

AD-A074 314

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
A MICROCOMPUTER-BASED DIGITAL DATA ACQUISITION CONTROLLER FOR A--ETC(U)
JUN 79 R A COLTON

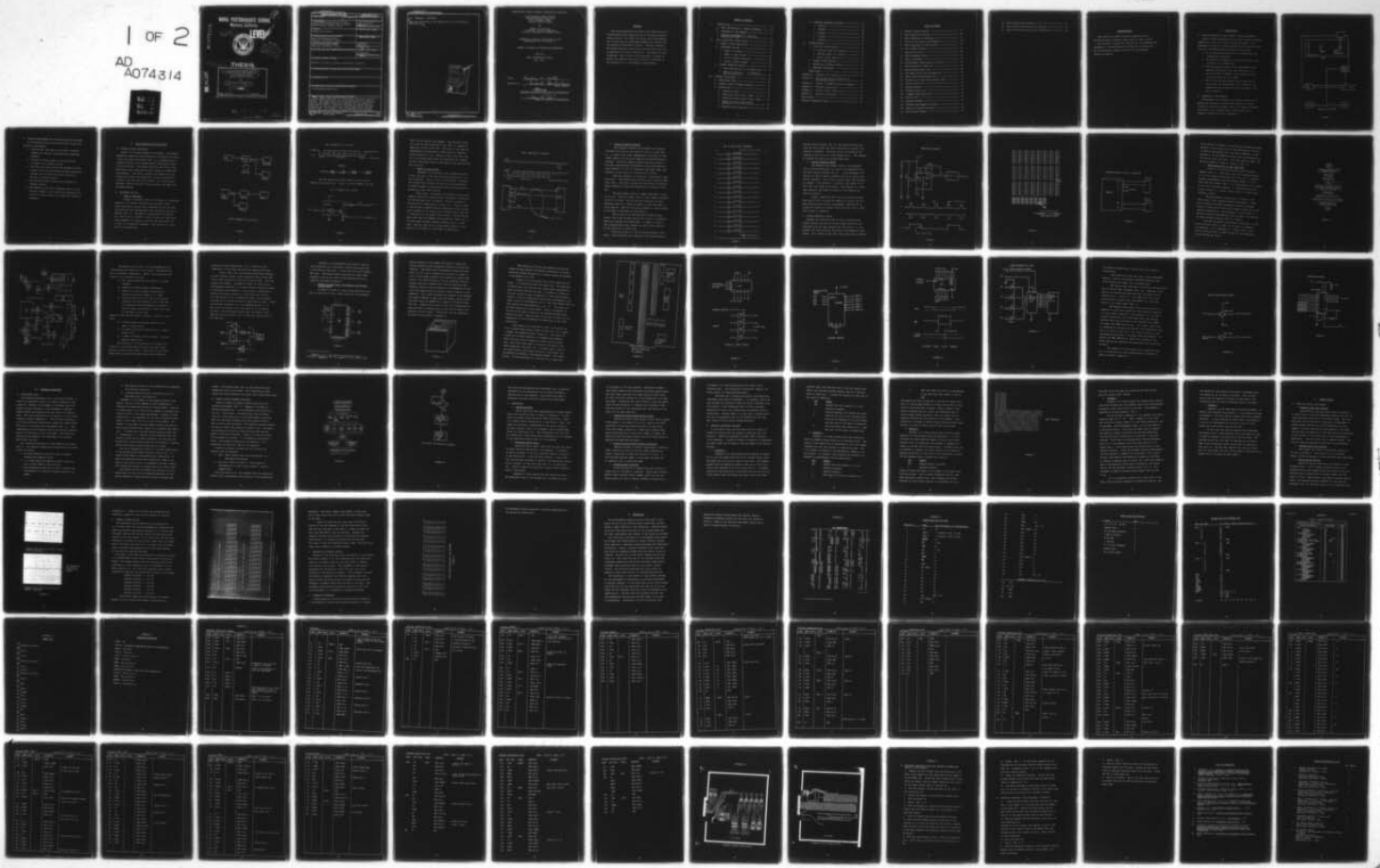
F/6 9/2

UNCLASSIFIED

NL

1 OF 2

AD
A074314



2

NAVAL POSTGRADUATE SCHOOL Monterey, California

ADA 074314

12 97p.

LEVEL II



D D C
RECEIVED
SEP 27 1979
R
L
G
I
T
V
E

9 Master's **THESIS,**

DDC FILE COPY

6 A MICROCOMPUTER-BASED DIGITAL
DATA ACQUISITION CONTROLLER
FOR A COMPUTER AIDED
ACOUSTIC IMAGING SYSTEM

by

10 Rodney Alvie Colton

111 June 79

Thesis Advisor: R. Panholzer

Approved for public release; distribution unlimited.

79 09 24 127

251 450

Gu

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Microcomputer-Based Digital Data Acquisition Controller for a Computer Aided Acoustic Imaging System		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1979
7. AUTHOR(s) Rodney Alvie Colton		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1979
		13. NUMBER OF PAGES 96
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Microcomputer Base Design		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis describes the design and construction of a micro-computer based controller for an ultrasonic acoustic imaging system. The INTEL 8748 single chip microcomputer was utilized and the associated hardware and software for the system are described in detail. Carefully designed and tested operating instructions are provided along with an explanation for each instruction. The system is fully documented, thus allowing future personnel to change or update the system as required to		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered

#20 - ABSTRACT - CONTINUED

take full advantage of the flexibility of a microcomputer based design. ↖

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

79 09 24 127

DD Form 1473
S/N 0102-014-6601

2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered

Approved for public release; distribution unlimited.

A Microcomputer-Based Digital
Data Acquisition Controller
for a Computer Aided
Acoustic Imaging System

by

Rodney Alvie Colton
Lieutenant, United States Navy
B.S.E.E., University of Nevada, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1979

Author

Rodney A. Colton

Approved by:

Rudolf Pankovyc
Thesis Advisor

L. E. Kik

Chairman, Department of Electrical Engineering

William M. Tolles

Dean of Science and Engineering

ABSTRACT

This thesis describes the design and construction of a microcomputer based controller for an ultrasonic acoustic imaging system. The INTEL 8748 single chip microcomputer was utilized and the associated hardware and software for the system are described in detail. Carefully designed and tested operating instructions are provided along with an explanation for each instruction. The system is fully documented, thus allowing future personnel to change or update the system as required to take full advantage of the flexibility of a microcomputer based design.

TABLE OF CONTENTS

I.	INTRODUCTION -----	10
	A. BASIC DESCRIPTION OF IMAGING APPARATUS -----	10
	B. STATEMENT OF THE PROBLEM -----	10
	C. DETAILED REQUIREMENTS FOR THE DATA ACQUISITION SYSTEM -----	12
II.	DATA ACQUISITION SYSTEM DESIGN -----	13
	A. OVERALL SYSTEM DESCRIPTION -----	13
	B. PERIPHERAL DEVICES -----	13
	1. Model 33 Teletype -----	13
	2. Model 40 Line Printer -----	16
	3. Digital Cassette Recorder -----	18
	4. Digital Cassette Reader -----	20
	C. SYSTEM CONTROLLER DESIGN -----	20
	1. Basic Description of the INTEL 8748 -----	24
	2. Design of Stand Alone Microcomputer and Printed Circuit Board -----	29
III.	SOFTWARE DEVELOPMENT -----	40
	A. DEVELOPMENT TOOLS -----	40
	B. OVERALL SYSTEM SOFTWARE STRUCTURE -----	42
	C. SUBROUTINES -----	45
	1. Subroutine Delay -----	45
	2. Subroutine ASCII output (ASCO) -----	45
	3. Subroutine Binary Coded Output (BCDO) ---	46
	4. Binary To Binary Coded Decimal Conversion -----	46
	5. Teletype Output Routines -----	46

D.	KEYBOARD SELECTABLE PROGRAMS -----	47
1.	Program 0 -----	47
2.	Program 1 -----	48
3.	Program 2 -----	49
4.	Program 3 -----	51
5.	Program 4 -----	52
IV.	SYSTEM TESTING -----	53
A.	INPUT AND OUTPUT TESTING -----	53
1.	Parallel Input Port Testing -----	53
2.	Latched Output Port Testing -----	53
3.	Serial Output Testing -----	53
B.	OVERALL SYSTEM TESTING -----	55
C.	MECHANICAL ALIGNMENT TESTING -----	57
D.	OPERATING PROCEDURES -----	57
V.	CONCLUSIONS -----	60
APPENDIX A:	Complete 8748 Instruction Set -----	62
APPENDIX B:	Wiring Data For 8748 Stand Alone and Extender Boxes -----	63
APPENDIX C:	Summary of PROMPT-48 Monitor Commands ---	67
APPENDIX D:	Software Documentation -----	68
APPENDIX E:	Printed Circuit Board Layout -----	89
APPENDIX F:	Operating Procedures -----	91
BIBLIOGRAPHY	-----	94
INITIAL DISTRIBUTION LIST	-----	95

LIST OF FIGURES

1. Acoustic Imaging System -----	11
2. Data Acquisition System -----	14
3. ASCII Character To Teletype -----	15
4. TTL To 20 mA Current Loop Adapter -----	15
5. ASCII Character To Line Printer -----	17
6. TTL To CMOS Converter -----	19
7. Revolution Counter -----	21
8. LPR-16 Readback Format -----	22
9. LPR-16 Interface -----	23
10. Microcomputer Based Design Flow Chart -----	25
11. Block Diagram of INTEL 8748 -----	26
12. Quasi-bidirectional Port -----	28
13. The Stand Alone 8748 Microcomputer -----	29
14. Stand Along Microcomputer Box -----	30
15. Controller Printed Circuit Board Layout -----	32
16. Typical Input Port -----	33
17. Address Decoder -----	34
18. Latched Output Port -----	35
19. 7 Segment LED Output Port -----	36
20. RS-232 Ports -----	38
21. Keyboard Encoder -----	39
22. Software Development Flow Chart -----	43
23. Executive Program Flow Chart -----	44
24. Test Program Number 2 -----	50

25.	RS-232 Square Wave Testing -----	54
26.	Data Verification Printout On Teletype -----	56
27.	Data Verification Printout On Line Printer -----	58

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. ENG 77-21600.

Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

I. INTRODUCTION

A. BASIC DESCRIPTION OF THE ACOUSTIC IMAGING APPARATUS

Figure 1 shows diagrammatically the basic acoustic imaging apparatus to which this thesis project was added. The mathematical details of image reconstruction are not covered here, but the basic physical process is as follows:

- 1) A 1.0 MHz plane acoustic wave is generated by the transmitting transducer.
- 2) The receiving transducer is moved horizontally and vertically in a raster scan.
- 3) The magnitude and phase of the received wave are sampled periodically such that the result is a 64 by 64 array of samples, each consisting of an 8 bit magnitude and 8 bit phase.
- 4) The 16 bits for each sample are provided at two 14 pin dual-in-line-package (DIP) sockets on the sample-hold and analog-to-digital-conversion (ADC) box of figure 1.

B. STATEMENT OF THE PROBLEM

The problem to be solved by this thesis project was to design and construct a digital data acquisition system controller which would accept the 64 by 64 array of samples and record it in a format which could be read into a remotely located PDP-11 for processing.

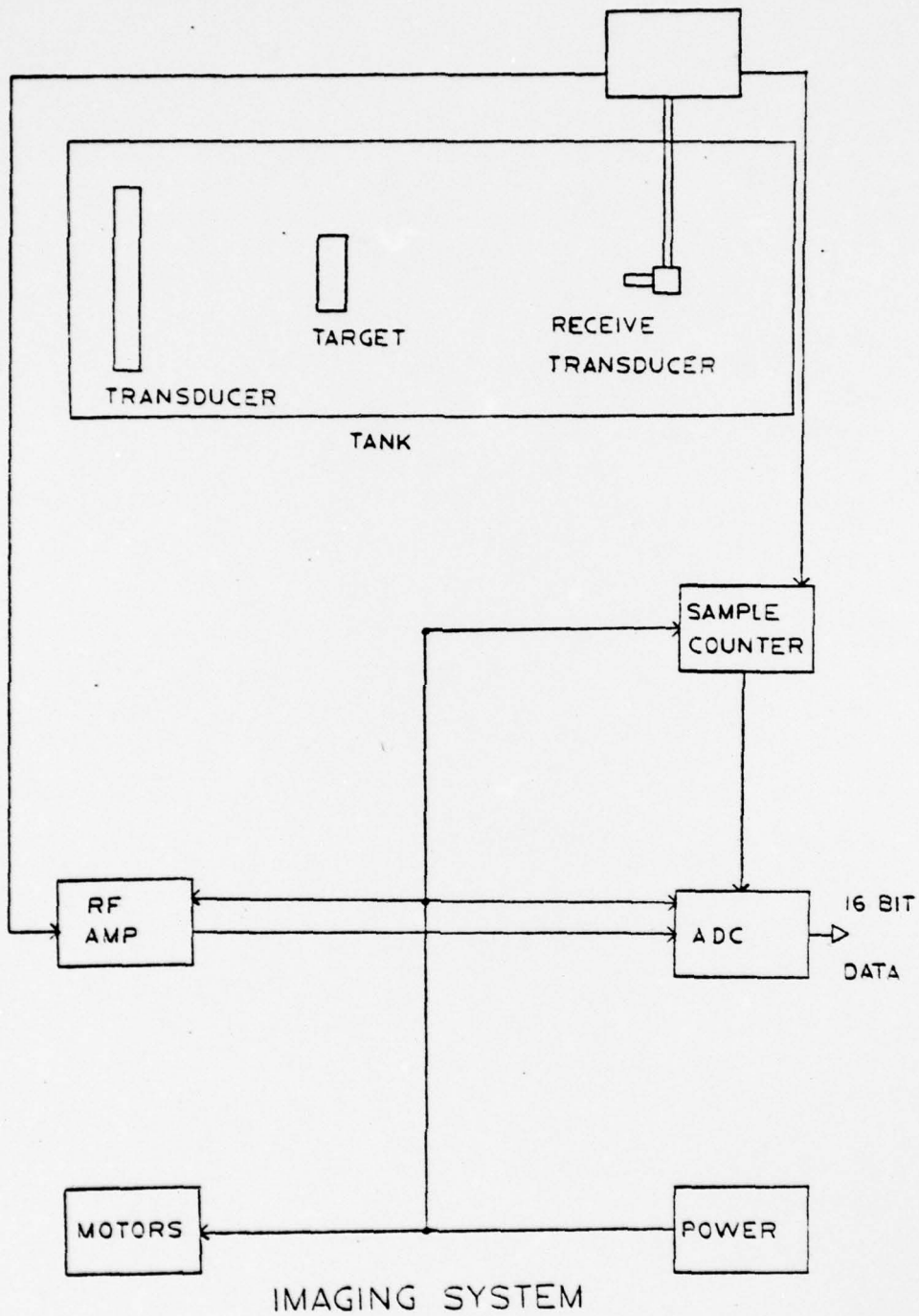


FIGURE 1

C. DETAILED REQUIREMENTS FOR THE DATA ACQUISITION SYSTEM

As a minimum the controller must be able to meet the following requirements:

- 1) Provide real time copy on a printer in decimal numbers for testing, calibration and comparison purposes.
- 2) Provide for manual sampling for testing and alignment of the imaging system.
- 3) The controller must provide the necessary control signals to set up the imaging and data acquisition systems, and to start and control them with a minimum of operator action.
- 4) The controller must provide test programs for hardware testing.
- 5) The system will be fully documented both in hardware and software, such that future personnel will be able to easily operate and update the system as necessary.

II. DATA ACQUISITION SYSTEM DESIGN

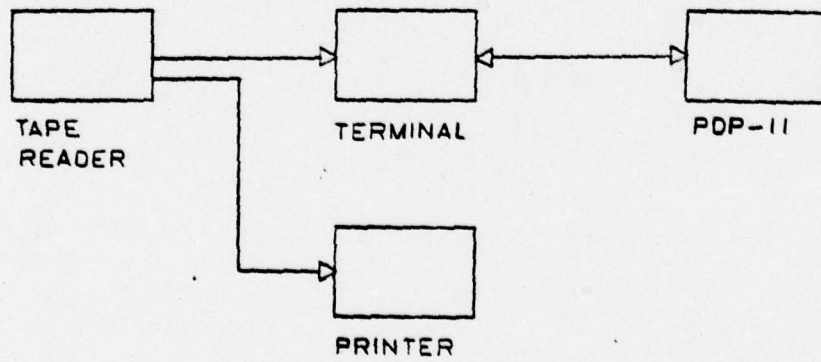
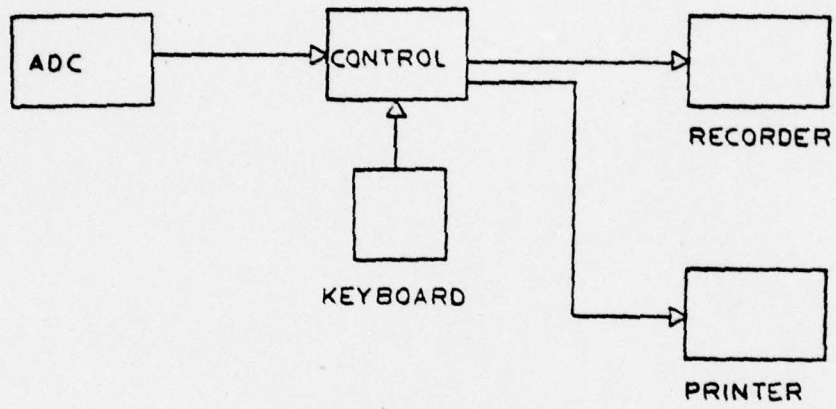
A. OVERALL SYSTEM DESCRIPTION

Figure 2 is a block diagram of the system. The electro-mechanical system and the electronics up the ADC's of figure 1 were previously constructed [Ref. 1], but required considerable testing and alignment before satisfactory operation was achieved. The system controller and control keyboard were custom designed and built around the INTEL 8748 microcomputer. The digital recorder and reader were purchased with options available which could be modified for interface with the rest of the system. The line printer, teletype, terminal and PDP-11 were available and used without modifications.

B. PERIPHERAL DEVICES

1. Model 33 Teletype

A model 33 teletype (TTY) was included as a peripheral because it was readily available and can readily make paper tapes which can be easily annotated as they are made and which are very transportable from one computer system to another [Ref. 2]. The model 33 accepts ASCII characters at a rate of up to 100 WPM. Figure 3 shows that each character consists of a start, 8 bits and two stop bauds, for a total of 11 bauds per character. This results in a baud rate of 110 bauds/sec:



DATA ACQUISITION SYSTEM

FIGURE 2

ASCII CHARACTER TO TELETYPE

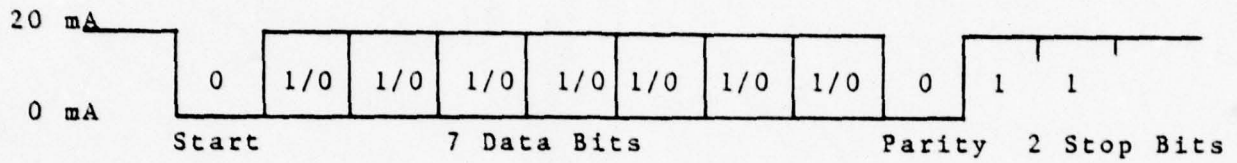


FIGURE 3

$$100 \frac{W}{M} \times 6 \frac{\text{Char.}}{\text{Word}} \times \frac{1 \text{ Min}}{60 \text{ Sec}} \times 11 \frac{\text{bauds}}{\text{Char.}} = 110 \frac{\text{Bauds}}{\text{Sec}}$$

The TTY requires the baud stream to be an on-off keyed 20 mA current loop. Figure 4 is the schematic for the

TTL TO CURRENT LOOP ADAPTER

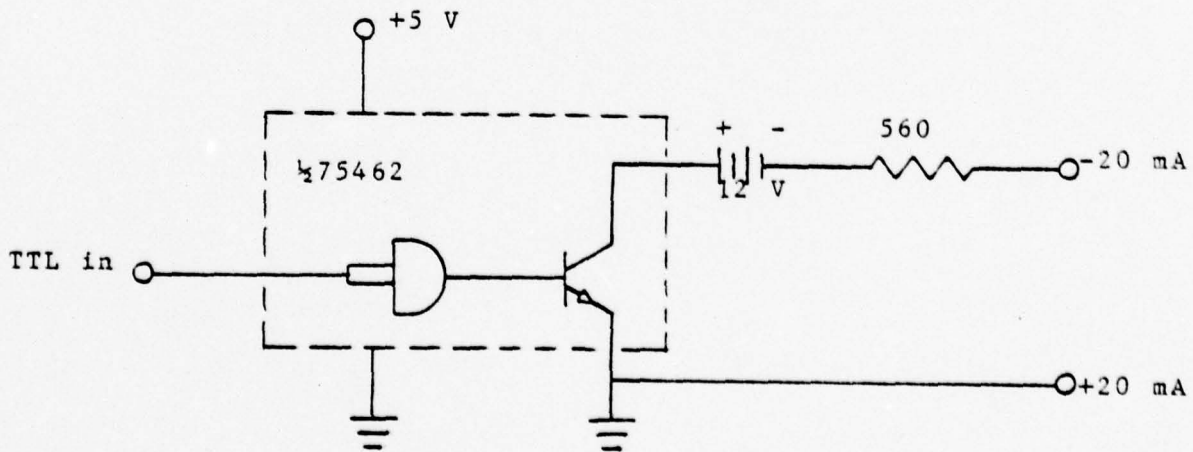


FIGURE 4

TTL to 20 mA current loop adapter. The Motorola 75462P is a high voltage peripheral driver [Ref. 3] capable of supplying 300 mA to the load. The value of 560 Ohms was experimentally determined to ensure that the "marking" current would be 20 mA. Worthy of note is the fact that the 12 VDC power supply must be ungrounded with respect to the 5 VDC supply which is the same as that used for the TTL circuitry.

2. Model 40 Line Printer

The model 40 line printer was included such that a real time copy of the data could be provided as it is being recorded. Since it has an "RS-232 like" interface, the hardware and software utilized to drive the printer is readily usable to drive other RS-232 devices such as a cathode ray tube (CRT) terminal.

The printer, as currently configured for the micro-computer laboratory, requires ASCII characters transmitted at 2400 baud/sec. with "no parity", which is defined as the parity bit always being a logical "1". Figure 5 shows a typical ASCII character being transmitted to the printer. Also included in the figure are the other signals [see Ref. 4) required by or provided by the printer. The controller must sense when the "Request Next Character" (RNC) line is low and cannot transmit another character until it goes high. RNC will stay low for significant periods of time during a "line feed" or "form feed" by the printer.

ASCII CHARACTER TO PRINTER

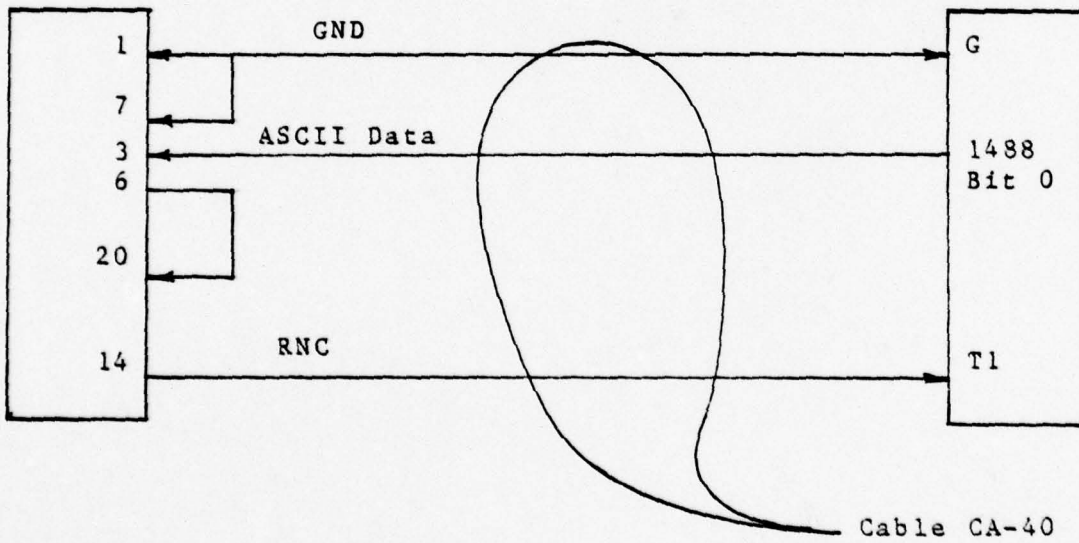
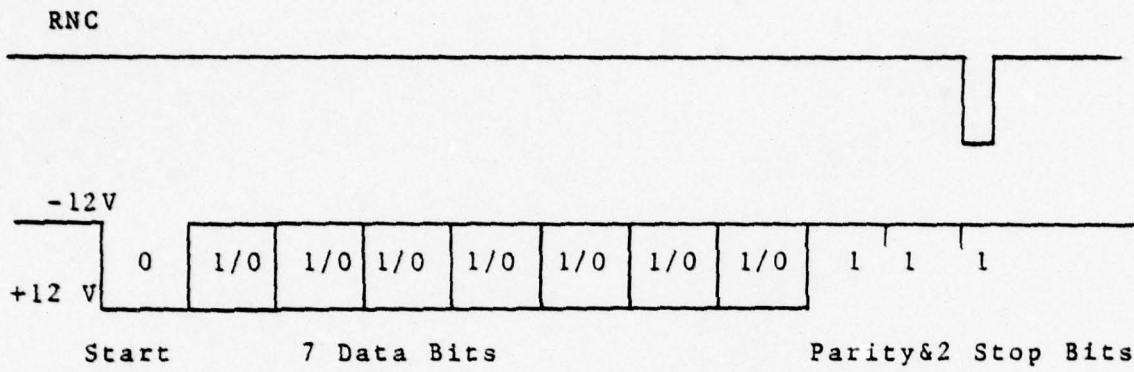


FIGURE 5

3. Digital Cassette Recorder

The cassette recorder was purchased with options available such that it can record 16 bits at a time [Ref. 5]. It uses a dual-track complementary non-return to zero (CNRZ) method for high noise immunity and self-clocking on playback. This high capacity method uses both tracks of the tape simultaneously and records in only one direction. Using separate "1" and "0" tracks and very small gaps, one cassette can hold up to 2.2 million bits of data.

The tape transport is driven by a four-winding stepping motor. The stepping motor drivers are clocked by CMOS block logic. The tape is moved only while data is actually being recorded or during the generation of a word or file gap.

The word length, which is jumper selectable, is set at 16 bits. The file length, which is also jumper selectable, is set at 64 words. The bit and word counters are automatically reset by a "power on reset" at the time power is first applied. They are also reset whenever a "load forward" of the tape is performed by the operator.

The 16 bits to be recorded must be CMOS voltage levels, and since the ADCs provide TTL levels, an interface board was designed and constructed. As shown in figure 6 the interface board also changes the start pulse provided by the controller to CMOS level.

The start pulse is a one millisecond negative true pulse. Since the ADCs are triggered by the falling edge of

TTL TO CMOS LEVEL CONVERTER

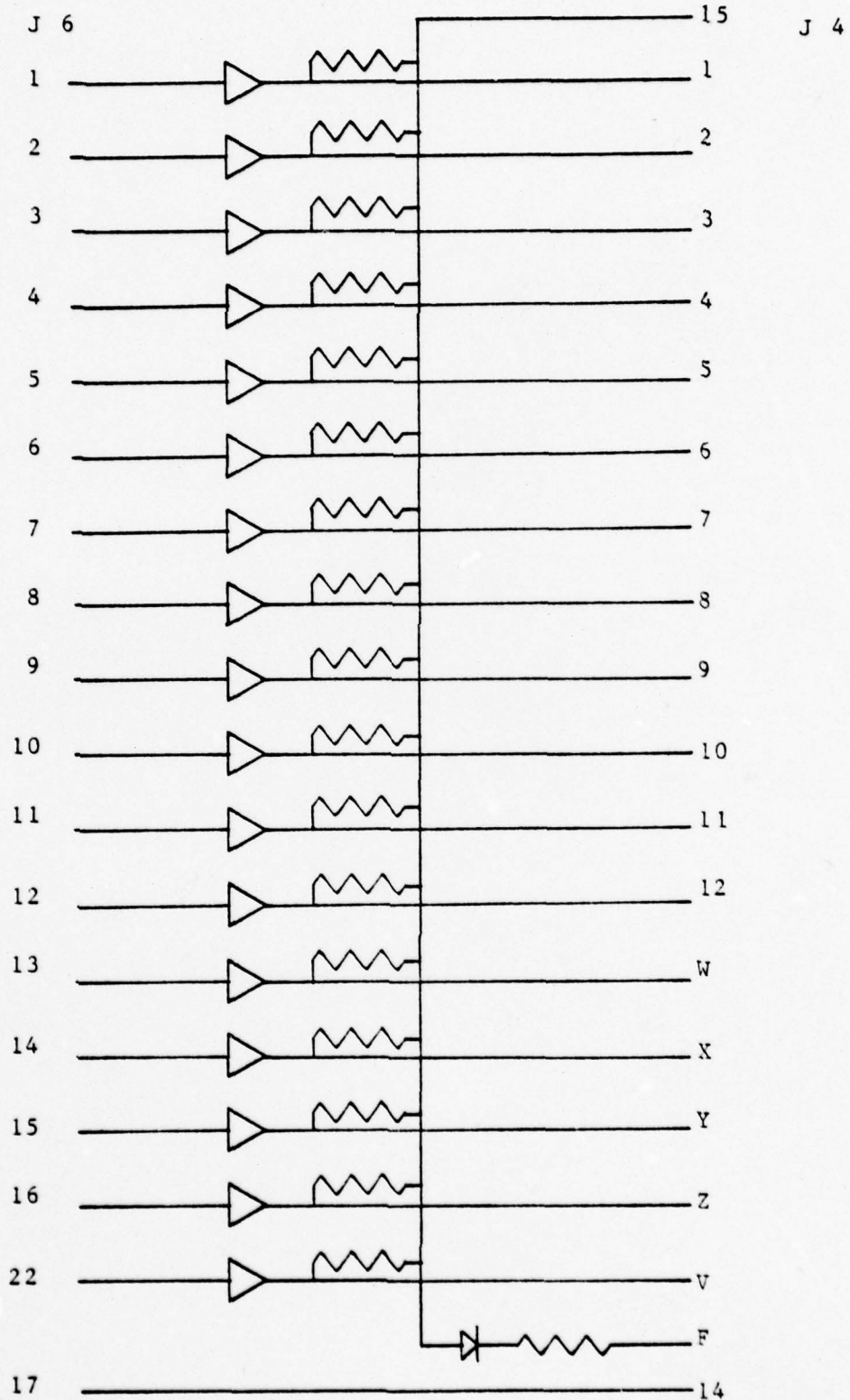


FIGURE 6

the revolution counter (Fig. 7), the controller must provide the start pulse only after the analog to digital conversion is complete. The start pulse is provided by the controller out of bit 2 of the latched port 6. The software to produce the pulse will be described later.

4. Digital Cassette Reader

The cassette tape reader (LPR-16) was purchased from the same manufacturer in order to be compatible with the CNRZ recording format [Ref. 6]. It was purchased with options for RS-232 serial interface and 20 mA current loop serial interface. The following options were user selected by jumpers: 2400 baud/sec., no parity, word length of 16 bits, and file length of 64 words. This results in a read-back format on the model 40 printer or CRT terminal as shown in figure 8.

Figure 9 shows the LPR-16 serial interface pinout. When switch SW-100 is open the reader will output one file each time the start button is depressed. When SW-100 is closed the reader will output files until the "end of tape" (EOT) signal is received.

C. SYSTEM CONTROLLER DESIGN

Because the controller must be able to perform such rather complex functions as binary-to-BCD conversion and interface with the tape recorder and line printer, it was decided that the controller should be a microcomputer based design. This further allows many timing and control problems

REVOLUTION COUNTER

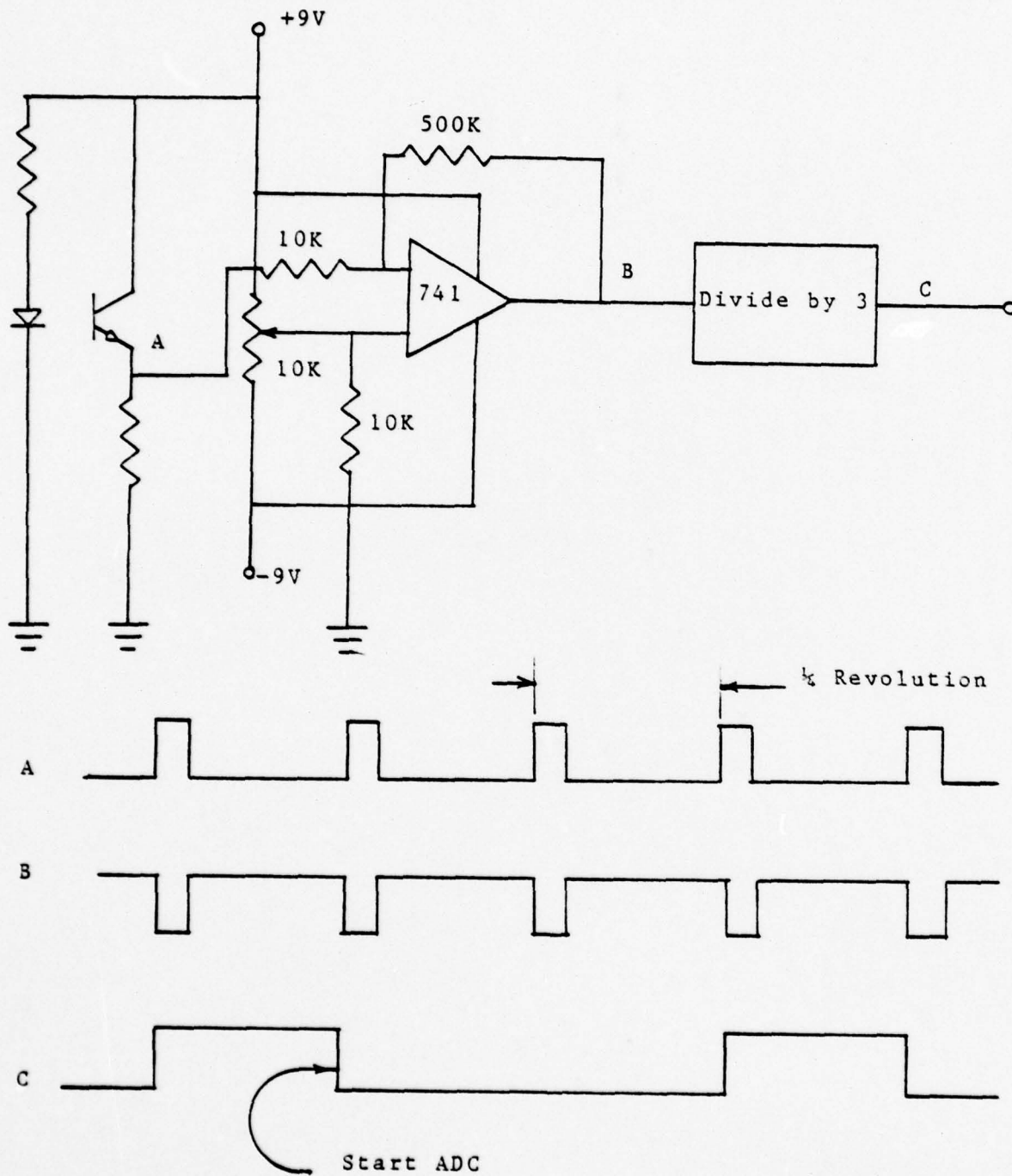


FIGURE 7

CCC 0	CCC 1	CCC 2	CCC 3	CCC 4	CCC 5	CCC 6	CCC 7
CCC 8	CCC 9	CCC A	CCC B	CCC C	CCC D	CCC E	CCC F
CCC 0	CCC 1	CCC 2	CCC 3	CCC 4	CCC 5	CCC 6	CCC 7
CCC 8	CCC 9	CCC A	CCC B	CCC C	CCC D	CCC E	CCC F
CCC 0	CCC 1	CCC 2	CCC 3	CCC 4	CCC 5	CCC 6	CCC 7
CCC 8	CCC 9	CCC A	CCC B	CCC C	CCC D	CCC E	CCC F
CCC 0	CCC 1	CCC 2	CCC 3	CCC 4	CCC 5	CCC 6	CCC 7
CCC 8	CCC 9	CCC A	CCC B	CCC C	CCC D	CCC E	CCC F

333 0	333 1	333 2	333 3	333 4	333 5	333 6	333 7
333 8	333 9	333 A	333 B	333 C	333 D	333 E	333 F
333 0	333 1	333 2	333 3	333 4	333 5	333 6	333 7
333 8	333 9	333 A	333 B	333 C	333 D	333 E	333 F
333 0	333 1	333 2	333 3	333 4	333 5	333 6	333 7
333 8	333 9	333 A	333 B	333 C	333 D	333 E	333 F
333 0	333 1	333 2	333 3	333 4	333 5	333 6	333 7
333 8	333 9	333 A	333 B	333 C	333 D	333 E	333 F

555 0	555 1	555 2	555 3	555 4	555 5	555 6	555 7
555 8	555 9	555 A	555 B	555 C	555 D	555 E	555 F
555 0	555 1	555 2	555 3	555 4	555 5	555 6	555 7
555 8	555 9	555 A	555 B	555 C	555 D	555 E	555 F
555 0	555 1	555 2	555 3	555 4	555 5	555 6	555 7
555 8	555 9	555 A	555 B	555 C	555 D	555 E	555 F
555 0	555 1	555 2	555 3	555 4	555 5	555 6	555 7
555 8	555 9	555 A	555 B	555 C	555 D	555 E	555 F

FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF

1111 1111 1111 1111

16-BIT DIGITAL WORD

FIGURE 8

CASSETTE READER (LPR-16) INTERFACE

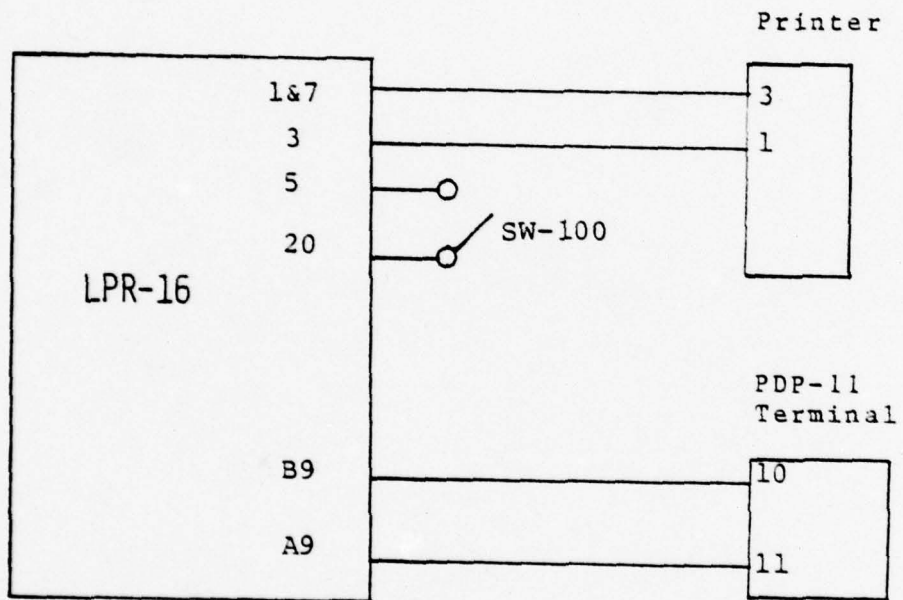


FIGURE 9

to be solved in software, thus minimizing hardware problems. Because of the availability of the microcomputer chip and a suitable development tool (the INTEL PROMPT-48) it was decided to use the INTEL 8748 single chip microcomputer as the heart of the controller. With that decision made, the design proceeded as shown in figure 10 [Ref. 7].

1. Basic Description of the INTEL 8748

Figure 11 is a block diagram of the 8748 microcomputer [Ref. 8]. The single 40 pin DIP package contains a) an 8 bit central processing unit (CPU), b) a 1 kilobyte erasable programmable read only memory (EPROM), c) a 64 word read/write memory (RAM), d) 27 input/output (I/O) lines, e) an 8 bit timer/event counter, f) oscillator and clock driver circuits, g) a reset circuit, and h) a single level interrupt circuit.

The microcomputer requires only a single 5 volt power supply, which reduces hardware requirements. There are two banks of directly addressable working registers either of which can be selected for added flexibility during subroutine executions. The remainder of the 64 words of RAM are indirectly addressable using the lower two registers of either working register bank as a pointer.

The CPU can perform the following functions: a) Add with or without carry, b) AND, OR, or EXOR, c) Increment or Decrement, d) Bit complement, e) rotate right or left with or without carry, f) SWAP nibbles of the accumulator, and g) BCD decimal adjust.

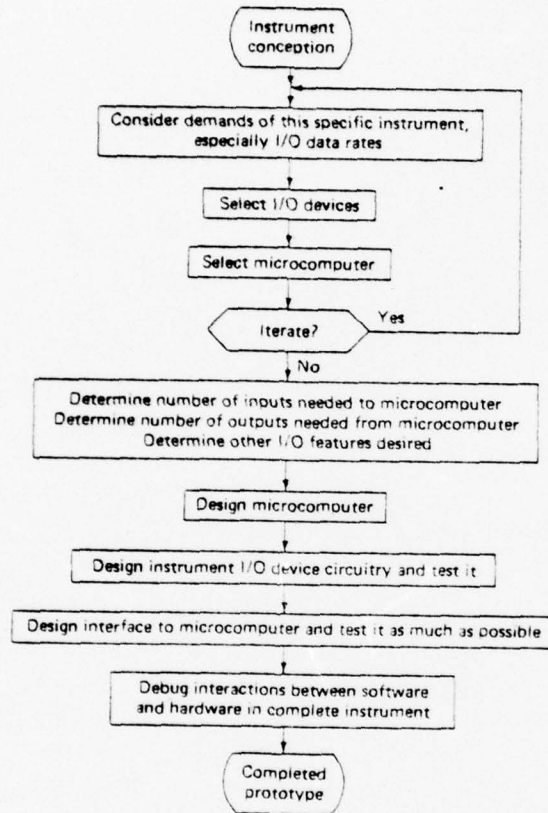


FIGURE 10

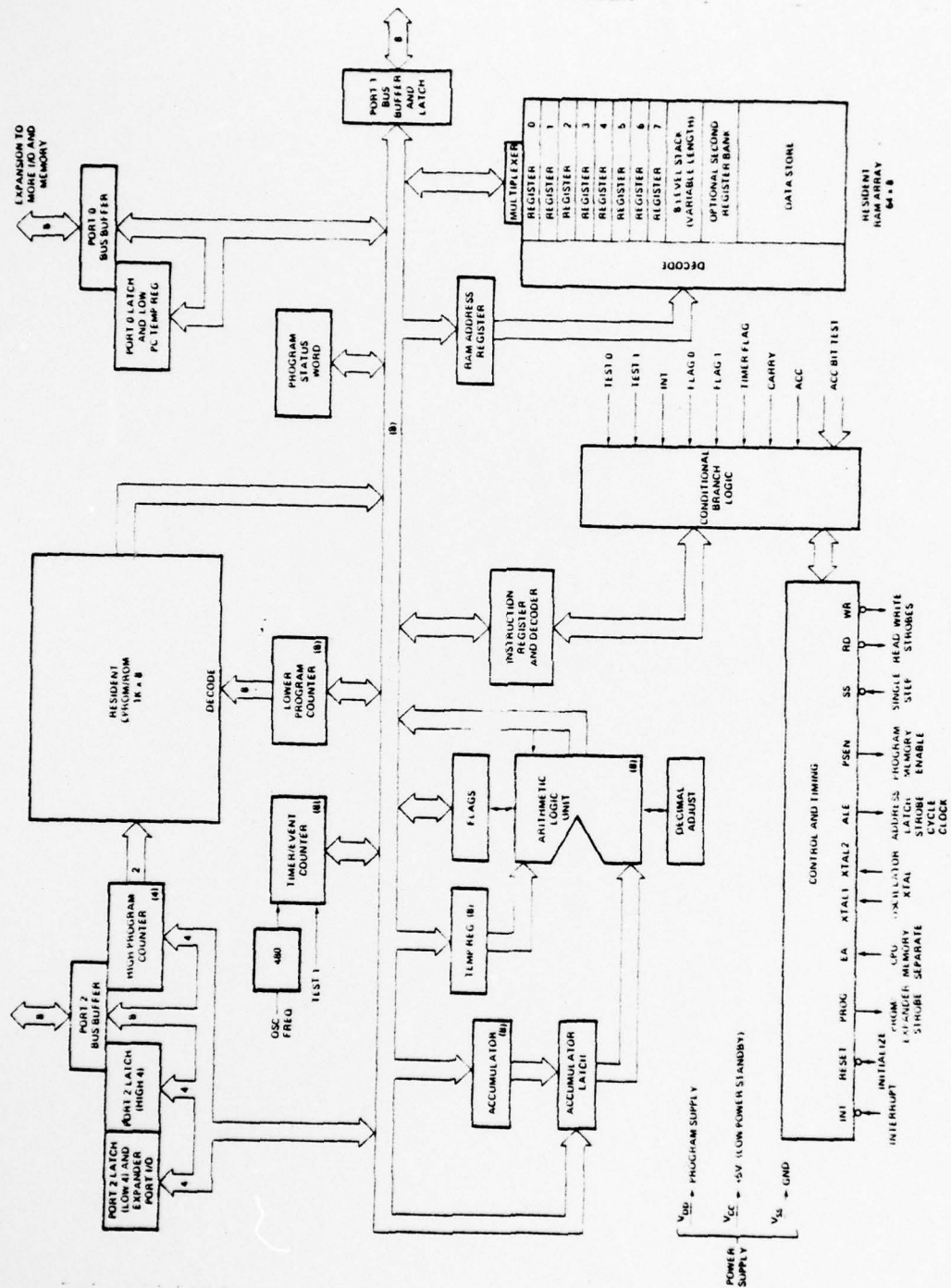


FIGURE 11

The machine cycle time is 2.5 microseconds and all instructions are either one or two cycles. The instruction set is included in Appendix A. Some of the significant features of the instruction set are:

- a) All jump instructions are relative to a page boundary.
- b) There is no direct compare instruction.
- c) There is no auto increment instruction so multiple precision arithmetic is clumsy.
- d) Very efficient use can be made of the working registers as loop counters by use of the DJNZ Rr instruction which decrements the register, test for zero, and executes a conditional jump if the result is not zero.

There are three program memory locations of special significance:

- a) A reset causes the program counter (PC) to jump to location 000.
- b) An external interrupt causes the PC to jump to location 003.
- c) A timer or counter interrupt causes a program jump to location 007.

The 3748 has 27 I/O lines which are grouped into 3 ports of 8 bits each which can serve as either inputs, outputs or bidirectional ports. There are also two "test" inputs which can alter program sequences when tested by

conditional jump instructions. All I/O ports are TTL compatible, in that they can drive one standard TTL load.

Ports 1 and 2 are called quasibidirectional because of their special output circuit structure which allows each line to serve as both an input and an output port. Only the outputs are latched. As shown in figure 12, each line is continuously pulled up to +5 volts through a resistive device of relatively high impedance. This pullup is sufficient to provide the source current for a TTL high level yet can still be pulled low by a standard TTL gate, thus allowing the same pin to be used as both an input and an output. Since the pulldown transistor is a low impedance device, a "1" must first be written to any line which is to be used as an input. This structure allows input and output on the same pin and also allows a mix of input and output lines on the same port.

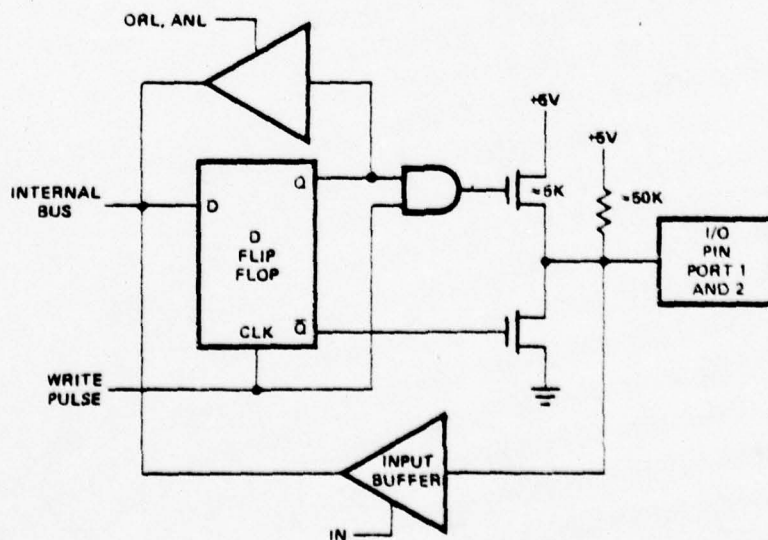


FIGURE 12

The BUS is a bidirectional port which is used in this application as a statically latched output port and a non-latching input port. It can also drive one standard TTL load. "The MCS-48 family satisfied the usual INTEL strategy of being the first in the marketplace with an imperfect but useful product."¹

2. Design of Stand Alone Microcomputer and Printed Circuit Board

As Shown in figure 13, very little additional hardware is required to form a stand alone 8748 microcomputer.

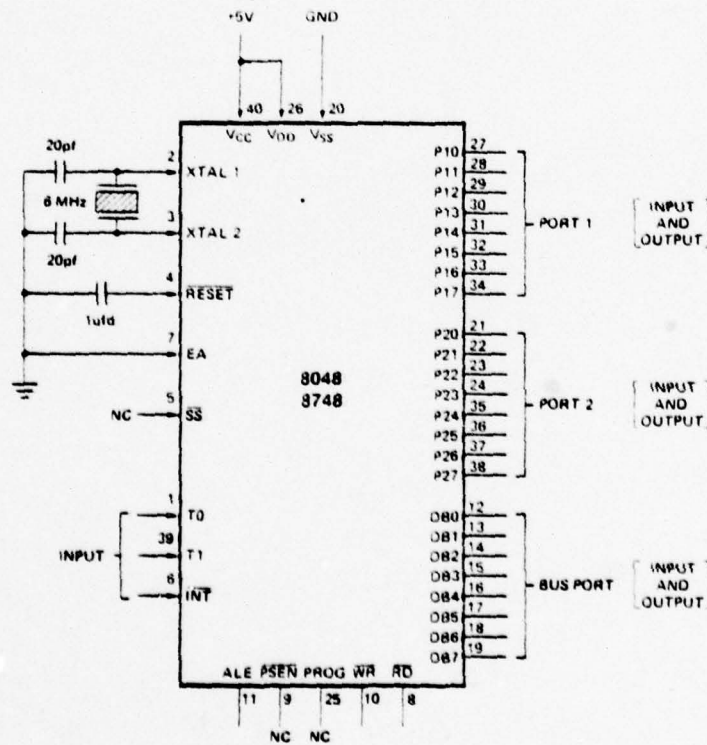


FIGURE 13

¹Wakerly, J.F., "The MCS-48 Microcomputer Family: A Critique," Computer, Vol. 12, Number 2, p. 30, Feb., 1979.

however, because of the number and kinds of inputs and outputs numerous other peripheral interface circuits were required. The stand alone microcomputer system was built into a 4" by 5" by 6" aluminum box as shown in figure 14 with a 50 pin edge connector for insertion of the printed circuit (PC) board holding all of the interface circuits. Another identical "extender box" with an identical edge connector was built which contains no microcomputer but has a 50 line ribbon cable and connector which connects to the PROMPT-48. This allows hardware and software to be debugged and tested together in what could be lossely called an emulate mode using the PROMPT-48 in place of the stand alone 8748 box. Unused pins on each of the 50 pin connectors are made available at banana plugs on each box for making connections to the PC board. Wiring data for the connectors and boxes are provided in Appendix B.

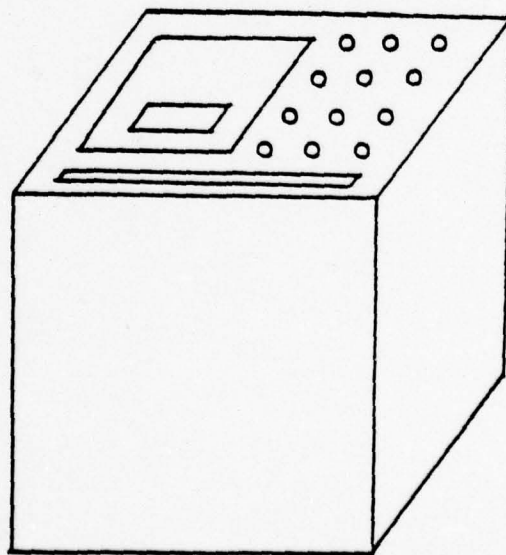
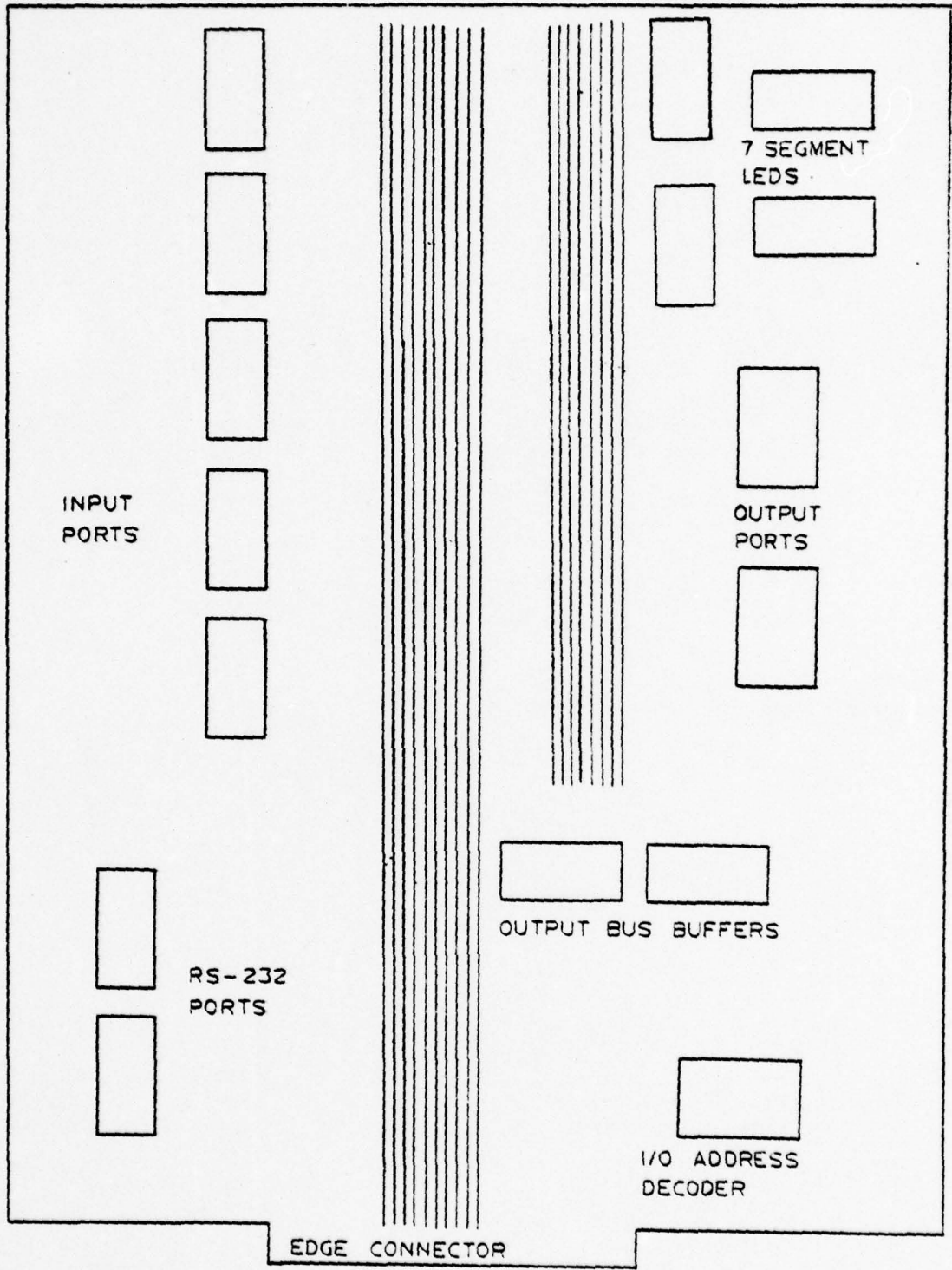


FIGURE 14

The controller PC board was designed with an eye toward minimum hardware and maximum flexibility in control functions through the ability to readily change the program in the EPROM of the 8748.

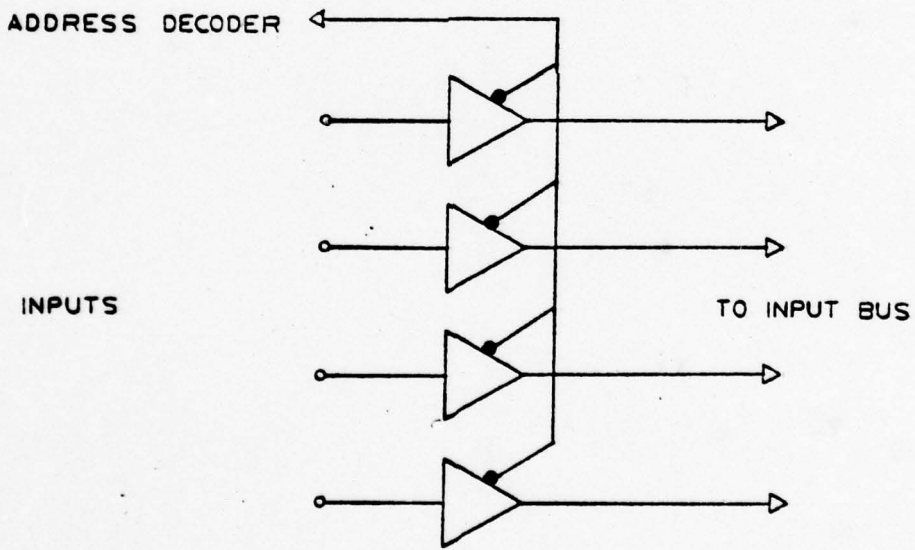
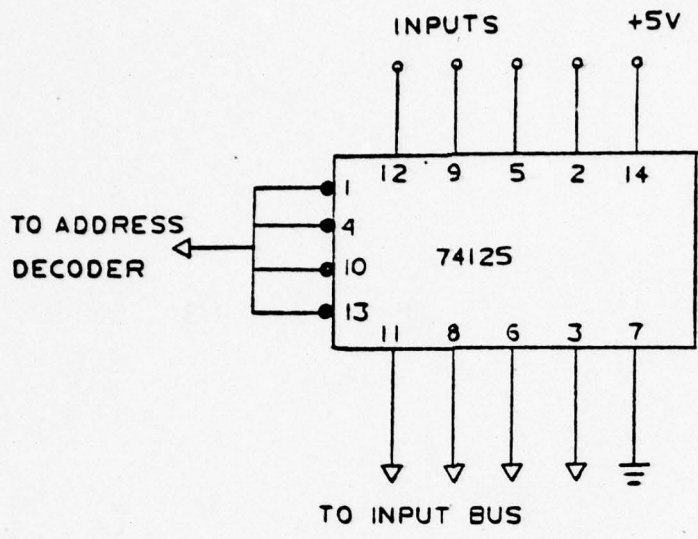
Figure 15 is a pictorial view of the controller PC board. It contains an input and output bus. The input bus is fed by one 4 bit and two 8 bit input ports consisting of 74125 tri-state buffers (see Ref. 9) as shown in figure 16. Only one input port may be enabled at a time under software control. A port is selected by outputting its address to the lower 4 bits of microcomputer port 2. The 74154 decode [Ref. 10] then enables the designated port (Fig. 17). The timing for an input operation is very simple. First, the address of the port to be read is output to the address decoder which selects the port. Then the data on the port is read into the accumulator of the microcomputer by executing an INS A,BUS instruction.

The output bus is buffered so that it can drive two 8 bit latched output ports (figure 18) and two 4 bit latched 7 segment LED decoder/drivers (figure 19). The timing required for an output operation is somewhat more complex, since the data is not latched into either type of output port until the rising edge of an active low pulse on the latch enable (LE) line [Ref. 10]. To output data to one of the output ports, first output the data to the output bus by sending its address to the address decoder. This takes LE low. To latch the data into the port LE is taken high



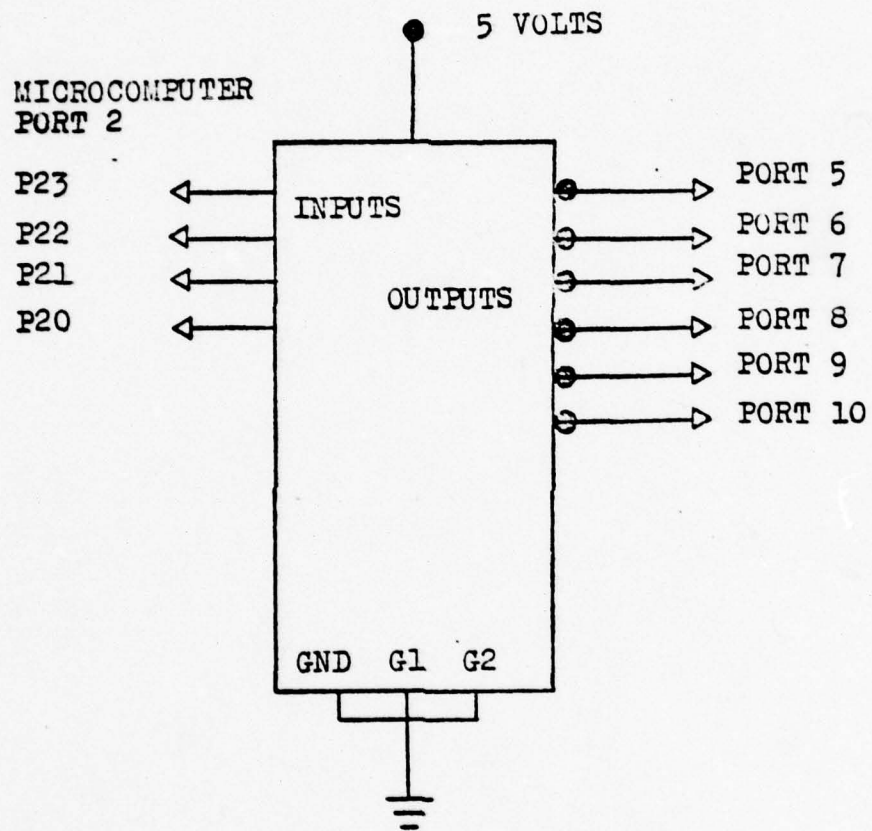
MICROCOMPUTER
PC BOARD

FIGURE 15



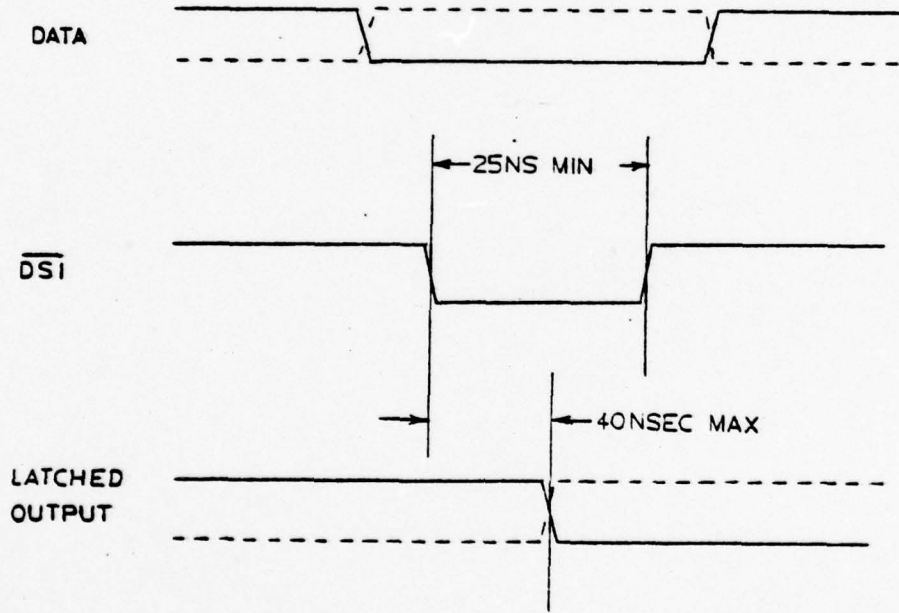
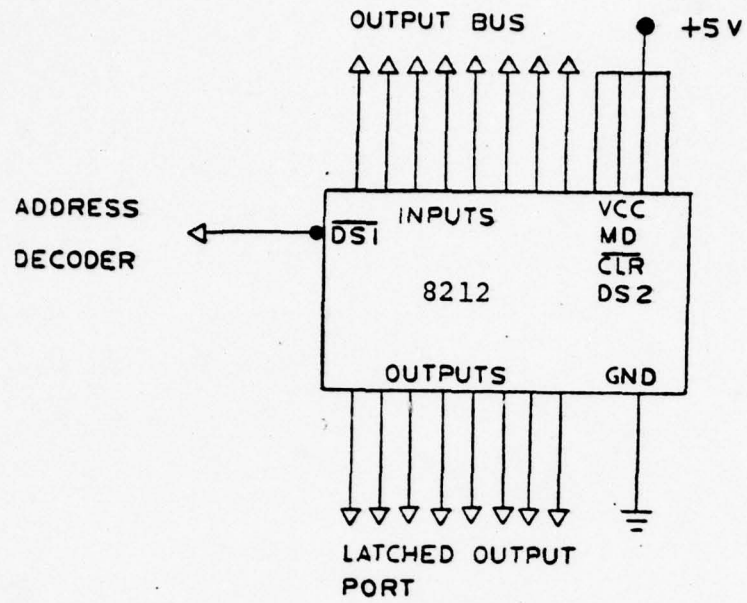
TYPICAL INPUT PORT

FIGURE 16



ADDRESS DECODER

FIGURE 17



OUTPUT PORT WITH TIMING

FIGURE 18

SEVEN SEGMENT OUT PORT

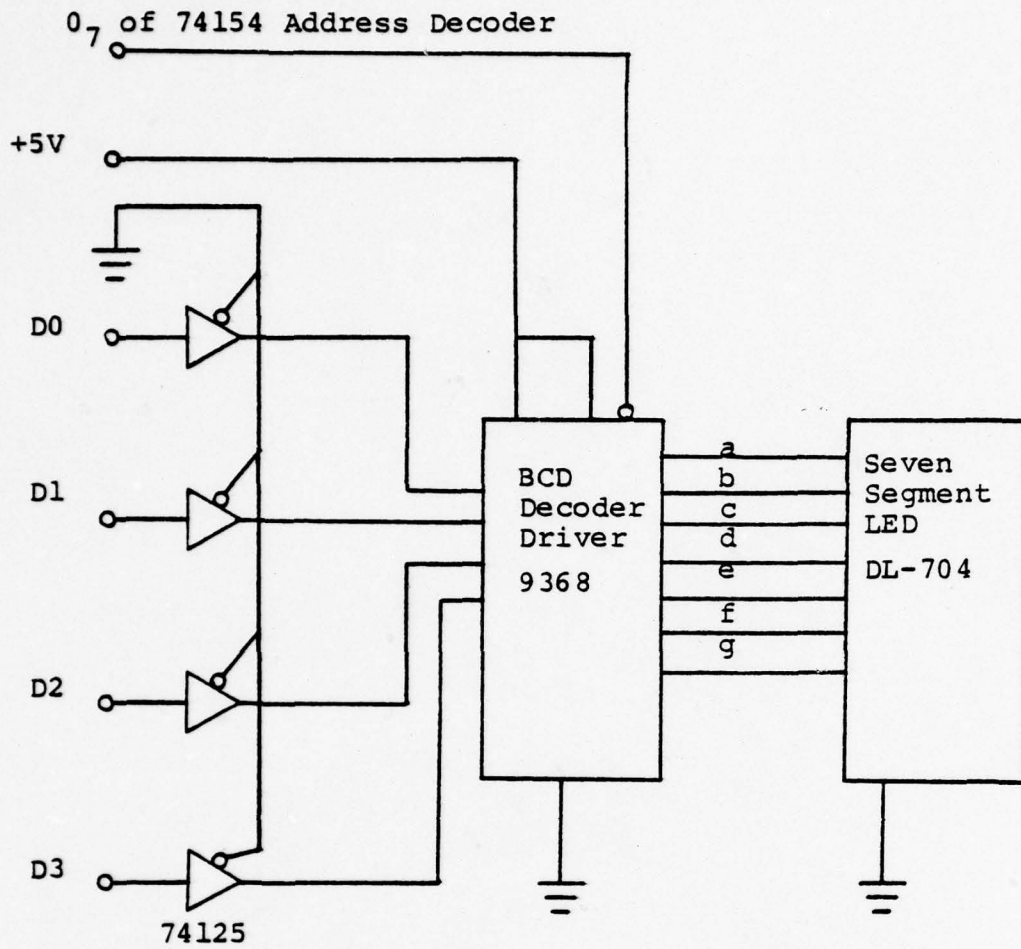


FIGURE 19

by selecting another port, usually port zero, which is non-existent.

Each input and output port has its own unambiguous address. Some of the possible addresses decoded by the 74154 decoder are unused, such as address zero.

The line printer that must be driven by the controller has an "RS-232 like" interface [Ref. 11]. Therefore, four RS-232 line drivers and receivers were provided on micro-computer port 1 (figure 20). The addition of the line drivers added the additional requirement for a plus and minus 12 volt power supply [Ref. 3].

The keyboard and encoder circuitry were built on a separate PC board and connected to the controller board by a 10 wire cable (figure 21). The keyboard is a 4 row by 4 column configuration with three additional user defined switches which are connected to the interrupt line (\overline{INT}), the reset line (\overline{RST}) and the est 1 (T1) lines. When a key is depressed the encoder provides the 4 bit hexadecimal code and a low level on the test 0 (T0) line to signify to the microcomputer that data is available from the keyboard. To prevent the \overline{RST} , \overline{INT} and T1 inputs from floating it was found that pullup resistors were required on the controller PC board.

The artwork for the printed circuit board was done on a 2-to-one scale and then photographically reduced for making the board (Appendix E).

RS-232 INPUT/OUTPUT PORTS

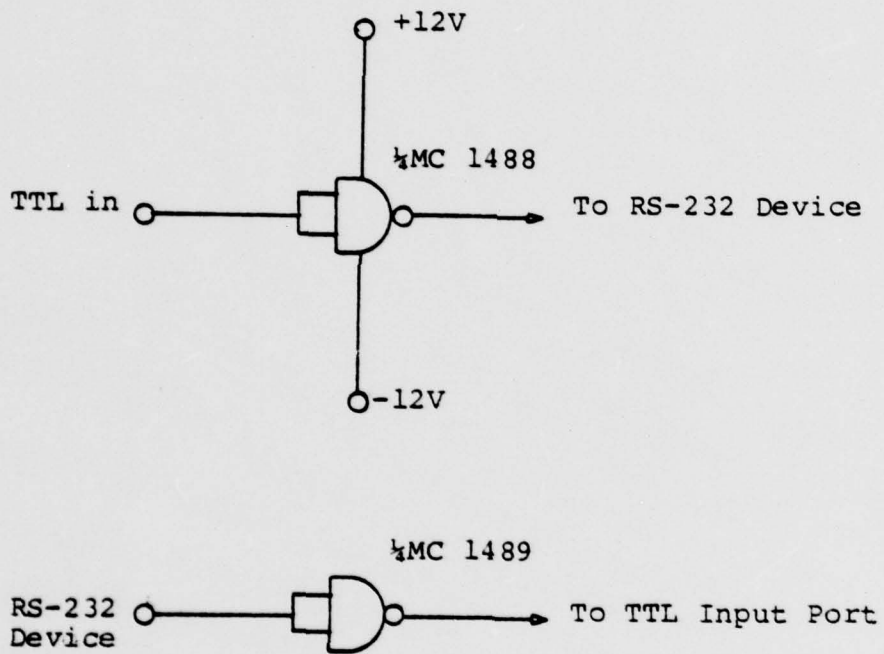


FIGURE 20

KEYBOARD ENCODER

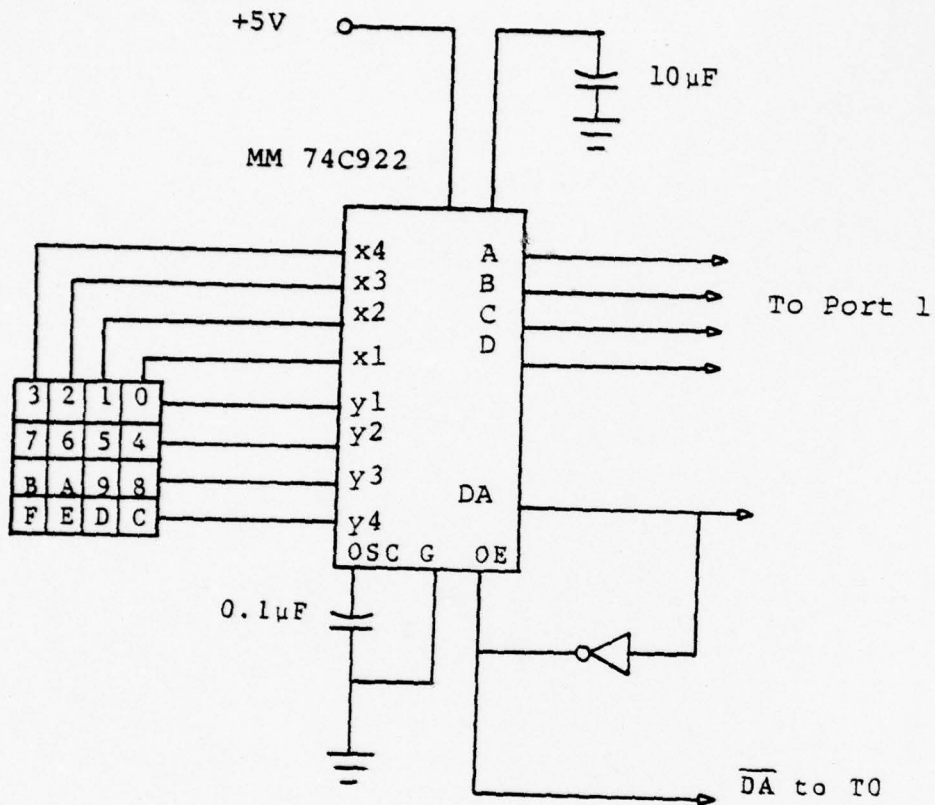


FIGURE 21

III. SOFTWARE DEVELOPMENT

A. DEVELOPMENT TOOLS

The software development was a "low level" process, in that all programming was done in machine language on the Prompt-48. Because of the number of different input and output routines required, it was necessary to develop a method whereby the Prompt-48 could be used in an emulate mode. This was accomplished by the design of a 50 pin edge connector adapter (Appendix B). The PC board containing all of the controller electronics, except the "stand alone" 8748 chip, is installed into the adapter [Ref. 12]. This configuration allows hardware testing and software development to be conducted at real time speed. This emulation capability was essential for development of the serial data output routines.

For a detailed description of the Prompt-48 see reference 12. The important capabilities provided by the Prompt-48 are as follows:

- 1) An eight-character display is used to display register and computer status.
- 2) A 1 K Byte read/write memory is used in place of the 1 K Byte EPROM of the 8748 chip.
- 3) The programming socket can be used to program the 8748 EPROM or to retrieve a program already in an EPROM.

- 4) The buses and ports of the Prompt-48 can be expanded with external circuitry.
- 5) A quite powerful monitor is installed in a 4 K Byte ROM within the Prompt-48.

Appendix C is a summary of the Prompt-48 monitor commands. The use of all commands is well documented in reference 12 except for the use of "Access Codes". The use of Access Codes is an artificiality imposed by hardware constraints while in the Prompt environment. Because of these constraints, it is necessary for the user to specify access codes to allow data flow into or out of the Prompt. For example, to read data into the Prompt requires the user to have selected access code 1 before the attempted read operation. To output data from the Prompt requires access code 0. Consequently, if a program is being debugged which requires an input and an output operation, the user must set breakpoints in the program solely for the purpose of changing access codes before program execution can resume.

The use of breakpoints results in a further complication if the program being tested involves a timing loop. In order for the Prompt to stop execution at breakpoints, it must stop execution after every instruction, jump to the monitor, and check to see if a breakpoint has been reached. This causes the program to proceed about five times slower.

This difficulty can be overcome by designing test programs which are used only for debugging purposes and that can be executed in their entirety without an access code

change. The program under test is thus executed without breakpoints and at full speed. This technique was used extensively while debugging the serial data output subroutines.

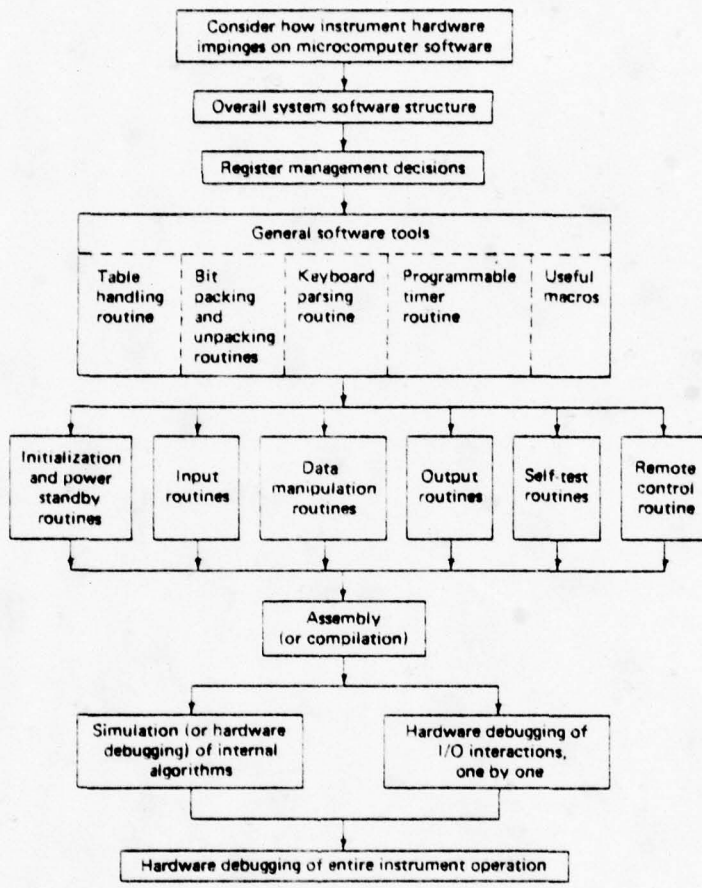
B. OVERALL SYSTEM SOFTWARE STRUCTURE

Figure 22 shows the general method used during the software development [Ref. 7]. Appendix D includes all software documentation. It was decided to rely heavily on the use of subroutines. The subroutines were developed first and stored generally at the higher memory locations. Then the individual programs which the operator must be able to select were developed. The executive program enables the user to select the program he wishes to use. The executive program was written so that more programs could be added as they were developed. Figure 23 is the flow chart for the executive program. It is entered by performing a reset. This happens automatically on power-up, or by depressing the reset key (RST) on the keyboard.

A specific program is entered by the following key sequence from the keyboard:

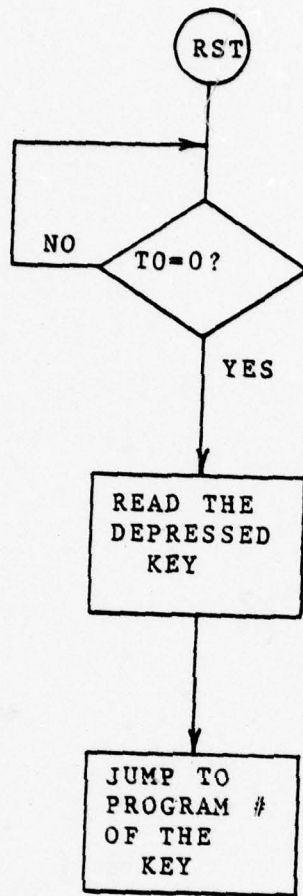
- 1) RST: Reset insures that the microcomputer is in the executive program.
- 2) Any one of keys 0 through 4: This causes the microcomputer to enter program number 0 through 4 respectively.

Since the operation of the programs cannot be understood without first understanding the operation of the subroutines,



Instrument software development.

FIGURE 22



FLOW CHART FOR EXECUTIVE PROGRAM

FIGURE 23

and since the subroutines were developed first, a detailed discussion of the subroutines will be next followed by descriptions of the programs. Flow charts were used in developing the more difficult programs and subroutines.

C. SUBROUTINES

1. Subroutine DELAY

One of the first tests performed was to output square waves from the RS-232 port to check its operation. To generate the square waves subroutine DELAY was written. DELAY is a simple routine consisting of a loop within a loop. The length of the delay is determined by the value in each of the loop counters, which is a parameter that is established before the subroutine call. The documentation for this and all subroutines clearly shows what parameters are to be passed to the subroutine and how they are to be passed.

2. Subroutine ASCII Output

Subroutine ASCII Output (ASCO) was the next and probably most difficult subroutine to be debugged. It receives an ASCII character in its 8 bit form and outputs it serially to the RS-232 port. Since the RS-232 device, in this case the line printer, requires "no parity", the subroutine outputs bits 0 through 7 of the ASCII code and a "1" as the parity bit. Figure 5 shows the timing for a typical ASCII character sent to the printer.

Appendix D fully explains how the subroutine works. Not described there is the process used to choose the value

to be passed to the loop counters. Approximate numbers were chosen based on previous tests for making square waves. The loop counts required for proper operation were found by trial and error through use of the Prompt-48 in its emulate mode. Subroutine ASCO also contains a trap loop which stops further execution if "Request Next Character" (RNC) from the printer is low indicating that the printer is not ready for the next character.

3. Subroutine Binary Coded Decimal Output

Subroutine Binary Coded Decimal Output (BDCO) accepts a BCD number in the range of 0 to 255, looks up the ASCII equivalent of each digit and calls ASCO for each digit. Since each sample consists of a three digit number representing magnitude and a three digit number for phase, the subroutine also provides a space after each three digits to separate the magnitude and phase.

4. Binary To Binary Coded Decimal Conversion

Subroutine Binary to Binary Coded Decimal Conversion (BCDC) receives an eight bit binary number representing either a magnitude or phase and converts it to BCD. It also provides the BCD number in the proper registers for an immediate call to subroutine BCDO.

5. Teletype Output Routines

A very similar set of subroutines were written to output data to the model 33 teletype. Subroutines Teletype Output (TTYO) and Baud Output (BAUDO) differ in that the serial output bit rate is greatly reduced, the parity bit

is always a "0", and the characters are output from a different port. The principles of operation, however, are quite similar to those described above.

The heavy use of subroutines proved very beneficial throughout the project development. For example, when a new test program was required to test the serial interface with the printer, it was relatively easy to write a program consisting of just a few instructions and several calls to subroutines. Program number 2 is just such a program which was left as a permanent part of the software and was used many times to verify proper system operation.

D. KEYBOARD SELECTABLE PROGRAMS

At the time of this writing there are five testing or operating programs that are operator selectable from the keyboard. These five programs are named PROG 0 through 4 in the software. A specific program is selected by depressing the reset (RST) key followed by the key number of the desired program.

1. Program 0

Program 0 is a test program which enables the operator to test the lower four bits of all latched output ports. The upper four bits are all ones in this test. For example, suppose that the operator wishes to test the 115 VAC solid state relay attached to bit 0 of output port 6. To prevent turning off the other bits of the port which are driving the 20 mA current loop and holding the start line of the tape

recorder high, the test word sent to the port should have ones in all positions except possibly the bit 0 position which is under test. A proper key sequence to test the 115 VAC relay is therefore:

<u>KEY</u>	<u>RESULT</u>
RST	Ensures that the computer is in the executive program.
0	Selects program 0 for testing the ports.
F	Turns on all bits of all latched ports. Note the FF displayed on the LED display and that the 115 VAC relay is energized.
E	All bits remain high except bit 0. Note that the 115 VAC relay is deenergized.

2. Program 1

Program 1 is a test program which uses subroutine delay to generate square waves out of the RS-232 port. The period of the square wave is determined by the 8 bit number inserted into the loop counters of subroutine delay. The 8 bit number is inserted as two hexadecimal numbers by two key strokes. For example, the following key sequence will generate an approximately 500 Hz square wave:

<u>KEY</u>	<u>RESULT</u>
RST	Ensures that the computer is in the executive program.
1	Selects program 1.
0	Sets upper four bits of counters to 0.

A Sets the lower four bits to a hexadecimal
 A, so that each loop counter is set to
 0A H
 0AH.

The square wave may be observed at the RS-232 output port bit 0 or can be made audible by closing switch SW-1 on the front of the controller PC board. Notice, however, that while an audible tone is a quick and convenient test to verify proper system operation, that the square wave is severely distorted by the presence of the speaker. This distortion is enough to prevent proper operation when transmitting to an RS-232 device such as the line printer.

3. Program 2

Program 2 is a test program used to verify proper operation of the software baud rate generation. It also verifies the other important aspect of serial data transmission, that of recognizing the presence or absence of a signal on the RNC line from the printer. The RNC signal can be disabled to observe the effect caused by its absence by opening switch SW-2 on the front of the controller PC board. The key sequence used to conduct this test is:

<u>KEY</u>	<u>RESULT</u>
RST	Enters executive program.
2	Selects program 2.

Figure 24 shows the effect of the absence of the RNC signal from the printer. Data is being transmitted during times when the printer cannot print. But, because the printer contains an input buffer register no characters are lost.

TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
TEST PROG#3
EST PROG#3TEST PROG#3TEST PROG#3TEST PROG#3TE
ST PROG#3TEST PROG#3TEST PROG#3TEST PROG#3TEST
PROG#3TEST PROG#3TEST PROG#3TEST PROG#3TEST PR
OG#3TEST PROG#3TEST PROG#3TEST PROG#3TEST PRO
G#3TEST PROG#3TEST PROG#3TEST PROG#3TEST P
ROG#3TEST PROG#3TEST PROG#3TEST PROG#3TEST
PROG#3TEST PROG#3TEST PROG#3TEST PROG#3TES
PROG#3TEST PROG#3TEST PROG#3TEST PROOG#3TE
ST PROG#3TEST PROG#3TEST PROG#3TEST PROG#3
TEST PROG#3TEST PROG#3TEST PROG#3TEST PROG#
3TEST PROG#3TEST PROG#3TEST PROG#3TEST PROG
#3TEST PROG#3TEST PROG#3TEST PROG#3TEST PRO
G#3TEST PROG#3T

[RNC DISABLED]

FIGURE 24

This may not be the case for another RS-232 device which does not contain such a buffer.

4. Program 3

Program 3 is a data acquisition program which records data from the ADCs onto the cassette tape while simultaneously providing a hard copy on the line printer. The program is selected by the key sequence: RST, 3.

As can be seen in Appendix D, the program loops while sampling test point 0 (TO). When a sample is to be taken the revolution counter sends a 0 to the Start Conversion line of the ADCs. The microcomputer sees a 0 on TO and starts a 1 millisecond delay. This gives the ADCs time to complete the analog to digital conversion and provides additional time for the data to stabilize at the inputs to the tape recorder. After the 1 millisecond delay the microcomputer puts out a 1 millisecond active low pulse from bit 2 of output port 6. This pulse starts the record cycle of the cassette recorder. Since the recorder requires approximately 300 milliseconds to complete the recording of the 16 bits of data, the speed of the mechanical scanner was adjusted to provide a sampling rate of about 2 Hz. That this sampling rate is satisfactory can be easily verified by the status LED on the tape recorder interface board (Fig. 6). The recorder is ready to achieve another sample when the LED is on.

If it is desired to obtain only a hard copy of the data, such as during alignment or calibration testing, then

the mechanical scan speed can be greatly increased within the capability of the model 40 printer. Also, the RNC signal must be disabled to prevent the RNC signal from interfering with the start conversion pulse to the ADCs.

5. Program 4

Program 4 is the other data acquisition program. It differs only slightly from program 3. It provides hard copy on the model 33 TTY while making a paper tape and/or cassette recording. In this program the serial data is provided at a rate of 110 Baud/second out of bit 4 of port 6 which drives the 20 mA current loop for the TTY.

In this case the limiting factor which determines the sampling rate is approximately twice the length of time it takes to execute a carriage return and line feed. The carriage return and line feed are executed on the rising edge of the revolution counter output which is applied to test point TO. The TTY must complete the carriage return and line feed before the next falling edge of the revolution counter which initiates the next sample.

IV. SYSTEM TESTING

A. INPUT AND OUTPUT PORT TESTING

1. Parallel Input Port Testing

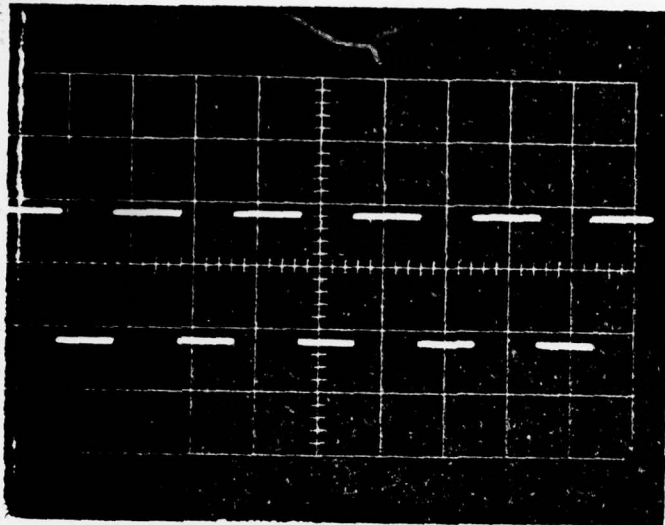
Each parallel input port was tested individually to verify that it was being addressed properly and that its input and output bits were in the proper order. The test data word was provided from a laboratory DIGIDESIGNER. A simple program was written in the Prompt-48 and single stepped to verify that the port was enabled at the proper time and that the execution of the read instruction resulted in the transfer of the proper data to the microcomputer accumulator. One port failed for no apparent reason. The 74125 IC chip was replaced and the test was satisfactory.

2. Latched Output Port Testing

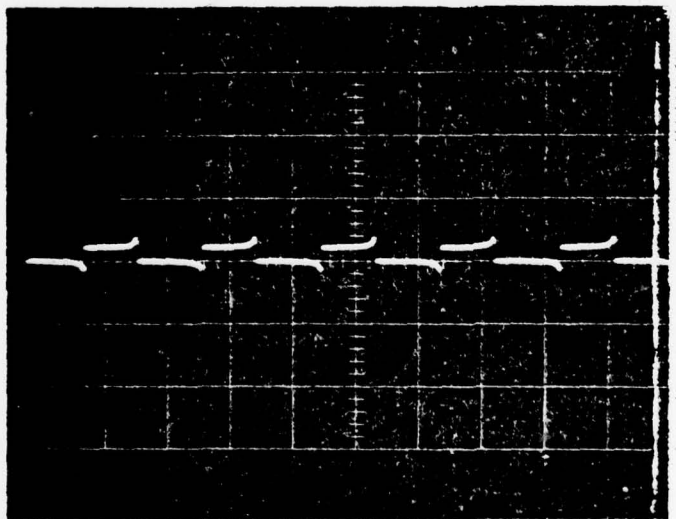
Each output was tested to the extent possible by the use of program 0. This proved to be a sufficient amount of testing because no output problems were experienced.

3. Serial Output Testing

The RS-232 serial output port was tested in three different ways. First a simple program written in the Prompt-48 was single stepped while the voltage at the port was monitored to ensure that it cycled between plus and minus 12 volts. Then program 1 was used to generate square waves. The square waves were checked for rise time and overshoot (Fig. 25) and were found to be in accordance with



SQUARE WAVES FROM RS-232 PORT WITHOUT
SPEAKER CONNECTED



Note attenuation
and distortion
caused by the
speaker.

SQUARE WAVES FROM RS-232 PORT WITH
SPEAKER CONNECTED

FIGURE 25

reference 11. Figure 26 also shows the attenuation and distortion caused by connecting the speaker to the port.

B. OVERALL SYSTEM TESTING

The objectives of the system testing program were:

(1) to verify that the 16 bits representing a magnitude and phase are properly converted to decimal; (2) to verify that the paper tape and TTY printout agree with the actual data present at the ADC outputs, (3) to verify that the hard copy produced on the model 40 line printer agrees with the actual data at the ADCs, and finally, (4) to verify that the paper tape or cassette tape agrees with the hard copy produced at the time the recording was made.

The output of the ADCs was simulated by the data switches on a DIGIDESIGNER so that known binary values could be processed. The sample strobe from the revolution counter was simulated by a pulse generator set at a frequency of 1 Hz. While sampling at a 1 Hz rate the following binary numbers representing magnitude and phase were provided to the system:

00000000	00000000	=	000	000
10001000	10001000	=	017	017
01000100	01000100	=	034	034
00100010	00100010	=	068	068
00010001	00010001	=	136	136
11111111	11111111	=	255	255

Each binary number was held constant for several samples so that a significant amount of data would be

generated. The binary numbers were chosen to test each bit of each input port and to cover the whole dynamic range of the ADCs.

Figure 26 shows the real time copy of the data printed by the TTY compared to the data read back on the same TTY and read back on the PDP-11. Figure 27 shows the real time copy made by the line printer. Because the cassette recorder was returned to the factory for repairs, verification of a cassette recording was not obtained. This type of test was conducted several times, and after each significant hardware or software change.

C. MECHANICAL ALIGNMENT TESTING

Because of the relatively short wavelength of the acoustic wave being sampled it is very important that the samples be taken at the proper times so that the columns of samples are vertical and straight. The alignment is mechanically adjustable by the position of the left and right limit switches of the mechanical scanner. This alignment can then be checked by disabling the vertical stepping motor and scanning back and forth across any target and noticing the agreement in sample values from one scan to the next. This alignment test should be conducted periodically and after any adjustments to the mechanical scanning mechanism.

D. OPERATING PROCEDURES

System operating instructions are provided in Appendix F for laboratory use and should enable personnel to operate

the equipment without requiring a detailed understanding of its design and construction.

ADDR	KEY CODE	LABEL	MNEMONIC	COMMENT
------	----------	-------	----------	---------

V. CONCLUSIONS

The microcomputer based controller described in this thesis met all of the original design objectives, and has become a useful addition to the laboratory. Because several input and output ports and some of the program memory are not used, improvements and changes to the system can be made.

The flexibility provided by a microcomputer based design has already been demonstrated by several software changes which resulted in improved system performance and operational convenience. Several interface problems which might have been solved by hardware changes were more easily solved in software. The ability to take manual samples and the provision of a real time hard copy of the recorded data were invaluable capabilities which were used many times during alignment and calibration tests of the overall system. Clearly, the controller described here could easily be adapted to many other control or interface applications.

The flexibility of the system is only slightly reduced by the requirement of performing all software development in machine language. It was found that as the author became more familiar with the instruction set and the use of the Prompt-48 that programs could be written and debugged quite expeditiously. The fact that the programs required very few mathematical manipulations further added to the ease of programming. Furthermore, for this controller type

application where program memory was limited, machine language programming clearly was the most cost effective choice in terms of the required development system and in terms of program memory utilization.

APPENDIX A

BY MNEMONIC

• ADD A, R0	68	DEC R5	CD	JNT1 addr	46	ORL A, R0	48
R1	69	R6	CE	JNZ addr	96	R1	49
R2	6A	R7	CF	JTF addr	16	R2	4A
R3	6B			JTO addr	36	R3	4B
R4	6C			JTI addr	56	R4	4C
R5	6D	DIS I	15	JZ addr	C6	R5	4D
R6	6E	DIS TCNTI	35			R6	4E
R7	6F					R7	4F
• ADD A, @R0	80	DJNZ R0, addr	E8	MOV A, #data	23	ORL BUS, #data	88
@R1	81	R1, addr	E9	MOV A, PSW	C7	P1, #data	89
• ADD A, #data	03	R2, addr	EA	MOV A, R0	F6	P2, #data	8A
		R3, addr	EB	R1	F9	ORLD P4, A	8C
• ADDC A, R0	78	R4, addr	EC	R2	FA	P5, A	8D
R1	79	R5, addr	ED	R3	FB	P6, A	8E
R2	7A	R6, addr	EE	R4	FC	ORLD P7, A	8F
R3	7B	R7, addr	EF	R5	FD		
R4	7C			R6	FE		
R5	7D	EN I	05	R7	FF	OUTL BUS, A	02
R6	7E	EN TCNTI	25	MOV A, @R0	F0	P1, A	39
R7	7F	ENTO CLK	75	@R1	F1	P2, A	3A
• ADDC A, @R0	70			MOV A, T	42		
@R1	71	INS A, BUS	08			RET	83
• ADDC A, #data	13	IN A, P1	09	• MOV PSW, A	07	RETR	93
		IN A, P2	0A	MOV R0, A	A8		
ANL A, R0	58			R1, A	A9	• RLA	E7
R1	59	INC A	17	R2, A	AA	• RLC A	F7
R2	5A	INC R0	18	R3, A	AB	• RRA A	77
R3	5B	INC R1	19	R4, A	AC	• RRC A	67
R4	5C	INC R2	1A	R5, A	AD		
R5	5D	INC R3	1B	R6, A	AE	SEL MB0	E5
R6	5E	INC R4	1C	R7, A	AF	SEL MB1	F5
R7	5F	INC R5	1D			SEL MB0	C5
ANL A, @R0	50	INC R6	1E	MOV R0, #data	88	SEL RB1	05
ANL A, @R1	51	INC R7	1F	R1, #data	89		
ANL A, #data	53	INC @R0	10	R2, #data	8A	STOP TCNT	65
ANL BUS, #data	98	INC @R1	11	R3, #data	8B	STRT CNT	45
P1, #data	99			R4, #data	8C	STRT T	55
P2, #data	9A			R5, #data	8D	SWAP A	47
				R6, #data	8E		
CALL 0addr	14	JB0 addr	12	R7, #data	8F	XCH A, R0	28
1addr	34	JB1 addr	32			R1	29
2addr	54	JB2 addr	52	MOV @R0, A	A0	R2	2A
3addr	74	JB3 addr	72	MOV @R1, A	A1	R3	2B
4addr	94	JB4 addr	94	MOV @R0, #data	80	R4	2C
5addr	B4	JB5 addr	B2	@R1, #data		R5	2D
6addr	D4	JB6 addr	D2			R6	2E
7addr	F4	JB7 addr	F2	MOV T, A	62	R7	2F
				MOV D A, P4	0C	XCH A, @R0	20
CLR A	27	JC addr	F8	P5	0D	XCH A, @R1	21
CLR C	97	JF0 addr	86	P6	0E		
CLR F0	85	JF1 addr		P7	0F	XCHD A, @R0	30
CLR F1	A5	JMP 0addr	04	P5, A	3D	@R1	31
CPL A	37	1addr	24	P6, A	3E		
• CPL C	A7	2addr	44	P7, A	3F	XRL A, R0	08
CPL F0	95	3addr	64			R1	09
CPL F1	85	4addr	84	MOV P A, #A	A3	R2	DA
		5addr	A4	MOV P3 A, #A	E3	R3	DB
• DA A	57	6addr	C4	MOV X A, @R0	80	R4	DC
DEC A	07	7addr	E4	@R1	81	R5	DD
DEC R0	C8	JMPP #A	83	MOV X @R0 A	90	R6	DE
R1	C9	JNC addr	E8	@R1, A	91	R7	DF
R2	CA	JNI addr	98			XRL A, @R0	00
R3	CB	JNTD addr	76	NOP	00	@R1	01
R4	CC						

• CARRY FLAG AFFECTED

All mnemonics copyright © 1976, 1977, 1978 Intel Corporation

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
------	----------	-------	----------	---------

APPENDIX B

WIRING DATA FOR 8748 BOX

<u>8748 Pin #</u>	<u>NAME</u>	<u>Edge Connector Pin #/Connection</u>
1	TO	14
2	XTAL 1	-- crystal & cap. to gnd
3	XTAL 2	-- crystal & cap. to gnd
4	$\overline{\text{RESET}}$	16
5	$\overline{\text{SS}}$	-- NC
6	$\overline{\text{INT}}$	49
7	EA	-- NC
8	$\overline{\text{RD}}$	9
9	$\overline{\text{PSEN}}$	15
10	$\overline{\text{WR}}$	11
11	ALE	13
12	DO (BUS)	17
13	D1	21
14	D2	25
15	D3	29
16	D4	31
17	D5	27
18	D6	23
19	D7	19
20	Vss	GND - 46
21	P20 (PORT 2)	7
22	P21	5

ADDRESS	DATA	LABEL	MNEMONIC	COMMENT

23	P22	3
24	P23	1
25	PROG	NC
26	VDD	+5V - 34
27	P10 (PORT 1)	18
28	P11	20
29	P12	22
30	P13	24
31	P14	26
32	P15	28
33	P16	30
34	P17	32
35	P24 (PORT 2)	4
36	P25	6
37	P26	8
38	P27	10
39	T1	12
40	Vcc	+5V - 34

EXTERNAL CONNECTIONS TO BOX

34	+5V
35 - 43	NC
44	GND
50	+5V
46	GND

Parts List For 8748 Box

<u>Item</u>	<u>QTY</u>
6" x 5" x 4" AL Box	1
Banana Jacks	13
50 Pin Edge Connector	1
6 MHz μ p crystal	1
20 pf Cap	2
1 μ fd Cap	1
4 - 40 x $\frac{1}{2}$ " hardware	6
ground lugs	2
40 pin Dip socket	1

WIRING DATA FOR EXTENDER BOX

8748 Pin Name	I/O Ports & Edge Connector pin #	
Bus - 0	17	LSB
1	21	
2	25	
3	29	
4	31	
5	27	
6	23	
7	19	MSB
Port 1 - 0	18	LSB
1	20	
2	22	
3	24	
4	26	
5	28	
6	30	
7	32	MSB
Port 2 - 0	7	LSB
1	5	
2	3	
3	1	
4	4	
5	6	
6	8	
7	10	MSB
+ALE	13	
+T0	14	
+T1	12	
-INT	49	
-PSEN	15	
-RD	9	
-WR	11	
-POWR	33	
-PROG	2	
-RESET	16	
GND	45, 46, 47, 48	
	34 - +5V	
	50 - +5V	
	44 - GND	
Unused:	35, 36, 37, 38, 39, 40, 41, 42, 43	

Table 5-7. Command List Summary

Command Prompts: "ACCESS=0" and "--"		
Command Key(s)/(Description)	Function Display	Section
[GO]:	"G	5-20
- [NO BREAK]	"Go	5-21
- [WITH BREAK]	"Gb	5-24
- [SINGLE STEP]	"GS	5-24
[EXAMINE/MODIFY]:	"E	5-17
- [PROGRAM MEMORY]	"EP	5-18
- [DATA MEMORY]	"Ed	5-17
- [REGISTER]	"Er	5-15
[2] (Port 2 Map)	"P2 MM"	5-16
[3] (Program PROM — 8741 or 8748)	"Pr 8741	5-53
[3] (Program PROM — 8755, with adapter)	"Pr 8755	5-53
[4] (Byte Search):	"S1	5-25
- [PROGRAM MEMORY]	"SP	5-26
- [DATA MEMORY]	"Sd	5-27
- [REGISTER]	"Sr	5-28
[5] (Word Search):	"S2	5-25
- [PROGRAM MEMORY]	"SP	5-28
- [DATA MEMORY]	"Sd	5-30
- [REGISTER]	"Sr	5-31
[6] (Hexadecimal Arithmetic)	"HE	5-49
[7] (Program PROM — 8748)	"Pr 8748	5-52
[8] (Compare PROM)	"Co	5-54
[9] (Move Memory):	"n	5-32
- [PROGRAM MEMORY]	"nP	5-33
- [DATA MEMORY]	"nd	5-34
- [REGISTER]	"nr	5-35
[A] (Access Mode Select)	"Ac CC"	5-14
[B] (Examine/Modify Breakpoint)	"br	5-23
[C] (Clear Memory):	"C	5-36
- [PROGRAM MEMORY]	"CP	5-37
- [DATA MEMORY]	"Cd	5-38
- [REGISTER]	"Cr	5-39
[D] (Dump Memory):	"d	5-40
- [PROGRAM MEMORY]	"dP	5-41
- [DATA MEMORY]	"dd	5-42
- [REGISTER]	"dr	5-43
[E] (Enter into Memory):	"r	5-44
- [PROGRAM MEMORY]	"rP	5-45
- [DATA MEMORY]	"rd	5-46
- [REGISTER]	"rR	5-47
[F] (Fetch PROM)	"FP	5-55

APPENDIX D

MEMORY MAP

000
Executive Program
014
015
Prog 0
034
035
Prog 1
069
06A
Executive Program
06D
076
Prog 3
0C6
Unused
0F0
Executive Program
0FF
100
Prog 2
135
136
BCDO
150
151
BAUDO
196
Unused
1A0
TTYO
1bA
Unused
200
Prog 4
245
Unused
34F
ADC
35A
35B
BCDC
3A2
Unused
3A6
ASCO
3DC
Unused
3E5
DELAY
3EF

APPENDIX D

REGISTER MANAGEMENT

ASCO: R7

Prog 0: No working registers except the Accumulator

DELAY: R4,5,6,7

Prog 1: R4,5,6,7

Prog 3: R1,2,3,4,5,6,7

BCDC: R3,4,5,6,7

BCDO: R1,2,3,4,5,6,7

AXCO: R2,3,4,5,6,7

Executive Program: Uses only the Accumulator

Prog 2: R2,3,4,5,6,7

TTYO: R1,2,3,4,5,6,7

BAUDO: R2,3,4,5,6,7

Prog 4: R1,2,3,4,5,6,7

APPENDIX D

PROGRAM EXECUTIVE PROGRAM

DATE 23 Feb. 79 PAGE 1 OF 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
000	0409	RST	JMP LOP 1	
009	260D	LOP 1	JNTO RDD1	
00B	0409		JMP LOP 1	
00D	230A	RDD1	MOV A,OA	
00F	3A		OUTL P2,A	
010	08		INS A, Bus	
011	00	LOP 2	NOP	
012	2611		JNTO LOP2	;pressing a key now will ; select a program
014	B3		JMPP@A	;jump to the address of ;the key depressed
0F0	15	PROG 0		
0F1	35	PROG 1		
0F2	6A	PROG 2		
0F3	76	PROG 3		
0F4	6C	PROG 4		;0F5 through 0FF are unused ;and available in the ;event more programs are ;added
06A	2400		JMP PROG 2	;PROG 2 is on page 1
06C	4400		JMP PROG 4	;PROG 4 is on page 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
				;this program tests the ;lower 4 bits of the output ;ports
15	00	PROG 1	NOP	
16	2615		JNTO PROG 1	;loop until key is presse
18	261C	LOP 1	JNTO RDD1	
1A	0418		JMP LOP 1	
1C	230A	RDD1	MOV A,0A	
1E	3A		OUTL P2,A	;select port 0A
1F	08		INS A, Bus	;read the depressed key
20	02		OUTL BUS, A	;output the depressed key
21	2307		MOV A, 07	
23	3A		OUTL P2,A	;select port 7
24	2300		MOV A,00	
26	3A		OUTL P2,A	;deselect port 7
27	2306		MOV A, 06	
29	3A		OUTL P2,A	;select port 6
2A	2300		MOV A, 00	
2C	3A		OUTL P2,A	;deselect port 6
2D	2305		MOV A,05	
2F	3A		OUTL P2,A	;select port 5
30	2300		MOV A, 00	
32	3A		OUTL P2,A	;deselect port 5
33	0415		JMP PROG 1	

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
3E5	FF	DELAY	MOV A,R7	;the length of delay
	AC		MOV R4,A	;is determined by the
	FE	LOP 2	MOV A,R6	;numbers placed in
	AD		MOV R5,A	;R6 and R7 before Delay
	00	LOP 1	NOP	;is called
	EDE9		DJNZR5,LOP 1	
	ECE7		DJNZR4,LOP2	
3EE	83		RET	

PROGRAM NUMBER 1

DATE 24 Feb 79 PAGE 1 OF 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
035	00	PROG 1	NOP	;this test program ;generates square waves
036	2635		JNTO,PROG 1	;for testing.
038	2636	LOP 1	JNTO,RDD 1	
03A	0438		JMP,LOP 1	
036	230A	RDD 1	MOVA,OA	;enter MS digit of ;delay
03E	3A		OUTL P2,A	
03F	08		INSA, BUS	
040	530F		ANL A,OF	;mark off unwanted ;ones
042	47		SWAP A	
043	AF	LOP 2	MOV R7,A	
044	2643		INTO,LOP 2	
046	230A	LOP 4	MOV A,OA	
048	2646		JNTO, LOP 4	
04A	264E	LOP 5	JNTORDD2	
04C	044A		JMP LOP 5	
04E	3A	RDD 2	OUTL P2,A	
04F	08		INSA, BUS	;enter LS digit of delay
050	530F		ANLA,OF	
052	6F		ADD A,R7	
053	AE		MOV R6,A	
054	AF		MOV R7,A	

PROGRAM NUMBER 1

DATE 24 Feb 79 PAGE 2 OF 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
055	02		OUTL BUS,A	
056	2307		MOV A,07	
058	3A		OUTL P2,A	
059	2300		MOV A,00	
05B	3A		OUTL P2,A	
05C	2310	LOP 3	MOV A,10	
05E	3A		OUTL P1,A	
05F	74E5		CALL DELAY	
061	2300		MOV A, 00	
063	39		OUTL P1,A	
064	74E5		CALL DELAY	
066	4635		INT 1,PROG 1	
068	045C		JMP, LOP 3	

PROGRAM SUBROUTINE BCDC

DATE 12 Mar 79 PAGE 1 OF 3

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
				BIN # is in R7 BCD Number will be in RS&6
35A	27	BCDC	CLR A	
5B	AD		MOV R5,A	;clear BCD registers
5C	AE		MOV R6,A	
5D	AB		MOV #3,A	
5E	FF		MOV A,R7	
5F	1270		JBO, ADD0	;Test each bit.
61	3278	J1	JB1, ADD1	
63	527E	J2	JB2, ADD2	
65	7284	J3	JB3, ADD3	
67	928A	J4	JB4, ADD4	
69	8290	J5	JB5, ADD5	
6B	D296	J6	JB6, ADD6	
6D	F29C	J7	JB7, ADD7	
6F	83		RFT	
70	BB00	ADD0	MOV R3,00	;Add 1
72	BC01		MOV R4,01	
74	744F		CALL ADC	
76	6461		JMP J1	
		ADD1		;Add 2
78	8C02		MOV R4,02	
7A	744F		CALL ADC	
7C	6463		JMP J2	

PROGRAM SUBROUTINE BCD

DATE 12 Mar 79 PAGE 2 OF 3

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
		ADD 2		
7E	8C04		MOV R4,04	;Add4
80	744F		CALL ADC	
82	6465		JMP J3	
		ADD 3		
84	8C0B		MOV R4,08	;Add 08
86	744F		CALL ADC	
88	6467		JMP J4	
8A	BC16	ADD 4	MOV R4,16	;Add 16
8C	744F		CALL ADC	
8E	6469		JMP J5	
90	BC32	ADD 5	MOV R4,32	;Add 32
92	744F		CALL ADC	
94	646B		JMP J6	
96	BC64	ADD 6	MOV R4,64	;Add 64
98	744F		CALL ADC	
9A	646D		JMP J7	
9C	BB01	ADD 7	MOV R3,01	
9E	BC28		MOV R4,28	
3A0	744f		CALL ADC	
				;BCD result is in R5&6
3A2	83		RET	

PROGRAM SUBROUTINE BCDC

DATE 12 Mar 79 PAGE 3 OF 3

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
34F	97	ADC	CLRC	
350	FE		MOV A,R6	
51	7C		ADDC A,R4	
52	57		DA A	
53	AE		MOV R6,A	
54	FD		MOV A,R5	
55	7B		ADDC A,R3	
56	57		DAA	
57	AD		MOV R5,A	
58	FF		MOV A,R7	
59	83		RET	

PROGRAM SUBROUTINE BCDO

DATE 20 Mar 79 PAGE 1 OF 1

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
136	FE	BCDO	MOV A,R6	
137	A9		MOV R1,A	
138	FD		MOV A,R5	;This converts BCD to
39	E3		MOV P3,A,@A	;ASCII, then outputs
3A	AB		MOV R2,A	;start with BCD #
3B	74A6		CALL ASCO	;in 5,6
3D	F9		MOV A,R1	
3E	47		SWAP A	;Put next BCD digit ;in Lower 4 bits
3F	530F		ANL A,0F	;mask off upper 4 bits
41	E3		MOV P3A,@A	;Loop up ASCII in Table
42	AB		MOV R3,A	
43	74A6		CALL ASCO	
45	F9		MOV A,R1	
46	530F		ANL A,0F	;This leaves last digit
48	E3		MOV P3A,@A	;in lower 4 bits
49	AB		MOV R3,A	
4A	74A6		CALL ASCO	
4C	BB20		MOV R3,20H	;Leaves space
4E	74A6		CALL ASCO	
50	83		RET	
300	3D	TABL		;ASCII loop up ;Table.
309	39			

PROGRAM SUBROUTINE ASCO

DATE 10 Mar 79 PAGE 1 OF 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
3A6	2300	ASCO	MOV A,00	
3A8	39		OUTL P1,A	;Output start bit
3A9	BF03		MOV R7,03	
3AB	BE0F		MOV R6,0F	
3AD	74E5		CALL DELAY	
3AF	BA08		MOV R2,08	;Set count for bits = 7
3B1	FB	LOP0	MOV A,R3	;Put char. into A.
3B2	EAB6		DJNZ R2,LOP 1	
3B4	64D1		JMP END	
3B6	97	LOP1	CLR C	
3B7	67		RRC A	
3B8	AB		MOV R3,A	
3B9	E6C6		JNC LOP 2	
3BB	2310		MOV A,10	
3BD	39		OUTL P1,A	;output "1"
3BE	BF03		MOV R7,03	;set loop count for Delay
3C0	BE0F		MOV R6,0F	;set loop count for Delay
3C2	74E5		CALL DELAY	
3C4	64B1		JMP LOP 0	
3C6	2300	LOP2	MOV A,00	;output "0"
3C8	39		OUTL P1,A	
3C9	BF03		MOV R7,03	;set loop ;count
3CB	BE0F		MOV R6,0F	;for Delay
3CD	74E5		CALL DELAY	
3FF	64B1		JMP LOP0	
3D1	2301	END	MOV A,10	;output stop bit

PROGRAM SUBROUTINE ASCO

DATE 10 Mar 79 PAGE 2 OF 2

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
3D3	39		OUTL P1,A	
3D4	BF0C		MOV R7,0C	;set loop count
306	BE0D		MOV R6,0D	;for Delay
3D8	74ES	LOP3	CALL DELAY	
3DA	46D8		JNT1 LOP3	;Loop if not ready for ;next character.
3DC	93		RET R	

PROGRAM PROG 2 (TEST ASCO)

DATE 11 Mar 79 PAGE 1 OF 1

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
100	BB54	PROG2	MOV R3,54	;T
102	74A6		CALL ASCO	
4	BB45		MOV R3,45	;E
6	74A6		CALL ASCO	
8	BB53		MOV R3,53	;S
A	74A6		CALL ASCO	
C	BB54		MOV R3,54	;T
E	74A6		CALL ASCO	
110	BB20		MOV R3,20	;Sp
	74A6		CALL ASCO	
	BB50		MOV R3,50	;P
	74A6		CALL ASCO	
118	BB52		MOV R3,52	;R
	74A6		CALL ASCO	
	BB4F		MOV R3,4F	'0
11E	74A6		CALL ASCO	
	BB47		MOV R3,47	;0
	74A6		CALL ASCO	
	BB23		MOV R3,23	;#
	74A6		CALL ASCO	
128	BB33		MOV R3,33	;3
12A	74A6		CALL ASCO	
12C	BB0A		MOV R3,0A	LF
12E	74A6		CALL ASCO	
130	BB0D		MOV R3,0D	CR
132	74A6		CALL ASCO	
134	2400		JMP PROG 2	

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
76	00	PROG4	NOP	
77	2676		JNTO, PROG3	
79	2305		MOV A,05	;turns on 115 VAC ;& set start high
7B	02		OUTL BUS,A	
7C	2306		MOV A,06	
7E	3A		OUTL P2,A	
7F	2300		MOV A,00	
81	3A		OUTL P2,A	
82	B808	LOP 2	MOV R0,08H	;8 samples per line
84	00	LOP 1	NOP	
85	5684		JT1, LOP1	;waits for sample strobe ;line to go low.
87	BFOA		MOV R7,0A	
89	BEOA		MOV R6,0A	
8B	74E5		CALL DELAY	
8D	2301		MOV A,01	;115 VAC still on ;but start is low.
8F	02		OUTL BUS,A	
90	2306		MOV A,06	
92	3A		OUTL P2,A	
93	2300		MOV A,00	
95	3A		OUTL P2,A	;latches above at P6
96	BF0A		MOV R7,0A	
98	BEOA		MOV R6,0A	
9A	74E5		CALL DELAY	
9C	2305		MOV A,05	;115 VAC & start on

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
9E	02		OUTL BUS A	
9F	2306		MOV A,06	
A1	3A		OUTL P2,A	
A2	2300		MOV A,00	
A4	3A		OUTL P2,A	;Ends starts pulse
A5	2308		MOV A,08	;select port 8
A7	3A		OUTL P2,A	
A8	08		INSA, BUS	;Read port 8
A9	AF		MOV R7,A	
AA	745A		CALL BCDC	
AC	3436		CALL BCDO	;print magnitude
AE	BB20		MOV R3,20	
B0	74A6		CALL ASCO	
B2	2309		MOV A,09	;Select port 9
B4	3A		OUTL P2,A	
B5	08		INSA BUS	;Read port 9
B6	AF		MOV R7,A	
B7	745A		CALL BCDC	
B9	3436		CALL BCDO	;Print phase
BB	E884		DJNZ Ro,LOP1	
BD	BB0A		MOV R3,0A	
BF	74A6		CALL ASCO	
C1	BB0D		MOV R3,0D	
C3	74A6		CALL ASCO	
C5	0482		JMP LOP2	

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
200	00	PROG4	NOP	
201	2600		JNTO, PROG 4	
03	2305		MOV A,05	
05	02		OUTL BUS,A	;Turns on 115 VAC & sets start high
06	3490		CALL SEL 6	
08	443C		JMP LOP 3	
0A	B808	LOP 2	MOV R0,08	;8 samples per line
0C	00	LOP 1	NOP	
0D	560C		JT1, LOP 1	
0F	BF0A		MOV R7,0A	
211	BE0A		MOV R6,0A	
13	74E5		CALL DELAY	
15	2319		MOV A,19	;115 VAC still on ;but start low.
17	02		OUTL BUS,A	
18	3490		CALL SEL 6	
1A	BF0A		MOV R7,0A	
1C	BE0A		MOV R6,0A	
1E	74E5		CALL DELAY	
20	231D		MOV A,1D	;115 VAC on & start high
22	02		OUTL BUS,A	
23	3490		CALL SEL6	
25	2308		MOV A,08	;select port 8
27	3A		OUTL P2,A	
28	08		INS A BUS	;Read port 8
29	AF		MOV R7,A	

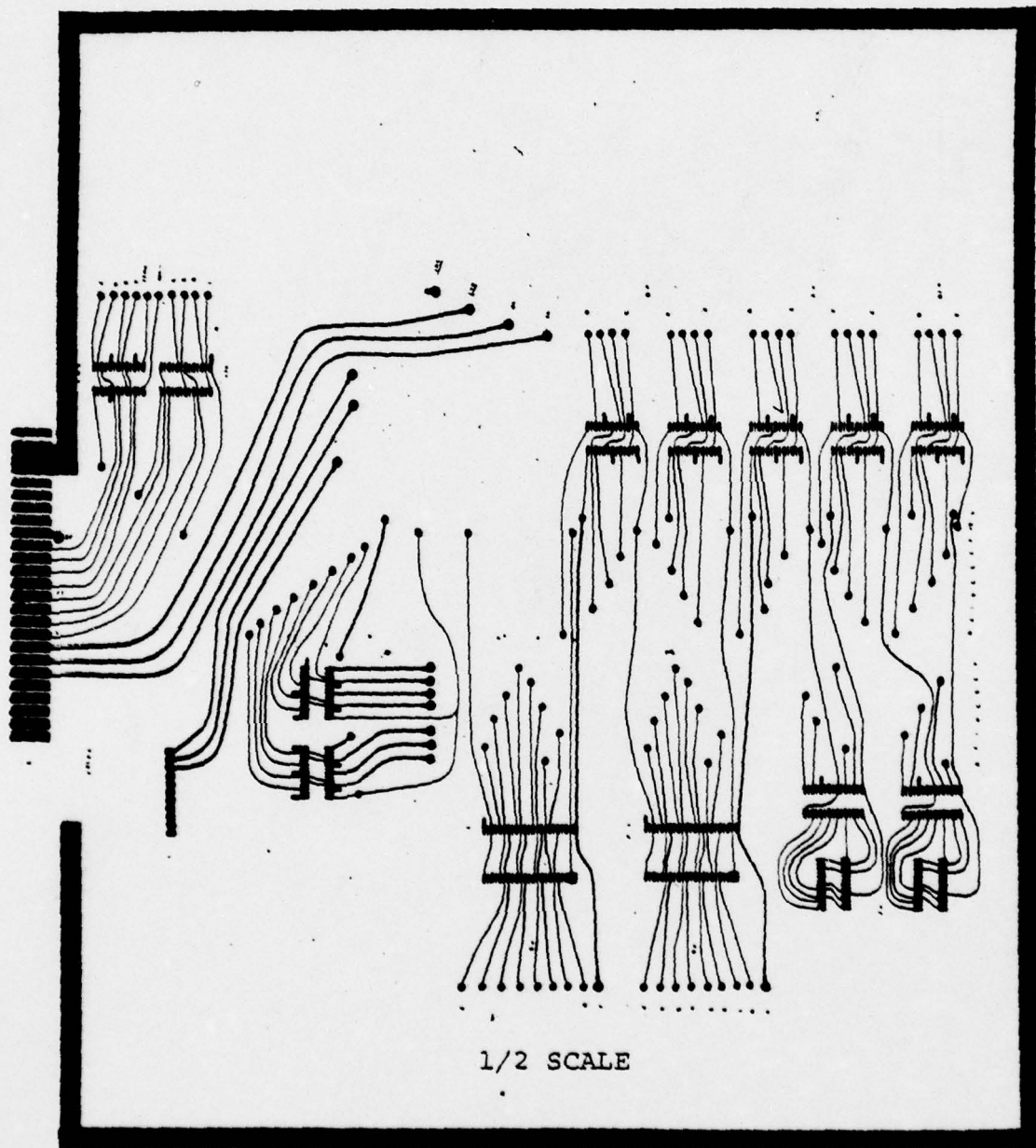
ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
2A	74SA		CALL BCDC	
2C	34A0		CALL TTYO	;Print Magnitude
2E	2309		MOV A,09	;select port 9
30	3A		OUTL P2,A	
31	08		INS A BUS	;Read port 9
32	AF		MOV R7,A	
33	745A		CALL BCDC	
35	34A0		CALL TTYO	;print phase
37	00	LOP4	NOP	
38	4637		JNT1, LOP4	
3A	E80C		DJNZ Ro,LOP1	
3C	BB0D	LOP3	MOV R3,0D	
3E	3451		CALL BAUDO	;carriage return
40	BB0A		MOV R3,0A	
42	3451		CALL BAUDO	;Line feed
44	440A		JMP LOP2	

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
1A0	FE	TTYO	MOV A,R6	;start with BCD# in
	A9		MOV R1,A	;R 5,6
	FD		MOV A,R5	
	E3		MOV P3 A,@A	;look up ASCII equivalent of
				;first digit
	AB		MOV R3,A	
	3451		CALL BAUDO	;output most sig. digit.
	F9		MOV A,R1	
	47		SWAP A	
	530F		ANL A,0F	
1AB	E3		MOV P3A,@A	
	AB		MOV R3,A	
	3451		CALL BAUDO	;output middle digit
	F9		MOV A,R1	
	530F		ANL A,0F	
	E3		MOV P3 A,@A	
	AB		MOV R3,A	
	3451		CALL BAUDO	;output LS digit
	BB20		MOV R3,20	;leave 1 space
	3451		CALL BAUDO	
1BA	83		RET	

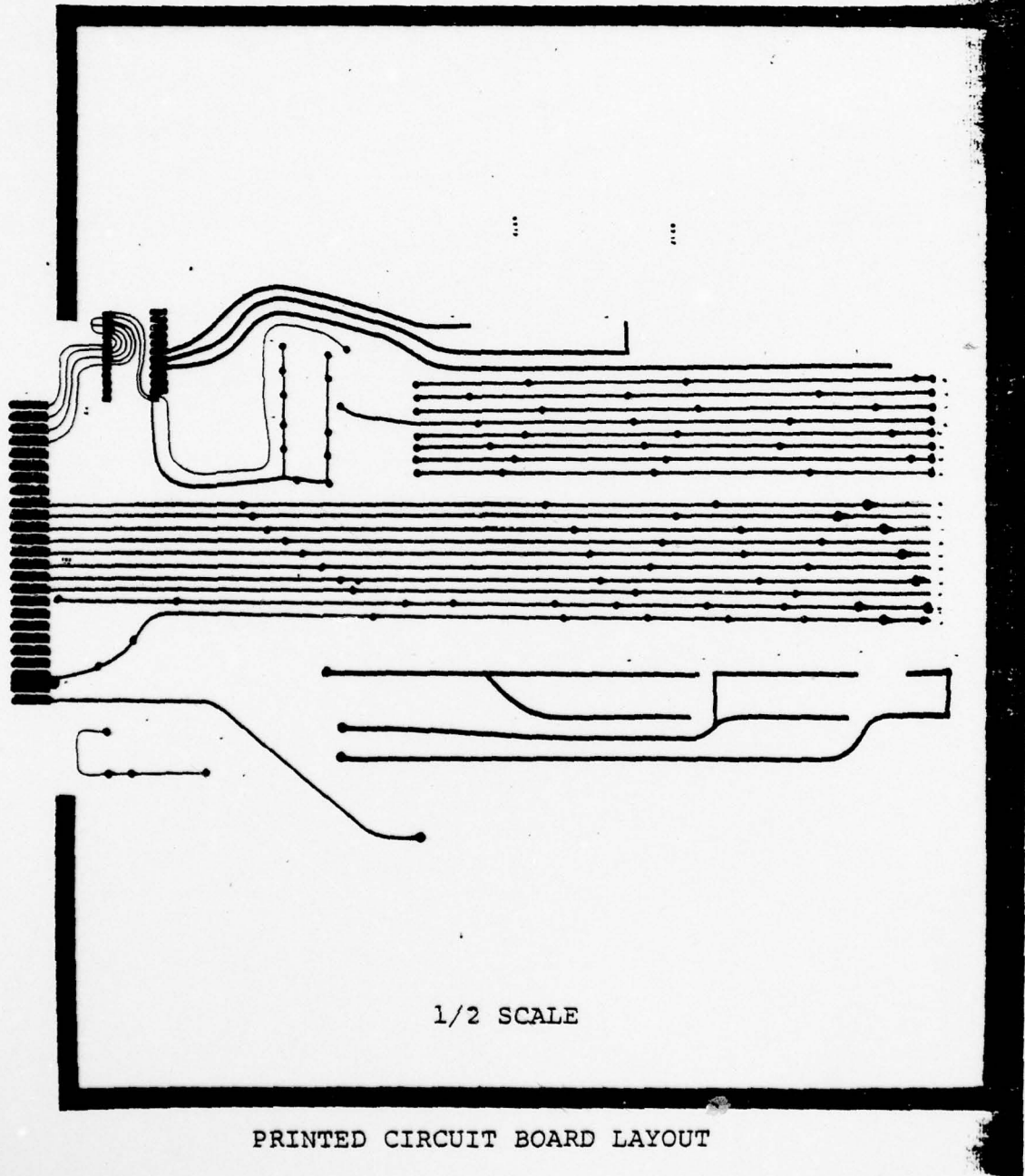
ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
151	2301	BAUDO	MOV A,01	
153	02		OUTL BUS A	
154	3490		CALL SEL6	;start the start bit.
156	BE22		MOV R6,22	
158	BF22		MOV R7,22	
15A	74E5		CALL DELAY	
15C	BA09		MOV R2,09	;set count for 8 bits
15E	FB	LOP0	MOV A,R3	;put char. into A
15F	EA63		DJNZ R2,LOP1	
161	2482		JMP END	
163	97	LOP1	CLR C	
164	67		RRC A	
165	AB		MOV R3,A	
166	E675		JNC LOP2	
168	2319		MOV A,19	;start "1" bit
16A	02		OUTL BUS A	
16B	3490		CALL SEL6	
16D	BF23		MOV R7,22	
16F	BE23		MOV R6,22	
171	74E5		CALL DELAY	
173	245E		JMP LOP0	
175	2301	LOP2	MOV A,01	
177	02		OUTL BUS A	;send out a "0"
178	3490		CALL SEL6	
17A	BE23		MOV R6,22	
17C	BF23		MOV R7,22	

ADDR	HEX CODE	LABEL	MNEMONIC	COMMENT
17E	74E5		CALL DELAY	
180	245E		JMP LOPØ	
182	2319	END	MOV a,19	;output 2 "1s"
184	02		OUTL BUS A	
	3490		CALL SEL6	
	BE45		MOV R6,45	
	BF45		MOV R7,45	
	74E5		CALL DELAY	
18D	83		RET	
190	2306	SEL6	MOV A,Ø6	
192	3A		OUTL P2,A	
193	23ØØ		MOV A,ØØ	
195	3A		OUTL P2,A	
196	83		RET	

APPENDIX E



PRINTED CIRCUIT BOARD LAYOUT



2 x 1/4 x 2

1/2 SCALE

PRINTED CIRCUIT BOARD LAYOUT



APPENDIX F

A. RECORDING PROCEDURE USING THE CASSETTE RECORDER AND MODEL 40 LINE PRINTER

1. Connect the data cables from the controller to the ADCs, cable number 8 to the magnitude ADC and number 9 to the phase ADC. Ground the black banana plug of each cable and connect the red sample strobe plug of cable 8 to the sample strobe input of the ADC box.
2. Turn the speaker and RNC switches on the front of the controller OFF.
3. Connect the model 40 line printer by cable CA-40.
4. Turn the power ON.
5. Press: RST, O, F.
6. Move the mechanical scanner to the starting position. Ensure that the sampling speed will not exceed 2 Hz. Stop the scanner.
7. Place an erased tape into the cassette recorder.
8. Move the Load Forward switch on the side of the recorder to the fully up position long enough to ensure that the tape is off the leader and that both sprockets of the tape transport are properly engaged and the tape is moving.
9. Move the Load Forward switch to the fully down position. Verify that the Status light on the recorder is ON.

10. Press: RST, 3. At this point samples will be recorded and printed each time the sample strobe line goes low. Depressing Test 1 (T1) on the keyboard will initiate a manual sample.
11. Start the mechanical scanner. Verify that the printer begins printing and that the recorder status light is flashing at 1 Hz.
12. The above procedure is applicable even if either the line printer or cassette recorder is not being used, such as during calibration or alignment testing.

B. RECORDING PROCEDURE USING THE MODEL 33 TELETYPE

1. Connect the data cables from the controller to the ADCs, cable number 8 to the magnitude ADC and number 9 to the phase ADC. Ground the black banana plug of each cable and connect the red sample strobe plug of cable 8 to the sample strobe input of the ADC box.

2. Turn the speaker and RNC switches on the front of the controller OFF.

Connect the 20 mA current loop adapter to port 6 and the TTY to the current loop in accordance with the instructions on the current loop box. Place the TTY in local operation.

4. Turn the power ON.

5. Press: RST, O, F.

6. Move the mechanical scanner to the starting position.

Ensure that the sampling rate will not exceed 1 Hz.

Stop the scanner.

7. Press: RST, 4.

8. While in local operation punch any information or required header onto the tape. Always end the header with at least one carriage return and line feed. Place the TTY in line operation.

9. Start the scanner. Verify that the scanning motor is running and that the TTY is printing and punching paper tape.

LIST OF REFERENCES

1. Culpepper, J.C., Analog to Digital Conversion and Hardware Improvement For A Computer Aided Acoustic Imaging System, MSEE Thesis, US Naval Postgraduate School, Monterey, Calif., 1978.
2. Teletype Corporation, Technical Manual, KSR-33, Bulletin 310B, 1972.
3. Motorola Semiconductor Products, Inc., Semiconductor Data Library/Linear, Vol. 6, p. 8-16, 1976.
4. Teletype Corporation, Technical Manual, Model 40 Line Printer, Specification 508945, 1977.
5. Datel Systems, Inc., Doc. No. MWZADH 1705, Instruction Manual Incremental Digital Cassette Recorder Systems Model ICT-WZ Series, 1977.
6. Datel Systems, Inc., Doc. No. LTYBMH2802, Model LPR-16-2 or 3 TTY/RS-232-6 Cassette Reader User's Instruction Manual, 1978.
7. Peatman, J.B., Microcomputer-Based Design, p. 16-18, McGraw-Hill, 1977.
8. INTEL Corporation, MCS-48 Microcomputer User's Manual, 1977.
9. National Semiconductor Corp., TTL Databook, 1976.
10. INTEL Corporation, Component Data Catalog, 1978.
11. Electronic Industries Association Standard RS-232-C, Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, p. 4-7, 1969.
12. INTEL Corporation, Prompt-48 Microcomputer User's Manual, 1978.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Assoc. Professor R. Panholzer, Code 52Pz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
5. Assoc. Professor M.L. Cotton, Code 52Cc Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Assoc. Professor J.P. Powers, Code 62 Po Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
7. Lieutenant Rodney A. Colton, USN 2461 E. Harmon Ave. Las Vegas, Nevada 89121	3
8. Dr. Newell Booth, Code 6513 Naval Ocean Systems Center San Diego, California 92152	2
9. Mr. Normal Caplan Automation, Bioengineering and Sensing System Program Engineering Division National Science Foundation 1800 G. Street Washington, D.C. 20550	1

AD-A074 314

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
A MICROCOMPUTER-BASED DIGITAL DATA ACQUISITION CONTROLLER FOR A--ETC(U)
JUN 79 R A COLTON

F/6 9/2

UNCLASSIFIED

NL

2 OF 2

AD
A074314



END
DATE
FILMED

10-79

DDC

10. Dr. G. Hutton
Central Institute for Industrial Research
Forskning sveien 1
Oslo 3, Norway

1