PC-166 Family

User's Manual

Second edition.

October 1993

Table of Contents

Glossary

Clock Pulse	A rising edge, then a falling edge, in that order applied to an 8254 timer/counter's clock input.
Counter Data Register	The eight-bit register of an 8254 timer/counter that corresponds to one of the two bytes in the counter's output latch for read operations and count register for write operations. The three counter data registers (CT2, CT1, CT0) appear in the PC-166 family board's register map.
Counter Gate	An input to a counter which enables or disables counting in that counter.
Counter Loading	The transfer of a count from an 8254 timer/counter's count register to its counting element.
Counter Trigger	A rising edge at an 8254 timer/counter's gate input.
DAC, D/A	Acronym and abbreviation for Digital to Analog Converter. A component of an analog output channel.
DIP	An abbreviation for Dual In-line Package.
IDC	Abbreviation for Insulation Displacement Connector.
Immediate Update	The mode where a D/A output channel is updated as soon as the digital code is written to the input latch.
Interrupt latency	The length time the host computer takes to begin executing an interrupt service routine from the time the interrupt request signal was generated.
LSB	Context sensitive abbreviation for either Least Significant Byte or Least Significant Bit
MSB	Context sensitive abbreviation for either Most Significant Byte or Most Significant Bit
Parallel Update	See synchronous update.
Programming a Counter	Writing a control word to the 8254 timer/counter's control register. The control word itself specifies which counter is being programmed.
Reference Gain	A D/A converter's output voltage range/span is determined by its voltage reference; an *n* volt reference implies a nominal *n* volt span. The reference gain is the factor by

which the output span is increased over the nominal span determined by the reference voltage.

Reference Voltage
: The voltage on which a D/A converter uses to generate its output voltage. The output voltage is commonly the reference voltage multiplied by the fraction of the D/A input digital code of the full scale digital code multiplied by the D/A's reference gain factor.

Span
: Or output voltage span. This is the total voltage range of an analog output channel. It is equal to the maximum output voltage minus the minimum output voltage. E.g. a +/-10 output range yields a span of: +10V - (-10V) = +20V

Synchronous Update
: The act of updating a number of D/A outputs simultaneously, so that the output voltages all change at the same time. Also called Parallel Update.

Update Trigger
: A signal used to transfer the digital codes from the D/A converter's input latches to its output latches.

1 Introduction

This chapter describes the PC-166 family and lists the contents of the PC-166 package.

1.1. Overview

The PC-166 is a family of analog output boards. There are six members of the family, all offering various features and functionality.

Analog output channels accept a digital code from the host processor an convert it to an analog value as required by the application in hand.

All boards have on-board timing which allow the analog outputs to be written to and/or updated at regular intervals. Flexible trigger and gating allows automatic operation in most systems without the need for external hardware or components.

A single digital input line is available; this may be used by the system to signal an external fault.

The board plugs into any fully bussed 16-bit slot of an IBM PC/AT or any compatible ISA or EISA machine, including 286, 386 or 486 based systems. It occupies sixty-four consecutive I/O addresses, in thirty-two 16-bit registers. The base I/O address can be set in the range 0h to 3FC0h with a DIP switch on the board.

Interrupts may be generated to the host computer to signal analog outputs needing more data or updating.

The PC-166 family feature wait state generation circuitry. This ensures that all host computer's I/O bus cycles are long enough for the PC-166. Zero, 1, 2, 4 or 8 additional wait states may be selected with a jumper block on the board.

The power supplies: +12V, +5V -12V, -5V and ground are available at the external connector of the board.

1.2. PC-166 Family

PC-166 family boards may be fitted with one or two types of analog output sub-systems, and various numbers of output channels.

PC-166 type boards carry 12-bit output channels. The PC-166 has 16 12-bit output channels while the PC-166B has 8 12-bit channels.

PC-266 boards carry 4 16-bit output channels.

PC-167 type boards carry both 16-bit and 12-bit output channels. The 16-bit 'outputs' are used to provide programmable references for the 12-bit outputs. The PC-167 has 16 12-bit output channels, and each group of 4 of these channels has an independent programmable reference. PC-167A boards have 16 12-bit channels, all of which share a single programmable reference. PC-167B boards have 8 12-bit channels, all of which share a single programmable reference.

For the remainder of this manual, the term PC-166x will be used to refer to a member of the PC-166 family. If the context requires it, the actual PC-166 family board will be used, for example PC-166B or PC-167A.

1.3. Features

- Up to 16 12-bit analog voltage output channels
- Full output span 20V
- Bipolar and monopolar outputs
- Each four D/A's output voltage range programmable through 16-bit resolution over entire output range of the D/A
- External asynchronous reset D/A outputs to zero lines
- Settles to $1/2$ LSB within 10us through full scale swing
- D/A data inputs double buffered
- Output modes include immediate load and update, preload and then synchronous update
- D/A update by program or interrupt
- Update triggers from software, internal clock or external clock with software or external gating
- Remote sense on all 12-bit D/As
- Outputs fully short circuit protected
- Output current boosting easily added with no sacrifice in accuracy
- 16-bit reference voltages available at the external connector
- Three independent timer/counters

- On board 2MHz reference oscillator
- Single digital input line
- For IBM-AT/286/386/486/ISA/EISA compatible and not so compatible systems
- Uses only the host computer's +5V power line
- Small board size, low power consumption
- 16-bit data transfer
- Interrupt level of each source of interrupts independently selectable from IRQ 2 to IRQ 7
- Jumper selectable wait state generator allows trouble free operation in all high speed systems
- All external connections (including all power supplies) available directly at the rear of the host computer
- Wide base address selection range; many boards may be paralleled
- Driver software provided, complete with C source code, run-time libraries and demos
- Full documentation: driver software manual and separate hardware manual

1.4. Applications

PC-166 family boards can be used:

- For instrument and tool control (e.g. plotters and chart recorders)
- As an element in process control systems
- In robotics (e.g. servo motor control)
- In automatic test equipment
- In portable diagnostic systems
- For waveform and function generation, including multi-channel in-phase and modulated waveforms
- For multiple programmable voltage sources
- In instruments and test equipment which have analog controls.
- With sensors, transducers, etc. which have analog controls.

These are a few examples of where PC-166 family boards can serve as an analog output system controller in laboratories, industry, production or part of a dedicated instrument for process monitoring and control.

1.5. PC-166 Package Contents

The PC-166 package should contain:

- PC-166 interface board and sufficient jumper caps for all possible operational configurations.
- The PC-166 User's Manual.
- A mating connector for the I/O connector (50 way).
- The Data Acquisition and Control Driver Software Manual.
- Two 5¼" 360 Kb diskettes containing PC-166 utility and demonstration software and the Data Acquisition and Control Driver Software Library.

If the package is missing any of the components or damaged in any way, then contact your distributor.

2

Installation

2.1. Configuration

There are four aspects of PC-166x boards that can be configured. These are:

- The base address. This is the address where the computer will find the board. It is set to an address between 0h and 3FC0h with the eight switches on the DIP switch block. Note that this address is in the computer's I/O address space.
- Wait state generation. If the host computer generates abnormally fast I/O bus cycles then it may be necessary to slow these down when the computer accesses PC-166x. The board can do this without affecting any other I/O bus cycles. The number of additional wait states inserted in a PC-166x bus cycle is controlled by a jumper on the board.
- The interrupt levels. There are two sources of PC-166x interrupts, and a corresponding jumper selection block for each.
- The analog output settings. The voltage reference(s) for non-programmable reference boards may be set, as well as whether the 16-bit analog voltage is allowed to reach the output connector. Note that these jumpers are normally factory set and do not usually require modification.

Setting the base address

The PC-166x uses a block of sixty-four (40h) I/O addresses. The base address setting controls the address where this block begins. This must be set so that the PC-166x does not use any addresses that are used by another device or card. If more than one PC-166x board is to be installed in the computer then each card must have a different base address setting.

The base address may be assigned to any location from 0h to 3FC0h, on sixty-four byte boundaries. The factory default base address is 280h. Refer to Table 1 below for a guide on addresses used by standard I/O devices and Table 5 in the appendices for a list of the base address switch settings. Since the board requires 64 I/O addresses, the base address must be chosen so that there is a full 64 bytes of unused addresses beginning from the selected base address.

For example, although the PC does not normally use base address 300h, the free address range from 300h only extends up to 31Fh, which is only 32 unused locations and is therefore insufficient for PC-166x boards.

The base address is set by adjusting the eight switches in the DIP switch block on the board. Switch number 1 is used to compare address line A13, number 2 for A12 and so on up to switch 8 which is used to compare address line A6. In general, if a switch is set to the *off* position, then its corresponding address weighting contributes to the base address. See Figure 1 for the DIP switch weightings.

For example switches 5 (corresponding to address line A9) and 7 (corresponding to A7) off yield a base address of 200h + 80h = 280h.

Address (hex)	Standard Device
000-1FF	Internal system board
200-20F	Games port
210-217	Expansion unit
220-24F	Reserved
250-257	Not assigned
258-25F	Intel "Above Board"
260-277	Not assigned
278-26F	Reserved
280-2EF	Not assigned (default PC-166x address)
2F0-2F7	LPT2
2F8-2FF	COM2
300-31F	Prototype board
320-32F	Hard disk
330-377	Not assigned
378-37F	LPT1
380-38F	SDLC communications
390-39F	Not assigned
3A0-3AF	Binary communications
3B0-3BF	Monochrome display adapter
3C0-3CF	Reserved
3D0-3DF	Colour graphics adapter
3E0-3E7	Reserved
3E3-3EF	Not assigned
3F0-3F7	Floppy disk
3F8-3FF	COM1
400 and up	Not used; refer to text

Table 1: Standard I/O Addresses.

I/O Addresses from 400h and up

Note that addresses from 400h and higher cannot normally be used because these addresses are not normally decoded by other I/O devices and cards in the 0h to 3FFh range.

PC-166 boards however (and most other members of the PC-XX family) **can** use these additional addresses if (and only if) there is no board or device at address 400h (or multiples of 400h) less than the address of the PC-166x **or** the board at the address 400h less also decodes the extra addresses.

A PC-166x may be installed at address 300h and another PC-166x at address 700h (400h locations apart). However it would not be advisable to install a PC-166x at address 778h because the printer port LPT1 uses a base address of 378h and does not normally decode the extra addresses.

Most other members of the PC-XX family of boards decode the extra addresses and may safely be installed 400h locations apart.

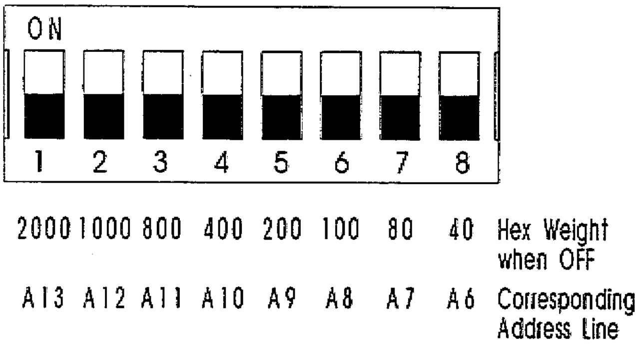


Figure 1: DIP Switch Address Weights [in hex].

If your computer has boards not listed in Table 1 installed (such as LAN adapters, back-up boards, scanner interfaces or other engineering boards) then consult the manuals for these boards for information on the address ranges used. In most cases a base address of 280h or 680h is a good choice. Address 280h is also the factory default address.

Generating additional wait states

The number of additional wait states inserted in I/O bus cycles addressing a PC-166x is controlled by the setting of the wait state jumper JP2. This is labelled on the board with the corresponding number of wait states. Refer to Figure 2 below for the wait state jumper positions.

The factory default setting is for one wait state.

PCBRDTST program

Most computers will require 1 wait state inserted, but if you think your computer can transfer information with no wait states (eg those with slower I/O busses), remove the wait state jumper and run the PCBRDTST.EXE program. This program is included on the PC-166 family Demonstration and Utility Diskette. The program will identify the board and if it reports that everything is okay after it has been run, the board may be used with no wait states. If the base address of the PC-166x board has been changed then the new base address may be supplied as a parameter to the PCBRDTST program.

For example, if the board has been set for I/O address 680 hex, enter PCBRDTST 680 to test the board.

If the PC-166x card in your computer seems to be returning incorrect readings then increase the number of wait states and re-run PCBRDTST.EXE until it reports that correct results are being obtained. If the card does not produce correct results with the maximum number of additional wait states inserted then either the host computer or the card is faulty, and should be serviced.

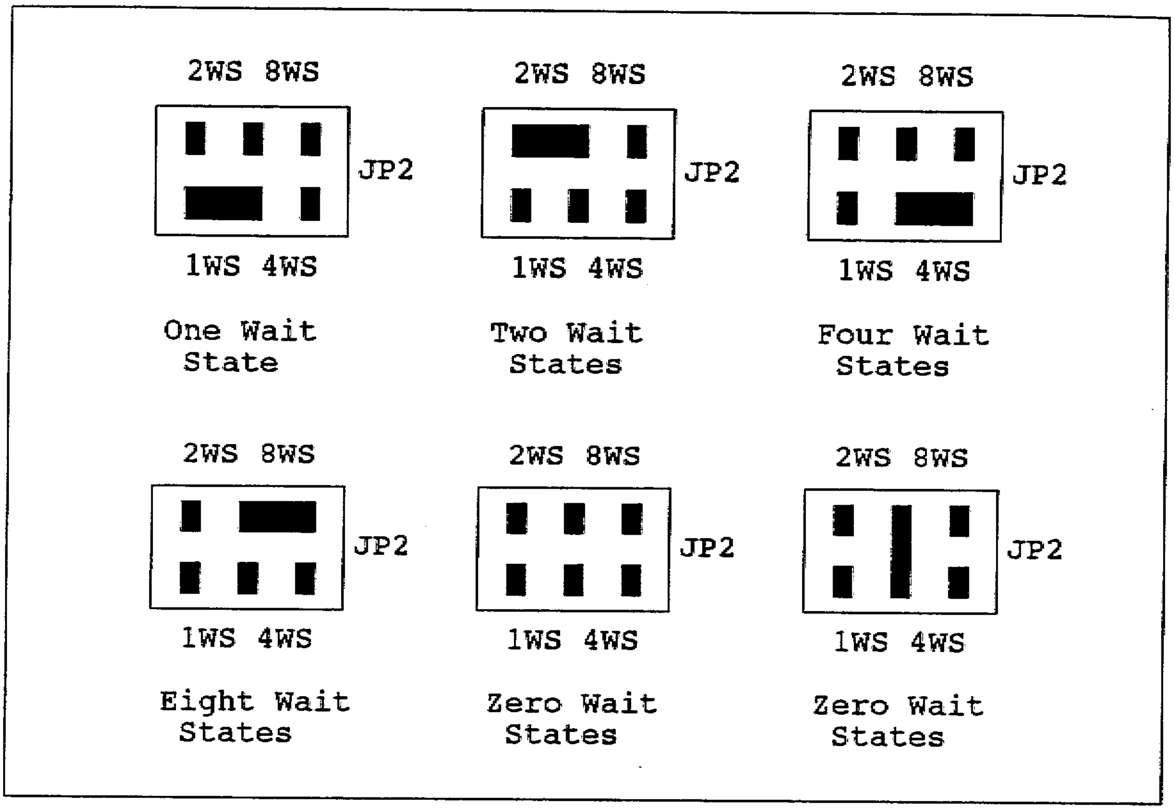


Figure 2: Wait State Jumper Settings.

PC-166x boards uses 16-bit data transfers, which on most computers default to very short access times. Hence the need for inserting the additional wait state. Any wait states inserted by the PC-166x in the computer's I/O cycle *only* affect operations to the PC-166x and **do not** affect anything else.

Even if the PCBRDTST program reports everything okay with no additional wait states, it is recommended to leave the board operating with one wait state, for maximum reliability. The additional I/O port time caused by the extra wait state is generally not significant to affect throughput.

Setting the interrupt levels

The PC-166 may generate interrupts to the host computer from:

- The output of the timing clock
- Analog output buffer empty

Full details on the function and use of the interrupts are given later in this manual.

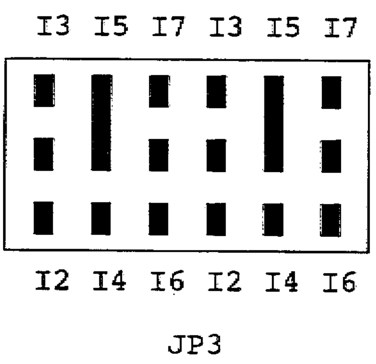
Each of the two sources of interrupts may be connected to any one of six available interrupts on the PC's I/O channel. They are IRQ2, IRQ3, IRQ4, IRQ5, IRQ6 and IRQ7. Note that on ISA systems (for example the PC AT) IRQ2 is used by the system board itself and any interrupt requests on IRQ2 are transparently routed by the system board to IRQ9. In other words, IRQ2 on the I/O channel simply becomes IRQ9 for ISA systems. See Table 2 below.

Interrupt Level	Standard Device	Available to PC-166x Board
IRQ0	System Timer	No
IRQ1	Keyboard	No
IRQ2/IRQ9	Display Adaptor (VGA)	Yes
IRQ3	Serial Port 2	Yes
IRQ4	Serial Port 1	Yes
IRQ5	Parallel Port 2	Yes
IRQ6	Floppy Disk Controller	Yes
IRQ7	Parallel Port 1	Yes

Table 2: Standard Interrupt Allocations.

When a PC-166x is shipped, all interrupt sources are set to their default level, IRQ5, as shown below in Figure 3. The leftmost jumper options select the interrupt level for the timing clock, while the rightmost jumpers in the block select the level for the output buffer empty interrupt. To change the interrupt level for an interrupt source, using jumper block JP3, remove the jumper cap from its original position and place it over the pins for another interrupt request line. Each sub-system that can generate interrupts should only have one jumper cap installed at a time. To disable an interrupt source completely, do not install the jumper cap at all, or place it over only one pin of the jumper block.

PC-166x boards can share interrupt lines with other devices and itself because it is possible to enable and disable a PC-166x's interrupt line drivers from software.



Note: leftmost jumpers select interrupt level for timing clock, rightmost jumpers select output buffer empty interrupt.

Figure 3: Interrupt Level Jumper Settings.

Adjusting the analog output settings

There are five mini-jumper blocks which are used for adjusting the analog output configuration. These jumpers are normally factory set, but in case adjustment is required the details are provided below.

D/A converter master reference

D/A converter master reference is set by JP1. Boards fitted with one or more 16-bit D/A converters (i.e. PC-167 and PC-266 boards) must apply +5V and -5V references to these converters; this is done by installing jumper caps over pins 1-2 and 3-4 (see below). Boards fitted with 12-bit D/A converters only (i.e. PC-166 boards) normally apply a +10V reference to these converters; this is done by placing jumper caps over pins 1-3 and 2-4. This provides for a maximum output voltage range of -10V to +10V. If the PC-166 type board (i.e. PC-166 or PC-166B) will be used to generate voltages in the range -5V to +5V only, its reference may be set to +5V for slightly better accuracy in the output voltage.

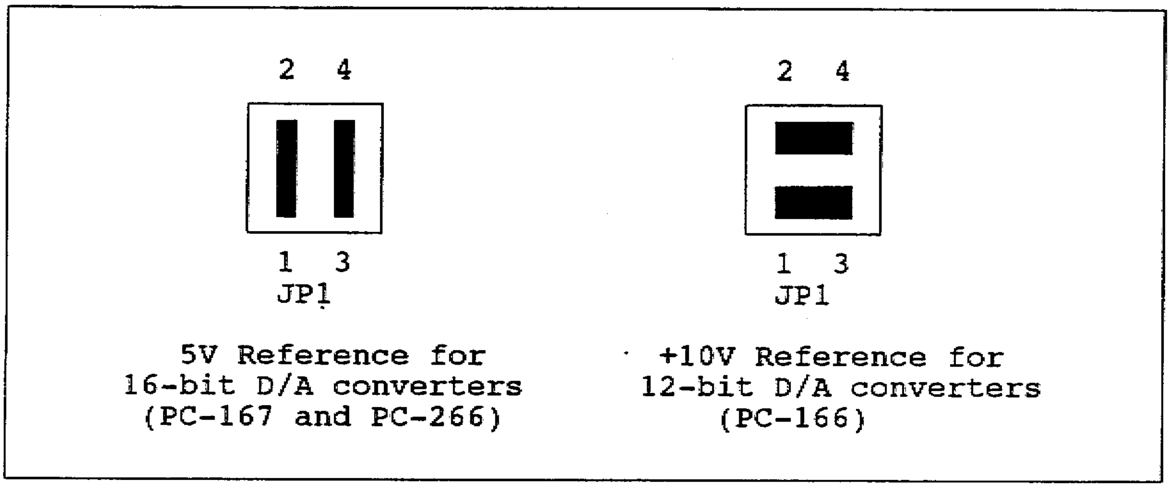


Figure 4: D/A converter master reference jumper settings.

/A Group ettings

Each of the four D/A groups has two configuration jumpers. The upper jumper selects the corresponding 12-bit D/A converter reference, while the lower jumper allows the D/A group's 16-bit D/A converter output to be connected to the output connector.

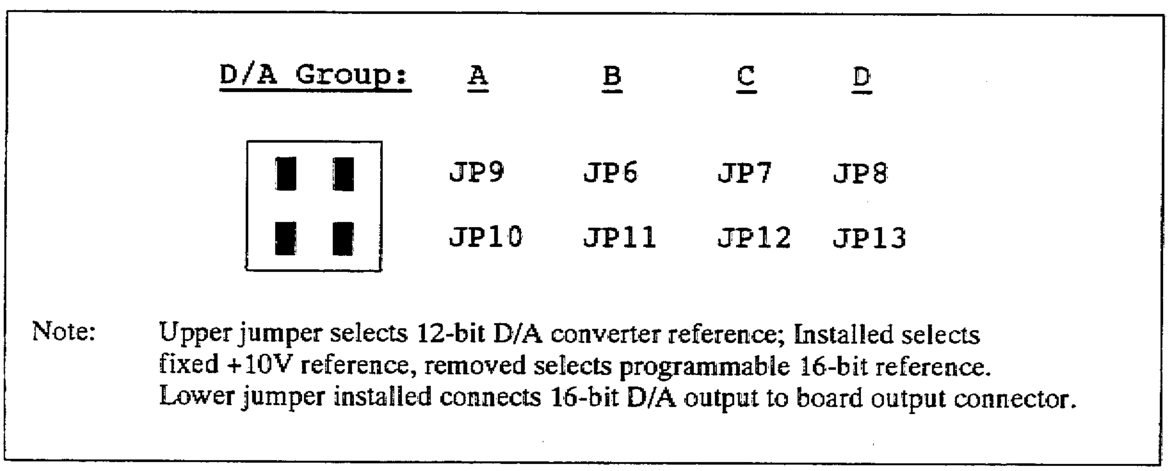


Figure 5: D/A group jumper settings.

D/A Group A

Group A has JP9 removed if the 16-bit D/A converter is present in the group and JP9 installed if it has no 16-bit D/A. ***Warning: Installing JP9 with a 16-bit D/A converter in the group may cause serious damage to the board.*** The default setting is JP9 removed on PC-167, PC-167A, PC-167B and PC-266 boards and JP9 installed on PC-166 and PC-166B boards.

If there is a 16-bit D/A converter in the group, installing JP10 will electrically connect its output to the board's output connector. The default setting is JP10 installed on PC-266 boards and removed on all other boards.

D/A Group B

Group B has JP6 removed if the 16-bit D/A converter is present in the group and JP6 installed if it has no 16-bit D/A. ***Warning: Installing JP6 with a 16-bit D/A converter in the group may cause serious damage to the board.*** The default setting is JP6 removed on PC-167 and PC-266 boards and JP6 installed on PC-166, PC-166B, PC-167A and PC-167B boards.

If there is a 16-bit D/A converter in the group, installing JP11 will electrically connect its output to the board's output connector. The default setting is JP11 installed on PC-266 boards and removed on all other boards.

D/A Group C

Group C has JP7 removed if the 16-bit D/A converter is present in the group and JP7 installed if it has no 16-bit D/A. ***Warning: Installing JP7 with a 16-bit D/A converter in the group may cause serious damage to the board.*** The default setting is JP7 removed on PC-167 and PC-266 boards and JP7 installed on PC-166, PC-166B, PC-167A and PC-167B boards.

If there is a 16-bit D/A converter in the group, installing JP12 will electrically connect its output to the board's output connector. The default setting is JP12 installed on PC-266 boards and removed on all other boards.

D/A Group D

Group D has JP8 removed if the 16-bit D/A converter is present in the group and JP8 installed if it has no 16-bit D/A. ***Warning: Installing JP8 with a 16-bit D/A converter in the group may cause serious damage to the board.*** The default setting is JP8 removed on PC-167 and PC-266 boards and JP8 installed on PC-166, PC-166B, PC-167A and PC-167B boards.

If there is a 16-bit D/A converter in the group, installing JP13 will electrically connect its output to the board's output connector. The default setting is JP13 installed on PC-266 boards and removed on all other boards.

Improving reference noise

The output of the 16-bit D/A converter(s) is not normally connected to the board's output connector (except for PC-266 boards). This is to reduce noise pickup when the 16-bit D/A converter is being used as a programmable reference for a 12-bit D/A quad in a group. If it is necessary to use the 16-bit voltage at the output connector then the corresponding jumper may be installed at any time without risk of damaging the board.

2.2. Calibration

PC-166x boards are calibrated at the factory, but it is recommended to re-calibrate the board on first installation. All that is required is a precision voltmeter and a small flat screwdriver. If two voltmeters are available, the process is made even easier.

There is only one device on the PC-166 board which requires calibration and this is the board's master reference. This is done with trim-pots R1 (to adjust the reference's gain) and R2 (to adjust the balance point). Calibration is as follows:

1) Make sure the D/A master reference jumper JP1 is set correctly for the desired master reference voltage. This is described in the previous section, and is normally set at the factory. Two master reference voltage values are possible, a bipolar +/-5V reference normally used with 16-bit output channels and a unipolar +10V reference, used only with 12-bit output channels. For the +/-5V reference, it is necessary to adjust the span (10V) and balance or centre point (0V), so that VRef+ and VRef- are +5V and -5V respectively. For the 10V reference, it is necessary only to adjust the span (10V), so that VRef+ is +10V. The VRef voltages are measured with respect to analog ground (pin 9 of U1).

2) +/-5V only: Connect the input of the voltmeter to pin 1 (VRef+) and the common to pin 15 (VRef-) of U1 (an AD588).

3) +/-5V only: Adjust R1 to produce exactly 10.0000V between these two pins.

4) Connect the common of the voltmeter to pin 9 (analog ground) of U1.

5) +/-5V: Adjust R2 (balance) to produce exactly 5.0000V between these two pins.

   10V: Adjust R1 (gain) to produce exactly 10.0000V between these two pins.

6) +/-5V: To check the balance, connect the input of the voltmeter (or connect up the second voltmeter) to pin 15 (VRef-) of U1. It should read -5.0000V.

7) +/-5V: Repeat steps 3, 5 and 6 until the voltage on pin 1 (VRef+) is 5.0000V and the voltage on pin 15 (VRef-) is -5.0000V.

   10V: After step 5, the voltage on pin 1 (VRef+) should be 10.0000V.

2.3. Installation

Installing a PC-166x card is straightforward. You will need a screwdriver to match the screws on the computer's cover and expansion slot bracket (normally a Philips type). Make sure that the board is configured correctly (see above).

a) Switch off the computer and all attached devices.

b) Unplug the power cord from the computer and all attached devices. Failure to do this may result in hazardous conditions, as there may be dangerous voltage levels present on externally connected cables.

c) Remove the top cover of the PC or the access port to the I/O channel. If you are not sure how to do this, consult the manual supplied with the system unit.

d) Choose any unused **16-bit** expansion adapter slot and remove the screw from the top of the blank bracket corresponding to the chosen slot. It is recommended to use a slot as far away from the PC's power supply as possible. Remove the bracket.

e) Align the gold plated edge connector of the PC-166x with the edge socket on the computer system board and align the board bracket with the rear adapter slot on the PC case. Firmly press the board down into the edge socket. Ensure that the board's edge connector is seated in the socket and has not slipped sideways past the socket.

f) Tighten the mounting bracket of the PC-166x to the back panel rail of the computer.

g) Replace the computer's cover. Plug in all cords and cables. Switch on the power. The PC-166x board is now installed and ready for operation.

3 Interconnections

The PC-166x board plugs into any of the computer's 16-bit expansion slots, along the board's gold plated edge connector. The board communicates to the external world via a connector mounted in its bracket. This chapter describes these connectors and the necessary functions of these external lines. It also describes recommended analog output connection schemes.

3.1. Connections to the Computer Backplane

A PC-166x board may be plugged into any **16-bit** slot of the computer backplane. All communication to and from the host computer is via this connector. The PC-166x board uses a block of 64 addresses in the host computer system's I/O address space.

If the PC-166x board sometimes behaves erratically then there is the possibility that the gold plated contacts on the edge connector may have become dirty or abraded, especially if the card is installed and removed many times from various different computers. This condition can be corrected simply by cleaning the contacts with an ordinary pencil eraser.

3.2. User I/O Connector

The PC-166/167/266 interfaces to the external world via a 50-pin male ribbon cable header mounted in the board's bracket.

The 50-pin connector carries the following signals, depending on the type of PC-166:

- The 12-bit analog outputs from channels 0 to 15 (CHx).
- The remote sense inputs for channels 0 to 15 (RFBx).
- Each D/A quad's reset line (RSTDx).

- The 16-bit analog outputs from channels 16 to 19 (CHx), corresponding to the group's programmable reference voltages.
- External update trigger input (EXTRIG).
- External gate inputs for the timing clock and trigger clock (EXGx).
- A single digital input line.
- Power supplies of +5V, -5V, +12V, -12V as well as analog and digital ground.

Figure 6 shows these connections, to which board type they apply, together with their pin assignments. Note that the pin connections refer to the pin numbers of the connectors when looking into the connector from the rear of the computer. Pin number 1 is marked with an arrow on the box surrounding the pins of the 50-way header connector.

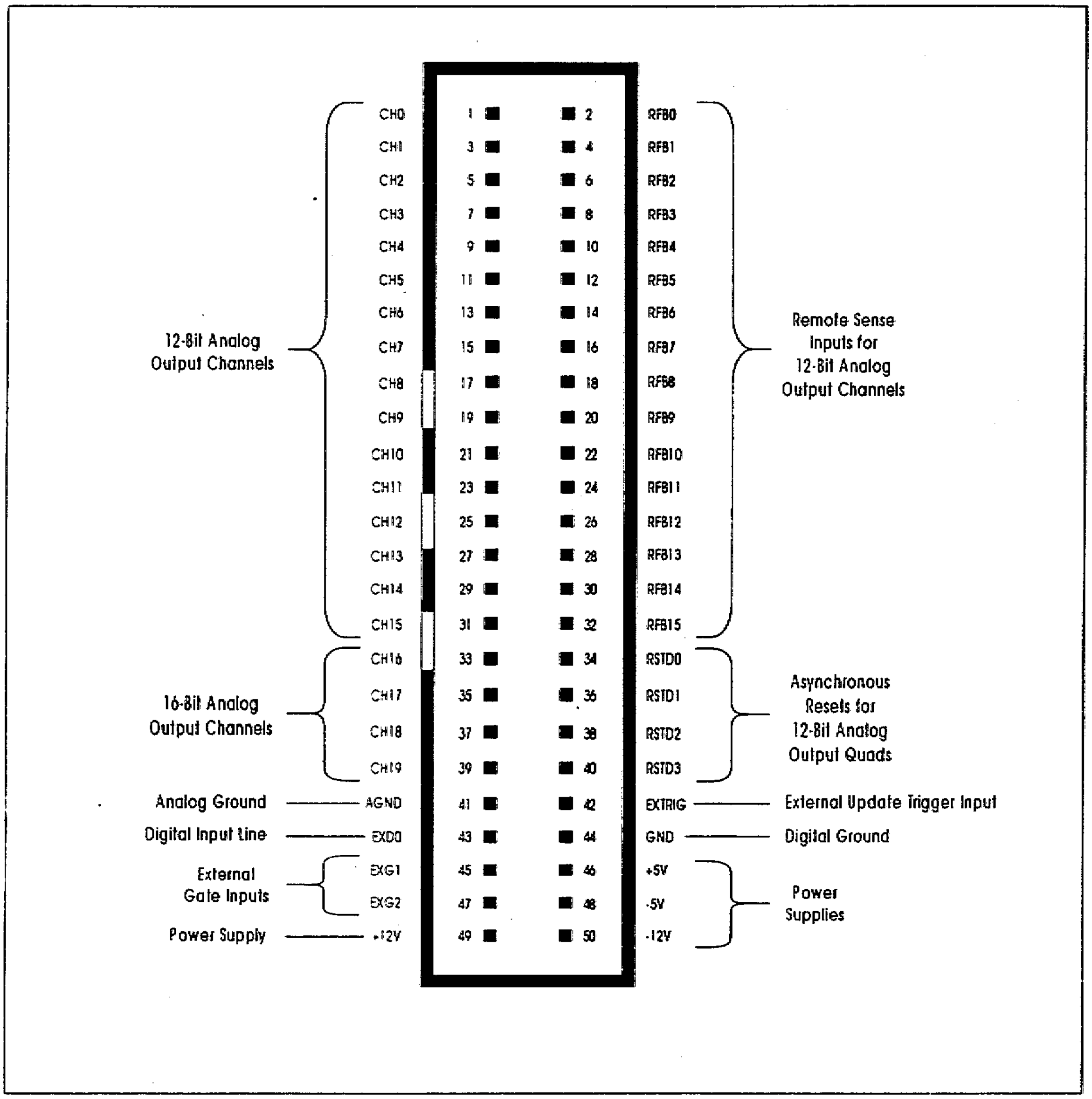


Figure 6: PC-166 family connector, as seen from the rear of the PC.

Power supplies

The +12V, -12V, +5V and -5V power supplies are available, together with analog and digital ground, from the I/O connector. These may be used to power external circuitry, if necessary. As a rough guide, no more than about 250mA should be drawn from these power supplies or the PC-166x board, or the computer or the power supply may be permanently damaged. Check the rating of the host computer's power supply for details on maximum current loading, especially on the -5V and -12V lines.

If at all possible, try not to use these power supplies if the PC-166x is being used in an application which is sensitive to noise. The overall accuracy of the board cannot be guaranteed and may be affected by the behaviour of external circuitry if it uses power from the PC-166x board.

3.3. Analog Output Connection Guidelines

PC-166/167/266 boards are high accuracy, precision analog output boards and thus great care must be taken when interfacing them to the external world. The host PC too, is an extremely good source of noise of a wide range of frequencies, and these easily find their way onto sensitive analog outputs. What follows are general guidelines which should be followed if the board is expected to perform to its specified accuracy.

The I/O connector is a 50-way IDC (insulation displacement) connector for ease of assembly, however they are prone to noise if not used carefully. Individually shielded ribbon cable **must** be used. The shield must be connected to the case ground (and **not** analog ground) at **one** end of the cable only.

The cable must be kept as short as possible, ideally no more than a few tens of centimeters long. This will prevent RF pickup, coupling from adjacent sources and voltage drops on the line. If long lines must be used, then the remote sense input to the corresponding D/A (12-bit only) may be used to overcome voltage drops. See Figure 7 below for a schematic. If the remote sense input is not used then it **must** be connected to the corresponding channel's output pin.

Ideally, the board should be situated in the slot furthermost from the host PC's power supply, at the very edge of the system unit or expansion unit case, and all other cards which may be a source of noise (eg fax/modem cards) be situated as far away as possible from the PC-166 board. The external cable should be routed directly away from the computer and ideally not run along any other cables or power leads.

Do not to use the power supplies on the I/O connector to power external circuitry. These are intended for quick starting during system development and where the external system can tolerate slightly reduced specifications.

Boards with 16-bit programmable reference voltages have these references disconnected (by jumpers) from the output connector. For maximum noise immunity, these references should remain disconnected since any noise on the reference finds its way onto the output voltages.

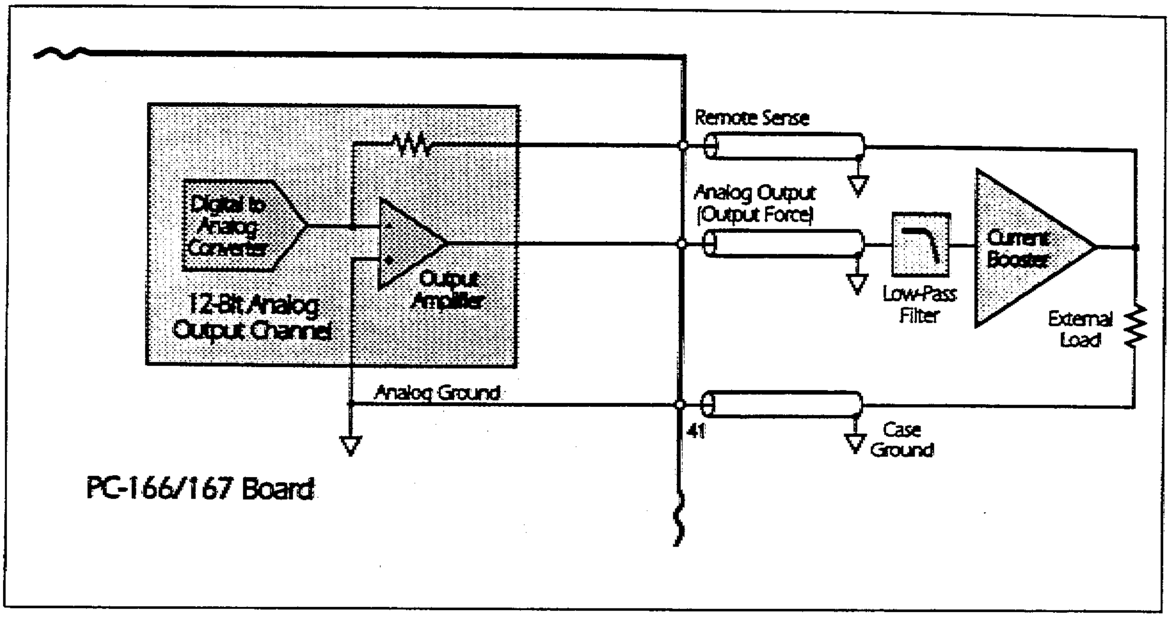


Figure 7: Current boosting with remote sense.

Always use analog ground for the ground reference for the analog outputs and **not** digital ground. Digital ground and the case/system ground must **never** be connected to analog ground, either directly or through some external device which may connect them or reference them internally.

The analog outputs do **not** have filtering to roll off the output signal at the maximum frequency that applies to the external system. This is because this maximum frequency may vary from system to system (or even from output to output), the characteristics of the roll-off vary considerably according to the characteristics of external system (from no roll-off needed to many orders of roll-off) and the physics of the external circuitry. Appropriate filtering **must** be used on output channels for accurate results.

4

Architecture

This chapter describes the architecture of PC-166x boards. There are three functional sub-systems on the board: the analog output groups, the timing system and single digital input line. In addition to these, there are some support sub-subsystems and an I/O connector. This chapter provides a brief overview of all these sub-systems, and explains their function and operation. Each part is described fully in appropriate sections later in this manual.

The block diagram below illustrates the overall architecture of the PC-167. PC-166/266 and A and B variants will contain a subset of the full PC-167's features.

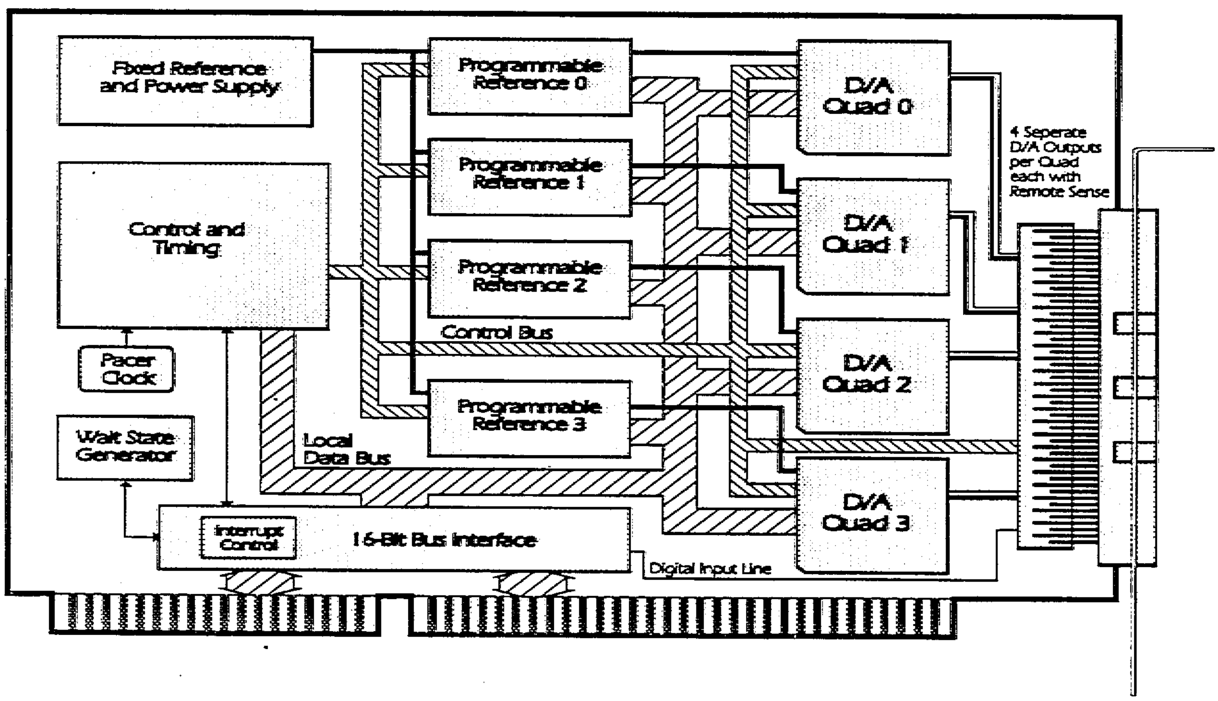


Figure 8: PC-166/167/266 Block Diagram.

4.1. Analog Output Groups

12-bit Analog Outputs

There are 16 12-bit analog outputs available from PC-166, PC-167 and PC-167A boards. The B variants have 8 12-bit analog outputs. Each analog output has a remote sense input which may be used for current boosting or overcoming voltage drops over long leads. The outputs can source or sink 5 mA and are fully short circuit protected.

Each analog output may be programmed fully from software. Bipolar or monopolar mode may be set, a reference voltage gain or 1 or 2 and whether the D/A output is to be updated immediately or synchronously with other output channels using an update trigger.

D/A Quads

The 16 12-bit analog outputs, are arranged in four **Quads**, each consisting of 4 analog outputs. Each quad has a dedicated external reset input, which may be used by the external system to reset the quad's outputs in case of a fault condition.

16-bit Analog Outputs

PC-167 and PC-266 boards have 4 16-bit D/A converters. On the PC-167 they are used as the control for programmable references to each of the D/A quads. On PC-266 boards, they are available as 4 16-bit analog output channels. The A variant boards have one 16-bit D/A converter installed.

4.2. Timing System

The timing system consists of a reference oscillator, three counters, two gate selection multiplexers and a trigger selection multiplexer all under control of the board control and software trigger registers.

An 8254 timer/counter device integrated into PC-166x boards provides the three independent timer/counters. The 2 MHz reference oscillator is divided down by the first timer, the prescaler. The second timer generates the D/A converter ready signal while the third timer generates D/A update triggers.

Counting/triggering is performed using a counter's gate input, which may come from the software in hand or an external source, as selected by the gate selection multiplexers. The counter's output pin indicates the status/result of counting.

The computer program may latch and read the current count at ant time.

There are six different timing/counting modes available.

The D/A converter ready signal from the timing clock may generate interrupt requests to the host computer.

Analog output update triggers may be chosen by the trigger selection multiplexer to come from the software, the update clock or an external source. These may also be enabled onto the host computer's bus to cause interrupt requests.

4.3. Digital Input Lines

PC-166 family boards have a single dedicated digital input line. If the external gate inputs are not used, then each of these lines may also be used as additional digital inputs.

4.4. Support Sub-Systems

Bus Interface

The bus interface provides for the base address of the PC-166x to be set, using an on-board DIP switch, within a wide range of PC I/O addresses.

It also provides bus transceivers and buffers which maintain proper bus flow discipline between the PC-166x and the host computer.

In the event of the host computer's I/O bus cycles being abnormally short, this sub-system is, under control of a selection jumper, able to insert additional wait states into any I/O cycle that is directed at a PC-166x board.

Interrupt Control

There are two sources of interrupts on PC-166x boards. This sub-system controls routing and enabling of the interrupt sources.

A source of interrupts from a device on a PC-166x is routed, via selection jumpers, to an interrupt request line on the PC I/O channel.

All interrupts from the board can be enabled or disabled from software by setting appropriate flags in PC-166x registers.

Board Control and Power Supply

This is an internal sub-system that is responsible for the general housekeeping of the board. Part of it ensures the correct power supply voltages for the analog circuitry. The board uses only the host computer's +5V supply, from which it generates a clean 15V analog supply.

4.5. External I/O Connector

All analog I/O lines, timing control lines and digital lines, together with the power supplies, are conveyed through a standard 50-way male IDC header connector. This allows quick cable assembly and mass termination.

5

Register Structure

At the lowest level, a PC-166x is programmed using I/O port input and output instructions. This chapter contains information on all of PC-166x registers. Although programming the board is not difficult, it is time consuming and requires detailed knowledge of both the PC-166x board and the host PC's operating system. This manual provides the former. As an alternative, a set of software library functions is provided. These cover most of the board's features and should suit most applications. The Driver Library is described in a companion manual to this. In case routines need to be custom written for a specific purpose, it is often quicker to base them on the driver system, by loading and modifying routines from the driver library source code. This chapter and Chapter 6: *Programming Guide* provide full information on how this is done.

The PC-166x board occupies 64 consecutive addresses in the computer's I/O space. Since the board has a bus size of 16-bits, the 64 addresses are used for 32 16-bit registers. Depending on the variant of PC-166x, some of the registers may not be used, they are therefore reserved. The layout of these registers is shown in Table 3: *PC-166 Family Register Structure*. This table also gives the page number of this manual on which the register is described. The offset of each register is given as an offset address from the base address of the board. The board's base address is set with the DIP switch as detailed in Chapter 2: *Installation*.

Quad Mode and Analog Output Data Registers [12-bit] (offsets 0, 8, 16 and 24: read/write)

These registers perform two functions, depending on the setting of the **quad mode select** bit (MS) in the PC-166 control register (CTRL, see below). The first function, if the quad mode select bit is clear, is to write the actual binary value corresponding to the desired analog output voltage for channels 0, 4, 8 and 12. If the quad mode select bit is set then the bits in this register are used to configure the corresponding D/A quad. The contents of this register may be read back at any time, and if the quad mode select bit is set, the quad's configuration may be read back.

Offset	Write	Read	Page No
0	Quad 0 Mode and Channel 0 Data Register	Quad 0 Mode and Channel 0	29
2	Channel 1 Data Register	Channel 1 Data Readback	31
4	Channel 2 Data Register	Channel 2 Data Readback	31
6	Channel 3 Data Register	Channel 3 Data Readback	31
8	Quad 1 Mode and Channel 4 Data Register	Quad 1 Mode and Channel 4	29
10	Channel 5 Data Register	Channel 5 Data Readback	31
12	Channel 6 Data Register	Channel 6 Data Readback	31
14	Channel 7 Data Register	Channel 7 Data Readback	31
16	Quad 2 Mode and Channel 8 Data Register	Quad 2 Mode and Channel 8	29
18	Channel 9 Data Register	Channel 9 Data Readback	31
20	Channel 10 Data Register	Channel 10 Data Readback	31
22	Channel 11 Data Register	Channel 11 Data Readback	31
24	Quad 3 Mode and Channel 12 Data Register	Quad 3 Mode and Chan 12	29
26	Channel 13 Data Register	Channel 13 Data Readback	31
28	Channel 14 Data Register	Channel 14 Data Readback	31
30	Channel 15 Data Register	Channel 15 Data Readback	31
32	Reference 0 or Channel 16 Register (REF0)	Ref 0 or Channel 16 (REF0)	32
34	Reference 1 or Channel 17 Register (REF1)	Ref 1 or Channel 17 (REF1)	32
36	Reference 2 or Channel 18 Register (REF2)	Ref 2 or Channel 18 (REF2)	32
38	Reference 3 or Channel 19 Register (REF3)	Ref 3 or Channel 19 (REF3)	32
40	Channel 0 to 15 Update Mode (UPDMODE)	---	32
42	Control/Status Register (CTRL)	Status Readback	33
44	Software Trigger Register (STRIG)	---	35
46	Reserved	---	
48	Prescaler Data (Counter/Timer 0) (CT0)	Prescaler Data	36
50	Timing Clock Data (Counter/Timer 1) (CT1)	Timing Clock Data	36
52	Trigger Clock Data (Counter/Timer 2) (CT2)	Trigger Clock Data	37
54	Counter/Timer Control Register (CTCTRL)	Counter Control (CTCTRL)	37
56	Reserved	---	
58	Reserved	---	
60	Reserved	---	
62	Reserved	---	

Table 3: PC-166 Family Register Structure.

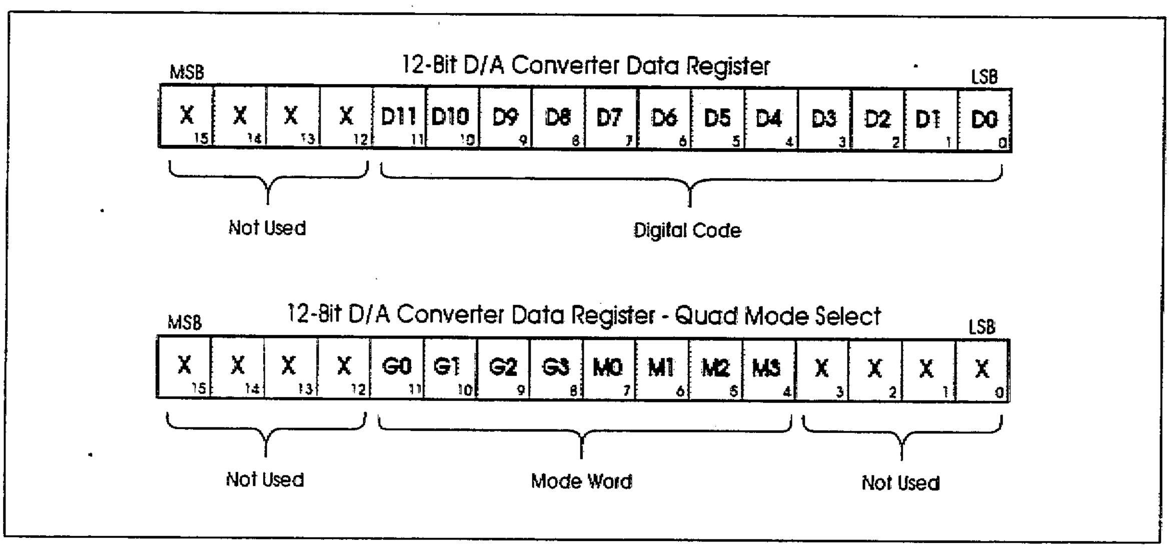


Figure 9: Quad Mode and Analog Output Data Register

Analog Output Data Mode

Bits 15-12 : These bits are not used and are undefined.

Bits 11-0 : The bits D11 (MSB) down to D0 (LSB) correspond to the 12-bit binary value corresponding to the channel's output voltage. See Chapter 6: *Programming Guide* for how to calculate input codes corresponding to output voltages.

Quad Mode Select Mode

If the quad mode select bit has been set in the PC-166 control register, then the bits D11 to D4 specify the reference gain and output mode of the corresponding D/A converter in the quad.

Bits 15-12 : These bits are not used and are undefined.

Bits 11-8 : Bit G0 sets the first channel in the quad's gain while G3 sets the fourth channel. A gain setting of 0 (Gx=0) configures channel x to have a reference gain of 1, while a gain setting of 1 configures the channel to have a gain of 2.

Bits 7-4 : Bit M0 sets the quad's first channel's output mode and bit M3 sets the fourth channel's mode. A mode setting of 0 (Mx=0) configures channel x for monopolar output, while a mode setting of 1 selects bipolar output.

Bits 3-0 : These bits are not used and are undefined.

Analog Output Data Registers [12-bit] (offsets 2, 4, 6, 10, 12, 14, 18, 20, 22, 26, 28 and 30: read/write)

These registers are used to write the actual binary value corresponding to the desired analog output voltage for channels 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14 and 15. These registers have the same format as the Quad Mode and Analog Output Data Registers when the quad mode select bit is clear. See Chapter 6: *Programming Guide* for how to calculate input codes corresponding to output voltages. The contents of this register may be read back at any time.

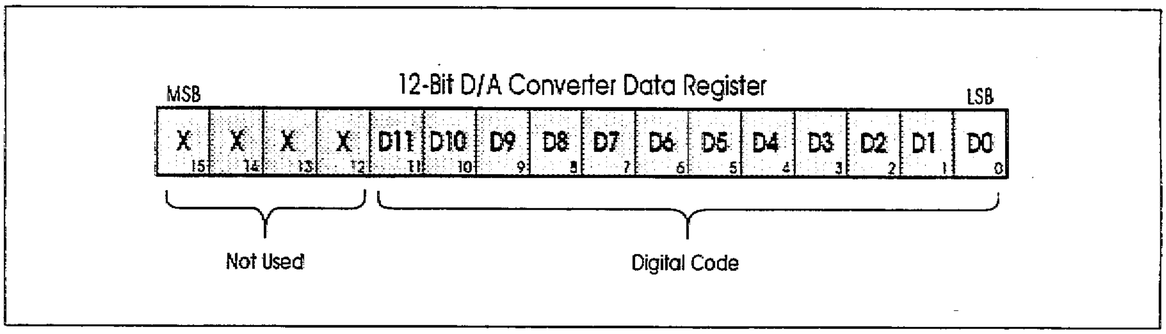


Figure 10: Analog Output Channel Data Register

Bits 15-12 : These bits are not used and are undefined.

Bits 11-0 : The bits D11 (MSB) down to D0 (LSB) correspond to the 12-bit binary value corresponding to the channel's output voltage. See Chapter 6: *Programming Guide* for how to calculate input codes corresponding to output voltages.

REF0 to REF3 - Analog Output Data Registers [16-bit] (offsets 32, 34, 36 and 38: read/write)

These registers are used to write the actual binary value corresponding to the desired analog output voltage (on PC-266 boards) or the programmable reference voltage for the 12-bit quads (on PC-167 boards). In both cases, these registers correspond to analog channels 16, 17, 18 and 19.

PC-167 On PC-167 boards, channel 16 controls the reference voltage for DAC group A (which includes channels 0, 1, 2 and 3); channel 17 controls the reference voltage for DAC group B (which includes channels 4, 5, 6 and 7); channel 18 controls the reference voltage for DAC group C (which includes channels 8, 9, 10 and 11) and channel 19 controls the reference voltage for DAC group D (which includes channels 12, 13, 14 and 15).

PC-167A On PC-167A boards, channel 16 controls the reference voltage for all 12-bit output channels (channels 0 to 15). Channels 17, 18 and 19 are not present.

PC-167B On PC-167B boards, channel 16 controls the reference voltage for all 12-bit output channels (channels 0 to 8, where channels 9 to 15 are not present). Channels 17, 18 and 19 are also not present.

If the channel is used to control reference voltages, then the reference voltage will be the actual analog voltage corresponding to the digital input code. See Chapter 6: *Programming Guide* for how to calculate input codes corresponding to reference or output voltages. The contents of this register may be read back at any time.

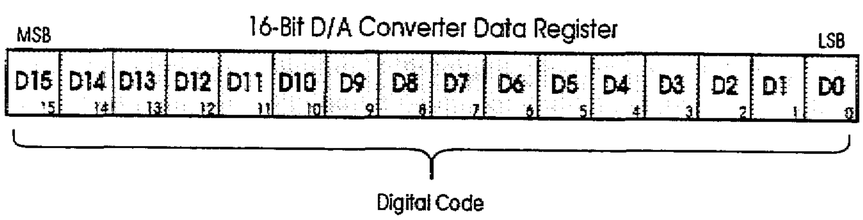


Figure 11: 16-Bit Analog Output Data Register

Bits 15-0 : The bits D15 (MSB) down to D0 (LSB) correspond to the 16-bit binary value corresponding to the channel's output voltage.

UPDMODE - Channel 0 to 15 Update Mode Register (offset 40: write only)

This register is used to specify the update mode for each of the 12-bit output channels. Each channel may either operate in immediate update mode or synchronous (or parallel or preload) update mode. In immediate mode, the output voltage changes as soon as a digital code is written to a D/A converter's input data register. In synchronous mode, one or more digital codes are preloaded into the one or more D/A converter's input latches and only transferred to their output registers when an update trigger is received. Hence the outputs of many D/A converters may change simultaneously when a trigger is received. On power-up or reset, all bits are reset to zero.

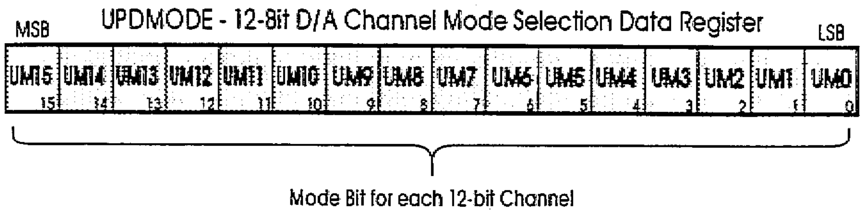


Figure 12: Update Mode Register for 12-Bit Analog Output Channels

Bits 15-0 : The bits UM15 (MSB, corresponding to channel 15) down to UM0 (LSB, corresponding to channel 0) correspond to the update mode setting for each channel. Specifying a UMx bit to be 0 causes channel x to updated immediately, while specifying UMx to 1 invokes synchronous mode for channel x.

UM_x	Channel Update Mode
0	Immediate
1	Synchronous

CTRL - Board Control Register (offset 42: read/write)

The board control register is used to write configuration information to the PC-166 board and read back status information from the board. It is used to select the gate source of the board timers; control a timer's gates if a software gate has been selected; enable interrupts from the board and set the source of D/A update triggers. Reading from this register provides status information on all the configuration that may be set above plus the status of the external gate lines; the status of the D/A buffer empty flag; the status of the trigger error flag and the status of the digital input line.

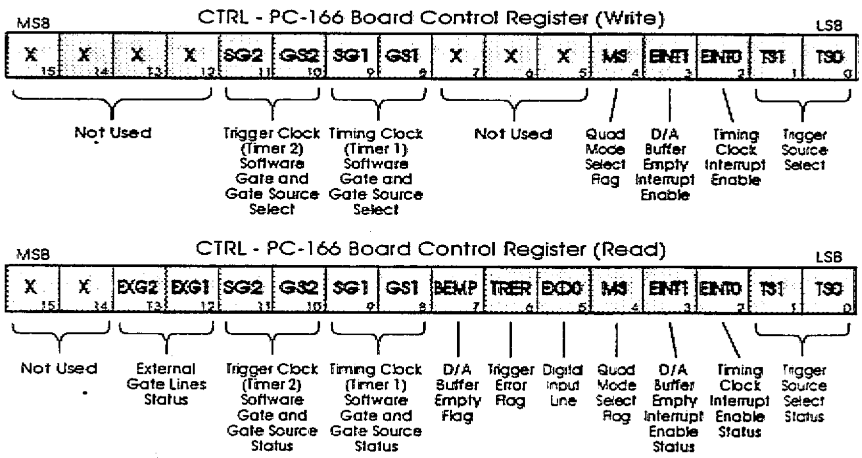


Figure 13: PC-166 Board Control Register

On power-up or reset, all bits are reset to zero, except bit 4, which is undefined under these conditions. Unused bits are reserved and read back undefined. Zero should be written to these bits.

Bits 15-14 : These bits are not used.

Bits 13-12 : These bits indicate the status of the external gate input lines. EXG1 is the status of the external gate input to timer 1, the timing clock. If the external gate has been enabled (with GS1 in this register) for timer 1 then, if this line is pulled high by an external device, the timer is enabled. If it is low, the timer is inhibited from counting. The trigger clock, timer 2, is controlled in similar manner by EXG2 and GS2. Note that if any of the external gate lines are not being used for controlling a timer's gate, then they may be used as additional digital input lines.

Bit 11 : Software Gate for the trigger clock (timer 2) (SG2): If the gate source bit for this timer (GS2) specifies a software gate (by being clear) then the gate of the timer is controlled by this bit. If the bit is set, then the timer is enabled whereas if it is clear, the timer is inhibited from counting.

Bit 10 : Gate Source for the trigger clock (timer 2) (GS2): This bit controls the source of the gate input of timer 2; if the bit is clear then a software gate is selected (see bit 11, SG2), whereas if it is set then the external gate input for this timer is selected.

GS2	Trigger Clock (Timer 2) Gate Source
0	Software gate
1	External gate input via EXG2

Bit 9 : Software Gate for the timing clock (timer 1) (SG1): If the gate source bit for this timer (GS1) specifies a software gate (by being clear) then the gate of the timer is controlled by this bit. If the bit is set, then the timer is enabled whereas if it is clear, the timer is inhibited from counting.

Bit 8 : Gate Source for the timing clock (timer 1) (GS1): This bit controls the source of the gate input of timer 1; if the bit is clear then a software gate is selected (see bit 9, SG1), whereas if it is set then the external gate input for this timer is selected.

GS1	Timing Clock (Timer 1) Gate Source
0	Software gate
1	External gate input via EXG1

Bit 7 : D/A Buffer Empty Flag (BEMP): This bit is set when the D/A sub-system receives an update trigger, thus indicating that new data may be written to the D/A converter's input data registers. The update trigger transfers the previous data written to the D/A input registers to their output registers. The BEMP flag is cleared by a write to any of the 12-bit D/A converter data registers. If the EINT1 bit is set in this register, then when the BEMP flag is set, an interrupt is generated too. Writing to this bit has no effect.

Bit 6 : D/A Trigger Error Flag (TRER): This bit is set when the D/A sub-system receives an update trigger before the D/A input registers have received new output data. It is cleared by a read from this register. If the flag is set by the board, it means that the operating program or interrupt service routine did not write new data to the D/A registers in sufficient time before the next update trigger was received. Writing to this bit has no effect.

Bit 5 : Digital Input Line 0 (EXD0): This bit reflects the status of the digital input line. Writing to this bit has no effect.

Bit 4 : Mode Select (MS): The system software sets this bit to indicate that the next write operation to one of the quad mode and analog output data registers will be to set the mode (gains and output ranges) of that quad's output channels. See the quad mode and analog output data register earlier in this chapter for the mode settings.

Bit 3 : Enable Interrupt Source 1 (EINT1): This bit is used to control the D/A buffer empty interrupt. If it is set, then the interrupt is enabled onto the host computer's I/O bus; thus whenever the D/A buffer empty flag becomes set by the board, an interrupt request will be sent to the host computer. Clearing the EINT1 bit disconnects the interrupt from the computer's bus.

Bit 2 : Enable Interrupt Source 0 (EINT0): This bit is used to control the timing clock interrupt. If it is set, then the interrupt is enabled onto the host computer's I/O bus; thus whenever the timing clock output goes from low to high, an interrupt request will be sent to the host computer. Clearing the EINT0 bit disconnects the interrupt from the computer's bus.

Bits 1-0 : Trigger Source Select (TS): These two bits select the source of update triggers for the 12-bit analog output channels. The trigger source is shown by the table below:

TS_1	TS_0	Update Trigger Source
0	0	Software trigger via STRIG register
0	1	Hardware trigger from trigger clock
1	0	External hardware trigger
1	1	Reserved, do not use

STRIG - Software Trigger Register (offset 44: write only)

When software update triggers are selected with the TS1 and TS0 bits (both 0), a write to bit 0 of this register causes an update trigger. This trigger updates those 12-bit channels programmed in synchronous update mode. Reading from this register is not defined.

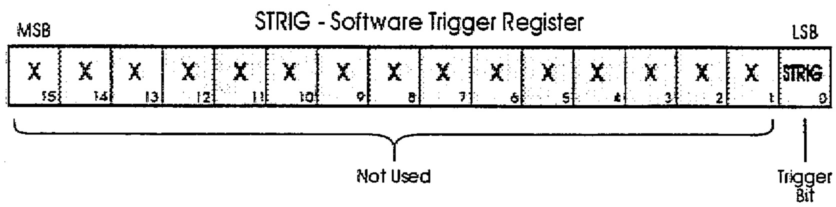


Figure 14: Software Trigger Register

Bits 15-1 : The bits are not used and are undefined.

Bit 0 : Writing a 1 to this bit causes an update trigger to occur.

CT0 - Prescaler Data Register (offset 48, read/write)

This register is the data register of the first board timer/counter. This counter is used to scale the on-board 2 MHz reference clock for the timing and trigger counters.

Before reading or writing from this register, a control word for this counter must be written to the timer/counter control register, CTCTRL. Then reads and writes to this register must follow the format specified in the control word.

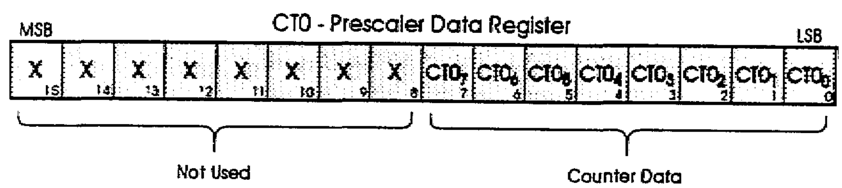


Figure 15: Prescaler Data Register

Bits 15-8 : These bits are not used and are undefined.

Bits 7-0 : The bits $CT0_7$ (MSB) down to $CT0_0$ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

CT1 - Timing Clock Data Register (offset 50, read/write)

This register is the data register of the second board timer/counter. This counter is used to generate regular interrupts to indicate that D/A channels require data to be written to them. The input to the timing counter is the prescaler output.

Before reading or writing from this register, a control word for this counter must be written to the timer/counter control register, CTCTRL. Then reads and writes to this register must follow the format specified in the control word.

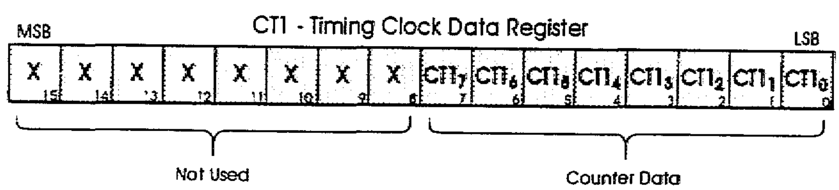


Figure 16: Timing Clock Data Register

Bits 15-8 : These bits are not used and are undefined.

Bits 7-0 : The bits $CT1_7$ (MSB) down to $CT1_0$ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

CT2 - Update Trigger Clock Data Register (offset 52, read/write)

This register is the data register of the third board timer/counter. This counter is used to generate regular update triggers to D/A channels that have been programmed for synchronous mode. The input to the trigger counter is the prescaler output.

Before reading or writing from this register, a control word for this counter must be written to the timer/counter control register, CTCTRL. Then reads and writes to this register must follow the format specified in the control word.

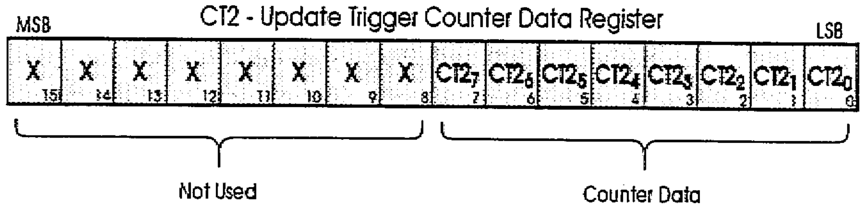


Figure 17: Update Trigger Clock Data Register

Bits 15-8 : These bits are not used and are undefined.

Bits 7-0 : The bits $CT2_7$ (MSB) down to $CT2_0$ (LSB) reflect the high byte or the low byte of the data read or written to this counter.

TCCTRL - Timer/Counter Control Register (offset 54, read/write)

The Timer/Counter Control Register is used to program, for each counter, the counting mode, the number of bytes to read/write and whether the counter counts in BCD or binary format. In addition, this register can be used to perform Read-Back Commands and Counter Latch Commands. Note that the function and bit names of this register differ according to whether Configuration Mode, Counter Latch Command or Read-Back Command is selected with bits 7-6 (SC_1 and SC_0) and bits 5-4 (RW_1 and RW_0).

See *'Using the Timing System'* in Chapter 6 for details on the read-back and counter latch commands and how the different counter modes operate.

The functions of the remaining bits are described below, depending on the setting of bits 7 and 6 and bits 5 and 4.

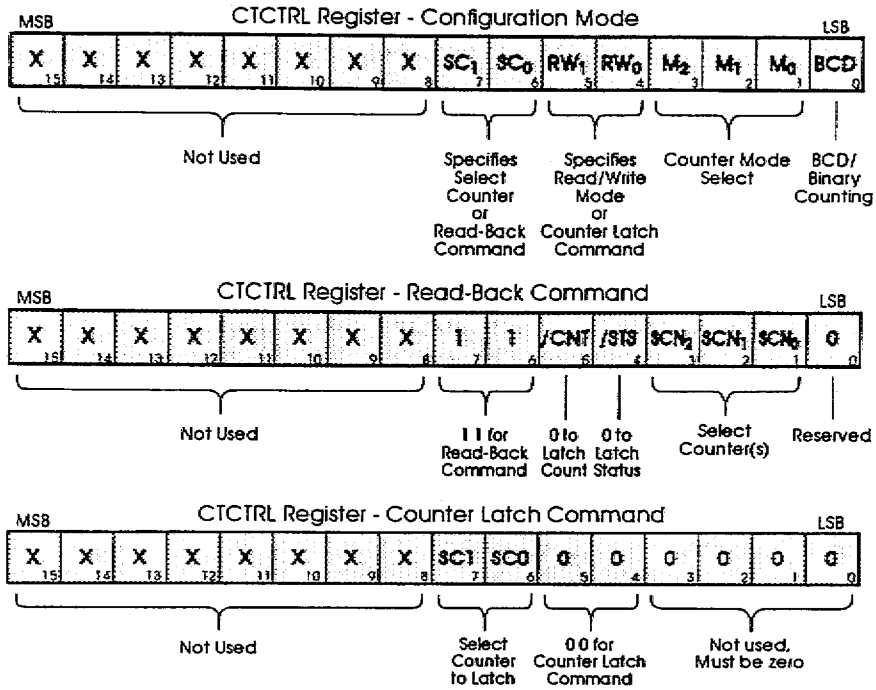


Figure 18: Timer/Counter Control Register

Configuration Mode

Bits 15-8 : These bits are not used and are undefined.

Bits 7-6 : Select Counter (SC): These two bits select the timer/counter to which the rest of the CTCTRL register bits will apply. The SC bits are defined as follows:

SC_1	SC_0	Operation Performed
0	0	Select Prescaler (Counter 0)
0	1	Select Timing Clock (Counter 1)
1	0	Select Trigger Clock (Counter 2)
1	1	Read-Back Command

Bits 5-4 : Read/Write Mode (RW): These two bits select the read/write mode of the selected timer/counter. The RW bits are defined as follows:

RW_1	RW_0	Operation Performed
0	0	Counter Latch Command
0	1	Read/Write Least Significant Byte only
1	0	Read/Write Most Significant Byte only
1	1	Read/Write Least Significant Byte first, then Most Significant Byte

Note that, even though the counters are 16-bit counters and the PC-166 is a 16-bit board, the counter interface to the PC-166 local bus is 8 bits wide. Therefore, if two byte counts are selected with RW1 and RW0 both set to 1, the counter must still be read or written to with two separate bytes (LSB and then MSB) forming the lower halves of two 16-bit words read or written to or from the counter's data register. The upper halves of the 16-bit words are not used.

Bits 3-1 : Counter Mode Select (M): These three bits select the operating mode of the timer/counter selected with bits 7-6. The mode bits are defined as follows:

M_2	M_1	M_0	Operating Mode
0	0	0	Mode 0
0	0	1	Mode 1
0	1	0	Mode 2
0	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Bit 0 : Counting Mode Select (BCD): This bit determines whether the selected counter is to count in BCD or binary format. A 0 specifies 16-bit binary counting and a 1 specifies 4 decade Binary Coded Decimal counting.

Read-Back Command

When the read-back command is specified, the bit definitions of the CTCTRL register are:

Bits 15-8 : These bits are not used and are undefined.

Bits 7-6 : These bits must both be set to 1 to invoke the read-back command.

Bit 5 : Count (/CNT): Setting this bit to 0 causes the timer/counter chip to latch the count(s) of the counters selected with bits 3-1 of this register (see below).

Bit 4 : Status (/STS): Setting this bit to 0 causes the timer/counter chip to latch certain status information from the counters selected with bits 3-1 of this register (see below). The format of the status byte is discussed on page 50 in Chapter 6.

Bits 3-1 : Select Counter (SCN): Setting one or more of these bits causes the corresponding counter to latch its count and/or status information, when the read-back command is issued. Setting SCN_2 latches the prescaler (counter 2) information, SCN_1: the timing clock (counter 1) and SCN_0: the trigger clock (counter 0).

Bit 0 : This bit performs no function and should be set to 0.

Counter Latch Command

The bit definitions of the CTCTRL Register, when the counter latch command is specified, are as follows:

Bits 15-8 : These bits are not used and are undefined.

Bits 7-6 : Select Counter (SC): These two bits select the timer/counter whose count is to be latched.

Bits 5-4 : These two bits must both be set to 0 to specify the counter latch command.

Bits 3-0 : These four bits perform no function and must all be set to 0.

This above is a summary of the 8254 timer/counter registers. See '*Using the Timing System*' in Chapter 6 for definitions on the read-back and counter latch commands and more information on programming the 8254, especially in relation to the additional PC-166 family hardware.

6

Programming Guide

6.1. Introduction

This chapter describes programming PC-166 family boards at the lowest level. While this is fairly straightforward, it requires detailed knowledge of the system hardware. As an alternative, the driver software supplied with the board can be used. Then the user need not read this section and can refer instead to the *Data Acquisition Driver Software Manual*, which describes the driver software which accompanies PC-166x boards. Advantages of using the supplied driver are:

- The time taken to develop most applications can be much reduced.
- Detailed knowledge is not required of the board, the host computer or the interface software.
- The Driver Library is callable from most high level libraries.
- The Driver Library takes into account multiple boards (both PC-166x and other supported boards) in the same computer.

Users and developers who need to incorporate special routines (especially interrupt handlers) into their application will need to read this chapter.

6.2. Register Programming

Once the PC-166x has been installed in the computer and the external connections made, the board is in an operational state. The PC-166x occupies 64 consecutive I/O addresses (in 32 16-bit wide registers), starting at the board's base address as set by the base address DIP switch. Programming a PC-166x is done solely with port input word and output word instructions, as in transferring analog output data. Reading and writing to its addresses allows the board to be configured and data transferred to and from the board. Refer to Chapter 5 for the I/O register map of the PC-166 family boards.

Reading or writing to the registers (or ports) typically takes the form of the one following instructions:

Language	Port Read	Port Write
Microsoft 'C'	`data = inpw( addr );`	`outpw( addr, data );`
Borland 'C'	`data = inport( addr );`	`outport( addr, data );`

Assembly Language	`mov ax, data`
	`mov dx, addr`
	`in ax, dx`
		`mov ax, data`
		`mov dx, addr`
		`out dx, al`

Where: `addr` is the address is the address in the address map of the PC-166x and `data` is the word read or written.

Port data size

The PC-166x registers are a word in width (16-bit), and hence must be read and written from accordingly. The registers start on ***even*** I/O addresses; i.e. the low byte of the data word is at the even or low address and the high byte thus at the odd or high address.

For example, if the base address is set to 280h, the first register will be at address 280h (low byte of the word) and 281h (high byte of the word). Therefore the address to which one writes is 280h. The next register will be at address 282h and 283h and so on.

The *entire* word must be transferred in **one** instruction; it will not do to for example, to write the low byte of the data word to the lower I/O address and the high byte to the next address; the board simply will not respond. It will also not do to write the low byte of the word to the lower I/O address and the high byte to the same address. Similarly, care must be taken when using certain compilers to ensure that they do not convert an input word or output word instruction into two separate byte instructions.

On power up or reset, the board's registers are generally reset to contain zeros. Before any meaningful analog output or waveform generation can be done, each analog channel must be set up with the correct operating parameters for the type of output required. If timed output is desired, then the board's timing system must also be set up. The next sections describe in full detail how all this is done.

6.3. Analog Output Groups

In the maximum configuration, the PC-167 board consists of four analog output groups. Each group contains a 16-bit D/A converter channel, a quad of four 12-bit D/A converter channels, some configuration jumpers and its input and output signals, as shown in Figure 19 below.

PC-166 and PC-166B boards do not have programmable references. PC-266 boards do not have 12-bit output channels. 'A' and 'B' variant boards are fitted with one programmable reference. 'B' variant boards are also only fitted with 8 12-bit output channels.

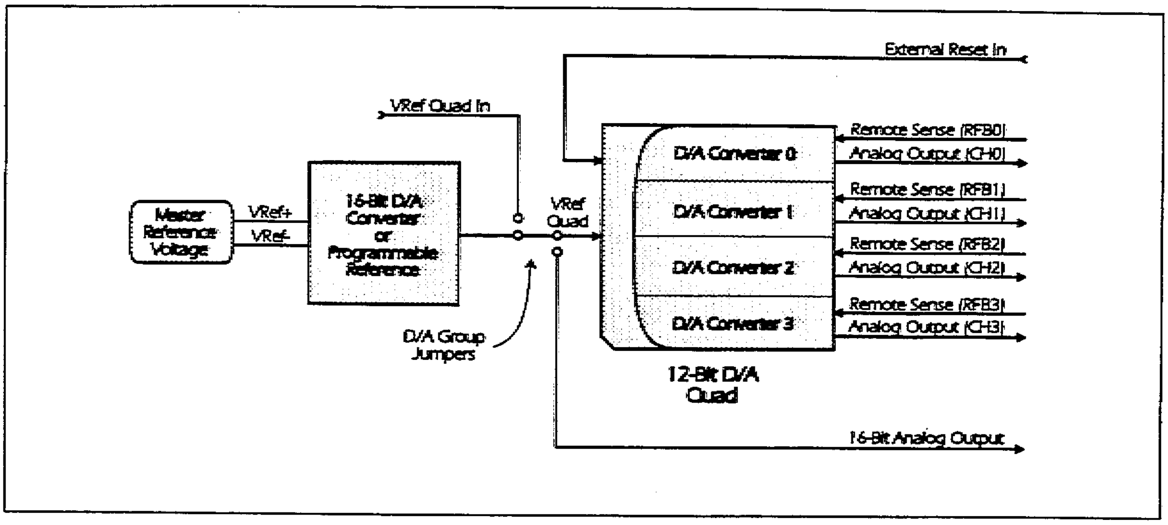


Figure 19: An Analog Output Group

6.4. Managing the 12-bit D/A Converters

Configuring the 12-bit D/A Channels

Before any voltages can be produced by a D/A converter channel, its update mode, reference gain and output range must be set. On power up or reset, all registers are reset to zero. Under this condition, the D/A channels are reset to immediate update, unipolar 10V span with a 0V output.

Output Mode and Reference Gain

Two choices of output mode are available per channel, **monopolar output** and **bipolar output**. A reference gain of 1 or 2 may also be specified for each channel. A D/A channel's output mode and reference gain is set by informing the board that a quad's mode is to be set and then writing to the corresponding D/A quad's mode register. The board is informed of a mode set instruction by setting bit 4 in the board's control register, CTRL (described on page 33). The next write operation to a quad's mode and analog output data register is then interpreted as a mode set instruction. The mode of all D/A channels in a quad is set with one mode word. The format of the word to write to the quad's mode register is repeated below.

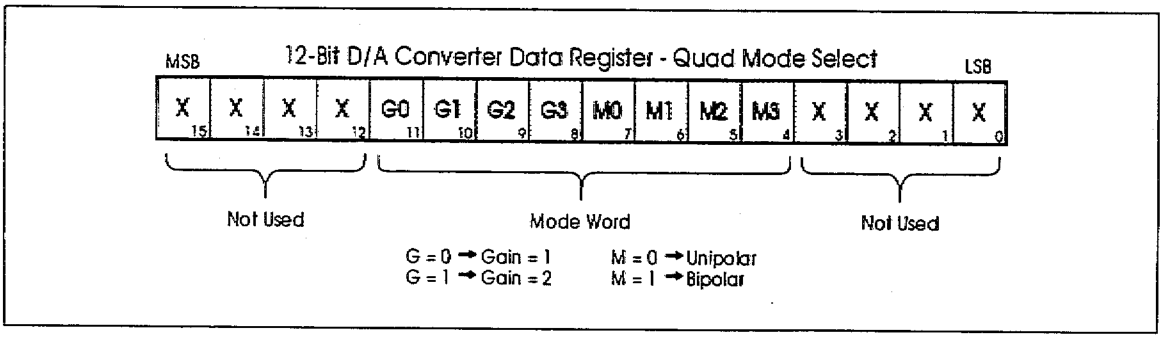


Figure 20: Quad Mode Register

M0 sets the quad's first channel's output mode and bit M3 sets the fourth channel's mode. A mode setting of 0 (M*x*=0) configures channel *x* for monopolar output, while a mode setting of 1 selects bipolar output.

Bit G0 sets the first channel in the quad's gain while G3 sets the fourth channel. A gain setting of 0 (G*x*=0) configures channel *x* to have a reference gain of 1. This means that the channel's output **span** will be equal to the quad's input reference voltage. A gain setting of 1 configures the channel to have a gain of 2, so that its output span is twice that of the reference voltage input.

If the quad's reference is +5V, a gain of 1 (Gx=0) in monopolar mode will produce an output range from 0V to +5V, while a gain of two will yield a 0V to +10V output range. In bipolar mode, (with the same reference input) a gain of 1 will give a -2.5V to +2.5V range, whilst a gain of 2 (Gx=1) will produce a -5V to +5V range. Thus for a channel to have a +/-10V (bipolar) range, the reference input must be +10V and the reference gain set to 2.

Quad Mode Readback

Mode words may be written at any time to the quads by the program in hand. If necessary, the program may check the mode settings by reading them back. This is done in the same manner as writing to the quad mode register (by setting the mode set bit in the control register), only the register is read from instead of written to.

Note that the program must reset the mode set bit once it has finished reading or writing mode words to the D/A quad.

Update Mode

It is also possible to set the update mode of each 12-bit channel. Two choices are available, **immediate update** and **synchronous** (or parallel) **update**. The update mode is set by writing a configuration word to the board's update register (UPDMODE). This is a 16-bit register (there is more information on this register on page 32) in which the setting of each bit corresponds to the update mode setting of the 12-bit output channels. Bit 0 specifies channel 0's update mode setting, bit 1 channel 1 and so on up to bit 15 for channel 15. If a channel's update mode bit is 0, immediate mode is selected whereas a 1 selects synchronous mode.

On reset or power up, all the bits in the update mode register are reset to zeros. The bits in this register must be written together, and since there is no read-back available for this register, the program in hand must maintain a copy of the current contents of the register, so that if it is necessary to change the update mode of one channel without affecting the mode of the other channels, the program, when it writes to this register, writes the updated information together with the current settings of the other channels.

In immediate mode, an analog channel's output voltage is updated as soon as the digital data is written to its data register. In synchronous mode, writing data to a channel simply stores the data in the channel's input buffer. The output voltage is not changed until the board issues an update trigger. This causes all the channels programmed in synchronous mode to transfer their preloaded data in their input buffers to their output latches, hence all the channel's output voltages change at the same time. An update trigger can come from software, the on-board update clock or an external source. See *Using the Timing System* on page 46 later in this chapter for full details on triggering.

Reset Input

Each of the D/A quads has an asynchronous reset input line. These can be used by the external system to reset all the outputs of the quad to 0V, irrespective of the state of the PC-166x board. Then that quad's output mode is set to monopolar with a gain of 1. The quad's reset lines are passively pulled high to +5V on the board; to reset a quad the external system must pull the corresponding line low for a minimum period of 90ns. Retaining the line low keeps the quad in the reset state until the line is allowed to go high again.

Transferring Data to the 12-bit D/A Channels

Once an output channel has been configured, digital data may be written to it to cause it to generate an output voltage. Each channel has an input buffer register and an output latch. Writing data to an analog output causes the data to be loaded into its input buffer. The channel's buffers are 16-bits wide. The lower 12 bits of the data word are the binary code corresponding to the output voltage, while the upper 4 bits are not used.

Monopolar Output Range

If an output channel is programmed for monopolar mode, the output voltage is given by:

Vout = (*Input Code* / 4096) x *Vref* x *Gain*

Where: *Vout* is the output voltage in volts,
Input Code is the actual right justified digital word (in binary) written to the channel's input buffer,
Vref is the quad's reference voltage input in volts (from -10V to +10V) and
Gain is the channel's programmed reference gain, either 1 or 2.

This is not very useful, we know the desired output voltage and need to know what digital code to write to produce this voltage. For a desired output voltage, the digital code is:

Input Code = 4096 x *Vout* / (*Vref* x *Gain*)

if *Vref* is not 0V and if *Vref* = 0V, it does not matter what the input code is, the output voltage will always be 0V.

Bipolar Output Range

If an output channel is programmed for bipolar mode, the output voltage is given by:

Vout = [(*Input Code* - 2048) / 4096] x *Vref* x *Gain*

Again, we need the digital code corresponding to this voltage. It is:

Input Code = 4096 x *Vout* / (*Vref* x *Gain*) - 2048

if *Vref* is not 0V and if *Vref* = 0V, it does not matter what the input code is, the output voltage will always be 0V.

Data Read-Back

The operating program may at any time read back the contents of any of the output channel's data registers. The digital code is returned in exactly the same format in which it was written.

References Values

PC-166 and PC-166B boards usually have fixed 10V references for the D/A quads, hence the value of *Vref* above is always 10V. Note that according to the equation for monopolar outputs, with a 10V reference and a gain of 2, an input code of 4095 would produce an output voltage of +20V - 1LSB. This is clearly impossible since the D/A's maximum output voltage is +10V - 1LSB; with this reference, output mode and gain setting, any input codes greater than 2048 will just saturate the output. This is not advisable.

If necessary, PC-166 and PC-166B boards can be jumpered so that the D/A converters have a +5V reference voltage. This is done with the D/A master reference jumper, JP1, described in Chapter 2: *Installation*.

PC-167 boards are fitted with a programmable reference for each of the D/A quads. Hence *Vref* in the equations above is the output of the programmable reference. Note that it is permissible for *Vref* to have a negative value. This allows full four quadrant multiplication by varying *Vref* and a 12-bit output channel (in bipolar mode) in that quad.

PC-167A and PC-167B boards are fitted with one programmable reference for all 12-bit output channels.

6.5. Using the 16-bit D/A Converters or Programmable References

PC-266 and PC-167 type boards are fitted with 16-bit D/A converters, which are used as higher accuracy outputs on the PC-266 and as programmable references for the 12-bit channels on the PC-167 type boards. PC-266 and PC-167 boards are fitted with four 16-bit D/A converters, and these are referred to as output channels 16 to 19. PC-167A and PC-167B boards are fitted with one 16-bit D/A converter, and is channel 16.

The 16-bit output channels are permanently configured for bipolar outputs with a 20V span (i.e. from -10V to +10V). These channels do not have to be configured in any way and are ready for data transfer immediately.

Each channel has an input buffer register and an output latch. Writing data to a 16-bit output causes the data to be loaded into its input buffer, and then immediately into its output latch. The channel's buffers are 16-bits wide. All 16 bits of the data word are used to hold the binary code corresponding to the output voltage.

The output voltage corresponding to a digital input code is:

$$Vout = 10 \times [\,(Input\ Code - 32768)\,/\,32768\,]$$

We need the digital code corresponding to this voltage. It is therefore:

$$Input\ Code = 3278.6 \times Vout - 32768$$

Data Read-Back

The operating program may at any time read back the contents of any of the 16-bit channel's data registers. The digital code is returned in exactly the same format in which it was written.

6.6. Using the Timing System

This section describes how to use the board's timing system to control the way analog values are generated. The system consists of a reference oscillator, three counters, two gate selection multiplexers and a trigger selection multiplexer all under control of the board control and software trigger registers, as depicted below.

The three counters on PC-166x boards are provided by an 8254 timer/counter integrated circuit. This chapter provides detailed programming information on the 8254 and associated circuitry for the PC-166x.

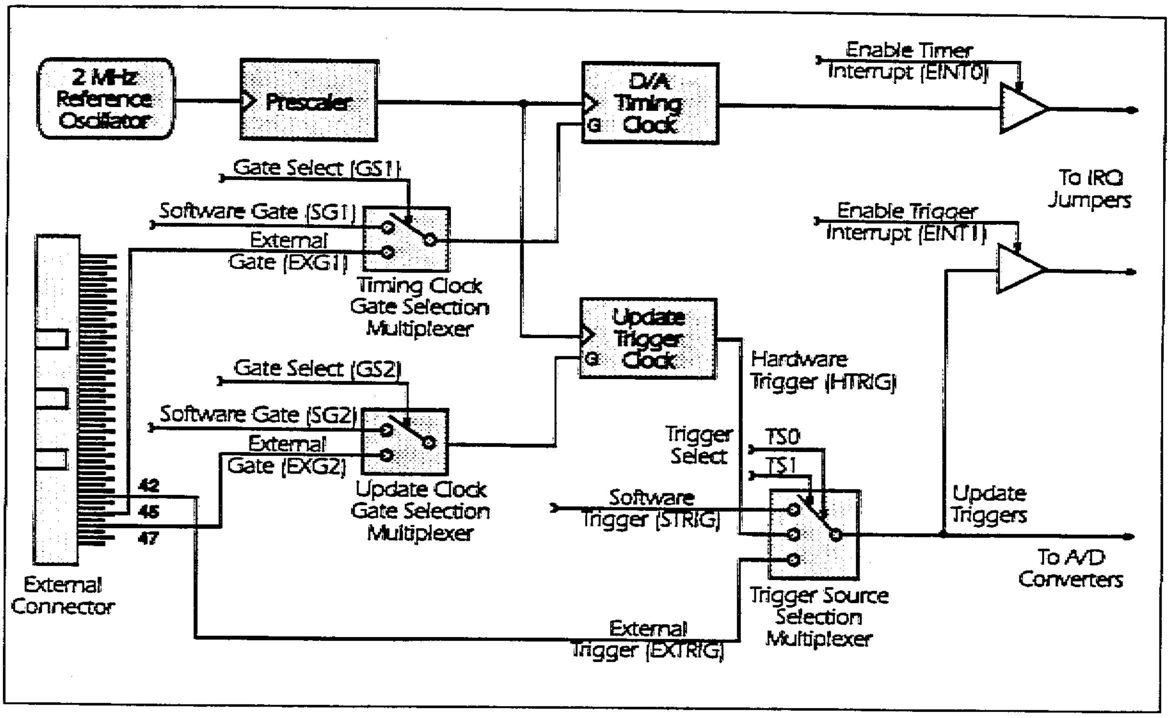


Figure 21: PC-166/167 Timing System.

Introduction

The 8254 is a general purpose three channel timer/counter device. Each timer/counter is totally independent, and each may be programmed in different modes and data formats. Since all three timers are identical, the information provided here applies equally to all the timers.

The PC-166x uses the first timer as a prescaler for the other two timers. Its clock input is driven by a fixed 2MHz reference oscillator. The next timer, timer 1, is used as an interrupt timer to indicate that the board needs service from the host program. The last timer, timer 2, is used to generate update triggers for output channels programmed in synchronous update mode. Both the timing clock and the update clock have their clock input driven by the output of the prescaler.

The timer/counter chip itself consists of a control word register, some logic circuitry and the three counters. Additional hardware on the PC-166x provides a programmable gate source for the timing and update clocks, a software or externally controlled counting gate and software controlled interrupt enable/disable.

Each counter consists of a two byte wide Count Register, a 16-bit Counting Element and a two byte wide Output Latch.

Count Register

The count register of a counter/timer stores the initial 16-bit count written to a counter. It consists of two, byte wide registers, which are written to separately even though the PC-166x has a 16-bit bus. When a counter is programmed with a control word, the count register is cleared. Both count register bytes are transferred (loaded) to the counting element simultaneously.

Counting Element

The counting element is simply a 16-bit presettable synchronous down counter. It cannot be read or written to directly. It is automatically loaded on specified conditions from data in the count register. The count value is always read from the output latch.

Output Latch

The output latch normally follows the counting element. It consists of two byte wide registers, which are read from separately. If a suitable counter latch command (see below) is sent to the counter, the current count value is latched in the output latch until it is read from the counter's data register (CT2, CT1 or CT0). Thereafter, the output latch continues to follow the counting element.

Gate source

The gate inputs of the timing counter (timer 1) and the update counter (timer 2) can be controlled independently by software to come from either software or an external source. The board control register contains two gate source bits (GS1 and GS2), corresponding to timer 1 and timer 2. Setting a gate source bit allows the timer's gate to be controlled via an external input (EXG1 or EXG2), while clearing the bit allows the program to control the gate via the board's control register (via SG1 or SG2). See page 33 for the format of the control register. Note that if a timer is programmed for a software controlled gate, the timer's external gate input line may be used as an additional general digital input line.

Software Gate

The board's control register also contains two software gate bits (SG1 for timer 1 and SG2 for timer 2), that directly control the corresponding timer's gate input. If the timer is programmed for a software controlled gate (via its gate source bit), then in general, setting a software gate bit enables counting and clearing the bit disables counting. See page 55 later on in this chapter for a full description of how the gate affects counting in various timer modes. If the timer is programmed for an external gate, writing to these bits has no effect.

Timing System Interrupts

The board control register also contains two interrupt enable bits; EINT0 for the timing clock output and EINT1 for the D/A input buffer empty signal. These bits control whether the corresponding signal is enabled onto a jumper selected IRQ line of the PC's I/O Channel. Setting an interrupt enable bit allows interrupt requests to reach the host computer and clearing the bit inhibits interrupt generation.

Programming a Counter

On power up or reset, the state of the 8254 is undefined. Before any timer/counter operations can be done, **each** timer to be used must be programmed with a control word, which is written to the counter/timer control register. This sets the individual counter's:

- Operating mode (Mode 0 to 5),
- Counting format (BCD or binary) and
- Read/Write format (LSB only, MSB only or LSB then MSB).

The programmed counter then operates in the specified format until it is reset or new configuration information is written to the Control Register. The format of this register when used for configuring a counter is repeated below for reference.

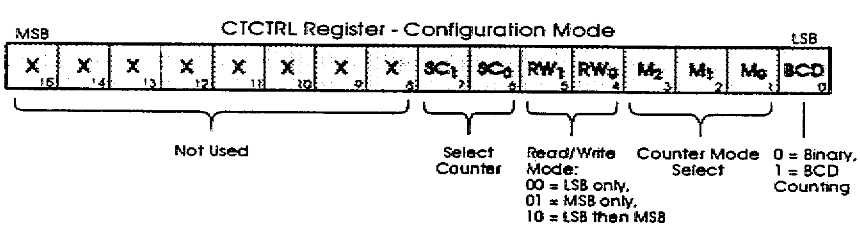


Figure 22: Timer/Counter Control Register - Timer Configuration Mode

See page 37 in Chapter 5 for the full bit definitions of this register.

Data Transfer Format

Using the Control Word, each counter may be programmed to transfer data in one of three ways:

- Read/write least significant byte only,
- Read/write most significant byte only or
- Read/write least significant byte first, then most significant byte.

A new initial count may be written to a counter without affecting the counter's programming in any way. Counting will be affected as described in the mode definitions below.

Writing a one byte initial count simply consists of outputting the byte to the counter's data register. Writing a two byte count consists of writing the first byte (the least significant byte) to the counter's data register and then, at any time later, writing the second or most significant byte to the same data register. This must be done as two 16-bit output operations, with the upper 8 bits all zero.

Hardware Connections

Each counter has three physical electrical connections which are:

- a clock pulse input,
- a gate input and
- a timer output.

Clock Pulse Input

The clock pulse input is the physical connection, where clock pulses are applied to a counter. A clock pulse is defined as a rising edge, then a falling edge, in that order, at a counter's clock input. New counts are loaded, and the counting element is decremented, on the falling edge of a clock pulse.

The source of clock pulses for timer 0 (the prescaler) is the on-board 2MHz oscillator, for timer 1 and 2 it is the output of the prescaler.

Gate Input

Depending on the counter's mode, the gate input provides for: enable/disable counting; count initiating (trigger) or setting/resetting the timer output. The prescaler's gate is permanently held high, while the gates of timer 1 and timer 2 are controlled from either an external source or the program in hand. The functioning of the gate input is described in the individual mode definitions which follow this, and summarised in Table 4 on page 55 below the mode definitions.

Read Operations

It is often necessary to read the value or status of a counter without disturbing the count in progress. This is easily done using one of three methods: a simple read operation; a counter latch command or a read-back command. The results of the read operation are read from the counter's data register, in the programmed format (LSB, MSB or LSB then MSB) of that counter. This will be termed **reading a counter**, and may take *one* physical CPU read instruction (LSB or MSB) or *two* physical CPU read instructions (LSB then MSB).

Simple Read Operation

The simple read operation consists of reading the contents of the desired counter's data register. The clock input of the counter must be disabled for this to be successful, otherwise the count may be in the process of changing when it is read, returning a completely erroneous count value. The clock input may be disabled with the gate input, the software gate (described above) or with external logic. Thus this method may not be used successfully with timer 0, which has its clock input permanently enabled.

Counter Latch Command

A counter latch command is invoked by writing a special type of control word to the timer/counter control register, CTCTRL.

The format of the control word for a counter latch command is diagrammed below.

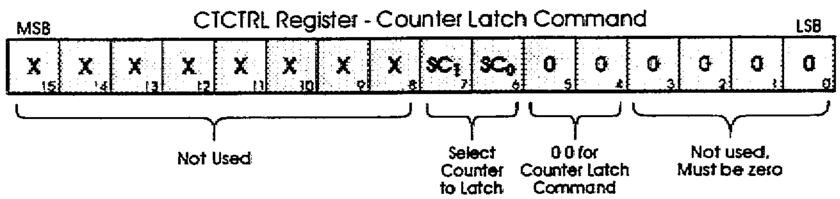


Figure 23: Timer/Counter Control Register - Counter Latch Command

Bits SC_1 and SC_0 *select* the counter whose count is to be latched. 00 selects counter 0, 01 selects counter 1 and 10 selects counter 2

The selected counter's output latch latches the count at the time the counter latch command is received. The count is held in the latch until it is read (or the counter reprogrammed). The count is then automatically unlatched and the output latch returns to following the counting element. Multiple counter latch commands may be used to latch more than one counter. Each latched counter holds its count until read. Counter latch commands do not affect the programming of the counter in any way.

If a counter is latched, any subsequent counter latch commands to the same counter, before the count has been read, will be ignored. When read, the count value returned will be the count at the time the first counter latch command was issued.

Read-Back Command

A read-back command is issued by writing a special type of control word to the timer/counter control register, CTCTRL. Read-back commands may be used to latch one or more counter's *current count value* and/or *status* information.

The format of the control word for a read-back command is diagrammed below.

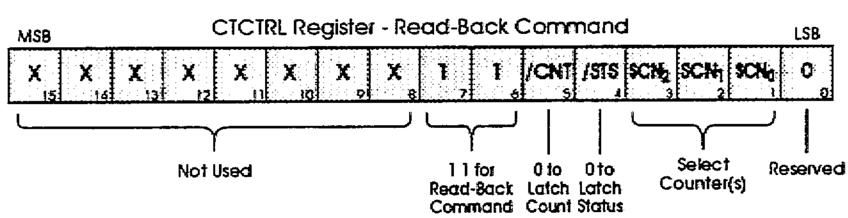


Figure 24: Timer/Counter Control Register - Read-Back Command

Setting any or all of bits 3 to 1 (SCN_2 to SCN_0) *selects* the counter(s) to which this command will apply. Setting the count (/CNT) bit to 0 causes the current count of the selected counter(s) to be latched, setting the status (/STS) bit to 0 causes the status byte of the selected counter(s) to be latched and setting both bits to 0 causes both the status and current count to be latched (see below).

Multiple Counter Latch

The read-back command is used to latch the current count of multiple counters in their respective output latches. This is done by setting the count bit to 0 in the control word for this command. This single command is functionally equivalent to multiple counter latch commands. Each counter's latched count is held until read (or the counter reprogrammed). That counter is automatically unlatched when read, but the other counters remain latched until they are read. If multiple read-back commands are issued to the same counter without reading the count, all but the first are ignored. The count returned is the current count at the time the first read-back command was issued.

Counter Status Information

The read-back command may also be used to latch certain status information of the selected counter(s), in their respective output latches. This is done by setting the status bit to 0 in the control word for this command.

The counter's status byte, when read, provides the information about the counter shown in the diagram below.

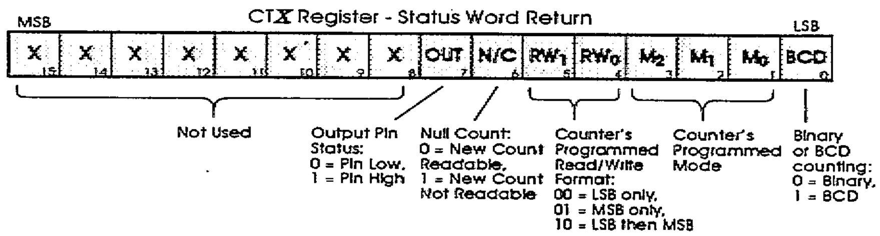


Figure 25: Counter/Timer Data Register - Status Word Return

Bits M_2 to M_0 return a binary number corresponding to the counter's programmed mode.

The null count bit indicates if the last count written to the count register has been loaded into the counting element. If the count has not yet been loaded then it cannot be read and the null count bit will be set to 1. Reading the count when the null count bit is set will return the current count from the previous initial count written to the count register. The exact time the new count is loaded depends on the mode of the counter (see below) but the null count bit clear indicates the new count has been loaded.

Bit 7 (OUT) reflects the state of the counter's output pin. This provides software the power to monitor this pin.

Latching both Status and Current Count

Setting both the Status (/STS) bit and the Count (/CNT) bit to 0 causes both the status byte and the current count of the selected counter(s) to be latched simultaneously. This is functionally the same as issuing two separate read-back commands at once. If both the status and current count are latched, the first read operation of the counter's data register will return that counter's status byte, regardless of which was latched first. The next one or two reads (depending on whether the counter has been programmed for one or two byte counts) return the latched count. Subsequent reads return unlatched counts.

Counter Mode Definitions

The following six sections describe in detail the different counting modes of the 8254 timer/counter. In the descriptions which follow, the output refers to the state of the output pin of the device, and gate refers to the state of the counter's gate input pin.

The word *Trigger* is used to mean a rising edge at a counter's gate input, *Counter Loading* the transfer of the initial count from the counter's count register to its counting element and *Clock Pulse* a rising edge and then a falling edge at a counter's clock input.

Mode 0: Interrupt on Terminal Count

After the mode byte is written to the control register, the output is low. Once an initial count has been written, the output remains low until the counter has counted down to zero. The output then goes high and remains high until a new count is written or the counter reprogrammed.

The gate input inhibits counting when low, and enables counting when high.

After the control word and initial count have been written, the counter is loaded on the next clock pulse. This clock pulse does not decrement the count, so for an initial count of N, the output goes high $N+1$ clock pulses after the initial count was written.

If a new count is written to the counter, it will be loaded on the next clock pulse and counting will continue from the new count. If a two-byte count is written, the first byte disables counting and sets the output low. After the second byte is written, the full count is loaded on the next clock pulse. This allows the counting sequence to be synchronised by software. Again the output goes high after $N+1$ clock pulses.

If the initial count is written when the gate is low, it will still be loaded on the next clock pulse. When the gate goes high, the output will go high N clock pulses later.

Using the internal oscillator, this mode can be used to generate a positive edge on the external output after a programmable time, or if the board is jumpered for interrupts, to generate an interrupt after a programmable time.

This mode can also be used to count events or frequency.

Mode 1: Hardware Retriggerable One-Shot

After the counter is programmed the output will be high. Writing an initial count arms the counter and a subsequent trigger loads the counter. The output goes low on the next clock pulse and remains low until the counter reaches zero. The output then goes high and remains high until the next clock pulse after the next trigger.

An initial count of N results in a one-shot pulse N clock cycles long. The one-shot is retriggerable, hence the output will remain low for N clock pulses after any trigger. The one-shot pulse can be repeated without rewriting the initial count to the counter. The gate input has no effect on the output.

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

Mode 2: Rate Generator

After the counter is programmed the output will be high. An initial count of *N* is loaded on the next clock pulse and when it has decremented down to 1, the output goes low for *one* clock pulse. The output then goes high, the counter automatically reloads the initial count and the process is repeated indefinitely. The sequence is repeated every *N* clock pulses.

The gate input enables counting when high, and inhibits counting when low. If the gate goes low during an output pulse, the output is set high immediately. A trigger reloads the initial count on the next clock pulse and the output goes low for one clock pulse after *N* clock pulses. Thus the gate input can be used to synchronise the counter.

Writing a new count does not affect the current counting sequence. If a trigger is subsequently received before the end of the current period, the counter will be reloaded on the next clock pulse and counting will continue from the new count. Otherwise the new count will be loaded at the end of the current cycle.

In this mode an initial count of 1 is invalid.

Mode 2 functions like a divide by *N* counter and is a good default mode for the functionality of the timers on the PC-166x. It can also be used to generate an output frequency or a periodic interrupt.

Mode 3: Square Wave Generator

Mode 3 is similar to Mode 2 except for the duty cycle of the output. After the counter is programmed the output will be high. An initial count of *N* is loaded on the next clock pulse. When *half* of the initial count has expired the output goes low for the remainder of the count. The output then goes high, the counter automatically reloads the initial count and the process is repeated indefinitely. This results in a square wave with a period of *N* clock cycles.

The gate input enables counting when high, and inhibits counting when low. If the gate goes low when the output is low, the output is set high immediately. A trigger reloads the counter with the initial count on the next clock pulse. Thus the gate input can be used to synchronise the counter.

Writing a new count does not affect the current counting sequence. If a trigger is subsequently received before the end of the current half-cycle of the square wave, the counter will be reloaded on the next clock pulse and counting will continue from the new count. Otherwise the new count will be loaded at the end of the current half-cycle.

Mode 3 functions slightly differently for even and odd initial count values.

For even counts: the output is initially high. On the next clock pulse the initial count is loaded. On subsequent clock pulses it is decremented by *two*. When the count expires, the output toggles and the counter is reloaded with the initial count. This process is repeated indefinitely.

For odd counts: the output is initially high. On the next clock pulse the initial count *minus one* (an even number) is loaded. On subsequent clock pulses it is decremented by *two*. One clock pulse *after* the count expires, the output goes low and the counter is reloaded with the initial count minus one. Subsequent clock pulses continue to decrement the count by two. When the count expires, the output goes high again and the counter is reloaded with the initial count minus one. This process is repeated indefinitely. So for odd counts, the output is high for $(N+1)/2$ counts and low for $(N-1)/2$ counts or high for one count longer than it is low.

Mode 3 is typically used to generate an output frequency.

Mode 4: Software Triggered Strobe.

After the mode byte is written to the control register, the output is high. Once an initial count has been written, the output remains high until the counter has counted down to zero. The output then goes low for *one* clock pulse and then goes high again. The counting sequence is triggered by writing an initial count.

The gate input inhibits counting when low, and enables counting when high. It has no effect on the output.

After the control word and initial count have been written, the counter is loaded on the next clock pulse. This clock pulse does not decrement the count, so for an initial count of N, the output strobes low $N+1$ clock pulses after the initial count was written.

If a new count is written while counting, it will be loaded on the next clock pulse and counting will continue from the new count. If a two-byte count is written, the first byte written has no effect on counting. After the second byte is written, the full count is loaded on the next clock pulse. This allows the counting sequence to be retriggered by software. Again the output strobes low after $N+1$ clock pulses.

Using the internal oscillator, this mode can be used to generate a negative pulse on the external output after a programmable time.

Mode 5: Hardware Triggered Strobe

This mode is similar to mode 4 except that the counting is triggered by a rising edge on the counter's gate input.

After the control word and initial count have been written, the output is high. The counter is loaded on the next clock pulse after a trigger is received. This clock pulse does not decrement the count. The output remains high until the counter has counted down to zero. The output goes low for *one* clock pulse and then goes high again. So for an initial count of N, the output strobes low $N+1$ clock pulses after the initial count was written.

The counting sequence is retriggerable: a trigger causes the counter to be loaded with the initial count on the next clock pulse. The output will not strobe low until $N+1$ clock pulses after any trigger.

The gate input has no effect on the output.

If a new count is written while counting, it will have no effect on the current count sequence. If a trigger is received after the new count is written but before the current count expires, the counter will be reloaded on the next clock pulse and counting will continue from there.

Using the internal oscillator, this mode can be used to generate a negative pulse on the external output after a programmable time from an external trigger.

Counter Gate Input

The gate inputs of the counter/timers have various functions depending on the mode and state of the gate input signal. The functions are summarised in the table below.

Operating Mode	Low or Going Low	Rising	High
0	Disables Counting	---	Enables Counting
1	---	Initiates Counting & Resets Output after Next Clock	---
2	Disables Counting & Sets Output Immediately High	Initiates Counting	Enables Counting
3	Disables Counting & Sets Output Immediately High	Initiates Counting	Enables Counting
4	Disables Counting	---	Enables Counting
5	---	Initiates Counting	---

Table 4: Summary of the Functions of the Counter Gate Inputs.

D/A Converter Update Triggering

Each of the 12-bit D/A converters has an input buffer and an output latch. The voltage it generates in a function of the contents of its output latch only, irrespective of the contents of the input latch. Now, the act of transferring the contents of the input buffer to the output latch is called updating the D/A converter, and is done with an update trigger. There are four types of update trigger; the first occurring when the output channel is programmed for immediate update:

- Immediate update trigger, in which the output latch is loaded with the contents of the input buffer as soon as the input buffer is written to by the operating program,

And then three types update triggering when the output channel is programmed for parallel or synchronous update:

- **Software update,** where a CPU write to the software trigger register (STRIG) causes all the D/A converters to be updated,
- **Hardware** or **clocked update,** where strobes from the update clock supply the update triggers and
- **External update,** where an externally connected signal updates the D/A converters.

The source of synchronous update triggers is selectable using the trigger select bits (TS0 and TS1) in the board's control register, as shown below.

TS_1	TS_0	Update Trigger Source
0	0	Software trigger via STRIG register
0	1	Hardware trigger from trigger clock
1	0	External hardware trigger
1	1	Reserved, do not use

The general principle behind delayed updates is that the controlling software writes data to the desired output channels in good time before an update trigger is received by the output channels. The software is able to determine when new data can be written to the output channels by monitoring the D/A converter buffer empty bit in the control register; if the bit is set it means that a trigger has occurred and new data can be written. This D/A converter buffer empty signal can also be used to interrupt the CPU so that the processor can write new data, in the background, as soon as the board is ready for it, without interfering with the main program's execution. More details follow in the next two sections, while the status bits are described fully on page 33 in Chapter 5: '*Register Structure*'.

Interrupt Handling

Both the output of the timing clock and the D/A converter buffer empty signal may be used to generate interrupt requests to the host computer. These two sources of interrupts are enabled respectively with the EINT0 and EINT1 bits in the board control register.

The timing clock may be used to generate a regular interrupt to the host system, which may invoke an interrupt service routine to write the next set of data points.

The D/A converter buffer empty signal indicates to an interrupt service routine that the next set of data points may also be written.

Whatever the source of interrupts, the software programmer's approach is the same:

- Select the desired interrupt line to use for the desired PC-166x interrupt source with the interrupt jumper block (JP3) on the PC-166x board.

The following procedures are all done from software:

- Program the counter and perform any other setup that has to be done.
- Save the state (enabled or disabled) of the selected interrupt level on the PC system board and then disable it.
- Save the old interrupt vector and install a new one, that will point to the interrupt service routine that will service the board's interrupts.
- Enable the interrupt level in the PC's interrupt controller.
- Enable the PC-166x interrupt onto the PC I/O channel by setting the appropriate bit in the board control register.

Now the software can continue with other tasks. The interrupt service routine must:

- Do whatever processing it needs to do, for example, writing the next set of data points or generating an update trigger.
- Either chain another interrupt routine if the interrupt line is being shared or write an end of interrupt command to the PC system board interrupt controller.

To clean up the interrupts after using them:

- Disable the interrupt in the PC's interrupt controller.
- Disable interrupt with the board's control register.
- Restore the interrupt vector to what it was before.
- Restore the interrupt level to its previous state.

Note that with respect to timer/counters, some modes are more suitable for generating interrupts (eg Mode 0 and 2) than others.

D/A Output Examples

What follows are a few sample methods for generating output waveforms or voltages. These are not the only methods for generating output voltages of waveforms, but they serve to illustrate some general principles.

Simple Program Output

This is the most elementary method for generating output voltages. The desired output channels are programmed for immediate update mode and whatever reference gain and output mode required. Then output data is simply written by the program, as required, to each channel's data register. The D/A converter loads the data automatically into its output latch and generates a corresponding output voltage.

Program Controlled Clocked Output

Using clocked output allows the output data points to be spaced accurately in time. The simplest way of doing this is described here.

- The program sets the board for hardware update triggers (i.e. from the update clock) with a software controlled gate and disables counting in the update clock with its gate.
- Next it programs the prescaler and update clock to generate update triggers at the desired frequency.
- The desired output channels are then programmed for synchronous update mode and whatever reference gain and output mode required.
- The first set of data points are written to the analog output channels' input buffers.
- Counting is then enabled in the update timer by enabling its gate.
- The program now monitors the D/A converter buffer empty bit in the board control register. As soon as the update timer generates an update trigger, all the D/A outputs are updated and the D/A converter buffer empty bit is set in the control register.
- The program detects this and writes the next set of data points. This causes the D/A buffer empty bit to be reset, and the monitoring and writing process continues.

If an update trigger is received by the D/A converters before the next set of data points has been written, the trigger error bit will be set in the board control register.

A variation on this is to supply the update triggers from an external source. The external source may also control the update timer's gate.

Interrupt Controlled Output

This method allows the output data to be written to the D/A converter channels totally in the background without interfering with the main program's execution. There are two separate components to the program in this case; the main program which sets up the board and flow of execution and the interrupt service routine. The main program must:

- Set up a data structure of the output data points that the interrupt service routine can access and write directly to the output channels. Raw binary data is preferable so that the ISR does not have the overhead of converting it.
- Set the board for software update triggers (i.e. from the software trigger register) with a software controlled gate and disable counting in the timing clock with its gate.
- Set the desired output channels to have synchronous update mode and whatever reference gain and output mode required.
- Program the prescaler and timing clock to generate interrupts at the desired frequency.
- Save the state (enabled or disabled) of the selected interrupt level on the PC system board and then disable it.
- Disable the timing clock interrupt from the PC I/O channel by clearing the EINT0 bit in the board control register.
- Save the old interrupt vector and install a new one, that will point to the interrupt service routine that will service the timing clock interrupts.
- Enable the interrupt level in the PC's interrupt controller.
- Enable the timing clock interrupt onto the PC I/O channel by setting the EINT0 bit in the board control register.
- Enable counting in the timing clock with its gate.

This allows the timing clock to generate interrupt requests. Now the software can continue with other tasks. Whenever the host PC receives an interrupt from the timing clock, the interrupt service routine is executed. It must:

- Write the next set of data points to the output channels.
- Write a 1 to bit 0 of the software trigger register to generate an output channel update trigger. This updates the contents of all the channels written to simultaneously.
- Write an end of interrupt command to the PC system board interrupt controller and return from the interrupt.

A variation on this is to program the output channels for immediate update mode. Then each channel is updated as soon as data is written to its input buffer and the ISR does not need to generate an update trigger. Also, some channels may be set to immediate update mode and others synchronous mode as desired.

Interrupt Controlled Clocked Output

This method is similar to the previous method, interrupt controlled output, only it uses the output buffer empty signal as an interrupt to indicate that new data should be written to the analog output channels. It also allows waveform generation totally in the background without interfering with the main program's execution. Again, there are two separate components to the program; the main program which sets up the board and flow of execution and the interrupt service routine. The main program must:

- Set up a data structure of the output data points that the interrupt service routine can access and write directly to the output channels. Raw data is preferable so that the ISR does not have the overhead of converting it.
- Set the board for hardware update triggers (i.e. from the update clock) with a software controlled gate and disable counting in the update clock with its gate.
- Set the desired output channels to have synchronous update mode and whatever reference gain and output mode required.
- Program the prescaler and update clock to generate interrupts at the desired frequency.
- Save the state (enabled or disabled) of the selected interrupt level on the PC system board and then disable it.
- Disable the update clock interrupt from the PC I/O channel by clearing the EINT1 bit in the board control register.
- Save the old interrupt vector and install a new one, that will point to the interrupt service routine that will service the update clock interrupts.
- Enable the interrupt level in the PC's interrupt controller.
- Enable the update clock interrupt onto the PC I/O channel by setting the EINT1 bit in the board control register.
- Write the first set of data points to the analog output channels' input buffers.
- Enable counting in the update clock with its gate.

This allows the update clock to generate update triggers as well as interrupt requests. Now the software can continue with other tasks. It can, at any time, check the trigger error bit in the board control register. If the bit is set, it indicates that the interrupt service routine did not write new data before the subsequent update trigger was received. Whenever the host PC receives an interrupt from the update buffer empty signal, the interrupt service routine is executed. It must:

- Write the next set of data points to the output channels.
- Write an end of interrupt command to the PC system board interrupt controller and return from the interrupt.

Using this method, the update triggers are not constrained to come from the update clock, they may come from any of the three possible sources; viz software from the main program via the software trigger register, the update clock or an external source.

Conclusions

Using interrupt timing or updating, a little care must be taken to ensure accurate output timing. The interrupt latency time should be kept to a minimum (by not disabling interrupts for any length of time, not allowing higher priority interrupts to be invoked for lengthy time periods and not executing in certain multi-tasking environments). Also the priority of the interrupt should be as high as possible.

The board is also not constrained to operate using only one of these methods; more than one may be used at the same time. Care must be taken to ensure that they do not interact with each other however.

6.7. Digital Input

A single line of digital input is provided as part of the PC-166x hardware. Its primary function is intended for the external system to signal to the host computer an error condition (for example, the external system's STOP button pressed), but it may be used for any purpose.

Using the digital input is straightforward. The line is TTL compatible, generally with a high being 2.5V to 5V and a low 0V to 0.4V. The status of the input line is communicated via the board's control register with bit 5 (EXD0). Reading this register and testing bit 5 reflects the current status of the digital input line.

Additional Digital Inputs

If the either or both of the external gate inputs for the timing and update clocks are not used for their function, they may also be used as digital input lines. Simply program the timers for software controlled gates (via SG1 and SG2 in the board control register), and the external gate inputs will have no effect on their respective timer. The EXG1 and EXG2 bits in the control register reflect the current status of these two lines, in exactly the same manner as the digital input line (EXD0) described above.

7

Driver Software

Full driver software is supplied as part of the PC-166x package. It is called the Data Acquisition and Control Driver Software Library, and is provided in two equivalent forms:

- A ready to use, precompiled, driver software library, with routines callable from various high level languages. These are:
  - 'C': Microsoft and Borland.
  - Pascal: Borland Turbo Pascal.
  - BASIC: Microsoft QuickBasic.
- Complete 'C' source code for all driver routines and complete source code for the language interface modules.

The Driver Library allows programmers to control PC-166x boards via high level function calls, so allowing the user to write custom software applications without having to understand the low level operation of the board.

Also included with the driver package is complete source code, in C for the entire driver package. This allows advanced users to modify existing code, rather than having to start writing low level code from scratch.

The Data Acquisition Driver Library supports not only PC-166x boards, but a number of other boards too. This chapter presents a quick reference to the functions in the driver library that apply to PC-166 family boards. Full details on using the library functions, the source code and a full function reference are provided in the companion manual to this, the *Data Acquisition and Control Driver Software Manual.*

A Note about Using the Driver Library

If you are using the driver library to operate the board, unless you know exactly what you are doing, **always** control the board with calls to the driver library and do **not** bypass the driver library and read or write directly to or from the board. This is because the driver library keeps an internal software copy of the state of the board. If you communicate to the board directly, this will obviously not update the software copy of the board's state. Hence, subsequent calls to the driver library may not produce the desired results, or the board may behave erratically or unpredictably.

In summary, do not allow the board's state to get out of sync with the software copy of its state.

7.1. Quick Function Reference

The Data Acquisition and Control Driver, when used with PC-166 family boards has the following appropriate function calls. These are:

Initialisation and General Functions

Function Name	Brief Description
`Init_brd`	Initialises one of up to eight boards. This function initialises the hardware of the board if necessary and informs the driver system of the board's presence. It must be called before any other function calls (besides `ID_Test`) are made to the board.
`Version`	Returns the version of the software library system in use.
`ID_Test`	Identify and test the board at the given base address.
`Int_set`	Enable and disable interrupts from a source of interrupts on a board or the host computer system board.

Analog Output Functions

Function Name	Brief Description
`AOut_Config`	Configures the output range, update mode and reference gain of an analog output channel.
`AOut_VRefConfig`	Informs the driver system of an analog output channel's reference voltage.
`AOut_VRefGet`	Returns the reference voltage of the specified output channel.
`AOut_TrigSource`	Sets the source of update triggers for synchronously updated channels.
`AOut_Update`	Updates output channels set for synchronous update mode.
`AOut_VWrite`	Writes a voltage to an analog output channel.
`AOut_Write`	Writes a binary value to an analog output channel.
`AOut_Read`	Reads back a binary value previously written to an analog output channel.
`AOut_V2Bin`	Converts a voltage to a corresponding binary value for the selected output channel.
`AOut_Bin2V`	Converts a binary value to the corresponding voltage for the selected output channel.

Counter/Timer Functions

Function Name	Brief Description
`CT_cfg`	Configures the operating mode, the counting method and the read/write format of the specified timer/counter.
`CT_read`	Latches and returns the current count value of a counter.

`CT_write`	Writes an initial counting value to a counter.
`CT_status`	Returns a status byte indicating a counter's programming, the state of the its output pin and whether the last initial count has been loaded.
`CT_gate_source`	Selects the source of the gate input of a counter.
`CT_soft_gate`	Gates - inhibits or allows clock pulses to reach - the selected counter.

8

Hardware Specifications

12-Bit D/A converters

Number of channels:	16 on PC-166, PC-167 and PC-167A boards, 8 on PC-166B and PC-167B boards
Resolution:	12-bit
Settling time to $^{1}/_{2}$ LSB:	Max 10us into a load of 500pF, 2kOhm
Throughput:	800kHz[1] at 50kHz max per channel
Output voltage range:	Programmable to any value in the range -10V to +10V, monopolar or bipolar
Output current:	5mA max
Reset line pulse width:	90ns min
Short circuit current:	Limited to 40mA to 0V or any power supply to +/-15V
Absolute maximum and minimum voltage applied to Vout and Rfb:	+/-15V (relative to digital ground)
Absolute maximum and minimum voltage applied to any digital line:	-0.3V to +7V (relative to digital ground)
Accuracy:	
Unipolar offset:	Typ $^{1}/_{4}$ LSB, Max 1 LSB
Bipolar zero:	Typ $^{1}/_{2}$ LSB, Max 2 LSB
Linearity error:	Typ $^{1}/_{4}$ LSB, Max $^{1}/_{2}$ LSB
Linearity over temp range:	Typ $^{1}/_{2}$ LSB, Max $^{3}/_{4}$ LSB
Differential linearity:	Max $^{1}/_{2}$ LSB
Gain error:	Typ 2 LSB, Max 5 LSB
Temperature drift:	10ppm/°C max of full scale range

[1] This figure is the maximum transfer rate of the PC-166x hardware. In practise, the limiting factors are the host computer and software speed. On an 8MHz AT, an assembly program continuously writing a constant to an output port achieves a transfer rate of around 500kHz.

Reference voltages

Number of references:	4 on PC-167 and PC-266 boards 1 on PC-176A and PC-167B boards
Resolution:	16-bit
Settling time:	7us to 4 LSBs 9us to 2 LSBs
Accuracy:	
Relative accuracy:	2 LSB at 25°C, 4 LSB over full temp range
Differential nonlinearity:	½ LSB
Gain offset and bipolar error (each):	4 LSB at 25°C, 8 LSB over temp range
Gain, offset and bipolar zero temp drift (each):	2 ppm/°C of full scale range
Output load:	2kOhm min, 1000pF max
Output resistance:	0.3Ohm
Short circuit current:	limited to 25mA to 0V or any power supply to +/-15V
Absolute maximum and minimum voltage applied to Vout:	+/-25V (relative to digital ground)

Timing

Internal reference oscillator:	2MHz
Internal prescaler, interrupt timer and update timer:	16-bit (each)
External clock:	TTL compatible
External gate:	TTL compatible

Digital Inputs

All digital inputs:	TTL compatible
Input voltage:	Logic 0: -0.5 - 0.8V Logic 1: 2V - 5V

Host Computer interface

Bus type:	AT, ISA, EISA	
Data path:	16-bit	
No of locations:	64 I/O mapped (ie 32 16-bit registers)	
Base address range:	0 to 3FC0h, on 64 byte boundaries, DIP switch selectable	
Wait states:	0, 1, 2, 4, 8 selectable; only affect cycles directed at the PC-166 boards	
Interrupts:	Timing clock output and analog output buffer empty, software controlled	
Power requirements:	+5V	Max 800 mA (PC-167)
	+12V	0.0 mA
	-12V	0.0 mA
	-5V	0.0 mA

Physical

Operating temperature range:	0°C to 70°C
Relative humidity:	5% to 90%, non-condensing
Dimensions:	15.9cm long, 11.1cm high (excluding connector and bracket); occupies less than one half AT slot
I/O Connector:	50 way IDC connector

Appendix A

Base Address Switch Settings

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
0h	ON	ON	ON	ON	ON	ON	ON	ON	0h
40h	ON	ON	ON	ON	ON	ON	ON	OFF	40h
80h	ON	ON	ON	ON	ON	ON	OFF	ON	80h
C0h	ON	ON	ON	ON	ON	ON	OFF	OFF	C0h
100h	ON	ON	ON	ON	ON	OFF	ON	ON	100h
140h	ON	ON	ON	ON	ON	OFF	ON	OFF	140h
180h	ON	ON	ON	ON	ON	OFF	OFF	ON	180h
1C0h	ON	ON	ON	ON	ON	OFF	OFF	OFF	1C0h
200h	ON	ON	ON	ON	OFF	ON	ON	ON	200h
240h	ON	ON	ON	ON	OFF	ON	ON	OFF	240h
280h	ON	ON	ON	ON	OFF	ON	OFF	ON	280h
2C0h	ON	ON	ON	ON	OFF	ON	OFF	OFF	2C0h
300h	ON	ON	ON	ON	OFF	OFF	ON	ON	300h
340h	ON	ON	ON	ON	OFF	OFF	ON	OFF	340h
380h	ON	ON	ON	ON	OFF	OFF	OFF	ON	380h
3C0h	ON	ON	ON	ON	OFF	OFF	OFF	OFF	3C0h
400h	ON	ON	ON	OFF	ON	ON	ON	ON	400h
440h	ON	ON	ON	OFF	ON	ON	ON	OFF	440h
480h	ON	ON	ON	OFF	ON	ON	OFF	ON	480h
4C0h	ON	ON	ON	OFF	ON	ON	OFF	OFF	4C0h
500h	ON	ON	ON	OFF	ON	OFF	ON	ON	500h
540h	ON	ON	ON	OFF	ON	OFF	ON	OFF	540h
580h	ON	ON	ON	OFF	ON	OFF	OFF	ON	580h
5C0h	ON	ON	ON	OFF	ON	OFF	OFF	OFF	5C0h
600h	ON	ON	ON	OFF	OFF	ON	ON	ON	600h
640h	ON	ON	ON	OFF	OFF	ON	ON	OFF	640h
680h	ON	ON	ON	OFF	OFF	ON	OFF	ON	680h
6C0h	ON	ON	ON	OFF	OFF	ON	OFF	OFF	6C0h
700h	ON	ON	ON	OFF	OFF	OFF	ON	ON	700h
740h	ON	ON	ON	OFF	OFF	OFF	ON	OFF	740h
780h	ON	ON	ON	OFF	OFF	OFF	OFF	ON	780h
7C0h	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	7C0h
800h	ON	ON	OFF	ON	ON	ON	ON	ON	800h
840h	ON	ON	OFF	ON	ON	ON	ON	OFF	840h
880h	ON	ON	OFF	ON	ON	ON	OFF	ON	880h
8C0h	ON	ON	OFF	ON	ON	ON	OFF	OFF	8C0h
900h	ON	ON	OFF	ON	ON	OFF	ON	ON	900h
940h	ON	ON	OFF	ON	ON	OFF	ON	OFF	940h
980h	ON	ON	OFF	ON	ON	OFF	OFF	ON	980h
9C0h	ON	ON	OFF	ON	ON	OFF	OFF	OFF	9C0h
A00h	ON	ON	OFF	ON	OFF	ON	ON	ON	A00h
A40h	ON	ON	OFF	ON	OFF	ON	ON	OFF	A40h
A80h	ON	ON	OFF	ON	OFF	ON	OFF	ON	A80h
AC0h	ON	ON	OFF	ON	OFF	ON	OFF	OFF	AC0h
B00h	ON	ON	OFF	ON	OFF	OFF	ON	ON	B00h
B40h	ON	ON	OFF	ON	OFF	OFF	ON	OFF	B40h
B80h	ON	ON	OFF	ON	OFF	OFF	OFF	ON	B80h
BC0h	ON	ON	OFF	ON	OFF	OFF	OFF	OFF	BC0h
C00h	ON	ON	OFF	OFF	ON	ON	ON	ON	C00h
C40h	ON	ON	OFF	OFF	ON	ON	ON	OFF	C40h
C80h	ON	ON	OFF	OFF	ON	ON	OFF	ON	C80h
CC0h	ON	ON	OFF	OFF	ON	ON	OFF	OFF	CC0h
D00h	ON	ON	OFF	OFF	ON	OFF	ON	ON	D00h
D40h	ON	ON	OFF	OFF	ON	OFF	ON	OFF	D40h
D80h	ON	ON	OFF	OFF	ON	OFF	OFF	ON	D80h
DC0h	ON	ON	OFF	OFF	ON	OFF	OFF	OFF	DC0h
E00h	ON	ON	OFF	OFF	OFF	ON	ON	ON	E00h
E40h	ON	ON	OFF	OFF	OFF	ON	ON	OFF	E40h
E80h	ON	ON	OFF	OFF	OFF	ON	OFF	ON	E80h
EC0h	ON	ON	OFF	OFF	OFF	ON	OFF	OFF	EC0h
F00h	ON	ON	OFF	OFF	OFF	OFF	ON	ON	F00h
F40h	ON	ON	OFF	OFF	OFF	OFF	ON	OFF	F40h
F80h	ON	ON	OFF	OFF	OFF	OFF	OFF	ON	F80h
FC0h	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	FC0h

Table 5: Base address switch settings.

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
1000h	ON	OFF	ON	ON	ON	ON	ON	ON	1000h
1040h	ON	OFF	ON	ON	ON	ON	ON	OFF	1040h
1080h	ON	OFF	ON	ON	ON	ON	OFF	ON	1080h
10C0h	ON	OFF	ON	ON	ON	ON	OFF	OFF	10C0h
1100h	ON	OFF	ON	ON	ON	OFF	ON	ON	1100h
1140h	ON	OFF	ON	ON	ON	OFF	ON	OFF	1140h
1180h	ON	OFF	ON	ON	ON	OFF	OFF	ON	1180h
11C0h	ON	OFF	ON	ON	ON	OFF	OFF	OFF	11C0h
1200h	ON	OFF	ON	ON	OFF	ON	ON	ON	1200h
1240h	ON	OFF	ON	ON	OFF	ON	ON	OFF	1240h
1280h	ON	OFF	ON	ON	OFF	ON	OFF	ON	1280h
12C0h	ON	OFF	ON	ON	OFF	ON	OFF	OFF	12C0h
1300h	ON	OFF	ON	ON	OFF	OFF	ON	ON	1300h
1340h	ON	OFF	ON	ON	OFF	OFF	ON	OFF	1340h
1380h	ON	OFF	ON	ON	OFF	OFF	OFF	ON	1380h
13C0h	ON	OFF	ON	ON	OFF	OFF	OFF	OFF	13C0h
1400h	ON	OFF	ON	OFF	ON	ON	ON	ON	1400h
1440h	ON	OFF	ON	OFF	ON	ON	ON	OFF	1440h
1480h	ON	OFF	ON	OFF	ON	ON	OFF	ON	1480h
14C0h	ON	OFF	ON	OFF	ON	ON	OFF	OFF	14C0h
1500h	ON	OFF	ON	OFF	ON	OFF	ON	ON	1500h
1540h	ON	OFF	ON	OFF	ON	OFF	ON	OFF	1540h
1580h	ON	OFF	ON	OFF	ON	OFF	OFF	ON	1580h
15C0h	ON	OFF	ON	OFF	ON	OFF	OFF	OFF	15C0h
1600h	ON	OFF	ON	OFF	OFF	ON	ON	ON	1600h
1640h	ON	OFF	ON	OFF	OFF	ON	ON	OFF	1640h
1680h	ON	OFF	ON	OFF	OFF	ON	OFF	ON	1680h
16C0h	ON	OFF	ON	OFF	OFF	ON	OFF	OFF	16C0h
1700h	ON	OFF	ON	OFF	OFF	OFF	ON	ON	1700h
1740h	ON	OFF	ON	OFF	OFF	OFF	ON	OFF	1740h
1780h	ON	OFF	ON	OFF	OFF	OFF	OFF	ON	1780h
17C0h	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	17C0h
1800h	ON	OFF	OFF	ON	ON	ON	ON	ON	1800h
1840h	ON	OFF	OFF	ON	ON	ON	ON	OFF	1840h
1880h	ON	OFF	OFF	ON	ON	ON	OFF	ON	1880h
18C0h	ON	OFF	OFF	ON	ON	ON	OFF	OFF	18C0h
1900h	ON	OFF	OFF	ON	ON	OFF	ON	ON	1900h
1940h	ON	OFF	OFF	ON	ON	OFF	ON	OFF	1940h
1980h	ON	OFF	OFF	ON	ON	OFF	OFF	ON	1980h
19C0h	ON	OFF	OFF	ON	ON	OFF	OFF	OFF	19C0h
1A00h	ON	OFF	OFF	ON	OFF	ON	ON	ON	1A00h
1A40h	ON	OFF	OFF	ON	OFF	ON	ON	OFF	1A40h
1A80h	ON	OFF	OFF	ON	OFF	ON	OFF	ON	1A80h
1AC0h	ON	OFF	OFF	ON	OFF	ON	OFF	OFF	1AC0h
1B00h	ON	OFF	OFF	ON	OFF	OFF	ON	ON	1B00h
1B40h	ON	OFF	OFF	ON	OFF	OFF	ON	OFF	1B40h
1B80h	ON	OFF	OFF	ON	OFF	OFF	OFF	ON	1B80h
1BC0h	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	1BC0h
1C00h	ON	OFF	OFF	OFF	ON	ON	ON	ON	1C00h
1C40h	ON	OFF	OFF	OFF	ON	ON	ON	OFF	1C40h
1C80h	ON	OFF	OFF	OFF	ON	ON	OFF	ON	1C80h
1CC0h	ON	OFF	OFF	OFF	ON	ON	OFF	OFF	1CC0h
1D00h	ON	OFF	OFF	OFF	ON	OFF	ON	ON	1D00h
1D40h	ON	OFF	OFF	OFF	ON	OFF	ON	OFF	1D40h
1D80h	ON	OFF	OFF	OFF	ON	OFF	OFF	ON	1D80h
1DC0h	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	1DC0h
1E00h	ON	OFF	OFF	OFF	OFF	ON	ON	ON	1E00h
1E40h	ON	OFF	OFF	OFF	OFF	ON	ON	OFF	1E40h
1E80h	ON	OFF	OFF	OFF	OFF	ON	OFF	ON	1E80h
1EC0h	ON	OFF	OFF	OFF	OFF	ON	OFF	OFF	1EC0h
1F00h	ON	OFF	OFF	OFF	OFF	OFF	ON	ON	1F00h
1F40h	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF	1F40h
1F80h	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	1F80h
1FC0h	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	1FC0h

Table 5: Base address switch settings (continued).

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
2000h	OFF	ON	ON	ON	ON	ON	ON	ON	2000h
2040h	OFF	ON	ON	ON	ON	ON	ON	OFF	2040h
2080h	OFF	ON	ON	ON	ON	ON	OFF	ON	2080h
20C0h	OFF	ON	ON	ON	ON	ON	OFF	OFF	20C0h
2100h	OFF	ON	ON	ON	ON	OFF	ON	ON	2100h
2140h	OFF	ON	ON	ON	ON	OFF	ON	OFF	2140h
2180h	OFF	ON	ON	ON	ON	OFF	OFF	ON	2180h
21C0h	OFF	ON	ON	ON	ON	OFF	OFF	OFF	21C0h
2200h	OFF	ON	ON	ON	OFF	ON	ON	ON	2200h
2240h	OFF	ON	ON	ON	OFF	ON	ON	OFF	2240h
2280h	OFF	ON	ON	ON	OFF	ON	OFF	ON	2280h
22C0h	OFF	ON	ON	ON	OFF	ON	OFF	OFF	22C0h
2300h	OFF	ON	ON	ON	OFF	OFF	ON	ON	2300h
2340h	OFF	ON	ON	ON	OFF	OFF	ON	OFF	2340h
2380h	OFF	ON	ON	ON	OFF	OFF	OFF	ON	2380h
23C0h	OFF	ON	ON	ON	OFF	OFF	OFF	OFF	23C0h
2400h	OFF	ON	ON	OFF	ON	ON	ON	ON	2400h
2440h	OFF	ON	ON	OFF	ON	ON	ON	OFF	2440h
2480h	OFF	ON	ON	OFF	ON	ON	OFF	ON	2480h
24C0h	OFF	ON	ON	OFF	ON	ON	OFF	OFF	24C0h
2500h	OFF	ON	ON	OFF	ON	OFF	ON	ON	2500h
2540h	OFF	ON	ON	OFF	ON	OFF	ON	OFF	2540h
2580h	OFF	ON	ON	OFF	ON	OFF	OFF	ON	2580h
25C0h	OFF	ON	ON	OFF	ON	OFF	OFF	OFF	25C0h
2600h	OFF	ON	ON	OFF	OFF	ON	ON	ON	2600h
2640h	OFF	ON	ON	OFF	OFF	ON	ON	OFF	2640h
2680h	OFF	ON	ON	OFF	OFF	ON	OFF	ON	2680h
26C0h	OFF	ON	ON	OFF	OFF	ON	OFF	OFF	26C0h
2700h	OFF	ON	ON	OFF	OFF	OFF	ON	ON	2700h
2740h	OFF	ON	ON	OFF	OFF	OFF	ON	OFF	2740h
2780h	OFF	ON	ON	OFF	OFF	OFF	OFF	ON	2780h
27C0h	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	27C0h
2800h	OFF	ON	OFF	ON	ON	ON	ON	ON	2800h
2840h	OFF	ON	OFF	ON	ON	ON	ON	OFF	2840h
2880h	OFF	ON	OFF	ON	ON	ON	OFF	ON	2880h
28C0h	OFF	ON	OFF	ON	ON	ON	OFF	OFF	28C0h
2900h	OFF	ON	OFF	ON	ON	OFF	ON	ON	2900h
2940h	OFF	ON	OFF	ON	ON	OFF	ON	OFF	2940h
2980h	OFF	ON	OFF	ON	ON	OFF	OFF	ON	2980h
29C0h	OFF	ON	OFF	ON	ON	OFF	OFF	OFF	29C0h
2A00h	OFF	ON	OFF	ON	OFF	ON	ON	ON	2A00h
2A40h	OFF	ON	OFF	ON	OFF	ON	ON	OFF	2A40h
2A80h	OFF	ON	OFF	ON	OFF	ON	OFF	ON	2A80h
2AC0h	OFF	ON	OFF	ON	OFF	ON	OFF	OFF	2AC0h
2B00h	OFF	ON	OFF	ON	OFF	OFF	ON	ON	2B00h
2B40h	OFF	ON	OFF	ON	OFF	OFF	ON	OFF	2B40h
2B80h	OFF	ON	OFF	ON	OFF	OFF	OFF	ON	2B80h
2BC0h	OFF	ON	OFF	ON	OFF	OFF	OFF	OFF	2BC0h
2C00h	OFF	ON	OFF	OFF	ON	ON	ON	ON	2C00h
2C40h	OFF	ON	OFF	OFF	ON	ON	ON	OFF	2C40h
2C80h	OFF	ON	OFF	OFF	ON	ON	OFF	ON	2C80h
2CC0h	OFF	ON	OFF	OFF	ON	ON	OFF	OFF	2CC0h
2D00h	OFF	ON	OFF	OFF	ON	OFF	ON	ON	2D00h
2D40h	OFF	ON	OFF	OFF	ON	OFF	ON	OFF	2D40h
2D80h	OFF	ON	OFF	OFF	ON	OFF	OFF	ON	2D80h
2DC0h	OFF	ON	OFF	OFF	ON	OFF	OFF	OFF	2DC0h
2E00h	OFF	ON	OFF	OFF	OFF	ON	ON	ON	2E00h
2E40h	OFF	ON	OFF	OFF	OFF	ON	ON	OFF	2E40h
2E80h	OFF	ON	OFF	OFF	OFF	ON	OFF	ON	2E80h
2EC0h	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF	2EC0h
2F00h	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	2F00h
2F40h	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF	2F40h
2F80h	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	2F80h
2FC0h	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	2FC0h

Table 5: Base address switch settings (continued).

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	Base Address
3000h	OFF	OFF	ON	ON	ON	ON	ON	ON	3000h
3040h	OFF	OFF	ON	ON	ON	ON	ON	OFF	3040h
3080h	OFF	OFF	ON	ON	ON	ON	OFF	ON	3080h
30C0h	OFF	OFF	ON	ON	ON	ON	OFF	OFF	30C0h
3100h	OFF	OFF	ON	ON	ON	OFF	ON	ON	3100h
3140h	OFF	OFF	ON	ON	ON	OFF	ON	OFF	3140h
3180h	OFF	OFF	ON	ON	ON	OFF	OFF	ON	3180h
31C0h	OFF	OFF	ON	ON	ON	OFF	OFF	OFF	31C0h
3200h	OFF	OFF	ON	ON	OFF	ON	ON	ON	3200h
3240h	OFF	OFF	ON	ON	OFF	ON	ON	OFF	3240h
3280h	OFF	OFF	ON	ON	OFF	ON	OFF	ON	3280h
32C0h	OFF	OFF	ON	ON	OFF	ON	OFF	OFF	32C0h
3300h	OFF	OFF	ON	ON	OFF	OFF	ON	ON	3300h
3340h	OFF	OFF	ON	ON	OFF	OFF	ON	OFF	3340h
3380h	OFF	OFF	ON	ON	OFF	OFF	OFF	ON	3380h
33C0h	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF	33C0h
3400h	OFF	OFF	ON	OFF	ON	ON	ON	ON	3400h
3440h	OFF	OFF	ON	OFF	ON	ON	ON	OFF	3440h
3480h	OFF	OFF	ON	OFF	ON	ON	OFF	ON	3480h
34C0h	OFF	OFF	ON	OFF	ON	ON	OFF	OFF	34C0h
3500h	OFF	OFF	ON	OFF	ON	OFF	ON	ON	3500h
3540h	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	3540h
3580h	OFF	OFF	ON	OFF	ON	OFF	OFF	ON	3580h
35C0h	OFF	OFF	ON	OFF	ON	OFF	OFF	OFF	35C0h
3600h	OFF	OFF	ON	OFF	OFF	ON	ON	ON	3600h
3640h	OFF	OFF	ON	OFF	OFF	ON	ON	OFF	3640h
3680h	OFF	OFF	ON	OFF	OFF	ON	OFF	ON	3680h
36C0h	OFF	OFF	ON	OFF	OFF	ON	OFF	OFF	36C0h
3700h	OFF	OFF	ON	OFF	OFF	OFF	ON	ON	3700h
3740h	OFF	OFF	ON	OFF	OFF	OFF	ON	OFF	3740h
3780h	OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	3780h
37C0h	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	37C0h
3800h	OFF	OFF	OFF	ON	ON	ON	ON	ON	3800h
3840h	OFF	OFF	OFF	ON	ON	ON	ON	OFF	3840h
3880h	OFF	OFF	OFF	ON	ON	ON	OFF	ON	3880h
38C0h	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	38C0h
3900h	OFF	OFF	OFF	ON	ON	OFF	ON	ON	3900h
3940h	OFF	OFF	OFF	ON	ON	OFF	ON	OFF	3940h
3980h	OFF	OFF	OFF	ON	ON	OFF	OFF	ON	3980h
39C0h	OFF	OFF	OFF	ON	ON	OFF	OFF	OFF	39C0h
3A00h	OFF	OFF	OFF	ON	OFF	ON	ON	ON	3A00h
3A40h	OFF	OFF	OFF	ON	OFF	ON	ON	OFF	3A40h
3A80h	OFF	OFF	OFF	ON	OFF	ON	OFF	ON	3A80h
3AC0h	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF	3AC0h
3B00h	OFF	OFF	OFF	ON	OFF	OFF	ON	ON	3B00h
3B40h	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF	3B40h
3B80h	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	3B80h
3BC0h	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	3BC0h
3C00h	OFF	OFF	OFF	OFF	ON	ON	ON	ON	3C00h
3C40h	OFF	OFF	OFF	OFF	ON	ON	ON	OFF	3C40h
3C80h	OFF	OFF	OFF	OFF	ON	ON	OFF	ON	3C80h
3CC0h	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF	3CC0h
3D00h	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	3D00h
3D40h	OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	3D40h
3D80h	OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	3D80h
3DC0h	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	3DC0h
3E00h	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	3E00h
3E40h	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF	3E40h
3E80h	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	3E80h
3EC0h	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	3EC0h
3F00h	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	3F00h
3F40h	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	3F40h
3F80h	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	3F80h
3FC0h	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3FC0h

Table 5: Base address switch settings (continued).