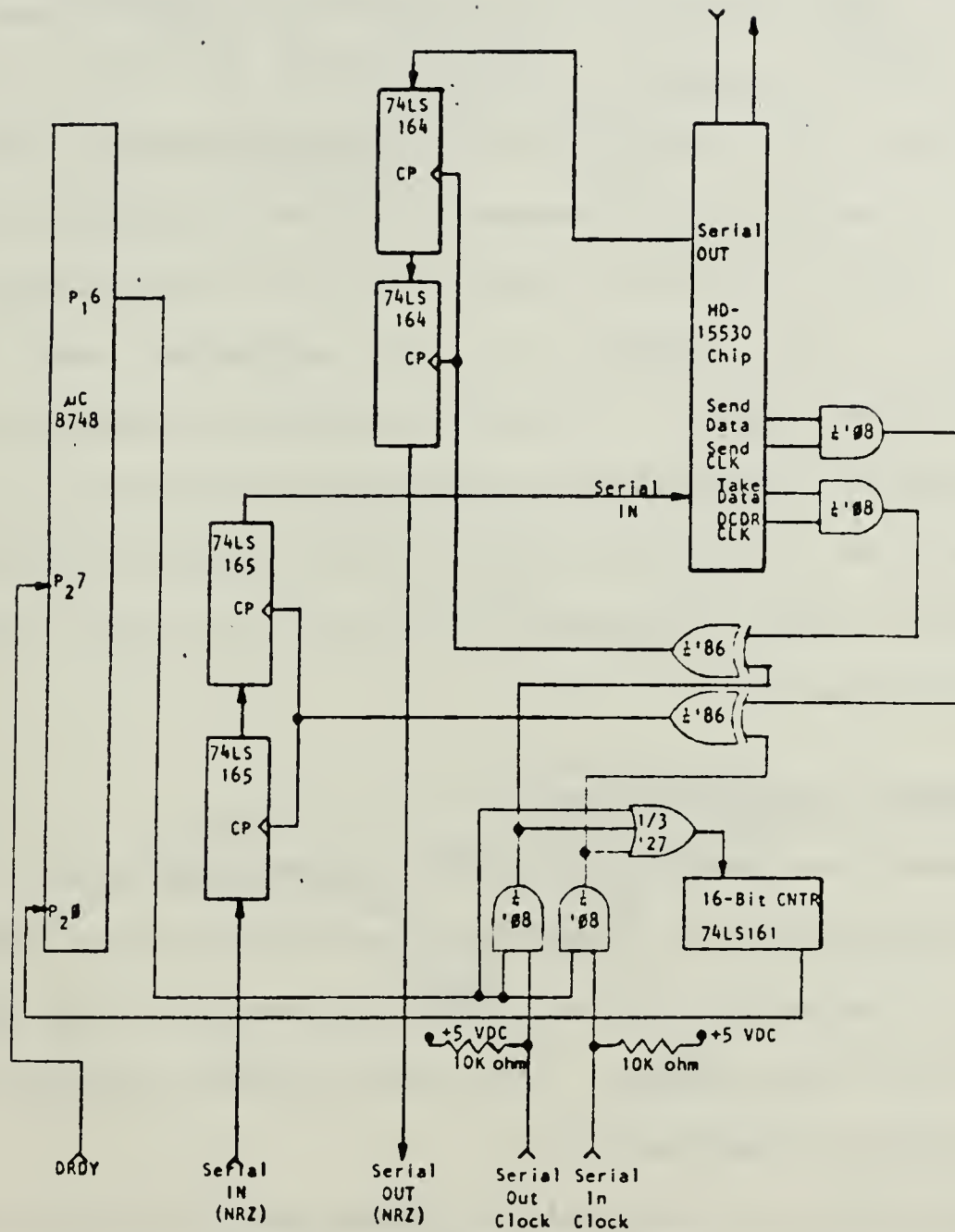
Figure 14. Serial Input/Output Logic and Data Paths

6. Fiber Optic Receiver/Driver Circuitry

The fiber optic front end was designed using a Hewlett-Packard HFBR1501/HFBR2501 transmitter/receiver snap-in optical link. This was chosen because of availability as well as performance characteristics. This link has a bandwidth which exceeds the Mil Std-1553 1 MHz requirement. Fabrication of the cable was relatively simple and the snap-in connectors ensured reasonable alignment and optical coupling.

Figure 15 gives the circuit design used in this effort. This is similar to one suggested in the Application Note 1009 from Hewlett-Packard [Ref. 5]. One minor change was incorporated to allow for adjustment of forward current across the Light Emitting Diode (LED). This was done to allow for "tuning" of the intensity on the LED to match the optical coupling characteristics of the link as well as to prolong the life of the LED.

B. SPECIFIC SOFTWARE ROUTINES

Program development for this effort was performed manually. Top down design was accomplished followed by flow charting, coding, assembly, and finally, linking and loading (EPROM programming). This process was tedious and time consuming. More user-friendly software development tools would have been beneficial.

The software can functionally be broken down into six routines. Each of the routines will be discussed along with their associated flow charts. Five of the routines are for the five modes of operation. The remaining routine performs the Restart and Initialization functions.
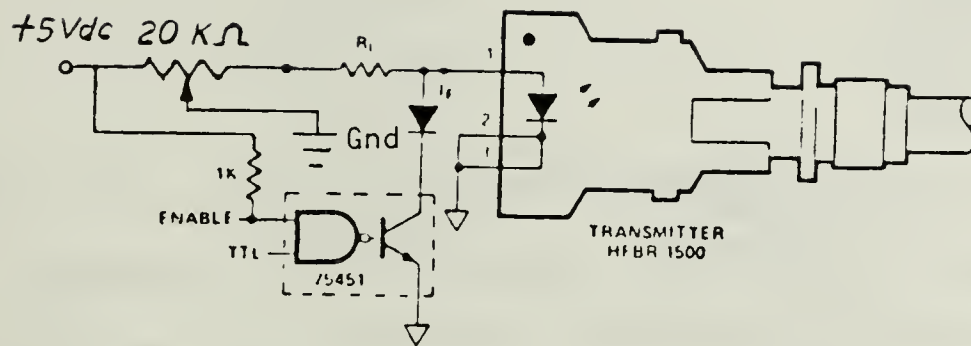
Fiber Optics Driver Circuit
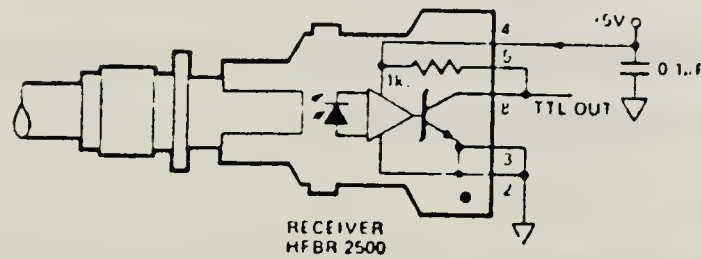
Fiber Optics Receiver Circuit

Figure 15. Fiber Optics Receiver/Driver Circuitry [Ref. 5]

Specifically, the routines are:

1. Restart/Init
2. TSTPAT
3. ECHOMODE
4. Par ByPass
5. Par Std
6. SerMode

Each routine is independent and forms a module. Each module, except the Restart/Init and TSTPAT routines, have a corresponding Interrupt Service Routine. To facilitate understanding and be succinct, only a brief discussion of each module will be presented. Complete program listings with comments are included in Appendix B.

1. Restart/Init

This module performs memory and port initialization as well as mode selection functions. All memory is cleared and Port 1 is zeroed. The Timer/Counter constant is loaded and the Mode Switch is read. The selected mode causes the flags F0 and F1 to be set or cleared appropriately. Control is then passed to the selected mode routine. Flow charts are shown in Figure 16a and 16b.

2. TSTPAT

This module generates a test pattern of alternating ones (1s) and zeros (0s) with an appropriate terminal address and command word. As implemented here, sixteen data words are transmitted each time the Timer/Counter interrupt occurs. This mode of operation is useful for hardware checkout as well as fiber optic link connectivity analysis. Used in conjunction with the EchoMode on a second board, a complete loop can be evaluated for proper operation. A flow chart is given in Figure 17.

38
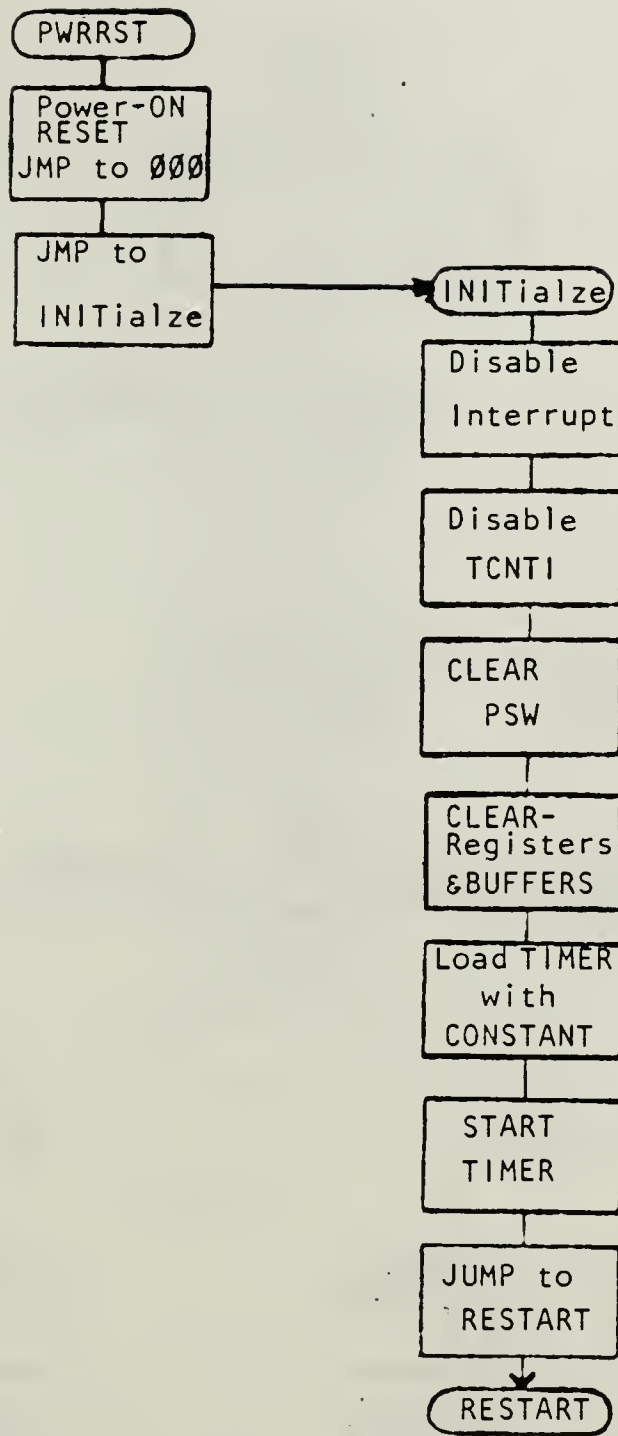
Figure 16a. Flow Chart for the INIT Routine
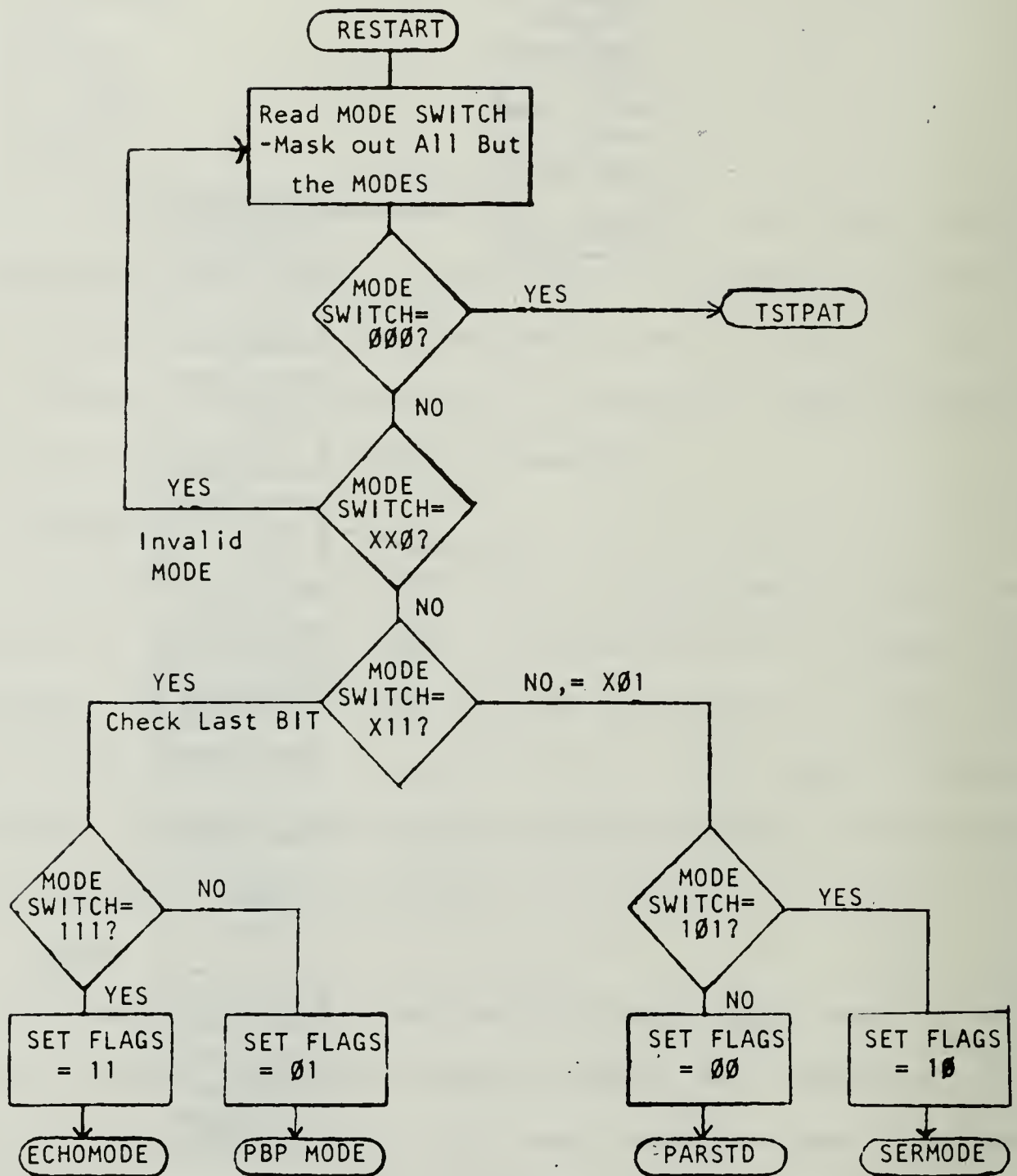
Figure 16b. Flow Chart for the RESTART Routine

40
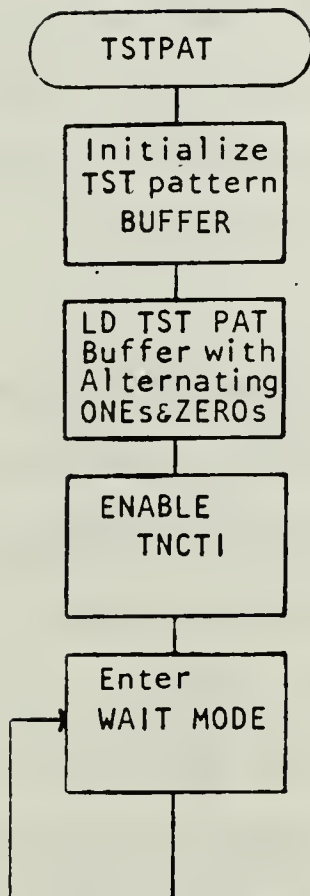
Figure 17. Flow Chart for the TSTPAT Routine

3. ECHOMODE

As mentioned above, this mode of operation could be used to test a complete fiber optic link loop. It could also be used to boost an optical signal on a extended link in a relay fashion. The module basically detects a command word addressed to it and echos the word along with associated data words back to the originating device (in this implementation), or to a third party board (if the Terminal address was changed). It could also be used in a delay-insertion loop configuration. A flow chart is included in Figure 18.

4. ParBypass

This is the least complex and fastest mode of operation. Data is not buffered in the μC8748 but passed directly to the Shift Register for input to the HD15530 chip. This module does, however, read in the command word and mask out the Word count. It saves the word count in a register for use as an event counter. The parallel bypass process begins by loading the 16-bit input port with a command word containing the terminal address and the number of words to transfer. The μC8748 reads in the command word and extracts the word count. It transmits the command word via the HD15530 chip, and then enters a wait mode pending loading of the next (first) data word. Upon detection of the next data word, it transfers the word directly to the HD15530 Shift Register and enables the encoder with a Data Sync. This process iterates until the word count is exhausted. It then returns and waits for the next sequence. A flow chart is given in Figure 19.
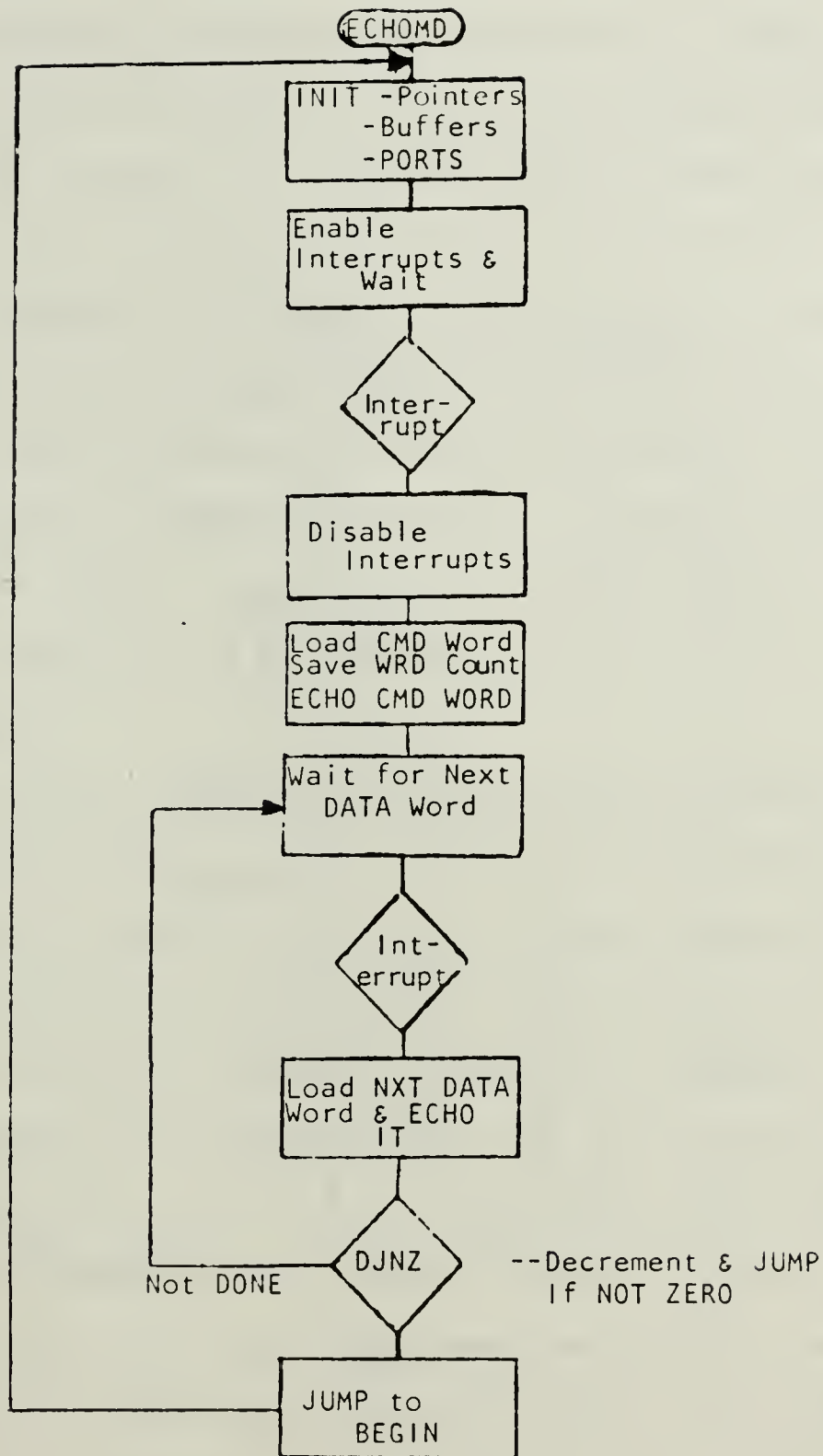
Figure 18. Flow Chart for the ECHOMODE Routine
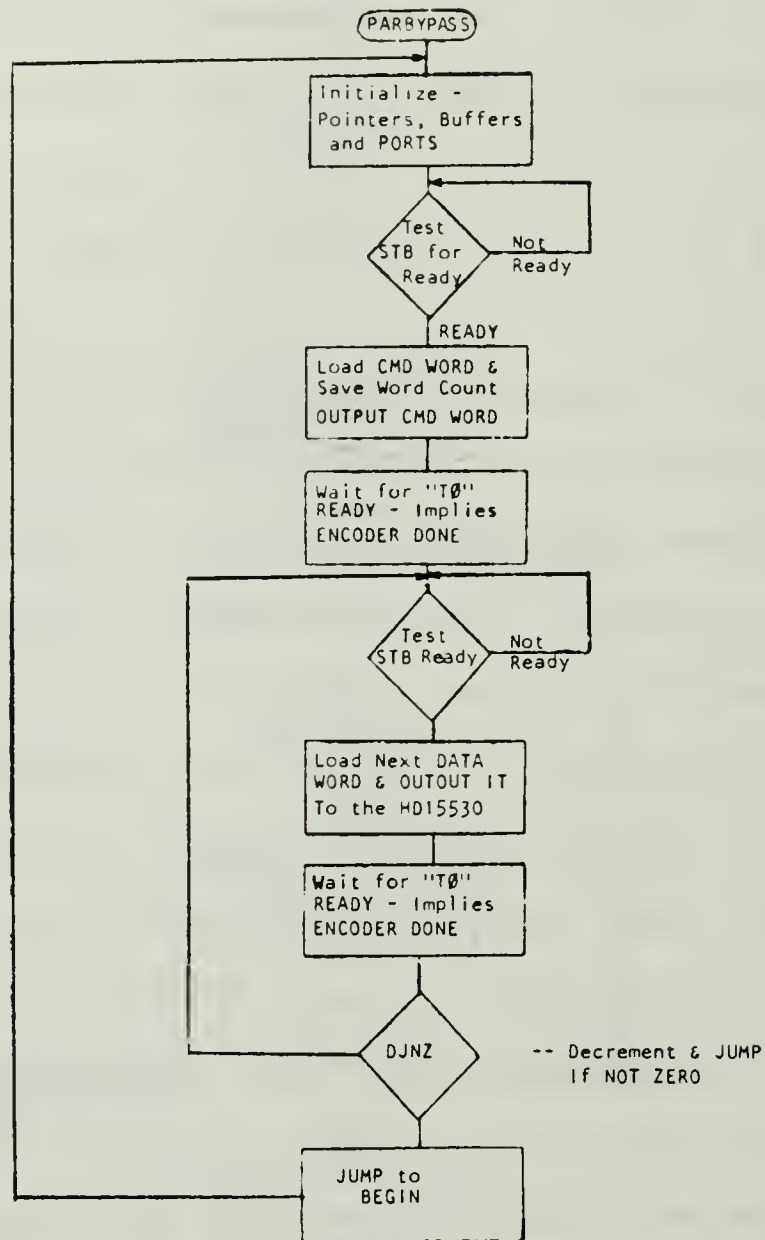
Figure 19. Flow Chart for the ParBypass Routine

5. ParStd

Data Buffering is provided for in this module. Parallel data is read into a buffer area within the /ЧC8748 and transmitted out after the entire block is buffered. The sequence begins by reading the command word, masking out the word count and saving it in the buffer. The buffer pointer is incremented and the next (first) data word is read in and buffered. This process iterates until the word count is exhausted. The maximum word count is limited to 18 decimal 16-bit words. This is due to the limited memory on board the /ЧC8748. This mode of operation is useful when the external parallel device is slower than the fiber optic link. A flow chart is given in Figure 20.

6. SerMode

There may be a requirement to provide for Serial input/output of data to the fiber link. An example might be a First-In-First-Out (FIFO) chip which can be loaded and read serially (such as the AMD 2813 FIFO). This module allows such an interface. The serial data can be directly loaded into or read from the Shift Registers by an external device. A point to recognize is that the word count must still be read on the LSH of the parallel port. The addition of one chip to the design could eliminate this peculiarity. The sequence begins by latching the word count on the parallel port. The /ЧC8748 reads the word count and saves it in a register. The /ЧC8748 then enables the Serial Control Input Strobe gate and awaits the loading of 16 bits of serial data. It then activates the HD15530 encoder with a Command sync, decrements the word count, and waits for the next 16-bit data word. Then the encoder is enabled with a data sync and the module returns for the next serial data word.
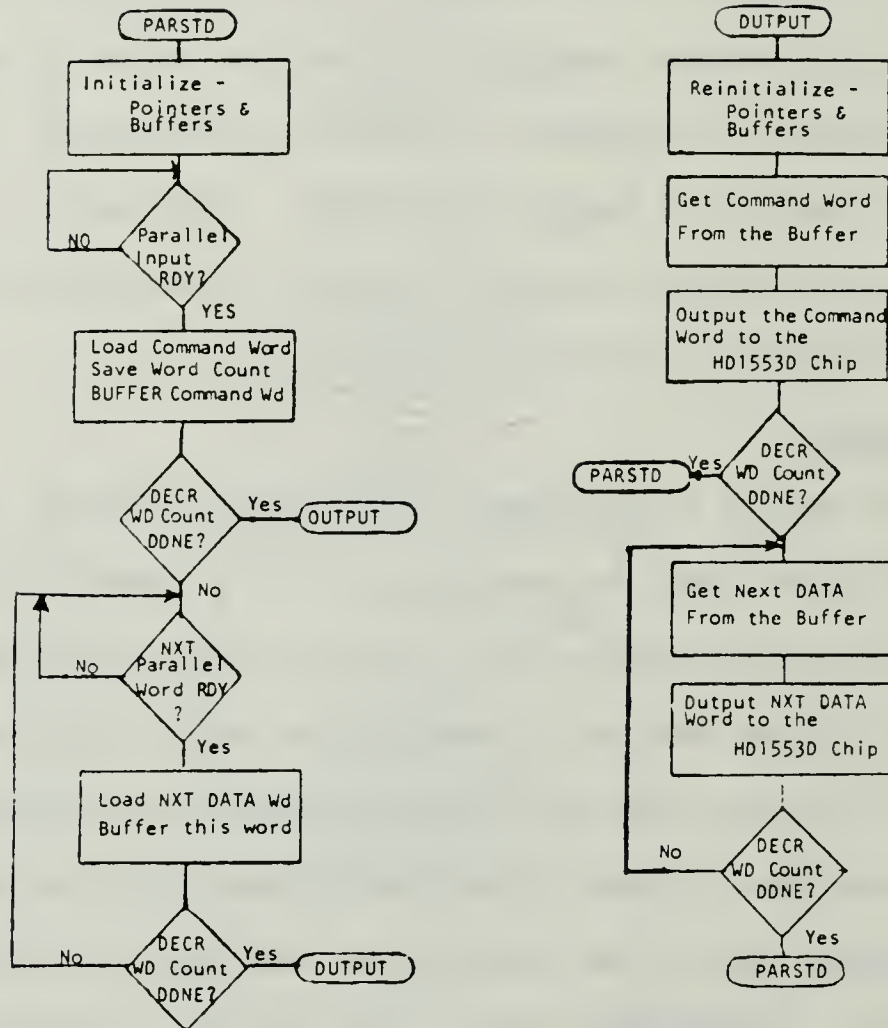
Figure 20. Flow Chart for the ParStd Routine

This process continues until the word count is exhausted. Following this, the module returns to the beginning for the next serial sequence. A flow chart is given in Figure 21.

C. INTERFACING AND TESTING

1. Z-8000 Evaluation Board

A means of demonstrating and verifying the completed design was needed. To address this issue the Zilog Z-8000 Evaluation Board from Advanced Micro-Devices (AMD-Z8000) was used. The Z-8000 Evaluation Board provided the requisite parallel I/O port and could operate at speeds fast enough to exercise the design properly. Resident on board was also a Line-by-Line Assembler for program development. It provides an RS232 port to interface to a dumb terminal for entry and display. A Commodore VIC-20 was configured to perform like a dumb terminal and interfaced to the Evaluation board.

The characteristics of the Evaluation board include: a Z-8000 16-bit microcomputer chip operating at 4 MHz; 8 kilobytes of RAM; up to 12 kilobytes of EPROM where the monitor and Line-by-Line Assembler resides; two programmable Serial I/O ports; and a 24-bit programmable I/O port. Additionally, it plugs into the Intel Multibus on which the design was based.

2. Parallel Interface Ports

The system design was accomplished on an Intel Multibus Prototype board using power, clock and databus lines to conform with the bus. As such, the 16-bit parallel I/O port was interfaced to the Multibus bus on the specified Data Bus lines. Additionally, the Constant Bus Clock pin

47
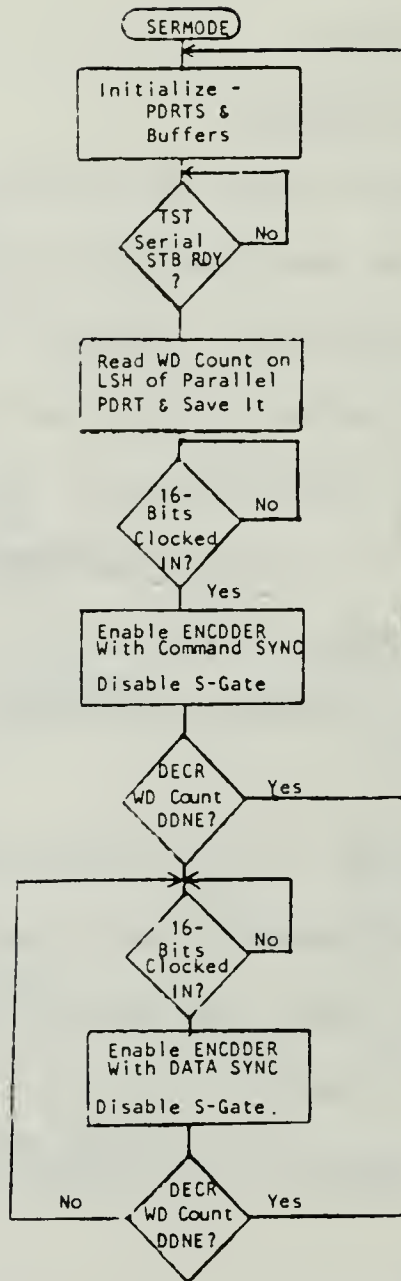
Figure 21. Flow Chart for the SERMODE Routine

48

was utilized to provide a clocking source to the μC8748 and the HD15530 chip. The Z-8000 board had to be interfaced to this bus to act like a standard Multibus Card. To accomplish this, a plug-in card was fabricated to take 16 lines from the Z-8000 Evaluation board's parallel port on P3 to the data bus on the Multibus back plane. Handshaking was provided via additional pins on the Multibus address bus. This is summarized in Figure 22.

3. Built-In-Test Function

One possible use for the MIL STD-1553 board could be connectivity analysis of a fiber optic link. An example would be the validation of a newly installed branch on an existing Local Area Network (LAN). It might be impractical to use two workstations on the existing network to check out the new branch on the LAN. The Built-In-Test function (BIT) incorporated on one of the MIL STD-1553 boards answers this need. Because the complete design was built on one multi-bus prototype board, it could be portable. With a battery pack to power the board, remote checkout of a fiber optic branch would be relatively simple as well as thorough.

To accomplish the BIT function, two seven-segment displays along with decoder/drivers were built on one of the boards. The inputs to the decoder/drivers were connected to the Least Significant Half (LSH) latch of the parallel output port. The Output Enable (OE) of this latch was strapped in the "on" state. The Latch Enable (LE) was strapped to port 1 bit 7 of the μC8748. Also included on this board was a momentary switch which was strapped to port 2 bit seven of the μC8748. This circuitry is

49
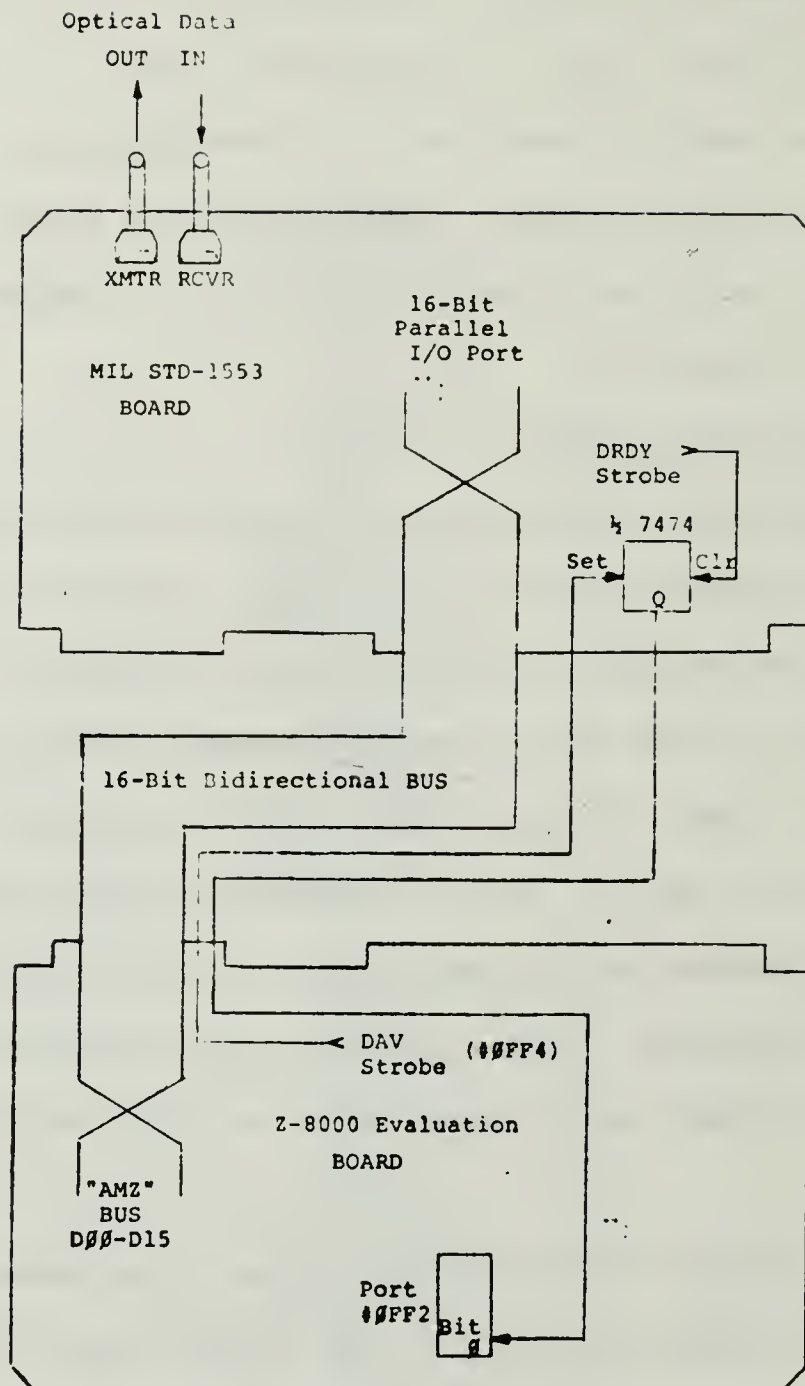
Figure 22. Z-8000 Evaluation Board to Mil Std-1553 Card Parallel
Interface

quite similar to the display portion of the Single Step and Display card given in Appendix C.

A flow chart of the BIT software routine is shown in Figure 23. As can be seen from the figure, the sequence of events begins by reading a pseudo-random byte from the free-running timer/counter. The Terminal Address byte is then concatenated with the pseudo-random byte to form a 16-bit Command Word for output to the HD15530 Encoder. Then the Encoder is enabled to transmit this Command Word across the fiber optic link. At this time a wait mode is entered pending arrival of the echoed word from a second board configured in the "Echo Mode". Upon arrival of the echoed word the BIT sequence reads the pseudo-random byte and compares it to the original byte transmitted. If the two bytes compare, a "word counter" is incremented and output to the seven segment displays. The software routine then enters a loop pending detection of closure of the momentary switch, at which time the sequence repeats.

By performing the above, a fiber optic link can be evaluated for connectivity. The procedure described above could be accomplished by technicians with minimal experience and verifies proper operation at a link with a fair degree of confidence. This capability was used to arrive at the results presented in Chapter VI.
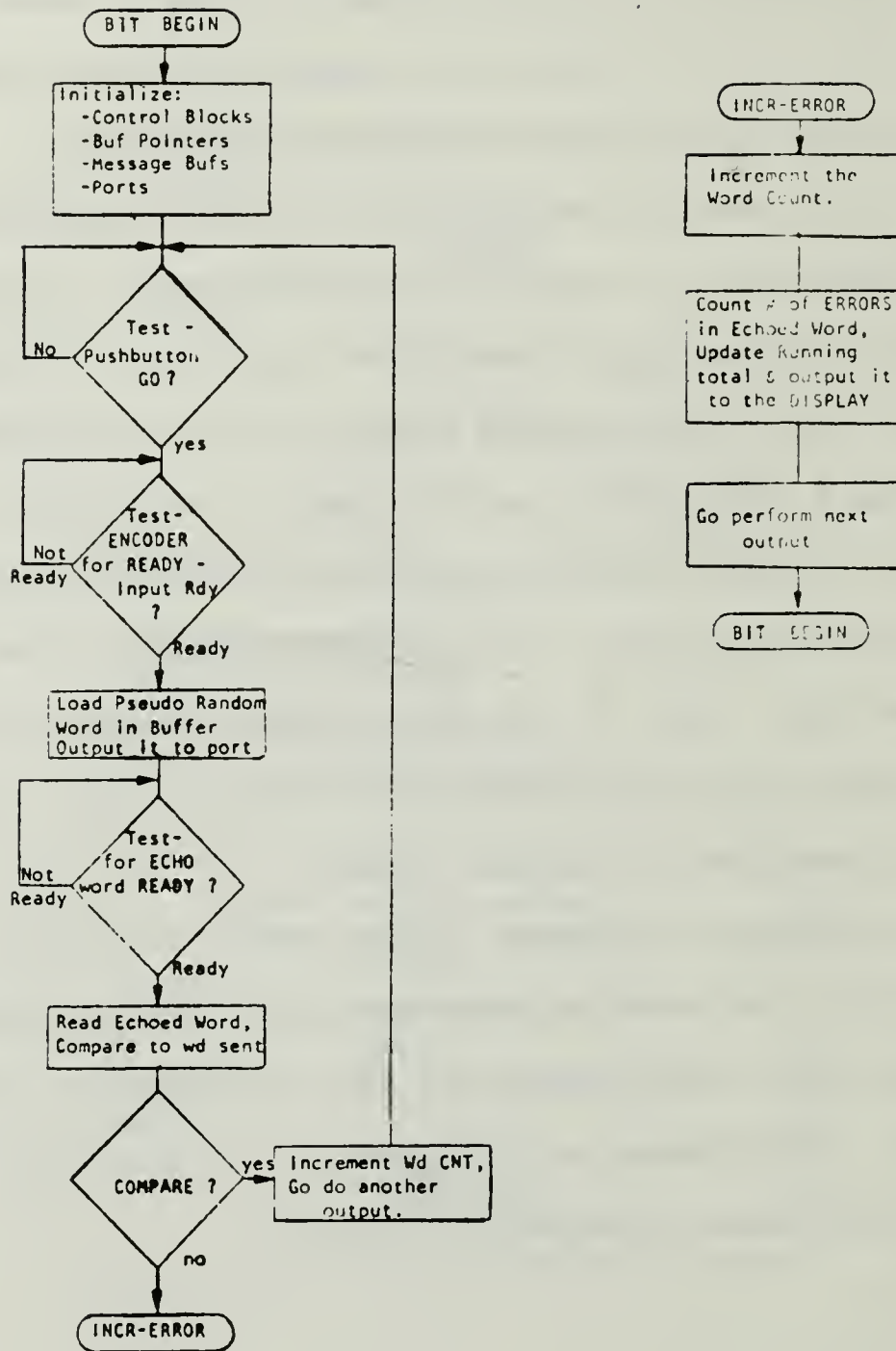
Figure 23. Flow Chart for the Built-In-Test (BIT) Routine

V. DESIGN DEVELOPMENT TOOLS

Any task undertaken with this degree of complexity requires tools and aids to make the job manageable. More efficient tools and aids usually yield quicker and more accurate results. As much of this effort was accomplished while separated from the academic environment with its laboratory and tools, support equipment had to either be borrowed, bought, or fabricated. The first item to be procured was a JDR Instruments Dual Trace Oscilloscope. This, by far, was the most critical tool. Another critical tool was the Intel Prompt-48 programmer to "burn" the software into the μC8748. The NPS staff graciously loaned this item for an extended period of time. Late in the design stages, it became evident that some sort of single step and display device was required to debug the software and check out the hardware. The Intel MCS-48 Family of Single Chip MicroComputers User's Guide [Ref. 4] provided a single step circuit for use with the μC8748. This circuit was incorporated on a plug-in card containing seven-segment displays and decoder/drivers. The plug-in card interfaced to the Mil Std-1553 cards via the Multibus backplane to provide single step and display capability. The final tool was to incorporate strapping options in the design to allow the plug-in card to access the data bus and address bus of the μC8748. More efficient tools are available and would have reduced the development time considerably. For example, an Assembler with a linking loader that would interface to the Prompt-48 and download assembled and linked code (machine code) for programming the μC8748 would have been

beneficial. A Logic State Analyzer would also have been helpful. Discussion of the available tools and their utilization follows.

A. THE PROMPT-48 PROGRAMMER

This unit allows for entry of the assembled machine code into its internal memory. This can be accomplished a couple of ways. The quickest and easiest is to load the memory with the machine code from a preprogrammed μC8748. This is useful in latter stages of development when only minor changes and additions to the code are required. If none of the original program exists, then entry of the machine code can be done using the keypad functions. This method is tedious at best. Once the coding has started, new codes should be placed so as to not overwrite the existing code. This requires sufficient allocation for each program module to permit future modifications without overwriting existing code which would require a manual relinking and loading. Once the code is resident in the Prompt-48 memory, it can be viewed and edited and finally programmed into the target μC8748.

B. SINGLE STEP AND DISPLAY CARD

As mentioned in Chapter II, the μC8748 design facilitates single stepping through program memory. The Single Step function can be accomplished as mentioned above. Figure 24 shows this circuit along with three-seven segment LED displays with the requisite BCD to Seven Segment Decoders/Drivers (7447). With this circuit, the next address, which appears on the databus and port 2, can be displayed and executed one instruction at a time. The timing diagram for this circuit is also given in Figure 24. This card became very useful late in the development for

54

SINGLE STEP CIRCUIT
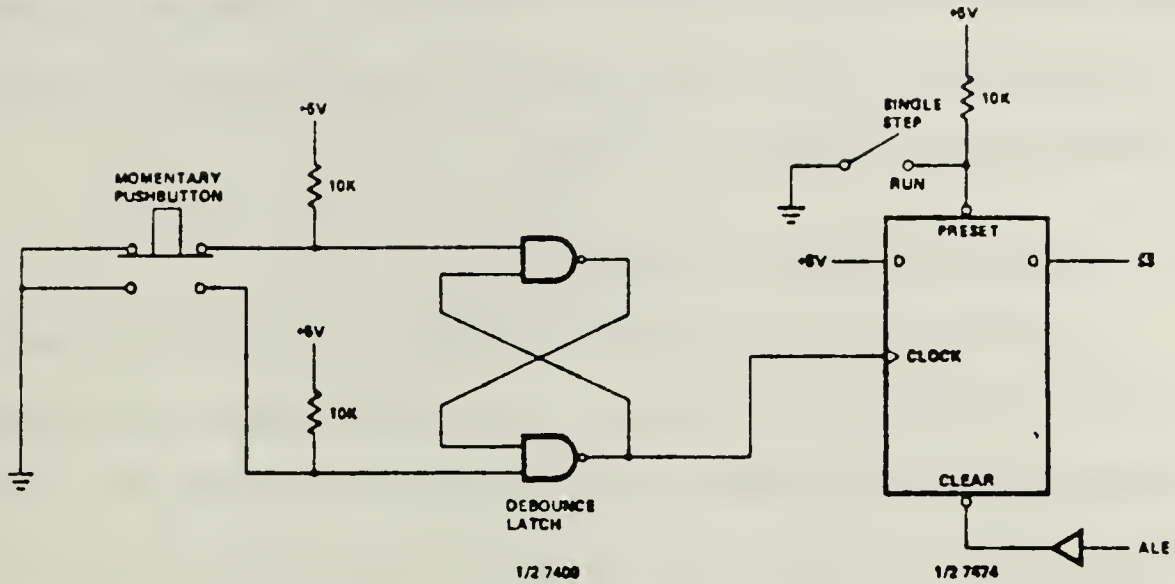
SINGLE STEP TIMING
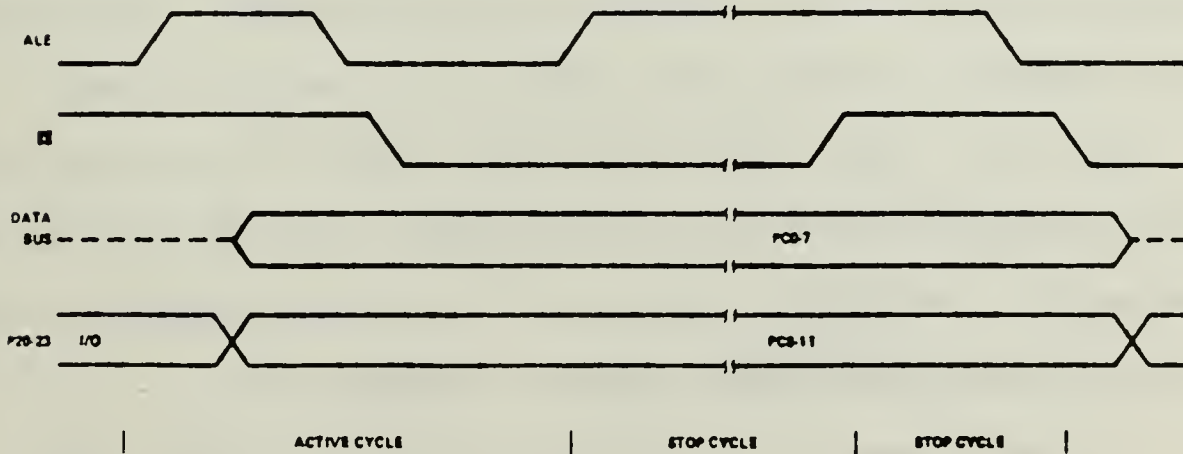
Figure 24. Timing Diagram for the "Single Step and Display" Card

[Ref. 4]

55

debugging the software and verifying the hardware. For example, a module could be single stepped through until the interrupts were enabled. At that time, an interrupt could be simulated by momentarily grounding the interrupt pin and then single stepping could resume to verify proper Interrupt Service Routine execution.

C. STRAPPING OPTIONS FOR TESTING

In order to use the plug-in card with a Mil Std-1553 card, the Mil Std-1553 card has to be strapped to allow the Address Latch Enable (ALE) signal to latch the address into the Most Significant Half (MSH) of the parallel port to display this address on the plug-in card for Single Step and Display. Additionally, the Output Enable for the MSH of the parallel port has to be tied low to allow for continuous display. Further, the Single Step pin on the /UC8748, which would normally float high, should be strapped to the Single Step output pin on the plug-in card.

When the Single Step and Display card is not being used, the ALE strap should be removed and the parallel latch strobe strap should be connected. The Single Step pin on the /UC8748 should be allowed to "float". This relationship is shown in Figure 25.

In normal operation, the following conditions should exist:

1. ALE unconnected
2. Single Step unconnected
3. Parallel Latch Strobe (MSH) should be strapped

The most valuable tool was the Single Step and Display plug-in card, since program debugging was possible with this card. The Prompt-48 was

Figure 25. Strapping Options for Testing

essential for programming the μC8748 after the code was developed. The oscilloscope was very useful in verifying hardware operation.

During development, the fiber optic link was disabled to prevent damage and proper operation was restored once the hardware and software designs were complete. A copper wire was used to link the two cards' Mil Std-1553 interfaces for testing purposes.

VI. <u>SYSTEM RESULTS</u>

The goal of this study was to demonstrate the implementation of Mil Std-1553 using a fiber optic link. This goal was achieved with qualifications. The first qualification is that strict compliance to Mil Std-1553 could not be achieved in some modes of operation. For example, in the Parallel Standard mode, only 18 words could be transferred due to the limitation of the internal RAM on the μC8748. Another example of non-compliance is the requirement for Status word to be transmitted not more than 3 microseconds after receipt of a frame of data. This capability was not included from the start in an attempt to limit the scope of this effort and thus to limit the time required to complete it. Several other Mil Std-1553 requirements were waived to facilitate reasonable time to completion and completion with limited resources.

The concepts of fiber optic implementation of Mil Std-1553 were sufficiently demonstrated to validate the feasibility of this approach. BiPolar Manchester II encoded data with proper sync and parity bit were successfully transferred over a fiber optic link in sufficient quantity and quality to demonstrate few or no barriers to full development of this concept.

A. MAXIMUM DATA RATE

As mentioned in the text, the fastest data transfer rates can be accomplished in the Parallel Bypass mode and the slowest rates appear in the Parallel Standard Mode. The slower rates are caused by the

59

requirement to divert the data into a buffer storage area internal to the /μC8748. Several additional instruction executions are required as can be seen from the program listings and flow charts. Handshaking requirements are doubled as can be seen from the control circuitry. The Serial mode transfer rate is comparable to the Parallel Bypass mode. Essentially, the same number and types of control signals exist in these two modes. The number of instructions in the critical loops are roughly equal except that in the Serial mode fewer instructions are needed due to hardware that handles some of the functions.

The maximum rates, therefore, will be based on the Parallel Bypass mode. Table 3 contains the maximum data rates for calculated and observed rates. The observed data rates are slower because both the /μC8748 and the HD15530 chip were not clocked at their maximum rates. The calculated data rates are based on assumed maximum clocking rates for these devices and the number of instruction cycles required (worst case) times the minimum instruction times.

TABLE 3

MAXIMUM DATA RATES (CALCULATED AND OBSERVED)

Unit	Maximum Data Rate Calculated/Specified	Observed
/μC8748	147 Kilobits/sec	134 Kilobits/sec
HD-15530	1.56 Megabits/sec	1.36 Megabits/sec
HFBR 0500 Link	5 Megabits/sec	1.36 Megabits/sec
Total System:	147 kilobits/sec	134 Kilobits/sec

The maximum data rates are not affected by either the HD15530 chip nor the fiber optic link because these devices are rated at higher than the requisite 1 Megabit data rate.

60

B.   INTERFERENCE/EMANATION AND ISOLATION

The greatest advantage of fiber optics lies in their ability to be immune to RFI and EMI.   An added benefit involves total electrical isolation. A further benefit is the extremely high data rates offered by fiber optics.   The coupled benefits of high data rates, RFI and EMI immunity, and electrical isolation suggest that fiber optic communication channels approach the ideal communications channel.   Prior to fiber optics, it was not possible to route communication links through fuel tanks, in close proximity to generators and high power equipment, or in water of any kind (underwater cables are hermetically sealed). Fiber optic links are unaffected by these environments. The mere weight of copper wire (large enough to carry information) is much greater than the weight of a fiber optic link capable of carrying significantly more information per unit of time.

Another area in which optical fibers proves superior to copper wire is data security. A typical copper wire channel carrying digital data can be tapped without making any contact with the conductor itself.   In fact, many copper wire channels radiate in free space enough to be detected and compromised at a considerable distance. This is referred to as compromising emanations. Fiber optics on the other hand cannot be easily tapped nor do they radiate in free space in the RF spectrum.

Applications which require high data security are already incorporating fiber optics because the alternative copper wire approach requires RFI shielding and physical security efforts which are very expensive compared to the improved data security payoff.

C. BUILT-IN-TEST (BIT) RESULTS

As mentioned in Chapter IV, the BIT capability was used to obtain some results that graphically demonstrate the operation of the design. The configuration required to obtain the results will be briefly described here.

To establish a complete loop, one MIL STD-1553 board was placed in the "Echo Mode" configuration and the second board (containing the seven-segment displays) was set up for the BIT function. Two optical fibers were used to interconnect the board to provide a closed loop.

When the push button was depressed (or held) a pseudo-random byte was concatenated with the Terminal Address to form a Command Word containing pseudo-random data. The word was transmitted across the link to the "Echo Mode" board. This board, in turn, echoed the word back to the originating board completing the circuit transfer. If both the transmitted Command Word and the echoed Command Word compared, the word count was incremented and displayed. When the pushbutton was held down a continuous stream of Command Words containing pseudo-random data were transmitted, echoed, and compared.

A means of plotting some results for inclusion here was desired. This was accomplished using a Hewlett-Packard 1641 (HP1641) Logic State Analyzer interfaced with a plotter via the Hewlett-Packard Bus (HPIB).

62

Two of the input channels of the HP1641 were connected to the transmit and receive pins on the HD15530 chip. The HP1641 External Clock input was connected to the HD15530 clock for synchronization. The HP1641 was triggered by connecting its External Trigger input to the Interrupt pin on the /μC8748. This was done to detect the arrival of the echoed Command Word. The trigger would occur only if the echoed word contained the correct Terminal Address and was properly received as a Command Word with correct parity and Manchester Bipolar encoding. A high degree of confidence in the proper operation of both cards using optical fibers could be demonstrated in this way.

The plot of the results are presented in Figure 26. Each pulse indicated on Channel 0 represents the transmission of a pseudo-random Command Word from the transmitting board. Channel 1 indicates the receipt of a properly echoed Command Word from the echoing board.

As can be seen from the Figure for this particular sample period there were approximately 115 20-bit Command Words containing 8 bits of random data successfully transmitted and echoed back. As mentioned above the following conditions must exist to have generated each of the 115 pulses in the plot:

1. Proper encoding, decoding, and re-encoding of all 20 bits per word.
2. Correct Parity generation, detection, and checking of all 20 bits.
3. Proper regeneration of the bit clock on both receiving ends of the fiber link.
4. Proper Terminal Address generation and detection on both ends (4 bits).
5. Successful generation, detection, and comparison of 8 bits of random data per 20-bit word.

There were no data "drop-outs" indicated in any of the plots produced during the design validation tests performed. This result should be

63

TIMING DIAGRAM   TRACE-COMPLETE
TALK ONLY
EXPAND INDICATOR [ON]   250US/CLK
  GLITCH DISPLAY [ON]   5MS/DIV
   MAGNIFICATION [X1]   [1.75MS]
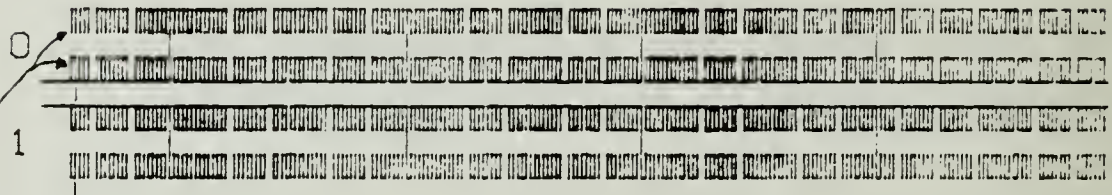
TIMING DIAGRAM   TRACE-COMPLETE
TALK ONLY
EXPAND INDICATOR [ON]   2.5US/CLK
  GLITCH DISPLAY [ON]   50US/DIV
   MAGNIFICATION [X1]   [17.5US]
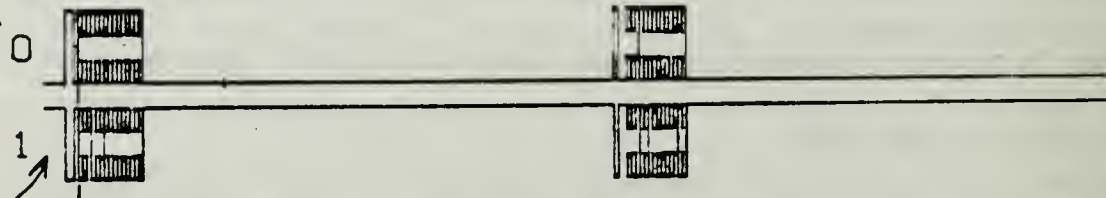
| BIT TIMES | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

COMMAND WORD

SYNC | REMOTE TERMINAL ADDRESS | T R | Fxudo Random DATA | P |

Figure 26   Built-In-Test (BIT) Results

64

qualified by the following. The validation test was performed using two MIL STD-1553 cards - one in the Echo Mode and one in the "BIT" mode. Two HFBR0500 Series Snap-In Optical Fiber Links were used that were each approximately 20 feet long. The forward current across both HFBR1501 transmitters was adjusted to match the characteristics of the optical fibers to ensure reliable optical transmission and reception on both ends of the link. The two MIL STD-1553 cards were powered by separate power supplies with no common ground connected. Thus, the boards were optically isolated.

A simple procedure was used to match the HFBR1501 transmitters to the optical fibers. An oscilloscope probe was placed on the "Valid Word" pin of the HD15530 chip to sense the receipt of a correctly encoded Manchester word containing the proper parity bit. Then one board was placed in the "Test Pattern Generation" mode and the second board in the "Echo" mode. The forward current potentiometers on both boards were adjusted to ensure proper reception of the 16 words transmitted and echoed by the TSTPAT software routine. Once this was accomplished, the plots mentioned above were generated and the results obtained.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Fiber optic implementaton of Mil Std-1553 has been accomplished in this effort in a relatively non-rigorous fashion. The basic concepts, however, have been demonstrated tenaciously. A relatively sound application of the standard has been implemented. Some general conclusions can be drawn from this study.

1. Implementation of Mil Std-1553 using fiber optics in airborne avionics is a reasonable approach.

2. The requisite fiber optic interface is simple and lends itself to many general applications.

3. The physical handling of the fiber optic cable during the course of this study was not overly sensitive--therefore, fiber optic cable installations can survive reasonable human environments. The critical parameter when handling fiber optics is the bending radius. Once this bending radius parameter is exceeded, cracking of the plastic or glass medium sets in and performance begins to degrade.

4. The data rate requirement of Mil Std-1553 (1 MBps) can be met and exceeded using fiber optics.

5. Data security applications requirements can be readily addressed with fiber optics.

6. Fiber optic cables are relatively light and therefore lend themselves to weight-critical applications such as space.

B. RECOMMENDATIONS

As mentioned above, the design tested was a relatively non-rigorous approach. Because it was non-rigorous, the conclusions drawn above are reasonable. Throughout this study, several design alternatives surfaced. They are summarized below.

66

1. The /⁴C8748 was acceptable for the purposes of this effort. It was a limiting factor in a few ways. First, the 8-bit wide data bus required a double read and write to transfer 16-bit words. Second, in the Parallel Standard mode, the buffer storage allocation was less than called for in Mil Std-1553. Third, the program execution time was one of the contributors to a slower data rate.

Recommendation: For higher speed data transfer environments, an alternative 16-bit microcomputer chip with more internal RAM should be selected.

2. A Very Large Scale Integration (VLSI) implementation of this design is well within current technology. The size, weight, and power consumption could be significantly reduced with VLSI technology. More intelligence could be designed in to incorporate all of the functions of Mil Std-1553. This would also facilitate much higher data transfer rates.

Recommendation: Develop and fabricate a custom VLSI implementation of this design using CAD/CAM technology for further study.

3. The fiber optic cable used in this design was single mode plastic core large diameter fiber. Much higher data rates and reliability could be achieved using small diameter glass fibers with laser excitation devices. Fifty megabit data rates are currently being achieved with higher quality fibers and driver/receivers.

Recommendation: For military and industrial applications, the use of higher quality fibers and drivers/receivers is suggested. The cost of these items is steadily falling because production techniques are rapidly improving.

4. The design as presented here is functional and could be used in general Local Area Networks as it exists. The data block size could be easily increased from 32 sixteen-bit words to 256 sixteen-bit words in either the Parallel Bypass mode or the Serial mode.

Recommendation: Consider using this design or a similar design for Local Area Networks environments that are susceptible to RFI and/or EMI.

APPENDIX A

MIL STD 1553 CARD LAYOUT AND DISCUSSION

The integrated circuits and discrete components used in the design were mounted on an INTEL Prototype board in a particular fashion. The intent was to keep the Mil Std-1553 peculiar components on one side of the board to allow expansion and customizing the design for particular applications. As an example, the Built-In-Test (BIT) function was added on one of the Mil Std-1553 cards. Figure Al gives this arrangement. Note that the BIT function components are shown in the dashed area of the figure.

The remaining area might be used to incorporate a First-In-First-Out (FIFO) buffer in a Multibus environment or any of a number of other uses.

The Mil Std-1553 peculiar components were also placed in such a way as to:

1. Minimize the data bus line lengths.

2. Isolate the clocking circuitry from other sensitive components.

3. Reduce much of the locally generated noise interactions.
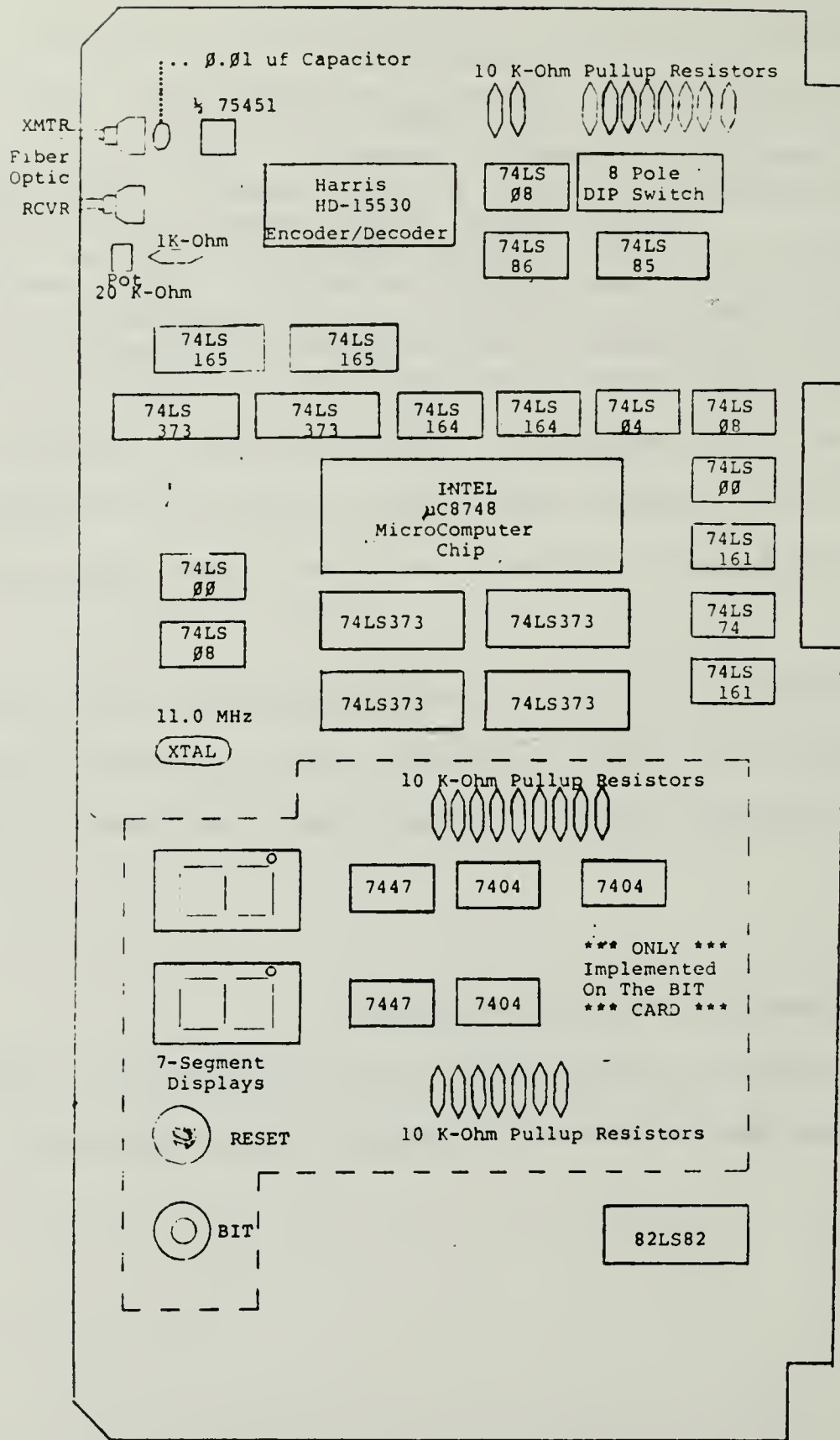
Figure A1. Mil Std-1553 Card Chip Placement

70
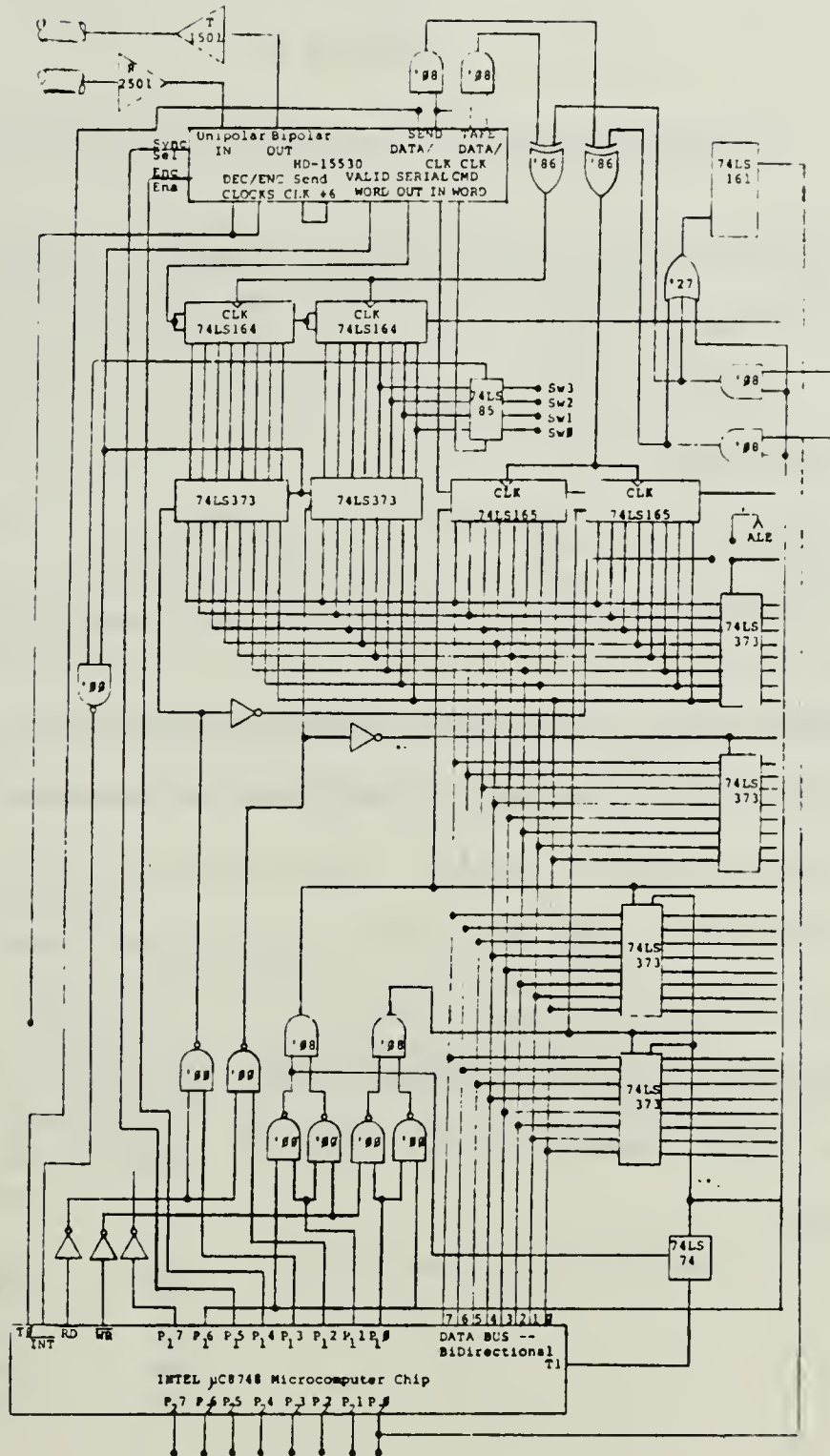
Figure A2. Complete Card Schematic for Mil Std-1553 Card

71

APPENDIX B

SOFTWARE LISTINGS: μC8748 MICROCOMPUTER CHIP

This Appendix contains the complete assembled listings for all of the software routines resident in the on-chip EPROM of the μC8748. It is organized in the following format:

1. A brief synopsis of the routine.

2. A definition for each of the variables used.

3. The beginning and ending hexidecimal addresses for the routine.

The intent here is to permit easy replication of any or all of the modes of operation as desired. If only one or two modes are implemented, it may be appropriate to "re-link" the various routines to provide more cohesive and compact code and room in the program memory for additional software.

As mentioned in the "RECOMMENDATIONS" section, more user-friendly development tools would reduce software buildup and checkout time significantly.

Synopsis:

Entry into this routine is accomplished via a "power-on" restart or "reset". When power is first applied, the μC8748 begins program execution at location "00" hexidecimal. At this location, a jump to the initialization routine was placed. At location 003 hexidecimal, the μC8748 expects to find the external interrupt service routine. A jump to the interrupt service routine was placed here. The μC8748 expects to find the Timer/Counter interrupt service routine at location 007 hexidecimal. The actual interrupt service routine was placed here to generate the test pattern words for the TSTPAT routine.

Variable Definitions:

       INIT   - Title of the initialization routine.
       EXTINT - Location of the external interrupt service routine.
       INTERPT - Title of the interrupt handling routine
       TIMINT - Title of the Timer/Counter interrupt Service Routine
       NXTWD   - Location of the "wait for encoder done" loop.

Routine Linkage:

Name   Begin   End   (addresses are in hexidecimal)

PWRRST: 000   001
EXTINT: 003   004
TIMINT: 007   039

addr	opcode/addr	label	mnemonic	Comments
000	0440	PWRRST	JMP INIT	JMP to INIT Routine
002				
003	4400	EXTINT	JMP INTERPT	JMP to External INIT Routine
005				
007	65	TIMINT	STOP TIMER	Timer Interrupt Routine
008	D5		SEL RB1	Select Register Bank 1
009	B90E		MOV R1',#0E	Load Reg 1 with # of Words
00B	B820		MOV R0',#20	Load Reg 0 With Buffer Addr
00D	9900		ANL P1, #00	Clear Port 1
00F	F0		MOV A, @R0'	Get Low Byte of Command Word
010	8901		ORL P1, #01	Output it to LSH Shift Reg
012	02		OUTL Bus,A	Strobe Low Byte Latch
013	9900		ANL P1, #00	Clear Port 1
015	18		INC R0'	Increment Buffer Pointer
016	F0		MOV A, @ R0'	Get High Byte of CMD Word
017	8902		ORL P1, #02	Output It to MSH Shift Reg
019	02		OUTL Bus, A	Strobe High Byte Latch
01A	9900		ANL P1, #00	Clear P1
01C	8930		ORL P1, #30	Enable Encoder With CMD Sync
01E	18		INC R0'	Increment Buffer Pointer
01F	9900		ANL P1, #00	Clear Port 1
021	3621	NEXTWD	JT0 NEXTWD	Wait For Encoder Done
023	F0		MOV A, @ R0'	Get Low Byte of NXT Data Word
024	8901		ORL P1, #01	Output it to LSH Shift Reg
026	02		OUTL BUS,A	Strobe Low Byte Latch
027	9900		ANL P1, #00	Clear Port 1
029	18		INC R0'	Increment Buffer Pointer
02A	F0		MOV A, @ R0'	Get High Byte of Nxt Data WD
02B	8902		ORL P1, #02	Output it to MSH Shift Reg
020	02		OUTL Bus,A0	Strobe High Byte Latch
02E	9900		ANL P1, #00	Clear Port 1
030	8910		ORL P1, #10	Enable Encoder w/Data Sync
032	18		INC R0'	Increment buffr Pointer Addr
033	9900		ANL P1, #00	Clear Port 1
035	E921		DJNZ R1',NEXTWD	Dec Wd CNTR & Jmp if notzero
037	C5		SEL RB0	Select Register Bank 0
038	55		STRT T	Start Timer
039	93		RETR	Return frm interrupt-restore

Synopsis:

This is the initialization routine. Entry is made via the PWRRST routine. INIT performs several setup functions. It clears the interrupts, ports, the accumulator, and the program status word. It loads the memory with all zeros and sets up the Timer/Counter for the time-out interrupts. It also clears the carry flag as well as flags "FØ" and "Fl".

Variable Definitions:

TNCTI   - The Timer Counter Interrupt enable/disable operand.
INITl   - A label for the "Clear Memory" loop.
MDTST   - A label where the mode flags are initialized.
RESTART - Title for the Mode selection software routine.

Program Linkage:

Name	Begin	End	(hex)
INIT	040	05C	

addr	opcode/addr	label	mnemonic	comments
040	15	INIT	DIS I	Disable Interrupts
041	35		DIS TNCTI	Disable T&C interrupts
042	9900		ANL P1, #00	Clear Port 1
044	8980		ORL P1,#80	Clear Int F.F.
046	9900		ANL P1,#00	Clear Port 1
048	27		CLR A	Zero Accumulator
049	D7		MOV PSW,A	Zero P.S.W.
04A	B83E		MOV R0,#3E	Top of Memory R0
04C	B93E		MOV R1,#3E	
04E	B100	INIT1	MOV @R1,#00	Zero all memory
050	C9		DEC R1	
051	E84E		DJNZ R0,INIT1	Not Done: Continue
053	23A0		MOV A,#A0	Set Up Timer Constant
055	62		MOV T,A	Load Time Constant
056	55		STRT TIMER	Start Timer
	15	MDTST	DIS I	
058	97		CLR C	Clear Carry Flag
059	A5		CLR F0	Clear Flag 0
05A	85		CLR F1	Clear Flag 1
05B	0477		JMP RESTART	Jump to Restart Routine

Synopsis:

RESTART performs the mode selection function based on the particular DIP switch settings selected. It is entered via the INIT routine. All of the possible modes available are entered via the RESTART routine. On the listing under the "NOTES" heading is a table that shows the relationship between the DIP switch positions and the modes they select. The two flags "F0" and "F1" are used by the interrupt service routine (INTERPT) to sense the current mode in which to handle the interrupts.

Variable Definitions:

```
RESTART    - Beginning of the RESTART routine.
TSTPAT     - Label for the test pattern generation routine.
MIDBIT     - A test loop label to check the state of the middle
             bit of the switch.
LASTBIT    - A test loop label for the last bit checking
             function.
SERMD      - A label for the jump to the Serial Mode routine.
ParStdMD   - "    "    "    "    "    "    "  Parallel Standard
             mode routine.
ECHOMD     - A label for the jump to the Echo mode routine.
PBPMD      - "    "    "    "    "    "    "  Parallel Bypass mode
             routine.
SERMODE    - The label for the Serial mode routine.
PARSTD     - "      "      "      "  Parallel Standard mode routine.
ECHOMODE   - "      "      "      "  Echo Mode routine.
PARBYPASS  - "      "      "      "  Parallel Bypass routine.
```

Program Linkage:

Name	Begin	End	(hex)
RESTART	077	097	

addr	opcode/addr	label	mnemonic	comments
077	OA	RESTART	IN A,P2	Read Mode Switch
078	5370		ANL A,#70	Mask Out-except Swtchs
07A	C660		JZ TSTPAT	000=>Test Pattern Mode
07C	D280		JB6 MIDBIT	If XX1=>Check Mid Bit
07E	0477		JMP RESTART	If XX0=>Invalid Mode
080	B286	MIDBIT	JB5 LASTBIT	If X11=>Check Last Bit
082	928A		JB4 SERMD	If 101=>Serial Mode
084	048D		JMP ParStdMD	If 001=>Parallel Std Md
086	9291	LASTBIT	JB4 ECHOMD	If 111=>ECHO MODE
088	0495		JMP PBPMD	If 011=>Parallel BP Md
08A	B5	SERMD	CPL F1	Set flags to :10
08B	2440		JMP SERMODE	Jump to Serial Mode
08D	85	ParStdMD	CLR F0	Set flags = 00
08E	A5		CLR F1	
08F	2480		JMP PARSTD	Jp to Parallel Std Md
091	95	ECHOMD	CPL F0	Set flags = 11
092	B5		CPL F1	
093	04A0		JMP ECHOMODE	Jump to Echo Mode
095	95	PBPMD	CPL F0	Set flags = 01
096	2400		JMP PARBYPASS	Jp to Parallel BP Md

NOTES: MODE SWITCH Settings with associated flag settings;

0 = ON
1 = OFF

S7	S6	S5	MODE	flags:	F1	F0
0	0	0	Test Pattern		X	X
0	0	1	Parallel Standard		0	0
0	1	0	Invalid		X	X
0	1	1	Parallel By Pass		0	1
1	0	0	Invalid		X	X
1	0	1	Serial		1	0
1	1	0	Invalid		X	X
1	1	1	Echo		1	1

Synopsis:

The test pattern generation process is initialized by the TSTPAT routine. The buffer that contains the test pattern words is loaded with the desired pattern. This buffer is 16-bytes long and begins at location 20 hex. Once this initialization is done, the routine enters a "wait" mode in wich it outputs an 8-bit counter to the most significant half of the parallel port for checkout purposes. Entry of this mode is made via the RESTART routine and switch selection.

Variable Definitions:

    TSTPAT   - Beginning of the test pattern generation routine.
    TSTPAT1 - Label for the buffer initialization loop.
    TSTPATWT - Label for the dummy loop waiting for a "TIME-OUT"
                 interrupt.
    RESTART - Label for the "RESTART" routine.

Program Linkage:

Name	Begin	End	(hex)
TSTAT	060	076	

addr	opcode/addr	label	mnemonic	comments
060	B90F	TSTPAT	MOV R1, #0F	0F hex = 16 Decimal Bytes
062	B820		MOV R0, #20	Buffer Starts at 20 hex
064	B005	TSTPAT1	MOV @R0,#05	TSTPAT Alternating 1's & 0's
066	18		INC R0	
067	B00E		MOV @R0, #0E	
069	18		INC R0	
06A	E964		DJNZ R1,TSTPAT1	Continue Until Done
06C	25		EN TNCTI	Enable T & C
06D	17	TSTPATWT	INC A	
06E	C677		JZ RESTART	
070	02		OUTL BUS,A	
071	8908		ORL P1, #08	
073	9900		ANL P1,#00	
075	046D		JMP TSTPATWT	

Synopsis:

The ECHOMODE routine is intended primarily for use in the testing and checkout phases, but could be used as a down-link signal booster in a long haul fiber optic link. Its function basically is to detect all data addressed to it and re-transmit the data back to the originating terminal. Functionally, it receives the Command Word and swaps the Terminal Address field with the Sub-address field and retransmits. The word count is used to echo all data words associated with a Command Word.

Variable Definitions:

```
ECHOMODE  - Beginning of the "ECHOMODE" routine.
WaitMode  - Return label from MDTST routine.
OUTCNT    - Dummy loop label pending reception of a word to echo.
ECHOINT   - Entry label for Received Word interrupt.
NRDY      - Encoder ready wait loop label.
NOTDONE   - Label for data word echos.
NRDY2     - Data Word Encoder ready wait loop label.
RETURN    - Label for return from ECHOINT.
```

Program Linkage:

Name	Begin	End	(hex)
ECHOMODE	ØAØ	ØB3	
ECHOINT	ØB6	ØFC	

addr	opcode/addr	label	mnemonic	comments
0A0	05	ECHOMODE	ENI	Enable Interrupts
0A1	35		DIS TNCTI	Disable T/C Int.
0A2	9900		ANL P1, #00	Clear Port 1
0A4	8980		ORL P1, #80	Clear Flip Flops
0A6	9900		ANL P1, #00	Clear Port 1
0A8	05		ENI	Enable Ints.
0A9	27	WaitMode	CLR A	
0AA	17	OUTCNT	INC A	
0AB	C657		JZ MDTST	
0AD	02		OUTL BUS,A	
0AE	8908		ORL P1,#08	
0B0	9900		ANL P1,#00	
0B2	04AA		JMP OUTCNT	
0B4			NOP-NOP	
0B6	15	ECHOINT	DIS I	Disable Ints.
0B7	8904		ORL P1, #04	Read LSH - CMD Wd
0B9	08		INS A, BUS	
0BA	531F		ANL A, 1F	Mask Out Word Cnt
0BC	AA		MOV R2, A	Save Wd Cnt in R2
0BD	9900		ANL P1, #00	Clear Port 1
0BF	8902		ORL P1, #02	Output LSH Enable
0C1	02		OUTL Bus, A	
0C2	9900		ANL P1, #00	Clear Port 1
0C4	8908		ORL P1, #08	Get MSB of CMD WD
0C6	08		INS A, BUS	
0C7	9900		ANL P1, 00	Clear Port 1
0C9	8901		ORL P1, #01	Set Up For MSH Xfer
0CB	02		OUTL BUS, A	Output MSH - CMD Wd
0CC	9900		ANL P1, 00	Clear Port 1
0CE	8930		ORL P1, #30	Ena ENCDR w/CMD SYNC
0D0	9900		ANL P1, #00	Clear Port 1
0D2	36D2	NRDY	JT0, NRDY	Wait - ENCDR Complete
0D4	EAD8		DJNZ R2 NOTDONE	Done Ret. to Begin
OD6	04F6		JMP RETURN	
0D8	56D8	NOTDONE	JT1, NOTDONE	Wait for Decoder
0DA	8904		ORL P1, #04	Read LSH - NXT Data Wd
0DC	08		INS A, Bus	
0DD	9900		ANL P1, #00	Clear Port 1
0DF	8902		ORL P1, #02	Xfer LSH - Nxt Data Wd
0E1	02		OUTL BUS, A	
0E2	9900		ANL P1, #00	Clear Port 1
0E4	8908		ORL P1, #08	Read MSB - Nxt Data Wd
0E6	08		INS A, BUS	
0E7	9900		ANL P1, #00	Clear Port 1
0E9	8901		ORL P1, #01	Xfer MSH - NXT Data Wd
0EB	02		OUTL Bus, A	

```
0EC     9900               ANL, P1, #00        Clear Port 1
0EE     8910               ORL P1, #10         Ena ENCDR  W/Data Sync
0F0     9900               ANL P1, #00         Clear Port 1
0F2     36F2    NRDY2      JT0, NRDY2          Wait - ENCDR Complete
0F4     EAD8               DJNZ R2, NOTDONE    Not Done, Wait -NXT WD
0F6     8980    RETURN     ORL P1, #80         Clear Flip Flops
0F8     9900               ANL P1, #00         Clear Port 1
0FA     C5                 SELRB0
0FB     93                 RETR                Ret. frm  interrupt
                                               w/Restore
```

Synopsis:

Parallel data is handled with the PARBYPASS routine in a "Bypass" mode. The data path is shown in Figure 12. This routine basically performs a "traffic cop" function. The first word latched on the parallel input port is always a command word containing the Terminal Address and the word count. The word count is read by the /ʎC8748 and used to manage the data transfers.

Variable Definitions:

    PARBYPASS - Entry point for the PARBYPASS routine.
    STBRD   - Wait loop pending Command Word ready on the parallel
                  input port.
    PBPNRDY   - Wait loop pending completion of a Command Word transfer.
    CONTINUE   - Wait loop pending data word ready on parallel port.
    ENCNRDY   - Wait loop pending completion of a data word transfer.

Program Linkage:

Name	Begin	End
PARBYPASS	100	136

addr	opcode/addr	label	mnemonic	comments
100	05	PARBYPASS	ENI	Enable Interrupts
101	35		DIS TNCTI	Disable T/C Ints
102	9900		ANL P1,#00	Clear Port 1
104	0A	STBRD	IN A,P2	Input Stb tst - RDY?
105	97		CLR C	
106	F7		RLC A	Rotate it thru Carry
107	E604		JNC STBRD	Strobe not ready
109	8941		ORL P1,#41	Read LSH of CMD Word
10B	08		INS A, BUS	
10C	9900		ANL P1,#00	Clear Port 1
10E	531F		ANL A,#1F	Mark Out Word Count
110	A9		MOV R1, A	Store Wd Count - R1
111	8942		ORL P1, #42	Latch MSH of CMD Wd
113	9900		ANL O1,#00	Clear Port 1
115	8930		ORL P1,#30	Ena ENCDRw/CMD Sync
117	9900		ANL P1,#00	Clear Port 1
119	3619	PBPNRDY	JT0, PBPNRDY	Wait - ENCDR Compl
11B	00		NOP	
11C	E920		DJNZ R1,CONTINUE	Not Done,Jp-CONTINUE
11E	2400		JMP PARBYPASS	Otherwise, - Nxt Seq
120	0A	CONTINUE	IN A, P2	Read Strobe
121	97		CLRC	
122	F7		RLC A	Test for Ready
123	E620		JNC,CONTINUE	Not Ready, Return
125	8940		ORL P1,#41	Latch LSH - Nxt Wd
127	9900		ANL P1,#00	Clear Port 1
129	8942		ORL P1,#42	Latch MSH-Nxt Data Wd
12B	9900		ANL P1,#00	Clear Port 1
12D	8910		ORL P1,#10	Ena ENCDR w/Data Sync
12F	9900		ANL P1,#00	Clear Port 1
131	3631	ENCNRDY	JT0,ENCNRDY	Wait - ENCDR compl
133	E920		DJNZ R1,CONTINUE	Not Done,- nxt Data Wd
135	2400		JMP PARBYPASS	DONE, RETURN - Nxt Seq

Synopsis:

Serial data transfers are managed by the SERMODE routine. The word count for the serial data must be read on the least significant half of the parallel port. The word count is read by the ⁄ᵁC8748 and used to manage the serial transfers. Serial data is clocked directly into the shift registers for the HD15530 Encoder. The serial transfer process is managed by the ⁄ᵁC8748.

Variable Definitions:

    SERMODE - Entry point for the Serial Mode of operation.
    WAITSTB - Wait loop pending latching of the word count on the
               parallel port.
    CNTRSTB - Wait loop pending clocking of 16 bits of serial Command
               Word data.
    ENCWAIT - Wait loop pending Encoder Complete state for the Command
               Word
    NXTGTSTB - Wait loop pending clocking of 16 bits of serial Data
               Word data.
    ENCWAIT1 - Wait loop pending Encoder Complete state for the Data
               Word data.

Program Linkage:

Name   Begin   End   (hex)

SERMODE   140   179

addr	opcode/addr	label	mnemonic	comments
140	05	SERMODE	En I	Enable Interrupts
141	35		DIS TNCTI	Disable T/C Ints.
142	9900		ANL P1,#00	Clear Port 1
144	0A	WAITSTB	IN A, P2	Input Stb for test
145	97		CLR C	Clear Carry
146	F7		RLC A	Test Strobe
147	E644		JNC, WAITSTB	Not Ready, tst agn
149	8901		ORL P1,#01	Input(Wd -Par Port)
14B	08		INS A, BUS	
14C	9900		ANL P1,#00	Clear Port 1
14E	531F		ANL A,#1F	MASK Word Count
150	A9		MOV R1,A	Store WdCNT - Reg1
151	8940		ORL P1,#40	Ena Serial IN Gate
153	0A	CNTRSTB	IN A, P2	Get Stb for Countr
154	F7		RLC A	Rotate thru Carry
155	97		CLR C	Clear Carry
156	F7		RLC A	Rot Stb thru Carry
157	E653		JNC, CNTRSTB	Wait - Strobe Rdy
159	9900		ANL P1,#00	Clear P1/dis S gte
15B	8930		ORL P1,#30	Ena ENCDR/CMD Sync
15D	9900		ANL P1,#00	
15F	365F	ENCWAIT	JT0, ENCWAIT	Wait - ENCDR Compl
161	E966		DJNZ R1,NXTGSTB	NOT DONE, continue
163	00		NOP	
164	2440		JMP SerMODE	Done, ret -nxt Seq
166	8940	NXTGSTB	ORL P1,#40	Ena Serial IN Gate
168	0A		IN A, P2	Test - Countr Rdy?
169	F7		RLC, A	Rotate thru Carry
16A	97		CLR C	Clear Carry
16B	F7		RLC, A	Rot.Stb thru Carry
16C	E666		JNC,NXTGSTB	Wait - Strobe Rdy
16E	9900		ANL P1,#00	Clear P1/Dis S gte
170	8910		ORL P1,#10	Ena ENCDR-Data Sync
172	9900		ANL P1,#00	Clear Port 1
174	3674	ENCWAIT1	JT0,ENCWAIT1	Wait - ENCDR Compl?
176	E966		DJNZ R1,NXTGSTB	Not Done, Continue
178	2440		JMP SerMODE	DONE, get NXT Seq

Synopsis:

Parallel input data is buffered by the PARSTD routine. The function of this routine is to provide data buffering to allow slower devices to interface to the fiber optic link. Parallel input data is read into the ΜC8748 buffer beginning at 20 hexidecimal. Up to 18 16-bit words can be buffered.

Variable Definitions:

    PARSTD - Entry point for the parallel standard routine.
    PARSTB - Wait loop pending parallel Command Word ready.
    DATAIN - Wait loop pending parallel Data Words ready.
    OUTPUT - Routine label for the Output routine.

Program Linkage:

Name	Begin	End	(hex)
PARSTD	180	1BB	
OUTPUT	1C0	1F1	

addr	opcode/addr	label	mnemonic	comments
180	0535	PARSTD	EnI	Enable ints.
182	B91A		MOV R1,#1A	Init Buf Pntr.
184	9900		ANL P1,#00	Clear Port 1
186	0A	PARSTB	IN A, P2	Input Stb for test
187	97		CLR C	Clear Carry flag
188	F7		RLC A	Rotate Parallel Stb
189	E686		JNC PARSTB	NOT CARRY, tst again
18B	8941		ORL P1,#41	Input LSH of CMD WD
18D	08		INS A, BUS	
18E	9900		ANL P1,#00	Clear Port 1
190	A1		MOV @R1, A	Save LSH of CMD WD
191	19		INC R1	Incrmt Buf Pointer
192	5313		ANL A,#13	Mask Off Wd Count
194	AA		MOV R2, A	Save Wd Cnt in Reg2
195	AB		MOV R3, A	Save Wd Cnt in Reg3
196	8942		ORL P1,#42	Input MSH of CMD WD
198	08		INS A, BUS	
199	9900		ANL P1,#00	Clear Port 1
19B	A1		MOV @R1, A	Save MSH of CMD Wd
19C	19		INC R1	Incrmt buf pointer
19D	EAA3		DJNZR2,DATAIN	If not done cont.
19F	B91A		MOV R1,#1A	Reinit Buf Pointer
1A1	24C0		JMP Output	IF DONE jp - OUTPUT
1A3	0A	DATAIN	IN A, P2	Input stb. for test
1A4	97		CLR C	Clear Carry Flag
1A5	F7		RLC A	Rotate Stb. thru Carry
1A6	E6A3		JNC DATAIN	Jump, if not carry
1A8	8941		ORL P1,#41	Input LSH - NXT Data Wd
1AA	08		INS A, BUS	
1AB	A1		MOV @R1, A	Save LSH - Nxt Data Wd in Buffer
1AC	19		INC R1	Incrmt Buf Pointer
1AD	9900		ANL P1,#00	Clear Port 1
1AF	8942		ORL P1,#42	Input MSH - Nxt Data Wd
1B1	08		INS A, BUS	
1B2	A1		MOV @R1, A	Save MSH - Nxt Data Wd
1B3	19		INC R1	Incrmt Buf Pointer
1B4	9900		ANL P1,#00	Clear Port 1
1B6	EAA3		DJNZ R2,DATAIN	NOT DONE, Continue
1B8	B91A		MOV R1,#1A	Restore Buf Pointer
1BA	24C0		JMP OUTPUT	Jmp to OUTPUT Routine

Synopsis:

Parallel input data buffered by the PARSTD routine is output to the HD15530 shift registers. Register R3 is used to pass the word count parameter from the PARSTD routine to the OUTPUT routine.

Variable Definitions:

> OUTPUT - Entry point for the OUTPUT routine.
> OUTSTB - Wait loop pending Encoder Complete state for the Command Word transfer.
> DATAOUT - Label for the parallel data Output routine.
> DATASTB - Wait loop pending Encoder Complete state for the data word transfers.
> PARSTD - Return label for the parallel standard routine.

Program Linkage:

Name	Begin	End
OUTPUT	1CØ	1F1
PARSTD	18Ø	1BB

addr	opcode/addr	label	mnemonic	comments
1C0	36C0	OUTPUT	JT0, OUTPUT	Wait for ENCDR Rdy
1C2	F1		MOV A,@R1	Ld A w/LSH of CMD Wd
1C3	8901		ORL P1,#01	Ena LSH S / Register
1C5	02		OUTL BUS, A	Out LSH-CMDWD to S/R
1C6	9900		ANL P1,#00	Clear Port 1
1C8	19		INC R1	Incrmt Buf Pointer
1C9	F1		MOV A,@R1	Ld A w/MSH of CMD Wd
1CA	8902		ORL P1,#02	Ena MSH S / Register
1CC	02		OUTL BUS, A	Out MSH-CMDWD to S/R
1CD	9900		ANL P1,#00	Clear Port P1
1CF	19		INC R1	Incrmt Buf Pointer
1D0	8930		ORL P1,#30	Ena ENCDR W/CMD Sync
1D2	9900		ANL P1,#00	Clear Port 1
1D4	36D4	OUTSTB	JT0 OUTSTB	Wait - ENCDR Compl
1D6	EBDA		DJNZ R3,DATAOUT	NOT DONE, Continue
1D8	2480		JMP PARSTD	IF DONE, JP to begin
1DA	F1	DATAOUT	MOV A,@R1	Get LSH - Nxt Data Wd
1DB	8901		ORL P1,#01	Ena LSH S / Register
1DD	02		OUTL BUS,A	Out LSH - Nxt Data Wd
1DE	9900		ANL P1,#00	Clear Port 1
1E0	19		INC R1	Incrmt Buf Pointer
1E1	F1		MOV A,@R1	Get MSH - Nxt Data Wd
1E2	8902		ORL P1,#02	Ena MSH S / Register
1E4	02		OUTL BUS,A	Out MSH- Nxt Data Wd
1E5	9900		ANL P1,#00	Clear Port 1
1E7	19		INC R1	Incmt Buffer Pointer
1E8	8910		ORL P1,#10	Ena ENCDR w/Data Sync
1EA	9900		ANL P1,#00	Clear Port 1
1EC	36EC	DATASTB	JT0,DATASTB	Wait - ENCDR Compl
1EE	EBDA		DJNZ R3,DATAOUT	NOT DONE, Get Nxt Wd
1F0	2480		JMP PARSTD	IF DONE, JMP to Begin

Synopsis:

All interrupt service routines originate in the INTERPT routine.
Flags "F0" and "F1" are set in the RESTART routine which indicate the
current mode in which the interrupt will be serviced. The table included
in the RESTART listing indicates this relationship. After processing the
appropriate interrupt, the return from interrupt is made via the RESTORE
segment which clears port 0 and returns to the interrupted routine via
popping the saved program counter from the stack and continuing that
routine at the current program counter. Entry is made via the jump to
INTERPT at location 003 hexidecimal.

Variable Definitions:

        INTERPT   - Entry point for all external interrupts.
        TSTFLG1   - Conditional jump based on Flag "F0" being set.
        INTSER   - Serial Interrupt Service Routine label.
        INTPARSTD   - Parallel Standard Interrupt Service Routine label.
        ECHOINT   - Echo mode Interrupt Service Routine label.
        INTPBB   - Parallel Bypass Interrupt Service Routine label.
        RESTORE   - Return from Interrupt routine label.

Program Linkage:

Name	Begin	End	(hex)
INTERPT	200	236	
SERINT	240	26A	
ECHOINT	0B6	0FC	
PARSTDINT	270	2AE	
PBPINT	2B0	2CB	

92

addr	opcode/addr	label	mnemonic	comments
200	15	INTERPT	DIS I	Disable Ints
201	9900		ANL P1,#00	Clear Port 1
203	8980		ORL P1,#80	Clear Flip Flops
205	9900		ANL P1,#00	Clear Port 1
207	B60D		JF0,TSTFLG1	If F0 Set, TST FLG1
209	7615		JF1,INTSER	If F1 Set,JP-SERINT
20B	4425		JMP,INTPARSTD	No Flags, JP-Parallel
20D	7620	STFLG1	JF1,ECHO-INT	Both Flags,JP to Echo
20F	442A		JMP PBPINT	F1 Set,JP to Parallel
				ByPass
215	D5	INTSER	SEL RB1	
216	4940		JMP SERINT	
220	D5		SEL RB1	
221	04B6	ECHO-INT	JMP ECHOINT	Jump to Echo Mode
225	D5	INTPARSTD	SEL RB1	
226	4470		JMP PARSTDINT	
22A	D5	INTPBP	SEL RB1	Sel Register Bank 1
22B	44B0		JMP PBPINT	
230	C5		SEL RB0	
231	9900	RESTORE	ANL P1,#00	Clear Port 1
234	05		EI	Enable Interrupts
235	93		RETR	RET.frm Ints-restore

Synopsis:

Serial data interrupts are serviced by the SERINT routine. The sequence executed here is basically the complement of that in the SERMODE routine. An interrupt is received upon receipt of a Command Word. Serial Data is received and clocked out on the Serial Output.

Variable Definitions:

    SERINT - Interrupt Service Routine entry point.
    CNTRWT - Loop to detect when 16-bits of serial data have been
              clocked out.
    NXTSWD - Loop to wait for decoder complete state.
    DATACNT - Loop to output data words serially.

Program Linkage:

Name	Begin	End
SERINT	240	26A
RESTORE	230	235

addr	opcode/addr	label	mnemonic	comments
240	8904	SERINT	ORL P1,#04	Enable LSH Latch O.E.
242	08		INS A,BUS	Read LSH
243	9900		ANL P1,#00	Clear Port 1
245	531F		ANL A,#1F	Mask Out Word Count
247	A9		MOV R1,A	Store it in Register R1
248	8904		ORL P1,#04	Enable LSH of Parallel Port
24A	02		OUTL BUS,A	Ouput it
24B	9900		ANL P1,#00	Clear Port 1
24D	8940		ORL P1,#40	Enable S.O. gate
24F	0A	CNTRWT	IN A,P2	Check for Counter Done
250	F7		RLC A	
251	97		CLR C	Clear Carry for Test
252	F7		RLC A	Test for done thru Carry Bit
253	E64F		JNC CNTRWT	If not set, go test again
255	9900		ANL P1,#00	Clear Port 1, and S.O. gate
257	E95B		DJNZ R1,NXTSWD	Decrement Word Counter; Done?
259	4430		JMP RESTORE	If done, Return with Restore
25B	365B	NXTSWD	JT0 NXTSWD	Wait for Decdr & Next Data Word
25D	8940		ORL P1,#40	Enable S.O. gate
25F	0A	DATACNT	IN A,P2	Check for Counter Done
260	F7		RLC A	
261	97		CLR C	
262	F7		RLC A	Test for done thru Carry Bit
263	E65F		JNC DATACNT	If not done; wait some more
265	9900		ANL P1,#00	Clear Port 1; and S.O. gate
267	E95B		DJNZ R1,NXTSWD	If not done, get next word
269	4430		JMP RESTORE	If done, Return with Restore

Synopsis:

Parallel data buffering occurs with the PARSTDINT interrupt service routine. The data is transferred to an internal μC8748 buffer. Once all data is buffered, this routine transfers the data to the parallel ports for output.

Variable Definitions:

    PARSTDINT - Entry point for the interrupt service routine in the
                parallel standard mode.
    DECWT   - Wait loop pending the Decoder Complete State.
    OUTDATA   - Output loop label for parallel output of the data.
    RESTORE   - Label for the return from interrupt routine.

Program Linkage:

Name	Begin	End	(hex)
PARSTDINT	270	2AE	
RESTORE	230	235	

-

addr	opcode/addr	label	mnemonic	comments
270	B91A	PARSTDINT	MOV R1,#1A	Init Buffer Pointer
272	8904		ORL P1,#04	Read LSH of CMD word
274	08		INS A, BUS	
275	9900		ANL P1,#00	Clear Port 1
277	5313		ANL A,#13	Mask Out Word Count
279	AA		MOV R2, A	Save Word Count in R2
27A	AB		MOV R3, A	Save Word Count in R3
27B	A1		MOV @R1,A	Save Wd Count in Buf
27C	19		INC R1	Incrmt Buffer Pointer
27D	8908		ORL P1,#08	Read MSH of CMD word
27F	9900		ANL P1,#00	Clear Port 1
281	A1		MOV @R1,A	Save MSH of CMD word
282	19		INC R1	Incrmt Buffer Pointer
283	EA89		DJNZ R2,DECWT	Test for done
285	B91A		MOV R1,#1A	
287	449D		JMP OUTDATA	Done,jp-OUTDATA rout.
289	5689	DECWT	JT1 DECWT	Wait for Decoder rdy
28B	8904		ORL P1,#04	Get LSH of nxt data wd
28D	08		INS A, BUS	
28E	9900		ANL P1,#00	Clear Port 1
290	A1		MOV @R1,A	Save LSH of data word
291	19		INC R1	Incrmt Buffer Pointer
292	8908		ORL P1,#08	Get MSH - nxt data wd
294	08		INS A, BUS	
295	9900		ANL P1,#00	Clear Port 1
297	A1		MOV @R1,A	Save MSH of data word
298	19		INC R1	Incrmt Buffer Pointer
299	EA89		DJNZ R2,DECWT	Done? if not,- nxt wd
29B	B91A		MOV R1,#1A	Done, reinit Buf Ptr
29D	F1	OUTDATA	MOV A, @R1	Get LSH of wd frm Buf
29E	19		INR R1	Incrmt Buffer Pointer
29F	8904		ORL P1,#04	Ena LSH-Parallel Port
2A1	02		OUTL BUS,A	Output LSH-wd to Port
2A2	9900		ANL P1,#00	Clear Port 1
2A4	F1		MOV A,@R1	Get MSH of wd frm Buf
2A5	19		INR R1	Incrmt Buffer Pointer
2A6	8908		ORL P1,#08	Ena MSH-Parallel Port
2A8	02		OUTL BUS,A	Output MSH-wd to Port
2A9	9900		ANL P1,#00	Clear Port 1
2AB	EB9D		DJNZ R3,OUTDATA	Not done, get nxt wd
2AD	4430		JMP RESTORE	Done, Return w/Restor

Synopsis:

Parallel data received for the parallel bypass mode is handled via the PBPINT interrupt service routine. The interrupt sequence is initiated upon receipt of a Command Word. Data is transferred directly to the parallel output ports bypassing the μC8748 internal buffer. Entry is made via the flag relationship given in the RESTART routine.

Variable Definitions:

    PBPINT - Entry point for the parallel bypass interrupt service
              routine.
    CONTPBP - Data word input routine.

Program Linkage:

Name	Begin	End
PBPINT	2B0	2CB
RESTORE	230	235

addr	opcode/addr	label	mnemonic	comments
2B0	8904	PBPINT	ORL P1,#04	Enable LSH Latch
2B2	08		INS A,BUS	Read LSH of CMD Word
2B3	9900		ANL P1,#00	Clear Port 1
2B5	531F		ANL A,#1F	Mask out word count
2B7	A9		MOV R1,A	Save Wd count in R1
2B8	8908		ORL P1,#08	Read MSH of CMD Word
2BA	E9BE		DJNZ R1,CONTPBP	Not done, continue
2BC	4430		JMP RESTORE	Done,Return w/Restor
2BE	9900	CONTPBP	ANL P1,#00	Clear Port 1
2C0	56BE		JT1 CONTPBP	Wait DECDR & nxt wd
2C2	8904		ORL P1,#04	Read LSH of nxt wd
2C4	9900		ANL P1,#00	Clear Port 1
2C6	8908		ORL P1,#08	Read MSH of nxt wd
2C8	E9BE		DJNZ R1,CONTPBP	Not done, continue
2CA	4430		JMP RESTORE	Done,Ret. w/Restore

BUILT-IN-TEST (BIT)

SOFTWARE DISCUSSION

The Built-In-Test function is accomplished via this routine. The purpose of this routine is to provide a means to easily demonstrate the functionality of a complete optical fiber link. This function was used to obtain the results given in Chapter V.

The BIT function could also be used to obtain a preliminary Bit Error Rate. A means of data logging the output over enough samples to calculate a reasonable Bit error count automatically would be required.

The placement of this routine was in high memory (300 hex) and is activated by the DIP switches indicated in Table 1 of Chapter III.

Synopsis:

The Built-In-Test Function (BIT) routine provides a means of validation of an optical fiber link. Functionally, the BIT routine reads a pseudo-random byte from the free running Timer and concatenates it with the appropriate Terminal Address field to form a Command Word. The routine outputs this word and then waits for it to be echoed back. Upon receipt of the echoed word, the routine compares the echoed word to the original word and displays the number of errors encountered as well as the running word count. It then returns to output another pseudo-random Command Word.

Variable Definitions:

    BIT   - Entry point for the Built-In-Test routine.
    WAIT   - Wait for "GO" from momentary pushbutton.
    CONTINUE - Begin Psuedo-Random Command Word Output.
    ENCWAIT   - Wait for Encoder to Complete.
    INTWAIT   - Wait for Echoed Word.
    ERRCNT   - Update the running error count.
    NOERR   - No errors encountered, output word count.
    MAXERR   - "FF" errors encountered, output current word count.
    MAXWDCNT - "FF" words tranferred without errors.

Program Linkage:

Name   Begin   End

BIT   300   35A

addr	opcode/addr	label	mnemonic	comments
300	2797	BIT	CLR A, CLR C	Clr Carry & ACC
302	A8		MOV R0, A	
303	A9		MOV R1, A	WORDCOUNT
304	AA		MOV R2, A	Running Error Cnt
305	BB0E		MOV R3, #0E	Term Addr
307	23A0		MOV A, #A0	
309	62		MOV T, A	Get pseudo-random
30A	55		STRT TIMER	byte
30B	0A	WAIT	IN A, P2	Wait for Momentary
30C	F210		JB7, CONTINUE	Switch to be De-
30E	6400		JMP WAIT	pressed
310	42	CONTINUE	MOV A, T	
311	A8		MOV R0, A	
312	8901		ORLP1, #01	
314	02		OUTL BUS, A	
315	9900		ANL P1, #00	
317	FB		MOV A, R3	
318	8902		ORL P1, #02	
31A	02		OUTL BUS, A	
31B	9900		ANL P1, #00	
31D	8930		ORL P1, #30	Output Pseudo-
				Random CMD WD
31F	9900		ANL P1, #00	
321	3621	ENCWAIT	JT0 ENCWAIT	Wait - ENCDR Com-
323	65		STOP TIMER	plete
324	55		START TIMER	
325	25	INTWAIT	EN TCNTI	
326	8625		JN 1 INTWAIT	Wait for Echoed
328	65		STOP TIMER	WORD
329	8908		ORL P1, #08	
32B	08		INS A, BUS	
32C	D8		XDR A, R0	Compare Echoed
				with original
32D	C640		JZ NOERR	
32F	FA	ERRCNT	MOV A, R2	
330	97		CLR C	
331	17		INC A	
332	F650		JC MAXERR	Disply Wd Cnt
334	AA		MOV R2, A	with dots.
335	F9		MOV A, R1	
336	97		CLR C	
337	17		INC A	
338	F660		JC MAXWDCNT	Disply Err Cnt
33A	A9		MOV R1, A	with dots

```
33B    6410                   JMP CONTINUE      Go back for NXT
340    3597      NO ERR       DIS TCNTI, CLR C
342    F9                     MOV A, R1
343    17                     INC A
344    F660                   JC MAXWDCNT       Exceeded Word -
346    A9                     MOV R1, A         count? If not -
                                                Update running
347    02                     OUTL BUS, A       Word Count total
348    8980                   ORL P1, #80       Output Counter
34A    9900                   ANL P1, #00
34C    6410                   JMP CONTINUE


350    35        MAXERR       DIS TCNTI         Errs exceeded "FF"
351    F9                     MOV A, R1
352    02                     OUTL BUS, A
353    8980                   ORL P1, #80       Output Curr Wd Cnt
355    9900                   ANL P1, #00
357    9A01                   ANL P2, #01
359    6459      WAIT 1       JMP WAIT 1


360    35        MAXWDCNT     DIS TCNTI         Wd CNT exceeded "FF"
361    FA                     MOV A, R2
362    02                     OUTL BUS, A
363    8980                   ORL P1, #80       Output Curr Err CNT
365    9900                   ANL P2, #00
367    9A00                   ANL P2, #00
369    6469      WAIT 2       JMP WAIT 2

007    65        TIMINT       STOP TIMER        Timer Int SerRout.
008    35                     DIS TCNTI         => Wd not echoed
                                                Before TIMEOUT Per
009    55                     START TIMER
00A    25                     EN TCNTI
00B    642F                   JMP ERRCNT        Incrmt error count
```

APPENDIX C

SINGLE STEP AND DISPLAY CARD SCHEMATIC

The "Single Step and Display" card was used primarily for program debugging and checkout. It consists of three Seven-Segment Displays along with the necessary "BCD- to-Seven-Segment" decoder/drivers. Because the displays used were "common-cathode" LEDs, each element of display required a "pull-up" resistor to provide enough current to make them visable.

The "Single-Step" circuitry for the card is a replica of that given in reference 4. The Address Latch Enable (ALE) from the μC8748 performs several functions. The ALE latches the the current address in the 74LS373's on-board the Mil Std-1553 card as well as latches the upper three bits of the address in a latch on-board the Single Step and Display card. The outputs of these latches are 'strapped' to provide a constantly enabled output. These outputs in turn drive the 7447's (BCD-to-Seven-Segment Decoder/Drivers), which in turn drive the displays. Note that the strapping options on the Mil Std-1553 board must be appropriatly selected to provide this function. Also note that the addresses displayed is presented in a peculiar fashion (see figure C1).
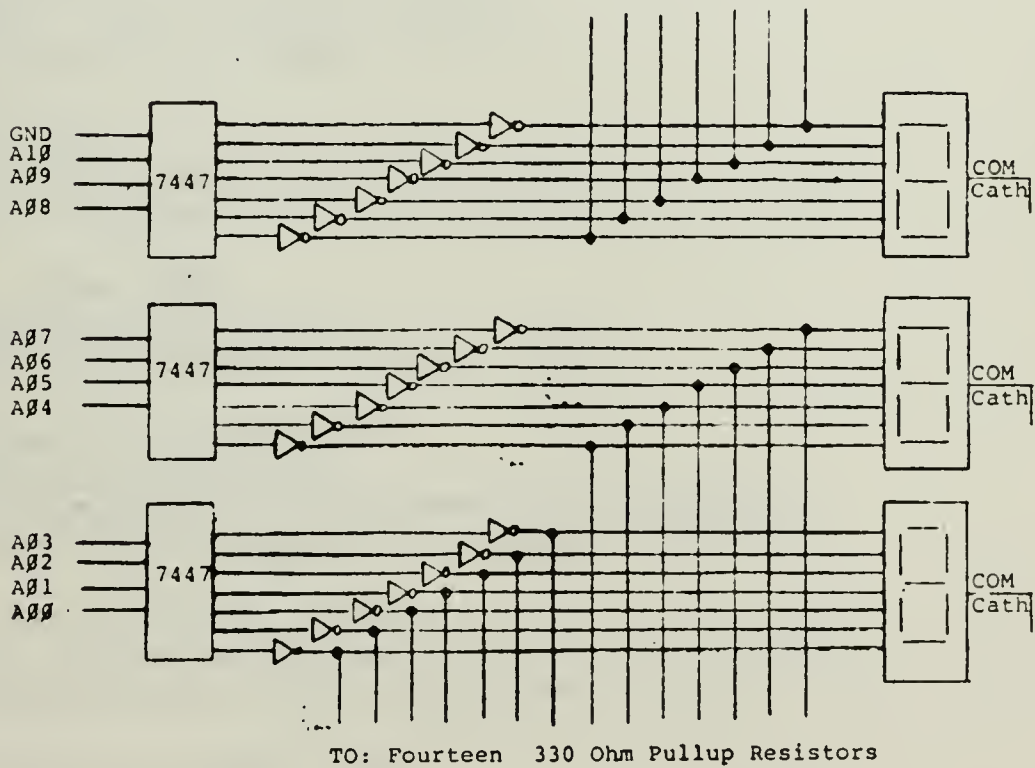
SINGLE STEP CIRCUIT
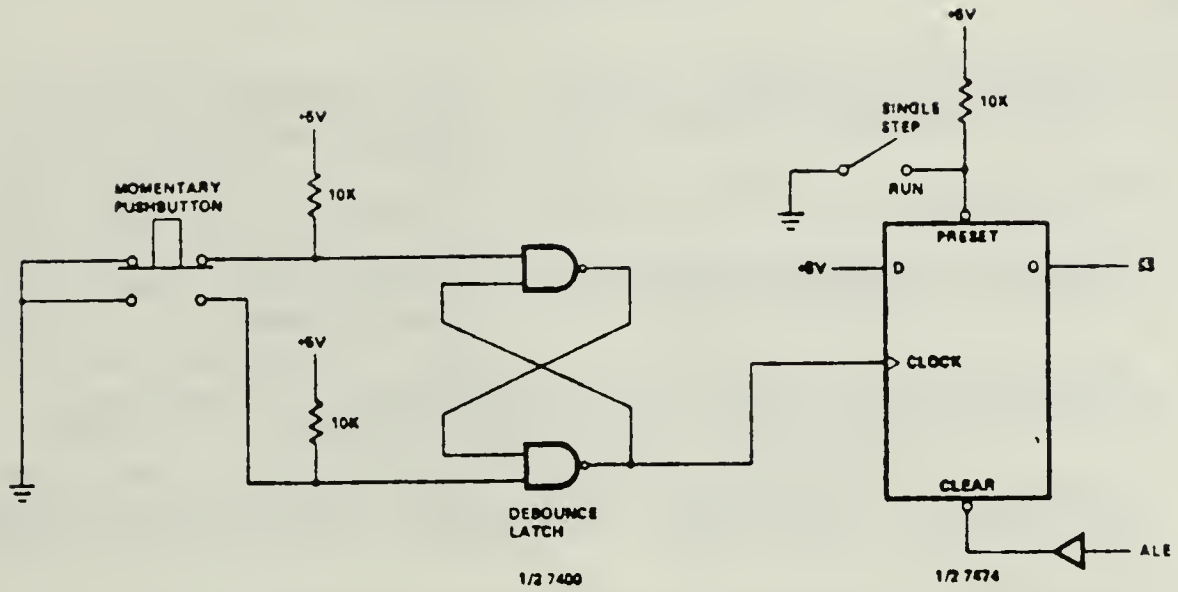
Figure Cl.   Single Step and Display Card Schematic

LIST OF REFERENCES

1. MIL-STD-1553A DESIGNER'S GUIDE, (third printing), ILC DATA
    DEVICE Corporation, Bohemia, N.Y., 1982.

2. EDN Magazine, pp. 60-72, (Volume 30, Number 8), 11 April 1985.

3. Digital IC Master, Hearst Business Communications, INC.,
    UPT Division, Santa Clara, CA., 1983, pp. 688-693.

4. MCS-48 Family of Single Chip MicroComputers - Users Manual,
    Intel Corp., Santa Clara, CA., April 1979.

5. Hewlett Packard Application Note #1009, Hewlett Packard Components,
    San Jose, CA., (Nov 1980).

6. MIL-STD-1553a DESIGNER'S GUIDE, (third printing), ILC DATA
    DEVICE Corporation, Bohemia, N.Y., 1982, pp. 1-25 -- 1-27.

INITIAL DISTRIBUTION LIST

No. Copies

1. Library, Code 0142   2
    Naval Postgraduate School
    Monterey, California 93943-5002

2. Department Chairman, Code 62   1
    Department of Electrial and Computer Engineering
    Naval Postgraduate School
    Monterey, California 93943

3. Professor J.P. Powers, Code 62 00   3
    Department of Electrical and Computer Engineering
    Naval Postgraduate School
    Monterey, California 93943

4. Professor C.R. Dunlap, Code 68DU   1
    Department of Oceanography
    Naval Postgraduate School
    Monterey, California 93943

5. COMMANDER   1
    Naval Space and Warfare Command
    (Attn: CDR R. Goldsby, PMW180-43)
    Washington, D.C. 20363

6. AFIT/CIRD   2
    Air Force Institute of Technology
    Wright Patterson Air Force Base, Ohio 45433-6583

7. Mr. Andrew Keith   1
    Armament Division/TFES
    Eglin Air Force Base, Florida 32542

8. Colonel N.W. Haars (USAF)   1
    Deputy for Operations
    Arnold Engineering Development Center (AEDC)
    Arnold Air Force Station, Tennessee 37389-5000

9. Captain F.J. Wayda (USAF)   1
    1912 Information Systems Support Group (ISSG/SYE)
    Langley Air Force Base, Alexandria, Virginia 23665

10. Captain R.S. Wester (USAF)   2
    Deputy for Operations, Flight Dynamics Division
    Arnold Engineering Development Center (AEDC)
    Arnold Air Force Station, Tennessee 37389-5000

11. Defense Technical Information Center   2
    Cameron Station
    Alexandria, Virginia 22304-6145