DAS-40

User Guide

for the

DAS-40G1 & DAS-40G2

A/D & D/A Data Acquisition Boards

Revision C • December 1992

Part Number: 24404

KEITHLEY DATA ACQUISITION - Keithley MetraByte/Asyst

440 MYLES STANDISH BLVD., Taunton, MA 02780
TEL. 508/880-3000, FAX 508/880-0179

Warranty Information

All products manufactured by Keithley Data Acquisition are warranted against defective materials and workmanship for a period of one year from the date of delivery to the original purchaser. Any product that is found to be defective within the warranty period will, at the option of Keithley Data Acquisition, be repaired or replaced. This warranty does not apply to products damaged by improper use.

Warning

Keithley Data Acquisition **assumes no liability for damages consequent to the use of this product. This product is not designed with components of a level of reliability suitable for use in life support or critical applications.**

Disclaimer

Information furnished by Keithley Data Acquisition is believed to be accurate and reliable. However, the Keithley Data Acquisition assumes no responsibility for the use of such information nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of Keithley Data Acquisition

Note:

MetraByte™ is a trademark of Keithley MetraByte Corporation.

Basic™ is a trademark of Dartmouth College.

IBM® is a registered trademark of International Business Machines Corporation.

PC, XT, AT, PS/2, and **Micro Channel Architecture®** are trademarks of International Business Machines Corporation.

Microsoft® is a registered trademark of Microsoft Corporation.

Turbo C® is a registered trademark of Borland International.

Contents

■ ■ ■

CHAPTER 1

INTRODUCTION

1.1 GENERAL

The DAS-40G1 and DAS-40G2 are high-speed, 12-bit, A/D and D/A data-acquisition boards that plug into an accessory slot of an IBM PC-AT or compatibles. The two boards differ only in A/D gains and throughputs. The G1 version offers gains of 1, 10, 100, and 500 at up to 100,000 samples per second; the G2 version offers gains of 1, 2, 4, and 8 with a throughput of 250,000 samples per second, see specifications on Page 7-2. Analog input configuration is switch-selectable, and the board is configurable for either eight differential input channels or 16 single-ended channels.

D/A subsections include two 12-bit, deglitched D/A converters able to provide either single outputs (the analog outputs change at different times) or simultaneous outputs (the D/A converter outputs change at the same time). In a DMA (Direct Memory Access) transfer, digital data converts to analog at 130KHz. The boards also feature two 8-line digital I/O ports, which can be set for input or output.

DAS-40 models contain a programmable Pacer Clock for controlling A/D and D/A conversion rates. Operating under program control, the Pacer Clock provides a usable range of 4.0us (250KHz) to 2s (0.5Hz). External clocking may also be selected for starting conversions, while an external trigger may be selected for gating the conversions (A/D and D/A). The A/D and D/A converters can operate simultaneously at the same clock speed, or the D/A subsystem can operate in single-conversion mode while the A/D subsystem operates from the pacer clock or from an external clock.

The DMA interface accommodates 16-bit data transfers and is switch-selectable for DMA Channel 5, 6, or 7. DMA buffers can reside anywhere in the DOS 640KB memory space of an PC/AT and may be up to 65,536 words (128KB) each.

The boards are also configurable for using two DMA channels to support Continuous Performance DMA. Continuous Performance DMA is a sampling method for providing gap-free transfers of large volumes of data from memory or disk (D/A conversions) or to memory or disk (A/D conversions) with no sample losses.

Interrupt Level support is jumper-selectable for Level 3, 5, 7, 10, or 15. Analog-input voltage protection extends to ±25V with power On or Off.

Board calibration is performed at the factory. While calibration of the A/D subsystem changes very little with a switch to a new range, the D/A subsystem may need recalibration for any range switching.

1.2 DISTRIBUTION SOFTWARE

The software package furnished with your board includes an installable software driver, BASIC and QuickBASIC Language Interface modules, example programs, and various utilities. Refer to the text file *FILES.DOC* on your Distribution Software diskette for a complete listing of the files contained thereon.

1.3 THE STA-40 ACCESSORY

The STA-40 is a Screw Terminal Accessory Box useful for connecting to the DAS-40. All I/O connections use screw terminals that are accessible through the side of the STA-40. Analog inputs connect to Terminals TB1 and TB2. Analog outputs, along with the external trigger and external clock, connect to TB5. Digital port lines connect to TB3 and TB4. Five "user" lines and a +5V input line are also available.

Two 50-pin connectors and a 26-pin connector provide additional I/O interface. Either 50-pin connector accepts a cable from the DAS-40 while the other accepts one or several daisy-chained STA-40s. The 26-pin connector is available for use with the "5B02 Rack" of signal-conditioning modules.

When connecting an analog input to the STA-40 for a differential measurement, connect the input signal line to a *HI* (High) terminal of Connector TB1 or TB2 (*HI0* through *HI7*) and the input ground to corresponding *L* (Low) terminal (*L08* through *L15*). For example, a differential connection for Channel 0 would use *HI0* and *L08* , while a differential connection for Channel 1 would use *HI1* and *L09* . Reference the ground of the DAS-40 to the system ground of the measurement source by connecting the *AGND* terminal either directly or via resistor to the system ground.

When connecting an analog input to the STA-40 for a single-ended measurement, connect the input signal line to a *HI* (High) or *L* (Low) terminal (*HI0* through *HI7* or *L08* through *L15*). For single-ended measurements, the *HI* and *L* prefixes on TB1 and TB2 have no significance. Connect the analog input return to the terminal labelled *ALO* , which should be tied to the ground reference of your system. If you want the ground reference to be at the STA-40, connect the 2-way jumper of the STA-40 (*W1*) to the *LOCAL* pin. If you want to reference the analog inputs to a point in your system, connect the *W1* jumper to the *REMOTE* pin. For single-ended measurements, connect the *AGND* terminal to your system ground, either directly or through a resistor to limit any common-mode voltage between *ALO* and *AGND* . As a guard against a situation where one might forget to reference the system ground to DAS-40 ground, the DAS-40 10K-Ohm Resistor R30 (Figure 2-1) connects *ALO* to *AGND* . You may remove this resistor if you use the REMOTE measurement scheme.

Note that when you selecting Single-Ended or Differential measurements, remember to set the DAS-40 selection switch.

The STA-40 also contains an area for breadboarding an application circuit. The following patches are available in that area:

PATCH	DESCRIPTION
AGND	Analog input ground return
+15V	+15V from the DAS-40 DC/DC Converter
-15V	-15V from the DAS-40 DC/DC Converter
PGND	Ground return for Â15V lines
DGND	Digital ground return
+5V	User-supplied input
USERA	User-supplied input
USERB	User-supplied input
USERC	User-supplied input
USERD	User-supplied input
USERE	User-supplied input

NOTE: Limit the ±15V output current to under 20mA.

CHAPTER 2

SETUP & INSTALLATION

2.1 GENERAL

This chapter provides instructions for the installing the DAS-40 in an IBM PC-AT or compatible. The chapter begins with procedures for making working copies of your DAS-40 Distribution Software. Next are instructions for unpacking and inspection, followed by descriptions of the options and methods for setting all configurable parameters.

2.2 BACKING UP DISTRIBUTION SOFTWARE

Distribution software is furnished on 5.25", 360K floppy diskette(s). To accommodate users with 3.5" floppy drives, the Software is also available on 720K diskette(s).

As soon as possible, make a working copy of your DAS-40 software using the procedures that follow. Store your original software copy in a safe place as a backup.

The following back-up procedures cover the more common computer configurations: a single-floppy drive (with hard disk), dual-floppy drives, and a hard drive.

Single-Floppy-Drive Machines

To copy your Distribution Software to your computer's hard disk, refer to the subsection below entitled *Hard-Drive Machines* . To copy to another diskette in a single-floppy-drive machine (with hard disk),

1. Turn on power to your computer and display.
2. After system boot-up, the DOS prompt should be C >
3. Be sure the DOS file *DISKCOPY.EXE* is in the C:\ directory. Then, type **DISKCOPY A: A:**
4. Insert the *source* diskette (your DAS-40 Distribution Software diskette) into Drive A. The system will prompt you through the disk copying process. When the source diskette has been copied into memory, the System will ask you to insert the *target* diskette into Drive A. The *target* diskette is a formatted, blank disk that is to be your back-up disk.
5. When a copy is complete, the computer will ask COPY ANOTHER (Y/N)?. Respond by typing **Y** for another diskette or **N** if you are finished copying. If you typed **Y** , repeat Steps 3 and 4.
6. Put the original DAS-40 diskette(s) in a safe place. Label the back-up diskette(s) *DAS-40 Working Disk* and use for running your DAS-40 programs.

Dual-Floppy-Drive Machines

To copy your Distribution Software to the computer's hard disk, refer to the next subsection, *Hard-Drive Machines* . To copy to another diskette(s) in a dual-floppy-drive machine,

1. Turn on power to your computer and display, and place your DOS diskette in Drive A.
2. The DOS prompt should be **A >** If not, type **A:** followed by < **Enter** > . Be sure the diskette in Drive A contains the DISKCOPY.EXE file.
3. Then, type **DISKCOPY A: B:**
4. Insert the *source* diskette (your DAS-40 diskette) into Drive A. The system will prompt you through the disk copying process. It will ask you to insert the *target* diskette into Drive B. The *target* diskette is a blank disk that is to be your back-up disk.
5. When a copy is complete, the computer will ask COPY ANOTHER (Y/N)?. Respond by typing **Y** for another diskette or **N** if you are finished copying. If you typed **Y** , repeat Steps 3 and 4.
6. When copying is complete, put the original DAS-40 diskette(s) in a safe place. Label the back-up diskette(s) *DAS-40 Working Disk* . Use this disk to run the software.

Hard-Drive Machines

To copy your DAS-40 files to a hard disk:

1. Start your computer. You should see a prompt, which indicates you are at the DOS level (for example, if your hard drive is designated as C , you should see the prompt C >).
2. The following instructions create a special directory for the DAS-40 Distribution Software files. At the DOS prompt, type: **mkdir D40** followed by < **Enter** > . Change to the DAS-40 directory by typing: **CD D40** followed by < **Enter** > .
3. Place the Distribution diskette into Floppy Drive A and type **A:** . When the prompt changes from C > to **A >** , type **copy *.* C:** followed by < **Enter** >.
4. When a copy is complete, the computer will ask COPY ANOTHER (Y/N)?. Respond by typing **Y** for another diskette or **N** if you are finished copying. If you typed **Y** , repeat Steps 3 and 4.
5. You have now copied the contents of the Distribution Software diskette to your hard disk. Store the original diskette in a safe place.

2.3 UNPACKING AND INSPECTING

After you remove the wrapped board from its outer shipping carton, proceed as follows:

1. Place one hand firmly on a metal portion of the computer chassis (the computer must be turned Off and grounded). You place your hand on the chassis to drain off static electricity from the package and your body, thereby preventing damage to board components.
2. Allow a moment for static electricity discharge; carefully unwrap the board from its anti-static wrapping material.
3. Inspect the board for signs of damage. If any damage is apparent, return the board to the factory.
4. Check the contents of your package against its packing list to be sure the order is complete. Report any missing items to the manufacturer immediately.

You may find it advisable to retain the packing material in case the board must be returned to the factory.

2.4 CONFIGURATION OPTIONS

The subsections that follow describe DAS-40 configuration options you may set prior to board installation. These options are either switch- or jumper-selectable and include the following:

- Setting the Base Address.
- Setting the DMA Channel(s).
- Setting the Interrupt Level.
- Setting A/D functions:

- Input Range
- Coding
- Input Configuration

- Setting D/A functions:

- DAC 0 Output Range
- DAC 1 Output Range

Figure 2-1 shows switch and jumper locations for these settings.

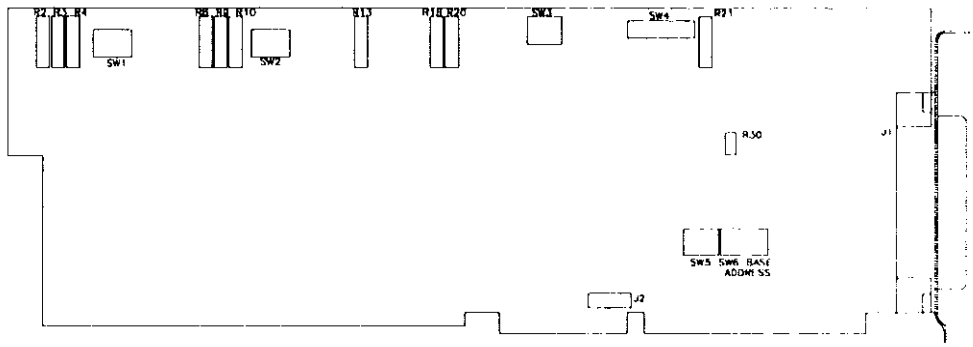


Figure 2-1. Switch and jumper locations. Note that R21 is not present on the Board's G2 version.

Included in the Distribution Software is the utility CONFIG40.EXE. This utility is provided as an aid to changing the jumpers and Dip Switches on the DAS-40 board. Refer to the section in Chapter 3 called *CHANGING THE FACTORY CONFIGURATION* for more detail.

Base Address

Check Base Address setting on the board's Base Address switch, which is a 6-position DIP switch labelled *SW6 BASE ADDRESS* . This switch is preset at the factory for an address of 240 Hex.

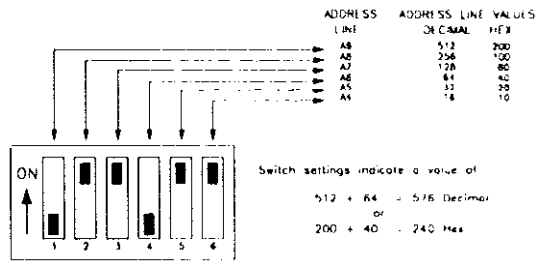


Figure 2-2. Diagram of the Base Address switch (SW6).

The factory-preset Base Address of 240 Hex is within the address range shown in the following table as *Reserved*. This default value will function in most computers without conflict, thereby eliminating any need for address selection and configuration. However, if you have a need to change the Base Address from its preset value, you must select an address within a range of 200 to 3E0 Hex (512 to 992 Decimal). In addition, the address must be on a 16-byte boundary and must not conflict with addresses already in use for other devices. As an aid to selecting a usable 3-digit Hex number, the following table is an industry-standard I/O address map for the full 000 to 3FF range.

Table of Industry-standard I/O addresses for peripheral devices.

HEX RANGE	USAGE	HEX RANGE	USAGE
000 to 1FF	Internal System	387 to 37F	LPT1:
200 to 20F	Game	380 to 38C	SDLC comm.
210 to 217	Expansion unit	380 to 389	Binary comm. 2
220 to 24F	Reserved	3A0 to 3A9	Binary comm. 1
278 to 27F	Reserved	3B0 to 3BF	Mono dsp/LPT1:
2F0 to 2F7	LPT2:	3C0 to 3CF	Reserved
2F8 to 2FF	COM2:	3D0 to 3DF	Color graphics
300 to 31F	Prototype card	3E0 to 3E7	Reserved
320 to 32F	Hard disk	3F0 to 3F7	Floppy disk
		3F8 to 3FF	COM1:

DMA Channel

The DAS-40 allows you to select DMA transfer channels. While you may select two channels for DMA (if two are available), the second channel is used only when Dual-DMA mode or continuous cycle is selected. Using SW5 (refer to Figure 2-1), you may select DMA Channel 5 (highest priority), Channel 6, or Channel 7 (lowest priority). The Board is factory-preset to select Channel 5 as the first channel and Channel 6 as the second. SW5 is a 4-position DIP switch whose setting options are as follows:

	FIRST DMA CHANNEL		SECOND DMA CHANNEL	
CHANNEL #	S1	S2	S3	S4
5	Off	On*	Off	On
6	On	Off	On	Off*
7	On	On	On	On
None	Off	Off	Off	Off

* Factory Configuration.

Note that a DMA channel used by one board may not be used by any other board (including another DAS-40) in the same PC system. In a multiple DAS-40 system, you must set each board to a different channel, limiting Dual-DMA and/or continuous cycle usage to a single board and setting remaining boards to use single-channel DMA.

Interrupt Level

The DAS-40 can interrupt the processor on any one of five different levels. Priority for these five

Interrupt Levels is Level 10 (highest), Level 15, Level 3, Level 5, and Level 7 (lowest).

The board is factory preset for Interrupt Level 15. You may change the Interrupt Level setting by repositioning the J2 jumper (refer to Figure 2-1) according to the following table.

INTERRUPT LEVEL	JUMPER POSITION
10	1
*15	2
3	3
5	4
7	5
None	6

* Factory Configuration.

A DAS-40 generates only one interrupt, regardless of cause. The Interrupt Service Routine must determine the cause of the interrupt by polling the ADCSR, DACSR, and SUPCSR (see Section 5.1). Upon finding the cause of the interrupt, the Interrupt Service Routine can act accordingly.

Note that an Interrupt used by one board in a PC must not be used by any other board in the PC. In a PC using Multiple DAS-40 boards, each board must be set to a different Interrupt Level. A board that needs no Interrupt should be set to disable its Interrupts.

A/D Functions

The Analog-to-Digital section of the DAS-40 is switch-selectable for the following:

- Input Range.
- Binary, Offset Binary, or 2's Complement coding.
- Single-ended or Differential input.

The following subsections describe settings for these parameters.

Input Range

If a board is configured for Unipolar inputs, it should receive only positive voltages. If configured for Bipolar inputs, it can accept both positive and negative voltages. The DAS-40 is factory preset for Bipolar inputs of ±10V. To change this range, reset positions S1 and S2 of DIP Switch SW3 according to the following table.

	SW3 POSITIONS	
RANGE	S1	S2
Bipolar ±10V *	Off	Off
Bipolar ±5V	On	Off
Unipolar +10V	On	On

* Factory Configuration.

NOTE: A change of input range may require a recalibration of the A/D converter to attain full-rated accuracy. See Chapter 6 for the calibration information.

Bipolar inputs may be either Offset Binary or 2's Complement Output Coding. Unipolar inputs should be straight Binary Output Coding.

Output Coding

A/D conversion changes an analog input signal to a corresponding digital format so that it can be processed by the PC. The output from a DAS-40 is a binary data word whose coding is selected with Switch SW3-3.

As mentioned in the preceding subsection, Bipolar inputs use either Offset Binary or 2's Complement Output Coding, while Unipolar inputs use straight Binary Output Coding. The DAS-40 is preset for Offset Binary coding but offers switch selectable Straight Binary and 2's Complement codings, as well. You may select output coding by setting Position S3 of DIP Switch SW3 according to the following table.

	SW3 POSITION
CODING	S3
Straight Binary (Unipolar)	On
Offset Binary (Bipolar)*	On
2's Complement (Bipolar)	Off

* Factory Configuration.

Single-Ended/Differential Inputs

The DAS-40 is switch-selectable for either Single-Ended or Differential Input modes. In Single-Ended mode, the board offers 16 channels for 16 different signals, each with a common return path. In Differential mode, the board uses a separate return path for each input signal and is thus limited to eight different signals. To configure the board for either input, set Switch SW4 according to the following table.

CONFIGURATION	POSITION
16 Single-Ended*	Left
8 Differential	Right

* Factory Configuration.

D/A Functions

A DAS-40 board contains two Digital-to-Analog Converters (DACs) which are individually selectable for Unipolar or Bipolar output ranges. In a Unipolar configuration, a DAC carries positive-voltage outputs only. In a Bipolar configuration, a DAC carries either positive or negative voltage, so long as the level is within the selected range (±10V, ±5V, ±2.5V, 0 to 10V, or 0 to 5V).

Set the DAC 0 output range selection using positions S1 through S5 of DIP Switch SW2, as follows:

	SW2 POSITIONS				
DAC 0 OUTPUT RANGE	S1	S2	S3	S4	S5
±10V*	Off	On	Off	Off	On
±5V	Off	On	On	Off	On
±2.5V	On	Off	On	Off	On
0 to 10V	Off	On	On	On	Off
0 to 5V	On	Off	On	On	Off

* Factory Configuration.

Set the DAC 1 output range selection using positions S1 through S5 of DIP Switch SW1, as follows:

	SW1 POSITIONS				
DAC 1 OUTPUT RANGE	S1	S2	S3	S4	S5
±10V*	Off	On	Off	Off	On
±5V	Off	On	On	Off	On
±2.5V	On	Off	On	Off	On
0 to 10V	Off	On	On	On	Off
0 to 5V	On	Off	On	On	Off

* Factory Configuration.

In Unipolar operation, the DAS-40 uses digital data in Straight Binary coding. In Bipolar mode, the board uses data in Binary Offset coding.

NOTE: After selecting a new output range, you must recalibrate the DAC (see calibration procedures).

2.5 HARDWARE INSTALLATION

Single-Ended vs. Differential Connections

Single-Ended configuration of the DAS-40 allows 16 channels to be made available for analog signals, while Differential configurations allows eight channels. Thus, Single-Ended configuration offers maximum channel density. However, Single-Ended configuration is more sensitive to noise from the input cables and is therefore better suited to applications using higher-level input voltages (over 1V Full-Scale) and shorter cable lengths (under 15').

When configuring for Single-Ended operation, connect the return sides of all analog channels to Amp Low, and connect the high side of each analog input to the corresponding input of the multiplexer in the A/D Converter.

When configuring for Differential operation, connect the high and low sides of an analog input signal to the corresponding inputs of the multiplexer stage. See Section 1.4 for more detail.

Installing the DAS-40 in a PC

WARNING: ANY ATTEMPT TO INSERT OR REMOVE A BOARD WITH THE COMPUTER POWER ON COULD DAMAGE YOUR COMPUTER!

1. Turn Off power to the PC and all attached equipment.
2. Remove the cover of the PC as follows: First remove the cover-mounting screws from the rear panel of the computer. Then, slide the cover of the computer about 3/4 of the way forward. Finally, tilt the cover upwards and remove.
3. Choose an available option slot. Loosen and remove the screw at the top of the blank adapter plate. Then slide the plate up and out to remove.
4. Hold the DAS-40 board in one hand placing your other hand on any metallic part of the PC/AT chassis (but not on any components). This will safely discharge any static electricity from your body.
5. Make sure the board switches have been properly set (refer to the preceding section).
6. Align the board connector with the desired accessory slot and with the corresponding rear-panel slot. Gently press the board downward into the socket. Secure the board in place by inserting the rear-panel adapter-plate screw.
7. Replace the computer's cover. Tilt the cover up and slide it onto the system's base, making sure the front of the cover is under the rail along the front of the frame. Replace the mounting screws.
8. Plug in all cords and cables. Turn the power to the computer back on.

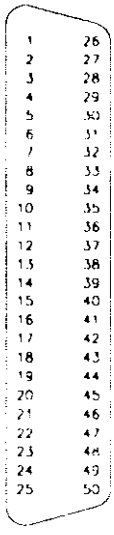
You are now ready to make any necessary system connections, install the DAS-40 software, and perform calibration and perform checks on calibration and adjustment, as described in the chapter on calibration.

The manufacturer recommends that you retain the static-shield packaging for possible future removal and handling of the DAS-40 board.

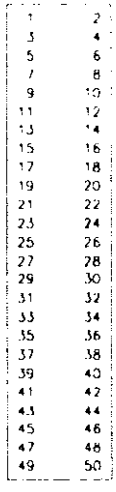
2.6 DAS-40 & STA-40 I/O CONNECTIONS

SIGNAL NAME	DAS-40 PIN NO.	STA-40 PIN NO.
Channel 0	1	1
Channel 8 (0 Ret)	26	2
Channel 1	2	3
Channel 9 (1 Ret)	27	4
Channel 2	3	5
Channel 10 (2 Ret)	28	6
Channel 3	4	7
Channel 11 (3 Ret)	29	8
Channel 4	5	9
Channel 12 (4 Ret)	30	10
Channel 5	6	11
Channel 13 (5 Ret)	31	12
Channel 6	7	13
Channel 14 (6 Ret)	32	14
Channel 7	8	15
Channel 15 (7 Ret)	33	16
AGND	9	17
Amp Low	34	18
+15V Out	10	19
-15V Out	35	20
PGND	11	21
DAC 0 Out	36	22
DAC 0 GND	12	23
DAC 1 Out	37	24
DAC 1 GND	13	25
DGND	38	26
DGND	14	27
DIO Port 0, Bit 0	39	28
DIO Port 0, Bit 1	15	29
DIO Port 0, Bit 2	40	30
DIO Port 0, Bit 3	16	31
DGND	41	32
DIO Port 0, Bit 4	17	33
DIO Port 0, Bit 6	42	35
DIO Port 0, Bit 5	18	34
DIO Port 0, Bit 7	43	36
DGND	19	37
DIO Port 1, Bit 0	44	38
DIO Port 1, Bit 1	20	39
DIO Port 1, Bit 2	45	40
DIO Port 1, Bit 3	21	41
DGND	46	42
DIO Port 1, Bit 4	22	43
DIO Port 1, Bit 5	47	44
DIO Port 1, Bit 6	23	45
DIO Port 1, Bit 7	48	46
DGND	24	47
DGND	49	48
Ext. Trigger IN	25	49
Ext. Clock IN	50	50

DAS-40 Board Connector



STA-40 Board Connector



❑

❑

❑

CHAPTER 3

PROGRAMMING NOTES

3.1 GENERAL

At the lowest level, DAS-40 is programmable with I/O (Input/Output) instructions. In BASIC, these are the IMP (X) and OUT X,Y functions. Assembly Language and most other high- level languages have equivalent instructions (IN AL,DX and OUT DX,AL). Use of these functions usually involves formatting data and dealing with absolute I/O addresses. Although not demanding, this can require many lines of code and necessitates an understanding of the devices, data format, and architecture of the DAS-40. To simplify DAS-40 programming, the Distribution Software contains a software driver (MDAS40.EXE) and Language Interface Modules for BASIC and QuickBasic (DAS40.BIN, DAS40,LIB and DAS40.QLB).

The MDAS40.EXE software driver installs readily from the DOS command line and is accessible from BASIC and QuickBASIC with a single-line CALL statement. The driver supports the majority of common operating MODEs. Using the CALL routine for DAS-40 MODEs, you may program your applications to select and perform any DAS-40 function, format and error-check data, and perform frequently used sequences of instructions. An example is MODE 3 which sets up the A/D Start/Stop channels and the Global Gain value.

The DAS-40 software driver saves programming time and supports data collection using DMA from an external clock source or the DAS-40 internal timer. Note that BASIC has no DMA processing functions; "background" data collection is available only by using the CALL routine.

Both methods of programming using INP and OUT functions and the CALL routine achieve the same result; you are free to choose either, although usually the BASIC programmer will find the CALL routine simpler to implement. If the DAS-40 MODEs described in this manual do not support your requirements, you may modify the DAS-40 driver as necessary. The fully commented assembly source is available from the manufacturer on a floppy disk (Part # PCF-40) and is a good starting point for Assembly Language programmers wishing to modify the standard driver routines.

3.2 THE LANGUAGE INTERFACE MODULES

DAS-40 Distribution Software includes the Language Interface Modules for BASIC and QuickBasic. These are:

DAS40.BIN	Interface Module for BASIC(A) that is loaded via the BLOAD Command.
DAS40.LIB	Interface Module for QuickBASIC (Ver 4.0 and higher) and Professional Basic (Ver 7.0 and higher) Stand-alone EXE programs.
DAS40.QLB	Interface Module for the QuickBasic (Ver 4.0 and higher) Programming Environment. It is specified at the QB invocation time using the /L switch as follows: **QB /L DAS40.QLB**
DAS40x.QLB	Interface Module for the QuickBasic Extended (Ver 7.0 and higher) Environment. It is specified at the QBX invocation time using the /L switch as follows: **QBX /L DAS40x.QLB**

All CALL mode communications with the DAS-40 driver is performed through these Interface Modules. Other interface modules are available from the manufacturer for Pascal, C and FORTRAN by ordering the PCF-40 option.

A typical CALL statement from your BASIC program to the driver is as follows:

```
380 MD% = 0                          'Initialize mode
390 FLAG% = 0                        'Clear error variable
400 D%(0) = 0                        'Specify Card #0
410 CALL DAS40(MD%, D%(0), FLAG%) 'Call the driver
420 IF FLAG% <> 0 THEN PRINT "MODE 0 Error # "; FLAG% AND 255: STOP:
```

The CALL from QuickBASIC differs only in the CALL statement (Line 410), as follows:

```
410 CALL BASDAS40(MD%, VARPTR(D%(0)), FLAG%)
```

The three variables within the parentheses (MD%, D%() and FLAG%) are the CALL parameters. In executing the CALL, the addresses of the variables (pointers) are passed in the sequence written to BASIC's stack. The CALL routine unloads these pointers from the stack and uses them to locate the variables in BASIC's data space so data can be exchanged with them.

Note that, FLAG% is a 16-bit quantity containing the error number in the least-significant eight bits and the mode number where the error occurred in the most-significant eight bits. Therefore to extract the error number, FLAG% must be ANDed with 255 (0FFH).

Also note, that D%() must be previously dimensioned as a ten-item integer array. For example,

```
160 DIM D%(10)
```

The following formatting requirements that must be met:

1. The CALL parameters are position-sensitive. The subroutine (DAS40 or BASDAS40) knows nothing of the names of the variables, just their locations from the order of their pointers on the stack.
2. The CALL routine expects its parameters to be integer-type variables; it writes and reads to the variables on this assumption. If you slip up and use a non-integer (real, single, or double-precision) variable in the CALL parameters, the routine will not function correctly. No error checking is done in the CALL on the variable type; so take care not to crash the computer!
3. You cannot perform any arithmetic functions within the parameter-list parentheses of the CALL statement. For example,

   ```
   410 CALL DAS40 (MD% + 2, D%(0) * 8, FLAG%)       'ILLEGAL!
   ```

   is illegal and will produce a syntax error.
4. You cannot use constants for any of the parameters in the CALL statement. The following is illegal:

   ```
   410 CALL DAS40 (7, 2, FLAG%)          'ILLEGAL!
   ```

   This must be programmed as shown above.

Apart from these restrictions, you can name the integer variables whatever you wish; the names in the examples are just convenient mnemonics. Strictly, you should declare the variables before executing the CALL. If you do not, the simple variables will be declared by default on execution, but array variables cannot be dimensioned by default and must be dimensioned before the CALL to pass data correctly if used as a CALL parameter. Most MODEs of the DAS40.BIN CALL routine require multiple items of data to be passed in an array. For this reason, D%(0) is specified as the data variable so that the CALL routine can locate the whole array from the position of its initial element.

Interface For BASIC(A) - DAS40.BIN

If you are new at using CALL statements in BASIC(A), the following may assist you in understanding how the CALL transfers execution to the Software Driver through the Interface Module (DAS40.BIN). Prior to entering the CALL, use **DEF SEG = SG** statement to set the segment address of the CALL subroutine as follows:

```
100 CLEAR, 49152!                              'reduce workspace to 48K
110 DEF SEG = 0
120 SG = 256 * PEEK(&H511) + PEEK(&H510)       'find BASIC's segment
130 SG = SG + 49152!/16                        'Find segment above BASIC
140 DEF SEG = SG                               'SG = load location
150 BLOAD "DAS40.BIN", 0                       'for the .BIN:
```

Note that the details of the DEF SEG statement are complicated and extend beyond the scope of this manual; therefore, they are not discussed here. For a working example of this routine, refer to the BASIC program BAEXAMPL.BAS in the Distribution Software.

After the successful BLOADing of DAS40.BIN, the typical CALL statement from your BASIC program is

```
380 MD% = 0                          'Initialize mode
390 FLAG% = 0                        'Clear error variable
400 D%(0) = 0                        'Specify Card #0
410 CALL DAS40(MD%, D%(0), FLAG%) 'Call the driver
420 IF FLAG% <> 0 THEN PRINT "MODE 0 Error # "; FLAG% AND 255: STOP:
```

The word **DAS40** in line 410 is the label assigned to be used from BASIC(A) only. Refer to the previous section for discussion on the CALL syntax.

Interface For QuickBASIC - DAS40.LIB, DAS40.QLB

Interfacing your QuickBASIC programs to the DAS-40 is a lot simpler than interfacing from BASIC(A); it is done simply by linking your program's object module(s) to either DAS40.LIB or DAS40.QLB (DAS40x.QLB if using QuickBASIC Extended Ver 7.0).

You use the .QLB module when running your programs from within the QuickBasic Integrated Environment. Specify the module with the **/L** switch when you first invoke QuickBASIC, as follows:

QB /L DAS40.QLB *your-file-name*

Use the .LIB module when you want to create stand-alone .EXE programs. To do this, you must compile and link your program as follows:

BC *example.bas /o* **;**
LINK *example* **,,,DAS40.LIB;**

where *example* is the name of your program, **BC** is the QuickBASIC Compiler and **LINK** is the Microsoft linker.

Regardless of the module you use (.LIB or .QLB), the actual CALL statement is as follows:

```
180 MD% = 0                              'initialize mode
190 FLAG% = 0                            'declare error variable
200 D%(0) = 0                            'Card #0
210 CALL BASDAS40(MD%, VARPTR(D%(0)), FLAG%)
220 IF FLAG% <> 0 THEN PRINT "MODE 0 Error # "; FLAG% AND 255: STOP
```

where BASDAS40 is the label assigned for use from QuickBASIC only. Refer to Section 3.2 for discussion on the CALL syntax and restrictions. For a working example in QuickBASIC, refer to the program QBEXAMPL in the Distribution Software.

3.3 LOADING/UNLOADING THE DAS-40 SOFTWARE DRIVER

The DAS-40 Device Driver consists of the two files MDAS40.EXE, and VIPARSE.EXE. As executable files, both are self-loading. Loading and unloading may be performed from the DOS command line.

At the DOS command line, type the following:

VIPARSE

followed by

MDAS40 */C0= file0 /C1= file1*

Where *file0* and *file1* are optional configuration files for Cards #0 and Card #1, respectively. These files are presumed created via the Distribution Software utility CONFIG40.EXE. Refer to the Section 3.4 (*CHANGING THE FACTORY CONFIGURATION*) for more detail. Specification of switches */C0=* and */C1=* is optional. If you specify no switches, Factory Configuration is assumed for up to two physical boards. If you specify both */C0=* and */C1=* , their order on the command line is irrelevant.

These loading commands may be entered automatically on power-up by adding them to your AUTOEXEC.BAT file. Use any ASCII text editor or a word processor to add the commands in the order shown above.

The DAS40 Device Driver (both files) may also be unloaded from the DOS Command line. Unloading the Device Driver returns the memory it occupies to DOS. The Device Driver files must be unloaded in the following order: first MDAS40.EXE, then VIPARSE.EXE.

To unload MDAS40.EXE, type the following:

MDAS40 */U*

and for VIPRASE.EXE,

VIPARSE */U*

Note that VIPARSE and MDAS40 occupy approximately 7600 bytes and 18000 bytes respectively.

3.4 CHANGING THE FACTORY CONFIGURATION

Provided with the release software is the utility: CONFIG40.EXE. This utility serves as an aid for setting the Jumper and Dip Switch settings and as a means for generating a disk file containing new board configuration information. To execute this utility, proceed as follows:

From the DOS Command Prompt, type the following:

CONFIG40 *file*

Where *file* is a valid DOS file name, which may include any necessary DOS Drive and path specification.

For example,

```
CONFIG40 C:\DAS40\CARD0.CFG
```

If the specified file does not exist, the default Factory Configuration is displayed; otherwise the contents of the file are shown.

If the file name is omitted, the CONFIG40 help screen is displayed. Follow the instructions as they appear on the screen.

The Configuration file created here is used when loading the DAS-40 Driver. Refer to Section 3.3 (*LOADING/UNLOADING THE DAS-40 SOFTWARE DRIVER*) for more detail.

Note that the files created by the CONFIG40 utility are in binary format and therefor not readable or suitable for editing using standard text editors. All changes to these files must occur using this utility.

3.5 THE CALL MODES

The following table identifies DAS-40 MODEs (see Chapter 4 for descriptions). The MODEs are selected by the MD% parameter in the CALL as follows:

MD%	FUNCTION
0	Initialize and Query Configuration of an installed DAS-40.
1	Assign a buffer and sample count to each DMA level for both A/D and D/A. Used before calling MODES 4 or 5.
2	Setup for N A/D or D/A conversions for transfer via DMA. Used before calling MODES 4 or 5.
3	Setup global start and stop channels and overall gain.
4	Start A/D conversions and transfer via DMA as setup in MODEs 1 + 2.
5	Start D/A conversions and transfer via DMA as setup in MODEs 1 + 2.
6	Setup Pacing clock rate.
7	Perform one A/D conversion on specified channel.
8	Perform one D/A conversion on DAC channel (1, 2, or both).
9	Monitor ADC/DAC DMA transfer status.
10	Setup for Digital Input/Output.
11	Input Byte/Word.
12	Set/Query current DAS-40 Card number
13	Terminate Current DMA transfer - A/D In or D/A Out.
14	Output Byte/Word.
15	Tag channel numbers to data.
16	A/D Triggered Block Scan (DMA).
17	Transfer A/D Data to a BASIC Array.
18	Allocate DMA Memory Block
19	Free DMA Memory Block

3.6 DEFAULT CONFIGURATION

The DAS-40 Software Driver assumes the following Factory Default conditions for DAS-40 Card Number 0:

BASE ADDRESS:	240H
DMA CHANNEL A:	5
DMA CHANNEL B:	6
INTERRUPT LEVEL:	15
A/D MODE:	16-CHANNEL SINGLE-ENDED
A/D INPUT RANGE:	±10V
A/D CODING:	OFFSET BINARY
DAC #0 RANGE:	±10V
DAC #1 RANGE:	±10V

Card Number 0 is assumed to be of the G2 variety (1/2/4/8 Gain).
If this configuration is not compatible with your system requirements, refer to Section 3.4 (*CHANGING THE FACTORY CONFIGURATION*) for instructions on making changes.

3.7 ACCOMMODATING MULTIPLE DAS-40s IN A SINGLE SYSTEM

The supplied software driver MDAS40.EXE supports up to two (2) DAS-40 boards simultaneously. The default configuration of a possible second board are identical to the Factory Configuration of the first board except for the following:

BASE ADDRESS:	250H
DMA CHANNEL A:	7
DMA CHANNEL B:	NONE
INTERRUPT LEVEL:	10

To override the factory settings, use the */C1=* switch as described in Section 3.3 (*LOADING/UNLOADING THE DAS-40 SOFTWARE DRIVER*).

Because the DAS-40 is designed to use DMA channels 5, 6, and 7, Dual-DMA modes are not allowed for the DAS-40 board configured with only one channel.

■ ■ ■

CHAPTER 4

CALL MODES

In each of the following CALL MODE discussions, the usage explanations include two subheadings, as follows:

- *Parameters:*
- *Returns:*

In a program that CALLs a particular MODE with the statement **CALL DAS40 (MD%, D%(0), FLAG%)**, the parameters are the acceptable values for the *MD%* array *D%()*. For example, the following Initialization Parameters are for MODE 3.

MD% = 3 (the MODE number).

D%(0) = Start channel # (0 - 15 for S.E.; 0 - 7 for Diff.).

D%(1) = Stop channel # (0 - 15 for S.E.; 0 - 7 for Diff.).

D%(2) = Overall gain code (0 - 3, according to the following table).

CODE	G1	G2
0	1	1
1	10	2
2	100	4
3	500	8

In a typical program (in BASIC) CALLing MODE 3, you might use these arguments to initialize MD% and D%(0) as follows:

```
MD% = 3                              ' Specify the MODE number.
D%(0) = 0                            ' Specify a starting channel number.
D%(1) = 12                           ' Specify an ending channel number.
D%(2) = 2                            ' Specify a Gain Code (from the table).
CALL DAS40 (MD%, D%(0), FLAG%)       ' Execute the CALL.
```

The parameters may be re-initialized for different values and the CALL repeated as many times as desired. Refer to the Distribution Software for fully functional example programs that use these MODEs.

After the CALL, any values or information that might be returned as a result of the CALL are listed under *Returns*.

Note that the *FLAG%* parameter (for Errors) does not require initialization. Error Codes that might be returned for FLAG% are discussed in Appendix C.

4.1 MODE 0 - QUERY CONFIGURATION AND INITIALIZE AN INSTALLED DAS-40

This mode initializes the board's hardware registers and subsystems to a known state. The board is initialized according to the information found in the internal data tables of the DAS-40 Driver. The information may be modified whenever the Driver is loaded from the DOS Command Line. If nothing is specified at load time, the Factory Configuration of the DAS-40 board(s) is assumed; refer to the section on switch and jumper settings in this manual for more detail.

To modify the Factory Configuration of the DAS-40, you must (with the DAS-40 out of the PC) run the utility CONFIG40.EXE (in the Distribution Software). This utility provides a means for creating a Board Configuration File and an aid for setting the on-board jumpers and dip switches. The Board Configuration File is used when loading the DAS-40 Driver to override the Factory Configuration. Refer to the section on changing the DAS-40 configuration for more detail.

The Pacer Clock defaults to 40KHz, and the Digital I/O Ports assume DIO Configuration Code 0 for word input.

MODE 0 Parameters:

D%(0) = Card # to initialize (0 or 1)

MODE 0 Returns:

D%(0) = The Card's Base Address

D%(1) = Interrupt request level

D%(2) = DMA Channel A

D%(3) = DMA Channel B

D%(4) = A/D Data Coding: 0 = Offset Straight Binary *
1 = Straight Binary
2 = 2's Complement

D%(5) = A/D Mode: 16 = Single Ended
8 = Differential

* For information on A/D coding, refer to Appendix A.

4.2 MODE 1 - SPECIFY DMA BUFFERS AND CONVERSION COUNTS

This mode is used to setup the address(es) and conversion count(s) for the next DMA A/D or D/A operation. Through this mode, you can setup addresses for single (Buffer A) or dual (Buffers A and B) DMA operations. The addresses and conversion counts passed to this mode are generally obtained through previous calls to Mode 18. Although the addresses of user arrays may be used as DMA buffers instead, these are not guaranteed to be suitable for DMA; i.e. Mode 1 may return error 13 or 22. For the BASIC languages, specifying a conversion count greater than 32767 into an integer variable causes an Overflow error. The following illustrates how to specify 50,000 samples in D%(2) without causing an Overflow:

D%(2) = 50000 - 65536

Buffer B parameters, D%(3), D%(4) and D%(5), are required only when Dual DMA operation is desired; specify zeroes when not used.

MODE 1 Parameters:

D%(0) = DMA Buffer A Offset

D%(1) = DMA Buffer A Segment

D%(2) = Samples for Buffer A

D%(3) = DMA Buffer B Offset (required for Dual DMA only)

D%(4) = DMA Buffer B Segment (required for Dual DMA only)

D%(5) = Samples for Buffer B (required for Dual DMA only)

MODE 1 Returns:

No returns.

Interpreted BASIC:

There are three ways to use this mode from Interpreted BASIC:

1. Specify the values returned by Mode 18 call, or
2. Specify an absolute address, or
3. Specify the address of a user array.

Method 1. is preferred while methods 2. and 3. are not recommended.

Example using Method 1:

```
100 ' ALLOCATE A MEMORY BUFFER
110 MD%  = 18
120 D%(0) = ASAMPLES     ' DESIRED SAMPLES
130 CALL DAS40 (MD%, D%(), FLAG%)
140 ' SAVE BUFFER PARAMETERS
150 ASAMPLES = D%(0)     ' ACTUAL SAMPLES
160 AOFFSET  = D%(1)     ' BUF A OFFSET
170 ASEGMENT = D%(2)     ' BUF A SEGMENT
180 ' PASS BUFFER PARAMETERS TO MD% 1
160 MD%  = 1
170 D%(0) = AOFFSET      ' BUF A OFFSET
180 D%(1) = ASEGMENT     ' BUF A SEGMENT
190 D%(2) = ASAMPLES     ' BUF A SAMPLES
200 D%(3) = 0
210 CALL DAS40 (MD%, D%(), FLAG%)
```

Example using Method 2:

```
520 MD% = 1
530 FLAG% = 0
540 D%(0) = 0                ' Offset 0
550 D%(1) = &H7000               ' Segment
560 D%(2) = 20000
```

When using an absolute address, you must make sure that memory location is not currently used as program or data area.

QuickBASIC:
There are two methods to use this mode from QuickBASIC:

1. Specify the values returned by Mode 18 call, or
2. Specify the address of a user array.

Method 1 is required for stand-alone EXE programs, while method 2 is required when running your programs from within the QuickBASIC integrated environment. User arrays used as DMA buffers must be declared as $DYNAMIC. Use VARSEG and VARPTR to get the segment and offset as shown below.

QuickBASIC example using Mode 18:

```
' ALLOCATE A MEMORY BUFFER
MD%  = 18
D%(0) = ASAMPLES     ' DESIRED SAMPLES
CALL DAS40 (MD%, D%(), FLAG%)
' SAVE BUFFER PARAMETERS
ASAMPLES = D%(0)     ' ACTUAL SAMPLES
AOFFSET  = D%(1)     ' BUF A OFFSET
ASEGMENT = D%(2)     ' BUF A SEGMENT

' PASS BUFFER PARAMETERS TO MD% 1
MD%  = 1
D%(0) = AOFFSET      ' BUF A OFFSET
D%(1) = ASEGMENT     ' BUF A SEGMENT
D%(2) = ASAMPLES     ' BUF A SAMPLES
D%(3) = 0
CALL DAS40 (MD%, D%(), FLAG%)
```

QuickBASIC example using a user array:

```
' DIMENSION A DYNAMIC ARRAY
REM $DYNAMIC
DIM BUFA(10000) AS INTEGER
REM $STATIC

' PASS BUFFER PARAMETERS TO MD% 1
MD%  = 1
D%(0) = VARPTR(BUFFA(0))          ' BUF A OFFSET
D%(1) = VARSEG(BUFFA(0))          ' BUF A SEGMENT
D%(2) = 10000                     ' BUF A SAMPLES
D%(3) = 0
CALL DAS40 (MD%, D%(), FLAG%)
```

4.3 MODE 2 - SETUP AN A/D OR D/A DMA OPERATION

Mode 2 specifies the Conversion Clock source, Start Trigger source, Single/Dual DMA mode, Single/Continuous cycle mode and optionally user Channel and Gain arrays. The Channel and Gain arrays are relevant when setting up an A/D operation. The Channel array is used when the desired channel scan consists of nonsequential channels or when sampling the same channel more than once in the same scan. The Gain array is relevant only when Channel array is specified. It is used when it is desired to sample the channels in the Channel array at different gains.

Alternatively, use Mode 3 for sequential channel scans at a fixed gain, and specify zeros in D%(0) thru D%(4). If specifying Channel and Gain array, D%(4) holds the number of entries in each array.

If using this mode for D/A operations, D%(0) thru D%(3) are not used and must be set to zero, and D%(4) specifies the DAC number as 0, 1 or 2 (both). The Conversion Clock source, Start Trigger

source, Single/Dual DMA mode, Single/Continuous cycle mode are common to both A/D and D/A operations. When specifying Internal Conversion Clock source, use Mode 6 to setup the desired rate.

MODE 2 Parameters:

D%(0) = Address Offset of a user Channel array

D%(1) = Address Segment of a user Channel array

D%(2) = Address Offset of a user Gain array

D%(3) = Address Segment of a user Gain array

D%(4) = Number of entries in channel/gain arrays: 1 - 16, OR

DAC channel number: 0, 1 or 2 (both channels)

D%(5) = Start Trigger source: 0 if internal, or 1 if external

D%(6) = Conversions Clock source: 0 if internal, or 1 if external

D%(7) = DMA mode: 0 if single DMA, or 1 if Dual DMA

D%(8) = DMA transfer mode: 0 if single cycle, or 1 if continuous

MODE 2 returns:

No returns

Interpreted BASIC:

The following code fragment illustrates how to specify channel and gain arrays from Interpreted BASIC.

```
640 MD% = 2
650 FLAG% = 0
660 D%(0) = VARPTR(CHAN%(0))      ' OFFSET TO CHANNEL ARRAY
670 D%(1) = -1                    ' FLAG TO USE DEFAULT DATA SEGMENT
680 D%(2) = VARPTR(GAIN%(0))      ' OFFSET TO GAIN ARRAY
690 D%(3) = -1                    ' FLAG TO USE DEFAULT DATA SEGMENT
700 D%(4) = 8                     ' NUMBER OF ENTRIES IN GAIN/CHAN ARR.
710 D%(5) = 0                     ' 0 ==> Internal Trigger
720 D%(6) = 0                     ' 0 ==> Internal Clock
730 D%(7) = 0                     ' 0 ==> Single Buffer DMA
740 D%(8) = 0                     ' 0 ==> Single Cycle
750 CALL DAS40(MD%, D%(0), FLAG%)
```

Where CHAN% and GAIN% are previously dimensioned as 16-integer arrays and initialized to the desired values. Refer to the supplied example BAEXAMPL.BAS for illustration.

Note when a Mode calls for segment and offset values, use VARPTR to obtain the offset and use -1 for the segment value. Specifying -1 informs the driver that this call is from Interpreted BASIC and that the Default Data Segment (DS) be used.

QuickBASIC:

The following code fragment illustrates how to specify channel and gain arrays from QuickBASIC.

```
MD% = 2
FLAG% = 0
D%(0) = VARPTR(CHAN%(0))      ' OFFSET OF CHANNEL ARRAY
D%(1) = VARSEG(CHAN%(0))      ' SEGMENT OF CHANNEL ARRAY
D%(2) = VARPTR(GAIN%(0))      ' OFFSET OF GAIN ARRAY
D%(3) = VARSEG(GAIN%(0))      ' SEGMENT OF GAIN ARRAY
D%(4) = 8                     ' NUMBER OF ENTRIES IN GAIN/CHAN ARRAY
```

```
D%(5) = 0                       ' 0 ==> Internal Trigger
D%(6) = 0                       ' 0 ==> Internal Clock
D%(7) = 0                       ' 0 ==> Single Buffer DMA
D%(8) = 0                       ' 0 ==> Single Cycle
CALL DAS40(MD%, D%(0), FLAG%)
```

Where CHAN% and GAIN% are previously dimensioned as 16-integer arrays and initialized to the desired values. Note when a Mode calls for segment and offset values, use the QuickBASIC functions VARPTR and VARSEG to obtain the offset and segment values of the given variable.

4.4 MODE 3 - SETUP THE A/D CHANNEL SCAN AND GLOBAL GAIN

This mode is used to setup the channel scan limits and a global gain value for the next A/D operation. Mode 3 is used when the desired channel scan consists of sequential channels that will be sampled at a fixed gain. Alternatively, use Mode 2 to specify Channel and Gain arrays whenever nonsequential channel scan and/or different Gains on different channels are desired.

MODE 3 Parameters:

D%(0) = Start channel number: 0 - 15 if Single Ended or 0 - 7 if Differential

D%(0) = Stop channel number: 0 - 15 if Single Ended or 0 - 7 if Differential

D%(2) = Global Gain code: 0 - 3

CODE	G1	G2
0	1	1
1	10	2
2	100	4
3	500	8

If a single channel is required, use that channel number for both the Start and the Stop channel #.

MODE 3 Returns:

No returns.

4.5 MODE 4: START A/D CONVERSIONS & TRANSFER VIA DMA

Mode 4 begins the A/D conversions and the transfer to memory via DMA according to the setup parameters specified in preceding calls to MODES 1, 2, 3, and 6. Once MODE 4 is initiated, the data conversion and transfer status may be monitored using calls to MODE 9. The DMA transfer will terminate when the specified conversion count(s) (as specified in MODE 1) is reached or a call to MODE 13 -- Terminate DMA Transfers -- is made.

Note that, once initiated, this mode operates in the background; therefore, you must use MODE 9 to determine when the data conversion(s) and transfer(s) are completed.

Also note that the maximum throughput of the DAS-40 (G2) is specified as 250 KHz in DMA modes. To achieve such throughput, Requests for DMA transfers by the DAS-40 must be serviced with minimum delay (latency). Screen I/O intensive applications are likely to increase the DMA Acknowledge latency and induce an A/D Data Overrun Error to be issued by the DAS-40 Driver.

When Internal trigger is specified in the last Mode 2 call, then this operation begins immediately. If External trigger is specified, then this operation begins when a high-to-low signal is detected at EXT/TRIG pin on the main connector.

If Internal Conversion Clock source is specified in the last Mode 2 call, then the conversions are paced according to the rate specified in the last call to Mode 6. If External Clock is specified, then conversions are paced at the clock rate as sensed at EXT/CLK pin on the main connector.

Note, A/D DMA and D/A DMA operations can not occur simultaneously.

Refer to Appendix B for more detail on DMA.

MODE 4 Parameters:

No parameters required.

MODE 4 Returns:

No returns.

4.6 MODE 5: START D/A CONVERSIONS & TRANSFER VIA DMA

Mode 5 begins transfer from memory and the D/A conversions via DMA according to the setup parameters specified in proceeding calls to MODES 1, 2 and 6.

Once MODE 5 is initiated, the data conversion and transfer status may be monitored using calls to MODE 9. The DMA transfer will terminate when the specified conversion count(s) (as specified in MODE 1) is reached or a call to MODE 13 -- Terminate DMA Transfers -- is made.

Note that, once initiated, this mode operates in background; therefore, you must use MODE 9 to determine when the data conversion(s) and transfer(s) are completed.

When Internal trigger is specified in the last Mode 2 call, then this operation begins immediately. If External trigger is specified, then this operation begins when a high-to-low signal is detected at EXT/TRIG pin on the main connector. If Internal Conversion Clock source is specified in the last Mode 2 call, then the conversions are paced according to the rate specified in the last call to Mode 6. If External Clock is specified, then conversions are paced at the clock rate as sensed at EXT/CLK pin on the main connector.

Note, A/D DMA and D/A DMA operations can not occur simultaneously.

MODE 5 Parameters:

No parameters required.

MODE 5 Returns:

No returns.

4.7 MODE 6: SETUP A/D & D/A CONVERSION PACING CLOCK

This mode sets up the Internal Conversion Clock rate for performing Clocked A/D and D/A operations. You must select for Internal Clock source through Mode 2 for this setup to relevant. This Mode accepts two integers that are collectively used to divide down the on-board 4 Mhz clock source. For a given frequency, you can determine these integer values as follows:

If the desired frequency is 5 KHz, then:

D%(0) * D%(1) = 4,000,000 / 5,000 = 800

Two possible integer values are:

D%(0) = 1
D%(1) = 800

Mode 0 initializes the default Clock rate to 40 KHz.

MODE 6 Parameters:

MD% = 6.

D%(0) = Number of 250 nSec pulses between conversions (1 - 65535).

D%(1) = Multiplier for value in D%(0) (16 - 65535).

MODE 6 Returns:

No returns.

4.8 MODE 7: DO ONE A/D CONVERSION

This mode performs one A/D conversion on the specified channel at the specified gain and requires no additional setup.

MODE 7 Parameters:

D%(0) = Channel number to read (S.E.: 0 - 15; Diff.: 0 - 7).

D%(1) = Gain Code value from following table.

CODE	G1	G2
0	1	1
1	10	2
2	100	4
3	500	8

MODE 7 Returns:

D%(0) = Digital Conversion value (0 - 4095).

D%(1) = Channel Gain (0 - 3).

4.9 MODE 8: DO ONE D/A CONVERSION

Mode 8 performs one D/A conversion on the specified DAC channel of the specified digital value. No previous setup is required for this mode. If D%(1) = 2 is specified, the digital value is simultaneously output on both DAC channels.

The allowed values for D%(0) are between 0 and 4095. A given value within this range reflects different voltage outputs depending on the current DAC range used (+/-10V, +/-5V, etc). For example, D%(0) = 1024 implies -5.0 V for ±10V range, 2.5V for +10V range, etc. (see Appendix A).

MODE 8 Parameters:

D%(0) = Digital value to be converted for output.

D%(1) = DAC channel number (0, 1 or 2).

MODE 8 Returns:

No returns.

4.10 MODE 9: MONITOR DMA TRANSFERS FOR ADCS & DACS

Following a call to MODE 4, MODE 5, or MODE 16, this mode should be used for monitoring progress of the A/D or D/A conversions. Data must not be accessed from the DMA buffers until either the terminal count is reached or DMA is terminated using MODE 13.

Note that a Data Lost condition, as stated below (D%(2)), means that an Interrupt Overrun Error has occurred during data acquisition/transmission. This will normally terminate the DMA transfer (set the DMA status inactive; D%(0) = 0). Refer to the Hardware Register sections on A/D Error D/A Error for explanations for a list of possible causes of this condition.

MODE 9 Parameters:

D%(0) = 0 to monitor ADCs;
1 to monitor DAC's.

MODE 9 Returns:

D%(0) = 1 if DMA is currently active; 0 otherwise.

D%(1) = 0 if DMA channel A is being used; 1 if DMA Channel B.

D%(2) = 0 if Data integrity is OK; 1 if Data Lost condition.

D%(3) = Number of conversions completed.

4.11 MODE 10: SETUP FOR DIGITAL I/O

Mode 10 is used to re-configure the 16-bit Digital I/O lines to one of four (4) possible configurations. A Configuration Code is passed into this MODE in D%(0). The Configuration Code may be one of the following:

CODE	CONFIGURATION
0	Word (16-bit) input
1	Byte (8-bit) input on Port1 and Byte output on Port0
2	Byte input on Port0 and Byte output on Port1
3	Word output

The Digital I/O lines are configured according to Code 0 whenever MODE 0 is called.

MODE 10 Parameters:

D%(0) = DIO configuration code (0, 1, 2 or 3)

MODE 10 Returns:

No returns.

4.12 MODE 11: INPUT BYTE/WORD

MODE 11 is called to read a digital value from the I/O lines. This value may be Byte or a Word depending on the last call made to MODE 0 or 10. The default value is a byte read from Port1.

MODE 11 Parameters:

No parameters required

MODE 11 Returns:

D%(0) = Byte/Word digital value.
Note that the input byte is in the least-significant eight bits of D%(0).

4.13 MODE 12: SET/QUERY CURRENT DAS-40 CARD NUMBER

The DAS-40 Driver supports one or two Physical DAS-40 cards simultaneously: Cards # 0 and 1. Mode 12 selects the card to be used. Selecting Card #1 when there is only one card in the system will NOT result in an immediate error; however, an error will be returned when attempting to initialize this card (MODE 0).

This mode is passed the card number to select for subsequent use. If a Card number other than 0 or 1 is passed to this MODE, the actual active card number is returned in D%(0).

This MODE Call is not required if you have one card in your system.

MODE 12 Parameters:

D%(0) = **0** or **1** to select an active card, or
other to query current active card.

MODE 12 Returns:

D%(0) = Current Active Card # (0 or 1).

4.14 MODE 13: TERMINATE CURRENT DMA OPERATION

This function terminates the current DMA transfer operation as initiated by MODEs 4, 5, or 16 and has no effect if DMA is not currently active.

WARNING: DO NOT TERMINATE A DMA OPERATION BY THE < **Ctrl** > - < **Break** > KEY SEQUENCE. IF YOU MUST TERMINATE A DMA OPERATION BEFORE ITS NORMAL COMPLETION, USE MODE 13 ONLY.

MODE 13 Parameters:

D%(0) = 0 to terminate ADCs, or 1 to terminate DACs.

MODE 13 Returns:

No returns.

4.15 MODE 14: OUTPUT BYTE/WORD

MODE 14 is called to write a digital value to the I/O lines. The value may be a Byte or a Word, depending on the last call made to MODE 0 or 10. The default value is a Byte written to Port 0.

MODE 14 Parameters:

D%(0) = Byte/Word digital value.
Note that the output byte is specified in the least-significant eight bits.

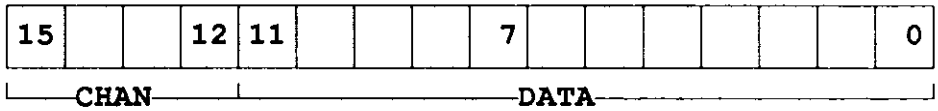
MODE 14 Returns:

No returns.

4.16 MODE 15: TAG CHANNEL NUMBERS TO A/D DATA SAMPLE

MODE 15 is optionally called immediately following a data acquisition operation (MODE 4 or 16) to tag all data samples collected with their corresponding source channel number. The Data as stored by MODEs 4 and 16 in the specified buffers is 12-bits long and occupies the Least Significant 12 bits of an integer (16-bits).

When called, this MODE adds the channel number (0 to 15) to the upper 4 bits (MSB) of the storage integer:



This MODE is useful when the sampled channels are not sequential and the data collected must be stored on disk for later processing. Note, that once a data buffer is tagged with channel numbers, these numbers must be stripped off by your program before the data can be analyzed.

MODE 15 Parameters:

D%(0) = 0 to tag channel numbers to data found in Buffer A;
1 to tag channel numbers to data found in Buffer B (Buffers A and B are as specified in the last call to MODE 1).

MODE 15 Returns:

No returns.

4.17 MODE 16: A/D TRIGGERED BLOCK SCAN (DMA)

MODE 16 is similar to MODE 4 in that it requires an identical setup. MODE 16, however, performs a block scan (one conversion from each channel specified) each time a trigger is detected. Once a Block Scan is triggered, the time between conversions is specified by the last call to MODE 6 or by an external clock signal received on the EXT CLK line.

If the trigger source is setup for Internal (see MODE 2), each call to this MODE effectively issues this trigger and a Channel Block Scan occurs. If the trigger source is setup for External, each high-to-low transition detected on the EXT TRG line (after calling MODE 16) causes a Channel Block Scan.

External triggers should be suppressed (EXT TRG held low or high) during a channel block scan; otherwise, a data overrun error may result. Note that the external trigger rate is a function of the conversion clock rate. The maximum external rate allowed can be determined as follows:

```
Max Ext Trigger Rate = conversion-rate / number-of-channels
```

Therefor if you want to scan 16 channels using the a conversion rate of 200KHz, the maximum External Trigger rate is 12.5KHz.

As with the Clocked DMA MODE 4, the progress of externally triggered Block Scans may be monitored using MODE 9 and terminated using MODE 13.

Some MODE 2 setup permutations such as Continuous Cycle and Internal trigger are not allowed.

MODE 16 Parameters:

No parameters required.

MODE 16 Returns:

No returns.

MODE 17: TRANSFER DATA TO BASIC ARRAY

MODE 17 is called after an A/D conversion and DMA transfer operation is completed (MODE 4 or 16). This mode will transfer the desired subset of the data collected to a user array and optionally provide the channel and gain information used during data acquisition. Specify the NULL pointer (0 Segment, 0 Offset) to suppress undesired information; Note that D%(0 - 5) are mandatory. The optional Gain and Channel arrays must be dimensioned to accommodate 17 integers each - the first integer value in each of the Channel and Gain Arrays is the number of channels scanned.

MODE 17 Parameters:

D%(0) = Address Offset of source data array.

D%(1) = Address Segment of source data array.

D%(2) = Address Offset of destination data array.

D%(3) = Address Segment of destination data array.

D%(4) = Number of samples to transfer.

D%(5) = Index of first sample to transfer.

D%(6) = Address Offset of destination Gain array (optional).

D%(7) = Address Segment of destination Gain array (optional)

D%(8) = Address Offset of destination Channel array (optional).

D%(9) = Address Segment of destination Channel array (optional).

Setting D%(5) to 0 implies that the transfer of data from the DMA buffer begins with the first sample collected. Specify other values for D%(4) and D%(5) to transfer any number of samples from anywhere within the DMA buffer.

MODE 17 Returns:

No returns.

Interpreted BASIC

The address (Segment and Offset) of the source data array: D%(0) and D%(1) is the same as the one passed to Mode 1 (Buffer A or B). All destination arrays are user dimensioned BASIC arrays and are specified to this mode using the using VARPTR to obtain the offset of the array and -1 for the segment. Specifying -1 informs the driver that this call is from Interpreted BASIC and that the Default Data Segment (DS) be used.

Example:

```
100 MD% = 17
110 FLAG% = 0
120 D%(0) = AOFFSET                ' OFFSET TO SOURCE DATA
130 D%(1) = ASEGMENT               ' SEGMENT TO SOURCE DATA
140 D%(2) = VARPTR(DBUF%(0))       ' OFFSET TO DEST DATA ARRAY
```

```
150 D%(3) = -1                      ' FLAG TO USE DEF DATA SEGMENT
160 D%(4) = 100                     ' NUMBER OF SAMPLES TO TRANSFER
170 D%(5) = 0                       ' INDEX OF 1ST SAMPLE TO TRANSFER
180 D%(6) = VARPTR(GBUF%(0))        ' OFFSET TO DEST GAIN ARRAY
190 D%(7) = -1                      ' FLAG TO USE DEF DATA SEGMENT
200 D%(8) = VARPTR(CBUF%(0))        ' OFFSET TO DEST CHANNEL ARRAY
210 D%(9) = -1                      ' FLAG TO USE DEF DATA SEGMENT
220 CALL DAS40(MD%, D%(0), FLAG%)
```

Where AOFFSET and ASEGMENT comprise the address of the buffer where the raw A/D data reside as passed to Mode 1 before the actual acquisition. DBUF%() is a user BASIC array dimensioned as 100 integers at a minimum (in this example!); it is used to store the request samples. GBUF%() and CBUF%() are two user BASIC arrays dimensioned as 17 integer each; they may be optionally used to store the gain and channel information associated with the data in DBUF%().

QuickBASIC

The address (Segment and Offset) of the source data array: D%(0) and D%(1) is the same as the one passed to Mode 1 (Buffer A or B). All destination arrays are user dimensioned BASIC arrays and are specified to this mode using the using the QuickBASIC functions VARPTR and VARSEG.

Example:

```
MD% = 17
FLAG% = 0
D%(0) = AOFFSET                 ' OFFSET TO SOURCE DATA
D%(1) = ASEGMENT                ' SEGMENT TO SOURCE DATA
D%(2) = VARPTR(DBUF%(0))        ' OFFSET TO DEST DATA ARRAY
D%(3) = VARSEG(DBUF%(0))        ' SEGMENT TO DEST DATA ARRAY
D%(4) = 100                     ' NUMBER OF SAMPLES TO TRANSFER
D%(5) = 0                       ' INDEX OF 1ST SAMPLE TO TRANSFER
D%(6) = VARPTR(GBUF%(0))        ' OFFSET TO DEST GAIN ARRAY
D%(7) = VARSEG(GBUF%(0))        ' SEGMENT TO DEST GAIN ARRAY
D%(8) = VARPTR(CBUF%(0))        ' OFFSET TO DEST CHANNEL ARRAY
D%(9) = VARSEG(CBUF%(0))        ' SEGMENT TO DEST CHANNEL ARRAY
CALL BASDAS40(MD%, D%(0), FLAG%)
```

Where AOFFSET and ASEGMENT comprise the address of the buffer where the raw A/D data reside as passed to Mode 1 before the actual acquisition. DBUF%() is a user array dimensioned as 100 integers at a minimum (in this example!); it is used to store the requested samples. GBUF%() and CBUF%() are two user arrays dimensioned as 17 integer each; they may be optionally used to store the gain and channel information associated with the data in DBUF%().

4.19 MODE 18 - ALLOCATE MEMORY BUFFER FOR DMA OPERATIONS

Mode 18 is used to allocate a memory buffer suitable for use by an A/D or D/A DMA operation. This mode is passed the desired size of the memory buffer as the number of samples (16-bit words) and returns two pointers (segment and offset pairs) that reference a memory area within the far heap that can accommodate the desired number of samples: a Buffer Pointer and a Block Pointer.

The Buffer Pointer (in D%(1) and D%(2)) references the address of a memory buffer that is guaranteed to be usable by MODE 1 without possible DMA page-wrap. The Block Pointer (in D%(3) and D%(4)) references the address of the allocated memory block; this pointer may or may not be the same as the Buffer Pointer. Both pointers should saved into user variables immediately following this Mode call. The Buffer Pointer is passed to MODE 1, while The Block Pointer is later used through MODE 19 to free (return to the heap) the memory area allocated by this mode.

Mode 18 is passed a desired number samples and it returns the actual number of samples that the memory buffer will accommodate. This actual number of samples may be less than the desired number due to memory limitations and DMA addressing rules. Always save the actual number of samples as returned in D%(0) into a user variable and later pass it to Mode 1.

The desired number of samples specified here may be between 0 and 65535; where 0 specifies the maximum possible: 65536. Some restriction apply to the number of samples depending on the language and/or language environment or the available free memory in your system. This MODE uses DOS' INT 21H function 48H to allocate a free memory block from the far heap. This will usually work whenever a far heap exists and it includes sufficient memory to accommodate the request.

For the BASIC languages, specifying a number greater than 32767 into an integer variable causes an Overflow error. The following illustrates how to specify 50,000 samples in D%(0):

```
D%(0) = 50000 - 65536
```

This Mode is usable from interpreted BASIC and QuickBASIC stand-alone EXE programs; however, it can not be used from programs running within the QuickBASIC Integrated Environment. When in the QB Environment, QB controls all available DOS memory making all calls to MODE 18 fail (Errors 43 or 44).

Finally, some definitions: a Page-wrap condition occurs when the effective DMA address and the sample count exceed 65535. The far memory heap is the memory area not occupied by any programs or data that is available while your program is running under DOS.

MODE 18 Parameters:

D%(0) = Number of samples; possible values: 0 to 65535; use 0 to specify 65536.

MODE 18 Returns:

D%(0) = Actual number of samples the memory buffer will accommodate. THIS NUMBER WILL BE LESS THAN OR EQUAL TO THE REQUESTED NUMBER DEPENDING ON THE AVAILABLE MEMORY.

D%(1) = Memory buffer address OFFSET that is suitable for use by the DMA A/D and D/A MODEs; use in MODE 1.

D%(2) = Memory buffer address SEGMENT that is suitable for use by the DMA A/D and D/A MODEs; use in MODE 1.

D%(3) = Actual Memory block address OFFSET that is passed to MODE 19 when this memory is no longer needed. This value is not necessarily the same as D%(1).

D%(4) = Actual Memory block address SEGMENT that is passed to MODE 19 when this memory is no longer needed. This value is not necessarily the same as D%(2).

4.20 MODE 19 - FREE MEMORY BUFFER

MODE 19 is used to free a memory block previously allocated through MODE 18. This MODE is passed the 'Actual' memory block address as returned by MODE 18 in D%(3) and D%(4). See MODE 18 for more detail.

MODE 19 Parameters

D%(0) = Actual Memory block address OFFSET as obtained through a previous call to MODE 18 (D%(3)).

D%(1) = Actual Memory block address SEGMENT as obtained through a previous call to MODE 18 (D%(4)).

MODE 19 Returns:

No returns.

■ ■ ■

❑

❑

❑

CHAPTER 5

REGISTER STRUCTURE & FORMAT

5.1 GENERAL

DAS-40 registers use 16 (eight 2-byte words) consecutive base addresses in the I/O space as follows (R = Read, W = Write):

LOCATION (HEX)		FUNCTION	TYPE
ADCSR Address	+0	A/D Control/Status (ADCSR)	R/W
"	+2	Channel-Gain List Control/Status (CHANCES)	R/W
"	+4	A/D Data (ADDAT)	R
"	+6	D/A Control/Status (DACSR)	R/W
"	+8	D/A Data (DADAT)	W
"	+A	DIO Data (DIODAT)	R/W
"	+C	Supervisory Control/Status (SUPCSR)	R/W
"	+E	Pacer Clock (TMRCTR)	R/W

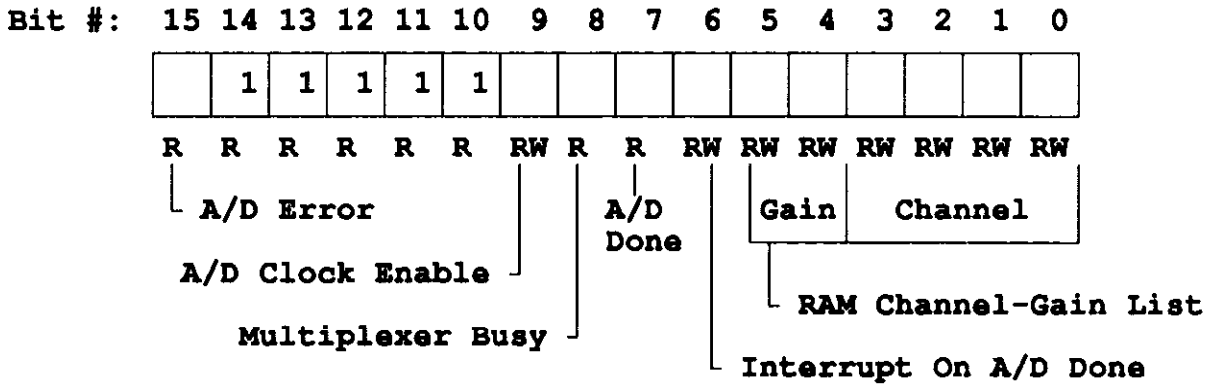
The I/O-area address for the ADCSR is jumper-selectable over a range of 200 to 3E0 (Hex) in increments of 20 (Hex). All other register addresses are offsets of this address. All registers are 16 bits wide and word-accessible only. The following sections describe each of these registers.

5.2 A/D CONTROL-STATUS REGISTER (ADCSR - BASE +0)

The ADCSR is a Read/Write register that controls the A/D section of the DAS-40. This register performs the following functions:

- Reports a Done state.
- Enables the A/D Done Interrupt.
- Loads the RAM Channel-Gain List.
- Reports a Multiplexer Busy condition.
- Enables the Pacer Clock to start A/D conversions.
- Reports an Error state.

ADCSR Register Map



ADCSR Bit Descriptions

Bit 15: A/D Error (A/DERR)

Bit 15 is Read Only. This bit sets to indicate an error caused by one of the following events:

- Starting a new A/D conversion before conclusion of the previous conversion causes a clock error. Since multiplexer settling time overlaps the A/D conversion, an A/D clock (internal or external) sets the error bit whenever the MUXBUSY signal (Bit 8) is 1.
- Starting an A/D conversion with a full DMA pipeline causes a data-overrun error. A full pipeline requires the ADDAT Register to be read before a new conversion can begin.
- During a triggered scan when SCDN (Bit 8 of the SUPCSR) is clear, re-triggering the board causes the error. Be sure SCDN is set before triggering the next scan.
- Triggering the board during a non-scan DMA (A/D or D/A) causes a trigger overrun error. One trigger enables the Pacer Clock for the entire DMA transfer. In A/D clocked DMA or D/A clocked DMA, do not trigger the board when DMA Done (SUPCSR Bit 15) is clear.

Bit 15 clears when the bus is reset, the board is initialized, or when ADCINIT (Bit 6 in the SUPCSR) is set.

Bits 14-10: Reserved

Bits 10-14 are unassigned. They are read back as 1s; writes to them get no response.

Bit 9: A/D Clock Enable (ADCLK)

Bit 9 is Read/Write. Setting this bit enables overflow from the Pacer Clock to start A/D conversion. Clearing this bit disables the Pacer Clock from causing conversions in the A/D section. You must set this bit in order to do A/D conversions. A bus reset or board initialization clears it.

Bit 8: Multiplexer Busy (MUXBUSY)

Bit 8 is Read Only. This bit sets when the multiplexer is settling. While MUXBUSY is set, a clock pulse (internal or external) causes A/DERR (Bit 15) to set. Bit 8 clears when the multiplexer settles and is ready to convert the analog value.

Bit 7: A/D Done (A/DDONE)

Bit 7 is Read Only. This bit sets when an A/D conversion is complete and the ADDAT Register contains valid data. Reading the ADDAT Register automatically clears this bit. If another conversion finishes before the A/D Data Register is read, two consecutive Reads are required to clear A/DDONE. When Bit 7 sets during DMA mode, a DMA transfer begins; the bit automatically clears after data transfer.

Bit 7 clears at bus reset, on board initialization, when ADCINIT (Bit 6 in the SUPCSR) sets, or with a reading of the ADDAT Register.

Bit 6: Interrupt On A/D Done (IA/DDONE)

Bit 6 is Read/Write. Setting the A/DDONE (Bit 7) while Bit 6 is set causes a CPU interrupt after completion of each conversion. In DMA mode, this bit enables DMAD (Bit 15 of the SUPCSR) to cause an interrupt.

Bit 6 sets and clears under programmed control, and it clears on a bus reset or board initialization.

Bits 5-4: Gain Select (GS)

Bits 4 and 5 are both Read/Write. They make up the programmable gain code for the RAM Channel-Gain List entry. These bits can be written to only after LLE (Bit 15 of the CHANCES) is set, but they can be read anytime. These bits represent the gain of the channel that is settling (not the gain of the channel under conversion).

Gain is coded according to the following table:

GAIN		GS1	GS0
G1	G2	BIT 5	BIT 4
1	1	0	0
10	2	0	1
100	4	1	0
500	8	1	1

Bits 4 and 5 set and clear under programmed control when LLE = 1. They are unaffected by bus reset and board initialization.

Bits 3-0: Channel Select (CHAN), Read/Write

Bits 0-3 are Read/Write. These bits specify the newest channel address to be added to the RAM

Channel-Gain List. Bits 0-3 can be written to only after LLE (Bit 15 of the CHANCES) is set, but they can be read anytime. When read, these bits represent the address of the channel that is settling (not the channel under conversion.

Channel addresses are coded according to the following table:

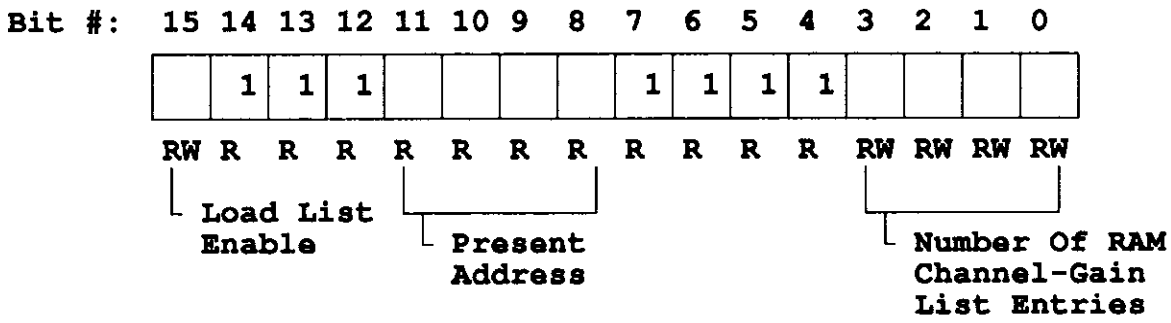
CHANNEL ADDRESS	CHANNEL ADDRESS BIT 3	2	1	0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Bits 3-0 can set and clear under programmed control when LLE = 1. They are random at power-up and are unaffected by bus reset or board initialization.

5.3 CHANNEL-GAIN LIST CONTROL-STATUS REGISTER (CHANCES - BASE +2)

CHANCES is a Read/Write register for activities affecting the RAM Channel-Gain List. You must take care to make only the specified number of writes to the RAM Channel-Gain List to avoid overwriting earlier entries. For example, if the CHANCES specifies three entries but you make six, the fourth, fifth, and sixth entries overwrite the first, second, and third entries.

CHANCES Register Map



CHANCES Bit Descriptions

Bit 15: Load List Enable (LLE)

Bit 15 is a Read/Write bit that must be set to load the RAM Channel-Gain List. When Bit 15 is clear, Writes to GS (Bits 4 and 5 of the ADCSR) and CHAN (Bits 3-0 of the ADCSR) are ignored. Bit 15 sets and clears under programmed control, and it clears on bus reset or board initialization.

Bits 14-12: Reserved

Bits 12-14 are unassigned. They are read as 1s, and writes to them are ignored.

Bits 11-8: Present List Address (PRESLA)

These are Read Only bits that represent the current address of the RAM Channel-Gain List. During loading, these bits give the address of the next entry.

Bits 11-8 clear during writes to the CHANCES. They also clear on a bus reset or board initialization.

Bits 7-4: Reserved

These bits are assigned. During reads, they appears as 1s. Writes to them are ignored.

Bits 3-0: Number Of RAM Entries (NUMB)

These are Read/Write bits that represent the number of entries in the RAM Channel-Gain List to be scanned. You may specify 1 to 16 entries for the scan.

NUMB is coded according to the following table.

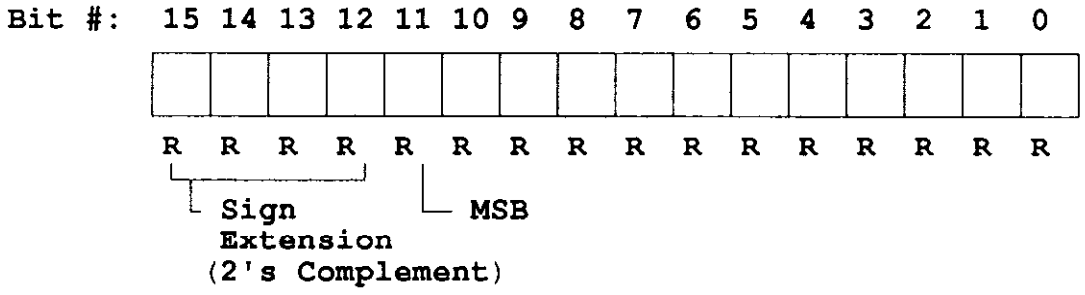
NUMBER OF ENTRIES	LAST RAM ADDRESS	NUMB CODING BIT 3	BIT 2	BIT 1	BIT 0
1	0	0	0	0	0
2	1	0	0	0	1
3	2	0	0	1	0
4	3	0	0	1	1
5	4	0	1	0	0
6	5	0	1	0	1
7	6	0	1	1	0
8	7	0	1	1	1
9	8	1	0	0	0
10	9	1	0	0	1
11	10	1	0	1	0
12	11	1	0	1	1
13	12	1	1	0	0
14	13	1	1	0	1
15	14	1	1	1	0
16	15	1	1	1	1

Bits 0-3 clear on a bus reset or board initialization.

5.4 A/D DATA REGISTER (ADDAT - BASE +4)

The ADDAT Register holds digital data from an A/D conversion. It is a Read Only register whose bit assignment appears in the following register map.

ADDAT Register Map



ADDAT Bit Descriptions

Bits 15-12: Sign Extension

Bits 12-15 read back as zeroes in Binary (Unipolar) and Offset Binary (Bipolar) Coding. In 2's Complement Coding, these bits provide sign extension for Bit 11.

Bits 11-0: Digital Data

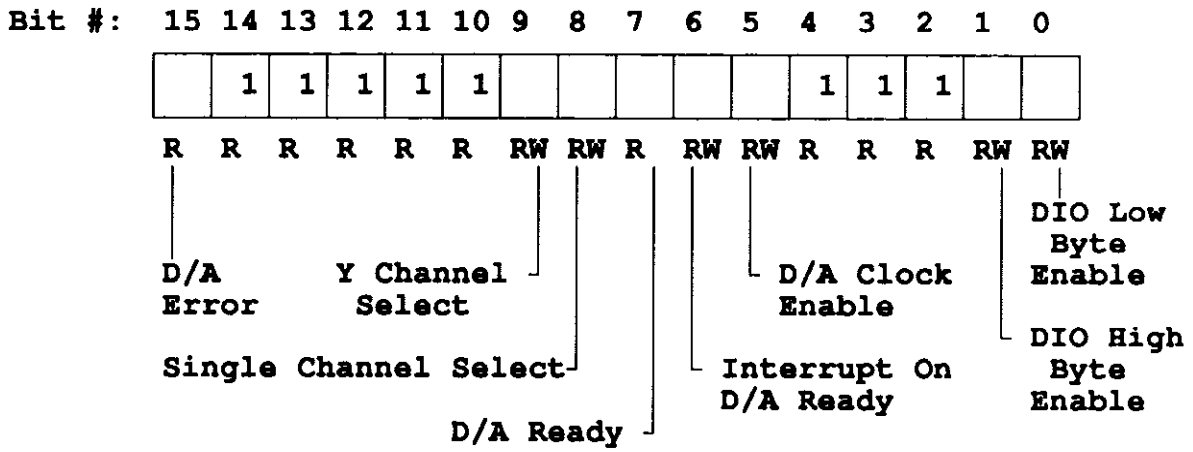
The DAS-40 uses only 12 bits of the ADDAT Register for digital data from an A/D conversion. Bit 11 is the MSB (Most Significant Bit).

Refer to the coding tables in Appendix A for examples of Binary, Offset Binary, and 2's Complement.

5.5 D/A CONTROL-STATUS REGISTER (DACSR - BASE +6)

DACSR is a Read/Write register that controls the D/A and DIO sections of the DAS-40.

DACSR Register Map



DACSR Bit Descriptions

Bit 15: D/A Error (D/AERR)

Bit 15 is a Read Only bit that sets in response to an error caused by one of the following events:

- Starting a conversion before a DMA transfer is complete is a Data Late Error in DMA Mode.
- Starting a conversion while DACRDY (Bit 7) is set is a Clock Error in Programmed I/O Mode.

Bit 15 clears on a bus reset or board initialization. It also clears by setting DACINIT (Bit 5 in the SUPCSR).

Bits 14-10: Reserved

Bits 10-14 are unassigned. They read back as 1s; they do not respond to writes.

Bit 9: Y Channel Select (YSEL)

Bit 9 is Read/Write. When SSEL (Bit 8) is set, Bit 9 determines whether the Y Channel (DAC 1) or the X Channel (DAC 0) is to receive data. Setting YSEL enables the Y Channel to receive data; clearing YSEL enables the X Channel to receive data. When SSEL is clear, YSEL is ignored.

Bit 9 sets and clears under programmed control; it is cleared on a bus reset or board initialization.

Bit 8: Single Channel Select (SSEL)

Bit 8 is a Read/Write bit that determines whether data switches between the two DACs or goes to one DAC. Setting SSEL enables Single Channel Mode and allows YSEL to determine the DAC. Clearing SSEL enables Dual-Channel (X-Y) mode disables response from YSEL (Bit 9).
Bit 8 sets and clears under programmed control; it is cleared on a bus reset or board initialization.

Bit 7: DAC Ready (DACRDY)

Bit 7 is a Read Only bit that sets when the D/A completes a conversion, indicating that new data can enter the DADAT Register. When this bit clears, it indicates that the D/A can be clocked.

Bit 7 clears with writes to the DADAT Register (once or twice, depending upon the state of SSEL, Bit 8); it sets following a clock-initiated D/A conversion (internal or external).

In DMA Mode, setting Bit 7 causes a DMA request; it clears upon completion of DMA transfer of the word (or words).

Bit 7 sets on a bus reset, on a board initialization, or by setting DACINIT (Bit 5 in the SUPCSR).

Bit 6: Interrupt On D/A Ready (ID/ARDY)

Bit 6 is Read/Write. Whenever it is set in non-DMA Mode, setting DACRDY (Bit 7) causes a CPU interrupt. When in D/A Clocked DMA Mode, Bit 6 enables DMAD (Bit 15 of the SUPCSR) to be set, which also causes a CPU interrupt.

In A/D Clocked or A/D Triggered Scan DMA Mode, DACRDY can not interrupt the CPU. You must therefore poll DACRDY to determine the state of the DAC.

Bit 6 sets and clears under programmed control; it also clears on a bus reset or board initialization.

Bit 5: D/A Clock Enable (DACLK)

Bit 5 is Read/Write. Setting this bit enables the Pacer Clock to start a D/A conversion.

This bit sets and clears under programmed control; it clears at bus reset or board initialization.

Bits 4-2: Reserved

Bits 2-4 are unassigned. They read back as 1s, and they ignore writes.

Bit 1: DIO High Byte Output Enable (HBOE)

Bit 1 is Read/Write. Setting this bit enables the high byte of the DIO port for digital output, and it enables the contents of the DIODAT Register to be present at Port 1. Clearing this bit enables the DIO high-byte-port lines (under external control) for digital input.

Bit 1 sets or clears under programmed control; it is cleared on a bus reset or board initialization.

Bit 0: DIO Low Byte Output Enable (LBOE)

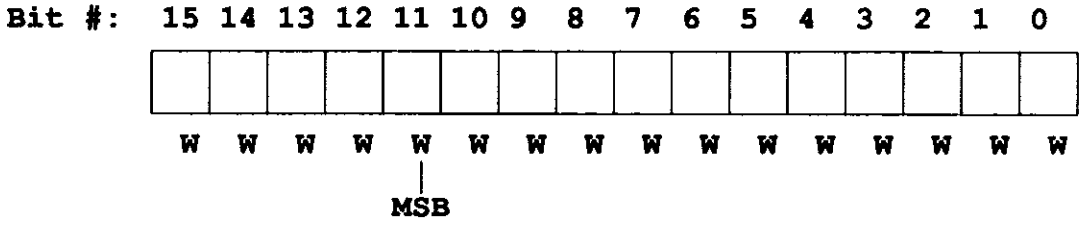
Bit 0 is Read/Write. Setting this bit enables the low byte of the DIO port for digital output, and it enables the contents of the DIODAT Register to be present on Port 0. Clearing this bit enables the DIO low-byte-port lines (under external control) for digital input.

Bit 0 sets or clears under programmed control; it is cleared on a bus reset or board initialization.

5.6 D/A DATA REGISTER (DADAT - BASE +8)

The DADAT Register is 12-bit, Write-only; it receives digital data to be converted by the DAC.

DADAT Register Map



DADAT Register Function

In Dual-Channel Mode, the first data word goes to the X Channel (DAC 0) and the second to the Y Channel (DAC 1); both are latched. Conversions on DAC 0 and 1 occur simultaneously with the first clock (internal or external) after the second data word is written to the DADAT.

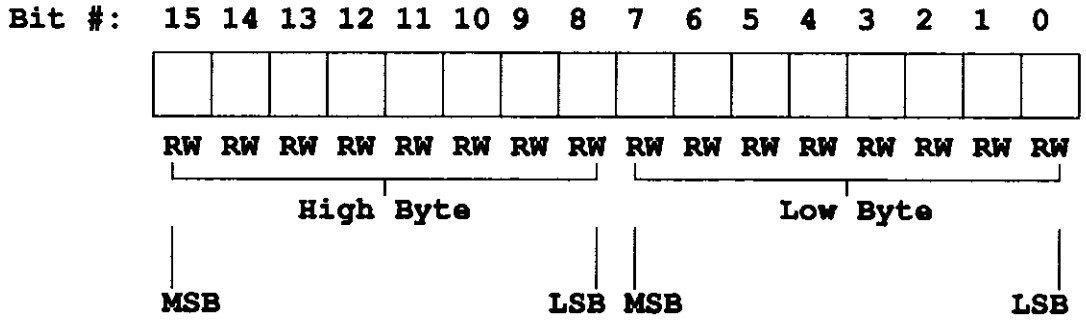
Refer to Appendix A for coding examples of Binary or Offset Binary. The DACs do not respond to 2's Complement coding.

5.7 DIO DATA REGISTER (DIODAT - BASE +0AH)

DIODAT is a Read/Write register that accepts digital data for transfer to or from the DAS-40. The states of LBOE and HBOE (DACSR Bits 0 and 1) determine the type of transfer at the DIO port. Setting a byte input causes writes to the corresponding byte in the DIODAT to go ignored, while reads of the corresponding DIODAT byte show the port's input line status. Changes in input lines appear immediately in the register. Setting a byte for output causes writes to the corresponding DIODAT byte to change the digital outputs while reads of the corresponding byte request digital output line status.

Since enabling a port for output causes transfer of the DIODAT Register contents to that port, the DIODAT Register is best loaded with required data **before** enabling the port for output.

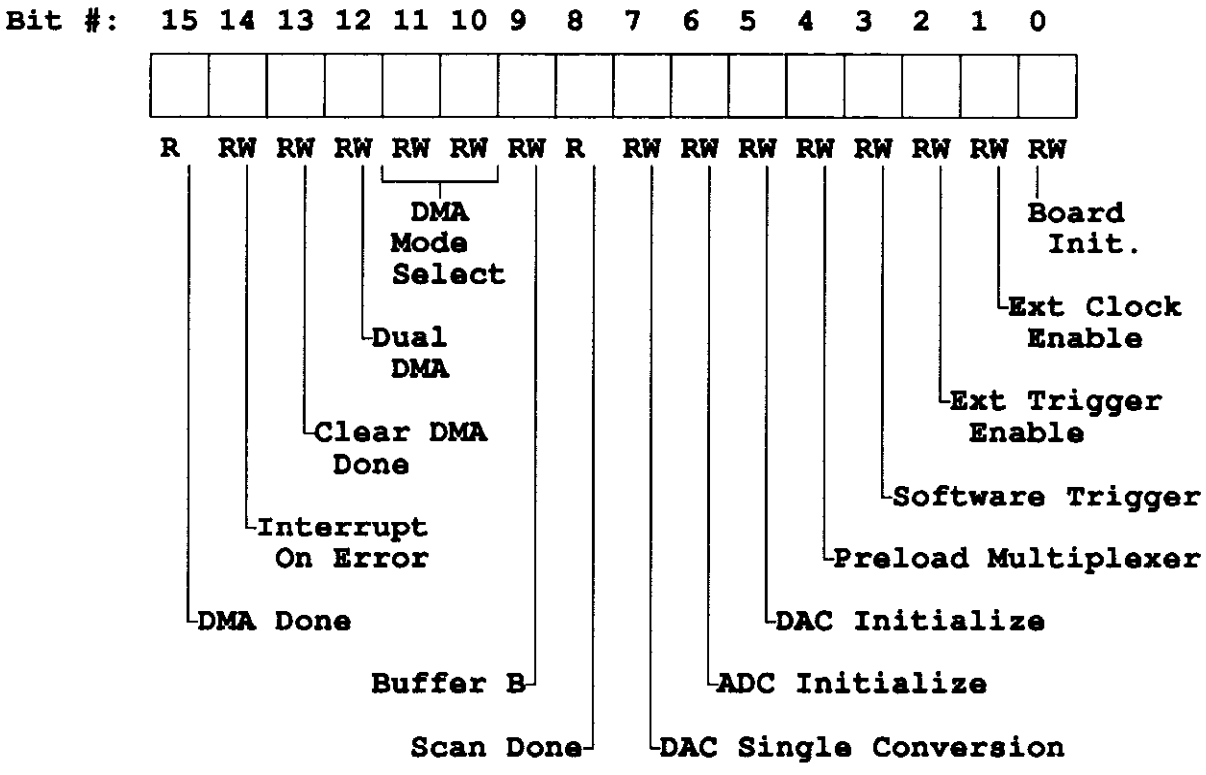
DIODAT contents are unaffected by a bus reset or board initialization. A map of this register is as follows:



5.8 SUPERVISORY CONTROL-STATUS REGISTER (SUPCSR - BASE +0CH)

SUPCSR is a Read/Write register for control of the Clock Source Control, the DMA Mode Select, and the Initialization Bits of the DAS-40 Board.

SUPCSR Register Map



SUPCSR Register Bit Descriptions

Bit 15: DMA Done (DMAD)

Bit 15 is Read Only. This bit sets when the Board has completed the requested transfer.

Bit 15 clears under programmed control (after setting Bit 13, CLRDMADNE). It also clears on a bus reset or board initialization.

Bit 14: Interrupt On Error (ERRINTEN)

Bit 14 is Read/Write. Setting this bit permits either A/DERR (Bit 15 of ADCSR) or D/AERR (Bit 15 of DACSR) to set and cause an interrupt. This process occurs according to the following table.

DMA MODE	BIT SET FOR INTERRUPT
0	A/DERR or D/AERR
1	A/DERR
2	D/AERR
3	A/DERR

See descriptions for Bits 10 and 11 for the DMA modes. Unless Bit 14 is set, no error will cause an interrupt.

Bit 14 sets and clears under programmed control. It also clears on a reset or board initialization.

Bit 13: Clear DMA Done (CLRDMADNE)

Writing a 1 to Bit 13 clears DMAD (Bit 15); a 0 has no effect.

Bit 13 sets under programmed control but reads as a 0.

Bit 12: Dual DMA (DDMA)

Bit 12 is Read/Write. Setting this bit selects Continuous Performance DMA Operation, in which two DMA channels transfer data continuously to or from memory.

Setting Bit 12 also produces the following two changes in DMA Control. First, a DMA transfer does not terminate when DMA Terminal Count occurs on Bus Finger B27. Second, the DMA switches with Terminal Count between two separate channels. The first DMA channel is switch-selected to be the default channel. Set Bit 12 only after switch-selecting the two DMA channels.

Bit 12 sets and clears under programmed control. It also clears on board initialization.

Bits 11-10: DMA Select (DS)

Bits 10 and 11 are Read/Write. These bits specify the DMA Mode according to the coding shown in the following table.

BIT 11 DS1	BIT 10 DS0	DMA MODE
0	0	No DMA (PIO)
0	1	A/D Clocked DMA
1	0	D/A Clocked DMA
1	1	A/D Triggered Scan DMA

Bits 10 and 11 set and clear under programmed control. They also clear on board initialization.

Bit 9: Buffer B (BUFFB)

Bit 9 is Read/Write. When Bit 12 is set, putting the Board in Dual-DMA Mode, Bit 9 indicates which DMA buffer is being used. When set, this bit signifies that Buffer 2 (Buffer B) is in use; when clear, this bit signifies that Buffer 1 (Buffer A) is in use. Single-buffer DMA must use Buffer 1; Dual-DMA must start with Buffer 1.

Writing a 1 to Bit 9 selects Buffer 1. A 0 is ignored. Write a 1 to this bit prior to starting DMA transfers.

Bit 9 is random at power-up and clears only by writing a 1 to it.

Bit 8: Scan Done (SCDN)

Bit 8 is Read Only. When this bit is set, it signifies completion of an A/D scan; and if the DMA Done bit is not set, Bit 8 indicates that another scan can be triggered. When Bit 8 is clear following a trigger, it signifies that the A/D is still scanning. Bit 8 can be ignored before the first trigger. When this bit sets during DMA Modes 0 and 3, clock pulses are disabled. This bit is not used in Modes 1 and 2, but in Mode 3 another trigger starts another scan. The occurrence of a trigger while Bit 8 is clear (after the first trigger) causes A/DERR (Bit 15) of the ADCSR to set.

Bit 8 is random on power-up and is unaffected by bus reset or board initialization.

Bit 7: DAC Single Conversion (DACON)

Setting Bit 7 initiates a single D/A conversion, but Bit 7 should be set only when DACRDY (Bit 7 of DACSR) is clear. Clearing this bit gets no response.

Bit 7 can force D/A conversions in programmed I/O Mode while the A/D is in DMA Mode.

Bit 7 clears on a board initialization. The bit always reads as 0.

Bit 6: A/D Initialize (ADCINIT)

Setting Bit 6 clears A/DERR (Bit 15 of the ADCSR) and initializes A/D data buffer circuits. Clearing this bit gets no response. This bit should be set prior to any A/D conversions.

Bit 6 sets under programmed control and always reads as 0.

Bit 5: D/A Initialize (DACINIT)

Setting Bit 5 clears D/AERR (Bit 15 of the DACSR) and initializes the DAC data buffers. Clearing this bit gets no response. Bit 5 should be set prior to any DAC conversions. Setting this bit does not change the DAC outputs.

Bit 5 sets under programmed control and always reads as a 0.

Bit 4: Preload Multiplexer (PRLD)

Setting Bit 4 preloads the first A/D channel (from the RAM Channel-Gain List) into the multiplexer. Clearing this bit gets no response. If the multiplexer is not preloaded, the first conversion takes place on the previous multiplexer channel. If multiplexer is not preloaded, the first conversion is not valid.

Bit 4 sets under programmed control and always reads as a 0.

Bit 3: Software Trigger (STRIG)

Setting Bit 3 starts the Pacer Clock. Clearing this bit gets no response. STRIG is a program-controlled trigger that can enable the internal or external clock (depending on XCLK–Bit 1).

NOTE: DO NOT SET STRIG AND XTRIG (BIT 2) AT THE SAME TIME.

Bit 3 sets under programmed control and always reads as 0.

Bit 2: External Trigger Enable (XTRIG)

Bit 2 is Read/Write. Setting Bit 2 allows the external trigger (on Pin 25 of J1) to start the clock. Either internal or external (depending on XCLK). A/D or D/A conversions occur via the clock after a trigger event.

NOTE: DO NOT SET XTRIG AND STRIG AT THE SAME TIME.

Bit 2 sets and clears under programmed control. It also clears on a bus reset or board initialization.

Bit 1: External Clock Enable (XCLK)

Bit 1 is Read/Write. Setting Bit 1 bit enables the external clock signal on Pin 50 of J1 as the Clock

Source for starting conversions. While this bit is set, the clocking rate is determined by the external clock signal. When this bit is clear, the clocking rate is determined by the internal oscillator and Pacer Clock Section.

Bit 1 sets or clears under programmed control. It also clears on a bus reset or board initialization.

Bit 0: Board Initialization (BDINIT)

Setting Bit 0 clears all error flags, the Control and Status Registers (ADCSR, DACSR, and SUPCSR), and the data buffers--returning the Board to the power-up state.. Clearing this bit gets no response. This bit does not affect the contents of DIODAT or DADAT.

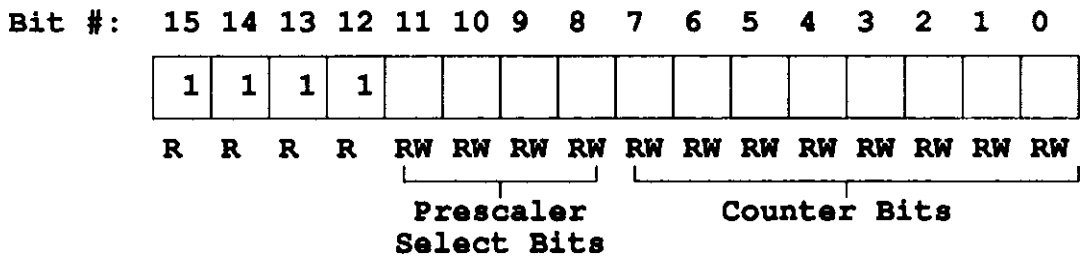
Bit 0 sets under programmed control and always reads as a 0.

5.9 PACER CLOCK REGISTER (TMRCTR - BASE +0EH)

The TMRCTR is a Read/Write register holding the counts for both the Pacer Clock and the prescale. When read, TMRCTR gives the current count and prescale of the divider. At overflow, the divider and prescale are reloaded with the original values.

Bits 8-11 make up a 4-bit prescale that divides the 4MHz clock by any power of 2 between 0 and 15 (1 to 32,768). Bits 0-7 form the 8-bit count for frequency set up by the prescaler bits. The frequency rate of A/D, D/A, or simultaneous A/D and D/A is determined by both the prescaler and counter bits.

TMRCTR Register Map



TMRCTR Register Bit Descriptions

Bits 15-12: Reserved

Bits 12-15 are unassigned and are read back as 1s.

Bits 11-8: Prescaler Select Bits (PRS3, PRS2, PRS1, and PRS0)

Bits 8-11 are Read/Write. These bits determine the value of the Prescaler Divider according to the coding shown in the table that follows. The Prescaler Divider output clocks the Pacer Clock.

Bits 8-11 set and clear under programmed control and are cleared on board initialization.

PRESCALE BITS 11	10	9	8	PRESCALE DIVISION	RESULTING FREQUENCY (HZ)
0	0	0	0	1	4000000.
0	0	0	1	1	4000000.
0	0	1	0	4	1000000.
0	0	1	1	8	500000.
0	1	0	0	16	250000.
0	1	0	1	32	125000.
0	1	1	0	64	62500.
0	1	1	1	128	31250.
1	0	0	0	256	15625.
1	0	0	1	512	7812.5
1	0	1	0	1024	3906.25
1	0	1	1	2048	1953.125
1	1	0	0	4096	976.5625
1	1	0	1	8192	488.2813
1	1	1	0	16384	244.1406
1	1	1	1	32768	122.0703

The resulting frequency is further divisible by the Counter Bits (Bits 0-7).

Bits 7-0: Counter Bits

Bits 0-7 are Read/Write. These bits are the divide-by count for the Pacer Clock in 1's Complement coding of the desired data value. A read of these bits provides the current count of the Pacer Clock.

Bits 0-7 set and clear under programmed control. They are random at power-up and are unaffected on a bus reset or board initialization.

DIVIDER (DECIMAL)	COUNTER BITS (1'S COMPLEMENT) HEX	DECIMAL
255	0	0
254	1	1
128	7F	127
127	80	128
126	81	129
1	FE	254
0	FF	255

■ ■ ■

CHAPTER 6

CALIBRATION

GENERAL

DAS-40 calibration uses a convenient software program to step you through the required measurements and adjustments. This program is entitled *DAS40CAL.EXE* and resides in your Distribution Software.

Equipment required for DAS40CAL.EXE is as follows:

- Voltage Calibrator (5- or 6-digit) such as an Electronic Development Corp. Model E100.
- Digital Voltmeter (5- or 6-digit) such as a Keithley Instruments Model 196 DMM.

■ ■ ■

CHAPTER 7

SPECIFICATIONS

NOTE: The following specifications are typical at 25°C unless stated otherwise.

7.1 A/D SUBSYSTEM

A/D Analog Inputs:

No. of Inputs	16 single-ended, 8 differential (switch-selectable).
Input Ranges	0 to +10V (Unipolar); ±5V (Bipolar); ±10V (Bipolar)
Output Data Codes	Straight binary (Unipolar), offset binary (Bipolar), or 2's complement (Bipolar)
Gain Range	G1: 1, 10, 100, & 500 ;
G2: 1, 2, 4, & 8	
Input Impedance	Off Channel: 100 MegOhms, 50pF; On Channel: 100 MegOhms, 120pF
Bias Current	±100pA
Common Mode Input Voltage	±11V max
Common Mode Rejection Ratio, Gain = 1	-80dB at 60Hz, 1 KOhm unbalanced
Max. Input Volts w/o Damage (Power ON)	±25V
Max. Input Volts Damage (Power OFF)	±25V
Channel-to-Channel Input Voltage Error	±5uV

A/D Accuracy:

Resolution	12 bits
Nonlinearity	Less than ±0.5 LSB
Differential Nonlinearity	Less than ±0.5 LSB
Inherent Quantizing Error	±0.5 LSB
System Accuracy	To within ±0.03% FSR
Channel Crosstalk	G1 = -94 dB
(At 1 KHz, with Rin = 1KOhm)	G2 = -100 dB
Gain Error	Adjustable to 0
Zero Error	Adjustable to 0

A/D Dynamic Performance:

Amplifier Input Noise - G1:	Gain = 1, 10:	0.05 LSB rms
	Gain = 100, 500:	0.005 LSB rms
Amplifier Input Noise - G2:	0.05 LSB rms	

NOTE: Where gain is greater than 1, input noise is multiplied by gain.

Input Bandwidth (-3dB)	G1: 150KHz (Gain of 1, 10) 6.6KHz (Gain of 100, 500) G2: 1.2MHz (Gain of 1) 1.1MHz (Gain of 2) 1.0MHz (Gain of 4) 0.9MHz (Gain of 8)
Chan. Acquisition Time to within 1/2 LSB	G1: 22us for Gain of 1 or 10; 400us for Gain of 100 or 500; G2: 2.5us
A/D Conversion Time	G1 & G2: 2.5us
A/D Converter Throughput	G1: 100 KHz for Gain of 1 50 Khz for Gain of 10 4 Khz for Gain of 100 2.5 KHz for Gain of 500 G2: 250KHz
A/D Throughput To System Memory	G1: 100,000 samples/s max; G2: 250,000 samples/s
Sample & Hold Aperture Uncertainty	G1 & G2: 0.2ns
Sample & Hold Delay	G1 & G2: 50ns
Harmonic Distortion	G1: -80 dB @ 100KHz (Gain of 10) G2: -80 dB @ 100KHZ (Gain of 8)

A/D Thermal Characteristics:

A/D Zero Drift	±1 ppm of FSR/°C
Amplifier Zero Drift	±5uV/°C
Gain Drift	±10 ppm of FSR/°C
Differential Linearity Drift	±1 ppm of FSR/°C
Monotonicity	0 to 70°C

7.2 D/A SUBSYSTEM

D/A Analog Outputs:

Number of Channels	2, deglitched
Output Ranges (Jumper Selectable)	0 to 5V (Unipolar); 0 to 10V (Unipolar): ±2.5V, ±5V, ±10V (Bipolar)
Output Data Coding (Jumper Selectable)	Straight binary (Unipolar) or Offset binary (Bipolar)
Throughput	130KHz max, single channel; 260KHz max, aggregate
Current Output	±5mA max
Output Impedance	0.1 Ohms max
Capacitive Drive Capability	100uF
Protection	Against short circuit to analog common
Glitch Energy	0.5mV-us

D/A Accuracy:

Resolution	12 bits
Nonlinearity	To within ±0.5 LSB
Differential Nonlinearity	To within ±0.5 LSB
Gain Error	Adjustable to 0
Zero Error	Adjustable to 0

D/A Dynamic Performance:

Settling Time to 0.01% of FSR	3us, 20V step; 1us, 100mV step
Slew Rate	150V/us

D/A Thermal Characteristics:

D/A Zero Drift	±1 ppm of FSR/°C
Gain Drift	±2 ppm of FSR/°C
Monotonicity	0 to 70°C

7.3 DIGITAL I/O SUBSYSTEM

Lines 16

Ports 2, 8-bit each (setable for input or output)

Digital Inputs:

Input Type	Level sensitive
Logic Family	ALSTTL
Logic Sense	Positive True
Logic Load	1 LSTTL load
Logical-High-Input Voltage	2.0V min
Logical-Low Input Voltage	0.8V max
Logical-High Input Current	20uA max
Logical-Low Input Current	-0.2mA max
Termination	None; unused inputs float

Digital Outputs:

Fanout 30 LSTTL loads

Logic Family	ALSTTL
Logic High Output Voltage	2.0V max
Logic Low Output Voltage	0.5V max
Logic High Output Current	-15mA max
Logic Low Output Current	24mA max
Throughput	As fast as programmed I/O allows (200KHz benchmark)

7.4 EXTERNAL TRIGGER

Input Schmitt Trigger, enables on TTL logic low

Logic Family	LSTTL
Logic Load	1 LSTTL load
Logical-High Input Voltage	2.0V min
Logical-Low Input Voltage	0.8V max
Logical-High Input Current	-0.25mA max
Logical-Low Input Current	25uA max
Min. Pulse Width:	High 100ns Low 100ns
Max. Pulse Width, Low	No restriction
Termination	22KOhm pull-up to +5V

7.5 CLOCK

Internal Clock:

Base Frequency	4.00MHz ±0.01%
Prescaler Range	Powers of 2 from 0 to 15 (1 to 32,768)
Divisor Range	Integer values 1 to 256
Frequency Range	4.0us (250KHz) to 2s (0.5Hz)

External Clock:

Input Schmitt Trigger; edge sensitive; clocks on falling edge.

Logic Family	LSTTL
Logic Load	1 LSTTL load
Logical-High Input Voltage	2.0V min
Logical-Low Input Voltage	0.8V max
Logical-High Input Current	-0.25mA max
Logical-Low Input Current	25uA max
Min. Pulse Width	High 200ns; Low 200ns
Termination	22KOhm pull-up to +5V

7.6 INTERFACE CHARACTERISTICS

Compatible Bus	IBM PC/AT
Interface	I/O mapped with 10-bit addressing
Occupied Locations	8 words reserved
Pre-set Base Address	240 (Hex); switch-selectable from 200 (Hex) to 3E0 (Hex) in increments of 20 (Hex)
Interrupts	1 interrupt; jumper configurable to any of 5 interrupt lines
Interrupt Lines	3, 5, 7, 10, and 15
Pre-set Interrupt Level	15
Interrupt Sources	A/D Error; D/A Error; A/D Done; D/A Ready; Trigger Error; DMA Done
Data Path	16 bits
DMA Channels	Channels 5, 6, or 7 (switch selected)

7.7 POWER REQUIREMENTS

+5V ±5% @ 3.3A max (2.4A typical)

NOTE: If an application uses the Â15V output, the load current should be limited to 20mA max.

7.8 PHYSICAL/ENVIRONMENTAL

Dimensions	4.5 x 13.25 x 0.75 in. (11.4 x 33.7 x 1.9 cm)
Weight 20 oz. (567g)	
Storage Temperatures	-13 to +185°F (-25 to +85°C)
Relative Humidity	To 95%, non-condensing
Operating Temperatures	32 to 158°F (0 to 70°C)

■ ■ ■

CHAPTER 8

FACTORY RETURNS

Before returning any equipment for repair, please call 508/880-3000 to notify MetraByte's technical service personnel. If possible, a technical representative will diagnose and resolve your problem by telephone. If a telephone resolution is not possible, the technical representative will issue you a Return Material Authorization (RMA) number and ask you to return the equipment. Please reference the RMA number in any documentation regarding the equipment and on the outside of the shipping container.

Note that if you are submitting your equipment for repair under warranty, you must furnish the invoice number and date of purchase.

When returning equipment for repair, please include the following information:

1. Your name, address, and telephone number.
2. The invoice number and date of equipment purchase.
3. A description of the problem or its symptoms.

Repackage the equipment. Handle it with ground protection; use its original anti-static wrapping, if possible.

Ship the equipment to

Repair Department
Keithley Data Acquisition Corporation
440 Myles Standish Boulevard
Taunton, Massachusetts 02780

Telephone 508/880-3000
Telex 503989
FAX 508/880-0179

Be sure to reference the RMA number on the outside of the package!

■ ■ ■

APPENDIX A

CODING TABLES

A.1 GENERAL

The binary-code representation of an analog voltage is referred to herein as an ADV, for *Analog Data Value*. The DAS-40 encodes/decodes ADVs as follows:

- In Straight Binary for voltages in a Unipolar range.
- In either 2's Complement (2's Complement coding is not available on the DACs) or Offset Binary for voltages in a Bipolar range.

To convert an ADV into a voltage value, you must know the FSR (Full Scale Range) and the NFS (Negative Full Scale) voltage values of the analog range. These values are described as follows:

RANGE	MODE	AVAILABILITY	FSR	NFS
0 to +5V	Unipolar	D/A	5V	0V
0 to 10V	Unipolar	D/A, A/D	10V	0V
±2.5V	Bipolar	D/A	5V	-2.5V
±5V	Bipolar	D/A, A/D	10V	-5V
±10V	Bipolar	D/A, A/D	20V	-10V

To convert an ADV into an analog voltage value for the output of an A/D converter working with Binary or Offset Binary coding, use the following equation:

Analog Voltage = ((ADV * FSR/4096) + NFS) / GAIN

Where *GAIN* = 1, 10, 100, or 500 for the DAS-40G1 or
1, 2, 4, or 8 for the DAS-40G2

To convert an ADV into an analog voltage value for the output of an A/D converter working with 2's Complement coding, use the following equation, with *GAIN* as above:

Analog Voltage = (ADV * FSR/4096) / GAIN

To determine the ADV from an analog voltage value from the output of a D/A converter working with Binary or Offset coding (2's Complement coding is not available on the DACs), use the following equation:

ADV = (Analog Voltage - NFS) * 4096/FSR

The remainder of this Appendix presents tables indicating ADVs for key points in the ranges of the A/D and D/A sections of the DAS-40. The tables list analog voltages in one column and corresponding digital code in another.

Table 1. 12-Bit A/D Bipolar Coding--Offset Binary.

ANALOG INPUT	DIGITAL CODING	ANALOG VOLTAGES GAIN = 1	GAIN = 2	GAIN = 4	GAIN = 8
+FS - 1 LSB	0FFF	+9.9951V	+4.9976V	+2.4988V	+1.2494V
+FS - 2 LSBs	0FFE	+9.9902V	+4.9951V	+2.4976V	+1.2488V
+1/2 FS	0C00	+5.0000V	+2.5000V	+1.2500V	+0.6250V
+1 LSB	0801	+4.883mV	+2.441mV	+1.221mV	+0.610mV
0	0800	0.0000V	0.0000V	0.0000V	0.0000V
-1 LSB	07FF	-4.883mV	-2.441mV	-1.221mV	-0.610mV
-1/2 FS	0400	-5.0000V	-2.5000V	-1.2500V	-0.6250V
-FS + 1 LSB	0001	-9.9951V	-4.9976V	-2.4988V	-1.2494V
-FS	0000	-10.0000V	-5.0000V	-2.5000V	-1.2500V

Table 2. 12-Bit A/D Bipolar Coding--2's Complement.

ANALOG INPUT	DIGITAL CODING	ANALOG VOLTAGES GAIN = 1	GAIN = 2	GAIN = 4	GAIN = 8
+FS - 1 LSB	07FF	+9.9951V	+4.9976V	+2.4988V	+1.2494V
+FS - 2 LSBs	07FE	+9.9902V	+4.9951V	+2.4976V	+1.2488V
+1/2 FS	0400	+5.0000V	+2.5000V	+1.2500V	+0.6250V
+1 LSB	0001	+4.883mV	+2.441mV	+1.221mV	+0.610mV
0	0000	0.0000V	0.0000V	0.0000V	0.0000V
-1 LSB	FFFF	-4.883mV	-2.441mV	-1.221mV	-0.610mV
-1/2 FS	FD00	-5.0000V	-2.5000V	-1.2500V	-0.6250V
-FS + 1 LSB	F801	-9.9951V	-4.9976V	-2.4988V	-1.2494V
-FS	F800	-10.0000V	-5.0000V	-2.5000V	-1.2500V

Table 3. 12-Bit A/D Unipolar Coding--Binary.

ANALOG INPUT	DIGITAL CODING	ANALOG VOLTAGES GAIN = 1	GAIN = 2	GAIN = 4	GAIN = 8
+FS - 1 LSB	0FFF	+9.9976V	+4.9988V	+2.4994V	+1.2497V
+FS - 2 LSBs	0FFE	+9.9951V	+4.9976V	+2.4988V	+1.2494V
+3/4 FS	0C00	+7.5000V	+3.7500V	+1.8750V	+0.9375V
+1/2 FS	0800	+5.0000V	+2.5000V	+1.2500V	+0.6250V
+1/4 FS	0400	+2.5000V	+1.2500V	+0.6250V	+0.3125V
0 + 1 LSB	0001	+2.441mV	+1.221mV	+0.610mV	+0.305mV
0	0000	0.0000V	0.0000V	0.0000V	0.0000V

Table 4. 12-Bit D/A Bipolar Coding--Offset Binary.

ANALOG INPUT	DIGITAL CODING	D/A RANGE ±10V	±5V	±2.5V
+FS - 1 LSB	0FFF	+9.9951V	+4.9976V	+2.4988V
+FS - 2 LSBs	0FFE	+9.9902V	+4.9951V	+2.4976V
+1/2 FS	0C00	+5.0000V	+2.5000V	+1.2500V
+1 LSB	0801	+4.883mV	+2.441mV	+1.221mV
0	0800	0.0000V	0.0000V	0.0000V
-1 LSB	07FF	-4.883mV	-2.441mV	-1.221mV
-1/2 FS	0400	-5.0000V	-2.5000V	-1.2500V
-FS + 1 LSB	0001	-9.9951V	-4.9976V	-2.4988V
-FS	0000	-10.0000V	-5.0000V	-2.5000V

Table 5. 12-Bit D/A Unipolar Coding--Binary.

ANALOG INPUT	DIGITAL CODING	D/A RANGE 0 TO +10V	0 TO +5V
+FS - 1 LSB	0FFF	+9.9976V	+4.9988V
+FS - 2 LSBs	0FFE	+9.9951V	+4.9976V
+3/4 FS	0C00	+7.5000V	+3.7500V
+1/2 FS	0800	+5.0000V	+2.5000V
+1/4 FS	0400	+2.5000V	+1.2500V
0 + 1 LSB	0001	+2.441mV	+1.221mV
0	0000	0.0000V	0.0000V

■ ■ ■

APPENDIX B

DIRECT MEMORY ACCESS (DMA)

B.1 THE IBM PC-AT DMA STRUCTURE

PC/XT/AT users may take advantage of the higher data-transfer speed of their computers by operating in the DMA mode. This mode relies on the computer's two Intel 8237 DMA Controllers (the First DMA Controller and the Second DMA Controller) to re-route the flow of data between an I/O device and memory for the greater speed. Normally, data transfers between an I/O device and memory pass through the CPU. But when the computer is in DMA mode, the Intel 8237s effect a CPU bypass to route data directly between the I/O device and memory. Bypassing the CPU avoids problems such as disruption of CPU activities by interrupts. Such problems limit the throughput of data.

The First DMA Controller provides four prioritized byte-transfer DMA channels, while the Second provides four additional word-transfer DMA channels. The First DMA Controller is not used by the DAS-40 and is therefore not discussed, here.

At the end of each A/D conversion, the ADC issues a DRQ (DMA Request). When the DMA Controller returns a DACK (DMA Acknowledge) signal, the ADC transfers the data sample (a 16-bit word) to a memory location put on the Address Bus by the DMA Controller. This transfer usually takes place in 3 to 4 micro-seconds and is undisturbed by interrupts, etc.

Events between a DRQ and the corresponding DACK begin with the DMA Controller receiving a DRQ from the DAS-40. Upon receipt of the DRQ, the DMA Controller issues an HRQ (Hold Request) to the CPU, asking it to relinquish control of the data and address buses. As soon as the CPU is able (within one machine cycle), it responds by returning an HLDA (Hold Acknowledge) to the DMA Controller. The DMA Controller then sends a memory location to the Address Bus and issues the DAS-40 a DACK, telling it to place the data sample on the Data Bus. The Controller also provides simultaneous I/O read (IOR) and memory write (MEMW) signals to effect the data transfer. Control is then returned to the CPU for at least one machine cycle before another DMA request is possible, thus preventing total bus takeover by the DMA Controller.

As mentioned earlier, the Second DMA Controller handles a total of four DMA channels. Three of these channels, Channels 5, 6, and 7, are unassigned and usable by the DAS-40. Up to two channels may be configured for a DAS-40, making the Board capable of continuous, gap-free data acquisition. When a DMA channel is assigned to a given DAS-40, it is unusable by any other devices in the computer.

B.2 THE DMA CONTROLLER REGISTERS

The Second DMA Controller (the word-transfer unit) handles 16-bit word data transfers. Similar to the First DMA Controller (XT), it handles only a 16-bit address but uses Address Lines A1-A16 with A0 always low for word accesses of up to 64K word space (one page). To address the full 16MB memory-address capability of the 80286 and 80386, a second register is used to specify an address page. The address page controls the remaining address lines and specifies one of 124 64K word pages.

The DMA Controller contains five registers that must be programmed before DMA transfers can occur. These registers are as follows:

Mode Register

This register is at Address 0D6H and is common to all channels. It controls the type of DMA transfer (I/O to Memory, Memory to I/O, or Channel Auto-initialization).

Mask Register

This register is at I/O Address 0D4H and is also common to all channels. It's primary use is to enable/disable the DMA channels.

Base and Current Address Register

This register is where the initial memory address is programmed. It is modified by the DMA Controller to reflect the address for the next transfer. Each DMA Channel has its own Address Register, as shown in the table below.

DMA Page select Register

This register is where the address-page information is programmed. Unlike the Base Address Register, this register remains fixed throughout the DMA operation. Each DMA Channel has it own Address Register as shown in the table below.

Word Count Register

This register is where the transfer count is programmed. It may be polled by an application programmed during DMA to determine the progress of the transfer. Each DMA Channel has it own Address Register as shown in the table below.

A sixth register, the Byte Pointer Flip/Flop (0D8H), is used throughout the programming of the above registers.

	REGISTER I/O ADDRESSES (HEX)		
DMA CHANNEL	**ADDRESS**	**PAGE**	**WORD COUNT**
5	0C4H	8BH	0C6H
6	0C8H	89H	0CAH
7	0CCH	8AH	0CEH

The DAS-40 driver (MDAS40.EXE) takes care of correctly programming these registers. If you intend to write your own driver, you will benefit by obtaining the source code for this driver (available from the manufacturer as Part # PCF-40).

B.3 DAS-40 A/D DMA MODES

The DAS-40 driver supports four types of A/D Clocked DMA operations, as follows:

1. Single Channel/Single Cycle
2. Single Channel/Continuous Cycle
3. Dual Channel/Single Cycle
4. Dual Channel/Continuous Cycle

All types except the first require the DAS-40 board be configured to use two DMA Channels.

When using Single-Channel/Single-Cycle (SCSC) mode, the DMA Controller is programmed to use the first DMA channel and the DMA Mode Register is setup for I/O to memory transfer in Non-Auto-initialize Mode. Thus, when the DMA Controller completes all requested transfers and issues its Terminal Count interrupt, the software interrupt Service Routine (ISR) performs all necessary I/O operations to the DAS-40 to halt further conversions. With this mode, you may acquire up to 64K Word data samples at the highest throughput rate allowed for the Board.

The Single-Channel/Continuous-Cycle (SCCC) is similar to the SCSC mode except that the DMA Controller's Mode register is programmed to auto-initialize and that it uses two DMA channels. Both Channels are programmed with the same information. The Terminal Count from one channel triggers the other channel, and the first channel automatically resets its Base Address and Word Count registers to their initial values; the data transfers continue. This mode continues indefinitely and must be terminated by the application program (DAS-40 Driver MODE 13). The progress of this mode may also be monitored by the application (DAS-40 Driver MODE 9). The interrupt caused by the Terminal Count is ignored by the ISR.

The Dual DMA/Single Cycle (DDSC) is similar to the SCSC except that two DMA channels are used, each of which could possibly use different Memory addresses and Word Counts. The Terminal Count of the first channel triggers the second channel; while the Terminal Count of the second channel is handled by the ISR as in the SCSC mode. This mode can provide up to 128K Word of continuous data (gap-free) at the board's maximum throughput.

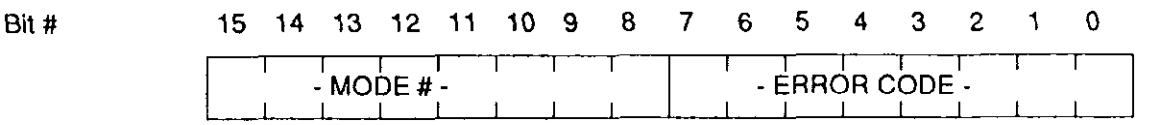
Finally, the Dual DMA/Continuous Cycle (DDCC) mode is similar to the DDSC mode except the two DMA channels are programmed to auto-initialize. Therefore, the Terminal Count of each channel triggers the other channel, and that channel automatically resets its Base Address and Word count, and it waits to be triggered by the channel currently running. In this mode, large amounts of continuous data may collected at speeds limited only by the speed of the transfer method used to store away the data in idle buffer. This mode continues indefinitely and must be terminated by the application program (DAS-40 Driver MODE 13). The interrupts caused by the Terminal Count signals are ignored by the ISR.

■ ■ ■

APPENDIX C

SUMMARY OF ERROR CODES

In general, you receive reports of error conditions via the Error Flag. This flag is of an unsigned integer type (16 bits) and contains the MODE number and Error Code number in the following format:



The Error Code is obtained by ANDing the error flag with 255 (0FFH). An Error Code of zero implies that the MODE Call was successful.

The following list contains Error Code definitions and suggested actions.

Error 1: Function not supported

Meaning:	A request is made to a function not supported by the DAS40 driver. This error should not occur in a standard release software.
Action:	Contact the manufacturer for Technical Support.

Error 2: Function out of bounds

Meaning:	Illegal function number is specified. This error should not occur in a standard release software.
Action:	Contact the manufacturer for Technical Support.

Error 3: Illegal board number

Meaning:	Illegal board number is detected.
Action:	Specify board number 0 or 1 through MODEs 0 or 12.

Error 5: Interrupt overrun

Meaning:	This error is issued from the Interrupt Service Routine (ISR) as invoked by the interrupt generated from the DAS-40. This error indicates that the A/D Error bit (ADCSR) or D/A Error bit (DACSR) is detected as set (1). Refer to the appropriate DAS-40 register description section for more detail. This error is usually caused by one or more of the following: 1. Conversion rate faster than the throughput of the board/PC system. 2. An External Trigger occurred during a channel scan.
Action:	Check A/D or D/A conversion parameters and retry.

Error 6: DMA already active

Meaning:	An attempt to start a DMA operation with DMA already active. Note that simultaneous A/D and D/A DMA are not allowed.
Action:	Terminate DMA operation using MODE 13, and retry.

Error 7: A/D subsystem not initialized

Meaning:	An attempt to perform a A/D operation without first initializing the A/D subsystem.
Action:	Repeat the operation or use MODE 0 to initialize the board.

Error 8: Invalid DMA channel number specified

Meaning:	An attempt is made to use a DMA mode without the proper DMA resources being available. For example, the Board is not configured to use DMA channel(s) as requested.
Action:	Reconfigure board to use DMA channels 5, 6 or 7.

Error 10: A/D conversion error

Meaning:	The A/D Error bit in the ADCSR Register is detected as set (1) during a synchronous A/D operation. Refer to the ADCSR register description section for more detail.
Action:	Repeat the operation or use MODE 0 to initialize the board.

Error 11: D/A subsystem not initialized

Meaning:	An attempt to perform a D/A operation without first initializing the D/A subsystem.
Action:	Repeat the operation or use MODE 0 to initialize the board.

Error 12: D/A conversion error

Meaning:	The D/A Error bit in the DACSR Register is detected as set (1) during a synchronous D/A operation. Refer to the DACSR register description section for more detail.
Action:	Repeat the operation or use MODE 0 to initialize the board.

Error 13: Odd DMA buffer pointer detected

Meaning:	This error is issued whenever the Address OFFSET value of a specified DMA buffer is odd; i.e. it falls on a byte boundary. Such an OFFSET is unusable by the DMA controller for word transfers. The error is returned from Mode 1.
Action:	Use MODE 18 to allocate a DMA buffer. The DMA buffer address returned from MODE 18 is guaranteed to be on a word boundary and therefore usable by the DMA controller.

Error 15: Digital input port not initialized

Meaning:	An attempt to perform a Digital Input operation without first initializing the Digital I/O subsystem.
Action:	Repeat the operation or use MODE 0 or 10.

Error 16: Digital output port not initialized

Meaning:	An attempt to perform a Digital Output operation without first initializing the Digital I/O subsystem.
Action:	Repeat the operation or use MODE 0 or 10.

Error 17: DAS-40 not at configured address

Meaning:	This error is issued from MODE 0 whenever the board presence test fails. This is normally caused by a conflict in the specified board I/O address and the actual I/O address the board is physically configured for. Also, this error is issued when the board is not present in the system.
Action:	Check the board's I/O address dip switch.

Error 20: Possible array index out-of-bounds condition.

Meaning:	This error is issued from MODE 17 whenever the Start sample index and the sample count specified will cause array index out-of-bounds.
Action:	Check input parameters D%(4) and D%(5) when calling MODE 17.

Error 22: DMA Address wrap condition

Meaning:	The combination of the DMA buffer address (Segment and Offset) and the sample count as specified through MODE 1 will cause a DMA page wrap.
Action:	Obtain the DMA buffer through a call to MODE 18 and retry.

Error 24: Illegal Channel/Gain array size

Meaning:	The Channel/Gain array size specified in MODE 2 must be between 1 and 16 inclusive.
Action:	Specify array size within valid range.

Error 25: Trigger source flag must be 0 or 1

Meaning:	Trigger signal source must be 0 (Internal) or 1 (External).
Action:	Re-specify Trigger signal source for MODE 2.

Error 26: Conversion Clock source flag must be 0 or 1

Meaning:	Clock signal source must be 0 (Internal) or 1 (External).
Action:	Re-specify Clock signal source for MODE 2.

Error 27: DMA Cycle flag must be 0 or 1

Meaning:	DMA Cycle mode flag must be 0 (Single Cycle) or 1 (Continuous Cycle).
Action:	Re-specify DMA Cycle flag for MODE 2.

Error 28: DMA data buffering flag must be 0 or 1

Meaning:	DMA data buffering flag must be 0 (Single buffer) or 1 (Double buffer).
Action:	Re-specify DMA data buffering flag for MODE 2.

Error 29: Channel number must be 0 - 15 (0 - 7 If Differential)

Meaning:	This error is issued from MODEs 2 or 3 whenever an illegal channel number is detected.
Action:	Specify channel number(s) within the board's current mode (S.E. or Differential).

Error 30: Gain code must be 0: 3

Meaning:	This error is issued from MODEs 2 or 3 whenever an illegal Gain code is detected.
Action:	Specify a valid Gain Code.

Error 31: DAC Channel number must be 0, 1 or 2

Meaning:	An invalid DAC channel number is specified. Note that specifying DAC Channel 2 implies that the next D/A output is sent to both DAC channels simultaneously.
Action:	Specify a valid DAC channel number for MODE 8.

Error 32: Unable to check DMA transfer status

Meaning:	This error is issued from MODE 9 whenever an attempt to interrogate the DMA controller and/or the DAS-40 for DMA transfer status fails.
Action:	Repeat DMA operation after first calling MODE 13 to terminate any DMA operations currently active.

Error 33: Source/Destination Buffer pointer is NULL

Meaning:	The specified buffer pointer is detected as NULL (0:0).
Action:	Check pointers passed to the MODE and repeat operation.

Error 34: Number of array transfers must be greater than 0

Meaning:	The number of sample transfers to array in MODE 17 must be other than 0.
Action:	Check number of transfers.

Error 35: Digital I/O ports Configuration Code must be 0, 1, 2, or 3

Meaning:	Illegal DIO configuration Code detected. This error is issued as required from all DIO MODEs.
Action:	Specify valid DIO Configuration Code.

Error 37: Clock multiplier must be greater than 15

Meaning: This error is issued from MODE 6.

Action: Specify multiplier value between 16 and 65535 inclusive.

Error 38: Clock ticks must be greater than 0

Meaning: This error is issued from MODE 6.

Action: Specify 250 nSec tick count greater than 0.

Error 39: Buffer number must be 0 (A) or 1 (B)

Meaning: This error is issued from MODE 15. You must specify 0 if you want to tag the data found in DMA buffer A, or 1 for DMA buffer B.

Action: Specify buffer code.

Error 40: Unsupported MODE number

Meaning: This error indicates that the MODE specified number is out of range.

Action: Specify a MODE number from those contained in Chapter 4.

Error 41: Driver not initialized

Meaning: This error indicates that the DAS-40 initialization (via MODE 0) did not precede a Call for a higher MODE.

Action: Call MODE 0 before calling any other MODE.

Error 42: FATAL ERROR WHILE ALLOCATING MEMORY

Meaning: You should never see this error. In it's attempt to determine the maximum available heap space, Mode 18 attempts to allocates 640K Byte (not a misprint!); This allocation should always fail and the size of the largest contiguous memory block is returned instead.

Action: If this error occurs, your program will immediately crash. If this error occurs repeatedly, contact the manufacturer.

Error 43: NOT ENOUGH FREE MEMORY TO ALLOCATE

Meaning: This error is reported by Mode 18 whenever the requested memory buffer can not be accommodated (for DMA purposes!) within the largest available contiguous memory block.

Action: Reduce the sample size, and retry.

Error 44: ERROR DURING MEMORY ALLOCATION

Meaning:	An error was reported by DOS following an INT 21H function 48H operation. This is typically a result of attempting to use Mode 18 from within the QB Integrated Environment.
Action:	Refer to Mode 18 for proper usage.

Error 255: $DAS40 Device not found

Meaning:	During board initialization the attempt to locate the DAS-40 device driver failed. This is normally due to the Device Driver not being installed.
Action:	Install MDAS40.EXE.

■ ■ ■