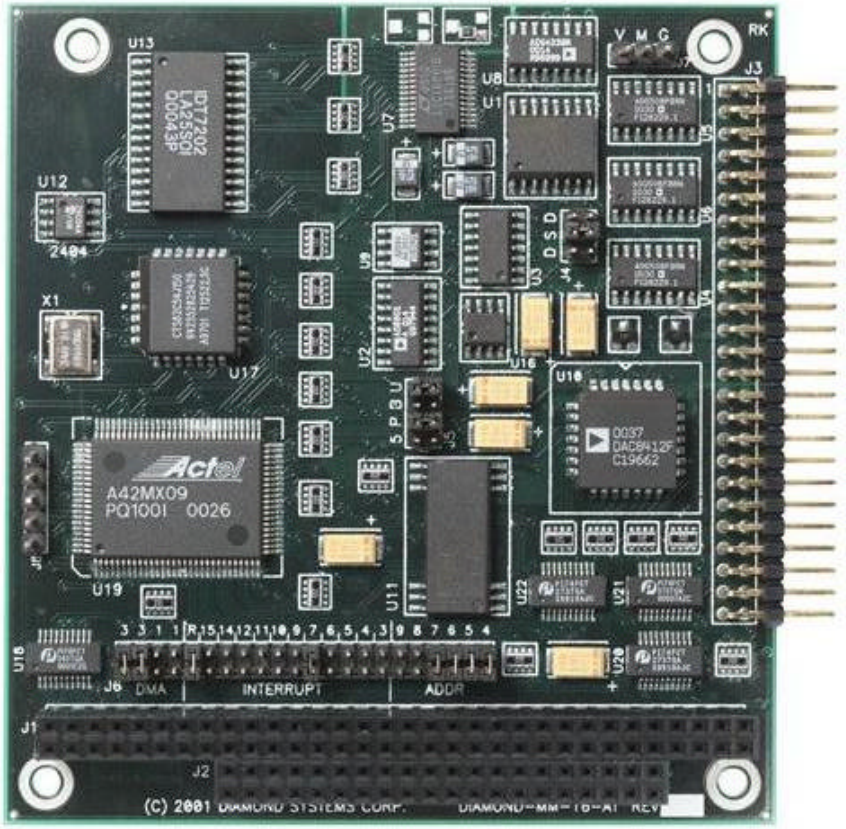
DIAMOND SYSTEMS CORPORATION

DIAMOND-MM-16-AT

Autocalibrating
16-bit Analog I/O PC/104 Module

User Manual V1.26



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

TABLE OF CONTENTS

DIAMOND-MM-16-AT *Autocalibrating 16-bit Analog I/O PC/104 Module*

1. DESCRIPTION

Diamond-MM-16-AT (DMM-16-AT) is a PC/104 expansion board offering embedded systems designers a full feature set of data acquisition capabilities. It is designed to be used in any PC-compatible embedded computer with a PC/104 (ISA-bus) expansion connector.

The board is an upgraded version of Diamond Systems' DMM-16-XT board with backwards hardware and software compatibility to allow system upgrade without any redesign in most applications. Upgraded features include autocalibration and integrated FIFO for higher reliability high-speed operation.

Key features include:

Analog Input

- 16 single-ended / 8 differential inputs
- 16-bit A/D resolution
- 100KHz maximum aggregate A/D sampling rate
- Programmable input ranges with maximum range of +/-10V
- Both bipolar and unipolar input ranges
- 512-sample FIFO for reliable high-speed sampling
- Autocalibrated inputs

Analog Output

- 4 optional analog outputs
- Fixed and user-programmable output ranges
- Simultaneous update
- Autocalibrated outputs

Digital I/O

- 8 dedicated digital outputs, TTL compatible
- 8 dedicated digital inputs, TTL compatible

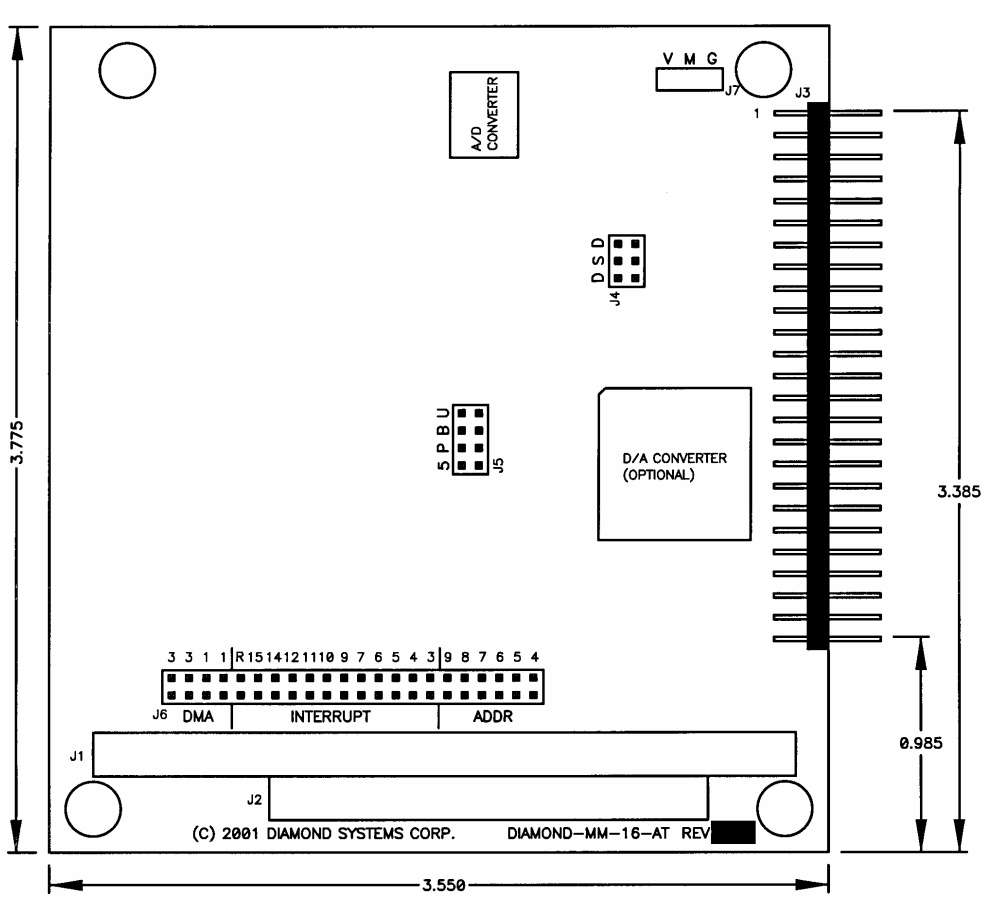
Counter/Timers

- 1 32-bit counter/timer for A/D sampling rate control
- 1 16-bit counter/timer for user counting and timing functions
- Programmable clock source for user counter/timer

System Features

- +5V-only operation
- Extended temperature range (-40 to +85^{o}C)
- Connector pinout compatible with Diamond-MM-16 board
- Register map compatible with Diamond-MM-16 board

2. DIAMOND-MM-16-AT BOARD DRAWING



Feature descriptions

J1	PC/104 8-bit bus connector
J2	PC/104 16-bit bus connector
J3	User I/O connector
J4	A/D single-ended / differential configuration
J5	D/A range and polarity configuration
J6	DMA / Interrupt / Address configuration
J7	Factory use only

3. I/O HEADER PINOUT AND PIN DESCRIPTION

Diamond-MM-16-AT provides a 50-pin header labeled J3 for all user I/O. This header is located on the right side of the board. Pin 1 is the upper left pin and is marked on the board.

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	Vout 2
Agnd	27	28	Vout 3
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
Vout0-3	Analog output channels 0 – 3
Vref Out	+5V precision reference voltage output
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
Agnd	Analog ground; all Agnd pins are tied together on the board. Pin 17 is a dedicated analog input ground in single-ended mode.
Dgnd	Digital ground

4. BOARD CONFIGURATION

Refer to the Drawing of Diamond-MM-16-AT on page 4 for locations of the configuration items mentioned here.

4.1 J6: Base Address

Each board in the system must have a different base address. Diamond-MM-16-AT's base address is set with a portion of jumper block 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 in the board's 10-bit I/O address. Bits 3-0 are always 0 for the base address, resulting in a 16-byte I/O address block. A jumper out is equal to a 1, and a jumper in is equal to a 0. 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-16-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

4.2 J4: A/D Single-Ended / Differential Mode

A single-ended input uses 2 wires, input and ground. The measured input voltage is the difference between these two wires. A differential input uses 3 wires: input +, input -, and ground. The measured input voltage is the difference between the + and - inputs.

A differential input has higher noise immunity than a single-ended input, since most noise affects both + and – input wires equally, whereas the noise in the ground signal will be a combination of radiated / conducted noise and noise injected into the ground signal by other electronic components on the board or in the signal source. The downside of differential inputs is that only half as many are available, since two input pins are required to produce a single differential input. DMM-16-AT can be configured for either 16 single-ended inputs or 8 differential inputs.

J4, a 2x3 jumper block, selects the analog input mode. For single-ended inputs, install a single jumper in the S position. An unused jumper may be stored for later use by installing it over only one pin. For differential inputs, install two jumpers in the two D positions. All inputs are configured in the same mode.

If you have a combination of single-ended and differential input signals, select differential mode. Then to measure the single-ended signals, connect the signal to the + input and connect analog ground to the - input.

4.3 J5: D/A Configuration

The 4 D/A channels can be configured in two ways:

- The full-scale output range can be selected between a fixed +5V output range or a user-programmable output range. The programmable range can be set anywhere between 1V and 10V.

- The outputs can be configured for unipolar (positive voltages only) or bipolar (both negative and positive output voltages). In unipolar mode, the outputs range from 0V minimum to the selected full-scale voltage (e.g. 5V in the fixed range). In bipolar mode, the outputs can range from – full-scale voltage to + full-scale voltage (e.g. +/-5V in the fixed range). Note that the maximum swing in bipolar mode is equal in both directions.

Two jumpers in J5 are used to configure the D/A channels. For fixed range, install a jumper in the location marked 5. For programmable range, install a jumper in the location marked P. Do not install a jumper in both locations simultaneously or across the two columns, as unpredictable behavior may result.

For bipolar inputs, install a jumper in the B location. For unipolar inputs, install a jumper in the U location. Do not install a jumper in both locations simultaneously or across the two columns, as unpredictable behavior may result.

Both the range and polarity jumpers must be installed for proper analog output behavior.

4.4 J6: DMA Level and Interrupt Level Selection

In addition to the base address described above, J6 is used to configure DMA and interrupt activity. Currently DMA activity is not supported on this board, so these jumper locations can be ignored.

Interrupts are used to transfer data from the board to memory at a rate higher than can be achieved through software sampling. During interrupt operation, the board will periodically generate an interrupt request. The processor will respond and run a user-supplied interrupt routine function (or the function supplied with the board's driver software). The interrupt routine reads the data from the board and makes it available to the user application program. DMM-16-AT allows you to select from levels 15, 14, 12, 11, 10, 9, 7, 6, 5, 4, and 3. Only one IRQ level is used by DMM-16-AT. To select the desired IRQ level install a jumper in that number's location in the Interrupt area of jumper block J6.

On the PC/104 bus each IRQ level in use must have a 1KΩ pull-down resistor attached. To enable the pull-down resistor for this board, install a jumper in the R location on J6.

Typically each board in the computer will use a different interrupt level, or IRQ level. However in special circumstances multiple boards may share the same IRQ level. In this case only one board should have the pull-down resistor enabled with the R location. The other boards should not have the resistor enabled.

5. I/O MAP

5.1 Overview

Diamond-MM-16-AT occupies 16 bytes in I/O memory space. A functional list of these 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
1	D/A LSB	A/D MSB
2	A/D channel register	A/D channel register
3	Digital output port	Digital input port
4	D/A 0 MSB	Update D/A
5	D/A 1 MSB	Update D/A
6	D/A 2 MSB	Update D/A
7	D/A 3 MSB	Update D/A
8	Clear interrupt flip flop	Status register
9	Interrupt control register	Interrupt control register readback
10	Ctr/Timer and FIFO Control Register	FIFO / status register
11	Analog Configuration Register	Analog and FIFO register readback

Addresses 12-15 form a window into 2 4-byte pages. The page is selected with a bit in register 10.

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 Control

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	EEPROM access key	FPGA code version

5.2 Register Map Bit Assignments

A blank location in the Write registers has no function.

A blank location in the Read registers has no function and reads back as 0.

WRITE operations

	7	6	5	4	3	2	1	0
0	Start A/D Conversion							
1	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
2	HIGH3	HIGH2	HIGH1	HIGH0	LOW3	LOW2	LOW1	LOW0
3	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
4					DA0-11	DA0-10	DA0-9	DA0-8
5					DA1-11	DA1-10	DA1-9	DA1-8
6					DA2-11	DA2-10	DA2-9	DA2-8
7					DA3-11	DA3-10	DA3-9	DA3-8
8	Clear Interrupt Request Flip Flop							
9	AINTE				TINTE	RSVD	CLKEN	CLKSEL
10	FIFORST	PAGE	FIFOEN	SCANEN	CLKFRQ	C2	C1	C0
11				SCNINT	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

	7	6	5	4	3	2	1	0
0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
1	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
2	HIGH3	HIGH2	HIGH1	HIGH0	LOW3	LOW2	LOW1	LOW0
3	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
4	Update D/A							
5	Update D/A							
6	Update D/A							
7	Update D/A							
8	STS	TINT	SD	AINT	CH3	CH2	CH1	CH0
9	AINTE				TINTE	DMAEN	CLKEN	CLKSEL
10	WAIT	PAGE	FIFOEN	SCANEN	CLKFRQ	OVF	HF	EF
11	C2	C1	C0		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/Status			
15	Page 0: 82C54 Control Register				Page 1: EEPROM Key			

6. REGISTER DEFINITIONS

Base + 0 Read A/D LSB

Bit No.	7	6	5	4	3	2	1	0
Name	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Definitions:

AD7 – 0 A/D data bits 7 - 0; AD0 is the LSB; A/D data is a signed 16-bit value ranging from -32768 to +32767.

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	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8

Definitions:

AD15 – 8 A/D data bits 15-8; AD15 is the MSB

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 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 appropriate pairs of bytes are read out together.

Base + 1 Write D/A 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 bits 7-0; DA0 is the LSB. D/A data is an unsigned 12-bit number ranging from 0 to 4095.

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.

A/D channels are automatically selected in sequence by the board. Each time an A/D conversion (A/D sample) starts, the board increments to the next channel in the range. When the high channel is sampled, the board resets to the low channel.

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 through Base + 7 Write DAC 0 – 3 MSB

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

Definitions:

DA11 – 8 D/A data bits 11 – 8 for the selected channel. DA11 is the MSB.

Base + 4 is used for D/A 0, Base + 5 is used for D/A 1, Base + 6 is used for D/A 2, and Base + 7 is used for D/A 3.

The final D/A value is constructed of the 4 upper bits written to these registers combined with the 8 lower bits of the D/A value written to Base + 1. Writing data to any of these 4 registers causes these 4 bits and the 8 lower bits from Base + 1 to be transferred to the selected D/A channel. However the D/A is not updated until a read operation is performed on one of these 4 addresses. This lets you write data to more than one D/A and then update all of them at the same time.

Since the Base + 1 register is shared by all 4 D/A channels, each D/A channel must have its data written in the proper sequence. First write the LSB to Base + 1, then write the MSB to one of the MSB registers 4-7 depending on the D/A selected. Repeat these two writes for each D/A you want to update. After all data is written, read from any of these registers to update all the D/A channels simultaneously.

Note that even though all channels are updated simultaneously, a channel will only change if it has new data written to it since the last update operation. Otherwise it will maintain its present value during the update operation.

Base + 4 through Base + 7 Read Update D/A channels

Reading from any of these 4 addresses will cause the 4 analog outputs to be updated. All outputs are updated simultaneously. See detailed description above.

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 request is generated by the board (i.e. during A/D conversions), and it must be cleared by software before another interrupt can be generated. Diamond-MM-16-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).

Base + 9 Read/Write Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	AINTE				TINTE	RSVD	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

⇒**:** Both AINTE and TINTE should not be set concurrently. Only one interrupt operation at a time is supported by the board.

RSVD DMA enable (reserved for future implementation, not currently supported)

CLKEN Enable hardware A/D clock:

1 Enable hardware A/D trigger (source is selected with CLKSEL bit); software triggers are disabled

0 Disable hardware trigger; A/D is triggered via software write to Base + 0

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 Write Counter/Timer and FIFO Control Register

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

FIFORST Reset FIFO. Writing a 1 to this bit causes the on-board FIFO to be reset to empty in preparation for an interrupt-based A/D operation. If this register is written to with a 1 in this bit, the board will ignore the other 7 bits of the written value, and the current values of the other 7 bits in the register will be preserved.

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

Page 0: 82C54 counter/timer access

Page 1: Calibration registers

FIFOEN FIFO enable:

1 Enable FIFO operation; if interrupts are enabled, interrupt requests will occur when the FIFO reaches or exceeds half-full (HF = 1).

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

SCANEN Scan enable:

1 A/D 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 remain high for a single conversion

CLKFRQ Clock frequency for counter/timer 1

0 10MHz

1 1MHz

C2 External Clock Gate Enable

1 IN0- (pin 29 on the I/O header) acts as a gate for A/D sample control when external A/D clock is enabled (CLKSEL = 0 above). When IN0- is high, falling edges on DI0 (pin 48 on the I/O header) 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 to Counter 0 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 to Counter 0 is IN0- (pin 29 on the I/O connector) inverted. Counter 0 counts on falling 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 + 10 Read Counter/Timer and FIFO Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	WAIT	PAGE	FIFOEN	SCANEN	CLKFRQ	OVF	HF	EF

WAIT Analog input circuit status. Whenever register 2 (channel register) or 11 (input range register) is written to, WAIT will go high for approximately 10μS as the circuit adjusts to the new signal.

1 The analog input circuit is busy settling on a new signal. Do not perform A/D conversions when WAIT = 1.

0 The circuit is ready for A/D conversions

PAGE Readback of PAGE bit described on previous page

FIFOEN Readback of FIFOEN bit described on previous page

SCANEN Readback of SCANEN bit described on previous page

CLKFRQ Readback of CLKFRQ bit described on previous page

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

Overflow is defined as the state when the FIFO is full and another A/D conversion occurs before any data is read out of the FIFO. In an overflow condition the FIFO contents are preserved, and no new data will be written to the FIFO. To clear an overflow condition, the FIFO must be reset with the FIFORST bit in register 10.

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

Base + 11 Write Analog Configuration Register

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

SCNINT Scan interval. This is the time between A/D samples during an A/D scan. An A/D scan occurs when SCANEN = 1 (Base + 10 bit 4) and an A/D conversion is triggered.

0 9.3µS

1 5.3µS

RANGE 5V or 10V A/D positive full-scale range; 0 = 5V, 1 = 10V

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

The table below lists the various combinations of ADBU and RANGE. The numbers shown are NOT the same as the input voltage range. See G1-0 below.

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

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 shown above and the effective input signal range. For example, if the A/D full-scale 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 (this register is cleared to 0).

Base + 11 Read Analog and Status Readback Register

Bit No.	7	6	5	4	3	2	1	0
Name	C2	C1	C0		RANGE	ADBU	G1	G0

This address provides a means of reading back the values written to the registers at Base + 10 (C2, C1, C0) and Base + 11 (RANGE, ADBU, G1, G0).

PAGE 0: 82C54 Counter/Timer

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

These registers map directly to the 82C54 counter/timer IC on the board. The definitions of these registers can be found in the 82C54 datasheet appended to the back of this manual.

Page 1: Calibration Control Registers

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 CALSTS = 0.

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	A7	A6	A5	A4	A3	A2	A1	A0

A7-A0 EEPROM / TrimDAC address. The EEPROM recognizes address 0 – 127 using address bits A6 – A0. The TrimDAC only recognizes addresses 0 – 7 using bits A2 – A0. In each case remaining address bits will be ignored.

Base + 14 Write Calibration Control Register

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

This register is used to initiate various commands related to autocalibration. More detailed information on autocalibration may be found elsewhere in this manual.

EE_EN EEPROM Enable. Writing a 1 to this bit will initiate a transfer to/from the EEPROM as indicated by the EE_RW bit.

EE_RW Selects read or write operation for the EEPROM: 0 = Write, 1 = Read.

RUNCAL Writing 1 to this bit causes the board to reload the calibration settings from EEPROM.

CMUXEN Calibration multiplexor enable. The cal mux is used to read precision on-board reference voltages that are used in the autocalibration process. It also can be used to read back the value of analog output 0.

1 = enable cal mux and disable user analog input channels

0 = disable cal mux, enable user inputs

TDACEN TrimDAC Enable. Writing 1 to this bit will initiate a transfer to the TrimDAC (used in the autocalibration process).

Base + 14 Read Calibration Status Register

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

TDBUSY TrimDAC busy indicator

0 User may access TrimDAC

1 TrimDAC is being accessed; user must wait

EEBUSY EEPROM busy indicator

0 User may access EEPROM

1 EEPROM is being accessed; user must wait

Base + 15 Write EEPROM Access Key Register

The user must write the value 0xA5 (binary 10100101) to this register each time after setting the PAGE bit in order to get access to the EEPROM. This helps prevent accidental corruption of the EEPROM contents.

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-16-AT uses a 16-bit A/D converter. This means that the analog input voltage can be measured to the precision of a 16-bit binary number. The maximum value of a 16-bit binary number is 2^{16} - 1, or 65535, so the full range of numerical values that you can get from a Diamond-MM-16-AT analog input channel is 0 - 65535.

The smallest change in input voltage that can be detected is $1/(2^{16})$, or 1/65536, 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-16-AT can measure both unipolar (positive only) and bipolar (positive and negative) analog voltages. In general you should select the highest gain (smallest input range) that will allow the A/D converter to read the full range of voltages for your input signals. 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 input voltages.

7.3 Single Ended and Differential Inputs

Diamond-MM-16-AT can handle both single-ended and differential inputs. A single-ended input is a two-wire input (one input signal and ground) 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 three-wire input (input +, input -, and ground), and the board will measure the difference between the voltages of the two inputs. Polarity is important for a differential input. Diamond-MM-16-AT will subtract the voltage on the low (-) input from the voltage of the high (+) input. Differential inputs are frequently used when the grounds of the input device and the measurement device (Diamond-MM-16-AT) are at different voltages, or when a low-level signal is being measured that has its own ground wire. Differential inputs also provide better noise immunity than single-ended inputs, because most of the noise will be present in equal amounts on both the + and – inputs, so in the subtraction process the noise will be cancelled out.

7.4 Input Ranges and Resolution

Range	ADBU	G1	G0	Input Range	Resolution (1 LSB)
0	0	0	0	±5V	153µV
0	0	0	1	±2.5V	76µV
0	0	1	0	±1.25V	38µV
0	0	1	1	±0.625V	19µV
0	1	0	0	Invalid setting	
0	1	0	1	Invalid setting	
0	1	1	0	Invalid setting	
0	1	1	1	Invalid setting	
1	0	0	0	±10V	305µV
1	0	0	1	±5V	153µV
1	0	1	0	±2.5V	76µV
1	0	1	1	±1.25V	38µV
1	1	0	0	0 – 10V	153µV **See Note**
1	1	0	1	0 – 5V	76µV
1	1	1	0	0 – 2.5V	38µV
1	1	1	1	0 – 1.25V	19µV

Note: On boards earlier than revision C, or with no revision mark, the 0-10V input range is linear only up to about 8.3V. On these boards, if you need to read inputs higher than 8.3V, use the bipolar ±10V input range. The revision mark is on the lower right edge of the board.

8. 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**
3. **Wait for analog input circuit to settle**
4. **Initiate an A/D conversion**
3. **Wait for the conversion to finish**
4. **Read the data from the board**
5. **Convert the numerical data to a meaningful value**

8.1 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. (See Chapter 7). 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.

For example:

To set the board to channel 4 only, write 0x44 to Base + 2.

To set the board to read channels 0 through 15, write 0xF0 to Base + 2.

⇒ **Note:** 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 and so on.

If you are sampling the same channel repeatedly, then you set both high and low to the same value as in the first example above. Then on subsequent conversions you do not need to set the channel again.

8.2 Select the input range

Select the input range from among the available ranges shown on page 21. If the range is the same as for the previous A/D conversion then it does not need to be set again. Write this value to the input range register at Base + 11.

For example:

For ±10V range, write 0x08 to Base + 11.

8.3 Wait for analog input circuit to settle

After writing to either the channel register (Base + 2) or the input range register (Base + 11), you must allow time for the analog input circuit to settle before starting an A/D conversion. The board has a built-in 10µS timer to assist with the wait period. Monitor the WAIT bit at Base + 10 bit 7. When it is 1 the circuit is actively settling on the input signal. When it is 0 the board is ready to perform A/D conversions.

8.4 Perform an A/D conversion on the current channel

After the above steps are completed, start the A/D conversion by writing to Base + 0. This write operation only triggers the A/D if the CLKEN bit is 0 to disable hardware triggering and enable software triggering. Otherwise the A/D will only trigger when the selected clock or trigger signal occurs. CLKEN should always be 0 when controlling A/D conversions in software.

8.5 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 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. Here is a pseudocode explanation:

```
Status = read(base+8) AND 128 // or Status = read(base+8) AND 80 Hex
If Status = 0 then conversion is complete, else A/D converter is busy
```

Keep repeating this procedure until Status = 0.

8.6 Read the data from the board

Once the conversion is complete, you can read the data back from the A/D converter. The data is 16 bits wide and is read back in two 8-bit bytes. The following pseudocode illustrates how to construct the 16-bit A/D value from these two bytes:

```
LSB = read(base)
MSB = read(base+1)
Data = MSB * 256 + LSB        // combine the 2 bytes into a 16-bit value
```

The final data is interpreted as a signed value ranging from –32768 to +32767.

⇒ **Note:** The data range always includes both positive and negative values, even if the board is set to a unipolar input range. The data must now be converted to volts or other engineering units by using a conversion formula as shown on the next page.

8.7 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. However you can combine both transformations into a single formula if desired.

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

Conversion Formula for Bipolar Input Ranges

Input voltage = A/D value / 32768 * Full-scale input range

Example: Input range is ±5V and A/D value is 17761:
Input voltage = 17761 / 32768 * 5V = 2.710V

For a bipolar input range, 1 LSB = 1/32768 * Full-scale voltage.

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 symbolic formula	Input voltage for ±5V range
-32768	$-V_{FS}$	-5.0000V
-32767	$-V_{FS}$ + 1 LSB	-4.9998V
…	…	…
-1	-1 LSB	-0.00015V
0	0	0.0000V
1	+1 LSB	0.00015V
…	…	…
32767	V_{FS} - 1 LSB	4.9998V

Conversion Formula for Unipolar Input Ranges

Input voltage = (A/D value + 32768) / 65536 * Full-scale input range

Example: Input range is 0-5V and A/D value is 17761:
Input voltage = (17761 + 32768) / 65536 * 5V = 3.855V

For a unipolar input range, 1 LSB = 1/65536 * Full-scale voltage.

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 symbolic formula	Input voltage for ±5V range
-32768	0V	0.0000V
-32767	1 LSB (V_{FS} / 65536)	0.000076V
…	…	…
-1	V_{FS} / 2 - 1 LSB	2.4999V
0	V_{FS} / 2	2.5000V
1	V_{FS} / 2 + 1 LSB	2.5001V
…	…	…
32767	V_{FS} - 1 LSB	4.9999V

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

The three control bits FIFOEN (FIFO enable), SCANEN (scan enable), and AINTE (A/D interrupt enable) determine the behavior of the board during A/D conversions.

In all cases, at the end of an AD conversion A/D data is latched into the FIFO in an interleaved fashion, first LSB, then MSB. A/D Data is read out of the FIFO with 2 read operations, first Base + 0 (LSB) and then Base + 1 (MSB).

When SCANEN = 1, each time an A/D trigger occurs, the board will perform an A/D conversion on all channels in the channel range. When SCANEN = 0, each time an A/D trigger occurs, the board will perform a single A/D conversion and then advance to the next channel and wait for the next trigger.

During interrupt operation, if FIFOEN = 1, then the FIFO will fill up with data until it reaches or exceeds half-full (half-full = 256 samples), and then the interrupt request will occur.

For all cases here where AINTE = 1, it is assumed that CLKEN = 1 as well, since in most applications interrupt operation is based on a hardware A/D clock.

AINTE	FIFOEN	SCANEN	Operation
0	0	0	Single A/D conversions are triggered by write to B+0. STS stays high during the A/D conversion. No interrupt occurs. The user program monitors STS 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. STS stays high during the entire scan (multiple A/D conversions). No interrupt occurs. The user program monitors STS and reads all A/D values when 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 the source selected with CLKSEL. STS stays high during the A/D conversion. A/D interrupt occurs 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 the source selected with CLKSEL. STS stays high during the entire scan (multiple A/D conversions). A/D interrupt occurs after the entire scan is complete. The interrupt routine reads out one entire A/D scan (multiple values) each time it runs.
1	1	0	Single A/D conversions are triggered by the source selected with CLKSEL. STS stays high during the A/D conversion. A/D interrupt occurs when HF goes high (256 A/D conversions have occurred). The interrupt routine reads out 256 samples (half the FIFO) each time it runs.
1	1	1	A/D scans are triggered by the source selected with CLKSEL. STS stays high during the entire scan (multiple A/D conversions). A/D interrupt occurs after the scan is complete 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.

10. ANALOG OUTPUT RANGES AND RESOLUTION

10.1 Description

Diamond-MM-16-AT uses a 4-channel 12-bit D/A converter (DAC) to provide 4 optional analog outputs. Model DMM-16-NA-AT does not include the 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-16-AT is 0 - 4095. The value 0 always corresponds to the lowest voltage in the output range, and the value 4095 always corresponds to the highest voltage.

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

10.2 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 = Output voltage range / 4096

Example: Output range = 0-5V;

Output voltage range = 5V – 0V = 5V

1 LSB = 5V / 4096 = 1.22mV

Example: Output range = ±5V;

Output voltage range = 5V – -5V = 10V

1 LSB = 10V / 4096 = 2.44mV

10.3 Output Range Selection

Jumper block J5 is used to select the D/A output range. Two selections need to be made.

First select whether you want fixed 5V or programmable D/A reference. For a fixed 5V reference install a jumper in location 5, and for programmable install a jumper in P. Do not install a jumper in both 5 and P simultaneously or unpredictable behavior will occur.

If you select programmable reference, the reference must then be programmed and calibrated using the Universal Driver software. After the reference has been programmed and calibrated one time, it will be stored in an EEPROM on the board and will be automatically recalled each time the board is powered up.

Next select bipolar or unipolar output. For bipolar output the outputs will swing between ± the selected reference, and for unipolar output the output will swing from 0V to the selected reference. For bipolar output install a jumper in the B position, and for unipolar output install a jumper in the U position. Again, do not install a jumper in both B and U simultaneously or unpredictable behavior will occur.

10.4 D/A Conversion Formulas and Tables

The formulas below explain how to convert between D/A codes and output voltages.

Conversion Formulas for Unipolar Output Ranges

Output voltage = (D/A code / 4096) * Reference voltage

D/A code = (Output voltage / Reference voltage) * 4096

Example: Reference voltage = 5V
Output range in unipolar mode = 0 – 5V
Full-scale range = 5V – 0V = 5V
Desired output voltage = 1V; D/A code = 1V / 5V * 4096 = 819.2 => 819

For a unipolar output range, 1 LSB = 1/4096 * Full-scale range, or 1.22mV in this example.

Here is an illustration of the relationship between D/A code and output voltage for a unipolar output range (V_{REF} = Reference voltage):

D/A Code	Output voltage symbolic formula	Output voltage for 0 – 5V range
0	0V	0.0000V
1	1 LSB (V_{REF} / 4096)	0.0012V
…	…	…
2047	V_{REF} / 2 - 1 LSB	2.4988V
2048	V_{REF} / 2	2.5000V
2049	V_{REF} / 2 + 1 LSB	2.5012V
…	…	…
4095	V_{REF} - 1 LSB	4.9988V

Conversion Formulas for Unipolar Output Ranges

Output voltage = ((D/A code – 2048) / 2048) * Output reference

D/A code = (Output voltage / Output reference) * 2048 + 2048

Example: Reference voltage = 5V
Output range in bipolar mode = ±5V
Full-scale range = 5V – (-5V) = 10V
Desired output voltage = 1V; D/A code = 1V / 5V * 2048 + 2048 = 2457.6 => 2458

For a bipolar output range, 1 LSB = 1/4096 * Full-scale range, or 2.44mV in this example.

Here is an illustration of the relationship between D/A code and output voltage for a bipolar output range (V_{REF} = Reference voltage):

D/A Code	Output voltage symbolic formula	Output voltage for ±5V range
0	$-V_{REF}$	-5.0000V
1	$-V_{REF}$ + 1 LSB	-4.9976V
…	…	…
2047	-1 LSB	-0.0024V
2048	0	0.0000V
2049	+1 LSB	0.0024V
…	…	…
4095	V_{REF} - 1 LSB	4.9976V

11. 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 three steps involved in performing a D/A conversion:

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

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

Use the formulas on the preceding page to compute the D/A code required to generate the desired voltage.

⇒ **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.

11.2 Write the value to the selected output channel

The four DACs share a single address for the LSB, Base + 1. Each DAC then has its own MSB address. Writing to the DAC's MSB address causes the 12-bit value to be loaded into the DAC. Therefore the LSB must be written first.

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)

Now write these values to the selected channel:

Write LSB to Base + 1

Write MSB to Base + 4 , Base + 5, Base + 6, or Base + 7 depending on the channel no.

11.3 Update the D/A

Read from any address in the range Base + 4 through Base + 7 to update the DACs. All four DACs are updated at the same time, so you can write data to multiple DACs and then perform a single read operation from any of the four addresses to update all of them at once.

Any DAC that has not had new data loaded into it will maintain its current value.

12. AUTOCALIBRATION OPERATION

Diamond-MM-16-AT includes a sophisticated autocalibration circuit that manages the calibration of both the A/D and the D/A circuitry. Operation is as follows.

12.1 Reference Voltages

The board contains a precision reference voltage chip that is selected for high stability over time and temperature. The value of the voltage output from this chip is measured at the factory. The board also contains some precision resistor divider ladders that produce intermediate voltages derived from the original reference. All these voltages are measured at the factory and their values are stored in an EEPROM on the board.

12.2 A/D calibration

When the A/D is calibrated, it measures these voltages using an extra input multiplexor reserved for calibration. The calibration software compares the measurements to the stored values and makes adjustments to the board to bring the measurements into tolerance (less than 2 LSBs max, in most cases less than 1 LSB). The adjustments are produced by controlling 4 8-bit DACs that are inserted at various points in the circuit. The DAC data is then stored in the EEPROM. Each of the valid A/D input ranges has its own set of calibration values, since each input range has slight offset and gain differences compared to each other range.

12.3 D/A Calibration

When the D/A is calibrated, the board performs a similar operation. The output of DAC 0 is routed through the calibration multiplexor. The offsets of the other three DACs relative to DAC 0 are measured at the factory and stored in the EEPROM. During calibration the average offset is added to the measured output of DAC 0, and this value is used as the comparison value to minimize overall errors. During D/A calibration the board can automatically determine the jumper configuration of the DACs based on their response to various output codes.

If the board is configured for programmable output range, the calibration software will first set and calibrate the output range, then calibrate the D/A.

12.4 Universal Driver Software Support

Calibration is simple when using the Diamond Systems Universal Driver software. Several functions are provided to manage the entire operation, and a demo program is included. For application developers targeting an operating system not supported by Universal Driver, the source code is included so you can incorporate it into your own program.

With the Universal Driver software, you have the option of recalling calibration values each time you change the input range, or leaving the current ones in place. Leaving the current ones in place will match the performance of other A/D boards which also use only a single set of calibration values. Recalling the values specific to the new input range will improve performance by a few LSBs but will result in a time delay since the data must be recalled from the EEPROM and loaded into the DACs.

A/D and D/A maybe calibrated separately. Calibration takes a few seconds and may be performed as often as desired, for example at system startup, once a day, etc.

NOTE: When calibrating the D/A channels, the output voltage of DAC0 will fluctuate between – full scale and + full-scale as part of the procedure. Any circuitry connected to the DAC during this time may be affected and produce unwanted results.

13. DIGITAL I/O OPERATION

Diamond-MM-16-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 Chapter 2, p. 4). They are CMOS TTL compatible. Each line can drive up to -15mA in a logic high state or sink up to 64mA in a logic low state. They do not have a readback feature, so your program must keep track of the latest output value. Diamond Systems' Universal Driver software keeps track of all digital output values automatically.

The inputs are located at pins 41 through 48 on the I/O header J3. They are also 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 digital input 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.

14. COUNTER/TIMER OPERATION

Diamond-MM-16-AT contains an 82C54 counter/timer IC that provides various timing functions on the board. The 82C54 contains 3 independent 16-bit counter/timers, numbered 0 – 2. Each counter has a clock input, a gate input, and an output. Each counter has a register that contains a 16-bit divisor value that is used to control the output. In timer mode, a clock is applied to the input pin and a divisor value is written to the counter. The output signal then pulses at a rate equal to the input signal divided by the 16-bit divisor. Thus the counters are often referred to as “divide-by-n” counters.

Each counter also has a 16-bit count register. The count register decrements by 1 each time a clock edge appears at the input. The number of input clocks can be determined by reading the current count register and subtracting it from the original value.

On DMM-16-AT counter 0 is made fully available to the user. The input is on pin 29 of the I/O header (IN0-). A falling edge on this input will cause the counter to decrement. The gate is on pin 46 of the I/O header (DIN2 / GATE0). When this pin is high, Counter 0 is enabled and can count. When this pin is low, Counter 0 holds its current value and ignores input clocks. Counter 0’s output is on pin 31 (OUT0). Pin 29 has a 10KΩ pull-up resistor.

Counters 1 and 2 are cascaded together. This means the output of counter 1 is connected to the input of counter 2. The resulting 32-bit counter/timer is used to control the A/D sampling rate. The input to counter 1 is a fixed 10MHz clock provided on the board. The gate signals for these two counters are tied together and are available on pin 48 (DIN0 / Gate 1/2).

Pins 46 and 48 (DIN2 and DIN0) are connected to 10KΩ pull-up resistors, so that if left unconnected they will be in the active / enabled state for the counters.

A datasheet on the 82C54 chip is included at the back of this manual.

15. SPECIFICATIONS

Analog Inputs

No. of inputs	8 differential or 16 single-ended (user selectable)
A/D resolution	16 bits (1/65,536 of full scale)
Input ranges	Bipolar: ±10V, ±5V, ±2.5V, ±1.25V Unipolar: 0-10V, 0-5V, 0-2.5V
Input bias current	50nA max
Maximum input voltage	±10V for linear operation
Overvoltage protection	±35V on any analog input without damage
Nonlinearity	±3LSB, no missing codes
Conversion rate	100,000 samples per second max
Conversion trigger	software trigger, internal pacer clock, or external TTL signal

Autocalibration

Circuits calibrated	A/D (all 8 input ranges) and D/A (fixed and programmable ranges)
A/D error	±1LSB (typical), ±2LSB (max) after autocalibration
D/A error	±1LSB (typical), ±2LSB (max) after autocalibration

Analog Outputs

No. of outputs	4, optional (not included on –NA version)
D/A resolution	12 bits (1/4096 of full scale)
Output ranges	Unipolar: 0 – 5V or 0 – (user-programmable) Bipolar: ±5V or ±(user-programmable)
Output current	±5mA max per channel
Settling time	4µS max to ±1/2 LSB
Relative accuracy	±1 LSB
Nonlinearity	±1 LSB, monotonic
Reference Voltage	5.000V ±3mV; 5mA max output current

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
Input current	±1µA 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; Logic 1: -15mA max

Counter/Timers

A/D Pacer clock	32-bit down counter (2 82C54 counters cascaded)
Pacer clock source	10MHz or 1MHz on-board clock source
General purpose	16-bit down counter (1 82C54 ctr); 100KHz or external input

General

Power supply	+5VDC ±10%
Current consumption	350mA typical
±15V output current	±10mA max with DACs unloaded; **not short-circuit protected**
+5V output current	Limited by PC/104 power supply; **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^oC to $+70^oC$**
  - **I82C54 . . . . . .-40^oC to $+85^oC$**
  - **M82C54 . . . . . .-55^oC to $+125^oC$**

Description

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

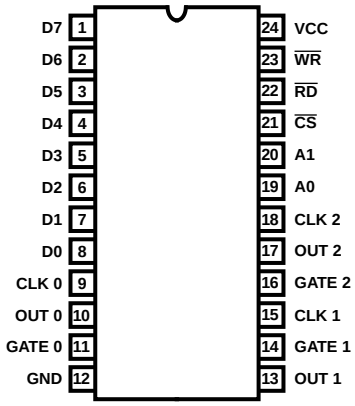
The 82C54 has three independently programmable and functional 16-bit counters, each capable of handling clock input frequencies of up to 8MHz (82C54) or 10MHz (82C54-10) or 12MHz (82C54-12).

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.

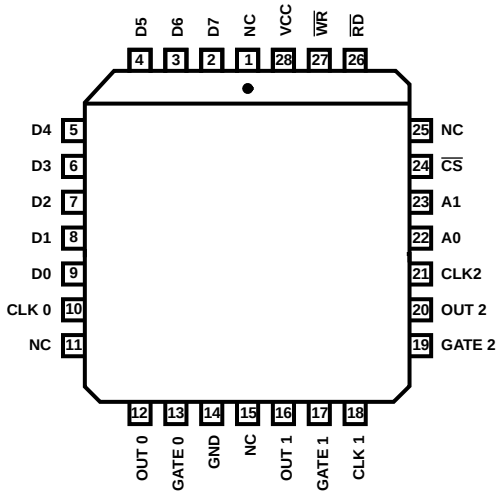
The Harris advanced CMOS process results in a significant reduction in power with performance equal to or greater than existing equivalent products.

Pinouts

82C54 (PDIP, CERDIP, SOIC)
TOP VIEW



82C54 (PLCC/CLCC)
TOP VIEW



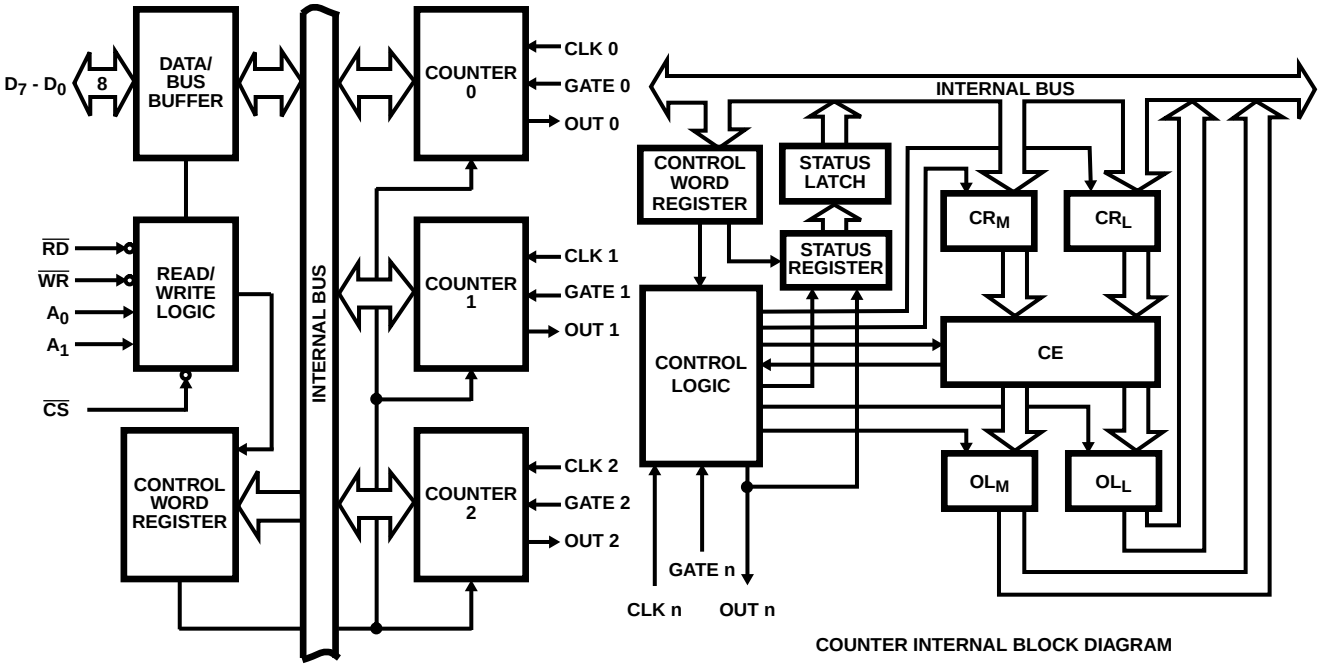
CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

File Number **2970.1**

Ordering Information

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

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

$\overline{CS}$	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.

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).

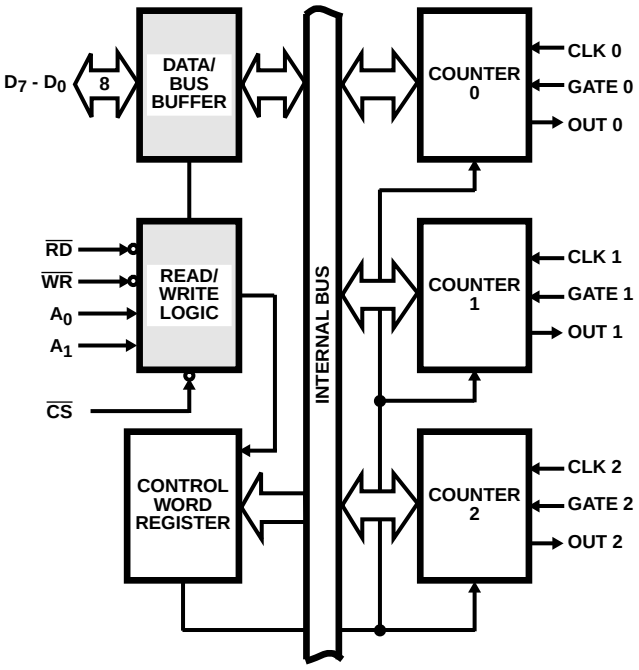


FIGURE 1. DATA BUS BUFFER AND READ/WRITE LOGIC FUNCTIONS

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.

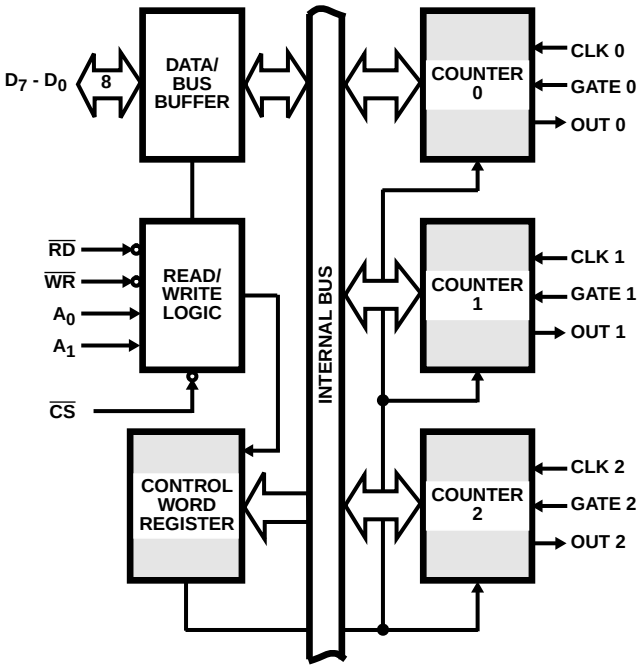


FIGURE 2. CONTROL WORD REGISTER AND COUNTER FUNCTIONS

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.

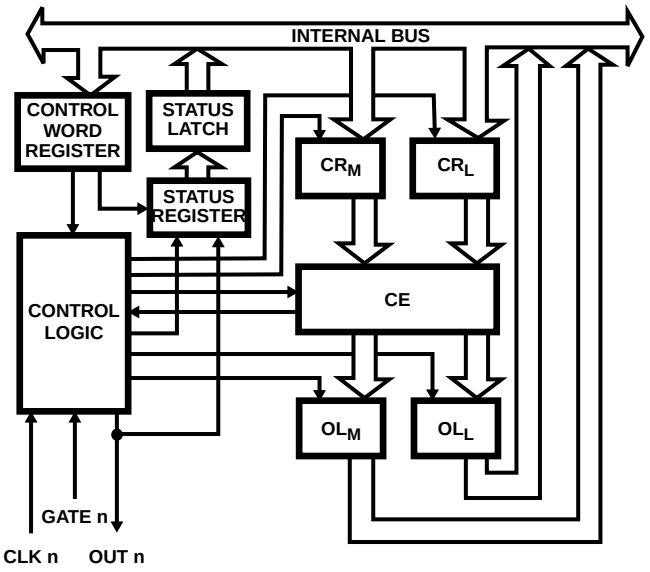


FIGURE 3. COUNTER INTERNAL BLOCK DIAGRAM

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.

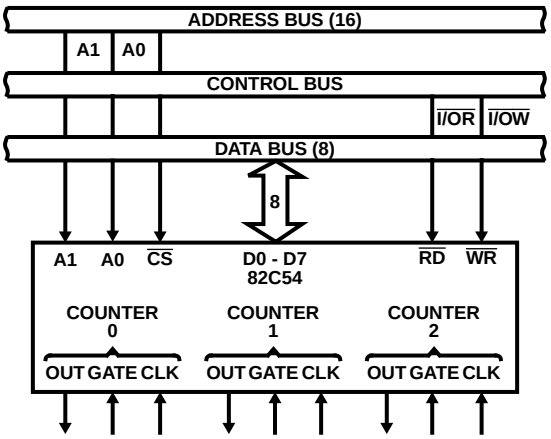


FIGURE 4. 82C54 SYSTEM INTERFACE

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; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D7	D6	D5	D4	D3	D2	D1	D0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	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.

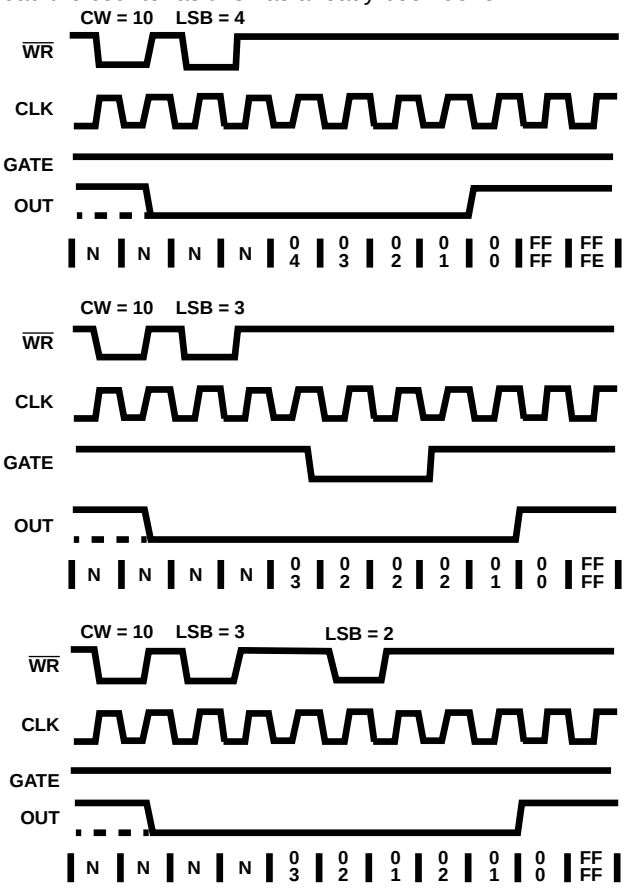


FIGURE 9. MODE 0

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.

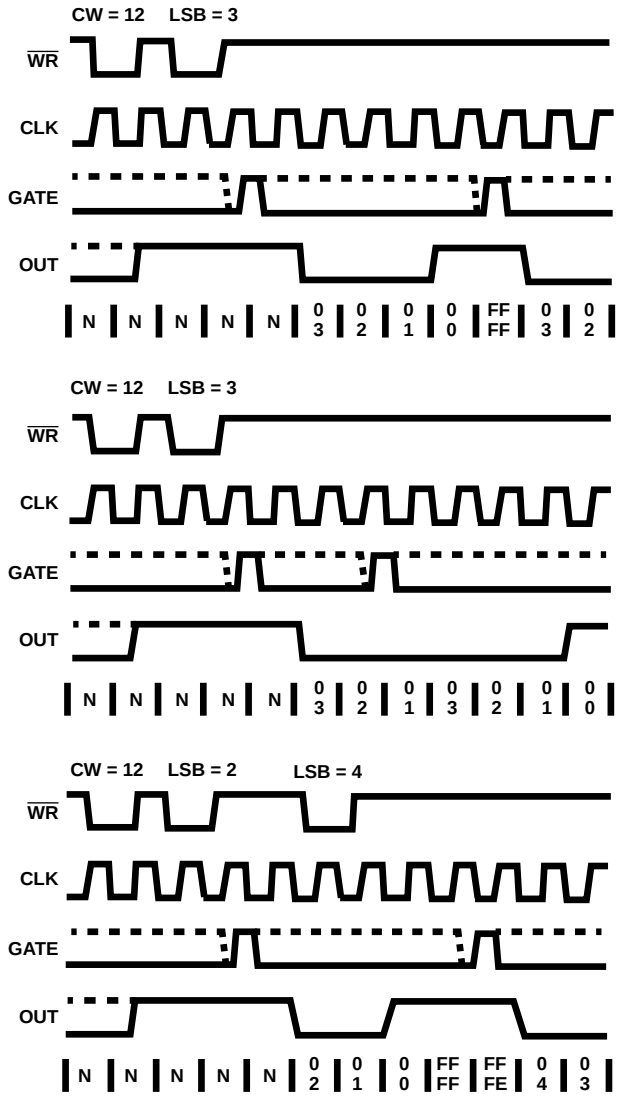


FIGURE 10. MODE 1

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.

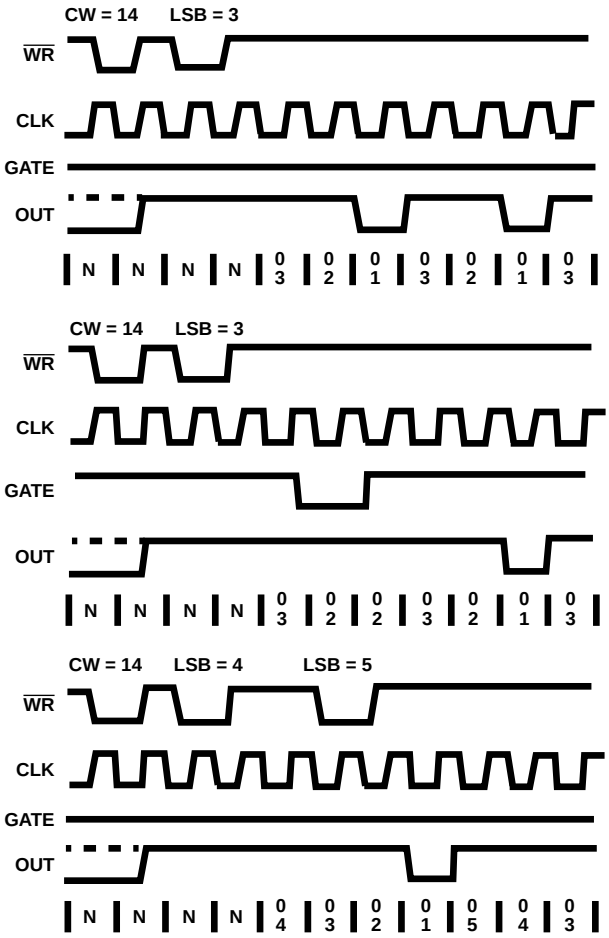


FIGURE 11. MODE 2

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.

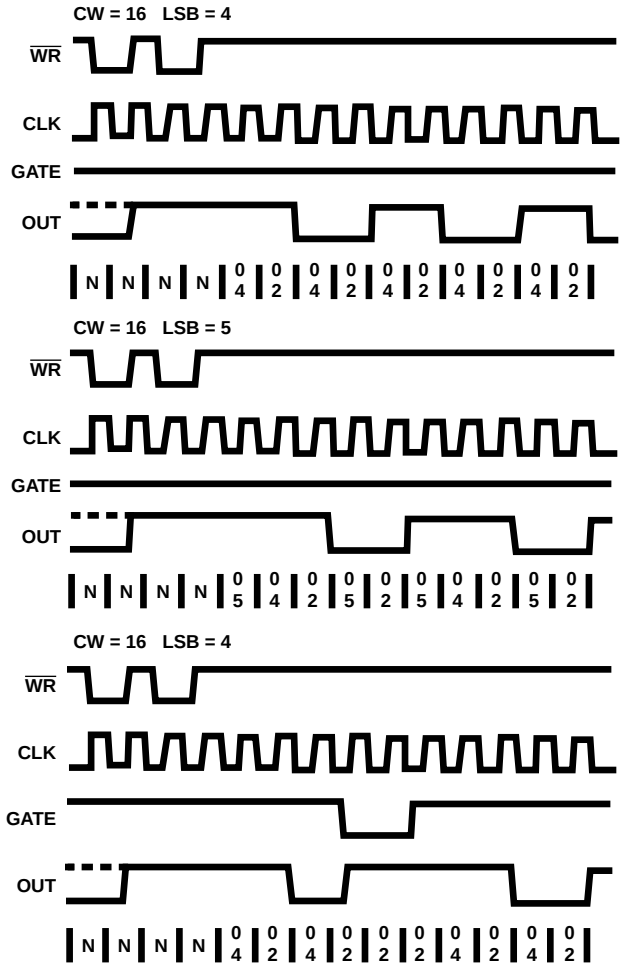


FIGURE 12. MODE 3

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.

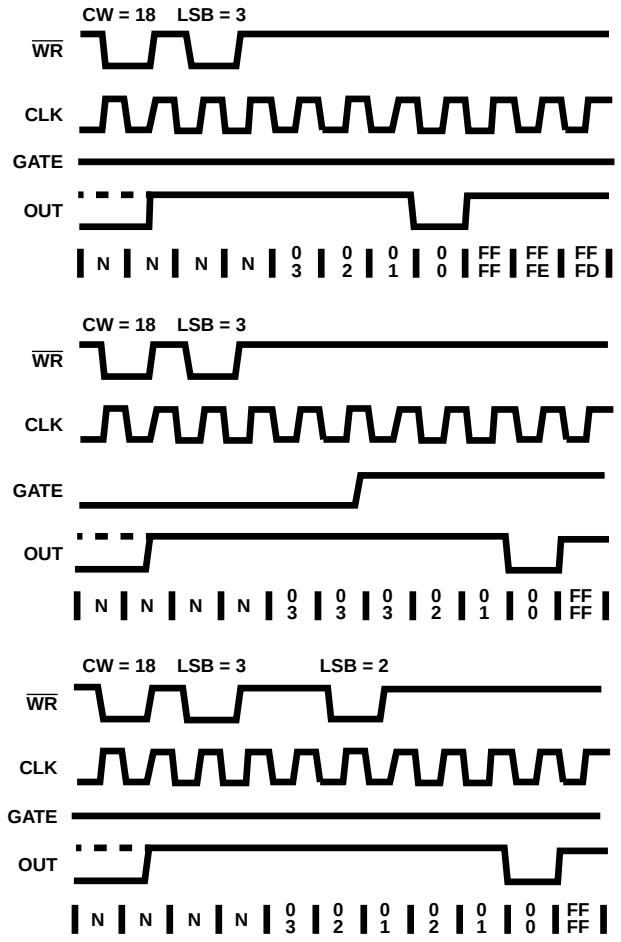


FIGURE 13. MODE 4

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.

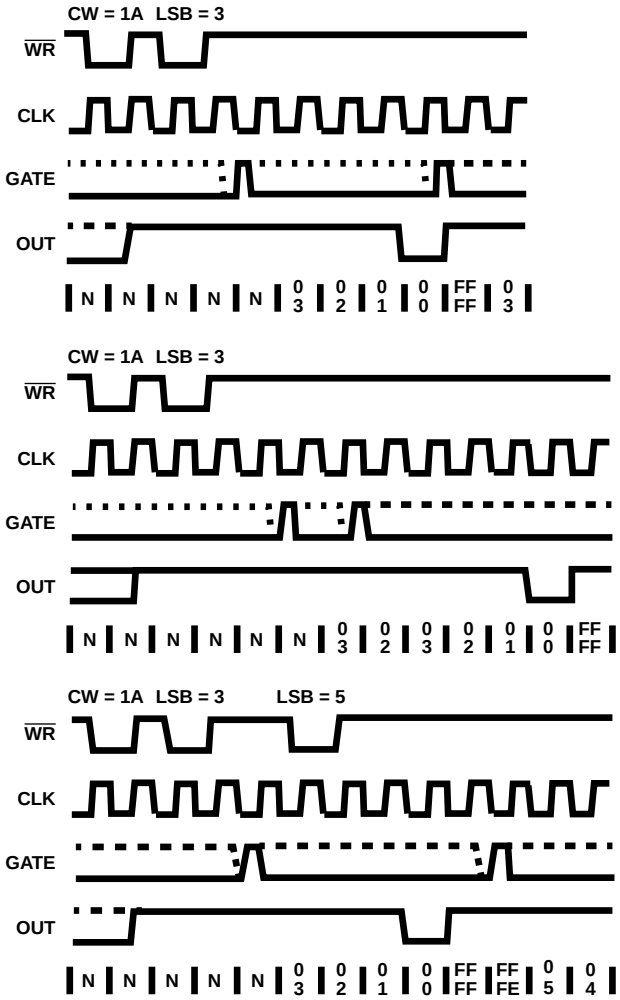


FIGURE 14. MODE 5

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