DAS-800 Series
Register-Level Programming
User's Guide

Revision A - October 1993
Part Number: 86890

The information contained in this manual is believed to be accurate and reliable. However, Keithley Instruments, Inc., assumes no responsibility for its use; nor for any infringements or patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of Keithley Instruments, Inc.

KEITHLEY INSTRUMENTS, INC., SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATED TO THE USE OF THIS PRODUCT. THIS PRODUCT IS NOT DESIGNED WITH COMPONENTS OF A LEVEL OF RELIABILITY THAT IS SUITED FOR USE IN LIFE SUPPORT OR CRITICAL APPLICATIONS.

Refer to your Keithley Instruments license agreement and Conditions of Sale document for specific warranty and liability information.

All brand and product names are trademarks or registered trademarks of their respective companies.

© Copyright Keithley Instruments, Inc., 1993.

All rights reserved. Reproduction or adaptation of any part of this documentation beyond that permitted by Section 117 of the 1976 United States Copyright Act without permission of the Copyright owner is unlawful.

Keithley Instruments, Data Acquisition Division

440 Myles Standish Blvd. Taunton, MA 02780

Telephone: (508) 880-3000 • FAX: (508) 880-0179

Table of Contents

List of Figures

List of Tables

Preface

The *DAS-800 Series Register I/O User's Guide* provides a description of the DAS-800 Series hardware registers.

The manual is intended for experienced programmers who want to gain direct access to the DAS-800 Series registers to perform analog input, digital I/O, or counter/timer I/O operations. It is assumed that users have read the *DAS-800 Series User's Guide* to familiarize themselves with the boards' functions, that they have completed the appropriate hardware installation and configuration, and that they are familiar with data acquisition principles.

Note: This manual is not intended for application programmers who are using the DAS-800 Function Call Driver or a menu-driven software package. If you are using the DAS-800 Function Call Driver, refer to the *DAS-800 Function Call Driver User's Guide*. If you are using a menu-driven software package, refer to the documentation supplied with the package.

The *DAS-800 Series Register I/O User's Guide* is organized as follows:

- Chapter 1 contains a description of the DAS-800 Series hardware registers.
- Chapter 2 contains guidelines you should follow when programming certain functions of DAS-800 Series boards.
- Appendix A contains a summary of the bits in the DAS-800 Series registers.

- Appendix B contains instructions for converting binary code to voltage.

Throughout the manual, keep in mind that references to DAS-800 Series boards apply to the DAS-800, DAS-801, and DAS-802 boards. When a feature applies to a particular board, that board's name is used.

1

Hardware Registers

The DAS-800 hardware registers require eight consecutive 8-bit locations. You assign the base address by setting switches on the DAS-800 Series board; refer to the *DAS-800 Series User's Guide* for more information on setting the base address. Since the base address is variable, the addresses of the individual registers are described as offsets from the selected base address.

Each register has both a write function and a read function. Table 1-1 lists the hardware registers and their write and read functions.

Table 1-1. DAS-800 Series Registers

Address	Register	Write Function	Read Function
Base + 0	A/D Conversion (LSB)	Initiates an analog-to-digital (A/D) conversion through software. Refer to page 1-4.	Contains the four least significant bits of data converted by the analog-to-digital converter (ADC); determines the status of the FIFO. Refer to page 1-5.
Base + 1	A/D Conversion (MSB)	Initiates an A/D conversion through software. Refer to page 1-4.	Contains the eight most significant bits of data converted by the ADC. Refer to page 1-5.

Table 1-1. DAS-800 Series Registers (cont.)

Address	Register	Write Function	Read Function
Base + 2	Control (Write)[1] Status 1 (Read)	Control Register 1 Specifies an analog input channel, enables interrupts, and specifies digital output values. Refer to page 1-8. Conversion Control Register Specifies the clock source, enables a digital trigger, specifies when to generate an interrupt, enables automatic channel scanning, enables a digital gate, and enables hardware conversions. Refer to page 1-11. Scan Limits Register Specifies the first and last channels used in automatic channel scanning. Refer to page 1-15.	Contains the currently selected analog input channel, the state of the digital input lines, and the status of the ADC. Refer to page 1-17.
Base + 3	Gain / Control Select (Write) Gain / Channel Status (Read)	Specifies the analog input range and the Control or Status register to access. Refer to page 1-19.	Contains the currently selected analog input range and analog input channel, and the status of automatic channel scanning. Refer to page 1-21.
Base + 4	8254 C/T0[2]	Loads counter/timer 0 (C/T0) on the 8254 counter/timer circuitry.	Contains the status of C/T0 on the 8254.
Base + 5	8254 C/T1[2]	Loads C/T1 on the 8254 counter/timer circuitry.	Contains the status of C/T1 on the 8254.

Table 1-1. DAS-800 Series Registers (cont.)

Address	Register	Write Function	Read Function
Base + 6	8254 C/T2[2]	Loads C/T2 on the 8254 counter/timer circuitry.	Contains the status of C/T2 on the 8254.
Base + 7	8254 Control (Write)[2] Status (Read)[3]	Controls the 8254 counter/timer circuitry.	Status 2 Register Contains the clock source, the status of the digital trigger, and the interrupt status. Refer to page 1-23. ID Register Contains the board type. Refer to page 1-25.

Notes

[1] An indirect addressing technique is used to determine which of three Control registers is accessed. Refer to page 1-8 for more information.

[2] The 8254 counter/timer registers are not described in this chapter. Refer to the *DAS-800 Series User's Guide* for a list of companies that provide documentation describing how to program the 8254 counter/timer registers. Refer to page 2-5 for a programming procedure to follow when using the 8254 counter/timers as an internal clock source.

[3] An indirect addressing technique is used to determine which of two Status registers is accessed. Refer to page 1-23 for more information.

Notes: On power-up or system reset, the DAS-800 Series registers are compatible with the DAS-8 Series registers. (On DAS-8 Series boards, only Control register 1 is available at Base Address + 2, Write, and no register is available at Base Address + 7, Read.) You can use application programs written to support the DAS-8 with the DAS-800, application programs written to support the DAS-8 PGA with the DAS-801, and application programs written to support the DAS-8 PGA/G2 with the DAS-802.

Whenever the board is powered up or reset, all output (write) register bits are set to 0.

The remainder of this chapter describes the hardware registers in detail.

A/D Conversion Registers (Base Address + 0, Base Address + 1, Write)

The A/D Conversion (LSB) register (Base Address + 0, Write) and the A/D Conversion (MSB) register (Base Address + 1, Write), shown in Figure 1-1, allow you to initiate an A/D conversion through software. Writing any value to either of these registers initiates an A/D conversion; the actual value written is irrelevant.

Notes: To initiate conversions through software, you must disable hardware conversions by setting the **HCEN** bit of the Conversion Control register to 0. Refer to page 1-11 for more information.

Refer to page 2-1 for programming guidelines to follow when initiating A/D conversions through software.

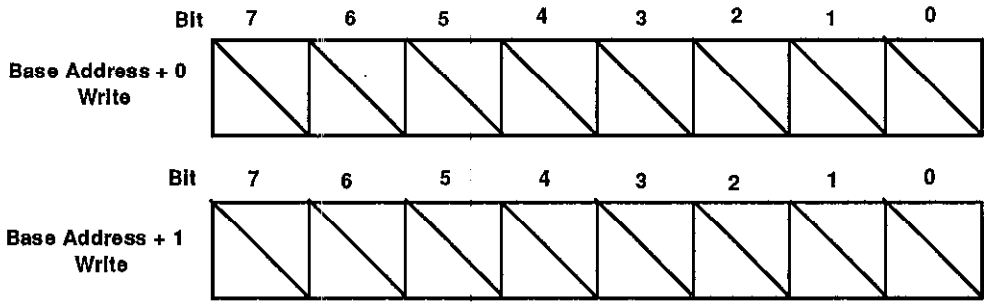


Figure 1-1. A/D Conversion Registers (Write)

A/D Conversion Registers (Base Address + 0, Base Address + 1, Read)

The A/D Conversion (LSB) register (Base Address + 0, Read) and the A/D Conversion (MSB) register (Base Address + 1, Read), shown in Figure 1-2, contain the result of an A/D conversion. The four least significant bits are stored in the A/D Conversion (LSB) register; the eight most significant bits are stored in the A/D Conversion (MSB) register. The A/D Conversion (LSB) register also contains the status of the data in the FIFO.

Notes: Bits 0 and 1 of the A/D Conversion (LSB) register (**FIFO Empty** and **FIFO OVF**) are used only when conversions are initiated by hardware (the **HCEN** bit of the Conversion Control register is set to 1). Refer to page 1-11 for more information about the **HCEN** bit. When conversions are initiated through software, **FIFO Empty** and **FIFO OVF** are always set to 0.

Bits 2 and 3 of the A/D Conversion (LSB) register are always set to 0.

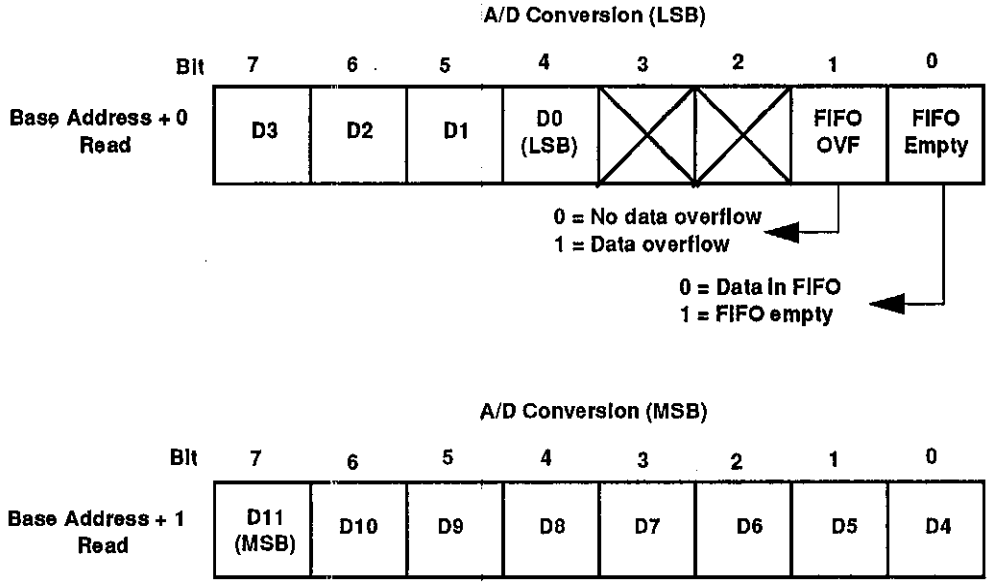


Figure 1-2. A/D Conversion Registers (Read)

Notes: To ensure proper operation of the FIFO, always read both A/D Conversion registers. Read the A/D Conversion (LSB) register first, followed by the A/D Conversion (MSB) register.

If you are acquiring multiple data samples, make sure that you read the A/D Conversion registers once more than the number of samples you are acquiring tc ensure that a FIFO overflow did not occur on the last data sample.

The bits in the A/D Conversion registers are described as follows:

- **FIFO Empty** - Determines the status of the data in the FIFO.

  If **FIFO Empty** = 0, the FIFO contains data.

  If **FIFO Empty** = 1, the FIFO is empty. If you read either of the A/D Conversion registers when **FIFO Empty = 1**, the value read is the last value stored in the FIFO.

- **FIFO OVF (FIFO Overflow)** - Determines the status of the data in the FIFO.

  If **FIFO OVF** = 0, no data in the FIFO was overwritten.

  If **FIFO OVF** = 1, data in the FIFO has been overwritten and one or more samples were lost.

  To restore normal operations after a FIFO overflow condition occurs (**FIFO OVF** = 1), you must disable conversions by setting the **HCEN** bit of the Conversion Control register to 0, and then restart the board and set up your operation again.

- **D0 through D11 (Data)** - A/D conversion data.

  When you initiate an A/D conversion through software, the data is read directly from the ADC and represents the result of the most recent conversion. When you initiate A/D conversions through hardware, the FIFO control logic selects the appropriate sample.

  When reading data in a bipolar input range, data is represented in offset binary format; a code of 0000 0000 0000 represents negative full scale, a code of 1111 1111 1111 represents positive full scale, and a code of 1000 0000 0000 represents 0 V. When reading data in a unipolar input range, the data represents the magnitude of the measured value; a code of 0000 0000 0000 represents 0 V and a code of 1111 1111 1111 represents positive full scale.

  Refer to Appendix B for more information on converting the binary code to voltage.

Control Register (Base Address + 2, Write)

The Control register is actually a group of three different Control registers that are accessed through a common port address (Base Address + 2, Write). An indirect addressing technique determines which of the three Control registers is accessed. The addressing technique uses the register select bits (**CS1** and **CS0**) of the Gain / Control Select register as a pointer to the appropriate register. Refer to page 1-20 for more information about **CS1** and **CS0**.

The three Control registers are described as follows:

- **Control register 1** - Specifies an analog input channel, enables interrupts, and specifies digital output values. Control register 1 is accessed when **CS1** = 0 and **CS0** = 0. This is the default Control register; it is selected whenever the board is powered up or reset.
- **Conversion Control register** - Specifies the clock source, enables a digital trigger, specifies when to generate an interrupt, enables automatic channel scanning, enables a digital gate, and enables hardware conversions. The Conversion Control register is accessed when **CS1** = 0 and **CS0** = 1.
- **Scan Limits register** - Specifies the first channel and the last channel used in automatic channel scanning. The Scan Limits register is accessed when **CS1** = 1 and **CS0** = 0.

Note: If you write to the Control register when **CS1 / CS0** = 11, the results are unpredictable.

The Control registers are described in the following sections.

Control Register 1 (Base Address + 2, Write)

Control register 1, shown in Figure 1-3, is accessed when **CS1 / CS0** = 00.

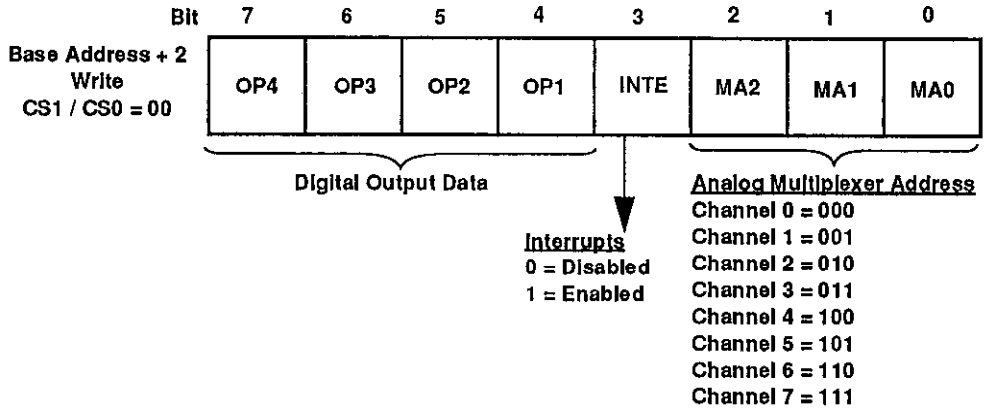


Figure 1-3. Control Register 1

The bits in Control register 1 are described as follows:

- **MA0 through MA2 (Analog Multiplexer Address Selection)** - Select the single analog input channel on which to perform an A/D conversion, as shown in Table 1-2.

Table 1-2. Analog Multiplexer Address Selection

Channel	MA2	MA1	MA0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Note: When you are using automatic channel scanning (the **EACS** bit of the Conversion Control register is set to 1), the values written to these bits are ignored. Refer to page 1-15 for more information about the **EACS** bit.

- **INTE (Interrupt Enable)** - Enables and disables interrupts on the host's PC bus.

  If **INTE** = 0, interrupts are disabled.

  If **INTE** = 1, interrupts are enabled.

Notes: The condition that causes an interrupt to occur is determined by the **IEOC** bit of the Conversion Control register. When the interrupt occurs, the **IRQ** bit of Status register 1 is set to 1. If **INTE** = 1 when **IRQ** is set to 1, the host receives the interrupt. Refer to page 1-14 for more information about the **IEOC** bit. Refer to page 1-18 for more information about the **IRQ** bit.

If interrupts are enabled, you must specify an interrupt level in the configuration file and by setting a jumper on the board. Refer to the *DAS-800 Series User's Guide* for more information.

- **OP1 (Digital Output 1 Data)** - Value written to the digital output line associated with the OP1 pin on the main I/O connector.
- **OP2 (Digital Output 2 Data)** - Value written to the digital output line associated with the OP2 pin on the main I/O connector.
- **OP3 (Digital Output 3 Data)** - Value written to the digital output line associated with the OP3 pin on the main I/O connector.
- **OP4 (Digital Output 4 Data)** - Value written to the digital output line associated with the OP4 pin on the main I/O connector.

Note: For all digital output data bits, a value of 1 forces the output high; a value of 0 forces the output low.

Conversion Control Register (Base Address + 2, Write)

The Conversion Control register, shown in Figure 1-4, is accessed when **CS1 / CS0** = 01.

Note: If you write to bit 6 of the Conversion Control register, the results are unpredictable.

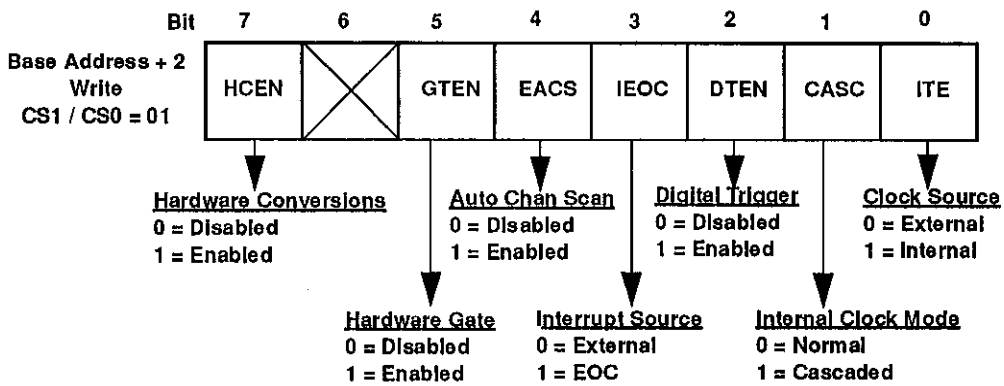


Figure 1-4. Conversion Control Register

The bits in the Conversion Control register are described as follows:

- **HCEN (Hardware Conversion Enable)** - Enables and disables hardware conversions.

  If **HCEN** = 0, hardware conversions are disabled. If a conversion is in progress when you set **HCEN** to 0, the conversion process stops.

Note: If you are using an internal clock and you stop the conversion process at any time, you must reload the 8254 counter/timer(s) with the desired count before restarting conversions. This ensures that the timing between conversions remains consistent.

If **HCEN** = 1, hardware conversions start. The **ITE** bit determines whether conversions begin immediately (internal clock) or are armed, waiting for the next falling edge of the external clock. In addition, the **DTEN** bit determines whether conversions wait for the digital trigger condition to occur.

You can modify the remaining bits of the Conversion Control register only if **HCEN** = 0. If you write to the Conversion Control register with **HCEN** set to 1, the conversion process starts; all other bits in the Conversion Control register retain their previous settings, regardless of the values written to them.

Caution: To ensure that the board works properly when initiating conversions under hardware control, you must follow a particular programming sequence. This sequence is described on page 2-2.

- **ITE (Internal Time Base Enabled)** - Selects the clock source for A/D conversions. This bit affects board operation only if hardware conversions are enabled (**HCEN** = 1).

  If **ITE** = 0, the falling edge of an external clock attached to the INT_IN pin on the main I/O connector initiates each conversion. (The external clock frequency cannot exceed 40 kHz.)

  If **ITE** = 1, the internal clock (the 8254 counter/timer circuitry) determines the time interval between conversions.

Note: If you are using an internal clock, the counter/timer(s) you are using (C/T2 or C/T1 and C/T2) must be programmed for rate generator mode (Mode 2). Refer to page 2-5 and to your 8254 documentation for information on programming a counter/timer mode. The *DAS-800 Series User's Guide* provides a list of companies that provide documentation describing how to program the 8254 counter/timers.

- **CASC (Internal Clock Mode)** - Determines the use of C/T2 and C/T1 of the 8254 counter/timer circuitry when using the internal clock. This bit affects board operation only if the internal clock is enabled (**ITE** = 1).

  If **CASC** = 0, only C/T2 of the 8254 counter/timer circuitry is used (normal mode). C/T2 is loaded with the number of 1 µs clock ticks between conversions.

  If **CASC** = 1, both C/T2 and C/T1 of the 8254 counter/timer circuitry are used (cascaded mode). C/T2 and C/T1 are both loaded with a count value. When C/T2 reaches terminal count, C/T1 decrements by 1. When both C/T2 and C/T1 reach terminal count, a conversion occurs.

  Refer to page 2-5 and to your 8254 documentation for information on loading C/T2 and C/T1. The *DAS-800 Series User's Guide* provides a list of companies that provide documentation describing how to program the 8254 counter/timers.

- **DTEN (Digital Trigger Enable)** - Enables and disables a digital trigger. This bit affects board operation only if hardware conversions are enabled (**HCEN** = 1).

  If **DTEN** = 0, the digital trigger is disabled. If **HCEN** = 1, A/D conversions begin immediately (if using an internal clock source) or at the next falling edge detected on the INT_IN pin on the main I/O connector (if using an external clock source).

  If **DTEN** = 1, the digital trigger is enabled. If **HCEN** = 1, A/D conversions begin when the board detects a rising edge on the IP1 pin on the main I/O connector (if using an internal clock source) or at the

next falling edge detected on the INT_IN pin after the board detects a rising edge on the IP1 pin (if using an external clock source).

Notes: To enable the digital trigger, **DTEN** must be set to 1 AND **GTEN** must be set to 0. If both these bits are set to 1, the hardware gate is enabled, but the digital trigger is disabled. Refer to page 1-15 for information about the **GTEN** bit.

To reset the digital trigger detection logic after the desired number of samples has been acquired, write any value to the Conversion Control register.

You can use software to perform an analog trigger function. Refer to page 2-8 for programming guidelines to follow when setting up an analog trigger.

- **IEOC (Interrupt Select)** - Determines when an interrupt occurs. This bit affects board operation only if interrupts are enabled (**INTE** = 1).

  If **IEOC** = 0, an interrupt occurs when a rising edge is detected on the INT_IN pin on the main I/O connector.

  If **IEOC** = 1, an interrupt occurs at the end of every conversion.

  When the interrupt occurs, the **IRQ** bit of Status register 1 is set to 1. If the **INTE** bit of Control register 1 is set to 1 when **IRQ** is set to 1, the host receives the interrupt. Refer to page 1-10 for more information about the **INTE** bit. Refer to page 1-18 for more information about the **IRQ** bit.

Caution: The **IEOC** bit affects the operation of the interrupt generation logic. When changing the state of this bit, you may generate a spurious interrupt. Therefore, before changing the state of **IEOC**, you should always disable interrupts by setting the **INTE** bit of Control register 1 to 0. Refer to page 1-10 for more information about the **INTE** bit.

- **EACS (Automatic Channel Scan Enable)** - Enables and disables automatic channel scanning. This bit affects board operation only if hardware conversions are enabled (**HCEN** = 1).

  If **EACS** = 0, automatic channel scanning is disabled. If **HCEN** = 1, conversions are performed on the single channel specified by the **MA0** through **MA2** bits of Control register 1. Refer to page 1-9 for more information about the **MA0** through **MA2** bits.

  If **EACS** = 1, automatic channel scanning is enabled. If **HCEN** = 1, conversions are performed on the range of channels specified by the Scan Limits register; the hardware increments the address shortly after the start of each conversion. Refer to the next section for more information about the Scan Limits register.

  Note: Make sure that you specify the start and end channels before you set **EACS** to 1.

- **GTEN (Gate Enable)** - Enables and disables the hardware gate. This bit affects board operation only if hardware conversions are enabled (**HCEN** = 1).

  If **GTEN** = 0, the hardware gate is disabled.

  If **GTEN** = 1, the hardware gate is enabled.

  Note: To enable the hardware gate, both **GTEN** and **DTEN** must be set to 1. Refer to page 1-13 for more information about the **DTEN** bit.

Scan Limits Register (Base Address + 2, Write)

The Scan Limits register, shown in Figure 1-5, is accessed when **CS1 / CS0** = 10.

Note: If you write to bits 6 and 7 of the Scan Limits register, the results are unpredictable.

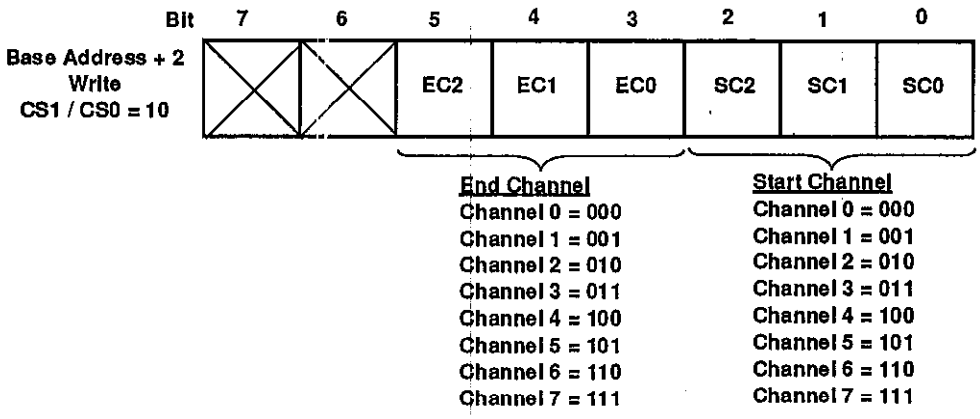


Figure 1-5. Scan Limits Register

The bits in the Scan Limits register are described as follows:

- **SC0 through SC2 (Start Channel Selection)** - Select the first channel used in automatic channel scanning.
- **EC0 through EC2 (End Channel Selection)** - Select the last channel used in automatic channel scanning.

Note: The start channel can be higher or lower than the end channel.

Make sure that you select the start and end channel before you enable automatic channel scanning by setting the **EACS** bit of the Conversion Control register to 1. Refer to page 1-15 for more information about the **EACS** bit.

Status Register 1 (Base Address + 2, Read)

Status register 1 (Base Address +2, Read), shown in Figure 1-6, contains the currently selected analog input channel, the interrupt status, the state of the digital input lines, and the status of the ADC.

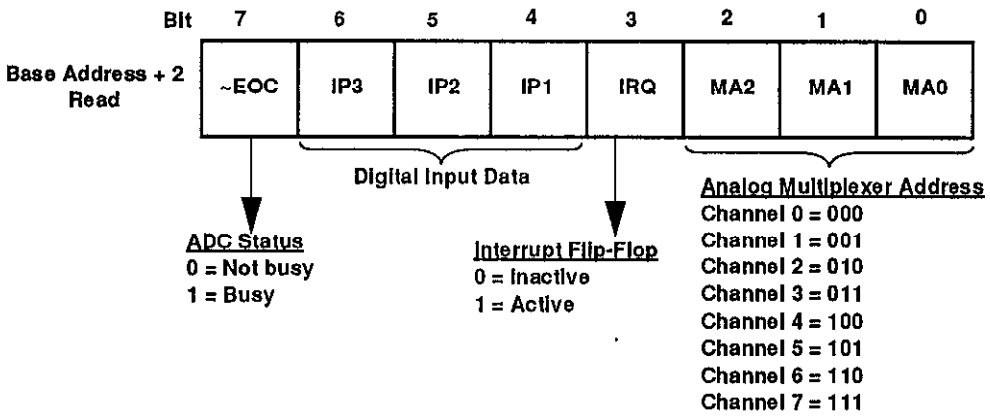


Figure 1-6. Status Register 1 (Read)

The bits in Status register 1 are described as follows:

- **MA0 through MA2 (Analog Multiplexer Address Status)** - Indicate the analog input channel specified in the **MA0** through **MA2** bits of Control register 1. Note that this represents the next channel to be converted. Refer to Table 1-2 on page 1-9 for more information on the meaning of the bits.

Note: These bits have the same meaning as the **MA0** through **MA2** bits in the Gain / Channel Status register. Refer to page 1-22 for more information.

- **IRQ (Interrupt Flip-Flop Status)** - Contains the status of the interrupt flip-flop.

  If **IRQ** = 0, the interrupt flip-flop is inactive, indicating that the interrupt condition has not occurred since this bit was last cleared.

  If **IRQ** = 1, the interrupt flip-flop is active, indicating that the interrupt condition occurred.

Note: The condition that causes the interrupt to occur is determined by the **IEOC** bit of the Conversion Control register. If the **INTE** bit of Control register 1 is set to 1 when the interrupt occurs (**IRQ** = 1), the host receives the interrupt. Refer to page 1-14 for more information about the **IEOC** bit. Refer to page 1-10 for more information about the **INTE** bit.

  You can clear this bit by writing any value to Control register 1.

- **IP1 (Digital Input 1 Data)** - State of the digital input line associated with the IP1 pin on the main I/O connector.

- **IP2 (Digital Input 2 Data)** - State of the digital input line associated with the IP2 pin on the main I/O connector.

- **IP3 (Digital Input 3 Data)** - State of the digital input line associated with the IP3 pin on the main I/O connector.

Note: For all digital input data bits, a value of 1 indicates that the input is high; a value of 0 indicates that the input is low.

- **~EOC** - Contains the status of the ADC End-of-Conversion (~EOC) bit.

  If **~EOC** = 0, the ADC is not performing a conversion. You can read the result of the last conversion from the A/D Conversion registers.

  If **~EOC** = 1, the ADC is performing a conversion and data is not available. Any data read while **~EOC** = 1 is undefined.

Note: The ~EOC bit is valid for software-initiated A/D conversions only.

Gain / Control Select Register (Base Address + 3, Write)

The Gain / Control Select register (Base Address + 3, Write), shown in Figure 1-7, allows you to select the analog input range and the Control or Status register to access.

Note: If you write to bit 4 of the Gain / Control Select register, the results are unpredictable.

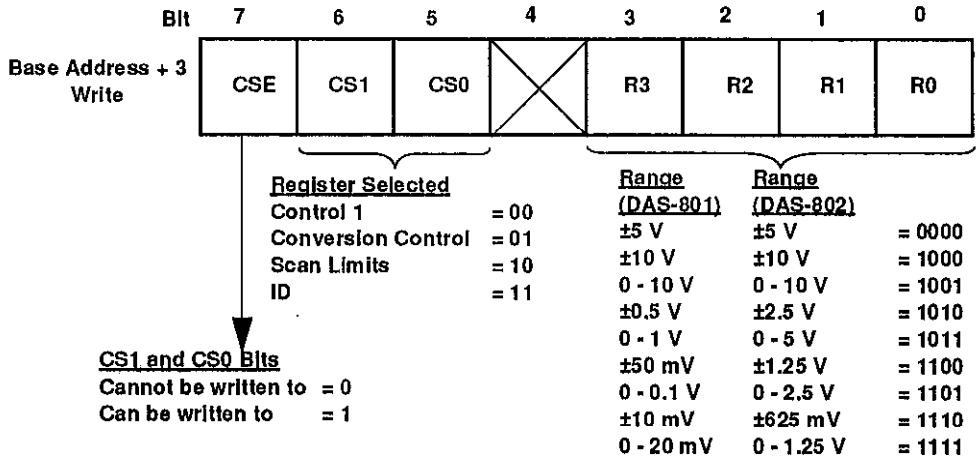


Figure 1-7. Gain / Control Select Register (Write)

The bits in the Gain / Control Select register are described as follows:

- **R0 through R3 (Range Selection)** - Select the analog input range for DAS-801 and DAS-802 boards, as shown in Table 1-3.

Table 1-3. Range Selection Bits

DAS-801 Range	DAS-802 Range	R3	R2	R1	R0
±5 V	±5 V	0	0	0	0
±10 V	±10 V	1	0	0	0
0 - 10 V	0 - 10 V	1	0	0	1
±0.5 V	±2.5 V	1	0	1	0
0 - 1 V	0 - 5 V	1	0	1	1
±50 mV	±1.25 V	1	1	0	0
0 - 100 mV	0 - 2.5 V	1	1	0	1
±10 mV	±625 mV	1	1	1	0
0 - 20 mV	0 - 1.25 V	1	1	1	1

Note: Since DAS-800 boards always have a ±5 V analog input range, these bits have no meaning for DAS-800 boards.

- **CS1 and CS0 (Register Selection)** - Determine the Control register that is accessed when you write data to Base Address + 2, Write, and the Status register that is accessed when you read from Base Address + 7, Read. The meaning of these bits is shown in Table 1-4.

Table 1-4. Register Selection Bits

Control Register	Status Register	CS1	CS0
Control 1	Status 2	0	0
Conversion Control	Status 2	0	1
Scan Limits	Status 2	1	0
Not applicable	ID	1	1

- **CSE (Register Selection Enable)** - Determines whether the register selection bits (**CS1** and **CS0**) can be written to.

  If **CSE** = 0, the register selection bits (**CS1** and **CS0**) cannot be written to; writing to the range selection bits (**R0** through **R3**) does not affect the state of **CS1** and **CS0**.

  If **CSE** = 1, the register selection bits (**CS1** and **CS0**) can be written to; writing to **CS1** and **CS0** does not affect the state of **R0** through **R3**.

Gain / Channel Status Register (Base Address + 3, Read)

The Gain / Channel Status register (Base Address + 3, Read), shown in Figure 1-8, contains the currently selected analog input range, the currently selected analog input channel, and the status of automatic channel scanning.

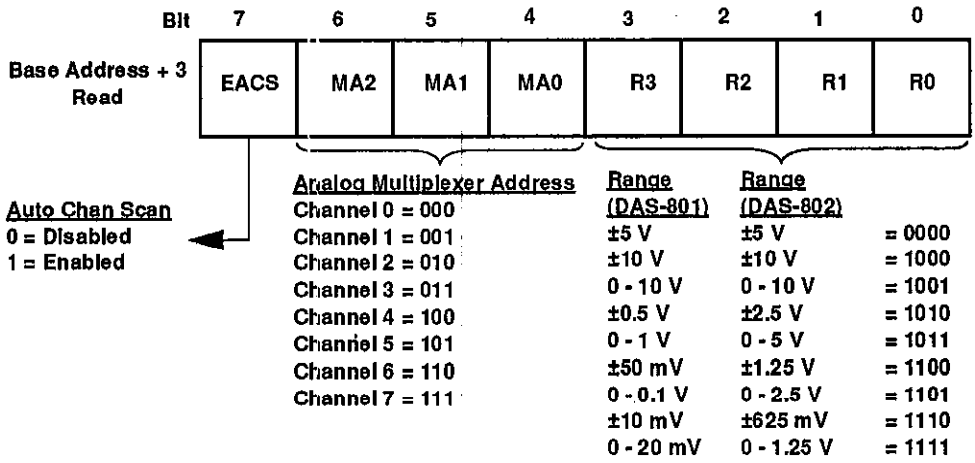


Figure 1-8. Gain / Channel Status Register (Read)

The bits in the Gain / Channel Status register are described as follows:

- **R0 through R3 (Range Status)** - Indicate the state of the **R0** through **R3** bits of the Gain / Control Select register. Refer to Table 1-3 on page 1-20 for more information on the meaning of the bits.

- **MA0 through MA2 (Analog Multiplexer Address Status)** - Indicate the state of the **MA0** through **MA2** bits of Control register 1. Refer to Table 1-2 on page 1-9 for more information on the meaning of these bits.

Note: These bits have the same meaning as the **MA0** through **MA2** bits in Status register 1. Refer to page 1-17 for more information.

- **EACS (Automatic Channel Scan Status)** - Indicates the state of the **EACS** bit of the Conversion Control register. Refer to page 1-15 for more information about the meaning of this bit.

Status Register (Base Address + 7, Read)

The Status register is actually a group of two different Status registers that are accessed through a common port address (Base Address + 7, Read). An indirect addressing technique determines which of the two Status registers is accessed. The addressing technique uses the register select bits (**CS1** and **CS0**) of the Gain / Control Select register as a pointer to the appropriate register. Refer to page 1-20 for more information about **CS1** and **CS0**.

The two Status registers are described as follows:

- **Status register 2** - Contains the clock source, the status of the digital trigger, and the interrupt status. Status register 2 is accessed when **CS1 / CS0** = 00, 01, or 10. This is the default Status register; it is selected whenever the board is powered up or reset.

- **ID register** - Contains the board type. The ID register is accessed when **CS1** = 1 and **CS0** = 1.

The Status registers are described in the following sections.

Status Register 2 (Base Address + 7, Read)

Status register 2, shown in Figure 1-9, is accessed when **CS1 / CS0** = 00, 01, or 10.

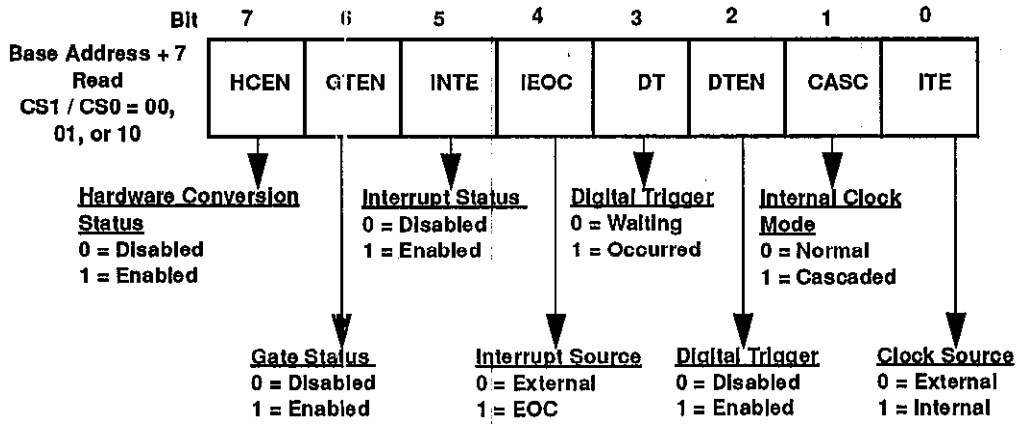


Figure 1-9. Status Register 2

The bits in Status register 2 are described as follows:

- **ITE (Internal Time Base Status)** - Indicates the state of the **ITE** bit of the Conversion Control register. Refer to page 1-12 for more information about the meaning of this bit.

- **CASC (Internal Clock Mode Status)** - Indicates the state of the **CASC** bit of the Conversion Control register. Refer to page 1-13 for more information about the meaning of this bit.

- **DTEN (Digital Trigger Status)** - Indicates the state of the **DTEN** bit of the Conversion Control register. Refer to page 1-13 for more information about the meaning of this bit.

- **DT (Digital Trigger Detect)** - Indicates whether the digital trigger event occurred.

  If **DT** = 0, the digital trigger circuitry is armed and waiting for the digital trigger event to occur.

  If **DT** = 1, the digital trigger event occurred.

This bit is meaningful only if the digital trigger is enabled (the **DTEN** bit of the Conversion Control register is set to 1). Refer to page 1-13 for more information about the **DTEN** bit.

- **IEOC (Interrupt Source)** - Indicates the state of the **IEOC** bit of the Conversion Control register. Refer to page 1-14 for more information about the meaning of this bit.

- **INTE (Interrupt Status)** - Indicates the state of the **INTE** bit of Control register 1. Refer to page 1-10 for more information about the meaning of this bit.

- **GTEN (Gate Status)** - Indicates the state of the **GTEN** bit of the Conversion Control register. Refer to page 1-15 for more information about the meaning of this bit.

- **HCEN (Hardware Conversion Status)** - Indicates the state of the **HCEN** bit of the Conversion Control register. Refer to page 1-11 for more information about the meaning of this bit.

ID Register (Base Address + 7, Read)

The ID register, shown in Figure 1-10, is accessed when **CS1** = 1 and **CS0** = 1.

Note: The values of bits 2 through 7 of the ID register are meaningless.

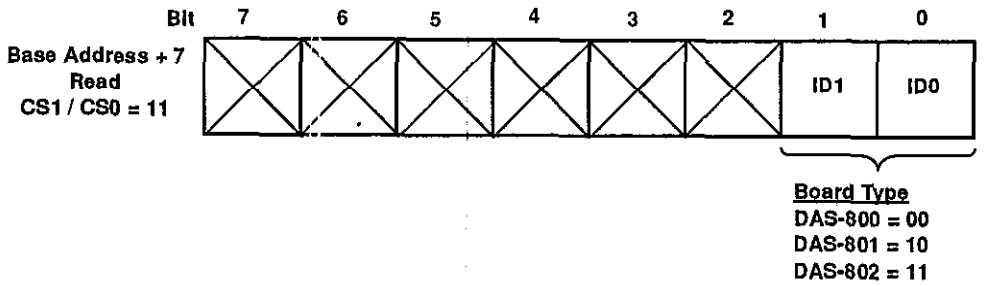


Figure 1-10. ID Register

The identification bits (**ID1** and **ID0**) indicate the DAS-800 Series board type, as shown in Table 1-5.

Table 1-5. Identification Bits

Board Type	ID1	ID0
DAS-800	0	0
DAS-801	1	0
DAS-802	1	1
Reserved	0	1

2

Programming Guidelines

This chapter contains guidelines you should follow when programming various functions of DAS-800 Series boards.

Performing a Software-Initiated A/D Conversion

To perform a software-initiated A/D conversion, perform the following steps:

1. In the Gain / Control Select register, set **R0** through **R3** to 0000 to specify a gain of ±5 V (DAS-801 and DAS-802 only) and set **CSE** to 1 and **CS1 / CS0** to 01 to access the Conversion Control register.

   Note: To prevent the internal amplifier circuitry from saturating, it is recommended that you initially specify a gain of ±5 V. You can set the desired gain (if different) at a later point (step 5).

2. Write to the Conversion Control register with all bits set to 0.

3. In the Gain / Control Select register, set **CS1 / CS0** to 00 to access Control register 1.

4. In Control register 1, select the analog input channel using the **MA0** through **MA2** bits. Refer to page 1-9 for more information.

5. In the Gain / Control Select register, set **R0** through **R3** to the desired gain. Refer to page 1-20 for more information.

6. Wait at least 50 µs for the amplifier circuitry to settle.

7. Initiate a software conversion by writing any value to either of the A/D Conversion registers.

8. Continue to read the **~EOC** bit of Status register 1 until it equals 0.

9. Read the four least significant bits of the last conversion from the A/D Conversion (LSB) register.

10. Read the eight most significant bits of the last conversion from the A/D Conversion (MSB) register.

11. Shift the resulting 16-bit value right by four bits to obtain the magnitude of the conversion (0 to 4095).

Performing a Hardware-Initiated A/D Conversion

To perform a hardware-initiated A/D conversion, perform the following steps:

1. In the Gain / Control Select register, set **CSE** to 1 and set **CS1 / CS0** to 01 to access the Conversion Control register.

2. Write to the Conversion Control register, making sure that **HCEN** and **EACS** are set to 0.

3. If you are using a single analog input channel, go to step 4.

   If you are using automatic channel scanning, go to step 6.

4. In the Gain / Control Select register, set **CS1 / CS0** to 00 to access Control register 1.

5. In Control register 1, select the analog input channel using **MA0** through **MA2**. Refer to page 1-9 for more information. Continue at step 8.

6. In the Gain / Control Select register, set **CS1 / CS0** to 10 to invoke the Scan Limits register.

7. In the Scan Limits register, specify the range of channels using the start channel bits (**SC0** through **SC2**) and the end channel bits (**EC0** through **EC2**). Refer to page 1-16 for more information. Continue at step 8.

8. In the Gain / Control Select register, set **CS1 / CS0** to 01 to invoke the Conversion Control register.

9. In the Conversion Control register, specify the appropriate hardware conversion options. Refer to page 1-11 for more information.

Note: If you intend to read data as a background task, make sure that you set **IEOC** to 1.

---

10. If you are using the internal clock, set the counter/timer mode for C/T2 and C/T1 of the 8254 counter/timer circuitry to Mode 2 and specify the initial count value. Refer to page 2-5 for additional information.

11. With **CS1 / CS0** still set to 01, write to the Conversion Control register again, making sure that **HCEN** and **EACS** (if applicable) are set to 1.

Note: When you write to the Conversion Control register with **HCEN** set to 1, all the other bits retain their prior settings, regardless of the value written at this time. This allows you to initialize the conversion control hardware to a known state before you initiate the first conversion and saves you from having to keep track of which bits were previously set when starting an operation.

---

12. If you specified a digital trigger in step 9 (**DTEN** = 1), the board waits for the trigger condition to be satisfied and then starts conversions.

13. If you want to read data as a foreground task, go to step 14.

    If you want to read data as a background task, go to step 15.

14. Continue to read the A/D Conversion registers, storing data only when **FIFO Empty** = 0 and **FIFO OVF** = 0. When the appropriate number of samples has been acquired, set the **HCEN** bit of the Conversion Control register to 0 and return to your application program.

15. Make sure that the hardware is set to generate an interrupt at the end of a conversion (**IEOC** = 1 and **INTE** = 1).

16. When the end-of-conversion interrupt occurs, your Interrupt Service Routine (ISR) must continue to read the A/D Conversion registers, storing data only when **FIFO Empty** = 0 and **FIFO OVF** = 0.

17. When **FIFO Empty** = 1, the FIFO is empty and all data has been read. At this point, the ISR can return to your application program.

Notes: If **FIFO OVF** = 1, one or more samples were lost. To restore normal operations after a FIFO overflow condition occurs, you must disable conversions by setting the **HCEN** bit of the Conversion Control register to 0, and then restart the board and set up your operation again, using a slower sample rate.

It is possible to read the least significant byte of a data sample before the FIFO overflows, but have the most significant byte of the same data sample overwritten. Therefore, it is recommended that you discard the last data byte read before the FIFO overflowed.

---

18. The process continues from step 16 until the required number of samples has been acquired. At this point, set the **HCEN** and **IEOC** bits of the Conversion Control register and the **INTE** bit of Control register 1 to 0.

Setting Up the Internal Clock

To set up the 8254 counter/timer circuitry for use as an internal clock source, perform the following steps:

1. In the Gain / Control Select register, set **CSE** to 1 and set **CS1 / CS0** to 01 to access the Conversion Control register.

2. Write to the Conversion Control register, making sure that **HCEN** is set to 0. This disables any hardware conversions already in progress.

3. Write to the Conversion Control register again, specifying the appropriate hardware conversion options. Refer to page 1-11 for more information.

Note: Make sure that you set **ITE** to 1. If you want to use cascaded mode, make sure that you set **CASC** to 1.

---

4. Decide which counter/timer(s) you want to use and the count you want to load into each, as follows:

   – For normal mode (**CASC** = 0), you use only C/T2. The count value you load into C/T2 equals the number of microseconds between conversions. For example, if you want to initiate a conversion every 100 µs (10 kHz), you load a count of 100.

     If you want to initiate conversions at rates of 15.25 Hz (one conversion every 65.536 ms) to 40 kHz (one conversion every 25 µs), it is recommended that you use normal mode. Since the maximum conversion rate of the ADC is 40 kHz, do not load a count value less than 25.

- For cascaded mode (**CASC** = 1), you use C/T2 and C/T1. C/T2 is clocked once every microsecond and acts as a prescaler to C/T1. A conversion is initiated each time both C/T2 and C/T1 reach terminal count. You load C/T2 and C/T1 so that the count in C/T2 multiplied by the count in C/T1 equals the number of microseconds between conversions. For example, if you want to initiate one conversion every 5 ms or 5,000 µs (2 kHz), you could load C/T2 with a count of 1,000 and C/T1 with a count of 5; you could also load C/T2 with a count of 100 and C/T1 with a count of 50.

  Note that cascaded mode may not provide the same degree of resolution as normal mode, since the conversion rate must be an integer divisible without a remainder. For example, you cannot initiate a conversion every 29 µs in cascaded mode, since 29 is a prime number. Also note that you cannot load C/T2 or C/T1 with a count value less than 2.

5. If you are using normal mode, perform step 6 and step 7 and then go to step 10.

   If you are using cascaded mode, perform step 6 through step 9 and then go to step 10.

6. Specify rate generator mode (Mode 2) for C/T2 by writing B4 hexadecimal (10110100) to the 8254 Control register (Base Address + 7).

7. Specify the count value for C/T2 by writing a 16-bit value to Base Address + 6.

   You load the 16-bit value as two 8-bit values. Write the eight least significant bits first, followed by the eight most significant bits. For example, to load C/T2 with a count value of 511 (01FF hexadecimal), write FF hexadecimal to Base Address + 6, and then write 01 hexadecimal to Base Address + 6. Continue at step 10.

8. Specify rate generator mode (Mode 2) for C/T1 by writing 74 hexadecimal (01110100) to the 8254 Control register (Base Address + 7).

9. Specify the count value for C/T1 by writing a 16-bit value to Base Address + 5.

   You load the 16-bit value as two 8-bit values. Write the eight least significant bits first, followed by the eight most significant bits. For example, to load C/T1 with a count value of 20 (0014 hexadecimal), write 14 hexadecimal to Base Address + 6, and then write 00 hexadecimal to Base Address + 6. Continue at step 10.

Note: Writing to the 8254 Control register resets the internal 8254 logic so that the next two writes set the appropriate count value. Therefore, for each counter/timer, write to the 8254 Control register first, then immediately write to the counter/timer register twice.

You can program C/T1 first, if desired. If you do, make sure that you first write to the 8254 Control register first with the code for C/T1, then write to Base Address + 5 twice, then write to the 8254 Control register with the code for C/T2, and finally write to Base Address + 6 twice.

---

10. Once the 8254 is initialized, you can start hardware-initiated conversions by writing to the Conversion Control register with **HCEN** set to 1.

Refer to your 8254 documentation for additional information on programming the 8254 counter/timer circuitry. The *DAS-800 Series User's Guide* provides a list of companies that provide documentation describing how to program the 8254 counter/timers.

Setting Up an Analog Trigger

To set up an analog trigger, perform the following steps:

1. In the Gain / Control Select register, set **CSE** to 1 and set **CS1 / CS0** to 01 to access the Conversion Control register.

2. Write to the Conversion Control register, making sure that **HCEN** and **EACS** are set to 0.

3. In the Gain / Control Select register, set **CS1 / CS0** to 00 to access Control register 1.

4. In Control register 1, specify the address of the channel you want to use as the analog trigger using **MA0** through **MA2**. Refer to page 1-9 for more information.

5. Periodically, initiate a software A/D conversion by writing any value to either of the A/D Conversion registers.

6. After each conversion, read the data from the A/D Conversion registers.

7. When the desired analog value is reached, set up the board to perform hardware-initiated conversions. Begin at step 3 on page 2-2.

A

Summary of Registers

Figure A-1 shows a summary of the DAS-800 Series write registers. Figure A-2 shows a summary of the DAS-800 Series read registers. Table A-1 contains an alphabetical list of all the bits provided in the DAS-800 Series registers, a brief description of the bits, and the pages you can refer to for additional information.

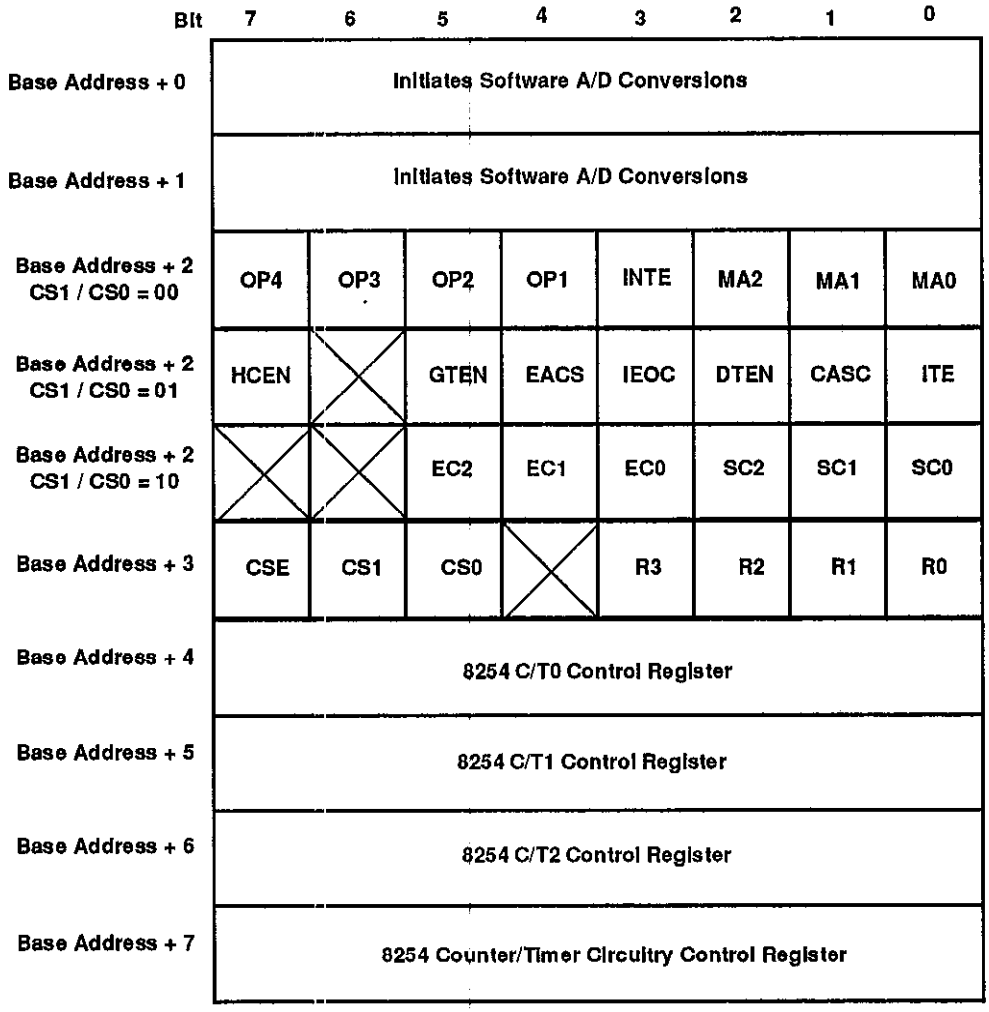
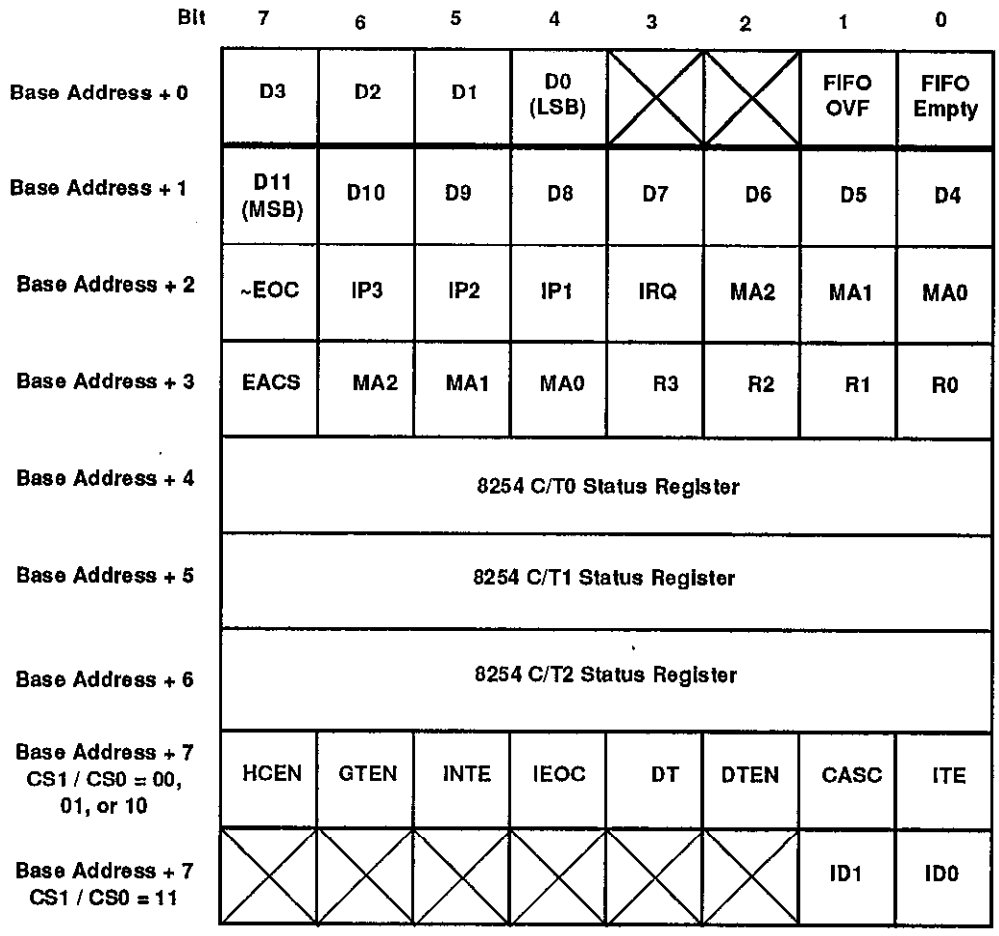


Figure A-1. DAS-800 Series Write Registers



Bit	7	6	5	4	3	2	1	0
Base Address + 0	D3	D2	D1	D0 (LSB)			FIFO OVF	FIFO Empty
Base Address + 1	D11 (MSB)	D10	D9	D8	D7	D6	D5	D4
Base Address + 2	~EOC	IP3	IP2	IP1	IRQ	MA2	MA1	MA0
Base Address + 3	EACS	MA2	MA1	MA0	R3	R2	R1	R0
Base Address + 4	8254 C/T0 Status Register							
Base Address + 5	8254 C/T1 Status Register							
Base Address + 6	8254 C/T2 Status Register							
Base Address + 7 CS1 / CS0 = 00, 01, or 10	HCEN	GTEN	INTE	IEOC	DT	DTEN	CASC	ITE
Base Address + 7 CS1 / CS0 = 11							ID1	ID0

Figure A-2. DAS-800 Series Read Registers

Table A-1. Summary of Register I/O Bits

Bit Name	Description	Page References
CASC	Specifies/contains internal clock mode (normal or cascaded).	page 1-13 page 1-24
CS0 and CS1	Specify Control or Status register to access.	page 1-20
CSE	Enables register select bits.	page 1-21
D0 through D11	Contains A/D conversion data.	page 1-7
DT	Indicates whether digital trigger condition was detected.	page 1-24
DTEN	Enables / contains status of digital trigger.	page 1-13 page 1-24
EACS	Enables / contains status of automatic channel scanning.	page 1-15 page 1-22
EC0 through EC2	Specifies last channel used in automatic channel scanning.	page 1-16
~EOC	Contains status of ADC ~EOC bit.	page 1-18
FIFO Empty	Indicates whether the FIFO contains data.	page 1-7
FIFO OVF	Indicates whether data in the FIFO was overwritten.	page 1-7
GTEN	Enables / contains status of the hardware gate.	page 1-15 page 1-25
HCEN	Enables / contains status of hardware-initiated conversions.	page 1-11 page 1-25
ID0 and ID1	Contain the DAS-800 Series board type.	page 1-26
IEOC	Specifies / contains interrupt source.	page 1-14 page 1-25
INTE	Enables / contains status of interrupt.	page 1-10 page 1-25
IP1 through IP3	Contains digital input data.	page 1-18
IRQ	Contains status of interrupt flip-flop.	page 1-18
ITE	Specifies / contains internal clock source.	page 1-12 page 1-24

Table A-1. Summary of Register I/O Bits (cont.)

Bit Name	Description	Page References
MA0 through MA2	Specifies / contains analog multiplexer address.	page 1-9 page 1-22
OP1 through OP4	Specifies digital output data.	page 1-10
R0 through R3	Specifies / contains analog input range.	page 1-20 page 1-22
SC0 through SC2	Specifies first channel used in automatic channel scanning.	page 1-16

B

Converting Binary Code to Voltage

Analog input data is stored in binary code in the upper four bits of the A/D Conversion (LSB) register and in the eight bits of the A/D Conversion (MSB) register. You may want to convert this binary code to a more meaningful voltage value.

To convert the binary code to voltage, use one of the following equations, where *count* is the count value stored in the A/D Conversion registers converted to decimal format (0 to 4095), 10 V is the span of the analog input range, 4096 is the number of counts available in 12 bits, *gain* is the gain of the analog input channel, and 2048 is the offset value:

DAS-800

Always bipolar input range type:

$$\text{Voltage} = (\text{count} - 2048) \times \frac{10}{4096}$$

DAS-801 / DAS-802

For unipolar input range type:

$$\text{Voltage} = \left(\text{count} \times \frac{10}{4096}\right) \div \text{gain}$$

For bipolar input range type:

$$\text{Voltage} = \left((\text{count} - 2048) \times \frac{10}{4096} \right) \div \text{gain}$$

For example, assume that you are reading analog input data from a channel on a DAS-801 board configured for a unipolar input range type; the channel collects the data at a gain of 10. The count value stored in the A/D Conversion registers (converted to decimal format) is 3072. The voltage is determined as follows:

$$\left(3072 \times \frac{10}{4096} \right) \div 10 = 0.75 \text{ V}$$

As another example, assume that you are reading analog input data from a channel on a DAS-802 board configured for a bipolar input range type; the channel collects the data at a gain of 2. The count value stored in the A/D Conversion registers (converted to decimal format) is 1024. The voltage is determined as follows:

$$\left((1024 - 2048) \times \frac{10}{4096} \right) \div 2 = -1.25 \text{ V}$$