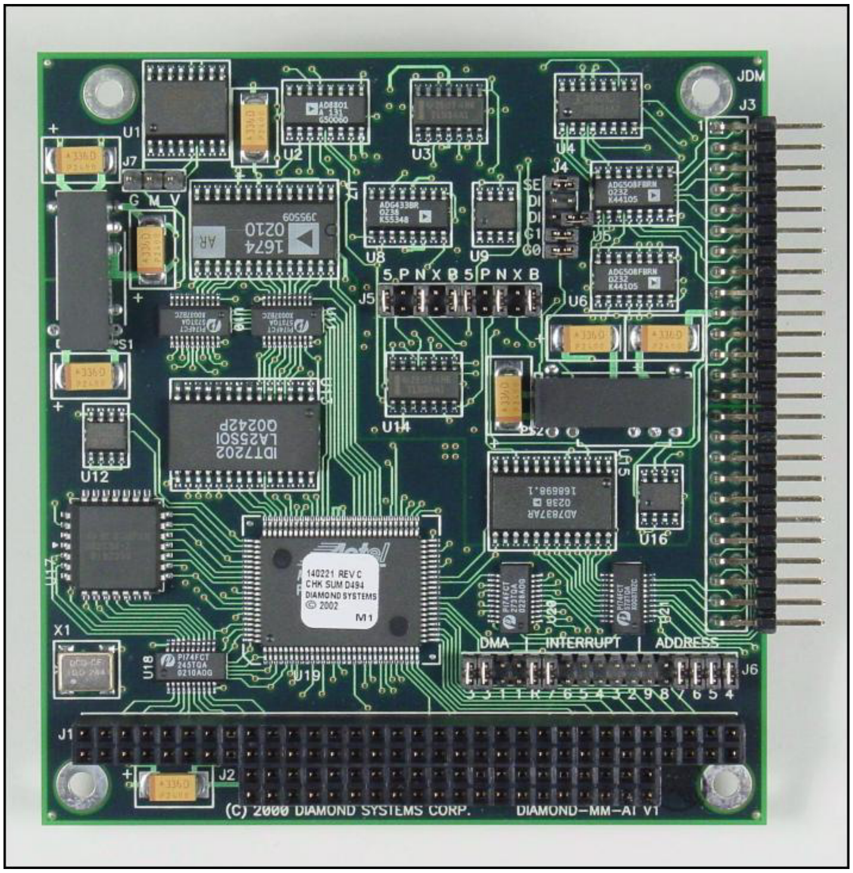
DIAMOND SYSTEMS CORPORATION

DIAMOND-MM-AT

Autocalibrating

12-bit Analog I/O PC/104 Module

User Manual V1.21



Diamond Systems Corporation
8430-D Central Ave.
Newark, CA 94560
Tel (510) 456-7800
Fax (510) 456-7878
techinfo@diamondsystems.com
www.diamondsystems.com

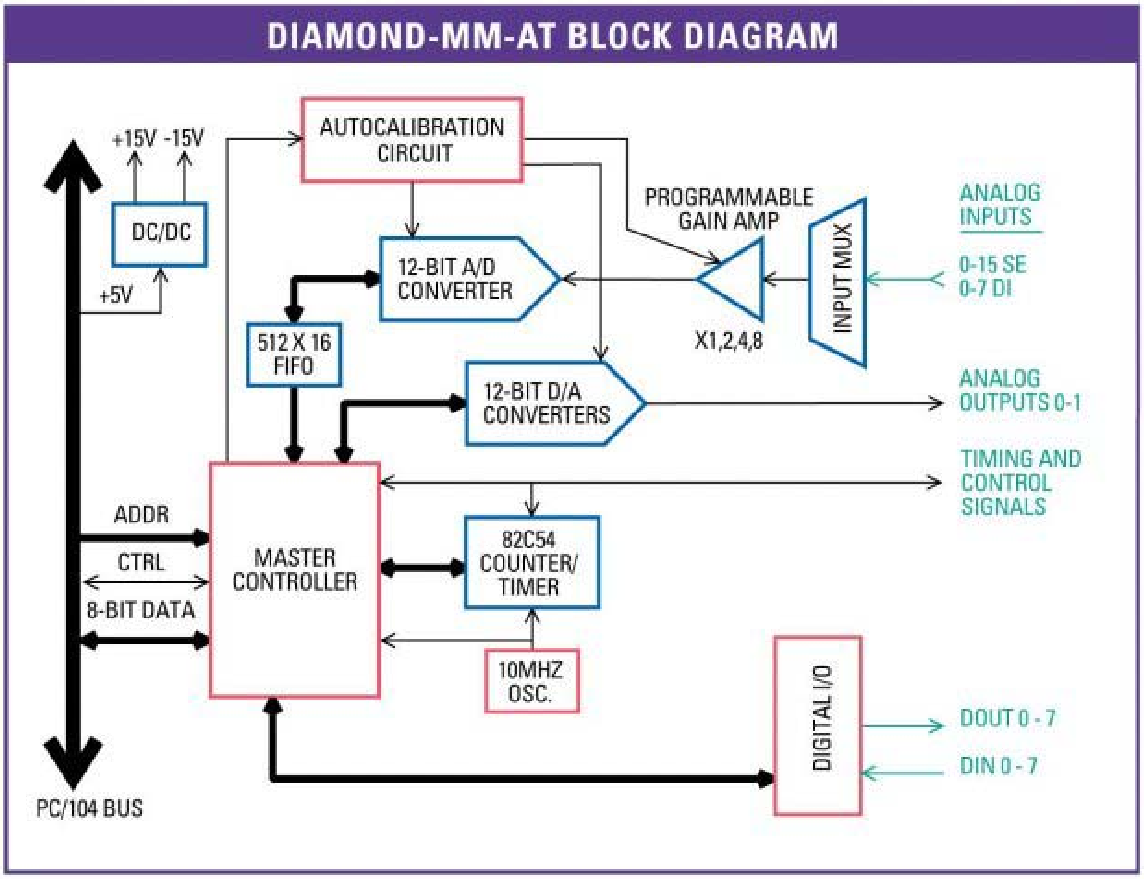
TABLE OF CONTENTS

1. DESCRIPTION

Diamond-MM-AT (DMM-AT) is a full-featured 12-bit analog I/O PC/104 module containing both analog input and analog output channels along with digital I/O and counter/timers. The register map and connector pinout of this board are the same as on our Diamond-MM (DMM) board, enabling DMM-AT to be a drop-in replacement for most existing DMM applications. DMA is not supported on this board.

Features

- 16 analog inputs
- 12-bit A/D converter, 100KHz maximum sampling rate
- Programmable gain
- A/D FIFO for reduced interrupt overhead
- 2 12-bit D/A channels
- 8 digital inputs
- 8 digital outputs
- 32-bit counter/timer for A/D timing
- 16-bit counter/timer for user application
- Programmable clock sources for counter/timers
- Autocalibration of analog input and output circuitry
- Programmable D/A full-scale range
- -40 to +85°C operation standard



2. I/O HEADER PINOUT AND PIN DESCRIPTION

Diamond-MM-AT provides a 50-pin header labeled J3 for all user I/O.

Vin 15 / 7-	1	2	Vin 7 / 7+
Vin 14 / 6-	3	4	Vin 6 / 6+
Vin 13 / 5-	5	6	Vin 5 / 5+
Vin 12 / 4-	7	8	Vin 4 / 4+
Vin 11 / 3-	9	10	Vin 3 / 3+
Vin 10 / 2-	11	12	Vin 2 / 2+
Vin 9 / 1-	13	14	Vin 1 / 1+
Vin 8 / 0-	15	16	Vin 0 / 0+
Agnd	17	18	Vref Out
Agnd	19	20	Vout 0
Agnd	21	22	Vout 1
Agnd	23	24	+15V
-15V	25	26	Vref In 0
Agnd	27	28	Vref In 1
In 0-	29	30	Dgnd
Out 0	31	32	Out 2
Dout 7	33	34	Dout 6
Dout 5	35	36	Dout 4
Dout 3	37	38	Dout 2
Dout 1	39	40	Dout 0
Din 7	41	42	Din 6
Din 5	43	44	Din 4
Din 3	45	46	Din 2 / Gate 0
Din 1	47	48	Din 0 / Gate 1/2
+5V	49	50	Dgnd

Signal Name	**Definition**
Vin 7/7+ ~ Vin 0/0+	Analog input channels 7 - 0 in single-ended mode; High side of input channels 7 - 0 in differential mode
Vin 15/7- ~ Vin 8/0-	Analog input channels 15 - 8 in both single-ended mode; Low side of input channels 7 - 0 in differential mode
Vout0, Vout 1	Analog output channels 0 and 1
Vref Out	+5V or -5V precision reference voltage output (user selectable)
Vref In 0, 1	External reference voltage inputs for custom D/A full-scale ranges
Dout7 - Dout0	Digital output port, TTL / CMOS compatible
Din7 - Din0	Digital input port, TTL / CMOS compatible
Din2 / Gate 0	Digital input line 2 doubles as the gate control for counter 0; Counter 0 counts when this line is high and holds when it is low
Din0 / Gate 1/2	Digital input line 0 doubles as a gate signal for counters 1 and 2 as determined by the control register at base + 11
In0-	Counter 0 input, negative polarity (negative edge trigger)
Out0, Out2	Counter 0 and Counter 2 output signals
±15V	Analog power supply; maximum current draw 10mA per line
+5V	Connected to PC/104 bus power supply **CAUTION: Power supply outputs are not short-circuit protected!**
Agnd	Analog ground

Dgnd	Digital ground

3. BOARD CONFIGURATION

Refer to the Drawing of Diamond-MM-AT in chapter 4 for locations of the configuration items mentioned here.

3.1 Base Address

Each board in the system must have a different base address. Diamond-MM-AT's base address is set with J6, located at the lower right corner of the board. Each of the six jumper locations marked 9, 8, 7, 6, 5, 4 corresponds to the same-numbered address bit. A jumper out is equal to a 1, and a jumper in is equal to a 0. The jumpers correspond to address bits 9-4 and leave bits 3-0 unselected, resulting in a 16-byte I/O decode. Although any 16-byte location is selectable, certain locations are reserved or may cause conflicts with other system resources. The table below lists recommended base address settings for Diamond-MM-AT. The default setting is 300 Hex.

Base Address		**Jumper Position**					
Hex	**Decimal**	**9**	**8**	**7**	**6**	**5**	**4**
220	544	Out	In	In	In	Out	In
240	576	Out	In	In	Out	In	In
250	592	Out	In	In	Out	In	Out
260	608	Out	In	In	Out	Out	In
280	640	Out	In	Out	In	In	In
290	656	Out	In	Out	In	In	Out
2A0	672	Out	In	Out	In	Out	In
2B0	688	Out	In	Out	In	Out	Out
2C0	704	Out	In	Out	Out	In	In
2D0	720	Out	In	Out	Out	In	Out
2E0	736	Out	In	Out	Out	Out	In
300	768 (Default)	Out	Out	In	In	In	In
330	816	Out	Out	In	In	Out	Out
340	832	Out	Out	In	Out	In	In
350	848	Out	Out	In	Out	In	Out
360	864	Out	Out	In	Out	Out	In
380	896	Out	Out	Out	In	In	In
390	912	Out	Out	Out	In	In	Out
3A0	928	Out	Out	Out	In	Out	In
3C0	960	Out	Out	Out	Out	In	In
3E0	992	Out	Out	Out	Out	Out	In

3.2 J4: A/D Single-Ended / Differential Mode and Programmable Gain

A/D input channels may be either single-ended (two wires consisting of input and ground) or differential (three wires consisting of input+, input-, and ground). All inputs on DMM-AT must be configured in the same manner. To select single-ended inputs, install one jumper in the SE position on J4. To select differential inputs, install two jumpers **horizontally** in the two DI positions. If you install these two jumpers vertically the inputs will not function properly.

Analog inputs may also be configured for unipolar (positive voltages only) or bipolar (both positive and negative voltages) operation. This configuration is done in software.

In most applications programmable gain is desirable. It enables you to select the gain for each input channel independently in software. To select programmable gain, install two jumpers **horizontally** in the G1 and G0 positions on J4. This connects the gain control pins on the Actel controller chip to the gain amplifier inputs so that software can control the gain setting.

For backward compatibility with Diamond-MM, the input gain may be set with jumpers. To do this, install jumpers in G1 and/or G0 to select the gain according to the table below. Also make sure that your program does not modify the two control bits G1 and G0 in the register at Base + 11. The Diamond-MM driver software does not use this register, so if you install a Diamond-MM-AT board in a system using software written for the Diamond-MM, the gain setting will remain constant according to your selection.

Jumper Settings for Fixed Gain Only

Gain	G1	G0
1	In	In
2	In	Out
4	Out	In
8	Out	Out

3.3 J5: D/A Configuration

The two analog outputs may be configured for unipolar/bipolar operation, and their full-scale may be either a fixed value from an on-board fixed reference, a programmable value from an on-board programmable reference, or a user-provided value from a pin on the I/O header. Each output channel may be configured independently. Jumper block J5 is used for the analog output configuration. Channel 0 uses the left group of 5 pairs, and channel 1 uses the right group of 5 pairs. Each group is labeled 5 P N X B.

The three items to configure are: internal/external reference; if internal, fixed or programmable; and finally unipolar/bipolar operation:

Position	Function
5	Selects 5V reference (if N jumper is installed)
P	Selects programmable reference (if N jumper is installed)
N	Selects internal reference, 5 or P above
X	Selects external reference, J3 pin 26 for DAC 0 or J3 pin 28 for DAC1
B	Selects bipolar mode if in, unipolar mode if out

For each DAC, one jumper must be installed in either N or X but not both.

For each DAC, if a jumper is installed in N, then one jumper must be installed in either 5 or P but not both.

If both DACs are set to P and N (internal programmable reference), then both DACs must have the same B jumper setting (in for both or out for both).

3.4 J6: Interrupt level and DMA level

To select the interrupt level, install a jumper in the location 7, 6, 5, 4, 3, or 2 in the interrupt portion of J6. Only one jumper should be installed in these locations. These are the only interrupt levels supported by Diamond-MM-AT.

In most instances you should also install a jumper in the R position to connect a 1KΩ pull-down resistor to the interrupt line. This enables interrupt sharing with other boards on the same interrupt level. Only one pull-down resistor should only be installed on any interrupt level on the PC/104 bus, so if you have two or more boards sharing the same interrupt level, be sure that only one of them has the R jumper installed.

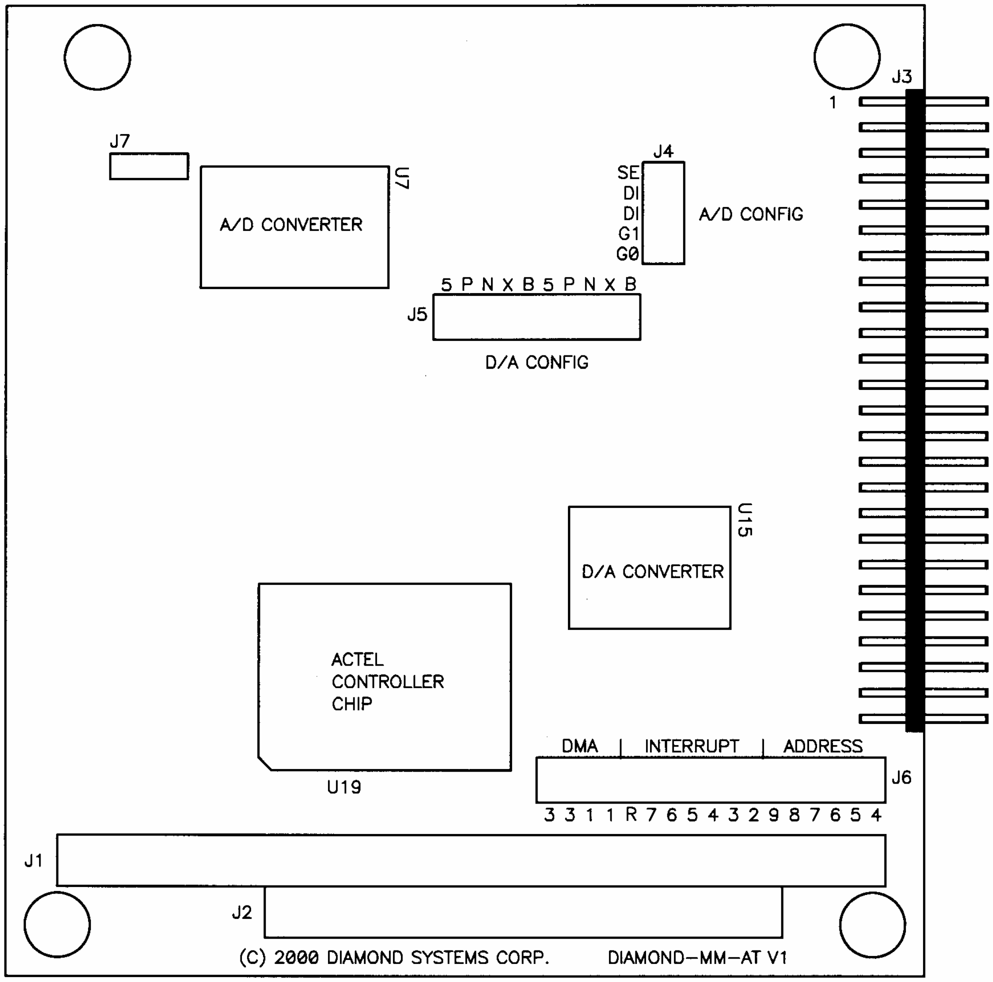
DMA is not currently supported on the Diamond-MM-AT board due to the fact that the FIFO combined with interrupt operation enable the board to operate at full speed in any operating system supported by the DSC Universal Driver. However the configuration locations are provided for future upgrade. If DMA is required please contact technical support.

3.5 J7: Test connector

J7 is used at the factory for initial board calibration and test. It is not needed by the user except in rare instances such as if the stored calibration values have accidentally been overwritten, or if you want to verify that the analog input circuit is functioning properly.

The 3 positions are marked G, M, and V. G is analog ground, M is the input signal from the output of the analog multiplexors (the currently selected input channel), and V is the output of the programmable gain amplifier (the voltage at M times the selected gain). Any gain and offset errors between M and V are corrected by the autocalibration circuit between this point and the A/D converter.

4. DIAMOND-MM-AT BOARD DRAWING



5. I/O MAP

5.1 Overview

Diamond-MM-AT occupies 16 bytes in I/O space. A functional list of registers is provided below, and detailed bit definitions are provided on the next page and the following chapter.

Base +	Write Function	Read Function
0	Start A/D conversion	A/D LSB + channel tag
1	Reset FIFO	A/D MSB
2	A/D channel register	A/D channel register
3	Digital output port	Digital input port
4	D/A 0 LSB	Not used
5	D/A 0 MSB + update	Not used
6	D/A 1 LSB	Not used
7	D/A 1 MSB + update	Not used
8	Clear interrupt flip flop	Status register
9	A/D operation control register	A/D operation register readback
10	Ctr/Timer and FIFO Control Register	Ctr/Timer / FIFO register readback
11	Analog Configuration Register	Analog and FIFO register readback

Addresses 12-15 form a window into 2 4-byte pages.

Page 0: 82C54 counter/timer

12	Counter/timer 0 data register	Counter/timer 0 data register
13	Counter/timer 1 data register	Counter/timer 1 data register
14	Counter/timer 2 data register	Counter/timer 2 data register
15	Counter/timer control register	Counter/timer control register

Page 1: Calibration

12	EEPROM / TrimDAC data register	EEPROM / TrimDAC data register
13	EEPROM / TrimDAC address register	EEPROM / TrimDAC address register
14	Calibration control register	Calibration status register
15	Not used	Not used

5.2 Register Map Bit Assignments

WRITE operations (blank bits are unused and have no effect)

	7	6	5	4	3	2	1	0
0	Start A/D Conversion							
1	Reset FIFO							
2	HIGH3	HIGH2	HIGH1	HIGH0	LOW3	LOW2	LOW1	LOW0
3	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
4	DA0-7	DA0-6	DA0-5	DA0-4	DA0-3	DA0-2	DA0-1	DA0-0
5	SUD				DA0-11	DA0-10	DA0-9	DA0-8
6	DA1-7	DA1-6	DA1-5	DA1-4	DA1-3	DA1-2	DA1-1	DA1-0
7					DA1-11	DA1-10	DA1-9	DA1-8
8	Clear Interrupt Request Flip Flop							
9	AINTE				TINTE	DMAEN	CLKEN	CLKSEL
10	PAGE		FIFOEN	SCANEN	CLKFRQ	C2	C1	C0
11					RANGE	ADBU	G1	G0
12	Page 0: 82C54 Counter 0 Page 1: Calibration Data							
13	Page 0: 82C54 Counter 1 Page 1: Calibration Address							
14	Page 0: 82C54 Counter 2 Page 1: Calibration Control							
15	Page 0: 82C54 Control Register Page 1: EEPROM access key register							

READ operations (blank bits are unused and read back as 0)

	7	6	5	4	3	2	1	0
0	AD3	AD2	AD1	AD0	ADCH3	ADCH2	ADCH1	ADCH0
1	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4
2	HIGH3	HIGH2	HIGH1	HIGH0	LOW3	LOW2	LOW1	LOW0
3	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
4	Not Used							
5	Not Used							
6	Not Used							
7	Not Used							
8	STS	TINT	SD	AINT	CH3	CH2	CH1	CH0
9	AINTE				TINTE	DMAEN	CLKEN	CLKSEL
10	PAGE		FIFOEN	SCANEN	CLKFRQ	C2	C1	C0
11	WAIT	OVF	HF	EF	RANGE	ADBU	G1	G0
12	Page 0: 82C54 Counter 0 Page 1: Calibration Data							
13	Page 0: 82C54 Counter 1 Page 1: Calibration Address							
14	Page 0: 82C54 Counter 2 Page 1: Calibration Status							
15	Page 0: 82C54 Control Register Page 1: FPGA revision code							

6. REGISTER DEFINITIONS

In the register maps below, any blank bit location is unused. Writing to it has no effect, and reading it returns a 0.

Base + 0 Read A/D LSB + Channel Tag

Bit No.	7	6	5	4	3	2	1	0
Name	AD3	AD2	AD1	AD0	CH3	CH2	CH1	CH0

Definitions:

AD3 - 0 A/D data bits 3 - 0; AD0 is the LSB; A/D data is an unsigned 12-bit value.

CH3 - 0 A/D channel data, 0 - 15 (0 - 7 in differential mode)

Base + 0 Write Start A/D Conversion

Writing to Base + 0 starts an A/D conversion. The value written does not matter.

Base + 1 Read A/D MSB

Bit No.	7	6	5	4	3	2	1	0
Name	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4

Definitions:

AD11 - 4 A/D data bits 11 - 4; AD11 is the MSB; A/D data is an unsigned 12-bit value.

Note: Reading from Base + 0 and Base + 1 result in the same physical operation, reading from the FIFO.

The FIFO is 8 bits wide, with A/D data stored and retrieved in interleaved fashion. Data from the A/D is put into the FIFO in little-endian mode, with the LSB and channel tag inserted first, and the MSB inserted second. Thus the data comes out of the FIFO in the same order. Each time a byte is read from either Base + 0 or Base + 1, the next byte will be read from the FIFO and the FIFO counter will be decremented.

Because the FIFO decrements after each read operation, you cannot read the same value more than once (unless the FIFO is empty, in which case the last byte may be read indefinitely). It is the programmer’s responsibility to ensure that data is read out of the FIFO properly so that data underrun does not occur from extra read operations.

Base + 1 Write Reset FIFO

Writing to Base + 1 resets the FIFO. The value written does not matter. After writing to this address, EF = 1, HF = 0, and OVF = 0.

Base + 2 Read/Write A/D Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	HIGH3	HIGH2	HIGH1	HIGH0	LOW3	LOW2	LOW1	LOW0

Definitions:

HIGH3 - 0 High channel of channel scan range

Ranges from 0 to 15 in single-ended mode, 0 - 7 in differential mode.

LOW3 - 0 Low channel of channel scan range

Ranges from 0 to 15 in single-ended mode, 0 - 7 in differential mode.

The high channel must be greater than or equal to the low channel.

When this register is written, the current A/D channel is set to the low channel. The WAIT bit (base + 11 bit 7) goes high for 10µS to indicate that the analog circuit is settling on the new input value. Do not start a new A/D conversion until WAIT returns to 0.

Base + 3 Read Digital Input Port

Bit No.	7	6	5	4	3	2	1	0
Name	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

These signals correspond directly to the same-named pins on I/O connector J3.

All lines are connected to 10KΩ pull-up resistors.

Base + 3 Write Digital Output Port

Bit No.	7	6	5	4	3	2	1	0
Name	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0

These pins correspond directly to the same-named pins on I/O connector J3.

On power-up or reset, the output register is cleared to all zeroes.

**Base + 4	Write	DAC 0 LSB**

Bit No.	7	6	5	4	3	2	1	0
Name	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Definitions:

DA7 - 0	D/A data bits 7 - 0 for output channel 0; DA0 is the LSB; D/A data is an unsigned 12-bit value.

**Base + 5	Write	DAC 0 MSB**

Bit No.	7	6	5	4	3	2	1	0
Name	SUD				DA11	DA10	DA9	DA8

SUD	Simultaneous Update

1	Do not update D/A (hold off update of DAC0 until DAC1 is loaded)

0	Normal operation. Update D/A when MSB is written

DA11 - 8	D/A bits 11 - 8 for output channel 0; DA11 is the MSB; D/A data is an unsigned 12-bit value.

**Base + 6	Write	DAC 1 LSB**

Bit No.	7	6	5	4	3	2	1	0
Name	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Definitions:

DA7 - 0	D/A data bits 7 - 0 for output channel 1; DA0 is the LSB; D/A data is an unsigned 12-bit value.

**Base + 7	Write	DAC 1 MSB**

Bit No.	7	6	5	4	3	2	1	0
Name					DA11	DA10	DA9	DA8

DA11 - 8	D/A bits 11 - 8 for output channel 1; DA11 is the MSB; D/A data is an unsigned 12-bit value.

When this register is written, DAC 1 will be updated. If SUD is set in Base + 5, DAC 0 will also be updated.

Base + 8 Write Clear Interrupt Request Flip Flop

Writing to this register clears the on-board interrupt flip flop. The value written does not matter.

The interrupt flip flop is set whenever an interrupt is generated on Diamond-MM-AT (i.e. during A/D conversions), and it must be cleared by software before another interrupt can be generated. Diamond-MM-AT's software driver includes an interrupt handler that performs this task automatically.

Base + 8 Read Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	STS	TINT	SD	AINT	ADCH3	ADCH2	ADCH1	ADCH0

STS A/D chip status:

1 A/D conversion or scan is in progress

0 A/D is idle

TINT Timer interrupt request status:

1 Interrupt request has been generated by timer 0

0 No interrupt is pending from timer 0

SD Single-ended / Differential A/D input mode setting (readback of jumper setting):

1 Single-ended (default)

0 Differential

AINT Analog interrupt request status:

1 Interrupt request is pending from A/D circuit

0 No interrupt is pending from A/D circuit

ADCH3 - 0 Current A/D channel; this is the channel currently selected on board and is the channel that will be used for the next A/D conversion (unless a new value is written to the channel register before then). This data is not necessarily the same as the channel no. in the LSB of the next A/D data from the board.

Base + 9 Read/Write Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	AINTE				TINTE	DMAEN	CLKEN	CLKSEL

AINTE Analog interrupt enable:

1 Enable A/D interrupts

0 Disable A/D interrupts

TINTE Timer interrupt enable:

1 Enable interrupts from timer 0

0 Disable interrupts from timer 0

DMAEN DMA enable (DMA operation is explained later in this manual):

1 Enable DMA operation

0 Disable DMA operation

CLKEN Enable hardware A/D clock:

1 Enable hardware trigger (source is selected with CLKSEL bit)

0 Disable hardware trigger

CLKSEL A/D clock select, used only when CLKEN = 1:

1 Internal trigger: on-board counter/timer (82C54) generates A/D conversions

0 External trigger: DIN0, pin 48 on I/O connector J3, generates A/D conversions

Base + 10 Read/Write Counter/Timer and FIFO Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	PAGE		FIFOEN	SCANEN	CLKFRQ	C2	C1	C0

PAGE Page number for registers at Base + 12 through Base + 15

Page 0: 82C54 counter/timer access (normal operation)

Page 1: Calibration registers (not intended for direct access by the user)

FIFOEN FIFO enable:

1 Enable FIFO operation; if interrupts are enabled, interrupts will occur when the FIFO is half full (HF = 1). This slows down the interrupt rate dramatically compared to the actual A/D sample rate.

0 Disable FIFO operation; if interrupts are enabled, interrupts will occur after each A/D conversion.

SCANEN Scan enable:

1 Scan mode enabled; FIFO will fill up with data for a single scan, and STS will stay high until an entire scan is complete; if interrupts are enabled, interrupts will occur on integral multiples of scans.

0 Scan mode disabled; The STS bit will correspond directly to the status indicator from the A/D converter

CLKFRQ Clock frequency for counter/timer 1 input

0 10MHz

1 1MHz

C2 External Clock Gate Enable

1 IN0- (pin 29 on the I/O header J3) acts as a gate for A/D sample control when external A/D clock is enabled (CLKSEL = 0). When IN0- is high, falling edges on DI0 (pin 48 on J3) will initiate A/D conversions. When IN0- is low, the DI0 signal is inhibited. IN0- is connected to a 10KΩ pull-up resistor.

0 IN0- does not act as a gate for external A/D clocking.

C1 Counter 0 input source:

1 Input is a 100kHz on-board reference frequency derived from the 10MHz oscillator. IN0- (pin 29 on the I/O header) gates this signal. When it is high (default), the 100kHz signal runs. When it is low, the 100kHz signal is stopped.

0 Input is IN0- (pin 29 on the I/O connector) inverted. Counter 0 counts on rising edges of IN0-. IN0- is connected to an on-board pull-up resistor, so only a connection to ground is necessary to toggle it.

C0 Counters 1 and 2 gate control:

1 Counters 1 and 2 are gated by DIN0 (pin 48 on the I/O connector). When DIN0 is high, the counters run; when DI0 is low, the counters are stopped. In this way pin 48 can be used as an A/D conversion gate signal when the counters are used for A/D conversion timing.

0 Counters 1 and 2 run freely with no gating.

Base + 11 Write Analog Configuration Control Register

Bit No.	7	6	5	4	3	2	1	0
Name					RANGE	ADBU	G1	G0

RANGE 5V or 10V A/D positive full-scale range

ADBU A/D bipolar / unipolar setting; 0 = bipolar, 1 = unipolar

The table below lists the effects of the various combinations of ADBU and RANGE:

RANGE	ADBU	A/D full-scale range
0	0	±5V
0	1	0 - 10V
1	0	±10V
1	1	Invalid setting

G1 - 0 A/D gain setting:

G1	G0	Gain
0	0	1
0	1	2
1	0	4
1	1	8

The gain setting is the ratio between the A/D full-scale range and the effective input signal range. Gain may be thought of as either amplifying the input signal to fit the A/D input range better, or dividing the A/D input range to create a smaller full-scale input range. For example, if the A/D input range is 0 - 10V, a gain setting of 2 creates an input signal range of 0 - 5V, and a gain setting of 4 creates an input signal range of 0 - 2.5V.

⇒: On power up or system reset, the board is configured for A/D bipolar mode, input range ±5V, gain = 1 (all bits set to 0).

⇒: Not all combinations of bits are valid. Please see the complete list of bit settings and corresponding input ranges on page 22.

Base + 11 Read Analog and Status Readback Register

Bit No.	7	6	5	4	3	2	1	0
Name	WAIT	OVF	HF	EF	RANGE	ADBU	G1	G0

This address provides a means of reading back the values written to the control register at Base + 11 along with FIFO and scan status signals.

WAIT Indicates that the analog input circuit is settling on a new value. This signal goes high whenever the user writes to register 2 (channel address register) or register 11 (analog configuration register). The duration of the WAIT period is 10uS, which is the settling time required for the analog input circuit.

OVF FIFO overflow flag; 0 = no overflow; 1 = overflow

HF FIFO half full flag; 0 = FIFO is less than half full; 1 = FIFO is at least half full

EF FIFO empty flag; 0 = FIFO is not empty; 1 = FIFO is empty

PAGE 0: 82C54 Counter/Timer

Base + 12 through Base + 15 Read/Write 82C54 Counter/Timer Registers

These registers map directly to the on-board 82C54 counter/timer IC used for A/D conversion timing and user functions. The definitions of these registers can be found in the 82C54 datasheet appended to the back of this manual.

Page 1: Calibration Control Registers

These register descriptions are provided for completeness. In most applications the programmer does not need to access them directly. Erroneous use of these registers can result in putting the board out of calibration.

Base + 12 Read/Write EEPROM / TrimDAC Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0

D7-0 Calibration data to be read or written to the EEPROM and/or TrimDAC.

During EEPROM or TrimDAC write operations, the data written to this register will be written to the selected device.

During EEPROM read operations this register contains the data to be read from the EEPROM and is valid after EEBUSY = 0 (bit 5 in register 14).

The TrimDAC data cannot be read back.

Base + 13 Read/Write EEPROM / TrimDAC Address Register

Bit No.	7	6	5	4	3	2	1	0
Name		A6	A5	A4	A3	A2	A1	A0

A6-A0 EEPROM / TrimDAC address. The EEPROM only recognizes address 0 – 127. The TrimDAC only recognizes addresses 0 – 7. Writing to an address beyond the recognized range will result in only the significant bits being accepted. For example, when accessing the EEPROM, selecting address 129 is the same as selecting address 1 (dropping address bit 7).

Base + 14 Write Calibration Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	EE_EN	EE_RW	LOADCAL	CMUXEN	TDACEN			

This register is used to access the EEPROM and TrimDAC during board calibration.

Do not access this register unless you have a clear understanding of what you are doing, as you may destroy calibration values and prevent the board from being autocalibrated without user intervention. If you have any questions please contact Technical Support first at support@diamondsystems.com.

CMUXEN Calibration multiplexor enable:

1 = enable cal mux and disable regular input muxes

0 = disable cal mux

When CMUXEN = 1, the analog input channels 0 – 15 are disabled, and the calibration multiplexor channels 0 – 7 may be read as analog input channels 16 – 23. The cal mux input channels are connected to low-drift analog voltages that are used to calibrate the analog input circuit. Two of the input channels are the two DACs, so they may be calibrated as well.

EE_EN EEPROM Enable. When this bit is set, a Read/Write command to the EEPROM will be processed. Read or Write is indicated by the EE_RW bit. Once this bit is set, the program must monitor EEBUSY (see next page)and wait until it is 0.

EE_RW Determines the direction of the EEPROM command. 0 = Write, 1 = Read.

LOADCAL Load Calibration. Setting this bit causes the EEPROM values in locations 0 – 7 to be transferred to the TrimDAC (forces a reloading of the default calibration values).

TDACEN TrimDAC Enable – writes the data in register 12 to the TrimDAC no. indicated by A2 – A0 in register 13. Data may not be read from the TrimDACs, so there is no read/write bit for this operation. Once this bit is set, the program must monitor TDBUSY (see next page)and wait until it is 0.

Note: EE_EN and TDACEN must not be set at the same time.

Base + 14 Read Calibration Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	0	TDBUSY	EEBUSY	CMUXEN	0	0	0	0

TDBUSY TrimDAC busy indicator

0	User may access TrimDAC or perform another operation
1	TrimDAC is being accessed; user must wait

EEBUSY EEPROM busy indicator

0	EEPROM is idle; user may access it or perform another operation
1	EEPROM access cycle is in progress; user must wait

The TrimDAC requires about 0.5ms to execute a single complete read or write cycle.

Base + 15 Write EEPROM Access Key Register

The user must write the value 0xA5 (binary 10100101) to this register in order to get access to the EEPROM. This helps prevent accidental corruption of the EEPROM contents. This register must be written to each time the PAGE bit is set to 1, as it is cleared when the PAGE bit is set to 0.

Base + 15 Read FPGA Revision Code

This register may be read back to indicate the revision level of the FPGA design.

7. ANALOG INPUT RANGES AND RESOLUTION

7.1 Resolution

Diamond-MM-AT uses a 12-bit A/D converter. This means that the analog input voltage can be measured to the precision of a 12-bit binary number. The maximum value of a 12-bit binary number is 2^{12} - 1, or 4095, so the full range of numerical values that you can get from a Diamond-MM-AT analog input channel is 0 - 4095.

The smallest change in input voltage that can be detected is $1/(2^{12})$, or 1/4096, of the full-scale input range. This smallest change results in an increase or decrease of 1 in the A/D code, and so this change is referred to as 1 LSB, or 1 least significant bit.

7.2 Unipolar and Bipolar Inputs

Diamond-MM-AT can measure both unipolar (positive only) and bipolar (positive and negative) analog voltages. In general you should select the highest gain you can that will allow the A/D converter to read the full range of voltages over which your input signals will vary. However, if you pick too high a gain, then the A/D converter will clip at either the high end or low end, and you will not be able to read the full range of voltages on your input signals.

7.3 Single Ended and Differential Inputs

Diamond-MM-AT can handle both single-ended and differential inputs. A single-ended input is a single-wire input that is referenced to analog ground on the board. This means that the input voltage will be measured with respect to the board’s analog ground. A differential input is a two-wire input, and the board will measure the difference between the voltages of the two inputs. Polarity is important for a differential input. Diamond-MM-AT will subtract the voltage on the low input (- pin) from the voltage of the high input (+ pin). Differential inputs are frequently used when the grounds of the input device and the measurement device (Diamond-MM-AT) are at different voltages, or when a low-level signal is being measured that has its own ground wire.

8. ANALOG INPUT RANGE SETTINGS

The table below lists all the valid analog input ranges on Diamond-MM-AT. Although there are 16 possible combinations for the 4 range control bits, only 12 configurations are valid due to the configuration of the A/D converter.

There is some overlap in the meaning of the bit combinations, so the total number of unique input ranges is 9.

The control bits RANGE, ADBU, G1, and G0 are in the control register at Base + 11.

Diamond-MM-AT Analog Input Ranges

Range No.	RANGE	ADBU	G1	G0	Input Range	1 LSB
0	0	0	0	0	±5V	2.44mV
1	0	0	0	1	±2.5V	1.22mV
2	0	0	1	0	±1.25V	0.61mV
3	0	0	1	1	±0.625V	0.305mV
4	0	1	0	0	0-10V	2.44mV
5	0	1	0	1	0-5V	1.22mV
6	0	1	1	0	0-2.5V	0.61mV
7	0	1	1	1	0-1.25V	0.305mV
8	1	0	0	0	±10V	4.88mV
9	1	0	0	1	±5V	2.44mV
10	1	0	1	0	±2.5V	1.22mV
11	1	0	1	1	±1.25V	0.61mV
12	1	1	0	0	INVALID	
13	1	1	0	1	INVALID	
14	1	1	1	0	INVALID	
15	1	1	1	1	INVALID	

9. PERFORMING AN A/D CONVERSION

This chapter describes the steps involved in performing an A/D conversion on a selected input channel using direct programming (not with the driver software).

There are five steps involved in performing an A/D conversion:

1. Select the input channel
2. Select the input range (if a new range is desired)
3. Perform an A/D conversion on the current channel
3. Wait for the conversion to finish
4. Read the data
5. Convert the numerical data to a meaningful value

9.1 Select the input channel

To select the input channel to read, write a low-channel/high-channel pair to the channel register at Base + 2. The low 4 bits select the low channel, and the high 4 bits select the high channel. When you write any value to this register, the current A/D channel is set to the low channel.

When you perform an A/D conversion, the current channel is automatically incremented to the next channel in the selected range. Therefore, to perform A/D conversions on a group of consecutively-numbered channels, you do not need to write the input channel prior to each conversion. For example, to read from channels 0 - 2, write Hex 20 to Base + 2. The first conversion is on channel 0, the second will be on channel 1, and the third will be on channel 2. Then the channel counter wraps around to the beginning again, so the fourth conversion will be on channel 0 again. This sequence will repeat until you change the settings at Base + 2 or reset the board.

About 10 microseconds is required as a settling time for the analog front end circuitry after you select the channel. If you are not selecting a new analog input range, you must wait for this settling time by monitoring the WAIT bit before starting the A/D conversion. Wait for WAIT = 0 before proceeding. If you are selecting a new input range, you may do this before waiting.

9.2 Select the input range

To select a new input range, write the desired setting to the register at Base + 11 (see page 17). If you want to use the same range as the previous A/D conversion, you can skip this step.

About 10 microseconds is required as a settling time for the analog front end circuitry after you select the input range. Monitor the WAIT bit (Base + 11 bit 7) before starting the A/D conversion. Wait for WAIT = 0 before proceeding.

Valid input channel / range sequences:

A. Select new input channel settings and new input range

Select input channel range with Base + 2; channel is set to low channel no.

Select input range with Base + 11

Wait for WAIT = 0 in Base + 11

B. Select new input channel, use same input range

Select input channel range with Base + 2; channel is set to low channel no.

Wait for WAIT = 0 in Base + 11

C. Use current settings with autoincrement of input channel

<No steps required>

9.3 Perform an A/D conversion on the current channel

After writing to the channel register and waiting for the analog circuitry to settle, you can perform an A/D conversion on the selected channel. To do this, simply write to base + 0 to start the conversion. The data value does not matter and is ignored.

9.4 Wait for the conversion to finish

The A/D converter takes up to 10 microseconds to complete a conversion. Most processors and software can operate fast enough so that if you try to read the A/D converter immediately after writing to base + 0, you will beat the A/D converter and get invalid data. Therefore the A/D converter provides a status signal STS to indicate whether it is busy or idle. This bit can be read back as bit 7 in the status register at Base + 8. When the A/D converter is busy (performing an A/D conversion), this bit is 1, and when the A/D converter is idle (conversion is done and data is available), this bit is 0. You must wait for STS = 0 before reading A/D data.

9.5 Read the data

Once the conversion is complete, you can read the data back from the A/D converter. The data is 12 bits wide and is read back in two 8-bit bytes. The LSB must be read first, followed by the MSB. Refer to the register definitions on p. 8 for the format of the A/D data. The following pseudocode illustrates how to construct the 12-bit A/D value from these two bytes:

```
LSB = read(base) / 16      ;shift LSB right 4 bits, delete channel #
MSB = read(base+1) * 16    ;shift MSB left 4 bits
Data = MSB + LSB           ;combine the 2 bytes into a 12-bit value
```

The final data ranges from 0 to 4095 (0 to 2^{12} - 1).

A/D data before manipulation:

Base + 1								Base + 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	CH3	CH2	CH1	CH0

A/D data after manipulation:

MSB								LSB							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	0	0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

9.6 Convert the numerical data to a meaningful value

Once you have the A/D value, you need to convert it to a meaningful value. The first step is to convert it back to the actual measured voltage. Afterwards you may need to convert the voltage to some other engineering units (for example, the voltage may come from a temperature sensor, and then you would need to convert the voltage to the corresponding temperature according to the temperature sensor's characteristics). Since there are a large number of possible input devices, this secondary step is not included here; only conversion to input voltage is described.

To convert the A/D value to the corresponding input voltage, use the following formulas:

Conversion Formula for Unipolar Input Ranges:

Input voltage = (A/D value / 4096) * Full-scale input range

Example: Input range is 0-5V and A/D value is 1776:
Input voltage = 1776 / 4096 * 5V = 2.168V

Here is an illustration of the relationship between A/D code and input voltage for a unipolar input range (V_{FS} = Full scale input voltage):

A/D Code	Input voltage
0	0V
1	1 LSB (V_{FS} / 4096)
2048	V_{FS} / 2
2049	V_{FS} / 2 + 1 LSB
4095	V_{FS} - 1 LSB (e.g. 4.9988V for 0-5V range)

Conversion Formula for Bipolar Input Ranges:

Input voltage = (A/D value/2048) * Full-scale input range - Full-scale input range

Example: Input range is ±5V and A/D value is 1776:
Input voltage = 1776 / 2048 * 5V - 5V = -0.664V

Here is an illustration of the relationship between A/D code and input voltage for a bipolar input range (V_{FS} = Full scale input voltage):

A/D Code	Input voltage
0	-V_{FS}
1	-V_{FS} + 1 LSB
2047	-1 LSB
2048	0V
2049	+1 LSB
4095	V_{FS} - 1 LSB (e.g. 4.9976V for ±5V range)

10. A/D SCAN, INTERRUPT, AND FIFO OPERATION

Three control bits determine the behavior of DMM-AT during A/D conversions.

SCANEN (scan enable, Base + 10 bit 4): A/D conversions may be performed one at a time or in scan mode. In non-scan mode, each A/D start command (software or external clock) causes the current channel to be sampled, and the internal channel register to increment to the next channel in the selected range. In scan mode, each A/D start command causes all the channels in the selected range to be sampled once in quick succession.

FIFOEN (FIFO enable, Base + 10 bit 5): The FIFO affects interrupt behavior. If the FIFO is not enabled, then each A/D start command will result in an interrupt at the end of the sample or scan. If the FIFO is enabled, then the interrupt will occur once the FIFO reaches its threshold (half full, or 256 samples). If interrupts are not enabled, then FIFOEN has no effect on the board's behavior. In all cases, at the end of an A/D conversion A/D data is latched into the FIFO. Data is read out of the FIFO with 2 read operations, the LSB and channel tag from Base + 0 and the MSB from Base + 1. This is true even when FIFOEN = 0.

AINTE (analog input interrupt enable, Base + 9 bit 7): This bit enables interrupt operation. The board will generate an interrupt when the appropriate set of conditions occur based on the SCANEN and FIFOEN bits. The interrupt routine (included in the Diamond Systems Universal Driver software or else provided by the programmer) reads A/D data out of the FIFO.

For all cases where AINTE = 1, CLKEN must also be set to 1 to enable hardware A/D clocking. This is because when interrupt operation is enabled, A/D conversions cannot be triggered by software and must be triggered by a hardware source.

AINTE	FIFOEN	SCANEN	Operation
0	0	0	Single A/D conversions are triggered by write to B+0. WAIT = STS. No interrupt occurs. The user program monitors STS or WAIT and reads A/D data when it goes low.
0	0	1	A/D Scans are triggered by write to B+0. All channels between LOW and HIGH will be sampled. WAIT goes high at the first STS high pulse and stays high until the last STS pulse goes low. No interrupt occurs. The user program monitors WAIT and reads the A/D data after it goes low.
0	1	0	Same operation as case 000 above.
0	1	1	Same operation as case 001 above.
1	0	0	Single A/D conversions are triggered by either TM_OUT2 or DIN0, depending on the value of INTTRIG. WAIT = STS. INT goes high after each conversion is done (when STS goes low). The interrupt routine reads one A/D sample each time it runs.
1	0	1	A/D scans are triggered by either TM_OUT2 or DI0, depending on the value of INTTRIG. WAIT goes high at the first STS high pulse and stays high until the last STS pulse goes low. INT goes high after the last STS pulse goes low (i.e. when WAIT goes low). The interrupt routine reads out one entire A/D scan each time it runs.
1	1	0	Single A/D conversions are triggered by either TM_OUT2 or DIN0, depending on the value of INTTRIG. WAIT = STS. INT goes high when HF goes high. The interrupt routine reads out 256 samples (half the FIFO) each time it runs.
1	1	1	A/D scans are triggered by either TM_OUT2 or DI0, depending on the value of INTTRIG. WAIT goes high at the first STS high pulse and stays high until the last STS pulse goes low. INT goes high after the last STS pulse goes low (i.e. when WAIT goes low) AND HF is high (i.e. an integral no. of scans has occurred and the FIFO is half full or more). The interrupt routine reads out enough complete scans to equal 256 or more samples each time it runs.

11. ANALOG OUTPUT RANGES AND RESOLUTION

11.1 Description

Diamond-MM-AT uses a two-channel 12-bit D/A converter (DAC) to provide two analog outputs. A 12-bit DAC can generate output voltages with the precision of a 12-bit binary number. The maximum value of a 12-bit binary number is 2^{12} - 1, or 4095, so the full range of numerical values that you can write to the analog outputs on Diamond-MM-AT is 0 - 4095.

The DACs on Diamond-MM-AT operate in both unipolar and bipolar modes. The user can select from an on-board fixed 5V reference for 0-10V and ±5V ranges, or an on-board programmable reference for custom ranges. In addition two pins on the I/O connector enable the user to provide an external reference for the DACs.

In most cases each DAC can be set to a different range. The only exception is that if both DACs are using the programmable range, then both DACs must be set to the same polarity (unipolar or bipolar mode).

⇒ **Note:** In this manual, the terms analog output, D/A, and DAC are all used interchangeably to mean the same thing.

11.2 Resolution

The *resolution* is the smallest possible change in output voltage. For a 12-bit DAC the resolution is $1/(2^{12})$, or 1/4096, of the full-scale output range. This smallest change results from an increase or decrease of 1 in the D/A code, and so this change is referred to as 1 LSB, or 1 least significant bit. The value of this LSB is calculated as follows:

1 LSB = Full-scale output voltage / 4096

Example: Full-scale output voltage = 10V (for 0-10V or ±5V ranges)
1 LSB = 10V / 4096 = 2.44mV

11.3 Full-Scale Voltage Selection

Jumper block J5 in the middle of the board is used for DAC configuration. The left group of 5 pairs of pins controls DAC 0, and the right group of 5 pins controls DAC 1. Configuration is done according to the instructions below.

Position	Function
5	Selects 5V reference (if N jumper is installed)
P	Selects programmable reference (if N jumper is installed)
N	Selects internal reference, 5 or P above
X	Selects external reference, pin 26 for DAC 0 or pin 28 for DAC1
B	Selects bipolar mode if in, unipolar mode if out

For each DAC, one jumper must be installed in either N or X but not both.

For each DAC, if a jumper is installed in N, then one jumper must be installed in either 5 or P but not both.

If both DACs are set to P and N (internal programmable reference), then both DACs must have the same B jumper setting (in for both or out for both).

12. GENERATING AN ANALOG OUTPUT

This chapter describes the steps involved in generating an analog output (also called performing a D/A conversion) on a selected output channel using direct programming (not with the driver software).

There are two steps involved in performing a D/A conversion:

1. Compute the D/A output value for the desired output voltage
2. Write the value to the selected output channel

12.1 Compute the D/A code for the desired output voltage

Use the following formulas to compute the D/A code required to generate the desired voltage:

For Unipolar operation:

Output value = Desired output voltage / Full-scale voltage * 4096

Example: Desired output voltage = 2.168V, full-scale voltage = 10V (0-10V range)
Output code = 2.168 / 5 * 4096 = 1776

For Bipolar operation:

Output value = Desired output voltage / Full-scale voltage * 2048 + 2048

Example: Desired output voltage = -3.0V, full-scale voltage = 5V (±5V range)
Output code = -3.0 / 5 * 2048 + 2048 = 819 (rounded to nearest integer value)

⇒ **Note:** The DAC cannot generate the actual full-scale reference voltage; to do so would require an output code of 4096, which is not possible with a 12-bit number. The maximum output value is 4095. Therefore the maximum possible output voltage is 1 LSB less than the full-scale reference voltage.

12.2 Write the value to the selected output channel

The two DACs are located at addresses base + 4 through base + 7 (see Chapter 6, I/O Map). Each DAC uses one pair of addresses. First write the LSB to the lower address, then write the MSB to the upper address. Writing the MSB to a DAC causes that DAC to be updated, outputting the new voltage.

First use the following formulas to compute the LSB and MSB values:

LSB = D/A Code AND 255 ;keep only the low 8 bits

MSB = int(D/A code / 256) ;strip off low 8 bits, keep 4 high bits

Example: Output code = 1776
LSB = 1776 AND 255 = 240 (F0 Hex); MSB = int(1776 / 256) = int(6.9375) = 6
(In other words, 1776 = 6*256 + 240)

Note that the LSB does not actually need to be computed, since Diamond-MM-AT uses an 8-bit databus and will ignore the high byte if a program attempts to write a 16-bit word to it.

Now write these values to the selected channel:

For Channel 0: Write 240 to base + 4, then write 6 to base + 5, updating Channel 0

For Channel 1: Write 240 to base + 6, then write 6 to base + 7, updating Channel 1

13. DIGITAL I/O OPERATION

Diamond-MM-AT contains an 8-bit digital output port and an 8-bit digital input port. Both ports are located at base + 3. To access the output lines, simply write an 8-bit value to base + 3. Similarly, to read the input lines, read from base + 3.

The output lines are located at pins 33 through 40 on the I/O header J3 (see p. 4). They use a 74FCT273 chip. They do not have a readback feature, so your program must keep track of the latest output value.

The inputs are located at pins 41 through 48 on the I/O header J3. They are CMOS/TTL compatible. There is no latch signal provided. However, the values are latched when being read to prevent transitions during the CPU read operation. All lines have 10KΩ pull-up resistors.

Input line 2 doubles as the gate control for counter 0. When it is high, counter 0 can count, and when it is low, counter 0 holds its present value.

Input line 0 doubles as a programmable gate control for counters 1 and 2. These counters are combined together and used as the A/D pacer clock. Bit 0 of the counter/timer control register at base + 10 determines whether these counters run freely or whether Input line 0 is the gate (see Chapter 6, page 13).

Digital I/O Specifications:

No. of inputs	8, HCT/TTL compatible
Input voltage	Logic 0: 0.0V min, 0.8V max; Logic 1: 2.0V min, 5.0V max
No. of outputs	8, FCT/TTL compatible
Output voltage	Logic 0: 0.0V min, 0.33V max; Logic 1: 3.8V min, 5.0V max
Output current	Logic 0: +64mA max per line; Logic 1: -15mA max per line

14. SPECIFICATIONS

Analog Inputs

No. of inputs	8 differential or 16 single-ended (user selectable)
A/D resolution	12 bits (1/4096 of full scale)
Input ranges	Bipolar: ±10V, ±5V, ±2.5V, ±1V, ±0.5V, Custom Unipolar: 0-10V, 0-5V, 0-2.5V, 0-1V, Custom
Input bias current	50nA max
Maximum input voltage	±10V for linear operation
Overvoltage protection	±35V on any analog input without damage
Nonlinearity	±1LSB, no missing codes
Conversion rate	100,000 samples per second max (with DMA operation)
Conversion trigger	software trigger, internal pacer clock, or external TTL signal

Autocalibration

Circuits calibrated	A/D (all 9 input ranges) and D/A
A/D error after calibration	±1LSB (typical), ±2LSB (max)
D/A error after calibration	±1LSB (typical), ±2LSB (max)

Analog Outputs

No. of outputs	2
D/A resolution	12 bits (1/4096 of full scale)
Output ranges	0-10V, ±5V, adjustable, or external reference input
Output current	±8mA max per channel
Settling time	4µS max to ±1/2 LSB
Relative accuracy	±1 LSB
Nonlinearity	±1 LSB, monotonic

Digital I/O

No. of inputs	8, HCT/TTL compatible
Input voltage	Logic 0: 0.0V min, 0.8V max; Logic 1: 2.0V min, 5.0V max
No. of outputs	8, HCT/TTL compatible
Output voltage	Logic 0: 0.0V min, 0.33V max; Logic 1: 3.8V min, 5.0V max
Output current	Logic 0: +64mA max per line; Logic 1: -15mA max per line

Counter/Timers

A/D Pacer clock	32-bit down counter (2 82C54 counters cascaded)
Clock source	10MHz on-board clock source or external signal
General purpose	16-bit down counter (1 82C54 counter)

General

Power supply	+5VDC ±10%
Current consumption	220mA typical
I/O header output current	±15V: ±10mA max with DACs unloaded +5V: Depends on PC/104 power supply **Power supply outputs are not short-circuit protected!**
Operating temperature	-40 to +85^{o}C
Operating humidity	5% to 95% noncondensing
PC/104 bus	8 bits

HARRIS
SEMICONDUCTOR

82C54

March 1997

CMOS Programmable Interval Timer

Features

- **8MHz to 12MHz Clock Input Frequency**
- **Compatible with NMOS 8254**
  - **Enhanced Version of NMOS 8253**
- **Three Independent 16-Bit Counters**
- **Six Programmable Counter Modes**
- **Status Read Back Command**
- **Binary or BCD Counting**
- **Fully TTL Compatible**
- **Single 5V Power Supply**
- **Low Power**
  - **ICCSB 10μA**
  - **ICCOP . . . . 10mA at 8MHz**
- **Operating Temperature Ranges**
  - **C82C54 . . . . 0°C to +70°C**
  - **I82C54 . . . . -40°C to +85°C**
  - **M82C54 . . . . -55°C to +125°C**

Description

The Harris 82C54 is a high performance CMOS Programmable Interval Timer manufactured using an advanced 2 micron CMOS process.

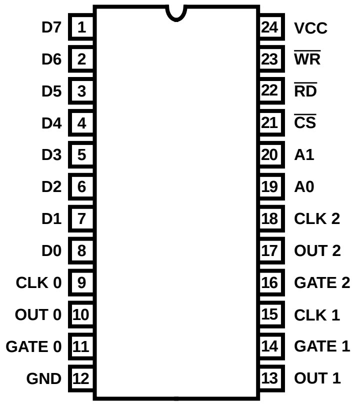
The 82C54 has three independently programmable and functional 16-bit counters, each capable of handling clock input frequencies of up to 8MHz (82C54) or 10MHz (82C54-10) or 12MHz (82C54-12).

The high speed and industry standard configuration of the 82C54 make it compatible with the Harris 80C86, 80C88, and 80C286 CMOS microprocessors along with many other industry standard processors. Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and many other applications. Static CMOS circuit design insures low power operation.

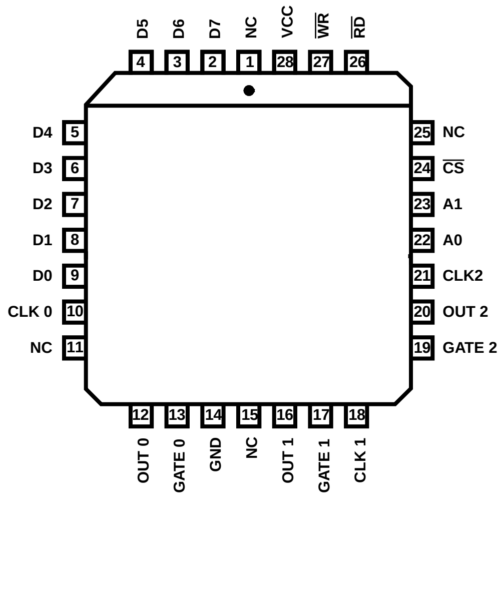
The Harris advanced CMOS process results in a significant reduction in power with performance equal to or greater than existing equivalent products.

Pinouts

82C54 (PDIP, CERDIP, SOIC)
TOP VIEW



82C54 (PLCC/CLCC)
TOP VIEW

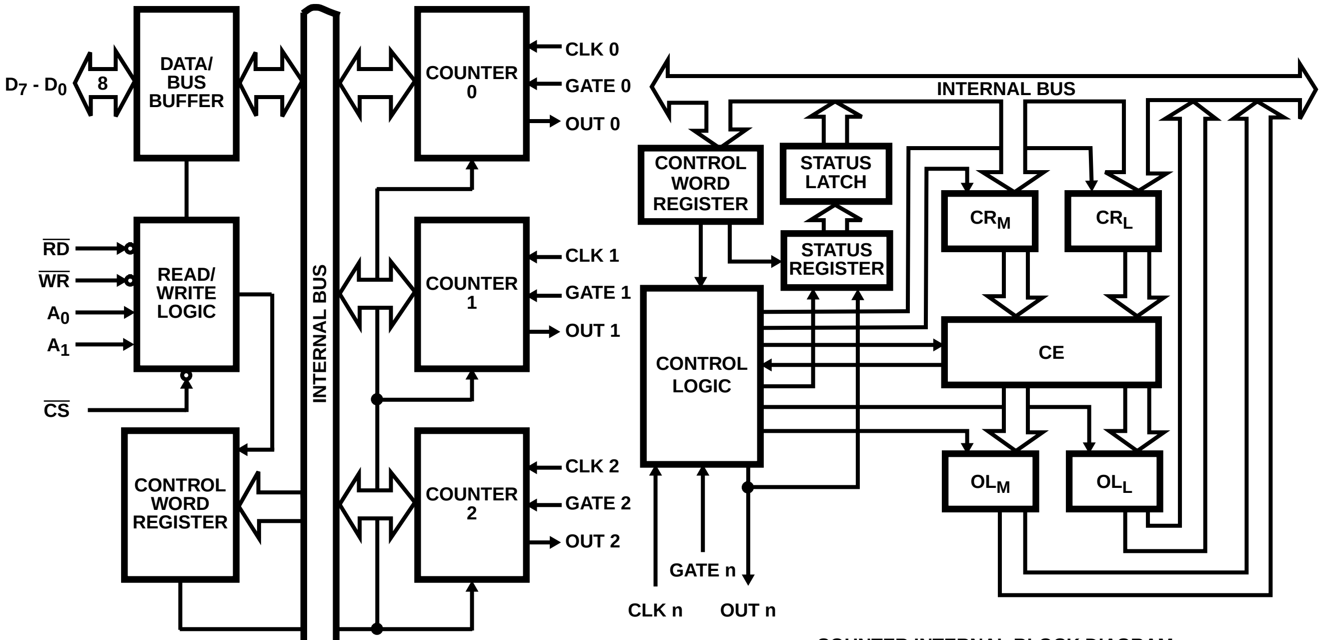


CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

Ordering Information

PART NUMBERS			TEMPERATURE RANGE	PACKAGE	PKG. NO.
8MHz	10MHz	12MHz			
CP82C54	CP82C54-10	CP82C54-12	0°C to +70°C	24 Lead PDIP	E24.6
IP82C54	IP82C54-10	IP82C54-12	-40°C to +85°C	24 Lead PDIP	E24.6
CS82C54	CS82C54-10	CS82C54-12	0°C to +70°C	28 Lead PLCC	N28.45
IS82C54	IS82C54-10	IS82C54-12	-40°C to +85°C	28 Lead PLCC	N28.45
CD82C54	CD82C54-10	CD82C54-12	0°C to +70°C	24 Lead CERDIP	F24.6
ID82C54	ID82C54-10	ID82C54-12	-40°C to +85°C	24 Lead CERDIP	F24.6
MD82C54/B	MD82C54-10/B	MD82C54-12/B	-55°C to +125°C	24 Lead CERDIP	F24.6
MR82C54/B	MR82C54-10/B	MR82C54-12/B	-55°C to +125°C	28 Lead CLCC	J28.A
SMD # 8406501JA	-	8406502JA	-55°C to +125°C	24 Lead CERDIP	F24.6
SMD# 84065013A	-	84065023A	-55°C to +125°C	28 Lead CLCC	J28.A
CM82C54	CM82C54-10	CM82C54-12	0°C to +70°C	24 Lead SOIC	M24.3

Functional Diagram



COUNTER INTERNAL BLOCK DIAGRAM

Pin Description

SYMBOL	DIP PIN NUMBER	TYPE	DEFINITION
D7 - D0	1 - 8	I/O	DATA: Bi-directional three-state data bus lines, connected to system data bus.
CLK 0	9	I	CLOCK 0: Clock input of Counter 0.
OUT 0	10	O	OUT 0: Output of Counter 0.
GATE 0	11	I	GATE 0: Gate input of Counter 0.
GND	12		GROUND: Power supply connection.
OUT 1	13	O	OUT 1: Output of Counter 1.
GATE 1	14	I	GATE 1: Gate input of Counter 1.
CLK 1	15	I	CLOCK 1: Clock input of Counter 1.
GATE 2	16	I	GATE 2: Gate input of Counter 2.
OUT 2	17	O	OUT 2: Output of Counter 2.

Pin Description (Continued)

SYMBOL	DIP PIN NUMBER	TYPE	DEFINITION
CLK 2	18	I	CLOCK 2: Clock input of Counter 2.
A0, A1	19 - 20	I	ADDRESS: Select inputs for one of the three counters or Control Word Register for read/write operations. Normally connected to the system address bus.
$\overline{CS}$	21	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	I	READ: This input is low during CPU read operations.
$\overline{WR}$	23	I	WRITE: This input is low during CPU write operations.
V_{CC}	24		V_{CC}: The +5V power supply pin. A 0.1μF capacitor between pins VCC and GND is recommended for decoupling.

A1	A0	SELECTS
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

Functional Description

General

The 82C54 is a programmable interval timer/counter designed for use with 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 computer/timer functions common to microcomputers which can be implemented with the 82C54 are:

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

Data Bus Buffer

This three-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 1).

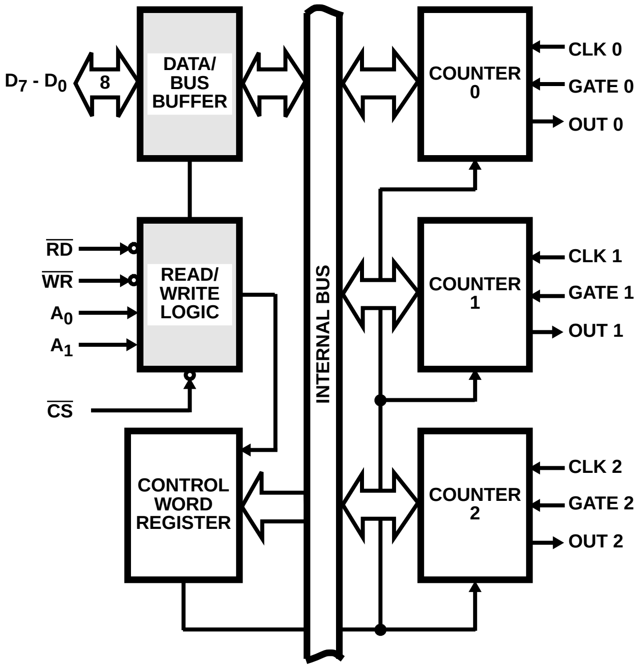


FIGURE 1. DATA BUS BUFFER AND READ/WRITE LOGIC FUNCTIONS

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. A1 and A0 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 (Figure 2) is selected by the Read/Write Logic when A1, A0 = 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 Counter operation.

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

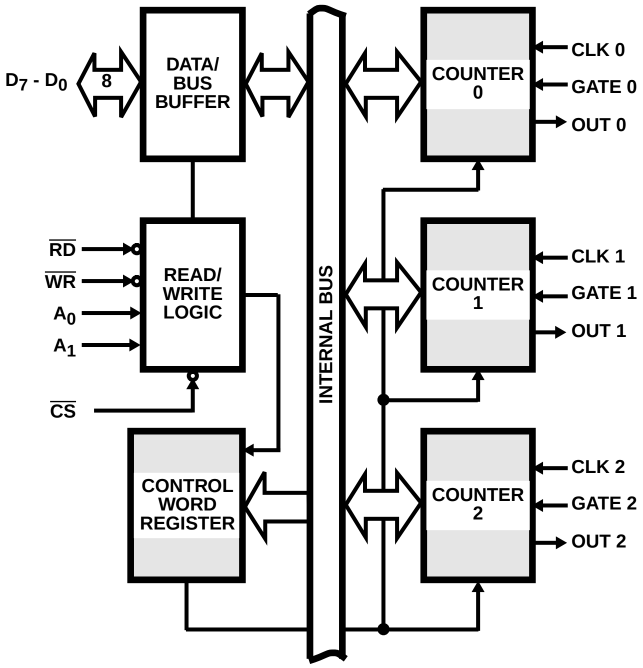


FIGURE 2. CONTROL WORD REGISTER AND COUNTER FUNCTIONS

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 signal counter is shown in Figure 3. 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.

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 labeled CE (for Counting Element). It is a 16-bit presettable synchronous down counter.

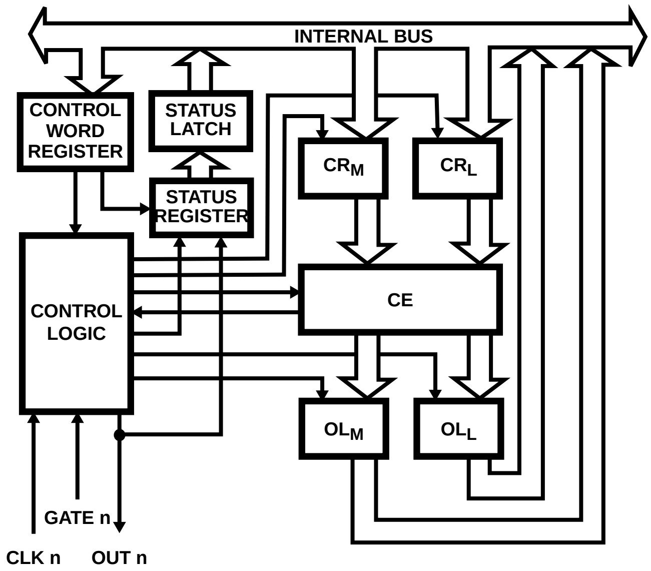


FIGURE 3. COUNTER INTERNAL BLOCK DIAGRAM

OLM and OLL are two 8-bit latches. OL stands for “Output Latch”; the subscripts M and L 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 CRM and CRL (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. CRM and CRL are cleared when the Counter is 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 system 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 A0, A1 connect to the A0, A1 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.

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.

All Control Words are written into the Control Word Register, which is selected when A1, A0 = 11. The Control Word specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A1, A0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

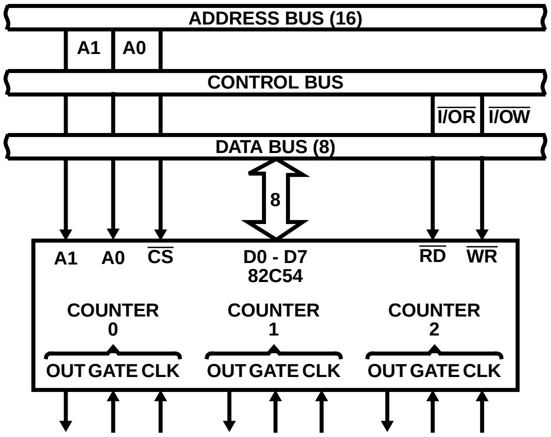


FIGURE 4. 82C54 SYSTEM INTERFACE

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 A1, A0 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.

Control Word Format

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

D7	D6	D5	D4	D3	D2	D1	D0
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.

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 - Binary Coded Decimal

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

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

Possible Programming Sequence

	A1	A0
Control Word - Counter 0	1	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0
Control Word - Counter 1	1	1
LSB of Count - Counter 1	0	1
MSB of Count - Counter 1	0	1
Control Word - Counter 2	1	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 2	1	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 0	1	1
Control Word - Counter 1	1	1
Control Word - Counter 2	1	1
LSB of Count - Counter 2	1	0

Possible Programming Sequence **(Continued)**

	A1	A0
LSB of Count - Counter 1	0	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0
MSB of Count - Counter 1	0	1
MSB of Count - Counter 2	1	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 2	1	1
Control Word - Counter 1	1	1
Control Word - Counter 0	1	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 2	1	0
LSB of Count - Counter 1	0	1
MSB of Count - Counter 1	0	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 1	1	1
Control Word - Counter 0	1	1
LSB of Count - Counter 1	0	1
Control Word - Counter 2	1	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 1	0	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 0	0	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 programming sequences.

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.

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. The first is through the Read-Back command, which is explained later. The second is a simple read operation of the Counter, which is selected with the A1, A0 inputs. The only requirement is that 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 process of changing when it is read, giving an undefined result.

Counter Latch Command

The other method for reading the Counters involves a special software command called the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A1, A0 = 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.

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

D7	D6	D5	D4	D3	D2	D1	D0
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 products.

The selected Counter's output latch (OL) latches the count when 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 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 or 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 read-back 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 5. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

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

D7	D6	D5	D4	D3	D2	D1	D0
1	1	$\overline{\text{COUNT}}$	$\overline{\text{STATUS}}$	CNT 2	CNT 1	CNT 0	0

D5: 0 = Latch count of selected Counter (s)
D4: 0 = Latch status of selected Counter(s)
D3: 1 = Select Counter 2
D2: 1 = Select Counter 1
D1: 1 = Select Counter 0
D0: Reserved for future expansion; Must be 0

FIGURE 5. READ-BACK COMMAND FORMAT

The read-back command may be used to latch multiple counter output latches (OL) by setting the COUNT bit D5 = 0 and selecting the desired counter(s). This signal 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 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 6. 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.

D7	D6	D5	D4	D3	D2	D1	D0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D7: 1 = Out pin is 1
0 = Out pin is 0
D6: 1 = Null count
0 = Count available for reading
D5 - D0 = Counter programmed mode (See Control Word Formats)

FIGURE 6. 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 counter 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 below.

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.

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.

COMMANDS								DESCRIPTION	RESULT
D7	D6	D5	D4	D3	D2	D1	D0		
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 7. READ-BACK COMMAND EXAMPLE

Both count and status of the selected counter(s) may be latched simultaneously by setting both COUNT and 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 7.

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.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	
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 (Three-State)
1	X	X	X	X	No-Operation (Three-State)
0	1	1	X	X	No-Operation (Three-State)

FIGURE 8. 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 (See "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 to 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 disables 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.

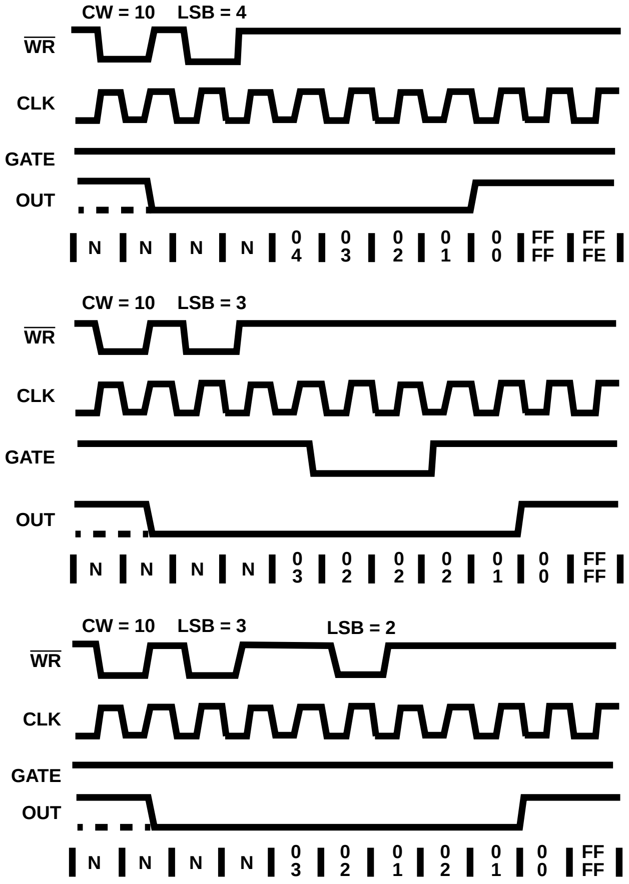

CW = 10 LSB = 4
WR
CLK
GATE
OUT
N N N N 0 4 0 3 0 2 0 1 0 0 FF FF FF FE
CW = 10 LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 2 0 2 0 1 0 0 FF FF
CW = 10 LSB = 3
LSB = 2
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 2 0 1 0 0 FF FF

FIGURE 9. MODE 0

NOTES: 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.
6. N stands for an undefined count.
7. Vertical lines show transitions between count values.

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 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 retriggerable. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

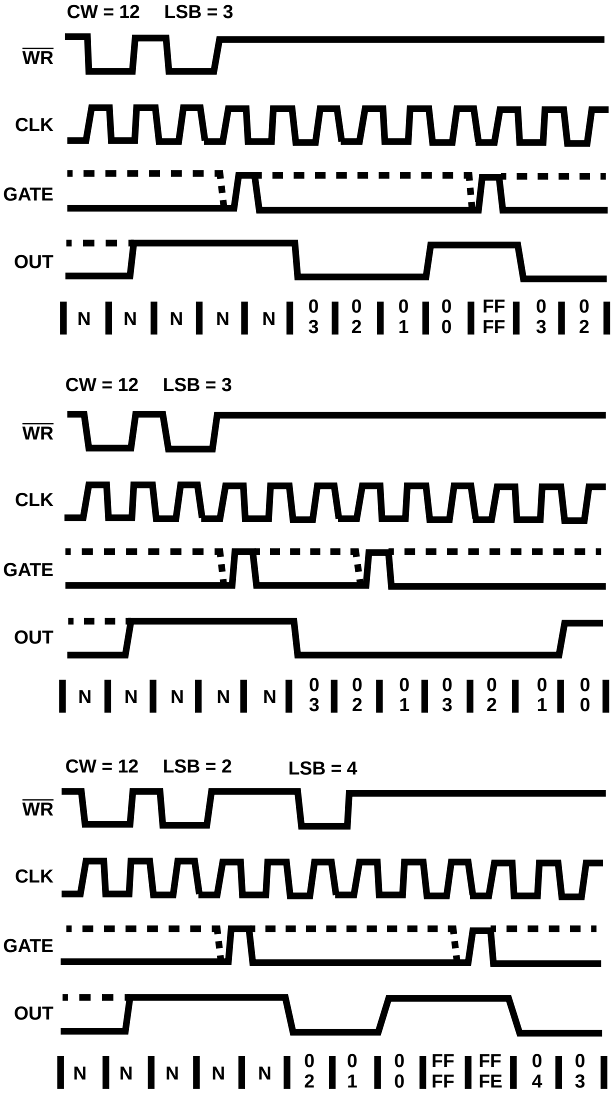

CW = 12 LSB = 3
WR
CLK
GATE
OUT
N N N N N 0 3 0 2 0 1 0 0 FF FF 0 3 0 2
CW = 12 LSB = 3
WR
CLK
GATE
OUT
N N N N N 0 3 0 2 0 1 0 3 0 2 0 1 0 0
CW = 12 LSB = 2 LSB = 4
WR
CLK
GATE
OUT
N N N N N 0 2 0 1 0 0 FF FF FF FE 0 4 0 3

FIGURE 10. MODE 1

Mode 2: Rate Generator

This Mode functions like a divide-by-N counter. It is typically 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.

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 end of the current counting cycle.

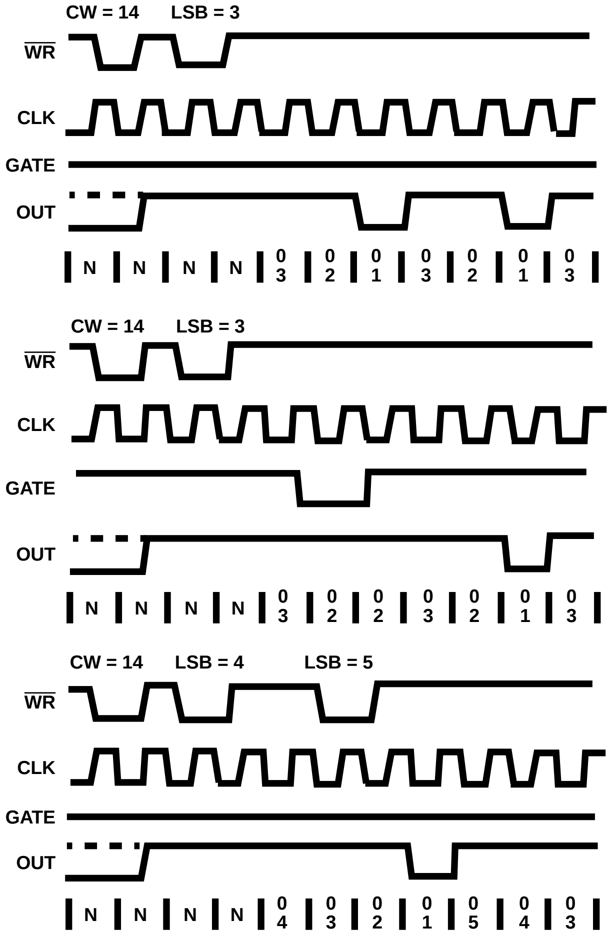

CW = 14 LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 1 0 3 0 2 0 1 0 3
CW = 14 LSB = 3
WR
CLK
GATE
OUT
N N N N 0 3 0 2 0 2 0 3 0 2 0 1 0 3
CW = 14 LSB = 4 LSB = 5
WR
CLK
GATE
OUT
N N N N 0 4 0 3 0 2 0 1 0 5 0 4 0 3

FIGURE 11. MODE 2

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.

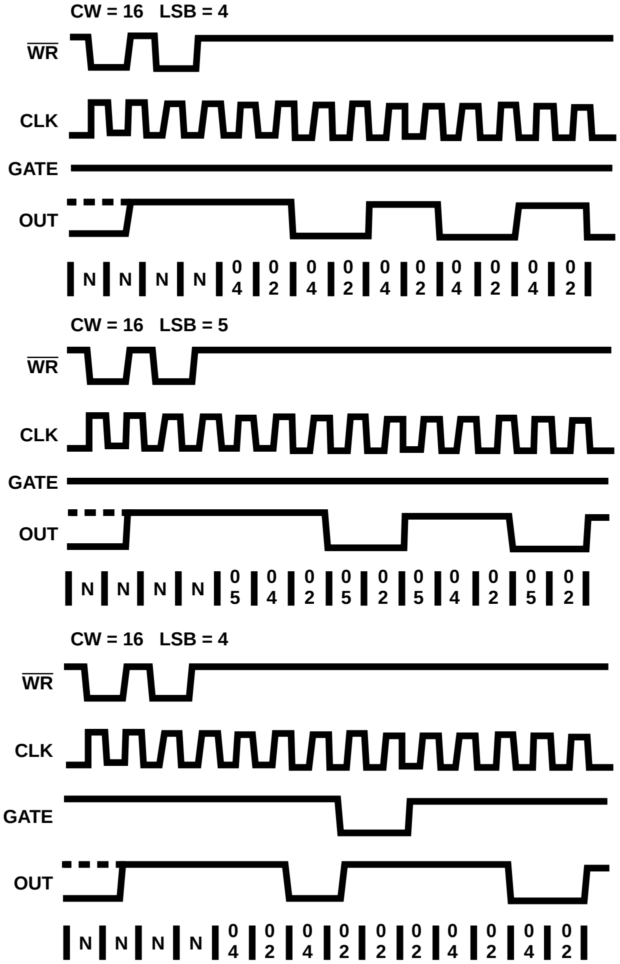

CW = 16 LSB = 4
$\overline{\text{WR}}$
CLK
GATE
OUT
N N N N 0 4 0 2 0 4 0 2 0 4 0 2 0 4 0 2 0 4 0 2
CW = 16 LSB = 5
$\overline{\text{WR}}$
CLK
GATE
OUT
N N N N 0 5 0 4 0 2 0 5 0 2 0 5 0 4 0 2 0 5 0 2
CW = 16 LSB = 4
$\overline{\text{WR}}$
CLK
GATE
OUT
N N N N 0 4 0 2 0 4 0 2 0 2 0 2 0 4 0 2 0 4 0 2

FIGURE 12. MODE 3

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 is loaded on one CLK pulse, decremented by one on the next CLK pulse, and then decremented by two on succeeding CLK pulses. When the count expires, OUT goes low and the Counter is reloaded with the initial count. The count is decremented by three on the next CLK pulse, and then by two on succeeding CLK pulses. When the count expires, OUT goes high again and the Counter is reloaded with the initial count. 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.

Mode 4: Software Triggered Mode

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse 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.

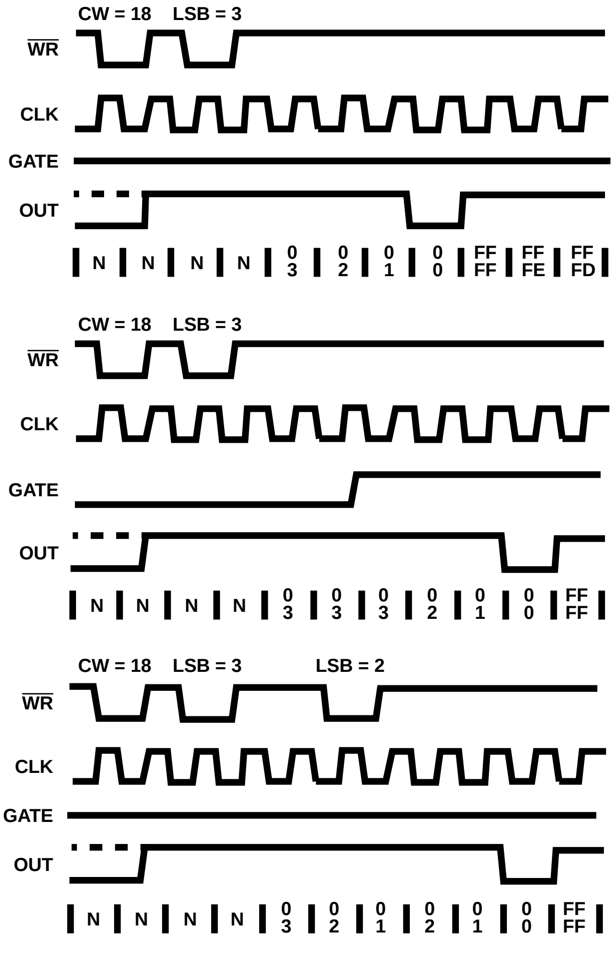

CW = 18 LSB = 3
WR
CLK
GATE
OUT
N N N N 03 02 01 00 FFFF FFFE FFFD
CW = 18 LSB = 3
WR
CLK
GATE
OUT
N N N N 03 03 03 02 01 00 FFFF
CW = 18 LSB = 3 LSB = 2
WR
CLK
GATE
OUT
N N N N 03 02 01 02 01 00 FFFF

FIGURE 13. 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 trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is triggerable. 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 new count on the next CLK pulse and counting will continue from there.

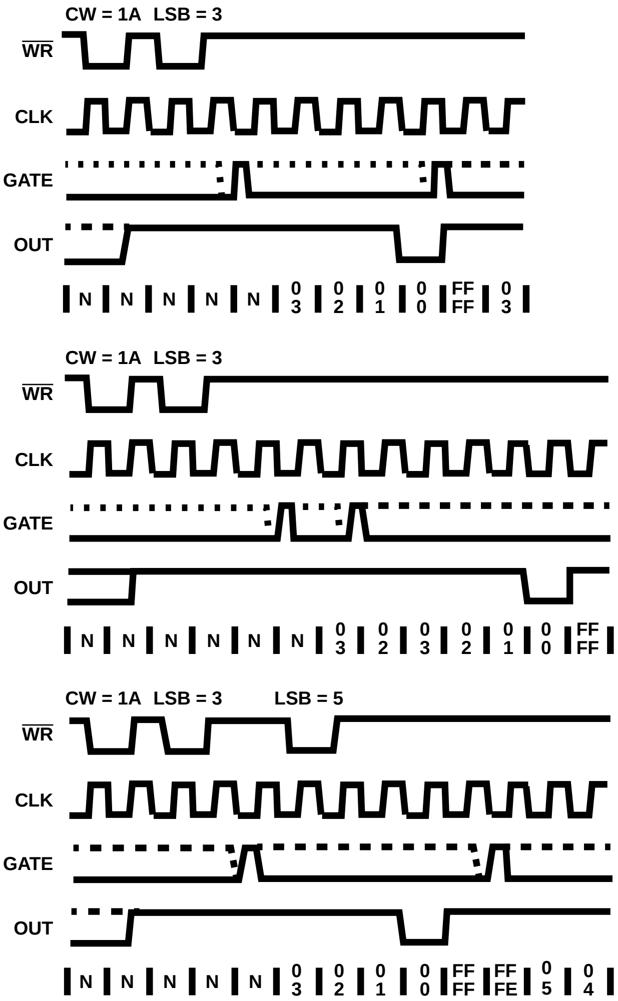

CW = 1A LSB = 3
WR
CLK
GATE
OUT
N N N N N 03 02 01 00 FFFF 03
CW = 1A LSB = 3
WR
CLK
GATE
OUT
N N N N N N 03 02 03 02 01 00 FFFF
CW = 1A LSB = 3 LSB = 5
WR
CLK
GATE
OUT
N N N N N 03 02 01 00 FFFF FFFE 05 04

FIGURE 14. MODE 5

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 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.

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.

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	1) Disables Counting	-	Enables Counting
5	-	Initiates Counting	-

FIGURE 15. GATE PIN OPERATIONS SUMMARY

MODE	MIN COUNT	MAX COUNT
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

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

FIGURE 16. MINIMUM AND MAXIMUM INITIAL COUNTS