UNITED STATES DEPARTMENT OF INTERIOR
GEOLOGICAL SURVEY

AN INEXPENSIVE, BUFFERED ANALOG-TO-DIGITAL CONVERTER FOR USE ON THE SCSI BUS WITH THE UNIX OPERATING SYSTEM

Peter L. Ward

U. S. Geological Survey
345 Middlefield Road
Menlo Park, CA 94025

Reese Cutler

Cutler Digital Design
2215 Sun Mor Ave.
Mountain View, CA 94040

November 3, 1987

Open-File Report 87-624

INTRODUCTION

We received a small amount of money to install a seismographic network in Katmai National Park, Alaska, in order to study the relationship of earthquakes in the region to the volcanoes. This network included 16 seismic channels in the first year and might be expanded over several years to 64 or more channels. The existing options for recording these data are limited and expensive. Using individual drum recorders or digital event recorders becomes cumbersome and expensive for more than a few stations. Film recorders are difficult to maintain and are no longer commercially available. Most magnetic tape recorders are best suited for networks of less than 7 channels or more than 64. None of these systems allow online detection and location of earthquakes. Digital computer systems have proven to be the more desirable method to monitor and record data from such networks but have typically cost between $50k and $100k. Given that the cost of computers has been rapidly declining while their processing power has been escalating, we decided to try to develop a much lower cost computing system. Our design ideas were heavily influenced by the time available. Funding for this network was received in late April, 1987, and the system was installed in Alaska in early September, 1987. This document describes briefly our design concepts and the specific design.

DESIGN PRINCIPLES

A major factor in the design of computer systems is the cost of the software both in terms of time and programmers' salary. Nearly all of the software that we need to detect, process online, and interactively reprocess earthquake data exists and is running on a variety of minicomputers throughout the world. Much of this software has been written on systems with virtual memory capability and it would be difficult to trim these programs to fit computers with a severely limited address space. Thus our first fundamental decision was that the system must have virtual memory capability (address space of at least 16 megabytes) and should have a processing capability of at least 1 MIPS (Million Instructions Per Second).

The next decision was that the processor must have true multi-user and multi-tasking capability so that many programs could be run simultaneously without our software having to distribute system resources. In this way the large pool of existing software could be implemented rapidly and the configuration of software could be changed easily depending on need.

The third basic decision was to use the UNIX operating system (UNIX is a trademark of Bell Laboratories). All of the software needed was already running under UNIX. UNIX has become the industry standard portable, multi-user, multi-tasking operating system. Nearly all manufacturers of minicomputers now sell and support UNIX for their hardware. IBM and Apple have announced their intent to sell and support UNIX on the new generation of personalsu
microcomputers being introduced this year. Thus by chosing UNIX, not only can existing software be readily ported to our system, but our system can then be readily ported to hundreds of models of computers being developed by dozens of manufacturers. The choice of UNIX put some constraints on design of the analog-to-digital converter as discussed below.

These decisions ruled out use of IBM PC, XT, and AT compatible and Apple II and Macintosh I personal computers. We reconsidered this result carefully because, after all, these computers are inexpensive, widely available, and easily maintainable worldwide. These machines do not have sufficient power for large seismic networks and programming to fit their addressing and processing limitations would negate any apparent cost savings. Furthermore these systems are soon to be replaced by new models that will meet the above criteria and that would make such special programming obsolete. While there are good arguments for ultimately using such mass-produced computers, we decided that the most expeditious way to proceed in 1987 was to do the development on the best machine available, but to use industry standards so that we could readily port our system to whichever hardware was best in the future.

The next decision involved how to interface the analog-to-digital converter (A/D) to the computer. The standard approach for large microcomputers and minicomputers is via a bus such as VME bus or Multibus. Processors using such high-performance buses, however, are typically expensive. The most powerful and least expensive processors typically use the Small Computer Systems Interface or

SCSI bus to connect to peripherals. SCSI is rapidly becoming an industry standard as increasing numbers of disk drive and tape drive manufacturers are producing systems to connect directly to this bus. Several manufacturers provide chips that do all of the low level interface to this bus so that interfacing a new device is relatively easy. SCSI will be available for many of the new generation personal computers announced this year. We decided that if we could make our A/D look like a read-only disk on the SCSI bus, then it could be readily interfaced to a wide variety of processors.

UNIX is not a realtime operating system. In other words it can not guarantee to service any specific process on a set schedule within fractions of a second. In fact most multi-user, multi-tasking operating systems are not as realtime oriented as single task processors. High performance A/D systems get around this problem by using buffering. Thus we concluded that if the A/D could buffer data for as much as a few seconds, it should be able to work properly even in a heavily laden UNIX environment. Within UNIX it is possible to lock a given process in core and to give it the highest priority, so that it will be serviced regularly.

Our design goal was to build a system capable of digitizing data from 128 seismic components at a rate of 100 samples-per-second each. We wanted to use as many off-the-shelf components as possible to reduce development time and to minimize cost.

OVERALL DESIGN

We researched all computer systems available that would meet the above criteria and would cost under $20k including disk and tape subsystems. We concluded that the most mature and powerful (1.5 MIPS) processor available in mid 1987 for the lowest price was the SUN Microsystems Model 3/50 Workstation. This system with a floating-point processor, 4 megabytes of memory, 142 megabytes of disk storage, and 60 megabytes of cartridge tape storage was available for about $10k.

We were unable to find a buffering A/D interfaced to the SCSI bus and were thus forced to build our own. A/D chips, however, can be bought off the shelf, so that what we had to build was only an A/D controller. SCSI controllers are also available off the shelf, but we were unable to get one within a reasonable time period and thus had to build our own. The overall design of the A/D unit is shown in Figure 1.

The dynamic range of the seismic telemetry system is less than 60db, so that a 12 bit A/D is more than adequate. This leaves 4 bits of each 16 bit word unused. We decided to use these extra bits to record time information. The Kinemetrics/True Time GOES Satellite Synchronized Clock (Model 468-DC) provides absolute time accurate to a millisecond and has an option for parallel output of the time information in BCD format. Thus by using each set of 4 bits for one BCD digit, the complete time of year from day of year thru millisecond can be encoded in 12 samples. The clock also puts out 4 error bits. Table 1 shows the information encoded in each sweep of 16 channels of seismic information. The clock data are latched into registers at the beginning of each sweep and then added 4 bits at a time to the A/D data stream. By placing the time information in the least significant bits, i.e. in the noise, the data can be processed without removing the time information.

The A/D is a Burr-Brown SDM854 Hybrid Data Acquisition System that contains not only an A/D, but a sample and hold amplifier, 16 channel multiplexer, clock, delay timer, and multiplexer address latch. Each channel of this internal multiplexer can then receive data from an external multiplexer (Burr-Brown MPC16S CMOS Analog Multiplexer) giving input from up to 256 channels. For ease of design and buffering, the number of channels in our system is selectable in units of 16, the number of multiplexers active. Similarly the sample rate is adjustable only to 50, 100, or 200 samples per second.

The output of the A/D and clock goes into a FIFO (Integrated Device Technology Model IDT7202) to provide some flexibility in the processor's response to the data stream. At high throughputs near 25.6K samples per second, the processor might otherwise miss some samples.

Our initial idea was to use a large FIFO as the whole data buffer and use Programmable Array Logic (PAL) to control the A/D and SCSI bus. The cost of 600K bytes of FIFO, however, is

prohibitive. Furthermore the SCSI interface chip is designed to talk to a processor. Several people who had designed SCSI interfaces advised us that debugging a PAL-based SCSI system would be very difficult. Thus we decided to use a processor for buffering and control. The most inexpensive and well proven processor readily available to us was an Intel 8088 in the form of an IBM-XT clone. A fully configured XT was available for development with a cross assembler and a prom burner. Thus we could write and compile the program in a standard working environment, burn a prom, and insert it into the A/D. This established the physical configuration of the A/D as an XT clone case, power supply, and motherboard containing processor and memory. We then used standard XT full-length plugin prototype boards to hold the custom wire-wrap circuitry. All of these components were readily available over the counter. In addition a serial RS-232 port board was purchased to provide printer output during debugging of the SCSI interface.

The wiring diagrams are shown in Appendix A. There are three types of boards. Board 1 is the digital board (JDR Microdevices wirewrap prototype card model IBM-PR2) containing I/O interface logic for the XT, timing logic, and SCSI interface. Board 2 is the analog board (JDR Microdevices wirewrap prototype card model IBM-PR1) containing time source input, A/D converter, a DC-DC converter for power regulation, buffers, and decoding circuitry. Board 3 (also model IBM-PR1) contains multiplexers for 128 channels. Board 4 (not built) is a copy of Board 3 for 128 more channels.

The software that runs the processor as a standalone program is loaded in a 2K ROM and is listed in Appendix B. It uses an 8K byte segment of processor memory (RAM) for program variables and the remaining 632K bytes for buffer space.

A binary digital counter operates 8 LEDs that display the numbers of buffers waiting to be read over the SCSI bus. A buffer is 4096 bytes. This display shows immediately how well the interface to UNIX is working and whether the A/D is functioning properly.

The SCSI bus is very powerful and has numerous options, but it is not documented for beginners. The bus is defined in a standard distributed by the American National Standards Institute, New York, and described in a document called "Small Computer System Interface (ANSI X3.131-1986)". Delivery from ANSI takes months, but the document can also be bought overnight at higher cost from Global Engineering Documents in Santa Ana, California (800-854-7179). The standard is so broad that a subset of the standard is being developed in a document called "Common Command Set of the SCSI(X3T9.2/85-52 Rev 4.B)". Interface chips are made by Western Digital (WD33C93), NCR Microelectronics (NCR 53C80), and now other vendors. The vendors provide design manuals and sample software. We used the WD33C93 on the recommendation of several designers who thought it had fewer bugs since it was developed later and because it had higher level commands. The greatest problem in the development of this A/D system was to clearly understand the SCSI bus and bus language. Now that we have done it, it seems easy, but the documentation available is exceedingly terse and assumes a prior knowledge. There is no question, however, that if we did it again, we would choose SCSI. After all, we did figure it out in only a few weeks!

SUN SOFTWARE

SUN Microsystems Consulting Group sells a generic SCSI bus driver called CONSULT-SIP. The source code is provided with a manual entitled "Software Interface Description to SCSI Intelligent Peripherals". This code works under SUN release 3.2 of UNIX but does not work under release 3.4. We spent considerable time discovering this fact and ended up reinstalling 3.2 and using it for this work. SUN will be correcting the problems, caused by a major rewrite of the SCSI system drivers after release 3.2. This software provides the appropriate interfaces for the C Language subroutine calls open, close, read, write, and several options under ioctl (the IO control subroutine) that we used to identify the A/D, to reset the A/D, and to report errors. We designed the A/D interface to provide what this software expected. Porting this device to another machine will require a similar appropriate driver. A real benefit of the SCSI bus is that such a generic driver can be used. Writing a new SCSI bus driver would be a major undertaking.

PERFORMANCE

The noise on the analog channels was about 5 mv peak-to-peak with reference to a plus and minus 5 volt maximum input causing jitter in the second data bit. Since the seismic signals input to the multiplexers had noise greater than this, no effort was made to reduce this noise.

The A/D was run for days into a program that read the header information, checking that the time between samples was always between 9 and 11 milliseconds, checking that the multiplexer numbers were sequential, and checking that the distinct numbers were always in their proper place. The only errors found were traced to poor reception by the Satellite Clock due to a misaligned antenna and automatic seeking between the two satellites. Over a period of hours the time between samples did not change 8 specific times and then suddenly changed 80 milliseconds as the internal clock in the receiver was resynchronized to the satellite. Reorientation of the antenna and setting the receiver to use only one satellite corrected this problem. No loss of data by the A/D was ever detected.

Throughput of data proved to be no problem for up to 256 channels at 100 samples per second or 25.6K samples per second. The Burr-Brown Hybrid Data Acquisition System is rated at 27K sps. The FIFO and processor had no problem maintaining this rate because DMA channels were used and the processor was dedicated to data collection and transmission. This rate would not be attainable using the MSDOS operating system. The bandwidth of the SCSI bus is up to 4 megabytes per second with DMA input and output. The SUN 3/50 was able to receive the data and write it to disk at 25.6K sps. For testing purposes we installed a do loop in the main program on the SUN doing integer multiplies and discovered that 4 such multiplies per sample could be done without losing data. There was some indication that data may have been lost by the disk at specific moments, perhaps during switching of disk tracks. This problem was not regular, however, and we did not have time to investigate it fully. In cases where the highest throughputs are required, it may be wise to use a higher cost SUN model including a VME bus so that data is input on the SCSI and output to disk on the VME bus. Throughput was influenced by system load. If the A/D reading program was run at standard priority (nice=0) and a large amount of other disk or tape I/O or computing was going on simultaneously, the number of buffers in the A/D would increase and decrease, sometimes causing buffer overflow. If the A/D program was run at highest priority (nice= -10) this was rarely a problem and never a problem for less than 200 channels (20K sps). When buffer overflow occurs, the A/D sends an error message and continues to collect data so that the buffer always contains the most recent data. The possibility of overflow can be further decreased by a program called plock that is available from SUN Consulting to lock a process in core. We did not buy or test the effect of this on throughput, because we never expected to use more than 128 channels.

A standard long-term short-term detection program was used to select and save earthquake data (Herriot, J. W., unpublished data). We could run 224 channels of data through this program without losing data and using all of the system bandwidth. We tested the program extensively with data from 32 stations near Parkfield, California. Running the complete detection algorithm on all 32 stations and saving suspected events on disk used less than 10 percent of the computer resources. While this task was running in background, we could use the rest of the computer resources for editing, compiling, and other computing. We also modified this program so that the data was not only scanned with suspected earthquakes saved to disk files, but also written continuously to magnetic tape. This allows us to collect continuous data from the network when desired.

The only task that would backup buffers in the A/D was extensive use of the cartridge streamer tape drive. This tape drive hogs the SCSI bus even while rewinding, a problem that should be corrected by a better controller. A major feature of the SCSI bus is that when a device is asked to do something that will take a long time, the device can disconnect from the bus and then reconnect once the data are available. We use this feature in reading the A/D. The SUN requests one buffer (4096 bytes). If the A/D does not have one ready, it disconnects and then reconnects when the buffer is ready. The tape controller should do this but does not. Thus it is possible to lose some data from the A/D when writing a very long cartridge tape. There may be ways to mitigate this problem, but in our case we concluded that the problem was likely to occur so rarely, that we would not spend more time on it.

A program is listed in Appendix 3 that allows the A/D data to be displayed as it is collected and also allows display of the events detected along with the response of the detector algorithm. This

program provides for auto-scaling of the traces and shows the trace offset and scale. This program was written quickly and has not been refined.

ACKNOWLEDGEMENTS

Many people were involved in discussions that led to the development of this system. In particular Bill Jolitz of Symmetric Computer Systems suggested several key ideas. Jim Ellis and Gray Jensen of the U.S.G.S. helped extensively in evaluating options. Jim Herriot provided the detection program and assisted in implementing it. John VanSchaack, Wes Hall, Anselmo Rodrigues and Dick Fry provided major assistance in connecting the prototype system to incoming seismic signals from Parkfield. Rex Allen provided many useful comments and reviewed this document.

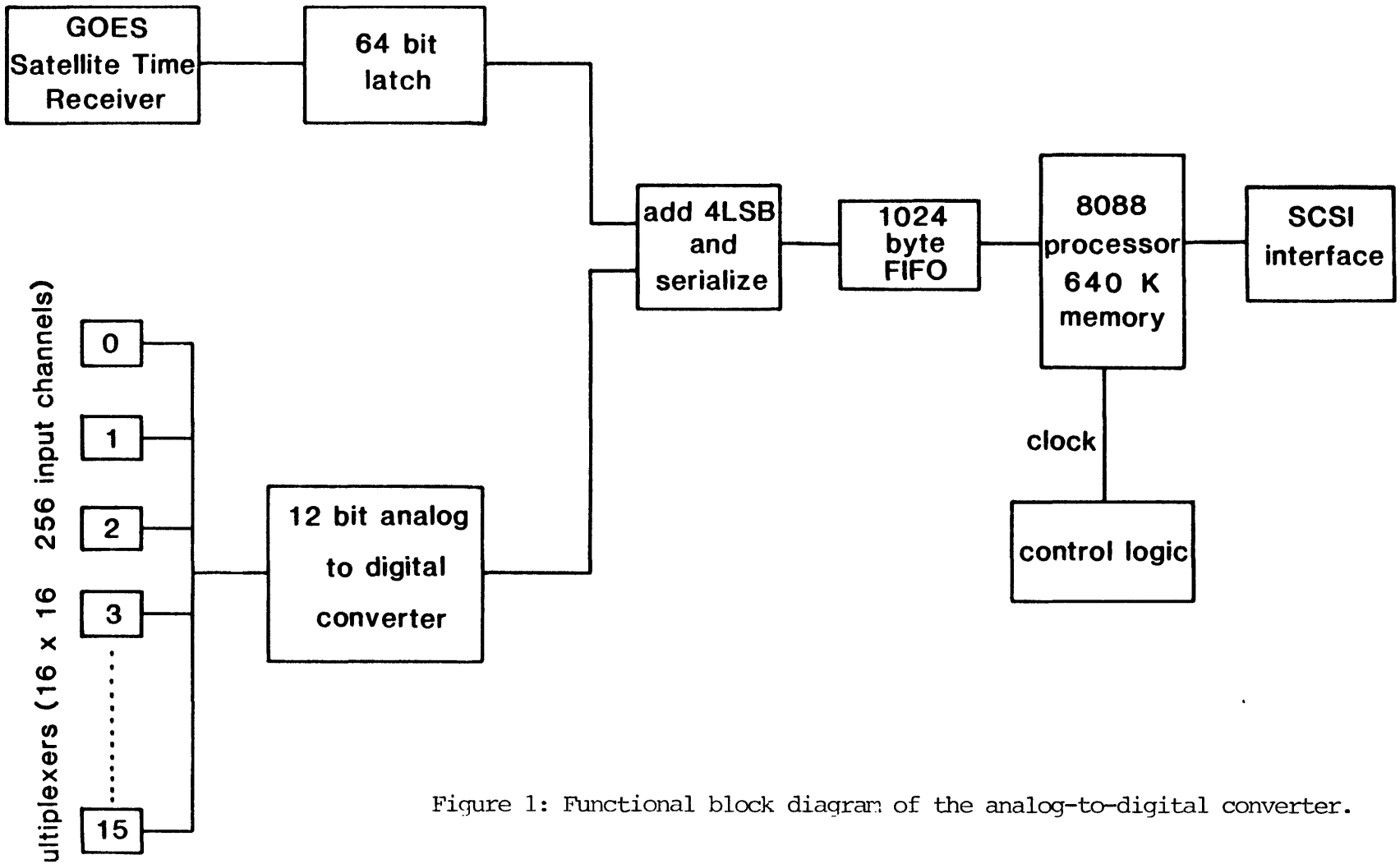
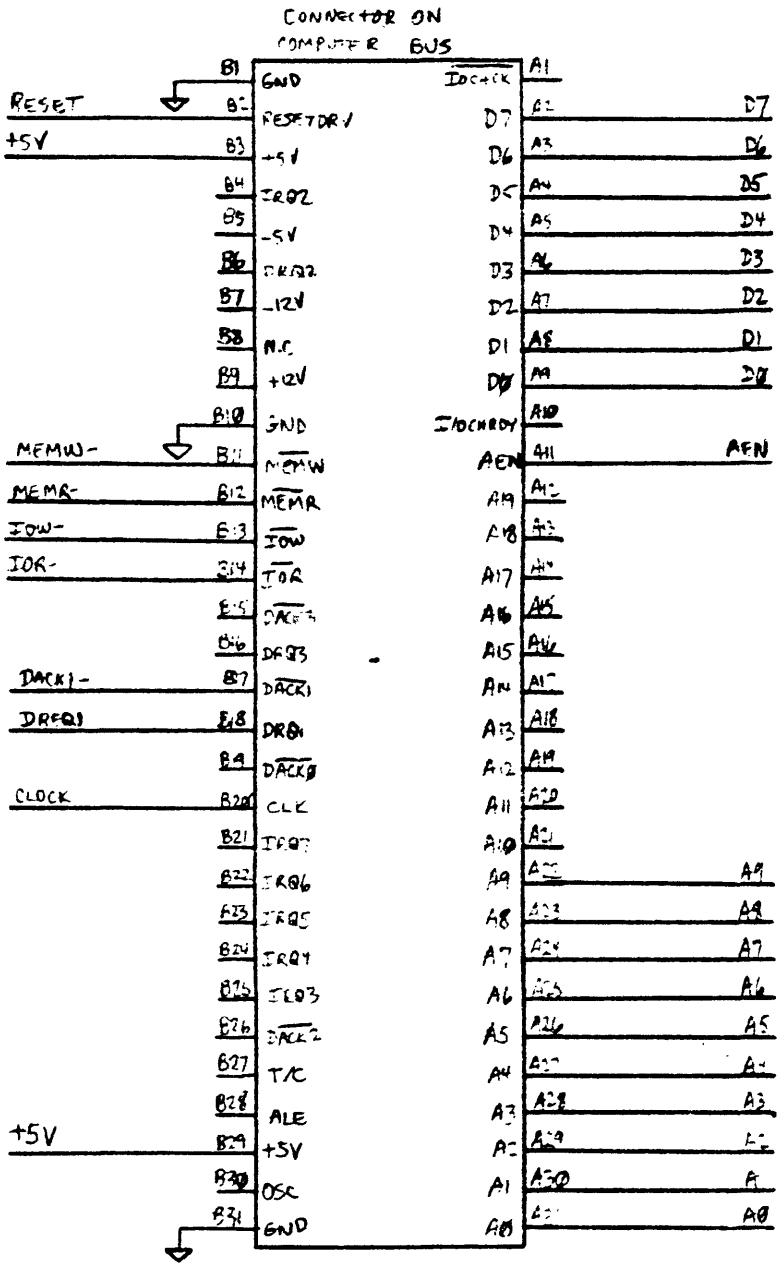


Figure 1: Functional block diagram of the analog-to-digital converter.

Table 1: BCD codes added to the least-significant 4 bits of each data word. These 16 codes are added to each sweep of 16 channels.

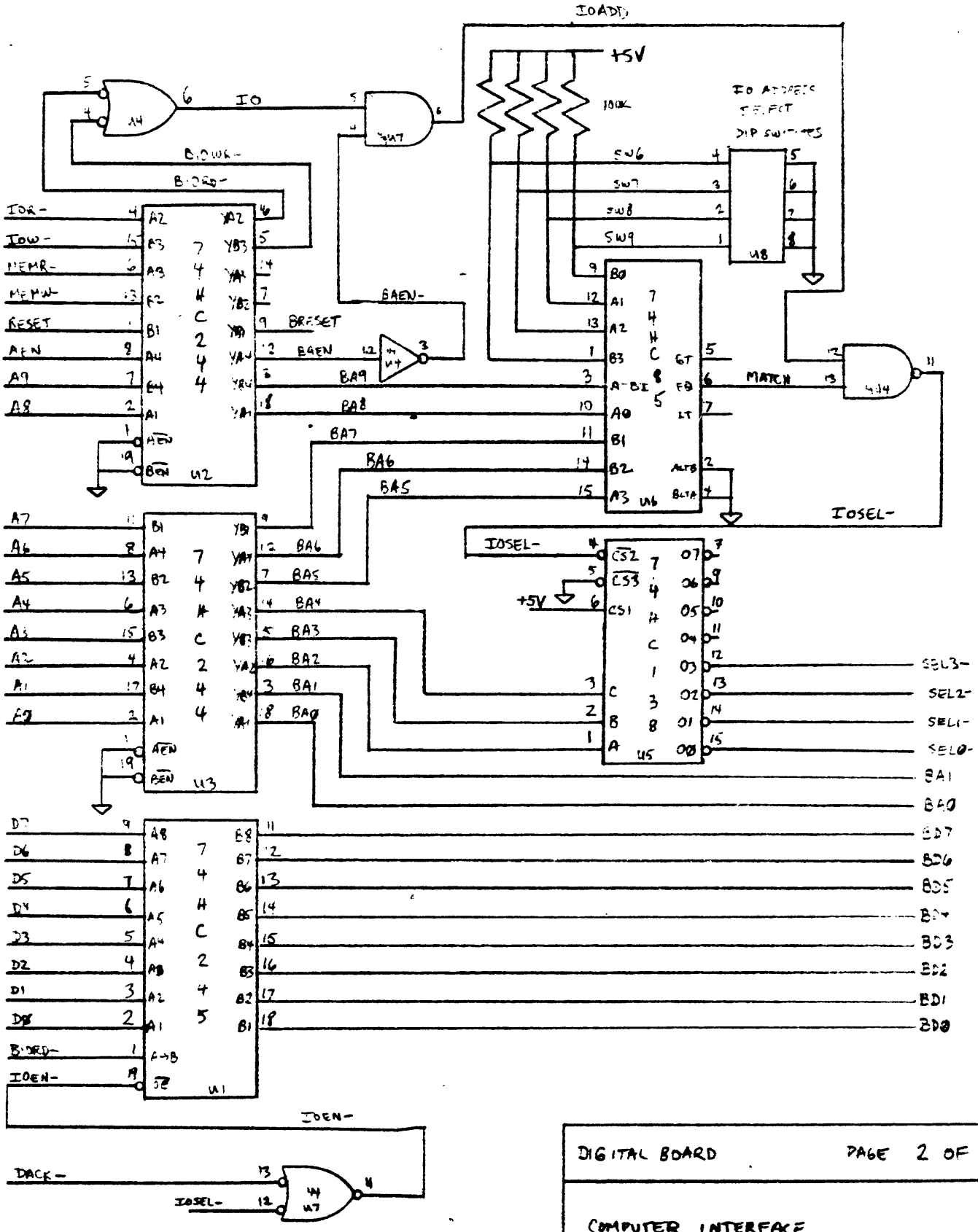
Distinct number (15)
Multiplexer number
Days (100s)
Days (10s)
Days (1s)
Hours(10s)
Hours(1s)
Minutes(10s)
Minutes(1s)
Seconds(10s)
Seconds(1s)
Milliseconds(100s)
Milliseconds(10s)
Milliseconds(1s)
Error bits
Distinct number (14)

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



DIGITAL BOARD	PAGE 1 OF 9
CONNECTOR TO COMPUTER BUS	

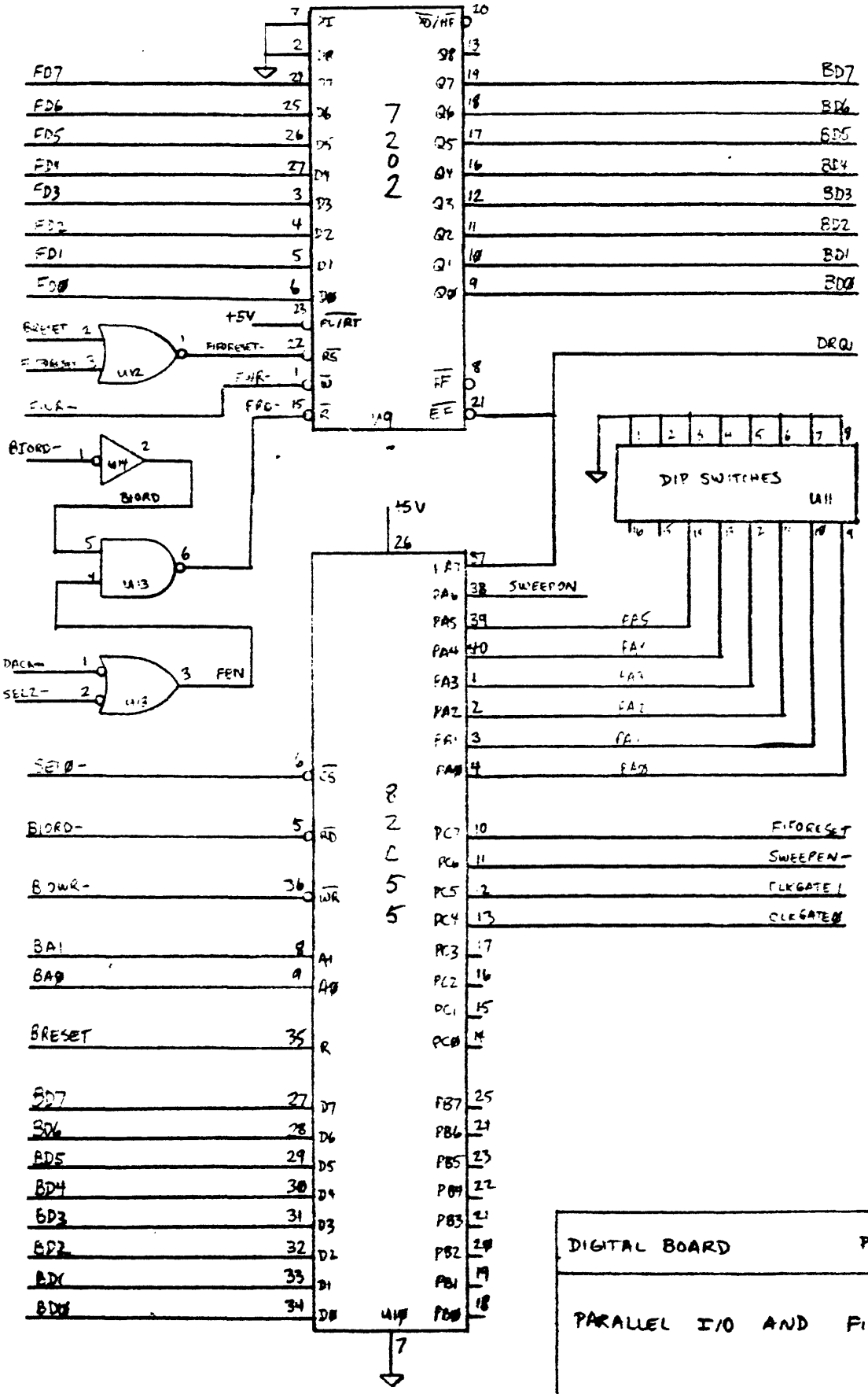
CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



DIGITAL BOARD PAGE 2 OF 9

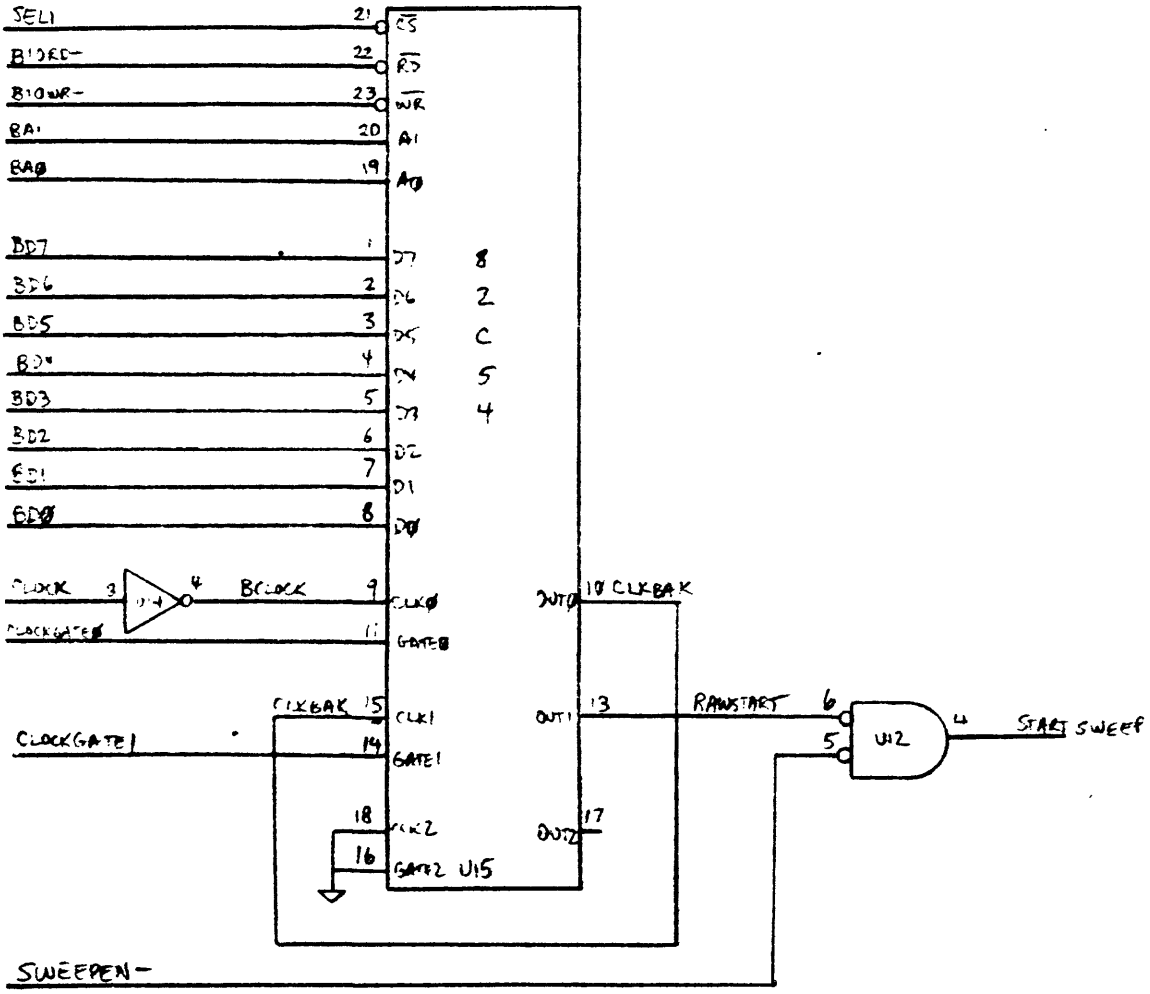
COMPUTER INTERFACE

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



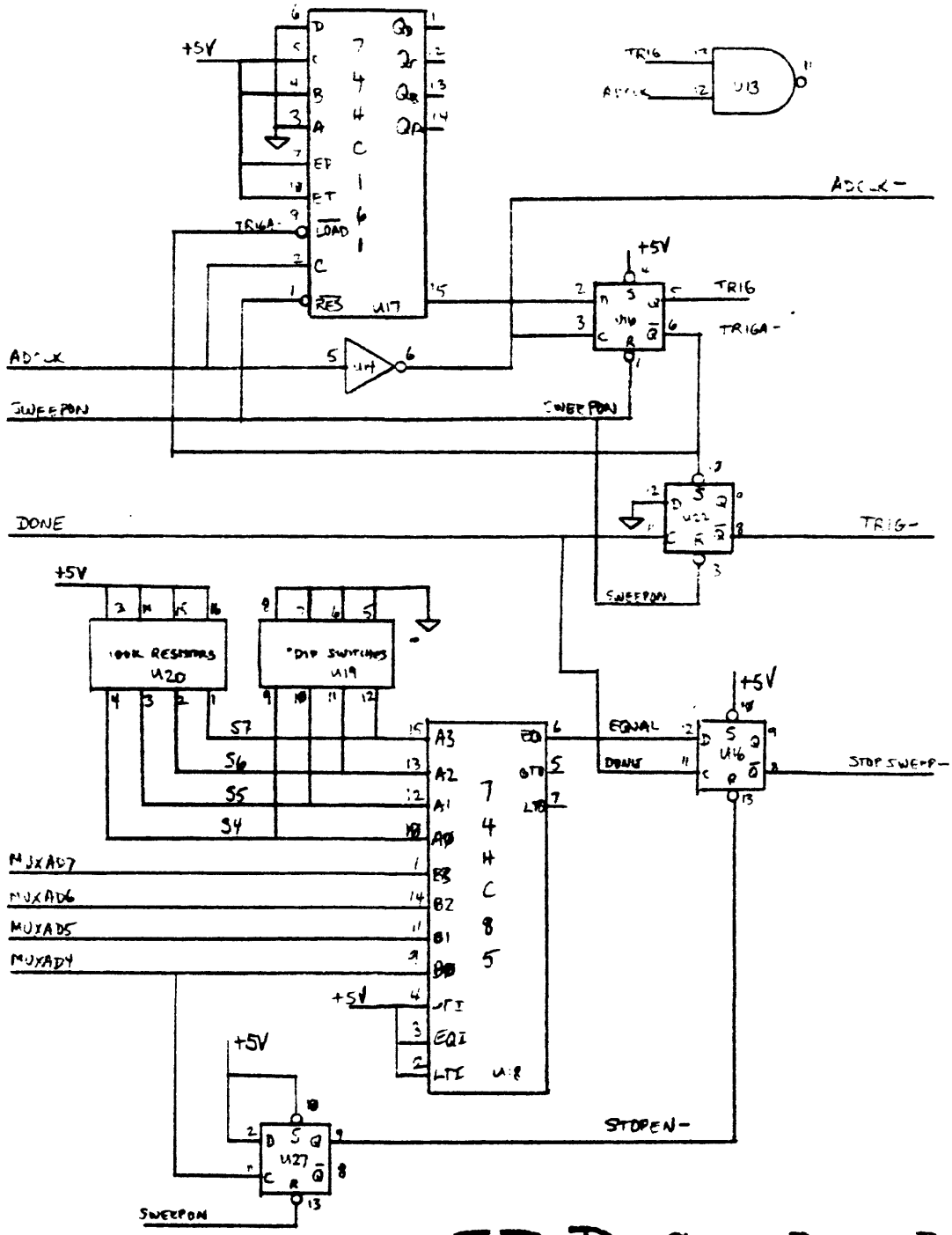
DIGITAL BOARD PAGE 3 OF 9

PARALLEL I/O AND FIFO



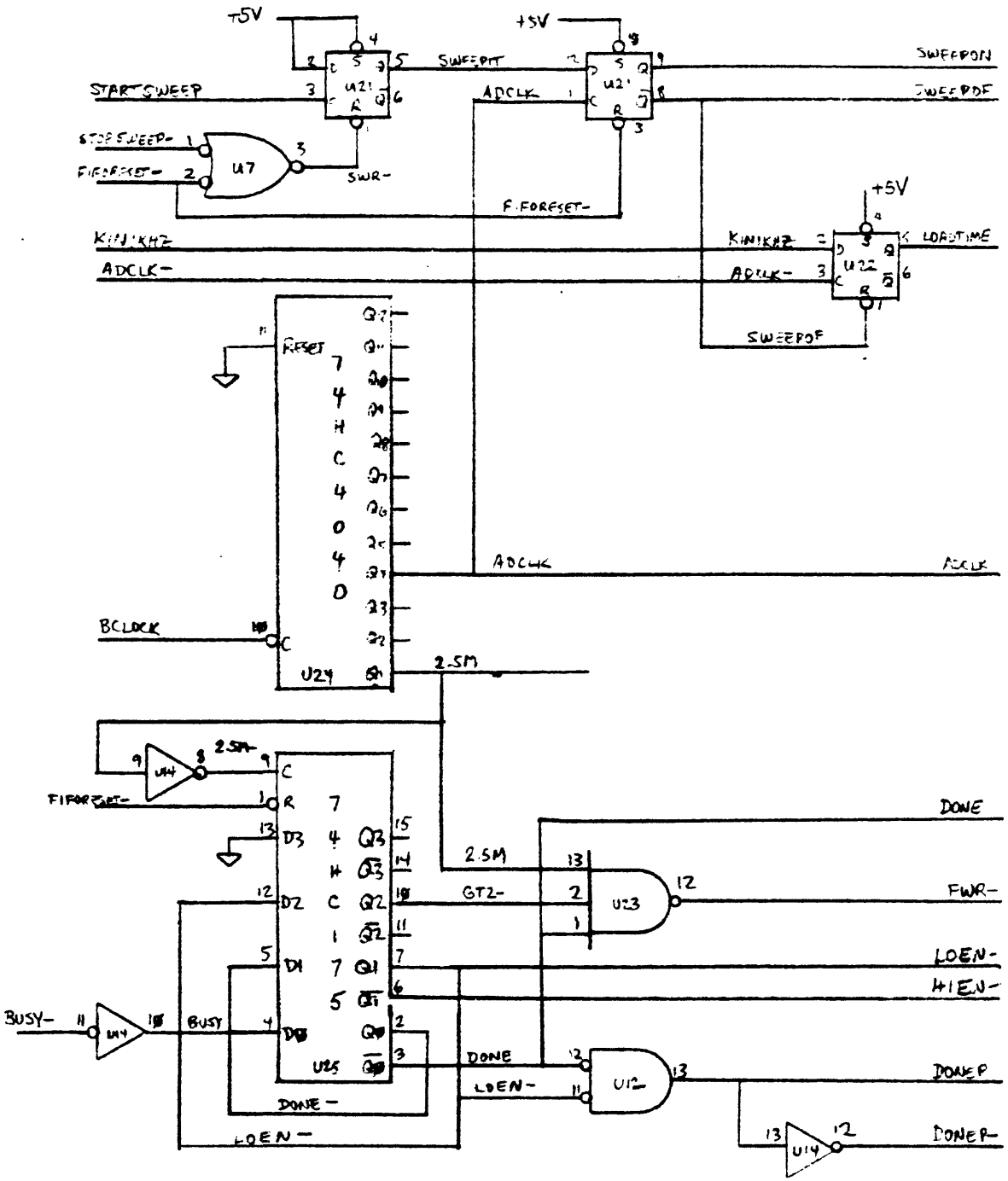
CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

DIGITAL BOARD	PAGE 4 OF 9
PROGRAMMABLE TIMER	



CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

DIGITAL BOARD	PAGE 5 OF 9
SWEEP TIMING LOGIC PART I	

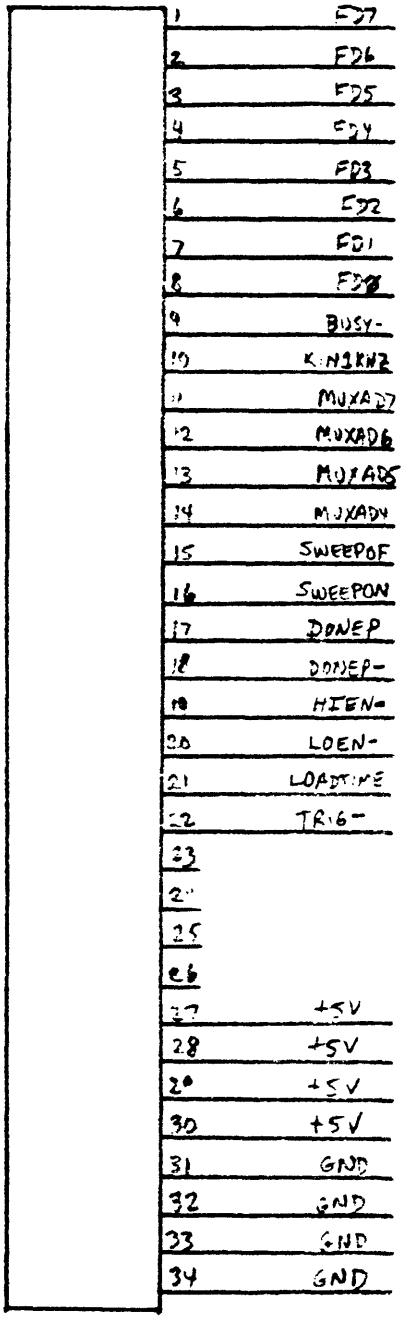


CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

DIGITAL BOARD	PAGE 6 OF 9
SWEEP TIMING LOGIC PART II	

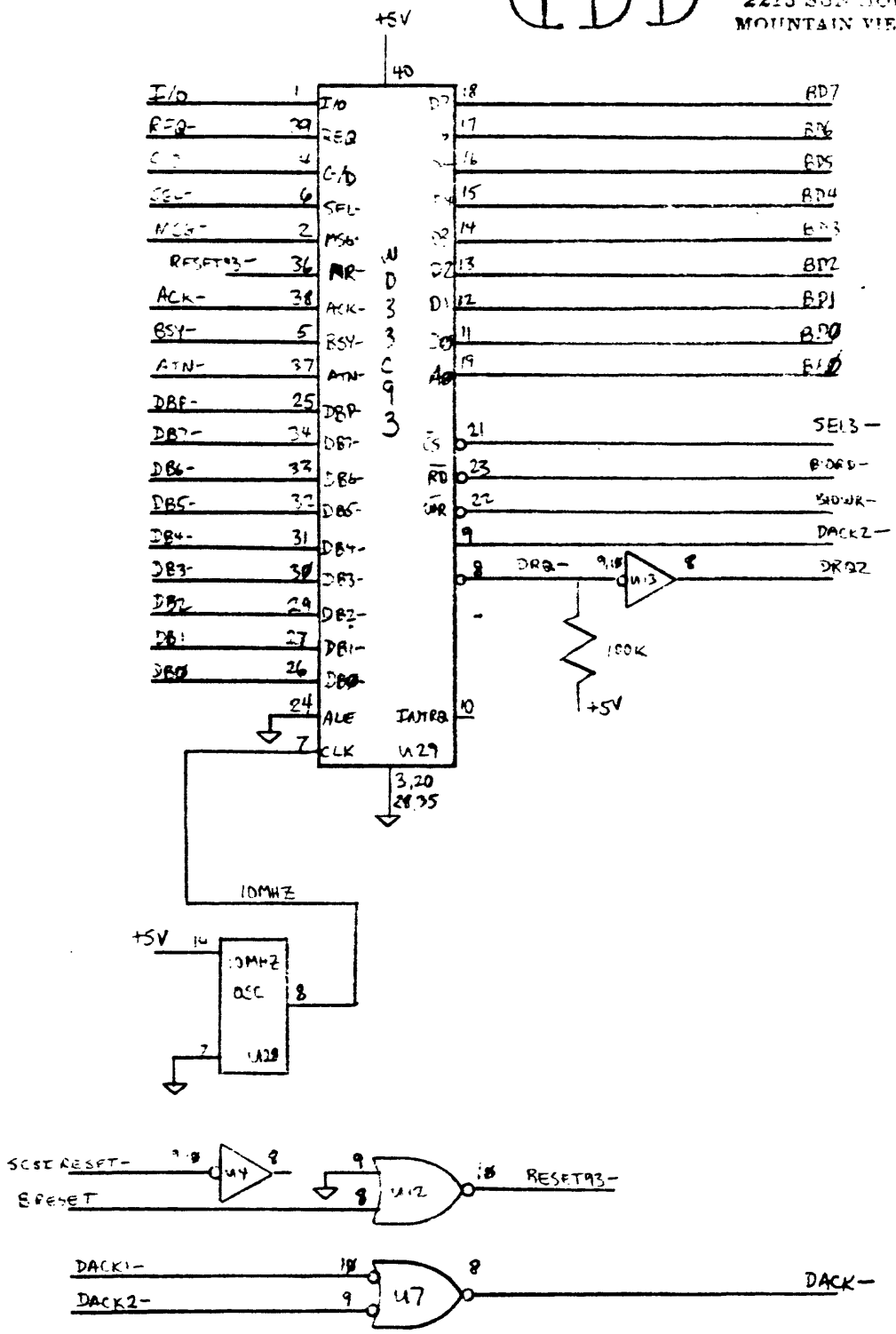
CUTLER DIGITAL DESIGN
2210 S[illegible] AVE.
MOUNTAIN VIEW, CALIF. 94040

(CONNECTOR → ANALOG BOARD)



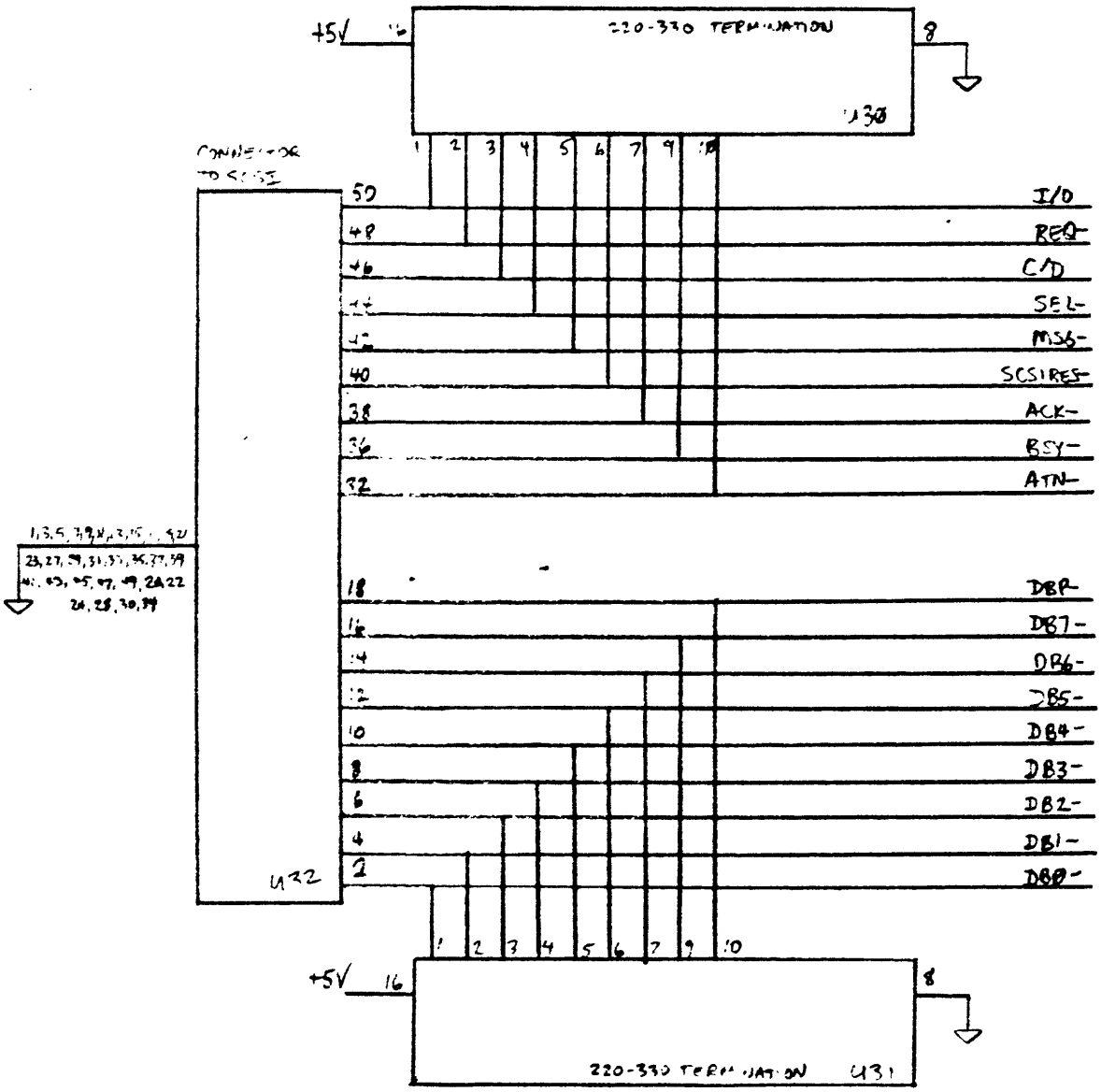
DIGITAL BOARD	PAGE 7 OF 9
CONNECTOR TO ANALOG BOARD	

CUTLER DIGITAL DESIGN
2215 SUNNOR AVE.
MOUNTAIN VIEW, CALIF. 94040



DIGITAL BOARD	PAGE 8 OF 9
SCSI CONTROLLER	

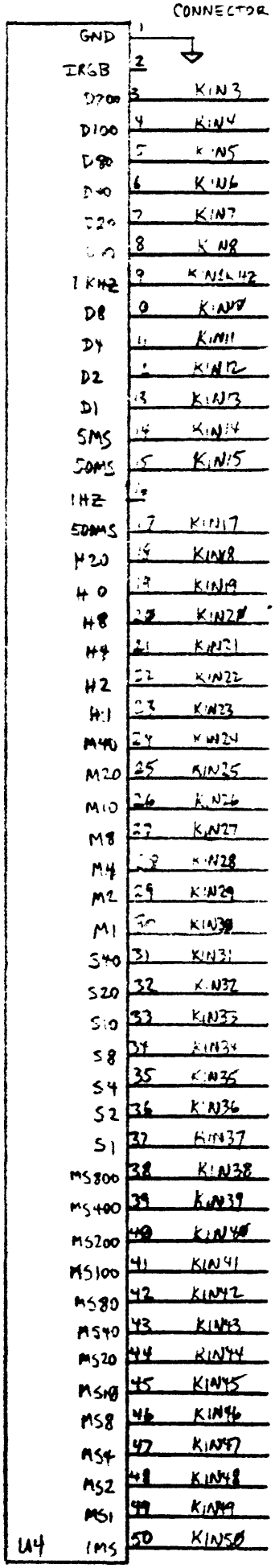
CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



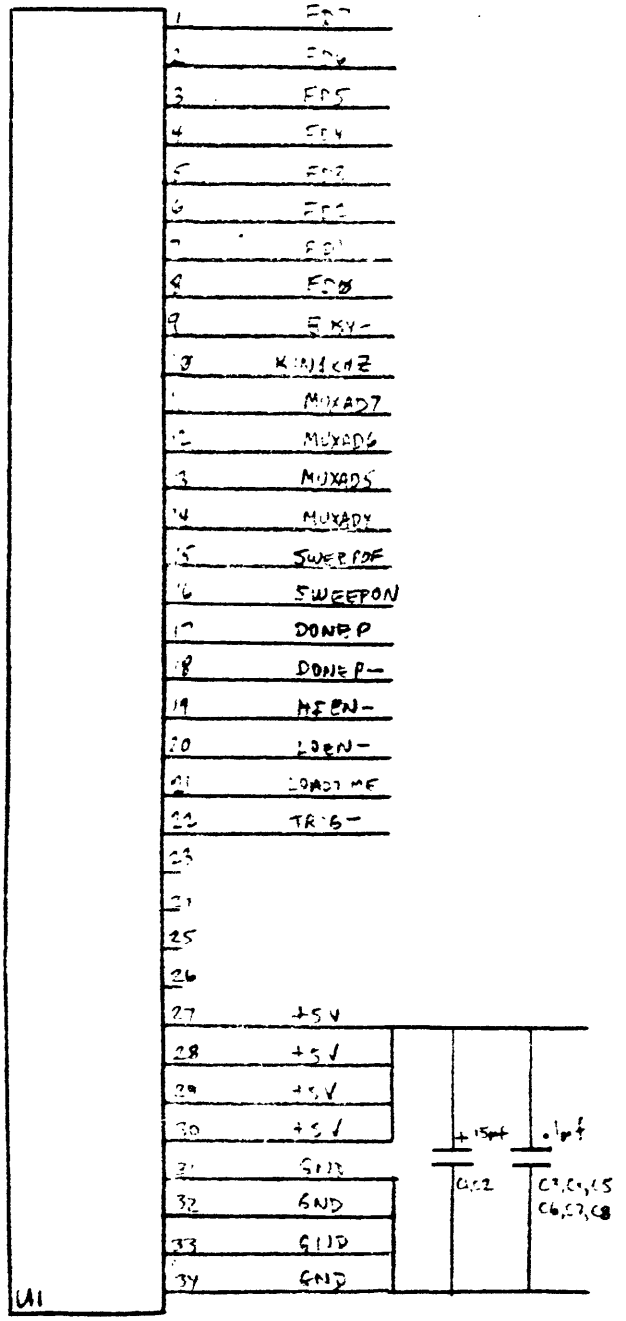
DIGITAL BOARD	PAGE 9 OF 9
SCSI CONNECTOR AND TERMINATION RESISTOR PACKS	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

CONNECTOR TO CLOCK



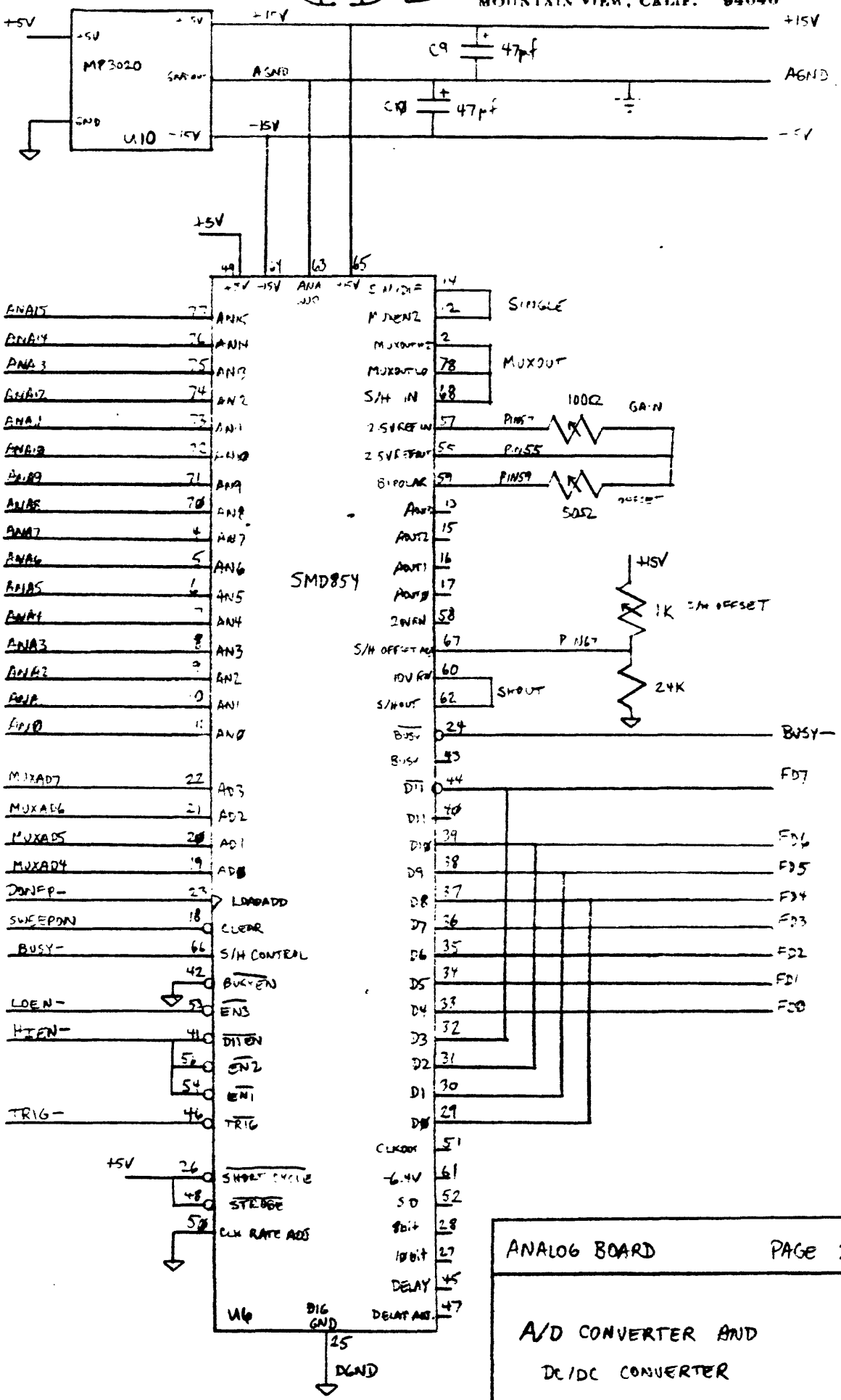
CONNECTOR TO DIGITAL BOARD



ANALOG BOARD PAGE 1 OF 6

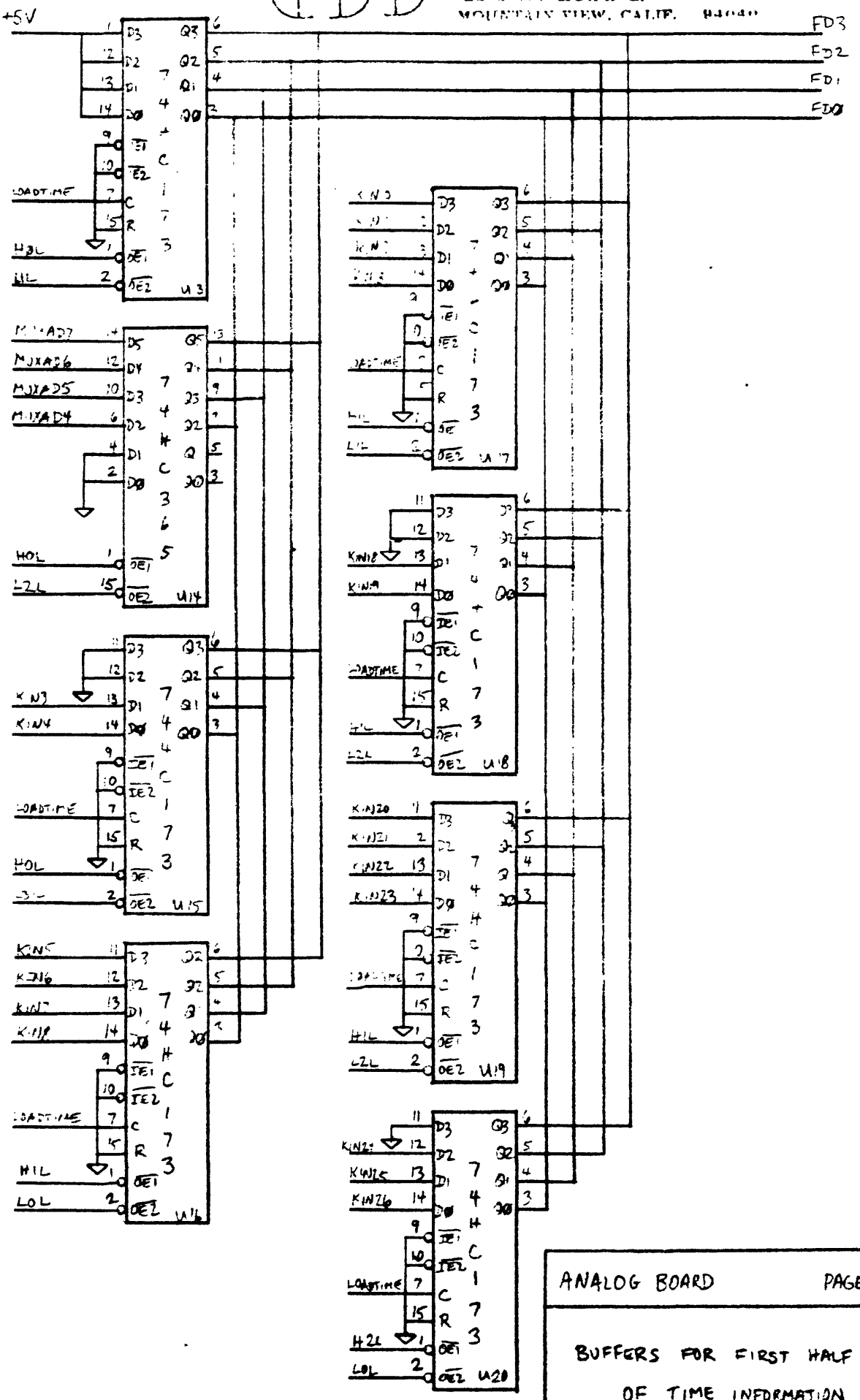
CONNECTORS TO DIGITAL BOARD AND TIME SOURCE

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



ANALOG BOARD PAGE 2 OF 6

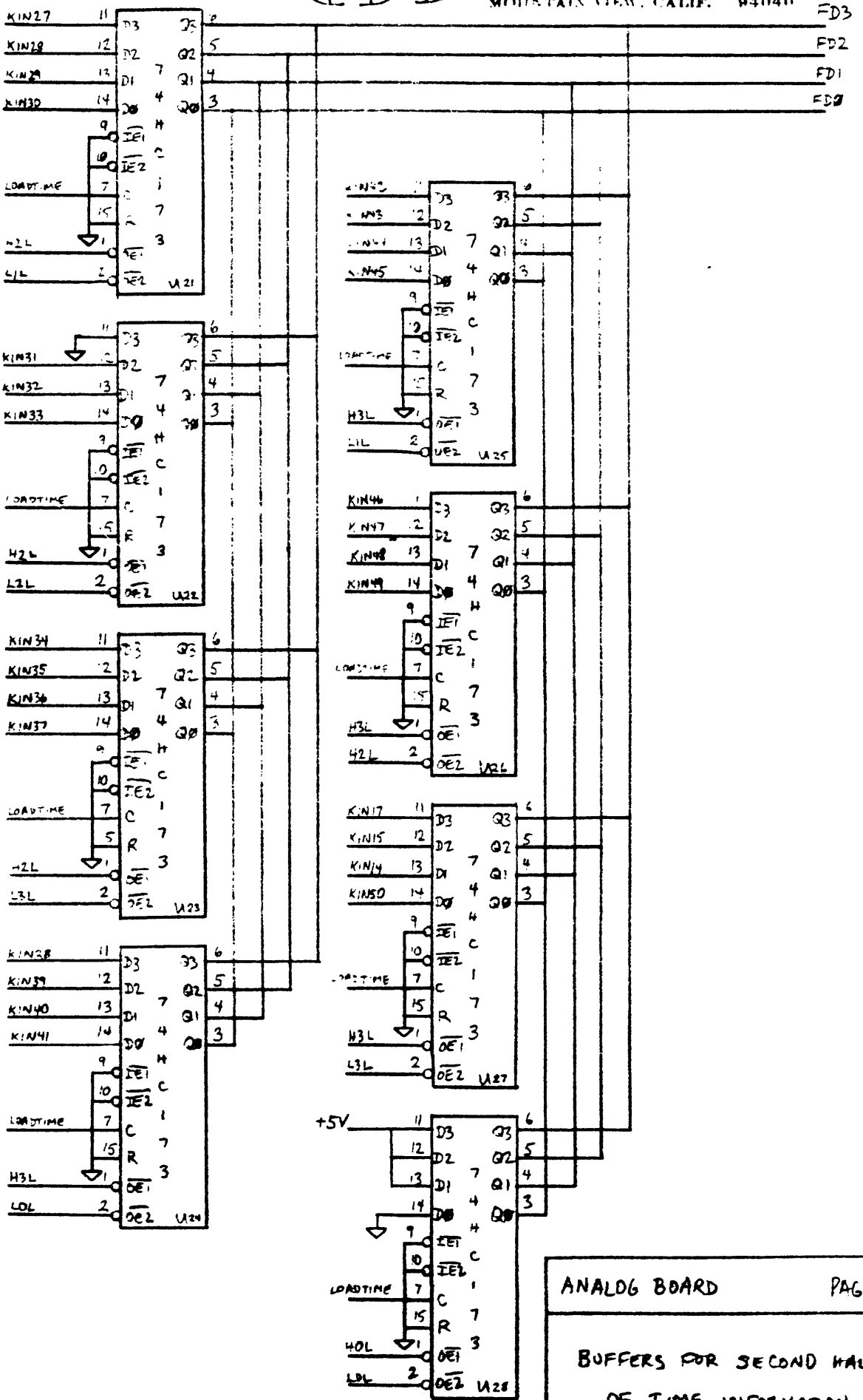
A/D CONVERTER AND DC/DC CONVERTER



ANALOG BOARD PAGE 3 OF 6

BUFFERS FOR FIRST HALF OF TIME INFORMATION

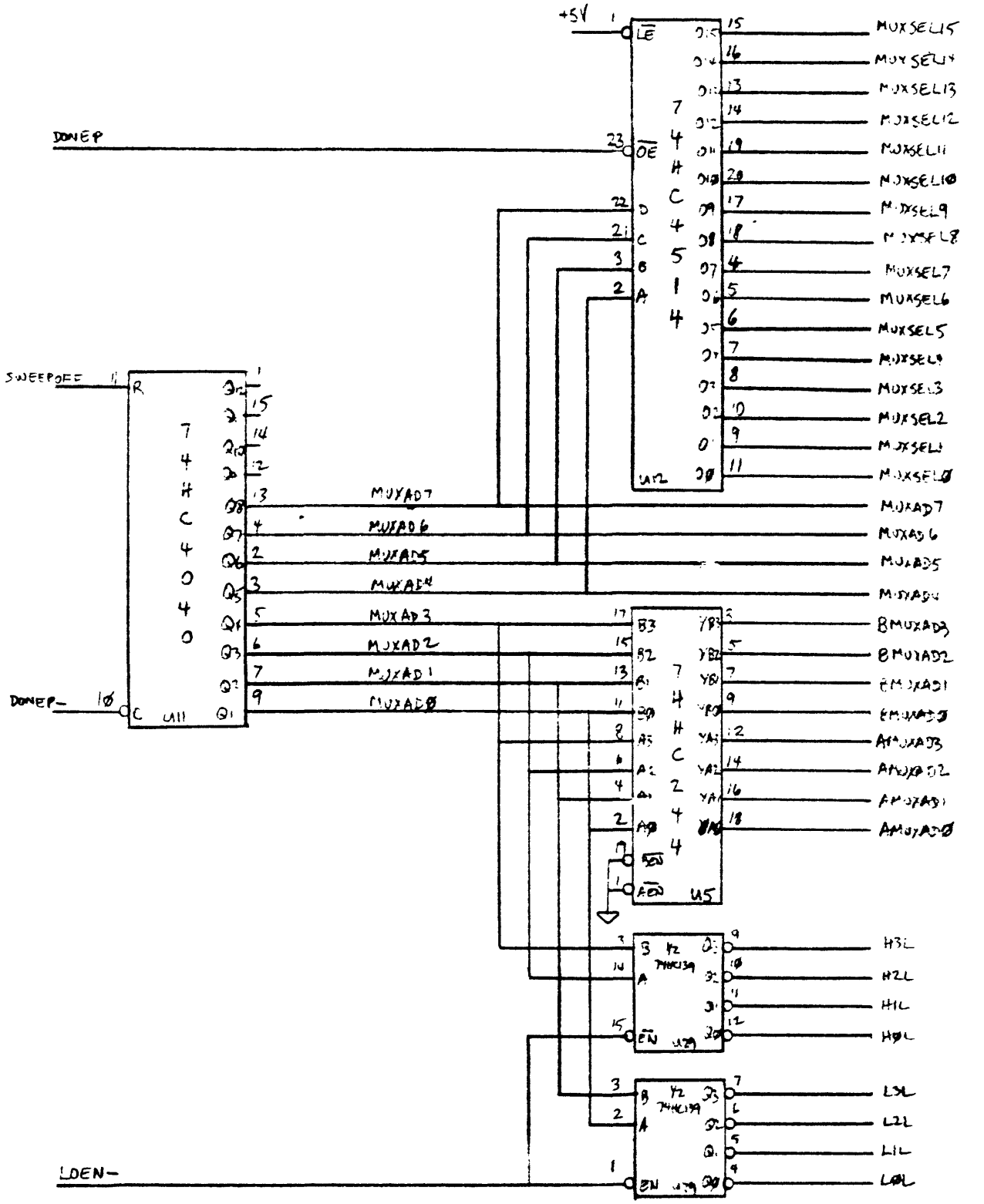
CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



ANALOG BOARD PAGE 4 OF 6

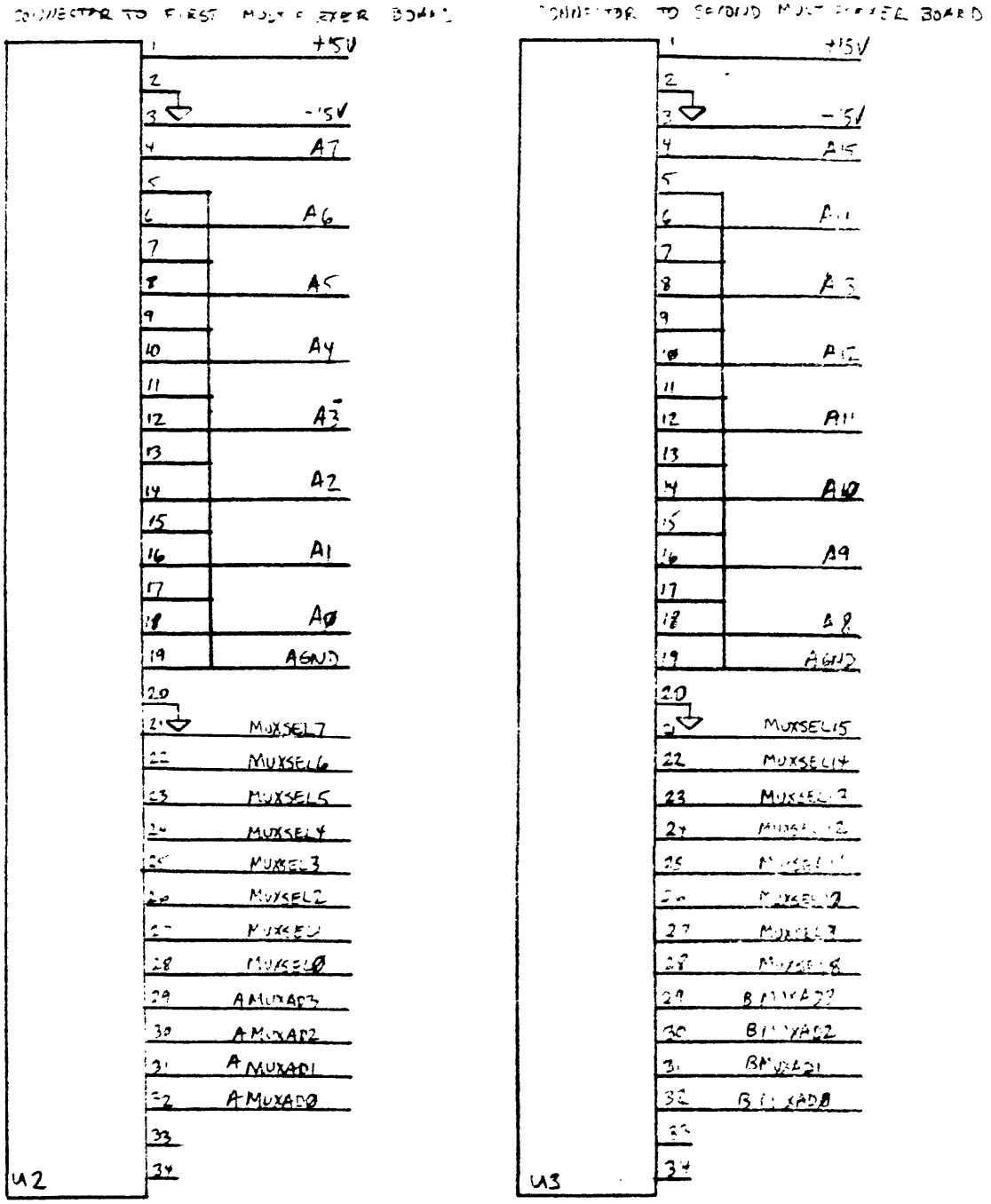
BUFFERS FOR SECOND HALF OF TIME INFORMATION

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



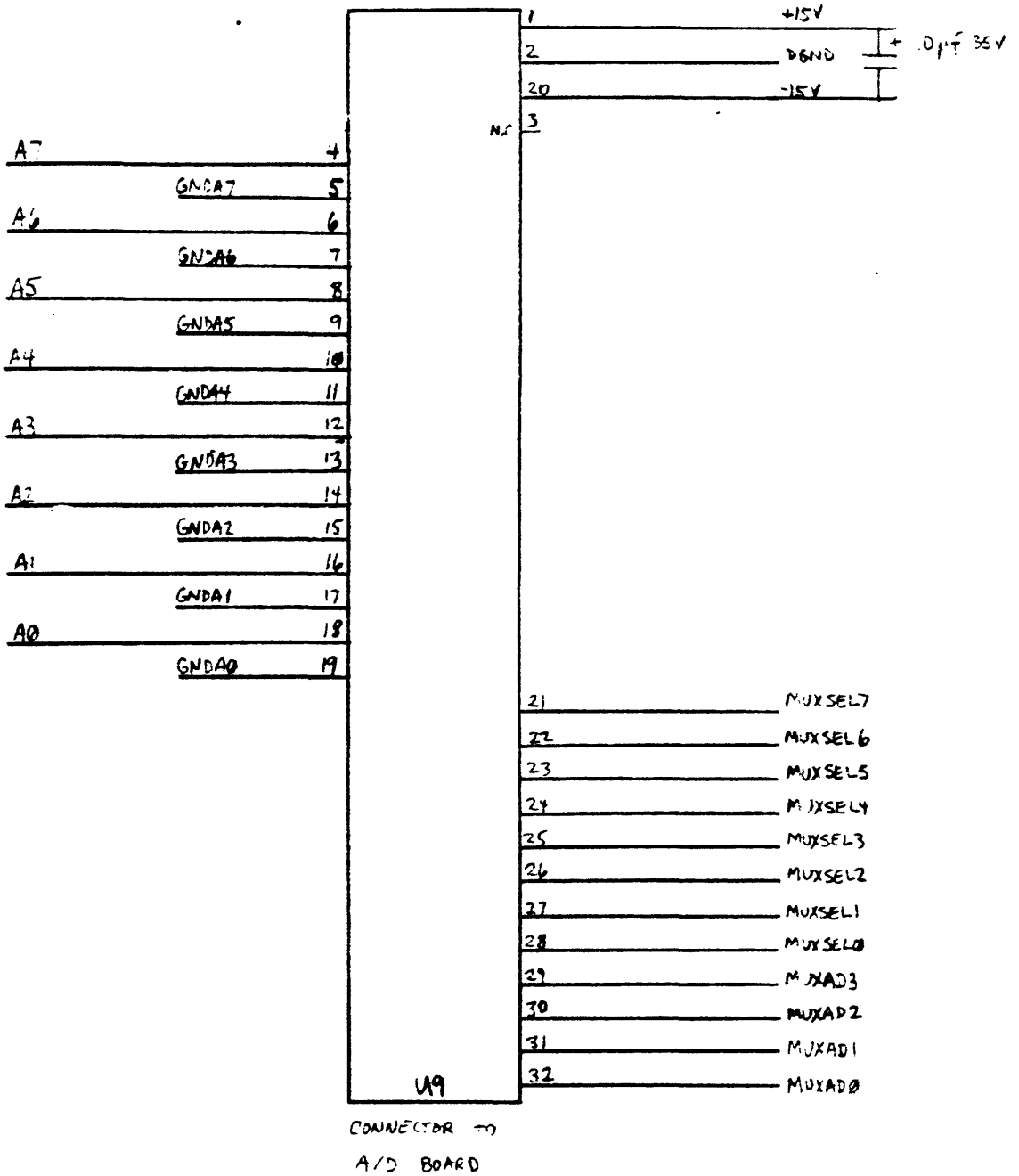
ANALOG BOARD	PAGE 5 OF 6
DECODING	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



ANALOG BOARD	PAGE 6 OF 6
CONNECTORS TO MULTIPLEXER BOARDS	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

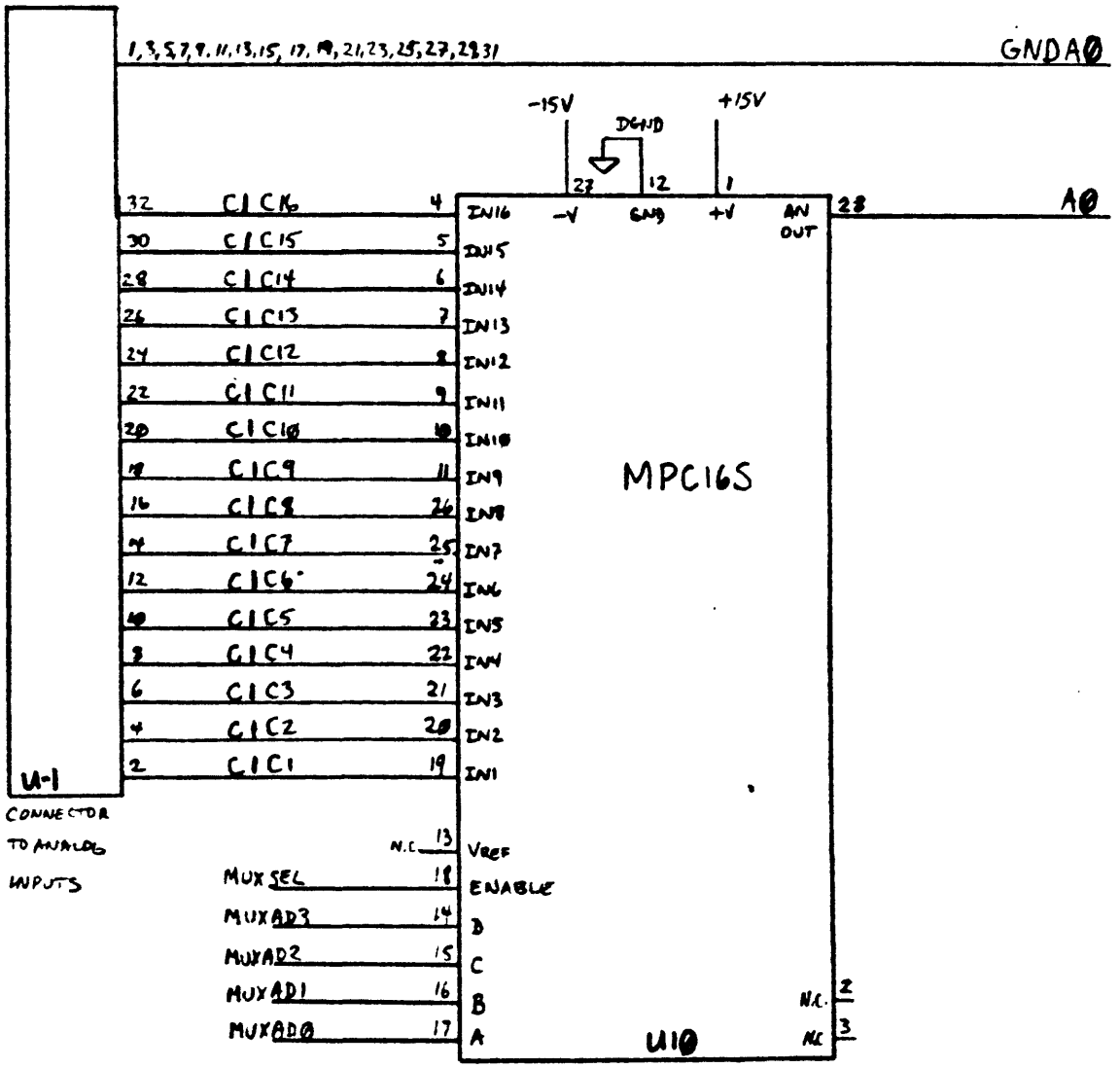


MULTIPLEXER BOARD PAGE 1 OF 9

CONNECTOR TO A/D BOARD

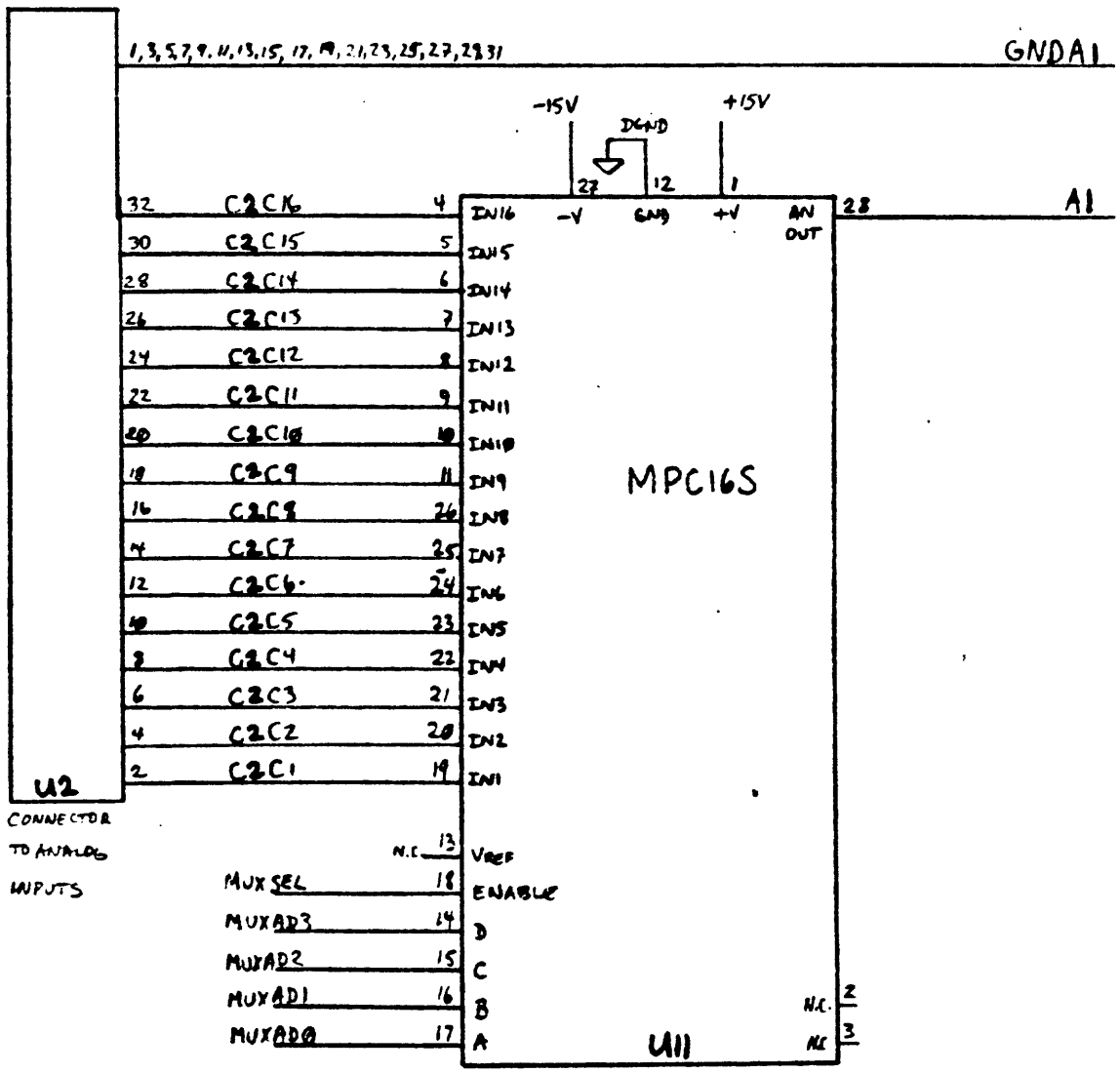
POWER BYPASSING

CHRYSLER DIGITAL DESIGN
2215 SUNMOR AVE.
MOUNTAIN VIEW, CALIF. 94040



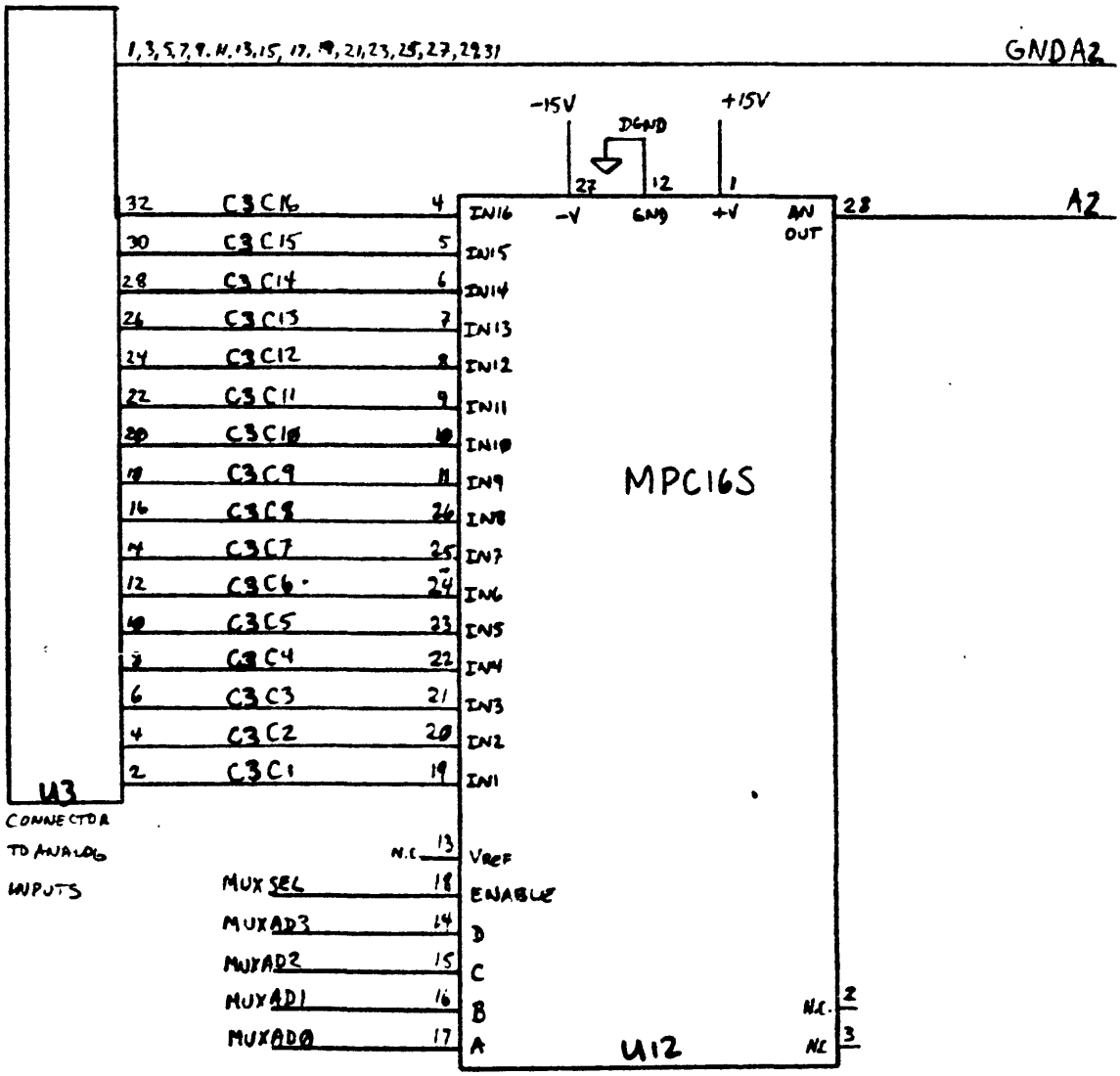
MULTIPLEXER BOARD	PAGE 2 OF 9
MULTIPLEXER Ø AND CONNECTOR	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040

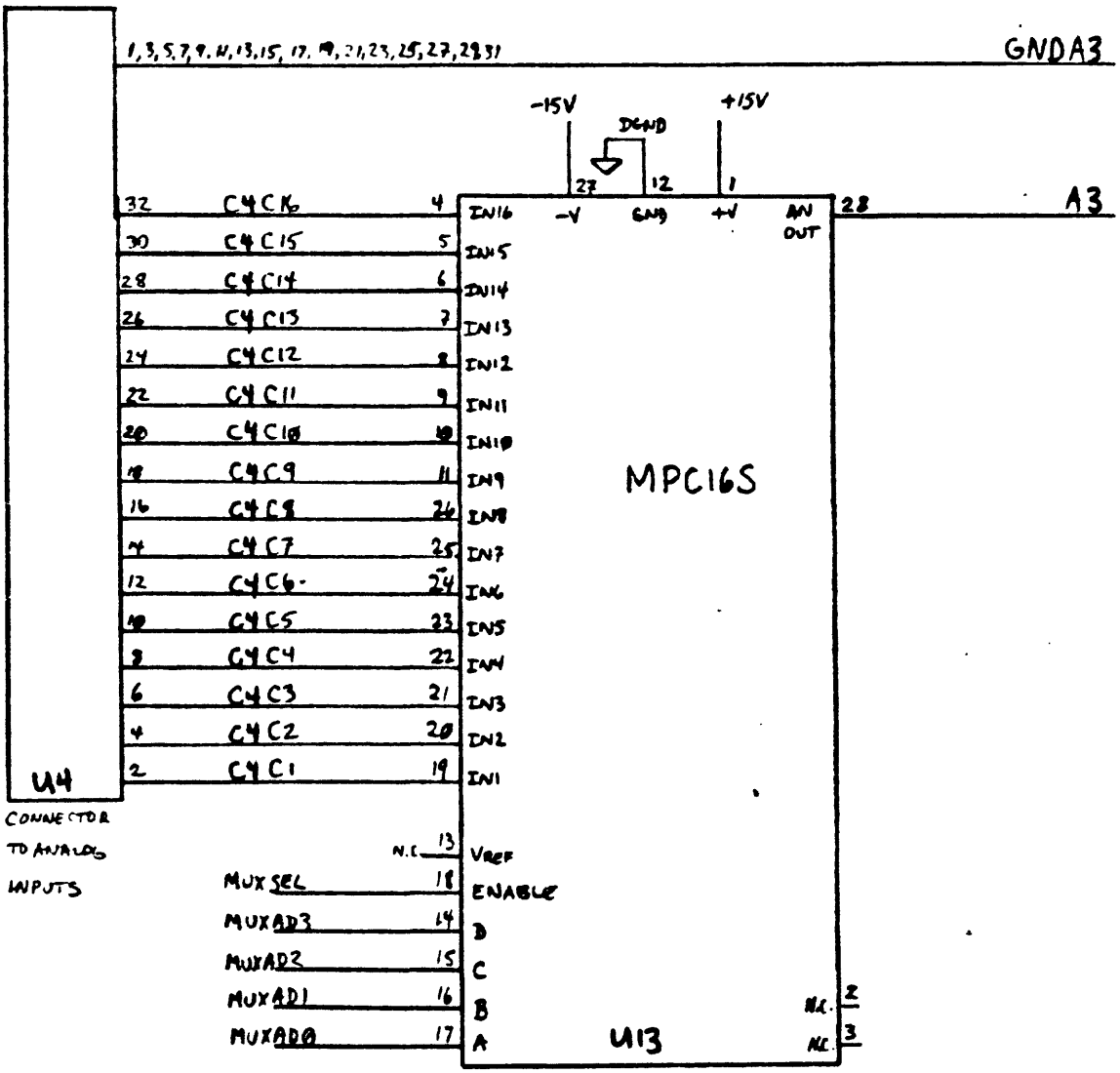


MULTIPLEXER BOARD	PAGE 3 OF 9
MULTIPLEXER 1 AND CONNECTOR	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 9404(



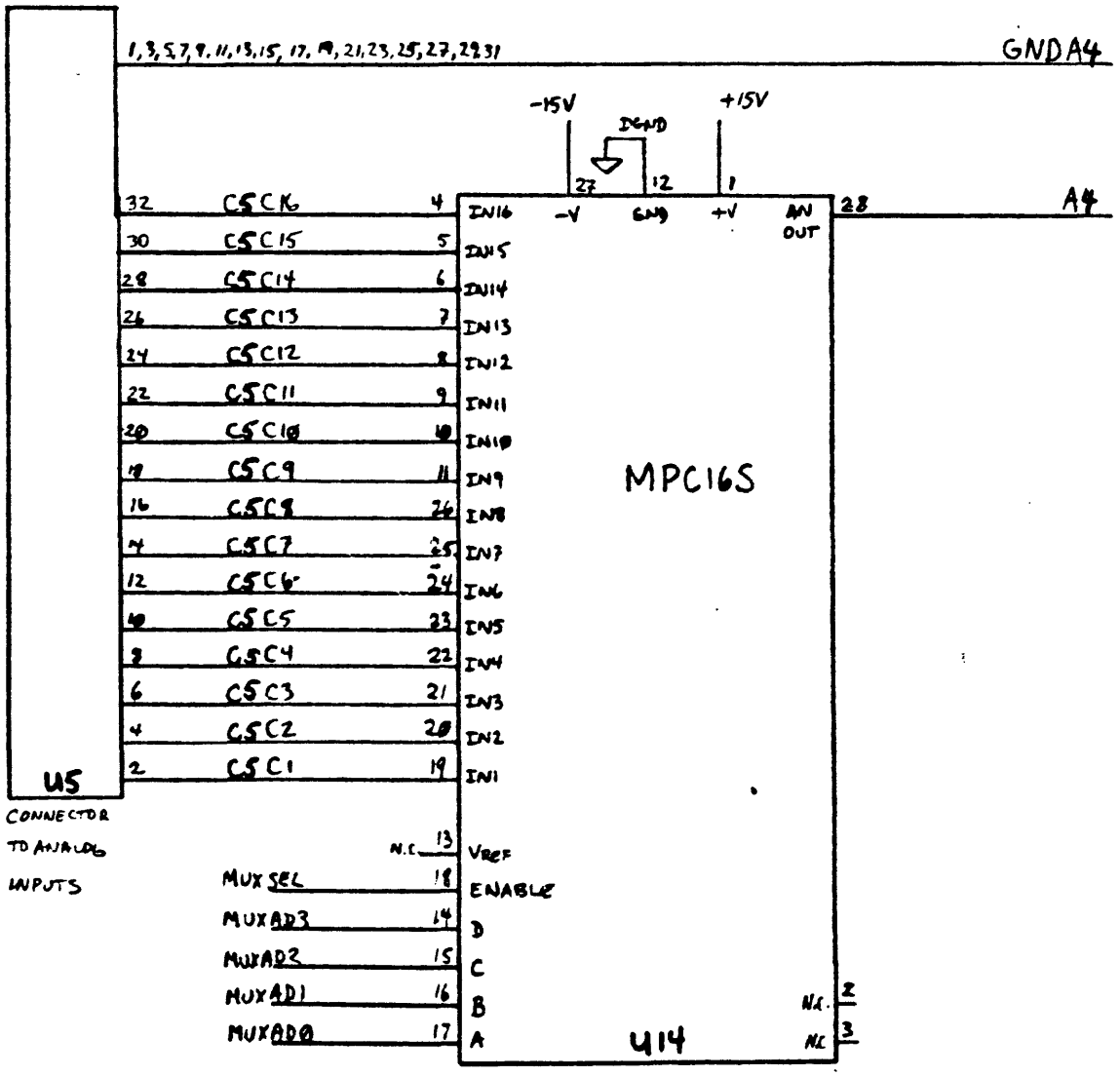
MULTIPLEXER BOARD	PAGE 4 OF 9
MULTIPLEXER 2 AND CONNECTOR	



MULTIPLEXER BOARD PAGE 5 OF 9

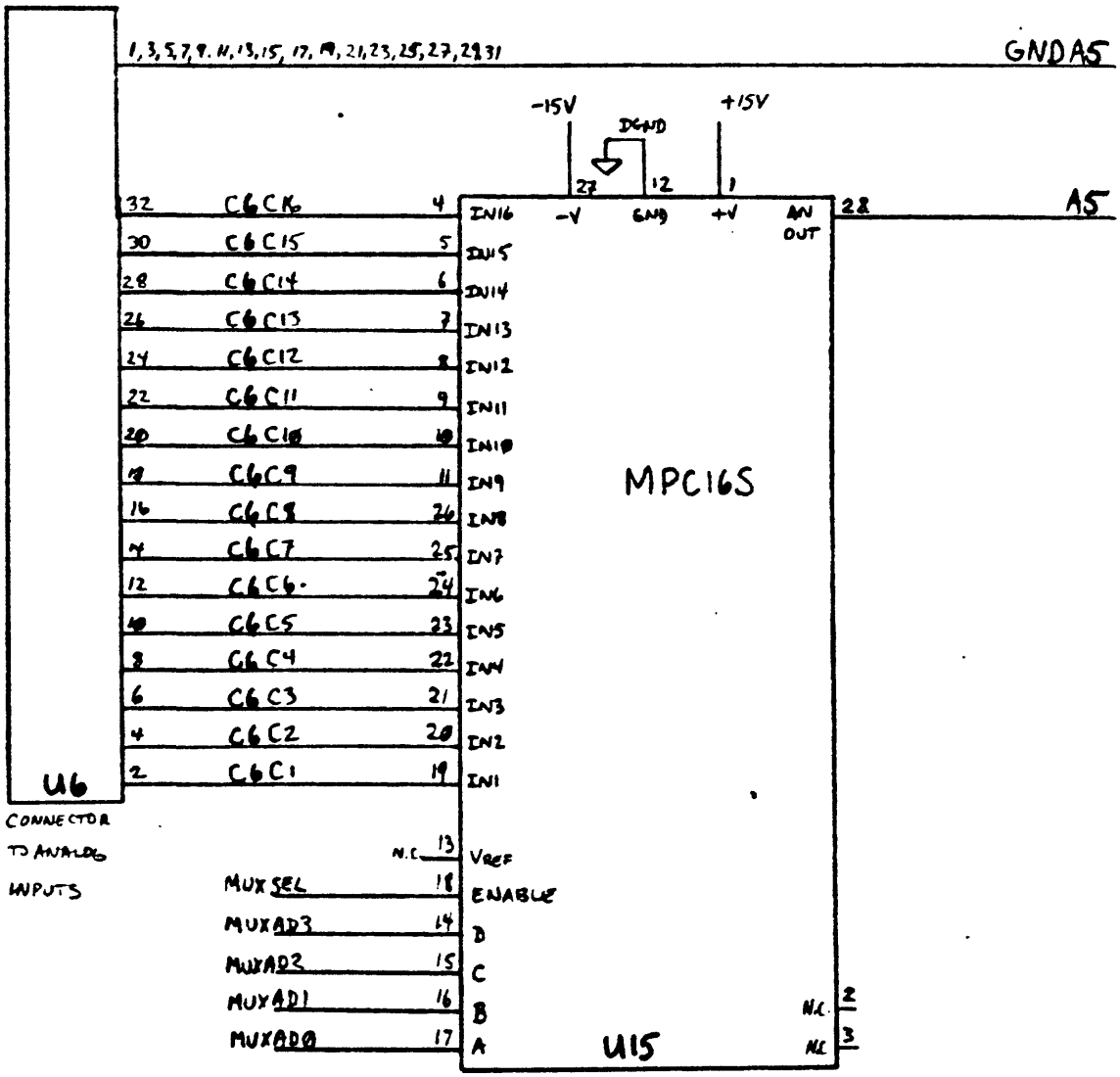
MULTIPLEXER 3 AND CONNECTOR

CUTLER DIGITAL DESIG
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 940



MULTIPLEXER BOARD	PAGE 6 OF 9
MULTIPLEXER 4 AND CONNECTOR	

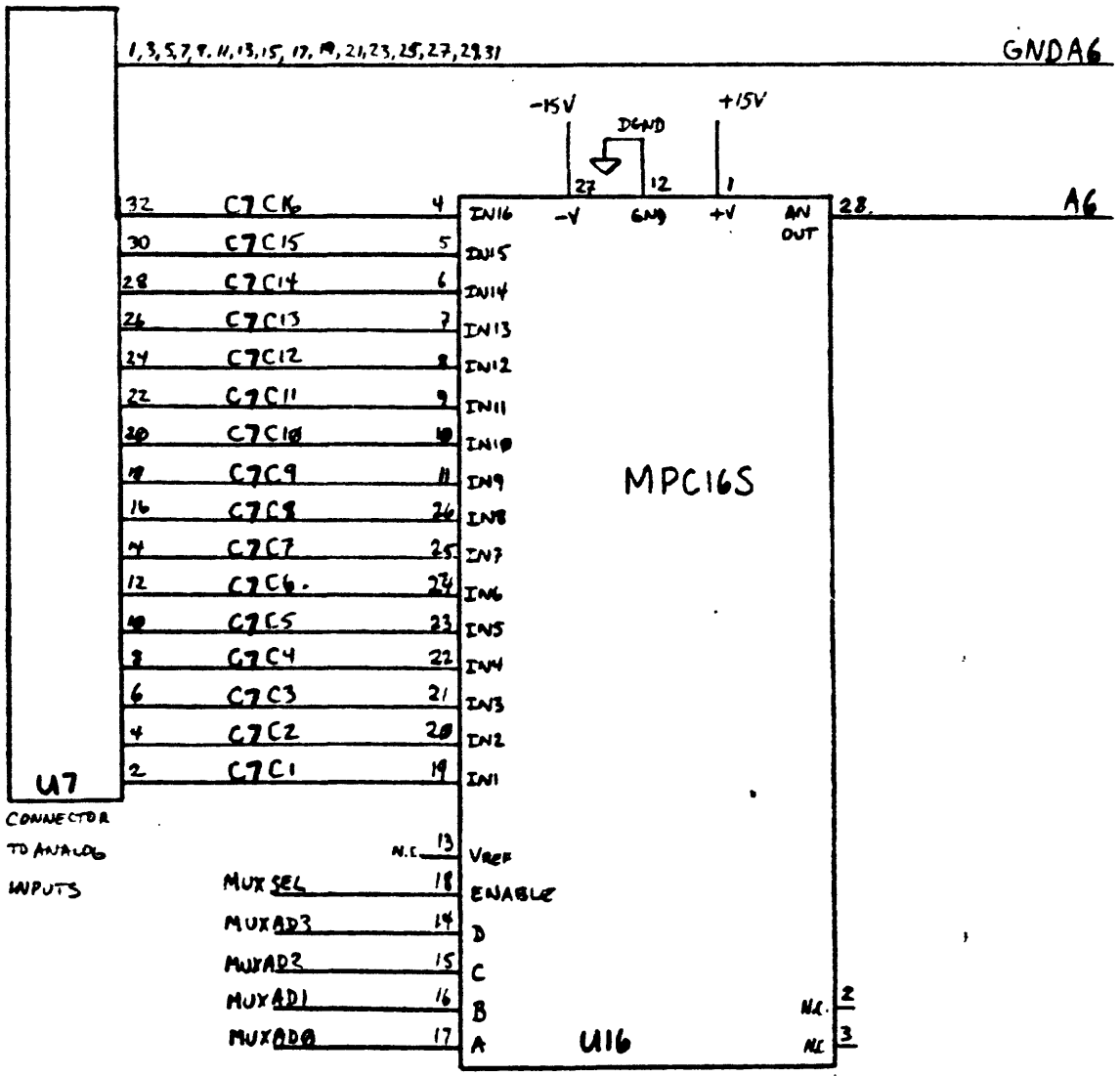
CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



MULTIPLEXER BOARD PAGE 7 OF 9

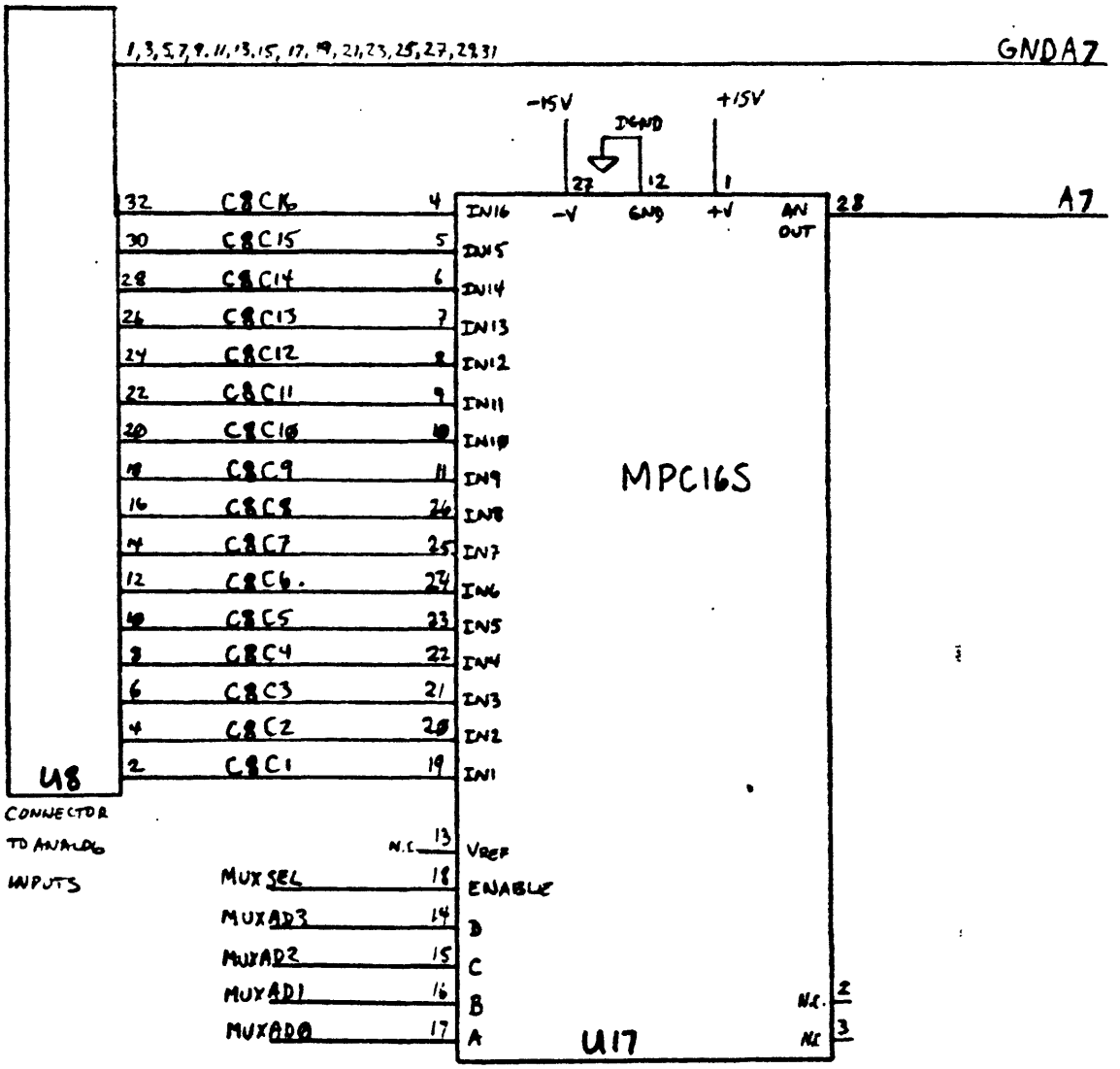
MULTIPLEXER 5 AND CONNECTOR

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



MULTIPLEXER BOARD	PAGE 8 OF 9
MULTIPLEXER 6 AND CONNECTOR	

CUTLER DIGITAL DESIGN
2215 SUN MOR AVE.
MOUNTAIN VIEW, CALIF. 94040



MULTIPLEXER BOARD	PAGE 9 OF 9
MULTIPLEXER 7 AND CONNECTOR	

```
          A/D ENGINE DRIVING SOFTWARE - FIRST IMPLEMENTATION
          --------------------------------------------------

                    INPUT  FILENAME :  TEST.ASM
                    OUTPUT FILENAME :  TEST.OBJ


 9                                          LIST    ON
10  0000:0000                              MACLIST OFF
11
12                                  ;       CUTLER DIGITAL DESIGN
13                                  ;       2215 SUN MOR AVENUE
14                                  ;       MOUNTAIN VIEW, CALIFORNIA 94040
15                                  ;       (415) 964-1481
16
17                                  ;       DEFINITIONS
18
19                                  ; RATE0 IS THE SYSTEM CLOCK DIVIDER USED TO PRODUCE 1 KHZ
20              0000 : 0000 12A5    RATE0:  EQU     4773
21
22                                  ; THE BUFFER POWER DETERMINES HOW LARGE OR SMALL A BUFFER IS USED FOR TRANSFERS
23                                  ;  FROM THE A/D TO MEMORY.  THE BUFFERS WILL BE A POWER OF TWO IN SIZE (ACTUAL
24                                  ;  BYTE COUNT), AND THE POWER SHOULD BE BETWEEN 8 AND 16
25              0000 : 0000 000C    BUFPWR: VAR     12
26              [01]                        IFTRUE  BUFPWR.LT.8
27                                  BUFPWR: VAR     8
28              [00]                        ENDIF
29              [01]                        IFTRUE  BUFPWR.GT.16
30                                  BUFPWR: VAR     16
31              [00]                        ENDIF
32
33              0000 : 0000 1000    BUFSIZE: EQU     2**BUFPWR
34              0000 : 0000 0004    SHIFTCNT: EQU    BUFPWR-8
35
36              [01]                        IFZ     STACKT.MOD.BUFSIZE
37                                  RESERVD: EQU     STACKT/BUFSIZE
38              [01]                        ELSE
39              0000 : 0000 0001    RESERVD: EQU     (STACKT/BUFSIZE)+1
40              [00]                        ENDIF
41
42              0000 : 0000 0FFF    XFERCNT: EQU     BUFSIZE-1
43              0000 : 0000 000F    XFERHI:  EQU     XFERCNT/100H
44              0000 : 0000 00FF    XFERLO:  EQU     XFERCNT.MOD.100H
45              0000 : 0000 009F    MAXBUF:  EQU     (A000H/(BUFSIZE/10H))-RESERVD
46
47              0000 : 0000 0000    ZERO:    EQU     00H
48              0000 : 0000 FFFF    MINUSONE: EQU    FFFFH
49              0000 : 0000 0001    ONE:     EQU     01H
50              0000 : 0000 0007    SEVEN:   EQU     07H
```

```
                TEST PROGRAM FOR A/D ENGINE


 51             0000 : 0000 0001      BIT0:    EQU     1H
 52             0000 : 0000 0002      BIT1:    EQU     2H
 53             0000 : 0000 0004      BIT2:    EQU     4H
 54             0000 : 0000 0008      BIT3:    EQU     8H
 55             0000 : 0000 0010      BIT4:    EQU     10H
 56             0000 : 0000 0020      BIT5:    EQU     20H
 57             0000 : 0000 0040      BIT6:    EQU     40H
 58             0000 : 0000 0080      BIT7:    EQU     80H
 59
 60             0000 : 0000 0000      DMA00:  EQU     00H
 61             0000 : 0000 0001      DMA01:  EQU     DMA00+1
 62             0000 : 0000 0002      DMA02:  EQU     DMA00+2
 63             0000 : 0000 0003      DMA03:  EQU     DMA00+3
 64             0000 : 0000 0004      DMA04:  EQU     DMA00+4
 65             0000 : 0000 0005      DMA05:  EQU     DMA00+5
 66             0000 : 0000 0006      DMA06:  EQU     DMA00+6
 67             0000 : 0000 0007      DMA07:  EQU     DMA00+7
 68             0000 : 0000 0008      DMA08:  EQU     DMA00+8
 69             0000 : 0000 0009      DMA09:  EQU     DMA00+9
 70             0000 : 0000 000A      DMA0A:  EQU     DMA00+10
 71             0000 : 0000 000B      DMA0B:  EQU     DMA00+11
 72             0000 : 0000 000C      DMA0C:  EQU     DMA00+12
 73             0000 : 0000 000D      DMA0D:  EQU     DMA00+13
 74             0000 : 0000 000E      DMA0E:  EQU     DMA00+14
 75             0000 : 0000 000F      DMA0F:  EQU     DMA00+15
 76
 77             0000 : 0000 0002      AMASKBIT: EQU    02H                    ATOD ON CHANNEL 1
 78
 79                                   ;  DMA MODE BYTE DEFINITION:
 80                                   ; BITS: 7 6 5 4   3 2 1 0
 81                                   ;       0 0                    DEMAND mode
 82                                   ;       0 1                    single mode
 83                                   ;       1 0                    BLOCK  mode
 84                                   ;       1 1                    CASCADE mode
 85                                   ;           0                  address increment
 86                                   ;           1                  address decrement
 87                                   ;             0                no auto init
 88                                   ;             1                AUTO INIT
 89                                   ;                 0 0          verify transfer
 90                                   ;                 0 1          write to memory
 91                                   ;                 1 0          read from memory
 92                                   ;                 1 1          illegal
 93                                   ;                     0 0      channel 0
 94                                   ;                     0 1      channel 1
 95                                   ;                     1 0      channel 2
 96                                   ;                     1 1      channel 3
 97             0000 : 0000 0000      DEMANDMD: EQU    00H
 98             0000 : 0000 0040      SINGLEMD: EQU    40H
 99             0000 : 0000 0080      BLOCKMD:  EQU     80H
100             0000 : 0000 00C0      CASCADMD: EQU    C0H
101             0000 : 0000 0000      ADDINC:   EQU     00H
```

```
                TEST PROGRAM FOR A/D ENGINE


102             0000 : 0000 0020        ADDDEC:    EQU     20H
103             0000 : 0000 0010        AUTOIN:    EQU     10H
104             0000 : 0000 0000        NOAUTOIN:  EQU     00H
105             0000 : 0000 0000        VERIFY:    EQU     00H
106             0000 : 0000 0004        TOMEM:     EQU     4H
107             0000 : 0000 0008        FROMEM:    EQU     8H
108             0000 : 0000 0000        CH0:       EQU     00H
109             0000 : 0000 0001        CH1:       EQU     01H
110             0000 : 0000 0002        CH2:       EQU     02H
111             0000 : 0000 0003        CH3:       EQU     03H
112             0000 : 0000 0000        SETON:     EQU     0H
113             0000 : 0000 0004        SETOFF:    EQU     4H
114
115             0000 : 0000 0058        CH0MODE:   EQU     SINGLEMD^ADDINC^AUTOIN^FROMEM^CH0
116             0000 : 0000 0041        CH1MODE:   EQU     SINGLEMD^ADDINC^NOAUTOIN^VERIFY^CH1
117             0000 : 0000 0042        CH2MODE:   EQU     SINGLEMD^ADDINC^NOAUTOIN^VERIFY^CH2
118             0000 : 0000 0043        CH3MODE:   EQU     SINGLEMD^ADDINC^NOAUTOIN^VERIFY^CH3
119
120             0000 : 0000 0083        DMASEG1:   EQU     83H
121             0000 : 0000 0081        DMASEG2:   EQU     81H
122             0000 : 0000 0082        DMASEG3:   EQU     82H
123
124             0000 : 0000 0001        INP:       EQU     CH1             ;A/D INPUT USES DMA CHANNEL ONE
125             0000 : 0000 0002        OUTP:      EQU     CH2             ;SCSI OUTPUT USES DMA CHANNEL TWO
126
127             [01]                            IFTRUE  OUTP.EQ.CH1
128                                     OUTSEG:    EQU     DMASEG1
129             [00]                            ENDIF
130
131             [01]                            IFTRUE  OUTP.EQ.CH2
132             0000 : 0000 0081        OUTSEG:    EQU     DMASEG2
133             [00]                            ENDIF
134
135             [01]                            IFTRUE  OUTP.EQ.CH3
136                                     OUTSEG:    EQU     DMASEG3
137             [00]                            ENDIF
138
139             0000 : 0000 0020        INTREG0:   EQU     20H
140             0000 : 0000 0021        INTREG1:   EQU     INTREG0+1
141
142             0000 : 0000 0040        TIM0:      EQU     40H
143             0000 : 0000 0041        TIM1:      EQU     41H
144             0000 : 0000 0042        TIM2:      EQU     42H
145             0000 : 0000 0043        TIMCON:    EQU     43H
146
147             0000 : 0000 0060        SPARA:     EQU     60H
148             0000 : 0000 0061        SPARB:     EQU     61H
149             0000 : 0000 0062        SPARC:     EQU     62H
150             0000 : 0000 0063        SPARCON:   EQU     63H
151
152             0000 : 0000 00A0        NMIREG:    EQU     0A0H
```

```
                TEST PROGRAM FOR A/D ENGINE


153
154             0000 : 0000 0300        IOBASE: EQU     300H
155             0000 : 0000 0300        PARA:   EQU     IOBASE
156             0000 : 0000 0301        PARB:   EQU     IOBASE+1
157             0000 : 0000 0302        PARC:   EQU     IOBASE+2
158             0000 : 0000 0303        PARCON: EQU     IOBASE+3
159             0000 : 0000 0304        CLK0:   EQU     IOBASE+4
160             0000 : 0000 0305        CLK1:   EQU     IOBASE+5
161             0000 : 0000 0306        CLK2:   EQU     IOBASE+6
162             0000 : 0000 0307        CLKCON: EQU     IOBASE+7
163             0000 : 0000 0308        FIFO:   EQU     IOBASE+8
164
165                                     ;       OUTPUT  76543210 TO PARALLEL CONTROL BYTE
166                                     ;               1       SET PARALLEL MODE
167                                     ;               0       BIT CHANGE FOR PORT C
168                                     ;                00      PORT A AND UPPER C MODE ZERO (BASIC I/O)
169                                     ;                01      PORT A AND UPPER C MODE ONE (STROBED I/O)
170                                     ;                10      PORT A AND UPPER C MODE TWO (BIDIR I/O)
171                                     ;                11      PORT A AND UPPER C MODE TWO (BIDIR I/O)
172                                     ;                  1      PORT A - INPUT
173                                     ;                  0      PORT A - OUTPUT
174                                     ;                   1      PORT C(UPPER) - INPUT
175                                     ;                   0      PORT C(UPPER) - OUTPUT
176                                     ;                    0      PORT B AND LOWER C MODE ZERO (BASIC I/O)
177                                     ;                    1      PORT B AND LOWER C MODE ONE (STROBED I/O)
178                                     ;                     1      PORT B - INPUT
179                                     ;                     0      PORT B - OUTPUT
180                                     ;                      1 PORT C(LOWER) - INPUT
181                                     ;                      0 PORT C(LOWER) - OUTPUT
182             0000 : 0000 0080        SETPARMODE:     EQU     BIT7
183             0000 : 0000 0000        BITWIDC:        EQU     ZERO
184             0000 : 0000 0000        ABASIC:         EQU     ZERO
185             0000 : 0000 0020        ASTROBE:        EQU     BIT5
186             0000 : 0000 0040        ABIDIR:         EQU     BIT6
187             0000 : 0000 0010        AINPUT:         EQU     BIT4
188             0000 : 0000 0000        AOUTPUT:        EQU     ZERO
189             0000 : 0000 0008        CUPIN:          EQU     BIT3
190             0000 : 0000 0000        CUPOUT:         EQU     ZERO
191             0000 : 0000 0000        BBASIC:         EQU     ZERO
192             0000 : 0000 0004        BSTROBE:        EQU     BIT2
193             0000 : 0000 0002        BINPUT:         EQU     BIT1
194             0000 : 0000 0000        BOUTPUT:        EQU     ZERO
195             0000 : 0000 0001        CDOWNIN:        EQU     BIT0
196             0000 : 0000 0000        CDOWNOUT:       EQU     ZERO
197
198                                     ; SCSI DEFINITIONS
199             0000 : 0000 030C        SCSI0:  EQU     30CH
200             0000 : 0000 030D        SCSI1:  EQU     SCSI0+1
201
202             0000 : 0000 0001        MJV:    EQU     01H             ;SOFTWARE MAJOR VERSION
203             0000 : 0000 0000        MNV:    EQU     00H             ;SOFTWARE MINOR VERSION
```

PAGE 5

TEST PROGRAM FOR A/D ENGINE

```
204          0000 : 0000 0040      BUFSZ:  EQU     64              ;BUFFER SIZE FOR MESSAGES
205          0000 : 0000 0000      GOOD:   EQU     0               ; COMMAND COMPLETE MESSAGE BYTE
206          0000 : 0000 0004      RELEASE: EQU     04H             ; DISCONNECTING MESSAGE BYTE
207          0000 : 0000 00FF      BAD     EQU     0FFH            ; COMMAND COMPLETE WITH ERROR STATUS BYTE
208          0000 : 0000 0002      CHKSTAT EQU     02H             ; CHECK STATUS VALUE
209          0000 : 0000 0080      INTBIT  EQU     80H
210
211          0000 : 0000 0000      NOSENSE EQU     00H             ;SENSEKEY VALUE FOR NO PROBLEM
212          0000 : 0000 0004      HARDERR EQU     04H             ;SENSEKEY VALUE FOR BUFFER OVERFLOW
213          0000 : 0000 0005      BADCOMM EQU     05H             ;SENSEKEY VALUE FOR BAD COMMAND
214
215                                                               ; WW33C93 COMMANDS:
216          0000 : 0000 0000      RESET   EQU     00H             ;       RESET
217          0000 : 0000 0001      ABORT   EQU     01H             ;       ABORT
218          0000 : 0000 0004      DISCON  EQU     04H             ;       DISCONNECT
219          0000 : 0000 0005      RESEL   EQU     05H             ;       RESELECT
220          0000 : 0000 000B      RESELSEN EQU    0BH             ;       RESELECT AND SEND
221          0000 : 0000 000C      WAITSEL EQU     0CH             ;       WAIT FOR SELECT
222          0000 : 0000 0014      SENDSTAT EQU    14H             ;       SEND STATUS
223          0000 : 0000 0015      SENDDATA EQU    15H             ;       SEND DATA
224          0000 : 0000 0016      SENDMESS EQU    16H             ;       SEND MESSAGE
225
226                                                               ; SCSI COMMANDS ISSUED BY INITIATOR
227          0000 : 0000 0000      TESTRDY EQU     00H             ;       TEST UNIT READY
228          0000 : 0000 0001      REZERO  EQU     01H             ;       REZERO UNIT
229          0000 : 0000 0004      SCSIDISCON EQU  04H             ;       DISCONNECT
230          0000 : 0000 0003      REQSENSE EQU    03H             ;       REQUEST SENSE
231          0000 : 0000 0008      RECEIVE EQU     08H             ;       RECEIVE
232          0000 : 0000 000A      SEND    EQU     0AH             ;       SEND
233          0000 : 0000 0012      INQUIRY EQU     12H             ;       INQUIRY
234          0000 : 0000 001A      SENDDIAG EQU    1AH             ;       SEND DIAGNOSTIC
235
236                                        PAGE
```

```
                        TEST PROGRAM FOR A/D ENGINE


                                                                    ; WD33C93 REGISTERS
0000 : 0000 0000        OWNID    EQU     00H                        ;       OWN ID REGISTER
0000 : 0000 0001        CONTROL  EQU     01H                        ;       CONTROL REGISTER
0000 : 0000 0002        TIMEOUT  EQU     02H                        ;       TIMEOUT REGISTER
0000 : 0000 0003        CDB1     EQU     03H                        ;       COMMAND DATA BYTE 1
0000 : 0000 0004        CDB2     EQU     04H                        ;       COMMAND DATA BYTE 2
0000 : 0000 0005        CDB3     EQU     05H                        ;       COMMAND DATA BYTE 3
0000 : 0000 0006        CDB4     EQU     06H                        ;       COMMAND DATA BYTE 4
0000 : 0000 0007        CDB5     EQU     07H                        ;       COMMAND DATA BYTE 5
0000 : 0000 0008        CDB6     EQU     08H                        ;       COMMAND DATA BYTE 6
0000 : 0000 000F        TARLUN   EQU     0FH                        ;       TARGET LOGICAL UNIT NUMBER
0000 : 0000 0010        COMPHASE EQU     10H                        ;       COMMAND PHASE
0000 : 0000 0011        SYNCHTR  EQU     11H                        ;       SYNCHRONOUS TRANSFER
0000 : 0000 0012        TRANSHI  EQU     12H                        ;       HIGH BYTE OF TRANSFER COUNT
0000 : 0000 0013        TRANSMID EQU     13H                        ;       MIDDLE BYTE OF TRANSFER COUNT
0000 : 0000 0014        TRANSLO  EQU     14H                        ;       LOW BYTE OF TRANSFER COUNT
0000 : 0000 0015        DESTID   EQU     15H                        ;       DESTINATION ID REGISTER
0000 : 0000 0016        SOURCID  EQU     16H                        ;       SOURCE ID REGISTER
0000 : 0000 0017        SCSISTAT EQU     17H                        ;       SCSI STATUS REGISTER
0000 : 0000 0018        COMMAND  EQU     18H                        ;       COMMAND REGISTER
0000 : 0000 0019        DATAREG  EQU     19H                        ;       DATA REGISTER


FE00 : 0000 0000        PUR:     EQU     FE00:0000H
0000 : 0000 FE00        PURHI:   EQU     FE00H
0000 : 0000 0000        PURLO:   EQU     0000H
0000 : 0000 0000        IPUR:    EQU     0000:0000H
0000 : 0000 0000        IPURHI:  EQU     0000H
0000 : 0000 0000        IPURLO:  EQU     0000H


                                PAGE
```

```
TEST PROGRAM FOR A/D ENGINE
MACRO DEFINITIONS

                                SUBTITLE MACRO DEFINITIONS

                        ; usage OUTPUT  OUTPUT-PORT , VALUE
                        OUTPUT: .MACRO  ARG1,ARG2
                                MOV     DX,ARG1
                                MOV     AL,ARG2
                                OUT     DX,AL
                                .MACEND

                        ; USAGE      INPUT      INPUT-PORT
                        INPUT:  .MACRO  ARG1
                                MOV     DX,ARG1
                                IN      AL,DX
                                .MACEND

                        ;       USAGE   INTON
                        INTON:  .MACRO
                                STI
                                .MACEND
                        INTOFF: .MACRO
                                CLI
                                .MACEND

                        SETREGIS: .MACRO   ARG1,ARG2
                                MOV     AL,ARG1
                                MOV     AH,ARG2
                                CALL    SETREG
                                .MACEND

                        GETREGIS: .MACRO   ARG1
                                MOV     AL,ARG1
                                CALL    READREG
                                .MACEND

                        COMMANDSCSI:    .MACRO  ARG1
                                MOV     AL,COMMAND
                                MOV     AH,ARG1
                                MOV     LASTCOMM,AH
                                CALL    SETREG
                                CALL    WAITFINT
                                MOV     AL,SCSISTAT
                                CALL    READREG
                                MOV     LASTSTAT,AL
                                MOV     AH,ARG1
                                MOV     AL,10H
                                CALL    SHOWSCSI                ;DIAGNOSTIC
                                MOV     AL,LASTSTAT
                                .MACEND

                        ;       END OF MACRO DEFINITIONS
```

```
TEST PROGRAM FOR A/D ENGINE
MACRO DEFINITIONS

320
321
322                                 PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

                                        SUBTITLE INITIALIZE COMPUTER
0000:0000                               .CODE

FE00:0000                               .ORG    PUR

                                START:      ; BEGIN OF PROGRAM


                                ; SET UP SEGMENT POINTERS
FE00:0000                               INTOFF                          DISABLE INTERRUPTS

FE00:0001  B8 00 00                     MOV     AX,IPURHI               LOAD SEGMENT REGISTERS
FE00:0004  8E D8                        MOV     DS,AX
FE00:0006  8E D0                        MOV     SS,AX
FE00:0008  8E C0                        MOV     ES,AX

FE00:000A  BD 00 00                     MOV     BP,ZERO                 ZERO INDEX AND OFFSET REGISTERS
FE00:000D  BB 00 00                     MOV     BX,ZERO
FE00:0010  BF 00 00                     MOV     DI,ZERO
FE00:0013  BE 00 00                     MOV     SI,ZERO

FE00:0016  BC 00 08                     MOV     SP,OFFSET STACKT        LOAD STACK POINTER

                                ; DISABLE NON MASKABLE INTERRUPT
FE00:0019  B0 00                        MOV     AL,ZERO                 DISABLE NMI
FE00:001B  E6 A0                        OUT     NMIREG,AL

                                ; SET UP SYSTEM PARALLEL PORT
FE00:001D  B0 99                        MOV     AL,SETPARMODE^ABASIC^AINPUT^CUPIN^BBASIC^BOUTPUT^CDOWNIN
FE00:001F  E6 63                        OUT     SPARCON,AL
FE00:0021  B0 F0                        MOV     AL,0F0H                 SETUP OUTPUT PORT B
FE00:0023  E6 61                        OUT     SPARB,AL

                                ; SET UP REFRESH AND DMA OPERATION
FE00:0025  B0 04                        MOV     AL,04H                  DISABLE DMA CHIP
FE00:0027  E6 08                        OUT     DMA08,AL

FE00:0029  E6 0D                        OUT     DMA0D,AL                MASTER CLEAR DMA

FE00:002B  B0 FF                        MOV     AL,0FFH                 SET COUNT FOR 64K
FE00:002D  E6 01                        OUT     DMA01,AL
FE00:002F  50                           PUSH    AX                      (THIS COMMAND IS TO GIVED DMA TIME)
FE00:0030  E6 01                        OUT     DMA01,AL

FE00:0032  50                           PUSH    AX                      (THIS COMMAND IS TO GIVED DMA TIME)
FE00:0033  B0 00                        MOV     AL,00H                  SET UP ADDRESS 0000H
FE00:0035  E6 00                        OUT     DMA00,AL
FE00:0037  50                           PUSH    AX                      (THIS COMMAND IS TO GIVED DMA TIME)
FE00:0038  E6 00                        OUT     DMA00,AL
                                ;SET TIMER 1 MODE
FE00:003A  B0 54                        MOV     AL,54H                  TIMER1, LSB, MODE 2
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

376  FE00:003C  E6 43                    OUT     TIMCON,AL
377                                    ;SET DMA MODE
378  FE00:003E  B0 58                    MOV     AL,CH0MODE          DMA MODE: CH0,INC,SING,READ,AUTO
379  FE00:0040  E6 0B                    OUT     DMA0B,AL
380                                    ;ENABLE DMA
381  FE00:0042  B0 00                    MOV     AL,00H
382  FE00:0044  E6 08                    OUT     DMA08,AL
383  FE00:0046  50                       PUSH    AX                  (THIS COMMAND IS TO GIVED DMA TIME)
384  FE00:0047  E6 0A                    OUT     DMA0A,AL            REMOVE CHANNEL 0 MASK
385                                    ;SET TIMER COUNT
386  FE00:0049  B0 12                    MOV     AL,12H              18 CLOCK CYCLES 15.12 MICROS
387  FE00:004B  E6 41                    OUT     TIM1,AL
388                                    ;SET OTHER DMA CHANNEL MODES
389  FE00:004D  B0 41                    MOV     AL,CH1MODE          DMA1: SIN,INC,NOAUTO,VERI
390  FE00:004F  E6 0B                    OUT     DMA0B,AL
391  FE00:0051  B0 42                    MOV     AL,CH2MODE          DMA2: SIN,INC,NOAUTO,VERI
392  FE00:0053  E6 0B                    OUT     DMA0B,AL
393  FE00:0055  B0 43                    MOV     AL,CH3MODE          DMA3: SIN,INC,NOAUTO,VERI
394  FE00:0057  E6 0B                    OUT     DMA0B,AL
395
396                                    ; INITIALIZE INTERRUPT CONTROLLER
397  FE00:0059  B0 13                    MOV     AL,13H
398  FE00:005B  E6 20                    OUT     INTREG0,AL
399  FE00:005D  B0 08                    MOV     AL,8
400  FE00:005F  E6 21                    OUT     INTREG1,AL
401  FE00:0061  B0 09                    MOV     AL,9
402  FE00:0063  E6 21                    OUT     INTREG1,AL
403  FE00:0065  B0 FF                    MOV     AL,0FFH             MASK ALL INTS OFF
404  FE00:0067  E6 21                    OUT     INTREG1,AL
405
406                                    ; INITIALIZE INTERRUPT VECTORS
407
408                                    ; INITIALIZE THE STACK
409  FE00:0069  BC 00 08                 MOV     SP,OFFSET STACKT    LOAD STACK POINTER
410
411                                    ; INITIALIZE PROTOTYPE PARALLEL PORT
412
413  FE00:006C                           OUTPUT  PARCON,SETPARMODE^ABASIC^AINPUT^CUPOUT^BBASIC^BOUTPUT^CDOWNOUT
418  FE00:0072                           OUTPUT  PARC,C0H
423                                    ;     OUTPUT 11000000 TO PARALEL PORT C
424                                    ;            1       FIFO RESET
425                                    ;             1        SWEEP DISABLE
426                                    ;              0       CLOCK 1 DISABLE
427                                    ;               0       CLOCK 0 DISABLE
428                                    ;                0
429                                    ;                 0
430                                    ;                  0
431                                    ;                   0
432
433                                    ;       INPUT   PARA
434                                    ;
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

435                                     ;       X               FIFO HAS DATA
436                                     ;        X              SWEEP IS ON
437                                     ;         X             DATA RATE B2    000     RATE = 200HZ
438                                     ;          X            DATA RATE B1    001     RATE = 50HZ
439                                     ;           X           DATA RATE B0    OTHER   RATE = 100HZ
440                                     ;            X          SCSI UNIT B2
441                                     ;             X         SCSI UNIT B1
442                                     ;              X        SCSI UNIT B0
443  FE00:0078                                  INPUT   PARA
447  FE00:007C  50                              PUSH    AX
448  FE00:007D  24 07                           AND     AL,SEVEN  MASK OUT HIGH BITS
449  FE00:007F  A2 5D 04                        MOV     DEVID,AL
450  FE00:0082  58                              POP     AX
451  FE00:0083  24 38                           AND     AL,38H
452  FE00:0085  74 0C                           JZ      SET200
453  FE00:0087  24 30                           AND     AL,30H
454  FE00:0089  74 10                           JZ      SET50
455  FE00:008B  C6 06 5A 04 0A                  MOV     BYTE PTR RATE1,10
456  FE00:0090  E9 10 00                        JMP     STX
457  FE00:0093  C6 06 5A 04 05          SET200  MOV     BYTE PTR RATE1,5
458  FE00:0098  E9 08 00                        JMP     STX
459  FE00:009B  C6 06 5A 04 14          SET50   MOV     BYTE PTR RATE1,20
460  FE00:00A0  E9 00 00                        JMP     STX
461  FE00:00A3                          STX
462
463
464  FE00:00A3  E8 EF 05                        CALL    SERRESET                ;RESET SERIAL PORT
465
466  FE00:00A6                          RESETIT
467  FE00:00A6  BC 00 08                        MOV     SP,OFFSET STACKT        LOAD STACK POINTER
468
469  FE00:00A9                          SCSICODE
470  FE00:00A9  C6 06 51 04 00                  MOV     BYTE PTR SENSEKEY,NOSENSE       ;RESET SENSE KEY
471  FE00:00AE  C6 06 49 04 00                  MOV     BYTE PTR ABNORMAL,ZERO          ;RESET ABNORMAL FLAG
472  FE00:00B3  C6 06 1D 04 00                  MOV     BYTE PTR ERRTYPE,ZERO
473  FE00:00B8                                  SETREGIS        OWNID,DEVID
478  FE00:00C1                                  COMMANDSCSI     RESET
492  FE00:00E1  3C 00                           CMP     AL,ZERO
493  FE00:00E3  74 05                           JE      SCSIL1
494  FE00:00E5  C6 06 1D 04 FF                  MOV     BYTE PTR ERRTYPE,0FFH
495  FE00:00EA                          SCSIL1
496  FE00:00EA  E8 82 02                        CALL    RESATOD
497  FE00:00ED                          SCSILOOP
498  FE00:00ED  E8 3B 04                        CALL    SHOWBUFFER              ;SET DISPLAY TO SHOW BUFFER COUNT
499  FE00:00F0                                  COMMANDSCSI WAITSEL
513
514  FE00:0110  A2 1C 04                        MOV     ERRCLASS,AL
515  FE00:0113                                  GETREGIS        SOURCID                 ;FIND OUT WHO IS THE BOSS
519  FE00:0118  24 07                           AND     AL,SEVEN                        ;STRIP OFF NON ID BITS
520  FE00:011A  A2 4B 04                        MOV     HOSTID,AL                       ;AND SAVE
521
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

 522  FE00:011D                              GETREGIS        TARLUN
 526  FE00:0122  A2 4E 04                    MOV     IDENTIFY,AL
 527
 528  FE00:0125                              GETREGIS        COMPHASE
 532  FE00:012A  A2 1D 04                    MOV     ERRTYPE,AL
 533  FE00:012D  3C 36                       CMP     AL,036H
 534  FE00:012F  74 03                       JE      SCSIL2
 535  FE00:0131  E9 38 02                    JMP     SCSIRSAV
 536  FE00:0134  A0 1C 04            SCSIL2  MOV     AL,ERRCLASS
 537  FE00:0137  3C 13                       CMP     AL,013H
 538  FE00:0139  74 03                       JE      SCSIL3
 539  FE00:013B  E9 2E 02                    JMP     SCSIRSAV
 540  FE00:013E                      SCSIL3
 541
 542  FE00:013E                              GETREGIS        CDB1
 546  FE00:0143  A2 47 04                    MOV     COMM,AL         ;SAVE COMMAND
 547
 548                                 ;       MOV     AL,'C'
 549                                 ;       CALL    PUTCH
 550                                 ;       CALL    SPACE
 551                                 ;       MOV     AL,COMM
 552                                 ;       CALL    PRBYTE
 553                                 ;       CALL    CRLF
 554                                 ;
 555                                 ;       MOV     AL,COMM
 556                                 ; branch according to command received
 557  FE00:0146  3C 01                       CMP     AL,REZERO
 558  FE00:0148  75 03                       JNE     SCSIL4
 559  FE00:014A  E9 F6 01                    JMP     SCSIREZ
 560  FE00:014D  3C 00               SCSIL4  CMP     AL,TESTRDY
 561  FE00:014F  75 03                       JNE     SCSIL5
 562  FE00:0151  E9 06 02                    JMP     SCSITRDY
 563  FE00:0154  3C 03               SCSIL5  CMP     AL,REQSENSE
 564  FE00:0156  75 03                       JNE     SCSIL6
 565  FE00:0158  E9 7C 01                    JMP     SCSIREQS
 566  FE00:015B  3C 08               SCSIL6  CMP     AL,RECEIVE
 567  FE00:015D  75 03                       JNE     SCSIL7
 568  FE00:015F  E9 77 00                    JMP     SCSISEND
 569  FE00:0162  3C 12               SCSIL7  CMP     AL,INQUIRY
 570  FE00:0164  75 03                       JNE     SCSIL8
 571  FE00:0166  E9 12 00                    JMP     SCSINQ
 572  FE00:0169  3C 1A               SCSIL8  CMP     AL,SENDDIAG
 573  FE00:016B  75 06                       JNE     SCSIDFLT
 574  FE00:016D  E9 00 00                    JMP     SCSISNDI
 575
 576  FE00:0170                      SCSISNDI
 577  FE00:0170  E9 00 00                    JMP     SCSIDFLT
 578
 579  FE00:0173                      SCSIDFLT
 580                                 ;               NON STANDARD COMMAND RECEIVED
 581  FE00:0173  C6 06 51 04 05              MOV     BYTE PTR SENSEKEY,BADCOMM       ;ILLEGAL REQUEST
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

582  FE00:0178  E9 D3 01                    JMP     SCSIQEND
583
584                                         PAGE
```

44

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

585
586  FE00:017B                          SCSINQ
587                                     ;               INQUIRY COMMAND RECEIVED
588  FE00:017B                                  GETREGIS CDB5           ;GET LENGTH
592  FE00:0180  A2 4F 04                       MOV      LENGTH,AL      ;STORE IT
593  FE00:0183                                  SETREGIS TRANSLO,08H
598  FE00:018A                                  SETREGIS TRANSMID,00H
603  FE00:0191                                  SETREGIS TRANSHI,00H
608  FE00:0198                                  SETREGIS SYNCHTR,0H
613  FE00:019F                                  SETREGIS CONTROL,00H    ;NOT DMA, OR DBA
618  FE00:01A6                                  SETREGIS COMMAND,SENDDATA
623
624  FE00:01AD  B0 AD                           MOV      AL,0ADH
625  FE00:01AF  E8 F2 03                        CALL     PUTDATA        ;FIRST BYTE
626
627  FE00:01B2  B0 00                           MOV      AL,0
628  FE00:01B4  E8 ED 03                        CALL     PUTDATA        ;SECOND BYTE
629
630  FE00:01B7  B0 00                           MOV      AL,0H
631  FE00:01B9  E8 E8 03                        CALL     PUTDATA        ;THIRD BYTE
632
633  FE00:01BC  B0 00                           MOV      AL,0
634  FE00:01BE  E8 E3 03                        CALL     PUTDATA        ;FOURTH BYTE
635
636  FE00:01C1  B0 03                           MOV      AL,03H
637  FE00:01C3  E8 DE 03                        CALL     PUTDATA        ;FIFTH BYTE
638
639  FE00:01C6  A0 52 04                        MOV      AL,BUFCNT
640  FE00:01C9  E8 D8 03                        CALL     PUTDATA        ;SIXTH BYTE
641
642  FE00:01CC  B0 01                           MOV      AL,MJV
643  FE00:01CE  E8 D3 03                        CALL     PUTDATA        ;SEVENTH BYTE
644
645  FE00:01D1  B0 00                           MOV      AL,MNV
646  FE00:01D3  E8 CE 03                        CALL     PUTDATA        ;EIGTH BYTE
647
648  FE00:01D6  E9 81 01                        JMP      SCSIEND
649
650                                             PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

 651  FE00:01D9                          SCSISEND:
 652                                     ;       MOV     AL,"S"
 653                                     ;       CALL    PUTCH
 654                                     ;       MOV     AL,"E"
 655                                     ;       CALL    PUTCH
 656  FE00:01D9                                  GETREGIS  CDB4                  ;SET LENGTH
 660  FE00:01DE  A2 45 04                        MOV     SCLNHI,AL
 661  FE00:01E1                                  GETREGIS  CDB5
 665  FE00:01E6  A2 44 04                        MOV     SCLNLO,AL
 666  FE00:01E9  A1 52 04                SCSENDA MOV     AX,BUFCNT
 667  FE00:01EC  3D 00 00                        CMP     AX,ZERO
 668  FE00:01EF  75 44                           JNE     SCSISEND1
 669                                     ;       CALL    CHKATOD
 670                                     ;       JMP     SCSENDA
 671
 672
 673                                     ;   SEND DISCONNECT MESSAGE TO INITIATOR
 674
 675
 676                                     ;       MOV     AL,"1"
 677                                     ;       CALL    PUTCH
 678                                     ;
 679
 680  FE00:01F1  B0 04                           MOV     AL,RELEASE
 681  FE00:01F3  E8 81 03                        CALL    XMITMESSAGE
 682                                     ;   DISCONNECT FROM BUS
 683
 684                                     ;       MOV     AL,"2"
 685                                     ;       CALL    PUTCH
 686                                     ;
 687
 688  FE00:01F6  E8 D9 03                        CALL   DISCONNECT       ;DISCONNECT FROM SCSI
 689
 690                                     ;       MOV     AL,"3"
 691                                     ;       CALL    PUTCH
 692                                     ;
 693  FE00:01F9                                  SETREGIS DESTID,HOSTID
 698  FE00:0202                          SCSISEND2
 699  FE00:0202  A1 52 04                        MOV     AX,BUFCNT
 700  FE00:0205  3D 00 00                        CMP     AX,ZERO
 701  FE00:0208  75 06                           JNE     SCSISEND3
 702  FE00:020A  E8 BA 01                        CALL    CHKATOD
 703  FE00:020D  E9 F2 FF                        JMP     SCSISEND2
 704  FE00:0210                          SCSISEND3
 705
 706                                     ;       MOV     AL,"4"
 707                                     ;       CALL    PUTCH
 708                                     ;
 709                                     ;       SETREGIS        SOURCID,040H            ;ENABLE SELECTION AND RESEL
 710  FE00:0210                                  COMMANDSCSI     RESEL
 724
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

                                        ;       MOV     AL,"5"
                                        ;       CALL    PUTCH
                                        ;
                                        ;       SEND IDENTIFY MESSAGE
FE00:0230  B0 80                                MOV     AL,BIT7
FE00:0232  E8 42 03                             CALL    XMITMESSAGE

                                        ;       MOV     AL,"6"
                                        ;       CALL    PUTCH
                                        ;
FE00:0235                               SCSISEND1
FE00:0235  A1 52 04                             MOV     AX,BUFCNT               ;UPDATE BUFFER COUNT
FE00:0238  48                                   DEC     AX
FE00:0239  A3 52 04                             MOV     BUFCNT,AX

FE00:023C  A1 56 04                             MOV     AX,OUTPTR               ;SET ADDRESS
FE00:023F  E8 22 02                             CALL    FIX
FE00:0242  88 26 43 04                          MOV     SCADHI,AH
FE00:0246  A2 42 04                             MOV     SCADMID,AL
FE00:0249  C6 06 41 04 00                       MOV     BYTE PTR SCADLO,ZERO
FE00:024E  E8 1B 02                             CALL    UPDOUT                  ;UP DATE POINTER TO NEXT BUFFER TO BE
                                                                                ; OUTPUT

FE00:0251                                       SETREGIS        TRANSLO,SCLNLO
FE00:025A                                       SETREGIS        TRANSMID,SCLNHI
FE00:0263                                       SETREGIS        TRANSHI,ZERO
FE00:026A                                       SETREGIS        CONTROL,BIT7^BIT3 ;TURN ON DMA ACCESS FOR SCSI CONTR.
FE00:0271                                       SETREGIS        SYNCHTR,00H
                                        ;       SETREGIS        SYNCHTR,045H

                                        ; SET UP DMA
FE00:0278                                       INTOFF
FE00:0279                                       OUTPUT  DMA0A,SETOFF^OUTP       ;MASK OUTPUT CHANNEL
FE00:027F                                       OUTPUT  DMA0B,SINGLEMD^ADDINC^NOAUTOIN^FROMEM^OUTP
                                                                                ;SET DMA CHANNEL MODE
FE00:0285                                       OUTPUT  DMA0C,00H               ;CLEAR BYTE FLIPFLOP
FE00:028B                                       OUTPUT  DMA04,SCADLO            ;SEND LOW ADDRESS
FE00:0292                                       OUTPUT  DMA04,SCADMID           ;SEND HIGH ADDRESS
FE00:0299  A1 44 04                             MOV     AX,SCLNLO               ;GET LENGTH
FE00:029C  48                                   DEC     AX                      ;DECREMENT THE COUNT
FE00:029D  50                                   PUSH    AX                      ;AND SAVE
FE00:029E                                       OUTPUT  DMA05,AL                ;SPECIFY LOW COUNT
FE00:02A4  58                                   POP     AX
FE00:02A5                                       OUTPUT  DMA05,AH                ;SEND HIGH COUNT
                                        ;       INPUT   DMA05
                                        ;       PUSH    AX
                                        ;       INPUT   DMA05
                                        ;       CALL    PRBYTE
                                        ;       POP     AX
                                        ;       CALL    PRBYTE                  ;PRINT COUNT - 1
FE00:02AB                                       OUTPUT  OUTSEG,SCADHI           ;SEND 64K SEGMENT NIBBLE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

                                        ; ENABLE DMA
FE00:02B2                                       OUTPUT  DMA0A,SETON^OUTP        ;REMOVE MASK
FE00:02B8                                       INTON
                                        ; SEND THE COMMAND, TO START THE TRANSFER
FE00:02B9                                       SETREGIS        COMMAND,SENDDATA
FE00:02C0  88 26 48 04                          MOV     LASTCOMM,AH
                                        ; WAIT FOR PROCESS DONE
FE00:02C4  E8 23 02                             CALL    WAITFINT
                                        ;       MOV     AL,SCADHI
                                        ;       CALL    PRBYTE
                                        ;       MOV     AL,SCADMID
                                        ;       CALL    PRBYTE
                                        ;       MOV     AL,SCADLO
                                        ;       CALL    PRBYTE
                                        ;       CALL    CRLF
                                        ;       CALL    SHOWDATA
                                        ;       FINISH OPERATION IF NOT DONE
                                        ;       CALL    AUXSTAT                 ;GET AUX STAT REG
                                        ;       AND     AL,BIT5^BIT4
                                        ;       JE      SS1
                                        ;       SETREGIS COMMAND,ABORT
                                        ;       MOV     AL,'A'
                                        ;       CALL    PUTCH
                                        ;       MOV     AL,'B'
                                        ;       CALL    PUTCH
                                        ;       CALL    SHOWAUX
                                        ;       CALL    WAITFREE

FE00:02C7                               SS1
                                        ; READ SCSI STATUS
FE00:02C7  E8 FF 02                             CALL    GETSTAT
                                        ;       GETREGIS        SCSISTAT
                                        ;       MOV     LASTSTAT,AL
                                        ; SHOW STATUS BEFORE ISSUING COMMAND
FE00:02CA  8A E0                                MOV     AH,AL
FE00:02CC  B0 40                                MOV     AL,040H
FE00:02CE  E8 36 04                             CALL    SHOWSCSI
FE00:02D1  A0 46 04                             MOV     AL,LASTSTAT

FE00:02D4  E9 77 00                             JMP     SCSIQEND

                                                PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

 882  FE00:02D7                          SCSIREQS
 883                                     ;               REQUEST SENSE
 884  FE00:02D7  C6 06 4F 04 0A                 MOV     BYTE PTR LENGTH,0AH ; SEND BACK ONLY 8 BYTES
 885  FE00:02DC                          REQ5
 886  FE00:02DC                                  SETREGIS TRANSLO,LENGTH
 891  FE00:02E5                                  SETREGIS TRANSMID,00H
 896  FE00:02EC                                  SETREGIS TRANSHI,00H
 901
 902                                     ;       MOV     AL,LENGTH
 903                                     ;       CALL    PRBYTE
 904                                     ;       CALL    CRLF
 905
 906  FE00:02F3                                  SETREGIS SYNCHTR,00H    ;ASYNCH
 911  FE00:02FA                                  SETREGIS CONTROL,00H    ;NOT DMA, OR DBA
 916  FE00:0301                                  SETREGIS COMMAND,SENDDATA
 921
 922  FE00:0308  B0 70                           MOV     AL,070H
 923  FE00:030A  E8 97 02                        CALL    PUTDATA         ;FIRST BYTE
 924                                     ;       MOV     AL,070H
 925                                     ;       CALL    PRBYTE
 926                                     ;       CALL    SPACE
 927
 928  FE00:030D  B0 00                           MOV     AL,0
 929  FE00:030F  E8 92 02                        CALL    PUTDATA         ;SECOND BYTE
 930                                     ;       MOV     AL,0
 931                                     ;       CALL    PRBYTE
 932                                     ;       CALL    SPACE
 933
 934  FE00:0312  A0 51 04                        MOV     AL,SENSEKEY
 935  FE00:0315  E8 8C 02                        CALL    PUTDATA         ;THIRD BYTE
 936                                     ;       MOV     AL,SENSEKEY
 937                                     ;       CALL    PRBYTE
 938                                     ;       CALL    SPACE
 939  FE00:0318  C6 06 51 04 00                 MOV     BYTE PTR SENSEKEY,NOSENSE
 940
 941  FE00:031D  B0 00                           MOV     AL,0
 942  FE00:031F  E8 82 02                        CALL    PUTDATA         ;FOURTH BYTE
 943                                     ;       MOV     AL,0
 944                                     ;       CALL    PRBYTE
 945                                     ;       CALL    SPACE
 946
 947  FE00:0322  B0 00                           MOV     AL,0H
 948  FE00:0324  E8 7D 02                        CALL    PUTDATA         ;FIFTH BYTE
 949                                     ;       MOV     AL,0H
 950                                     ;       CALL    PRBYTE
 951                                     ;       CALL    SPACE
 952
 953  FE00:0327  B0 00                           MOV     AL,0
 954  FE00:0329  E8 78 02                        CALL    PUTDATA         ;SIXTH BYTE
 955                                     ;       MOV     AL,0
 956                                     ;       CALL    PRBYTE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

 957                                    ;       CALL    SPACE
 958
 959  FE00:032C  B0 00                          MOV     AL,0H
 960  FE00:032E  E8 73 02                       CALL    PUTDATA         ;SEVENTH BYTE
 961                                    ;       MOV     AL,0H
 962                                    ;       CALL    PRBYTE
 963                                    ;       CALL    SPACE
 964
 965  FE00:0331  B0 02                          MOV     AL,2
 966  FE00:0333  E8 6E 02                       CALL    PUTDATA         ;EIGHTH BYTE
 967                                    ;       CALL    PRBYTE
 968  FE00:0336                         REQX
 969  FE00:0336  B0 A5                          MOV     AL,0A5H
 970  FE00:0338  E8 69 02                       CALL    PUTDATA         ;NINTH BYTE
 971                                    ;       MOV     AL,2H
 972                                    ;       CALL    PRBYTE
 973                                    ;       CALL    SPACE
 974
 975  FE00:033B  B0 5A                          MOV     AL,05AH
 976  FE00:033D  E8 64 02                       CALL    PUTDATA         ;TENTH BYTE
 977                                    ;       MOV     AL,2H
 978                                    ;       CALL    PRBYTE
 979                                    ;       CALL    SPACE
 980
 981                                    ;       CALL    CRLF
 982
 983  FE00:0340  E9 17 00                       JMP     SCSIEND
 984
 985
 986
 987                                            PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                INITIALIZE COMPUTER

 988  FE00:0343                      SCSIREZ
 989                                 ;       MOV     AL,1
 990                                 ;       CALL    FREEZE
 991                                 ;               REZERO UNIT
 992  FE00:0343  E8 29 00                    CALL    RESATOD
 993  FE00:0346  C6 06 51 04 00              MOV     BYTE PTR SENSEKEY,NOSENSE    ;RESET SENSE KEY
 994  FE00:034B  E9 0C 00                    JMP     SCSITRDY
 995
 996
 997  FE00:034E                      SCSIQEND
 998  FE00:034E  A0 51 04                    MOV     AL,SENSEKEY
 999  FE00:0351  3C 00                       CMP     AL,NOSENSE
1000  FE00:0353  74 05                       JE      SCSIEND1
1001                                ;  SEND REQUEST TO CHECK STATUS
1002  FE00:0355  B0 02                       MOV     AL,CHKSTAT
1003  FE00:0357  E9 02 00                    JMP     SCSIEND2
1004  FE00:035A                      SCSITRDY
1005                                ;               TEST UNIT READY
1006  FE00:035A                      SCSIEND:
1007  FE00:035A                      SCSIEND1:
1008                                ;  SEND STATUS = GOOD
1009  FE00:035A  B0 00                       MOV     AL,GOOD
1010  FE00:035C                      SCSIEND2:
1011  FE00:035C  E8 EB 01                    CALL    XMITSTATUS
1012  FE00:035F  B0 0E                       MOV     AL,0EH
1013                                ;       CALL    DISPAUX
1014                                ;       SEND END MESSAGE = 0
1015  FE00:0361  B0 00                       MOV     AL,GOOD                 ;SET END MESSAGE TO ZERO
1016  FE00:0363  E8 11 02                    CALL    XMITMESSAGE
1017                                ;       DISCONNECT
1018  FE00:0366  E8 69 02                    CALL    DISCONNECT
1019  FE00:0369  E9 81 FD                    JMP     SCSILOOP
1020
1021  FE00:036C                      SCSIRSAV
1022  FE00:036C  E9 04 FE                    JMP     SCSIDFLT
1023                                        PAGE
```

PAGE 21

TEST PROGRAM FOR A/D ENGINE
SUBROUTINES

```
1024                                    SUBTITLE SUBROUTINES
1025  FE00:036F                        RESATOD:
1026                                    ; SET UP A/D CLOCK RATE
1027  FE00:036F  E8 1D 00                      CALL    SETCLK
1028
1029                                    ;       WAIT UNTIL SOME DATA IS THERE
1030
1031  FE00:0372                         T0:     INPUT   PARA
1035  FE00:0376  24 80                          AND     AL,80H
1036  FE00:0378  74 F8                          JE      T0
1037                                    ; TURN OFF CLOCK
1038  FE00:037A                                 OUTPUT  PARC,00H
1043
1044                                    ; WAIT UNTIL SAMPLE IS DONE
1045  FE00:0380                         T1:     INPUT   PARA
1049  FE00:0384  24 40                          AND     AL,40H
1050  FE00:0386  75 F8                          JNE     T1
1051
1052                                    ; READ ALL DATA FROM FIFO
1053                                    ;T2:     INPUT   FIFO
1054                                    ;        INPUT   PARA
1055                                    ;        AND     AL,80H
1056                                    ;        JE      T2
1057
1058                                    ; RESET CLOCK AGAIN
1059  FE00:0388  E8 04 00                       CALL    SETCLK
1060
1061                                    ; START A-TO-D DMA OPERATION
1062  FE00:038B  E8 59 00                       CALL    INITATOD
1063
1064  FE00:038E  C3                             RET
1065
1066  FE00:038F                         SETCLK: OUTPUT  PARC,C0H
1071  FE00:0395                                 OUTPUT  CLKCON,36H
1076  FE00:039B                                 OUTPUT  CLKCON,76H
1081  FE00:03A1                                 OUTPUT  CLK0,RATE0.MOD.256
1086  FE00:03A7                                 OUTPUT  CLK0,RATE0/256
1091  FE00:03AD                                 OUTPUT  CLK1,RATE1
1096  FE00:03B4                                 OUTPUT  CLK1,00H
1101  FE00:03BA                                 OUTPUT  PARC,F0H
1106  FE00:03C0                                 OUTPUT  PARC,30H
1111  FE00:03C6  C3                             RET
1112
1113                                    PAGE
```

51

```
                TEST PROGRAM FOR A/D ENGINE
                SUBROUTINES

1114  FE00:03C7                         CHKATOD:
1115  FE00:03C7                                 INPUT   DMA08                   READ DMA STATUS REGISTER
1119  FE00:03CB  24 02                          AND     AL,OFFSET AMASKBIT
1120  FE00:03CD  74 03                          JZ      CHKATODX
1121
1122  FE00:03CF  E8 2E 00                       CALL    NEXTATOD
1123                                    ;       CALL    INCFLAG
1124  FE00:03D2                         CHKATODX:
1125  FE00:03D2  BA 01 03                       MOV     DX,PARB
1126  FE00:03D5  A0 5C 04                       MOV     AL,SSF
1127  FE00:03D8  3C 00                          CMP     AL,ZERO
1128  FE00:03DA  74 06                          JE      CHKATOD1
1129  FE00:03DC  A0 5B 04                       MOV     AL,FLAG
1130  FE00:03DF  E9 03 00                       JMP     CHKATOD2
1131  FE00:03E2  A1 52 04               CHKATOD1: MOV     AX,BUFCNT
1132                                    ;       ADD     AX,256-MAXBUF
1133                                    ;       MOV     AL,INPTR
1134  FE00:03E5  EE                     CHKATOD2: OUT     DX,AL
1135  FE00:03E6  C3                             RET
1136
1137  FE00:03E7                         INITATOD:
1138  FE00:03E7  C7 06 54 04 00 00              MOV     WORD PTR INPTR,ZERO     CLEAR INPUT POINTER
1139  FE00:03ED  E8 66 00                       CALL    IFIX                    FIX UP NEXT DMA POINTERS
1140  FE00:03F0  E8 0D 00                       CALL    NEXTATOD                INVOKE AND ATOD/DMA CYCLE
1141  FE00:03F3  C7 06 56 04 00 00              MOV     WORD PTR OUTPTR,ZERO    CLEAR OUTPUT POINTER
1142  FE00:03F9  C7 06 52 04 00 00              MOV     WORD PTR BUFCNT,ZERO    CLEAR BUFFER COUNT
1143  FE00:03FF  C3                             RET
1144
1145                                            PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SUBROUTINES

1146  FE00:0400                         NEXTATOD:
1147  FE00:0400                                 INTOFF
1150  FE00:0401  50                             PUSH    AX
1151  FE00:0402  52                             PUSH    DX
1152  FE00:0403  51                             PUSH    CX
1153  FE00:0404                                 OUTPUT  DMA0A,SETOFF^INP    ;MASK INPUT CHANNEL
1158  FE00:040A                                 OUTPUT  DMA0B,SINGLEMD^ADDINC^NOAUTOIN^TOMEM^INP
1163                                                                        ;SET A-D DMA MODE
1164  FE00:0410                                 OUTPUT  DMA0C,00H           ;CLEAR BYTE FLIPFLOP
1169  FE00:0416                                 OUTPUT  DMA02,00H           ;SEND LOW ADDRESS
1174  FE00:041C                                 OUTPUT  DMA02,HIADD         ;SEND HIGH ADDRESS
1179  FE00:0423                                 OUTPUT  DMA03,XFERLO        ;SPECIFY LOW COUNT
1184  FE00:0429                                 OUTPUT  DMA03,XFERHI        ;SEND HIGH COUNT
1189  FE00:042F                                 OUTPUT  DMASEG1,HINIB       ;SEND 64K SEGMENT NIBBLE
1194  FE00:0436                                 OUTPUT  DMA0A,SETON^INP     ;REMOVE MASK
1199  FE00:043C                                 INTON
1202  FE00:043D  E8 07 00                       CALL    UPDIN                     UPDATE INPUT POINTER
1203  FE00:0440  E8 39 00                       CALL    UPDBC                     UPDATE BUFFER COUNT
1204  FE00:0443  59                             POP     CX
1205  FE00:0444  5A                             POP     DX
1206  FE00:0445  58                             POP     AX
1207  FE00:0446  C3                             RET
1208
1209                                    ; UPDIN - UPDATES THE INPUT POINTER INPTR, WRAPPING IT AROUND
1210                                    ;                 WHEN NECESSARY
1211  FE00:0447  A1 54 04               UPDIN:  MOV     AX,INPTR
1212  FE00:044A  40                             INC     AX
1213  FE00:044B  3D 9F 00                       CMP     AX,OFFSET MAXBUF
1214  FE00:044E  75 03                          JNE     UPDINX
1215  FE00:0450  B8 00 00                       MOV     AX,0H
1216  FE00:0453  A3 54 04               UPDINX: MOV     INPTR,AX
1217  FE00:0456  A1 54 04               IFIX:   MOV     AX,INPTR
1218  FE00:0459  E8 08 00                       CALL    FIX
1219  FE00:045C  A2 58 04                       MOV     HIADD,AL
1220  FE00:045F  88 26 59 04                    MOV     HINIB,AH
1221  FE00:0463  C3                             RET
1222
1223  FE00:0464                         FIX
1224                                    ;       MOV     AX,0040H                ;DUMMY POINTER
1225                                    ;       RET
1226
1227  FE00:0464  05 01 00                       ADD     AX,OFFSET RESERVD
1228
1229             [01]                           IFTRUE  SHIFTCNT.GT.0
1230  FE00:0467  B1 04                          MOV     CL,SHIFTCNT
1231  FE00:0469  D3 E0                          SHL     AX,CL
1232             [00]                           ENDIF
1233
1234  FE00:046B  C3                             RET
1235
1236                                           PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SUBROUTINES

1237  FE00:046C A1 56 04            UPDOUT  MOV     AX,OUTPTR
1238  FE00:046F 40                          INC     AX
1239  FE00:0470 3D 9F 00                    CMP     AX,OFFSET MAXBUF
1240  FE00:0473 75 03                       JNE     UPDOUTX
1241  FE00:0475 B8 00 00                    MOV     AX,0H
1242  FE00:0478 A3 56 04            UPDOUTX MOV     OUTPTR,AX
1243  FE00:047B C3                          RET
1244
1245  FE00:047C A1 52 04            UPDBC:  MOV     AX,BUFCNT
1246  FE00:047F 40                          INC     AX
1247  FE00:0480 3D 9F 00                    CMP     AX,OFFSET MAXBUF
1248  FE00:0483 74 04                       JE      UPDBCX
1249  FE00:0485 A3 52 04                    MOV     BUFCNT,AX
1250  FE00:0488 C3                          RET
1251  FE00:0489 E8 E0 FF            UPDBCX: CALL    UPDOUT
1252  FE00:048C C6 06 51 04 04              MOV     BYTE PTR SENSEKEY,HARDERR
1253  FE00:0491 C3                          RET
1254
1255                                        PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1256                                            SUBTITLE SCSI SUBROUTINES
1257
1258                                    DELAY   ;       ASSUMES DELAY COUNT IS IN CX
1259    FE00:0492                       DELAY1
1260    FE00:0492   E2 FE                       LOOP    DELAY1
1261    FE00:0494   C3                          RET
1262                FE00 : 0000 0020    PAUSECONST EQU  20H
1263    FE00:0495                       DELAYP
1264    FE00:0495   51                          PUSH    CX
1265    FE00:0496   B9 20 00                    MOV     CX,PAUSECONST
1266    FE00:0499   E8 F6 FF                    CALL    DELAY
1267    FE00:049C   59                          POP     CX
1268    FE00:049D   C3                          RET
1269
1270                                    ;       AUXSTAT RETURNS THE VALUE OF THE AUXILIARY STATUS IN AL
1271                                    ;
1272    FE00:049E                       AUXSTAT
1273    FE00:049E   BA 0C 03                    MOV     DX,SCSI0
1274    FE00:04A1   EC                          IN      AL,DX
1275    FE00:04A2   C3                          RET
1276
1277                                    ;       READREG READS A SCSI REGISTER
1278                                    ;         THE REGISTER DESIGNATOR MUST BE STORED IN AL
1279                                    ;         USAGE:        MOV     AL,REGISTER_DESIGNATOR
1280                                    ;                       CALL    READREG
1281                                    ;
1282                                    ;         THE VALUE FROM THE REGISTER IS LEFT IN AL
1283                                    ;
1284    FE00:04A3                       READREG
1285    FE00:04A3   3C 17                       CMP     AL,SCSISTAT
1286    FE00:04A5   74 0B                       JE      RREG1
1287    FE00:04A7   BA 0C 03                    MOV     DX,SCSI0
1288    FE00:04AA   EE                          OUT     DX,AL
1289    FE00:04AB   BA 0D 03                    MOV     DX,SCSI1
1290
1291    FE00:04AE   EB 00                       JMP     SHORT $+2               WAIT FOR SLOW I/O
1292
1293    FE00:04B0   EC                          IN      AL,DX
1294    FE00:04B1   C3                          RET
1295
1296    FE00:04B2   BA 0C 03            RREG1   MOV     DX,SCSI0
1297    FE00:04B5   EE                          OUT     DX,AL
1298    FE00:04B6   BA 0D 03                    MOV     DX,SCSI1
1299
1300    FE00:04B9   EB 00                       JMP     SHORT $+2               WAIT FOR SLOW I/O
1301
1302    FE00:04BB   EC                          IN      AL,DX
1303    FE00:04BC   B9 07 00                    MOV     CX,SEVEN
1304    FE00:04BF   E8 D0 FF                    CALL    DELAY
1305    FE00:04C2   C3                          RET
1306
```

```
TEST PROGRAM FOR A/D ENGINE
SCSI SUBROUTINES

1307                                    PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1308                                    ;       SETREG READS A SCSI REGISTER
1309                                    ;         THE REGISTER DESIGNATOR MUST BE STORED IN AL
1310                                    ;         THE REGISTER VALUE MUST BE STORED IN AH
1311                                    ;         USAGE:          MOV     AL,REGISTER_DESIGNATOR
1312                                    ;                         MOV     AH,VALUE
1313                                    ;                         CALL    SETREG
1314                                    ;
1315                                    ;
1316  FE00:04C3                         SETREG
1317  FE00:04C3  BA 0C 03                       MOV     DX,SCSI0
1318  FE00:04C6  EE                             OUT     DX,AL
1319
1320  FE00:04C7  BA 0D 03                       MOV     DX,SCSI1
1321  FE00:04CA  8A C4                          MOV     AL,AH
1322  FE00:04CC  EB 00                          JMP     SHORT $+2               THIS IS A WAIT FOR SLOW I/OPORTS
1323                                    ;                                         AS SPECIFIED IN THE WD33C92 STARTER KIT
1324  FE00:04CE  EE                             OUT     DX,AL
1325  FE00:04CF  C3                             RET
1326
1327                                    ;       WAITFINT POLLS THE AUXILIARY STATUS REGISTER FOR AN INTERRUPT
1328                                    ;       CONDITION.  IT ALSO INVOKES CHKATOD
1329               FE00 : 0000 8000     WAITTIME EQU    8000H
1330  FE00:04D0                         WAITFINTA
1331  FE00:04D0  51                             PUSH    CX
1332  FE00:04D1  52                             PUSH    DX
1333  FE00:04D2  50                             PUSH    AX
1334  FE00:04D3  53                             PUSH    BX
1335                                    ;       MOV     AL,'W'
1336                                    ;       CALL    PUTCH
1337                                    ;       MOV     AL,'A'
1338                                    ;       CALL    PUTCH
1339                                    ;       CALL    CRLF
1340  FE00:04D4  B9 00 80                       MOV     CX,WAITTIME
1341  FE00:04D7                         WAITFINTB
1342  FE00:04D7  E8 C4 FF                       CALL    AUXSTAT
1343  FE00:04DA  24 80                          AND     AL,INTBIT
1344  FE00:04DC  75 20                          JNE     WAITX
1345                                    ;       MOV     AL,"A"
1346  FE00:04DE  E8 5D 00                       CALL    POLLING
1347  FE00:04E1  81 F9 00 80                    CMP     CX,WAITTIME
1348  FE00:04E5  75 F0                          JNE     WAITFINTB
1349                                    ;       MOV     AL,"?"
1350                                    ;       CALL    PUTCH
1351                                    ;       CALL    CRLF
1352  FE00:04E7  E9 14 00                       JMP     WAITX
1353  FE00:04EA                         WAITFINT
1354  FE00:04EA  51                             PUSH    CX
1355  FE00:04EB  52                             PUSH    DX
1356  FE00:04EC  50                             PUSH    AX
1357  FE00:04ED  53                             PUSH    BX
1358                                    ;       MOV     AL,'W'
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1359                                    ;       CALL    PUTCH
1360                                    ;       MOV     AL,'I'
1361                                    ;       CALL    PUTCH
1362                                    ;       CALL    CRLF
1363  FE00:04EE  B9 00 80                       MOV     CX,WAITTIME
1364  FE00:04F1                         WAITFINT1
1365  FE00:04F1  E8 AA FF                       CALL    AUXSTAT
1366  FE00:04F4  24 80                          AND     AL,INTBIT
1367  FE00:04F6  75 06                          JNE     WAITX
1368                                    ;       MOV     AL,"I"
1369  FE00:04F8  E8 43 00                       CALL    POLLING
1370  FE00:04FB  E9 F3 FF                       JMP     WAITFINT1
1371  FE00:04FE                         WAITX
1372  FE00:04FE  5B                              POP     BX
1373  FE00:04FF  58                              POP     AX
1374  FE00:0500  5A                              POP     DX
1375  FE00:0501  59                              POP     CX
1376  FE00:0502  C3                              RET
1377  FE00:0503                         WAITFREE
1378  FE00:0503  51                              PUSH    CX
1379  FE00:0504  52                              PUSH    DX
1380  FE00:0505  50                              PUSH    AX
1381  FE00:0506  53                              PUSH    BX
1382                                    ;       MOV     AL,'W'
1383                                    ;       CALL    PUTCH
1384                                    ;       MOV     AL,'F'
1385                                    ;       CALL    PUTCH
1386                                    ;       CALL    CRLF
1387  FE00:0507  B9 00 80                       MOV     CX,WAITTIME
1388  FE00:050A                         WAITFREE1
1389  FE00:050A  E8 91 FF                       CALL    AUXSTAT
1390  FE00:050D  24 30                          AND     AL,BIT5^BIT4    ;CHECK FOR BSY OR CIP
1391  FE00:050F  74 ED                          JE      WAITX
1392                                    ;       MOV     AL,"F"
1393  FE00:0511  E8 2A 00                       CALL    POLLING
1394  FE00:0514  E9 F3 FF                       JMP     WAITFREE1
1395
1396  FE00:0517                         DISPAUX:
1397  FE00:0517  A2 5B 04                       MOV     FLAG,AL
1398  FE00:051A  E8 81 FF                       CALL    AUXSTAT
1399  FE00:051D  0A 06 5B 04                    OR      AL,FLAG
1400  FE00:0521  A2 5B 04                       MOV     FLAG,AL
1401  FE00:0524  C3                              RET
1402  FE00:0525                         SHOWFLAG:
1403  FE00:0525  C6 06 5C 04 FF                 MOV     BYTE PTR SSF,0FFH
1404  FE00:052A  C3                              RET
1405  FE00:052B                         SHOWBUFFER:
1406  FE00:052B  C6 06 5C 04 00                 MOV     BYTE PTR SSF,ZERO
1407  FE00:0530  C3                              RET
1408  FE00:0531                         SHOWAUX:
1409  FE00:0531  E8 AC 01                       CALL    SPACE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1410  FE00:0534  E8 67 FF                    CALL    AUXSTAT
1411  FE00:0537  E8 AC 01                    CALL    PRBYTE
1412  FE00:053A  E8 98 01                    CALL    CRLF
1413  FE00:053D  C3                          RET
1414  FE00:053E                       POLLING:
1415                                  ;       PUSH    AX
1416  FE00:053E  E8 86 FE                    CALL    CHKATOD
1417  FE00:0541  E8 51 FF                    CALL    DELAYP          ;PAUSECONST DETERMINES THE LENGTH OF THIS DELAY
1418  FE00:0544  E2 03                        LOOP    POLLINGX
1419
1420                                  ;       POP     AX
1421                                  ;       PUSH    AX
1422                                  ;       CALL    PUTCH
1423                                  ;       CALL    SPACE
1424                                  ;DISABLE DMA OPERATION ON OUTP
1425                                  ;       OUTPUT  DMA0A,SETOFF^OUTP
1426                                  ;CLEAR FLIP FLOP
1427                                  ;       OUTPUT  DMA0C,00H
1428                                  ;READ LOW COUNT AND PUSH ONTO STACK
1429                                  ;       INPUT   DMA05
1430                                  ;       PUSH    AX
1431                                  ;READ HI BYTE AND PRINT
1432                                  ;       INPUT   DMA05
1433                                  ;       CALL    PRBYTE
1434                                  ;POP LOW BYTE AND PRINT
1435                                  ;       POP     AX
1436                                  ;       CALL    PRBYTE
1437                                  ;ENABLE DMA OPERATION ON OUTP
1438                                  ;       OUTPUT  DMA0A,SETON^OUTP
1439                                  ;       CALL    SHOWDATA
1440                                  ;       CALL    SHOWAUX
1441  FE00:0546  B9 00 80                    MOV     CX,WAITTIME
1442  FE00:0549                       POLLINGX:
1443                                  ;       POP     AX
1444  FE00:0549  C3                          RET
1445
1446                                          PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1447                                    ; XMITSTATUS EXPECTS STATUS IN REGISTER AL
1448
1449  FE00:054A                         XMITSTATUS:
1450  FE00:054A  A2 4C 04                      MOV     XSTATUS,AL              ;SAVE STATUS
1451  FE00:054D                                 GETREGIS        SCSISTAT        ;CLEAR ANY INTERRUPTS BY READING STAT
1455  FE00:0552                                 SETREGIS        CONTROL,00H     ;TURN OFF DMA, DBA
1460  FE00:0559                                 SETREGIS        COMMAND,SENDSTAT^BIT7
1465                                                                            ;ISSUE THE SEND_STAT COMMAND, SNG BYT
1466  FE00:0560  A0 4C 04                      MOV     AL,XSTATUS
1467  FE00:0563  E8 3E 00                      CALL    PUTDATA                 ;SEND XSTATUS TO DATA REGISTER WHEN RDY
1468
1469                                    ;XMIT1   CALL    AUXSTAT                 ;GET AUX STAT
1470                                    ;        AND     AL,01H                  ;LOOK AT DBA BIT
1471                                    ;        JE      XMIT1                   ;LOOP UNTIL READY
1472                                    ;        SETREGIS        DATAREG,XSTATUS ; WRITE STATUS BYTE
1473
1474  FE00:0566  E8 81 FF                      CALL    WAITFINT
1475  FE00:0569  E8 5D 00                      CALL    GETSTAT
1476                                    ;       MOV     AL,'S'
1477                                    ;       CALL    PUTCH
1478                                    ;       CALL    SPACE
1479  FE00:056C  8A 26 46 04                   MOV     AH,LASTSTAT
1480  FE00:0570  A0 4C 04                      MOV     AL,XSTATUS
1481  FE00:0573  E8 91 01                      CALL    SHOWSCSI
1482  FE00:0576  C3                            RET
1483
1484                                    ; XMITMESSAGE EXPECTS MESSAGE IN REGISTER AL
1485  FE00:0577                         XMITMESSAGE:
1486  FE00:0577  A2 4D 04                      MOV     XMESSAGE,AL             ;SAVE MESSAGE
1487  FE00:057A                                 GETREGIS        SCSISTAT        ;CLEAR ANY INTERRUPTS BY READING STAT
1491  FE00:057F                                 SETREGIS        CONTROL,00H     ;TURN OFF DMA, DBA
1496  FE00:0586                                 SETREGIS        COMMAND,SENDMESS^BIT7
1501                                                                            ;ISSUE THE SEND_MESS COMMAND, SNG BYT
1502  FE00:058D  A0 4D 04                      MOV     AL,XMESSAGE
1503  FE00:0590  E8 11 00                      CALL    PUTDATA
1504
1505                                    ;XMIT2   CALL    AUXSTAT                 ;GET AUX STAT
1506                                    ;        AND     AL,01H                  ;LOOK AT DBA BIT
1507                                    ;        JE      XMIT2                   ;LOOP UNTIL READY
1508                                    ;        SETREGIS        DATAREG,XMESSAGE ; WRITE STATUS BYTE
1509  FE00:0593  E8 54 FF                      CALL    WAITFINT
1510  FE00:0596  E8 30 00                      CALL    GETSTAT
1511                                    ;       MOV     AL,'M'
1512                                    ;       CALL    PUTCH
1513                                    ;       CALL    SPACE
1514  FE00:0599  8A 26 46 04                   MOV     AH,LASTSTAT
1515  FE00:059D  A0 4D 04                      MOV     AL,XMESSAGE
1516  FE00:05A0  E8 64 01                      CALL    SHOWSCSI
1517  FE00:05A3  C3                            RET
1518
1519  FE00:05A4                         PUTDATA:
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1520  FE00:05A4  A2 50 04                   MOV     DATATEMP,AL
1521  FE00:05A7  B9 00 80                   MOV     CX,WAITTIME
1522  FE00:05AA                     PUTDATA1
1523  FE00:05AA  E8 F1 FE                   CALL    AUXSTAT         ;GET AUXI STAT
1524  FE00:05AD  24 01                      AND     AL,01H          ;LOOK AT DBA BIT
1525  FE00:05AF  75 0B                      JNE     PUTDATAX
1526                                ;       MOV     AL,'P'
1527  FE00:05B1  E8 8A FF                   CALL    POLLING
1528  FE00:05B4  8B C1                      MOV     AX,CX
1529  FE00:05B6  3D 00 80                   CMP     AX,WAITTIME
1530                                ;       JNE     PUTDATA1
1531                                ;       SETREGIS        COMMAND,ABORT
1532                                ;       JMP     RESETIT
1533                                ;       SETREGIS        COMMAND,DISCON
1534                                ;       COMMANDSCSI     ABORT
1535                                ;       COMMANDSCSI     DISCON
1536                                ;       MOV     AL,7
1537                                ;       CALL    FREEZE
1538
1539  FE00:05B9  E9 EE FF                   JMP     PUTDATA1
1540  FE00:05BC                     PUTDATAX
1541  FE00:05BC                             SETREGIS        DATAREG,DATATEMP
1546  FE00:05C5  E8 CD FE                   CALL    DELAYP
1547  FE00:05C8  C3                         RET
1548
1549
1550  FE00:05C9                     GETSTAT GETREGIS        SCSISTAT
1554  FE00:05CE  A2 46 04                   MOV     LASTSTAT,AL
1555  FE00:05D1  C3                         RET
1556
1557  FE00:05D2                     DISCONNECT
1558  FE00:05D2                             SETREGIS        CONTROL,00H
1563  FE00:05D9                             SETREGIS        COMMAND,DISCON
1568  FE00:05E0  E8 20 FF                   CALL    WAITFREE
1569                                ;       CALL    WAITFINT
1570  FE00:05E3  E8 E3 FF                   CALL    GETSTAT
1571                                ;       MOV     AL,'D'
1572                                ;       CALL    PUTCH
1573                                ;       CALL    CRLF
1574                                ;       CALL    SPACE
1575                                ;       MOV     AH,LASTSTAT
1576                                ;       MOV     AL,0EEH
1577                                ;       CALL    SHOWSCSI
1578  FE00:05E6  C3                         RET
1579
1580                                        PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1581                                    ; MODE TO BE FOUND IN SCMODE
1582                                    ; ADDRESS TO BE FOUND IN SCADHI,SCADMID,SCADLO
1583                                    ; LENGTH TO BE LIMITED TO 64K, FOUND IN SCLNHI,SCLNLO
1584  FE00:05E7                         WRITSCSI
1585  FE00:05E7                               SETREGIS        TRANSLO,SCLNLO
1590  FE00:05F0                               SETREGIS        TRANSMID,SCLNHI
1595  FE00:05F9                               SETREGIS        TRANSHI,ZERO
1600  FE00:0600                               SETREGIS        CONTROL,BIT7
1605  FE00:0607                               SETREGIS        DESTID,HOSTID
1610
1611                                    ; SET UP DMA
1612  FE00:0610                               INTOFF
1615  FE00:0611                               OUTPUT  DMA0A,SETOFF^OUTP       ;MASK OUTPUT CHANNEL
1620  FE00:0617                               OUTPUT  DMA0B,SINGLEMD^ADDINC^NOAUTOIN^FROMEM^OUTP
1625                                                                           ;SET DMA CHANNEL MODE
1626  FE00:061D                               OUTPUT  DMA0C,00H               ;CLEAR BYTE FLIPFLOP
1631  FE00:0623                               OUTPUT  DMA04,SCADLO            ;SEND LOW ADDRESS
1636  FE00:062A                               OUTPUT  DMA04,SCADMID           ;SEND HIGH ADDRESS
1641  FE00:0631  A1 44 04                     MOV     AX,SCLNLO               ;GET LENGTH
1642  FE00:0634  48                           DEC     AX                      ;DECREMENT THE COUNT
1643  FE00:0635  50                           PUSH    AX                      ;AND SAVE
1644  FE00:0636                               OUTPUT  DMA05,AL                ;SPECIFY LOW COUNT
1649  FE00:063C  58                           POP     AX
1650  FE00:063D                               OUTPUT  DMA05,AH                ;SEND HIGH COUNT
1655  FE00:0643                               OUTPUT  DMASEG1,SCADHI          ;SEND 64K SEGMENT NIBBLE
1660                                    ; ENABLE DMA
1661  FE00:064A                               OUTPUT  DMA0A,SETON^OUTP        ;REMOVE MASK
1666  FE00:0650                               INTON
1669                                    ; SHOW STATUS BEFORE ISSUING COMMAND
1670                                    ;       MOV     AH,SCMODE
1671                                    ;       MOV     AL,30H
1672                                    ;       CALL    SHOWSCSI
1673                                    ; SEND THE COMMAND, TO START THE TRANSFER
1674  FE00:0651                               SETREGIS        COMMAND,SCMODE
1679  FE00:065A  88 26 48 04                  MOV     LASTCOMM,AH
1680                                    ; WAIT FOR PROCESS DONE
1681  FE00:065E  E8 89 FE                     CALL    WAITFINT
1682                                    ; READ SCSI STATUS
1683  FE00:0661                               GETREGIS        SCSISTAT
1687  FE00:0666  A2 46 04                     MOV     LASTSTAT,AL
1688                                    ; SHOW STATUS BEFORE ISSUING COMMAND
1689  FE00:0669  8A E0                        MOV     AH,AL
1690  FE00:066B  B0 40                        MOV     AL,040H
1691  FE00:066D  E8 97 00                     CALL    SHOWSCSI
1692  FE00:0670  A0 46 04                     MOV     AL,LASTSTAT
1693  FE00:0673  C3                           RET
1694
1695                                    ;       GET THE SCSI REGISTERS INTO THE DATA ARRAY CDAT
1696  FE00:0674                         GETSCSIREG
1697  FE00:0674  E8 27 FE                     CALL    AUXSTAT         ;GET AUX STATUS BEFORE STATUS
1698  FE00:0677  A2 00 04                     MOV     CDATA,AL
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1699  FE00:067A  B9 1A 00                    MOV     CX,26
1700  FE00:067D  BB 01 04                    MOV     BX,OFFSET CDATA+1
1701  FE00:0680  BA 00 00                    MOV     DX,ZERO
1702  FE00:0683  52                  GSREG1: PUSH    DX
1703  FE00:0684  51                          PUSH    CX
1704  FE00:0685  53                          PUSH    BX
1705  FE00:0686                              GETREGIS DL
1709  FE00:068B  5B                          POP     BX
1710  FE00:068C  88 07                       MOV     [BX],AL
1711  FE00:068E  43                          INC     BX
1712  FE00:068F  59                          POP     CX
1713  FE00:0690  5A                          POP     DX
1714  FE00:0691  42                          INC     DX
1715  FE00:0692  E2 EF                       LOOP    GSREG1
1716  FE00:0694  C3                          RET
1717
1718                                         PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1719                                    ; DIAGNOSTIC SUBROUTINES
1720            FE00 : 0000 03F8        SERBASE EQU     3F8H
1721            FE00 : 0000 03F8        TX      EQU     SERBASE
1722            FE00 : 0000 03F8        RX      EQU     SERBASE
1723            FE00 : 0000 03F8        DLLSB   EQU     SERBASE
1724            FE00 : 0000 03F9        DLMSB   EQU     SERBASE+1
1725            FE00 : 0000 03F9        IEREG   EQU     SERBASE+1
1726            FE00 : 0000 03FA        IIREG   EQU     SERBASE+2
1727            FE00 : 0000 03FB        LCREG   EQU     SERBASE+3
1728            FE00 : 0000 03FC        MCREG   EQU     SERBASE+4
1729            FE00 : 0000 03FD        LSREG   EQU     SERBASE+5
1730            FE00 : 0000 03FE        MSREG   EQU     SERBASE+6
1731
1732  FE00:0695                         SERRESET
1733                                    ; DISABLE INTERRUPTS
1734  FE00:0695                                 OUTPUT  LCREG,03H             ;TURN OFF DLAB
1739  FE00:069B                                 OUTPUT  IEREG,00H
1744                                    ; SET BAUD RATE
1745  FE00:06A1                                 OUTPUT  LCREG,83H             ;TURN ON DLAB
1750  FE00:06A7                                 OUTPUT  DLLSB,0CH
1755  FE00:06AD                                 OUTPUT  DLMSB,00H
1760                                    ; SET TO READ OR WRITE DATA
1761  FE00:06B3                                 OUTPUT  LCREG,03H             ;TURN OFF DLAB
1766                                    ; SET MODEM CONTROL TO DTR,RTS
1767  FE00:06B9                                 OUTPUT  MCREG,03H
1772  FE00:06BF  C3                             RET
1773
1774  FE00:06C0                         PUTCH:
1775  FE00:06C0  C3                             RET
1776  FE00:06C1  50                             PUSH    AX
1777  FE00:06C2  E8 02 FD              PUTCH1   CALL    CHKATOD
1778  FE00:06C5                                 INPUT   LSREG
1782  FE00:06C9  24 20                          AND     AL,20H                ;SEE IF HOLDING
1783  FE00:06CB  74 F5                          JE      PUTCH1
1784  FE00:06CD  58                             POP     AX
1785  FE00:06CE                                 OUTPUT  TX,AL
1790  FE00:06D4  C3                             RET
1791  FE00:06D5                         CRLF
1792  FE00:06D5  B0 0D                          MOV     AL,0DH
1793  FE00:06D7  E8 E6 FF                       CALL    PUTCH
1794  FE00:06DA  B0 0A                          MOV     AL,0AH
1795  FE00:06DC  E8 E1 FF                       CALL    PUTCH
1796  FE00:06DF  C3                             RET
1797  FE00:06E0                         SPACE
1798  FE00:06E0  B0 20                          MOV     AL,20H
1799  FE00:06E2  E8 DB FF                       CALL    PUTCH
1800  FE00:06E5  C3                             RET
1801  FE00:06E6                         PRBYTE
1802  FE00:06E6  51                             PUSH    CX              SAVE CX
1803  FE00:06E7  50                             PUSH    AX              SAVE AX
1804  FE00:06E8  D0 E8                          SHR     AL,1
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1805  FE00:06EA  D0 E8                        SHR     AL,1
1806  FE00:06EC  D0 E8                        SHR     AL,1
1807  FE00:06EE  D0 E8                        SHR     AL,1
1808  FE00:06F0  E8 06 00                     CALL    PUTCHNIB
1809  FE00:06F3  58                           POP     AX
1810  FE00:06F4  E8 02 00                     CALL    PUTCHNIB
1811  FE00:06F7  59                           POP     CX
1812  FE00:06F8  C3                           RET
1813  FE00:06F9                       PUTCHNIB
1814  FE00:06F9  24 0F                        AND     AL,0FH
1815  FE00:06FB  3C 0A                        CMP     AL,0AH
1816  FE00:06FD  7C 02                        JL      PN1
1817  FE00:06FF  04 07                        ADD     AL,07H
1818  FE00:0701  04 30                PN1     ADD     AL,30H
1819  FE00:0703  E8 BA FF                     CALL    PUTCH
1820  FE00:0706  C3                           RET
1821
1822  FE00:0707                       SHOWSCSI
1823  FE00:0707  50                           PUSH    AX
1824  FE00:0708  53                           PUSH    BX
1825  FE00:0709  51                           PUSH    CX
1826  FE00:070A  52                           PUSH    DX
1827  FE00:070B  50                           PUSH    AX
1828  FE00:070C  E8 D7 FF                     CALL    PRBYTE
1829  FE00:070F  E8 CE FF                     CALL    SPACE
1830  FE00:0712  58                           POP     AX
1831  FE00:0713  8A C4                        MOV     AL,AH
1832  FE00:0715  E8 CE FF                     CALL    PRBYTE
1833  FE00:0718  E8 BA FF                     CALL    CRLF
1834  FE00:071B  E8 56 FF                     CALL    GETSCSIREG
1835  FE00:071E  BB 00 04                     MOV     BX,OFFSET CDATA
1836  FE00:0721  B9 10 00                     MOV     CX,16
1837  FE00:0724                       SHOWSC1
1838  FE00:0724  8A 07                        MOV     AL,[BX]
1839  FE00:0726  43                           INC     BX
1840  FE00:0727  E8 BC FF                     CALL    PRBYTE
1841  FE00:072A  E8 B3 FF                     CALL    SPACE
1842  FE00:072D  E2 F5                        LOOP    SHOWSC1
1843  FE00:072F  E8 A3 FF                     CALL    CRLF
1844
1845  FE00:0732  B9 0B 00                     MOV     CX,11
1846  FE00:0735                       SHOWSC2
1847  FE00:0735  8A 07                        MOV     AL,[BX]
1848  FE00:0737  43                           INC     BX
1849  FE00:0738  E8 AB FF                     CALL    PRBYTE
1850  FE00:073B  E8 A2 FF                     CALL    SPACE
1851  FE00:073E  E2 F5                        LOOP    SHOWSC2
1852  FE00:0740  E8 92 FF                     CALL    CRLF
1853  FE00:0743  5A                           POP     DX
1854  FE00:0744  59                           POP     CX
1855  FE00:0745  5B                           POP     BX
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1856  FE00:0746  58                          POP     AX
1857  FE00:0747  C3                          RET
1858
1859                                  ;SHOWDATA
1860                                  ;       PUSH    AX
1861                                  ;       PUSH    BX
1862                                  ;       PUSH    CX
1863                                  ;       PUSH    DX
1864                                  ;       MOV     BX,OFFSET DUMMY
1865                                  ;       MOV     CX,16
1866                                  ;
1867                                  ;SHOWDA1
1868                                  ;       MOV     AL,[BX]
1869                                  ;       PUSH    AX
1870                                  ;       INC     BX
1871                                  ;       MOV     AL,[BX]
1872                                  ;       INC     BX
1873                                  ;       CALL    PRBYTE
1874                                  ;       POP     AX
1875                                  ;       CALL    PRBYTE
1876                                  ;       CALL    SPACE
1877                                  ;       LOOP    SHOWDA1
1878                                  ;       CALL    CRLF
1879                                  ;
1880                                  ;       POP     DX
1881                                  ;       POP     CX
1882                                  ;       POP     BX
1883                                  ;       POP     AX
1884                                  ;       RET
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901                                  ; INCFLAG INCREMENTS AND OUTPUTS A FLAG BYTE
1902  FE00:0748  A0 5B 04             SET5    MOV     AL,FLAG
1903  FE00:074B  0C 20                        OR      AL,BIT5
1904  FE00:074D  E9 25 00                     JMP     INCF1
1905
1906  FE00:0750  A0 5B 04             SET6    MOV     AL,FLAG
```

```
                TEST PROGRAM FOR A/D ENGINE
                SCSI SUBROUTINES

1907  FE00:0753  0C 40                        OR      AL,BIT6
1908  FE00:0755  E9 1D 00                     JMP     INCF1
1909
1910  FE00:0758  A0 5B 04             SET7     MOV     AL,FLAG
1911  FE00:075B  0C 80                        OR      AL,BIT7
1912  FE00:075D  E9 15 00                     JMP     INCF1
1913
1914  FE00:0760  A0 5B 04             SET0     MOV     AL,FLAG
1915  FE00:0763  0C 01                        OR      AL,BIT0
1916  FE00:0765  E9 0D 00                     JMP     INCF1
1917
1918  FE00:0768                       FREEZE:
1919  FE00:0768  0C F0                        OR      AL,0F0H
1920  FE00:076A                               OUTPUT  PARB,AL
1925  FE00:0770  EB FE                FREEZE1  JMP     SHORT   FREEZE1
1926  FE00:0772  C3                           RET
1927  FE00:0773                       ZEROFLAG
1928  FE00:0773  B0 00                        MOV     AL,00H
1929                                  ;INCFLAG INC      FLAG
1930  FE00:0775  A2 5B 04             INCF1    MOV     FLAG,AL
1931  FE00:0778                               OUTPUT  PARB,FLAG
1936  FE00:077F  C3                           RET
1937
1938  FE00:0780  A0 5B 04             SET1     MOV     AL,FLAG
1939  FE00:0783  0C 02                        OR      AL,BIT1
1940  FE00:0785  E9 ED FF                     JMP     INCF1
1941
1942  FE00:0788  A0 5B 04             SET2     MOV     AL,FLAG
1943  FE00:078B  0C 04                        OR      AL,BIT2
1944  FE00:078D  E9 E5 FF                     JMP     INCF1
1945
1946  FE00:0790  A0 5B 04             SET3     MOV     AL,FLAG
1947  FE00:0793  0C 08                        OR      AL,BIT3
1948  FE00:0795  E9 DD FF                     JMP     INCF1
1949
1950  FE00:0798  A0 5B 04             SET4     MOV     AL,FLAG
1951  FE00:079B  0C 10                        OR      AL,BIT4
1952  FE00:079D  E9 D5 FF                     JMP     INCF1
1953
1954  FE00:1FF0                               ORG     1FF0H           ;SET UP RESET
1955  FE00:1FF0  EA 00 00 00 FE               JMP FAR START
1956
1957                                          PAGE
```

```
                TEST PROGRAM FOR A/D ENGINE
                DATA SPACE

1958                                            SUBTITLE DATA SPACE
1959  0000:0000                                 .DATA
1960  0000:0000                                 .ORG    IPUR
1961  0000:0000                         IVECS:          .BLKW   256*2
1962
1963                                    ; SCSI VARIABLES
1964  0000:0400                         CDATA   BLKB    BUFSZ
1965  0000:0440  01                     SCMODE  .BYTE   1
1966  0000:0441  01                     SCADLO  .BYTE   1
1967  0000:0442  01                     SCADMID .BYTE   1
1968  0000:0443  01                     SCADHI  .BYTE   1
1969  0000:0444                         SCLN:
1970  0000:0444  01                     SCLNLO  .BYTE   1
1971  0000:0445  01                     SCLNHI  .BYTE   1
1972  0000:0446  01                     LASTSTAT BYTE   1
1973  0000:0447  01                     COMM     BYTE   1
1974  0000:0448  01                     LASTCOMM BYTE   1
1975  0000:0449  01                     ABNORMAL BYTE   1
1976  0000:044A  01                     ENDSTAT  BYTE   1
1977  0000:044B  01                     HOSTID   BYTE   1
1978  0000:044C  01                     XSTATUS  BYTE   1
1979  0000:044D  01                     XMESSAGE BYTE   1
1980  0000:044E  01                     IDENTIFY BYTE   1
1981  0000:044F  01                     LENGTH   BYTE   1
1982  0000:0450  01                     DATATEMP BYTE   1
1983  0000:0451  01                     SENSEKEY BYTE   1
1984
1985            0000 : 0000 041A        AUXREG  EQU     CDATA+26        ;       BYTE PTR AUXREG ETC.
1986                                    ;SENSEKEY EQU    CDATA+27
1987            0000 : 0000 041C        ERRCLASS EQU    CDATA+28
1988            0000 : 0000 041D        ERRTYPE  EQU    CDATA+29
1989                                    ; ERRTYPE EQUALS ZERO ON START UP
1990                                    ; ERRTYPE EQUALS FFH IF RESET DOESN'T SEEM TO WORK
1991                                    ;       0FEH                    ;DISCONNECT ERROR
1992            0000 : 0000 041E        OLDSTAT  EQU    CDATA+30
1993            0000 : 0000 041F        LASTCOM  EQU    CDATA+31
1994
1995                                    ; A TO D VARIABLES, BUFFER VARIABLES
1996
1997
1998  0000:0452                         BUFCNT:         .BLKW   1
1999  0000:0454                         INPTR:          .BLKW   1
2000  0000:0456                         OUTPTR:         .BLKW   1
2001  0000:0458  01                     HIADD:          .BYTE   1
2002  0000:0459  01                     HINIB:          .BYTE   1
2003  0000:045A  01                     RATE1:          .BYTE   1
2004
2005                                    ; TEST VARIABLES
2006  0000:045B  01                     FLAG:           .BYTE   1
2007  0000:045C  01                     SSF:            .BYTE   1
2008
```

```
                TEST PROGRAM FOR A/D ENGINE
                DATA SPACE

2009                                    ; SCSI VARIABLES
2010  0000:045D  01                     DEVID:          .BYTE   1
2011
2012  0000:0600                                 ORG     600H
2013  0000:0600                         STACK:  .BLKB   512
2014  0000:0800                         STACKT:
2015  0000:4000                                 ORG     4000H
2016  0000:4000                         DUMMY:
2017  0000:4000                                 .END
```

TEST PROGRAM FOR A/D ENGINE
DATA SPACE

********** S Y M B O L I C R E F E R E N C E T A B L E **********

ABASIC	= 0000 : 0000 0000	ABIDIR	= 0000 : 0000 0040	ABNORMAL	0000 : 0000 0449	ABORT	= 0000 : 0000 0001
ADDDEC	= 0000 : 0000 0020	ADDINC	= 0000 : 0000 0000	AINPUT	= 0000 : 0000 0010	AMASKBIT	= 0000 : 0000 0002
AOUTPUT	= 0000 : 0000 0000	ASTROBE	= 0000 : 0000 0020	AUTOIN	= 0000 : 0000 0010	AUXREG	= 0000 : 0000 041A
AUXSTAT	7E00 : 0000 049E	BAD	= 0000 : 0000 00FF	BADCOMM	= 0000 : 0000 0005	BBASIC	= 0000 : 0000 0000
BINPUT	= 0000 : 0000 0002	BIT0	= 0000 : 0000 0001	BIT1	= 0000 : 0000 0002	BIT2	= 0000 : 0000 0004
BIT3	= 0000 : 0000 0008	BIT4	= 0000 : 0000 0010	BIT5	= 0000 : 0000 0020	BIT6	= 0000 : 0000 0040
BIT7	= 0000 : 0000 0080	BITWIDC	= 0000 : 0000 0000	BLOCKMD	= 0000 : 0000 0080	BOUTPUT	= 0000 : 0000 0000
BSTROBE	= 0000 : 0000 0004	BUFCNT	0000 : 0000 0452	BUFPWR	= 0000 : 0000 000C	BUFSIZE	= 0000 : 0000 1000
BUFSZ	= 0000 : 0000 0040	CASCADMD	= 0000 : 0000 00C0	CDATA	0000 : 0000 0400	CDB1	= 0000 : 0000 0003
CDB2	= 0000 : 0000 0004	CDB3	= 0000 : 0000 0005	CDB4	= 0000 : 0000 0006	CDB5	= 0000 : 0000 0007
CDB6	= 0000 : 0000 0008	CDOWNIN	= 0000 : 0000 0001	CDOWNOUT	= 0000 : 0000 0000	CH0	= 0000 : 0000 0000
CH0MODE	= 0000 : 0000 0058	CH1	= 0000 : 0000 0001	CH1MODE	= 0000 : 0000 0041	CH2	= 0000 : 0000 0002
CH2MODE	= 0000 : 0000 0042	CH3	= 0000 : 0000 0003	CH3MODE	= 0000 : 0000 0043	CHKATOD	7E00 : 0000 03C7
CHKATOD1	7E00 : 0000 03E2	CHKATOD2	7E00 : 0000 03E5	CHKATODX	7E00 : 0000 03D2	CHKSTAT	= 0000 : 0000 0002
CLK0	= 0000 : 0000 0304	CLK1	= 0000 : 0000 0305	CLK2	= 0000 : 0000 0306	CLKCON	= 0000 : 0000 0307
COMM	0000 : 0000 0447	COMMAND	= 0000 : 0000 0018	COMPHASE	= 0000 : 0000 0010	CONTROL	= 0000 : 0000 0001
CRLF	7E00 : 0000 06D5	CUPIN	= 0000 : 0000 0008	CUPOUT	= 0000 : 0000 0000	DATAREG	= 0000 : 0000 0019
DATATEMP	0000 : 0000 0450	DELAY	7E00 : 0000 0492	DELAY1	7E00 : 0000 0492	DELAYP	7E00 : 0000 0495
DEMANDMD	= 0000 : 0000 0000	DESTID	= 0000 : 0000 0015	DEVID	0000 : 0000 045D	DISCON	= 0000 : 0000 0004
DISCONNECT	7E00 : 0000 05D2	DISPAUX	7E00 : 0000 0517	DLLSB	= 7E00 : 0000 03F8	DLMSB	= 7E00 : 0000 03F9
DMA00	= 0000 : 0000 0000	DMA01	= 0000 : 0000 0001	DMA02	= 0000 : 0000 0002	DMA03	= 0000 : 0000 0003
DMA04	= 0000 : 0000 0004	DMA05	= 0000 : 0000 0005	DMA06	= 0000 : 0000 0006	DMA07	= 0000 : 0000 0007
DMA08	= 0000 : 0000 0008	DMA09	= 0000 : 0000 0009	DMA0A	= 0000 : 0000 000A	DMA0B	= 0000 : 0000 000B
DMA0C	= 0000 : 0000 000C	DMA0D	= 0000 : 0000 000D	DMA0E	= 0000 : 0000 000E	DMA0F	= 0000 : 0000 000F
DMASEG1	= 0000 : 0000 0083	DMASEG2	= 0000 : 0000 0081	DMASEG3	= 0000 : 0000 0082	ENDSTAT	0000 : 0000 044A
ERRCLASS	= 0000 : 0000 041C	ERRTYPE	= 0000 : 0000 041D	FIFO	= 0000 : 0000 0308	FIX	7E00 : 0000 0464
FLAG	0000 : 0000 045B	FREEZE	7E00 : 0000 0768	FREEZE1	7E00 : 0000 0770	FROMEM	= 0000 : 0000 0008
GETSCSIREG	7E00 : 0000 0674	GETSTAT	7E00 : 0000 05C9	GOOD	= 0000 : 0000 0000	GSREG1	7E00 : 0000 0683
HARDERR	= 0000 : 0000 0004	HIADD	0000 : 0000 0458	HINIB	0000 : 0000 0459	HOSTID	0000 : 0000 044B
IDENTIFY	0000 : 0000 044E	IEREG	= 7E00 : 0000 03F9	IFIX	7E00 : 0000 0456	IIREG	= 7E00 : 0000 03FA
INCF1	7E00 : 0000 0775	INITATOD	7E00 : 0000 03E7	INP	= 0000 : 0000 0001	INPTR	0000 : 0000 0454
INQUIRY	= 0000 : 0000 0012	INTBIT	= 0000 : 0000 0080	INTREG0	= 0000 : 0000 0020	INTREG1	= 0000 : 0000 0021
IOBASE	= 0000 : 0000 0300	IPUR	= 0000 : 0000 0000	IPURHI	= 0000 : 0000 0000	IPURLO	= 0000 : 0000 0000
IVECS	0000 : 0000 0000	LASTCOM	= 0000 : 0000 041F	LASTCOMM	0000 : 0000 0448	LASTSTAT	0000 : 0000 0446
LCREG	= 7E00 : 0000 03FB	LENGTH	0000 : 0000 044F	LSREG	= 7E00 : 0000 03FD	MAXBUF	= 0000 : 0000 009F
MCREG	= 7E00 : 0000 03FC	MINUSONE	= 0000 : 0000 FFFF	MJV	= 0000 : 0000 0001	MNV	= 0000 : 0000 0000
MSREG	= 7E00 : 0000 03FE	NEXTATOD	7E00 : 0000 0400	NMIREG	= 0000 : 0000 00A0	NOAUTOIN	= 0000 : 0000 0000
NOSENSE	= 0000 : 0000 0000	OLDSTAT	= 0000 : 0000 041E	ONE	= 0000 : 0000 0001	OUTP	= 0000 : 0000 0002
OUTPTR	0000 : 0000 0456	OUTSEG	= 0000 : 0000 0081	OWNID	= 0000 : 0000 0000	PARA	= 0000 : 0000 0300
PARB	= 0000 : 0000 0301	PARC	= 0000 : 0000 0302	PARCON	= 0000 : 0000 0303	PAUSECONST	= 7E00 : 0000 0020
PN1	7E00 : 0000 0701	POLLING	7E00 : 0000 053E	POLLINGX	7E00 : 0000 0549	PRBYTE	7E00 : 0000 06E6
PUR	= 7E00 : 0000 0000	PURHI	= 0000 : 0000 FE00	PURLO	= 0000 : 0000 0000	PUTCH	7E00 : 0000 06C0
PUTCH1	7E00 : 0000 06C2	PUTCHNIB	7E00 : 0000 06F9	PUTDATA	7E00 : 0000 05A4	PUTDATA1	7E00 : 0000 05AA
PUTDATAX	7E00 : 0000 05BC	DUMMY	0000 : 0000 4000	RATE0	= 0000 : 0000 12A5	RATE1	0000 : 0000 045A
READREG	7E00 : 0000 04A3	RECEIVE	= 0000 : 0000 0008	RELEASE	= 0000 : 0000 0004	REQ5	7E00 : 0000 02DC
REQSENSE	= 0000 : 0000 0003	REQX	7E00 : 0000 0336	RESATOD	7E00 : 0000 036F	RESEL	= 0000 : 0000 0005
RESELSEN	= 0000 : 0000 000B	RESERVD	= 0000 : 0000 0001	RESET	= 0000 : 0000 0000	RESETIT	7E00 : 0000 00A6

DATA SPACE

SCADLO		0000	0000	0441	SCADMID		0000	0000	0442	SCLN		0000	0000	0444	SCLNHI		0000	0000	0445
SCLNLO		0000	0000	0444	SCMODE		0000	0000	0440	SCSENDA		7E00	0000	01E9	SCSI0	=	0000	0000	030C
SCSI1	=	0000	0000	030D	SCSICODE		7E00	0000	00A9	SCSIDFLT		7E00	0000	0173	SCSIDISCON	=	0000	0000	0004
SCSIEND		7E00	0000	035A	SCSIEND1		7E00	0000	035A	SCSIEND2		7E00	0000	035C	SCSIL1		7E00	0000	00EA
SCSIL2		7E00	0000	0134	SCSIL3		7E00	0000	013E	SCSIL4		7E00	0000	014D	SCSIL5		7E00	0000	0154
SCSIL6		7E00	0000	015B	SCSIL7		7E00	0000	0162	SCSIL8		7E00	0000	0169	SCSILOOP		7E00	0000	00ED
SCSINQ		7E00	0000	017B	SCSIQEND		7E00	0000	034E	SCSIREQS		7E00	0000	02D7	SCSIREZ		7E00	0000	0343
SCSIRSAV		7E00	0000	036C	SCSISEND		7E00	0000	01D9	SCSISEND1		7E00	0000	0235	SCSISEND2		7E00	0000	0202
SCSISEND3		7E00	0000	0210	SCSISNDI		7E00	0000	0170	SCSISTAT	=	0000	0000	0017	SCSITRDY		7E00	0000	035A
SEND	=	0000	0000	000A	SENDDATA	=	0000	0000	0015	SENDDIAG	=	0000	0000	001A	SENDMESS	=	0000	0000	0016
SENDSTAT	=	0000	0000	0014	SENSEKEY		0000	0000	0451	SERBASE	=	7E00	0000	03F8	SERRESET		7E00	0000	0695
SET0		7E00	0000	0760	SET1		7E00	0000	0780	SET2		7E00	0000	0788	SET200		7E00	0000	0093
SET3		7E00	0000	0790	SET4		7E00	0000	0798	SET5		7E00	0000	0748	SET50		7E00	0000	009B
SET6		7E00	0000	0750	SET7		7E00	0000	0758	SETCLK		7E00	0000	038F	SETOFF	=	0000	0000	0004
SETON	=	0000	0000	0000	SETPARMODE	=	0000	0000	0080	SETREG		7E00	0000	04C3	SEVEN	=	0000	0000	0007
SHIFTCNT	=	0000	0000	0004	SHOWAUX		7E00	0000	0531	SHOWBUFFER		7E00	0000	052B	SHOWFLAG		7E00	0000	0525
SHOWSC1		7E00	0000	0724	SHOWSC2		7E00	0000	0735	SHOWSCSI		7E00	0000	0707	SINGLEMD	=	0000	0000	0040
SOURCID	=	0000	0000	0016	SPACE		7E00	0000	06E0	SPARA	=	0000	0000	0060	SPARB	=	0000	0000	0061
SPARC	=	0000	0000	0062	SPARCON	=	0000	0000	0063	SS1		7E00	0000	02C7	START		7E00	0000	0000
STX		7E00	0000	00A3	SYNCHTR	=	0000	0000	0011	T0		7E00	0000	0372	T1		7E00	0000	0380
TARLUN	=	0000	0000	000F	SSF		0000	0000	045C	TESTRDY	=	0000	0000	0000	TIM0	=	0000	0000	0040
TIM1	=	0000	0000	0041	TIM2	=	0000	0000	0042	TIMCON	=	0000	0000	0043	TIMEOUT	=	0000	0000	0002
STACK		0000	0000	0600	TOMEM	=	0000	0000	0004	TRANSHI	=	0000	0000	0012	TRANSLO	=	0000	0000	0014
TRANSMID	=	0000	0000	0013	TX	=	7E00	0000	03F8	UPDBC		7E00	0000	047C	UPDBCX		7E00	0000	0489
UPDIN		7E00	0000	0447	UPDINX		7E00	0000	0453	UPDOUT		7E00	0000	046C	UPDOUTX		7E00	0000	0478
STACKT		0000	0000	0800	VERIFY	=	0000	0000											

```
#include "../online/header.h"
struct header h;

#define MINETA (-50)
#define MAXETA  50
#define SHOWWIDTH 4
#define DECIM  2
#define NUMGET 4096
#define TAPEBLOCK 8192
#define XZERO  0
#define BEGINTRACE 60
#define YZERO    65
#define XSIZE  1152
#define YSIZE   820
#define MAXCHAN  64
#define PERNOISE 100
#define GAINSWEEPS  1000
#define ATODDEV "/dev/rsip0"
#define BOUNDS 5
int numget=NUMGET;

struct stats {
   int wirenum;
   char name[5];
   int discr;
} park[]={
    1,"KSV ",680,
    2,"KBTE",1020,
    3,"KBTN",1360,
    4,"KBTV",1700,
    5,"KRBE",2040,
    6,"KRBN",2380,
    7,"KRBV",2720,
    8,"KSM ",3060,
    9,"KTP ",680,
   10,"KKCN",1020,
   11,"PUB ",1360,
   12,"BLM ",1700,
   13,"KKCE",2040,
   14,"KKCV",2380,
   15,"WWR ",2720,
   16,"KKR ",3060,
   17,"KDME",400,
   18,"KDMN",404,
   19,"KDMV",346,
   20,"XXXX",347,
   21,"XXXX",462,
   22,"XXXX",466,
   23,"XXXX",585,
   24,"XXXX",586,
   25,"XXXX",587,
   26,"XXXX",590,
   27,"XXXX",494,
   28,"XXXX",495,
   29,"XXXX",313,
   30,"XXXX",500,
   31,"XXXX",625,
   32,"XXXX",309,
};

#define LOW4BITS 0xF
#include <stdio.h>
#include <sys/file.h>
#include <sys/ioctl.h>
```

```
#include <signal.h>
#include <sys/types.h>

#include "/usr/sip/sundev/sipreg.h"
struct sip_errorcode sen;

#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <suntool/panel.h>
#include <suntool/icon.h>
#include <pixrect/pixrect_hs.h>

struct rect framesize = {XZERO,YZERO,XSIZE,YSIZE}; /* size and position of frame */

static int my_done=0;

static short seis_icon[]={
#include "seis.icon"
};
DEFINE_ICON_FROM_IMAGE(my_icon,seis_icon);

char buf[TAPEBLOCK];
char decstr[][7]= {"","other ","3rd "};
int endit=0;

int intr=1;
onintr() {
   intr=0;
}

static Notify_value my_frame_destroyer(frame,status)
   Frame frame;
   Destroy_status status;
{
   onintr();
   if(endit) return(notify_next_destroy_func(frame,status));
   notify_veto_destroy(frame);
   return(NOTIFY_DONE);
}

#define ATODIN  0
#define FILEOUT 1
#define FILESIN 10
#define TMPFILE 10
#define EQFILE  11
#define TAPEIN  12

/*------------ variables used in triggering calculations -------------*/
#define MAX_STA 32
long    ssum        [MAX_STA];      /* short-term sums for one sweep   */
long    rsum        [MAX_STA];      /* rectified short-term sums       */
int     sbars       [MAX_STA];      /* regular short-term averages     */
int     rbars       [MAX_STA];      /* rectified short-term averages   */
int     sbarbar     [MAX_STA];      /* regular long-term averages      */
int     rbarbar     [MAX_STA];      /* rectified long-term averages    */
int     etas        [MAX_STA];      /* eta values for each station     */
int     ploteta     [MAX_STA];
int     oploteta    [MAX_STA];
int     sta_trigger[MAX_STA];       /* individual station trigger      */
int     subnet[] = {3,0,1,2,3,4,5,6,7,-1,3,8,9,10,11,12,13,14,15,-1,3,16,
17,18,19,20,21,22,23,-1,3,24,25,26,27,28,29,30,31,-1,-1};

int eta_num   =    2 ;              /* numerator of eta constant       */
int eta_den   =    1 ;              /* denominator of eta constant     */
```

```
int eta_con   =    5 ;              /* another eta constant          */
int diffsam   =    0 ;              /* don't diff samps bef computatn */
int sweeps    =    3.;              /* secs of data for short term av */
int aperture  =   20.;              /* secs for coincident statn trig */
int nsw_aperture,ns_sweep,sweep;

int dec;
int dev;

int ptr= -1;
int maxptr= -1;
float gain=0.0;
double atof();
int first= -1;
int last= -1;
int print_ave=0;
int detector=0;
int percent=0;
int rate=100;
int trigged=0;
int plotx=1;
int yincr;
int firsttime=1;
int tapein=0;

main(argc,argv)
   int argc;
   char **argv;
{
   int i;

   signal(SIGINT,onintr);

   dec=DECIM;
   if(dec<1)dec=1;
   if(argc==1) plotit(ATODDEV,NULL,ATODIN);
   else for(i=1;i<argc;i++) {
      endit=0;
      ptr= -1;
      maxptr= -1;
      intr=1;
      firsttime=1;
      if(argv[i][0]=='-') {
         switch(argv[i][1]) {
            case 'd': dec=atoi(argv[++i]);
                      if(dec<1)dec=1; break;
            case 'f': first=atoi(argv[++i]); break;
            case 'g': gain =atof(argv[++i]); break;
            case 'l': last =atoi(argv[++i]); break;
            case 'p': percent=atoi(argv[++i]); break;
            case 'x': plotx=atoi(argv[++i]); break;
            case 'D': detector=1; dec=1; break;
            case 'P': print_ave=1; break;
            default: printf("plotatod: Unknown argument %s\n",argv[i]);
         }
      }
      else if(strncmp("atod",argv[i],4)==0)
         plotit(ATODDEV,NULL,ATODIN);
      else if(strncmp("eq.",argv[i],3)==0)
         plotit(argv[i],NULL,EQFILE);
      else if(strncmp("tmp",argv[i],3)==0)
         plotit("/usr/tmp/atod.out",NULL,TMPFILE);
      else if(strncmp("tape",argv[i],4)==0) {
         plotit("/dev/rst0",NULL,TAPEIN);
```

```
            numget=TAPEBLOCK;
        }
          else plotit(ATODDEV,argv[i],FILEOUT);
        }
}

plotit(input,output,type)
    char *input,*output;
    int type;
{
    int i,j,k,s,numchan,numst,numstin,ioc,yy,yyl,swidth,samp,num;
    int ysize;
    short *ibuf;
    short x,x0,ex0,y[MAXCHAN],y0[MAXCHAN],yzero;
    char argp[16];
    Frame frame;
    Canvas canvas;
    Pixwin *pw;
    char frame_label[120];
    Icon icon;
    struct pr_prpos where;
    struct pixfont *font;
    int yz,xshow,dx,tempp;
    extern char *malloc();
    long stasum[32],stamax[32],stamin[32];
    float millivolts[32],const,val;
    char temp[6],ttime[16];
    int lastsam[MAX_STA],thissam,diff,xcounts;

    if(percent==0)percent=PERNOISE;
    const=2048*16/5000;     /* samples per millivolt including time 4 bits */
    dev=open(input,O_RDONLY);
    if(dev== -1) {
        perror("plotatod");
        return(1);
    }
    if(type<FILESIN) {
        ioc=ioctl(dev,SIPIOC_REZERO);
        if(ioc== -1) perror("rezero");
    }
    if(type==EQFILE) {
        ioc=read(dev,&h,HEADSIZE);
        if(ioc!=HEADSIZE) {
            perror("reading eq.file header");
            return(1);
        }
    }

    ioc=read(dev,buf,numget);
    if(ioc!=numget) {
        perror("first read");
        return(1);
    }
    ibuf=(short *)buf;
    for(numchan=0,i=1;i<257;i+=16){
        if((j=ibuf[i]&LOW4BITS) > numchan) numchan=j;
    }
    numchan++;                              /* number of multiplexers */
    numst=numstin=numchan*16;                       /* number of channels */
    printf("Number of channels input is %d.\n",numst);
    if(first>last) { tempp=first; first=last; last=tempp; }
    if(last== -1 || last>numst)last=numst;
    if(first<=0)first=1;
    numst=last-first+1;
```

```
printf("Plot channels %d through %d.\n",first,last);
if(dec<4)sprintf(frame_label,
   "Input from %s for %d channels and plotting every %spoint.",
   input,numst,decstr[dec-1]);
else sprintf(frame_label,
   "Input from %s for %d channels and plotting every %dth point.",
   input,numst,dec);
yincr=(YSIZE-12)/numst;
ysize=yincr*numst+12;
framesize.r_height=ysize+25;

printf("Begin autoscaling by reading data for %d seconds.\n",GAINSWEEPS/100);
for(i=first-1;i<last;i++) {
   stamax[i]= -10000;
   stamin[i]=10000;
   stasum[i]=0;
}
if(numchan<=2) {                        /* determine scaling of traces */
   if(type>=FILESIN) {
      close(dev);
      dev=open(input,O_RDONLY);
      if(dev== -1) perror("plotatod");
      if(type==EQFILE) {
         ioc=read(dev,&h,HEADSIZE);
         if(ioc!=HEADSIZE)perror("Reading header");
      }
   }
   for(j=0,num=1;j<GAINSWEEPS;j++,num++){
      ioc=getsweep(ibuf,numstin,1,type);
      if(ioc==EOF) {
         printf("Only %d sweeps of data decimated by %d. \n",j,dec);
         j=GAINSWEEPS-1;
      }
      for(i=first-1;i<last;i++) {
         samp=ibuf[i];
         if(samp<stamin[i])stamin[i]=samp;
         if(samp>stamax[i])stamax[i]=samp;
         stasum[i]+=samp;
         if(j==GAINSWEEPS-1) {
            stasum[i]=stasum[i]/num;
            if(print_ave)printf("%5s min=%5d max=%5d ave=%5d\n",
               park[i].name,stamin[i],stamax[i],stasum[i]);
            /* calculate samples per pixel */
            if(gain!=0.0) {
               millivolts[i]=gain;
               stamin[i]=gain*const/yincr;
            }
            else  {
               millivolts[i]=((stamax[i]-stamin[i])*100.0/percent)/const;
               stamin[i]=(stamax[i]-stamin[i])*100/(percent*yincr);
            }
            if(stamin[i]<1)stamin[i]=1;
            sprintf(park[i].name,"%4s",park[i].name);
         }
         /*printf("%x ",samp&LOW4BITS);*/
      }
      /*printf("\n");*/
   }
}
font=pf_open("/usr/lib/fonts/fixedwidthfonts/screen.r.7");
where.pr=(Pixrect *) icon_get(&my_icon,ICON_IMAGE);
where.pos.x=2;
where.pos.y=60;
pf_text(where,PIX_SRC,font,input);
```

```
icon_set(&my_icon,ICON_IMAGE,where.pr,0);

frame=window_create(0,FRAME,
   FRAME_LABEL,       frame_label,
   FRAME_ICON,        &my_icon,
   FRAME_OPEN_RECT,   &framesize,
   0);
canvas=window_create(frame,CANVAS,
   CANVAS_AUTO_SHRINK,        FALSE,
   CANVAS_WIDTH,              XSIZE,
   CANVAS_HEIGHT,             ysize,
   0);

(void) notify_interpose_destroy_func(frame,my_frame_destroyer);
pw=canvas_pixwin(canvas);
window_set(frame,WIN_SHOW,TRUE,0);
yyl=0;
swidth=(XSIZE-BEGINTRACE+SHOWWIDTH)/SHOWWIDTH;
yzero=yincr/2;
for(i=0;i<numst;i++) y0[i]=yzero+i * yincr;
dx=0;
yz=yincr/2;
pw_batch_on(pw);
xshow=BEGINTRACE+swidth;
yzero=0;
for(i=first-1;i<last;i++) {
   pw_vector(pw,0,yzero,BEGINTRACE,yzero,PIX_SRC,1);
   pw_text(pw,1,yzero+yz+6,PIX_SRC,NULL,park[i].name);
   sprintf(temp,"%3.0f",(float)stasum[i]/const);
   pw_text(pw,35,yzero+yz+8,PIX_SRC,font,temp);
   sprintf(temp,"%3.0f",millivolts[i]);
   pw_text(pw,35,yzero+yz-1,PIX_SRC,font,temp);
   yzero+=yincr;
}
printf("Now begin plotting data.\n");
pw_vector(pw,0,yzero,BEGINTRACE,yzero,PIX_SRC,1);
if(type<FILESIN) {
   ioc=ioctl(dev,SIPIOC_REZERO,argp);
   if(ioc== -1) perror("rezero");
}
else {
   close(dev);
   dev=open(input,O_RDONLY);
   if(dev== -1) perror("plotatod");
   if(type==EQFILE) {
      ioc=read(dev,&h,HEADSIZE);
   }
}
ptr= -1;            /* reset getsweep */
maxptr= -1;

if(detector) {
   ns_sweep     = (sweeps        * rate) / dec;   /* Note: decim div */
   nsw_aperture = aperture      / sweeps;        /* aperture of sweeps */
}
x=BEGINTRACE;
x0=BEGINTRACE;
ex0=BEGINTRACE;
sweep=0;
xcounts=0;
while(intr && ((ioc=getsweep(ibuf,numstin,dec,type))!=EOF) ) {
   yzero=yz;
   if(x==BEGINTRACE) {
      for(i=0,j=2;i<15;i++) {
```

```
            if(i==3 || i==8)ttime[i]=' ';
            else if(i==11)ttime[i]='.';
            else ttime[i]=(char)(ibuf[j++]&LOW4BITS)+'0';
         }
         ttime[15]='\0';
      }
      if(detector){
         for(s=first-1; s<last; s++){                    /* just some stations */
            samp     = ibuf[s];                        /* get a2d sample     */
            if(diffsam){                                /* use differences?   */
               thissam    = samp;                       /* save curr sample   */
               samp       = thissam - lastsam[s];       /* diff with last     */
               lastsam[s] = thissam;
            }                                           /* remember last samp */
            diff     = samp - sbarbar[s];               /* diff for abs func  */
            ssum[s] += samp;                            /* add to reg sums    */
            rsum[s] += diff<0? -diff:diff;              /* and rectified sums */
         }
         if(++sweep == ns_sweep){
            sweep=0;
            if( trigger() ) {
               pw_text(pw,x,ysize-1,PIX_SRC,NULL,"TRIGGER");
            }
         }
      }
      for(i=first-1;i<last;i++) {
         yy=yzero-((ibuf[i]-stasum[i])/ stamin[i]);
         pw_vector(pw,x0,y0[i],x,yy,PIX_SRC,1);
         if(detector && trigged){
            if(ploteta[i]>0) {
               pw_vector(pw,ex0,yzero-oploteta[i],ex0,yzero-ploteta[i],PIX_SRC,1);
               if(x<ex0) {
                  pw_vector(pw,ex0,yzero-ploteta[i],XSIZE-2,yzero-ploteta[i],PIX_SRC,1);
                  pw_vector(pw,BEGINTRACE,yzero-ploteta[i],x,yzero-ploteta[i],PIX_SRC,1);
               }
               else pw_vector(pw,ex0,yzero-ploteta[i],x,yzero-ploteta[i],PIX_SRC,1);
            }
            oploteta[i]=ploteta[i];
            if(i==(last-1))ex0=x;
         }
         y0[i]=yy;
         yzero+=yincr;
      }
      trigged=0;
      x0=x;
      if(++xcounts>=plotx) {xcounts=0; x+=dec;}
      if(x>=xshow) {
         pw_text(pw,BEGINTRACE,ysize-1,PIX_SRC,NULL,ttime);
         pw_show(pw);
         if(x>=XSIZE) {
            x=BEGINTRACE;
            x0=BEGINTRACE;
            xshow=BEGINTRACE+swidth;
            pw_writebackground(pw,BEGINTRACE,0,swidth+10,ysize,PIX_SRC);
         }
         else {
            xshow+=swidth;
            pw_writebackground(pw,x,0,swidth+10,ysize,PIX_SRC);
         }
      }
      (void)notify_dispatch();
   }
   if(intr) {
      notify_dispatch();
```

```
        pw_batch_off(pw);
        endit=1;
        window_main_loop(frame);
     }
     close(dev);
}

static short *ibuffr;

int getsweep(buffer,numch,decim,type)
   short *buffer;
   int numch,decim,type;
{
   register i,ii;
   int got,next;

   ptr+=numch*decim-numch;
   for(i=0;i<numch;i++) {
      if(ptr<maxptr)buffer[i]=ibuffr[ptr++];
      else {
         got=read(dev,buf,numget);
         if(intr==0)return(EOF);
         ibuffr=(short *)buf;
         if(got<numget) {
            if(type<FILESIN) {
               got=ioctl(dev,SIPIOC_GETERRC,&sen);
               if(sen.siperr_sensekey ==4) {
                  printf("  AtoD buffer overflow. Resetting.\n");
                  got=ioctl(dev,SIPIOC_REZERO);
                  if(got== -1) perror("plotatod.rezero");
               }
            }
            else return(EOF);
         }
         ptr=ptr-maxptr;
         maxptr=(got/2);
         buffer[i]=ibuffr[ptr++];
      }
   }
   return(0);
}

/*---------------------------- trigger -------------------------------*/

trigger() {                    /* calc etas, station and master triggers */
   register i, s;              /* return: 1=master trigger, 0=no trigger */
   int eta, sbar, rbar, new_sbarbar, new_rbarbar, thresh;

   if(firsttime){
      for(s=first-1; s< last; s++) {
         sbarbar[s]=ssum[s]/ns_sweep;
         rbarbar[s]=rsum[s]/ns_sweep; }
      firsttime=0;
      oploteta[s]= MINETA*(yincr-BOUNDS)/(MAXETA-MINETA);
      return(0); }


   trigged=1;
   for(s=first-1; s< last; s++) {
      sbar=ssum[s]/ns_sweep;
      rbar=rsum[s]/ns_sweep;

      eta = rbar -
            (eta_num * rbarbar[s]) / eta_den -
```

```
            abs(sbarbar[s]-sbar) - eta_con;

      new_sbarbar = (sbar  +  7 * sbarbar[s]) / 8;  /* weighted avg   */
      if(new_sbarbar == sbarbar[s]){                /* if no change   */
         if(sbar > new_sbarbar)new_sbarbar++;       /* favor sbar...  */
         if(sbar < new_sbarbar)new_sbarbar--; }     /* over long avg  */

      new_rbarbar = (rbar  +  7 * rbarbar[s]) / 8;  /* weighted avg   */
      if(new_rbarbar == rbarbar[s]){                /* if no change   */
         if(rbar > new_rbarbar)new_rbarbar++;       /* favor sbar...  */
         if(rbar < new_rbarbar)new_rbarbar--; }     /* over long avg  */

      sbarbar[s] = new_sbarbar;
      rbarbar[s] = new_rbarbar;

      if(sta_trigger[s])sta_trigger[s]--;           /* age trigger    */
      if(eta > 0)sta_trigger[s] = nsw_aperture;     /* set statn trig */

      etas[s]=eta; sbars[s]=sbar; rbars[s]=rbar;    /* save for diagn */
      ploteta[s]=eta;
      if(ploteta[s]< MINETA)ploteta[s]=MINETA;
      if(ploteta[s]> MAXETA)ploteta[s]=MAXETA;
      ploteta[s]=ploteta[s]*(yincr-BOUNDS)/(MAXETA-MINETA);
   }


   /* loop thru subnet vector looking for a particular subnet
    * whose number of triggered stations is >= to thresh for that net.
    * Structure of subnet where thresh=threshhold and sn=station num is:
    * <thresh> <sn> <sn> ... -1 <thresh> <sn> <sn> ... -1 ... -1 -1
    * ie, thresh-sn groups separated by -1 all terminated by -1 -1
    */

   i=0;
   while(    (thresh  = subnet[i++])  != -1 )
      while( (s       = subnet[i++])  != -1 )
         if( (thresh -= sta_trigger[s] >  0 )  <= 0)return(1);

   return(0);
}

int trig_on(){};
```