DRA

OSU

The Ohio State University

A MICROCOMPUTER BASED DATA ACQUISITION SYSTEM AND EXPERIMENT CONTROLLER

Matthew W. Ganz

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 712759-3
February 1981
Contract NASW-3393

National Aeronautics and Space Administration Headquarters
Washington, D.C. 20546

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

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 DATA ACQUISITION SYSTEM AND EXPERIMENT CONTROLLER		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER ESL 712759-3
7. AUTHOR(s) Matthew W. Ganz		8. CONTRACT OR GRANT NUMBER(s) Contract NASW-3393
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering Columbus, Ohio 43212		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS NASA Headquarters HQ Contracts and Grants Division Washington, D.C. 20546		12. REPORT DATE February 1981 13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
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) AIM-65 microcomputer Analog to digital converter Digital input/output Direct memory access Analog conditioning circuit Microformatter Expansion motherboard		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes a data acquisition system which was designed and implemented at The Ohio State University's ElectroScience Laboratory. The system monitors and records the signal strength of a radio beacon sent to earth from a geosynchronous satellite. It acquires data from several devices such as a radar, a radiometer, and a rain gauge which can monitor the meteorological conditions along the Earth-space propagation path. The acquired data are stored in digital format on magnetic tape for analysis at the laboratory's computer		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

center. This data acquisition system is based on the Rockwell AIM-65 Advanced Interactive Microcomputer. It replaces a previous system which used 2 minicomputers to perform the same tasks.

This report gives a detailed description of the design and operation of the system's various hardware components. The report also presents schematic diagrams, the theory of operation, and normal operating procedures.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. Overview

This report describes a data acquisition system designed, constructed, and implemented at the Satellite Communications Facility of the Ohio State University ElectroScience Laboratory. The system replaced a former system which used a Hewlett-Packard 2116B Minicomputer as its intelligent controller. The old system worked very well but it underwent frequent modifications over the past decade and became quite complicated. This system was not extensively documented and many of the modifications were not documented at all. Operating and troubleshooting this system were often difficult and frustrating. With the advent of modern microprocessor technology it was decided that a new data acquisition system based on a small microcomputer should be built.

The new system was designed to be easily understood, easily expanded, and well documented. The design stresses simplicity of design and programming techniques so that people unfamiliar with the system's operation would be able to master it quickly. The hardware design is general enough that the data acquisition system can easily be used for other applications.

The system is designed for easy maintenance and repairs. All components, including the microcomputer, are inexpensive enough to keep duplicates on hand. This will help reduce down time and costly repairs.

The system's operating program can be stored in Eraseable Programmable Read Only Memory (EPROM), eliminating the need to load the program during the system's start-up. The program in the EPROM can be easily modified as the system's needs change.

This report is organized so that it will be useful to both the system operator and the troubleshooter. Each of the several subsystems has a chapter devoted to it. Each chapter begins with an overview of the subsystem and instructions for proper operation. This is followed by an indepth look at the subsystem's hardware and theory of operation. Schematic drawings and wire lists appear with the explanatory text.

A glossary at the end of the report gives the meanings of some of the commonly used words, abbreviations, acronyms and phrases.

B. System Description

The data acquisition system comprises several hardware components. A block diagram of the system is shown in Figure 1-1. The Rockwell AIM-65 Advanced Interactive Microcomputer is used as the intelligent processor. Added to the basic microcomputer is an expansion mother board. This board contains five sockets into which boards containing Input/Output (I/O) or memory devices can be inserted. Two such boards were constructed and a brief description of each of them follows.

The first board designed was a digital I/O board. This board contains 128 digital lines each of which can act as an input or output. The digital I/O board also contains several counters, timers, and special I/O functions. Chapter 2 gives operational and functional details of this board.

The second board is an analog input board which contains 16 Analog to Digital Converters (ADCs). Each of the ADCs monitors an analog voltage and produces a digital number proportional to the voltage. The computer can then read, process, and store this number. Thus, the computer can observe the analog voltages produced by sensors which monitor different enviornmental conditions such as temperature, pressure and wind speed, in addition to signals such as the received satellite beacon amplitude. Chapter 3 discusses the design and operation of the analog input board.

A third board containing 8 kilobytes of additional Random Access Memory (RAM) for the AIM-65 was designed but has not yet been constructed. This board can provide expansion memory for user programs or data in the system, although the present system does not require the use of this board. If the need for additional RAM arises, this board can be quickly constructed and implemented.

In addition to the boards on the expansion motherboard, several other hardware devices support the AIM-65. A cassette deck is connected

to the cassette interface of the AIM-65. This cassette deck allows the computer's operator to store programs or data on a standard cassette tape. These data or programs can then be quickly reloaded into the computer. The AIM-65 also provides a teletype (TTY) interface. This allows a standard teletype to be used as an interactive I/O device.

The AIM-65 is also interfaced with a Pertec FT 8840A-9, 9-track digital tape deck. The data which are acquired are stored on magnetic tape using this deck. A special interface, called a Direct Memory Access (DMA) controller, was designed to allow easy transfer of data between the AIM-65 and the magnetic tape. Chapter 4 presents details of the operation of the DMA controller and the digital tape deck.

C. The AIM-65

The heart of the data acquisition system is the AIM-65 microcomputer. This computer was chosen for several reasons. It is very inexpensive, reliable and versatile. It is fairly well documented and is easily expandable. The operation of the AIM-65 is the subject of study in an undergraduate course at The Ohio State University. Some of the future students working at the ElectroScience Lab will have gained some experience with the AIM-65 through this course.

The AIM-65 contains many versatile features. The computer is based on a Rockwell R6502 8-bit microprocessor. It has 4096 (4K) bytes of RAM which can be used for program or data storage. The AIM-65's addressing scheme permits expansion up to 64K bytes. The AIM-65 also has a monitor program in ROM (permanent memory). This monitor program allows programs and data to be easily entered, debugged and modified. It also has several powerful subroutines which can be used in other programs to do several special functions. The microcomputer has a full 54 key keyboard which allows the operator to control the machine. A 20 character alpha-numeric display and thermal printer are provided to allow the display and printout of messages and data. Information can easily be transferred to these output devices using the user-available subroutines mentioned above.

The AIM-65 also has one very versatile integrated circuit (IC) on board. This IC is the Rockwell R6522 Versatile Interface Adapter (VIA). The terms VIA and 6522 are both used in this report to refer to this IC. The 6522 has two 8-bit bidirectional I/O ports. The computer sees each port as an ordinary memory register.

To the outside world they look like standard TTL digital inputs or outputs. These ports are programmable; that is, each can be programmed to be either an input or an output port. The 6522 also has two programmable counter/timers and several other useful features. The 6522 is the basic building block used in the digital I/O board and is treated in more detail in Chapter 2.

The AIM-65's manufacturer, Rockwell, has published several books which provide quite extensive hardware and software support for the computer. They are The AIM-65 User's Guide, The R6500 Hardware Manual, The R6500 Programming Guide, and The AIM-65 Monitor Program Listing. These items are referenced as items [1], [2], [3], and [5] respectively, in the bibliography.

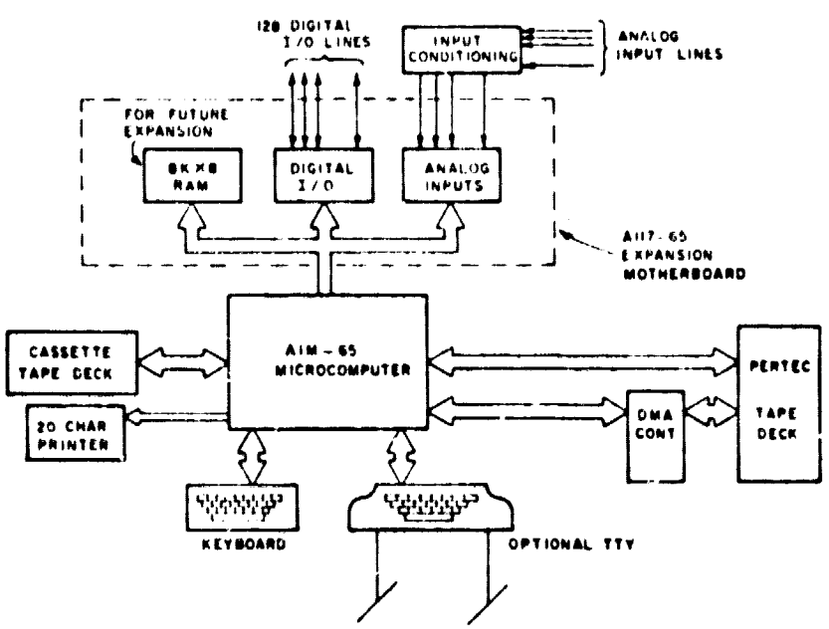


Figure 1-1 Data Acquisition System

II. DIGITAL I/O SUBSYSTEM

A. Introduction

The digital input/output section of the data acquisition system is based on the 6522 VIA integrated circuit. A circuit board containing 8 VIA's was designed and fabricated. Printed circuit techniques were used for fabrication of the circuit board to allow easy duplication.

The digital I/O board is designed to be accessed quite easily by the AIM-65. It is assumed that the reader is somewhat familiar with the programming and use of the 6522. Extensive coverage of this subject is given in [1] and [2].

B. Operation

Before we examine the programming of the VIA, we will see where the VIAs are located in the AIM-65's memory. The VIA supplied with the computer occupies the addresses (memory locations) labeled AØØØ through AØØF (in hexadecimal format). Each of the VIAs on the digital I/O board also occupies 16 memory locations. Thus the eight VIAs on the digital I/O board require a total of 128 memory locations.

Figure 2-1(a) shows a map of the section of memory used by the digital I/O board. This board occupies locations 81ØØ through 817F in the AIM-65's memory. The block of memory which the board occupies can be changed simply by changing the configuration of a set of switches on the board. This allows several boards to be used in the system simultaneously by assigning each to a different memory block. The second section of this chapter gives instructions for properly setting these switches. For the following discussion we will assume that the switches are set to be consistant with Figure 2-1.

Figure 2-1(b) shows the memory map of a typical VIA's organization. The VIA (#3), like all of the others, occupies 16 memory locations. The name of each of these 16 registers is shown in the figure. Details of the significance of each register can be found in Reference [1] and Reference [2]. This chapter discusses only the most important registers and functions.

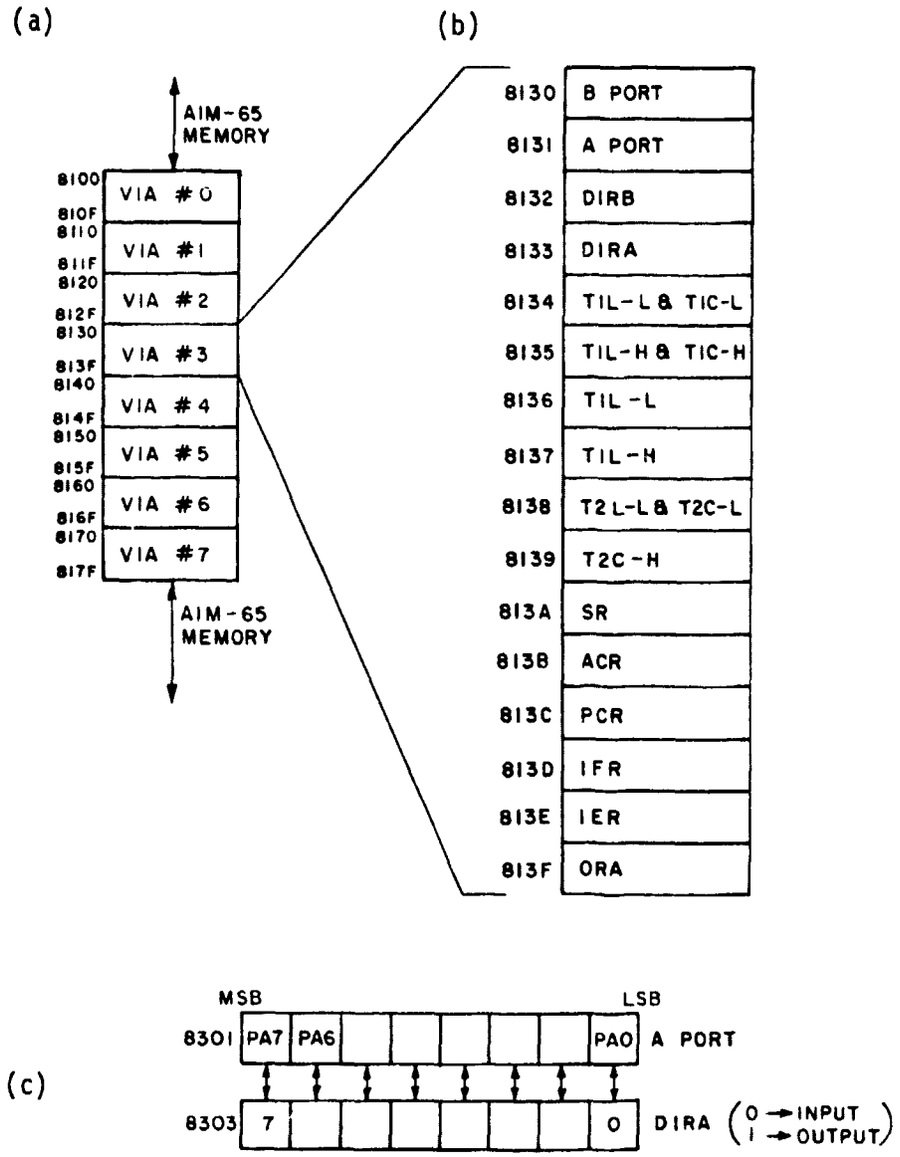


Figure 2-1 (a) Partial AIM-65 Memory Map
(b) Detailed Memory Map of VIA #3
(c) Detail of Port-Direction Register Relationship

The only mode of V.I.A. operation that we will deal with here is the simple I/O port mode. Each V.I.A. has two 8-bit bidirectional I/O ports. The computer sees each of these ports as if it were an ordinary memory register. To the outside world, the ports each look like a set of 8 standard logic (TTL) gates. Each port can be designated as either an input port or an output port. In fact, each of the eight bits within a port can be programmed to be either an input bit or an output bit. Our programming, however, will be much easier if we designate the bits within a particular port to be all inputs or all outputs.

The next concept we must understand is how to designate (or program) a particular port to be an input port or an output port. From Figure 2-1 we see that the two I/O ports in the V.I.A. are labled "A port" and "B port". In V.I.A. #3 we see that they are located at locations 8131 and 8130 respectively. Locations 8133 and 8132 are labled DIRA (for DIRection of A port) and DIRB respectively. These two 8-bit registers are called direction registers. Each bit in a direction register corresponds with a bit in its associated port. Thus, by writing the proper bits into a direction register we can define each bit in the associated port to be either an input or an output.

The details of the correspondance between a typical port and direction register is shown in Figure 2-1(c). A logical 1 in a direction register bit makes the corresponding port bit an output. A logical Ø in a direction register bit makes the corresponding port bit an input. Thus, if we wanted the A port in V.I.A. #3 to be an output port and the B port to be an input port we might use the following sequence of instructions

```
LDA  #FF     ;A←(11111111)
STA 8133     ;DIRA← A
LDA  #ØØ     ;A← (ØØØØØØØØ)
STA 8132     ;DIRB← A
```

These instructions would typically be placed at the very beginning of a program as part of the initialization procedure. More examples of V.I.A. programming are given in Chapter 8 of Reference [1].

Now that we have given our ports direction we will discuss the passing of data between the computer and the outside world through the port. The contents of our output port (A port) are controlled by writing into location 8131. Writing a logic 1 into a bit in 8131 causes the corresponding output bit to go to the logic 1 state (+3VDC). A logic Ø written in this same A port bit causes a logic Ø (≈ØV) to appear on the corresponding output port bit. Thus, to make all of our A port bits go to the logic Ø state, a typical program segment might look like:

```
LDA  #ØØ      ;A ←(ØØØØØØØØ)
STA 8131      ;APort  ←  A
```

The B Port is our input register and it can be read by a simple LDA 813Ø instruction. If any input is at a potential greater than 2 volts, a logical 1 will be found in the corresponding port location when the port is read.

Each pair of I/O ports can be accessed by the outside world through a 20-pin connector. One such connector is located next to each 6522 I.C. Connector pin assignments are shown in Figure 2-3. Each V.I.A. output is capable of driving one standard TTL load. The B Port outputs have a higher current sourcing ability which allows them the capability of driving a Darlington transistor pair.

There are two control lines for each port. These lines (CA1,CA2 for the A port and CB1, CB2 for the B Port) can be used for some of the more advanced features of the 6522 (such as handshaking or interrupt control). Each of the 20 pin connectors has two spare pins (pins #10 and #11). These spare pins can be jumpered to two of the control lines if the user desires external access for these lines.

C. Hardware Details

Standard address decoding techniques are used to select the digital I/O board. Some preliminary decoding of the address bus is done internally by the AIM-65. Additional decoding is done on the digital I/O board. This decoding is done by comparing five address bus bits with the output of a set of five switches. Setting the configuration of these switches allows the user to select a particular memory range for each board. When it is determined that a particular board has been selected, an additional decoder on that board selects one of the eight V.I.A.'s. Details of this decoding and other aspects of the hardware configuration of the digital I/O section are presented in this section.

A schematic diagram of a V.I.A. board is shown in Figure 2-2. The AIM-65 has a 16-bit address bus which is capable of addressing 65,536 (2^{16}) memory locations. The four most significant bits of this bus (A15, A14, A13, A12) are internally decoded by the AIM-65. The output of this internal decoder which interests us is the one labled $\overline{CS8}$ in Figure 2-2. This line is normally at the logic 1 state. It only assumes the logic Ø state when an address between 8ØØØ and 8FFF appears on the address bus (i.e. during a read or write operation to any memory location between 8ØØØ and 8FFF).

The $\overline{CS8}$ line and the five next most significant address lines (A11, A10, A9, A8, and A7) are connected to the input of a 6-bit magnitude comparator. This comparator is composed of two cascaded 74LS85 4-bit comparators (Z_1 and Z_2). The address lines mentioned above are connected to the A inputs of the comparators and the B inputs are connected to switches S1-S6. These switches allow each B input to be set to either the logic 1 or the logic Ø state. When one of these swithches is in the "ON" position, the corresponding B input is grounded (logic Ø). The same switch in the "OFF" position causes the B input to be pulled up to +5VDC (logic 1). Thus by setting S1-S6, we can change the address range for which the comparator's output will be active. Table 2-1 gives a summary of the possible address ranges and the necessary switch configurations for

proper operation in each range.

When an address in the selected range appears on the address bus, the comparator's output becomes active (pin 6 of Z_2 assumes a logic 1 state). This enables a 1 of 8 decoder, Z_3. When Z_3 is enabled, it selects one of the eight V.I.A.s. The V.I.A. selected depends on the status of the select inputs to Z_3 (A6, A5, and A4). An example will be presented shortly which will clarify this address decoding process.

Before the address decoding example is presented, a few words should be said about the 6522 I.C. itself. Figure 2-3 is a schematic diagram showing the connection details of a typical 6522. Several control signals from the AIM-65's control bus (SYS R/$\overline{W}$, ϕ2, $\overline{IRQ}$, $\overline{RES}$,) and data bus (DBØ-DB7) are connected to every V.I.A. These lines are used for control and data transfer purposes. The V.I.A. will only respond to signals on these buses when its $\overline{CS}$ line is low. This $\overline{CS}$ line is connected to an output of Z_3 (the 1 of 8 decoder). Thus, when a particular V.I.A. is selected by the address decoding process (mentioned above), it becomes able to communicate with the computer through the control and data buses.

The four remaining address bus lines, A3, A2, A1, and AØ, are connected to RS3, RS2, RS1, and RSØ of each V.I.A. (the RS stands for Register Select). These four lines select one of the V.I.A.'s 16 internal registers. A data word can then be transferred to or from the register on the data bus.

Now, the promised addressing example will be presented. Let's first suppose that all of the address selection switches except S5 are closed (ON position). Referring to Table 2-1, we see that this sets our address range to 81ØØ-817F.

We can verify this by examining the state of all of the logic elements when we try to read from (or write to) a location in this range. Let's suppose the computer was writing to the A port of V.I.A. #3. The program step executing this would be (see Section 2B):

STA 8131

TABLE 2-1

Switch and Corresponding Bit						Address Range
Switch	S_1	S_2	S_3	S_4	S_5	S_6	
Corresponding Bit	CS8	A_{11}	A_{10}	A_1	A_8	A_7	
	0	0	0	0	0	0	8000-807F
	0	0	0	0	0	1	8080-80FF
	0	0	0	0	1	0	8100-817F
	0	0	0	0	1	1	8180-81FF
	0	0	0	1	0	0	8200-827F
	0	0	0	1	0	1	8280-82FF
	0	0	0	1	1	0	8300-837F
	0	0	0	1	1	1	8380-83FF
	0	0	1	0	0	0	8400-847F
	0	0	1	0	0	1	8480-84FF
	0	0	1	0	1	0	8500-857F
	0	0	1	0	1	1	8580-85FF
	0	0	1	1	0	0	8600-867F
	0	0	1	1	0	1	8680-86FF
	0	0	1	1	1	0	8700-877F
	0	0	1	1	1	1	8780-87FF
	0	1	0	0	0	0	8800-887F
	0	1	0	0	0	1	8680-88FF
	0	1	0	0	1	0	8900-897F
	0	1	0	0	1	1	8980-89FF
	0	1	0	1	0	0	8A00-8A7F
	0	1	0	1	0	1	8A80-8AFF
	0	1	0	1	1	0	8B00-8B7F
	0	1	0	1	1	1	8B80-8BFF
	0	1	1	0	0	0	8C00-8C7F
	0	1	1	0	0	1	8C80-8CFF
	0	1	1	0	1	0	8D00-8D7F
	0	1	1	0	1	1	8D80-8DFF
	0	1	1	1	0	0	8E00-8E7F
	0	1	1	1	0	1	8E80-8EFF
	0	1	1	1	1	0	8F00-8F7F
	0	1	1	1	1	1	8F80-8FFF
	1	X	X	X	X	X	None

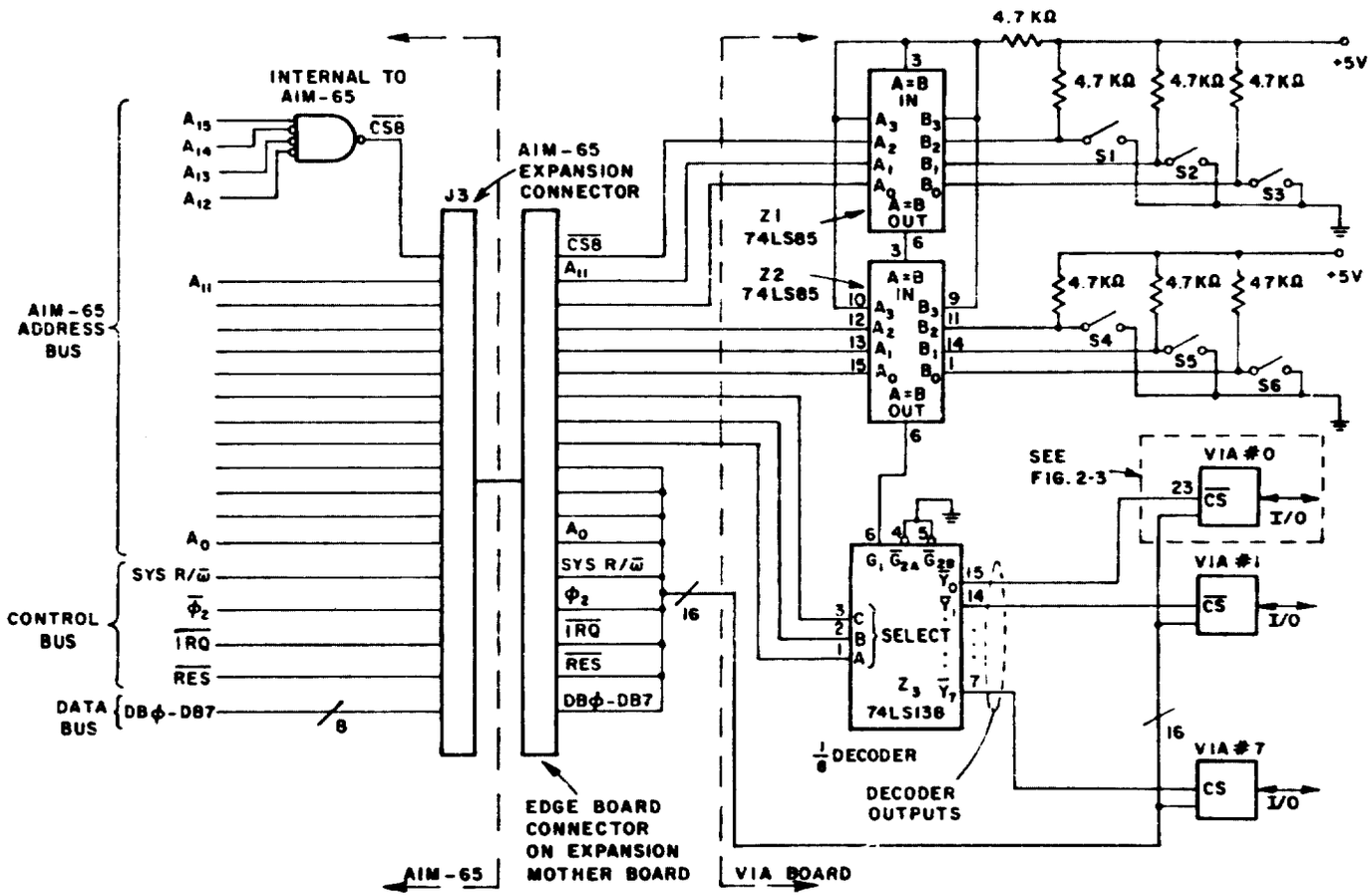


Figure 2-2 VIA Board

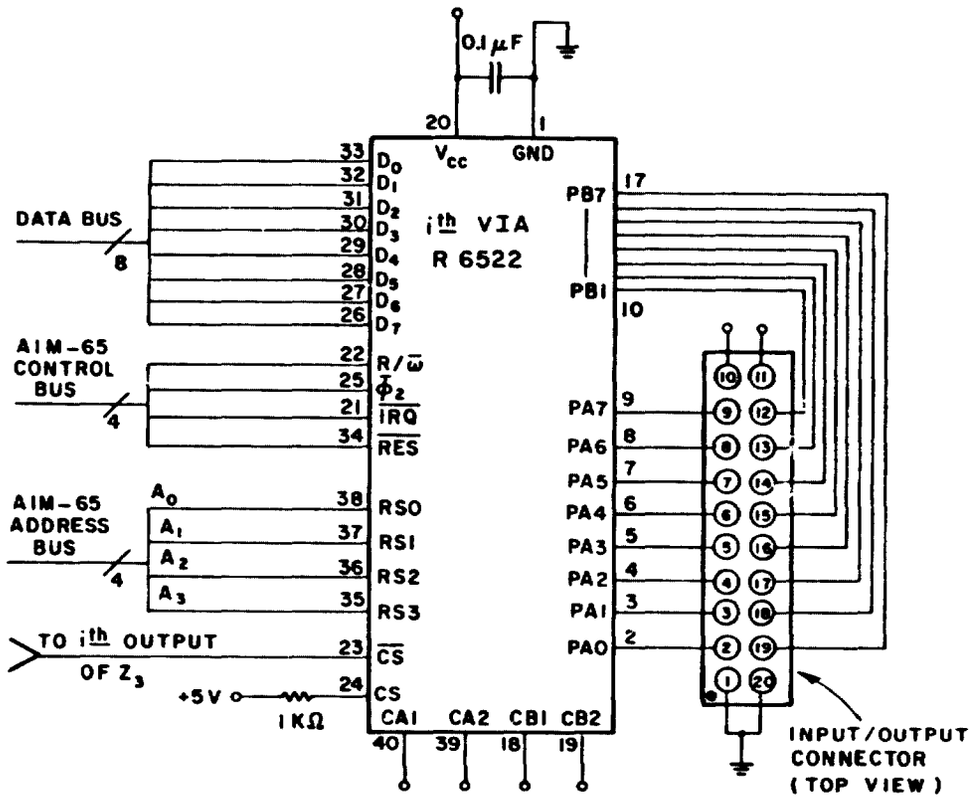


Figure 2-3 VIA Connection Details

Thus, 8131 would be the address appearing on the address bus. The address bus state would be:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1
8				1				3				1			

Since (A15, A14, A13, A12) = (1,Ø,Ø,Ø), $\overline{CS8}$ will be at logic Ø. Now ($\overline{CS8}$, A_{11}, A_{10}, A_9, A_8, A_7) = (Ø,Ø,Ø,Ø,1,Ø), and switches (S_1, S_2, S_3, S_4, S_5, S_6) also equal (Ø,Ø,Ø,Ø,1,Ø), thus the same pattern of 1's and Ø's appears at both the A and B inputs of the comparator. The A=B output (pin 6, Z2), will go to the logic 1 state. This enables Z3. Now, the C,B, and A inputs to Z3 are set to Ø, 1, and 1 respectively, (i.e. the state of A_6, A_5, and A_4 respectively). Since Z3 is enabled and (C,B,A) = (0,1,1), the Y3 output goes low (logic Ø); thus, the $\overline{CS}$ line of V.I.A. #3 goes low and V.I.A. #3 is enabled. Since (A3, A2, A1, AØ)= (RS3, RS2, RS1, RSØ)=(Ø,Ø,Ø,1)=1 (Hex), the #1 register in V.I.A. #3 (the A port) is selected.

Now, since the V.I.A. sees that we have selected the A port, it looks at the R/$\overline{W}$ line. Since this line is in the write (logic Ø) state, the data on the data bus (which equals the accumulator since we are doing an STA operation) is written into the A Port and our operation is complete.

There are several other functions which the 6522 V.I.A. can perform and the versatility of the I.C. cannot be stressed enough. In this chapter, we've seen how to accomplish some very simple digital I/O operations. Several additional programming and hardware details are presented in the Rockwell literature and the interested reader is encouraged to examine these sources.

Additional information on the digital I/O subsystem, including the PC board layout, is given in the appendices.

III. ANALOG INPUT SUBSYSTEM

A. Introduction

The analog input section of the data acquisition system is described in this chapter. This section (or subsystem) allows the microprocessor to acquire data derived from analog quantities. Such analog signals might typically be the outputs of a beacon receiver or temperature, pressure, or wind speed transducers.

The analog input section consists of 16 analog to digital (A/D) converters and the hardware necessary to interface them with the computer and the analog signals. Analog Devices AD7574JN monolithic A/D converters were chosen for use in the analog input section. Sixteen such A/D converters are contained on the A/D board which can be inserted into any slot on the AIM-65 expansion motherboard. The AD7574JN was chosen because it is inexpensive, readily available, and can be easily interfaced with the microcomputer system.

Another board provides input conditioning for each A/D converter on the A/D board. The conditioning board provides input buffering and level shifting for the analog input signals before they are applied to the A/D converters. Details of both the A/D board and the analog conditioning board are given in the third section of this chapter.

B. Operation and Calibration

This section presents the operational details of the analog input section. The basics of A/D conversion are presented and soft- details are given. The final portion of this section gives a suggested calibration procedure.

The AD7574JN is an 8-bit Analog to Digital Converter (ADC). It converts an analog voltage between ØV and 10V to a digital number between Ø and 255 (base 10). This digital number is directly proportional to the magnitude of the analog input voltage. Thus, the system resolution is:

$$\text{Resolution} = \frac{V_{max} - V_{min}}{\text{\# of digital intervals}} = \frac{1ØV - ØV}{255} = .039V$$

Most of the analog signals presented to the data acquisition system are in the -5V to +5V range. Since the AD7574 requires an input in the ØV to 1ØV range, the analog signals must be shifted up by 5V to make them compatible with the ADC. A board called the analog conditioning board provides this 5 volt level shift. This board is located in a chassis, separate from the A/D board. The analog conditioning board contains 16 identical circuits, each with the transfer function:

$$V_{out} = V_{in} + 5V.$$

A few of the analog signals are in the ØV to 5V range. By changing the configuration of a few jumpers on a particular analog conditioning circuit, the circuit's transfer function can be changed to

$$V_{out} = 2(V_{in}).$$

This allows the ØV to +5V input range to become transformed into our desired ØV to +10V range. The next section of this chapter gives the details for setting the jumpers to provide the desired input range.

The analog signals are brought into the analog input system through BNC connectors located in the back of the chasis which contains the analog conditioning board. These connectors are labled with the channel numbers Ø through F. In this system, the word channel is used to describe one A/D converter, its conditioning circuit, and the cable which connects them.

Each A/D converter occupies one location in the AIM-65's memory. The addressing scheme for the ADCs is quite simple. The first three digits in the address are always 8ØØ. The fourth digit is the channel number. Thus, ADC #Ø is addressed by 8ØØØ, ADC #1 by 8ØØ1, etc., up to 8ØØF for ADC #F (#15 in base 10).

Before an A/D conversion can be made, the ADC must be reset. This is done simply by "reading" the ADC. For example, ADC #9 can be reset by the instruction:

LDA 8009.

Each ADC automatically resets itself after it performs a conversion. Thus, "manual" resetting only needs to be done once (before the first conversion). This manual resetting is usually done in the initialization routine in the user's program.

An A/D conversion is initiated by "writing" to the ADC's address. The conversion takes approximately 15μs. When the conversion is complete, the digital result can be transferred to the accumulator by "reading" the ADC. Thus, a program segment which will perform an A/D conversion on ADC #0 and transfer the result to the accumulator might look like:

```
LDA 8000                          Reset converter
    .                             (During initialization
    .                             procedure)
    .
    .
STA 8000                          Start conversion
NOP  ┐
NOP  │
NOP  │
NOP  │   Delay 16μs (2μs/NOP)
NOP  │
NOP  │
NOP  │
NOP  ┘
NOP
LDA 8000                          A ← Result
    .
    .
    .
```

The calibration procedure for the ADC's is quite straightforward. The easiest method to examine the outputs of the ADC's is to use some of the AIM-65's user subroutines to display them. A program which will display the output of an ADC on the AIM-65's display is given below. The letter X is used in place of the channel number (0-F) of the ADC to be calibrated.

```
        LDA  8ØØX    ;     Reset ADC
Begin:  STA  8ØØX    ;     Start Conversion
        NOP  ⎫
        NOP  ⎪
        NOP  ⎬   Delay
        NOP  ⎪
        NOP  ⎪
        NOP  ⎭
        JSR  EB44    ;     Clear Display
        LDA  8ØØX    ;     A  Conversion Result
        JSR  EA46    ;     Display ← A
        JMP  begin         Jump Back to Beginning
```

When this program is run, a 2-digit hexadecimal number (which is the 8-bit output of the ADC) will appear on the display.

The GAIN and OFFSET ADJ. of the analog conditioning circuit being calibrated must be adjusted for proper operation. The procedure for proper adjustment is given below:

1) With Ø.ØV applied to the input to the channel, adjust the OFFSET ADJ. potentiometer until the number shown on the display is 8Ø or flickers between 7F and 8Ø.

2) Apply +5.0V to the channel input. Adjust the GAIN potentiometer until the number shown on the display is FF or flickers between FE and FF.

3) Repeat the above 2 steps until no trimming is required to get the desired output with either ØV or +5V applied to the input.

This completes the operational description of the analog input section

C. Hardware Details

The AD7574JN analog to digital converter requires very little supporting hardware. The basic ADC circuit is shown in Figure 3-1. This circuit is duplicated 16 times on the A/D board. The necessary address decoding and data buffering is also done on this board.

An external resistor and capacitor connected to the CLK input (pin 17) of the ADC determine the internal clock rate and, thus, the A/D

conversion time. Details of the selection criteria for the R-C combination can be found in the manufacturer's data sheet. The values selected (R=125KΩ, C=100pF) yield a conversion time of about 16 microseconds. This is the minimum conversion time recommended by the manufacturer.

The ADC is selected (enabled) when a logic Ø appears at the $\overline{CS}$ input (pin 16). The state of the $\overline{RD}$ input (pin 15) determines what function the I.C. will perform once it is selected. A conversion is started when a logic Ø pulse is applied to $\overline{CS}$ while $\overline{RD}$ is at logic 1. When the conversion is complete, the result can be read by pulling both $\overline{CS}$ and $\overline{RD}$ to logic Ø. This enables the tri-state buffered data outputs (pins 6-13) which are directly connected to the microcomputer's data bus.

The AD7574JN requires a reference voltage (Vref) of -10V. This reference voltage is generated by the reference supply shown in Figure 3-2. An Analog Devices AD580 provides a precision voltage of +10 volts. A unity gain inverter using a CA3140 op-amp follows the AD580. This inverts the +10.0V into our desired -10.0V reference. A pnp transistor placed in the inverter's feedback loop allows the circuit to tolerate substantial loading (approx. 100mA). This reference supply circuit is located on the A/D board.

The AD7574JN accepts an analog input in the 0.0V to 10.0V range. It generates an 8-bit number proportional to this input. The analog input is applied to pin 4 of the I.C.

Since many of the existing analog signals at the Satellite Communications Facility are in the -5V to +5V range, a conditioning circuit is required to shift each analog signal up by 5V. The analog conditioning board contains 16 such conditioning circuits. Figure 3-3 shows the schematic diagram of one of the 16 identical conditioning circuits. The first stage is a simple unity gain inverter. An offset adjust voltage is connected to the non-inverting input of IC1A to allow the elimination of any offset errors introduced in the channel. The value of R_3 was chosen to minimize offset errors caused by the op-amp's input bias currents. Its value of 5.1KΩ is determined by the equation:

$(R_1 \,//\, R_2) \approx R_3 \,//(R_4 \,//\, R_5)$

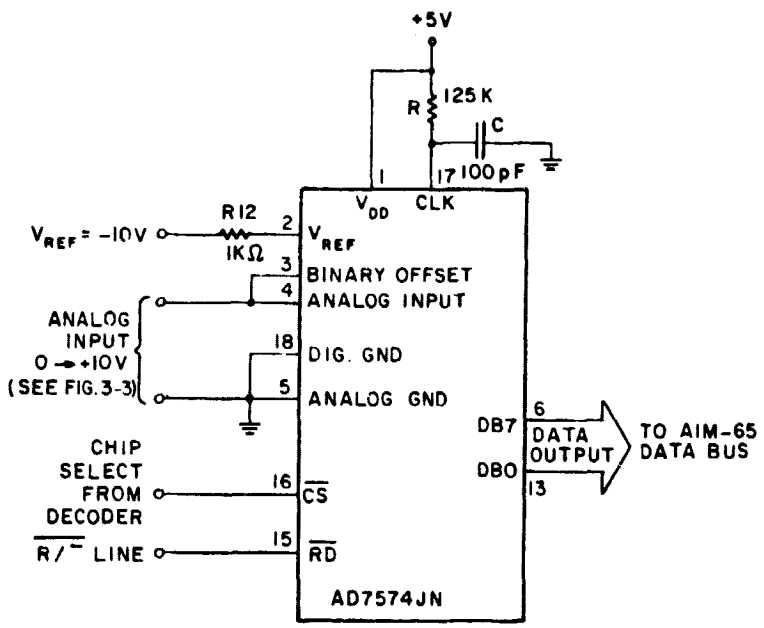


Figure 3-1 AD7574JN Configuration

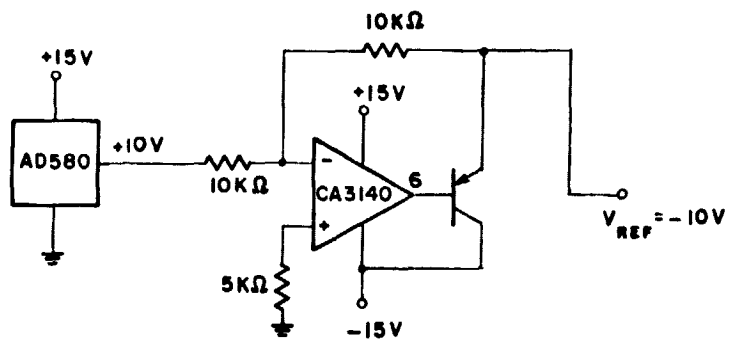


Figure 3-2 Reference Supply

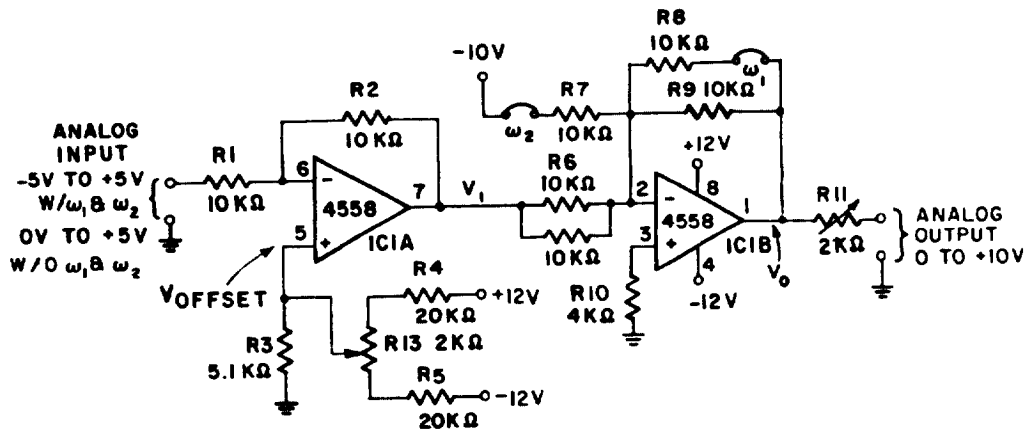


Figure 3-3 Analog Conditioning Circuit

If this equation is approximately satisfied, any voltages caused by input bias currents will appear as common mode signals to the op-amp and will therefore be suppressed. The transfer function of the first stage is given by:

$$V_1 = -V_{in} + V_{offset}^{*} \quad (1)$$

The second stage of the analog conditioning circuit is an inverting amplifier with the transfer function:

$$V_o = -V_1 + 5V \quad (2)$$

Combining equations (1) and (2) gives us the overall transfer function of the conditioning circuit:

$$V_o = -(-V_{in} + V_{offset}) + 5V \quad (3)$$

or

$$V_o = V_{in} + 5V - V_{offset}$$

Since a few analog signals with limits of 0V to 5V were anticipated, jumpers w_1 and w_2 were added to the conditioning circuit. Removing these jumpers provides a second stage transfer function of:

$$V_o = -2V_1 \quad (4)$$

Combining equations (1) and (4) yields the overall conditioning circuit transfer function:

$$V_o = 2V_{in} - 2V_{offset}$$

Each of the op-amps in the conditioning circuit requires several

---

* Note that V_{offset} can be either positive or negative.

resistors which meet at the op-amp's inverting input. Single in-line resistor networks are used here since they are small, inexpensive, and track well with varying temperature.

Potentiometer R11 provides a gain adjustment (GAIN ADJ.) control. This potentiometer, in conjunction with resistor R12 in Figure 3-1 allows the ADC's input gain to be adjusted during the calibration procedure.

The output of each analog conditioning circuit is brought to the corresponding ADC through a short length of sub-miniature coaxial cable. A disconnect for each cable is provided on the conditioning board.

The A/D board occupies a slot in the AIM-65 expansion motherboard. The interfacing between the ADC's and the AIM-65 buses is done on the A/D board. The address decoding techniques used to select the ADCs are very similar to those used in the digital I/O boards. Therefore, the discussion of this decoding will not be quite as detailed as that presented for the digital I/O boards in Chapter 2.

Figure 3-4 shows a schematic diagram of the ADC decoding system. Two 4-bit magnitude comparators do the preliminary decoding. The A inputs to the comparators examine address lines A_{11}-A_4. The B inputs are all grounded. The B inputs corresponding to A_{11}, A_{10}, and A_9 are grounded by jumper wires on the A/D board. By jumpering one or more of these B inputs to +5V (logic 1), the A/D boards address range can be changed. This allows multiple A/D boards to be used, each with a unique address range.

INPUT CONFIGURATION*			ADDRESS RANGE OF A/D BOARD
C	B	A	
0	0	0	8ØØØ - 8ØØF
0	0	1	82ØØ - 82ØF
0	1	0	84ØØ - 84ØF
0	1	1	86ØØ - 86ØF
1	0	0	88ØØ - 88ØF
1	0	1	8AØØ - 8AØF
1	1	0	8CØØ - 8CØF
1	1	1	8EØØ - 8EØF

* 0=0Volts, 1=+5Volts

TABLE 3-1

The A=B output of this 8-bit comparator is connected to the G1 "enable" input on each of two 74LS138 1 of 8 decoders (I.C. 3 and I.C. 4). $\overline{CS8}$ is connected to the $\overline{G2A}$ "enable" input of each decoder. A3 is connected to the $\overline{G2B}$ "enable" input of I.C. 3. A3 is inverted before it is applied to the $\overline{G2B}$ "enable" input of I.C. 4. A particular decoder is only enabled when G1 is at logic 1, and both $\overline{G2A}$ and $\overline{G2B}$ are at logic Ø. When one of the two decoders is enabled, it selects one of eight ADCs by pulling the ADC's $\overline{CS}$ line to logic Ø.

TABLE 3-1 lists the address range corresponding to each possible configuration of the three jumper wires mentioned above. Details of the decoding operation might be more thoroughly understood if a decoding example similar to the one given in Chapter 2 is worked. Since the procedure is so similar, such an example will not be presented here but the results shown in TABLE 3-1 can easily be verified.

As mentioned in the beginning of this section, each ADC has only two control lines, $\overline{CS}$ and $\overline{RD}$. The address decoders (just discussed) control the ADCs' $\overline{CS}$ (Chip Select) lines. The $\overline{RD}$ line should be at logic Ø during a "read" operation, and logic 1 during a "write" (start conversion) operation. The microcomputer's $R/\overline{W}$ line has these desired states and therefore it is directly connected to the $\overline{RD}$ pin of each ADC. Thus, when the computer performs a STA (write) operation to an ADC, the $\overline{CS}$ line of that ADC is pulled to logic Ø, while the $\overline{RD}$ line is at logic 1. This causes a conversion to start in this ADC.

When the computer "reads" an ADC (an LDA operation), both the $\overline{CS}$ and $\overline{RD}$ lines of that ADC are set to the logic Ø state. This enables the data outputs of the ADC. The 8-bit output is buffered by a 74LS245 octal line transceiver (I.C. 5) before being applied to the AIM-65 data bus. A few simple logic gates (I.C. 6) are used to insure that the transceiver isolates the ADCs from the data bus when no ADC is being read.

If we look at the programming examples given in the previous section, we can now analyze how the circuit shown in Figure 3-4 operates. Let us first assume that the C,B, and A inputs to I.C. 1 are grounded. TABLE 3-1 tells us that our A/D board is addressed by 8ØØØ - 8ØØF. Let's suppose we wish ADC #1 to perform a conversion. Our first program

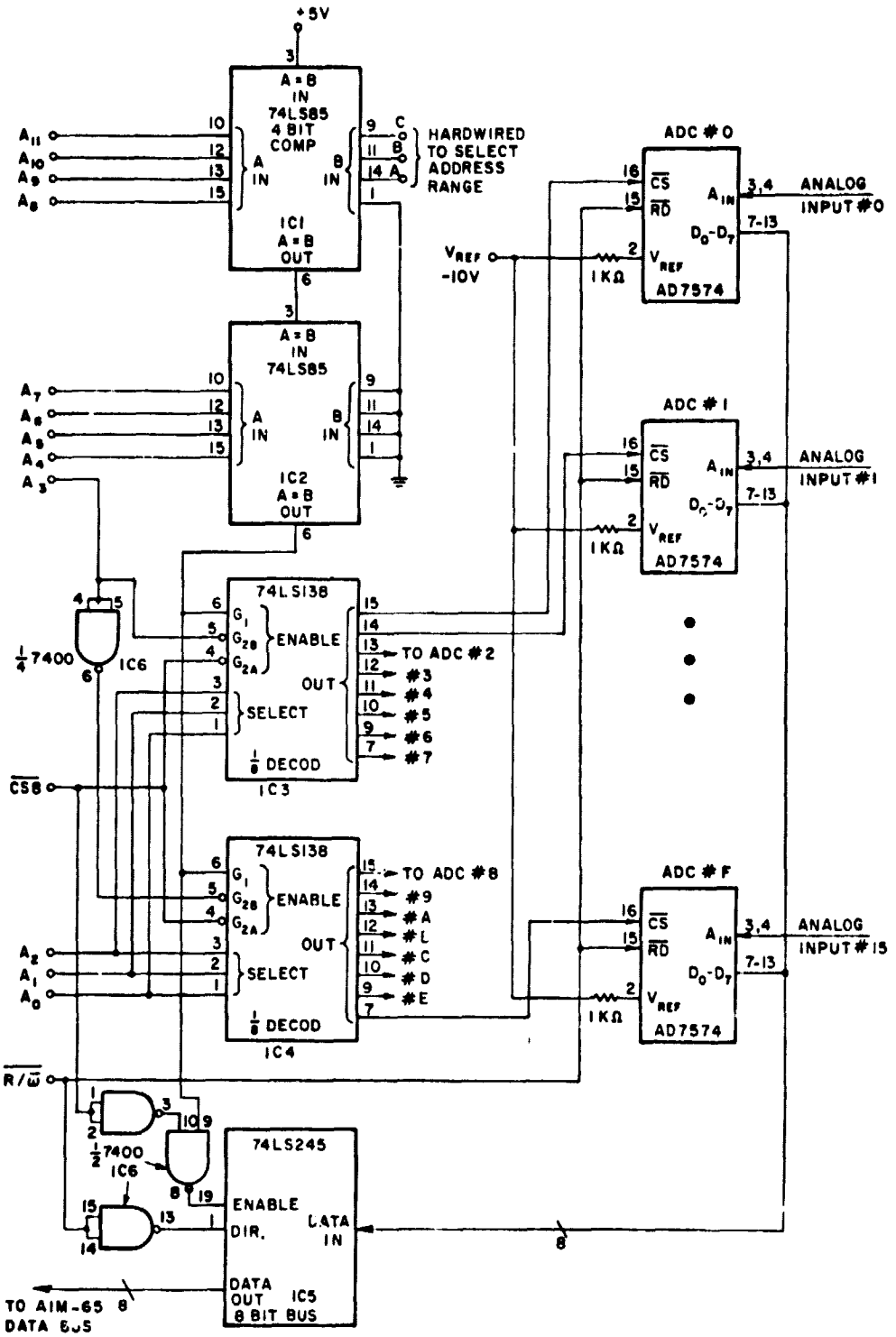


Figure 3-4 A/D Converter System

step would be a STA 8ØØ1 operation. When 8ØØ1 appears on the address bus, $\overline{CS8}$ is at logic Ø, our comparator's output (pin 6, I.C. 2) is at logic 1 and A3 is at logic Ø. Thus I.C. 3 is enabled. Since (A_2, A_1, A_0) = (0,0,1), pin 14 of I.C. 3 (and therefore $\overline{CS}$-pin 16 of ADC #1) goes to logic Ø. During a STA operation, $\overline{R/\bar{w}}$ is a logic 1 and an A/D conversion is started on ADC #1.

Sixteen microseconds later the A/D conversion is complete. Our LDA 8ØØ1 instruction causes 8ØØ1 to again appear on the address bus. ADC #1 is again selected; but this time its $\overline{RD}$ pin is at logic Ø. The 8-bit result of the conversion is presented to I.C. 5 (our output buffer). Since pin 6 of I.C. 2 is high, and the $\overline{R/\bar{w}}$ line is low, the enable line to I.C. 5 is pulled to logic Ø and our data is transferred to the computer via the data bus.

Additional details of the analog input subsystem including PC board layouts are given in the appendices.

The entire analog input section of the data aquisition system has been presented in this chapter. We have seen how to calibrate and operate the ADCs under computer control. We have also seen the theory behind the operation of the controlling logic. This completes the discussion of the main data input sections of the data acquisition system and prepares us for the discussion of the DMA controller which will allow us to store our newly acquired data on magnetic tape.

IV. DMA CONTROLLER AND TAPE CONTROL

A. Introduction

The Direct Memory Access (DMA) Controller described in this Chapter serves as an interface between the AIM-65 and the Pertec FT8840A-9 nine track digital tape recorder. Direct memory access is necessary in this system because the AIM-65 cannot supply data directly from its memory as fast as the tape deck requests it. The DMA controller acts as a buffer into which the computer can store data. When a substantial quantity of data has been stored, the contents of the buffer can be quickly dumped to the tape deck. The DMA Controller can also be used to read a data record from the tape deck and transfer it to the computer.

A block diagram of the DMA Controller is shown in Figure 4-1(a). The heart of the DMA Controller is 4096 bytes of static Random Access Memory (RAM). This memory acts as the data buffer which stores the data to be transferred to the tape. The controller also contains circuitry which controls this memory and allows data to be transferred to the tape deck.

The controller is constructed on an Augat wire-wrap board. A wire list for this board is presented in Appendix B.

This chapter first presents some fundamentals of the operation of the Pertec FT8840A-9 digital tape deck. Understanding these fundamentals will make the DMA Controller's operational theory much easier to understand. Some examples of tape deck operations are presented with flow diagrams and program listings of the programs which perform the operations. The final section of this chapter presents the hardware details of the DMA Controller.

B. Tape Control

The Pertec tape deck has two primary components, the tape transport and the microformatter. The tape transport consists of the electronic and servomechanical systems which cause tape motion and the transfer of data to and from the tape. The microformatter is a device which acts as an interface between the tape transport and the controller (the DMA Controller in this case.)

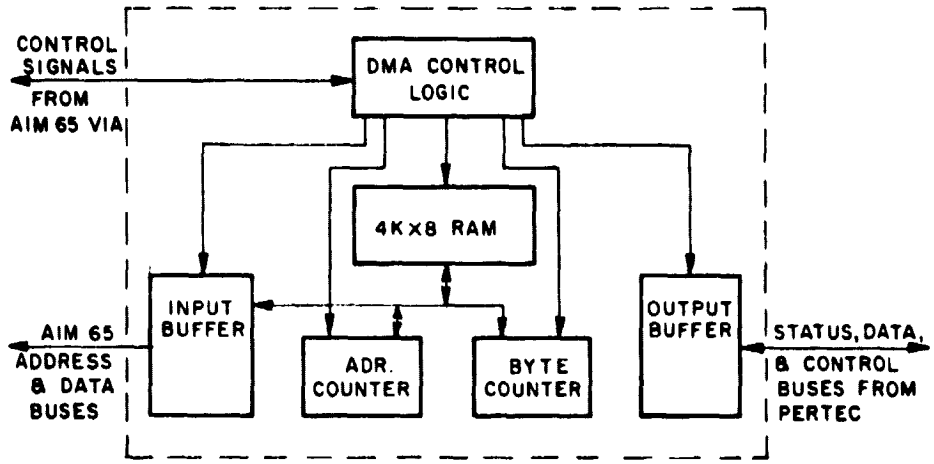


Figure 4-1(a) DMA Controller Block Diagram

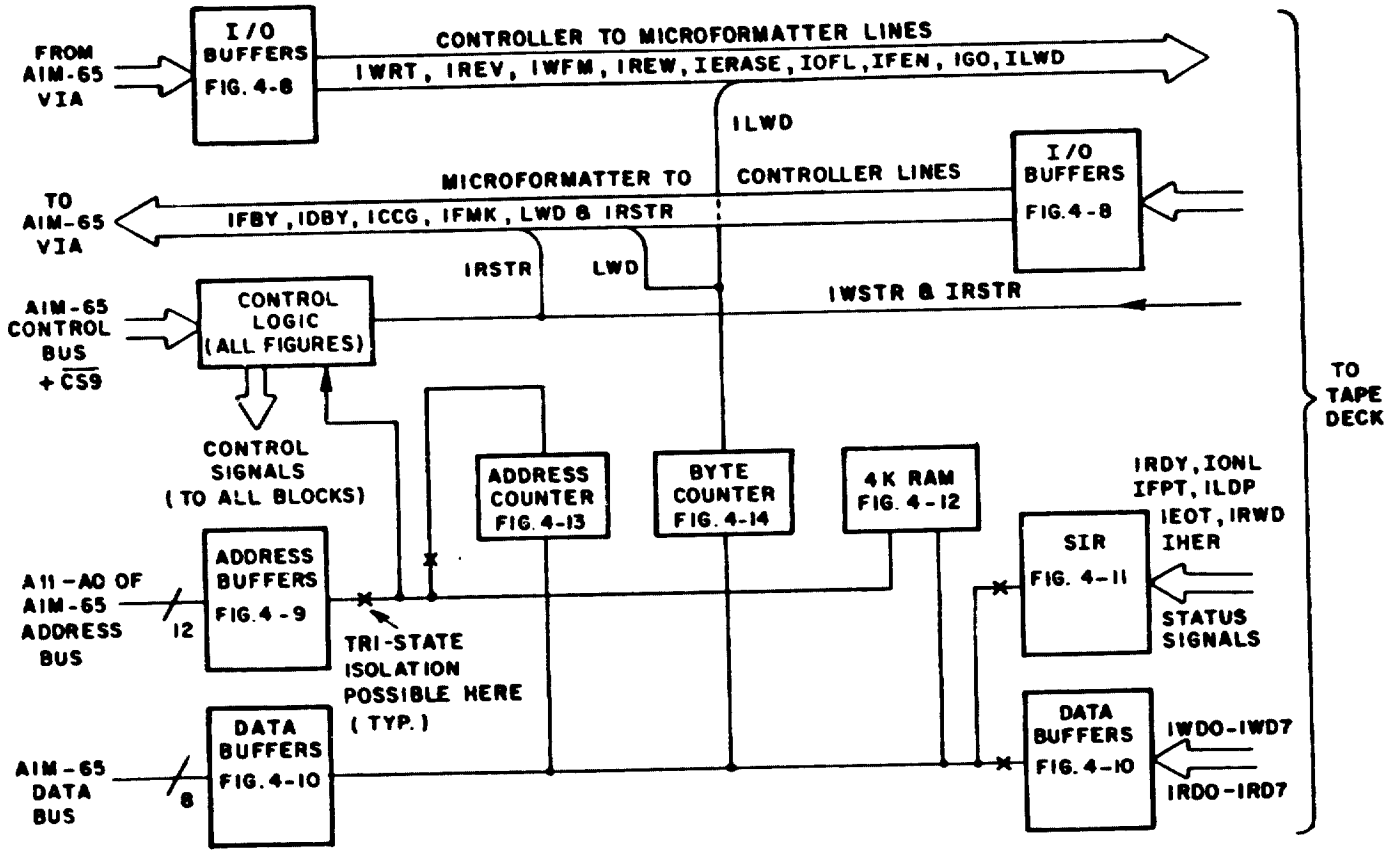


Figure 4-1(b). DMA Controller - Detailled Block Diagram

Figure 4-1(b) shows a detailed block diagram of the DMA Controller. Each block in this diagram has a schematic diagram and discussion explaining it in the final section of this chapter. This diagram shows how all of the DMA Controller's subsystems are integrated.

Figure 4-1(b) also shows the control signals which connect the microformatter and the controller. These signals fall into the following three categories: microformatter to controller signals, controller to microformatter signals, and tape status signals. The tape deck's operating manual, [4], gives a description of each of these signals. A brief summary of the most important signals is given in this section. Note that the microformatter uses negative logic. Thus, a line is considered active (or true) when it is in the logic Ø state.

We will first examine the controller to microformatter signals. Those which concern us are [4]:

IFEN-FORMATTER ENABLE

- This signal, when low (logic Ø) clears the microformatter and enables it to accept a command from the controller.

IWRT-WRITE/READ

- This signal specifies the mode of the system.
  Logic Ø = Write
  Logic 1 = Read

IREV-REVERSE/FORWARD

- This signal specifies forward or reverse tape motion.
  Logic Ø = Reverse
  Logic 1 = Forward

IWFM-WRITE FILE MARK

- This pulse causes a Write File Mark to be written on the tape if IWRT is also low.

IERASE-ERASE

- This signal, when low, cuases a length of tape to be erased if IWRT is low. If IWRT is high (logic 1) a record will be spaced over instead of being erased.

ILWD-LAST WORD

- During a write operation this pulse is used to tell the microformatter that the word which is being sent from the controller is the last word in the record.

IGO-INITIATE COMMAND

- This pulse initiates a command. It is given to the microformatter after all of the necessary input signals to the microformatter have been set to their desired states.

IWØ-IW7-WRITE DATA LINES

- These are the lines on which the data to be written are transferred to the microformatter.

The tape deck can perform many tape operations. Table 4-1 shows the operations with which we are most concerned. The operations listed in Table 4-1 are self-explanatory and details can be found in the Pertec Microformatter Addendum [4]. A tape operation is initiated by setting the control lines for the desired operation and then sending the IGO pulse.

In addition to the controller to microformatter signals, several microformatter to controller signals also exist. These signals allow the computer to monitor the tape deck as it performs an operation. The most important of these microformatter to controller signals are [4]:

IFBY-FORMATTER BUSY

- This line is low between the time of the IGO command and the time at which tape motion ceases after the excecution of a command.

IHER-HARD ERROR

- This is a pulse which indicates that a parity error has occurred during either a read or write operation.

IWSTR-WRITE STROBE

- This line is pulsed (to logic Ø) each time a word (byte) is written onto the tape.

TABLE 4-1 Microformatter Commands

Operations	IREV	IWRT	IWFM	IERASE
READ FORWARD	1	1	1	1
READ REVERSE	Ø	1	1	1
WRITE	1	Ø	1	1
WRITE FILE MARK	1	Ø	Ø	1
SPACE FORWARD	1	1	1	Ø
SPACE REVERSE	Ø	1	1	Ø
FILE MARK SEARCH FORWARD	1	1	Ø	Ø
FILE MARK SEARCH REVERSE	Ø	1	Ø	Ø

TABLE 4-2 - AIM-65 VIA Port Assignment

Port Address	Bit	Function
AØØØ	0	IFBY
	1	IDBY
	2	ICCG
	3	LWD
	4	IFMK
	5	-
	6	IRSTR
	7	IGO
AØØ1	0	$\overline{DMA}$
	1	IWRT
	2	IREV
	3	IWFM
	4	IREW
	5	IERASE
	6	IOFL
	7	IFEN

IRSTR-READ STROBE

- This line is pulsed each time a word is read from the tape.

IFMK-FILE MARK

- This line is pulsed whenever the microformatter encounters a file mark.

IRØ-IR7-READ DATA LINES

- These are the lines on which the data which are read are transferred to the DMA controller.

Several of these control signals are connected to the VIA on the AIM-65 board. The port assignments for this VIA are given in Table 4-2. The signals shown in Table 4-2 which were not mentioned above are discussed in the Pertec Microformatter Addendum [4]. These signals are not used in the elementary tape operations which we will discuss. They are, however, connected to the AIM-65 to allow the excecution of some commands which are not presented in this chapter.

Most of the lines shown in Table 4-2 have been mentioned already; but there are a few which must be explained before the tape deck operating procedures can be presented. The most important of these signals is the line labled $\overline{DMA}$. This line determines in which of two modes the DMA controller will operate. When this line is in the logic 1 state the DMA controller is connected to the AIM-65 and isolated (by tri-state buffers) from the tape deck. In this mode the 4Ø96 bytes of memory on the DMA controller board can be accessed by the computer. This memory occupies the address range 9ØØØ-9FFF (see memory map in Appendix A).

When the $\overline{DMA}$ line is in the logic Ø state, the computer is isolated from the DMA memory (again by tri-state buffers) and the tape deck is connected to this memory. It is in this mode that data are transferred between the magnetic tape and the DMA controller. Since the $\overline{DMA}$ line can only be in one of the two states at a particular instant, it is impossible for both the computer and the tape deck to have control of the DMA memory at the same time.

The DMA memory occupies locations 9ØØØ through 9FFF in the AIM-65's memory map. Several other devices, which occupy addresses in the 83ØØ through 83Ø7 range, are also located on the DMA controller board. These devices include two 12-bit counters, two flip-flops which the computer controls, and a register which contains the status signals from the tape deck. These devices and the significance of their addresses are explained as the need arises during the following discussion of the DMA Controller's operation.

The DMA Controller can transfer the contents of any contiguous block of its memory to the magnetic tape. To dump a block of memory to the tape, the starting address and block length must first be loaded into the DMA controller. The $\overline{\text{DMA}}$ line is then pulled to the logic Ø state (by writing a Ø bit into bit Ø of AØØ1). The computer sets up and starts the tape deck, waits for the write operation to be complete, and then regains command of the DMA Controller by returning the $\overline{\text{DMA}}$ line to the logic 1 state.

The starting address is loaded into the DMA controller by writing the address into the starting address counter. Since all of the memory on the DMA controller board is in the 9ØØØ-9FFF range, the leading 9 need not be loaded into the counter. Only the last 3 hexadecimal digits (12 bits) must be loaded. Since the AIM-65 is an 8-bit machine, two operations must be performed to load all 12 bits into the counter. Figure 4-2(a) shows the addresses and bit assignments for the starting address counter. The 8 least significant bits are loaded into location 83ØØ and the 4 most significant bits are loaded into bit 3-bit Ø of location 83Ø1.

For convenience, the starting address is normally set to be the first byte of memory, 9ØØØ. Thus $(ØØØ)_{16}$ should be loaded into the starting address counter. This can simply be accomplished by the following program segment:

```
LDA #ØØ     A←(ØØØØØØØØ)
STA 83ØØ    Adr. Ctr.-Lo←A
STA 83Ø1    Adr. Ctr.-Hi←A
```

	Address	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit Ø
(a)	83ØØ	Starting Adr. CTR-LO	AC 7	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC Ø
	83Ø1	Starting Adr. CTR-HI	X	X	X	X	AC 11	AC 1Ø	AC 9	AC 8
(b)	83Ø3	BYTE CTR.-LO	BC 7	BC 6	BC 5	BC 4	BC 3	BC 2	BC 1	BC Ø
	83Ø2	BYTE CTR.-HI	X	X	X	X	BC 11	BC 1Ø	BC 9	BC 8

AC = Address Counter
BC = Byte Counter
X = Don't Care

Figure 4-2 (a) Address Counter
(b) Byte Counter

The number of bytes to be dumped is then loaded into the byte counter of the DMA controller. This is not quite as simple as the address counter loading procedure. The byte counter is a 12-bit counter which is incremented each time a data word is transferred to the tape deck. When the counter reaches the all 1s state (FFF), it sends the IWLD (last word) pulse to the microformatter. This causes the termination of the write operation.

Since the ILWD signal is sent when the counter reaches FFF, the number loaded into the counter must be the difference between FFF and the number of bytes to be dumped. Thus, we must load the counter with the 2's complement of the number of bytes to be dumped. The 2's complement is easily calculated using the formula:

2's Complement = 1's Complement +1.

The 1's complement is obtained by simply replacing each 1 in the number with a Ø and vice versa. Thus if we wanted to dump 1Ø24 bytes we would determine the number to be loaded into the byte counter by finding the 2's complement of 1Ø24 (base 10):

$$(1Ø24)_{\text{Base 10}} = (4ØØ)_{\text{Base 16}} = (Ø1ØØØØØØØØØØ)_{\text{Base 2}}$$

the 1's complement of this number is given by:

$$\overline{(Ø1ØØØØØØØØØØ)} = (101111111111)$$

therefore, the 2's complement is given by:

$$(1Ø1111111111) + 1 = (11ØØØØØØØØØØ) = (CØØ)_{16}$$

Thus, we must load CØØ (hex) into the byte counter to dump 1Ø24 bytes of memory to the tape. Figure 4-2(b) shows the addresses and bit assignments of the byte counter. The 8 least significant bits of the number to be loaded are written into location 83Ø3 and the 4 most significant bits are written into 83Ø2. Thus, a program segment which will set our byte counter to dump 1Ø24 bytes of data is given by:

```
LDA  #ØØ      A←ØØØØØØØØ
STA 83Ø3      BYTE CTR.-LO←A
LDA  #ØC      A←ØØØØ11ØØ
STA 83Ø2      BYTE CTR. -Hi←A
```

When the byte counter reaches the all 1's state, it sends the ILWD pulse to the tape deck and sets a flip-flop on the DMA controller board. The output of this flip-flop, labled LWD, is connected to bit 3 of the AIM-65's input port (see Table 4-2). During a write operation, the computer observes the LWD signal and waits for it to make the low to high transition which signals the completion of the operation. This LWD flip-flop must be manually reset before each write operation. This is done by excecuting the instruction, STA 83Ø5.

Before any tape operation is attempted, the computer must examine several of the status lines which come from the tape deck. The status lines with which we are concerned are [4]:

IRDY-READY

- This line only becomes a logic Ø when the microformatter is ready to accept a command.

IONL-ON LINE

- This line is at the logic Ø state whenever the microformatter is on line.

IFPT-FILE PROTECT

- This line is at the logic Ø state if the tape on the deck has had its file protect ring removed.

ILDP-LOAD POINT

- This line is at the logic Ø state when the tape is at the load point (beginning of tape).

IEOT-END OF TAPE

- This line is at the logic Ø state when the tape has been advanced to its end.

IRWD-REWINDING

- This line is at the logic Ø state while the tape is rewinding.

IHER-HARD ERROR

- A logic Ø pulse appears on this line whenever a word with a parity error is written onto or read from the tape.

The computer can examine the state of each of these lines by reading location 83Ø7. Figure 4-3 shows how these status signals are assigned to the various bits of register 83Ø7. This register is called the Status Input Register (SIR).

Bit 7 of the SIR is labled HER, not IHER. This is due to the fact that, like the ILWD signal, the IHER signal is a short pulse (approx. 20 microseconds). To ensure that this pulse is observable to the computer, the IHER pulse, like the ILWD pulse is used to set a flip-flop. It is the output of this flip-flop, labled HER, that is connected to the SIR. This flip-flop is cleared by executing the operation, STA 83Ø4.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit Ø
HER	-	IRWD	IEOT	ILDP	IFPT	IONL	IRDY

Figure 4-3 - Status Input Register

The status input register transfers data to the computer on the data bus. Since the register is located on the DMA controller board, the $\overline{\text{DMA}}$ line must be in the logic 1 state when the SIR is read.

A memory map summarizing all of the registers which affect DMA operations is shown in Table 4-3. A complete memory map of the entire data acquisition system is shown in Appendix A.

Now that the groundwork has been laid for tape operations, the details will be explained with several examples. We will examine three short programs which will perform these simple tape operations: 1) write a record, 2) space back one record, 3) read a record. A flow diagram and program listing will be presented for each of these programs.

Before we do any operations we must set the directions of the AIM-65's I/O ports so they will be in agreement with the directions shown in Table 4-2. We should also load a logic 1 into each output bit so no tape command is inadvertantly given to the microformatter.

TABLE 4-3 - DMA CONTROLLER MEMORY MAP

ADDRESS	FUNCTION
83ØØ	ADR. CTR. - LO
83Ø1	ADR. CTR. - HI
83Ø2	BYTE CTR. - HI
83Ø3	BYTE CTR. - LO
83Ø4	LWD flip-flop CLR
83Ø5	HER flip-flop CLR
83Ø7	Status Input Reg. (Figure 4-3)
9ØØØ ↓ 9FFF	DMA memory (4K x 8)
AØØØ ↓ AØØ1	AIM-65's I/O Ports (Fig. 4-2)

This is done by the following program:

```
LDA      #FF      A←(11111111)
STA      AØØ3     DIRA←A
STA      AØØ1     A Port←A
LDA      #80      A←(1ØØØØØØØ)
STA      AØØ2     DIRB←A
STA      AØØØ     B Port←A
BRK
```

Before each tape operation, we must check the tape deck status to insure that the deck is ready to accept our instruction. Since this must be done before each operation, it might be wise for us to write a subroutine which checks the tape status. This subroutine (or subprogram) can then be called from any of our 3 programs. This eliminates the need for an identical status check program segment in each of the 3 programs. This subroutine will act as a good first example since it will be quite short and simple.

Figure 4-4 shows a flow diagram for our status check subroutine. The program first checks to see if we are at the end of the tape or off line. If either of these conditions exists, the program is aborted and the machine is told to stop. The computer then examines the IFBY line to see if the microformatter is busy. If it is busy, the computer goes back and checks it again. This is repeated utnil the microformatter is no longer busy. The computer then checks the IRDY line. If the deck isn't ready, the computer waits until it is before returning. A listing of our status check subroutine is given below:

```
Ø22Ø    LDA    8307  }
Ø223    AND     #1Ø  }    Stop if end of tape
Ø225    BNE    Ø228  }
Ø227    BRK
Ø228    LDA    8307  }
Ø22B    AND     #Ø2  }    Stop if off line
Ø22D    BEQ    Ø23Ø  }
Ø22F    BRK
Ø23Ø    LDA    8307  }
Ø233    AND     #Ø1  }    Wait if busy
Ø235    BNE    Ø23Ø  }
```

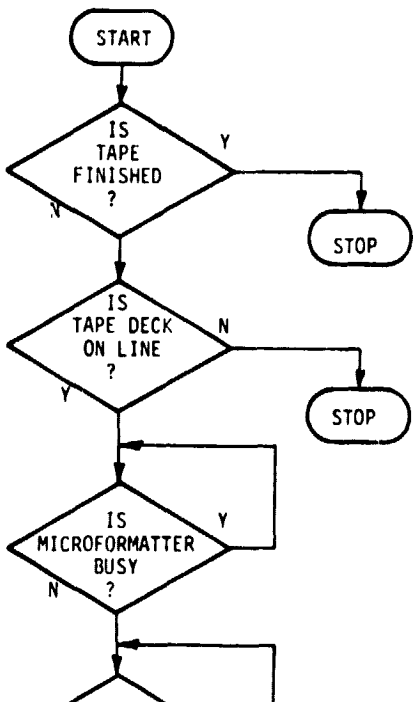


Figure 4-4 Status Check Subroutine Flowchart

```
0237    LDA    A000
023A    AND     #01    Wait if not ready
023C    BEQ    0237
023E    RTS           Return
```

This subroutine occupies locations 0220 through 023E in the AIM-65's memory. There is no special significance to this address range, but each of the programs presented in this chapter occupies a unique range. This subroutine can be excecuted by a JSR 0220 operation in each of our 3 programs.

The first program to be presented is the write program. A flow diagram is shown in Figure 4-5. Our first step is to call the status subroutine. If the tape deck is ready for our instructions we then load the address and byte counters and clear the HER and LWD flip-flops. The necessary control lines (IWRT and IFEN) and the $\overline{\text{DMA}}$ line are then pulled down to the logic 0 state, the IGO pulse is sent, and the operation begins. When the last word has been sent and the formatter is no longer busy, all of the control lines are returned to the logic 1 state and we are finished. A program listing for the write program is given below:

```
0300    JSR    0220    Check status
0303    LDA     #00
0305    STA    8300
0308    STA    8301    (00)  into Adr. Ctr.
030B    STA    8303    (C0)  into Byte Ctr.
030E    LDA     #0C
0310    STA    8302
0313    STA    8304    Clear HER & LWD flip-flops
0316    STA    8305
0319    LDA     #7C
031B    STA    A001    IWRT & IFEN & DMA to logic 0
031E    LDA     #00
0320    STA    A000    Send IGO pulse
0323    LDA     #80
0325    STA    A000
0328    LDA    A000
032B    AND     #09    Wait for IFBY=LWD=logic 1
032D    CMP     #09
032F    BNE    0328
0331    LDA     #FF    IWRT & IFEN & DMA to logic 1
0333    STA    A001
0336    BRK           Stop
```

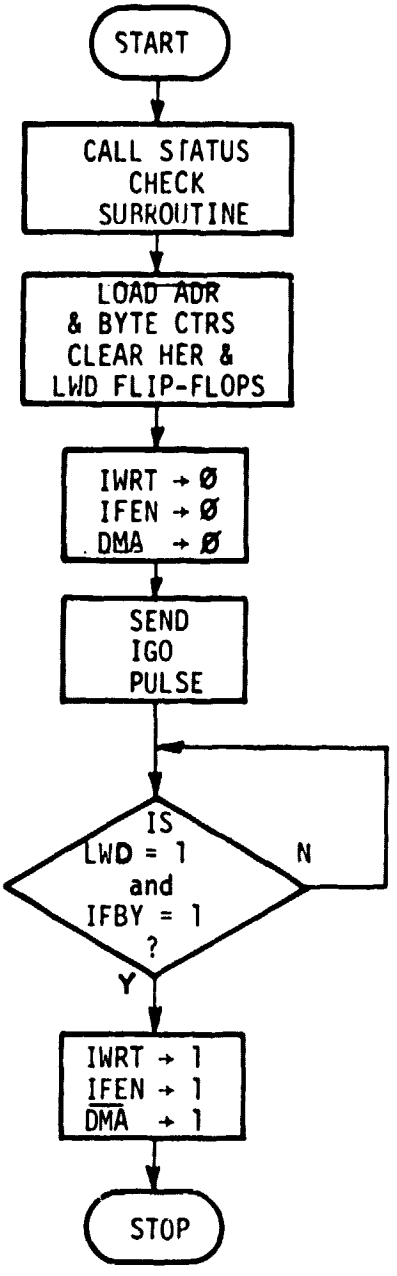


Figure 4-5. Write Operation Flow Diagram

If a parity error occurred during the write operation, the HER flip-flop will have been set and a logic 1 will appear in bit 7 of the SIR. If such an error occurs we might want to space back 1 record and try to write the record again. This exemplifies the need for our "space back 1 record" program.

A flow diagram for this second program is given in Figure 4-6. We first call the status check subroutine. If the deck is ready for our instructions, we set IFEN, IERASE, and IREV (see Table 4-2) to logic Ø and send the IGO pulse. When the tape has spaced back by one record and stopped, IFBY will go to the logic 1 state and the computer is stopped. A program listing for this program is given below.

```
Ø35Ø    JSR    Ø22Ø        Check Status Check subroutine
Ø353    LDA     #5B}
Ø355    STA    AØØ1}       IFEN & IERASE & IREV to logic Ø
Ø358    LDA     #ØØ}       IGO to logic Ø
Ø35A    STA    AØØØ}
Ø35D    LDA     #8Ø}       IGO to logic 1
Ø35F    STA    AØØØ}
Ø362    LDA    AØØØ)
Ø365    AND     #Ø1}       Go back to Ø362 if IFBY=Ø
Ø367    DEQ    Ø362)
Ø36A    LDA     #FF        IFEN & IREASE & IREV to logic 1
Ø36C    STA    AØØ1
Ø36F    BRK                Stop
```

The third program we will examine will read a record and transfer the data from the tape to the DMA memory. This read program is very similar to our write program. There are a few significant differences which must be discussed. A flow diagram of the read program is shown in Figure 4-7. The byte counter is not used during a read operation. Therefore, only the address counter needs to be preset (normally to ØØØ). The operation begins when the computer pulls the IFEN and $\overline{\text{DMA}}$ lines to logic Ø and sends the IGO pulse. The record is read; and the data are transferred to the DMA memory. Then the tape stops, IFBY becomes a logic 1, IFEN and $\overline{\text{DMA}}$ are returned to logic 1 and the computer stops. A program listing of the read program is given below:

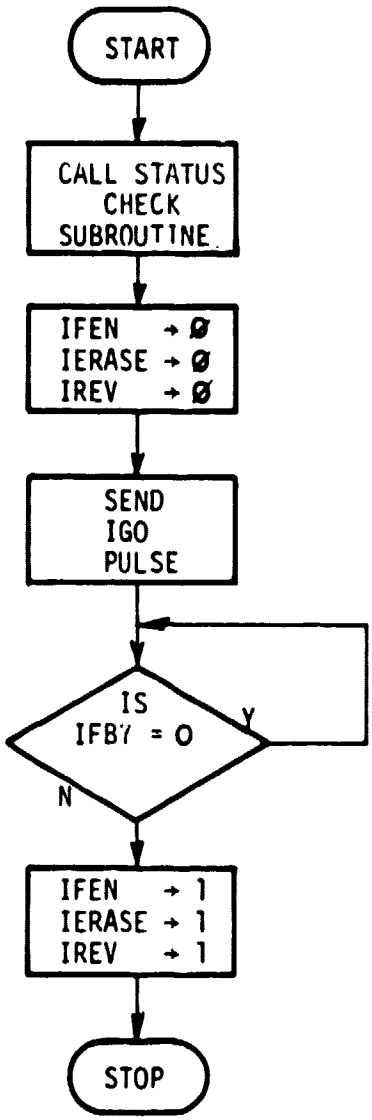


Figure 4-6. Space Back 1 Record Flow Diagram

```
0380    JSR     0220        Check status
0383    LDA      #00
0385    STA     8300        Set Address Ctr. to 000
0388    STA     8301
038B    STA     8304        Clear last word flip-flop
038E    LDA      #7E        IFEN & DMA to logic 0
0390    STA     A001
0393    LDA      #00
0395    STA     A000
0398    LDA      #80        Send IGO pulse
039A    STA     A000
039D    LDA     A000
03A0    AND      #01        Wait until IFBY=logic 1
03A2    BEQ     039D
03A4    LDA      #FF        IFEN & DMA to logic 1
03A6    STA     A00i
0349    BRK                 Stop
```

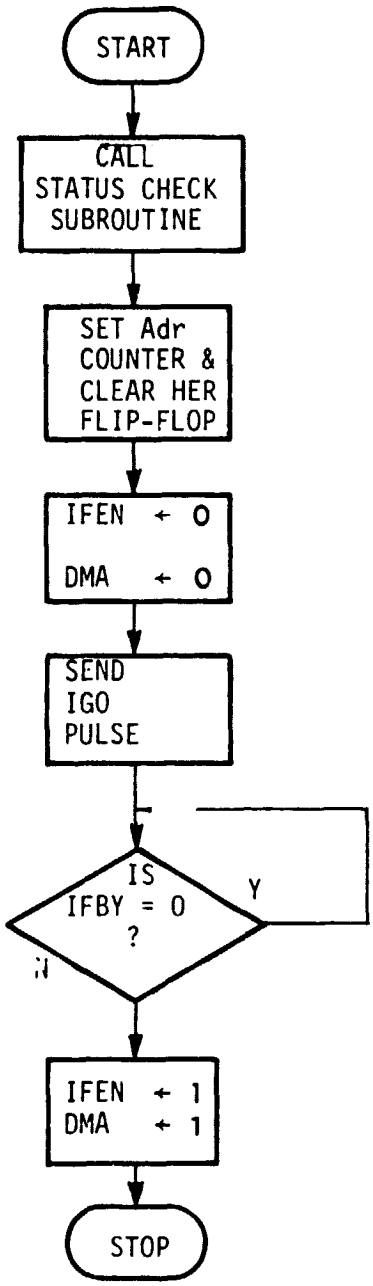


Figure 4-7. Read Program Flow Diagram

The data which are read from the tape are transferred to a contiguous block of the DMA controller's memory. This block starts at the address which was loaded into the address counter (preceded by 9). Thus, if the address counter is preset to ØØØ, this block of data will start at 9ØØØ. The block of data which was transferred to the DMA controller can be read by examining 9ØØØ,9ØØ1,...

Since we don't necessarily know the number of bytes in a record in advance, a method of counting the number of bytes transferred is provided. Each time a byte is transferred, a pulse appears on the IRSTR line. This line is connected to bit 6 of the B Port of the AIM-65's VIA. The VIA has the ability to count pulses on bit 6 of its B Port and this ability can be used to count the incoming read strobes. Details of this counter's operation (timer 2, pulse counting mode) can be found in Chapter 6 of Reference [2].

These 3 programming examples have shown how the computer can use the DMA controller to perform some simple tape operations. All of the tape operations shown in Table 4-1 can be performed using the same basic methods. This section has given all of the operational details necessary to use the DMA controller. The next section will explain the hardware details.

C. Hardware Details

In this section, the hardware details of the DMA controller will be presented. A good understanding of the operational concepts presented in the previous section will make it much easier to understand the hardware features of the DMA controller. The DMA controller consists of several integrated subsystems. This section presents a schematic diagram and description of each of these subsystems.

1. Tape Control Line Buffers

The first subsystem which we will discuss contains the buffers which are inserted into the lines connecting the DMA controller and the tape deck's microformatter. These buffers are shown schematically in Figure 4-8. The buffers give the controller to microformatter lines the ability to drive the microformatter. This buffering is necessary

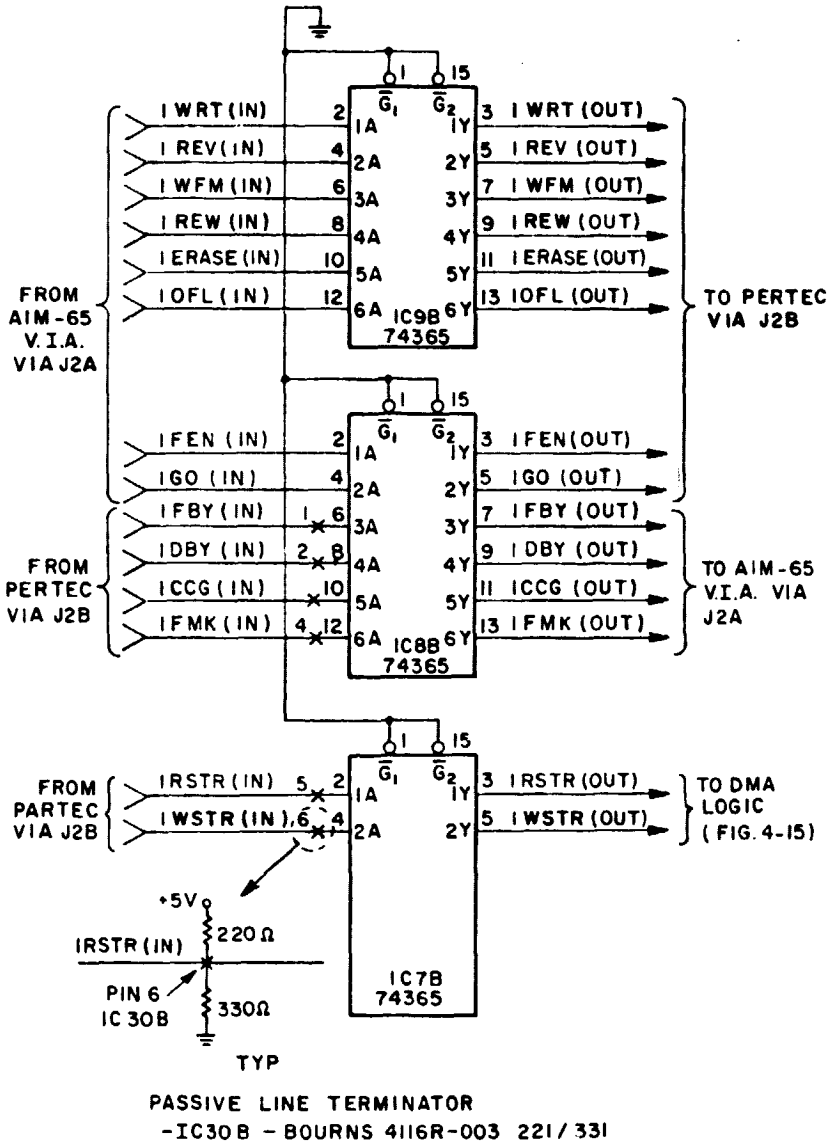


Figure 4-8 DMA Board Control I/O Buffering

because the AIM-65's VIA outputs are each capable of driving only 1 TTL load. The microformatter's inputs present a greater load. Therefore, the controller to microformatter signals are bufferred to give them the extra driving ability they need.

The microformatter to controller lines are also bufferred to reduce the effects of any noise which may have been introduced along the relatively long lines. Passive line terminators, (shown as X's in Figure 4-8) are connected to each line at the input to the buffers. These terminators also help to eliminate noise and are suggested for use in noisy applications by the tape deck manufacturer. This buffering system consists of 3 74365 IC's which are always enabled ($\overline{G}_1$ and $\overline{G}_2$ are permanently tied to ground).

2. The Address Bus

The address bus ($\overline{CS8}$ and A11-AØ) from the AIM-65 is also buffered as it enters the DMA controller. This address buffering is shown in Figure 4-9. Two 74LS365 IC's (IC4A and IC5A) are used for the address buffers. They perform 2 functions. First, they give the address lines the ability to drive several TTL loads. Secondly, they can be used to isolate the AIM-65 from the DMA controller. The buffers go to a high-impedance state (isolating the controller from the computer) whenever the $\overline{G}_1$ and $\overline{G}_2$ inputs go to the logic 1 state. These inputs are tied together and connected to the line labled SYS DMA. This line is the compliment of the $\overline{DMA}$ line from the AIM-65 VIA. Thus, the $\overline{DMA}$ line must be in the logic 1 state for any data transfer between the computer and the DMA controller.

Figure 4-9 also shows three 74LS85 ICs configured as a 10-bit address comparator. The operation of this comparator is analogous to that of the address comparators on the digital I/O and analog input boards. Its output, labled 83ØX, assumes the logic 1 state whenever an address between 83ØØ and 83ØF appears on the address bus. This line is used to select several of the registers shown in the DMA controller's memory map (Table 4-3). Details of this selection process will be presented later in this section.

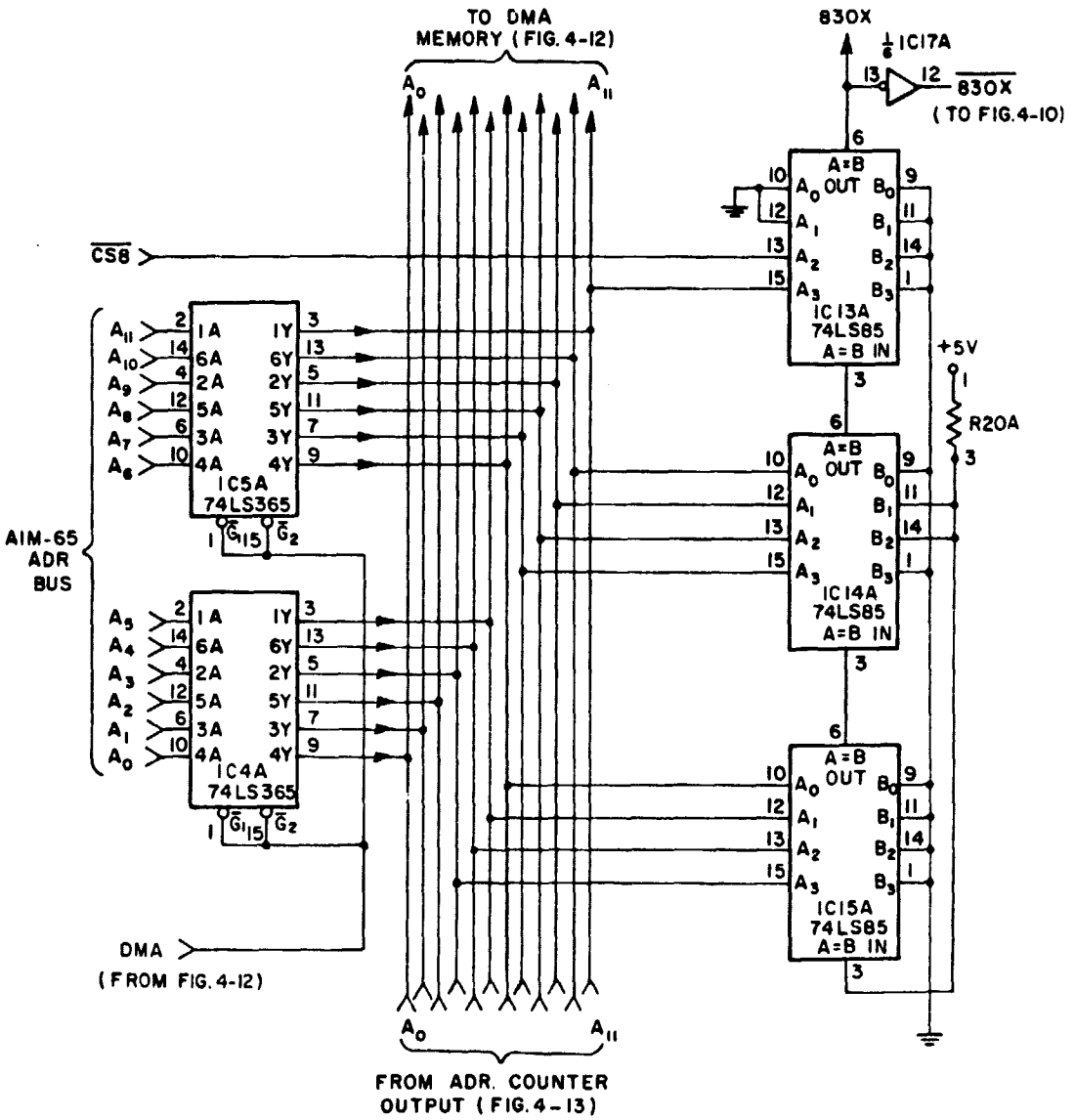


Figure 4-9 DMA Address Buffering and Partial Decoding

3. The Data Bus

Figure 4-10 shows the data bus section of the DMA controller. This bus is brought into the controller through a buffer which, like the address buffers, can isolate the AIM-65 from the controller. Similar buffers interface the data bus and the tape deck's data lines. Additional logic provides the necessary enable and direction signals for the buffers.

The AIM-65's data bus is brought into the DMA controller through a 74LS245 octal line transceiver (IC3A). The AIM-65's SYS R/$\overline{W}$ line is connected to the direction input of the transceiver to set the direction of data flow. Data are transferred from the computer to the controller when this line is logic Ø ($\overline{WRITE}$) and transferred in the opposite direction when this line is logic 1 (READ).

IC3A's enable input (pin 19) is connected to the inverted output of ½ of IC 11A. The transceiver is enabled only when: (1) the SYS DMA line is at the logic Ø state (i.e. $\overline{DMA}$=logic 1) and (2) either $\overline{83ØX}$ or $\overline{CS9}$ is at the logic Ø state. Thus, no data can be transferred between the controller and the computer when the $\overline{DMA}$ line from the computer is at logic Ø. The $\overline{CS9}$ line, like the $\overline{CS8}$ line mentioned previously, is an output from the AIM-65's internal address decoders. It assumes the logic Ø state whenever an address between 9ØØØ and 9FFF appears on the computer's address bus. This is the address range reserved for the DMA controller's memory. The $\overline{83ØX}$ line is the complement of the 83ØX line shown in Figure 4-9. Thus, IC3A is only enabled when we address a register in either the 83ØØ-83ØF or the 9ØØØ-9FFF range (while $\overline{DMA}$=1).

Figure 4-10 also shows 2 bidirectional data buffers which connect the DMA controller's data bus to the data lines from the tape deck's microformatter. These buffers are only enabled when the SYS DMA line is at the logic 1 level ($\overline{DMA}$=logic Ø). The state of the IWRT line determines the direction of data transfer. When IWRT=logic Ø, data are transferred from the controller to the microformatter. When IWRT=logic 1, data are transferred in the opposite direction.

The data bus also connects the computer with the Status Input

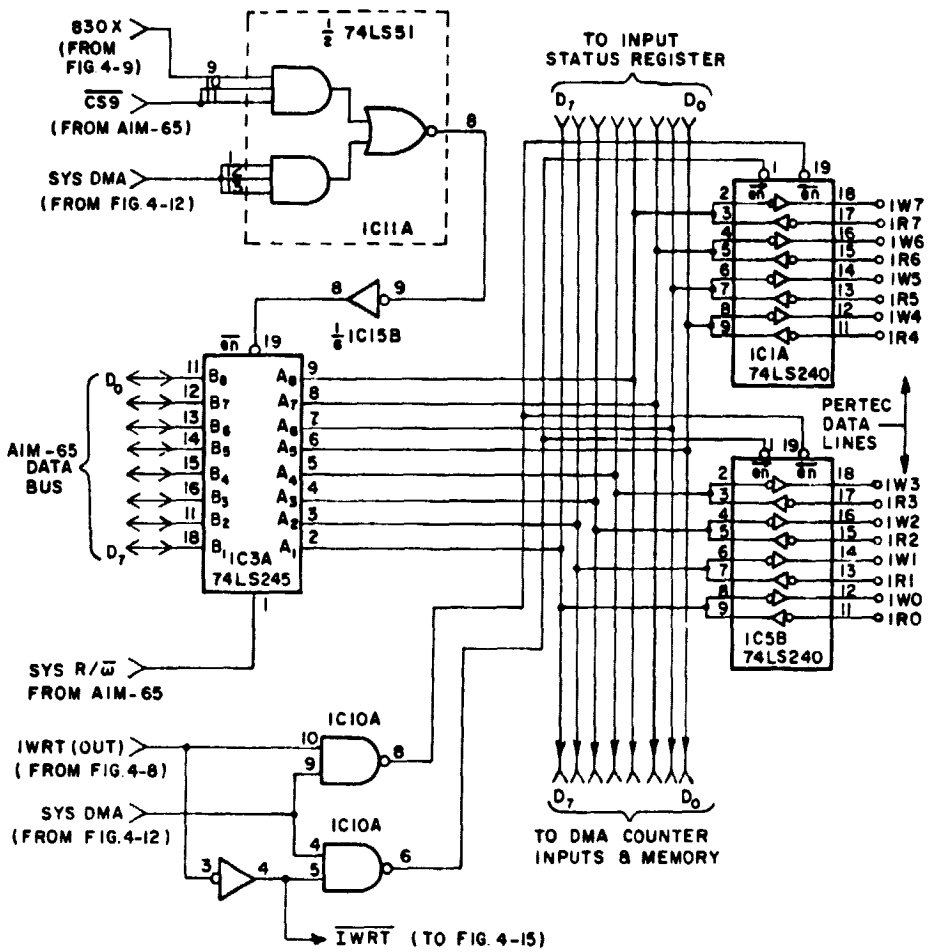


Figure 4-10 Data Buffering

Register, the DMA memory, the address counter and the byte counter. The details of each of these subsystems will be examined next.

4. Status Input Register

Figure 4-11 shows the schematic diagram of the Status Input Register (SIR). The status signals from the tape deck are connected to the B inputs of a 74LS245 octal line transceiver (IC2A). The direction input of this IC is tied to ground so the direction of data flow is fixed (B1-B8 are inputs, A1-A8 are outputs). The outputs of this transceiver are directly connected to the data bus.

The SIR is only enabled when 83ØX, A2, A1, and AØ are all at the logic 1 state. This occurs when 83Ø7 appears on the address bus. When these 4 lines are all at logic 1 the A=B output of IC12A (pin 6) assumes the logic 1 state. This A=B output is inverted by 1/6 of IC15B. This inverted output enables IC2A (the SIR). Thus, when the computer excecutes a LDA 83Ø7 instruction, IC2A is enabled and the status signals appear on the data bus to be transferred to the accumulator.

The IHER signal from the tape deck is a logic Ø pulse which is sent each time a parity error occurs during a read or write operation. This IHER pulse is used to preset a flip-flop (½ of IC2ØB). The output of this flip-flop is connected to a SIR input to allow the computer to detect a parity error after a tape operation is complete. The clearing of this flip-flop will be discussed shortly. Note that since the SIR is connected to the buffered data bus, the $\overline{\text{DMA}}$ line must be a logic 1 for the SIR to be read by the computer.

5. The DMA Memory

Figure 4-12 shows the memory which is the heart of the DMA controller. This memory is comprised of eight 2114-L static RAM ICs. Each IC contains 1Ø24 4-bit registers (1K x 4). They are connected in pairs (IC1B with IC11B, IC2B with IC12B, IC3B with IC13B, IC4B with IC14B), and each pair then has 1Ø24 8-bit registers. Thus, there is a total of 4Ø96 eight-bit memory registers.

The data lines from the RAM ICs are directly connected to the buffered data bus. Ten buffered address bits, A9-AØ, are connected to the address inputs of each memory IC. These ten lines address the

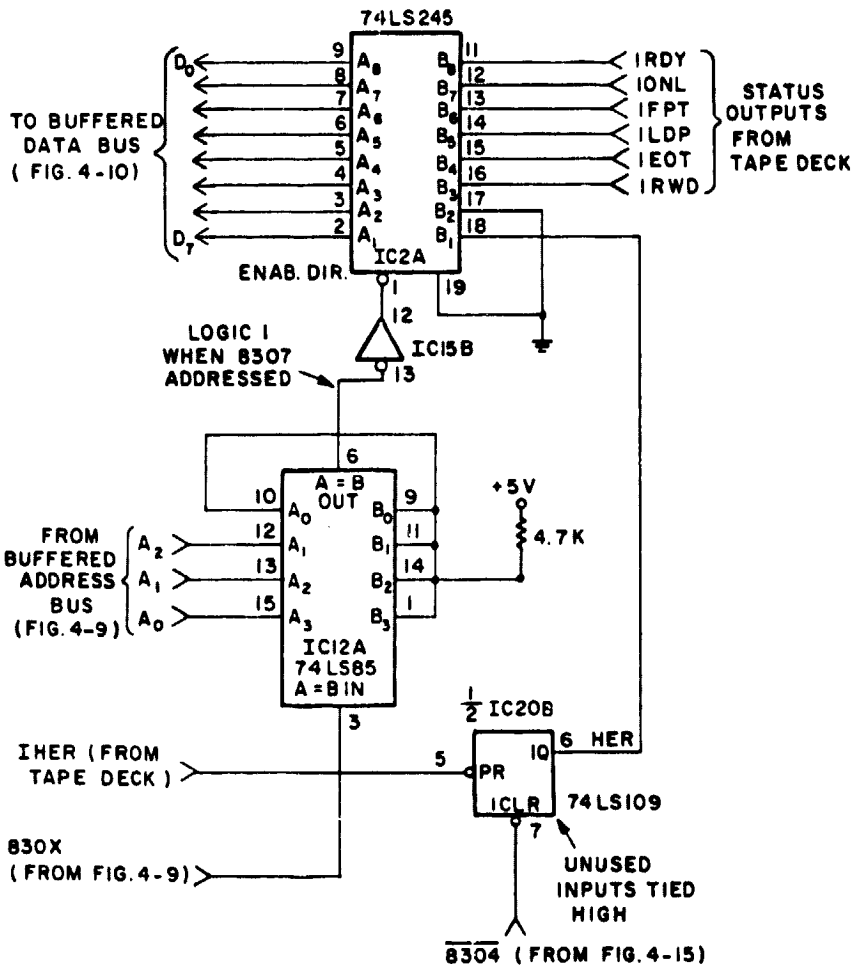


Figure 4-11 Status Input Register

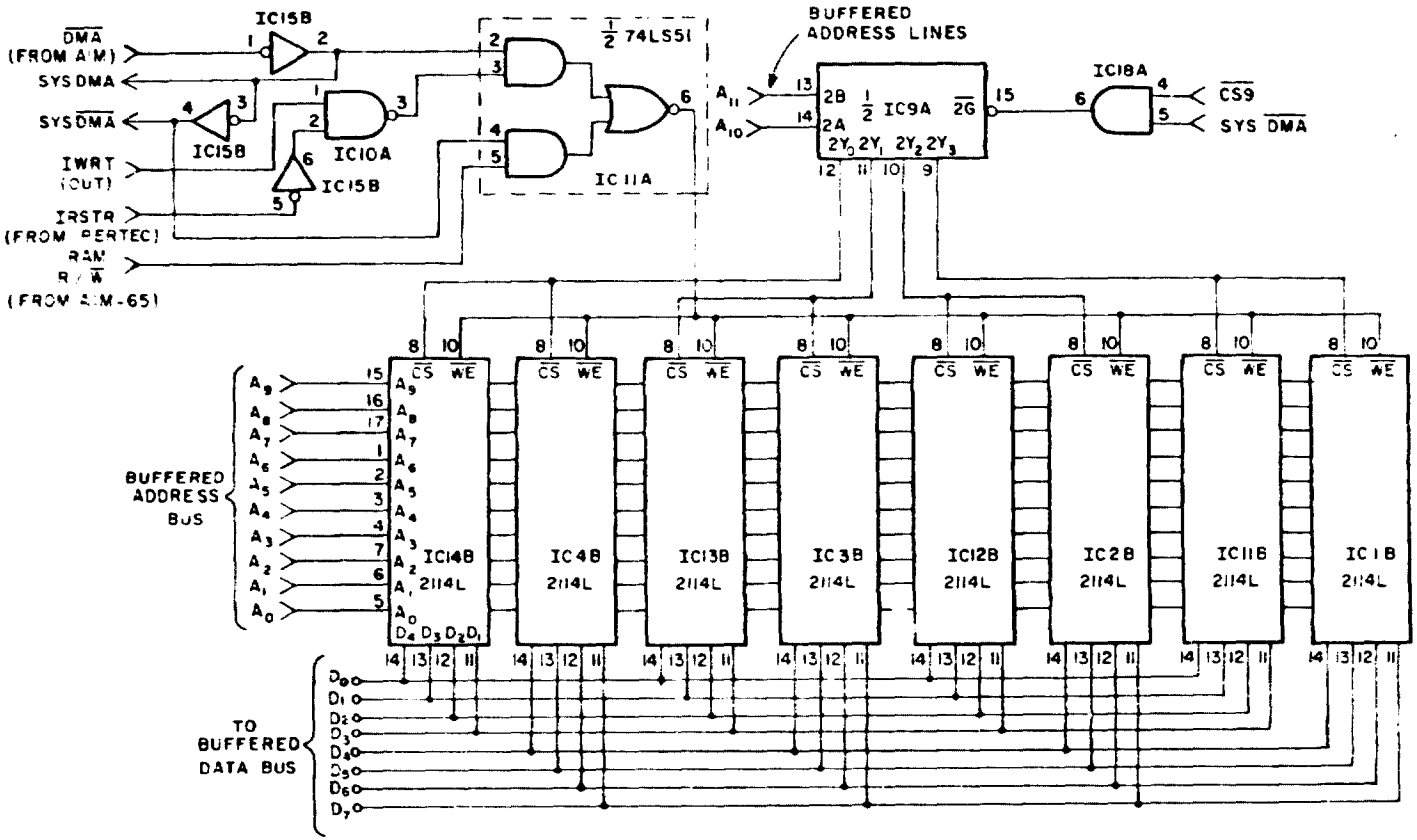


Figure 4-12 DMA Memory and R/W Logic

1Ø24 (=2^{10}) registers within each memory IC. Our two most significant bits, A_{11} and A_{10}, are used to select one of the four pairs of memory ICs. A 1 of 4 decoder (½ of IC9A) is used for this selection process. This decoder (and thus the memory) is enabled when $\overline{CS9}$ or $\overline{DMA}$ is at the logic Ø state, i.e., when reading or writing from 9ØØØ-9FFF or during DMA operations. When the decoder is enabled, it examines its 2A and 2B inputs which are connected to A_{10} and A_{11}. The state of these two lines determines which 1 of the 4 outputs (2YØ, 2Y1, 2Y2, & 2Y3) will be pulled low. When one of the outputs is low, a pair of memory ICs is selected by pulling their $\overline{CS}$ (Chip Select) inputs (pins 8) to logic Ø. Table 4-4 summarizes the memory selection process.

The read/write control for the memory system is provided by parts of IC15B, IC10A, and IC11A. The output of this read/write logic (pin 6 of IC11A) is connected to the $\overline{WE}$ (Write Enable) line of each memory IC (pin 10). Table 4-5 summarizes the function of the read/write logic. When the $\overline{WE}$ and $\overline{CS}$ to a memory IC are both low, the data on the data bus are transferred to the memory register which has been selected (i.e., the data word is written into the memory). When $\overline{CS}$ is a logic Ø while $\overline{WE}$ is logic 1, the contents of the selected memory register is presented to the data bus, i.e., the memory is read.

Table 4-5 shows us that when the $\overline{DMA}$ line is at logic 1, the $\overline{WE}$ line assumes the state of the computer's RAM R/$\overline{W}$ line and the computer provides the read/write signal. During a DMA operation (i.e. $\overline{DMA}$ = logic Ø), the $\overline{WE}$ line only becomes a logic Ø when the IWRT line is a logic 1 (we are reading from the tape) and a read strobe is being sent. Thus a data word will be transferred from the data bus to the memory at each read strobe during a tape read operation.

Figure 4-12 also shows where the SYS DMA and SYS $\overline{DMA}$ lines are generated. The $\overline{DMA}$ line is brought from the computer to pin 1 of IC15B. This signal is inverted twice and the outputs of these inverters are used as our SYS DMA and SYS $\overline{DMA}$ signals. These signals are then used in several places throughout the DMA controller. Note that the SYS $\overline{DMA}$ line is at the same state as the $\overline{DMA}$ line but has the ability to drive several TTL inputs (Remember the $\overline{DMA}$ line can only drive 1 TTL input since it comes from a VIA).

6. Address and Byte Counters

We now must see how the address and byte counters operate. The address counter is a 12-bit counter which generates the addresses for the memory during DMA operations. The byte counter (also 12 bits) counts the number of bytes which the DMA controller has transferred, and sends the last word signal when the preset byte count has been reached. The basic hardware configuration of both counters will be presented, followed by a discussion of their loading and counting operations.

Figures 4-13 and 4-14 show the address and byte counters. Each counter consisits of 3 cascaded 74LS163 4-bit counters. The counters are enabled (allowed to count) only during DMA operations (i.e. SYS DMA is a logic 1). The counting and loading signals are provided by a set of logic gates which will be discused shortly.

We will first examine the address counter. We will assume that it has already been loaded with our desired starting address. When the $\overline{\text{DMA}}$ line goes to logic Ø our counters are enabled and will increment each time a count pulse appears at their clock inputs. The address counter outputs are tri-state buffered by two 74LS365 ICs (IC25A and IC24A). When the SYS $\overline{\text{DMA}}$ line is at logic Ø, these buffers are enabled and the output of the address counter appears on the address bus. Remember, when SYS $\overline{\text{DMA}}$ is at logic Ø the AIM-65's address bus is isolated from the DMA controller.

The address which the counter supplies to the address bus is presented to the memory's address inputs. The contents of the selected memory register can then be transferred to or from the tape deck on the data bus. As the address counter is incremented, sequential memory locations are selected and sequential data words are transferred.

Each time the address counter is incremented, the byte counter is also incremented. When the byte counter reaches the all logic 1s state, the RCO (Ripple Carry Output) of IC 25B (pin 15) goes to the logic 1 state. This signal is inverted by 1/6 of IC 17A. The inverted RCO signal presets the LWD flip-flop (1/2 of IC 2ØB) when the counter indicates that the last byte has been sent. The inverted RCO

TABLE 4-4 - Memory Selection Summary

Pin 6 IC18A	A_{11}	A_{10}	Address Range	ICs Selected
1	X	X	-	none
0	0	0	9000-93FF	4B & 14B
0	0	1	9400-97FF	3B & 13B
0	1	0	9800-9BFF	2B & 12B
0	1	1	9C00-9FFF	1B & 11B
			X = Don't Care	

TABLE 4-5 - Read/Write Logic Summary

LINE	$\overline{DMA}$	RAM R/$\overline{W}$	IWRT	IRSTR	$\overline{WE}$
STATE	1	1	X	X	1
	1	0	X	X	0
	0	X	0	X	1
	0	X	1	1	1
	0	X	1	0	0
	X = Don't Care				

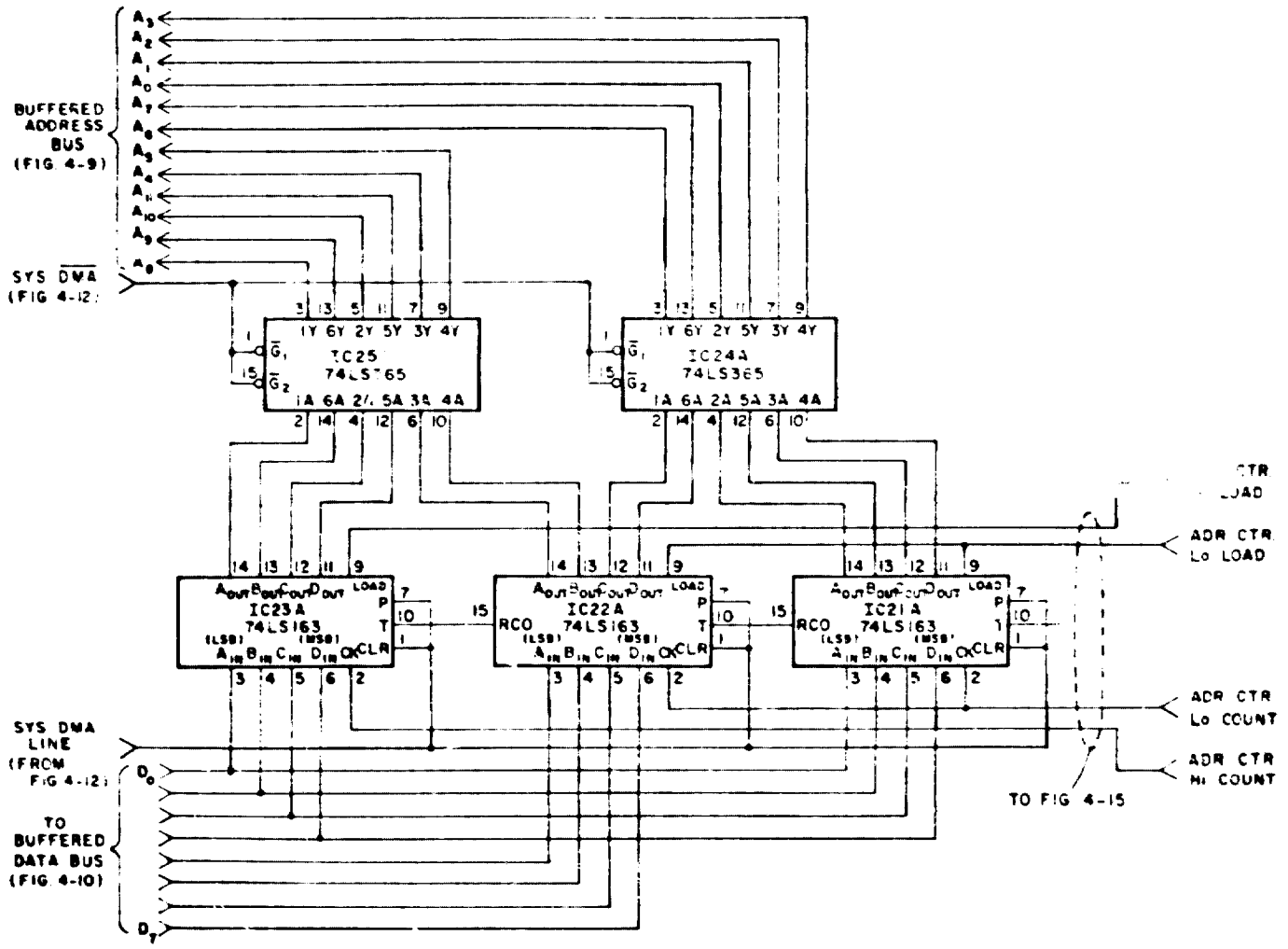


Figure 4-13 Address Counter

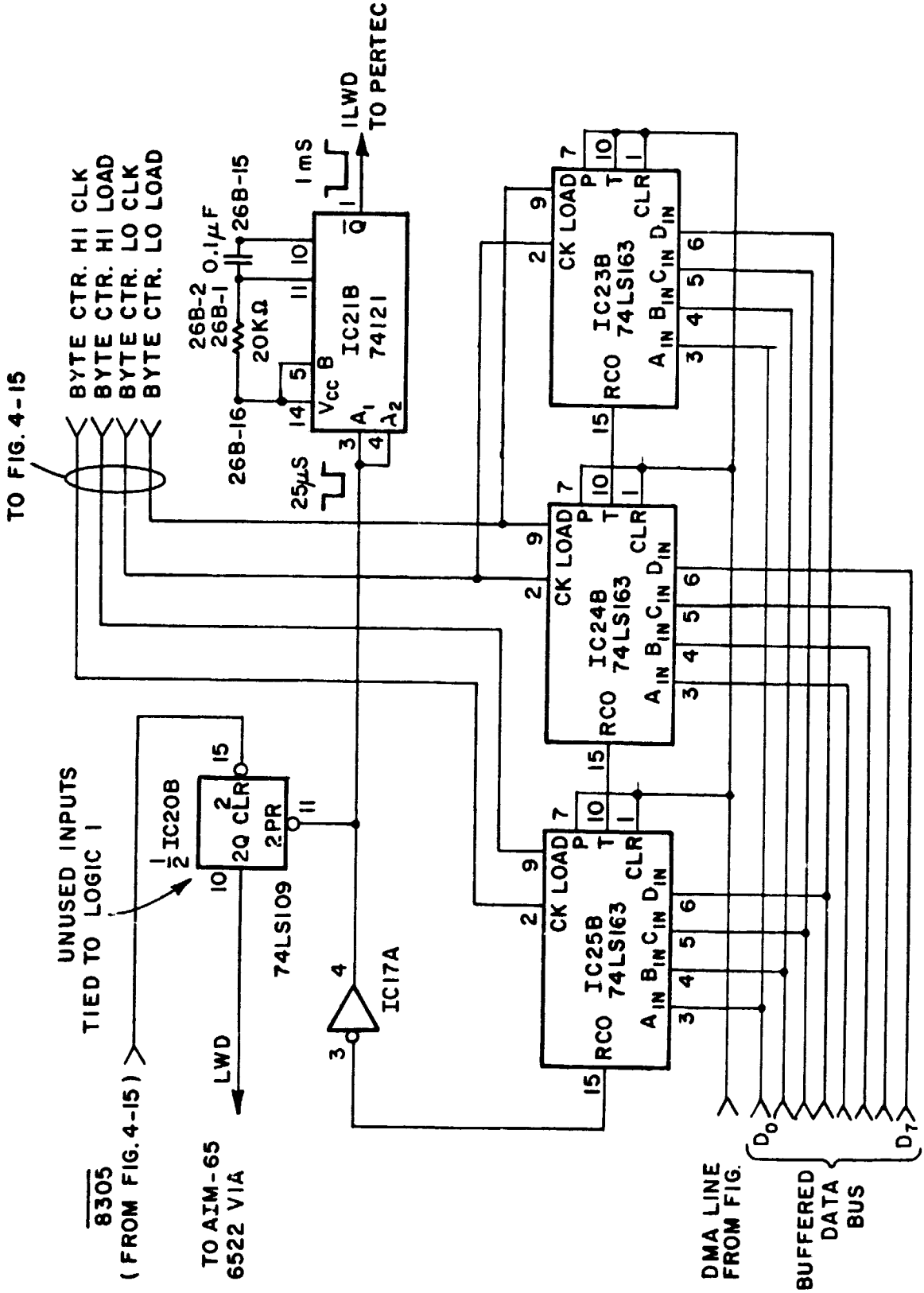


Figure 4-14. Byte Counter

signal also is used to fire a one-shot multivibration (IC21B). This one shot produces an extended pulse of approximately 1 millisecond which is sent to the microformatter as the ILWD pulse. During a tape write operation, this ILWD pulse will tell the tape deck that the last word is being sent. The tape deck will automatically stop upon receipt of this pulse.

The circuit shown in Figure 4-15 generates the clock and load signals for the address and byte counters. A 1 of 8 decoder (IC22B) is used to control the load and clock circuitry. The outputs of this decoder are enabled when the decoder's G1 input (pin 6) is at logic 1 and both $\overline{G2A}$ and $\overline{G2B}$ (pins 5 & 4) are at logic Ø. G1 is connected to the 83ØX line (which goes high when 83ØØ-83OF is addressed). The computer's R/$\overline{W}$ line (which goes to logic Ø during STA operations) is connected to $\overline{G2A}$. The computer's $\overline{\phi 2}$ clock signal is connected to the $\overline{G2B}$ input. This signal becomes a logic Ø during the latter half of each computer cycle. It is during this last half of the cycle that the computer puts a data word on data bus and guarantees that the word is valid (correct and stable). Thus, using the $\overline{\phi 2}$ clock to enable our loading operation insures that the counters will be loaded during the time when the data are on the data bus and stable.

When IC22B is enabled, it examines its A, B and C inputs and pulls the corresponding output to logic Ø. Thus, a STA 83ØØ operation causes a logic Ø pulse to appear on YØ (pin 15), STA 83Ø1 causes a pulse on Y1, etc.

A STA 83Ø4 operation causes a logic Ø pulse on Y4 (pin 11). This output is connected to the clear input of our HER flip-flop (Figure 4-11). Therefore, when the computer excecutes a STA 83Ø4 instruction, the HER flip-flop is cleared. Similarly, STA 83Ø5 clears the LWD flip-flop (Figure 4-14).

We will now examine the loading of the counters. We will make some simplifying assumptions which will make the initial circuit analysis much easier. Let us first ignore IC16A and IC17A. We will also assume that pin 8 of IC6B is at the logic 1 state (which it will be during the loading operations).

The 74LS163 counters (Figures 4-13 and 4-14) require their load inputs to be logic Ø during a load operation. If, while the load line is low, a positive (logic Ø to logic 1) transition occurs on the IC's clock input, the counter will load the number on the data bus.

We will examine the loading of the least significant byte of the address counter as an example. If the computer executes a STA 83ØØ instruction, a logic Ø pulse will appear at the YØ output of IC22B. This causes both the ADR. CTR-LO CLOCK and ADR. CTR-LO LOAD signals to also momentarily go to logic Ø (remember pin 8 of IC6B is at logic 1).

Since the clock signal must go to logic 1 while the load signal is still logic Ø, 2 inverters are placed in the load signal's path to cause a small delay (approximately 50 nanoseconds) for the load pulse. Thus, when YØ of 1C22B returns to logic 1, the clock pulse will return to logic 1 approximately 50 nanoseconds before the load pulse, and the counter will be loaded with the data on the data bus. The other counter bytes are loaded similarly. The only difference is the unique address which selects each counter byte (see Table 4-3).

After the counters have been loaded and the tape deck started, read or write strobes (logic Ø pulses) are sent from the microformatter. IC6B examines these strobes and the status of the IWRT line. If IWRT is set for a write operation (IWRT = logic Ø), the write strobes will be transferred through IC6B and will appear on pin 8 of this IC. They will then appear at each output of IC19A and each counter will be incremented. Similarly, if IWRT is at logic 1, the read strobes will be sent to the clock inputs of the counters.

The strobes from the tape deck must be gated with the IWRT line because both read and write strobes are sent by the microformatter during a write operation. This is because the tape deck reads the freshly written data to determine if a parity error occurred during the write operation. If the strobes are not gated with the IWRT line (by IC6B), read strobes which occur during a write operation can increment the counters causing valid data words to be skipped.

This concludes the discussion of the Direct Memory Access Controller. A complete wire list giving all of the wire wrap connections is presented

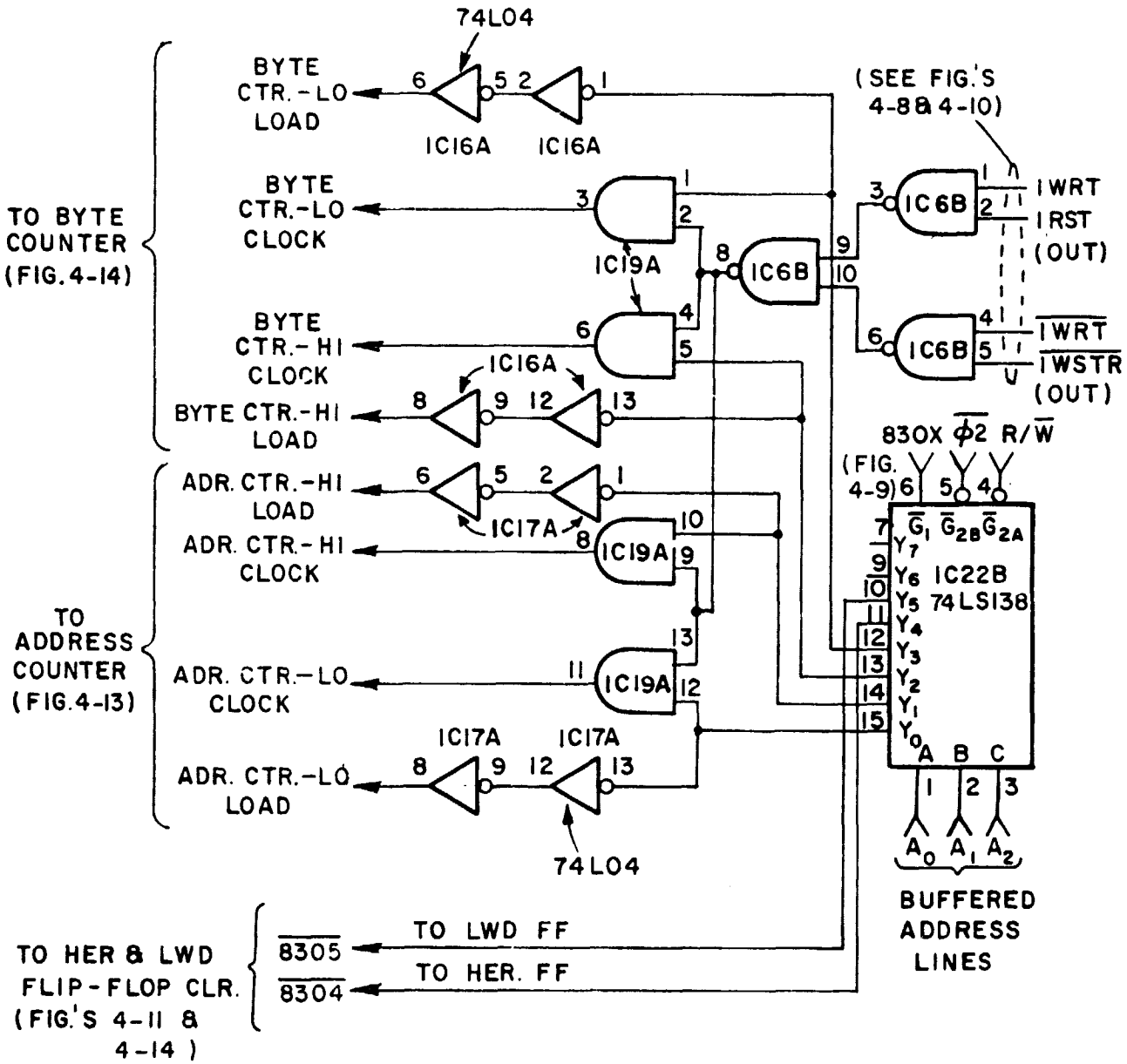


Figure 4-15. Address and Byte Counters - Load and Clock Logic

in Appendix B. A connector key giving the connector and pin number of each line which enters or leaves the DMA controller, is also given in this Appendix.

V. HIGH RESOLUTION RADAR CONTROL

A. Introduction

A high resolution S-band radar which can determine the spatial distribution of water along an Earth-space propagation path is connected to the data acquisition system through a VIA. This radar is controlled by a radar interface which was originally designed to interface the radar with the I/O section of the Hewlett-Packard 2116B minicomputer. This chapter describes the hardware which was used to connect the radar interface with the AIM-65. This is followed by a discussion of the signals which drive the interface and retrieve the radar video. The last section of this chapter gives the software required to generate and receive these signals.

B. Hardware Interconnections and Control Sequence

The radar interface is designed to operate under the control of a Hewlett-Packard 16-bit duplex register. This duplex register is an I/O device which is very similar to a VIA. There are, however, some very important differences. The H-P 16-bit duplex register considers +12V on an I/O pin to be a logic Ø. Ø V on the same pin corresponds to a logic 1. The radar interface's inputs and outputs are designed to be consistent with these logic level definitions. Therefore, to connect a VIA to the interface we must shift the VIA's output signals from TTL levels (+5V logic) to the +12V logic the interface requires. We must also change the logic signals which go from the interface to the VIA from +12V logic to +5 logic to avoid damaging the VIA inputs.

There are 4 signals which run from the VIA to the interface which must be changed from the ØV to +12V range. A circuit was designed and constructed which performs this function. It is shown schematically in Figure 5-1. The 4 signals are buffered by a 7417 hex open collector buffer IC. The open collector outputs are pulled up to +12 Volts by four 1KΩ resistors. This circuit is contained on a small printed circuit

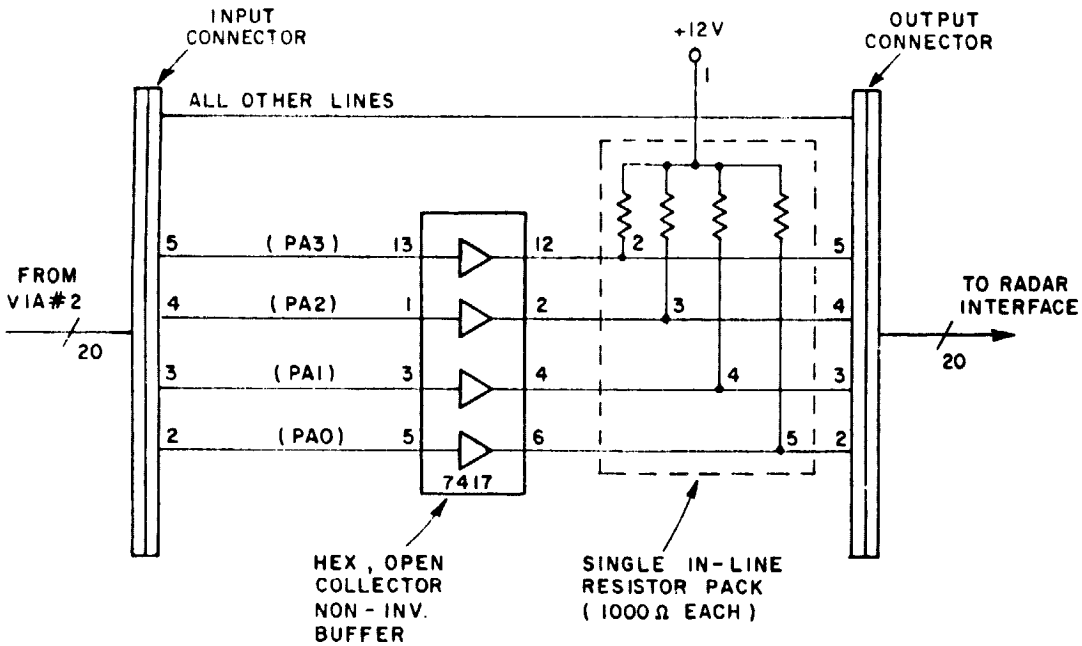


Figure 5-1 Radar Control Word Buffers

board located in the chassis containing the analog conditioning board.

The signals which go from the interface to the VIA are shifted down to the ØV to +5V range from the ØV to 12V range by changing the voltage on their pull-up resistors (in the interface) from +12V to +5V. A switch was installed to allow this voltage to be switched from +12V to +5V. This switch is located near the radar interface. The switch should be set to the AIM position when the interface is to be run by the AIM-65, and set to the H.P. position when the interface when operation with the 2116B is desired.

C. Radar Interface Operation

Before describing the actual timing and control signals necessary for radar operations, a brief overview of the radar interface's operation will be presented. The interface fires the radar when the computer sends the interface a RESET command. A radar pulse is fired and a receiver gate is opened during the trailing edge of the pulse. The received radar video is then sampled 100 times and each sample is converted into a digital word which is stored in a shift register memory. Each of these digital words is proportional to the relative strength of the signal received from one of the 100 samples (range bins). The interface automatically sends 31 additional pulses and samples the video 100 times for each pulse. The results from all of the 32 pulses are digitally integrated (averaged) for each of the 100 range bins. The result of this process is 100 11-bit video words representing the average signal returned for each of the 100 range bins. This entire process takes 320 milliseconds.

After the 32 pulses have been integrated, the interface sends a logic Ø pulse (the HP flag) to the computer. This pulse tells the computer that the interface has stopped and the data can be retrieved. The computer can then take the data from the interface and the entire process is complete.

The radar interface is connected to VIA #2 of the data acquisition system. Figure 5-2 shows the bit assignments for the I/O ports in this VIA. The 4 least significant bits of the A Port are outputs and all of the other port bits in this VIA are inputs. The 4 output bits

comprise the radar control word. These bits are labled DEVICE COMMAND (DC), RESET, S-SELECT, and SHIFT. Bit 4 of the A port is labled HP FLAG. This is the line on which a logic Ø pulse appears when the interface has completed the radar firing routine. The remaining 3 bits of the A port and the entire B port are connected to the 11 data output lines of the interface.

All of the lines going between the computer and the tape deck use negative logic, i.e., a line is considered active when it is at the logic Ø state. The 4 control word signals are explained below:

DEVICE COMMAND (DC) - Each of the other 3 control signals and the HP flag is gated with this line. The line must be at the logic Ø level for any signals or data to be transferred between the computer and the radar interface.

RESET - A logic Ø pulse on this line (while DC is at logic Ø) causes the radar interface to be reset and initiates the radar firing sequence.

S-SELECT - This line, when logic Ø (with DC at logic Ø), causes a video data word to appear on the outputs of the interface. The word can then be read by the computer.

SHIFT - A logic Ø pulse on this line (while DC and S-SELECT are at logic Ø) causes the next sequential data word (of the 100) to be placed on the interface's outputs.

The proper command sequence for driving the radar interface is most easily explained with the aid of a timing diagram. Such a diagram is shown in Figure 5-3. The radar firing sequence is started by pulling the DEVICE COMMAND (DC) and RESET lines to logic Ø. The DC line remains at logic Ø during the entire radar operation. The RESET line returns to logic 1 and the radar begins its firing sequence.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit Ø
B121 = A Port of V.I.A. #2	V2	V1	VØ	HP Flag	DEVICE* COMMAND	RESET*	S - SELECT*	SHIFT*
B12Ø = B Port of V.I.A. #2	V1Ø	V9	V8	V7	V6	V5	V4	V3

CA1
HP FLAG

*V.I.A. Output

Note: V1Ø - VØ are the interface's video data outputs (V1Ø = MSB)

Figure 5-2 - V.I.A.#2 Port Assignments

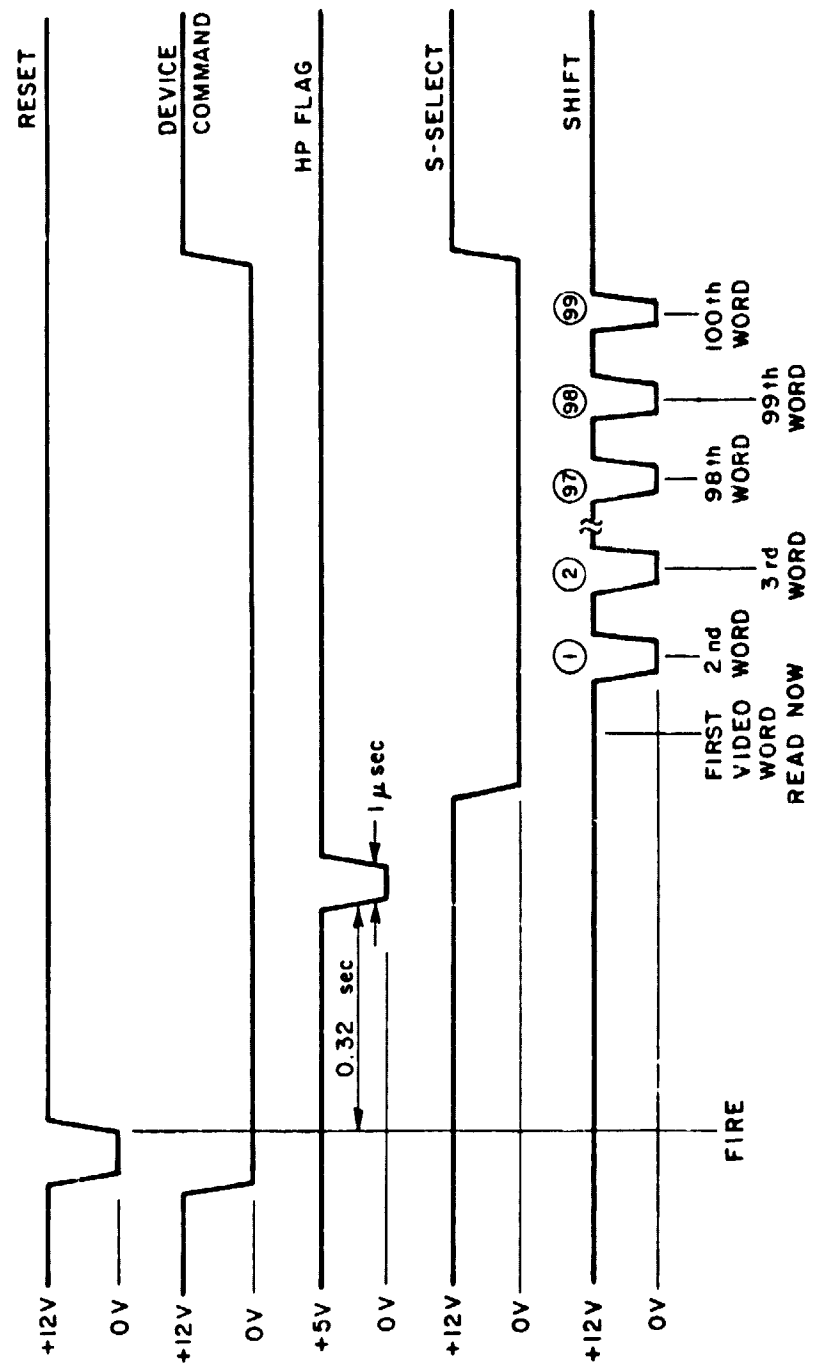


Figure 5-3. Radar Control Timing Diagram

After 320 milliseconds have elapsed, a logic Ø pulse appears on the HP FLAG line. This pulse tells the computer that the 32 radar pulses have been fired and the integrated results can be transferred to the computer. The computer then pulls the S-SELECT line to logic Ø and the first video word appears on the outputs of the interface. The computer then reads this word. The SHIFT line is then pulled to logic Ø and the second word appears on the interface's outputs. After the computer reads this word it toggles the SHIFT line to logic 1 and back to logic Ø. The third word is then read. This toggle and read process is repeated until all 100 video words have been read. The computer then returns all of the control lines, including the DC line, to logic 1 and the process is finished.

D. Software Considerations

This section presents some basic software which will perform the radar control sequence and acquire the video data.

Before presenting the software details we must first deal with a few minor problems. The first problem concerns the HP FLAG signal. This signal is a pulse of approximately 1 microsecond duration. Since most of the AIM-65's instructions require 3 or 4 microseconds to be excecuted, the HP FLAG pulse is too short to be observed by simply examining the port bit connected to the HP FLAG. A simple program loop which examines this bit and waits for it to be a logic Ø might look like:

```
START:  LDA  8121        A←A Port of VIA #2
        AND   #10        A←A • (ØØØ1ØØØØ)
        BNE START        Jump Back to Start if Bit 5 of the A
                         Port = 1
```

Since the HP FLAG is only 1 microsecond long this loop will miss this signal unless it occurs during the program cycle in which the accumulator is loaded with the contents of location 8121. Therefore, we must find a method of positively detecting this short pulse. One

method might be to have the pulse set a flip flop which the computer can read and then clear after the pulse is detected. This would be quite straightforward and the required hardware would be small and easy to build.

There is an easier method of detecting this pulse using a feature of the VIA. The VIA contains several internal flip-flops which are virtually identical to the one which was just described. Use of one of these flip-flops will eliminate the need for any additional hardware, but will require that we learn the necessary software to control the flip-flop.

Space does not permit us to discuss the details of all of the VIA's internal features. Several of the VIA's internal registers will be used to allow us to set and read a flip-flop which is connected to the HP FLAG line. Reference to Chapter 8 of Reference [1] and Chapter 6 of Reference [2] is strongly suggested if you are not thoroughly familiar with the VIA's operation. Particular attention should be given to the discussions of the Interrupt Enable Register (IER), the Interrupt Flag Register (IFR) and the Peripheral Control Register (PCR).

We will detect the pulse on the HP FLAG line by connecting this line to the CA1 input of VIA #2. Since we will never need to interrupt the processor, all bits of the IER should be cleared. This can be done by writing (7F) into the IER (location 812E) during the intialization routine of our program.

By writing a logical Ø into bit Ø of the PCR (location 812C), we can cause the CA1 interrupt flag to set whenever a logic 1 to logic Ø transition occurs on the CA1 input. This flag can be examined by reading bit Ø of the IFR. Since we have cleared all of the IER bits, no interrupt will occur when this CA1 interrupt flag is set. The computer can, however, monitor bit Ø of the IFR. When the HP FLAG is sent, this bit will become a logic 1, and the computer can then retrieve the radar data.

Before each radar operation, the computer must clear bit Ø of the IFR. If a logic 1 is written into any IFR bit, that bit will be cleared. Thus we can easily clear the IFR by writing (FF) into it.

The second problem which we must recognize before we write our radar control program is the fact that the data which the radar interface outputs to the computer is inverted. This is because the 16-bit duplex register from the Hewlett-Packard computer recognizes ground potential as a logic 1 and a positive voltage as a logic Ø. Since AIM-65 uses conventional TTL logic levels, the data outputs from the radar interface should be inverted (complemented) before the computer writes it onto the magnetic tape. This complementing operation can easily be done by the computer without the need for any additional hardware. When the inverted data word is in the computer's accumulator, excecution of an EOR #FF instruction will cause the word to be complemented.

We are now ready to write a simple program which will fire and read the radar. A flow diagram for this program is given in Figure 5-4. A program listing is given below:

```
Ø5ØØ    LDX    #ØØ
Ø5Ø2    LDA    #ØØ
Ø5Ø4    STA    8122      Set Direction Registers
Ø5Ø7    LDA    #ØF
Ø5Ø9    STA    8123
5Ø5C    LDA    #ØØ       PCR←(ØØ)
Ø5ØE    STA    812C
Ø511    LDA    #7F       IER←(7F)
Ø513    STA    812E
Ø516    LDA    #FF       Clear IFR
Ø518    STA    812D
Ø51B    LDA    #Ø3       D.C. & RESET to Logic Ø
Ø51D    STA    8121
Ø52Ø    LDA    #ØF       Reset to logic 1
Ø522    STA    8121
Ø525    LDA    812D      Wait for HP FLAG
Ø528    AND    #Ø1
Ø52A    BEQ    Ø525
Ø52C    LDA    #Ø5       S-Select to Logic Ø
Ø52E    STA    8121
Ø531    LDA    8121      Compliment First Word and Store in
Ø534    EOR    #FF
Ø536    STA    Ø6ØØ      - Jump to Print Subroutine
Ø539    JSR    Ø7ØØ
Ø53C    LDX    #Ø1
Ø53E    LDA    #Ø4       SHIFT to Logic Ø
Ø54Ø    STA    8121
```

```
0543    LDA   8120  }
0546    EOR    #FF  }   Put Next Word in 0600, X
0548    STA  0600,X }
054B    JSR   0700      - Jump to Print Subroutine
054E    LDA    #05      - SHIFT to Logic 1
0550    STA   8121 }
0553    INX        }
0554    TXA        }    Jump Back to 053E if all 100
0555    CMP    #64 }    Words Have Not Been Read
0557    BNE   053E }
0559    LDA    #FF }    All Control Bits to Logic 1
055B    STA   8121 }
055E    BRK
```

Note that this program only recovers the 8 most significant bits of each video word. All 11 bits are brought to the VIA from the radar interface and the program could be easily modified to read the entire 11 bit word if extended precision is required.

It should also be noted that a JSR 0700 instruction is excecuted each time a word is retrieved. If the following subroutine is entered into the AIM-65, a facsimile of the A-scope radar display will be printed by the computer.

```
0700    ROR    A  }
0701    ROR    A  }   Rotate Accumulator 4 bits to
0702    ROR    A  }   the Right
0703    ROR    A  }
0704    AND   #0F      - Mask Off 4 m.s.b.'s
0706    STA    0A      - 0A←A
0708    TXA        }
0709    SBC    #0F }
070B    BPL   0721 }   RTS if x > 16
070D    JSR   EB44     - Clear Print Buffer
0710    LDA    0A  }
0712    BEQ   071E }
0714    LDA    #23 }   Put a Row of # Signs Proportional
0716    JSR   EEFC }   to Original Contents of 0A into
0719    DEC    0A  }   Print Buffer
071B    JMP   0710 }
071E    JSR   E9F0     - Print Contents of Print Buffer
0721    RTS
```

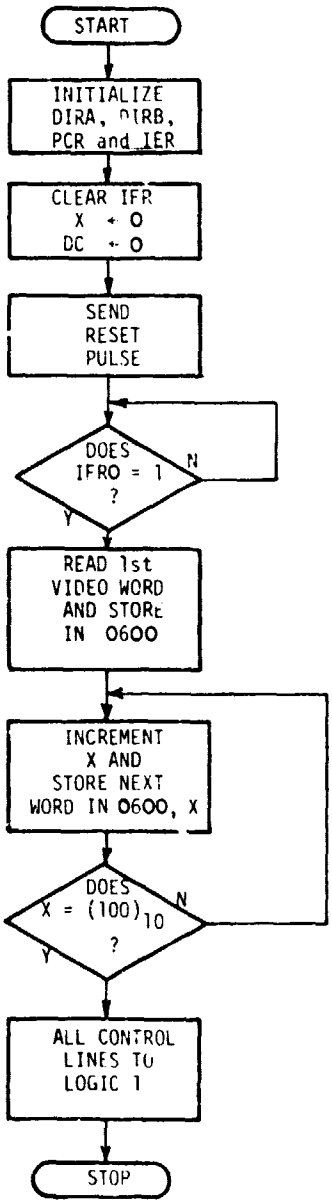


FIGURE 5-4 - RADAR CONTROL PROGRAM

This subroutine prints a row of # symbols (ASCII 23) proportional to the magnitude of each video word. The result is a printed pattern which resembles the radar A-scope display. In order to save time and paper, only the first 16 video words (range bins) are displayed. This number can be varied by varying the object of the SBC operation (step ØØ9). This subroutine might be very useful during repairs or tests of the radar interface or radar. If the printed display is not necessary, it can be eliminated by turning the printer off or by inserting a RTS instruction into location ØØØ.

This concludes the discussion of the control of the high resolution radar. Wire lists for the cables which connect the computer and the radar interface are given in Appendix B.

VI. SENSE SWITCH INPUTS

A set of 16 switches is connected to the 2 input ports of VIA#4. These switches can be set by the system's operator. The computer can then determine the configuration of the switches by simply reading the input ports. The switches might typically be used to stop or idle the system or to tell the computer that the data being returned from each particular experiment are or are not valid.

The computer can periodically examine the switch configuration and operate the system accordingly. Since mechanical switches are prone to contact bounce upon closure, an electronic debouncing circuit was installed so jitter free logic signals will be presented to the VIA inputs.

Figure 6-1 shows the schematic diagram of the sense switch debouncing circuit. The 16 switches are connected to the inputs of three MC14490 hex keybounce eliminators. A debounced version of the inputs to these IC's appears at their outputs

Figure 6-1 also shows the port bit assignments for the switches. For VIA#4 the A port occupies location 8141 and the B port occupies location 8140.

Both the switches and the debouncing circuit are located in the

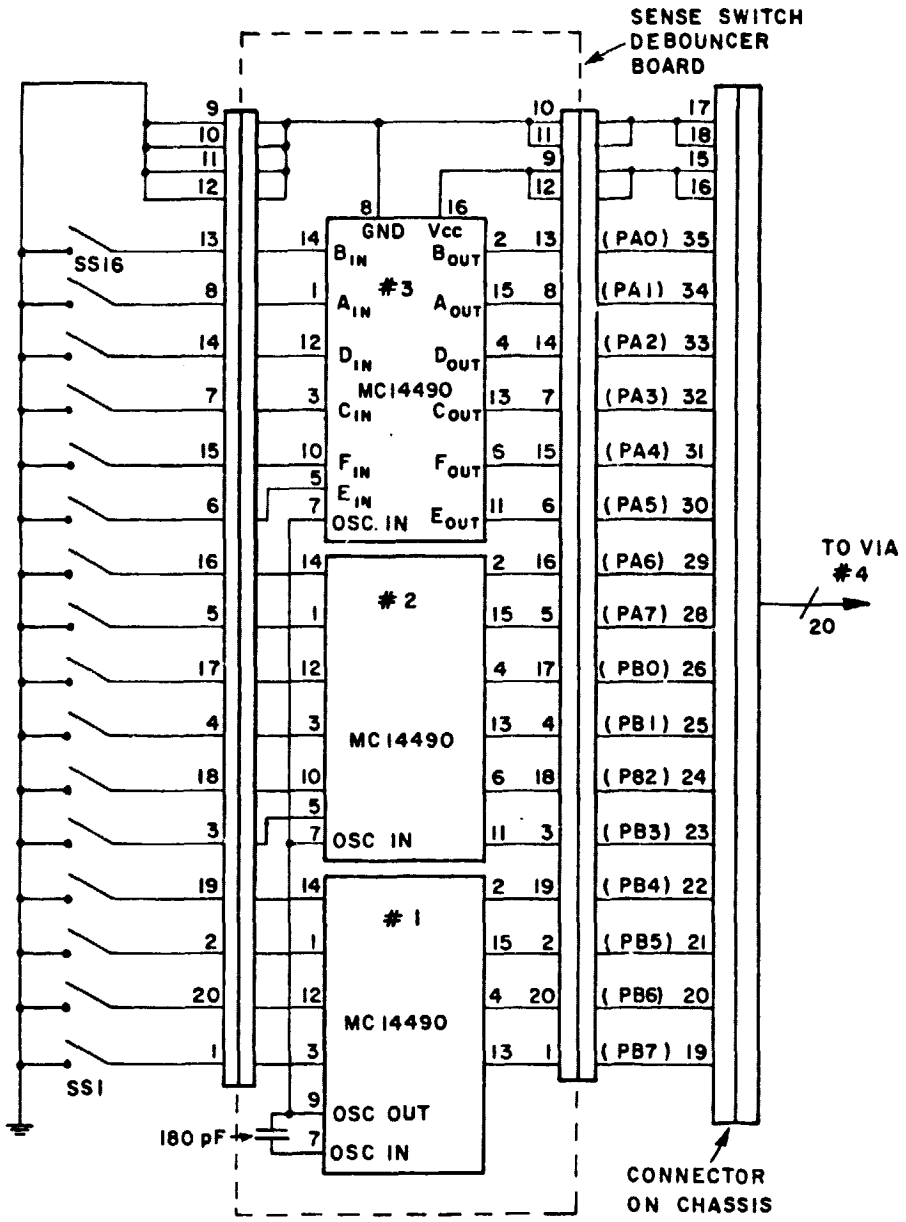


Figure 6-1 Sense Switch Debouncing Circuit

chassis which contains the analog conditioning board. The circuit is constructed using wire-wrap techniques. Connectors are provided so the board can be disconnected from the system for repairs or modifications.

VII. CONCLUSIONS AND FUTURE CONSIDERATIONS

This report has presented a detailed description of each of the data acquisition subsystems. Details of both hardware and software design techniques have been given. We have seen how this system has the ability to acquire digital and analog data, manipulate them and store them on magnetic tape for future analysis. The system also has powerful digital I/O features which allow it to control the system from which it acquires the data.

The AIM-65 is a powerful yet easy to use computer. It's basic operation can be learned in a few hours. This ease of operation can lead to some very useful modifications and extensions of the system's hardware and software. In this chapter we will discuss a few of these possible future extensions and their implications.

An Erasable Programable Read Only Memory (EPROM) Programmer might be a very useful first addition to the system. This programmer can be used to write the system's operating program in a permanent memory. This would eliminate the need to load the program from cassette tape when starting the system. The EPROMS can be erased and reprogrammed if the system's operating program must be changed.

Another useful addition to the system might be a video monitor. This monitor could be used to output real-time displays of the system status or data. A graphic display of the data being acquired could be of great aid to the operator or troubleshooter.

An interface could be designed to connect the interface to a standard telephone line. This would allow the system's operator to monitor or change the system's status from a remote location. This feature might make it much easier to insure that good data are taken during late night thunderstorms when no operator is present.

Audible alarms might be added to the system which could indicate that the system has encountered a problem. This problem might be the end of the magnetic tape, loss of the receiver's lock on the beacon, or the failure of a sensor or hardware component.

There is practically no limit to the software features which can be added to the system. Useful programs, program segments, and subroutines can be designed to perform many operations. Software extentions can be designed which will make the system more automatic by making its own operational decisions. Component calibration and troubleshooting programs can turn the AIM-65 into an extremely powerful piece of test equipment. Anyone who is involved in any way with the system or the data it acquires should master the programming techniques which can modify and improve the system's performance.

We have seen that the microcomputer based data acquisition system is quite versatile and powerful. It can be used in many applications where automatic data acquisition and/or experiment control are required. The straightforward hardware design and ease of software design should make this system a most useful addition to the laboratory's facilities.

REFERENCES

[1] AIM-65 Microcomputer User's Guide. Rockwell International, document No. 29650 N36, 1979.

[2] R6500 Microcomputer System Hardware Manual. Rockwell International, document No. 29650 N31, 1979.

[3] R6500 Microcomputer System Programming Manual. Rockwell International, document No. 29650 N30, 1979.

[4] Model FT8840A Formatted Tape Transport Operating and Service Manual No. 104927 with Microformatter Addendum, Chatsworth Calif., Pertec Computer Corp. 1977.

[5] AIM-65 Microcomputer Monitor Program Listing. Rockwell International, document No. 98650 N36L, 1979.

APPENDIX A - DATA ACQUISITION SYSTEM - MEMORY MAP

0000 + 00FF	Scratch pad memory for operating program
0100 + 01FF	Reserved for AIM-65 stack and monitor subroutines
0200 + 0FFF	User available RAM for operating program
1000 + 7FFF	Not currently used
8000 8001 8002 + 800F	ADC #0 ADC #1 ADC #2 . . . ADC #F
8010 + 80FF	Not currently used
8100 + 810F	V.I.A. #1
8110 + 811F	V.I.A. #2
	. . .
8170 + 817F	V.I.A. #7

8180 + 82FF	Not currently used
8300	ADR. CTR. -LO
8301	ADR. CTR. -Hi
8302	BYTE CTR. -Hi
8303	BYTE CTR. -Lo
8304	LWD - Clear
8305	HER - Clear
8306	Not Used
8307	STATUS INPUT REG.
8308 + 8FFF	Not currently used
9000	DMA Memory (4K RAM on DMA board)
A000 + A00F	AIM-65 V.I.A.
A00F + HFFF	AIM-65 I/O and RAM
B000 + DFFF	Reserved for ROM (or EPROM) sockets on AIM-65 Board
E000 + FFFF	AIM-65 Monitor Program (ROM)

APPENDIX B - WIRE LISTS, PC LAYOUTS, ETC.

DMA Controller - Connector Key

	J1A	J2A	J1B	J2B
1	D_0	CS9	IRØ	IRDY
2	D_1	$\overline{\phi 2}$	IR1	IONL
3	D_2	RAM R/$\overline{W}$	IR2	IFPT
4	D_3	IRSTR (OUT)	IR3	ILDP
5	D_4	LWD	IR4	IFOT
6	D_5	IWRT (OUT)	IR5	IRWD
7	D_6	IWRT (IN)	IR6	IWRT (OUT)
8	D_7	$\overline{DMA}$ (IN)	IR7	IWSTR (IN)
9	A_{11}	IREV (IN)	NC	IRSTR (IN)
10	A0	IWFM (IN)	NC	IHER
11	SYS R/$\overline{W}$	IREW (IN)	NC	ILWD
12	$\overline{CS9}$	NC	NC	IREV (OUT)
13	$\overline{\phi 2}$	NC	NC	NC
14	A10	SYS R/$\overline{W}$	NC	IWFM
15	A9	$\overline{CS8}$	IWØ	IREW
16	A8	IERASE (IN)	IW1	IERASE
17	A7	IOFL (IN)	IW2	IOFL
18	A6	IFEN (IN)	IW3	IFEN
19	A5	IGO (IN)	IW4	IGO
20	A4	IFBY (OUT)	IW5	IFBY
21	A3	IDBY (OUT)	IW6	IDBY
22	A2	ICCG (OUT)	IW7	ICCG
23	A1	IFMK (OUT)	NC	IFMK
24	NC	NC	NC	SIG GND
25	$\overline{CS8}$	NC	NC	NC
26	RAM R/$\overline{W}$	NC	NC	NC

CABLE KEY - J1A TO AIM-65 EXPANSION CONNECTOR

J1A pin #	Expansion Connector #	Label
1	15	D_0
2	14	D_1
3	13	D_2
4	12	D_3
5	11	D_4
6	10	D_5
7	9	D_6
8	8	D_7
9	N	A_{11}
10	A	A_0
11	V	SYS R/$\overline{W}$
12	19	$\overline{CS9}$
13	Y	$\overline{\phi 2}$
14	M	A_{10}
15	L	A_9
16	K	A_8
17	J	A_7
18	H	A_6
19	F	A_5
20	E	A_4
21	D	A_3
22	C	A_2
23	B	A_1
24	NC	
25	18	$\overline{CS8}$
26	Z	RAM R/$\overline{W}$

CABLE KEY - J2A TO AIM-65 APPLICATION CONNECTOR

J2A pin #	Application Connector #	Label
1	NC	$\overline{CS9}$
2	NC	$\overline{\phi 2}$
3	NC	RAM R/$\overline{W}$
4	17	IRSTR
5	12	LWD
6	NC	
7	4	IWRT (IN)
8	14	$\overline{DMA}$
9	3	IREV (IN)
10	2	IWFM (IN)
11	5	IREW (in)
12	NC	
13	NC	
14	NC	SYS R/$\overline{W}$
15	NC	$\overline{CS8}$
16	6	IERASE (IN)
17	7	IOFL (IN)
18	8	IFEN (IN)
19	15	IGO (IN)
20	9	IFBY (OUT)
21	10	IDBY (OUT)
22	11	ICCG (OUT)
23	13	IFMK
24	NC	
25	NC	
26	NC	

CABLE KEY - J1B TO PERTEC INTERFACE

J1B	Pertec Connector #	Label
1	B37	IRØ
2	A37	IR1
3	B39	IR2
4	A39	IR3
5	B40	IR4
6	A40	IR5
7	B42	IR6
8	A42	IR7
9	A35	GND
10	A35	GND
11	NC	
12	NC	
13	NC	
14	B16	IWØ
15	A16	IW1
16	B18	IW2
17	A18	IW3
18	B19	IW4
19	A19	IW5
20	B21	IW6
21	A21	IW7
22	A35	GND
23	A35	GND
24	NC	
25	NC	
26	NC	

CABLE KEY - J2B TO PERTEC INTERFACE

J2B #	Pertec Connector #	Label
1	B27	IRDY
2	A27	IONL
3	A28	IFPT
4	B30	ILDP
5	A30	IEOT
6	B28	IRWD
7	A4	IWRT
8	A34	IWSTR
9	B36	IRSTR
10	A24	IHER
11	B13	ILWD
12	B4	IREV
13	NC	-
14	B6	IWFM
15	B12	IREW
16	B7	IERASE
17	A12	IOFL
18	A13	IFEN
19	A3	IGO
20	B22	IFBY
21	A22	IDBY
22	B24	ICCG
23	A25	IFMK
24	B35	GND
25	NC	-
26	NC	-

DMA CONTROLLER - IC LIST

A side		B side	
IC #	TYPE	IC #	TYPE
1	74LS240	1	2114L
2	74LS245	2	2114L
3	74LS245	3	2114L
4	74LS365	4	2114L
5	74LS365	5	74LS240
6		6	7400
7		7	74365
8		8	74365
9	74LS139	9	74365
10	74LS00	10	
11	74LS51	11	2114L
12	74LS85	12	2114L
13	74LS85	13	2114L
14	74LS85	14	2114L
15	74LS85	15	74LS04
16	74L04	16	
17	74L04	17	
18	74LS08	18	
19	74LS08	19	
20	Resistor Pack	20	74109
21	74LS169	21	74121
22	74LS169	22	74LS136
23	74LS169	23	74LS169
24	74LS365	24	74LS169
25	74LS365	25	74LS169
26		26	Discrete Components
27		27	
28		28	
29		29	

CABLE KEY - VIA #0 TO DATATRON CLOCK CABLE

VIA #0 Connector #	Clock Cable #	Label
1	13	GND
2	1	USC 1
3	2	2
4	3	4
5	4	8
6	5	TSC 1
7	6	2
8	7	4
9	13	GND
10	NC	-
11	NC	-
12	8	UMC 1
13	9	2
14	10	4
15	11	8
16	12	TMC 1
17	14	2
18	15	4
19	13	GND
20	13	GND

CABLE KEY - VIA #1 TO DATATRON CLOCK CABLE

VIA # Connector #	Clock Cable #	Label
1	13	GND
2	26	UDC 1
3	27	2
4	28	4
5	29	8
6	30	TDC 1
7	31	2
8	32	4
9	33	8
10	NC	-
11	NC	-
12	16	UHC 1
13	17	2
14	18	4
15	19	8
16	20	THC 1
17	21	2
18	34	HDC 1
19	35	2
20	13	GND

C2

CABLE KEY - VIA #2* TO RADAR INTERFACE (R.I.)

VIA #2 pin #	R.I. #	Label
1	36	GND
2	23	SHIFT
3	19	S-SELECT
4	24	RESET
5	35	D.C.
6	17	HP FLAG
7	1	Bit 0
8	2	1
9	3	2
10**	17	HP FLAG
11	NC	-
12	4	Bit 3
13	5	4
14	6	5
15	7	6
16	8	7
17	9	8
18	10	9
19	11	10
20	36	GND

* This connector is located on the circuit board shown in Figure 5-1.

** This pin is connected to VIA #2's CA1 input.

Note: Unused control word inputs (pins 20, 21, 22, of the R.I. connector) must be tied to +12V.

CABLE CONNECTOR SENSE SWITCH CONNECTOR (ON CHASIS) TO VIA #4

Sense Switch Connector #	VIA #4 pin #	Label
15	-	+5V
16	-	+5V
17	1	GND
18	20	GND
19	19	S.S. D-1
20	18	S.S. D-2
21	17	S.S. D-3
22	16	S.S. D-4
23	15	S.S. D-5
24	14	S.S. D-6
25	13	S.S. D-7
26	12	S.S. D-8
27	-	NC
28	9	S.S. D-9
29	8	S.S. D-10
30	7	S.S. D-11
31	6	S.S. D-12
32	5	S.S. D-13
33	4	S.S. D-14
34	3	S.S. D-15
35	2	S.S. D-16

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC1A				74LS244	20
1	1	10A-6			
1	1	5B-1	5B-1		
2	2	1A-3			
2	2	3A-9	3A added pin-Lo		
4	4	1A-5	1A-5		
4	4	3A-8	3A-8		
6	6	1A-7	1A-7		
6	6	3A-7	3A-7		
8	8	1A-9	1A-9		
8	8	3A-6	3A-6		
10	5A-1			GND	
11	5A-16		J1B-5	1R4	
12	added pin-Hi		J1B-18	1W4	
13	9		J1B-6	1R5	
14	10		J1B-19	1W5	
15	11		J1B-7	1R6	
16	12		J1B-20	1W6	
17	13		J1B-8	1R7	
18	14		J1B-21	1W7	
19	15	10A-8	10A-8		
19	15	5B-19	5B-15		
20	16			V_{CC}	
IC2A				74LS245	20
1	1	GND		GND	
2	2	5B-9	5B added pin-Lo		
3	3	5B-7	5B-7		
4	4	5B-5	5B-5		
5	5	5B-3	5B-3		
6	6	1A-9	1A added pin-Lo		
7	7	1A-7	1A-7		
8	8	1A-5	1A-5		

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
9	added $_{LO}$ pin-	1A-3	1A-3		
10	7A-1			GND	
11	7A-16		J2B-1	IRDY	
12	added $_{HI}$ pin-		J2B-2	IONL	
13	9		J2B-3	IFPT	
14	10		J2B-4	ILDP	
15	11		J2B-5	IEOT	
16	12		J2B-6	IRWD	
17	13			GND	
18	14	20B-6	20B-6		
19	15	15B-12	15B-14		
20	16			5V	
IC3A				74LS245	20
1			J2A-7	R/$\overline{W}$	
10	8A-1			GND	
11	8A-16			D_0	
12	added $_{HI}$ pin-			D_1	
13	9			D_2	
14	10			D_3	
15	11			D_4	
16	12			D_5	
17	13			D_6	
18	14			D_7	
19	15	15B-8	15B-8		
20	16				
IC4A				74LS365	16
1	1	4A-15	4A-15		
1	1	5A-1	5A-1		
1	1	15B-3	15B-3		
2	2		J1A-19	A_5	
4	4		J1A-21	A_3	

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
6	6		J1A-23	A_1	
10	10		J1A-24	A_0	
12	12		J1A-22	A_2	
14	14		J1A-20	A_4	
IC5A				74LS365	16
1	1	5A-15	5A-15		
2	2		J1A-13	A_{11}	
4	4		J1A-15	A_9	
6	6		J1A-17	A_7	
10	10		J1A-18	A_6	
12	12		J1A-16	A_8	
14	14		J1A-14	A_{10}	
IC9				74LS139	16
9	9	11B-8	11B-8		
10	10	12B-8	12B-8		
11	11	13B-8	13B-8		
12	12	14B-8	14B-8		
13	13	5A-3	5A-3		
14	14	5A-13	5A-13		
15	15	18A-6	18A-6		
IC10A				74LS00	14
1	1		J2A-2	IWRT	
2	2	15B-6	15B-6		
3	3	11A-3	11A-3		
4	4	16A-4	16A-4		
5	5	10A-9	10A-11		
5	5	15B-2	15B-2		
5	5	11A-1	11A-1		
7	7		10A-8	GND	

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
10	12	16A-3	16A-3		
10	12	10A-1	10A-1		
IC11A				74LS51	14
1	1	11A-13	11A-15		
1	1	11A-12	11A-14		
2	2	15B-2	15B-2		
2	2	15B-3	15B-3		
4	4	15B-4	15B-4		
5			J2A-3	RAM R/$\overline{W}$	
4	4	18A-5	18A-5		
6	6	15B-11	15B-13		
7			11A-8	GND	
8	10	15B-9	15B-11		
9	11	17A-12	17A-14		
10	12	11A-11	11A-13		
10	12	18A-4	18A-4		
10	12		J2A-1	$\overline{CS9}$	
IC12A				74LS85	16
1	1	20A-2	20A-2		
1	1	12A-14	12A-14		
3	3	13A-6	13A-6		
3	3	22B-6	22B-6		
6	6	15B-13	15B-15		
9	9	12A-11	12A-11		
9	9	12A-10	12A-10		
11	11	12A-14	12A-14		
12	12	22B-3	22B-3		
12	12	4A-11	4A-11		
13	13	22B-2	22B-2		
13	13	4A-7	4A-7		
15	15	4A-9	4A-9		

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
15		22B-1			
IC13A				74LS85	16
1		13A-14			
3		14A-6			
6		17A-11	17A-13		
9		13A-11			
10		13A-12			
10		13A-8		GND	
11		13A-14			
11		13A-10			
13			J2A-9	$\overline{CS8}$	
15		5A-3			
IC14A					
1		14A-9			
3		15A-6			
9		14A-8		GND	
10		SA-13			
11		20A-3			
11		14A-14			
12		5A-5			
13		5A-11			
15		5A-7			
15A				74LS85	16
1		15A-14			
3		14A-14			
9		15A-11			
9		15A-14			
10		5A-9			
11		15A-8		GND	
12		4A-3			
13		4A-13			

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
15		4A-5			
IC16A				74L04	14
2		16A-5			
1		19A-1			
1		22B-12			
6		23B-9			
9	11	16A-12	16A-14		
8	10	25B-9			
13	15	19A-5			
13	15	22B-13			
IC17A				74L04	14
1		19A-10	19A-12		
1		22B-14			
2		17A-5			
6		23A-9			
8	10	21A-9			
9	11	17A-12			
13	15	19A-12	19A-14		
13	15	22B-15			
IC18A				74LS08	14
7			18A-8	GND	

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC19A				74LS08	14
2		19A-4			
3		24B-2			
6		25B-2			
7			19A-8	GND	
8	10	23A-2			
9	11	19A-13	19A-15		
11	13	21A-2			
IC20A				R-Pack	16
1		20A-16		+5V	
4		21A-1			
5		23A-7			
IC21A				74LS163	16
1		22A-1			
1		23A-1			
2		22A-2			
3		3A-9	3A added L8		
3		23A-3			
3		23B-3			
4		3A-8			
4		23A-4			
4		23B-4			
5		3A-7			
5		23A-5			
5		23B-5			
6		3A-6			
6		23A-6			
6		23B-6			
7		21A-10			
9		22A-9			
11		24A-10			
12		24A-6			
13		24A-12			

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
14		24A-4			
15		22A-10			
IC22A				74LS163	16
3		3A-5			
3		24B-3			
4		3A-4			
4		24B-4			
5		3A-3			
5		24B-5			
6		3A-2			
6		24B-6			
11		24A-14			
12		24A-2			
13		25A-10			
14		25A-6			
15		23A-10			
IC23A				74LS163	16
7		22A-7			
11		25A-12			
12		25A-4			
13		25A-14			
14		25A-2			

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC24A				74LS365	1
1		24A-15			
1		25A-15			
3		5A-9			
5		4A-9			
7		4A-11			
9		4A-5			
11		4A-7			
13		5A-7			
IC25A				74LS365	16
1		25A-15			
3		5A-11			
5		5A-13			
7		4A-13			
9		4A-3			
11		5A-3			
13		5A-5			
15		5B-4			

pins 1,2,3,4,5,6, 7,8,15,16, & 17 of IC's 1B, 2B, 3B, 4B, 11B, 12B, 13B, & 14B are Daisy Chained

Some of these connections are made on IC3B to minimize crowding at IC4B

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC1B				2114L	18
8		11B-8			
IC2B				2114L	18
8		12B-8			
IC3B				2114L	18
8		13B-8			
IC4B				2114L	18
8		14B-8			
1		5A-9			
2		4A-3			
3		4A-13			
4		4A-5			
5		4A-9			
6		4A-7			
7		4A-11			
9	Added pin-Lc			GND	
10	Added pin-Hf	15B-10	15B-12		
11		5B-9			
12		5B-7			
13		5B-5			
14		5B-3			
15		5A-5			
16		5A-11			
17		5A-7			

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC5B				74LS244	20
2		5B-3			
2		3A-5			
4		5B-5			
4		3A-4			
6		5B-7			
6		3A-3			
8		5B-9			
8		3A-2			
10	10B-1		10B-8	GND	
11	10B-16		J1B-1	IRO	
12	Added pin-Hi		J1B-14	IWO	
13	9		J1B-2	IRI	
14	10		J1B-15	IWI	
15	11		J1B-3	IR2	
16	12		J1B-16	Iw2	
17	13		J1B-4	IR3	
18	14		J1B-17	IW3	
IC14B				2114L	18
11		1A-9	1A added pin-Lo		
12		1A-7			
13	1A-5				
14	1A-3				
IC15B				74LS04	14
1			J2A-8	DMA	
5			J2A-4	IRSTR	
7			15B-8	GND	

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC6B				7400	4
1	1	9B-3	9B-3	IWRT	
4	4	16A-4	16A-4	$\overline{\text{IWRT}}$	
2	2	7B-3	7B-3	IRSTR (out)	
5	5	7B-5	7B-5	IWSTR (out)	
3	3	9	11		
6	6	10	12		
7	7		8	GND	
8		19A-9	19A-11		

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
IC20B				74LS109	16
1		22B-11			
5			J2B-10	IHER	
10			J2A-5	LWD	
11			J2B-11	ILWD	
15		22B-10			
IC22B				74LS138	
5			J2A-6	$\overline{\phi 2}$	
4			J2A-7	R/$\overline{W}$	
IC23B				74LS163	16
1		23B-10			
1		23B-7			
1		21A-7			
3		25B-3			
4		25B-4			
5		25B-5			
6		25B-6			
15		24B-10			
2		24B-2			
9		24B-9			
1		15B-2			
IC24B				74LS163	16
1		24B-7			
1		22A-7			
15		25B-10			
IC25B				74LS163	16
1		25B-7			
1		24B-1			
15		17A-3			

FROM IC PIN NO.	FROM (AUGAT NO.)	TO IC PIN NO.	TO (AUGAT NO.)	LABEL	✓
	J2A-7	9B-2	9B-2	IWRT (in)	
	J2A-9	9B-4	9B-4	IREV (in)	
	J2A-10	9B-6	9B-6	IWFM (in)	
	J2A-11	9B-10	9B-10	IREW (in)	
	J2A-16	9B-12	9B-12	IERASE (in)	
	J2A-17	9B-14	9B-14	IOFL (in)	
9B-3	9B-3		J2B-7	IWRT (out)	
9B-5	9B-5		J2B-12	IREV (out)	
9B-7	9B-7		J2B-14	IWFM (out)	
9B-9	9B-9		J2B-15	IREW (out)	
9B-11	9B-11		J2B-16	IERASE (out)	
9B-13	9B-13		J2B-17	IOFL (out)	
	J2A-18	8B-2	8B-2	IFEN (in)	
	J2A-19	8B-4	8B-4	IGO (in)	
		8B-6	8B-6	IFBY (in)	
		8B-10	8B-10	IDBY (in)	
		8B-12	8B-12	(CCG (in)	
		8B-14	8B-14	IFML (in)	
8B-3	8B-3		J2B-18	IFEN (out)	
8B-5	8B-5		J2B-19	IGO (out)	
8B-7	8B-7		J2A-20	IFBY (out)	
8B-9	8B-9		J2A-21	IDBY (out)	
8B-11	8B-11		J2A-22	ICCG (out)	
8B-13	8B-13		J2A-23	IFMK (out)	
	J2B-9	7B-2	7B-2	IRSTR (in)	
	J2B-8	7B-4	7B-4	IWSRT (in)	

WIRE LIST DMA BOARD

JUMPERS J2A-J1A

	FROM	TO	LABEL
	J2A1	J1A12	$\overline{CS9}$
	J2A2	J2A13	$\overline{\phi 2}$
	J2A14	J1A11	SYS R/$\overline{W}$
	J2A15	J1A25	$\overline{CS8}$
	J2A3	J1A26	

	FROM	TO	LABEL

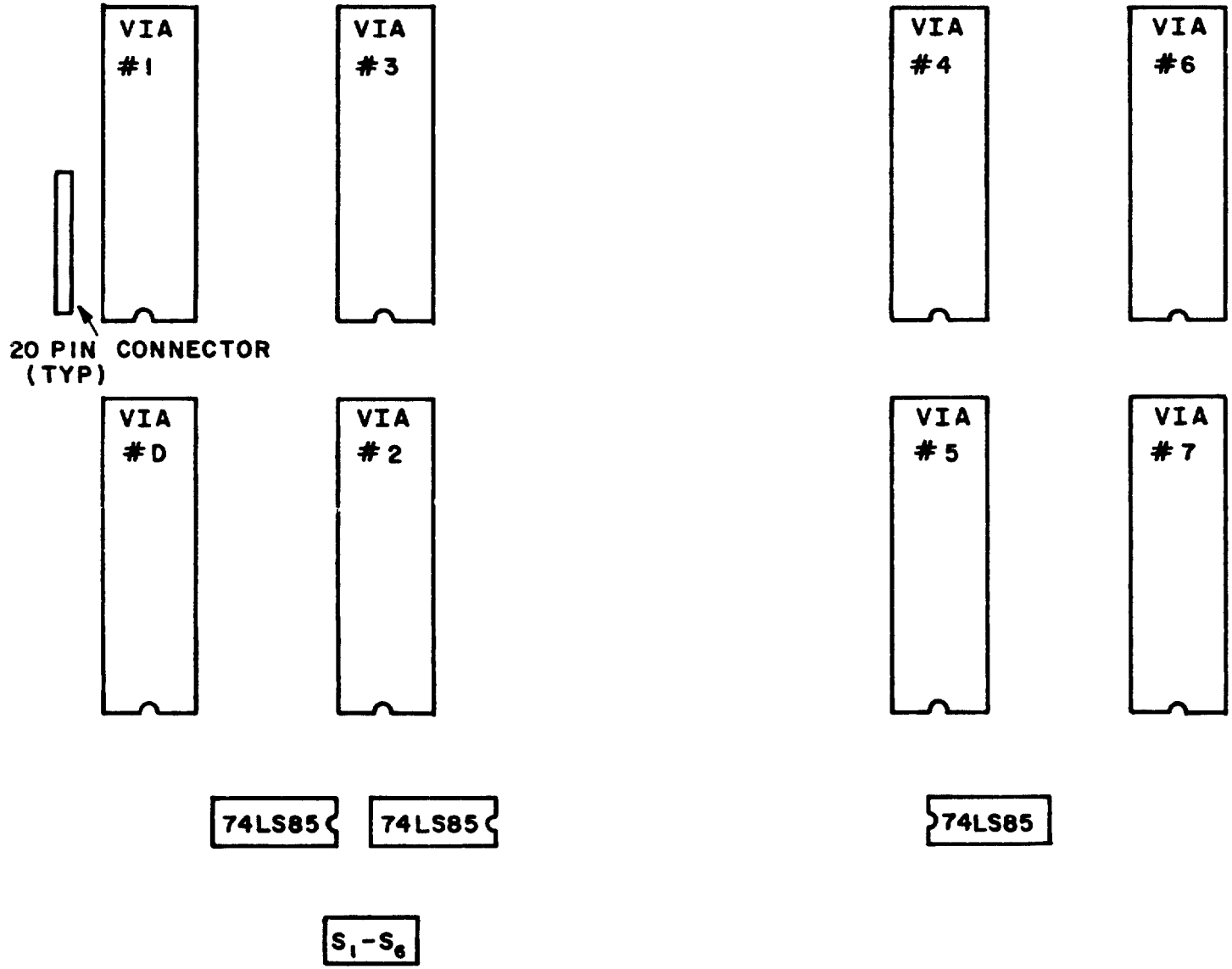


Figure B-1. Digital I/O Board Component Layout - Top View (Full Scale)

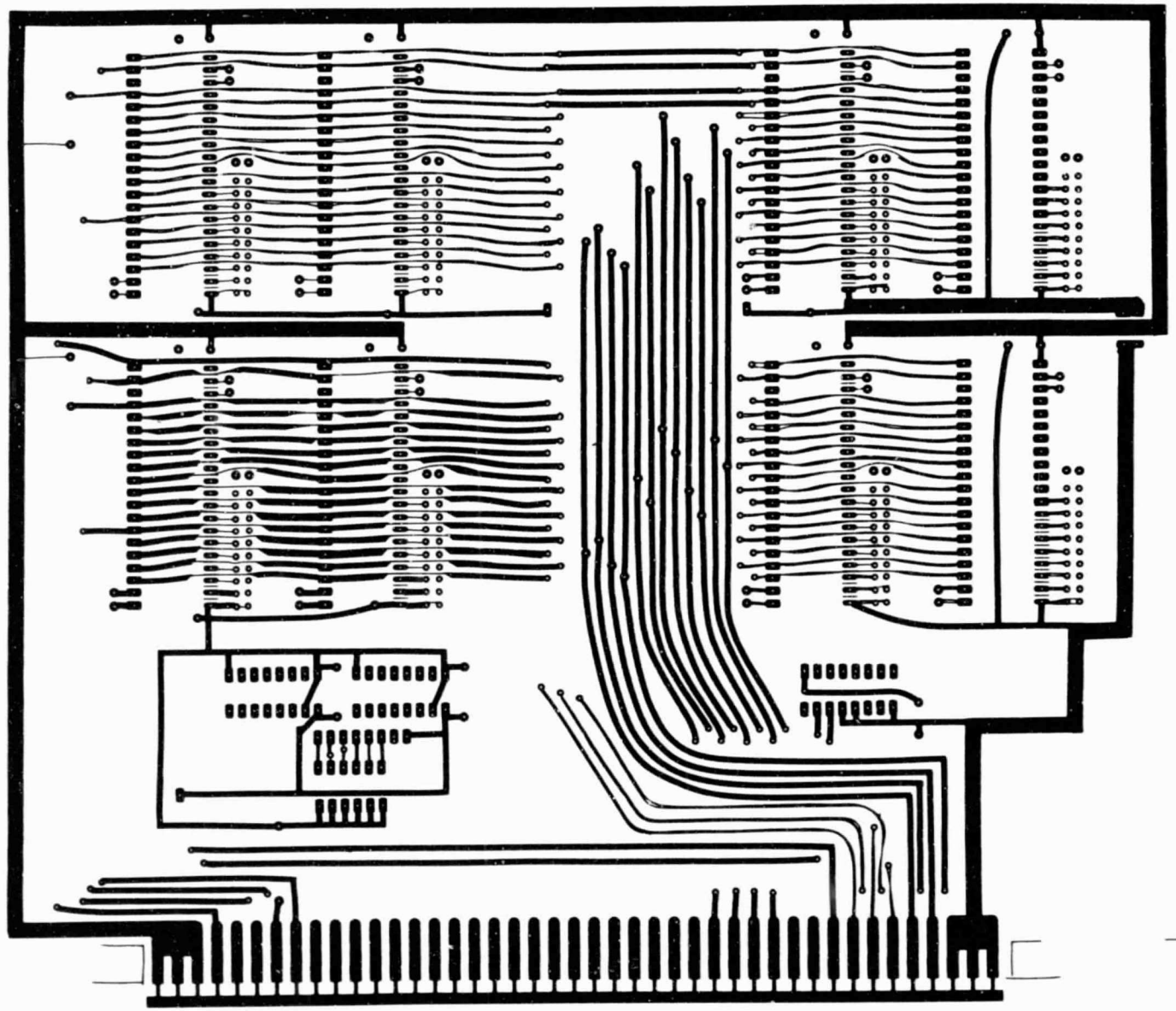


Figure B-2 Digital I/O Board PC Pattern - Top Side (.80 Scale)

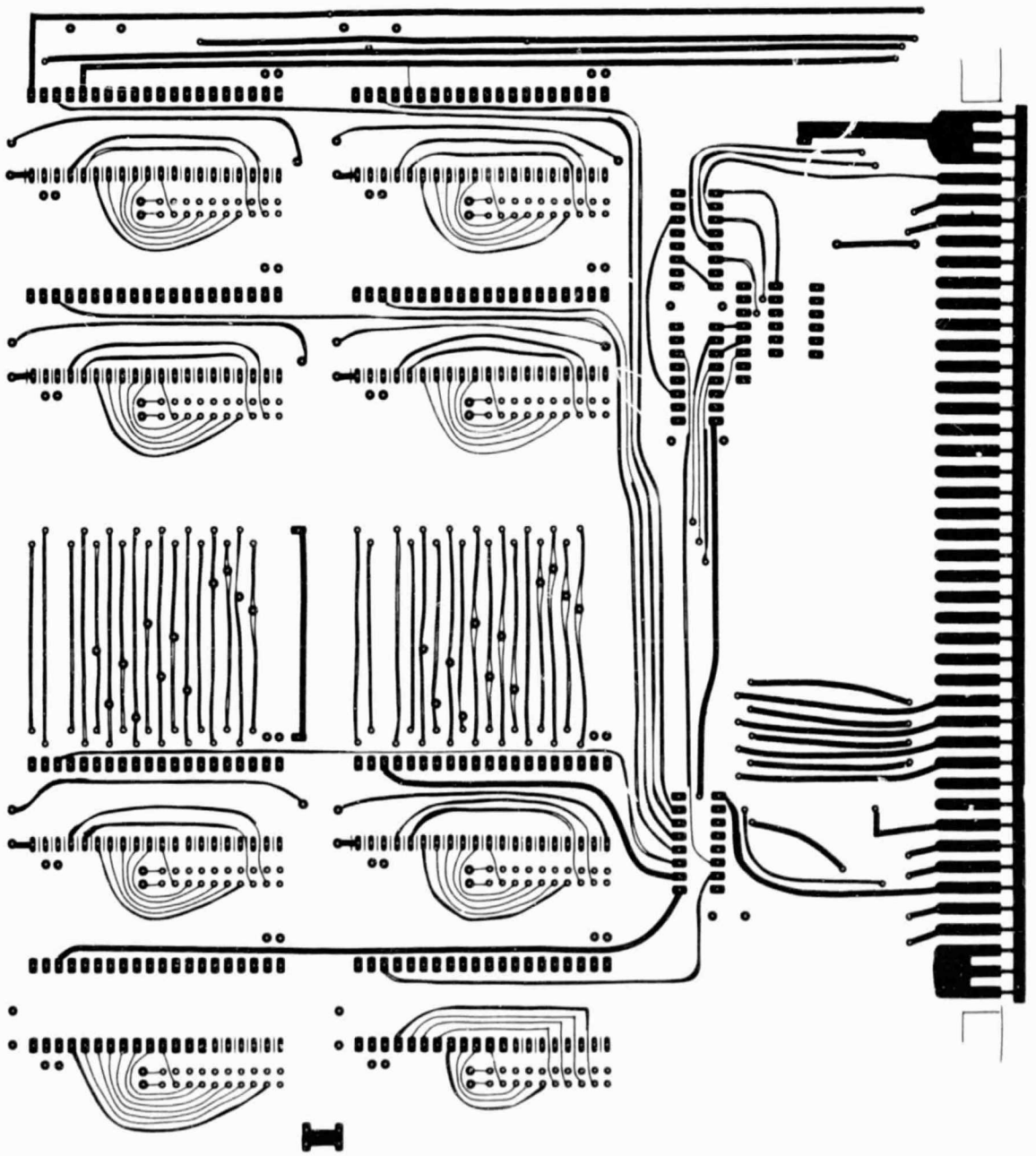


Figure B-3 Digital I/O Board - Bottom Side (.80 Scale)

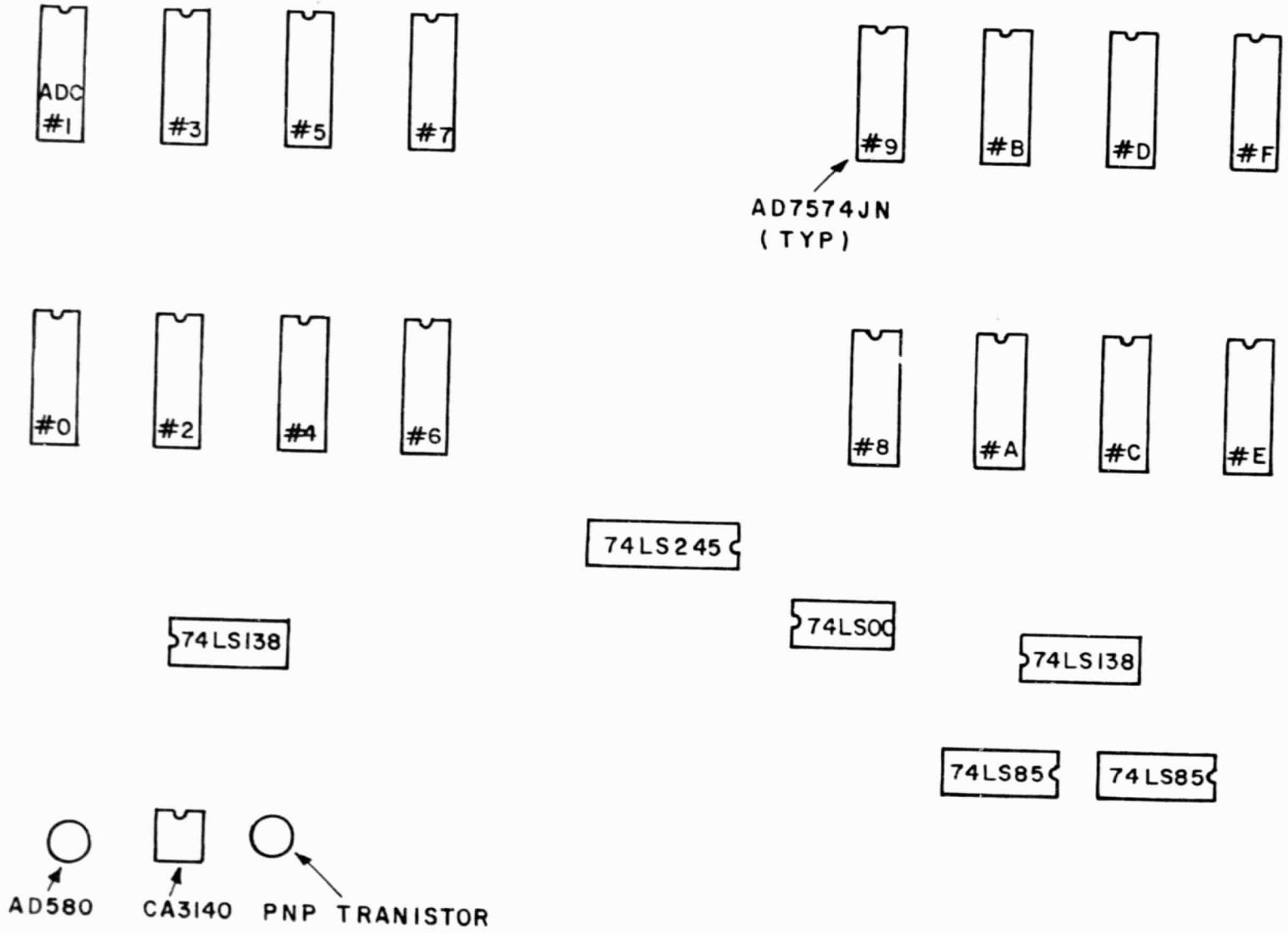


Figure B-4 A/D Board Component Layout - Top View (Full Scale)

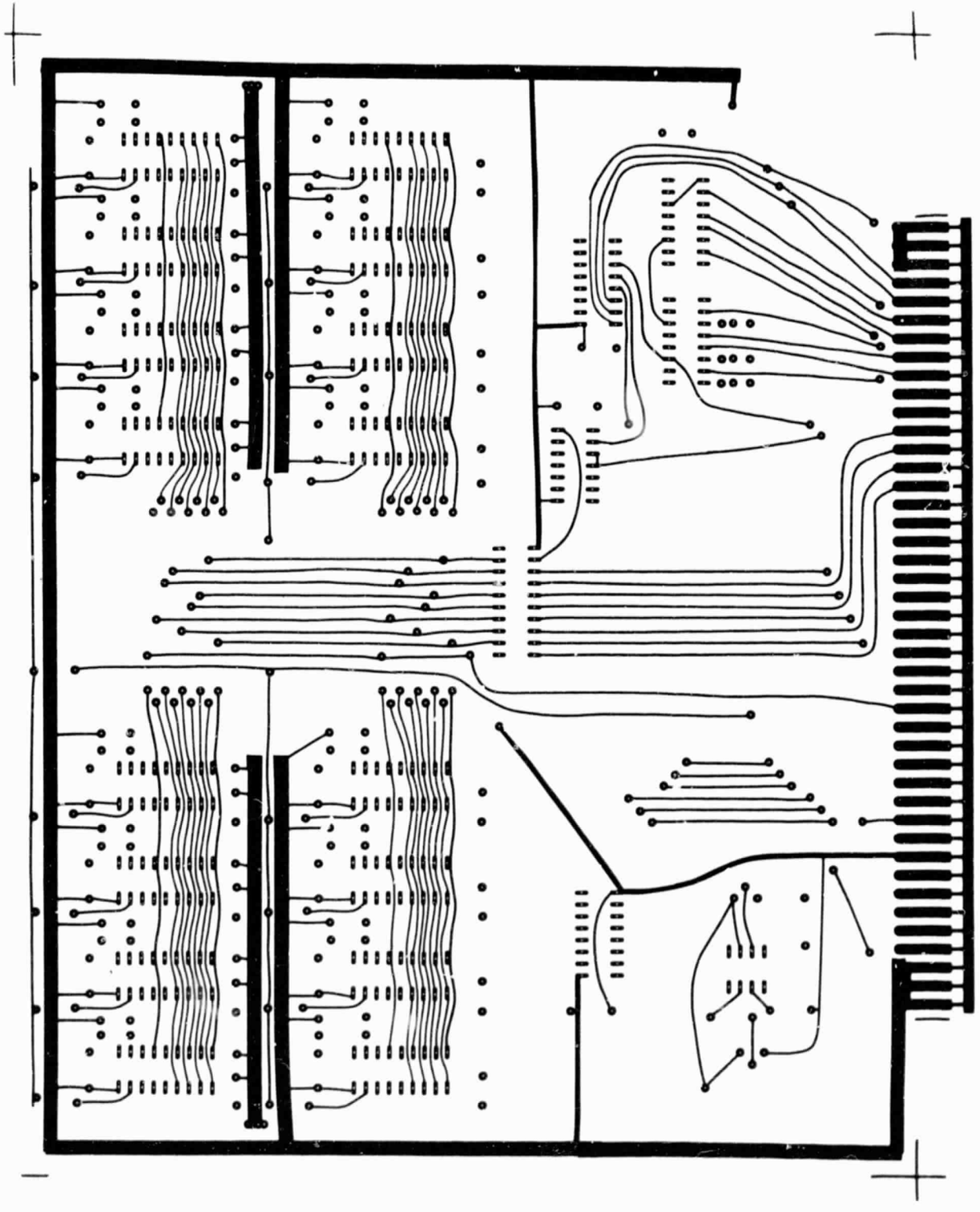


Figure B-5 A/D Board - Top Side (.80 Scale)

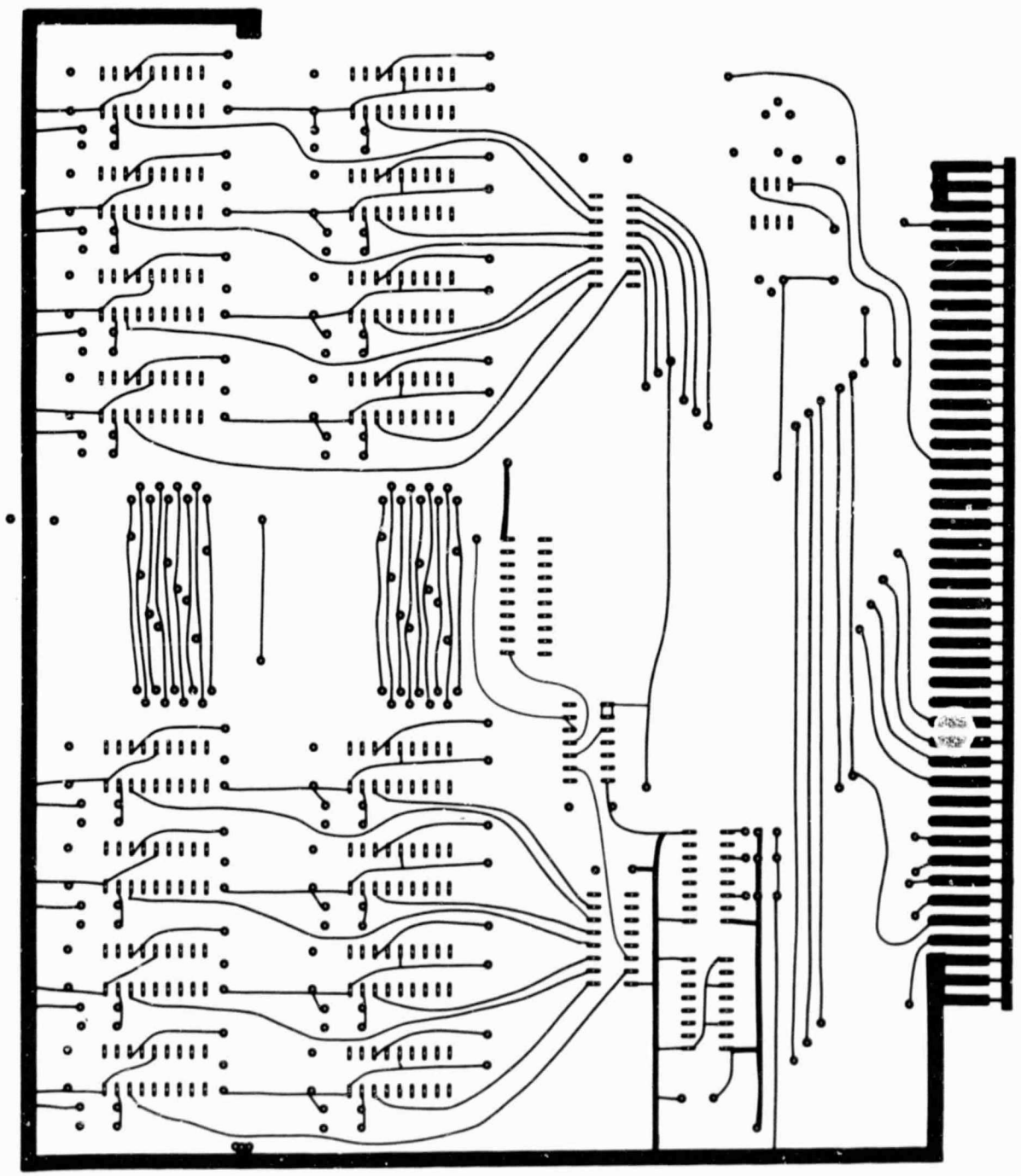


Figure B-6 A/D Board - Bottom Side (.80 Scale)

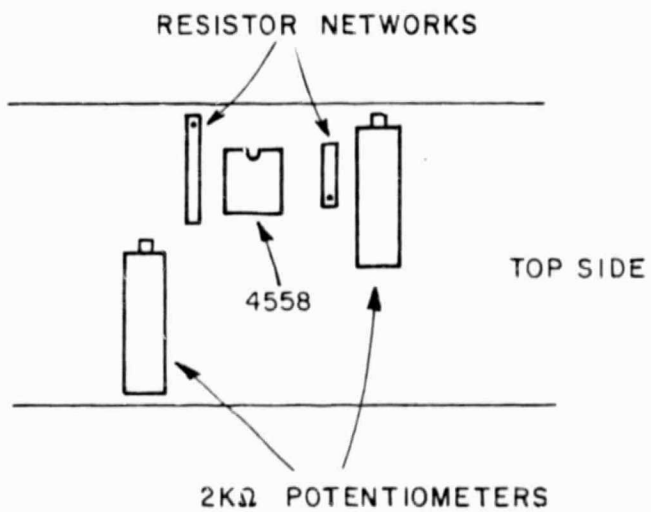
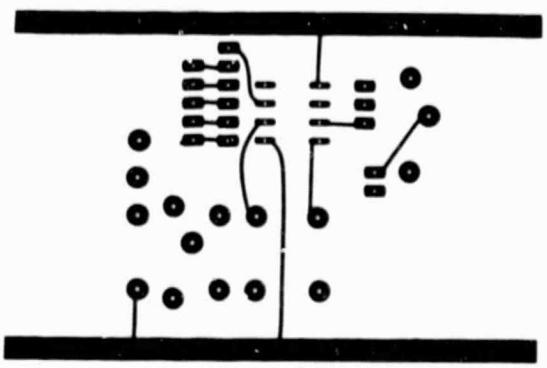
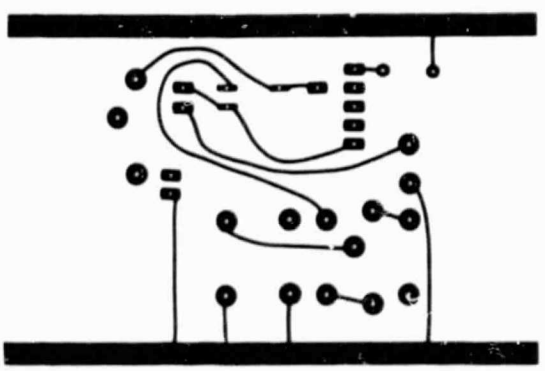


Figure B-7 Analog Conditioning Circuit Component Layout



a.) Top View



b.) Bottom View

Figure B-8 Analog Conditioning Circuit PC Pattern (full scale)

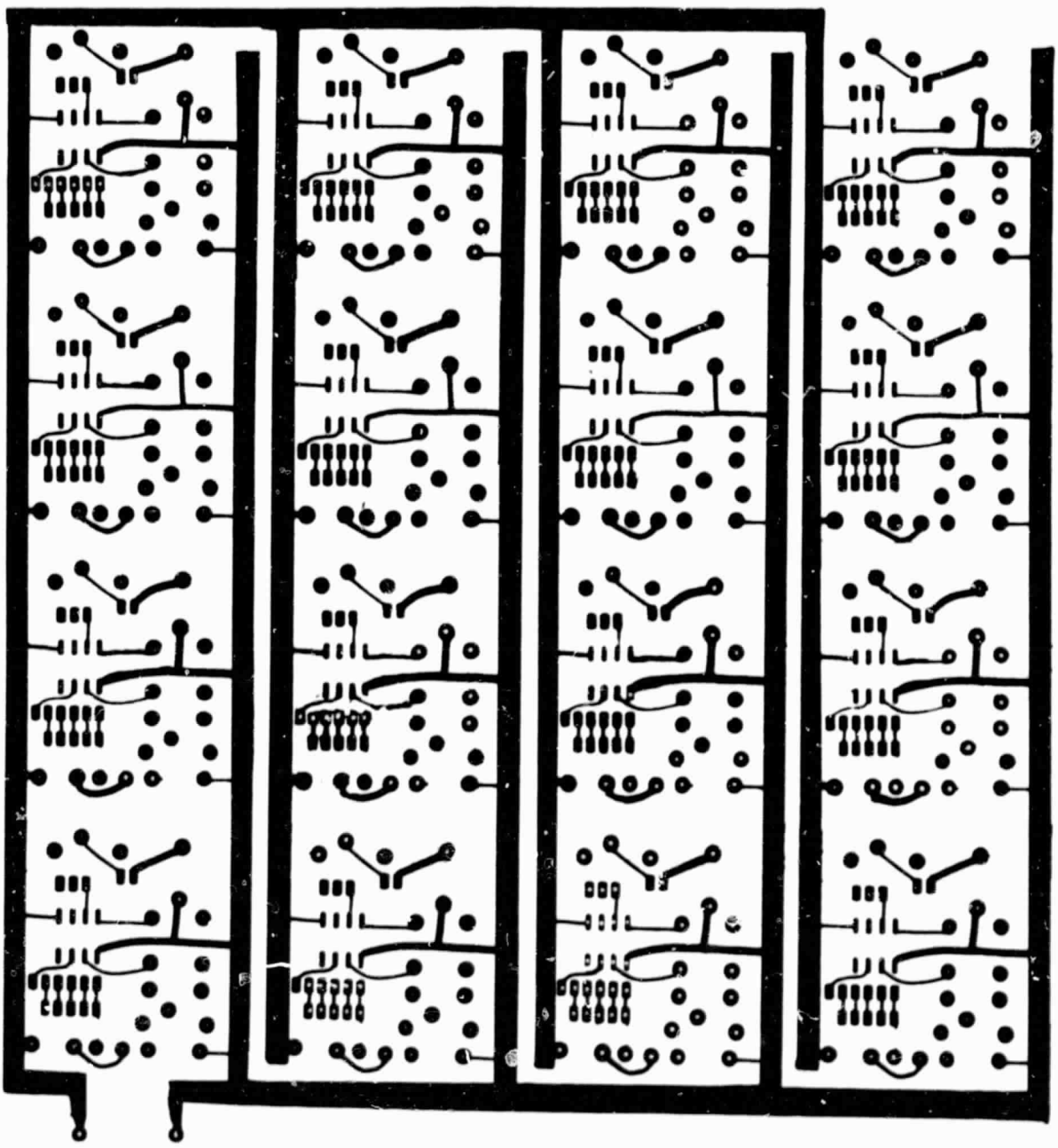


Figure B-9 Analog Conditioning Board - Top Side (.80 Scale)

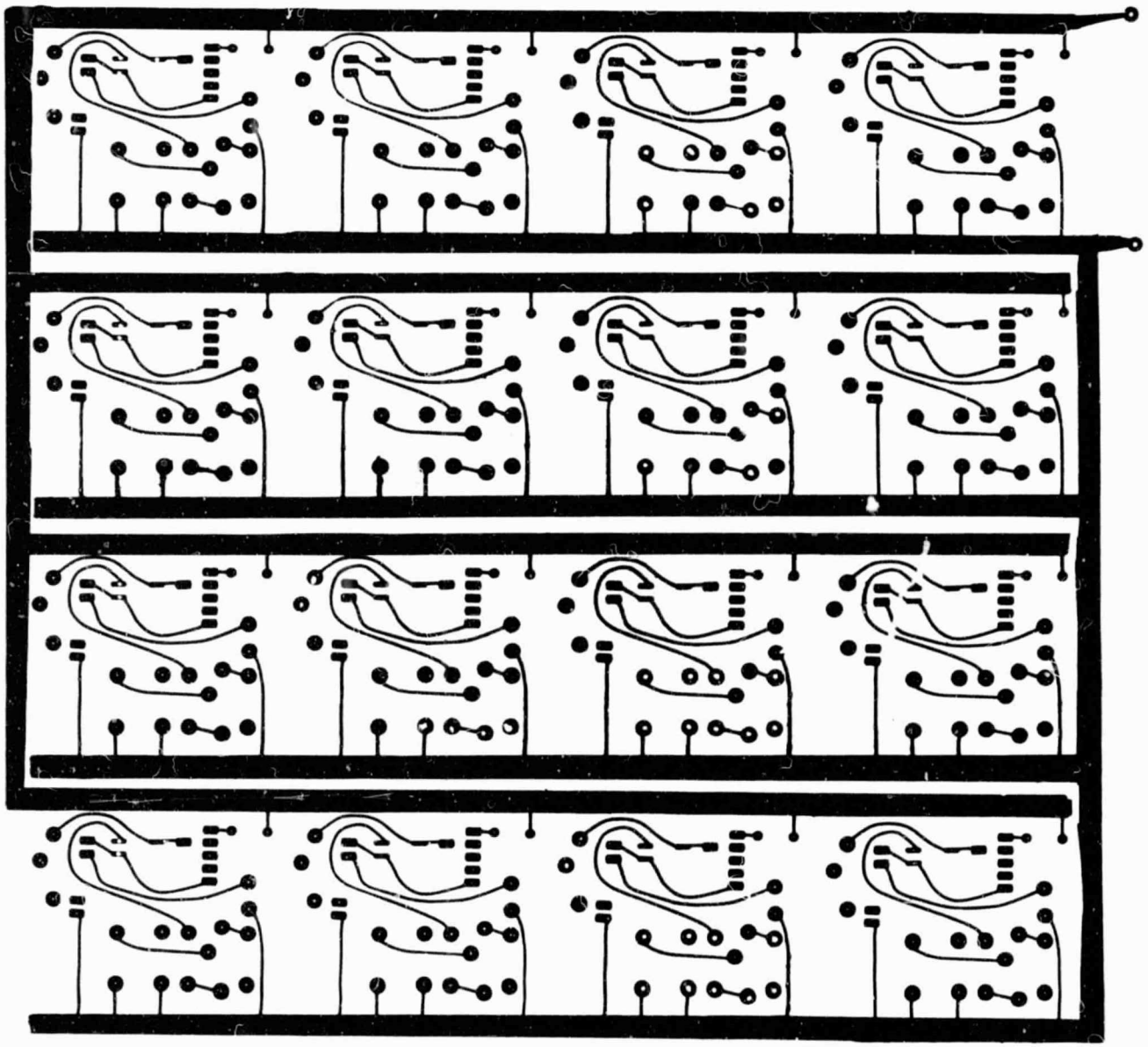


Figure B-10 Analog Conditioning Board Bottom Side (.80 Scale)

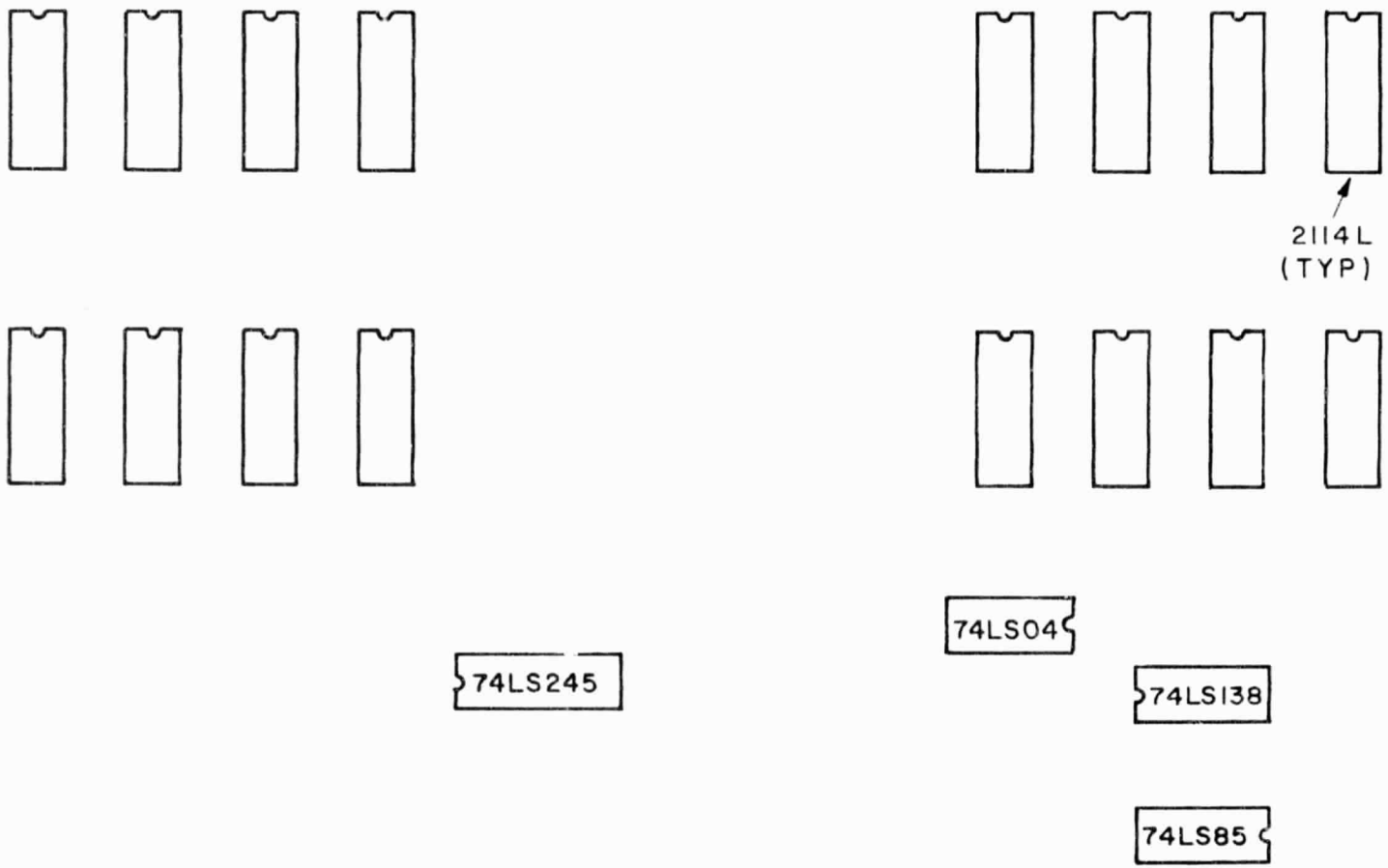


Figure B-11 Expansion Memory Board Component Layout - Top View (Full Scale)

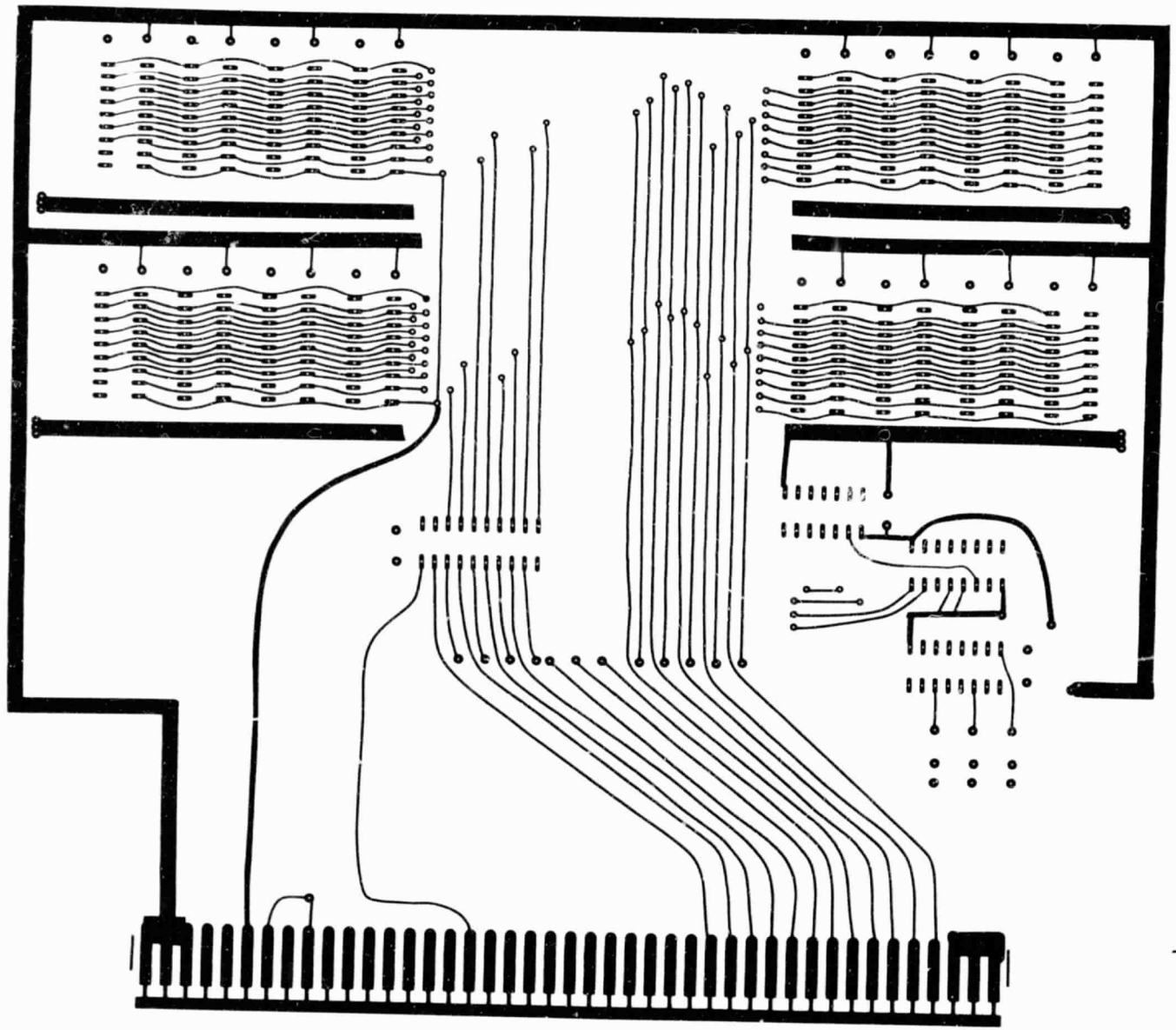


Figure B-12 Expansion Memory Board Top Side (.80 Scale)

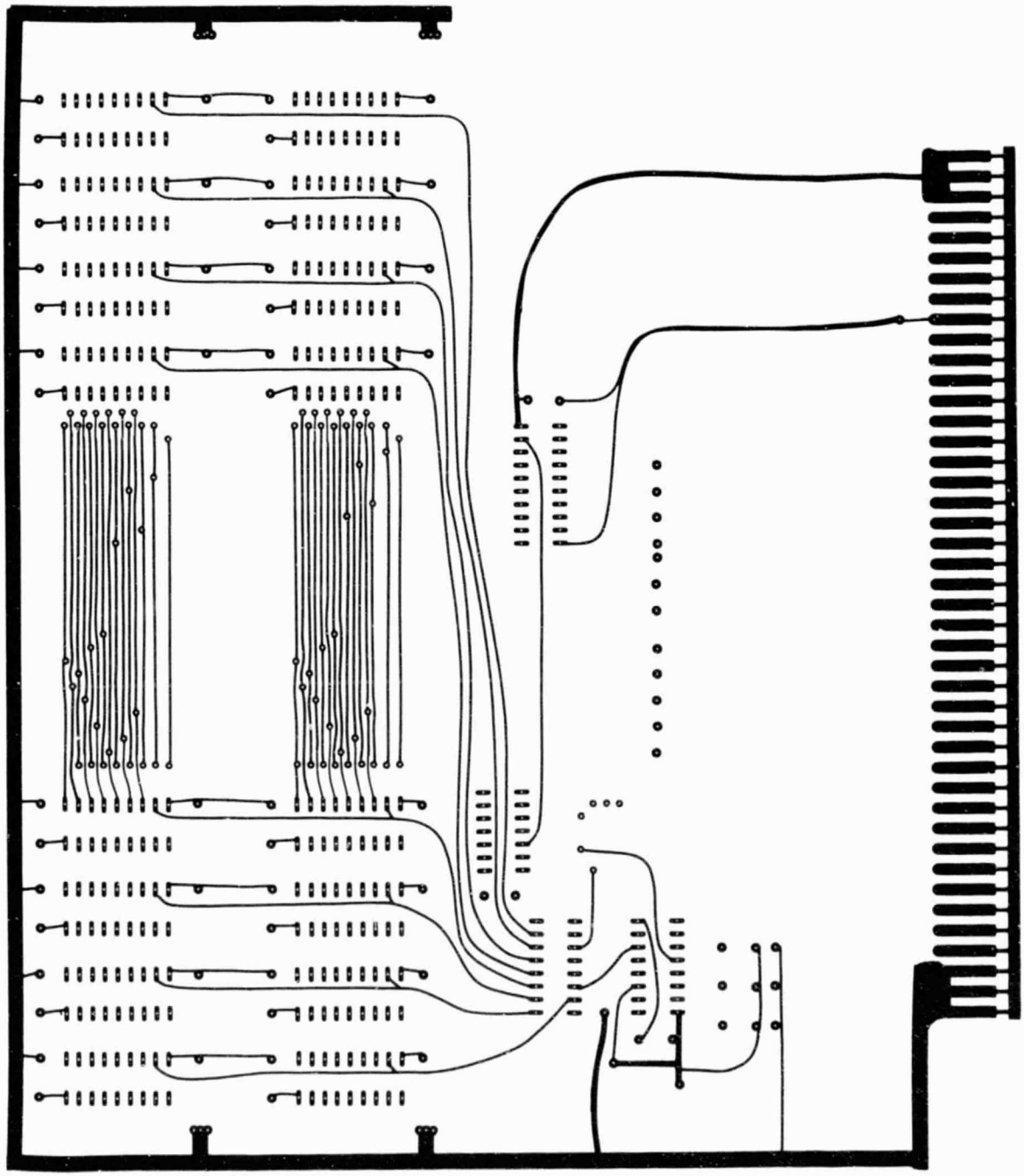


Figure B-13 Expansion Memory Board Bottom Side (.80 Scale)

APPENDIX C - GLOSSARY OF SELECTED MNEMONIES

ADC	-	Analog to Digital Converter
A/D	-	Analog to Digital
D/A	-	Digital to Analog
DMA	-	Direct Memory Access
HER	-	Hard (Parity) Error
IC	-	Integrated Circuit
IER	-	Interrupt Enable Register
IFR	-	Interrupt Flag Register
I/O	-	Input/Output
LSB	-	Least Significant Bit
LWD	-	Last Word
MSB	-	Most Significant Bit
PCR	-	Peripheral Control Register
VIA	-	Versatile Interface Adapter (a 6522 IC)