DTI-UM-176Ø-4

PRICE: $25.00

USER MANUAL

FOR

DATA TRANSLATION, INCORPORATED

DT1760 SERIES

INTERFACE SUBSYSTEMS

DT1761 DT1762

DT1764 DT1765

DATA TRANSLATION
4 STRATHMORE ROAD
NATICK. MASSACHUSETTS. 01760.

Information furnished by Data Translation is believed to be accurate and reliable. However, no responsibility is assumed by Data Translation for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Data Translation, Inc.

August 1978

TABLE OF CONTENTS

Chapter 1 Introduction

Chapter 2 Analog Input Specifications

Chapter 3 Analog Output Specifications

Chapter 4 Interface Register Description

Chapter 5 Operation and Programming

Chapter 6 Configuration

Chapter 1

INTRODUCTION

1.1 Survey of DT176Ø series

The DT176Ø series are advanced, standardized, low cost data acquisistion systems designed for users of the Digital Equipment Corporation LSI-11 micro-computer series. These data acquisition systems are on standard quad-size boards that plug into the LSI-11, PDP-11/Ø3 or Heath H11 backplanes.

The DT1762 data acquisition system can process up to 64 channels of analog data at a throughput rate of 25,ØØØ channels per second. Throughput rates of1ØØ,ØØØ channels per second are also available. The DT1764 offers the sameanalog input capability as the DT1762 but accepts wide range signal inputs down to 3uV resolution. The DT1761 analog input/output system offers 16 single ended A/D input channels and 2 D/A output channels - each D/A with a power amplifier output for driving industrial loads. The DT1765 offers the analog I/O capability of the DT1761 but accepts wide range A/D inputs down to 3uV resolution. A complete DMA (Direct Memory Access) interface is available on all models.

These systems contain several standard features that are usually found only in much higher priced systems. Included as standard are acquisition of data on an external event and a complete data acquisition system containing a fully protected input analog multiplexer, sample/hold amplifier, and an analog-to-digital converter. The entire system is powered from the computer + 5 volt power supply through a highly isolated DC-DC converter contained on the interface system. Full interrupt control and optional DMA interface provide the user with full system compatibility under RT-11 and RSX-11 operating system environments.

1.2 Product Line

The DT176Ø series product line encompasses a wide range of analog requirements. Members of this series include the following interfaces:

TYPE	MODEL	DESCRIPTION
		(Types of outputs, number of input channels)
High Level Analog Input Systems	DT1762	8 DI or 16 SE 16 DI, 32 DI, 32 SE, or 64 SE with on-board expander module
Wide Range Analog Input Systems	DT1764	8 DI or 16 SE 16 DI, 32 DI 32 SE, or 64 SE with on-board expander module Range 1ØmV to 1Ø volts
High Level Analog I/O Systems	DT1761	8 DI or 16 SE 2 D/A with point plotting capability
Wide Range Analog I/O Systems	DT1765	8 DI or 16SE Range 1ØmV to 1Ø volts 2 D/A with point plotting capability

Note: ALL UNITS ARE 12 BIT RESOLUTION.

1.3 OPTIONS

All Models	
- SE	single ended input
- DI	differential input
- 8, - 16	number of input channels
- FD	diagnostic software on floppy diskette rather paper tape
- DMA	DMA hardware
Models	DT1761 DT1762
- PG	programmable gain amplifier
- C	100 KHZ A/D throughput
- A	125 KHZ A/D throughput
- M	1-5V input (4-20 mA)
Models	DT1764 DT1765
- R	precision resistor kit included
Models	DT1762 DT1764
-16, -32, -64	Number of input channels
Models	DT1761 DT1765
- 0	Analog Output only (no A/D input)
Cables	Order DT186916-2 cable assembly for a 10 foot ribbon cable with connector. Mates with DT176Ø series, One used for Models DT1761 and DT1765, Models DT1762 and DT1764 require 2.

Chapter 2

ANALOG INPUT SPECIFICATIONS

2.1 Explanation of Input Types

Data Translation offers the user three distinct types of analog inputs. High level systems are available where the input range can be selected to be Ø-5V, ±5V, Ø-1ØV, or ±1ØV (optional programmable gain amplifier extends dynamic range of system to 14 bits). The wide range input systems offer user-selectable gain, allowing input ranges from 1ØmV full scale to 1ØV full scale, depending on the choice of a single external gain resistor. Finally, 1-5 Volt (4-2ØmA) inputs are provided for industrial loop control applications (requires customer installation of a resistor for each channel - see section 2.5).

The customer must select one of these three input types at the time of purchase. It is not possible to field-change the input type (although units may be returned to factory for such a change on special arrangement).

2.2 High Level Analog Input Specifications: (models DT1761 and DT1762)

NUMBER OF CHANNELS:	DT1761 8 DI or 16 SE with 2 D/A channels DT1762 8, 16, or 32 DI channels 16, 32, or 64 SE channels
MUX INPUT CHARACTERISTICS	Break before Make on new channel selection ± 35 volts overvoltage protection
INPUT IMPEDANCE:	1ØØ megohms in "OFF" or "ON" position of multiplexer
A/D CONVERTER RESOLUTION:	12 bits binary weighted
CODING:	Straight binary,offset binary, or 2's complement, jumper selectable.
RANGES:	Ø to 5 volts, ± 5 volts, Ø to 1Ø volts, or ± 1Ø volts. All jumper selectable.

PERFORMANCE:

a.	Gain Error	Adjustable to zero
b.	Offset Error	Adjustable to zero
c.	Linearity	± ½ LSB
d.	Accuracy	± Ø.Ø3% FSR
e.	Stability	± 25 ppm/°C FSR
f.	Throughput	35 KHz Standard 1ØØ KHz Option "C" 125 KHZ Option "A"

Note: ALL HIGH LEVEL ANALOG INPUT INTERFACES MUST BE ORDERED AS SE OR DI. THEY ARE NOT FIELD SELECTABLE.

2.3 Wide Range Analog Input Specifications: (models DT1764 and DT1765)

NUMBER OF CHANNELS:	DT1764 8, 16, or 32 DI channels 16, 32, or 64 SE channels DT1765 8 DI, or 16 SE with 2 D/A channels
MUX INPUT CHARACTERISTICS:	Inputs are protected to ± 15 volts non-destructive.
INPUT IMPEDANCE:	100 megohms // 10 pf "OFF" 100 megohms // 60 pf "ON"
BIAS CURRENT:	20nA @ + 25°C. 50nA @ + 75°C.
A/D CONVERTER RESOLUTION:	12 bits binary weighted
CODING	Straight binary,offset binary, or 2's complement,jumper selectable.
RANGE:	0 to 10mV, 0 to 10V (or ±10mV, ±10V) These ranges represent the minimum and maximum input ranges. The full spectrum of intermediate ranges is given in table 2-1. All ranges are selected via a single resistor and timing capacitor on the interface.
PERFORMANCE:	
a. Gain Error	Adjustable to zero
b. Offset Error	Adjustable to zero
c. Linearity	± ½ LSB
d. Accuracy	± 0.03% FSR @ gain equal to 1
e. Stability	± 30 ppm /°C
f. Throughput	31 KHz @ a gain of 1 to 10. For higher gains with corresponding throughput rates see table m.

Note: ALL WIDE RANGE ANALOG INPUT INTERFACES CAN BE FIELD CONFIGURED TO BE SE OR DI BY THE USER VIA JUMPERS PROVIDED ON THE INTERFACE BOARD.

2.4 Instrumentation Amplifier

The wide range interface systems (models DT1764 and DT1765) have an on-board high-precision variable gain instrumentation amplifier. As the gain of the amplifier is increased, the time allowed for settling of the multiplexer, sample/hold and amplifier must also be increased. The gain (and hence the input range) is set with the use of an external gain resistor (R_g). The time allowed for settling is set by an external capacitor (C_t). If neither of these two parts (R_g and C_t) are installed, the instrumentation amplifier will operate at a gain of 1 (representing a Ø-1ØV or ±1ØV input range) and the proper settling time will be allowed. To select the gain required and the value of the gain resistor, the following formulas are useful: (the gain resistor must be a precision resistor with ±Ø.Ø2% tolerance)

$$G = \frac{\text{1ØV}}{V_{\text{input range desired}}} = 1 + \frac{20 \times 1\text{Ø}^3}{R_g}$$

$$R_g = \frac{2Q \times 1\text{Ø}^3}{G-1}$$ (for a gain of 1, no resistor is required)

The following table illustrates the values of R_g and C_t required for various gains and amplifier settling times. It should be noted that the total throughput time is the time required for amplifier settling plus the 2ØuSEC required for the A/D conversion.

INPUT RANGE	GAIN	R_g	C_t	AMP SETTLING TIME	SYSTEM ACCURACY	THROUGHPUT (SAMPLES/SECOND)
± 1Ø mV	1ØØØ	2Ø.Ø2	Ø.Ø15uf	25ØuS	±Ø.1%	3.7 KHz
± 25 mV	4ØØ	5Ø.13	68ØØpf	12ØuS	±Ø.Ø8%	7.1 KHz
± 5Ø mV	2ØØ	1ØØ.5	33ØØpf	7ØuS	±Ø,Ø7%	11.1 KHz
± 1ØØ mV	1ØØ	2Ø2.Ø	15ØØpf	4ØuS	±Ø.Ø5%	16.75 KHz
± 1 volt	1Ø	2222	None	12uS	±Ø.Ø3%	31.Ø KHz
± 2.5 volts	4	6667	None	12uS	±Ø.03%	31.Ø KHz
± 5.Ø volts	2	2Ø.ØK	None	12uS	±Ø.Ø3%	31.Ø KHz
± 1Ø.Ø volts	1	None	None	12uS	±Ø.Ø3%	31.Ø KHz

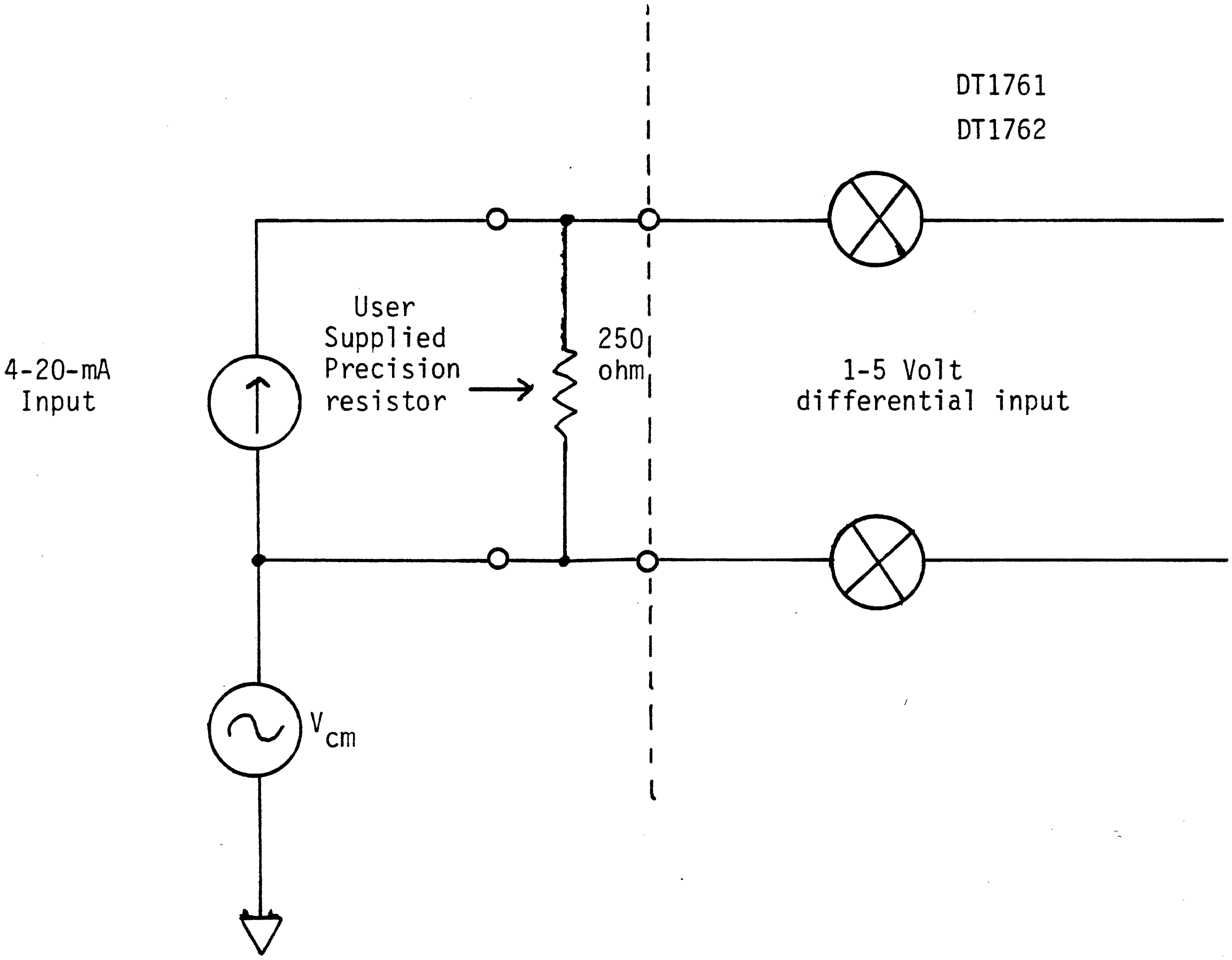
Note: Throughput time = Amplifier Settling time + A/D conversion time (2Ø uSEC)

$$\text{Throughput} = \frac{1}{\text{throughput time}}$$

Table 2-1

2.5 4-2ØmA Analog Input Option

Data Translation analog input systems can be configured for 4-2ØmA loop control inputs. This is accomplished by ordering the 1-5V input option (models DT1761 and DT1762 only). The user must provide a 25Ø ohm resistor to convert the 4-2ØmA input to a 1-5 volt signal. This resistor should be a precision resistor with a ± 0.02% tolerance. All 1-5 volt inputs are supplied in the differential configuration. The 4-2ØmA input option may only be factory-installed - it may not be field-applied.



NOTE: V_{cm} (common mode voltage) should be $\leq$ 0.1V

SAMPLE 4-2Ø MA INPUT CODES

	ADDBR		
OCTAL	BINARY	Vin	Iin
7777	111111111111	1.ØØØØ	4.ØØØmA
7776	11111111111Ø	1.ØØ1Ø	4.ØØ4mA
7775	1111111111Ø1	1.ØØ2Ø	4.ØØ8mA
7773	111111111Ø11	1.ØØ39	4.Ø15mA
7767	11111111Ø111	1.ØØ78	4.Ø31mA
7757	1111111Ø1111	1.Ø156	4.Ø62mA
7737	111111Ø11111	1.Ø313	4.125mA
7677	11111Ø111111	1.Ø625	4.25ØmA
7577	1111Ø1111111	1.125Ø	4.5ØØmA
7377	111Ø11111111	1.25ØØ	5.ØØØmA
6777	11Ø111111111	1.5ØØØ	6.ØØØmA
5777	1Ø1111111111	2.ØØØØ	8.ØØØmA
3777	Ø11111111111	3.ØØØØ	12.ØØØmA
1777	ØØ1111111111	4.ØØØØ	16.ØØØmA
Ø777	ØØØ111111111	4.5ØØØ	18.ØØØmA
Ø377	ØØØØ11111111	4.75ØØ	19.ØØØmA
Ø177	ØØØØØ1111111	4.875Ø	19.5ØØmA
ØØ77	ØØØØØØ111111	4.9375	19.75ØmA
ØØ37	ØØØØØØØ11111	4.9688	19.875mA
ØØ17	ØØØØØØØØ1111	4.9844	19.938mA
ØØØ7	ØØØØØØØØØ111	4.9922	19.969mA
ØØØ3	ØØØØØØØØØØ11	4.9961	19.984mA
ØØØ1	ØØØØØØØØØØØ1	4.998Ø	19.992mA
ØØØØ	ØØØØØØØØØØØØ	4.999Ø	19.996mA

Chapter 3
ANALOG OUTPUT SPECIFICATIONS

3.1 Dual Channel D/A Specifications

The DT1761 and DT1765 offer the user a sophisticated dual channel D/A Converter. The converter and its associated interface circuitry simplify a number of control tasks. A complete guide to the use of the analog output circuitry is given in Chapter 5.

RESOLUTION:	12 bit binary weighted
LINEARITY:	± ½ LSB
RANGES:	Ø to 1Ø volts, ± 1Ø volts; @ 25 mA minimum output current, all ranges jumper selectable
RELATIVE ACCURACY:	± Ø.Ø25%
SETTLING TIME:	Ø.1% in 1 uS, Ø.Ø1% in 3 uS driven into 50 ft. coaxial cable terminated with 47Ø ohms.
TEMPERATURE COEFFICIENT:	25 ppm/°C.
Z AXIS CONTROL:	The DT1761 and DT1765 contain all the control circuitry for Z axis and scope control mode bits.
Z OUTPUT (INTENSIFY):	TTL compatable pulse driven into 5Ø ohm termination. Normally LOW (max. Ø.8 volts), generates HIGH (min. 2.4 volts) pulse on intensify.
Z RISETIME:	1ØØ nS into 5Ø ft. of coaxial cable terminated with 47Ø ohms.
Z PULSE WIDTH:	Ø.5 uS or 5 uS jumper selectable. External capacitor supplied by the user can lengthen the pulse width from Ø.2 uS to 1ØØ uS.
SET-UP DELAY:	3 uS or 7Ø uS jumper selectable. External capacitor supplied by the user can lengthen the delay from 1 uS to Ø.5 mS.
GAIN & OFFSET:	Both outputs adjustable with external pot.
MODE BITS:	Two input bits are decoded to four mode control output lines. The output lines are LOW true and can drive up to 1Ø standard TTL loads (consult Chapter 5 for application information).

Chapter 4

COMPUTER INTERFACE REGISTERS

4.1 Introduction

The interface registers of the DT176Ø series data acquisition systems are designed to meet the requirements of standard DEC PDP-11 interfaces. As such, they are structured around a Control and Status register for complete software control of the interface. The registers are standardized, providing a common register set among the entire series. This chapter describes in detail the bit definitions of this register set.

4.2 Summary of Registers

The following is a brief summary of the registers common to the DT176Ø series interfaces. The term "BASE" refers to the device address (see Chapter 6).

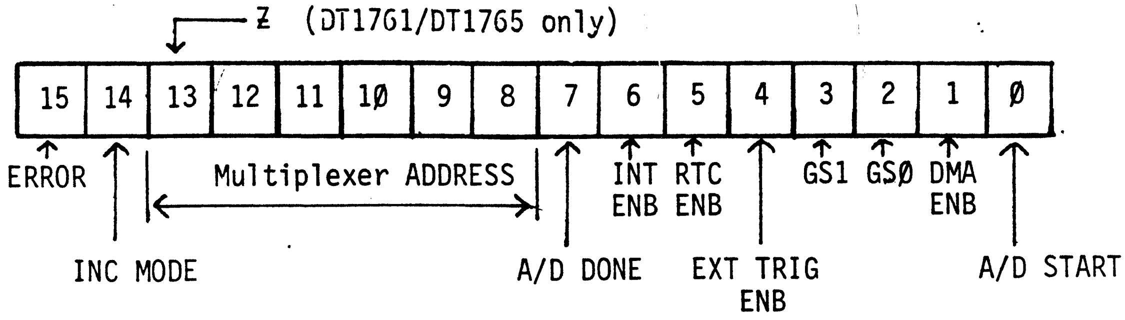
Register Name	Address	Comments
A/D Control-Status Register (ADCSR)	BASE	R/W, byte addressable all models
A/D Data Buffer Register (ADDBR)	BASE + 2	Read only, word addressable all models
D/A Data Buffer Register (DACDBR)	BASE + 2	Write only, word addressable DT1761 and DT1765 only
DMA Word Count Register (DMWCR)	BASE + 4	R/W, word addressable DMA option only, all models
DMA Current Address Register (DMCAR)	BASE + 6	R/W, word addressable DMA option only, all models

4.3 Detailed Register Description

The following sections describe the interface registers in detail.

Control - Status Register   Base address

ADCSR



Upper Byte (loading this byte will trigger a conversion - see section 5.1)

ERROR bit (bit 15)   READ/WRITE TO ZERO

This bit indicates an error condition on the board when set. The following error conditions will set this bit:

- an attempt to start a new conversion during the multiplexer settling interval. This can be due to external inputs (external trigger or Real Time Clock inputs) or program attempts to start a conversion.
  Note: Selecting Ext. trigger or RTC trigger modes with the respective input floating (not tied to a digital high or digital low) can generate this error.
- an attempt to start a new conversion while a previous conversion was still in progress. This error can be generated in all triggering modes.
- a read of the A/D data register while a previous conversion was still in progress. Invalid data will be read if this is done, hence the error indication.

This bit may be cleared in two different ways. The backplane INIT signal (generated by power-up or execution of a RESET instruction) will clear this bit, as will writing a zero into this bit position. (example: CLR @ADCSR
MOV #Ø, @#ADCSR
note that either of these two examples will reset the error bit but will also start a new conversion by loading an address into the upper byte of the CSR)

INC MODE (bit 14) READ/WRITE

When this bit is set the A/D converter is placed in increment mode. When in this mode, every start conversion command will increment the multiplexer address automatically. Thus, the program can sample channels sequentially without the requirement of a software maintained channel address counter. The multiplexer address is incremented before the conversion is initiated. This means that if the multiplexer address is read back upon completion of a conversion initiated under increment mode the address bits will indicate the channel address from which the waiting data was obtained. Increment mode can be used under all triggering modes. See Chapter 5 for information on how to utilize increment mode. The bit can be set and reset under program control, and is cleared by the INIT signal.

Ƶ (bit 13) READ ONLY (DT1761/DT1765 only)

This bit is internally connected to the Ƶ axis intensify output of the DT212 dual DAC module. During a Ƶ axis intensify pulse this bit will read as a one.

MUX ADDRESS (bits 8-13 READ/WRITE

These six bits provide the means for addressing up to 64 analog input channels. On the DT1762 and DT1764, all six of these bits are used to address the input channel desired. On the DT1761 and DT1765, only the low order 4 bits (bits 8-11) of this field are required to address the maximum of 16 analog input channels available on these models. CSR bit 12 becomes a read-only bit, always reading as a zero. CSR bit 13 is utilized differently - See description of Ƶ bit above.

The multiplexer address bits are read/write bits. Whenever a new address is written into these bit positions, a conversion is automatically initiated on that channel. Thus, any operation that writes into the upper byte of the CSR will start a new conversion. See Chapter 5 for information on how to utilize this feature to best advantage. Whenever a conversion is completed, a read of the multiplexer address will always indicate the channel from which the waiting A/D data was obtained.

Lower Byte

A/D DONE (bit 7) READ ONLY

This bit is set when the A/D module signals that the End of Conversion (EOC) has occurred. The bit remains set until the A/D data is read, at which time the bit is cleared. Thus, the A/D done bit indicates that valid data is available to the program. This bit is also cleared by the INIT signal.

INT ENB (bit 6) READ/WRITE

This bit when set enables interrupts to be generated upon either the A/D DONE bit or the ERROR bit becoming set. The A/D done and the error interrupts use the same interrupt vector (see Chapter 6 for information on setting the interrupt vector). This bit can be set and reset under program control, and is cleared by the INIT signal.

RTC ENB (bit 5) READ/WRITE

This bit when set enables the A/D converter to be triggered from the Real Time Clock input on connector J1. If a Real Time Clock is connected to the RTC input, the user may then control the converter through the Real Time Clock. This can be set and reset under program control, and is cleared by the INIT signal.

EXTTRIG ENB (bit 4) READ/WRITE

This bit is similar in operation to the RTC ENB bit but instead allows the A/D converter to be triggered by the External Trigger input. This bit can be set and reset under program control and is cleared by the INIT signal.

GAIN SELECT (bits 2-3) READ/WRITE

GS1 bit 3
GSØ bit 2

These bits function only on boards equipped with the programmable gain option. With the option installed, these bits control the gain of the switch gain amplifier and thus the gain of the converter. The gain can be selected according to the following table:

GS1	GSØ	GAIN
Ø	Ø	1
Ø	1	2
1	Ø	4
1	1	8

The gain bits can be set and reset under program control, and are cleared by the INIT signal..

DMA ENB (bit 1) READ/WRITE

This bit controls the operation of the Direct Memory Acess (DMA) option if installed. On boards with the DMA option, setting this bit instructs the board to transfer A/D data directly into memory using DMA. Before setting this bit the two DMA registers should be set correctly with the transfer parameters. See Chapter 5 for information on using this very useful option. The DMA ENB bit can be set and reset under program control. It is automatically cleared when the given transfer is complete. It is also cleared by the INIT signal.

A/D Start (bit Ø) WRITE ONLY

This bit is a write only bit and will always read back as a zero. Writing into the CSR with this bit set will initiate a conversion. See Chapter 5 for information on triggering conversions.

A/D Data Buffer Register — Base address + 2

ADDBR READ ONLY — This register is not byte-addressable.

This location will return the A/D data as follows when read:

15	14	13	12	11	1Ø	9	8	7	6	5	4	3	2	1	Ø
not used *				MSB					A/D DATA						LSB

D/A Data Buffer Register — Base address + 2

DACDBR WRITE ONLY — This register is not byte-addressable

15	14	13	12	11	1Ø	9	8	7	6	5	4	3	2	1	Ø
X-Y SEL	Z INTENSIFY	MODE 1	Ø	MSB					DAC DATA						LSB

*These bits are not driven on the bus by the interface, thus they are indeterminate.

X-Y (bit 15)

When set DAC data is loaded into Y-DAC
When clear DAC data is loaded into X-DAC

Z INTENSIFY (bit 14)

When set will trigger a Z-axis intensify output.

MODE (bits 13-12)

These two bits are decoded to produce four digital outputs according to the following table:

MODE	MODE	Connector Pin			
1	Ø	43	45	47	49
Ø	Ø	Ø	1	1	1
Ø	1	1	Ø	1	1
1	Ø	1	1	Ø	1
1	1	1	1	1	Ø

DAC DATA (bits 11-Ø)

Twelve bits of DAC data.

DMA option registers
When the DMA option is installed, the following registers have meaning.

DMA Word Count
DMWCR

BASE address + 4
This register is not byte-addressable.

15	14	13	12	11	1Ø	9	8	7	6	5	4	3	2	1	Ø
not used				WORD COUNT DATA											

The word count register is a 12 bit counter/ register which determines the number of transfers that will occur. The register is loaded with the one's complement of the desired transfer count. This register is incremented after every transfer. When the transfer is completed an End of Range (EOR) interrupt will be generated. For a complete description of DMA operation, see Chapter 5. Upon readback of this register, the upper four bits of this register are indeterminate. The low order 12 bits can be set and reset under program control, and are also modified by the DMA logic as the transfer progresses.

DMA Current Address Register
DMCAR

Base address +6
This register is not byte-addressable

15	14	13	12	11	1Ø	9	8	7	6	5	4	3	2	1	Ø
CURRENT ADDRESS															

The current address register is used to supply the actual memory location where data is to be transferred. Before initiating a DMA transfer this register is loaded with the first location of the memory block to be filled. The register is incremented by the DMA logic after every word transferred. Due to the fact that the A/D data is 12 bits in length, a complete PDP-11 word is transferred for every conversion. The user program may write either even or odd addresses into this register, but the logic will strip bit Ø and start its transfer at an even address. Upon read-back of this register, bit Ø will be indeterminate. The 15 bits of this register may be set and reset under program control, and are also modified by the DMA logic.

Chapter 5

OPERATION AND PROGRAMMING

5.1 Triggering Modes

There are several means of initiating an A/D conversion on the DT176Ø series, giving the user great flexibility. The following conditions will trigger the start of an A/D conversiun:

A. Loading a Multiplexer Channel Address (upper byte of ADCSR) Whenever a new channel address ia loaded, the board will automatically initiate a conversion after the multiplexer has settled. A new address may be loaded in several ways:

```
MOVB  #channel, @#ADCSR+1; loads channel address in upper
                         ; byte of ADCSR, starts conversion
MOV   #csr,@#ADCSR       ; loads an entire new CSR
CLR   @#ADCSR            ; clears ADCSR, starts conversion
                         ; on channel zero.
CLRB @#ADCSR+1           ; clears only channel address, starts
                         ; conversion on channel zero.
```

If the user desires to start a conversion without loading a new channel address, the program should use a variation of (B), below.

B. Setting bit Ø (A/D START) of ADCSR
This technique can be applied in several ways:

```
MOVB  #1Ø1, @#ADCSR.   ; enable interrupts, start conversion,
                       ; but do not load new channel
MOV   #1Ø1, @#ADCSR    ; enable interrupts, start conversion
                       ; on channel zero
(Note that the last example is equivalent to
    MOV #1ØØ, @#ADCSR   due to (A), above)
```

To start a conversion without changing any of the parameters already loaded into ADCSR, use

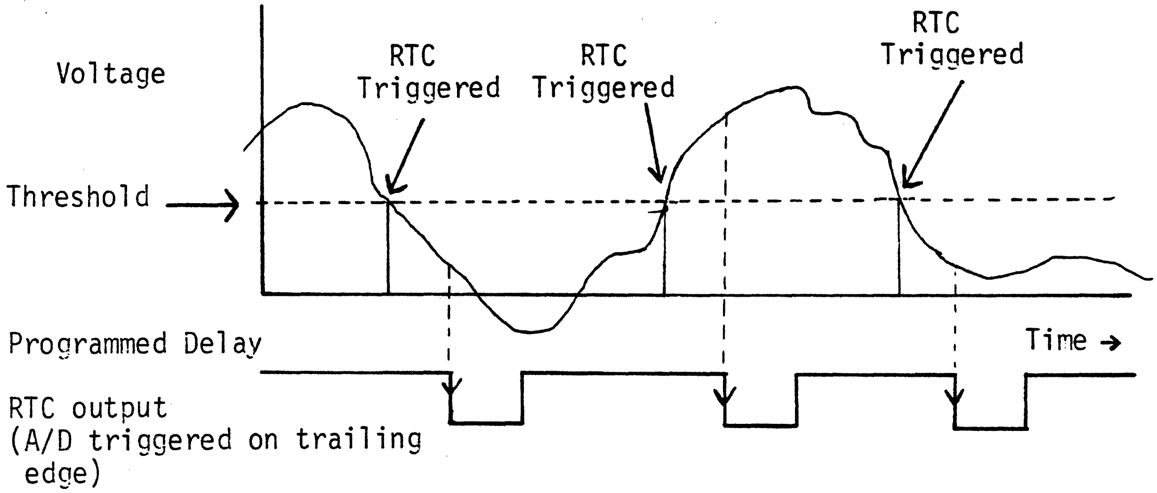
```
INCB   @#ADCSR
```

This sets the start bit without modifying any other bits in the CSR and without loading a channel address. If INC were to be used instead of INCB, the upper byte of ADCSR would be read and reloaded during the execution of the INC instruction. Under some conditions this may cause an error. Consequently, the use of INC is discouraged and INCB is recommended.

C. Real Time Clock input when RTC ENB bit is set, and
External Trigger input when EXTTRIG ENB bit is set

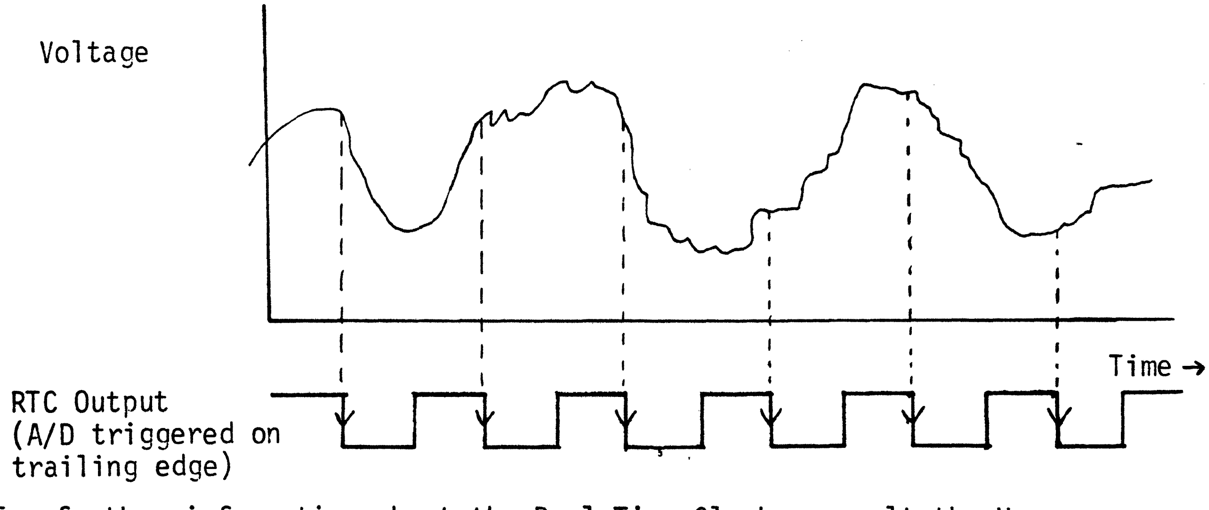
When the RTC ENB bit is set, the operation of the converter is under the control of the Real Time Clock input, a TTL compatible input. The trailing edge of this input will switch the analog sample and hold to hold, initiating a conversion. A typical application of this input would be to connect a waveform to the Real Time Clock, triggering the RTC at a certain threshold level. After a programmed time interval, the RTC would then generate a pulse to trigger the A/D converter. (Note: the pulse output of the DEC KWV11A is too narrow to be used directly.)

Delayed Trigger Application:



Another application that is perhaps more common is utilizing the RTC to generate a software controlled clock. If the A/D converter is triggered by this controlled frequency clock, an incoming analog waveform can be digitized to any desired degree of accuracy (limited only by the sampling frequency and the A/D throughput). Digital filters are greatly simplified using this combination of the RTC and an A/D converter.

Digital Filter Application:



For further information about the Real Time Clock, consult the User Manual for the Data Translation model DT2769, a dual-width LSI-11 interface providing complete compatability with DEC's KWV11-A option.

The EXTTRIG ENB bit and the EXTERNAL TRIGGER input function in an identical manner to the RTC input and enable bit. The purpose of the TTL compatible external trigger input is to provide the user with yet another means of triggering the converter. A typical application of the external trigger input is to synchronize the converter with some external event (s). If an RTC board is available in the system, the Schmitt Trigger inputs of the RTC may be used to detect a preset threshold voltage and immediately issue a TTL trigger to the A/D converter.

Before enabling RTC or external trigger mode the user program must set up the channel address parameters on the board (selecting the channel and sequential mode if desired). Setting up these parameters generally requires performing an A/D conversion, as loading a new address will start a conversion. Once the board has been instructed as to the proper channels, the RTC ENB bit or the EXTTRIG ENB bit should be set, along with any other mode bits that are appropriate (such as the gain setting, interrupt enable, DMA enable, etc. as desired).

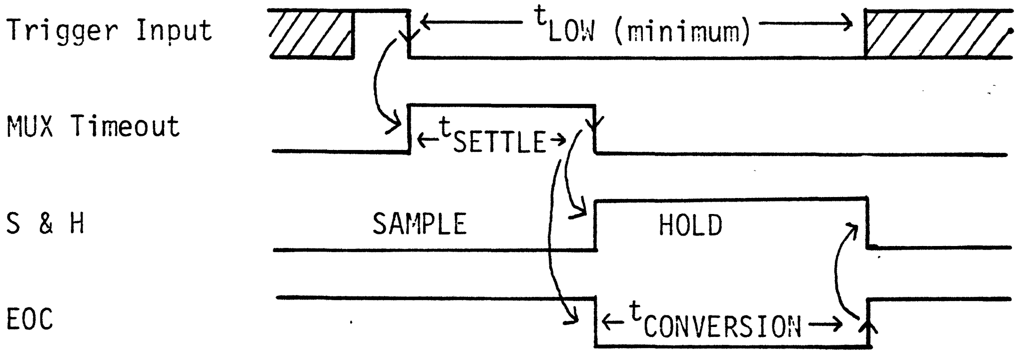
```
For RTC mode:
MOVB   #4Ø,@#ADCSR    ; set RTC bit, clear other bits
MOVB   #14Ø,@#ADCSR   ; set RTC bit and enable interrupts
BISB   #4Ø,@#ADCSR    ; set RTC bit, leave others alone

For External trigger mode:
MOVB   #2Ø,@#ADCSR    ; set EXTTRIG bit, clear others
MOVB   #12Ø,@#ADCSR   ; set EXTTRIG bit, enable interrupts
BISB   #2Ø,@#ADCSR    ; set EXTTRIG bit, leave others alone
```

Note that all of the examples are "byte" instructions, operating only on the lower byte of ADCSR. The use of "byte" instructions will prevent modification of the upper byte of ADCSR, keeping unwanted triggers from interacting with the RTC or External Trigger inputs. Take care in using the BISB method, as it is possible to generate an unwanted combination of modes (such as RTC and interrupt, RTC and EXTTRIG, etc.)

Once the RTC ENB or EXTTRIG ENB bit is set, an A/D conversion will be triggered on every negative-going edge of the respective inputs. If both inputs are in use simultaneously, (both RTC ENB and EXTTRIG ENB are set), the inactive input (i.e. the input that did not trigger the current conversion) must remain in the low state until the conversion is finished. The RTC input and the EXTTRIG input are subject to the following constraints:

Under sequential mode:



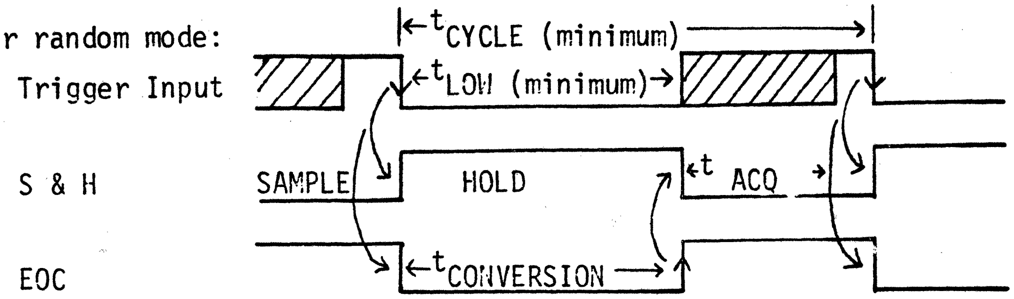
t_{SETTLE} = multiplexer settling time

$t_{CONVERSION}$ = A/D conversion time

(see Appendix C for numerical values for these two parameters)

t_{LOW} (minimum) = $t_{SETTLE} + t_{CONVERSION}$, a low time less than or equal to this value will set the ERROR bit

Under random mode:



$\leftarrow t_{CYCLE}$ (minimum) $\rightarrow$
Trigger Input
$\leftarrow t_{LOW}$ (minimum) $\rightarrow$
S & H
SAMPLE
HOLD
$\leftarrow t$ ACQ $\rightarrow$
EOC
$\leftarrow t_{CONVERSION} \rightarrow$

t_{ACQ} = Sample & Hold acquisition time

$t_{CONVERSION}$ = A/D conversion time

(see Appendix C for numerical values for these two parameters)

$t_{LOW\ (minimum)} = t_{CONVERSION}$, a low time less than or equal to this value will set the ERROR bit

$t_{CYCLE\ (minimum)} = t_{ACQ} + t_{CONVERSION}$, a cycle time less than or equal to this value will result in incomplete sample & hold acquisition

5.2 Channel Addressing Modes

The DT176∅ series analog boards offer the user two methods of specifying input channel addresses to the A/D converter. The desired channel address can be directly loaded or the multiplexer auto-increment method can be selected (sequential mode).

A. Direct Loading

Utilizing this mode of addressing, the desired channel address along with any appropriate mode bits is loaded into ADCSR. Examples of this method and a discussion of triggering considerations are given in the Triggering Modes section in the paragraph on loading a multiplexer address. Complete random access to any channel is possible with direct loading of the address.

B. Sequential Mode

Many applications for data acquisition systems require the periodic scanning of all or a group of input channels. To simplify software requirements in programming for these applications, the DT176∅ series allow the program to set a bit (INC MODE) in the CSR to allow the hardwired logic on the board to cycle through input channels automatically.

Once the INC MODE bit is set to allow sequential mode and an initial channel address is specified, every new trigger will increment the multiplexer channel address and perform a conversion on the new channel. When the multiplexer reaches the last channel installed on the board, the channel address is forced to zero and the cycle continues. With this mode of operation, valuable CPU time is not required to update a software channel address counter and direct-load a new address for every conversion. Sequential mode can be used in combination with any triggering and data transfer mode desired (DMA, interrupts, RTC, etc.).

One concern for the user application software is the need to know on what channel waiting A/D data was taken. To meet this need, the board CSR always contains the address of the channel on which the current valid A/D data was taken. The multiplexer address is not incremented until the next trigger command is received, at which time the logic will automatically wait a preset time for the analog front end to settle before triggering the conversion. The need for a software channel address counter is completely obviated by this feature.

Configuring the board for sequential mode, although not completely straight forward, is simple. The procedure is slightly complicated by the fact that the INC MODE bit (ADCSR bit 14) is located in the upper byte of the CSR, and any modifications to the upper byte will trigger a conversion. Before setting the INC mode bit, the initial channel of the scan must be selected. To prepare the board, load a channel address corresponding to the channel immediately preceding the desired start channel (if the first conversion of the scan is to be on Channel 3, load channel address 2, etc). Do not set the INC MODE bit until the channel address in ADCSR is one less than the desired start address. After waiting for the converter to finish its current conversion (if any), execute

```
                        BIS  #4ØØØØ, @#ADCSR
```

setting the INC MODE bit. This will increment the channel address (to the desired start address) and start a conversion on that channel. Once sequential mode has been enabled, it is only necessary to trigger the converter with the A/D start bit or by an external trigger source to automatically step the multiplexer to the next channel and perform a conversion. Note: when preparing the board to be operated under sequential mode with triggering supplied by the RTC or EXT TRIG inputs, the channel address loaded should be two less than the desired start address.

5.3 Data Transfer Modes

There are several methods that may be utilized to transfer data to and from the DT176Ø series data acquisition systems. The methods and some programming examples follow.

A. Programmed I/O

All data transfers occur under explicit software control when using programmed I/O. Once the converter is triggered, the software must wait for the A/D DONE bit (bit 7 of the CSR) to become set before data can be obtained. The program is not informed of when the done bit becomes set - it must sample the CSR until the bit is set,

Example:

```
MOV  #4ØØ,@#ADCSR                   ;direct load channel address 1,
                                    ;trigger conversion
=user code=
; program is ready for A/D data, see if board is ready too
Loop: TSTB @#ADCSR                  ; is bit 7 set? (A/D DONE)
      BMI  GETDATA                  ; Yes - go get data
=user code (if any)=
BR LOOP
```

The TSTB instruction can be replaced by a BIT #2ØØ,@#ADCSR or a BITB #2ØØ, @#ADCSR. All test the state of the A/D done bit (make sure the proper conditional branch follows whatever test instruction is used). Before the program gets the A/D data, it is good practice to check the state of the ERROR bit (bit 15 of the CSR) to verify that the conversion was error-free. Once the program has determined that A/D data is ready, the software must move the data from the A/D data buffer register to whatever location is appropriate to the program.

```
GETDATA:        TST         @#ADCSR          : error bit set?
                BMI         ERROR            ; Yes - go process
                MOV         @#ADDBR,dst      ; get data
                =user code=
```

Programmed I/O may be used in conjunction with any triggering mode and any channel addressing mode.

B. Interrupt driven I/O

In many applications, the time consumed by the CPU in programmed I/O just waiting for the A/D converter is too valuable to waste. For these applications and applications that require an instant (real time response to the A/D data, interrupt mode is provided. In this mode the program not only selects the channel addressing mode and triggering mode desired but also sets bit 6 of the CSR, the INT ENB bit. Thereafter, an interrupt will be generated every time the A/D DONE bit (bit 7 of the CSR) or the ERROR bit (bit 15 of the CSR) become set. The program now does not need to loop waiting for the A/D DONE bit - instead the CPU can proceed to some other task. The program is assured of being informed as soon as the board requires attention. One interesting application of interrupt mode is when interrupts are used in conjunction with an external trigger source: after a conversion is triggered externally the A/D done condition generates an interrupt. This tells the processor that some external event has occurred and A/D data is ready - a dual use of the interrupt.

Before using interrupt mode, consult the processor handbook for the LS1-11 to become familiar with the DEC interrupt procedure. The DT176Ø series boards generate vectors that are completely DIP switch selectable - any desired vector can be set (see Chapter 6). The interrupt generated by the A/D DONE and ERROR bits uses the "base" vector.

The user interrupt service routine (ISR) must remember that an interrupt can also be generated by the ERROR bit. Typically the first function of the ISR is to check this bit - if clear, then the interrupt signals valid A/D data. If set, an error condition has occurred.

```
     Example:
     MOV        #14Ø,@#ADCSR          ;enable int., load address Ø,enable RTC
     =user code=
ISR: TST    @#ADCSR                   ; error bit set?
     BMI    ERROR                     ; yes - go process
     MOV    @#ADDBR,(R2)+             ; store A/D in memory
     RTI
```

In the example above, the program loads address zero, enables interrupts and the RTC trigger, and then proceeds to a different task. Thereafter, every time a conversion completes an interrupt will be generated and the service routine executed. In this example, the service routine checks the error bit to see if the interrupt was generated by an error condition, then moves the A/D data into memory using processor register 2 as a pointer. The interrupt service routine should end in an RTI instruction (exception: RT-11 users consult RT-11 manual for information on programming interrupt service routines.)

Interrupt driven I/O may be used with any triggering mode and either channel addressing mode.

C. Direct Memory Access (DMA)

A unique option for the DT176Ø series interfaces is the Direct Memory Access (DMA) option. Some applications such as high speed waveform analysis, digital filtering, etc. require the high speed acquisition of a number of continuous conversions. Programmed I/O and interrupt operation still require processor action for every conversion taken, using valuable time and reducing the potential throughout of the A/D converter. The DMA option allows the interface itself to completely handle the transfer of such a block of A/D data directly into memory with no processor action. The interface is free to run at its own highest throughput under DMA making possible a number of applications that were prohibitively expensive in the past.

Normal triggering under DMA causes a wrap-around effect. Every time the A/D DONE bit becomes set, the interface makes a DMA request to the processor. After the interface becomes bus master, the A/D data is sent to memory. The Word Count Register and the Current Address Register are incremented and the A/D DONE bit is cleared. The clearing of the A/D DONE bit automatically triggers the next conversion and the cycle repeats. Thus the data transfer rate is dependent upon the speed of the Data Acquisition Module and the time required to perform the transfer.

An external triggering source (RTC or EXT TRIG) can be used under DMA mode. When an external trigger source is enabled, the wrap-around triggering that is normal for DMA mode is disabled and the interface will wait for explicit triggers. This is the only difference from normal triggering - the rest of the transfer cycle is identical.

Sequential mode may be used under DMA, but the desired start address must be set up before DMA operation is enabled. Under sequential mode, each trigger will increment the multiplexer channel address and perform a conversion on the next channel. If sequential mode is not selected, all conversions will be taken on one channel.

The operation of the interface under DMA mode is as follows:

1. Load the DMA Word Count Register (DMWCR) with the complement of the number of words (A/D conversions)
2. Load the DMA Current Address Register (DMCAR with the starting memory address of the block.
3. If sequential mode is desired, prepare the interface for it now. (see section 5-2B)
4. Load the low byte of DCSR with the DMA ENB bit set and the appropriate triggering mode bits set (if neither RTC or EXT TRIG modes are desired, set the A/D START bit)
5. When the transfer is complete, and End of Range (EOR) interrupt will be generated. The vector for this interrupt is the selected vector for the interface plus 4.

```
Example:
WC:             .WORD          2Ø47.
ADD:            .WORD          8192.
START:          MOV            WC,@#DMWCR       ;load word count into register
                COM            @#DMWCR          ;complement word count
                MOV            ADD,@#DMCAR      ; load address register
                MOVB           #42,@#ADCSR      ; start transfer under RTC
                =user code=
EOR:            =user end of range interrupt service routine=
                RTI
```

5.4 Using the D/A Converter

The DT1761 and DT1765 offer the user a sophisticated dual channel D/A converter. The converter and its associated interface circuitry simplify a number of control tasks. For a description of the interface register, see Chapter 4. Chapter 7 will inform the user where the output connections are to be found.

All data transfers to the DACs occur under programmed I/Ø - to update the DACs, the program merely writes a new word into the DAC Data Buffer Register (DACDBR).

If the two D/A converters are being used for controlling equipment other than display oscilloscopes, the digital mode outputs and the Ƶ axis outputs are available to the user. The Ƶ axis intensify output, for example, is useful if there is a need for a program-generated output pulse. The user may program the pulse through the installation of capacitors for the set-up delay and the pulse width. The digital mode outputs are useful as generalized TTL program outputs.

For driving a CRT point plotting display, the hardware provided on the interface greatly simplifies the refresh operation. To display a point, write both X and Y co-ordinates to the appropriate DACs. If a storage oscilloscope is being used, the digital mode output may be used to control the mode of the CRT (erase, write-through, etc.). When writing the second of the two coordinates, set the Ƶ axis intensify bit. The hardware will automatically wait a preset time for the CRT deflection amplifiers to settle before issuing a Z intensify pulse of selectable duration. When this operation has completed, a point has been displayed on the CRT screen. If a scope with electrostatic deflection amplifiers is being used (7Ø uS amplifier settling time), it is important for the refresh software to be able to detect when the display hardware is ready to display another point. To meet this need, ADCSR bit 13 is internally connected to the Z intensify output. During the interval that a Z intensify pulse is being produced, this bit will read as a logic 1. The falling edge of this bit (the change from a logic 1 to a logic Ø indicates that another dot may be displayed.

Chapter 6

CONFIGURATION

6.1 Introduction

This chapter describes in detail the customer-applied configuration jumpers in addition to explaining the configuration options. The location of all switches, jumpers, and adjustment points are given in Appendix B. If any questions arise as to the location of any jumper or switch mentioned in this chapter, consult this appendix. (Space limitations precluded printing diagrams at every point in the text where a jumper or adjustment point is mentioned)

6.2 Interface Address

The DT176Ø series interfaces allow the user to configure the interface so that the interface will respond to any desired peripheral address in the standard LS1-11 peripheral address space. This "base" or "device" address is set with miniature switches, allowing the user to change the address without any tools.

The interfaces will respond at 8 consecutive locations (with each location corresponding to part of an interface register.) The user selects the base address and the inteface will automatically respond to the desired address and the 7 following locations. The base address may be any value between 16ØØØØ$_8$ and 17777Ø$_8$

To select the base address, DIP switches B8 and B11 are set. Before setting the switches, the user must determine the binary address on a bit by bit basis.

Record the address here:

|←user-selected bits→|

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Ø
1	1	1											x	x	x

(Note: bits 15-13 are fixed as ones (selecting the LS1-11 peripheral address area) and bits 2-Ø are decoded internally to select the 4 double-byte registers).

After choosing the states of bits 12-3, appropriate switches must be set on the interface to make the board respond to the desired addresses. The following table maps each of the 1Ø bits that the user may select onto 1Ø switches. When setting the DIP switch, and "on" or "closed" position represents a zero in the respective address bit.

	Address bit									
Switch	12	11	1Ø	9	8	7	6	5	4	3
B8	5	7	4	2	1	3	6	8	▨	▨
B11	▨	▨	▨	▨	▨	▨	▨	▨	3	2

Numbers given inside table represent sections or individual switches of each DIP switch. For example Address bit 12 maps onto DIP switch B8, switch #5.

Once the base address has been set, bits 2-Ø decode internally to address the following registers:

Address bits 2	1	Ø	Register
Ø	Ø	Ø	ADCSR Lo Byte
Ø	Ø	1	ADCSR Hi Byte
Ø	1	Ø	Data Buffer *
Ø	1	1	Data Buffer *
1	Ø	Ø	DMWCR *
1	Ø	1	DMWCR *
1	1	Ø	DMCAR *
1	1	1	DMCAR *

*These registers are not byte - addressable.

Unless otherwise requested, all DT176Ø series interfaces are factory-set at a base address of 177ØØØ$_8$.

6.3 Interrupt Vectors

In addition to the flexibility of setting the base address, the DT176Ø series interfaces allow the user to select the interrupt vectors used by the interfaces. Like the base address, the base interrupt vector is also set by opening or closing DIP switches. Before setting the switches however the user must select the desired vector. Record the chosen vector here.

|←User - Selected→|

7	6	5	4	3	2	1	Ø
					*	*	*

* Bits 2,1,Ø are generated by the interface as follows:

Bit 2; set if DMA EOR interrupt
cleared if A/D DONE or ERROR interrupt
Bit 1,Ø : always Ø

Once the vector has been chosen, the following table should be used to map the vector address bits onto DIP switch sections.

Vector Address bit →	8	7	6	5	4	3
B 11 switch section →	5	4	1	6	7	8

Unless otherwise requested, all DT176Ø series interfaces are factory-set at a base-vector address of 130$_8$.

6.4 Analog Input Configuration

6.4.1 Selecting SE or DI

Model DT1764 and DT1765 allow the user to configure the analog front end for either single-ended (SE) or differential operation (CI). Thc table below summarizes the connections required:

Input configuration	Jumper connection
SE (Single Ended)	P1-P2, P3-P4
DI (differential)	P5 installed, P2 to P3

Models DT1761 and DT1762 must be factory-configured for either SE or DI input. SE or DI operation is not user selectable. These two models must be ordered according to use. For SE-configured DT1761 and DT1762, P3-P2-P4 are connected at the factory.

6.4.2 Selecting an input range

6.4.2.1 High level (DT1761 and DT1762)

Models DT1761 and DT1762 allow the user to select bipolar or unipolar ranges, the magnitude of the range, and the notation of the resultant A/D data. These are configured through the installation of jumpers on the printed circuit card. The following table summarizes these connections:

Range	Data Notation	Jumper connection
Ø-5	binary	R5-R6 ,R3-R4,S3-S1
± 5	offset binary	R5-R6,R2-R4,S3-S1
± 5	two's complement	R5-R6,R2-R4,S2-S1
Ø-1Ø	binary	R3-R4,S3-S1,R5-R1
± 1Ø	offset binary	R2-R4,S3-S1,R5-R1
± 1Ø	two's complement	R2-R4,S2-S1,R5-R1

6.4.22 Wide Range (DT1764, DT1765)

The wide range systems have the ability for gain and timing selection. The Instrumentation Amplifier may be programmed for any gain from 1 to 1ØØØ. Because the A/D has an input voltage range of either Ø to + 10V or ± 1ØV, the gain of this instrumentation amplifier will determine the input signal voltage range to be digitized. At a gain of 1ØØØ, the input signal for full scale would be 1Ø mV (1Ø mV x 1ØØØ = 1ØV). For a gain of 1, no external resistor is needed, since the instrumentation amplifier has been set to this gain internally. The gain equation for the Instrumentation Amplifier is given below: (see Section 2.4 for additional information)

$$G = \frac{\text{1ØV}}{V_{\text{input range desired}}} = 1 + \frac{20 \times \text{1Ø}^3}{R_g}$$

$$R_g = \frac{2Q \times \text{1Ø}^3}{G\text{-}1}$$ (for a gain of 1, no resistor is required)

6.4.3 Selecting expander channels (DT1762 and DT1764)
Models DT1762 and DT1764 allow configuration of the interface with up to 64SE or 32DI channels. These two models feature an extended addressing mechanism to allow the interface to address more input channels than are present on a single A/D converter module. Model DT1764 utilizes wide range inputs and may be field selectable as either SE or DI. Model DT 1762 utilizes high level inputs and may only be factors set as SE or DI. The following table summarizes the connections required for the extended addressing:

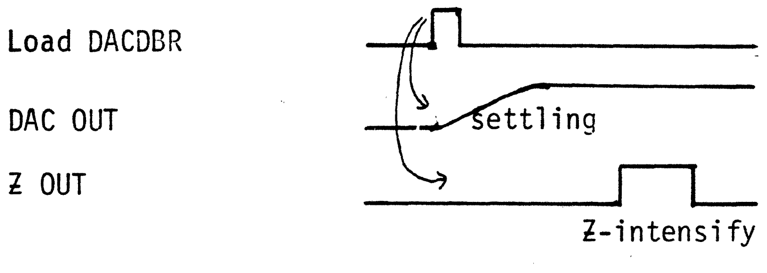
Input mode	#of channels	Jumper connections on expander jumper header
SE	64	jumper across all pins
	32	2-15,3-14,4-13,6-11,7-10,5-8
	16	3-14,4-13,7-10,5-6-8
DI	32	2-16-,3-15,6-12,7-11,8-5
	16	3-15,7-11,5-8
	8	5-6-7-8

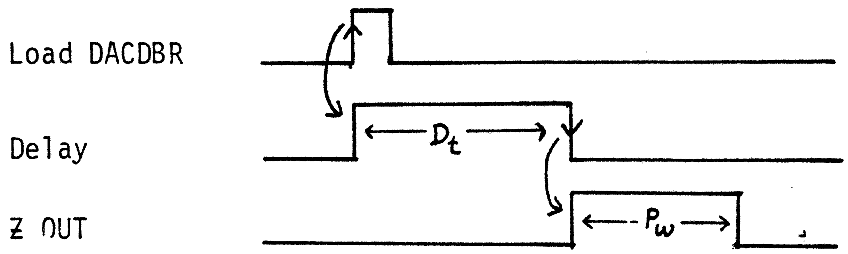
6.5 Analog Output Configuration (DT1761 and DT1765)

6.5.1 Selecting an output range
The output ranges of the two D/A converters provide maximum flexibility as each channel can be configured independently. The following table gives the available configurations:

Range	YDAC jumper	XDAC jumper	Offset binary/ binary	Two's Complement
±5v	Y1-Y2 , Y3-Y4	X1-X2 , X3-X4	M2-M3	M1-M3
Ø-1Øv	Y3-Y4	X3-X4	M2-M3	N/A
±1Øv	Y1- Y2	X1-X2	M2-M3	M1-M3

6.5.2 Configuring Z-intensify
The DT176Ø series interfaces equipped with D/A outputs allow the user the maximum flexibility in mating the D/A channels with any of several standard display oscilloscopes, if that is indeed the use of the outputs. This flexibility is achieved through the on-board timing circuitry associated with the Z intensify output. When the D/A's are used with display oscilloscopes, there is a minimum setup time for the xy values to settle through the horizontal and vertical deflection amplifiers. Once this setup time has been met, a fixed-width output pulse is required to display a dot at the (x,y) location. The timing relationships for the Z intensify output pulse are as follows:





Jumper Connections

Set up delay (D_T)	3 usec D-D1	70 usec D-D2	1 usec-Ø.5 sec. install C_{Delay} between D and D3

Note: D_t= ø.5796 x C_{Delay}
C_{delay} must be greater than 1ØØØ pf

Pulse width (P_w)	Ø.5 usec I-I2	5 usec I-I3	Ø.2 usec - 1ØØ usec install C_{pulse} between I and I1

Note: P_w = Ø.5796 x C_{pulse}

C_{pulse} must be greater than 1ØØØ pf

6.6 Factory-set configuration.
Unless otherwise requested all DT176Ø series interfaces are shipped in the following configurations:

All Models:	Base address 177ØØØ Base Vector 13Ø
Models DT1761, DT1762	A/D range ± 1ØV, 2's complement
Models DT1764, DT1765	A/D range ± 1ØmV, 2's complement
Models DT1761, DT1765	D/A range ± 1ØV, offset binary Ƶ pulse width Ø.5 uS Ƶ delay 3 uS

In addition, interfaces will be shipped configured for either SE or DI operation as requested by the customer, and will be shipped with the requested number of input channels.

Chapter 7

USER CONNECTIONS

7.1 Introduction

User connections of analog signals require a discussion of the various connection schemes available. This includes both single ended and differential inputs with both high and low level signals.

7.2 Single ended analog inputs

Single ended connections are those which have a common side that is referenced back to analog common of the system. The advantage of this scheme is that the user gets twice the number of channels in the same space. The major disadvantage is that the user gives up any common mode rejection he might obtain from a differential system.

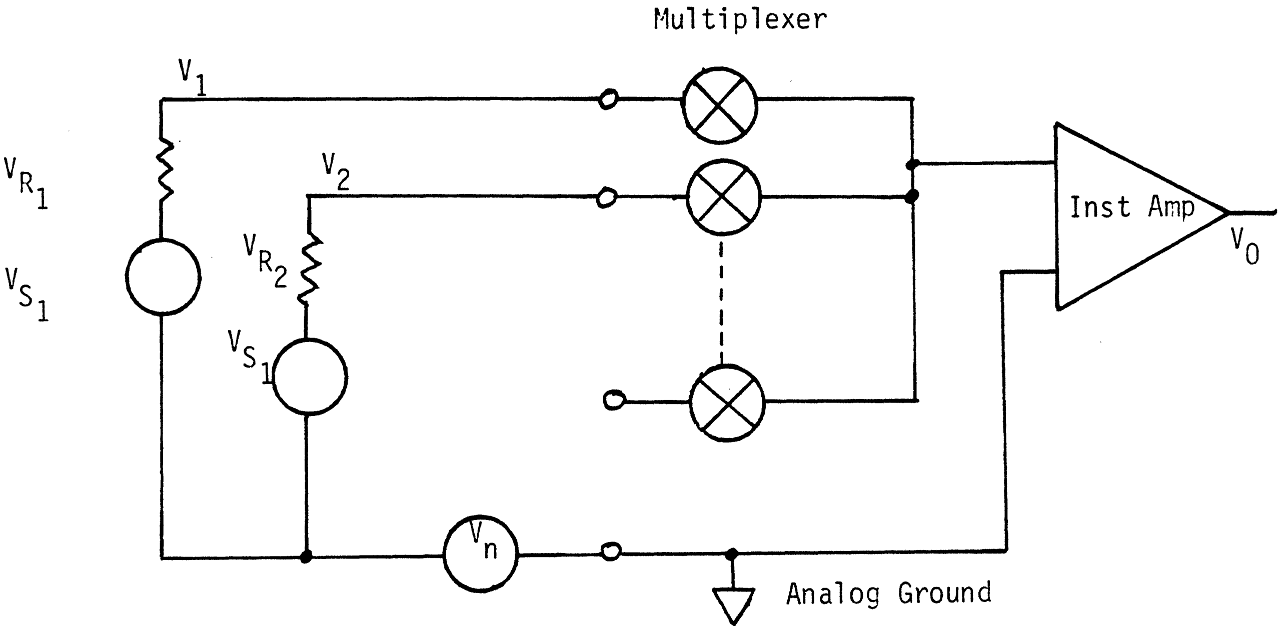
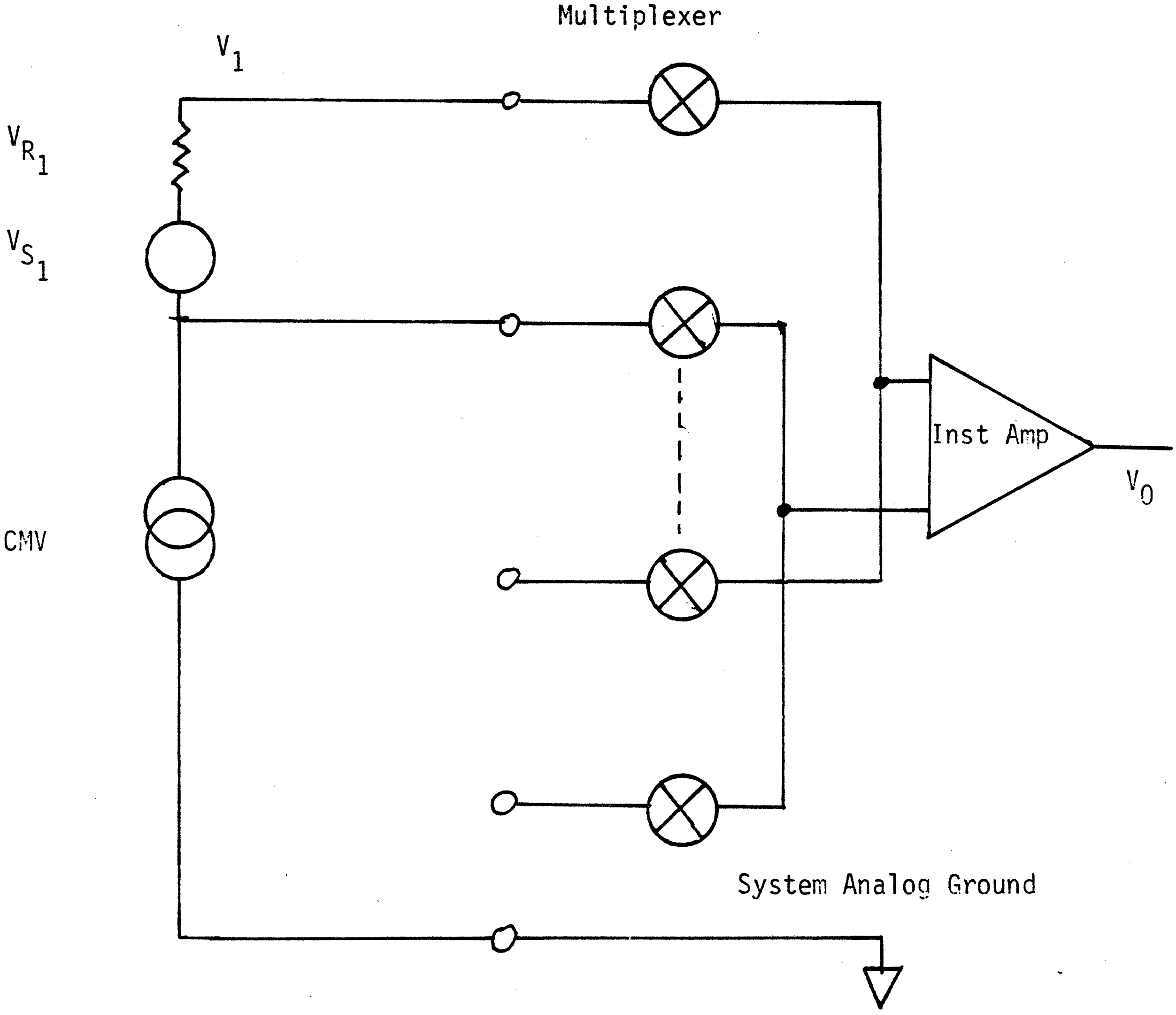


Figure 7.1

Figure 7.1 shows the single ended connection. The output of the instrumentation amplifier, $Vo = Vn + Vs + V_{R1}$, thus error can be introduced due to V_{R1} and Vn, where Vn is the noise voltage and V_{R1} is the voltage drop due to input cabling resistance.

7.3 Differential Inputs

When the differential input scheme is utilized, there are two switches per channel, thus the number of channels are cut in half. The benefits are that common mode voltages, ie. voltages appearing on both sides of the source simultation amplifier. This CMR accounts for a much quieter system. The amount of CMRR depends on how well balanced the instrumentation amplifier is. This can be calculated as follows: Two differential inputs are tied together and connected to a voltage source of plus 10 volts. Than the output of the instrumentaion amplifier is measured as Vo.



$$CMRR = 20 \ Log_{10} \quad \frac{CMV}{VOUT/A}$$

Where CMRR = common mode rejection in db
CMV = common mode voltage
VOUT = The change in the amplifier output voltage due to the CMV.
A = Amplifier gain

For example, the specification for CMRR on Data Translation's wide range interface is 100db at 60HZ at a gain of 1000. Thus with a CMV of 10 volts, the V out of the amplifier at a gain of 1000 should be:

$$100 = 20 \text{ Log }_{10} \frac{\text{CMV}}{\Delta \text{VOUT/A}}$$

$$\text{Antilog } \frac{100}{20} = \frac{10^4}{\text{VOUT/1000}}$$

$$10^5 = \frac{10^4}{\text{VOUT}}$$

$$\text{VOUT} = 10^{-1}\text{V.}$$

Thus with a CMV of 10 volts, the output of the instrumentation amplifier is 100 milli volts. The differential inputs on these systems are not isolated and require a ground return from the system analog ground to the local ground reference of the source.

7.4 Avoiding spurious signals

In order to obtain the best performance from a system, certain guidelines in connecting analog signals to the system should be utilized. These guidelines and precautions will minimize the pickup of electrical noise by the measuring circuits.

7.4.1 Twisted pair input lines

The effects of magneti coupling on the input signals may be reduced for differential input configuration by twisting the signal and return lines. This is effective since the inductive pickup voltages on the two lines tends to match, thus not having an effect on the measurement. This, however, is not the case for a ground referenced single ended system.

7.4.2 Shielded input lines

The effects of electrostatic coupling may be reduced by shielding the input lines. This becomes important if the source has a high impedance. The shield should only be tied to a ground at the instrument end. This will prevent ground loop currents.

7.4.3 Input settling with high source impedance.

Solid state multiplexers inject a small amount of charge into the input lines when channels are switched. This can cause a transient error due to the input source impedance time constants. All Data Translation systems allow for input settling upon new channel selection. The settling time varies for the different input systems available.

Data Acquisition Module	Input Settling Time	Conversion Time
DT57Ø1 (high level)	1Ø uSec.	18.8 uSec.
DT571Ø (1ØØ KHz)	3 uSec	7.Ø uSec.
DT571ØA(125 KHz)	3 uSec	5.Ø uSec.
DT57Ø2 (wide range)	Set by user	2Ø.Ø uSec.

Settling time for input multiplexer

Normally, the control logic allows sufficient time for this charge to settle to less than ½ LSB of error (nine time constants to .Ø12 percent). However, more time may be needed when the multiplexer is switching an input channel with high source impedance, particularly when large amounts of shunt capacitance exists in the interconnecting cables. Source impedance / cable shunt capacitance products greater than Ø.3 uSec. (1K-3ØØPF) on 1ØØKHZ units and 1 uSec. (1K-1ØØØPF) on 3ØKHZ units should be avoided if less than ½ LSB error is desired. Assuming a twisted pair cable capacitance of 5Ø PF / foot and 1K source impedance this translates into a maximum run of six feet on 1ØØKHZ models and twenty feet on 30KHZ models. Note also that settling errors can be minimized by increasing the internal time out with an external capacitor C_t (60pF per uSec.). This time out can only be adjusted on the wide range units which utilize the DT57Ø2 module.

7.5 Connector pin assignments

The remaining pages of this chapter contain tables detailing the pin assignments on the user interface connectors.

7.5.1 Models DT1761-SE, DT1765-SE

CONNECTOR J1			
PIN	SIGNAL	PIN	SIGNAL
1	CHØ	26	A. GND
2	A. GND	27	CH14
3	CH8	28	A. GND
4	A. GND	29	CH7
5	CH1	30	A. GND
6	A. GND	31	CH15
7	CH9	32	A. GND
8	A. GND	33	YDAC OUT
9	CH2	34	A. GND
10	A. GND	35	XDAC OUT
11	CH1Ø	36	A. GND
12	A. GND	37	RTC IN
13	CH3	38	D. GND
14	A. GND	39	EXT TRIG IN
15	CH11	40	D. GND
16	A. GND	41	Ƶ OUT
17	CH4	42	D. GND
18	A. GND	43	DAC MDØ
19	CH12	44	D. GND
20	A. GND	45	DAC MD1
21	CH5	46	D. GND
22	A. GND	47	DAC MD2
23	CH13	48	D. GND
24	A. GND	49	DAC MD3
25	CH6	50	D. GND

7.5.2 Models DT1761-DI, DT1765-DI

CONNECTOR J1			
PIN	SIGNAL	PIN	SIGNAL
1	CHØ	26	A. GND
2	A. GND	27	RET6
3	RETØ	28	A. GND
4	A. GND	29	CH7
5	CH1	30	A. GND
6	A. GND	31	RET7
7	RET1	32	A. GND
8	A. GND	33	YDAC OUT
9	CH2	34	A. GND
10	A. GND	35	XDAC OUT
11	RET2	36	A. GND
12	A. GND	37	RTC IN
13	CH3	38	D. GND
14	A. GND	39	EXT TRIG IN
15	RET3	40	D. GND
16	A. GND	41	Ƶ OUT
17	CH4	42	D. GND
18	A. GND	43	D. GND
19	RET4	44	D. GND
20	A. GND	45	DAC MD1
21	CH5	46	D. GND
22	A. GND	47	DAC MD2
23	RET5	48	D. GND
24	A. GND	49	DAC MD3
25	CH6	50	D. GND

7.5.3 Models DT1762-SE, DT1764-SE

CONNECTOR J1			
PIN	SIGNAL	PIN	SIGNAL
1	CHØ	26	A. GND
2	A. GND	27	CH14
3	CH8	28	A. GND
4	A. GND	29	CH7
5	CH1	30	A. GND
6	A. GND	31	CH15
7	CH9	32	A. GND
8	A. GND	33	
9	CH2	34	
10	A. GND	35	
11	CH1Ø	36	
12	A. GND	37	RTC IN
13	CH3	38	D. GND
14	A. GND	39	EXT TRIG IN
15	CH11	40	D. GND
16	A. GND	41	
17	CH4	42	
18	A. GND	43	
19	CH12	44	
20	A. GND	45	
21	CH5	46	
22	A. GND	47	
23	CH13	48	
24	A. GND	49	
25	CH6	50	

CONNECTOR J2			
PIN	SIGNAL	PIN	SIGNAL
1	CH24	26	CH36
2	CH16	27	CH45
3	CH25	28	CH37
4	CH17	29	CH46
5	CH26	30	CH38
6	CH18	31	CH47
7	CH27	32	CH39
8	CH19	33	CH56
9	CH28	34	CH48
10	CH20	35	CH57
11	CH29	36	CH49
12	CH21	37	CH58
13	CH30	38	CH50
14	CH22	39	CH59
15	CH31	40	CH51
16	CH23	41	CH60
17	CH40	42	CH52
18	CH32	43	CH61
19	CH41	44	CH53
20	CH33	45	CH62
21	CH42	46	CH54
22	CH34	47	CH63
23	CH43	48	CH55
24	CH35	49	A. GND
25	CH44	50	A. GND

7.5.4 Models DT1762-DI, DT1764-DI

CONNECTOR J1			
PIN	SIGNAL	PIN	SIGNAL
1	CHØ	26	A. GND
2	A. GND	27	RET6
3	RETØ	28	A. GND
4	A. GND	29	CH7
5	CH1	30	A. GND
6	A. GND	31	RET7
7	RET1	32	A. GND
8	A. GND	33	
9	CH2	34	
10	A. GND	35	
11	RET2	36	
12	A. GND	37	RTC IN
13	CH3	38	D. GND
14	A. GND	39	EXT TRIG IN
15	RET3	40	D. GND
16	A. GND	41	
17	CH4	42	
18	A. GND	43	
19	RET4	44	
20	A. GND	45	
21	CH5	46	
22	A. GND	47	
23	RET5	48	
24	A. GND	49	
25	CH6	50	

CONNECTOR J2			
PIN	SIGNAL	PIN	SIGNAL
1	RET8	26	CH20
2	CH8	27	RET21
3	RET9	28	CH21
4	CH9	29	RET22
5	RET10	30	CH22
6	CH10	31	RET23
7	RET11	32	CH23
8	CH11	33	RET24
9	RET12	34	CH24
10	CH12	35	RET25
11	RET13	36	CH25
12	CH13	37	RET26
13	RET14	38	CH26
14	CH14	39	RET27
15	RET15	40	CH27
16	CH15	41	RET28
17	RET16	42	CH28
18	CH16	43	RET29
19	RET17	44	CH29
20	CH17	45	RET30
21	RET18	46	CH30
22	CH18	47	RET31
23	RET19	48	CH31
24	CH19	49	A. GND
25	RET20	50	A. GND

Chapter 8

CALIBRATION AND TESTING

8.1 Equipment and System Requirements

In order to assist the user in calibrating and testing the operation of DT1760 series data acquisition systems, Data Translation has developed a comprehensive software calibration aid designated SP0024. This software is provided in either of the two media: Paper tape for minimum, paper-tape based LSI-11 systems and floppy disk for sophisticated RT-11 systems. The system and test equipment requirements for this software are given in the following tables:

SP-0024 System requirements

Paper Tape

KD11-F (LSI-11) processor, ECO #1Ø or greater
or
KD11-HA (LSI-11/2) processor
Minimum of 4k words RAM (on processor card or elsewhere)
Serial Interface at standard DEC console address
Low Speed paper tape reader (TTY) or high speed paper tape reader
Data Translation DT1760 series data acquisition system(s).

Floppy Disk

KD11-F (LSI-11) processor, ECO #1Ø or greater
or
KD11-HA (LSI-11/2) processor
Minimum of 8k words RAM
System console terminal at standard DEC console address
DEC Compatable dual floppy disc drive system
RT-11 operating system (version 2 or version 3)
Data Translation DT1760 series data acquisition system(s).

Test Equipment Requirements

For A/D calibration (all models):
Laboratory quality voltage standard
EDC Model E1ØØC or equivalent

For D/A calibration (models DT1761 and DT1765 only):
Laboratory quality 4½ digit or greater Digital Volt Meter (DVM)
Data Technology Model 4Ø or equivalent
1Ø MHz or greater band width oscilloscope
Tektronix Model T922 or equivalent

8.2.1 Loading SP-ØØ24 From Paper Tape

SP-ØØ24 is supplied in PDP-11 absolute loader format. To load this release into memory the following steps must be taken:

1. Load the LSI-11 absolute loader (see DEC documentation for information on this).

2. Place the paper tape in the paper type reader.

3. Start the absolute loader at location XXX5ØØ when XXX is determined by the following table:

SYSTEM MEMORY SIZE	XXX
4K	017
8K	037
12K	057
16K	077
20K	117
24K	137
28K or greater	157

Following this procedure will cause SP-ØØ24 to be loaded into memory and executed.

8.2.2 Loading Floppy Disk

SP-ØØ24 contains an RT-11 memory image file namded SPØØ24 SAV that is prestructured to run under the DEC RT-11 operation system. This file is linked to load into memory and start execuition.

To load SP-ØØ24, boot RT-11 up on floppy dirve zero and leave the system floppy disk in the drive. Insert SP-ØØ24 in drive one. Type "RUN DX1:SPØØ24 carriage return " in response to the RT-11 monitor's "." prompt character. When SPØØ24 has loaded, the initialization routine will automatically execute.

8.2.3 Using SP-ØØ24

SP-ØØ24 runs under a test executive which allows the user to control the operation of the program and to configure the software for the proper option configuration of his particular interface. Under this test executive the user may ask for a directory of products which the program supports, he may specify what ests to run and he may loop on a test if an error occurs, or halt an error. The commands for the SP-ØØ24 test executive are listed.

COMMAND	FUNCTION
ALL	Runs all logic tests that are present for the current device. Generates an error if there is no current device.
BOOT	Jumps to the standard hardware bootstrap (173ØØØ). Generates an error if there is no bootstrap present.
DIRECTORY	Displays the contents of the current directory. Generates an error if there is no current directory.
EXIT	Halts the processor
MODEL (space)	Displays the parameters associated with the current device. Generates an error if there is no current device.
MODEL (space) DTXXXX	Searches the current directory for the given model number. If found, makes that model the current device. Generates an error of there is no current directory or if the model number can not be found.
TEST (space)	Runs the last test executed
TEST (space) ❬number❭	Runs the indicated test

Test Command prefixes - The following command prefixes are to be used with the TEST command to control the execution of the various tests.

COMMAND PREFIXES TO TEST EXECUTION COMMANDS

R	(TEST command only)	repeat this test continuously
L		loop on this test if an error is detected
H		halt test stream if an error is detected
I		inhibit error printout

A control C will terminate any test.

Example: User wants to run a scope loop on Test 2 because an error is encountered. Type:

ILT (space) 2

In this case the program will loop on test 2 and inhibit error printouts.

SP0024 Error Codes - SP0024 will print an error code when a specific error is encountered. All codes are in the program listing and show the error and what test the program was in when a failure occurred.

8.2.4 Initialization Routine

When the program begins execution a message will be printed on the system console.

```
DATA TRANSLATION   TST-11  MONITOR V02-04.
DIRECTORY OF SP-0024 V01-03.

THIS DIAGNOSTIC CONTAINS ROUTINES FOR

MODEL  DT1761
MODEL  DT1762
MODEL  DT1764
MODEL  DT1765
```

The program will be waiting for the next command with a prompt character. In this case a ">".

The user then enters his model number by utilizing the "M" command. EX. > M < space > DT1761 CR.

The program will then ask for the particular configuration of the board. Once all the configurations are entered the program will give the Base address and Interrupt vector address and print a prompt character.

8.2.5 SP-0024 Program Description

SP-0024 consists of two main sections. The first section which contains 16 tests will test all the logic of the interface board. These tests contain scope loops for debug purposes and will provide an error message if a problem exists. The next section of tests actually produces printouts of A/D data and provides D/A outputs for calibration and verification of operation. The tests contained in SP-0024 are listed.

LOGIC TESTS

TEST 1:	BRPLY FROM ALL REGISTERS
TEST 2:	CHECK A/D START BIT
TEST 3:	CHECK ADCSR R/W BITS
TEST 4:	BINITL ACTION
TEST 5:	BYTE OPERATION OF ADCSR
TEST 6:	CHECK DMWCR BITS
TEST 7:	CHECK DMCAR BITS
TEST 10:	HIGH BYTE TRIGGERING
TEST 11:	A/D START BIT TRIGGERING
TEST 12:	INCREMENT MODE OPERATION
TEST 13:	ERROR BIT OPERATION

TEST 14:	A/D DONE INTERRUPT
TEST 15:	ERROR INTERRUPT
TEST 16:	DMA LOGIC
TEST 17:	DMA DATA TRANSFER
TEST 20:	END OF LOGIC TESTS
TEST 21:	D/A OUT TO A/D IN
TEST 22:	A/D IN TO D/A OUT
TEST 23:	A/D CALIBRATION
TEST 24:	A/D DATA DISPLAY UNDER DMA
TEST 25:	A/D INPUT CHANNEL SCAN
TEST 26:	A/D INPUT GAIN/CHANNEL SCAN
TEST 27:	D/A CALIBRATION
TEST 30:	D/A SQUARE WAVES
TEST 31:	D/A RAMPS
TEST 32:	SCOPE CONTROL OUTPUTS

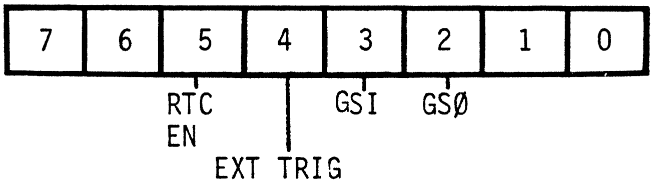
8.2.6 <u>Test Descriptions</u>

All description of tests are located in the program listing at the beginning of each test.

8.2.7 <u>Modes of Operation</u>

Test 23 and test 24 will ask for a MODES input when they are called. This MODES input is the lower byte of the CSR.

When a user wants to run an interface under RTC control or EXT TRIG he can set these bits when the program asks for MODES. DMA mode can only be run with Test 24. Ex. If a user wants to run DMA under EXT TRIG he will set the MODES as 20_8.



MODES BITS FOR USE IN TEST 23 AND TEST 24

8.2.8 <u>Calibration</u>

Calibration of the system requires a voltage standard for highly accurate analog inputs and a DVM for calibration of analog outputs.

Equipment Required:

Voltage standard - EDC Model MV-100 or equivalent
DVM - Data Technology Model 40 or equivalent

8.2.8.1 <u>Calibration of Analog Inputs</u>

Configuration ±10V FS, 2's compliment notation:

1. Connect a Voltage Standard to CHØ input
2. Set standard to -2.4mV
3. Start Test 23 at CHØ, Mode Ø

4. Adjust A/D offset for the printout between 7777 and ØØØØ
5. Set voltage standard to +9.9927V
6. Adjust A/D range control for printout between 3776 and 3777

NOTE: for ±10mV systems (DT1765,DT1764) divide above values by 1000.

8.2.8.2 Calibration of Analog Inputs with Programmable gain amplifier

1. Adjust the A/D offset and range as in section 8.2.8.1
2. Set voltage standard to -600mV
3. Start Test 23 at CHØ Mode 14_8
4. Adjust the Programmable gain offset pot for a printout between 7777 and ØØØØ

8.2.9.3 Calibration of Analog Output

Configuration - ±10V offset binary notation:

1. Connect DVM to YDAC output pin
2. Run Test 27 - this test will produce plus and minus full scale outputs on each D/A connector each time the space bar is hit on the system console
3. Adjust the offset pot for each D/A for -10,000V
4. Adjust the range pot for each D/A for +9.9915V

APPENDIX A

Power Requirements

Although the precision analog-to-digital and digital-to-analog converters manufactured by Data Translation require a highly regulated ± 15 volt power supply, all DT176Ø series interfaces contain an on-board DC-DC converter to meet this requirement. Thus the DT176Ø series interfaces require only +5V for complete operation. This power is supplied by the computer backplane via pins dedicated by Digital Equipment Corporation for supplying logic power to Q-bus interface boards.

Power specifications for all DT176Ø series interfaces:

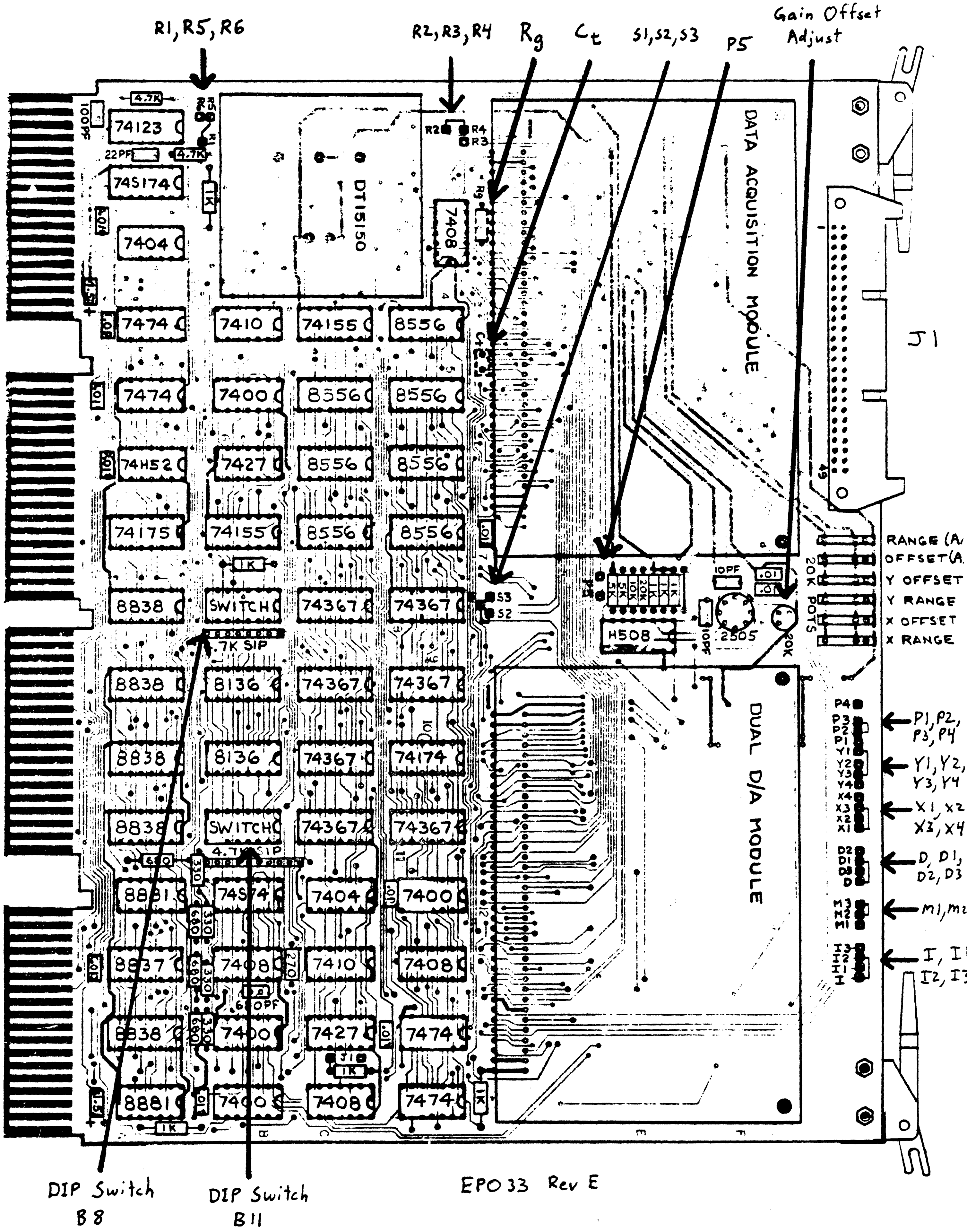
+ 5V @ 2A maximum

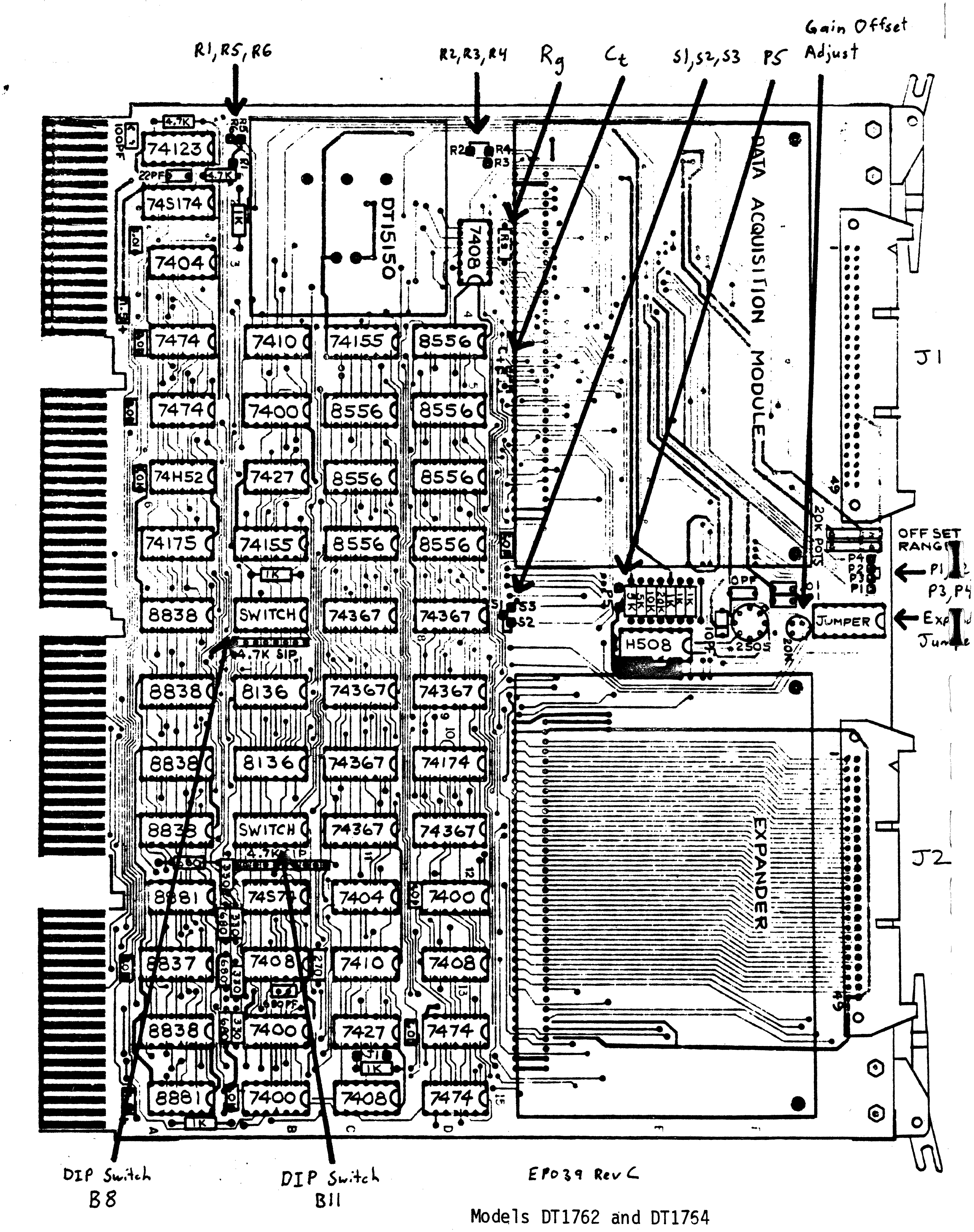
APPENDIX B

Physical Locations of all jumpers, switches and adjustment points.

Models DT1761 and DT1765: see page B - 2

Models DT1762 and DT1764: see page B - 3


R1, R5, R6
R2, R3, R4
Rg
Ct
S1, S2, S3
P5
Gain Offset Adjust
J1
DATA ACQUISITION MODULE
DT15150
DUAL D/A MODULE
74123
74S174
7404
7474
7410
74155
8556
7400
74H52
7427
74175
8838
SWITCH
74367
4.7K SIP
8136
74174
8881
74S74
7408
8837
H508
2505
20K POTS
RANGE (A
OFFSET (A
Y OFFSET
Y RANGE
X OFFSET
X RANGE
P1, P2, P3, P4
Y1, Y2, Y3, Y4
X1, X2, X3, X4
D, D1, D2, D3
M1, M2
I, I1, I2, I3
DIP Switch B8
DIP Switch B11
EPO33 Rev E


R1, R5, R6
R2, R3, R4
Rg
Ct
S1, S2, S3
P5
Gain Offset Adjust
J1
J2
OFF SET RANGE
P1, P2
P3, P4
Exp'd Jumper
DIP Switch B8
DIP Switch B11
EP039 Rev C
DATA ACQUISITION MODULE
EXPANDER
DT15150
74123
74S174
7404
7474
7410
74155
8556
7400
74H52
7427
74175
8838
SWITCH
74367
4.7K SIP
8136
74174
8881
74S74
7408
8837
H508
JUMPER
2505
20K POTS
4.7K
22PF
1K
100PF
R2
R4
R3
S1
S3
S2
49

Data Translation, Inc.
Software Engineering

READER'S COMMENTS

We are interested in hearing of any problems encountered in the use of our software and calibration manuals. Please fill out this form and return it to us if you have any comments or suggestions regarding this release.

Title of manual/release:______________________________

Did you find any errors in this release? If so, please describe.

____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____

Is there sufficient documentation on the software, regarding both system requirements and usage? If not, what material is missing and where should it be placed?

____
____
____
____
____

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Non-programmer
- □ Systems Engineer

Name______________________________Date____________________

Organization____

Street____

City____________________State__________________Zip Code________

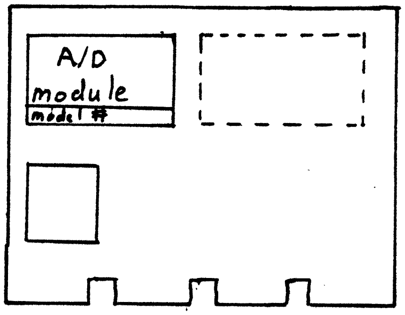
If you require a written reply, please check here. □

APPENDIX C

GENERAL NOTES

C.1 Timing Values for Data Translation A/D modules

(Note: the A/D module number is printed in bold digits on the top of the module. The module can be located on the interface according to this diagram.



Data Translation Module Number	Required Settling Time	A/D Conversion Time	Throughput Time	Maximum Samples/Second
DT5701	10 uSec	18.8 uSec	28.8 uSec	34722
DT5710	3 uSec	7.0 uSec	10.0 uSec	100000
DT5710A	3 uSec	5.0 uSec	8.0	125000
DT5702	see table below	20.0uSec	see table below	

DT5702 Parameters

INPUT RANGE	GAIN	R_g	C_t	SETTLING TIME	SYSTEM ACCURACY	MAXIMUM SAMPLES/SECOND
± 10 mV	1000	20.02	0.015uf	250uS	±0.1%	3703
± 25 mV	400	50.13	6800pf	120uS	±0.08%	7142
± 50 mV	200	100.5	3300pf	70uS	±0.07%	11111
± 100 mV	100	202.0	1500pf	40uS	±0.05%	16667
± 1 volt	10	2222	None	12uS	±0.03%	31250
± 2.5 volts	4	6667	None	12uS	±0.03%	31250
± 5.0 volts	2	20.0K	None	12uS	±0.03%	31250
± 10.0 volts	1	None	None	12uS	±0.03%	31250

C.2 Notes on Full Scale (FS)

Full Scale (FS) is the amount of input voltage required to turn ON all the bits of the A/D converter. For a D/A converter, the inverse is true: Full Scale is the voltage that results when all bits of the converter are turned ON.

In a 12 bit converter there are 4096 possible states (2^{12}). Because one of these states is given to zero, the converter lacks one state at its high or positive FS end. Hence even though the converter is rated at 1Ø volts Full Scale, the positive Full Scale value will actually be 1 state (1 least significant bit value) below 1Ø volts. For example, a Ø-1Ø volt range converter has a least significant bit value of 4.88 mV (1Ø volts / 4096 states). The positive Full Scale will be reached at 1Ø volts -4.88 mV or 9.9951 volts. The negative full scale (in this instance taken to mean the voltage associated with all converter bits OFF) will be Ø volts.

Full Scale Range (FSR)

The Full Scale Range is the difference in voltage between positive Full Scale (all converter bits ON) and negative full scale (all converter bits OFF). Thus a Ø-1Ø volt converter has a Full Scale Range of 1Ø volts while a ±1Ø volt converter has a FSR of 2Ø volts.

C.3 Computing Calibration Values for use with SP-Ø13

(Note: LSB (least significant bit) = FSR ÷ 4096, see C.2 above)

A/D Offset Adjustment

Unipolar Ranges (Ø-n Volts, n ≤ 1Ø)

		Printed Data	
Input Range	Input Voltage	Low Value	High Value
any	+½ LSB	Ø	1 LSB
any, octal output	+½ LSB	ØØØØ	ØØØ1
Ø-1ØV	+1.22Ø7mV	ØØØØ	Ø.ØØ2
Ø.5V	+Ø.61Ø4mV	Ø.ØØØ	Ø.ØØ1
Ø.1ØmV	+1.22Ø7mV	Ø.ØØØ	Ø.ØØ2
4-2ØmA	4.ØØ39mA	4.ØØØmA	4.ØØ4mA

Bipolar Ranges (±n Volts, n ≤ 1Ø)

		Printed	Data
Input Range	Input Voltage	Low Value	High Value
any	Ø volts -½ LSB	Ø volts -1 LSB	Ø volts
any, octal output	Ø volts -½ LSB	7777 3777	ØØØØ (two's complement) 4ØØØ (offset binary)
± 1ØV	-2.4414mV	-Ø.ØØ5	+ Ø.ØØØ
± 5V	-1.22Ø7mV	-Ø.ØØ2	+ Ø.ØØØ
± 1Ø mV	-2.4414uV	-Ø.ØØ5	+ Ø.ØØØ

A/D Range Adjustment

Unipolar Ranges (Ø-n Volts, n ≤ 1Ø)

		Printed Data	
Input Range	Input Voltage	Low Value	High Value
any	+FS-1½ LSB	+FS-2LSB	+FS-1LSB
any, octal output	+FS-1½ LSB	7776	7777
Ø-1ØV	+9.9963V	9.995	9.998
Ø-5V	+4.9982V	4.998	4.999
Ø-1ØmV	+9.9963mV	9.995	9.998
4-2ØmA	+19.9941mA	19.992mA	19.996mA

Bipolar Ranges (±n Volts, n≤1Ø)

		Printed Data	
Input Range	Input Voltage	Low Value	High Value
any	+FS - 1½ LSB	+FS-2 LSB	+FS - 1 LSB
any, octal output	+FS - 1½ LSB	3776 7776	3777 (two's complement) 7777 (offset binary)
±10V	+9.9927	+9.99Ø	+9.995
±5V	+4.9964	+4.995	+4.998
±10mV	+9.9927mV	+9.99Ø	+9.995

A/D Gain Adjustment (LSB now = FSR/(4Ø96*8))

Unipolar Ranges (Ø-n Volts, n ≤ 1Ø)

		Printed Data	
Input Range	Input Voltage	Low Value	High Value
any	+½ LSB	Ø	1 LSB
any, octal output	+½ LSB	ØØØØ	ØØØ1
Ø-1ØV	+ Ø.15259mV	Ø.ØØØ	Ø.ØØØ
Ø-5V	+ Ø.Ø7629mV	Ø.ØØØ	Ø.ØØØ

Bipolar Ranges (±n Volts, n ≤ 1Ø)

		Printed Data	
Input Range	Input Voltage	Low Value	High Value
any	Ø Volts - ½ LSB	Ø Volts - 1 LSB	Ø Volts
any, octal output	Ø Volts - ½ LSB	7777 3777	ØØØØ (two's complement) 4ØØØ (offset binary)
± 1ØV	- Ø.3Ø518mV	-Ø.ØØ1	+Ø.ØØØ
± 5V	- Ø.15259mV	+Ø.ØØØ	+Ø.ØØØ

D/A Offset Adjustment (-FS)

Output Range	-FS
Ø-5V	Ø.ØØØØ
± 5V	-5.ØØØØ
Ø-1ØV	Ø.ØØØØ
±10	-1Ø.ØØØØ

D/A Range Adjustment (+FS)

Output Range	+FS
Ø-5V	+4.9988
±5V	+4.9976
Ø-1ØV	+9.9976
±1ØV	+9.9951

APPENDIX D

SP-0024 PROGRAM LISTINGS

```
RT-11 LINK  V05.04A     LOAD MAP        SAT 31-MAR-79 14:06:07
SP0024.SAV      TITLE:  SP0024  IDENT:  V01.04

SECTION  ADDR   SIZE    GLOBAL  VALUE   GLOBAL  VALUE   GLOBAL  VALUE

. ABS.  000000  001000   (RW,I,GBL,ABS,OVR)
SP0024  001000  000070   (RW,I,LCL,REL,CON)
                        DIRECT  001000
DT1760  001070  006152   (RW,I,LCL,REL,CON)
                        I1761   001244  I1762   001254  I1764   001262
                        I1765   001272  PATCH   007216  T1761   001070
                        T1762   001070  T1764   001070  T1765   001070
TST11   007242  006434   (RW,I,LCL,REL,CON)
                        START   007242

TRANSFER ADDRESS = 007242, HIGH LIMIT = 015676 =  3551. WORDS
```

TABLE OF CONTENTS

```
                        .LIST   TTM
                        .ENABL  LC
                        .TITLE  DT1760  TST-11 MODULE
                        .IDENT  /V01.04/
000000                  .PSECT  DT1760
                .NLIST  BIN
        ;
        ;
                .SBTTL  GENERAL INFORMATION
        ;
        ;
        ; COPYRIGHT (C) 1979, DATA TRANSLATION INCORPORATED. ALL
        ; RIGHTS RESERVED. NO PART OF THIS PROGRAM OR PUBLICATION
        ; MAY BE REPRODUCED WITHOUT THE PRIOR WRITTEN PERMISSION
        ; OF  DATA TRANSLATION INCORPORATED,  4 STRATHMORE ROAD,
        ; NATICK, MASS. 01760.
        ;
        ;
        ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
        ; WITHOUT  NOTICE  AND  SHOULD  NOT  BE  CONSTRUED AS A
        ; COMMITMENT BY DATA TRANSLATION INCORPORATED.
        ;
        ;
        ; DATA TRANSLATION CANNOT ASSUME ANY RESPONSIBILITY FOR
        ; THE USE OF ANY PORTION OF THIS SOFTWARE FOR OTHER THAN
        ; ITS INTENDED  DIAGNOSTIC  PURPOSE  IN CALIBRATING AND
        ; TESTING  DATA  TRANSLATION  MANUFACTURED  ANALOG  AND
        ; DIGITAL INTERFACE BOARDS.
        ;
        ;
        ;
        ; VERSION 01-04
        ;
        ; EDWIN KROEKER 31-MAR-79
        ;
        ;
        ; THIS PROGRAM MODULE CONTAINS ROUTINES TO TEST AND
        ; CALIBRATE DTI DT1760 SERIES ANALOG INTERFACE SYSTEMS
        ; FOR THE LSI-11. A COMPLETE LISTING OF THE MODELS
        ; TESTED BY THIS CODE MODULE WILL BE FOUND ON THE
        ; FOLLOWING PAGES. THIS MODULE IS DESIGNED TO OPERATE
        ; UNDER TST-11 SUPERVISION, AND WILL EXECUTE UNDER
        ; THE LSI-11, LSI-11/2, AND LSI-11/23 CPUS.
        ;
                        .LIST   BIN
                        .SBTTL  TEST PARAMETER BLOCK (TPB)
                ;
                ; TEST-11 DECLARATION
                ;
                        .MCALL  TST11
000000                  TST11
                ;
                ;
                ; ADDITIONAL PARAMETERS USED BY THIS DIAGNOSTIC
                ;
                ;
        000546  DELAY   =546            ; A/D DELAY COUNT STORAGE
                ;
        000522  ERRCNT  =522
                ;
                ;
                ; TEST PARAMETER BLOCK
                ;
                .NLIST  BIN
        ;
000000  TPB:    .WORD   PARAM           ; ADDRESS OF PARAMETER
                                        ; PRINT-OUT ROUTINE
000002          .BYTE   377             ; RESERVED
000003          .BYTE   32              ; # OF TESTS
        ;
        ; TEST ADDRESS TABLE FOR USE BY TST-11
        ;
000004          .WORD   TEST1,PR7
000010          .WORD   TEST2,PR7
000014          .WORD   TEST3,PR7
000020          .WORD   TEST4,PR7
000024          .WORD   TEST5,PR7
000030          .WORD   TEST6,PR7
000034          .WORD   TEST7,PR7
000040          .WORD   TEST10,PR7
000044          .WORD   TEST11,PR7
000050          .WORD   TEST12,PR7
000054          .WORD   TEST13,PR7
000060          .WORD   TEST14,PR7
000064          .WORD   TEST15,PR7
000070          .WORD   TEST16,PR7
000074          .WORD   TEST17,PR7
000100          .WORD   TEST20,PR7
000104          .WORD   TEST21,0
000110          .WORD   TEST22,0
000114          .WORD   TEST23,0
000120          .WORD   TEST24,0
000124          .WORD   TEST25,0
000130          .WORD   TEST26,0
000134          .WORD   TEST27,0
000140          .WORD   TEST30,0
000144          .WORD   TEST31,0
000150          .WORD   TEST32,0
        ;
                        .LIST   BIN
```

```
     1                                  .SBTTL  INITIALIZATION
     2                          ;
     3 000154   012737  INIT61: MOV     #BIT2,@#SWR      ; I/O MODEL
                000004
                000540
     4 000162   000413          BR      INITH
     5                          ;
     6 000164   005037  INIT62: CLR     @#SWR            ; INPUT ONLY MODEL
                000540
     7 000170   000410          BR      INITH
     8                          ;
     9 000172   012737  INIT64: MOV     #BIT4,@#SWR      ; LOW LEVEL INPUT ONLY
                000020
                000540
    10 000200   000430          BR      INITL
    11                          ;
    12 000202   012737  INIT65: MOV     #<BIT4+BIT2>,@#SWR       ; LOW LEVEL I/O
                000024
                000540
    13 000210   000424          BR      INITL
    14                          ;
    15                          ; HIGH LEVEL MODELS INITIALIZATION
    16                          ;
    17 000212           INITH:  RELMOV  #SPEED,R1        ; QUERY USER
    18 000220   004767          CALL    QUERY
                000240
    19 000224   103407          BCS     DT5701           ; TYPED "N"
    20 000226           DT5710: SET     BIT6,SWR         ; DT5710 MODULE
    21 000234   012737          MOV     #14,@#DELAY      ; SET DELAY CONSTANT
                000014
                000546
    22 000242   000412          BR      INIT             ; GO TO COMMON INIT.
    23                          ;
    24 000244           DT5701: SET     BIT5,SWR         ; DT5701 MODULE
    25 000252   012737          MOV     #50,@#DELAY      ; SET DELAY CONSTANT
                000050
                000546
    26 000260   000403          BR      INIT             ; GO TO COMMON INIT.
    27                          ;
    28                          ; LOW LEVEL MODELS INITIALIZATION
    29                          ;
    30 000262   012737  INITL:  MOV     #524,@#DELAY     ; SET LOW LEVEL DELAY
                000524
                000546
     1                         ; INITIALIZATION COMMON TO ALL MODELS
     2                         ;
     3 000270                  INIT:   RELMOV  #DMAOPT,R1        ; QUERY USER
     4 000276  004767                  CALL    QUERY
               000162
     5 000302  103403                  BCS     NODMA             ; TYPED "N"
     6 000304                  DMA:    SET     BIT0,SWR
     7 000312                  NODMA:  PRINT   <# OF A/D INPUT CHANNELS (IN OCTAL): >
     8 000362                          GETOCT                    ; GET OCTAL INPUT
     9 000364  103402                  BCS     1$                ; CR - CONTINUE
    10 000366                          CRLF                      ; ASK AGAIN
    11 000370  000750                  BR      NODMA
    12                         ;
    13 000372  113737          1$:     MOVB    @#ODTACC,@#SWR+1
               000514
               000541
    14 000400  105737                  TSTB    @#ODTACC          ; ZERO VALUE?
               000514
    15 000404  001003                  BNE     2$                ; NO - SKIP
    16 000406                          CLEAR   171,SWR           ; CLEAR SOME BITS
    17 000414                  2$:     TEST    BIT2,SWR          ; I/O MODEL?
    18 000422  001411                  BEQ     3$                ; NO - SKIP
    19 000424                          RELMOV  #DACODE,R1        ; ASK ABOUT CODING
    20 000432  004767                  CALL    QUERY
               000026
    21 000436  103003                  BCC     3$                ; 2'S COMP., SKIP
    22 000440                          SET     BIT1,SWR          ; SET OFF. BIN. FLAG
    23 000446  012737          3$:     MOV     #177000,@#BASE    ; SET DEFAULT ADDRESS
               177000    170400
               000542
    24 000454  012737                  MOV     #130,@#VECTOR     ; SET DEFAULT VECTOR
               000130    400
               000544
    25 000462  000207                  RETURN                    ; ALL DONE
    26                         ;
    27                         ;
    28                         ; SOFTWARE SWITCH REGISTER BIT RESERVATIONS
    29                         ;
    30                         ; BITS 13-8:    # OF A/D CHANNELS
    31                         ; BIT 6:        DT5710 A/D MODULE PRESENT
    32                         ; BIT 5:        DT5701 A/D MODULE PRESENT
    33                         ; BIT 4:        DT5702 A/D MODULE PRESENT
    34                         ; BIT 3:        DT5703 A/D MODULE PRESENT
    35                         ; BIT 2:        DT212 D/A MODULE PRESENT
    36                         ; BIT 1:        DAC OFFSET BINARY CODING
    37                         ; BIT 0:        DMA LOGIC PRESENT
```

```
000464                  QUERY:  PUSH    R1                ; SAVE POINTER
000466                          PRINTS                    ; PRINT PROMPT
000470                          PRINT   < (Y OR N)? >
000506                          TTYIN
000510  122700                  CMPB    #'Y,R0            ; A "Y"?
        000131
000514  001427                  BEQ     2$
000516  122700                  CMPB    #'N,R0            ; A "N"?
        000116
000522  001422                  BEQ     1$                ; YES
000524                          CRLF
000526                          PRINTC  <SEE MANUAL FOR ASSISTANCE.>
000564                          POP     R1                ; RESTORE POINTER
000566  000736                  BR      QUERY
                        ;
000570  000261          1$:     SEC                       ; SET CARRY FLAG
000572  000401                  BR      3$
000574  000241          2$:     CLC                       ; CLEAR CARRY FLAG
000576                  3$:     TTYOUT                    ; ECHO CHARACTER
000600                          CRLF
000602                          POP     R1                ; RESTORE R1
000604  000207                  RETURN
                        ;
                        ;
                        .NLIST  BIN
000606  SPEED:          .ASCIZ  "100 KHZ A/D MODULE"
000631  DMAOPT:         .ASCIZ  "IS THE DMA OPTION PRESENT"
000663  DACODE:         .ASCIZ  "DACS: TWO'S COMPLEMENT CODING"
                        .EVEN
                                .LIST   BIN
                        ;
                        ;
                        ;
                                .SBTTL  DISPLAY PARAMETERS
                        ;
                        ;       THIS ROUTINE DISPLAYS THE CURRENT SETTING
                        ;       OF 'BASE' AND 'VECTOR' ON THE SYSTEM CONSOLE
                        ;       TERMINAL.
                        ;
000722                  PARAM:  PRINT   <  BASE ADDRESS = >
000746  013700                  MOV     @#BASE,R0         ; GET BASE ADDRESS
        000542
000752                          OCT16                     ; DISPLAY
000754                          CRLF
000756                          PRINT   <VECTOR ADDRESS = >
001002  013700                  MOV     @#VECTOR,R0       ; GET VECTOR ADDRESS
        000544
001006                          OCT16                     ; DISPLAY
001010                          CRLF
001012  000207                  RETURN                    ; ALL DONE
```

```
                         .SBTTL  ERROR REPORTERS
                 ;
                 ;
                 ; THIS ROUTINE PROVIDES ERROR REPORTING FOR BUS
                 ; TIME-OUT ERRORS (NO BRPLY FROM INTERFACE).
                 ;
001014           NORPLY: PRINT   <NO BRPLY WHEN ACCESSING LOCATION >
001060  010100           MOV     R1,R0
001062                   OCT16                   ; DISPLAY ADDRESS
001064                   CRLF
001066  000207           RETURN                  ; DONE
                 ;
                 ;
                 ; THIS ROUTINE PROVIDES ERROR REPORTING FOR REGISTER
                 ; BIT ERRORS (ONE OR MORE INCORRECT BITS IN A REGISTER)
                 ;
001070           REG:    PRINTC  <REGISTER ERROR>
001112                   PRINT   <ADDRESS: >
001126                   PUSH    R0              ; SAVE R0
001130  010100           MOV     R1,R0           ; GET ADDRESS
001132                   OCT16
001134                   CRLF
001136                   PRINT   <EXPECTED:>
001152  010200           MOV     R2,R0           ; GET EXPECTED VALUE
001154                   OCT16                   ; DISPLAY
001156                   CRLF
001160                   PRINT   <FOUND:    >
001174  011600           MOV     (SP),R0         ; GET BAD BITS
001176  074200           XOR     R2,R0           ; GENERATE SNAPSHOT
001200                   OCT16                   ; DISPLAY
001202                   CRLF                    ; FORMATTING
001204                   PRINT   <BITS:     >
001220                   POP     R0              ; GET ERROR BITS
001222                   OCT16                   ; DISPLAY
001224                   CRLF
001226  000207           RETURN                  ; DONE
                 ;
                 ;
                 ; THIS ROUTINE PROVIDES ERROR REPORTING FOR THE DMA
                 ; DATA TRANSFER CHECK TEST.  THE LOCATION IN MEMORY
                 ; WHERE BAD DATA WAS FOUND IS REPORTED.
                 ;
                 ;
001230           DMAERR: PRINTC  <DMA DATA TRANSFER ERROR>
001262                   PRINT   <BAD DATA AT LOCATION >
001312  010300           MOV     R3,R0           ; GET ADDRESS
001314  162700           SUB     #2,R0           ; MODIFY
        000002
001320                   OCT16                   ; DISPLAY
001322                   CRLF
001324  000207           RETURN
```

```
                                .SBTTL  MODEL TESTING INFORMATION
                        .NLIST  BIN
        ;
        ; THIS CODE MODULE CONTAINS THE ROUTINES NECESSARY
        ; TO TEST THE FOLLOWING DTI INTERFACE MODELS:
        ;
        ;
        ;
        ;
        ;       DT1761  HIGH LEVEL ANALOG I/O
        ;
        I1761   ==INIT61
        T1761   ==TPB
        ;
        ;
        ;       DT1762  HIGH LEVEL ANALOG INPUT
        ;
        I1762   ==INIT62
        T1762   ==TPB
        ;
        ;
        ;       DT1764  LOW LEVEL ANALOG INPUT
        ;
        I1764   ==INIT64
        T1764   ==TPB
        ;
        ;
        ;       DT1765 LOW LEVEL ANALOG I/O
        ;
        I1765   ==INIT65
        T1765   ==TPB
        ;
                                .LIST   BIN
```

```
                          .SBTTL  LOGIC TESTS
                          .SBTTL  TEST 1:  BRPLY FROM ALL REGISTERS
                  ;
                  ; THIS TEST VERIFIES THAT THE INTERFACE SYSTEM RESPONDS
                  ; WITH A BUS REPLY SIGNAL DURING A BUS DATIO BUS CYCLE.
                  ; ALL REGISTERS AVAILABLE ON THE BOARD ARE CHECKED.
                  ;
                  ;
001326  010602    TEST1:  MOV     SP,R2           ; SAVE SP
001330                    RELMOV  #3$,R0          ; SET UP TRAP TO 4
001336  010037            MOV     R0,@#4
        000004
001342  013701    2$:     MOV     @#BASE,R1       ; GET ADDRESS
        000542
001346                    SCOPE                   ; DECLARE LOOP POINT
001350  005011            CLR     (R1)            ; DATIO BUS CYCLE
001352  062701            ADD     #2,R1           ; NEXT REGISTER
        000002
001356                    SCOPE                   ; DECLARE LOOP POINT
001360  005011            CLR     (R1)            ; DATIO BUS CYCLE
001362  062701            ADD     #2,R1           ; NEXT REGISTER
        000002
001366                    SCOPE                   ; DECLARE LOOP POINT
001370  005011            CLR     (R1)            ; DATIO BUS CYCLE
001372  062701            ADD     #2,R1           ; NEXT REGISTER
        000002
001376                    SCOPE                   ; DECLARE LOOP POINT
001400  005011            CLR     (R1)            ; DATIO CYCLE
001402                    EXIT
                  ;
                  ; **************************************************
                  ;
                  ;       ERROR CODE 1 - BUS TIMEOUT
                  ;
                  ; **************************************************
                  ;
001404  011603    3$:     MOV     (SP),R3         ; GET OFFENDING PC
001406  010206            MOV     R2,SP           ; RESTORE STACK
001410                    ERROR   1,NORPLY        ; REPORT ERROR
001414  000113            JMP     (R3)            ; CONTINUE TEST
```

```
                        .SBTTL  TEST 2:  CHECK A/D START BIT
                ;
                ; THIS TEST CHECKS THE OPERATION OF THE A/D START BIT.
                ; THE BIT IS CHECKED FOR READBACK OF ZERO AND READ-ONLY
                ; OPERATION.
                ;
001416  013701  TEST2:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001422  005002          CLR     R2              ; INIT. TEST REGISTER
001424                  SCOPE                   ; DECLARE LOOP POINT
001426  005011          CLR     (R1)            ; CLEAR BOARD
001430  011100          MOV     (R1),R0         ; GET BIT
001432  042700          BIC     #177776,R0      ; TEST BIT
        177776
001436  001402          BEQ     2$              ; OK - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 2 - A/D START BIT ERROR
                ;                       READS AS A 1
                ;
                ; **************************************************
                ;
001440                  ERROR   2,REG           ; REPORT ERROR
001444          2$:     EXIT
```

```
                                .SBTTL  TEST 3:  CHECK ADCSR R/W BITS
                        ;
                        ; THIS TEST CHECKS ALL OF THE READ/WRITE BITS IN ADCSR.
                        ; BITS ARE CHECKED FOR BOTH SET AND RESET CAPABILITY.
                        ; BITS THAT ARE NOT CHECKED ARE
                        ;
                        ;       A/D START, DMA ENB, A/D DONE, ERROR
                        ;
                        ;
001446  013701  TEST3:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001452  005002          CLR     R2              ; INIT TEST REG.
001454  012703          MOV     #40,R3          ; SET # OF STATES
        000040
001460                  SCOPE                   ; DECLARE LOOP POINT
001462  010211  1$:     MOV     R2,(R1)         ; SET BITS
001464  011100          MOV     (R1),R0         ; GET BITS
001466  042700          BIC     #177603,R0      ; IGNORE SOME BITS
        177603
001472  074200          XOR     R2,R0           ; TEST BITS
001474  001402          BEQ     2$              ; NO ERROR - SKIP
                        ;
                        ; ****************************************************
                        ;
                        ;       ERROR CODE 3 - BIT ERROR, ADCSR
                        ;                       BITS 3-6
                        ;
                        ; ****************************************************
                        ;
001476                  ERROR   3,REG           ; REPORT ERROR
                        ;
001502  062702  2$:     ADD     #4,R2           ; NEXT STATE
        000004
001506  077313          SOB     R3,1$           ; LOOP UNTIL DONE
001510                  TEST    170,SWR         ; A/D PRESENT?
001516  001430          BEQ     7$              ; NO - SKIP
001520  005002          CLR     R2              ; INIT TEST REG.
001522  113703          MOVB    @#SWR+1,R3      ; SET # OF STATES
        000541
001526                  TEST    BIT2,SWR        ; I/O MODEL?
001534  001402          BEQ     3$              ; NO - SKIP
001536  012703          MOV     #20,R3          ; FORCE VALUE
        000020
001542          3$:     SCOPE                   ; DECLARE LOOP POINT
001544  010211  4$:     MOV     R2,(R1)         ; SET BITS
001546  011100          MOV     (R1),R0         ; GET BITS
001550  042700          BIC     #140377,R0      ; IGNORE SOME BITS
        140377
001554  074200          XOR     R2,R0           ; TEST BITS
001556  001402          BEQ     5$              ; NO ERROR - SKIP
```

```
                        ; ************************************************
                        ;
                        ;       ERROR CODE 4 - BIT ERROR, ADCSR
                        ;                        BITS 8-13
                        ;
                        ; ************************************************
                        ;
001560                          ERROR   4,REG           ; REPORT ERROR
                        ;
001564  062702          5$:     ADD     #400,R2         ; NEXT STATE
        000400
001570  013705                  MOV     @#DELAY,R5      ; WAIT
        000546
001574  077501          6$:     SOB     R5,6$           ; DELAY
001576  077316                  SOB     R3,4$           ; LOOP UNTIL DONE
001600  005002          7$:     CLR     R2              ; INIT TEST REG.
001602                          SCOPE                   ; DECLARE LOOP POINT
001604  010211                  MOV     R2,(R1)         ; SET BIT
001606  011100                  MOV     (R1),R0         ; GET BIT
001610  042700                  BIC     #137777,R0      ; TEST BIT
        137777
001614  001402                  BEQ     8$              ; NO ERROR - SKIP
                        ;
                        ; ************************************************
                        ;
                        ;       ERROR CODE 5 - BIT ERROR, ADCSR
                        ;                        INC MODE BIT NOT CLEAR
                        ;
                        ; ************************************************
                        ;
001616                          ERROR   5,REG           ; REPORT ERROR
001622  012702          8$:     MOV     #40000,R2       ; GET BIT
        040000
001626                          SCOPE                   ; DECLARE LOOP POINT
001630  010211                  MOV     R2,(R1)         ; SET BIT
001632  011100                  MOV     (R1),R0         ; GET BIT
001634  042700                  BIC     #137777,R0      ; TEST BIT
        137777
001640  001002                  BNE     9$              ; OK - SKIP ERROR
                        ;
                        ; ************************************************
                        ;
                        ;       ERROR CODE 6 - BIT ERROR, ADCSR
                        ;                        INC MODE BIT NOT SET
                        ;
                        ; ************************************************
                        ;
001642                          ERROR   6,REG           ; REPORT ERROR
001646                  9$:     EXIT                    ; DONE
```

```
                                .SBTTL  TEST 4:  BINITL ACTION
                        ;
                        ; THIS TEST VERIFIES THAT THE BINITL SIGNAL CLEARS
                        ; THE PROPER ADCSR BITS.  THE BITS THAT ARE NOT
                        ; CHECKED ARE
                        ;
                        ;       RTC ENB, EXTTRIG ENB, DMA ENB, MUX ADDRESS BITS
                        ;
                        ;
001650  013701  TEST4:  MOV     @#BASE,R1       ; GET ADDRESS
        000542
001654  012702  1$:     MOV     #40114,R2       ; SET ADCSR BITS
        040114
001660  010211          MOV     R2,(R1)
001662  010211          MOV     R2,(R1)         ; SET ERROR BIT
001664  013700          MOV     @#DELAY,R0      ; WAIT FOR A/D DONE
        000546
001670  077001  2$:     SOB     R0,2$
001672  005002          CLR     R2              ; CLR TEST REG.
001674                  SCOPE                   ; DECLARE LOOP POINT
001676  000005          RESET                   ; ISSUE BINITL
001700  011100          MOV     (R1),R0         ; GET BITS
001702  042700          BIC     #37400,R0       ; IGNORE SOME BITS
        037400
001706  074200          XOR     R2,R0           ; TEST BITS
001710  001402          BEQ     3$              ; OK - SKIP ERROR
                        ;
                        ; ***************************************************
                        ;
                        ;       ERROR CODE 7 - PROPER BIT(S) NOT CLEARED
                        ;                       BY BINITL
                        ;
                        ; ***************************************************
                        ;
001712                  ERROR   7,REG           ; REPORT ERROR
                        ;
001716          3$:     EXIT                    ; ALL DONE
```

```
                              .SBTTL  TEST 5:  BYTE OPERATION OF ADCSR
                      ;
                      ; THIS TEST VERIFIES HIGH AND LOW BYTE OPERATIONS
                      ; INVOLVING THE ADCSR.
                      ;
                      ;
 001720  013701  TEST5:  MOV     @#BASE,R1        ; GET ADDRESS
         000542
 001724  005011          CLR     (R1)             ; CLEAR ADCSR
 001726  013700          MOV     @#DELAY,R0       ; WAIT
         000546
 001732  077001  1$:     SOB     R0,1$
 001734  012702          MOV     #174,R2          ; INIT. TEST REGISTER
         000174
 001740                  SCOPE                    ; DECLARE LOOP POINT
 001742  112711          MOVB    #47574,(R1)      ; SET R/W BITS
         047574
 001746  011100          MOV     (R1),R0          ; GET ADCSR AS WORD
 001750  013703          MOV     @#DELAY,R3       ; LET BOARD FINISH
         000546
 001754  077301  2$:     SOB     R3,2$
 001756  042700          BIC     #100203,R0       ; IGNORE STATUS BITS
         100203
 001762  074200          XOR     R2,R0            ; TEST BITS
 001764  001402          BEQ     3$               ; OK - SKIP ERROR
                 ;
                 ; **************************************************
                 ;
                 ;       ERROR CODE 10 - HIGH BYTE LOADED DURING
                 ;                       A LOW BYTE OPERATION
                 ;
                 ; **************************************************
                 ;
 001766                  ERROR   10,REG           ; REPORT ERROR
```

```
001772  005011  3$:     CLR     (R1)            ; CLEAR ADCSR
001774  013700          MOV     @#DELAY,R0      ; WAIT
        000546
002000  077001  4$:     SOB     R0,4$
002002  113702          MOVB    @#SWR+1,R2      ; INIT TEST REGISTER
        000541
002006  105302          DECB    R2              ; ADJUST
002010  052702          BIS     #76000,R2       ; SET ALL EXTRA BITS
        076000
002014  005201          INC     R1              ; POINT TO HIGH BYTE
002016                  SCOPE                   ; DECLARE LOOP POINT
002020  110211          MOVB    R2,(R1)         ; SET R/W BITS
002022  016100          MOV     -1(R1),R0       ; GET ADCSR AS WORD
        177777
002026  013703          MOV     @#DELAY,R3      ; LET BOARD FINISH
        000546
002032  077301  5$:     SOB     R3,5$
002034  042700          BIC     #100203,R0      ; IGNORE STATUS BITS
        100203
002040  000302          SWAB    R2              ; ADJUST R2
002042  042702          BIC     #100377,R2      ; CLEAR RANDOM BITS
        100377
002046  074200          XOR     R2,R0           ; TEST BITS
002050  001402          BEQ     6$              ; OK - SKIP ERROR
                ;
                ; ************************************************
                ;
                ;       ERROR CODE 11 - LOW BYTE LOADED DURING
                ;                       A HIGH BYTE OPERATION
                ;
                ; ************************************************
                ;
002052                  ERROR   11,REG          ; REPORT ERROR
                ;
002056          6$:     EXIT
```

```
                                .SBTTL  TEST 6:  CHECK DMWCR BITS
                        ;
                        ; THIS TEST CHECKS ALL LEGAL COMBINATIONS OF BITS
                        ; IN THE DMWCR (DMA WORD COUNT REGISTER).
                        ;
                        ;
002060                  TEST6:  TEST    BIT0,SWR        ; DMA PRESENT?
002066  001422                  BEQ     3$              ; NO - SKIP TEST
002070  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
002074  062701                  ADD     #4,R1           ; ADJUST
        000004
002100  005002                  CLR     R2              ; INIT TEST REG.
002102  012703                  MOV     #4096.,R3       ; SET # OF STATES
        010000
002106                          SCOPE                   ; DECLARE LOOP POINT
002110  010211          1$:     MOV     R2,(R1)         ; SET BITS
002112  011100                  MOV     (R1),R0         ; GET BITS
002114  042700                  BIC     #170000,R0      ; IGNORE SOME BITS
        170000
002120  074200                  XOR     R2,R0           ; TEST BITS
002122  001402                  BEQ     2$              ; NO ERROR - SKIP
                        ;
                        ; ************************************************************
                        ;
                        ;       ERROR CODE 12 - BIT ERROR, DMWCR
                        ;
                        ; ************************************************************
                        ;
002124                          ERROR   12,REG          ; REPORT ERROR
                        ;
002130  005202          2$:     INC     R2              ; NEXT STATE
002132  077312                  SOB     R3,1$           ; LOOP UNTIL DONE
002134                  3$:     EXIT
```

```
                                .SBTTL  TEST 7:  CHECK DMCAR BITS
                        ;
                        ; THIS TEST CHECKS ALL LEGAL COMBINATIONS OF BITS
                        ; IN THE DMCAR (DMA CURRENT ADDRESS REGISTER).
                        ;
                        ;
002136                  TEST7:  TEST    BIT0,SWR        ; DMA PRESENT?
002144  001423                  BEQ     3$              ; NO - SKIP TEST
002146  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
002152  062701                  ADD     #6,R1           ; ADJUST
        000006
002156  005002                  CLR     R2              ; INIT TEST REG.
002160  012703                  MOV     #32768.,R3      ; SET # OF STATES
        100000
002164                          SCOPE                   ; DECLARE LOOP POINT
002166  010211          1$:     MOV     R2,(R1)         ; SET BITS
002170  011100                  MOV     (R1),R0         ; GET BITS
002172  042700                  BIC     #1,R0           ; IGNORE BIT 0
        000001
002176  074200                  XOR     R2,R0           ; TEST BITS
002200  001402                  BEQ     2$              ; NO ERROR - SKIP
                        ;
                        ; ***************************************************
                        ;
                        ;       ERROR CODE 13 - BIT ERROR, DMCAR
                        ;
                        ; ***************************************************
                        ;
002202                          ERROR   13,REG          ; REPORT ERROR
                        ;
002206  062702          2$:     ADD     #2,R2           ; NEXT STATE
        000002
002212  077313                  SOB     R3,1$           ; LOOP UNTIL DONE
002214                  3$:     EXIT
```

```
                                .SBTTL  TEST 10: HIGH BYTE TRIGGERING
                        ;
                        ; THIS TEST VERIFIES THAT LOADING THE HIGH BYTE OF
                        ; ADCSR TRIGGERS A CONVERSION AND THAT THE A/D DONE
                        ; BIT IS CLEARED WHEN DATA IS READ.
                        ;
                        ;
 002216                 TEST10: TEST    170,SWR         ; A/D PRESENT?
 002224 001434                  BEQ     5$              ; NO - SKIP TEST
 002226 013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
 002232 005011                  CLR     (R1)            ; CLEAR ADCSR
 002234 013700                  MOV     @#DELAY,R0      ; WAIT
        000546
 002240 077001          1$:     SOB     R0,1$
 002242                         SCOPE                   ; DECLARE LOOP POINT
 002244 016100                  MOV     2(R1),R0        ; READ DATA
        000002
 002250 105711                  TSTB    (R1)            ; DONE BIT CLEAR?
 002252 100001                  BPL     2$              ; YES - SKIP ERROR
                        ;
                        ; ******************************************************
                        ;
                        ;       ERROR CODE 14 - A/D DONE BIT NOT CLEARED
                        ;                       AFTER A/D DATA WAS READ
                        ;
                        ; ******************************************************
                        ;
 002254                         ERROR   14              ; REPORT ERROR
```

TEST 10: HIGH BYTE TRIGGERING

```
  002256              2$:     SCOPE                      ; DECLARE LOOP POINT
  002260  112761              MOVB    #0,1(R1)           ; LOAD HIGH BYTE,ADCSR
          000000
          000001
  002266  013700              MOV     @#DELAY,R0         ; WAIT
          000546
  002272  077001      3$:     SOB     R0,3$
  002274  105711              TSTB    (R1)               ; DONE BIT SET?
  002276  100401              BMI     4$                 ; YES - SKIP ERROR
                      ;
                      ; ************************************************
                      ;
                      ;       ERROR CODE 15 - A/D DONE BIT NOT SET
                      ;
                      ; ************************************************
                      ;
  002300                      ERROR   15                 ; REPORT ERROR
                      ;
  002302              4$:     SCOPE                      ; DECLARE LOOP POINT
  002304  016100              MOV     2(R1),R0           ; READ A/D DATA
          000002
  002310  105711              TSTB    (R1)               ; DONE BIT CLEAR?
  002312  100001              BPL     5$                 ; YES - SKIP ERROR
                      ;
                      ; ************************************************
                      ;
                      ;       ERROR CODE 16 - A/D DONE BIT NOT CLEARED
                      ;                       AFTER A/D DATA WAS READ
                      ;
                      ; ************************************************
                      ;
  002314                      ERROR   16                 ; REPORT ERROR
                      ;
  002316              5$:     EXIT
```

```
                                .SBTTL  TEST 11: A/D START BIT TRIGGERING
                        ;
                        ; THIS TEST VERIFIES THAT THE A/D START BIT OF THE
                        ; ADCSR TRIGGERS A CONVERSION AND THAT THE A/D DONE
                        ; BIT IS CLEARED WHEN DATA IS READ.
                        ;
                        ;
002320                  TEST11: TEST    170,SWR         ; A/D PRESENT?
002326  001432                  BEQ     5$              ; NO - SKIP TEST
002330  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
002334  005011                  CLR     (R1)            ; CLEAR ADCSR
002336  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
002342  077001          1$:     SOB     R0,1$
002344                          SCOPE                   ; DECLARE LOOP POINT
002346  016100                  MOV     2(R1),R0        ; READ DATA
        000002
002352  105711                  TSTB    (R1)            ; DONE BIT CLEAR?
002354  100001                  BPL     2$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 17 - A/D DONE BIT NOT CLEARED
                        ;                       AFTER A/D DATA WAS READ
                        ;
                        ; **************************************************
                        ;
002356                          ERROR   17              ; REPORT ERROR
002360                  2$:     SCOPE                   ; DECLARE LOOP POINT
002362  105211                  INCB    (R1)            ; SET A/D START BIT
002364  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
002370  077001          3$:     SOB     R0,3$
002372  105711                  TSTB    (R1)            ; DONE BIT SET?
002374  100401                  BMI     4$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 20 - A/D DONE BIT NOT SET
                        ;
                        ; **************************************************
                        ;
002376                          ERROR   20              ; REPORT ERROR
```

```
002400          4$:     SCOPE                           ; DECLARE LOOP POINT
002402  016100          MOV     2(R1),R0                ; READ A/D DATA
        000002
002406  105711          TSTB    (R1)                    ; DONE BIT CLEAR?
002410  100001          BPL     5$                      ; YES - SKIP ERROR
                ;
                ; ***********************************************************
                ;
                ;       ERROR CODE 21 - A/D DONE BIT NOT CLEARED
                ;                       AFTER A/D DATA WAS READ
                ;
                ; ***********************************************************
                ;
002412                  ERROR   21                      ; REPORT ERROR
002414          5$:     EXIT
```

```
                                .SBTTL  TEST 12: INCREMENT MODE OPERATION
                        ;
                        ; THIS TEST VERIFIES THAT THE HARDWARE CHANNEL
                        ; SEQUENCING LOGIC FUNCTIONS PROPERLY.
                        ;
                        ;
002416                  TEST12: TEST    170,SWR         ; A/D PRESENT?
002424  001464                  BEQ     10$             ; NO - SKIP TEST
002426  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
002432  005011                  CLR     (R1)            ; CLEAR ADCSR
002434  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
002440  077001          1$:     SOB     R0,1$
002442  113704                  MOVB    @#SWR+1,R4      ; GET # OF CHANNELS
        000541
002446                          TEST    BIT2,SWR        ; I/O MODEL?
002454  001402                  BEQ     2$              ; NO - SKIP
002456  012704                  MOV     #20,R4          ; FORCE VALUE
        000020
002462  110403          2$:     MOVB    R4,R3           ; COPY TO R3
002464  006303                  ASL     R3              ; NOW # OF ITERATIONS
002466  012711                  MOV     #40000,(R1)     ; SET INC. MODE BIT
        040000
002472  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
002476  077001          3$:     SOB     R0,3$
002500  012702                  MOV     #400,R2         ; PRIME TEST REGISTER
        000400
002504                          SCOPE                   ; DECLARE LOOP POINT
002506  011100          4$:     MOV     (R1),R0         ; CHECK ADCSR
002510  042700                  BIC     #140377,R0      ; IGNORE SOME BITS
        140377
002514  074200                  XOR     R2,R0           ; TEST BITS
002516  001402                  BEQ     5$              ; OK- SKIP ERROR
                        ;
                        ; *************************************************
                        ;
                        ;       ERROR CODE 22 - MUX ADDRESSES NOT
                        ;                       INCREMENTING PROPERLY
                        ;
                        ; *************************************************
                        ;
002520                          ERROR   22,REG          ; REPORT ERROR
```

```
002524  062702  5$:     ADD     #400,R2         ; BUMP COUNT
        000400
002530  000302          SWAB    R2              ; SWAP BYTES
002532  120204          CMPB    R2,R4           ; END-OF-CHANNELS?
002534  001001          BNE     6$              ; NO - SKIP
002536  005002          CLR     R2              ; YES - RESET TO ZERO
002540  000302  6$:     SWAB    R2              ; SWAP BYTES BACK
002542  105211          INCB    (R1)            ; TRIGGER CONVERTER
002544  013700          MOV     @#DELAY,R0      ; WAIT
        000546
002550  077001  7$:     SOB     R0,7$
002552  105711          TSTB    (R1)            ; DONE BIT SET?
002554  100401          BMI     8$              ; YES - SKIP
                ;
                ; ***************************************************
                ;
                ;       ERROR CODE 23 - A/D DONE BIT NOT SET
                ;
                ; ***************************************************
                ;
002556                  ERROR   23              ; REPORT ERROR
                ;
002560  077326  8$:     SOB     R3,4$           ; LOOP UNTIL DONE
002562  005011          CLR     (R1)            ; CLEAR BOARD
002564  013700          MOV     @#DELAY,R0      ; WAIT
        000546
002570  077001  9$:     SOB     R0,9$
002572  005761          TST     2(R1)           ; CLEAR A/D DONE
        000002
002576          10$:    EXIT
```

```
                              .SBTTL  TEST 13: ERROR BIT OPERATION
                      ;
                      ; THIS TEST VERIFIES THAT THE ERROR BIT OF THE ADCSR
                      ; CAN BE SET AND CLEARED PROPERLY.
                      ;
                      ;
002600                TEST13: TEST    170,SWR         ; A/D PRESENT?
002606  001440                BEQ     6$              ; NO - SKIP TEST
002610  013701                MOV     @#BASE,R1       ; GET ADDRESS
        000542
002614  005761                TST     2(R1)           ; CLEAR A/D DONE
        000002
002620  005011                CLR     (R1)            ; CLEAR ADCSR
002622  013700                MOV     @#DELAY,R0      ; WAIT
        000546
002626  077001        1$:     SOB     R0,1$
002630  005711                TST     (R1)            ; ERROR BIT CLEAR?
002632  100001                BPL     2$              ; YES - SKIP ERROR
                      ;
                      ; ***************************************************
                      ;
                      ;       ERROR CODE 24 - ERROR BIT NOT CLEAR
                      ;
                      ; ***************************************************
                      ;
002634                        ERROR   24              ; REPORT ERROR
                      ;
002636  012700        2$:     MOV     #1,R0           ; A/D START BIT IN R0
        000001
002642                        SCOPE                   ; DECLARE LOOP POINT
002644  110011                MOVB    R0,(R1)         ; TWO QUICK TRIGGERS
002646  110011                MOVB    R0,(R1)
002650  013702                MOV     @#DELAY,R2      ; WAIT
        000546
002654  077201        3$:     SOB     R2,3$
002656  005711                TST     (R1)            ; ERROR BIT SET?
002660  100401                BMI     4$              ; YES - SKIP ERROR
                      ;
                      ; ***************************************************
                      ;
                      ;       ERROR CODE 25 - ERROR BIT NOT SET
                      ;
                      ; ***************************************************
                      ;
002662                        ERROR   25              ; REPORT ERROR
```

```
002664                  4$:     SCOPE                   ; DECLARE LOOP POINT
002666  005761                  TST     2(R1)           ; CLEAR A/D DONE BIT
        000002
002672  005011                  CLR     (R1)            ; CLEAR ADCSR
002674  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
002700  077001          5$:     SOB     R0,5$
002702  005711                  TST     (R1)            ; ERROR BIT CLEAR?
002704  100001                  BPL     6$              ; YES - SKIP ERROR
                        ;
                        ;  **************************************************
                        ;
                        ;       ERROR CODE 26 - ERROR BIT NOT CLEAR
                        ;
                        ;  **************************************************
                        ;
002706                          ERROR   26              ; REPORT ERROR
                        ;
002710                  6$:     EXIT
```

```
                                .SBTTL  TEST 14: A/D DONE INTERRUPT
                        ;
                        ; THIS TEST VERIFIES THAT THE A/D DONE BIT CAN
                        ; PROPERLY GENERATE AN INTERRUPT.
                        ;
                        ;
002712                  TEST14: TEST    170,SWR         ; A/D PRESENT?
002720  001445                  BEQ     4$              ; NO - SKIP TEST
002722  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
002726  013702                  MOV     @#VECTOR,R2     ; GET VECTOR ADDRESS
        000544
002732  005761                  TST     2(R1)           ; CLEAR A/D DONE BIT
        000002
002736  005011                  CLR     (R1)            ; CLEAR ADCSR
002740  013700                  MOV     @#DELAY,R0      ; WAIT FOR A/D DONE
        000546
002744  077001          1$:     SOB     R0,1$
002746  105711                  TSTB    (R1)            ; IS BIT SET?
002750  100402                  BMI     2$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 27 - A/D DONE BIT NOT SET
                        ;
                        ; **************************************************
                        ;
002752                          ERROR   27              ; REPORT ERROR
002754  000427                  BR      4$              ; CAN'T CONTINUE
002756  005003          2$:     CLR     R3              ; PREPARE STATUS WORDS
002760  012704                  MOV     #PR7,R4
        000340
002764                          RELMOV  #3$,R0          ; GET ISR ADDRESS
002772  010012                  MOV     R0,(R2)         ; STORE
002774  010462                  MOV     R4,2(R2)        ; STORE STATUS TOO
        000002
```

```
003000                            SCOPE                        ; DECLARE LOOP POINT
003002  052711                    BIS      #100,(R1)           ; ENABLE INTERRUPTS
        000100
003006  106403                    MTPS     R3                  ; ENABLE CPU INTERRUPTS
003010  000240                    NOP                          ; STALL TIME
003012  106404                    MTPS     R4                  ; TURN OFF CPU
003014  142711                    BICB     #100,(R1)           ; CLEAR ENABLE BIT
        000100
                        ;
                        ; ************************************************
                        ;
                        ;         ERROR CODE 30 - NO INTERRUPT ON A/D DONE
                        ;
                        ; ************************************************
                        ;
003020                            ERROR    30                  ; REPORT ERROR
003022  000404                    BR       4$                  ; CAN'T CONTINUE
                        ;
003024  062706          3$:       ADD      #4,SP               ; ADJUST STACK
        000004
003030  142711                    BICB     #100,(R1)           ; CLEAR ENABLE BIT
        000100
003034                  4$:       EXIT
```

```
                                .SBTTL  TEST 15: ERROR INTERRUPT
                        ;
                        ; THIS TEST VERIFIES THAT THE ERROR BIT CAN
                        ; PROPERLY GENERATE AN INTERRUPT.
                        ;
                        ;
003036                  TEST15: TEST    170,SWR         ; A/D PRESENT?
003044  001457                  BEQ     5$              ; NO - SKIP TEST
003046  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
003052  013702                  MOV     @#VECTOR,R2     ; GET VECTOR ADDRESS
        000544
003056  005761                  TST     2(R1)           ; CLEAR A/D DONE BIT
        000002
003062  005011                  CLR     (R1)            ; CLEAR ADCSR
003064  012700                  MOV     #1,R0           ; GENERATE ERROR
        000001
003070  010011                  MOV     R0,(R1)         ; TWO QUICK TRIGGERS
003072  010011                  MOV     R0,(R1)
003074  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
003100  077001          1$:     SOB     R0,1$
003102  005711                  TST     (R1)            ; IS BIT SET?
003104  100402                  BMI     2$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 31 - ERROR BIT NOT SET
                        ;
                        ; **************************************************
                        ;
003106                          ERROR   31              ; REPORT ERROR
003110  000435                  BR      5$              ; CAN'T CONTINUE
003112  016100          2$:     MOV     2(R1),R0        ; READ A/D DATA
        000002
003116  105711                  TSTB    (R1)            ; IS DONE BIT CLEAR?
003120  100002                  BPL     3$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 32 - A/D DONE BIT NOT CLEARED
                        ;                       AFTER A/D DATA WAS READ
                        ;
                        ; **************************************************
                        ;
003122                          ERROR   32              ; REPORT ERROR
003124  000427                  BR      5$              ; CAN'T CONTINUE
```

```
     1 003126  005003  3$:     CLR     R3                ; PREPARE STATUS WORDS
     2 003130  012704          MOV     #PR7,R4
               000340
     3 003134                  RELMOV  #4$,R0            ; GET ISR ADDRESS
     4 003142  010012          MOV     R0,(R2)           ; STORE
     5 003144  010462          MOV     R4,2(R2)          ; STORE STATUS TOO
               000002
     6 003150                  SCOPE                     ; DECLARE LOOP POINT
     7 003152  052711          BIS     #100,(R1)         ; ENABLE INTERRUPTS
               000100
     8 003156  106403          MTPS    R3                ; ENABLE CPU INTERRUPTS
     9 003160  000240          NOP                       ; STALL TIME
    10 003162  106404          MTPS    R4                ; TURN OFF CPU
    11 003164  142711          BICB    #100,(R1)         ; CLEAR ENABLE BIT
               000100
    12                 ;
    13                 ;  ***************************************************
    14                 ;
    15                 ;       ERROR CODE 33 - NO INTERRUPT ON ERROR
    16                 ;
    17                 ;  ***************************************************
    18                 ;
    19 003170                  ERROR   33                ; REPORT ERROR
    20 003172  000404          BR      5$                ; EXIT
    21                 ;
    22 003174  062706  4$:     ADD     #4,SP             ; ADJUST STACK
               000004
    23 003200  142711          BICB    #100,(R1)         ; CLEAR ENABLE BIT
               000100
    24 003204          5$:     EXIT
```

```
                                .SBTTL  TEST 16: DMA LOGIC
                        ;
                        ; THIS TEST VERIFIES THAT THE DMA LOGIC IS CYCLING
                        ; PROPERLY.  THIS TEST DOES NOT CHECK THE ACTUAL
                        ; TRANSFERRED DATA.
                        ;
                        ;
003206                  TEST16: TEST    BIT0,SWR        ; DMA PRESENT?
003214  001550                  BEQ     13$             ; NO - SKIP.TEST
003216  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
003222  013702                  MOV     @#VECTOR,R2     ; GET VECTOR
        000544
003226  012703                  MOV     #176777,R3      ; GET WORD COUNT
        176777
003232  012704                  MOV     #16000,R4       ; GET BUFFER ADDRESS
        016000
003236  005011                  CLR     (R1)            ; CLEAR BOARD
003240  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
003244  077001          1$:     SOB     R0,1$
003246  005761                  TST     2(R1)           ; CLEAR DONE BIT
        000002
003252                          SCOPE                   ; DECLARE LOOP POINT
003254  010361                  MOV     R3,4(R1)        ; LOAD DMWCR
        000004
003260  010461                  MOV     R4,6(R1)        ; LOAD DMCAR
        000006
003264  152711                  BISB    #2,(R1)         ; START TRANSFER
        000002
003270  132711                  BITB    #2,(R1)         ; BIT SET?
        000002
003274  001002                  BNE     2$              ; YES - SKIP ERROR
                        ;
                        ; **************************************************
                        ;
                        ;       ERROR CODE 34 - DMA ENABLE BIT NOT SET
                        ;
                        ; **************************************************
                        ;
003276                          ERROR   34              ; REPORT ERROR
003300  000516                  BR      13$             ; CAN'T CONTINUE
```

```
 003302  000005  2$:     RESET                       ; BINITL CLEAR BIT?
 003304  132711          BITB    #2,(R1)             ; TEST
         000002
 003310  001402          BEQ     3$                  ; YES - SKIP ERROR
                 ;
                 ; **************************************************
                 ;
                 ;       ERROR CODE 35 - BINITL FAILED TO CLEAR
                 ;                       DMA ENABLE BIT
                 ;
                 ; **************************************************
                 ;
 003312                  ERROR   35                  ; REPORT ERROR
 003314  000510          BR      13$                 ; CAN'T CONTINUE
                 ;
 003316          3$:     RELMOV  #8$,R0              ; GET ISR ADDRESS
 003324  010062          MOV     R0,4(R2)            ; STORE
         000004
 003330  012762          MOV     #PR7,6(R2)          ; STORE STATUS TOO
         000340
         000006
 003336  005000          CLR     R0                  ; ENABLE CPU INTERRUPTS
 003340  106400          MTPS    R0
 003342                  SCOPE                       ; DECLARE LOOP POINT
 003344  010361          MOV     R3,4(R1)            ; LOAD DMWCR
         000004
 003350  010461          MOV     R4,6(R1)            ; LOAD DMCAR
         000006
 003354  005761          TST     2(R1)               ; CLEAR DONE
         000002
 003360  152711          BISB    #16,(R1)            ; START TRANSFER
         000016
                 ;
         003362' OLD1    =.-2                        ; FOR OLD REV. PATCH
                 ;
 003364  016105  4$:     MOV     4(R1),R5            ; SAMPLE DMWCR
         000004
 003370  042705          BIC     #170000,R5
         170000
 003374  022705          CMP     #7777,R5            ; END OF RANGE?
         007777
 003400  001413          BEQ     6$                  ; YES - WAIT FOR EOR
 003402  013700          MOV     @#DELAY,R0          ; WAIT
         000546
 003406  077001  5$:     SOB     R0,5$
 003410  016100          MOV     4(R1),R0            ; SAMPLE DMWCR
         000004
 003414  042700          BIC     #170000,R0
         170000
 003420  020005          CMP     R0,R5               ; ANY TRANSFERS?
 003422  001360          BNE     4$                  ; YES - CONTINUE
```

```
                ; **************************************************
                ;
                ;       ERROR CODE 36 - DMA FAILING TO CYCLE
                ;
                ; **************************************************
                ;
003424                  ERROR   36              ; REPORT ERROR
003426  000443          BR      13$             ; CAN'T CONTINUE
                ;
003430  013700  6$:     MOV     @#DELAY,R0      ; WAIT ONE MORE CYCLE
        000546
003434  077001  7$:     SOB     R0,7$
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 37 - NO EOR INTERRUPT
                ;
                ; **************************************************
                ;
003436                  ERROR   37              ; REPORT ERROR
003440  000436          BR      13$             ; CAN'T CONTINUE
                ;
003442  062706  8$:     ADD     #4,SP           ; ADJUST STACK
        000004
003446  012705          MOV     #7777,R5        ; DMWCR = 7777?
        007777
003452  046105          BIC     4(R1),R5
        000004
003456  001402          BEQ     9$              ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 40 - EOR INTERRUPT WITHOUT
                ;                       DMWCR = 7777
                ;
                ; **************************************************
                ;
003460                  ERROR   40              ; REPORT ERROR
003462  000425          BR      13$             ; CAN'T CONTINUE
                ;
003464  032711  9$:     BIT     #2,(R1)         ; DMA ENB BIT CLEARED?
        000002
003470  001402          BEQ     10$             ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 41 - DMA ENABLE BIT NOT CLEARED
                ;                       AT END OF TRANSFER
                ;
                ; **************************************************
                ;
003472                  ERROR   41              ; REPORT ERROR
003474  000420          BR      13$             ; CAN'T CONTINUE
```

```
003476  016105  10$:    MOV     6(R1),R5        ; CHECK DMCAR
        000006
003502  042705          BIC     #1,R5           ; IGNORE BIT 0
        000001
003506  022705          CMP     #20000,R5       ; CORRECT?
        020000
003512  001402          BEQ     11$             ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 42 - DMCAR INCORRECT AT END
                ;                       OF DMA TRANSFER
                ;
                ; **************************************************
                ;
003514                  ERROR   42              ; REPORT ERROR
003516  000407          BR      13$             ; CAN'T CONTINUE
                ;
003520  005711  11$:    TST     (R1)            ; ERROR BIT CLEAR?
003522  100001          BPL     12$             ; YES - ALL DONE
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 43 - ERROR BIT SET DURING
                ;                       DMA TRANSFER
                ;
                ; **************************************************
                ;
003524                  ERROR   43              ; REPORT ERROR
                ;
003526  132711  12$:    BITB    #14,(R1)        ; BITS CLEARED?
        000014
003532  001401          BEQ     13$             ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 44 - IMPROPER ADCSR STATE
                ;
                ; **************************************************
                ;
003534                  ERROR   44              ; REPORT ERROR
                ;
003536          13$:    EXIT                    ; ALL DONE
```

```
                                .SBTTL  TEST 17: DMA DATA TRANSFER
                        ;
                        ; THIS TEST VERIFIES THAT THE DMA LOGIC IS ACTUALLY
                        ; TRANSFERRING DATA INTO MEMORY.
                        ;
                        ;
003540                  TEST17: TEST    BIT0,SWR        ; DMA PRESENT?
003546  001526                  BEQ     14$             ; NO - SKIP TEST
003550  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
003554  013702                  MOV     @#VECTOR,R2     ; GET VECTOR
        000544
003560  005011                  CLR     (R1)            ; CLEAR BOARD
003562  013700                  MOV     @#DELAY,R0      ; WAIT
        000546
003566  077001          1$:     SOB     R0,1$
003570  012703                  MOV     #4,R3           ; SET # OF ITERATIONS
        000004
003574  005005                  CLR     R5              ; CLEAR AVERAGE
003576  105211          2$:     INCB    (R1)            ; TRIGGER CONVERSION
003600  013700                  MOV     @#DELAY,R0      ; WAIT FOR A/D DONE
        000546
003604  077001          3$:     SOB     R0,3$
003606  105711                  TSTB    (R1)            ; SET?
003610  100402                  BMI     4$              ; YES - SKIP ERROR
                        ;
                        ; ************************************************
                        ;
                        ;       ERROR CODE 45 - A/D DONE BIT NOT SET
                        ;
                        ; ************************************************
                        ;
003612                          ERROR   45              ; REPORT ERROR
003614  000503                  BR      14$             ; EXIT
                        ;
003616  066105          4$:     ADD     2(R1),R5        ; ADD IN DATA
        000002
003622  077313                  SOB     R3,2$           ; GET MORE
003624  006205                  ASR     R5              ; DIVIDE BY 4
003626  006205                  ASR     R5
003630  005105                  COM     R5              ; COMPLEMENT
003632  042705                  BIC     #170000,R5      ; CLEAR UPPER 4 BITS
        170000
003636  012703                  MOV     #16000,R3       ; INIT. TRANSFER AREA
        016000
003642  012704                  MOV     #1000,R4        ; NUMBER OF WORDS
        001000
003646  010523          5$:     MOV     R5,(R3)+        ; LOAD DATA
003650  077402                  SOB     R4,5$           ; LOOP UNTIL DONE
```

TEST 17: DMA DATA TRANSFER

```
 003652  012761              MOV     #177377,4(R1)    ; LOAD DMWCR
         177377
         000004
 003660  012761              MOV     #16400,6(R1)     ; LOAD DMCAR
         016400
         000006
 003666                      RELMOV  #8$,R0           ; GET ISR ADDRESS
 003674  010062              MOV     R0,4(R2)         ; STORE
         000004
 003700  012762              MOV     #PR7,6(R2)       ; STORE STATUS TOO
         000340
         000006
 003706  005000              CLR     R0               ; ENABLE CPU INTERRUPTS
 003710  106400              MTPS    R0
 003712  152711              BISB    #2,(R1)          ; START TRANSFER
         000002
                     ;
         003714'     OLD2    =.-2                     ; FOR OLD REV. PATCH
                     ;
 003716  012703              MOV     #400,R3          ; WAIT FOR TRANSFER
         000400
 003722  013700      6$:     MOV     @#DELAY,R0
         000546
 003726  077001      7$:     SOB     R0,7$
 003730  077304              SOB     R3,6$
                     ;
                     ; **************************************************
                     ;
                     ;       ERROR CODE 46 - NO EOR INTERRUPT
                     ;
                     ; **************************************************
                     ;
 003732                      ERROR   46               ; REPORT ERROR
 003734  000433              BR      14$              ; CAN'T CONTINUE
                     ;
 003736  062706      8$:     ADD     #4,SP            ; ADJUST STACK
         000004
 003742  012703              MOV     #16000,R3        ; CHECK LOW BUFFER
         016000
 003746  012704              MOV     #300,R4          ; (16000 TO 16600)
         000300
 003752  022305      9$:     CMP     (R3)+,R5         ; A/D DATA?
 003754  001001              BNE     10$              ; YES - EXIT LOOP
 003756  077403              SOB     R4,9$            ; CHECK REST OF BUFFER
```

```
003760  022703  10$:    CMP     #16402,R3       ; CORRECT FIRST WORD?
        016402
003764  001403          BEQ     11$             ; YES - SKIP ERROR
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 47 - ERROR IN DMA TRANSFER,
                ;                       STARTING LOCATION
                ;
                ; **************************************************
                ;
003766                  ERROR   47,DMAERR       ; REPORT ERROR
003772  000414          BR      14$
                ;
003774  012703  11$:    MOV     #16400,R3       ; CHECK REST OF BUFFER
        016400
004000  012704          MOV     #500,R4         ; (16400 TO 17600)
        000500
004004  022305  12$:    CMP     (R3)+,R5        ; A/D DATA?
004006  001401          BEQ     13$             ; NO - EXIT LOOP
004010  077403          SOB     R4,12$          ; CHECK REST OF BUFFER
004012  022703  13$:    CMP     #17402,R3       ; CORRECT LAST WORD?
        017402
004016  001402          BEQ     14$             ; YES - EXIT
                ;
                ; **************************************************
                ;
                ;       ERROR CODE 50 - ERROR IN DMA TRANSFER,
                ;                       ENDING LOCATION
                ;
                ; **************************************************
                ;
004020                  ERROR   50,DMAERR       ; REPORT ERROR
004024          14$:    EXIT
```

```
                                .SBTTL  TEST 20: END OF LOGIC TESTS
                        ;
                        ; THIS TEST IS PRESENT TO INFORM THE TST-11 MONITOR
                        ; THAT THERE ARE NO MORE ADDITIONAL LOGIC TESTS
                        ; TO BE EXECUTED.  WHEN THE "ALL" COMMAND IS USED,
                        ; THIS TEST WILL FORCE A RETURN TO THE COMMAND
                        ; LEVEL OF TST-11 WHEN THE TEST SEQUENCER REACHES
                        ; THIS TEST.
                        ;
                        ;
004026                  TEST20: ESCAPE
                        ;
                        ;
                                .SBTTL  CALIBRATION INITIALIZATION
                        ;
                        ; THIS ROUTINE PERFORMS INITIALIZATION FUNCTIONS
                        ; FOR SOME OF THE CALIBRATION TESTS.
                        ;
                        ;
004030                  GETCH:  PRINT   <CHANNEL? >     ; OUTPUT PROMPT
004044                          GETOCT                  ; GET VALUE
004046  103422                  BCS     1$              ; CR AT END - CONT.
004050                          PRINTC  <ENTER AN OCTAL CHANNEL ADDRESS>
004112  000746                  BR      GETCH           ; TRY AGAIN
                        ;
004114  013702          1$:     MOV     @#ODTACC,R2     ; GET VALUE
        000514
004120  042702                  BIC     #177700,R2      ; CLEAR UPPER BITS
        177700
004124  000302                  SWAB    R2              ; ADJUST
004126                  2$:     PRINT   <MODE BITS? >   ; OUTPUT PROMPT
004144                          GETOCT                  ; GET VALUE
004146  103420                  BCS     3$              ; CR AT END - CONT.
004150                          PRINTC  <ENTER AN OCTAL BYTE VALUE.>
004206  000747                  BR      2$
                        ;
004210  013700          3$:     MOV     @#ODTACC,R0     ; GET VALUE
        000514
004214  042700                  BIC     #177400,R0      ; CLEAR HIGH BYTE,R0
        177400
004220  060002                  ADD     R0,R2           ; ADD IN MODE BYTE
004222  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
004226  005761                  TST     2(R1)           ; CLEAR DONE
        000002
004232  005011                  CLR     (R1)            ; CLEAR BOARD
004234  105711          4$:     TSTB    (R1)            ; WAIT FOR DONE
004236  100376                  BPL     4$
004240  005761                  TST     2(R1)           ; CLEAR DONE
        000002
004244  000207                  RETURN
```

```
                                   .SBTTL  DAC INITIALIZATION
                          ;
                          ; THIS ROUTINE PROVIDES INITIALIZATION FUNCTIONS
                          ; FOR SOME OF THE TESTS THAT EXERCISE THE DACS.
                          ;
                          ;
004246                    DAC:     TEST    BIT2,SWR          ; I/O MODEL?
004254  001001                     BNE     1$
004256                             EXIT                      ; NO - EXIT
                          ;
004260  012702            1$:      MOV     #4000,R2          ; SET FOR TWO'S COMP
        004000
004264  012703                     MOV     #3777,R3
        003777
004270                             TEST    BIT1,SWR          ; OFFSET BINARY?
004276  001404                     BEQ     2$                ; NO - SKIP
004300  042702                     BIC     #4000,R2          ; MODIFY
        004000
004304  052703                     BIS     #4000,R3
        004000
004310  010204            2$:      MOV     R2,R4             ; MAKE COPIES FOR
004312  052704                     BIS     #100000,R4        ; SECOND DAC CHANNEL
        100000
004316  010305                     MOV     R3,R5
004320  052705                     BIS     #100000,R5
        100000
004324  013701                     MOV     @#BASE,R1         ; GET ADDRESS
        000542
004330  000207                     RETURN
```

```
                        .SBTTL  TEST 21: D/A OUT TO A/D IN
                ;
                ; THIS TEST OUTPUTS A RAMP ON THE X DAC WHILE
                ; DISPLAYING A/D INPUT DATA FROM CHANNEL 0.  IF THE
                ; X DAC IS CONNECTED TO THE CHANNEL 0 INPUT, THE
                ; USER CAN VERIFY THAT THE ANALOG SUBSYSTEMS ARE
                ; FUNCTIONING (ALTHOUGH NOT CALIBRATED).
                ;
                ;
004332  004767  TEST21: CALL    DAC
        177710
004336                  PRINTC  <CONNECT THE X DAC TO CHANNEL 0>
004400                  KBEXIT                  ; SET UP KEYBOARD
004402  005761          TST     2(R1)           ; CLEAR DONE
        000002
004406  005011          CLR     (R1)            ; CLEAR CSR
004410  105711  2$:     TSTB    (R1)            ; WAIT FOR DONE
004412  100376          BPL     2$
004414  005761          TST     2(R1)           ; CLEAR DONE
        000002
004420          3$:     CRLF
004422  112705          MOVB    #10,R5          ; LINE COUNTER
        000010
004426  105211  4$:     INCB    (R1)            ; TRIGGER CONVERTER
004430  105711  5$:     TSTB    (R1)            ; WAIT FOR DONE
004432  100376          BPL     5$
004434  010261          MOV     R2,2(R1)        ; SET UP NEXT STATE
        000002
004440  016100          MOV     2(R1),R0        ; GET A/D DATA
        000002
004444                  OCT12                   ; DISPLAY
004446  005202          INC     R2              ; NEXT DAC STATE
004450  042702          BIC     #70000,R2       ; 12 BITS ONLY
        070000
004454  105305          DECB    R5              ; LINE OVER?
004456  001760          BEQ     3$              ; YES - NEXT LINE
004460                  TAB                     ; NEXT DATA
004462  000761          BR      4$
```

```
                                .SBTTL  TEST 22: A/D IN TO D/A OUT
                        ;
                        ; THIS TEST ALLOWS A USER TO CONNECT A SIGNAL SOURCE
                        ; TO INPUT CHANNEL 0 AND SEE WHAT SHOULD BE THE
                        ; SAME SIGNAL APPEARING ON THE X DAC OUTPUT.
                        ;
                        ;
004464  004767  TEST22: CALL    DAC
        177556
004470                  PRINTC  <CONNECT SIGNAL SOURCE TO CHANNEL 0>
004536                  KBEXIT                          ; SET UP KEYBOARD
004540  013701          MOV     @#BASE,R1               ; GET ADDRESS
        000542
004544  005761          TST     2(R1)                   ; CLEAR DONE
        000002
004550  005011          CLR     (R1)                    ; START LOOP
004552  105711  1$:     TSTB    (R1)                    ; WAIT FOR DONE
004554  100376          BPL     1$
004556  016100          MOV     2(R1),R0                ; GET DATA
        000002
004562  105211          INCB    (R1)                    ; START NEXT CONVERSION
004564  042700          BIC     #170000,R0              ; PREPARE Y DAC DATA
        170000
004570  010061          MOV     R0,2(R1)                ; LOAD DAC DATA
        000002
004574  000766          BR      1$
```

```
                        .SBTTL  TEST 23: A/D CALIBRATION
                ;
                ; THIS TEST ACCEPTS A CHANNEL ADDRESS FROM THE USER
                ; AND DISPLAYS THE DATA FROM THAT CHANNEL CONTINUOUSLY
                ;
                ;
004576          TEST23: PRINTC  <A/D CALIBRATION>
004620  004767          CALL    GETCH           ; GET CHANNEL ADDRESS
        177204
004624                  KBEXIT                  ; SET UP KEYBOARD
004626  042702          BIC     #140303,R2      ; CLEAR SOME BITS
        140303
004632  010200          MOV     R2,R0           ; GET CSR IN R0
004634  042700          BIC     #60,R0          ; NO EXTERNAL TRIGGER
        000060
004640  010011          MOV     R0,(R1)         ; SET UP CHANNEL
004642  105711  1$:     TSTB    (R1)            ; WAIT FOR DONE
004644  100376          BPL     1$
004646  005761          TST     2(R1)           ; CLEAR DONE
        000002
004652  032702          BIT     #60,R2          ; EXTERNAL?
        000060
004656  001002          BNE     2$              ; YES - NO START BIT
004660  052702          BIS     #1,R2           ; SET START BIT
        000001
004664  112704  2$:     MOVB    #10,R4          ; LINE COUNTER
        000010
004670                  CRLF
004672  110211  3$:     MOVB    R2,(R1)         ; START CONVERSION
004674  105711  4$:     TSTB    (R1)            ; WAIT FOR DONE
004676  100376          BPL     4$
004700  016100          MOV     2(R1),R0        ; GET DATA
        000002
004704                  OCT12                   ; DISPLAY
004706  005711          TST     (R1)            ; ERROR BIT SET?
004710  100011          BPL     5$              ; NO - SKIP
004712  112700          MOVB    #'E,R0          ; PRINT 'E'
        000105
004716                  TTYOUT
004720  005237          INC     @#ERRCNT        ; INCREMENT COUNT
        000522
004724  001003          BNE     5$              ; NO OVERFLOW
004726  012737          MOV     #-1,@#ERRCNT
        177777
        000522
004734  105304  5$:     DECB    R4              ; LINE OVER?
004736  001752          BEQ     2$              ; YES - NEW LINE
004740                  TAB
004742  000753          BR      3$              ; NEXT CONVERSION
```

```
                          .SBTTL  TEST 24: A/D DATA DISPLAY UNDER DMA
                  ;
                  ; THIS TEST ALLOWS THE USER TO ACTUALLY CHECK THE
                  ; A/D DATA ACQUIRED AT HIGH SPEED AND TRANSFERRED
                  ; DIRECTLY INTO MEMORY.
                  ;
                  ;
004744            TEST24: TEST    BIT0,SWR          ; DMA PRESENT?
004752  001001            BNE     1$                ; YES - CONTINUE
004754                    EXIT                      ; NO - EXIT
                  ;
004756            1$:     PRINTC  <A/D DATA DISPLAY UNDER DMA>
005014  004767            CALL    GETCH             ; GET CHANNEL ADDRESS
        177010
005020                    KBEXIT                    ; SET UP KEYBOARD
005022  042702            BIC     #140303,R2        ; CLEAR SOME BITS
        140303
005026  010200            MOV     R2,R0             ; GET CSR IN R0
005030  042700            BIC     #60,R0            ; NO EXTERNAL TRIGGER
        000060
005034  010011            MOV     R0,(R1)           ; SET UP CHANNEL
005036  105711    2$:     TSTB    (R1)              ; WAIT FOR DONE
005040  100376            BPL     2$
005042  005761            TST     2(R1)             ; CLEAR DONE
        000002
005046  013703            MOV     @#VECTOR,R3       ; GET VECTOR ADDRESS
        000544
005052  062703            ADD     #4,R3             ; ADJUST
        000004
005056                    RELMOV  #5$,R0            ; GET ISR ADDRESS
005064  010013            MOV     R0,(R3)           ; STORE
005066            3$:     PRINT   <FILLING BUFFER...>
005112  012761            MOV     #16000,6(R1)      ; LOAD DMCAR
        016000
        000006
005120  012761            MOV     #176777,4(R1)     ; LOAD DMWCR
        176777
        000004
005126  005761            TST     2(R1)             ; CLEAR DONE
        000002
005132  110211            MOVB    R2,(R1)           ; LOAD MODE BITS
005134  152711            BISB    #2,(R1)           ; START TRANSFER
        000002
                  ;
        005136'   OLD3    =.-2                      ; OLD REV. PATCH
                  ;
005140  000001    4$:     WAIT                      ; WAIT FOR INTERRUPT
005142  000776            BR      4$                ; HANG UNTIL RECEIVED
                  ;
                  ;  NOTE: KEYBOARDS INTERRUPTS ALSO ON, STILL PROVIDES
                  ;        ABILITY TO ABORT OUT OF THIS WAIT.
```

```
005144  105011  5$:     CLRB    (R1)                ; TURN OFF MODE BITS
005146  062706          ADD     #4,SP               ; ADJUST STACK
        000004
005152  005711          TST     (R1)                ; ERROR BIT SET?
005154  100013          BPL     6$                  ; NO - SKIP
005156                  CRLF
005160                  PRINT   <ERROR BIT IS SET>
005204  012703  6$:     MOV     #16000,R3           ; DISPLAY BUFFER
        016000
005210  012704          MOV     #1000,R4            ; CONVERSION COUNT
        001000
005214  112705  7$:     MOVB    #10,R5              ; LINE COUNTER
        000010
005220                  CRLF
005222  012300  8$:     MOV     (R3)+,R0            ; GET DATA
005224                  OCT12                       ; DISPLAY
005226  105737          TSTB    @#RCSR              ; CHARACTER?
        177560
005232  100005          BPL     10$                 ; NO - SKIP
005234                  TTYIN                       ; PROCESS
005236          9$:     CRLF                        ; NEW BUFFER
005240  005000          CLR     R0                  ; RE-ENABLE INT.
005242  106400          MTPS    R0
005244  000710          BR      3$
                ;
005246  005304  10$:    DEC     R4                  ; ALL DONE?
005250  001772          BEQ     9$                  ; YES - NEW BUFFER
005252  105305          DECB    R5                  ; LINE DONE?
005254  001757          BEQ     7$                  ; YES - LOOP
005256                  TAB                         ; MORE DATA
005260  000760          BR      8$
```

```
                                .SBTTL  TEST 25: A/D INPUT CHANNEL SCAN
                        ;
                        ; THIS TEST SCANS THE INPUT CHANNELS WHILE
                        ; DISPLAYING THE A/D DATA ON THE TERMINAL.
                        ;
                        ;
005262                  TEST25: PRINTC  <A/D INPUT CHANNEL SCAN>
005314                          KBEXIT                  ; SET UP KEYBOARD
005316  013701                  MOV     @#BASE,R1       ; GET ADDRESS
        000542
005322  005761                  TST     2(R1)           ; CLEAR DONE
        000002
005326  005011                  CLR     (R1)            ; CLEAR CSR
005330  105711          1$:     TSTB    (R1)            ; WAIT FOR DONE
005332  100376                  BPL     1$
005334  005761                  TST     2(R1)           ; CLEAR DONE
        000002
005340  005002                  CLR     R2              ; INIT. CHANNEL COUNT
005342                  2$:     CRLF
005344  112703                  MOVB    #10,R3          ; LINE COUNTER
        000010
005350                          PRINT   <CH=>           ; DISPLAY ADDRESS
005356  010200                  MOV     R2,R0
005360                          OCT8
005362  010200          3$:     MOV     R2,R0           ; GET CHANNEL ADDRESS
005364  000300                  SWAB    R0              ; PUT IN HIGH BYTE
005366  010011                  MOV     R0,(R1)         ; TRIGGER CONVERTER
005370  105711          4$:     TSTB    (R1)            ; WAIT FOR DONE
005372  100376                  BPL     4$
005374  016100                  MOV     2(R1),R0        ; GET DATA
        000002
005400                          TAB
005402                          OCT12                   ; DISPLAY
005404  105303                  DECB    R3              ; LINE DONE?
005406  001365                  BNE     3$              ; NO - CONTINUE
005410  005202                  INC     R2              ; INC. CHANNEL
005412  123702                  CMPB    @#SWR+1,R2      ; END OF RANGE?
        000541
005416  001351                  BNE     2$              ; NO - CONTINUE
005420  005002                  CLR     R2              ; YES
005422                          CRLF
005424  000746                  BR      2$
```

```
                                .SBTTL  TEST 26: A/D INPUT GAIN/CHANNEL SCAN
                        ;
                        ; THIS TEST SCANS THE INPUT CHANNELS WHILE
                        ; CHANGING THE GAIN OF THE CONVERTER.  THE
                        ; A/D DATA IS DISPLAYED ON THE TERMINAL.
                        ;
                        ;
005426                  TEST26: PRINTC  <A/D INPUT GAIN/CHANNEL SCAN>
005464                          KBEXIT                          ; SET UP KEYBOARD
005466  013701                  MOV     @#BASE,R1               ; GET ADDRESS
        000542
005472  005761                  TST     2(R1)                   ; CLEAR DONE
        000002
005476  005011                  CLR     (R1)                    ; CLEAR CSR
005500  105711          1$:     TSTB    (R1)                    ; WAIT FOR DONE
005502  100376                  BPL     1$
005504  005761                  TST     2(R1)                   ; CLEAR DONE
        000002
005510  005002                  CLR     R2                      ; INIT. CHANNEL COUNT
005512  005003                  CLR     R3                      ; INIT. GAIN
005514                  2$:     CRLF
005516  112704                  MOVB    #7,R4                   ; LINE COUNTER
        000007
005522                          PRINT   <CH=>                   ; DISPLAY ADDRESS
005530  010200                  MOV     R2,R0
005532                          OCT8
005534  112700                  MOVB    #',,R0                  ; DISPLAY GAIN
        000054
005540                          TTYOUT
005542  112700                  MOVB    #'0,R0
        000060
005546  032703                  BIT     #10,R3                  ; GS1 SET?
        000010
005552  001401                  BEQ     3$                      ; NO - SKIP
005554  105200                  INCB    R0
005556                  3$:     TTYOUT                          ; DISPLAY BIT
005560  112700                  MOVB    #'0,R0
        000060
005564  032703                  BIT     #4,R3                   ; GS0 SET?
        000004
005570  001401                  BEQ     4$                      ; NO - SKIP
005572  105200                  INCB    R0
005574                  4$:     TTYOUT                          ; DISPLAY BIT
```

```
  005576  010200  5$:     MOV     R2,R0           ; GET CHANNEL ADDRESS
  005600  000300          SWAB    R0              ; PUT IN HIGH BYTE
  005602  050300          BIS     R3,R0           ; SET GAIN BITS
  005604  010011          MOV     R0,(R1)         ; TRIGGER CONVERTER
  005606  105711  6$:     TSTB    (R1)            ; WAIT FOR DONE
  005610  100376          BPL     6$
  005612  016100          MOV     2(R1),R0        ; GET DATA
          000002
  005616                  TAB
  005620                  OCT12                   ; DISPLAY
  005622  105304          DECB    R4              ; LINE DONE?
  005624  001364          BNE     5$              ; NO - CONTINUE
  005626  005202          INC     R2              ; INC. CHANNEL
  005630  062703          ADD     #4,R3           ; INCREMENT GAIN
          000004
  005634  042703          BIC     #177763,R3      ; CLEAR EXTRA BITS
          177763
  005640  123702          CMPB    @#SWR+1,R2      ; END OF RANGE?
          000541
  005644  001323          BNE     2$              ; NO - CONTINUE
  005646  005002          CLR     R2              ; YES
  005650                  CRLF
  005652  000720          BR      2$
```

```
                                .SBTTL  TEST 27: D/A CALIBRATION
                        ;
                        ; THIS TEST ALTERNATES POSITIVE AND NEGATIVE FULL
                        ; SCALE OUTPUTS ON THE TWO DAC CHANNELS TO ALLOW
                        ; THE USER TO CALIBRATE THE DACS.
                        ;
                        ;
005654  004767  TEST27: CALL    DAC                     ; INIT. DACS
        176366
005660                  PRINTC  <D/A CALIBRATION>
005702  010461  1$:     MOV     R4,2(R1)                ; +FS, Y
        000002
005706                  TTYIN
005710  010561          MOV     R5,2(R1)                ; -FS, Y
        000002
005714                  TTYIN
005716  010261          MOV     R2,2(R1)                ; +FS, X
        000002
005722                  TTYIN
005724  010361          MOV     R3,2(R1)                ; -FS, X
        000002
005730                  TTYIN
005732  000763          BR      1$
                        ;
                        ;
                        ;
                        ;
                        ;
                                .SBTTL  TEST 30: D/A SQUARE WAVES
                        ;
                        ; THIS TEST OUTPUTS HIGH SPEED FULL SCALE
                        ; SQUARE WAVES TO BOTH DACS.
                        ;
                        ;
005734  004767  TEST30: CALL    DAC
        176306
005740                  PRINTC  <D/A SQUARE WAVES>
005764  062701          ADD     #2,R1                   ; ADJUST POINTER
        000002
005770                  KBEXIT                          ; SET UP KEYBOARD
005772  010211  1$:     MOV     R2,(R1)
005774  010411          MOV     R4,(R1)
005776  000400          BR      2$
006000  010311  2$:     MOV     R3,(R1)
006002  010511          MOV     R5,(R1)
006004  000772          BR      1$
```

```
                                .SBTTL  TEST 31: D/A RAMPS
                        ;
                        ; THIS TEST GENERATES RAMPS ON THE DAC OUTPUTS.
                        ;
                        ;
006006  004767          TEST31: CALL    DAC                     ; INIT. DACS
        176234
006012                          PRINTC  <D/A RAMPS>
006026  062701                  ADD     #2,R1                   ; ADJUST POINTER
        000002
006032                          KBEXIT                          ; SET UP KEYBOARD
006034  010211          1$:     MOV     R2,(R1)
006036  010411                  MOV     R4,(R1)
006040  005202                  INC     R2
006042  005204                  INC     R4
006044  000773                  BR      1$
                        ;
                        ;
                        ;
                                .SBTTL  TEST 32: SCOPE CONTROL OUTPUTS
                        ;
                        ; THIS TEST EXERCISES THE DAC MODE OUTPUTS AND THE
                        ; Z AXIS INTENSIFY OUTPUT.
                        ;
                        ;
006046  004767          TEST32: CALL    DAC
        176174
006052                          PRINTC  <SCOPE CONTROL OUTPUTS>
006102                          KBEXIT                          ; SET UP KEYBOARD
006104  013701                  MOV     @#BASE,R1               ; GET ADDRESS
        000542
006110  062701                  ADD     #2,R1                   ; ADJUST POINTER
        000002
006114  005002                  CLR     R2                      ; INIT. TEST REGISTER
006116  010211          1$:     MOV     R2,(R1)                 ; SET BITS
006120  062702                  ADD     #10000,R2               ; NEXT STATE
        010000
006124  000774                  BR      1$                      ; LOOP BACK
                        ;
                        ; THIS SECTION OF CODE MAKES CHANGES TO THREE LOCATIONS
                        ; IN THIS SOFTWARE RELEASE.  THIS CAUSES THE DMA TESTS TO
                        ; COMPATABLE WITH CERTAIN OLD REVISION BOARDS.  ONLY THE
                        ; DMA STARTING CONDITIONS ARE MODIFIED.
                        ;
006126  012767          PATCH:: MOV     #3,OLD1                 ; PATCH 1
        000003
        175226
006134  012767                  MOV     #3,OLD2                 ; PATCH 2
        000003
        175552
006142  012767                  MOV     #3,OLD3                 ; PATCH 3
        000003
        176766
006150                          ESCAPE                          ; RETURN TO TST-11
                        ;
        000001                  .END
```

```
SYMBOL TABLE

BASE  = 000542          INITH   000212R     002 TEST14  002712R     002
BIT0  = 000001          INITL   000262R     002 TEST15  003036R     002
BIT1  = 000002          INIT61  000154R     002 TEST16  003206R     002
BIT10 = 002000          INIT62  000164R     002 TEST17  003540R     002
BIT11 = 004000          INIT64  000172R     002 TEST2   001416R     002
BIT12 = 010000          INIT65  000202R     002 TEST20  004026R     002
BIT13 = 020000          I1761 = 000154RG    002 TEST21  004332R     002
BIT14 = 040000          I1762 = 000164RG    002 TEST22  004464R     002
BIT15 = 100000          I1764 = 000172RG    002 TEST23  004576R     002
BIT2  = 000004          I1765 = 000202RG    002 TEST24  004744R     002
BIT3  = 000010          LF    = 000012          TEST25  005262R     002
BIT4  = 000020          LPERR = 000002          TEST26  005426R     002
BIT5  = 000040          LPTST = 000001          TEST27  005654R     002
BIT6  = 000100          NEWFLG= 010000          TEST3   001446R     002
BIT7  = 000200          NODMA   000312R     002 TEST30  005734R     002
BIT8  = 000400          NORPLY  001014R     002 TEST31  006006R     002
BIT9  = 001000          ODTACC= 000514          TEST32  006046R     002
CR    = 000015          OLD1  = 003362R     002 TEST4   001650R     002
CTRLC = 000003          OLD2  = 003714R     002 TEST5   001720R     002
CTRLI = 000011          OLD3  = 005136R     002 TEST6   002060R     002
DAC     004246R     002 PARAM   000722R     002 TEST7   002136R     002
DACODE  000663R     002 PATCH   006126RG    002 TPB     000000R     002
DELAY = 000546          PR7   = 000340          TSTALL= 000010
DMA     000304R     002 QUERY   000464R     002 TSTCSR= 000500
DMAERR  001230R     002 RBUF  = 177562          TSTNUM= 000520
DMAOPT  000631R     002 RCSR  = 177560          TSTONE= 000040
DT5701  000244R     002 REG     001070R     002 TSTSEQ= 000020
DT5710  000226R     002 SPACE = 000040          T1761 = 000000RG    002
ERRCNT= 000522          SPEED   000606R     002 T1762 = 000000RG    002
ERRNUM= 000521          SWR   = 000540          T1764 = 000000RG    002
FF    = 000014          TEST1   001326R     002 T1765 = 000000RG    002
GETCH   004030R     002 TEST10  002216R     002 VECTOR= 000544
HLTERR= 000004          TEST11  002320R     002 XBUF  = 177566
INHERR= 000100          TEST12  002416R     002 XCSR  = 177564
INIT    000270R     002 TEST13  002600R     002

. ABS.  000000     000
        000000     001
DT1760  006152     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  2238 WORDS  ( 9 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  74 PAGES
DX0:1760,DX1:1760=DX1:TST11.ML/M,DX1:1760.V01
```

APPENDIX E

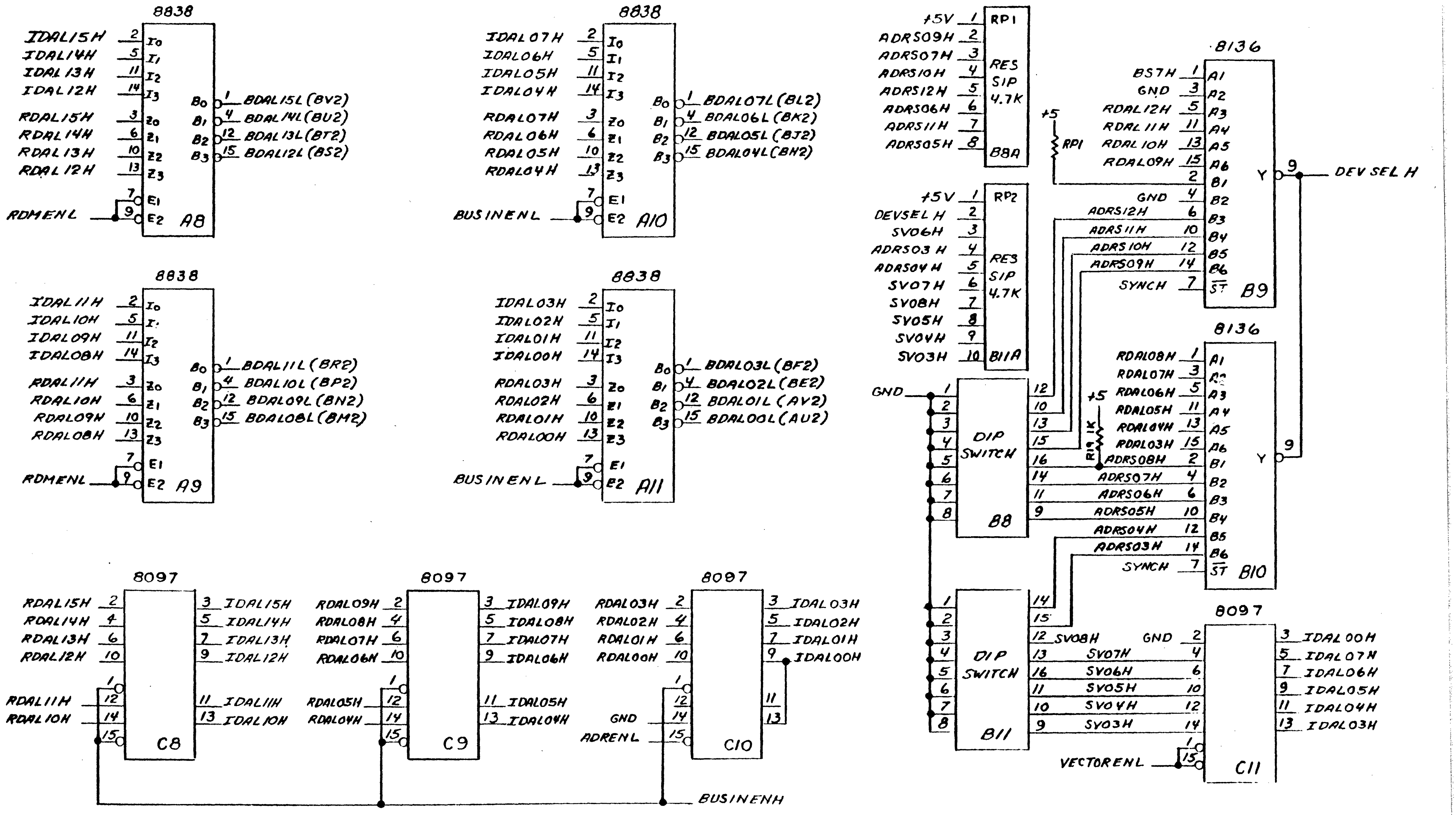
LOGIC DRAWINGS

F-1	236	2-14-80	K.Y.C.	K.E.D
F	150	2-29-79	K.Y.C.	K.E.D.
REVISION	ECO NO.	DATE	DWN	ENG

DATA TRANSLATION INC

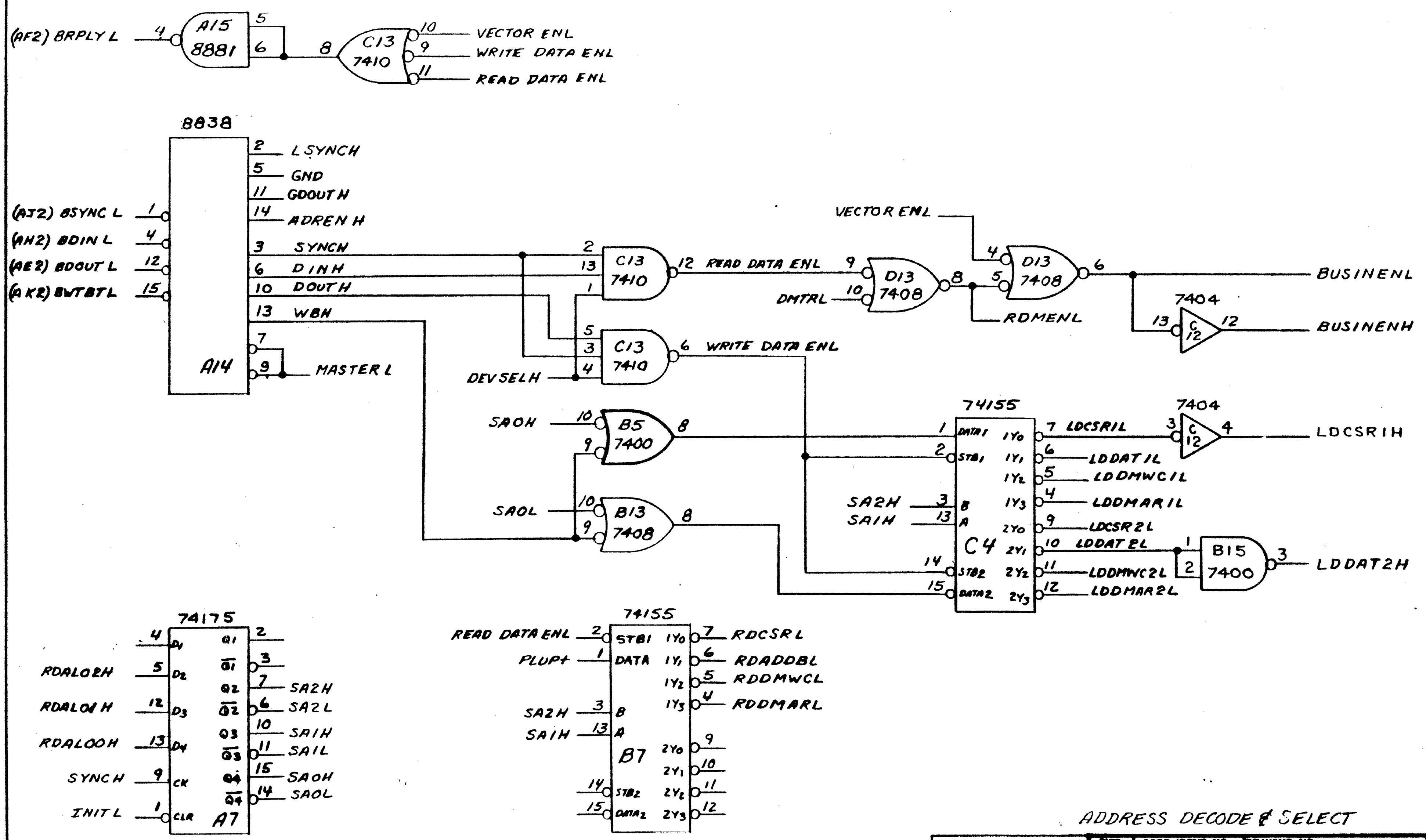
DRAWN: K.E.D.	DATE: 8-29-78	TITLE: SCHEMATIC LSI 11 ANALOG I/O SYSTEM DT 1761
CHECKED: P.J.S.	DATE. 8-30-78	
ENG: P.J.S.	DATE: 8-30-78	

EP033	SHT. 1 OF 11	SIZE B	DWG.NO. 21020033	REV F-1



SENSOR BASED ACQUISITION AND CONTROL UNIT

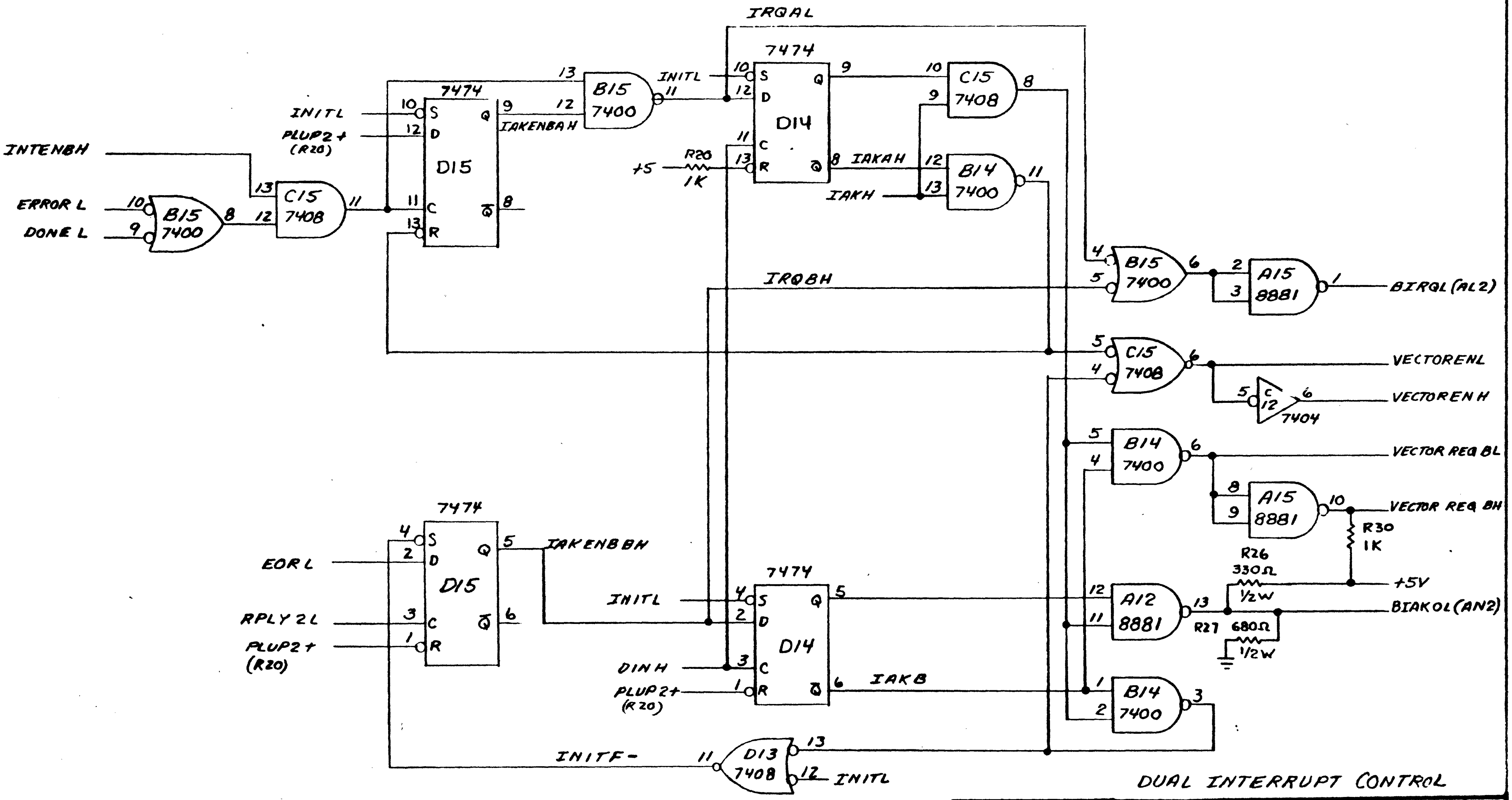
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 2	



ADDRESS DECODE & SELECT

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION, INC. AND SHALL NOT BE REPRODUCED, COPIED, OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION, INC

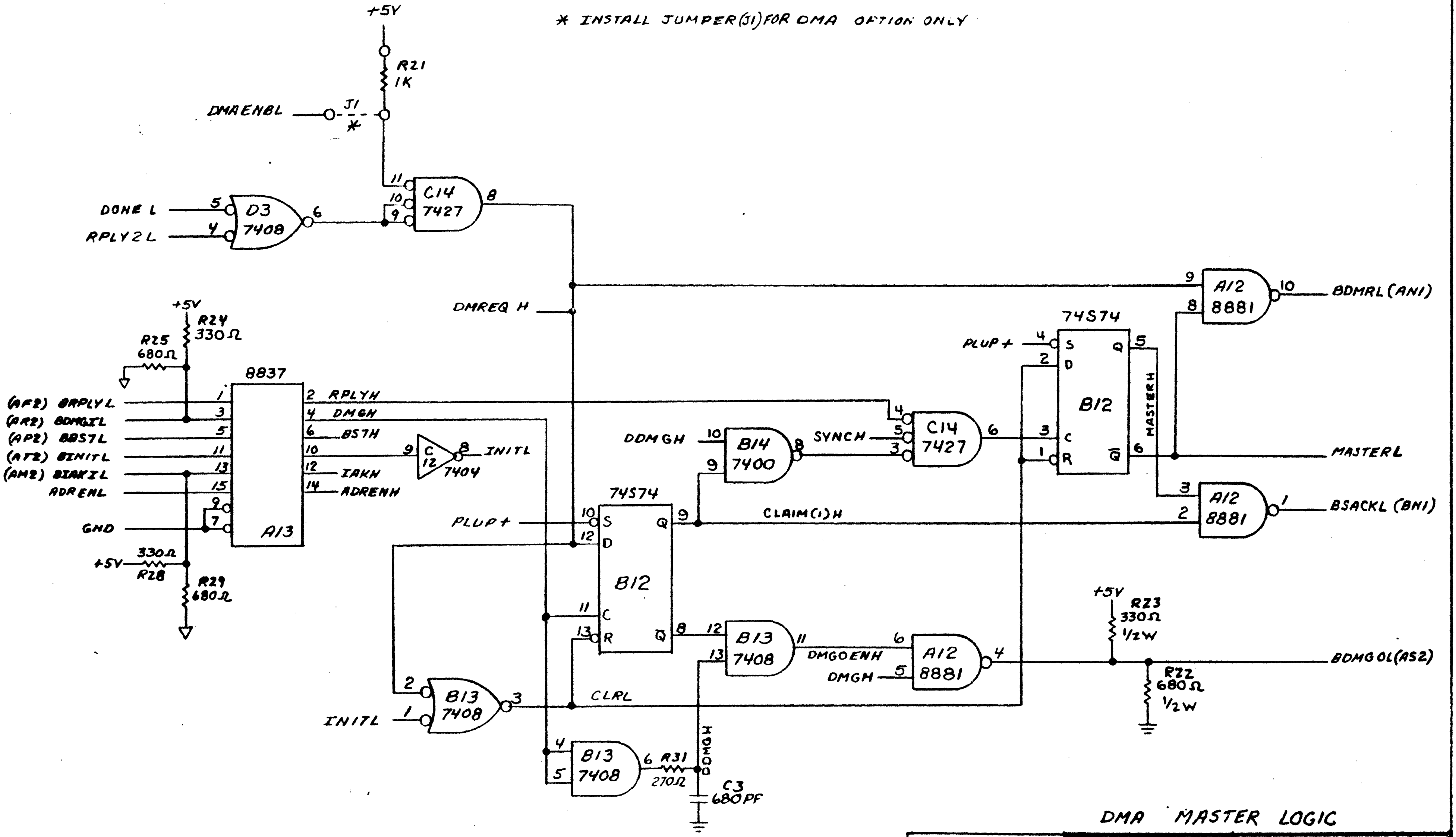
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 3	



DUAL INTERRUPT CONTROL

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION, INC AND SHALL NOT BE REPRODUCED, COPIED, OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION, INC

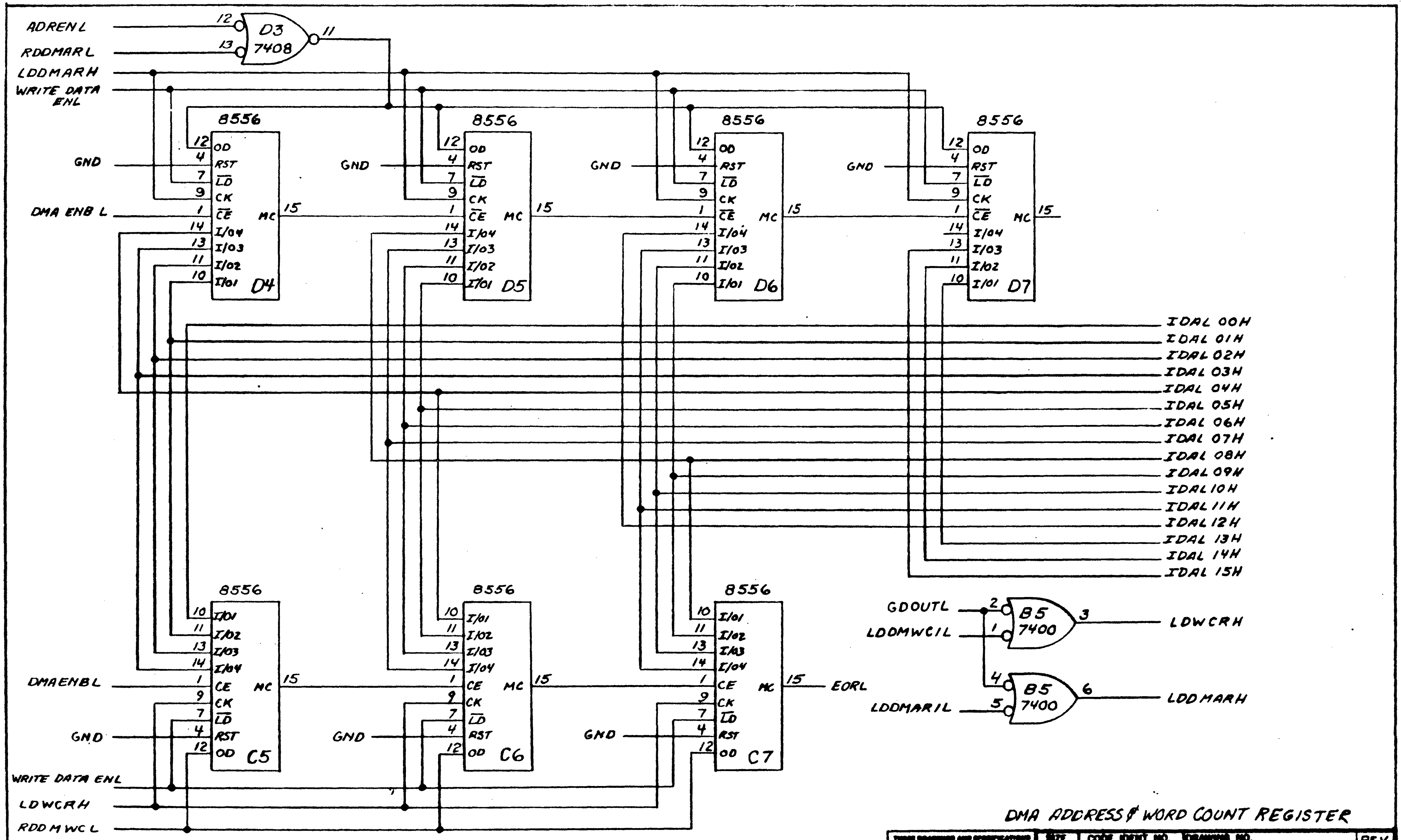
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE			SHEET 4



✱ INSTALL JUMPER (J1) FOR DMA OPTION ONLY

DMA MASTER LOGIC

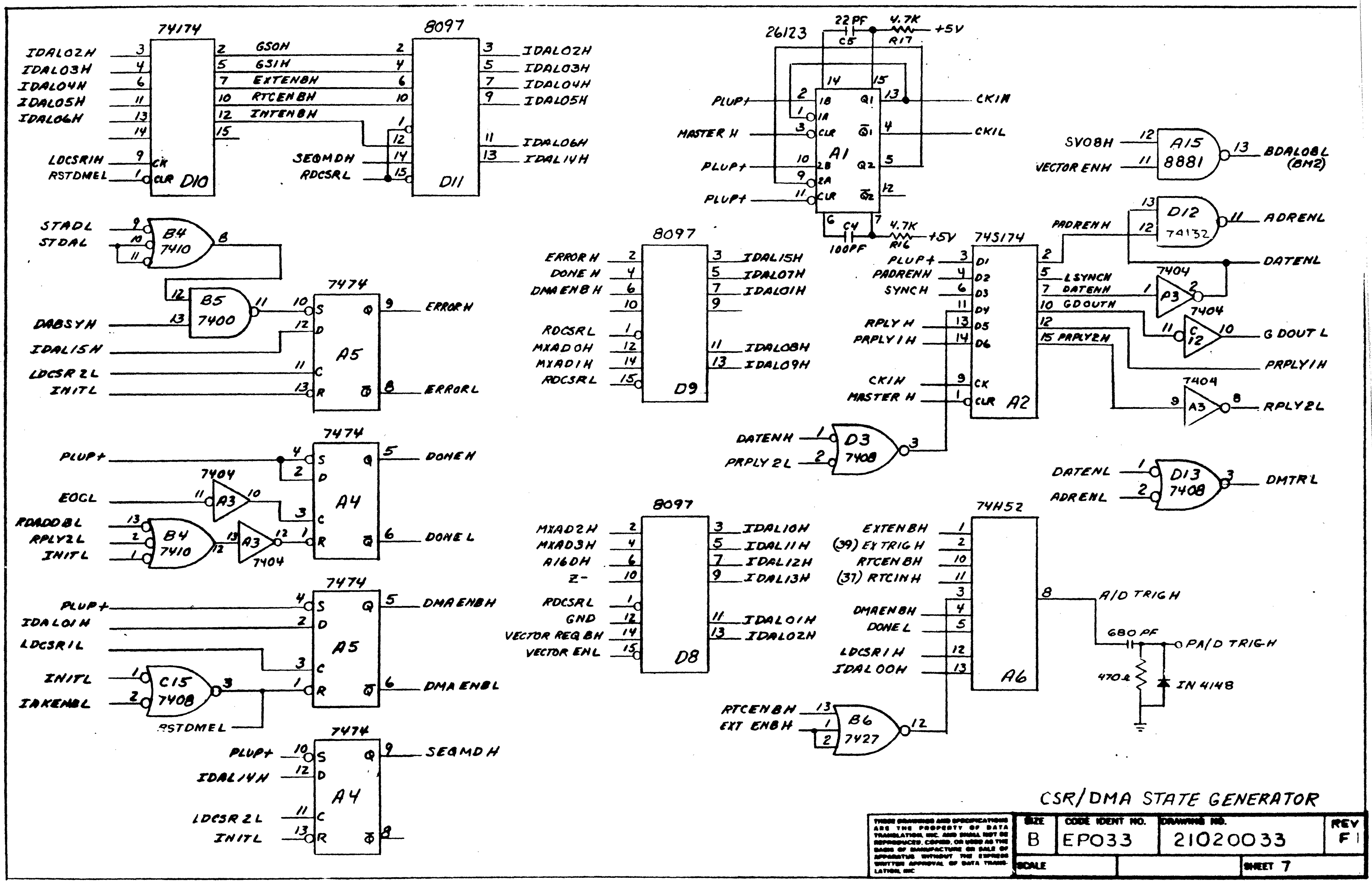
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 5	



DMA ADDRESS & WORD COUNT REGISTER

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION, INC. AND SHALL NOT BE REPRODUCED, COPIED, OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION, INC.

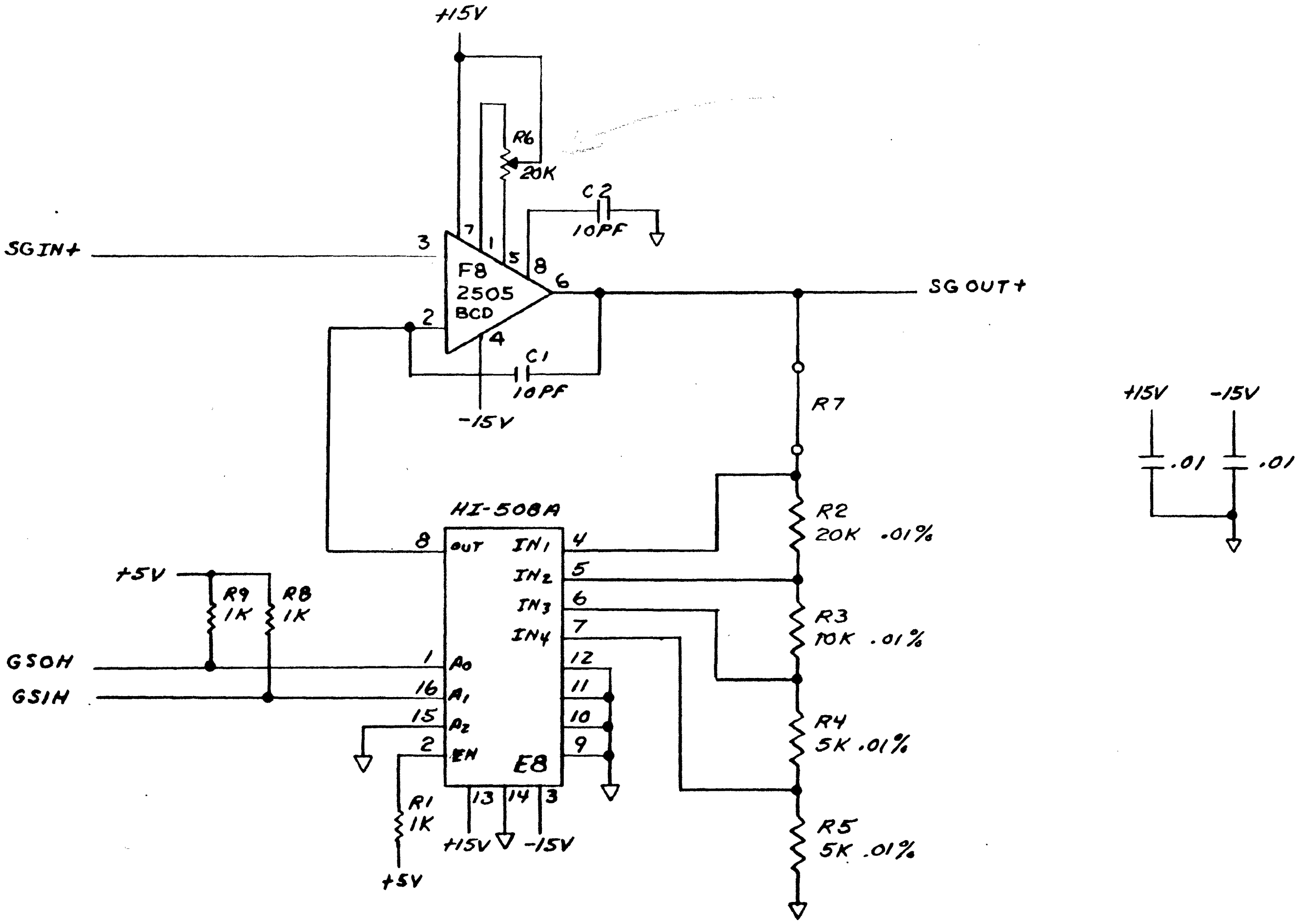
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 6	



CSR/DMA STATE GENERATOR

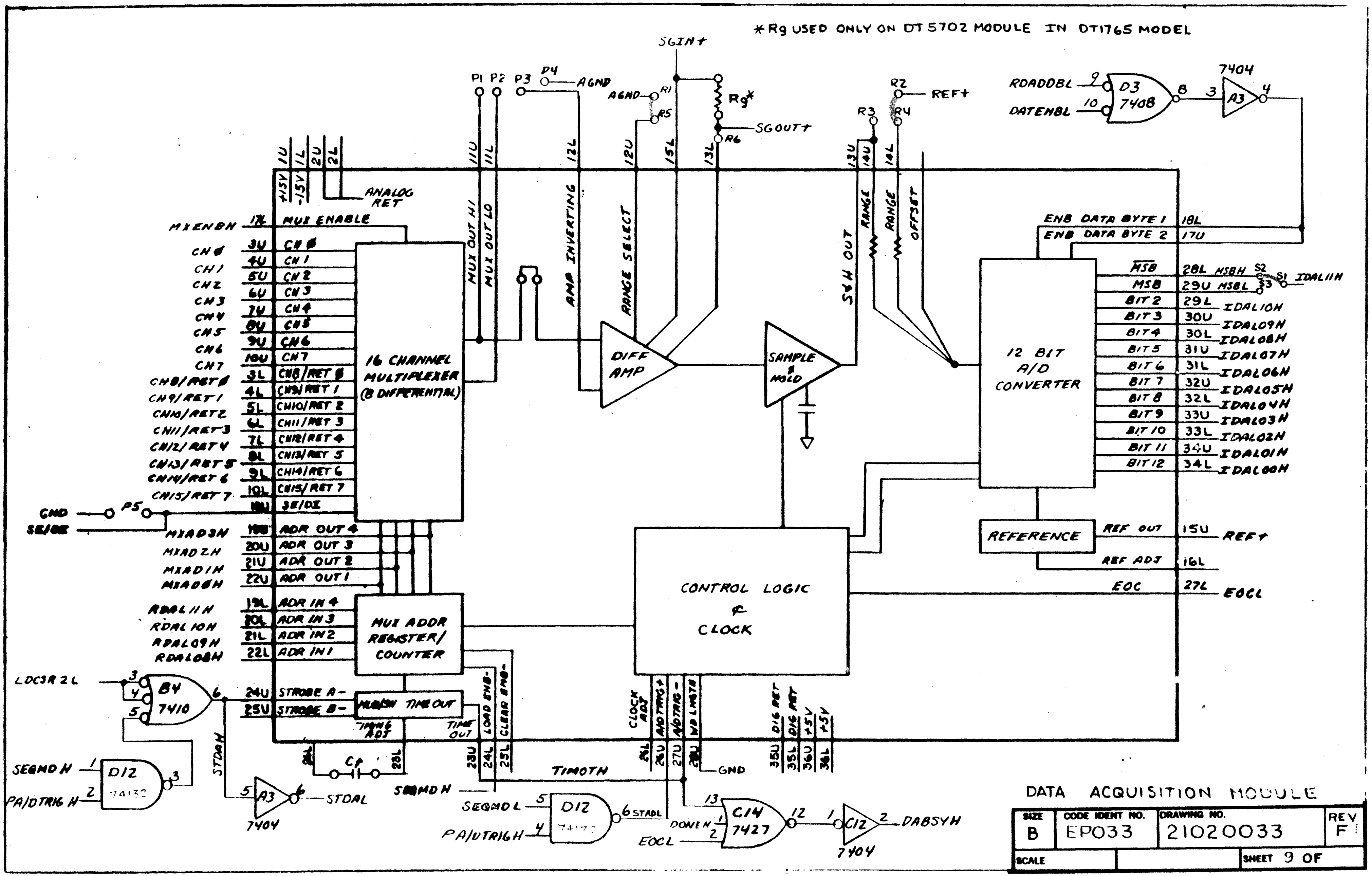
THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION, INC. AND SHALL NOT BE REPRODUCED, COPIED, OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION, INC.

SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EPO33	21020033	F1
SCALE		SHEET 7	



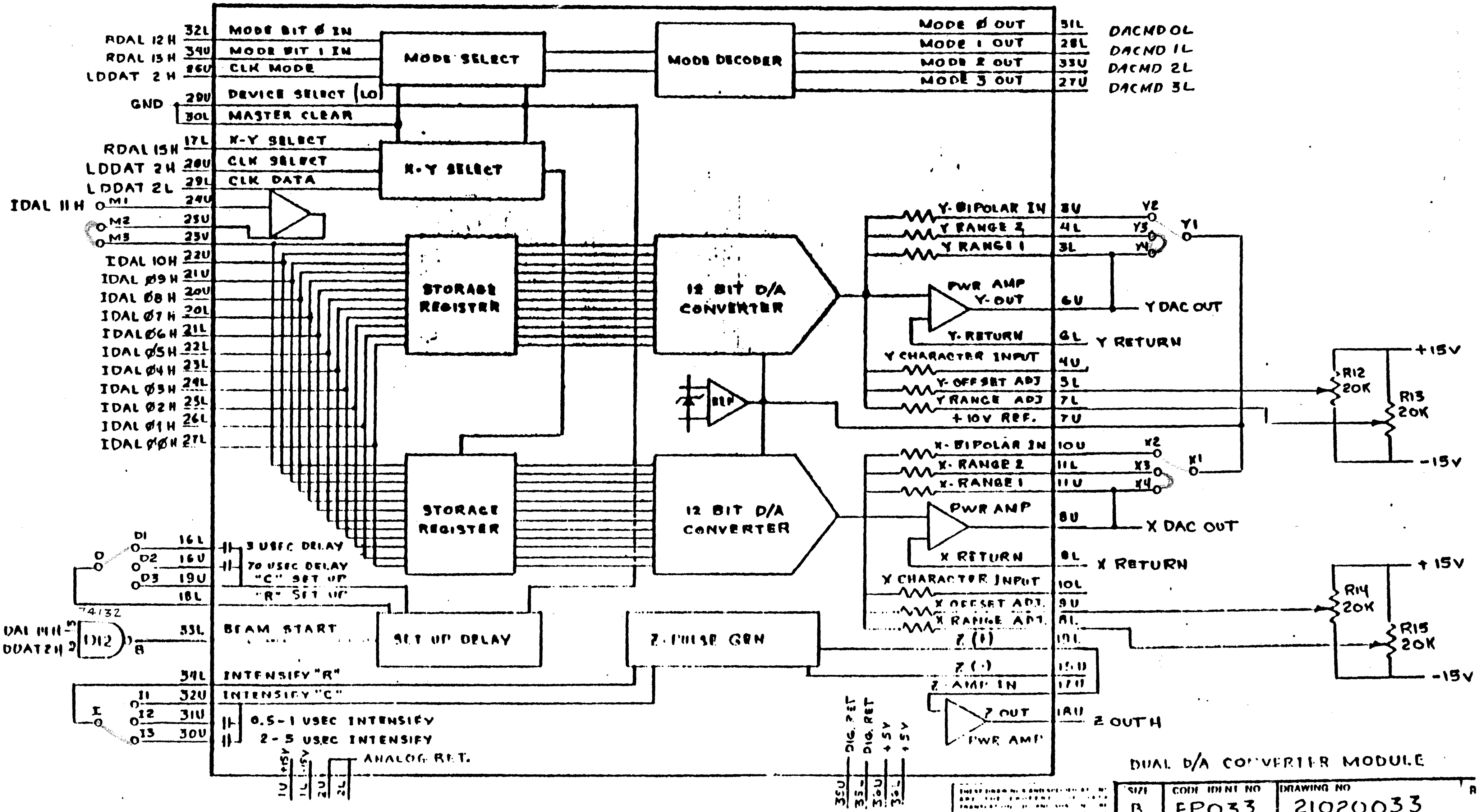
PROG: GAIN AMP

SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 8	



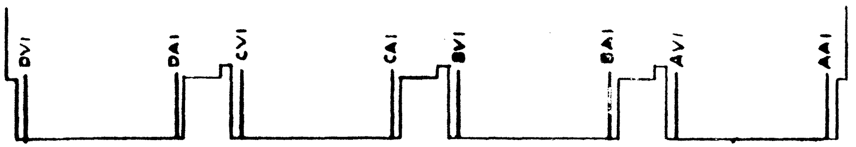
DATA ACQUISITION MODULE

SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 9 OF	



DUAL D/A CONVERTER MODULE

SIZE	CODE IDENT NO	DRAWING NO
B	EP033	21020033
SCALE		SHEET 10 OF



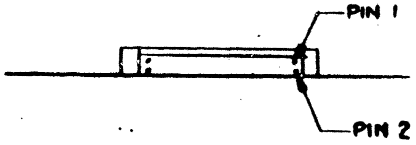
COMPONENT SIDE

COMPONENT SIDE		SOLDER SIDE		COMPONENT SIDE		SOLDER SIDE	
PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME
AA1		AA2	+5V	BA1	BDCOK H	BA2	+5V
AB1		AB2	-12V	BB1	BPOK H	BB2	-12V
AC1	BAD16	AC2	D GND	BC1		BC2	D GND
AD1	BAD17	AD2	+12V	BD1		BD2	+12V
AE1		AE2	BD OUTL	BE1		BE2	BDAL2L
AF1		AF2	BRPLY L	BF1		BF2	BDAL3L
AH1		AH2	BDIN L	BH1		BH2	BDAL4L
AJ1	D GND	AJ2	BSYNC L	BJ1	D GND	BJ2	BDAL 5L
AK1		AK2	BWTBT L	BK1		BK2	BDAL 6L
AL1		AL2	BIRQL	BL1		BL2	BDAL 7L
AM1	D GND	AM2	BIAKI L	BM1	D GND	BM2	BDAL 8L
AN1	BDMRL	AN2	BIAKOL	BN1	BSACK L	BN2	BDAL 9L
AP1	BHALT L	AP2	BBS7 L	BP1		BP2	BDAL 10L
AR1	BREF L	AR2	BDMGI L	BR1	BEVNT L	BR2	BDAL 11L
AS1		AS2	BDMGOL	BS1		BS2	BDAL 12L
AT1	D GND	AT2	BINIT L	BT1	D GND	BT2	BDAL 13L
AU1		AU2	BDALOL	BU1		BU2	BDAL 14L
AV1	+5B	AV2	BDAL1 L	BV1	+5V	BV2	BDAL 15L

COMPONENT SIDE		SOLDER SIDE		COMPONENT SIDE		SOLDER SIDE	
PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME
CA1		CA2	+5V	DA1		DA2	+5V
CB1		CB2	-12V	DB1		DB2	-12V
CC1		CC2	D GND	DC1		DC2	D GND
CD1		CD2	+12V	DD1		DD2	+12V
CE1		CE2		DE1		DE2	
CF1		CF2		DF1		DF2	
CH1		CH2		DH1		DH2	
CJ1	D GND	CJ2		DJ1	D GND	DJ2	
CK1		CK2		DK1		DK2	
CL1		CL2		DL1		DL2	
CM1	D GND	CM2		DM1	D GND	DM2	
CN1		CN2		DN1		DN2	
CP1		CP2		DP1		DP2	
CR1		CR2		DR1		DR2	
CS1		CS2		DS1		DS2	
CT1	D GND	CT2		DT1	D GND	DT2	
CU1		CU2		DU1		DU2	
CV1	+5B	CV2		DV1		DV2	

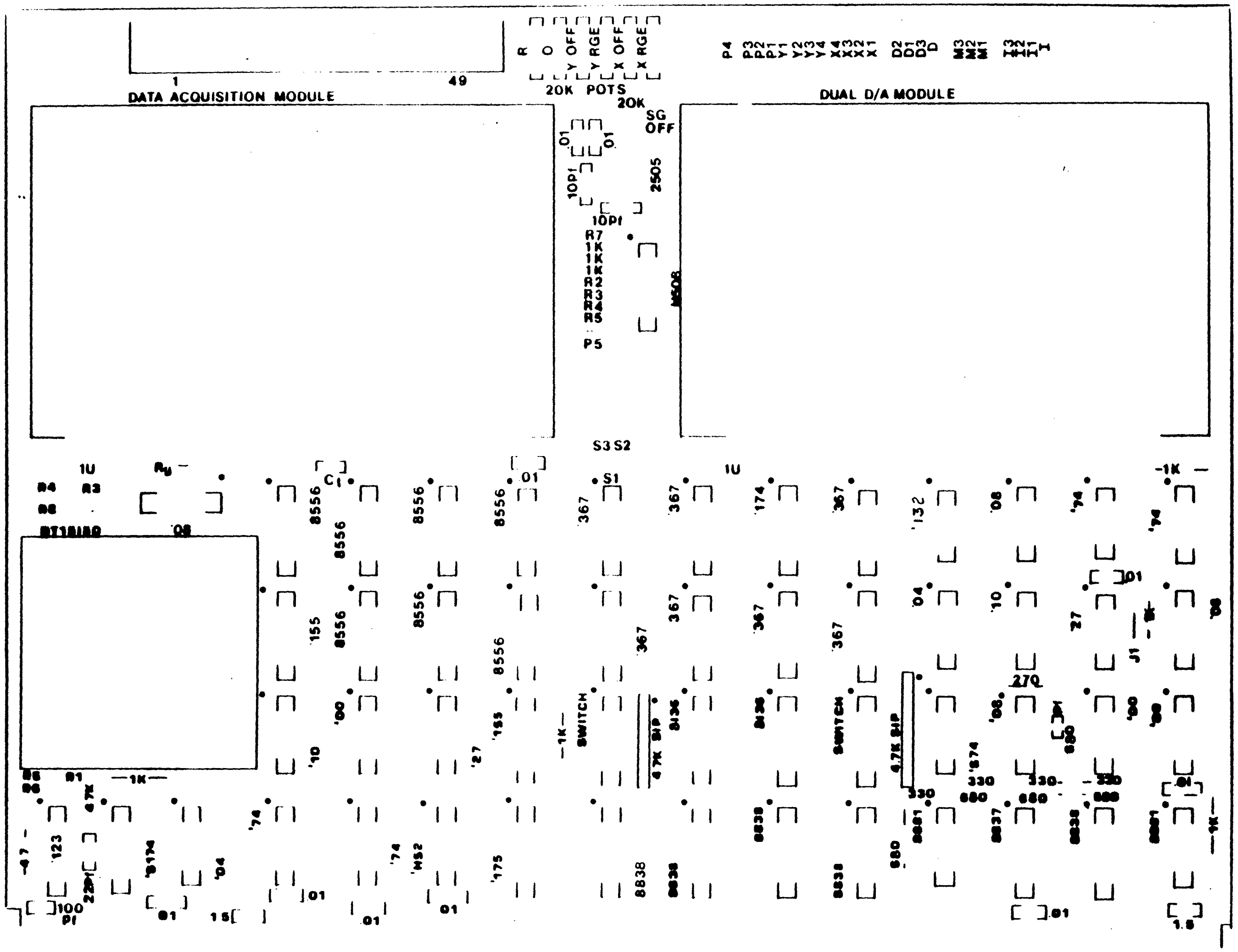
COMPONENT SIDE

J1			
PIN	SIG. NAME	PIN	SIG NAME
1	CHØ	2	A GND
3	CH8/RET Ø	4	A GND
5	CH1	6	A GND
7	CH9/RET1	8	A GND
9	CH2	10	A GND
11	CH10/RET2	12	A GND
13	CH3	14	A GND
15	CH11/RET3	16	A GND
17	CH4	18	A GND
19	CH12/RET4	20	A GND
21	CH5	22	A GND
23	CH13/RET5	24	A GND
25	CH6	26	A GND
27	CH14/RET6	28	A GND
29	CH7	30	A GND
31	CH15/RET7	32	A GND
33	Y DAC OUT	34	A GND
35	X DAC OUT	36	A GND
37	RTC IN	38	D GND
39	EXT TRIG IN	40	D GND
41	Z OUT	42	D GND
43	DAC MDØ	44	D GND
45	DAC MD1	46	D GND
47	DAC MD2	48	D GND
49	DAC MD3	50	D GND



[illegible] PROPERTY OF DATA [illegible] OF APPARATUS WITHOUT THE [illegible] WRITTEN APPROVAL OF DATA TRANSLATION INC

SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP033	21020033	F
SCALE		SHEET 11 OF 11	



COMPONENT DESIGNATIONS

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION, INC AND SHALL NOT BE REPRODUCED, COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION, INC

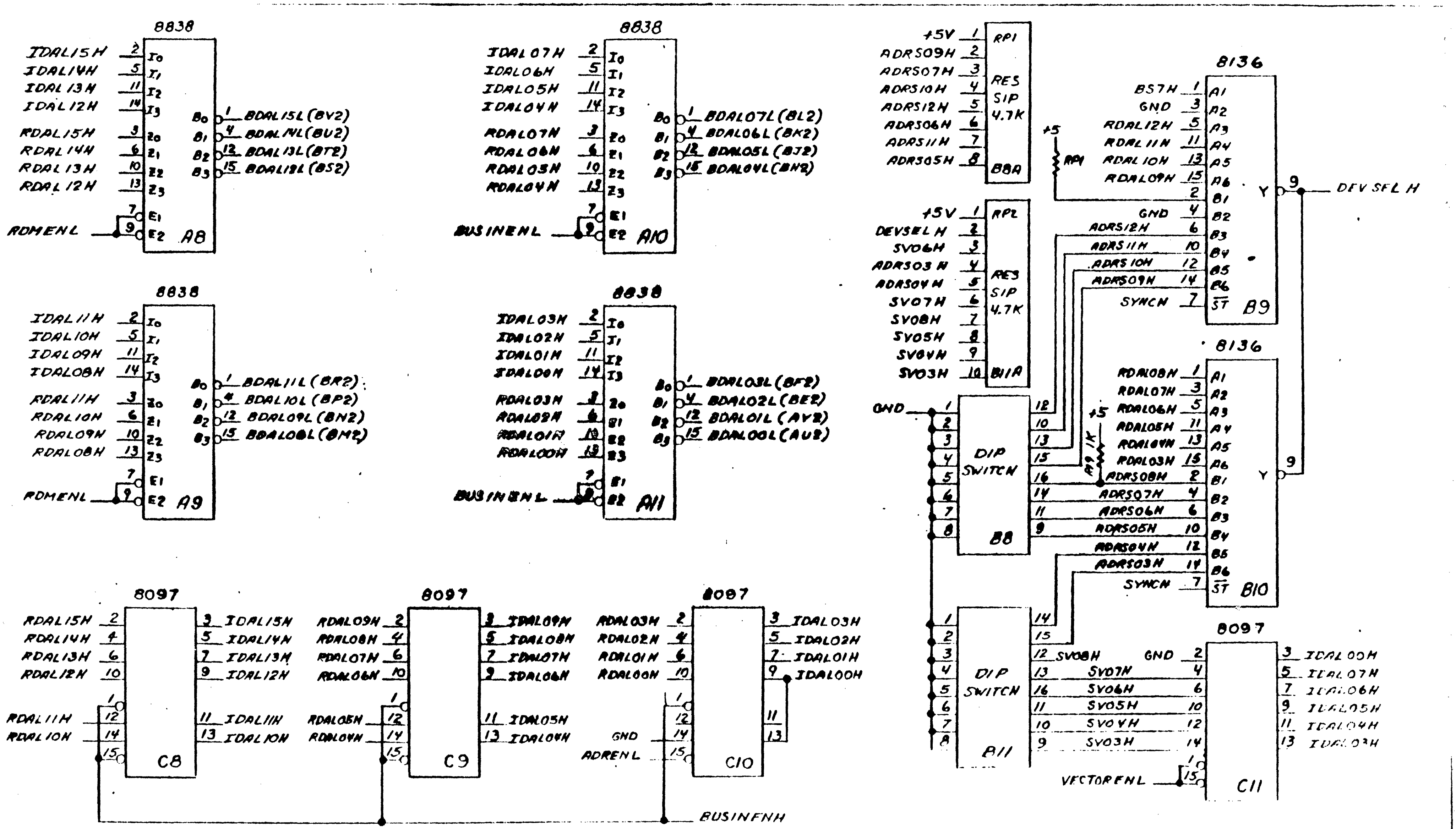
SIZE	CODE IDENT NO.	DRAWING NO.	REV
A	EP033	11070033	F-1
SCALE		SHEET 1 OF 1	

REVISION	ECO NO.	DATE	DWN	ENG
D-2	236	2-14-80	K.F.C	K.E.D.
D-1	216	12-3-79	K.F.C.	K.E.D.
D	151	2-7-79	K.F.C.	K.E.D.

DATA TRANSLATION INC

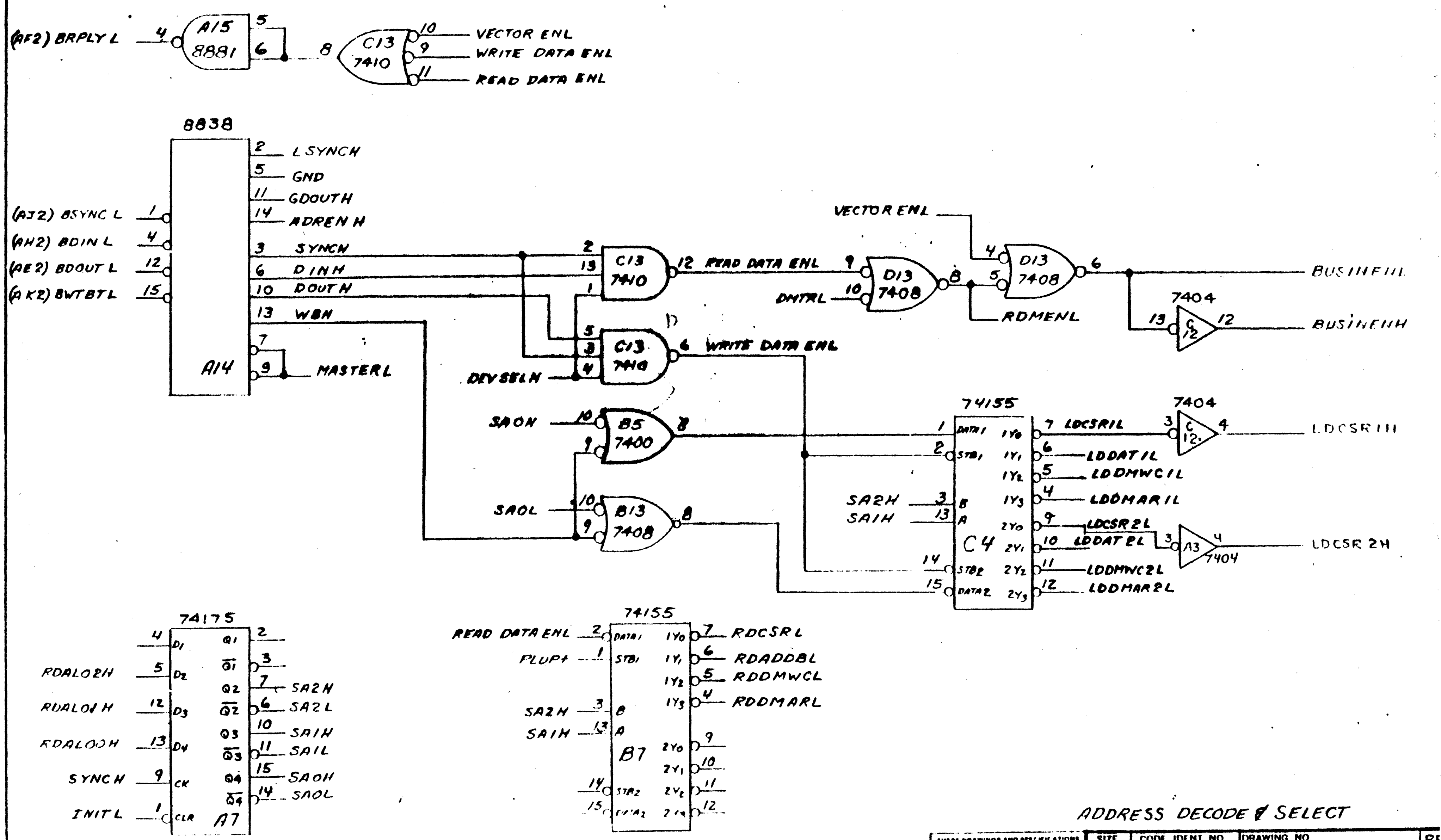
DRAWN: K.E.D.	DATE: 8-23-78	TITLE: SCHFMATIC LSI II ANALOG INPUT SYSTEM DT 1762
CHECKED: P.J.S.	DATE. 9-6-78	
ENG: P.J.S.	DATE: 9-6-78	

EP039	SHT. 1 OF 11	SIZE B	DWG.NO. 21020039	REV. D2



BUS XCVRS AND ADDRESS DECODERS

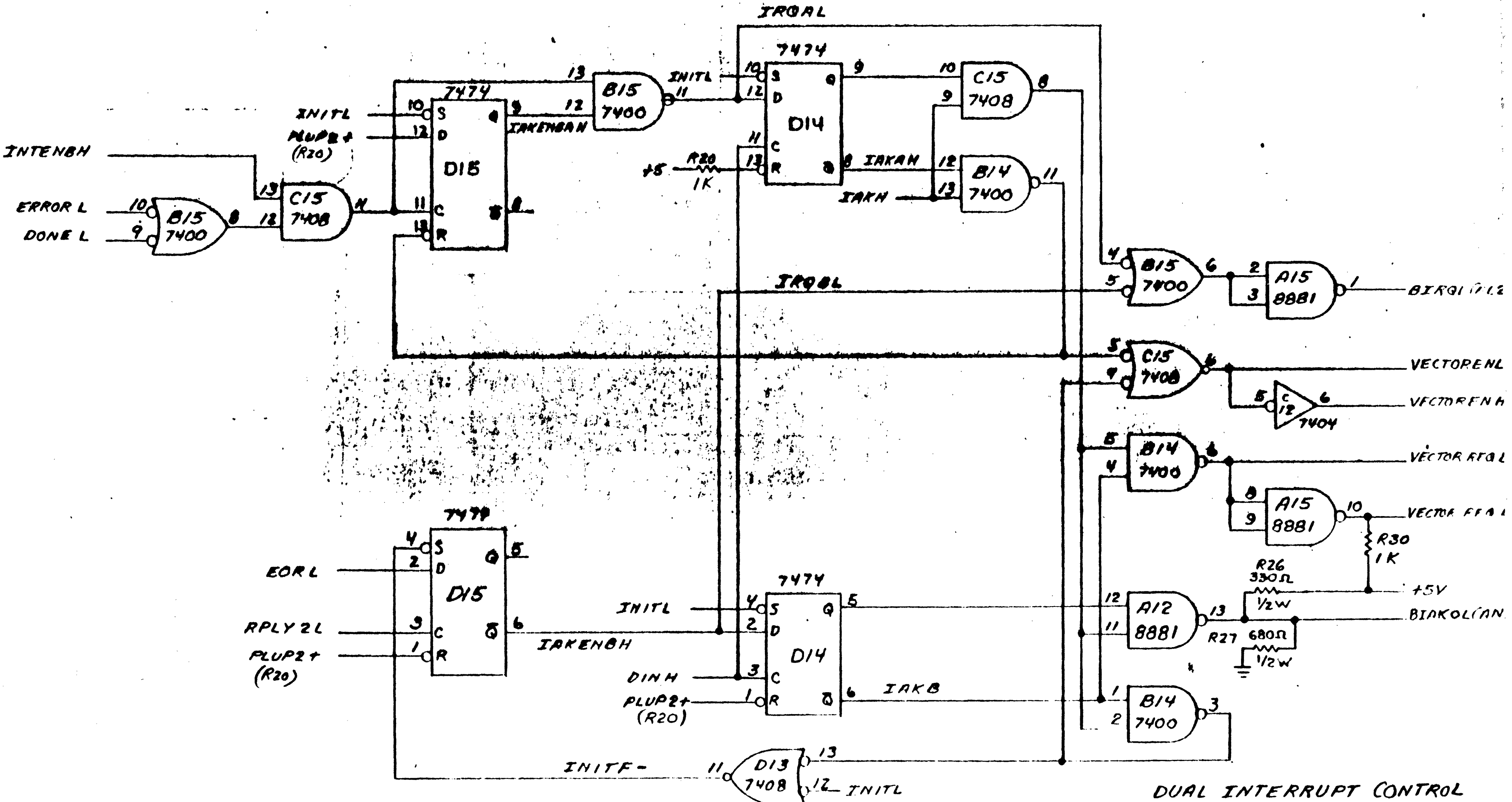
THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC. AND SHALL NOT BE REPRODUCED, COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC.	SIZE B	CODE IDENT NO. EF039	DRAWING NO. 21020039	REV D2
	SCALE			SHEET 2



ADDRESS DECODE & SELECT

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC AND SHALL NOT BE REPRODUCED COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC

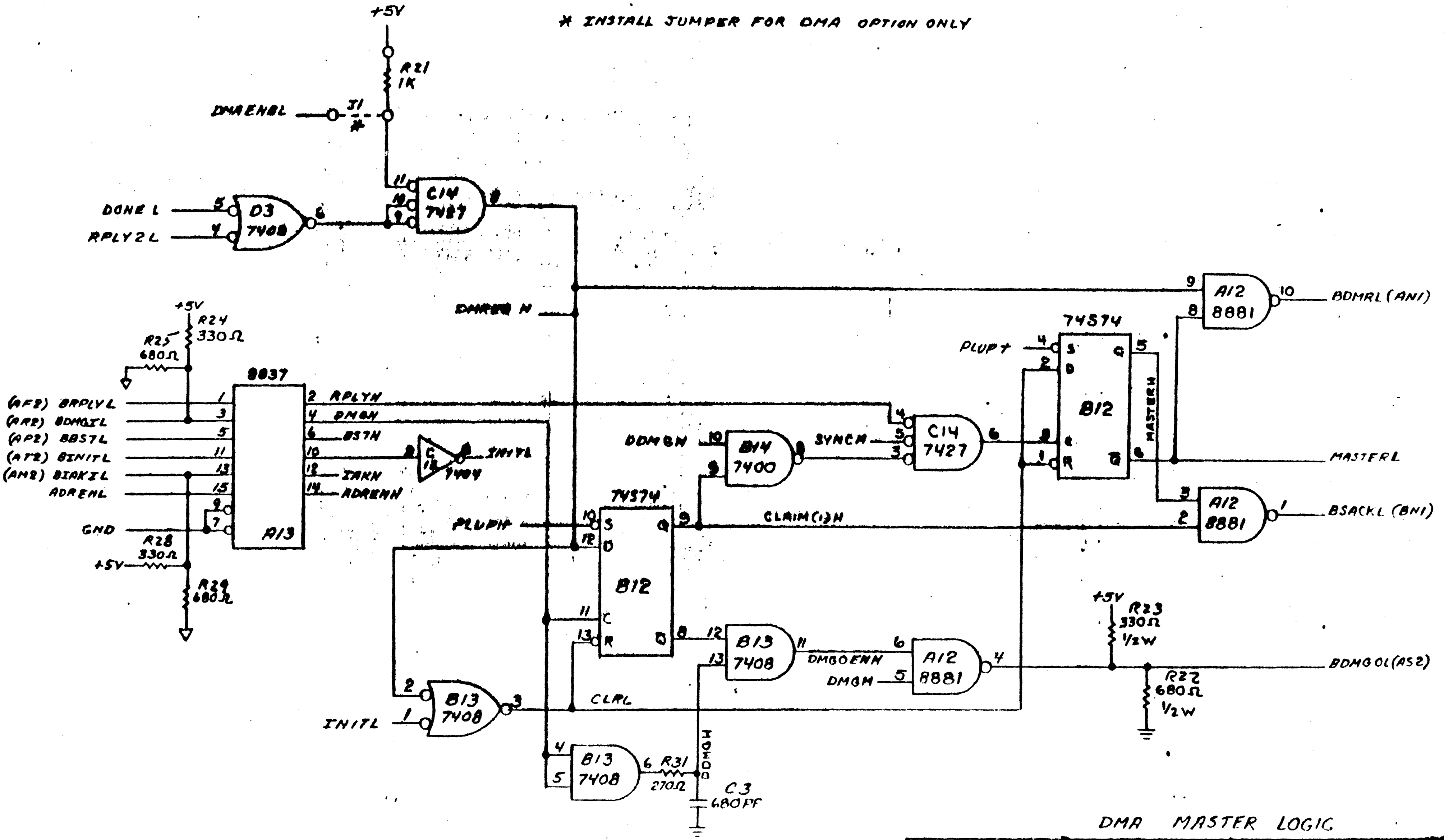
SIZE	CODE IDENT NO.	DRAWING NO.	RE
B	EP039	21020039	D
SCALE		SHEET 3	



DUAL INTERRUPT CONTROL

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC AND SHALL NOT BE REPRODUCED COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC

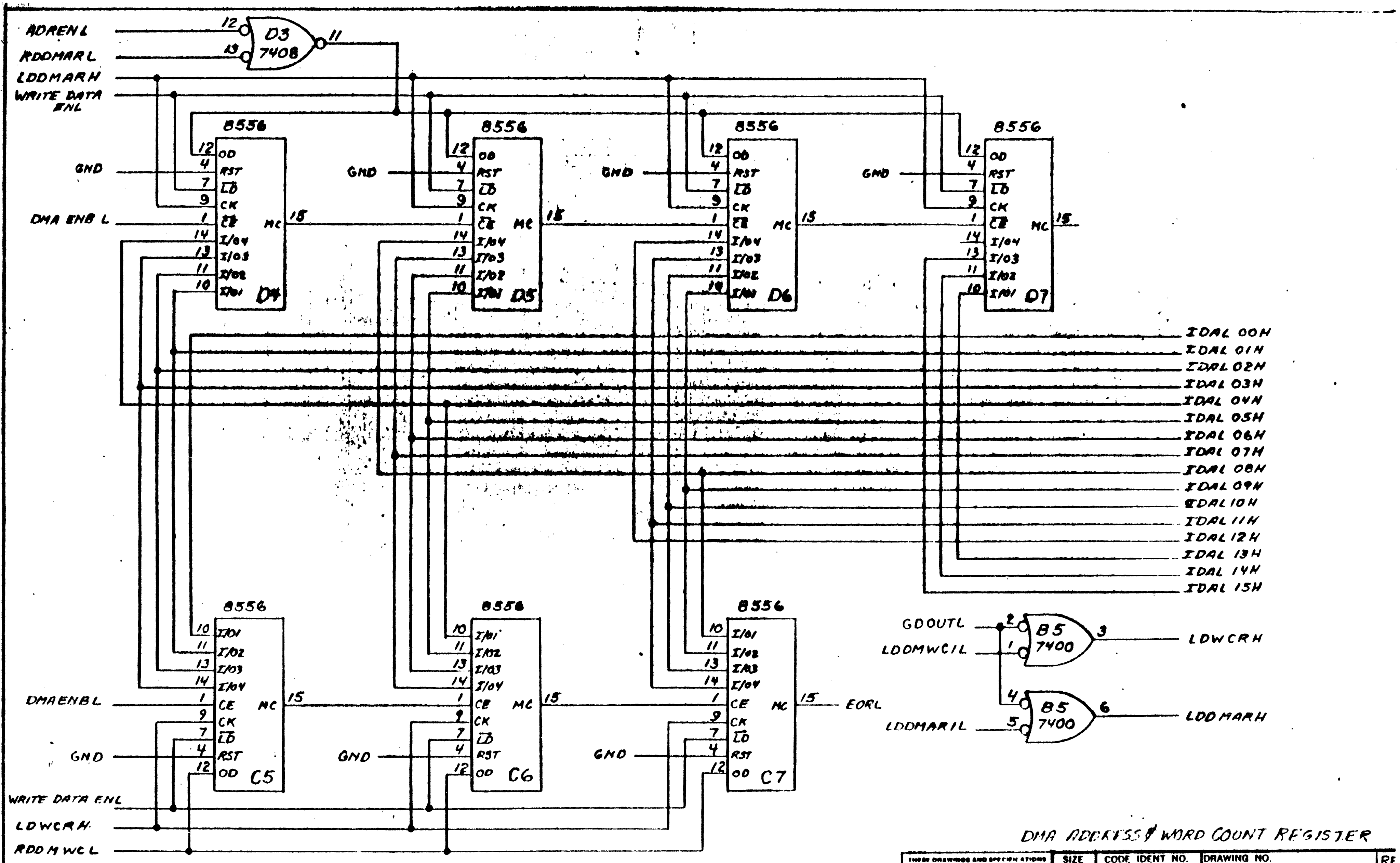
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	1P039	21020039	D
SCALE		SHEET 4	



DMA MASTER LOGIC

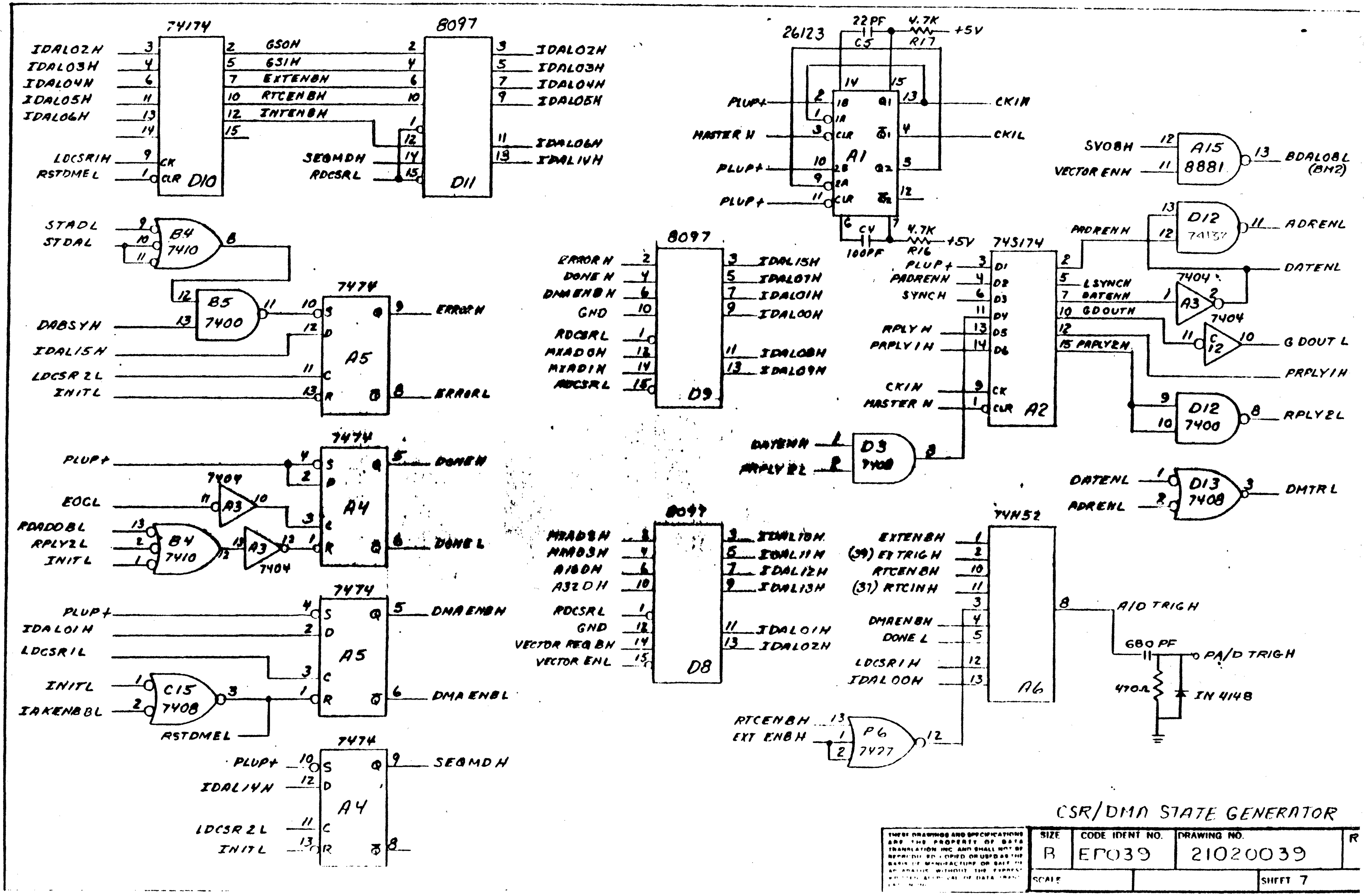
THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC AND SHALL NOT BE REPRODUCED COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC

SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP039	21020039	D2
SCALE		SHEET 5	



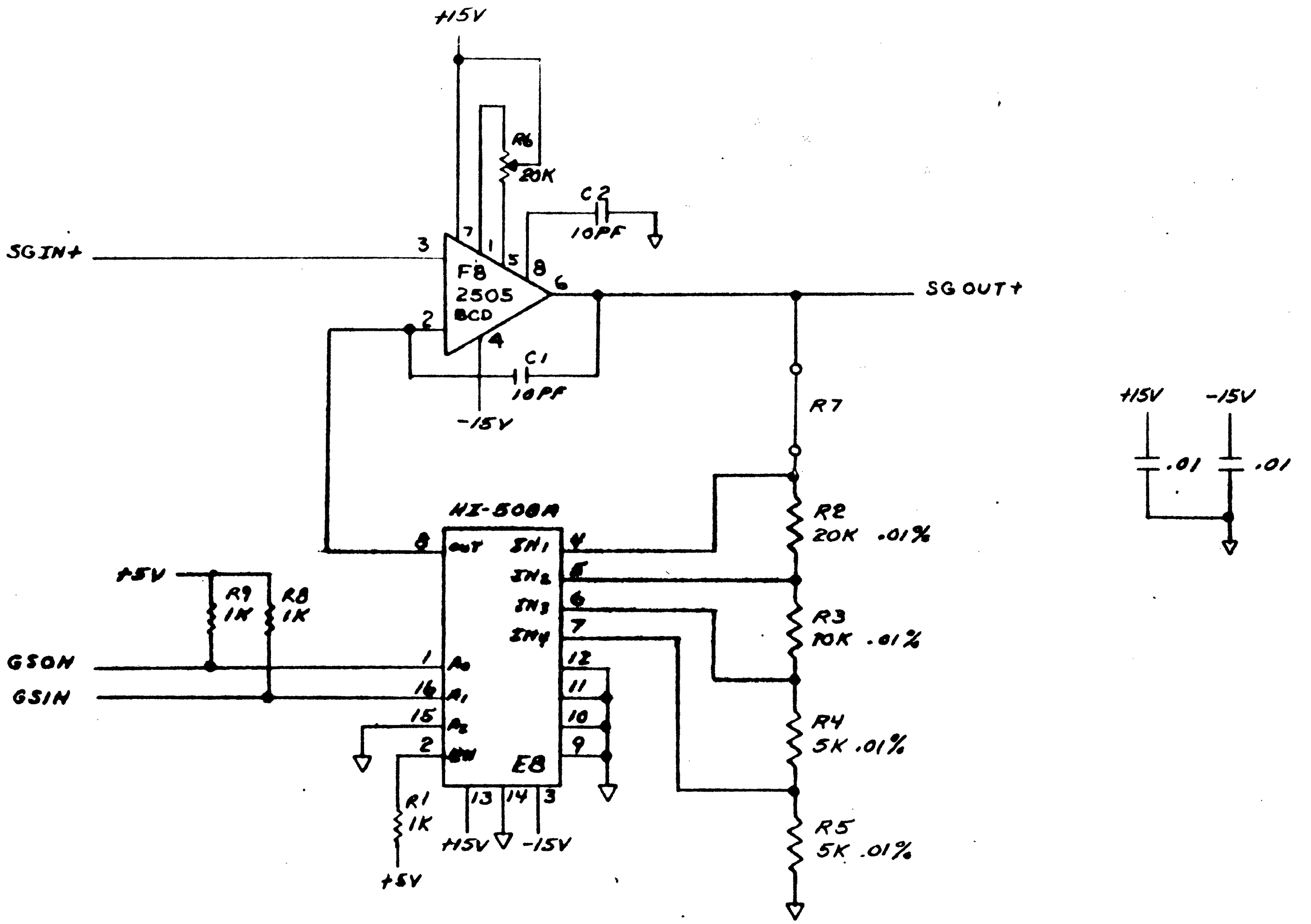
DMA ADDRESS & WORD COUNT REGISTER

SIZE	CODE IDENT NO.	DRAWING NO.	RE
B	FPO39	21020039	C
SCALE		SHEET 6	



CSR/DMA STATE GENERATOR

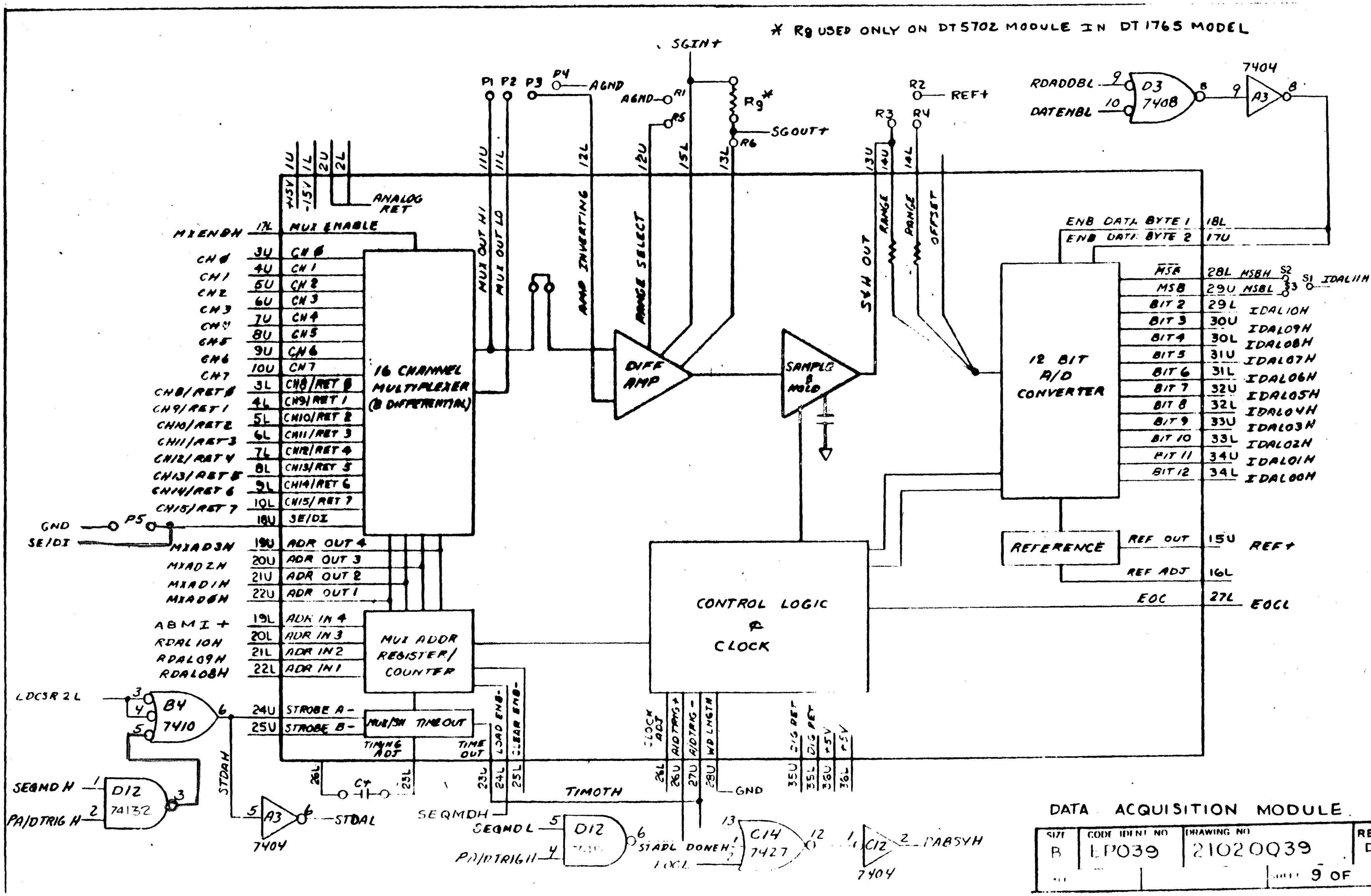
THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC AND SHALL NOT BE REPRODUCED, COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC.	SIZE	CODE IDENT NO.	DRAWING NO.	R
	B	EP039	21020039	
	SCALE			SHEET 7



PROG. GAIN AMP

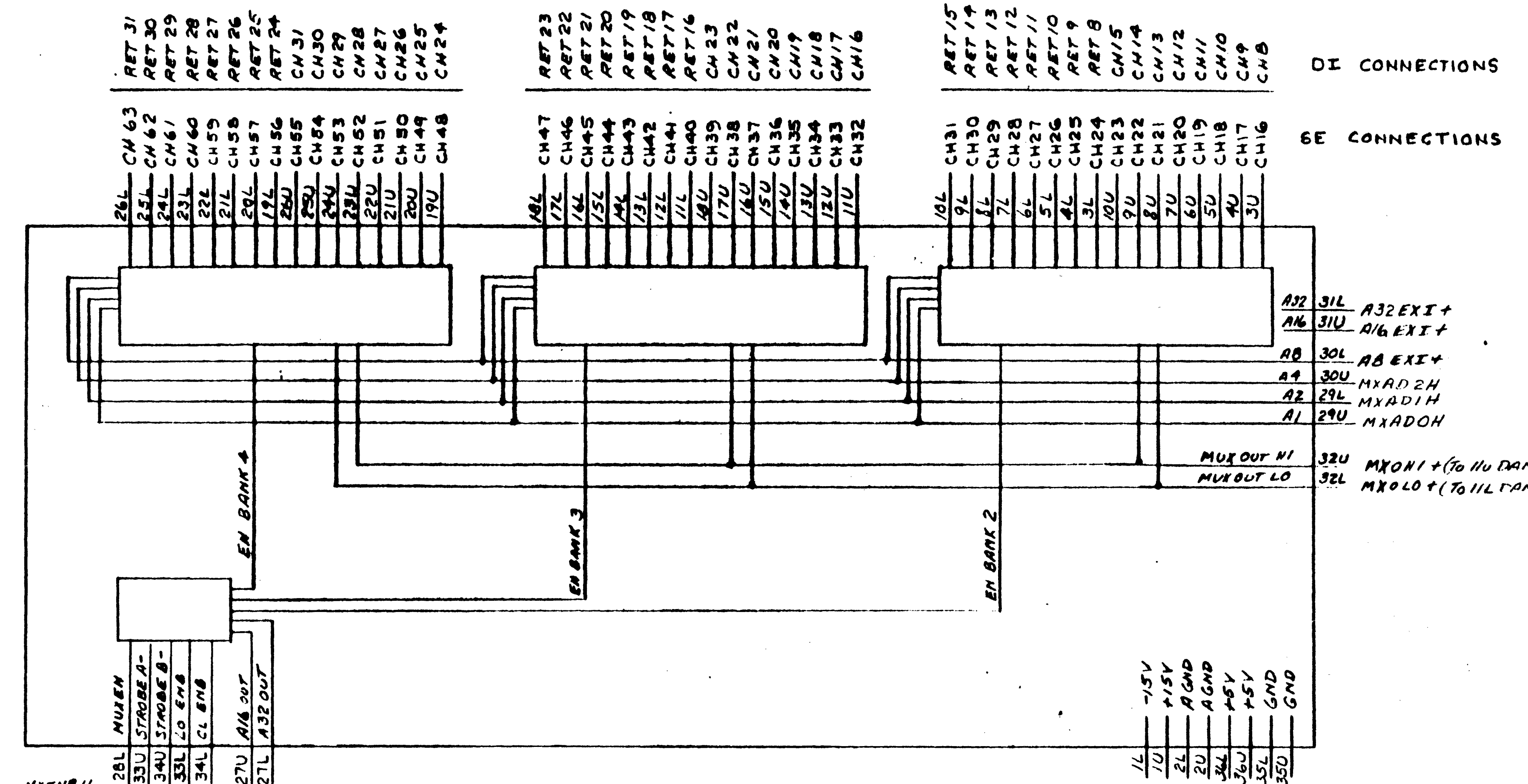
SIZE	CODE IDENT NO.	DRAWING NO.	REV
B	EP039	21020039	D
SCALE		SHEET 8	

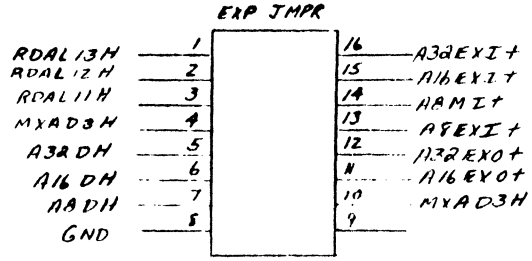
* Rg USED ONLY ON DT5702 MODULE IN DT1765 MODEL



DATA ACQUISITION MODULE

SIZE	CODE IDENT NO	DRAWING NO	RE
B	LP039	21020039	D
		SHEET 9 OF	

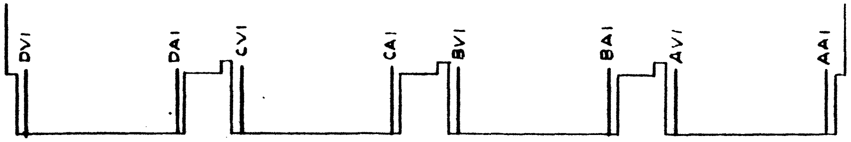




JUMPER	CONFIGURATIONS
64SE	JUMPER ACCROSS ALL PINS
32SE	2-15, 3-14, 4-13, 6-11, 7-10, 5-8
16 SE	3-14, 4-13, 7-10, 5-6-8
32 DI	2-16, 3-15, 6-12, 7-11, 8-5
16 DI	3-15, 7-11, 5-8
8 DI	5-6-7-8

MULTIPLEXER EXPANDER

SIZE	CODE IDENT NO.	DRAWING NO.	R
B	EP039	21020039	
SCALE		SHEET 10 OF	



COMPONENT SIDE

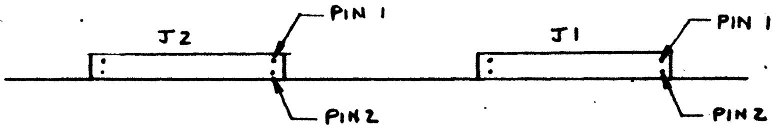
COMPONENT SIDE		SOLDER SIDE		COMPONENT SIDE		SOLDER SIDE	
PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME
AA1		AA2	+5V	BA1	BDCOK H	BA2	+5V
AB1		AB2	-12V	BB1	BPOK H	BB2	-12V
AC1	BAD16	AC2	D GND	BC1		BC2	D GND
AD1	BAD17	AD2	+12V	BD1		BD2	+12V
AE1		AE2	BD OUTL	BE1		BE2	BDAL 2L
AF1		AF2	BRPLY L	BF1		BF2	BDAL 3L
AH1		AH2	BDIN L	BH1		BH2	BDAL 4L
AJ1	D GND	AJ2	BSYNC L	BJ1	D GND	BJ2	BDAL 5L
AK1		AK2	BWTBT L	BK1		BK2	BDAL 6L
AL1		AL2	BIRQL	BL1		BL2	BDAL 7L
AM1	D GND	AM2	BIAKI L	BM1	D GND	BM2	BDAL 8L
AN1	BDMRL	AN2	BIAKOL	BN1	BSACK L	BN2	BDAL 9L
AP1	BHALT L	AP2	BBS7 L	BP1		BP2	BDAL 10L
AR1	BREF L	AR2	BDMGI L	BR1	BEVNT L	BR2	BDAL 11L
AS1		AS2	BDMGOL	BS1		BS2	BDAL 12L
AT1	D GND	AT2	BINIT L	BT1	D GND	BT2	BDAL 13L
AU1		AU2	BDALOL	BU1		BU2	BDAL 14L
AV1	+5B	AV2	BDAL1 L	BV1	+5V	BV2	BDAL 15L

COMPONENT SIDE		SOLDER SIDE		COMPONENT SIDE		SOLDER SIDE	
PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME	PIN	SIGNAL NAME
CA1		CA2	+5V	DA1		DA2	+5V
CB1		CB2	-12V	DB1		DB2	-12V
CC1		CC2	D GND	DC1		DC2	D GND
CD1		CD2	+12V	DD1		DD2	+12V
CE1		CE2		DE1		DE2	
CF1		CF2		DF1		DF2	
CH1		CH2		DH1		DH2	
CJ1	D GND	CJ2		DJ1	D GND	DJ2	
CK1		CK2		DK1		DK2	
CL1		CL2		DL1		DL2	
CM1	D GND	CM2		DM1	D GND	DM2	
CN1		CN2		DN1		DN2	
CP1		CP2		DP1		DP2	
CR1		CR2		DR1		DR2	
CS1		CS2		DS1		DS2	
CT1	D GND	CT2		DT1	D GND	DT2	
CU1		CU2		DU1		DU2	
CV1	+5B	CV2		DV1		DV2	

COMPONENT SIDE

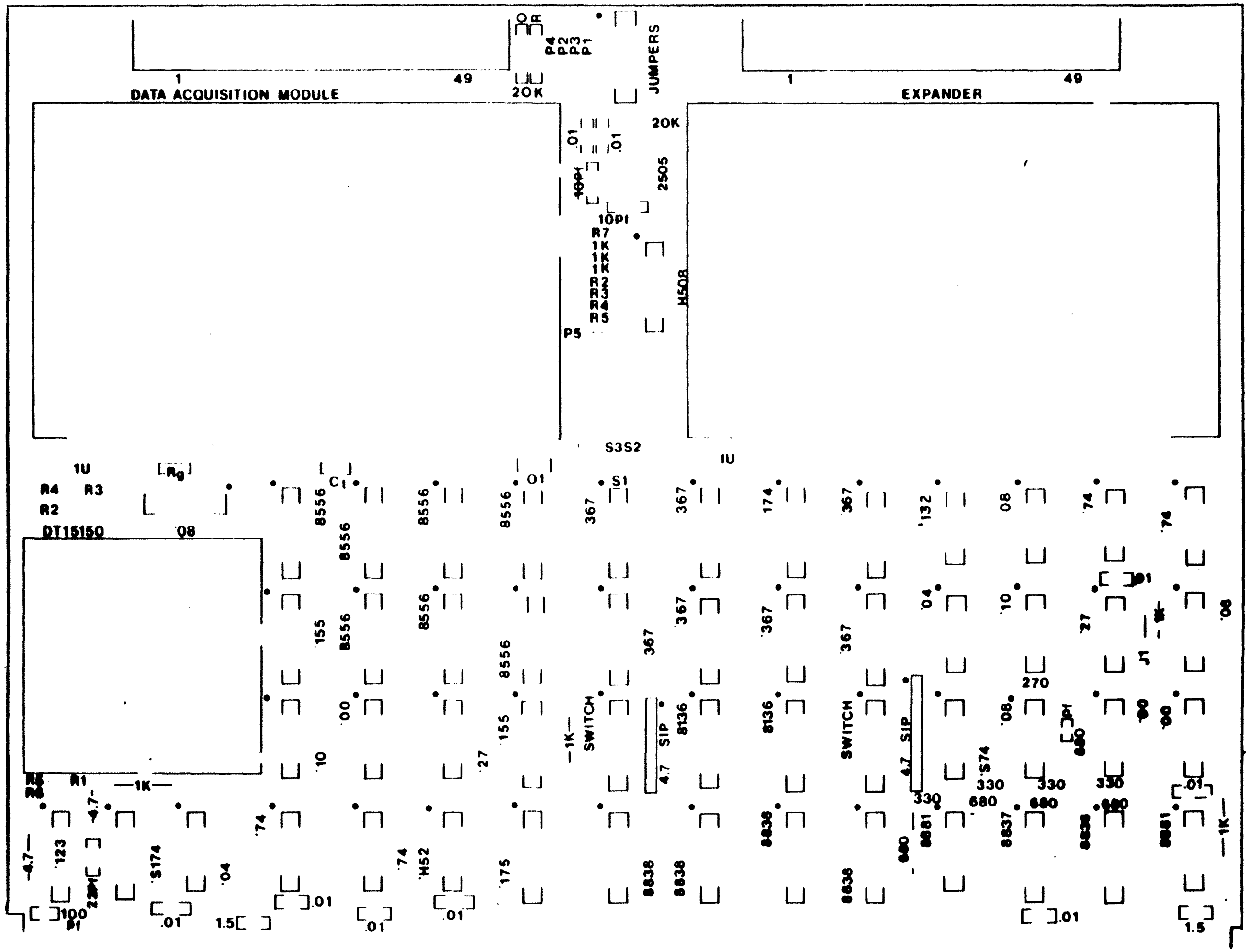
J2			
PIN	SIG. NAME	PIN	SIG. NAME
1	CH24/ RET8	2	CH16/ CH8
3	CH25/ RET9	4	CH17/ CH9
5	CH26/ RET10	6	CH18/ CH10
7	CH27/ RET11	8	CH19/ CH11
9	CH28/ RET12	10	CH20/ CH12
11	CH29/ RET13	12	CH21/ CH13
13	CH30/ RET14	14	CH22/ CH14
15	CH31/ RET15	16	CH23/ CH15
17	CH40/ RET16	18	CH32/ CH16
19	CH41/ RET17	20	CH33/ CH17
21	CH42/ RET18	22	CH34/ CH18
23	CH43/ RET19	24	CH35/ CH19
25	CH44/ RET20	26	CH36/ CH20
27	CH45/ RET21	28	CH37/ CH21
29	CH46/ RET22	30	CH38/ CH22
31	CH47/ RET23	32	CH39/ CH23
33	CH56/ RET24	34	CH48/ CH24
35	CH57/ RET25	36	CH49/ CH25
37	CH58/ RET26	38	CH50/ CH26
39	CH59/ RET27	40	CH51/ CH27
41	CH60/ RET28	42	CH52/ CH28
43	CH61/ RET29	44	CH53/ CH29
45	CH62/ RET30	46	CH54/ CH30
47	CH63/ RET31	48	CH55/ CH31
49	A GND	50	A GND

J1			
PIN	SIG. NAME	PIN	SIG NAME
1	CHØ	2	A GND
3	CH8/RET Ø	4	A GND
5	CH1	6	A GND
7	CH9/RET1	8	A GND
9	CH2	10	A GND
11	CH10/RET2	12	A GND
13	CH3	14	A GND
15	CH11/RET3	16	A GND
17	CH4	18	A GND
19	CH12/RET4	20	A GND
21	CH5	22	A GND
23	CH13/RET5	24	A GND
25	CH6	26	A GND
27	CH14/RET6	28	A GND
29	CH7	30	A GND
31	CH15/RET7	32	A GND
33		34	A GND
35		36	A GND
37	RTC IN	38	D GND
39	EXT TRIG IN	40	D GND
41		42	D GND
43		44	D GND
45		46	D GND
47		48	D GND
49		50	D GND



THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC. AND SHALL NOT BE REPRODUCED, COPIED, OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC.

SIZE B	CODE IDENT NO. EP039	DRAWING NO. 21020039	REV D
SCALE		SHEET 11 OF 11	



COMPONENT DESIGNATIONS

THESE DRAWINGS AND SPECIFICATIONS ARE THE PROPERTY OF DATA TRANSLATION INC AND SHALL NOT BE REPRODUCED COPIED OR USED AS THE BASIS OF MANUFACTURE OR SALE OF APPARATUS WITHOUT THE EXPRESS WRITTEN APPROVAL OF DATA TRANSLATION INC

SIZE	CODE IDENT NO.	DRAWING NO.	REV
A	EP039	11070039	D-2
SCALE		SHEET 1 OF 1	

DATA TRANSLATION INC
4 Strathmore Rd., Natick MA 01760
(617) 655-5300 Telex 948474