DAS-50

Keithley MetraByte Corporation

A Subsidiary of Keithley Instruments, Inc.
440 Myles Standish Boulevard
Taunton, Massachusetts 02780

Part Number: 24851

First Printing: August 1988
Revision Level C: July 1990

Keithley MetraByte Corporation
440 Myles Standish Boulevard
Taunton, Massachusetts 02780

WARNING
Keithley MetraByte Corporation 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.

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 GENERAL INTRODUCTION

Keithley MetraByte's DAS-50 is a high-speed, self-contained Analog-to-Digital Interface Board. The DAS-50 is designed for use with the IBM ® PC/XT/AT and compatible computers. Features of the DAS-50 include a 1 MHz conversion rate and up to 1 Megaword of on-board system memory. These make the DAS-50 the ideal board in applications such as:

- Laboratory Automation
- Process Monitoring Control
- Vibration/Stress Analysis
- Event/Transient Analysis
- Signal/Sensor Interface

The DAS-50 is easily programmable using MetraByte provided software. One alternative is a user-friendly Setup Menu. This can be used to set the DAS-50's operating parameters without a complicated application program. The PC's keyboard or a Microsoft compatible mouse are used to enter values for the operating parameters. If necessary, these values are automatically rounded to a legitimate value. Additionally, traces can be started or stopped, and diagnostics run from the Pop-Up Menu. Thus, even a novice computer user can program the DAS-50 to operate at its full potential.

For users who have more sophisticated applications, MetraByte provides a library of Language Independent Interface commands. These ASCII commands are incorporated into a user-designed application program. Application programs can be programmed in several languages including: BASIC ®, PASCAL, C, and FORTRAN.

1.2 FUNCTIONAL DESCRIPTION

The DAS-50 accepts data from up to four single-ended input channels. An analog multiplexer can be programmed to accept data from 1,2 or 4 of the input channels. (See the Set Channel command for more information.) If more than one channel is selected, the multiplexer automatically switches to the next channel in the sequence. When the multiplexer changes channels, the track and hold amplifier maintains the status of the previous channel. The data is then converted by the A/D converter and stored sequentially in the on-board memory. The data can be read from memory via the PC Bus. Data may be read sequentially or by channel.

MEMORY

The on-board memory is configured as banks of twelve 256K x 1 DRAM (Dynamic Random Access Memory). Between 1 and 4 banks of DRAM can be installed on the DAS-50 Memory Board, allowing for total memory capacities (in words) of 256K, 512K, 768K, and 1M.

INPUT VOLTAGE RATE

The A/D Converter can operate in either unipolar or bipolar mode with software selectable input ranges of: 0 to 5V, 0 to 10V, ± 2.5V, ± 5V, or ± 10 V. The input voltage range must be specified prior to the start of a trace. The voltage ranges can not be changed after a trace has begun.

ACQUISITION RATE

The rate of data acquisition can be determined by either a variable internal clock or an external TTL clock. If the internal clock is used, a conversion rate in the range 1 MHz to 137 Hz must be assigned. If an external clock is used, the maximum clock frequency is 1 MHz.

TRACE MODES

A trace is a series of a defined number of samples. Triggers are used to signal the stop or start of a trace. Traces can be initiated before, after, or about the time a trigger occurs. If the Before trigger mode is selected, data is sampled until a trigger occurs. In the After trigger mode, the trigger initiates data collection to begin. The About trigger mode allows data to be collected unconditionally until a trigger occurs, then a specified number of samples are taken after the trigger occurs.

TRIGGERS

The trigger pulse can originate from any of three sources: the program, a Digital Pulse, or an Analog Voltage level from Channel 0. The Digital Pulse and Analog Voltage triggering can further be specified to be either level or edge sensitive. Each of these has different set-up parameters which are described below:

Program - When the trigger origin is the user program, only the Trace After trigger mode can be selected. (Refer to Chapter 3 for more information regarding trigger modes.)

Digital Pulse - When the trace is triggered by a digital pulse, a positive or negative level or edge must be chosen. This is a TTL-trigger pulse with timing requirements as described in section 4.6. Several trace modes can be selected which allow data to be collected after a trigger occurs, before a trigger occurs, or before and after a trigger takes place.

Analog Voltage - This trigger-method uses the voltage level found on Channel 0. In this type of triggering, a positive or negative level or edge, and the trigger voltage level must be chosen. Valid voltage levels are between -9.99 and + 9.99 V. For a level sensitive trigger, the DAS-50 will trigger above the specified voltage level for positive level triggering, and below the specified voltage level for negative level triggering. The edge sensitive trigger causes the DAS-50 to trigger on the positive-going or negative-going edges.

Note that the significant difference between the level and edge triggering for both Digital and Analog modes is at the start of a trace. If level triggering is selected, the trace may be immediately triggered when a trace is started if the trigger conditions are already valid. For example, suppose an Analog positive level trigger of +2.00 volts is selected. If the Channel 0 voltage is +3.00 volts at the start of the trace, the trace will be triggered immediately, regardless of whether the voltage is rising or falling. This may or may not be what is really desired. On the other hand, if positive edge sensitive triggering is selected, the trace would not be triggered until the Channel 0 voltage falls lower than + 2.00 volts and then crosses the +2.00 volt boundary in the specified positive direction.

1.3 FILES ON THE DAS-50 UTILITY DISK

The files contained on the DAS-50 Utility Disk are listed in Table 1-1. They are also listed in the *FILES.DOC* file on the Utility Disk.

Table 1-1. Files on DAS-50 Utility Disk

Filename	Description
VI.SYS	Support files for Setup Screen and command handler.
DAS50DRV.SYS	System files for the DAS-50.
FILES.DOC	Listing of files contained on disk.
README.DOC	File providing a quick installation guide and any additional information regarding the DAS-50 not contained in this manual.
D50SETUP.EXE	Program which provides automatic setup of the user's *CONFIG.SYS* file.
CAL50.EXE	Base Address selection, calibration, and memory test program. (Also provided as source code).
CLOG50_1.EXE	Executable example program. Shows acquisition of 48 samples from 2 channels in TRace-After-Trigger mode, logged sequentially. (Also provided as source code)
CAL50.BAS	Source code (in BASIC) for DAS-50 Calibration Procedure.
D50CMD.BAS	Example program (in BASIC) illustrating the use of the Language Interface Commands. Requires keyboard input and displays results.
LOG50_1.BAS	QBASIC example for the acquisition of 48 data samples from 2 channels in the TRace-After-Trigger mode, data is logged sequentially.

Table 1-1. Files on DAS-50 Utility Disk

LOG50_2.BAS	Example Program (in BASIC). Shows acquisition of 60 samples from 2 channels in Trace-About-Trigger mode, data logged sequentially .
PLOT50.BAS	BASIC example program showing collection of data and quick plot to screen.
CLOG50_1.C	Source Code for *LOG50_1* written in C.
PLOG50_1.PAS	Source Code for *LOG50_1* written in PASCAL.
FLOG50_1.FOR	Source Code for *LOG50_1* written in FORTRAN.
PCIPMOD.EXE	Utility for modifying instrument device driver file if multiple DAS-50'S are to be used in one PC.
LS.COM	Loads PCIP device drivers from DOS.
US.COM	Unloads PCIP device drivers that were loaded using LS.COM.
VIKEY.COM	Keyboard conversion for foreign keyboards.
VIKEY.DOC	Guide to installation and use of *VIKEY.COM*

1.4 SPECIFICATIONS

INPUT

Channels:	4 single-ended (selectable for sequence)
Ranges:	0 to ± 5V, 0 to ±10V, ± 2.5V, ± 5V, ± 10V
Maximum Input:	± 25V
Resistance:	100 K Ohms
Capacitance:	10 pF
Bandwidth (-3 dB):	6 MHz (BW flat within 0.5, DC to 500 KHz)
Channel to Channel: (Signal Isolation)	62 dB (500 KHz Analog Input)

CONVERTER

Resolution:	12 bits
Conversion Rate:	1 µsecond

ACCURACY

Overall Error:	2 LSB max.
Differential Linearity Error:	1 LSB max.
Integral Linearity Error:	1.5 LSB max.
No Missing Codes:	Guaranteed

POWER SUPPLY

+5V:	2 A, typ.
+12V:	80 mA, typ
Power Consumption:	11 watts typ.

ENVIRONMENTAL

Operating:	0°C to + 50 °C
Storage:	-25°C to + 85°C
Humidity:	0 to 90%, non-condensing
Weight:	19.5 oz. (585 grams)

1.5 ORDERING INFORMATION

Table 1-2 lists the part numbers for the DAS-50 and its accessories. A brief description of each is provided. Refer to the MetraByte Catalog for more information or call MetraByte's Technical Support Department. (See Chapter 7.)

Table 1-2. Ordering Information

Part Number	Description
DAS-50/1	DAS-50 High-Speed Analog to Digital Converter with 256K word memory.
DAS-50/2	DAS-50 High-Speed Analog to Digital Converter with 512K word memory.
DAS-50/4	DAS-50 High-Speed Analog to Digital Converter with 1024K word memory.
BNC-50	BNC-50 Input Terminal Box

1.6 HOW TO USE THIS MANUAL

This manual is designed for a knowledgeable computer user. It assumes that the user is familiar with his PC and its operation. At times, it may be necessary to refer to the Operator's Manual provided with the computer and a reference manual for the language used in custom application programs. Be sure to have these handy.

Chapter 2,"Installation" describes unpacking and inspection procedures, setting the base address and memory configuration switches, installation of the board, and how to connect peripherals.

Chapter 3, "Getting Started" tells you how to make copies of the DAS-50 Utility Diskette. Additionally, the chapter describes how to install the device drivers and modify your config.sys file. This chapter also describes how to use the DAS-50's Pop-Up Menu to set the operating parameters.

Chapter 4, "Operating the DAS-50", discusses the DAS-50'S Pop-Up Menu options.

Chapter 5, "Logging Data" describes the format of the file to which data is saved.

Chapter 6,"Programming", describes the Language Independent Interface commands. These commands can be integrated into programs written in any of several languages. This allows the creation of custom data collection programs. Each command's function, syntax, and usage is described. Two sample programs, written in C and BASIC, are also given. This chapter also provides a brief description of the DAS-50's registers for reference purposes.

Chapter 7, "Maintenance and Repair" gives a procedure for calibrating the DAS-50. Warranty information and Return-to-Factory instructions are also provided.

The Appendices contain other useful information. Appendix A provides the Intel 82C54 Data Sheet for programmers requiring more detailed information. Appendix B gives the error messages which apply to the Language Interface Routines.

CHAPTER 2
INSTALLATION

2.1 GENERAL

This chapter describes how to install your DAS-50. The following information is provided: unpacking and inspection procedures, setting of the base address and memory configuration switches, installation of the board, and system connections.

If you have installed optional peripheral boards in your PC before, you may want to skip this chapter. Be sure, however, that the base address switch and memory configuration switch have been set according to the instructions in section 2.3. The DAS-50 is factory-calibrated and should be re-calibrated when necessary. Calibration procedures are given in section 7.2.

2.2 UNPACKING AND INSPECTION

After you unpack the board from its outer shipping material:

1. Remove the item from its packing material, while placing one hand firmly on a metal portion of the system chassis. This will prevent any damage to the board components due to possible static electricity build-up during transit.

2. After allowing a moment for static electricity discharge, carefully unwrap the board from the anti-static material.

3. Inspect the board for any possible damage. If any sign of damage is detected, return the board to the factory as described in section 7.4.

Confirm that each item on the packing list has been shipped. It is a good idea to retain the packing material in the event that the board must be returned to the factory for repair.

2.3 SWITCHES

This section describes the DIP switches on the DAS-50 board. There are two switches: the Base Address Switch and the Memory Configuration Switch. The location of each of these is shown in Figure 2-1.

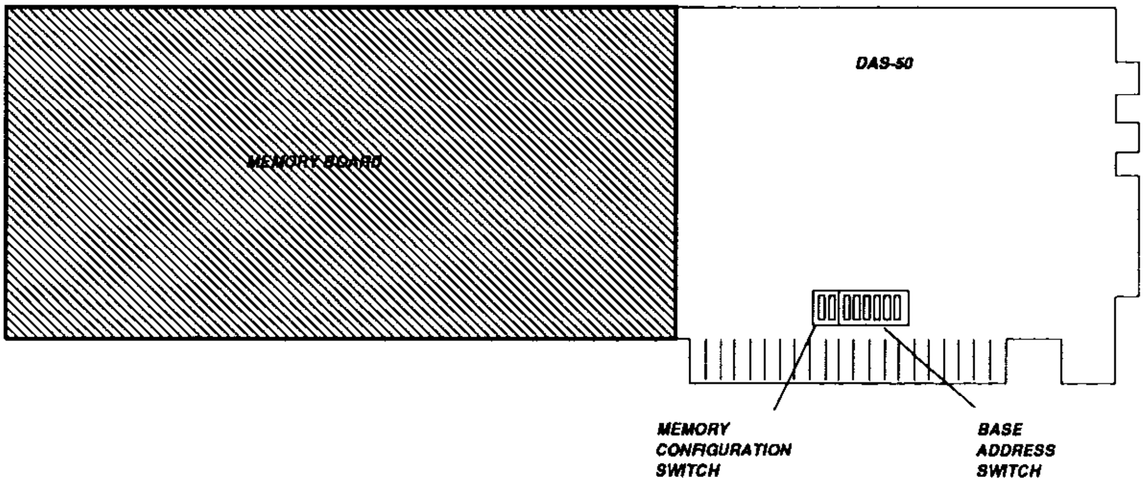


Figure 2-1. Location of Switches

Base Address Switch

The DAS-50 uses a block of sixteen non-overlapping I/O addresses. The Base Address Switch is used to set the Base Address of the Board. The DAS-50 has a default base address setting of 310 (hex). This slot may be occupied within your computer. Base addresses may be assigned within the range 100 to 3FF (hex). Check your system configuration and Operator's Manual for available addresses.

The Base Address Switch is a six-position, DIP switch located in the lower right-hand corner of the board. (See Figure 2-1.) To set the appropriate base address, use a pen-tip to move the individual switches into the OFF position. For example, to set a base address of 300 hex (768 decimal), move switches 1 and 2 into the OFF position as shown in Figure 2-2.

If you are still unsure of how to set the Base Address Switch, you may find it helpful to run the DAS-50 Calibration Program, *CAL50.EXE.* (Refer to section 7.2.) The third step of the calibration procedure asks you to input the desired base address. A picture of the switch setting is then drawn on the screen and the program verifies that the board is found at the given address.

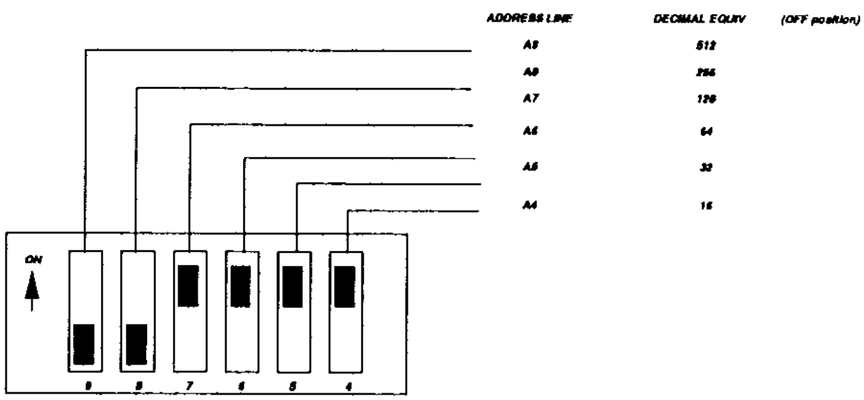
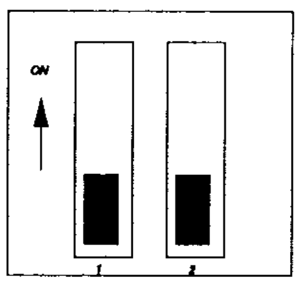


Figure 2-2. Setting the Base Address Switch

Memory Configuration Switch

The Memory Configuration Switch is a two-position, DIP switch located next to the Base Address Switch. (See Figure 2-1.) This switch is factory-set for the amount of memory installed on the board. You should only have to re-set this switch if you change the amount of memory. To set the Memory Configuration Switch, use a pen-tip to move the individual switches into the desired position. (See Figure 2-3.) For example, the switch shown in Figure 2-3 is set to 1024K of memory.



S1	S2	AMOUNT OF MEMORY	SOCKETS FILLED
ON	ON	256K	U1-12
OFF	ON	512K	U1-24
ON	OFF	768K	U1-36
OFF	OFF	1024K	U1-48

Figure 2-3. Memory Configuration Switch

2.4 BOARD INSTALLATION

This section provides general instructions for installing the DAS-50 Board. For more detailed information regarding installation of peripheral boards, consult the documentation provided with your computer.

> **WARNING**
>
> DO NOT ATTEMPT TO INSERT OR REMOVE ANY ADAPTER BOARD WITH THE COMPUTER POWER ON! THIS COULD CAUSE DAMAGE TO YOUR COMPUTER!

To install the DAS-50 Board:

1. Turn the power to the PC and to all attached options OFF.

2. Unplug the power cords of all attached options from the electrical outlets. Make a note of where all the cables and cords are attached to the rear of the system unit and disconnect.

3. Remove the cover of the PC. To do this, first remove the five cover mounting screws on the rear panel of the computer. Then, slide the cover of the computer about 3/4 of the way forward. Tilt the cover upwards to remove.

4. Choose an available option slot. Make sure that there is an additional empty slot to its right. Loosen and remove the screw at the top of the blank adapter plate. Then slide the plate up and out to remove.

5. Hold the DAS-50 in one hand. With the other hand, touch any metallic part of the PC/AT cabinet. This will safely discharge any static electricity which has built-up in your body.

6. Be sure the desired Base Address has been set as described in section 2.3.

7. Align the gold edge connector with the edge socket and the back adapter

place with the adapter plate screw. Gently press the board downward into the socket. Re-install the adapter plate screw.

8. 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. Install the mounting screws.

9. Plug in all cords and cables. Turn the power to the computer back on.

You are now ready to make any necessary system connections and install the software.

2.5 SYSTEM CONNECTIONS

All connections to the DAS-50 are made through the DB-25 connector located on the rear edge of the card. This connector is shown in Figure 2-4.

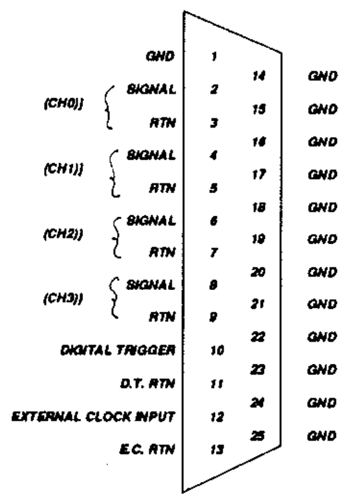


Figure 2-4. DB-25 Connector

Connections through the BNC-50 Interface Box

We recommend that MetraByte's BNC-50 Interface Box be used if multiple BNC connections are to be made. The BNC-50 provides connections for the four input channels, Digital Trigger, and External Clock.

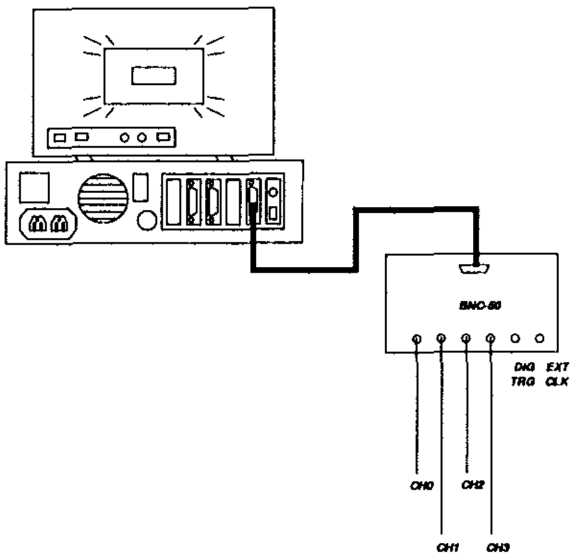


Figure 2-5. System Block Diagram Using the BNC-50

NOTE: The DAS-50 has an input resistance of 100KOhms, which facilitates measuring high impedance points. However, high impedance sources connected to the BNC-50 can become corrupted due to crosstalk on the flat ribbon cable. This can be particularly noticeable when the external clock is specified.

If this should occur, there are several solutions:

1. Try using a multi-coaxial cable instead of the flat ribbon cable.
2. Use a shorter ribbon cable.
3. Try using a low-impedance source, such as the output of a wide-bandwidth op amp, with a flat ribbon cable. This combination should show little effect of crosstalk between channels or from the external clock.

This page intentionally left blank.

CHAPTER 3
GETTING STARTED

3.1 GENERAL

Before you begin operating the DAS-50, you must:

- Make a Back-up Copy of the DAS-50 Utility Disk
- Install the DAS-50 Software
- Modify your computer's *CONFIG.SYS* File to include *ANSI.SYS*
- Install the DAS-50 Device Drivers

These tasks are described in the following sections.

3.2 MAKING A BACK-UP COPY OF THE DISKETTE

Before installing the DAS-50 Utility Program, first create a back-up copy of the DAS-50 Utility Diskette following the general procedure outlined below:

1. Turn on your computer and display.

2. Switch to the directory containing your *DISKCOPY.EXE* DOS utility.

3. At the DOS prompt, type:

   `DISKCOPY {source drive designator}: {copy drive designator}:`

4. The system will prompt you through the disk copying process. Insert the source diskette into the specified drive. Then, it will ask you to insert the target disk into the other drive. This is a blank, formatted disk which is to be your back-up disk. When the copying process has been completed, the computer will ask **`Copy another (Y/N)?`**. Enter **`N`** .

5. When your copy has been completed, put the original disk is a safe place. Label the back-up copy "DAS-50 Utility Disk Back-Up Copy". Use this disk to install the software.

3.3 INSTALLING THE DAS-50 UTILITY SOFTWARE

If at all possible, the DAS-50 software should be installed on your computer's hard drive. Installing the DAS-50 software in this manner requires the use of the DOS **COPY** command. To install the DAS-50 software:

1. Turn on your PC and its display. You should see a prompt which indicates you are at the DOS level.

2. The following instructions create a special directory for the DAS-50 Utility Disk files and assumes this directory is called **DAS50**. At the DOS prompt, change to the root directory (or other directory you want the DAS-50 directory to be branched off from), i.e. type:

   `cd \`

3. Next, create the DAS50 directory. Type:

   `mkdir \das50` OR **`md \das50`**

   Change to the das50 directory by typing:

   `cd \das50`

3. Place the DAS-50 Utility Disk into the floppy drive (assume this is drive a:) and type:

   `copy a:*.*`

You are now ready to install the driver software.

3.4 ADDING ANSI.SYS to CONFIG.SYS

The DAS-50 software requires that the *ANSI.SYS* be loaded from your *CONFIG.SYS*. Be sure to add the following instruction to your *CONFIG.SYS* file:

`DEVICE = {path}ANSI.SYS`

If the device drivers are installed from the *CONFIG.SYS* file, the *ANSI.SYS* instruction must precede the device driver instructions (See 3.5.1 and 3.5.2.).

3.5 INSTALLING THE DEVICE DRIVERS

Before operating your DAS-50, you must load its driver files (*VI.SYS* and *DAS50DRV.SYS*). This can be done in one of two ways: either by loading the drivers from your *CONFIG.SYS* or by loading them directly from DOS using the Load/Unload Option.

If you choose to load the drivers from your *CONFIG.SYS*, they will be automatically loaded every time your computer is booted up. This keeps the drivers memory-resident, allowing the DAS-50 to be used at any time. There are two ways to modify your CONFIG.SYS file: (1) Editing your CONFIG.SYS file (See section 3.5.1.) and (2) Using the DAS50DRV.SYS Setup Program (See section 3.5.2).

If unoccupied memory is limited in your computer, you will probably want to manually load and unload the drivers from the DOS command line. After the drivers have been unloaded using this technique, 500 driver file bytes will remain memory resident. We suggest creating batch files to invoke the load and unload options quickly. Section 3.5.3 discusses how to load the device drivers from the command line.

NOTE: If you have more than one DAS-50 installed in the same computer - or if you have additional MetraByte PCIP family Virtual Instruments, refer to Appendix C for Driver Installation procedure.

Syntax Conventions

The following syntax conventions are used in this section:

1. { } - Anything enclosed within curly brackets is optional. Do not include the curly brackets in the command.

2. **UPPER CASE** is used to define the abbrievated command name. The full name may be used. Command statements should be entered in UPPER CASE.

3. ***Italic print*** is used to indicate command statement parameters.

3.5.1 Editing the CONFIG.SYS File

If you do not want to run the DAS-50 Set-Up Program and are familiar with your *CONFIG.SYS* file, you may want to edit the existing *CONFIG.SYS* file with a standard word-processor which handles ASCII input. Start by bringing the *CONFIG.SYS* file into the editor. Then, after the **DEVICE = ANSI.SYS** instruction:

STEP 1. ADD THE VI.SYS DRIVER

Add the following line to the CONFIG.SYS file :

```
DEVICE = {PATH}VI.SYS {MONITOR} /HK = x  /MK = m /SK = s
```

Where:

HK designates the **Help Key**. Whenever the indicated key (x) is pressed, the help screen will be displayed. (If the DAS-50 is visible.) On-line help consists of a brief description of the DAS-50, a listing of current Key settings, and a listing of all commands which can be incorporated into your application programs.

x represents the key combination -- [Ctrl] and/or [Alt]) plus one of the following: [a] through [z], [F1] through [F10], [0] through [9], [→], [Esc], or [?]), spelled out. For example, "CTRL D" would indicate that [Ctrl] and [d] would be pressed at the same time to bring up the help screen. The default setting for the help key parameter is [Alt][H].

MK designates the **Mode Select Key**. This key combination places the visible DAS-50 into keyboard entry mode. Refer to the following section for more information.

m is the name of the key combination -- [Ctrl] and/or [Alt] plus one of the following: [a] through [z], [F1] through [F10], [0] through [9], [Tab], [Esc], or [?]. It must be spelled out, for example, "F1" assigns the function key [F1] to be the Mode Select Key. The default Mode Select Key is [Alt][M].

SK defines the **Instrument Select Key**. If you have multiple MetraByte PCIP boards installed in your computer, this key or key combination is used to toggle between their setup menus.

s is the name of the key or key combination -- Ctrl and/or Alt plus one of the following: a through z, F1 through F10, 0 through 9, →, Esc, or ? spelled out. For example, "Alt I" would indicate that Alt and I must be pressed simultaneously. The default Instrument Select Key is Alt →.

MONITOR determines which type of monitor is being used with the system. Enter **MONO** for a monochrome monitor or **COLOR** for a color monitor. COLOR is the default setting.

NOTE: If optional parameters are not specified, their default values will be used.

Example:

```
DEVICE = C:\VI.SYS /HK=ALT H /MK=ALT M /SK=ALT TAB
```

If this example line were to be placed in the *CONFIG.SYS* file, the default selections for the Help Key (Alt H), Mode Select Key (Alt M),and Instrument Select Key (Alt →) are enabled.

STEP 2. ADD THE DAS50DRV.SYS DRIVER

Add the following line to the *CONFIG.SYS* file:

```
DEVICE = {PATH}DAS50DRV.SYS /PK=p  /BA=b
```

Where:

PK designates the Pop-Up Menu Key. Whenever the indicated key or key combination is pressed, the instrument's Pop-Up Setup Menu will be displayed. Each type of instrument must be assigned a different Pop-Up Key.

p is the name of the key combination. It must be spelled out. For example, "CTRL P" would indicate that Ctrl and P must be pressed at the same time to bring up the help screen. The default setting for the Pop-Up Menu Key is Ctrl F1.

> *NOTE*
> We suggest that you avoid assigning only one key as a Pop-Up key -- unless it is a function key. Assigning one key will most likely will disable that key for use in other software applications.

BA defines the base address setting for the DAS-50. If this parameter is not given, the default base address of &H310 (784 decimal) will be used.

b is the value of the base address. Base address values may be given in either hex or decimal; however, if they are given in hex they must be preceded by an ampersand and an H (i.e., &H). Make certain that the base address you give has not been already assigned to another peripheral.

NOTE: If optional parameters are not specified, their default values will be used.

STEP 3. RE-BOOT THE PC.

Press Ctrl Alt Del. After the system start-up screens are displayed, a screen similar to Figure 3-1 should appear.

```
******* VI.SYS loaded
        o Help Key is ALT H
        o Instrument Select Key is ALT TAB
        o Mode Select Key is ALT M

******* DAS50DRV.SYS loaded
        o Pop-Up Key is CTRL F1
        o Base Address is 0310 hex
```

Figure 3-1. Start-Up Screen

3.5.2 Using D50SETUP.EXE to Modify the CONFIG.SYS File

If you prefer not to manipulate the *CONFIG.SYS* file, use the DAS-50 Setup Program. This is a self-explanatory program which is contained on the DAS-50 Utility Disk.

Before you begin to run the DAS-50, make certain that the DAS-50 and its software have been installed correctly. Make note of the Base Address Switch setting, you will need this information later.

1. Then, change to the directory where your DAS-50 software has been loaded. At the DOS prompt, type:

   D50SETUP

   The program will then prompt you for the information required by the system driver (*VI.SYS*). Follow the instructions given. Sample screen dialogue is provided in Figures 3-2 through Figure 3-8.

   First, you will be asked to specify the drive where you want the *CONFIG.SYS* file to reside. Enter the appropriate letter (i.e., C).

```
                     DAS-50 Setup Program

   Drive letter for CONFIG.SYS file : C


                     Select Boot Drive

   Enter a drive letter where new/revised CONFIG.SYS
   is to be placed.  Then hit 'ENTER'.
   Hitting 'ESC' will exit program.
```

Figure 3-2. Selecting the Boot Drive

2. You will then be asked for the path name. If you are unsure of what a path name is, refer to the operator's manual provide with your PC. Otherwise, provide the path name, being certain to include the correct drive specifier and directory name(s). See Figure 3-3.

```
                    DAS-50 Setup Program

    Drive letter for CONFIG.SYS file : C
    Full path name to VI.SYS:  C:\DAS50


             Generate the path name to VI.SYS

    The default path is .... ( C:\ )
    The path in the current CONFIG.SYS is .... ( C:\DAS50\ )

    Enter one of the following..

    o  The complete path to VI.SYS followed by 'ENTER'
    o  'ENTER' to use default path
    o  'ESC' to exit Setup
```

Figure 3-3. Generating the Path Name to VI.SYS

3. Specify the type of monitor you are using: MONO or COLOR.

DAS-50 Setup Program

```
Drive letter for CONFIG.SYS file : C
Full path name to VI.SYS:  C:\DAS50

Declare type of monitor:  COLOR
```

Specify the type of monitor in system

```
The default monitor is .... ( COLOR )
The current monitor is .... ( COLOR )

Enter one of the following..

o  The word COLOR or MONO followed by 'ENTER'
o  'ENTER' to default monitor
o  'ESC' to exit setup
```

Figure 3-4. Specifying the type of Monitor

3. Set the Help Select Key, Mode Select Key, and Instrument Select Key as illustrated in Figures 3-5 through 3-7.

   Whenever the *Help Key* is pressed and the instrument is visible, the help screen will be displayed. On-line help consists of a brief description of the DAS-50, a listing of current key settings, and a listing of all commands which can be incorporated into your application programs. The default setting for the help key is [Alt][H].

The *Mode Select Key* is used to toggle between the mouse and keyboard setup modes. Refer to the following section for more information. The default Mode Select Key is `Alt` `M`.

If you have other PCIP boards installed in your computer, the *Instrument Select Key* is used to toggle between their setup menus. The currently selected instrument's menu will be highlighted. The default Instrument Select Key is `Alt` `Tab`.

Keys may be a single function key (F1 through F10) or a combination of keys (Ctrl or Alt with another valid key). Hot keys are specified by "spelling out" the key(s) used. For example:

"Alt I" means that `Alt` and `I` are pressed simultaneously.

"F1" indicates `F1`.

"Ctrl I" is `Ctrl` and `I` pressed together.

If an invalid key or key combination is entered, the following error message will appear:

```
The Definition for the Key is not valid.
Strike any key to try another.
```

> *CAUTION*
>
> Single keystrokes may be defined as Keys; however, it is suggested that only function keys be assigned in this manner. If the *CONFIG.SYS* file resides in the root directory, assigning a single keystroke will disable that key from being used in other software applications.

```
                    DAS-50 Setup Program
    Drive letter for CONFIG.SYS file : C
    Full path name to VI.SYS:  C:\DAS50

    Declare type of monitor:  COLOR
    HELP KEY definition: ALT ?
```

```
                    Help Key Definition
    The Help Key is defined by spelling out either or both the
    legal SHIFT KEYS (CTRL and ALT) plus 1 of the following
    character keys.

            A through Z, F1 through F10, 0 through 9
                          TAB, ESC, ?

    The default HELP KEY is....( ALT H )
    The current HELP KEY is....( ALT ? )

    Enter one of the following..

    o  The definition for Help Key followed by 'ENTER'
    o  'ENTER' to use default HELP KEY
    o  'ESC' to end entry
```

Figure 3-5. Defining the Help Key

```
                       DAS-50 Setup Program

    Drive letter for CONFIG.SYS file : C
    Full path name to VI.SYS:  C:\DAS50

    Declare type of monitor:  COLOR
    HELP KEY definition: ALT ?
    MODE SELECT KEY definition: F2
```

```
                    Mode Select Key Definition

    The Mode Select Key is defined by spelling out either or
    both the legal
    SHIFT KEYS (CTRL and ALT) plus 1 of the following character
    keys.

            A through Z, F1 through F10, 0 through 9
                           TAB, ESC, ?

    The default MODE SELECT KEY is....( ALT M )
    The current MODE SELECT KEY is....( F2 )

    Enter one of the following..

    o  The definition for Mode Select Key followed by 'ENTER'
    o  'ENTER' to use default HELP KEY
    o  'ESC' to end entry
```

Figure 3-6. Defining the Mode Select Key

```
                    DAS-50 Setup Program

    Drive letter for CONFIG.SYS file : C
    Full path name to VI.SYS:  C:\DAS50

    Declare type of monitor:  COLOR
    HELP KEY definition: ALT ?
    MODE SELECT KEY definition: F2
    INSTRUMENT SELECT KEY definition: CTRL I
```

```
                Instrument Select Key Definition

The Instrument Select Key is defined by spelling out either or
both the legal SHIFT KEYS (CTRL and ALT) plus 1 of the follow-
ing character keys.

          A through Z, F1 through F10, 0 through 9
                        TAB, ESC, ?

    The default INSTRUMENT SELECT KEY is....( ALT TAB )
    The current INSTRUMENT SELECT KEY is....( CTRL I )

    Enter one of the following..

    o  The definition for INSTRUMENT SELECT KEY followed by
    'ENTER'
    o  'ENTER' to use default INSTRUMENT SELECT KEY
    o  'ESC' to end entry
```

Figure 3-7. Defining the Instrument Select Key

4. You will now be asked to confirm your selections. (See Figure 3-8.) If you wish to change an entry, press N and the program will return to the Boot Drive Selection Screen. To return to DOS, press Esc. Otherwise, press ↵.

```
                 DAS-50 Setup Program

    Drive letter for CONFIG.SYS file : C
    Full path name to VI.SYS:  C:\DAS50

    Declare type of monitor:  COLOR
    HELP KEY definition: ALT ?
    MODE SELECT KEY definition: F2
    INSTRUMENT SELECT KEY definition: CTRL I


                 VI Setup CONFIRMATION

    Enter one of the following..

    o  'N' if above entries are NOT OK
    o  'ENTER' to continue
    o  'ESC' to exit setup
```

Figure 3-8. Confirming the System Setup

5. Next, the program will ask you for the information required to setup the DAS-50 driver. It first prompts you for the path name to the file *DAS50DRV.SYS*. (See Figure 3-9.)

```
                    DAS-50 Setup Program
   Full path name to VI.SYS:  C:\DAS50


              Generate the path name to DAS50DRV.SYS
   The default path is .... ( C:\ )
   The path in the current CONFIG.SYS is .... ( C:\DAS50\ )

   Enter one of the following..

   o  The complete path to DAS50DRV.SYS followed by 'ENTER'
   o  'ENTER' to use default path
   o  'Esc' to exit Setup
```

Figure 3-9. Generating the DAS50DRV.SYS Path

6. The program then asks you to select a Pop-Up Menu Key as shown in Figure 3-10. Whenever the *Pop-Up Menu Key* is pressed, the instrument's Pop-Up Setup Menu will be displayed. It is suggested that each type of instrument be assigned a different Pop-Up Key. The default setting for the Pop-Up Menu Key is [Ctrl][F2].

   Keys may be a single function key (F1 through F10) or a combination of keys (Ctrl or Alt with another valid key). Hot keys are specified by "spelling out" the key(s) used. (See Step #3 for specifics.)

```
                    DAS-50 Setup Program
    Full path name to DAS50DRV.SYS:  C:\DAS50
    POP UP MENU KEY definition: ALT P
```

```
                    POP UP Key Definition
The Pop Up Key is defined by spelling out either or both the
legal SHIFT KEYS (CTRL and ALT) plus 1 of the following charac-
ter keys.

           A through Z, F1 through F10, 0 through 9
                         TAB, ESC, ?

    The default POP UP KEY is....( CTRL F2 )
    The current POP UP KEY is....( ALT P )

    Enter one of the following..

    o  The definition for POP UP KEY followed by 'ENTER'
    o  'ENTER' to use default POP UP KEY
    o  'ESC' to end entry
```

Figure 3-10. Pop Up Key Definition

7. You now will be asked for the Base Address of the DAS-50. Base address values may be given in either hex or decimal; however, if they are given in hex they must be preceded by an ampersand and an H (i.e., &H). Make certain that the base address you give has not been already assigned to another peripheral and agrees with the Base Address Switch setting on the DAS-50.

```
                    DAS-50 Setup Program
    Full path name to DAS50DRV.SYS:  C:\DAS50
    POP UP MENU KEY definition: ALT P
    Base Address for DAS-50:  <Enter>
```

```
                D50 I/O Base Address Selection
    The Base Address can be entered in either decimal or hex.
    If entered in hex, the number must be preceded by &H
    (i.e., &H310 is the same as 784)

    The default DAS-50 Base Address is ( &H310 ).
    The current DAS-50 Base Address is ( &H310 ).

    Enter one of the following..

    o  The BASE ADDRESS to use for the DAS-50 followed by 'EN-
    TER'
    o  'ENTER' to use default BASE ADDRESS
    o  'ESC' to exit Setup
```

Figure 3-11. Selecting a Base Address

8. You will now be asked to confirm your selections. (See Figure 3-12.) If you wish to change an entry, press N and the program will return to the screen requesting a path name for the DAS-50 driver. To return to DOS, press Esc. Otherwise, press ↵.

```
                DAS-50 Setup Program
    Full path name to DAS50DRV.SYS:  C:\DAS50
    POP UP KEY definition: ALT P
    Base Address for the DAS-50:&H310
```

```
              DAS-50 Setup CONFIRMATION
    Enter one of the following..

    o  'N' if above entries are NOT OK
    o  'ENTER' to continue
    o  'ESC' to exit setup
```

Figure 3-12. Confirming the DAS-50 Setup

9. The program will back-up your present *CONFIG.SYS* file to the file *CONFIG.BAK* and write the new set-up information to the *CONFIG.SYS* File. The *CONFIG.SYS* file is displayed. An example is given in Figure 3-13. If your system already has a *CONFIG.BAK* file, you will be asked if it should be deleted as shown in Figure 3-14.

```
Backing old CONFIG.SYS file to CONFIG.BAK
and creating new CONFIG.SYS

DEVICE = C:\SYS\ANSI.SYS
DEVICE = C:\VI.SYS /HK=ALT H  /MK=ALT M  /SK=ALT TAB
DEVICE = C:\DAS50DRV.SYS  /PK= CTRL F2  /BA = &H310

files = 20
buffers = 10
```

Figure 3-13. Creation of the new CONFIG.SYS

```
Backing old CONFIG.SYS file to CONFIG.BAK
and creating new CONFIG.SYS

Backup file already exists
Delete it? (Y/N)


DEVICE = C:\SYS\ANSI.SYS
DEVICE = C:\VI.SYS /HK=ALT H  /MK=ALT M  /SK=ALT TAB
DEVICE = C:\DAS50DRV.SYS  /PK= CTRL F2  /BA = &H310

files = 20
buffers = 10
```

Figure 3-14. CONFIG.BAK Already Exists

10. In order to activate the *CONFIG.SYS* file that was created, press Ctrl Alt Del simultaneously to re-boot the system.

3.5.3 Loading/Unloading the Drivers from DOS

The *VI.SYS* and *DAS50DRV.SYS* drivers can be "manually" loaded from the DOS command line, if desired. This is useful if you frequently run memory-resident programs. Using this technique, a maximum of 500 driver file bytes will remain memory resident once the drivers have been unloaded. You can create batch files which will alleviate the tediousness of this method. See *Notes* for specifics on batch files.

NOTE: Both of these drivers must be loaded for the DAS-50 to operate. Be sure that *ANSI.SYS* has been loaded. Refer to Appendix C if you have multiple DAS-50's or another MetraByte PCIP instrument installed.

Syntax Conventions

The following syntax conventions are used in this section:

1. { } - Anything enclosed within curly brackets is optional. Do not include the curly brackets in the command.

2. [ ] - Anything enclosed within square brackets is mandatory. Do not include the square brackets in the command.

3. **UPPER CASE** is used to define the abbrievated command name. The full name may be used. Command statements should be entered in UPPER CASE.

4. *Italic print* is used to indicate command statement parameters.

Loading the Drivers

The *LS.COM* utility lets you load the DAS-50 driver files from the DOS command line. Help is available for this utility. The help screen briefly describes the function of, syntax for, and a sample usage for *LS.COM*. To access help, type **LS** at the DOS prompt.

VI.SYS

To load the *VI.SYS* driver, type the following at the DOS prompt:

```
LS [Path\VI.SYS] {/HK = x} {/MK = m} {/SK = s}
```

Where

HK designates the **Help Key**. Whenever the indicated key (**x**) is pressed, the help screen will be displayed if the DAS-50 is visible. On-line help consists of a brief description of the DAS-50, a listing of current Key settings, and a listing of all commands which can be incorporated into your application programs.

x represents the key combination ([Ctrl] and/or [Alt] plus one of the following: [a] through [z], [F1] through [F10], [0] through [9], [→], [Esc], or [?]) spelled out. For example, "CTRL D" would indicate that [Ctrl] and [D] would be pressed at the same time to bring up the help screen. The default setting for the help key parameter is [Alt][H].

MK designates the **Mode Select Key**. This key combination places the visible DAS-50 into keyboard entry mode. Refer to the following section for more information.

m is the name of the key combination -- [Ctrl] and/or [Alt] plus one of the following: [a] through [z], [F1] through [F10], [0] through [9], [Tab], [Esc], or [?]. It must be spelled out, for example, "F1" assigns the function key [F1] to be the Mode Select Key. The default Mode Select Key is [Alt][M].

SK defines the **Instrument Select Key**. If you have multiple MetraByte PCIP boards installed in your computer, this key or key combination is used to toggle between their setup menus.

s is the name of the key or key combination -- [Ctrl] and/or [Alt] plus one of the following: [a] through [z], [F1] through [F10], [0] through [9], [→], [Esc], or [?] spelled out. For example, "Alt I" would indicate that [Alt] and [I] must be pressed simultaneously. The default Instrument Select Key is [Alt][→].

Examples

```
LS VI.SYS /HK=ALT H /MK=ALT M /SK=ALT TAB
LS VI.SYS
LS C:\PCIP\VI.SYS  /HK=?
```

DAS50DRV.SYS DRIVER

To load the *VI.SYS* driver, type the following at the DOS prompt:

```
LS [PATH/DAS50DRV.SYS] {/PK=p}  {/BA=b}
```

Where:

PK designates the Pop-Up Menu Key. Whenever the indicated key or key combination is pressed, the instrument's Pop-Up Setup Menu will be displayed. Each type of PCIP or DAS-50 instrument must be assigned a different Pop-Up Key.

p is the name of the key combination. It must be spelled out. For example, "CTRL P" would indicate that Ctrl and P must be pressed at the same time to bring up the help screen. The default setting for the Pop-Up Menu Key is Ctrl F2.

> *NOTE*
> We suggest that you avoid assigning only one key as a Pop-Up key -- unless it is a function key. Assigning one key will most likely will disable that key for use in other software applications.

BA defines the base address setting for the DAS-50. If this parameter is not given, the default base address of &H310 (784 decimal) will be used.

b is the value of the base address. Base address values may be given in either hex or decimal; however, if they are given in hex they must be preceded by an ampersand and an H (i.e., &H). Make certain that the base address you give has not been already assigned to another peripheral.

NOTE: If optional parameters are not specified, their default values will be used.

Examples

```
LS D:\DAS50\DAS50DRV.SYS  /PK= CTRL F2  /BA = &H310
LS C:\PCIP\DAS50DRV.SYS /BA=784
LS DAS50DRV.SYS
```

Unloading the Drivers

The *US.COM* utility lets you unload the DAS-50 driver files from the DOS command line. Each driver must be unloaded separately. **Help** is available for this utility. The help screen briefly describes the function of, syntax for, and a sample usage for *US.COM*. To access help, type **US** at the DOS prompt.

To unload a driver, at the DOS prompt, type:

```
US [PATH/DRIVER FILENAME]
```

Where:

DRIVER FILENAME	is *VI.SYS* or *DAS50DRV.SYS*.

Examples

```
US D:\PCIP\VI.SYS
US C:\DAS50\DAS50DRV.SYS
```

Notes

- *VI.SYS* loads into memory in two portions: a resident portion and a transient portion. The resident portion occupies about 500 bytes of memory and stays resident until the computer is rebooted or turned OFF. *VI.SYS* should be uninstalled after *DAS50DRV.SYS*.

- When loading *VI.SYS* from a batch file, load it twice, i.e.:

```
LS {path}VI.SYS
LS {path}VI.SYS
. . .
```

  This is necessary because *VI.SYS* loads in two portions. Normally, this occurs automatically; however batch files do not allow this.

- Load *VI.SYS* as soon as possible after power-up and then unload it if you do not need it. This action locates the resident portion as low as possible in memory, avoiding the creation of memory "holes". (DOS has no mechanism for recovering fragmented memory.)

- When *LS.COM* loads a *.SYS* file, it displays an **`Amount of Memory`** statement. The amount indicated may exceed the actual *.SYS* file size, because additional memory is allocated to the *.SYS* file as "working" space.

- In some instances, the *LS.COM* utility can be used to load non-MetraByte drivers - for example *MSMOUSE.SYS*. However, in general, MetraByte does not suggest this practice. Not all non-MetraByte drivers work and those that do cannot be unloaded without disrupting certain interrupt vectors. If you choose to experiment with this procedure, the *VI.SYS* driver must be loaded first.

- It is important that you specify the path to the driver files when calling *LS.COM* from the command line. If no path is specified, *LS.COM* will search the current directory and then the paths listed in the *CONFIG.SYS* file (using the **`PATH =`** command). *LS.COM* will load the first file it finds in the specified path, displaying the filename and path as well as the load destination.

- If two instrument drivers are installed, you must have two boards installed. DO NOT install two drivers at the same base address.

CHAPTER 4
OPERATING THE DAS-50

4.1 GENERAL

The DAS-50 can be programmed by using the Pop-Up Menu or by writing your own application programs incorporating the DAS50 driver commands. These commands are described in Chapter 6. The Pop-up Menus are convenient when you first begin to use the DAS-50. They provide an easy means to exercise all of the DAS-50's capabilities. They are also useful in suitations where writing any type of a program is difficult.

This chapter describes the Pop-Up Menu. To access this menu, press the Define Pop-Up Hot Key(s) (see Chapter 3). The default Pop-Up Hot Key sequence is [Ctrl][F2]. Once the DAS-50 Set-up Menu is popped-up, you can use a mouse or the keyboard to select the DAS-50's operating parameters.

4.2 USING A MOUSE

If your computer is equipped with a mouse, the mouse can be used to set the operating parameters of the DAS-50. Once the DAS-50's setup display is popped up, the mouse will be enabled. (Be sure to enable the mouse driver before bringing up the DAS-50's setup display.) As you move the mouse around, you wil see the rectangular cursor move. Make your selection by moving the cursor to the parameter to be changed or to select the operating mode. Then "click" the mouse.

4.3 USING THE KEYBOARD

Keyboard mode can be invoked by pressing the Mode Select Key combination (See Chapter 3.) or, if in mouse mode, by selecting the **Kybd Entry** mode option. The default Mode Select Key is [Alt][M]. When Keyboard Entry mode is selected, you change from field to field using the cursor keys([↑], [↓], [→], and [←]). The selected parameter will flash and appear in reverse type. Each parameter or value is changed individually. Once the value has been changed, you must press [↵]. Information describing the legal strokes for the current parameter will be displayed at the bottom of the menu.

4.4 ON-LINE HELP

On-line Help is always available by pressing the Help Key combination. (Refer to Chapter 3.) The programmed key combination displays in the upper left corner of the Pop-Up Menu. The default Help Key combination is [Alt][H].

4.5 THE POP-UP MENU

Before you try to access the Pop-Up Menu, ensure that the device drivers and the DAS-50 itself have been installed according to the directions given in the previous chapters. Then, press the Pop-Up Menu Key (Ctrl F2 or other programmed Pop-Up Key). Figure 4-1 shows the DAS-50's Pop-Up Menu.

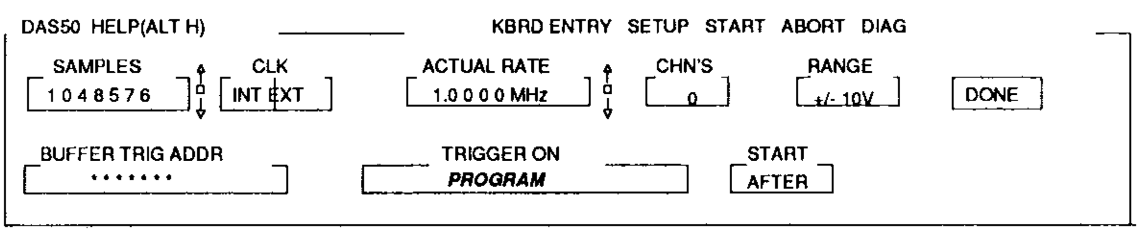


Figure 4-1. Pop-Up Menu

> *NOTE*
> At this point, the data can only be changed by using the mouse. If you do not have a mouse installed, switch to keyboard entry by pressing the *Mode Select Key* combination. (Default is Alt M.) Refer to section 3.3 for more information regarding the *Mode Select Key.*

The Pop-Up Menu is divided into two sections. The main portion of the menu contains the DAS-50's **Operating Parameters**. These are:

- **Samples**
- **Clk**
- **Actual Rate**
- **Chn's**
- **Range**
- **Trigger On**
- **Start**
- **Buffer Trig Address**

Section 4.6 describes the operating parameters and their settings.

The **Operating Modes** are accessed by selecting the options on the top line. These are discussed in section 4.7 and include:

- **KybrdEntry**
- **Acquire**
- **Abort**
- **Save**
- **Diag**

4.6 OPERATING PARAMETERS

The DAS-50's operating parameters consist of:

- **Samples**
- **Clk**
- **Actual Rate**
- **Chn's**
- **Range**
- **Trigger On**
- **Start**
- **Buffer Trig Address**

SAMPLES

This parameter sets the number of samples which are collected during a trace. The minimum number of samples which can be specified is 48. The maximum number is dependent on the amount of memory installed on the DAS-50. (i.e., 262,144; 524,288; 786,432; 1,048,576) Numbers must occur in steps of 16. If the number given is not a multiple of 16, it will be automatically rounded to the next highest. Note that the number of samples specified has no meaning if the Before Trigger Mode is selected. (Refer to "Start" parameter description.)

This parameter has an ↑, ↓, and a square button in between. (Refer to Figure 4-2.) If you are using a mouse to enter data, you can increment or decrement the current value by steps of 16. For example, to increment the number by 16, move the cursor to the ↑ and click the mouse once. The square button is used to indicate that data entry in the mouse mode has been completed. Notice that as you change the digits of the sample number, the square button will blink. When you have completed the changes, move the cursor to the square button and click the mouse once. The square button will stop blinking.

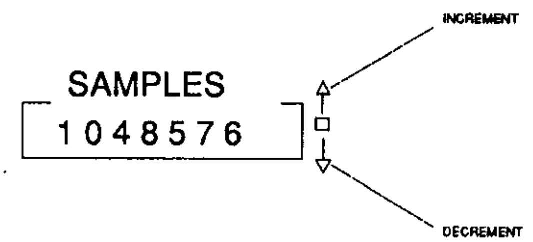


Figure 4-2. Sample Parameter

CLOCK

This parameter allows you to select the clock source. If **Int** is selected, the conversions will be initiated by the internal clock. Otherwise, an external TTL clock will initiate conversions. If the internal clock is specified, a clock rate must also be given. Figure 4-3 provides a timing diagram for each clock type.

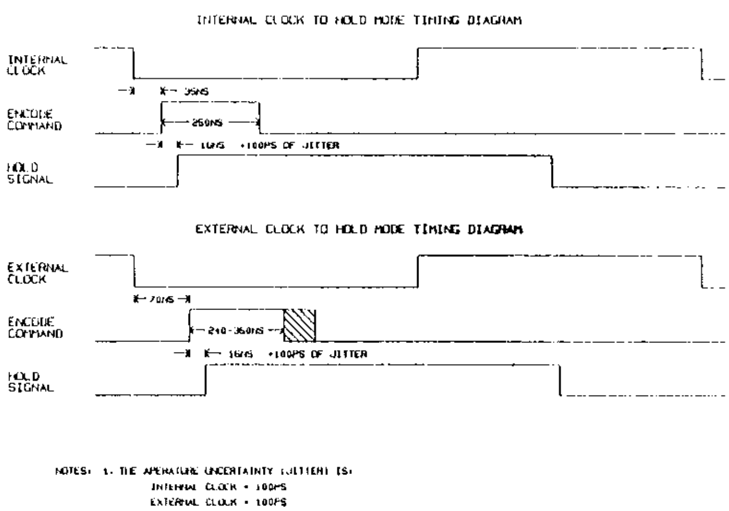


Figure 4-3. Clock Timing

ACTUAL RATE This parameter appears only if an internal clock rate is specified. The conversion rate range is 1 MHz to 137 Hz and is determined by the formula:

Rate (Mhz) = 9/N, where N is 9 to 65535

If an invalid rate is given, the number will be automatically rounded to the nearest valid rate.

When using the mouse to input data, you can re-define the units by moving the cursor to the MHz indicator and clicking the mouse until the correct unit is reached.

This operating parameter, like the Samples parameter, also provides ↑ and ↓ to facilitate data entry. Refer to the Samples parameter description for more information.

CHN'S This parameter selects which channels are to be used. Channels can be selected in the following combinations: 0, 1, 2, 3, 0&1, 0&2, 0&3, 1&2, 1&3, 2&3, 0&1&2&3.

RANGE The range parameter selects the input voltage range The following ranges may be selected: ±2.5V, ±5V, ±10V, 0-5V, and 0-10V.

TRIGGER ON The Trigger On parameter allows you to specify how a trace is to be triggered. If **Program** is selected, trace is triggered from the application program.

If **Digital Level** is chosen, an externally generated digital pulse triggers the trace. You must also specify the Level parameter (HI or LO).

When **Channel 0 Level** is selected, an analog signal on channel 0 is the trigger source. The trace is triggered on either the high or low side of a specified voltage. You must also specify the Level and Trigger > (voltage) parameters.

Digital Edge selects a digital pulse as the trace trigger. The trigger occurs on the POSitive or NEGative edge of the signal.

Channel 0 Edge, selects an analog signal on channel 0 as the trigger source. The trigger occurs when the specified voltage is reached on the given edge of the signal. When selecting this type of trigger, you must also specify the Edge and Trig At parameters.

START The start parameter lets you select when a trace will begin. As the samples are collected, the status square will change to "BUSY" and the characters will flash. Once the trace has been completed, the status block will display "DONE".

Start modes are as follows:

After - This mode starts a trace after the trigger has occurred. The specified number of samples are collected. This is the only mode which can be selected when the Trigger On mode is **Program**. The timing of this mode is illustrated in Figure 4-4.

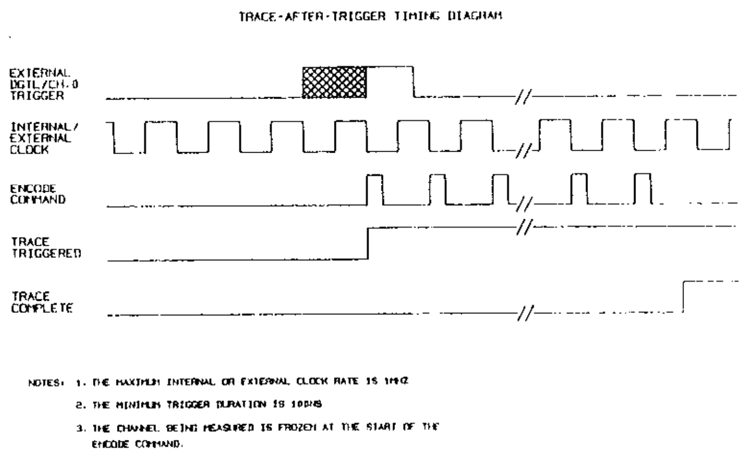


Figure 4-4. Trace After Timing Diagram

Before - When the Before mode is specified, the trace begins immediately and an unlimited number of samples is collected until the trigger occurs. When the trigger is detected, the trace stops. When in this mode, the Samples parameter does not need to be set. A timing diagram for this mode is given in Figure 4-5.

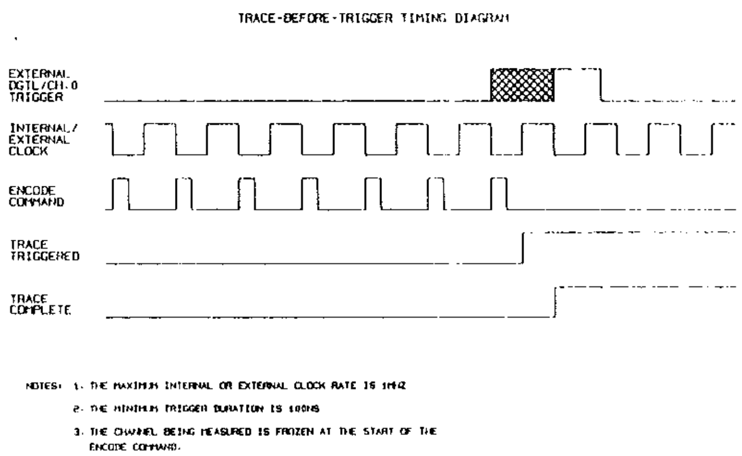


Figure 4-5. Trace Before Timing Diagram

About - In the about mode, samples are collected until a trigger signal is received. The DAS-50 then continues to take the number of samples specified by the Samples parameter. See Figure 4-6 for an illustration of the timing for this mode.

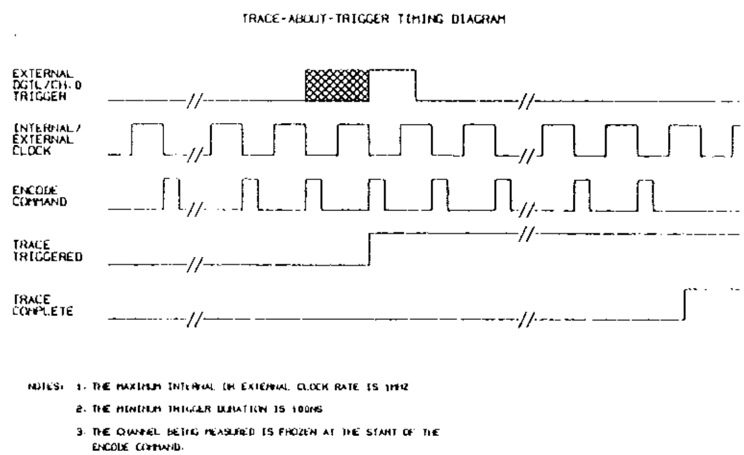


Figure 4-6. Trace About Timing Diagram

EDGE - This parameter selects the edge of the signal the trigger will occur on. A positive (POS) or negative (NEG) edge can be specified for digital and analog triggers.

(digital and analog triggers)

LEVEL - This parameter selects the level of the signal the trigger will occur on. A high (HI) or low (LO) level can be specified.

(digital and analog triggers)

TRIG AT - This parameter determines at what voltage the trigger will occur. Both positive and negative voltages can be specified. (Toggle the + or - sign by moving the cursor to the current sign and clicking the mouse.)

(analog edge triggers)

TRIG ≥ - When the voltage specified by this parameter is exceeded, the trigger will occur. Both positive and negative voltages can be specified. (Toggle the + or - sign by moving the cursor to the current sign and clicking the mouse.)

(analog level triggers)

4.7 OPERATING MODES

This section describes the operating modes. They are:

- ***Kbrd Entry***

When **Kbrd Entry** is selected, the mouse is deactivated and the keyboard is enabled. When keyboard entry is activated, the words "Kbrd Entry" will flash. Keyboard Entry only affects the Setup and Diagnostic Menus. To re-enable the mouse, press [Esc].

- ***Acquire***

This option signals to the DAS-50 to start acquiring data.

- ***Abort***

The **Abort** option allows you to stop a trace.

- ***Save***

The **Save** option access the Save menu (See Figure 4-7.), which allows you to set the various paremeters used in saving data. You can specify the file which data is to be saved to, which sample the DAS-50 is to start the save procedure at, and the mode and format used to save the data.

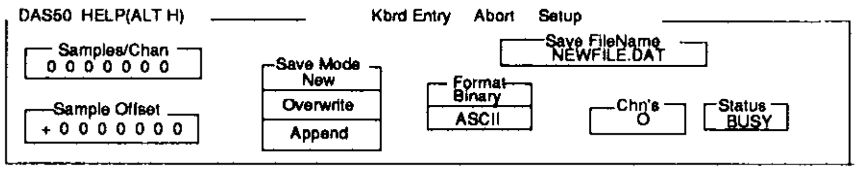


Figure 4-7. Save Menu

Samples/Channel This parameter specifies how many samples per channel are to be saved. After an acquisition takes place, the maximum number of samples taken is displayed.

Save Filename This parameter determines which file the data will be saved to. To enter a new filename, select **Save Filename** and type the new filename. Be sure to include the proper drive and path information.

See also *Save Mode.*

Sample Offset This designates the first sample to be saved relative to the trigger. This can be either a positive (+) or negative (-) value. After an acquisition has taken place, this parameter is set to the value where the first sample is taken.

Save Mode This selects how data is added to the data file. The new data can replace the contents of an existing file (*Overwrite*), be added to the end of an existing file (*Append*), or be written to a new file (*New*).

Format This parameter determines whether data is saved in *ASCII* or *Binary* form.

Chn's **Chn's** selects the channel from which data is to be saved. This can be 0, 1, 2, 3 or any combination thereof (e.g., 0 & 1, 0 & 2, 0 & 3, 1 & 2, 1 & 3, 2 & 3, 0 & 1 & 2 & 3). Note that data must have been collected for channels in order to be saved.

Status This block will display the current "save data status". This can be either BUSY or READY.

- ***Diag***

This option lets you access the DAS-50 Diagnostic Tests. DIAG is selected, the setup menu is wiped out and the Diagnostic Menu is shown. Diagnostics allows you to read/write either a 5555 hex or AAAA hex pattern to the board.

This page intentionally left blank.

CHAPTER 5
LOGGING DATA

5.1 GENERAL

All PCIP-type instruments, including the DAS-50, use the same structure for storing data to a file. In this structure, the data is preceded by a header. The header contains the details and conditions of the data recording. The data is arranged to correspond to the order of channel sampling. The data headers are readable ASCII text files. Data may be ASCII or Binary. In this chapter, the term *data set* refers to the data stored during a sampling session. A *data file* may contain one or more data sets. Each data set in the data file must have its own header.

This chapter is offered as guide to understanding and "manually" creating headers for data sets (PCIP instruments automatically generate a data set complete with header and data upon each save data command). It identifies and describes the required contents and structure for data sets. It also presents typical data set examples.

5.2 DATA-SET FILE (SAMPLE)

The following data file is a typical example of file structure and recorded data. The rules for file structure and parameters are given in the following section.

```
*****sample data file****
*
Start Global
Date:  05/15/90
Time:  10:53:31
SampleSetSize  18
RateMantissa 10000
RateExponent -10
RecordSize 4
DataMode ASCII
Instrument DAS-50
Channels 2
End
Start Channel
Channel 0
Polarity Bipolar
FSMantissa 1
FSExponent 1
```

```
DataSize 12
AnalogMSB 11
AnalogLSB 0
DigitalMSB -1
DigitalLSB -1
End
Start Channel
Channel 2
Polarity Bipolar
FSMantissa 1
FSExponent 1
DataSize 12
AnalogMSB 11
AnalogLSB 0
DigitalMSB -1
DigitalLSB -1
End
Start Command
# Set Sample=1048576
# Set Rate Int 1.0000E6
# Set Chan=0&2
# Set Range +/-10V
# Set Trigger Mode 0 Start AFTER
End
* Trigger Address : 0000000
* Sample Relative to Trigger : +0000000
Start Data
+0010,+2000
+0020,+1980
+0032,+1967
+0052,+1903
+0098,+1867
+0133,+1790
+0167,+1642
+0235,+1542
+0389,+1498
End
*
***End of file****
```

Every PCIP data set contains one *Start global* group, one or more *Start Channel* group, and a *Start Data* section. A *Start Command* group is added automatically during instrument data-saves; but is not necessary in a "manually" generated data set. It may be omitted from a "manually" generated data set header.

A data file may contain more than one data set, but each data set must have its own Global and Channel Information.

5.3 DAS-50 DATA-SET FILE STRUCTURE

PCIP data-set headers may contain up to three types of information: *Global, Channel,* and *Command.* In a data set, these information types appear under the headings of *Start Global, Start Channel,* and *Start Command.* Similarly, the actual data in the data set appears under the heading of *Start Data.* For example:

```
*****Beginning of File *****
*
Start Global
Date:
Time:
SampleSetSize
RateMantissa
RateExponent
RecordSize
DataMode
Instrument
Product
Channels
End
*
Start Channel
Channel
Polarity
FSMantissa
FSExponent
Datasize
AnalogMSB
AnalogLSB
DigitalMSB
DigitalLSB
```

```
End
*
```
Start Command
```
Instrument Commands
in effect at time of
recording.
End
*
```
Start Data
```
Data Here
End
*
******End of File****
```

Every data set contains one *Start Global* group, at least one *Start Channel* group, and a *Start Data* section. A *Start Command* group is added automatically during instrument data-saves; but is not necessary in a "manually" generated data set, it may be omitted from a "manually" generated data set. A data file may contain more than one data set and each must have its own Global and Channel information. The different sections are described below.

5.3.1 Start Global Group

`Start Global`	Indicates start of global information area.
`Date:`	Enter date as *mm/dd/yy* where *mm* is month, *dd* is day, and *yy* is year.
`Time:`	Enter time as *hh:mm:ss* where *hh* is hours, *mm* is minutes, and *ss* is seconds.
`SampleSetSize:`	Refers to number of samples in data set. Enter total number of data samples to be taken.
`RateMantissa`	Used with the RateExponent to express the sample rate (time per data point). Decimal points are not allowed; use RateExponent to adjust.

RateExponent	Use as a power of 10 with RateMantissa to express the sample rate. Example: to express a value of 25.6us, enter a RateMantissa of 256 with a RateExponent of -7.
RecordSize	Enter the number of channels sample times two (for the number of bytes in each channel sample).
DataMode	Enter *ASCII* or *BINARY* to label the mode of data-save. When *BINARY* is selected, the header is in ASCII although the data is in Binary.
Instrument	Specify the name of the PCIP Instrument presently in use, as shown in the upper-left corner of the Pop-Up menu - for example, *DAS-50*. (**NOTE:** This should be the same name assigned by the *PCIPMOD* program.)
Product	The name of the original recording instrument, such as *DAS-50*.
Channels	Enter the number of channels in the data set. There will be a separate *Start Channel* group for each specified channel.
End	Indicates the end of the Global information block.

5.3.2 START CHANNEL GROUP

Start Channel	Indicates the start of a Channel information block.
Channel	Enter the number of the channel being described (first channel is 0).
Polarity	Enter either *UNIPOLAR* or *BIPOLAR*.
DataSize	Number of bits per sample (8,12, or 16 bits). DataSize is used with FSMantissa and FSExponent to calculate the Full Scale Voltage. (See also FSExponent description below.)

FSMantissa

FSMantissa, DataSize, and FSExponent are used together to calculate the value of Full Scale Voltage. (See also FSExponent description below.)

FSExponent

FSExponent (*e*, as a power of 10), DataSize, and FSMantissa (*m*) are used to calculate the Full Scale Voltage (*FSV*) according to the following formulae:

(1) In general, *FSV* is calculated using:

$$FSV = m * 10^{e}$$

(2) The *voltage* can then be found using one of the following:

For Bipolar Data

$$Voltage = FSV * [Data/2^{(DataSize-1)}]$$

For Unipolar Data

$$Voltage = FSV * [Data/2^{(DataSize)}]$$

Example

Assume:

m = 1
e = 1
Data = 1280
DataSize = 12.

Then: $FSV = 1 * 10^{1} = 10V$.

The **Bipolar** voltage is:

$$Voltage = 10*[1280/2^{11}] = 6.25V$$

The Unipolar voltage is:

`Voltage = 10*[1280/2^12] = 3.125V`

AnalogMSB	Bit position of Most Significant Bit of analog data.*
AnalogLSB	Bit position of Least Significant Bit of analog data.*
DigitalMSB	Bit position of Most Significant bit of digital data.*
DigitalLSB	Bit position of Least Significant bit of digital data.*
End	End of Channel Information block.

******NOTES:*

(1) If negative ones (-1) are given for *AnalogMSB* and *AnalogLSB*, no analog data exists.

(2) If negative ones (-1) are given for *DigitalMSB* and *DigitalLSB*, no digital data exists.

(3) The Least Significant Bit is 0.

5.3.3 Start Command Group

A *Start Command* group is automatically added with any data-save command; however, it is not a requirement for a manually generated data set.

Start Command	Indicates the beginning of syntax code. This code contains the commands required to setup the instrument to the configuration that was in effect when data was saved. Each line must be preceded by a #.
End	End of Start Command block.

5.3.4 Start Data Section

`Start Data`	Data which follows appears in the format chosen in the Global Information block (*ASCII* or *BINARY*). For *ASCII* data, each row will be carriage-return and line-feed terminated, and represents a single sampling of data from one or more channels. In each row, data for each channel will be presented in the order sampled, and it will be separated from adjacent channel data by commas. For example, data from a 3-channel sampling will appear as: **`xx, xx, xx`**
`End`	End of Start Data block.

5.3.5 Comments

Comment entries in a data file must be preceded by an asterisk (*). Thus, in a single line of a data file, all characters between the opening asterisk and the carriage return (`Enter`) are ignored during a read of the data-set.

CHAPTER 6
PROGRAMMING

6.1 GENERAL

This chapter discusses the on-board registers and the language interface. These descriptions will be useful to programmers who intend to create custom data collection programs. Data logging must be done by means of these types of programs.

The chapter begins by describing the Independent Language Interface. A series of ASCII commands allows you to control the operation of the DAS-50 from a program written in any of several languages. Each of the commands is described in detail. The description includes a synopsis of the command's function, its syntax, and a few examples illustrating its use. Refer to Appendix B for a complete listing of error messages associated with the execution of these commands.

The chapter also contains a short procedure for incorporating the commands into a program. Two sample programs, written in C and BASIC, are given.

For those who require more specific control over the DAS-50, register descriptions are also given. It should be noted, however, that these are only an outline of the registers' functions. Refer to the Intel ® 82C54 Data Sheet in Appendix A for more information.

6.2 LANGUAGE INDEPENDENT INTERFACE

The DAS-50 is provided with a Language Independent Interface. To open the interface file, use the standard file open command for the language you are using and the filename "$DAS50". Once the device file has been opened, commands are passed to the device in ASCII using whatever print or write instructions are available with the language. The commands which are with the language interface are described below.

Examples of opening the language interface file in BASIC and C are provided below.

BASIC EXAMPLE

```
20   CLS
30   ON ERROR GOTO 340
40   OPEN "$DAS50" FOR OUTPUT AS #1
50   PRINT #1, "CLEAR"
60   OPEN "$DAS50" FOR INPUT AS #2
70   '
80   'Clear area on screen for command
90   '
100  LOCATE 11,1:PRINT"                                        "
110  LOCATE 12,1:PRINT"                                        "
120  LOCATE 13,1:PRINT"                                        "
130  LOCATE 11,1
140  '
150  INPUT "Enter command ", A$
160  '
170  'determine if it's a read command
180  C$ = MID$(A$,1,2)
190  IF (C$="re"OR C$="RE" OR C$= "re" OR C$= "RE") THEN MODE=1
     ELSE MODE=0
200  '
210  PRINT #1,A$
220  IF MODE=0 THEN GOTO 100
230  '
240  'if READ then retrieve data
250  '
260  INPUT #2,A$
270  PRINT "The Value returned was...", A$
280  PRINT "Strike any key to continue"
290  IF INKEY$="" GOTO 290
300  GOTO 100
310  '
328  'Error Handler
330  '
340  BEEP
350  PRINT "Basic Error number : ";ERR
355  REM CHECK for syntax error in DAS-50
360  IF ERR=68  OR ERR = 57 OR ERR = 75 THEN GOTO 430
370  IF ERR <> 62 GOTO 470
380  'must be error 62 ... nothing to read
390  PRINT "   Nothing to READ"
400  CLOSE 2
410  OPEN "$DAS50" FOR INPUT AS #2
```

```
420 GOTO 470
430 INPUT #2,EN:  LINE INPUT #2,A$: LINW INPUT #2,B$
440 PRINT "DAS50 Error number ";EN
450 PRINT "ERROR - "B$
460 PRINT "On command line of ...";A$
470 PRINT "Strike any key to continue"
480 IF INKEY$="" GOTO 480
490 CLS
500 RESUME 100
```

C EXAMPLE

```
FILE *VI
char                                              *ErrMess1[200],
ErrMess2;
int  ErrNum;

main();
{
if ((VI = fopen("$DAS50","r+"))==NULL)
     printf("Could not find device DAS-50");
if (!(fputs("Set SA 1234 rate 1e6 ch 0 Acquire",VI)GetError;
if (fflush(VI) = EOF) GetError;
...
...
...

}
rewind (VI);
void  GetError()
\* retrieve error number and error message from device*\
{
fscanf(VI,"%d",ErrNum):                           \* get error num-
ber\
fgets(ErrMess1,80,VI);                            \* get error
message 1\
fgets(ErrMess2,80,VI);                            \* get error mes-
sage 2\
printf("ERROR Number %d", ErrNum);
printf("ERROR - %s", ErrMess2);
printf("On command line of ...%s", ErrMess1);
rewind (VI);
}
```

Nomenclature Rules

This section presents a few nomenclature rules to bear in mind while reading the command descriptions:

(1) Anything appearing in curly brackets, (i.e, { }) is optional.
Don't enter the curly brackets.

(2) Anything appearing in square brackets indicates a mandatory choice.
The square brackets should not be entered.

(3) Uppercase defines the command's key letters which are the minimum set to be used. Upper and lower case in use do not matter.

(4) Variables denoting parameters will be in italics.

(5) Multiple commands may appear on one command line.

Commands

Table 6-1 provides a listing of all the Language Interface Commands. You may find it useful to run the *D50CMD.BAS* program provided on the Utility Disk. This provides an example of the use of each Language Interface Command.

Table 6-1. Language Interface Commands

Commands
ACQUIRE
CLEAR
HIDE
LOCK
READ BASE ADDRESS
READ CHANNELS
READ CLOCK
READ RANGE
READ RATE
READ SAMPLE
READ SAVE CHANNELS
READ SAVE CHANNELS
READ SAVE FILE
READ SAVE FORMAT
READ SAVE MODE
READ SAVE OFFSET
READ SAVE SAMPLES
READ SAVE STATUS
READ START
READ STATUS
READ TRIGGER ADDRESS
READ TRIGGER MODE
READ TRIGGER VOLTAGE
SAVE ABORT
SAVE DATA
SET ADDRESS *address*
SET CHANNELS *channel_number(s)*

Table 6-1. Language Interface Commands

Commands
SET RANGE *r*
SET RATE EXT *rate_value*
SET RATE INT *rate_value*
SET RATE *rate_value*
SET SAMPLES *s*
SET SAVE CHANNELS *channel_number(s)*
SET SAVE FILE *filename*
SET SAVE FORMAT *[BInary, AScii]*
SET SAVE MODE *[NEw, OVerwrite, APpend]*
SET SAVE OFFSET *x*
SET SAVE SAMPLES
SET START *trace_mode*
SET TRIGGER MODE *trigger_mode* {[AT,@] *trigger_voltage*}
SET TRIGGER VOLTAGE {[AT, @] *trigger_voltage*}
SET TRIGGER VOLTAGE {[AT, @] *trigger_voltage*}
SHOW
STOP
TRANSFER *buffer_segment offset size (step)*
UNLOCK

ACQUIRE

The *Acquire* command tells the DAS-50 to begin collecting data using the trigger mode, voltages, and data collection rate already specified. Refer to the *set* commands for more information.

Syntax

```
ACquire
```

Example:

```
acquire
```

Associated Error Messages

```
101 Unknown character encountered
```

CL - The Clear Command

The Clear command clears the DAS-50's device buffers. It is recommended that this command be performed after each `OPEN` of the DAS-50.

Syntax

```
CLear
```

Examples

```
cl
clear
```

Associated Error Messages

```
101 Unknown character encountered
```

HIDE

This command is used to remove an instrument's operating panel from the screen. If multiple instruments are installed, the panel appearing on the top will be hidden. Otherwise, the instrument last opened in the file, will be hidden.

Syntax

```
HIde
```

Example:

```
hide
```

Associated Error Messages

```
101 Unknown character encountered
```

LOCK

When this command is executed, the parameters in the Set-Up Menu can not be altered. It disables both the keyboard and the mouse.

Syntax

```
LOck
```

Example

```
lo
```

Associated Error Messages

```
101 Unknown character encountered
```

REad BA

The *Read Base Address* command returns the base address of the DAS-50.

Syntax

```
REad BAse
```

Example:

`read ba`	Returns the base address of the DAS-50.

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
```

REad CHannel

This command returns an integer which represents the channel selected at the last acquire.

Syntax

```
REad CHannel
```

Returns

is an integer representing the channel number. This can be a single channel, switched between any two channels, or selected to scan from channel 0 to channel 3. Integers and their corresponding Channel numbers include:

0	0
1	1
2	2
3	3
4	0&1
5	0&2
6	0&3
7	1&2
8	1&3
9	2&3
10	0&1&2&3

Examples

```
rea channel
re ch
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
105 Illegal word following Channel
```

REad CLock

This command returns an integer, representing the clock mode status (internal/external).

Syntax

```
REad CLock
```

Returns

If a *0* is returned, an internal clock is used. If a *1* is returned, the clock is external.

Examples

```
read clock

re cl
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
```

REad RANge

This command returns an integer representing the voltage range at the last acquire.

Syntax

```
REad RANge
```

Returns

The return integers and their corresponding ranges are shown below:

0	± 2.5V
1	± 5.0V
2	± 10.0V
3	0 to 5 V
4	0 to 10 V

Examples

```
rea range
re ran
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
106 Illegal word following Range
```

REad RAte

This command returns a string in floating point format, representing the sample rate at the time of the last acquisition. This command should be used with Read Clock which returns the clock mode status (internal/external).

Syntax

```
REad RAte
```

Examples

```
rea rate

re ra
```

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following READ
110 Illegal word following Rate
111 Illegal or missing Rate value
```

REad SAmples

The Read Sample Command returns a string which represents the number of samples set at the last acquisition.

Syntax

```
REad SAmples
```

Example

```
read sample

rea sa
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
116 Illegal or missing Sample count
```

REad SAVe CHannels

Returns an integer designating which channels data is written from.

Syntax

```
REad SAVe CHannels
```

Returns

n is an integer representing the channel number. This can be a single channel, switched between any two channels, or selected to scan from channel 0 to channel 3. Integers and their corresponding Channel numbers include:

0	0
1	1
2	2
3	3
4	0&1
5	0&2
6	0&3
7	1&2
8	1&3
9	2&3
10	0&1&2&3

Examples

`re sav ch` ⇒ Selects channel 3.

Associated Error Messages

```
119 Illegal read SAVE command
120 Nothing following Save command
```

REad SAVe FIle

Returns the name of the current file in which data is being saved.

Syntax

```
REad SAVe FIle
```

Example

`RE SAV FI`	⇒	Will return the name of the currently open file.

Associated Error Messages

```
119 Illegal read SAVE command
120 Nothing following SAVE command
```

REad SAVe FOrmat

Returns an integer indicating whether data is saved in ASCII or Binary format.

Syntax

```
REad SAVe FOrmat
```

Returns

0	if data is saved in Binary form.
1	if data is saved in ASCII form.

Examples

```
READ SAVE FORMAT
re sav for
```

Associated Error Messages

```
119 Illegal read SAVE command
125 Nothing following Save command
```

REad SAVe MOde

This command returns an integer which represents the mode used to save data to a data file. Data can replace the contents of an existing file (*Overwrite*), be added to the end of an existing file (*Append*), or be written to a brand-new file (*New*).

Syntax

```
REad SAVe MOde
```

Returns

0 *NEW* - Creates a new file and adds the data to it. The file can not already exist.

1 *OVERWRITE* - New data will overwrite existing data in the file.

2 *APPEND* - Adds data to the end of an existing file.

Examples

```
RE SAV MO
rea sav mod
read save mode
```

Associated Error Messages

```
119  Illegal read SAVE command
I20  Nothing following Save command
```

REad SAVe OFFSet

Returns current save offset.

Syntax

```
REad SAVe OFFSet
```

Returns

x represents the offset from the trigger address.

Example

```
RE SAV OFF
```

Associated Error Messages

```
119 Illegal read SAVE command
120 Nothing following Save command
```

REad SAVe SAMples

Returns how many samples per channel are to be saved.

Syntax

```
REad SAVe SAMples
```

Returns

s represents the number of samples to be read per channel.

Example

```
RE SAV SAM
```

Associated Error Messages

```
119 Illegal read SAVE command
120 Nothing following Save command
```

REad SAVe STatus

This command returns an integer indicating the current "Save Data" status.

Syntax

```
REad SAVe STatus
```

Returns

n is an integer which represents the current save data status, where:

-1	BUSY
0	READY
>0	DOS Error Number

Examples

```
RE SAV ST
```

Associated Error Messages

```
119 Illegal read SAVE command
120 Nothing following Save command
```

REad STatus

The *Read Status* command returns a status string indicating the current bit settings of the status register.

Syntax

```
REad STatus
```

Returns

The status string can be interpreted as shown in the table below.

Status String Interpretation

Bit #	Description
0 - 1	Memory Size. These two bits are used to indicate the amount of on-board memory which has been installed on your DAS-50. Possible bit settings are : 00 = 256K memory, 01 = 512K memory, 10 = 768K memory, and 11 = 1M memory.
2	Trigger Address Latched. This bit is set by the board following a trigger, and reset by the board upon a PC bus read of the ADC data.
3	Trace Triggered. This bit is set by the board when the trace is triggered, and reset by the board when the Start trace bit of the Trigger/Timer Control Register is reset.
4	Trace Completed. This bit is set by the board once the trace is complete, and reset by the board when the start trace bit of the Trigger/Timer Control Register is reset.
5	Trigger Level. This bit reflects the state of the voltage comparator used for external triggering. A "1" indicates the voltage on Channel 0 is higher than the voltage set by the Trigger Level DAC.
6	Buffer Full. This bit is set by the board when the Trace Buffer is full, and reset by the board when the start trace bit of the Trigger/Timer Control Register is reset.
7	Not used.

Examples

```
rea st
re stat
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
```

REad STARTMode

This command returns a string integer representing the start mode at the time of the last acquire. For a description of the various start modes, refer to the Set Start Command description.

Syntax

```
REad STARTMode trace_mode
```

Returns

trace_mode is an integer which represents the current save data status, where:

0	Before Start Mode
1	About Start Mode
2	After Start Mode

Examples

```
get startm
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
107 Illegal word following Start
```

REad TRIGgerAddress

When this command is executed, data is first read from the device. After this, the current memory address is returned.

Syntax

```
REad TRIGAddress
```

Example:

```
re triga
```

Associated Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
```

REad TriggerMode

This command returns a string integer representing the trigger mode at the time of the last acquisition.

Syntax

```
REad TRigger MOde
REad TM
```

Returns

An integer which represents the current trigger mode, where:

0	Program
1	Digital High Level
2	Digital Low Level
3	Analog High Level
4	Analog Low Level
5	Digital Positive Edge
6	Digital Negative Edge
7	Analog Positive Edge
8	Analog Negative Edge

Examples

```
rea tr MO
```

Associated Error Messages

```
101  Unknown character following Get
104  Illegal word following READ
108  Illegal word following TRIG
109  Illegal number following TM or Trig Mode
```

REad TRIGgerVoltage

This command returns a string representing a voltage between +9.99V and -9.99V; the trigger voltage set at the time of the last data acquisition.

Syntax

```
REad TRIGVoltage
```

Examples

```
re trigv
```

Associate Error Messages

```
101 Unknown character encountered
104 Illegal word following READ
```

SAVe ABort

When this command is executed, the program stops saving data to a file. A message to this effect is inserted in the data file.

Syntax

```
SAVe ABort
```

Examples

```
SAV AB
```

Associated Error Messages

```
NONE
```

SAVe DAta

When this command is executed, data is saved to a data file.

Syntax

```
SAVe DAta
```

Examples

```
SAV DA
```

Associated Error Messages

```
119 Illegal read SAVE command
121 DAS50 has never acquired data
```

{SEt} ADdress

This command sets the local address pointer on the DAS-50 to the given address. This enables the user to read data back from any portion of the on-board memory.

Syntax

```
{SEt} ADdress address
```

Where:

address	Valid addresses are between 0 and 1048575.

Examples:

`ad 45123`	⇒	Sets address pointer to address 45123.
`set address 0`	⇒	Sets address pointer to 0.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
115 Illegal or missing Set Address value
```

{SEt} CHannels

This command tells the DAS-50 which channels are to be opened. There are a total of 4 channels available for data acquisition.

Syntax

```
SEt CHannels n
```

Where

n is the channel number. The DAS-50's input may be selected as a single channel, switched between any two channels, or selected to scan from channel 0 to channel 3. Channel number combinations which may be selected include:

0	0&3
1	1&2
2	1&3
3	2&3
0&1	0&1&2&3
0&2	

Examples

`set channel 3`	⇒	Selects channel 3.
`chan 2&3`	⇒	Selects channels 2 and 3.
`ch 0&1&2&3`	⇒	Selects channels 0, 1, 2, and 3.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
105 Illegal word following Channels
```

{SEt} RANge

This command defines the voltage range.

Syntax

```
SEt RANge r
```

Where

r is the voltage range. Allowed voltage ranges are:

{+}5V	+-2.5V
{+}10V	+-5V
+-10V	

Examples

`set range +-5V`	⇒	Defines voltage range as ±5V.
`ran 10V`	⇒	Defines voltage range as 10V.
`+5V`	⇒	Defines voltage range as 0 to 5V.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
106 Illegal word following Range
```

{SEt} RAte

This command determines how frequently samples will be taken. The Set Rate command can be tied to an internal or external clock.

Syntax

`{SEt} RAte INt` *rate_value*	Internal Clock
`{SEt} RAte EXt`	External Clock

Where:

rate_value	determines how frequently the samples will be taken. Rate_values can be any value in the range 137.33 to 1,000,000 Hz. If a value above 1 MHz. Is given, it will become 1 MHz.

Examples:

`set ra 1e4`	⇒	Sets the sample rate to 10KHz.
`set rate ext`	⇒	Ties the sample rate to the external clock
`rate int 7.5e5`	⇒	Sample rate is 750 khz.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
110 Illegal word following Rate
111 Illegal or missing Rate value
```

{SEt} SAmples

This command determines how many samples will be taken.

Syntax

```
SEt SAmples s
```

Where

s	represents the number of samples to be taken. *S* can be a value anywhere between 48 and 1,048,576. The value given should occur is steps of 16. If it does not, it will be rounded up to the nearest modulo 16 value.

Examples:

samples 233008	⇒	Sets number of samples to be gathered as 23338.
sa 1E6	⇒	Defines number of samples to be taken as 1,000,000.

Associated Error Messages

```
101 Unknown character encountered
116 Illegal or missing Sample count
103 Illegal word following SET
```

{SEt} SAVe CHannels

This command designates which channels data is to be saved from.

Syntax

```
{SEt} SAVe CHannels n
```

Where

n is the channel number. This can be a single channel, switched between any two channels, or selected to scan from channel 0 to channel 3. Channel number combinations which may be selected include:

0	0&3
1	1&2
2	1&3
3	2&3
0&1	0&1&2&3
0&2	

Examples

`set save channel 3`	⇒	Selects channel 3.
`sav chan 2&3`	⇒	Selects channels 2 and 3.
`sav ch 0&1&2&3`	⇒	Selects channels 0, 1, 2, and 3.

Associated Error Messages

```
119 Illegal read SAVE command
105 Illegal word following Channels
```

{SEt} SAVe FIle

Designates a file to which data is to be saved.

Syntax

```
{SEt} SAVe FIle 'filename'
```

Where

filename is the file to which the data is to be saved. The filename must include the appropriate drive and path information.

Examples

`SE SAV FI 'c:\junk\data.log'`	⇒	Designates that data is to be saved into the *DATA.LOG* file located in the directory *junk* on the C drive.

Associated Error Messages

```
119 Illegal read SAVE command
124 Illegal Save Filespec
```

{SEt} SAVe FOrmat

Determines whether data is saved in ASCII or Binary format.

Syntax

```
{SEt} SAVe FOrmat[BInary,AScii]
```

Where

BInary	indicates that data is to be saved in Binary form.
AScii	indicates data is to be saved in ASCII form.

Examples

`SE SAV FO BI`	⇒	Data is saved in Binary form.
`SAVE FORMAT ASC`	⇒	Data is saved in ASCII form.

Associated Error Messages

```
119 Illegal read SAVE command
125 Illegal word after Save Format
```

{SEt} SAVe MOde

This command selects how data is added to a data file. The new data can replace the contents of an existing file (*Overwrite*), be added to the end of an existing file (*Append*), or be written to a brand-new file (*New*).

Syntax

```
{SEt} SAVe MOde mode
```

Where

mode can be one of the following:

OVerwrite - New data will overwrite existing data in the file.

APpend - Adds data to the end of an existing file.

NEw - Creates a new file and adds the data to it. The file can not already exist.

Examples

`SE SAV MO NEW`	⇒	Writes data to a new file.
`sav sav mo over`	⇒	Overwrites existing data in file.
`sav sav mode ap`	⇒	Appends data to an existing file.

Associated Error Messages

```
119  Illegal read SAVE command
I23  Illegal Save Mode
```

{SEt} SAVe OFFSet

Designates the first sample to be saved relative to the trigger point.

Syntax

```
{SEt} SAVe OFFSet x
```

Where

x represents the first sample to be saved in relation to the trigger.

Example

```
SE SAV OFF 3
```

Associated Error Messages

```
119 Illegal read SAVE command
127 Illegal offset relative to trigger address
```

{SEt} SAVe SAMples

Designates how many samples per channel are to be saved.

Syntax

```
{SEt} SAVe SAMples s
```

Where

s represents the number of samples to be read per channel.

Example

```
SE SAV SAM 52
```

Associated Error Messages

```
119 Illegal read SAVE command
126 Illegal Sample Size
```

{SEt} STart

This command defines when the sampling trace will begin. Traces can be taken before, after, or before and after a trigger signal is received. Check the Help Menu for additional valid syntax options.

Syntax

```
{SEt} STart trace_mode
```

Where

trace_mode is the trace mode performed. (NOTE: The trace modes can be selected by using either the first letter or the first two letters of the *trace_mode* name, i.e., ab for about.) Modes include:

B	*Before*. In this mode, the DAS-50 will collect unlimited samples until a trigger signal is received.
AB	*ABout*. In this mode, samples are taken until a trigger signal is received. The DAS-50 will continue to take the number of samples specified by the *Set Sample* command after the command trigger is received.
AF	*AFter*. In this mode, the number of samples specified by the *Set Sample* command is collected after a trigger is received.

Examples:

START AF	⇒	Start collecting samples after trigger is received.
ST AB	⇒	Start collecting samples before trigger is received. When trigger received, collect x samples more and stop.
SAB	⇒	Start collecting samples immediately and stop when trigger signal is received.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
107 Illegal word following Start
```

{SEt} TriggerMode

This command determines what type of signal trigger is to be used and what the trigger signal voltage is to be. A trigger can be generated by the PC internally or externally - either digital or analog on the negative or positive edge. (Analog triggers originate from channel 0.) The trigger signal's voltage can occur anywhere between -9.99V and +9.99V.

Syntax

```
{SEt} TRigger MOde trigger_mode {[AT,@] trigger_voltage}
{SEt} TM trigger_mode {[AT,@] trigger_voltage}
```

Where:

trigger_mode is a value between 0 and 4 which defines the mode of triggering to be used. Valid trigger modes are:

0 Program
1 Digital High Level
2 Digital Low Level
3 Analog High Level
4 Analog Low Level
5 Digital Positive Edge
6 Digital Negative Edge
7 Analog Positive Edge
8 Analog Negative Edge

trigger_voltage is the voltage of the trigger signal. Voltages may range between -9.99V and +9.99V.

Examples

`tr MO 0`	⇒	Sets the trigger mode as being via the user program.
`trig mode 3 @ 2.3`	⇒	Defines an analog trigger when Channel 0 is greater than 2.3V.
`tm 2`	⇒	Defines the trigger mode to be an digital trigger at a TTL "0".

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
109 Illegal word following Mode
117 No Trigger Voltage
```

{SEt} TriggerVoltage

The trigger voltage can be set separately (i.e., not included as part of the *Set Trigger* command) by using the Set Trigger Voltage command.

Syntax

```
{SEt} TRigger VOltage {[AT,@] trigger_voltage}
{SEt} TV {[AT,@]} trigger_voltage}
```

Where

trigger_voltage is the voltage of the trigger signal. Voltages may range between -9.99V and +9.99V.

Examples

trigger at -3.4 ⇒ Defines trigger voltage to be -3.4V.

Associated Error Messages

```
101 Unknown character encountered
103 Illegal word following SET
117 Illegal or missing Trig volt value
```

SHow

Displays either the Setup/Acquire Pop-Up Menu or the Save Data screen.

Syntax

```
SHow {[n]}
```

Where:

n	can be 1 or 2. Enter 1 for Setup/Acquire screen or 2 for the Save/Data screen.

Example:

`show [1]`	⇒	Displays the Setup/Acquire Screen.
`show`	⇒	Displays the SHOW1 screen.

Associated Error Messages

```
101  Unknown character encountered
```

STOp

The Stop command is used to halt data acquisition.

Syntax

```
STop
```

Example

```
stop
```

Associated Error Messages

```
101  Unknown character encountered
```

TRAnsfer

The Transfer command is used to move data from the on-board memory of the DAS-50 to an array in the computer's memory.

Syntax

```
TRAnsfer buffer_seg offset size (step)
```

Where

buffer_seg	If this is set to -1, the routine is being called from Interpretive BASIC.
offset	This is the offset of the buffer array.
size	This indicates the number of words to transfer.
step	Determines which data readings will be returned. Allowed step values are 1, 2, and 4 whose meanings are as follows: 1 - DAS-50 returns consecutive data readings. 2 - DAS-50 returns every other reading. This is useful in applications where 2 channels are being used to collect data and only 1 channel's data is to be read. 4 - Every fourth reading is returned. This is useful in applications where data is being acquired in 4 channels and you wish to store or read 1 channel's data. The default value is 1.

Example

```
TRA, -1, VARPTR(x%(0)), 20
```

Associated Error Messages

```
101  Unknown Character encountered...
102  Space Expected ...
112  Illegal or missing TRAns Segment value
113  Illegal or missing TRAns Offset value
114  Illegal or missing TRAns Size value
```

UNLock

When this command is executed, the parameters in the Set-Up Menu can be changed by using the keyboard or the mouse.

Syntax

```
UNLock
```

Example

```
Unl
unloc
```

Associated Error Messages

```
101  Unknown character encountered
```

6.3 PROGRAMMING SEQUENCE

This section provides a brief discussion of various sequences. There are two ways of programming the DAS-50: using direct access, or using the Independent Language Interface. The direct approach consists of reading and writing to the hardware. This method is not suggested. Instead, we recommend that you use the Interface routines described in the previous section. You may wish to refer to the description of the I/O map found in section 6.7. Three examples of programming sequences are provided below.

Example 1. Trace After Trigger Mode

1. Open a communication path to the DAS-50.

2. Setup the DAS-50 by writing the following values: channels, range, rate, samples, start, trigger mode (trigger voltage).

3. Start the trace by writing the Acquire command.

4. Check for the trace to be complete by using the status commands. Note that reading the status involves a write of "Get Status" followed by a read of the DAS-50 communication input buffer. See the programming examples on the Utility Disk.

5. Set the DAS-50 Memory Address Pointer to 0 by writing Address 0.

6. Move the desired number of samples to the computer's memory by writing the Transfer command.

   Note that when reading a single channel of a multi-channel trace, the address pointer must be set to the first word of that channel (address 0, 1, 2, or 3) to be read.

Example 2. Trace Before Trigger Mode

1. Open a communication path to the DAS-50.

2. Setup the DAS-50 by writing the following values: channels, range, rate start, trigger mode (trigger voltage).

3. Start the trace by writing the Acquire command.

4. Check for the Trace to be complete by using the Status command. Note that reading the status involves a write of Get Status followed by a read of the DAS-50 communication input buffer. See the example programs on DAS-50 Utility Disk.

5. Determine the Trigger Address by using the Get Trigger Address command.

6. Read the Status to see if the DAS-50 buffer is full and calculate from where you want to start reading data and set the DAS-50 Memory Address Pointer by using the Address command.

   Note that if the DAS-50 buffer is not full, the first valid data word is at address 0. If the buffer is full, the first valid data word is at the address following the trigger address.

7. Move the desired number of samples to the computer's memory by writing the Transfer command.

Example 3. Trace About Trigger

1. Open a communication path to the DAS-50.

2. Setup the DAS-50 by writing the following values: channels, range, rate, samples, start, trigger mode (trigger voltage).

3. Start the Trace by writing the Acquire command.

4. Check for the Trace to be complete by using the Status command. Note that reading the status involves a write of Get Status followed by a read of the DAS-50 communication buffer input buffer. See the programming examples on the Utility Disk.

5. Determine the Trigger Address by using the Get Trigger Address command.

6. Read the Status to see if the DAS-50 buffer is full and calculate from where you want to start reading data and set the DAS-50 Memory Address Pointer by using the Address command.

   Note that if the DAS-50 buffer is not full, the first valid data word is at address 0. If the buffer is full, the first valid data word is at the address following the trigger address plus the number of samples requested.

7. Move the desired number of samples to the computer's memory by writing the Transfer command.

6.4 ANALOG TO DIGITAL CORRESPONDENCE

The basic algorithm that converts the 12-bit digital output of the DAS-50 to its corresponding analog voltage value is as follows:

$$V = (FSR)\ ((DR/4096) - 0.5\ N))$$

where:

V is the analog voltage
FSR is the full scale range : 5, 10, 20 (volts)
DR is the Digital Output from DAS-50 registers
N is 1 for Bipolar, 0 for Unipolar

The language interface driver takes into account the bipolar/unipolar factor and the transfer equation becomes:

$$V = (FS)\ (DD/2048)$$

where:

V is the analog voltage
FS is the full scale 2.5, 5, 10 (volts)
DD is the digital output from the Language Interface Drivers

Note the difference between Full Scale Range and Full Scale in the two above equations. On a unipolar range, the two values are the same, while on a bipolar range the Full Scale Range is twice the Full Scale value.

6.5 DIRECT SCREEN I/O

The *VI.SYS* driver maintains all the instruments' visual characteristics. In doing this , the *VI.SYS* prevents any screen writes to the instrument area by intercepting and interpreting all INT 10 h calls that would result in affecting the visible instruments. Any program that uses DOS or INT 10's to display information will work with out interference to the visible instruments. If, however, a program is being used that performs direct screen I/O (i.e., writes directly to the display's memory) it is impossible to prevent the visible screen from being overwritten, one may redraw it by doing a pop-down followed by a pop-up. The instrument itself will be unaffected.

Most BASICs will have no problems, although it has been noted that some BASICs do direct screen I/O when scrolling the screen up. If this is the case, one could pop the instrument down before the scroll would occur.

MS QuickC, will not change to the user's screen unless there has been a write to the screen. If the user programs the instrument to pop-up, using the SHOW command, before any normal screen I/O has occurred, the instrument will pop-up over QuickC'S SCREEN and not the user's screen.

BASIC files Specification

If BASICA is loaded without any command line switches, there will be only 3 file permitted to be open at one time. Since the instrument require two files to be open (1 for output and 1 for input) it may be necessary for your program to have more than 3 files. To do this, enter:

`BASIC \fn` where n is the number of files (up to 16).

Microsoft C

If the DAS-50 is opened for reading and writing as below:

```
FILE *D50;
D50 = fopen( "$DAS50", "r+");
```

It is important to execute a rewind command between an input and output or an output and input command to the DAS-50. In addition, in Microsoft C, it is necessary to issue a `n fflush ()` command after the `fprint()` to ensure that the command is flushed from DOS' buffer.

6.6 REGISTER DESCRIPTIONS

Table 6-2 lists and describes the Registers and their base offsets. More detailed information regarding each register is provided in the paragraphs following the table. It should be noted that R is used to denote Read Only Registers and W is used to denote Write Only Registers.

Table 6-2. DAS-50 I/O Address Map

Base Offset	Description	Type
Base + 0	ADC/(Memory Test) Low Byte Memory Test Low Byte	R W
Base + 1	ADC/(Memory Test) High Byte Memory Test High Byte	R W
Base + 2	ADC/(Memory Test) Low/High Byte	R
Base + 3	Trigger/Timer Control Register Trigger/Timer Control Register	R W
Base + 4	Timer - Counter 0 Data	W
Base + 5	Timer - Counter 1 Data	W
Base + 6	Timer - Counter 2 Data	W
Base + 7	Timer - Control Word	W
Base + 8	Address Generator Reset Register	W
Base + 9	Trigger Address Register - LSB Address Generator Offset Register - LSB	R W
Base + 10	Trigger Address Register - MSB Address Generator Offset Register - MSB	R W
Base + 11	Trigger Address Register - MS4 Address Generator Offset Register - MS4	R W
Base + 12	Channel Control Register	W
Base + 13	Analog Input Range Control Register	W
Base + 14	Analog Voltage Trigger Level Register	W
Base + 15	Status Register	R

ADC Low Byte (Base + 0)

This is a read only register containing the four low-order bits. The register is left justified and zero-filled. This address must be read before the ADC high byte.

Memory Test Low Byte (Base + 0)

This is a read/write register. The memory test byte corresponds to the ADC low byte. The Memory Test Mode is selected by writing a 3 (decimal) to the Trig/Timer Control Register (Base + 3).

ADC High Byte (Base + 1)

This is a read only register, which contains the ADC's eight high-order bits. This address can be read consecutively to obtain 8-bit data resolution. Each read will increment the Memory Address Register.

ADC /(Memory Test) Low/High Byte (Base + 1)

This is a read/write register. This byte corresponds to the ADC high byte. The Memory Test Mode is selected by writing a 3 (decimal) to the Trig/Timer Control Register (Base + 3).

ADC Low/High Byte (Base + 2)

This is a Read Only Register. The ADC Low and High bytes can be read consecutively from this one address, starting with the Low Byte. When the second byte is read, the Memory Address Register is incremented to the next address.

Trigger/Timer Control Register (Base + 3)

The Trigger/Timer Control Register is described in the following paragraphs .

B1 and B0 **Trace Selection**. Bits B1 and B0 select the Trace Mode to be used. The Trace Mode determines when the trace is started in relation to when the trigger occurs. There are three modes which can be selected: Before, About, and After. A memory test mode can also be invoked. Settings of the bits are described in Table 6-3.

Table 6-3. Trace Selection

B1	B0	Description
0	0	Trace Before Trigger
0	1	Trace About Trigger
1	0	Trace After Trigger
1	1	Memory Test Mode

Before - In the Before trace mode, the trace is begun immediately and samples are taken until a trigger occurs. The sample number parameter does not does not affect this trace mode.

About - In the About trace mode, samples are taken until a trigger takes place. Then the specified number of samples are collected (See set samples command or the samples description in the Pop-Up menu discussion for more information.)

After - In this mode, the trace is begun after a trigger is detected. The trace continues until the number of samples specified is collected.

B2 **Trigger Internal/External.** Bit B2 selects whether the trigger signal is generated from an external or internal source. Triggers can be generated from three sources: the PC's bus, a digital TTL pulse, or the voltage level from Channel 0. If the trigger originates from the PC's bus, set this bit to 1. Otherwise, set the bit to 0.

B3 **Trigger Digital/Analog.** This value of this bit indicates whether the trigger occurs on an external digital or Channel 0 voltage input. If a digital pulse is desired, set the bit to 1. Likewise, to trigger on the Channel 0 voltage, set the bit to 0. This bit is only used if an external trigger is selected by bit 2.

B4 **Trigger Polarity Positive/Negative.** This bit is used to select whether a trigger will occur on the negative or positive going edge of the signal. If the trigger occurs on the positive edge, set the bit to 1; otherwise it should be set to 0. This bit is only used if an external trigger is selected by bit 2.

B5 **Trigger Armed/Disarmed.** This bit is used in conjunction with B6 to start a trace.

B6 **Start/Stop Trace.** The bit is set by the PC Bus to begin a trace. The PC can also reset the bit to stop a trace.

B7 **Acquisition Rate Timer Internal/External.** If the acquisition rate is to be generated according to an on-board timer, select Internal (Set the bit to 1.) If the rate is generated by an external timer, set the bit to 0 (External). *NOTE:* Refer to the description of Bit 7 in the Analog Input Range Control Register (Base + 13) for more information.

Timer - Counter 0 Data (Base + 4)

This register is only active when bit 7 of the Trigger/Timer Control Register is set for internal timing (Set to 1.). The register is used in Mode 3 (Refer to the Intel ® 82C54 Data Sheet in Appendix A for more information.)

The 9 MHz system clock rate is divided by the value of this counter. The result is an internal ADC conversion rate in the range 1 MHz to 137 Hz. For example, if the value of the counter is 11, the resultant ADC conversion rate is 818 KHz. Some example values are listed in Table 6-4.

Table 6-4. Conversion Rates

Divisor	Clock Rate
9	1 Mhz
10	900 Khz
11	818 Khz
12	750 Khz
13	692 Khz
.	
.	
.	
65535	137 Hz

Timer Counter 1 Data (Base + 5)

This counter provides a terminal count number from 1 to 65534. This register operates in Mode 0 as described in the Intel ® 82C54 Data Sheet in Appendix A. It will contain the modular count of the sample size. The driver software sets this to 16 (decimal), providing a sample count in increments of 16.

Timer Counter 2 Data (Base + 6)

This counter operates in conjunction with Timer Counter 1 to select the ADC sample size in the Trace About Trigger and Trace After Trigger modes. This counter operates in Mode 2 (See the Data Sheet in Appendix A.) to divide the sample count by 16. While counter 1 in Mode 0 provides a terminal count from 1 to 65534. This results in a modulo-16 sample count with a minimum of 48 and maximum of 1,048,576. Note that although it simplifies programming to always have Counter 2 as a divide by 16, it can be programmed for a minimum count of 2 for finer resolution of small sample sizes.

Timer Control Word (Base + 7)

Used in conjunction with the previously described timer registers, this register selects counting modes and control operation.

Address Generator Reset Register (Base + 8)

A write to this register will reset the Address Generator to "zero minus 1" prior to starting a trace. The data value is not significant.

LSB Trigger Address(RO)/Address Generator Offset Register(WO)-(Base + 9)

The LSB Trigger Address Register (RO) contains the trigger address (after the trigger but before the first ADC data read) or the present Address Generator Address (after the first

ADC read). Bit 2 of the Status Register is set when a trigger is detected. The register contains the LSB of the 20-bit address corresponding to the buffer location following the trigger. When ADC data is read, bit 2 of the status register is reset and this register will track the address generator.

The LSB Address Generator Register (WO) is loaded with the LSB of the 20-bit Offset Address to be entered into the Address Generator.

MSB Trigger Address(RO)/Address Generator Offset Register(WO) (Base + 10)

The MSB Trigger Address Register (RO) contains the trigger address (after the trigger but before the first ADC data read) or the present Address Generator Address (after the first ADC read). Bit 2 of the Status Register is set when a trigger is detected. The register contains the MSB of the 20-bit address corresponding to the buffer location following the trigger. When ADC data is read, bit 2 is reset and this register will track the address generator.

The MSB Address Offset Register (WO) is loaded with the MSB of the 20-bit Offset Address to be entered into the Address Generator.

MS4 Trigger Address(RO)/Address Generator Offset Register(WO) (Base + 11)

The MS4 Trigger Address Register (RO) contains the trigger address (after the trigger but before the first ADC data read) or the present Address Generator Address (after the first ADC read). Bit 2 of the Status Register is set when either a trigger is detected or a trace is stopped. The register contains the MS4 of the 20-bit address corresponding to the buffer location following the trigger. When ADC data is read, bit 2 is reset and this register will track the address generator.

The MS4 Address Generator Offset Register is loaded with the MS4 of the 20-bit Offset Address to be entered into the Address Generator.

Channel Control Register (Base + 12)

The channel control register is described in Figure 6-1 and Table 6-5.

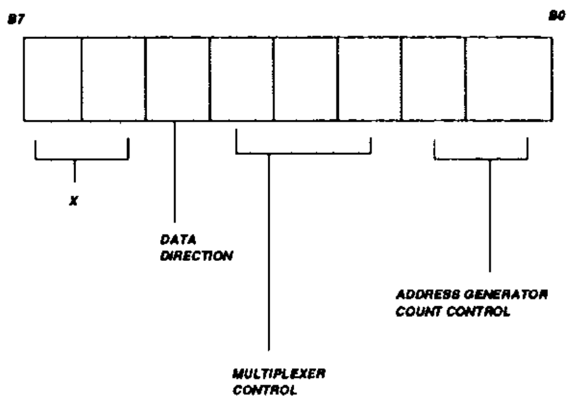


Figure 6-1. Channel Control Register

Table 6-5. Channel Control Register

Bit Settings	# of Channels	Data Memory Direction	Multiplexer Firing Order
XX000000	1	IN	0
XX000100	"	"	1
XX001000	"	"	2
XX001100	"	"	3
XX100000	"	OUT	
XX000001	2	IN	0-1
XX000101	"	"	0-2
XX001001	"	"	0-3
XX001101	"	"	1-2
XX010001	"	"	1-3
XX010101	"	"	2-3
XX100001	"	OUT	
XX000010	4	IN	0-1-2-3
XX100010	"	OUT	

Analog Input Range Control Register (Base + 13)

The contents of this register are shown in Figure 6-2.

B0 - B6 **Input Voltage Range.** The bits are set as indicated in Table 6-6 to achieve the desired INput Voltage Range.

Table 6-6. Input Voltages

Bit Settings (B0-B6)	Input Voltage Range
X0010001	-2.5 to +2.5 V
X0100010	-5 to + 5 V
X1000100	-10 to + 10 V
X0101001	0 to + 5 V
X1001010	0 to + 10 V

B7 **Trigger Edge/Level.** When this bit is set to 1, the trigger is edge-sensitive. When this bit is set to 0, the trigger is level-sensitive.

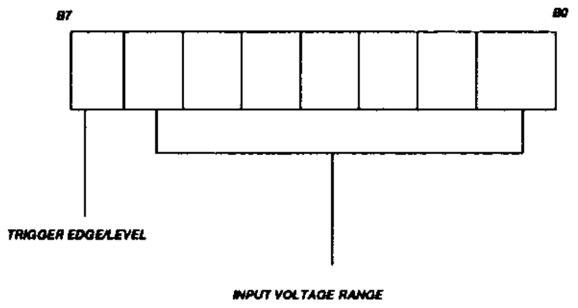


Figure 6-2. Analog Input Register

Analog Voltage Trigger Level Register (Base + 14)

This is a write-only register. The register contains the 8-bit DAC data for the External Analog Voltage Level triggering. The DAC provides a 256-step trim resolution covering -10 to + 10 volts.

Status Register (Base + 15)

The Status Register is shown in Figure 6-3.

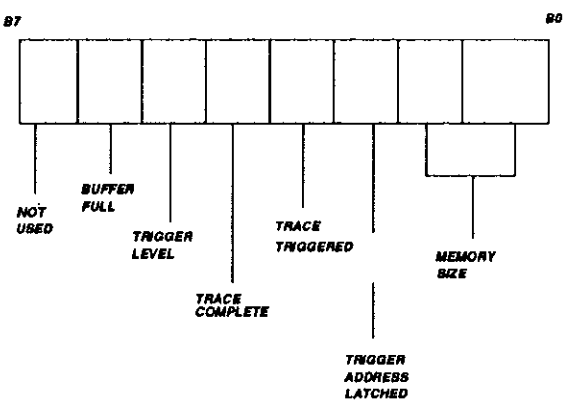


Figure 6-3. Status Register

Memory Size (Bits 0 -1) - Table 6-7 lists the settings of bits 0 and 1.

Table 6-7. Memory Size

Bit 1	Bit 0	Memory Size
0	0	256K
0	1	512K
1	0	768K
1	1	1M

Trigger Address Latched (Bit 2) - Indicator bit is set by board following a trigger, and reset by board on PC bus read of ADC data.

Trace Triggered (Bit 3) - This bit is set by the board when the trace is triggered, and reset by the board when the start trace bit of the Trigger/Timer Control Register is reset.

Trace Completed (Bit 4) - This bit is set by the board once the trace is complete, and reset by the board when the start trace bit of the Trigger/Timer Control Register is reset.

Trigger Level (Bit 5) - This bit reflects the state of the Voltage Comparator used for external triggering. A "1" indicates the voltage on Channel 0 is higher than the voltage set by the trigger level DAC.

Buffer Full (Bit 6) - This bit is set by the board when the Trace Buffer is full, and reset by the board when the Start Trace bit of the Trigger/Timer Control Register is reset.

This page intentionally left blank.

CHAPTER 7
CALIBRATION AND WARRANTY INFORMATION

7.1 GENERAL

This chapter provides instructions for calibrating the DAS-50. The return to factory procedure is also discussed in the event that your unit should require repair.

7.2 CALIBRATION

The Utility Disk contains the program *CAL50.EXE*, a user-friendly calibration program written in BASIC. The *CAL50.EXE* program leads you through each step of the calibration procedure.

1. Allow the DAS-50 to warm-up for at least 1 hour.

2. To invoke the Calibration Program, at the DOS prompt, type:

   ```
   CAL50
   ```

3. The program will respond with the screen shown in Figure 5-1. Enter the appropriate Base Address setting using the appropriate format. Only Base Addresses greater than or equal to 256 decimal (100 hex) are valid. If an invalid base address is entered, a beep sounds and a base address is prompted for again. If a hex value is given in an unrecognizable format (Valid format is **&H???**.), the following error message appears:

   ```
   ?Redo from start
   Enter the Base Address Setting
   ```

4. The program then checks to see if the Base Address Switch is appropriately

set for the given base address. If a valid base address is given, the program draws a picture of the corresponding base address switch setting and checks to see that the installed board is at the given address. If it is, this message is displayed:

```
DAS-50 is responding at selected base address-
Press 'Enter' to continue
```

If the board is not installed at the given base address, you will be prompted as follows:

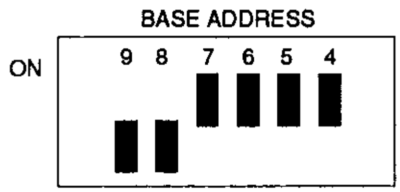
```
DAS-50 is not responding at selected base address-
Check switch setting or press 'Enter' to try again.
```

5. Once the board has been installed at the specified address, and [Enter] has been pressed, the screen shown in Figure 7-1 is displayed. Press [1] or [2] for the appropriate Diagnostic procedure. Note that you can exit from the routine at any point by pressing [Esc].

This is the DAS-50 Calibration and Memory Test Program

Enter the Base Address Switch setting (decimal or &H---):

For address 768 (Hex 300) set switch as shown



DAS-50 is responding at selected base address -
Press 'Enter to continue'

Figure 7-1. Calibration Start-Up Screen

1. DAS-50 Calibration Routine

2. DAS-50 Memory Test Routine

Select from above or 'Esc' to Exit

Figure 7-2. Calibration/Memory Test Screen

6. To begin the calibration routine, press 1 . The first screen of the calibration procedure will appear. (Refer to Figure 7-3.) Enter the appropriate channel number and follow the procedures outlined in the corresponding screens. Make certain to perform the calibration according to the instructions. Maintain the sequence because some adjustments are dependent on the others.

DAS-50 Calibration Routine 1. The only equipment that is needed to calibrate the DAS-50 is a Precision DC Voltage Reference Source, such as an Electronic Development Corp. Mode E100EC and a screwdriver. 2. Connect the DC Voltage Source to any Input Channel 3. Enter the Input Channel Number (0 is the default):

Figure 7-3. First Calibration Screen

7. To begin the memory test procedure, press ②. The Memory Test screen will appear. (Refer to Figure 7-4.) Enter the appropriate number for the memory bank to be tested. The program will then begin to read and write to the address.

```
DAS-50 Memory Test Routine

The Memory Capacity switch indicates that you have
# banks(#K) installed

If this is correct, press 'Enter' to continue

Otherwise, enter the correct number of banks installed (1-4):
```

Figure 7-4. Memory Test Screen

7.3 WARRANTY INFORMATION

All products manufactured by MetraByte 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 MetraByte, be repaired or replaced. This warranty does not apply to products damaged by improper use.

WARNING

MetraByte Corporation assumes no liability to 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.

7.4 RETURN-TO-FACTORY INFORMATION

Before returning any equipment to the factory for repair, you must first call the Technical Support Department at (508) 880-3000. They will try to diagnose and solve your problem over the phone. If they ascertain that the unit has to be returned to the factory for repair, they will issue a Return Material Authorization (RMA) number. Note that if the board is to be repaired under warranty, the Technical Support Department will need your invoice number and the date the board was purchased. Please reference the RMA number on any correspondence regarding the board.

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

1. A brief description of the problem.

2. Your name, address, and telephone number.

3. The invoice number and the date when the board was purchased.

4. Repackage the board in its original anti-static wrapping (and handle it with ground protection) and ship it back to:

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

FAX:(508)880-0179
Telephone:(508)880-3000
Telex:503989

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

This page intentionally left blank.

intel.

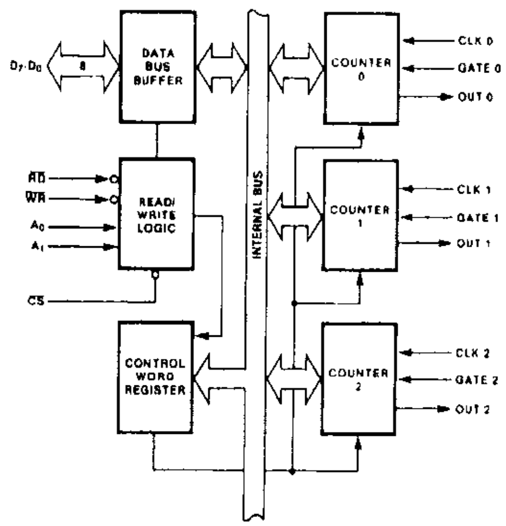
82C54
CHMOS PROGRAMMABLE INTERVAL TIMER

- **Compatible with all Intel and most other microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **Handles Inputs from DC to 8 MHz**
  - **— 10 MHz for 82C54-2**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **Three independent 16-bit counters**
- **Low Power CHMOS**
  - **— I_{CC} = 10 mA @ 8 MHz Count frequency**
- **Completely TTL Compatible**
- **Six Programmable Counter Modes**
- **Binary or BCD counting**
- **Status Read Back Command**
- **Available in 24-Pin DIP and 28-Pin PLCC**

The Intel 82C54 is a high-performance, CHMOS version of the industry standard 8254 counter/timer which is designed to solve the timing control problems common in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 82C54 is pin compatible with the HMOS 8254, and is a superset of the 8253.

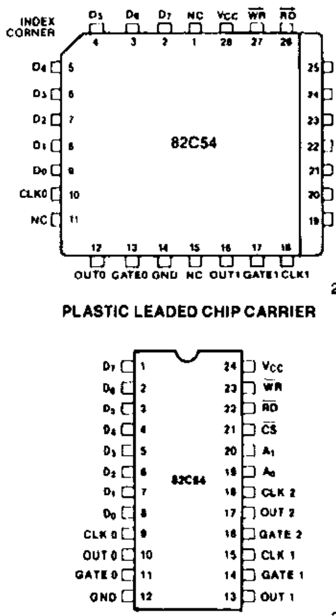
Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and in many other applications.

The 82C54 is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent HMOS product. The 82C54 is available in 24-pin DIP and 28-pin plastic leaded chip carrier (PLCC) packages.



231244-1

Figure 1. 82C54 Block Diagram



231244-3

PLASTIC LEADED CHIP CARRIER

231244-2

Diagrams are for pin reference only.
Package sizes are not to scale.

Figure 2. 82C54 Pinout

 August 1991
 Order Number: 231244-005

Table 1. Pin Description

Symbol	Pin Number DIP	Pin Number PLCC	Type	Function
D_7-D_0	1-8	2-9	I/O	Data: Bidirectional tri-state data bus lines, connected to system data bus.
CLK 0	9	10	I	Clock 0: Clock input of Counter 0.
OUT 0	10	12	O	Output 0: Output of Counter 0.
GATE 0	11	13	I	Gate 0: Gate input of Counter 0.
GND	12	14		Ground: Power supply connection.
OUT 1	13	16	O	Out 1: Output of Counter 1.
GATE 1	14	17	I	Gate 1: Gate input of Counter 1.
CLK 1	15	18	I	Clock 1: Clock Input of Counter 1.
GATE 2	16	19	I	Gate 2: Gate input of Counter 2.
OUT 2	17	20	O	Out 2: Output of Counter 2.
CLK 2	18	21	I	Clock 2: Clock input of Counter 2.
A_1, A_0	20-19	23-22	I	Address: Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus. **A_1 / A_0 / Selects** 0 / 0 / Counter 0 0 / 1 / Counter 1 1 / 0 / Counter 2 1 / 1 / Control Word Register
$\overline{CS}$	21	24	I	Chip Select: A low on this input enables the 82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
$\overline{RD}$	22	26	I	Read Control: This input is low during CPU read operations.
$\overline{WR}$	23	27	I	Write Control: This input is low during CPU write operations.
V_{CC}	24	28		Power: +5V power supply connection.
NC		1, 11, 15, 25		No Connect

FUNCTIONAL DESCRIPTION

General

The 82C54 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 82C54 are:

- Real time clock
- Even counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 3).

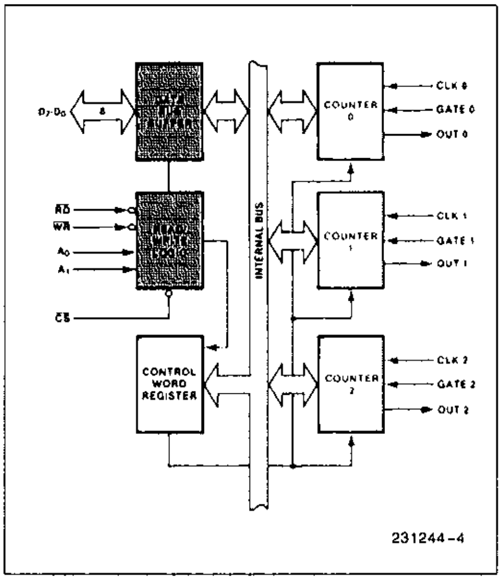


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 82C54 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 82C54 has been selected by holding $\overline{CS}$ low.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when A_1, A_0 = 11. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

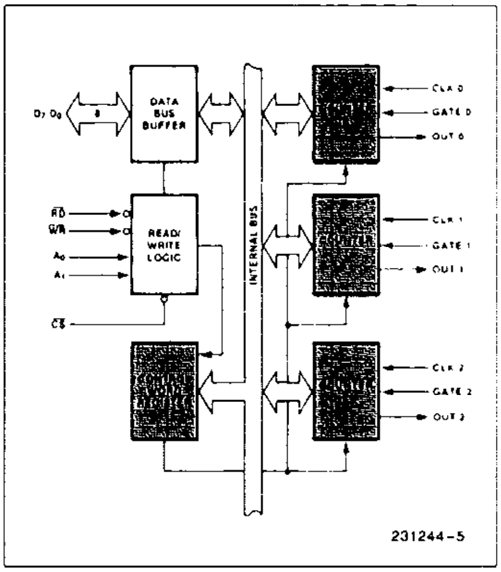


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

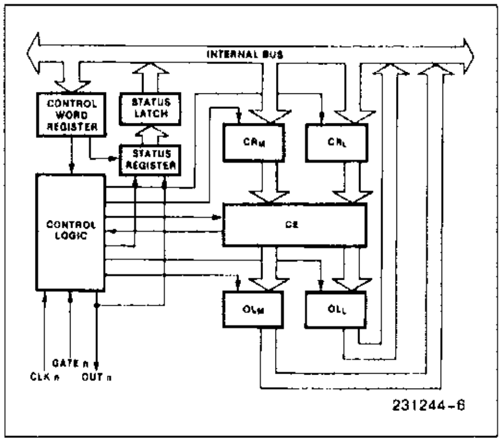


Figure 5. Internal Block Diagram of a Counter

The status register, shown in the Figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte" respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

82C54 SYSTEM INTERFACE

The 82C54 is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0, A_1 connect to the A_0, A_1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

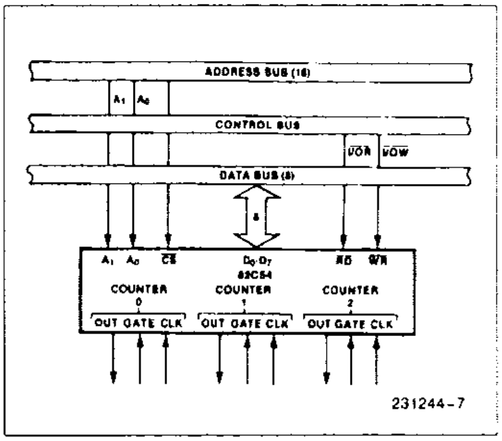


Figure 6. 82C54 System Interface

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 82C54 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 82C54

Counters are programmed by writing a Control Word and then an initial count. The control word format is shown in Figure 7.

All Control Words are written into the Control Word Register, which is selected when $A_1, A_0 = 11$. The Control Word itself specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1, A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Control Word Format

$A_1, A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW — Read/Write:

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

NOTE: Don't care bits (X) should be 0 to insure compatibility with future Intel products.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Figure 7. Control Word Format

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1, A_0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

		A_1	A_0			A_1	A_0
Control Word —	Counter 0	1	1	Control Word —	Counter 2	1	1
LSB of count —	Counter 0	0	0	Control Word —	Counter 1	1	1
MSB of count —	Counter 0	0	0	Control Word —	Counter 0	1	1
Control Word —	Counter 1	1	1	LSB of count —	Counter 2	1	0
LSB of count —	Counter 1	0	1	MSB of count —	Counter 2	1	0
MSB of count —	Counter 1	0	1	LSB of count —	Counter 1	0	1
Control Word —	Counter 2	1	1	MSB of count —	Counter 1	0	1
LSB of count —	Counter 2	1	0	LSB of count —	Counter 0	0	0
MSB of count —	Counter 2	1	0	MSB of count —	Counter 0	0	0

		A_1	A_0			A_1	A_0
Control Word —	Counter 0	1	1	Control Word —	Counter 1	1	1
Counter Word —	Counter 1	1	1	Control Word —	Counter 0	1	1
Control Word —	Counter 2	1	1	LSB of count —	Counter 1	0	1
LSB of count —	Counter 2	1	0	Control Word —	Counter 2	1	1
LSB of count —	Counter 1	0	1	LSB of count —	Counter 0	0	0
LSB of count —	Counter 0	0	0	MSB of count —	Counter 1	0	1
MSB of count —	Counter 0	0	0	LSB of count —	Counter 2	1	0
MSB of count —	Counter 1	0	1	MSB of count —	Counter 0	0	0
MSB of count —	Counter 2	1	0	MSB of count —	Counter 2	1	0

NOTE:
In all four examples, all counters are programmed to read/write two-byte counts.
These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 82C54.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A_1, A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

$A_1, A_0 = 11$; $\overline{CS} = 0$; $\overline{RD} = 1$; $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1, SC0 - specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4 - 00 designates Counter Latch Command

X - don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies; A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back command. This command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3,D2,D1 = 1.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counter(s)
D_3: 1 = Select counter 2
D_2: 1 = Select counter 1
D_1: 1 = Select counter 0
D_0: Reserved for future expansion; must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{STATUS}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D_7 1 = Out Pin is 1
 0 = Out Pin is 0
D_6 1 = Null count
 0 = Count available for reading
D_5-D_0 Counter Programmed Mode (See Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

THIS ACTION:	CAUSES:
A. Write to the control word register:[1]	Null count = 1
B. Write to the count register (CR);[2]	Null count = 1
C. New count is loaded into CE (CR → CE);	Null count = 0

[1] Only the counter specified by the control word will have its null count set to 1. Null count bits of other counters are unaffected.

[2] If the counter is programmed for two-byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{COUNT}$ and $\overline{STATUS}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

Command D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Description	Results
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the 82C54.

CLK PULSE: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

TRIGGER: a rising edge of a Counter's GATE input.

COUNTER LOADING: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

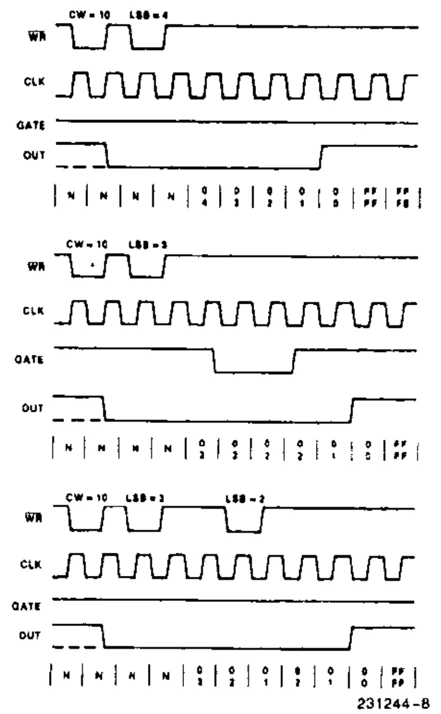
After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte does not disable counting. OUT is set low immediately (no clock pulse required).
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.



NOTE:
The Following Conventions Apply To All Mode Timing Diagrams:
1. Counters are programmed for binary (not BCD) counting and for Reading/Writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10, hex is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values.
The lower number is the least significant byte.
The upper number is the most significant byte. Since the counter is programmed to Read/Write LSB only, the most significant byte cannot be read.
N stands for an undefined count.
Vertical lines show transitions between count values.

Figure 15. Mode 0

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

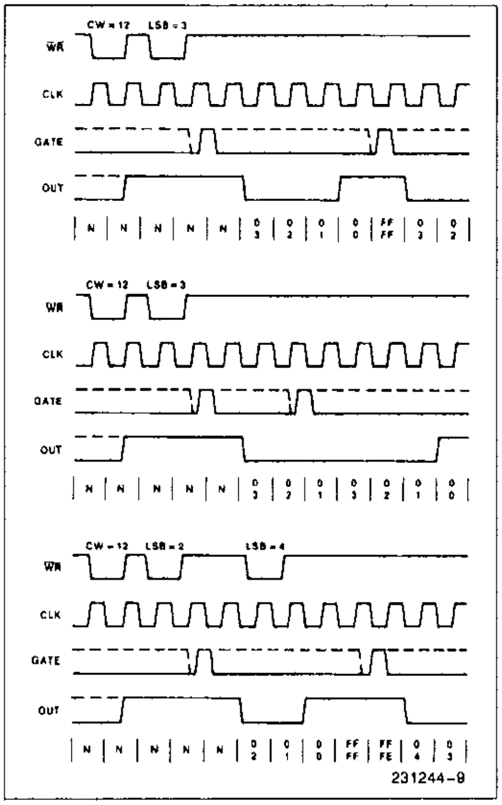


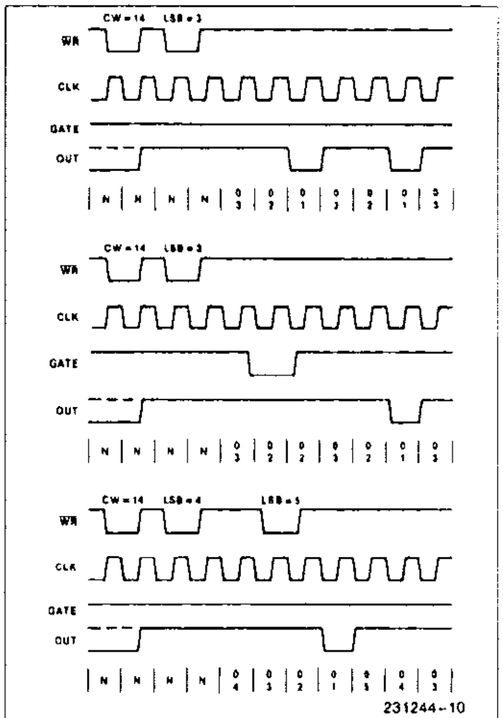
Figure 16. Mode 1

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typicially used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 17. Mode 2

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

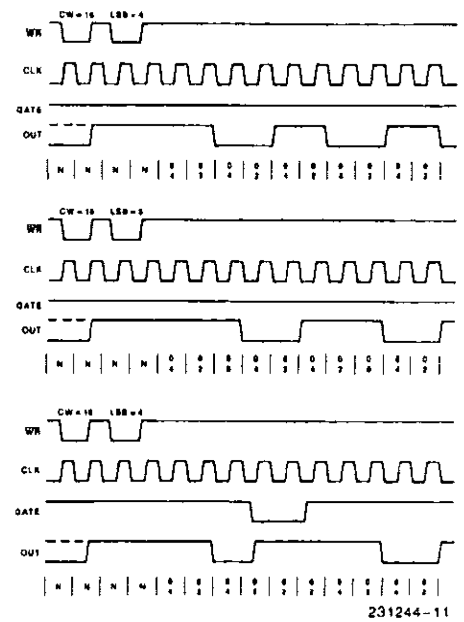
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N + 1)/2 counts and low for (N − 1)/2 counts.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N + 1 CLK pulses after the new count of N is written.

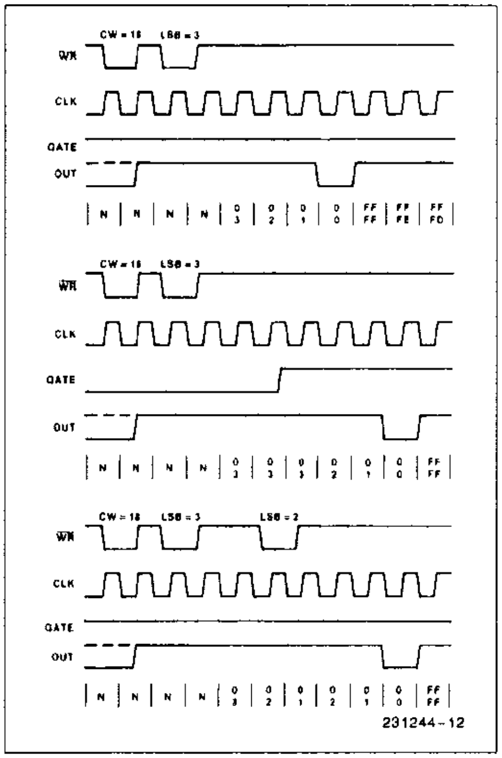


Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

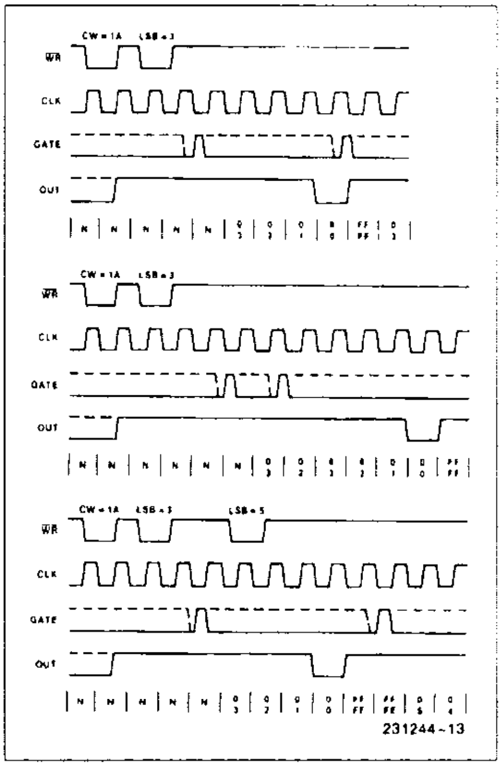


Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 21. Gate Pin Operations Summary

MODE	MIN COUNT	MAX COUNT
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting

Figure 22. Minimum and Maximum Initial Counts

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias....... 0°C to 70°C
Storage Temperature .... −65° to +150°C
Supply Voltage .... −0.5 to +8.0V
Operating Voltage .... +4V to +7V
Voltage on any Input.......... GND −2V to +6.5V
Voltage on any Output .. GND−0.5V to V_{CC} + 0.5V
Power Dissipation .... 1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

***WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, GND = 0V) (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 μA
I_{IL}	Input Load Current		± 2.0	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage Current		± 10	μA	V_{OUT} = V_{CC} to 0.0V
I_{CC}	V_{CC} Supply Current		20	mA	Clk Freq = 8MHz 82C54 10MHz 82C54-2
I_{CCSB}	V_{CC} Supply Current-Standby		10	μA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Inputs/Data Bus V_{CC} All Outputs Floating
I_{CCSB1}	V_{CC} Supply Current-Standby		150	μA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Other Inputs, I/O Pins = V_{GND}, Outputs Open
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins
C_{OUT}	Output Capacitance		20	pF	returned to GND(5)

A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, GND = 0V) (T_A = −40°C to +85°C for Extended Temperature)

BUS PARAMETERS (Note 1)

READ CYCLE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	45		30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		0		ns
t_{RR}	$\overline{RD}$ Pulse Width	150		95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		120		85	ns
t_{AD}	Data Delay from Address		220		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	90	5	65	ns
t_{RV}	Command Recovery Time	200		165		ns

NOTE:
1. AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓	0		0		ns
t_{WA}	Address Hold Time After $\overline{WR}$ ↑	0		0		ns
t_{WW}	$\overline{WR}$ Pulse Width	150		95		ns
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	120		95		ns
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	0		0		ns
t_{RV}	Command Recovery Time	200		165		ns

CLOCK AND GATE

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{CLK}	Clock Period	125	DC	100	DC	ns
t_{PWH}	High Pulse Width	60(3)		30(3)		ns
t_{PWL}	Low Pulse Width	60(3)		50(3)		ns
T_R	Clock Rise Time		25		25	ns
t_F	Clock Fall Time		25		25	ns
t_{GW}	Gate Width High	50		50		ns
t_{GL}	Gate Width Low	50		50		ns
t_{GS}	Gate Setup Time to CLK ↑	50		40		ns
t_{GH}	Gate Hold Time After CLK ↑	50(2)		50(2)		ns
T_{OD}	Output Delay from CLK ↓		150		100	ns
t_{ODG}	Output Delay from Gate ↓		120		100	ns
t_{WC}	CLK Delay for Loading(4)	0	55	0	55	ns
t_{WG}	Gate Delay for Sampling(4)	−5	50	−5	40	ns
t_{WO}	OUT Delay from Mode Write		260		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	45	−40	40	ns

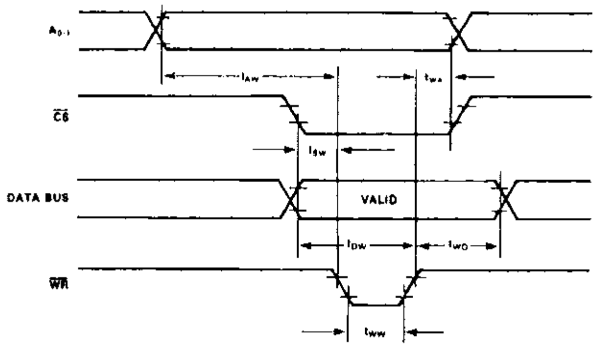
NOTES:
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 82C54-2) of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Except for Extended Temp., See Extended Temp. A.C. Characteristics below.
5. Sampled not 100% tested. $T_A = 25°C$.
6. If CLK present at T_{WC} min then Count equals N + 2 CLK pulses, T_{WC} max equals Count N + 1 CLK pulse. T_{WC} min to T_{WC} max, count will be either N + 1 or N + 2 CLK pulses.
7. In Modes 1 and 5, if GATE is present when writing a new Count value, at T_{WG} min Counter will not be triggered, at T_{WG} max Counter will be triggered.
8. If CLK present when writing a Counter Latch or ReadBack Command, at T_{CL} min CLK will be reflected in count value latched, at T_{CL} max CLK will not be reflected in the count value latched. Writing a Counter Latch or ReadBack Command between T_{CL} min and T_{WL} max will result in a latched count vallue which is ± one least significant bit.

EXTENDED TEMPERATURE (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	82C54		82C54-2		Units
		Min	Max	Min	Max	
t_{WC}	CLK Delay for Loading	−25	25	−25	25	ns
t_{WG}	Gate Delay for Sampling	−25	25	−25	25	ns

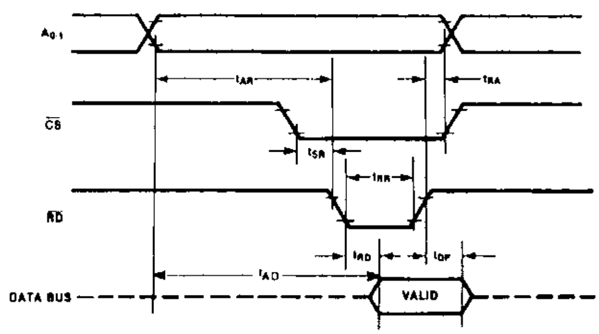
WAVEFORMS

WRITE



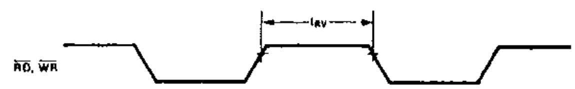
231244-14

READ



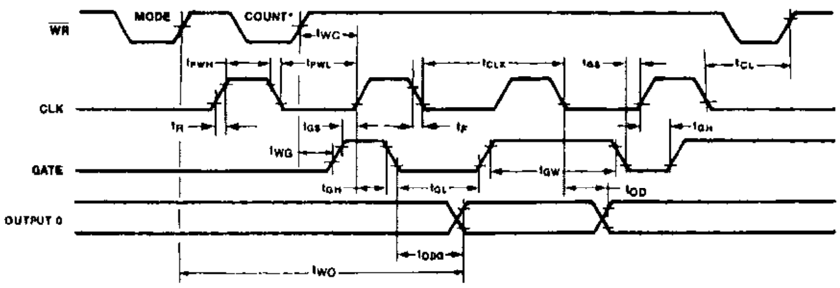
231244-15

RECOVERY



231244-16

CLOCK AND GATE



231244-17

* Last byte of count being written

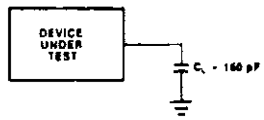
A.C. TESTING INPUT, OUTPUT WAVEFORM

INPUT/OUTPUT

231244-18

A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

A.C. TESTING LOAD CIRCUIT

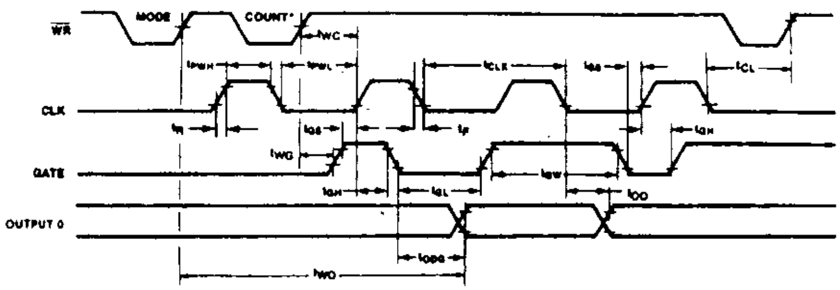


231244-19

C_L = 150 pF

C_L Includes jig capacitance

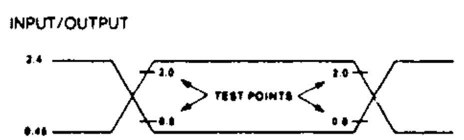
CLOCK AND GATE



231244-17

* Last byte of count being written

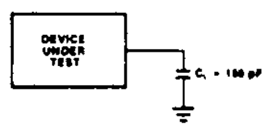
A.C. TESTING INPUT, OUTPUT WAVEFORM



231244-18

A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

A.C. TESTING LOAD CIRCUIT



231244-19

C_L = 150 pF

C_L includes jig capacitance

ERROR CODES

101	Unknown character encountered....
102	Space Expected
103	Illegal word following SET
104	Illegal word following READ
105	Illegal word following Channel
106	Illegal word following Range
107	Illegal word following STart
108	Illegal word following TRIG
109	Illegal number following TM or Trig Mode
110	Illegal word following Rate
111	Illegal or missing Rate value
112	Illegal or missing TRAns Segment value
113	Illegal or missing TRAns Offset value
114	Illegal or missing TRAns Size value
115	Illegal or missing Set Address value
116	Illegal or missing Sample count
117	Illegal or missing trig volt value
118	Illegal command when DAS-50 is BUSY
119	Illegal read SAVE command
120	Nothing following SAVE command
121	DAS50 has never acquired data
122	Too many DOS tasks
123	Illegal Save Mode
124	Illegal Save Filespec
125	Illegal word after Save Format
126	Illegal Sample Size
127	Illegal offset relative to trigger address
128	Illegal value after SHOW command
129	Attempt to read trigger address before it happened

APPENDIX C
INSTALLING MULTIPLE DAS-50's OR PCIP-TYPE INSTRUMENTS

PCIPMOD.EXE is a utility program which enables you to create unique drivers for each DAS-50 or PCIP type board in your application. In general, the *PCIPMOD.EXE* utility should be run under the following circumstances:

- If you have installed multiple DAS-50's or other type of MetraByte PCIP product in one computer.

> *NOTE*
> If two or more drivers for the same device type are to be loaded, define different Hot Keys (See Chapter 3) for each menu. This will allow you to display several Pop-Up Menus at once.

Using PCIPMOD.EXE

Table 1 names the PCIP device type supported by *PCIPMOD.EXE*:

Table 1. Devices and their SHOWNAMES

SHOWNAME	DEVICE NAME	DESCRIPTION
PCIP-AWFG	$AWFG	Arbitrary Waveform Generator
PCIP-DMM	$DMM	Digital Multimeter
DAS-50	$DAS50	High-speed Data Acquisition Interface
PCIP-CNTR	$COUNT	Frequency Counter/Timer
PCIP-CAL	$CAL	Voltage Calibrator
PCIP-SCAN	$SCAN	8/16 Channel Scanner/Multiplexer
PCIP-SCOPE	$SCOPE	Sampling Oscilloscope
PCIP-RES	$RES	Programmable Resistor Network
PCIP-SST	$SST	Sine/Square/Triangle Wave Source

To use *PCIPMOD.EXE*:

1. Change to the device directory containing *PCIPMOD.EXE*. At the DOS prompt, type:

```
PCIPMOD
```

The program will open, as follows:

```
This program will modify the ORIGINAL PCIP Driver's
1)  Name that is to be used to Open the device
2)  Name that is displayed in the upper left corner of
    the instrument when the instrument is visible

Enter the name of the ORIGINAL driver to be modified.
Please include the entire path name and drive where the
device can now be found. For example, ...
C:\xxx\DAS50DRV.SYS

ENTER DEVICE TO MODIFY
```

2. Enter the name of and path to the ORIGINAL device driver file. For example, if you have two DAS-50's and the DAS-50 driver files are loaded on the D: drive under the directory DAS50, enter:

```
d:\pcip\das50drv.sys
```

3. You will now be prompted for the new DEVICE NAME (See Table 1 for existing ones) as follows:

```
The current name used to open the device is       $DAS50
Enter the Name to be used for opening the device
```

Enter any name, up to eight characters long. You must include the $. For example, you could name the device **$MYDEV1**.

4. The program then prompts for the SHOWNAME (See Table 1 above.) of the original device:

```
The current name use for the instrument popup is      DAS-50
Enter the new name to be used.  This name can not be longer
than 7 characters
```

Enter the new SHOWNAME. (This is the name which appears in the upper-left corner of the pop-up menu.) Be sure not to exceed 7 characters. For example, you could enter **dev1**.

5. You will then be prompted for the new driver filename and its path:

```
Enter the name to be used for the New Driver.  Please
include the drive and complete path of where the driver is
to be placed.
```

Give the new filename making sure to include the correct path, i.e.: **C:\PCIP\DEVDR1.SYS**.

When the driver creation process has been completed, the message **NEW DEVICE HAS BEEN SUCCESSFULLY WRITTEN** will appear.