User manual for

PC-26, PC-30 and PC-39

High Performance Analog I/O Boards for IBM PC, PC/XT, PC/AT, PS/2 Model 25 and 30 and compatible Computer Systems.

First edition.

January 1990

January 1990 Printing

Table of Contents

Preface i

Chapter 1. Introduction 1

1.1. Overview 1
1.2. Features 1
1.3. Software support 3
1.4. Throughput 5
1.5. Getting Started 5
1.6. Accessories 6

Chapter 2. Architecture 7

2.1. D/A Subsystem 7
2.2. A/D Subsystem 7
2.3. Bus interface. 8
2.4. Timing and control 8
2.5. Counter/timer. 10
2.6. A/D operations. 10
2.7. Digital I/O. 11

Chapter 3. Configuring the PC-30 board 13

3.1. Introduction 13
3.2. Changing the Configuration 13
3.3. Bus Interface Configuration. 15
3.4. D/A jumper selections. 17
3.5. A/D configuration. 18

Chapter 4. Interconnections 23

4.1. Introduction 23
4.2. Connections to the IBM backplane. 23
4.3. User connection. 23

4.4. Analog I/O 26
4.5. Digital I/O. 27

Chapter 5. Register Structure 35

5.1. Introduction 35
5.2. Register structure. 35

Chapter 6. Programming Guide 55

6.1. Converting from binary to analog values 55
6.2. Initialization 56
6.3. Clearing the A/D subsystem 57
6.4. Writing to the D/A converters 57
6.5. Digital I/O 57
6.6. Obtaining a single A/D reading - PC-26/PC-30 57
6.7. Obtaining a single A/D reading - PC-39 58
6.8. Setting the sample rate 58
6.9. Detecting the end of conversion 58
6.10. Obtaining a series of A/D conversions by polled I/O 59
6.11. Interrupts 59
6.12. Single channel DMA 60
6.13. DMA data format 61

Chapter 7. Calibration 63

7.1. Introduction 63
7.2. A/D calibration 63
7.3. A/D calibration Software 65
7.4. DAC0 and DAC1 calibration 65
7.5. DAC2 and DAC3 calibration 67
7.6. D/A calibration Software 67

Appendix A. Specifications 69

Appendix B. PC-26/PC-30/PC-39 Compatibility 75

Appendix C. Layout Diagrams 77

Appendix D. Problem Determination Guide 81

Index

Preface

This manual is written for users of the PC-26, PC-30 and PC-39 series of analog I/O boards. It provides all information necessary to successfully program and operate all boards in the series.

The companion manual to this, "User Manual for PC-30 Driver Software", describes the use of the supplied driver software.

This manual assumes:

- That you have a basic knowledge of electronic circuitry and measurement techniques.
- That you are familiar with the host PC which you are using.
- That you are capable of writing your own programs.

The manual contains the following sections.

Chapter 1 - Introduction.

- Chapter 1 contains an overview of the PC-30[1] series of boards.

Chapter 2 - Architecture.

- Chapter 2 discusses the basic operation of the PC-30 board.

Chapter 3 - Configuration.

- Chapter 3 discusses the selection of various board parameters and the configuration of the board for various operating requirements.

Chapter 4 - Interconnection.

- Chapter 4 describes the connection of the PC-30 series of boards to the host computer and to user inputs.

Chapter 5 - Register structure.

[1] For the rest of the manual, PC-30 will be used to refer to the PC-26, PC-30 and PC-39. Where information is specific to a particular board, this will be stated in the text.

- Chapter 5 describes the register structure of the PC-30 series of boards.

Chapter 6 - Programming guide.

- Chapter 6 provides a tutorial style guide to programming the PC-30. Information is provided as to where in the driver software to obtain examples of the topics discussed.

Chapter 7 - Calibration.

- Chapter 7 describes the procedures and equipment required to calibrate the PC-30 series of boards. Calibration software included with the PC-30 is also described.

Appendix A - Hardware Specifications.

- Appendix A provides complete electrical specifications for the PC-30 series of boards.

Appendix B - PC-26/PC-30/PC-39 Differences.

- Appendix B discusses the differences between the PC-26/PC-30, and the PC-39.

Appendix C - PC-30 Component Layout.

- Appendix C contains layout diagrams of the PC-26, PC-30 and PC-39.

Appendix D - Problem Determination guide.

- Appendix D contains information which may help you if you are experiencing problems with your PC-30.

Chapter 1

Introduction

1.1. Overview

The PC-26, PC-30 and PC-39 series of boards are full size, low cost, high accuracy analog and digital I/O boards for the IBM PC, PC/XT, PC/AT, PS/2 model 25, PS/2 model 30 and compatible series of computers.

- PC-30. The PC-30 forms the basis of the range, and features 16 analog input channels with 25 KHz throughput, two 12 bit D/A outputs, two 8 bit D/A outputs and 24 digital I/O lines.
- PC-39. This board has all the features of the PC-30, but also provides A/D throughput of 80 KHz, DMA capability and improved analog specifications.
- PC-26. The PC-26 has similar features to the PC-30, but does not have D/A outputs or digital I/O capability.

1.2. Features

The PC-26, PC-30 can be plugged into any of the fully bussed slots in a PC/XT or compatible computer. Installation in PC/AT, PS/2 or compatible computers is not recommended.

The PC-39 can be plugged into any of the fully bussed slots in a PC/XT/AT, PS/2 model 25 or 30 or compatible computer.

1.2.1. A/D subsystem

The A/D subsystem's major component is a monolithic analog to digital converter, which accepts analog voltage inputs from sensors, such as pressure transducers and thermocouples, and converts them into 12 bit digital codes.

This code is transmitted to the host processor, which processes it according to the software in use at

the time.

The A/D section allows for 16 single-ended inputs, and can be configured for unipolar (input range of 0 to 10V) or bipolar (input ranges of +/-5V or +/-10V) operation. Resolution is 12 bits. For unipolar inputs, the output code is straight binary, and for bipolar, offset binary.

The A/D may be operated in either single conversion or continuous conversion mode. In single conversion mode the board performs a single conversion on the selected input channel and stops on completion of this conversion. In continuous conversion mode conversions are performed at a set rate. This rate is set by programming the PC-30's internal timer or an external clock source.

A/D conversions may be monitored by either polled I/O, DMA or by interrupts. In polled I/O mode the software continuously polls the board's status register to check for completion of the current A/D conversion. DMA (Direct Memory Access) is used to transfer data direct from the A/D to memory. Only the PC-39 has DMA capability. In interrupt mode, the board automatically generates a hardware interrupt on completion of each conversion.

Key specifications

- A/D resolution: 12 Bits
- Nonlinearity: Less than +/-0.75 LSB
- A/D full scale input ranges: 0 to +10V, -5 to +5V, and -10 to +10V jumper selectable.
- Number of A/D inputs: 16 single ended.
- A/D throughput rate: 80 KHz (PC-39), 25KHz (PC-26 and PC-30).

1.2.2. D/A Subsystem

The D/A subsystem consists of two 12-bit D/A converters, DAC0 and DAC1, and two 8-bit D/A converters, DAC2 and DAC3. The D/A subsystem is not present on the PC-26. Digital outputs are received from the host processor and converted to an analog voltage output required by the application in hand. The four DACs are independent of one another, and can operate at a throughput of up to 130KHz. Output ranges are independently configurable as 0-+10V unipolar, or as +/-10v bipolar.

Key specifications: DAC0 and DAC1

- D/A resolution: 12 Bits
- D/A nonlinearity: Within 0.01% FSR
- Full scale output ranges: 0 to +10V, -10V to +10V.
- D/A throughput rate: 130 KHz.

Key specifications: DAC2 and DAC3

- D/A resolution: 8 Bits
- D/A nonlinearity: Within 0.38% FSR

- Full scale output ranges: 0 to +10V, -10V to +10.
- D/A throughput rate: 130 KHz.

1.2.3. Digital I/O subsystem

The digital I/O subsystem is an interface for the transfer of digital data from and to the PC bus to and from one or more peripheral devices connected to the PC-30. The digital I/O subsystem is not present on the PC-26. There are three bidirectional eight bit digital I/O ports, which can each be used in a variety of operating modes.

1.2.4. Interface logic

The PC-30 is accessed via I/O operations performed by the host processor. Of the 14 bit address received by the board, the most significant 9 bits select the board, and the least significant 5 bits select the register to be accessed.

The PC-30 occupies 32 byte locations: six byte locations for the A/D subsystem, six for the D/A subsystem, four for the digital I/O subsystem, four for the counter/timer system, and twelve for control and manufacturing test functions. The base address of the PC-26 and PC-30 can be selected as either 700 or 780 (hex), while the PC-39 can be located at 700, 780, F00 or F80 (hex).

The PC-30 operates from the +5V, +12V and -12V lines of the PC bus.

1.3. Software support

Software support for the PC-26, PC-30 and PC-39 is available in a variety of forms :

1.3.1. PC-30 software support pack

The PC-30 software support pack provides two packages; the PC-30 driver software, and Status-30.

1.3.1.1 Driver software

This consists of a set of real time device drivers for use with the PC-26, PC-30 and PC-39 boards. These device drivers are written in C, and are callable from most compiled languages, including the following :

- Microsoft C.
- Turbo C.
- Turbo Pascal versions 4 and 5.

The driver software allows programers to control the PC-30 via high level function calls, so allowing users to write custom software without understanding the low level operation of the PC-30. The driver package provides access to all PC-30 capabilities, including DMA and interrupts. 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.

1.3.1.2 Status-30

Status-30 is an application program, which combines the features of an oscilloscope, spectrum

analyzer and data logger in one package. Now in release 2, Status features a graphical interface, pull-down menus, and extensive on-line help. The program can be controlled either by a mouse or from the keyboard, and has sophisticated zoom, pan, and cursor measurement capabilities.

Status's data analysis capabilities not only include FFT (Fast Fourier Transforms), but also include Chirp-Z transforms. Chirp-Z transforms can provide more than 60 times better frequency resolution than conventional FFTs. For example, sampling at 50KHz, Status-30 can provide frequency resolution of 0.2 Hz with a record length of only 4096 samples.

Waveforms and signal spectrums can be plotted to a variety of plotters. In addition, Status-30 can store data in a variety of formats, compatible with most spreadsheet and graphing programs, as well as special purpose programs such as MathCad and Hypersignal.

1.3.2. IoCalc

IoCalc is a spreadsheet program, with the ability to acquire, process and output data to and from analog and digital I/O devices in real-time. Not just an add-on for a spreadsheet program designed for business applications, IoCalc is a custom written real-time program, optimized for engineering and scientific users. Under OS/2, IoCalc provides full multi-tasking operation, allied with powerful inter-task communication capabilities.

Data is acquired directly into spreadsheet cells, processed, and in turn sent to output devices direct from spreadsheet cells.

The entire spreadsheet, or part of it, can be updated at fixed intervals, with timing resolution down to 10 milliseconds. Any section of the spreadsheet can also be logged to disk or printer at fixed intervals.

IoCalc can be used to implement virtually any process that can be represented mathematically. Sample spreadsheets supplied with IoCalc include : Control loops (PID and 'Bang-bang'), intelligent data loggers, digital filters, multi-channel multimeters, thermocouple compensation and oscillators.

1.3.2.1 Features

- True real-time, multi-tasking data processing
- Familiar user interface
- Can be used to implement control loops, data loggers, digital filters and more
- Menu driven, with context sensitive help.
- Either DOS or OS/2 protected mode operation.

1.3.2.2 Applications

- Process control
- Data logging
- Monitoring
- Automatic test
- Smart instrumentation

- Laboratory data collection
- Virtual instruments

1.4. Throughput

The throughput of the PC-30 series of boards is dependent on several factors, principally whether DMA or program transfer techniques are used to read data from the A/D converter.

1.4.1. DMA

DMA is Direct Memory Access and, as the name implies, data from the A/D is transferred to the PC's memory directly, without the data acquisition software in use taking any action (other than setting the hardware up initially, and waiting for the DMA to complete). This is only available on the PC-39. In this case, the processing power of the host PC system is of no consequence, and the throughput of the PC-39 will be at its maximum.

1.4.2. Program transfer.

If program transfer techniques are used (polled I/O or interrupts), the situation becomes more complex. In this case the maximum possible throughput is limited by the processing power of the CPU in the host PC, and the efficiency of the software in use. In general, throughput of greater than 30 KHz is very seldom achieved.

Note that the software drivers for the PC-30, although efficient, are written as general purpose, "idiot proof" routines. Custom assembly language routines can easily be written which will outperform these.

1.5. Getting Started

If you want to get started quickly and have not changed any of the factory installed jumpers on the PC-30, here's what to do:

i. Install the PC-30 in your computer. (Chapter 3 provides brief instructions on this, but if you are not sure, it is better to get someone who is qualified to do this).

ii. Connect up a voltage source to any (or all) of the input channels. (You can also loop the analog outputs back to the inputs). The pin-out of the PC-30 connector is shown in figure 4.1 later in the manual.

iii. Run the DIAG.EXE program on the Calibration disk. This program will execute the PC-30 diagnostics routine, and should print out the following message:

PC-30 Driver Version 1.00
PC-30 diagnostics report the following:

PC-30X found, operating correctly.

iv. If the "found" message is displayed, then the PC-30 is installed correctly. Note that the diagnostic function identifies a PC-26 as a PC-30. If the DIAG program displays some other message, then you probably need to reconfigure the board. This is discussed in chapter 3 of this manual.

1.6. Accessories

In order to assist in applying the PC-30, several accessories are available. Only a brief description is given here. Consult your dealer for full details.

1.6.1. PC-81

The PC-81 is an input expander board. Multiple PC-81s may be used to expand the input channel capability of the PC-30 to more than 65000 channels. Each PC-81 has 64 screw terminal inputs.

1.6.2. PC-22

The PC-22 is a Euro-card format single channel signal conditioning module. It provides programmable gain, and filtering functions.

1.6.3. PC-68

The PC-68 is a Euro-card format four channel strain gage signal conditioning board. It provides four independent channels with user programmable excitation, differential inputs, and a high performance instrumentation amplifier. The PC-68 can also be used simply as a four channel ultra-high performance instrumentation amplifier board.

Chapter 2

Architecture

This chapter describes the architecture of the PC-30 series of boards. The block diagram in figure 2.1 highlights the major elements contained on the board, and their interrelationship. There are four major subsections. These are the following:

2.1. D/A Subsystem

The D/A subsystem contains two 12-bit D/A converters, two 8-bit D/A converters as well as their associated circuitry, including buffer registers. The 8-bit D/A converters' outputs are updated immediately a value is written to them. In the case of the 12 D/A converters, when the low byte of the value is written, the data in the buffer register is transferred to the D/A, and hence to the analog output.

2.2. A/D Subsystem

The A/D subsystem contains several separate components:

- The input multiplexer. The multiplexer selects one of sixteen single ended input channels. This channel is selected by a channel address, obtained from the Control/Channel register.
- The sample and hold unit. The sample and hold unit holds the selected input channel steady for the duration of the A/D converter's conversion process.
- The A/D converter performs the actual A/D conversion. An A/D conversion is begun by a A/D strobe. This is generated by the timing and control section, described later.

Data may be transferred from A/D either by polled I/O or DMA. This is discussed in section 2.8.

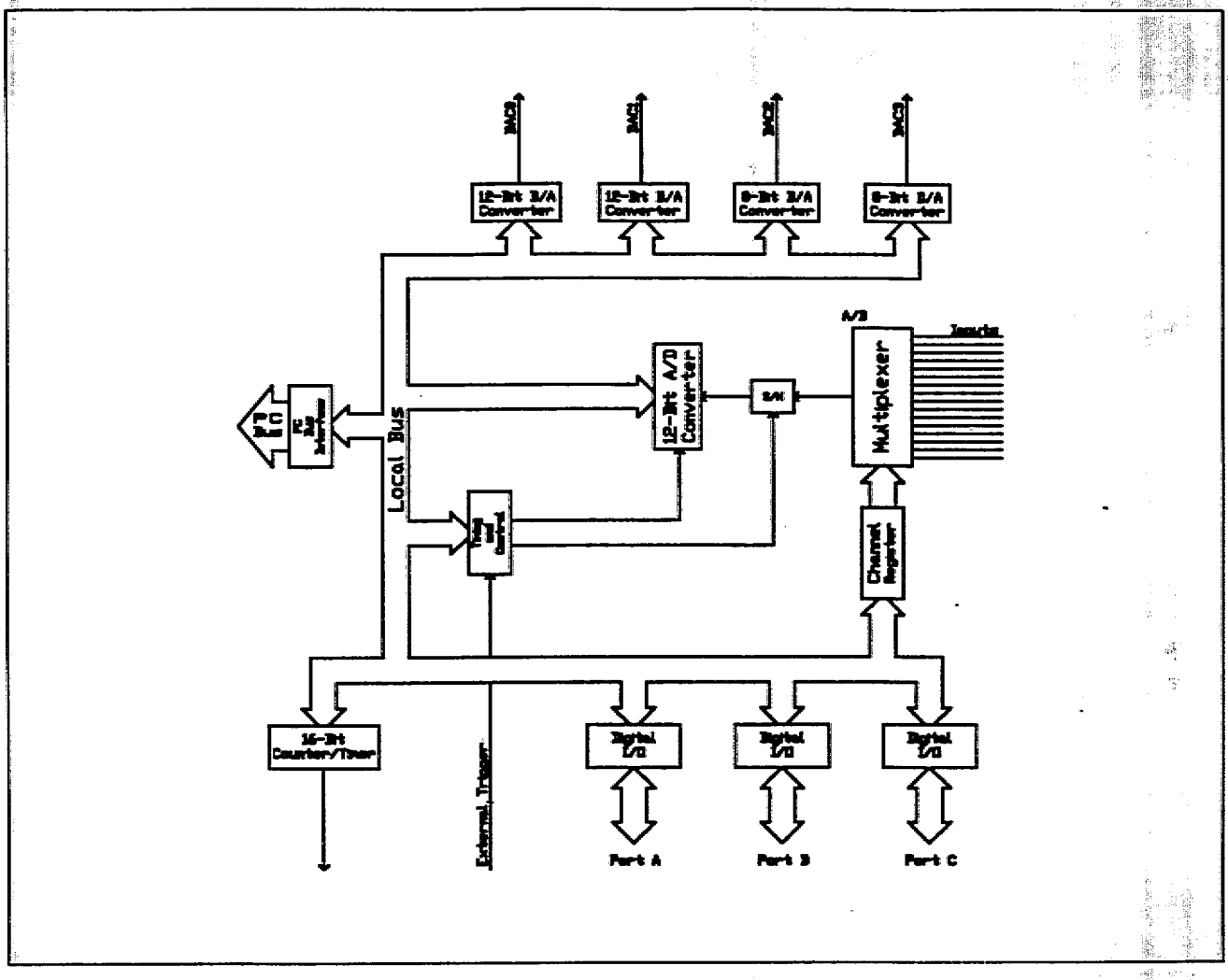


Figure 2.1.

2.3. Bus interface.

The bus interface is responsible for three functions:

i. The decoding of the board's base address. The board's base address is set by jumpers.

ii. The generation of interrupts. Interrupts can be generated at the end of each A/D conversion. The interrupt level is factory preset to 5.

iii. The generation of DMA signals (PC-39 only). DMA operations are described later in this chapter.

2.4. Timing and control

The timing and control subsection is responsible for the generation of A/D strobes, and also

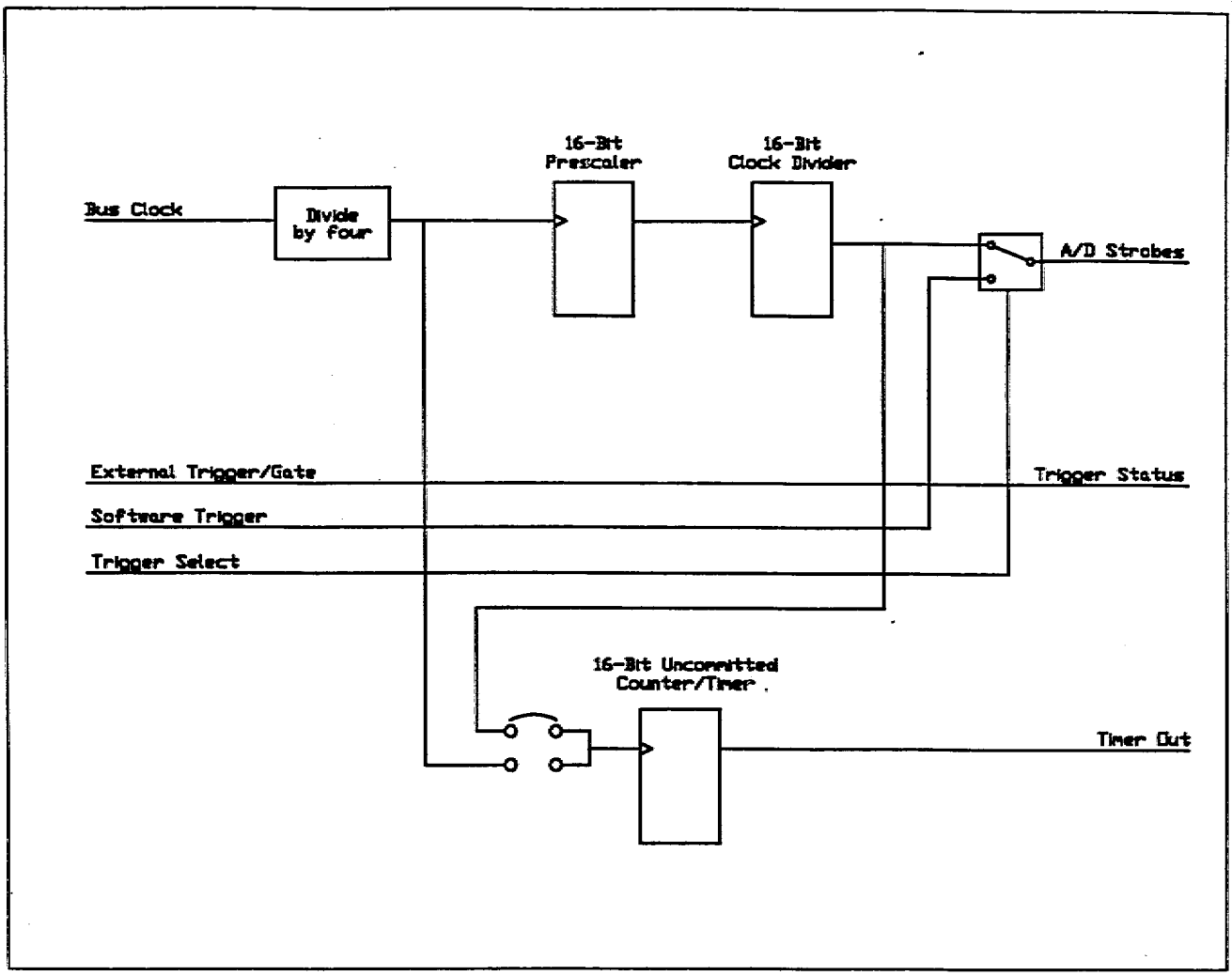


Figure 2.2.

contains an uncommitted counter/timer which can be used for signal generation, or as a frequency or pulse period counter. A/D strobes cause the A/D converter to begin a conversion. A simplified block diagram of this section is shown in figure 2.2.

A/D strobes may be selected under program control to be either hardware or software strobes.

i. Software strobes. Software strobes are generated by a write operation to a control register. They hence allow a single conversion to be started under program control.

ii. Hardware strobes. The source of hardware strobes is the internal clock. This is derived from the bus clock, which varies depending on the host PC. This signal is then divided down by a programmable ratio.

The timing and control section contains three major subsections:

2.4.1. Clock Prescaler.

The host PC's bus clock is divided by four, and then used to drive the clock prescaler. This is a 16

bit counter, the output from which drives the clock divider.

2.4.2. Clock divider.

The clock divider divides the signal from the prescaler by a programmable ratio. No strobes are generated unless the divider is enabled, so allowing the start of a set continuous conversion to be synchronized to the trigger input. Note that this divider is active only for an internally generated clock signal.

2.4.3. Clock selection multiplexer.

The Clock selection multiplexer determines whether the A/D strobe signal is derived from the hardware clock (which may be either derived from the internal clock, or from the external input) or from the software clock. The software clock is generated by a write operation to a control register, and hence allows a single conversion to be started under program control.

2.5. Counter/timer.

The PC-30 contains an uncommitted counter/timer, connected as shown in figure 2.2. This counts either pulses from the bus clock divided by four, or from the A/D prescaler. Selection between these two sources is by jumper. The output of this counter is made available on the PC-30 user connector.

2.6. A/D operations.

2.6.1. Sampling data

A/D operations proceed as follows:

i. The board is initialized. This comprises the following steps:

   a) The appropriate clock mode is selected, and the clock divider programmed.

   b) The A/D buffer and the trigger system are reset.

   c) The channel to be converted is written to the channel register.

ii. The system is then enabled, either by a trigger command, or by an external signal.

iii. As soon as conversions are enabled, A/D conversions start. These conversions occur at the rate set by either the external clock or the internal clock and the value programmed into the clock divider.

iv. Conversions continue until the board is disabled. There are two methods of transferring data from the A/D to memory. These are the following:

2.6.2. Simple Polled I/O.

Polled I/O is the simplest possible method of data transfer. It proceeds as follows:

i. The program continuously waits for a conversion cycle to complete.

ii. The data from the A/D conversion is then read, and stored in the PC's memory by the program.

iii. This process repeats until however many samples are required have been read.

Polled I/O has the advantage of extreme simplicity, but has two disadvantages.

- Transfer speed is limited by the speed of the CPU in the host PC.
- While polled I/O is being performed, the CPU is totally dedicated to this process, and cannot deal with anything else (such as keyboard input). Note that for this reason simple polled I/O is generally not suitable for use with multi-tasking operating systems such as UNIX or OS/2.

Polled I/O is generally used for single conversions, or continuous conversions at low sampling rates (less than 30 KHz)

2.6.3. Single block DMA

DMA stands for Direct Memory Access, and is available only on the PC-39. It proceeds as follows:

i. The host PC's DMA hardware is first set up with the address of the memory into which the A/D samples are to be put, and the number of samples to be obtained. The board is then initialized, and sampling started, as described above. The program can then continue with any other task, such as checking the keyboard for input.

ii. When the A/D buffer contains data, a DMA cycle is initiated.

iii. This DMA cycle reads the data from the A/D buffer, and stores it in the PC's memory, without the CPU taking any action.

iv. This process repeats until however many samples are required have been read.

v. The program checks the PC-39 to see if all the required samples have been transferred. Note that this check can be done at any time, unlike for polled I/O, where the A/D status must be checked quickly enough to ensure that data is read before the next A/D conversion completes.

The primary advantage of DMA operation is the very high transfer rate, but the number of samples which can be acquired in a single block is limited by the PC hardware to 32768.

2.7. Digital I/O.

The digital I/O section of the PC-30 consists of three 8-bit ports (port A, B and C), which can be configured in a variety of operating modes. The digital I/O portion of the PC-30 emulates, and is 100% compatible with, an 8255 type PPI (Programmable Peripheral Interface).

The three ports are divided into two groups, group A (consisting of port A and the upper half of port C) and group B (consisting of port B and the lower half of port C). Each of these groups can be individually configured into one of three operating modes.

2.7.1. Operating modes.

The three basic modes of operation are as follows:

- Mode 0 - Basic I/O.
- Mode 1 - Strobed I/O.
- Mode 2 - Bidirectional bus operations.

At power up, the interface is set to mode 0. Each mode will be described in detail in the next sections.

2.7.2. Mode 0 (Basic Input/Output).

Mode 0 characteristics are as follows:

- Two 8-bit ports (A and B) and two 4-bit ports (upper and lower halves of port C).
- Any port can either operate as an input or an output.
- This mode of operation provides simple I/O operations. Any port can be used either for input or output, but not for both.

2.7.3. Mode 1 (Strobed Input/Output).

Mode 1 characteristics are as follows:

- Two groups, each consisting of one 8-bit, and one 4-bit port.
- The 4-bit port is used for control and status of the 8-bit port.
- The 8-bit port may be used for either input or output.
- Input and output operations are latched.
- This mode of operation provides I/O operations with a simple handshake protocol.

2.7.4. Mode 2 (Strobed Bidirectional Input/Output).

Mode 2 characteristics are as follows:

- One 8-bit bidirectional port, and one 5 bit control port.
- Can be used in group A only.
- The 5-bit port is used for control and status of the 8-bit port.
- Input and output operations are latched.
- Port B can still be used in mode 0.
- This mode of operation provides a means for bidirectional I/O operations on a single 8-bit port.

Bit definitions for each mode are described in chapter 4, and programming the various modes in chapter 5.

Chapter 3

Configuring the board

3.1. Introduction

The PC-26, PC-30 and PC-39 boards can be configured in many different ways to suit each user's individual requirements. This configuration is set by the position of the various mini-jumps on the board. Each set of mini-jumps controls a specific aspect of the operation of the board. These are as follows :

i. Bus interface. The base address of the board and the DMA level used by the board can be set. As supplied by the factory, the base address is set to 700H, the DMA level to 1 (PC-39), and the interrupt level to 5. This allows operation in a standard PC/XT/AT which contains only conventional boards (Multifunction boards, disk controller boards, display boards etc), but may require modification if exotic boards (other scientific boards, certain backup systems etc) are installed. Note that the DMA request lines are electronically disconnected from the PC bus unless specifically enabled by software, even if a DMA level selection jumper is installed.

ii. D/A operation. The output range of all four D/A converters may be selected independently by mini-jumps. As supplied by the factory, all D/A outputs are set for bipolar output, -10 to +10 volts.

iii. A/D operation. The input range of the A/D may be set, by jumper, for uni- or bipolar operation, for input voltage ranges 0 - +10V, +/-5V +/-10V.

3.2. Changing the Configuration

The jumpers may be located either from the diagram in appendix C, or from the labels on the PC-30 board itself.

In order to change the jumpers settings, follow the procedure below :

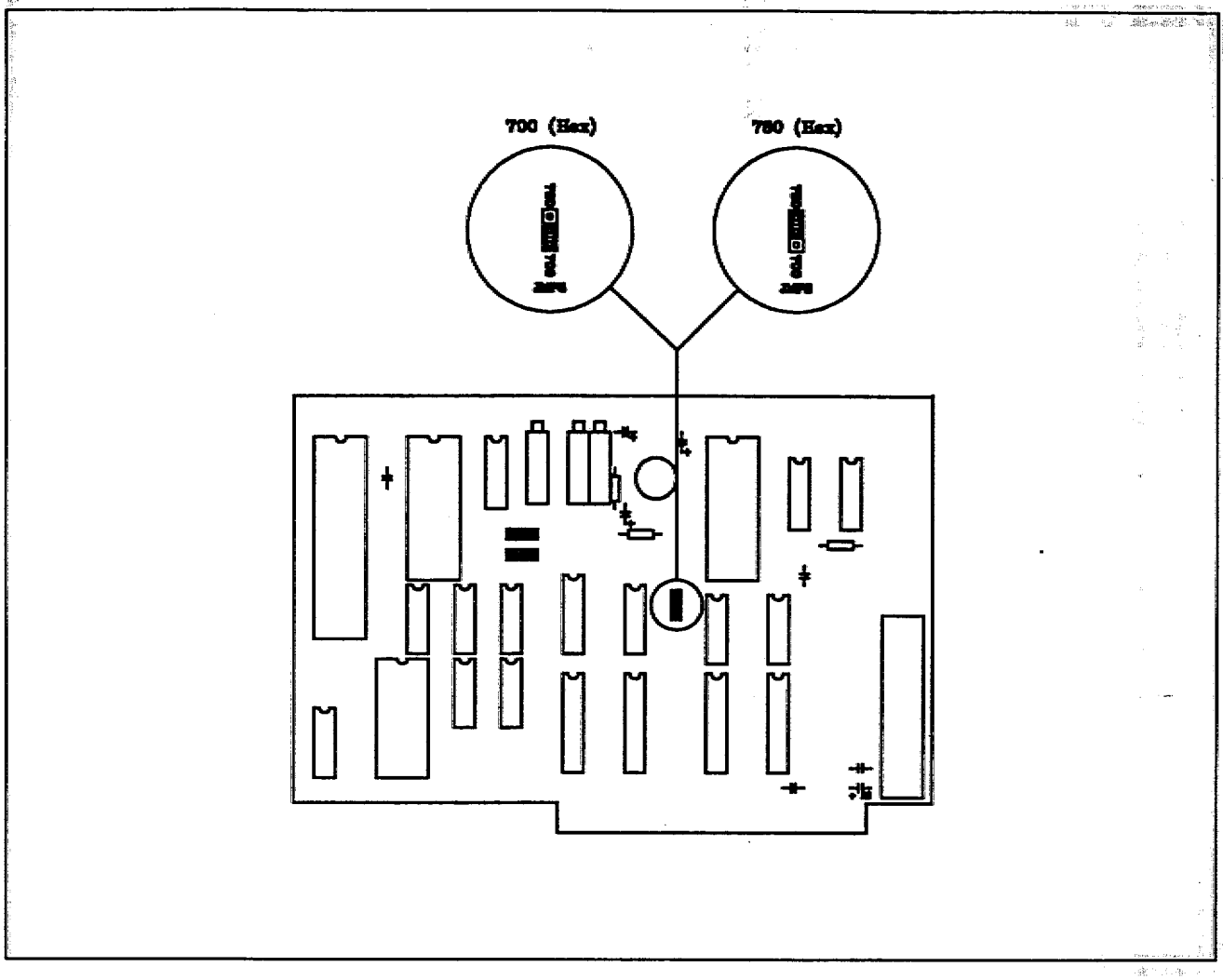


Figure 3.1. PC-26 base address jumper settings

i. Switch off the computer.

ii. Remove the board.

iii. Change the required jumpers settings.

iv. Replace the board in the PC.

v. Power up, and run a program (such as DIAG), which executes the PC-30 diagnostics routines.

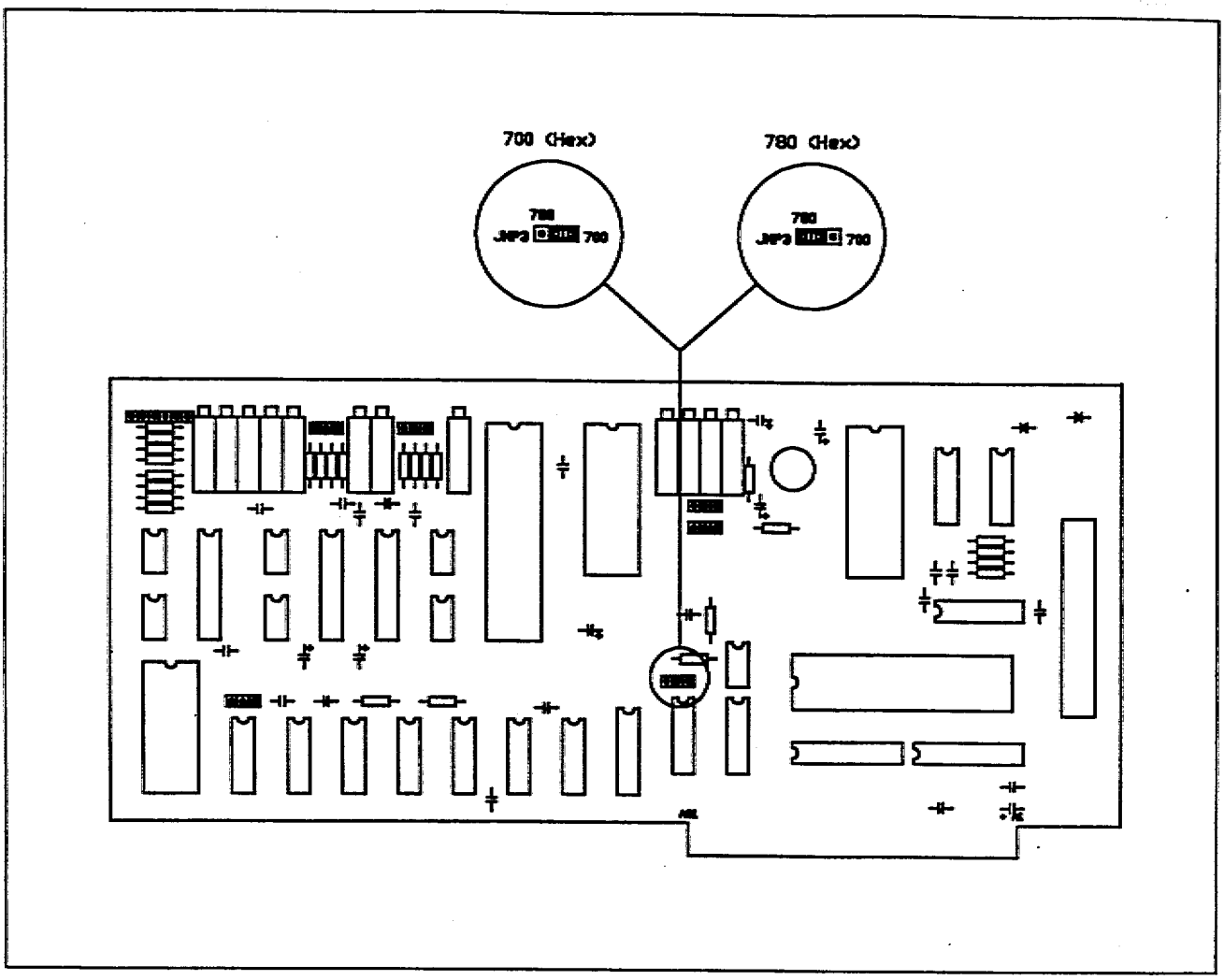


Figure 3.2. PC-30 base address jumper settings.

3.3. Bus Interface Configuration.

3.3.1. Base address

The base address setting is controlled by JMP3 and, in the case of the PC-39, by JMP10 as well. As supplied by the factory, the address is set to 700H. The board occupies 32 consecutive locations. Figure 3.1 shows the jumper selections for the PC-26, figure 3.2 those for the PC-30, and figure 3.3 those for the PC-39.

3.3.2. DMA level

3.3.2.1 PC-39

On the PC-39, the DMA setting block consists of jumper JMP9. The DMA level may be set to level

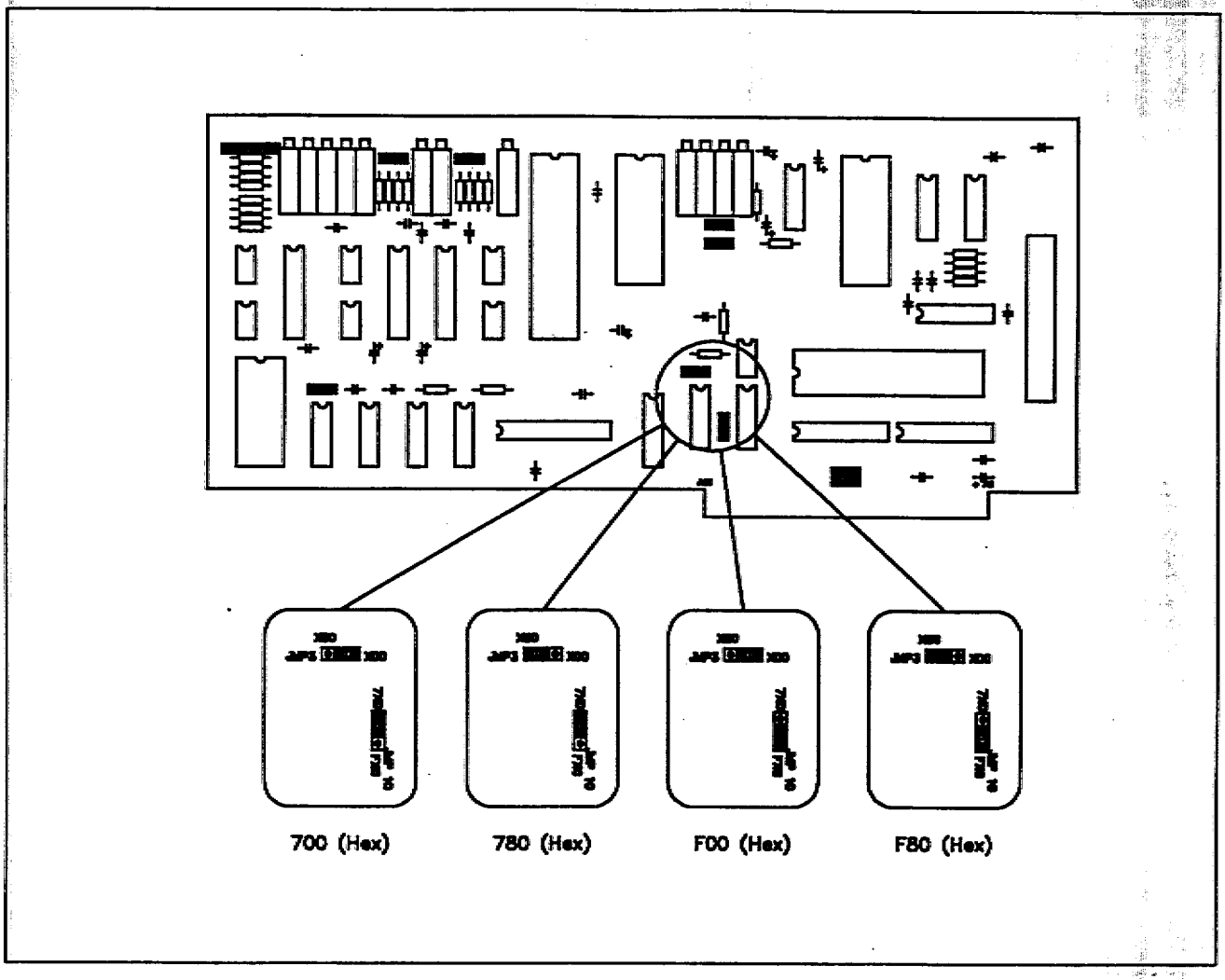


Figure 3.3. PC-39 base address jumper settings.

1 or 3. The DMA jumper settings for the PC-39 are shown in figure 3.4.

Warning
Only one set of the jumpers in the above block may be installed at any one time.

3.3.2.2 Selection of DMA levels

In a standard PC, DMA channels are allocated as follows :

level 1	Unused
level 3	Used by fixed disks (XT and AT)

A DMA level which is not already used must be selected. As level 1 is least often used, this is normally selected for PC-39s. Note that unless DMA is specifically enabled by software, the DMA outputs from the board are tri-stated (does not have any effect on the PC bus).

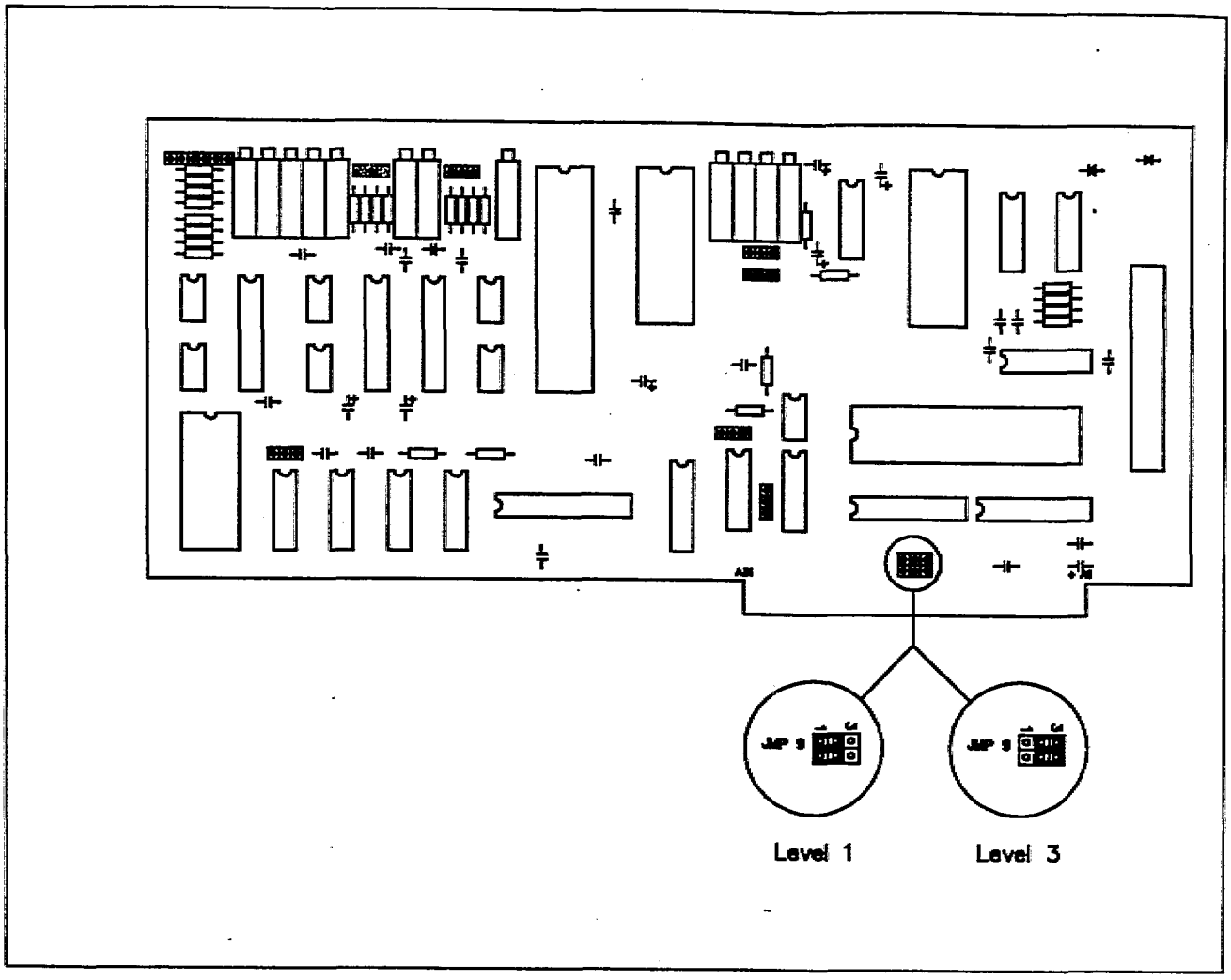


Figure 3.4. PC-39 DMA level jumper settings.

3.4. D/A jumper selections.

The four D/A converters may be jumpered for either monopolar (0 to 10V) or bipolar (-10 to +10V) output ranges. The D/A jumpers are numbered from JMP4 to JMP7, as follows :

DAC0 (12-bit)	JMP4
DAC1 (12-bit)	JMP5
DAC2 (8-bit)	JMP6
DAC3 (8-bit)	JMP7

Jumper settings are shown in figure 3.5.

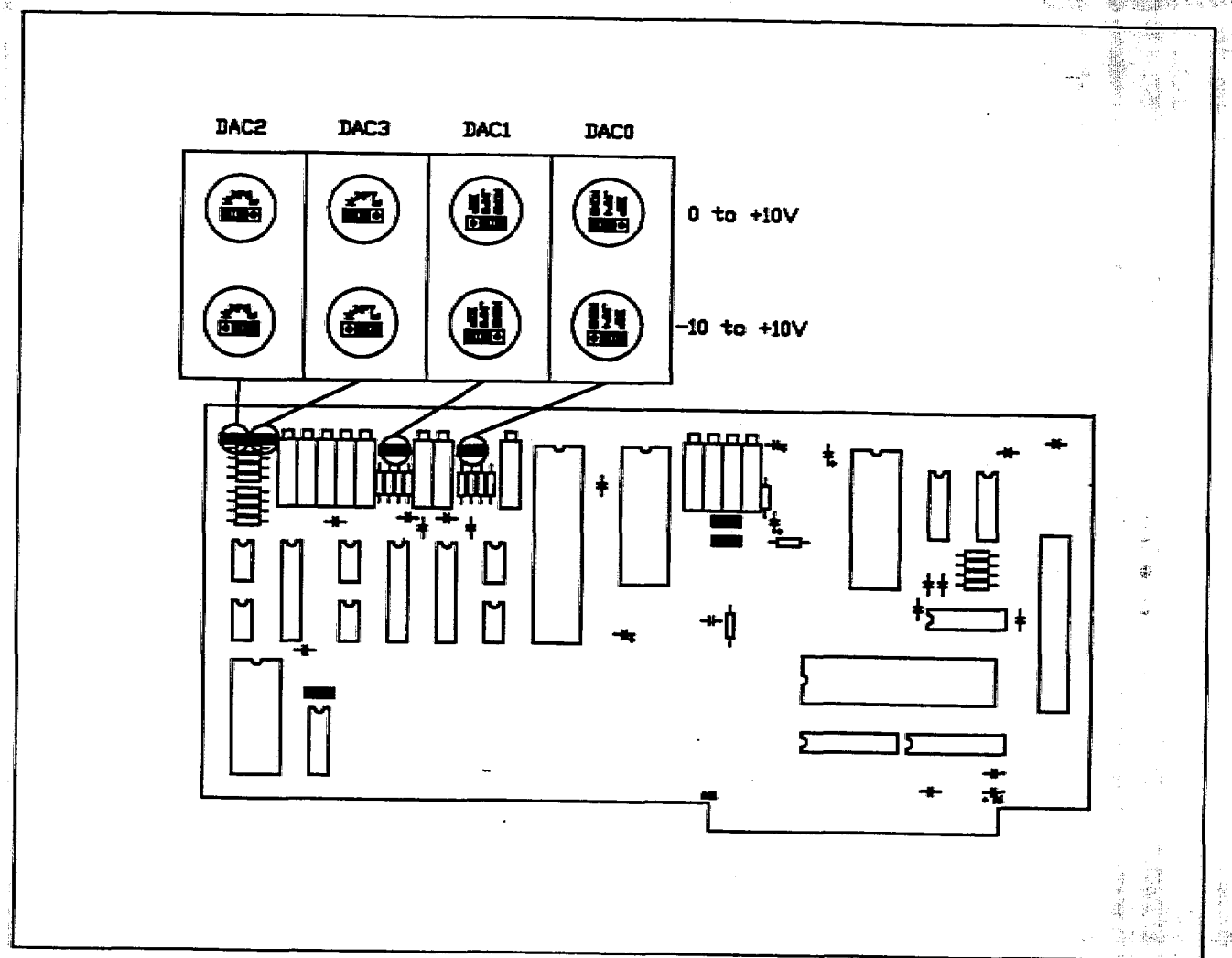


Figure 3.5. PC-30 and PC-39 D/A jumper settings.

3.5. A/D configuration.

3.5.1. A/D voltage range jumper settings

The A/D may be configured for input range. This is controlled by JMP1 and JMP2. The A/D jumper settings for the PC-26 are shown in figure 3.6, and those for the PC-30 and PC-39 in figure 3.7.

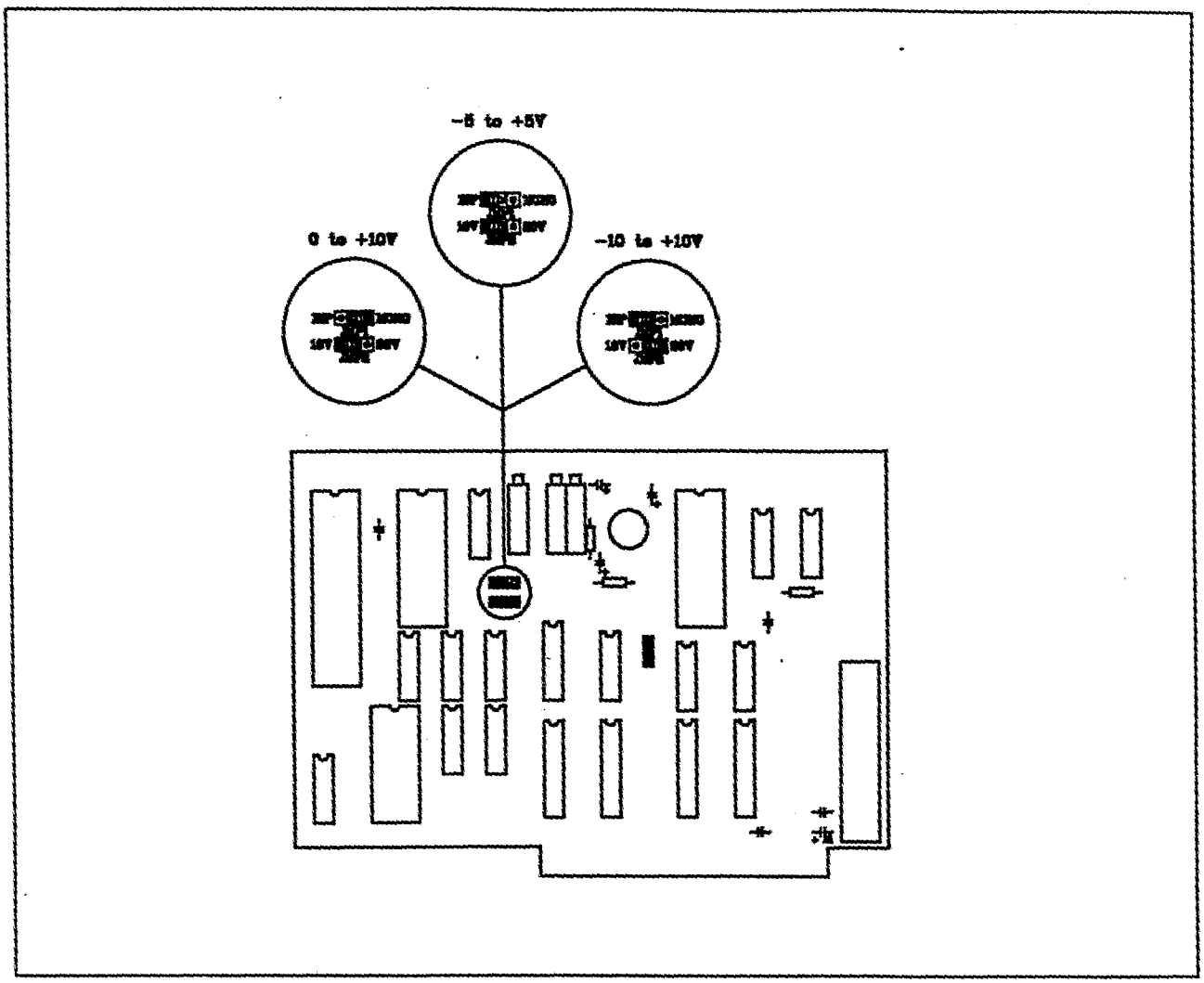


Figure 3.6. PC-26 A/D jumper settings

3.6. Counter/timer (PC-30/PC-39 only).

Figure 2.2 shows a partial schematic of the PC-30 clock/trigger system. Counter 0 of the 8253 serves as the A/D clock prescaler, and counter 1 as the A/D clock divider. Counter 2 is the uncommitted counter/timer. The output of this counter is available on pin 21 of J1.

Note
This counter is unused on the PC-26.

The input to the counter can be obtained from one of two sources :

i. Prescaler. In this case the input to the counter is obtained from the A/D prescaler.

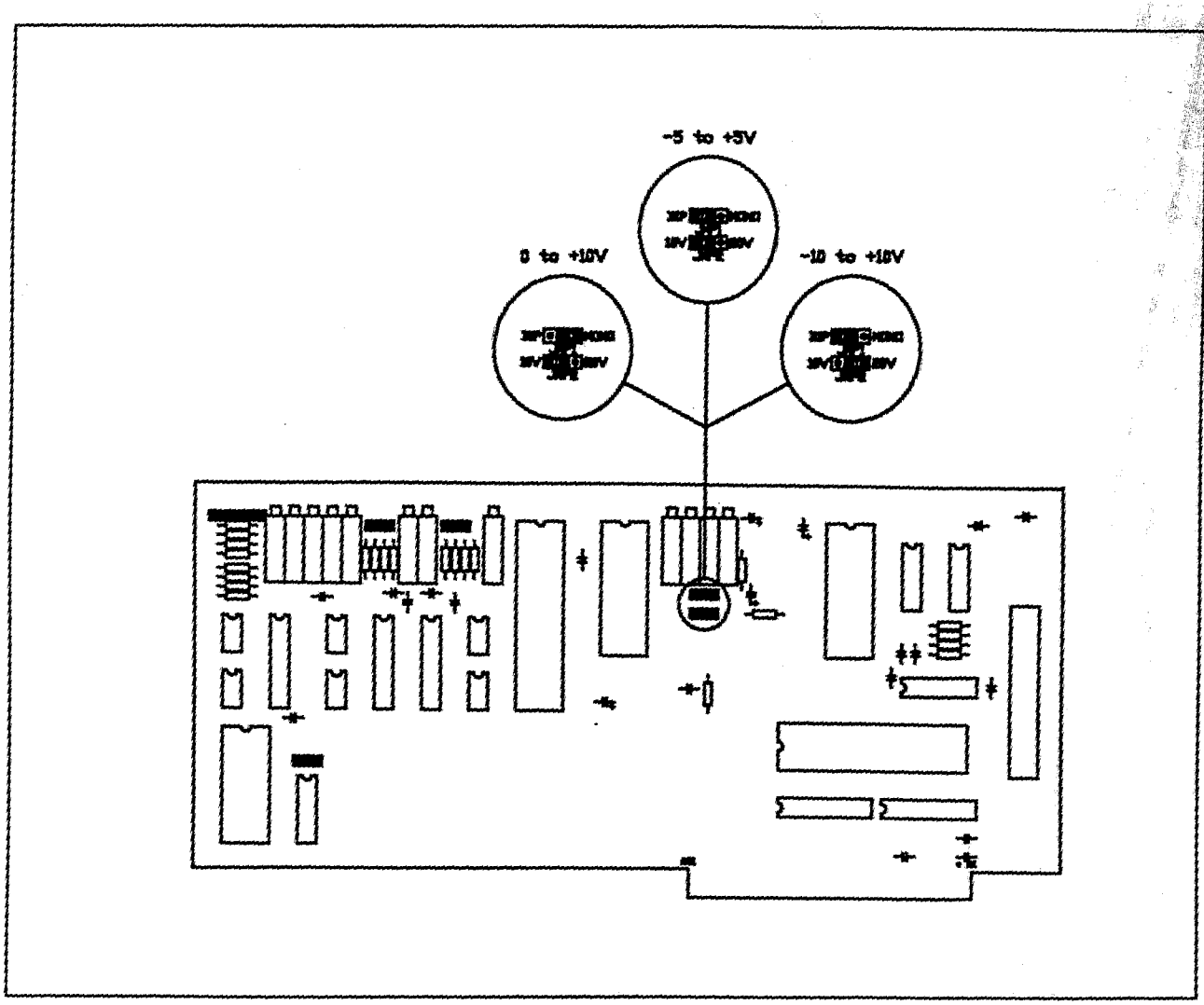


Figure 3.7. PC-30 and PC-39 A/D jumper settings

ii. Bus clock. In this case the input to the counter is obtained from the PC's bus clock divided by four.

The clock source is controlled by jumper JMP8. Settings are shown in figure 3.8.

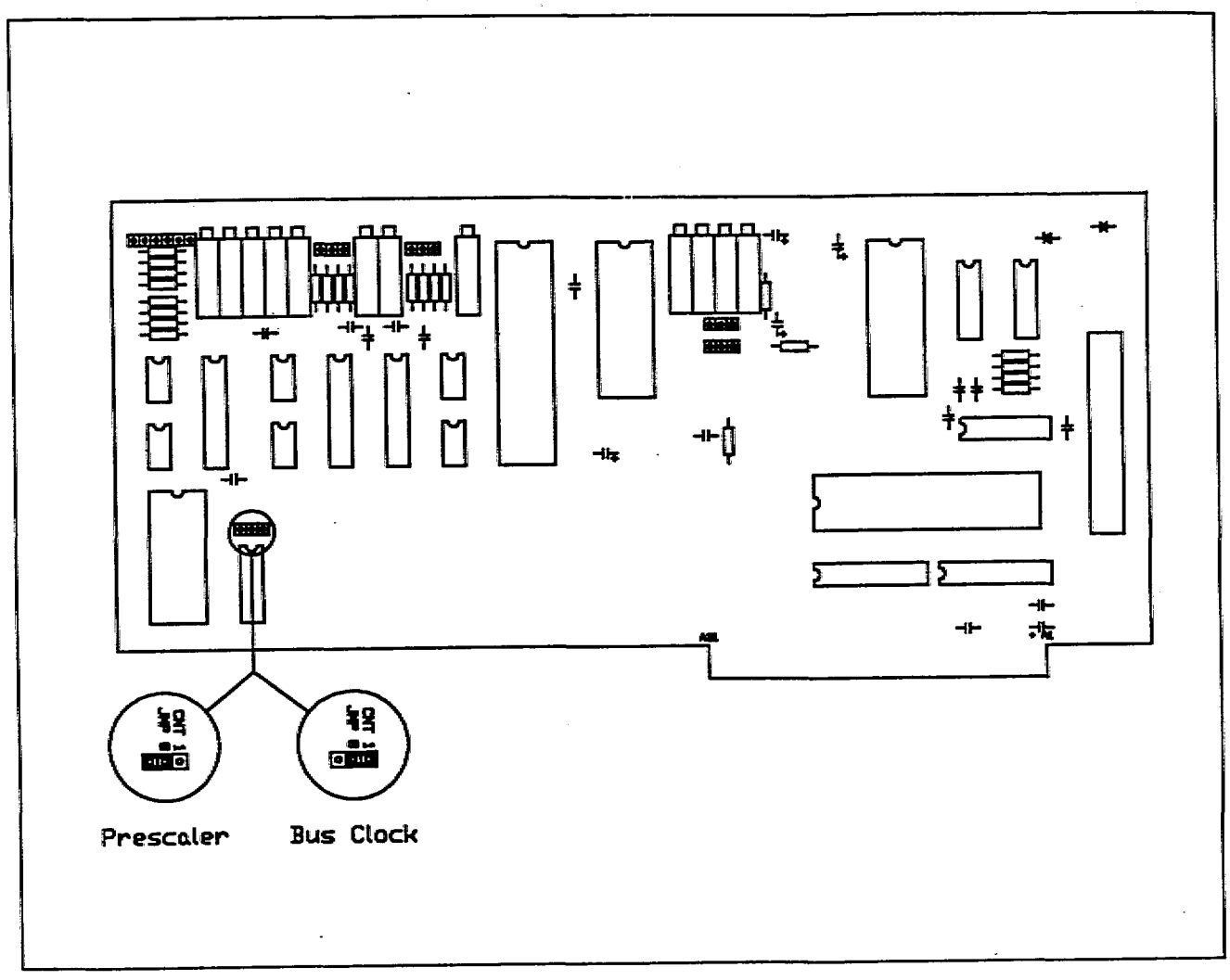


Figure 3.7. PC-30 and PC-39 Timer jumper settings

[This page intentionally left blank.]

Chapter 4

Interconnections

4.1. Introduction

The PC-26, PC-30 and PC-39 plug into IBM PC/XT/AT or compatible expansion slots at connector P1. All boards in the family connect to the user's circuitry at connector J1. This chapter describes these two connectors.

4.2. Connections to the IBM backplane.

PC-26, PC-30 and PC-39 boards can be plugged into any slot of the IBM backplane, with the exception of the J8 slot of the XT. This particular slot requires the -CARDSLCT signal, which is not used on other slots. All communication to and from the host processor is carried out via this connector.

4.3. User connection.

4.3.1. PC-26

The PC-26 is connected to the user interface via a male 25 way D-type connector. This connector accommodates the following signals :

- 16 single ended lines of analog input.
- External trigger.

J1 also provides + and -12V power supply, with limited current output.

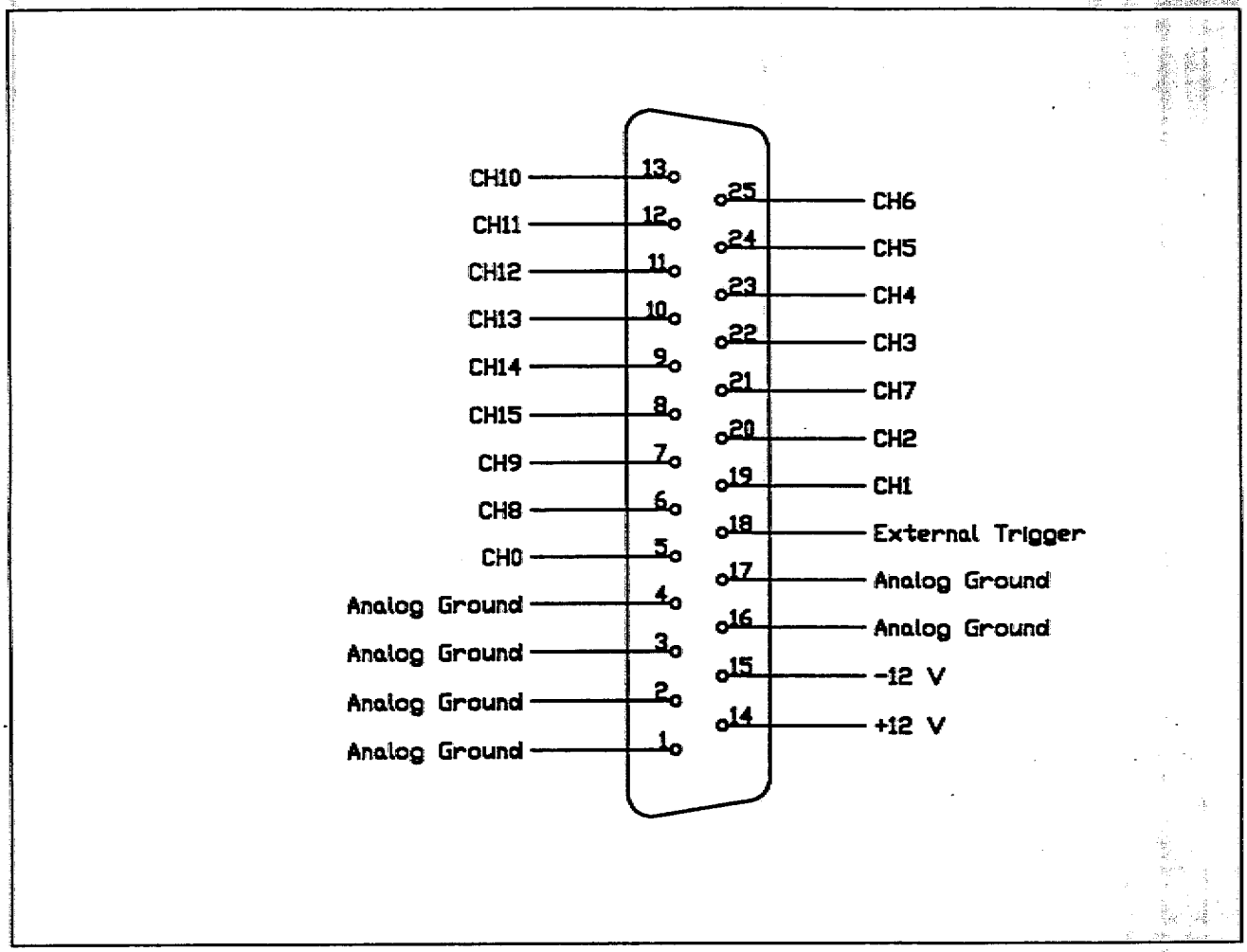


Figure 4.1. PC-26 connector, as seen from the rear of the PC.

Figure 4.2. shows these connections, together with their pin assignments. Note that the pin connections refer to the pin DB25 pin numbers. These are embossed onto the connector itself. Do not make use of any numbers on the PC-26 board.

4.3.2. PC-30/PC-39

The PC-30 and PC-39 are connected to the user interface via a male 50 way D-type connector. This connector accommodates the following signals :

- 16 single ended lines of analog input.
- 4 lines of analog output.
- 24 digital I/O lines.
- External trigger and clock.

J1 also provides + and -12V power supply, with limited current output, as well as a +5V supply.

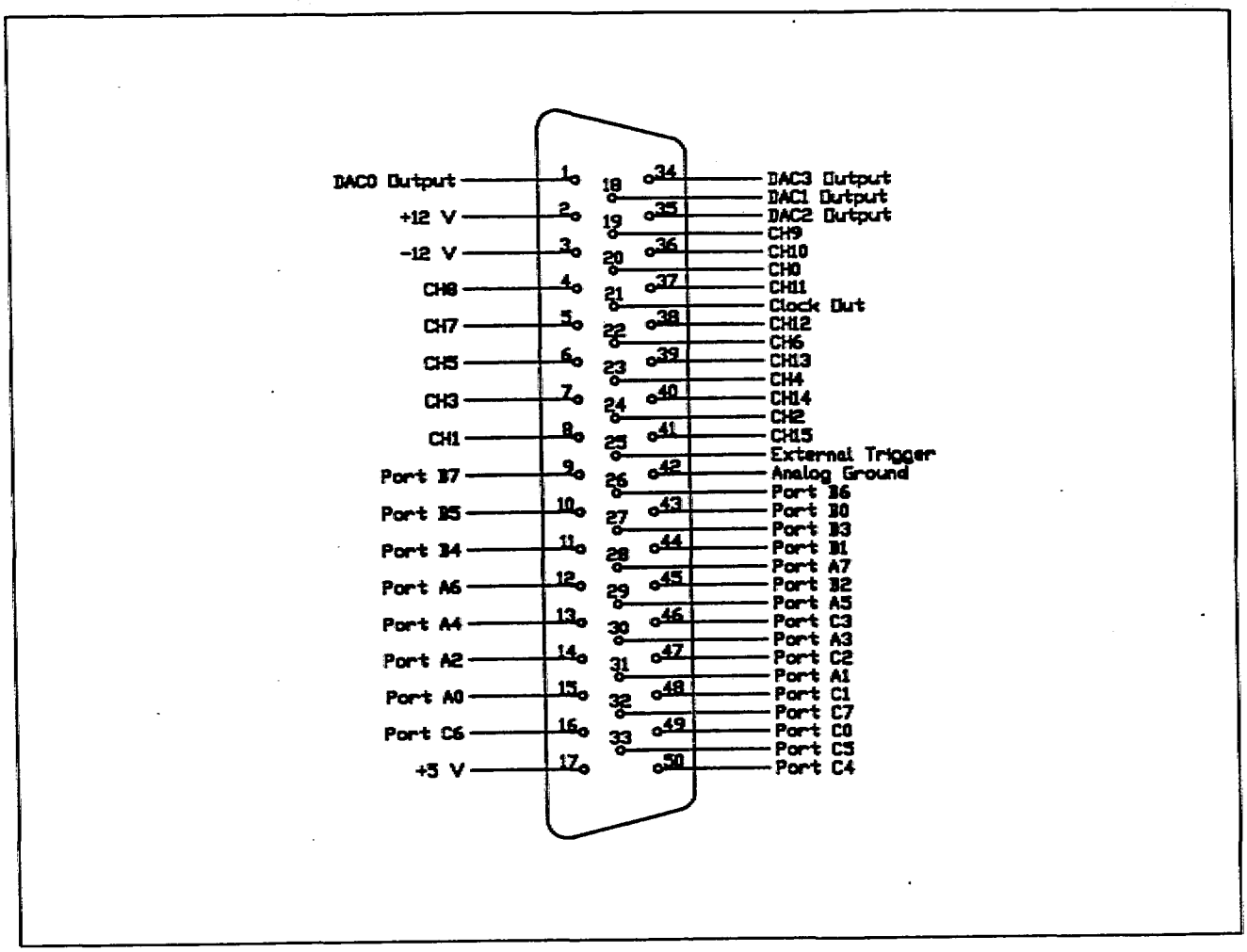


Figure 4.2. PC-30 connector, as seen from the rear of the PC.

Figure 4.2. shows these connections, together with their pin assignments. Note that the pin connections refer to the pin DB50 pin numbers. These are embossed onto the connector itself. Do not make use of any numbers on the PC-30 board.

4.3.3. Signal definitions.

i. CH0 - CH15. These are the analog input lines. Note that no more than +- 10V must be applied to these pins.

ii. ANALOG GROUND. One analog ground line is provided. The analog input lines are measured relative to AGND.

iii. DAC0 OUTPUT. This is the analog output line for DAC0.

iv. DAC1 OUTPUT. This is the analog output line for DAC1.

v. DAC2 OUTPUT. This is the analog output line for DAC2.

vi. DAC3 OUTPUT. This is the analog output line for DAC3.

vii. +12. This line provides a +12 V power supply to the user's interface. Maximum permissible current draw is 20 mA.

viii. -12. This line provides a -12 V power supply to the user's interface. Maximum permissible current draw is 20 mA.

ix. +5V. This line provides a -12 V power supply to the user's interface. Maximum permissible current draw is 20 mA.

x. Port A0 - A7. This is the first digital I/O port, digital I/O port 0. It is configurable into a number of operating modes under software control.

xi. Port B0 - B7. Digital I/O port 1. It is configurable into a number of operating modes under software control.

xii. Port C0 - C7. Digital I/O port 2. It is configurable into a number of operating modes under software control.

xiii. External Trigger. This line may be read under software control, and is TTL compatible.

xiv. Clock Out. This line is the output of the uncommitted counter/timer. It is TTL compatible.

4.4. Analog I/O

4.4.1. Recommended analog input schemes.

Analog signals are input into the PC-30 as single ended inputs. Single ended inputs are shown diagrammatically in figure 4.3.

Analog inputs are limited to a voltage of between -10 and +10 V.

Warning:
Overloading any analog input may cause other input channels to become inaccurate or noisy.

4.4.2. Analog output

The analog output lines are referenced the analog ground line. The analog output lines may be jumpered to give either monopolar or bipolar outputs.

4.4.3. Connection guidelines.

The PC-30 is a very high performance I/O subsystem, and was designed to have lower input noise levels than any similar analog I/O board currently available. Its performance may however be severely affected by incorrect connection techniques. This especially true of noise levels.

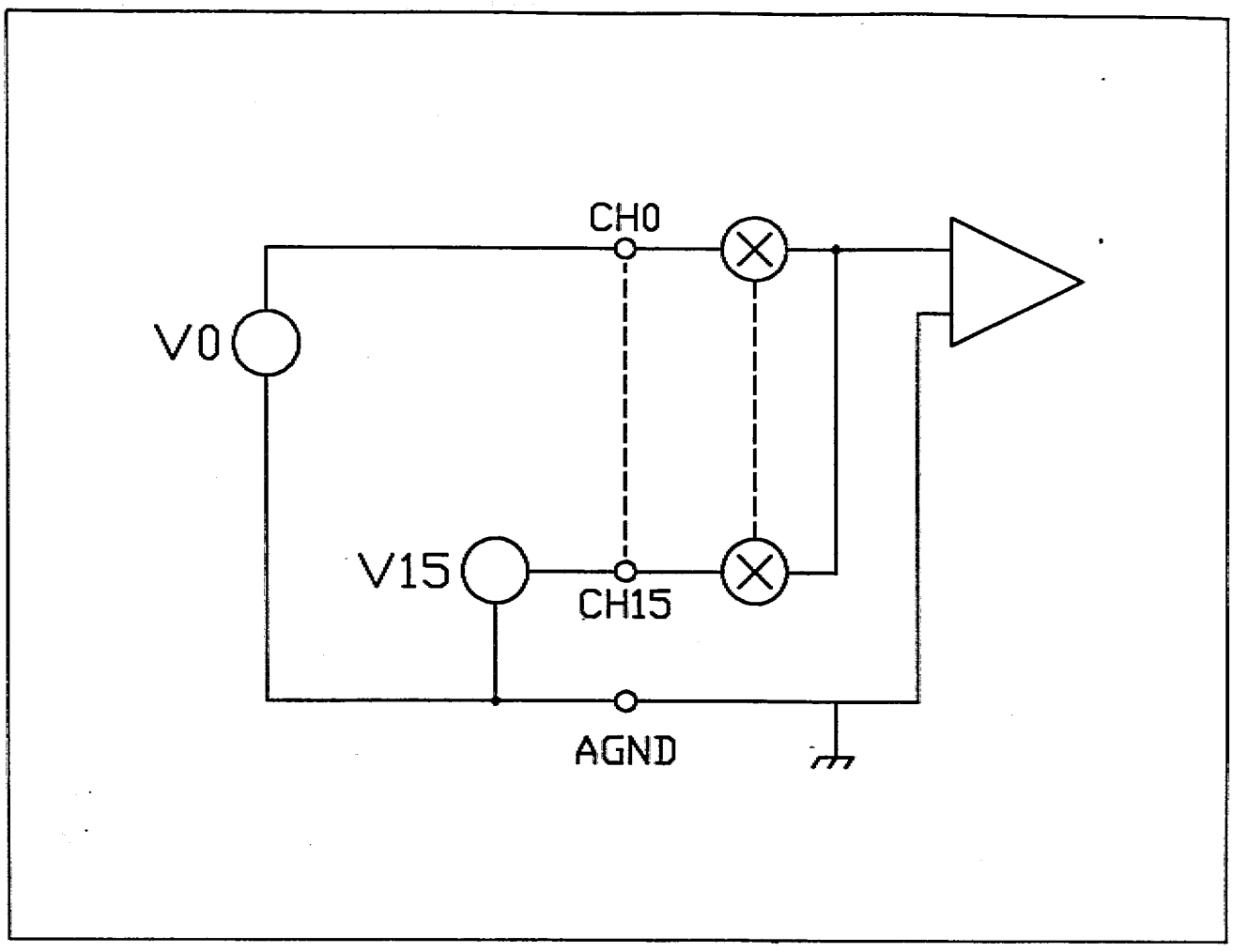


Figure 4.3. Single ended analog inputs.

4.4.3.1 Shielded input lines.

Wherever possible, leads should be shielded. Optimally, each input line should be individually shielded. The shield should be tied to analog ground at the instrument end of the connection only.

4.4.3.2 Input voltages.

To maintain stated accuracy, all inputs to the PC-30 must be within +- 10 V.

4.4.3.3 Source Impedance.

To maintain stated accuracy, all devices connected to the analog inputs of the PC-26 or PC-30 must have a source impedance of less than 2.5 Ohm, and those connected to a PC-39 less than 1 KOhm.

4.5. Digital I/O.

The digital I/O section of the PC-30 consists of three 8-bit ports (port A, B and C), which can be

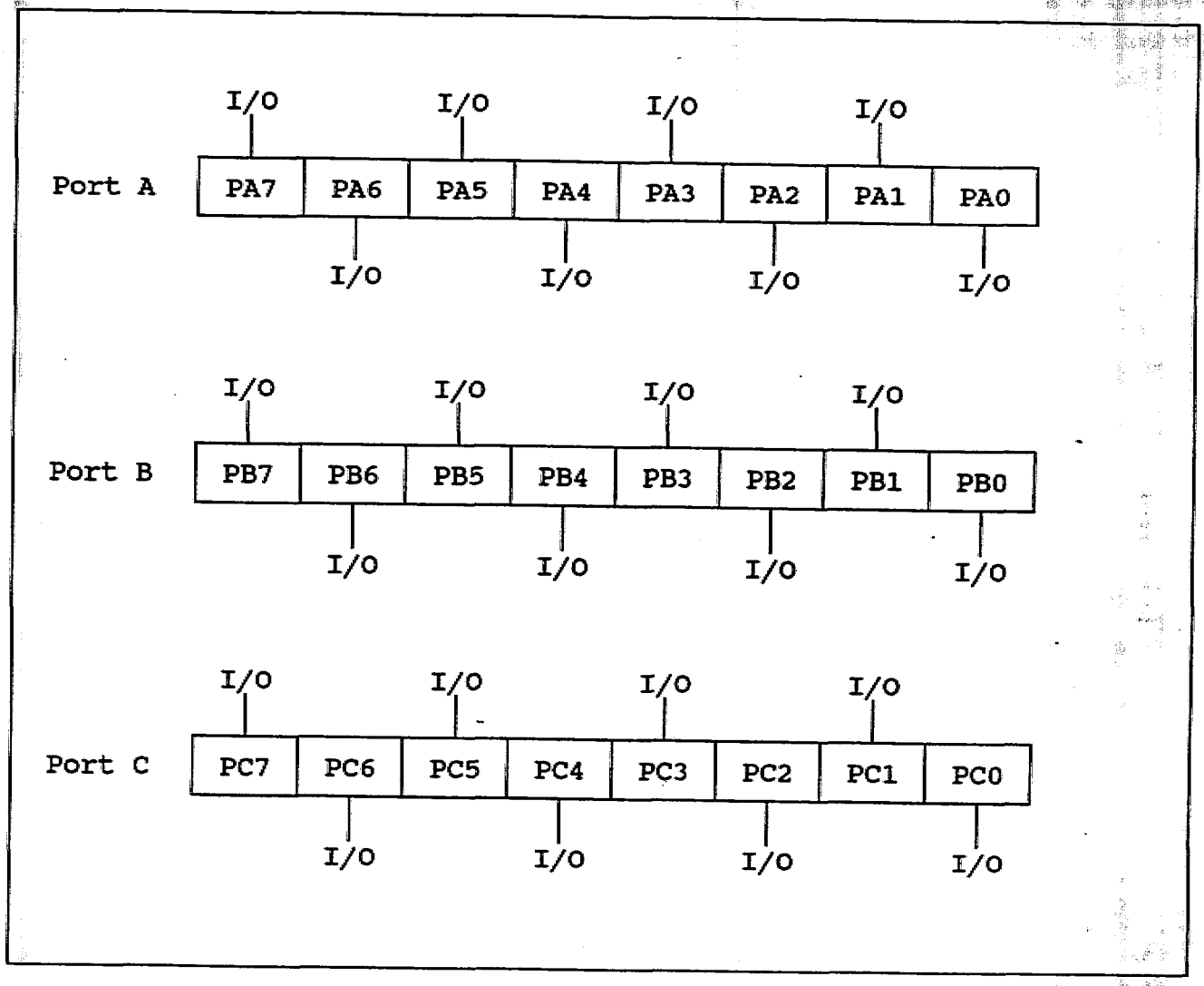


Figure 4.4. Mode 0 digital I/O.

configured in a variety of operating modes. The digital I/O portion of the PC-30 emulates, and is 100% compatible with, an 8255 type PPI (Programmable Peripheral Interface).

The three ports are divided into two groups, group A (consisting of port A and the upper half of port C) and group B (consisting of port B and the lower half of port C). Each of these groups can be individually configured into one of three operating modes.

4.5.1. Operating modes.

The three basic modes of operation are as follows:

- Mode 0 - Basic I/O.
- Mode 1 - Strobed I/O.

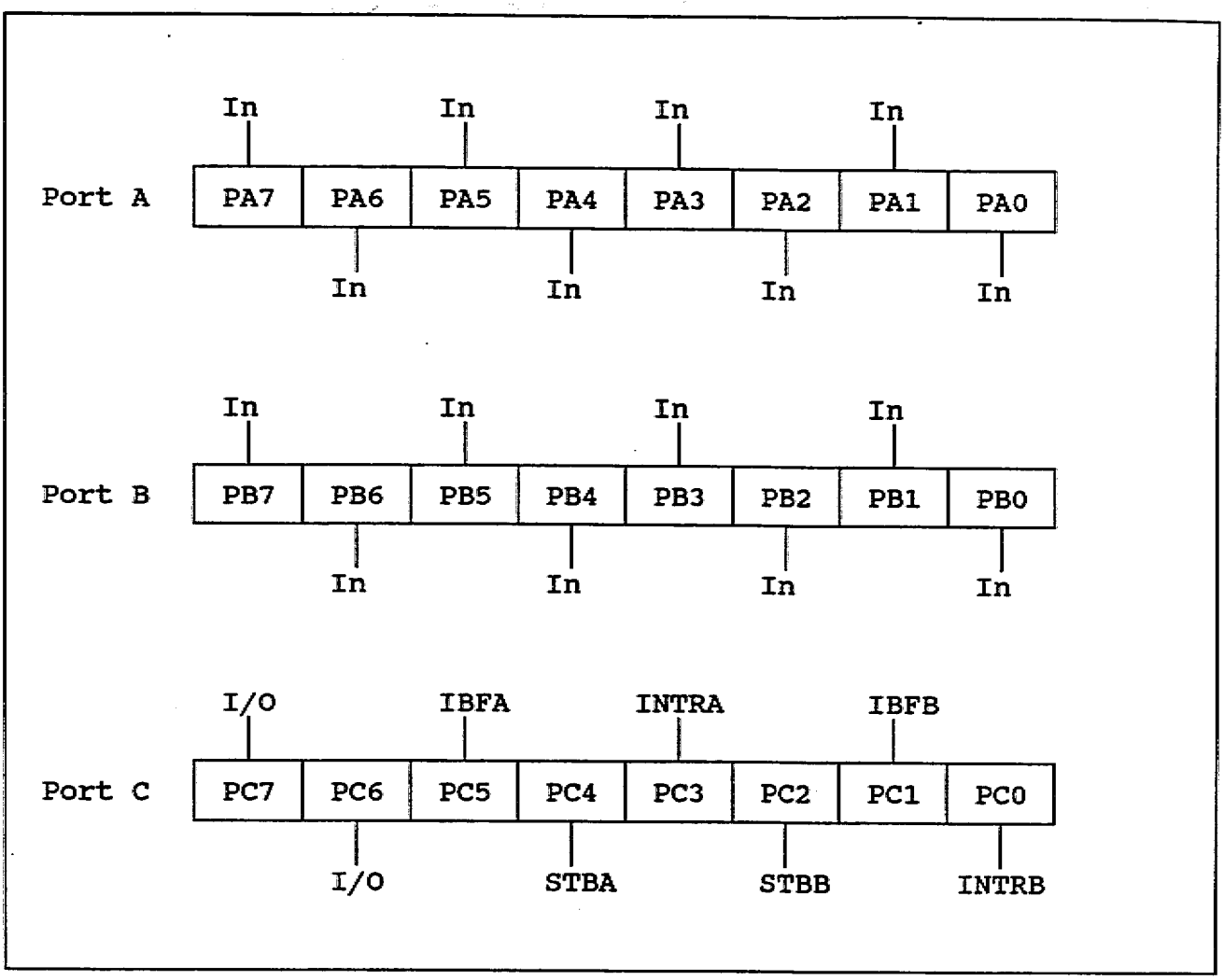


Figure 4.5. Mode 1 digital input.

- Mode 2 - Bidirectional bus operations.

At power up, the interface is set to mode 0. Each mode will be described in detail in the next sections.

For all three modes, various functions are assigned to the assorted I/O lines. These assignments are described in the next sections.

4.5.2. Mode 0 (Basic Input/Output).

Mode 0 characteristics are as follows:

- Two 8-bit ports (A and B) and two 4-bit ports (upper and lower halves of port C).
- Any port can either operate as an input or an output.

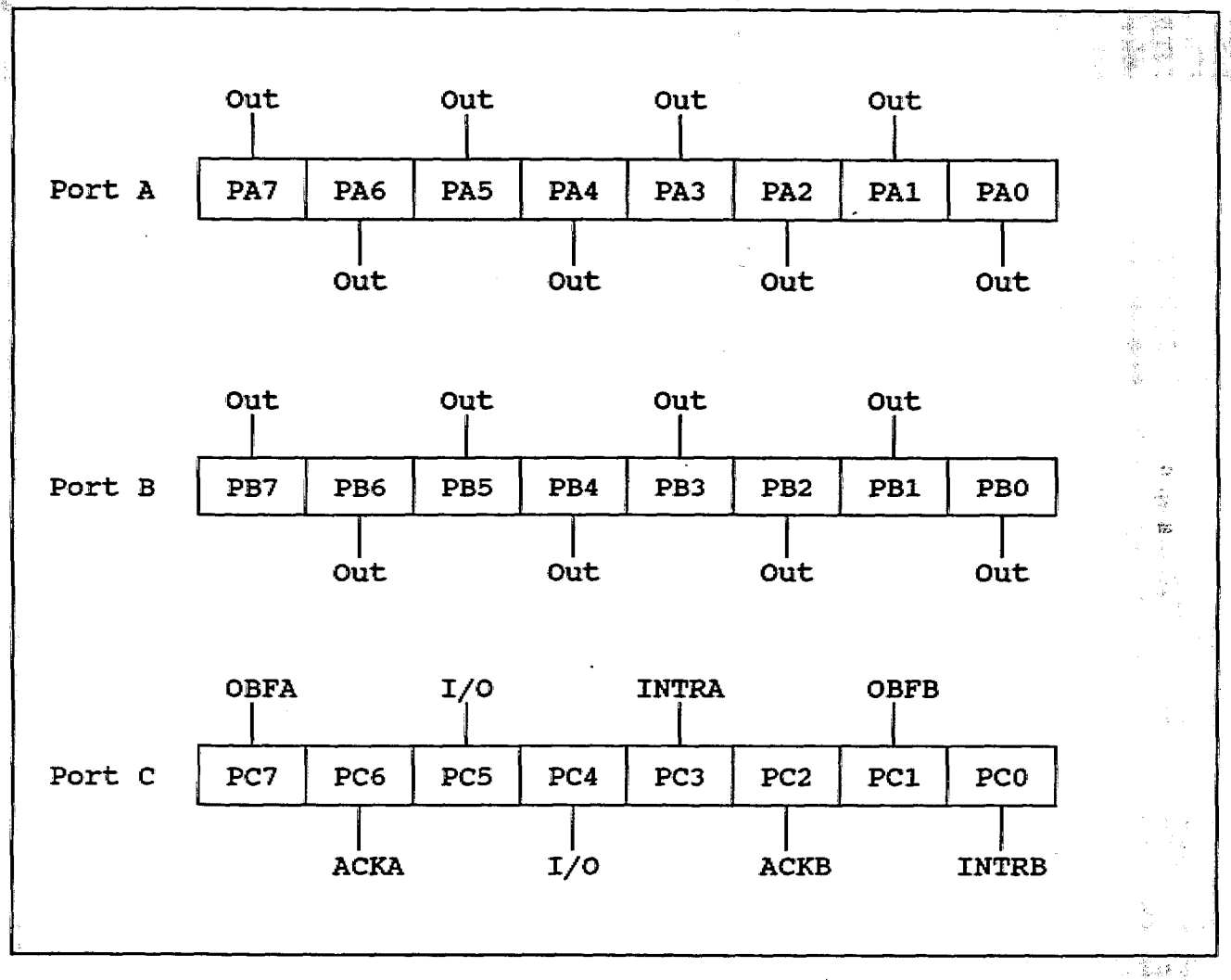


Figure 4.6. Mode 1 digital output.

Bit definitions for mode 0 are shown in figure 4.4.

This mode of operation provides simple I/O operations. Ports defined as inputs when read reflect the digital inputs on the port. Port defined as outputs are set to the value most recently written to the port. Any port can be used either for input or output, but not for both.

4.5.3. Mode 1 (Strobed Input/Output).

Mode 1 characteristics are as follows:

- Two groups, each consisting of one 8-bit and one 4-bit port.
- The 4-bit port is used for control and status of the 8-bit port.
- The 8-bit port may be used for either input or output.

- Input and output operations are latched.

This mode of operation provides I/O operations with a simple handshake protocol. The assignment of handshake signals to port C is shown in figure 4.5 and 4.6. The handshake signals are as follows:

4.5.3.1 Handshake Signals

i. Input operations

a) STB (Strobe input). A low on this input loads data into the input buffer. The data can then be read by the program.

b) IBF (Input Buffer Full). A high on this output indicates that there is data in the input buffer. This can be used as either a acknowledgment or buffer full signal. The output is reset when the CPU reads the data.

c) INTR (Interrupt). This output is set high if STB is low, IBF is high, and the INTE bit of the PC-30 internal register is set.

ii. Output operations

a) OBF (Output Buffer Full). A low on this output indicates that the CPU has written data to the port. It is set high by the ACK input going low.

b) ACK (Acknowledge input). A low on this input indicates to the PC-30 that the data has been accepted by the external circuitry.

c) INTR (Interrupt). This output is set high if ACK is high, OBF is high, and the INTE bit of the PC-30 internal register is set.

d) Bits of port C not used for handshake lines can be used for simple (mode 0) operations.

4.5.4. Mode 2 (Strobed Bidirectional Input/Output).

Mode 2 characteristics are as follows:

- One 8-bit bidirectional port, and one 5 bit control port.
- Can be used in group A only.
- The 5-bit port is used for control and status of the 8-bit port.
- Input and output operations are latched.
- Port B can still be used in mode 0 or 1.

This mode of operation provides a means for bidirectional I/O operations on port A. The assignment of handshake signals to port C is shown in figure 4.7. The handshake signals are as follows:

4.5.4.1 Handshake Signals

i. OBF (Output Buffer Full). A low on this output indicates that the CPU has written data to the port. It is set high by the ACK input going low.

ii. ACK (Acknowledge input). A low on this input enables the port A outputs, allowing

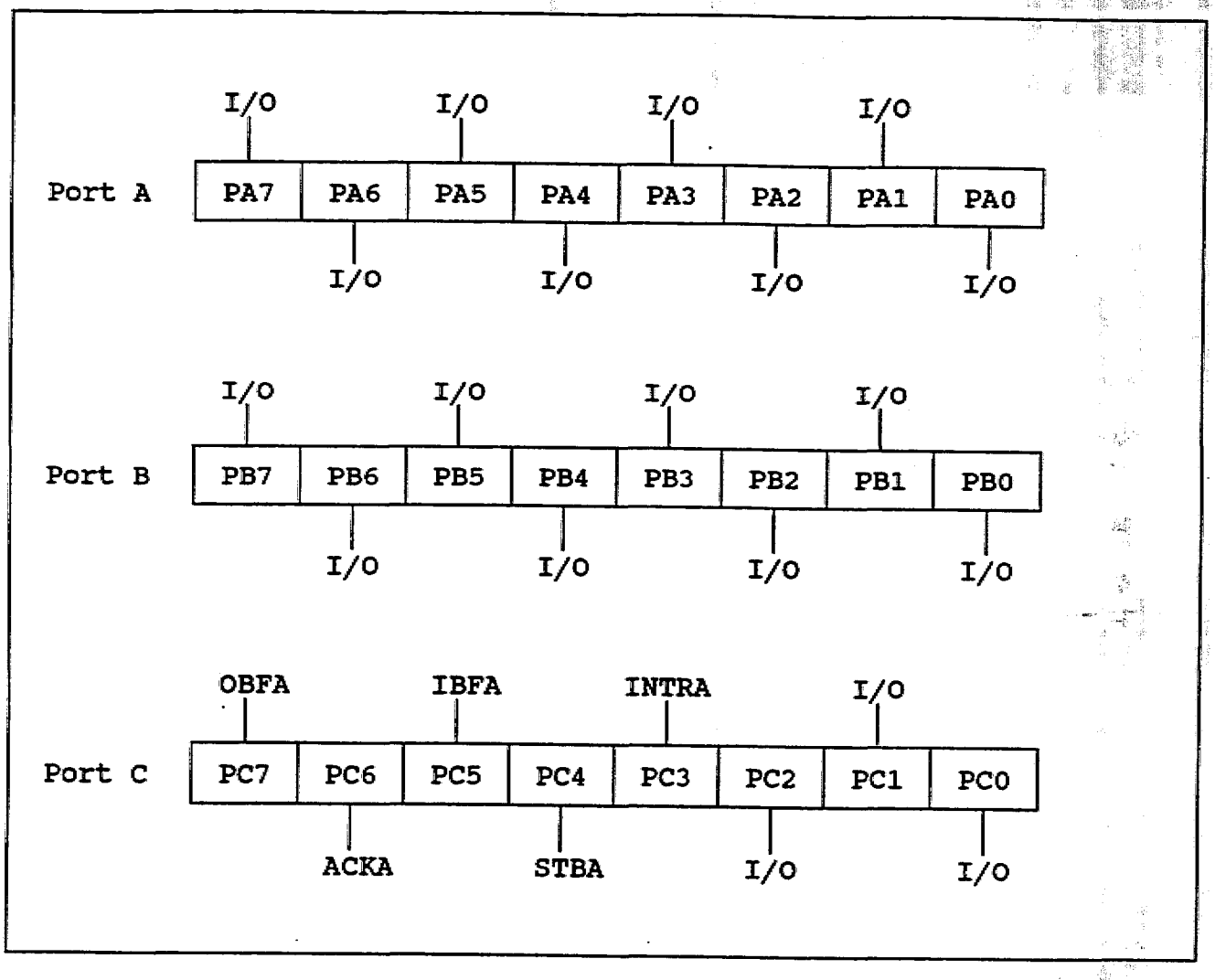


Figure 4.7. Mode 2 Digital I/O.

external circuitry to read the value written to the port. If this output is high, port A is in the input mode.

iii. STB (Strobe input). A low on this input loads data into the input buffer. It can then be read by the program.

iv. IBF (Input Buffer Full). A high on this output indicates that there is data in the input buffer. This can be used as either a acknowledgment or buffer full signal. The output is reset when the CPU reads the data.

v. INTR (Interrupt). This output is set high under either of two conditions:

a) If ACK is high, OBF is high, and the INTE1 bit of the PC-30 internal register is set.

b) If STB is low, IBF is high, and the INTE2 bit of the PC-30 internal register is set.

Bits of port C not used for handshake lines can be used for simple (mode 0) I/O operations.

4.5.5. Mode combination considerations.

It is possible to configure the parallel interface of the PC-30 in several different modes which leave some bits of port C unused for control or status. These bits can be used as follows:

i. If programmed as inputs, these bits can be accessed as usual by port read commands.

ii. If programmed as outputs, the bits can be written by the bit set reset functions described in chapter 5.

[This page intentionally left blank.]

Chapter 5

Register structure

5.1. Introduction

At the lowest level, the PC-30 can be programmed using I/O input and output instructions. This chapter contains the information required to do this. Although not difficult, this is time consuming, and requires detailed knowledge of the PC-30, as well as the operation of the host PC and its operating system. In order to simplify this process, a set of device drivers is available for use with the board. The use of these allows access to all board functions. These drivers are described in the "User Manual for PC-30 Driver Software" manual.

The next chapter discusses various programming techniques and tips.

5.2. Register structure.

The PC-30 uses 32 consecutive address locations in I/O space. The layout of these registers is shown in figure 5.1. Note that certain addresses have different read and write register functions.

Note also that the addresses above are given as offsets from the base address of the board. This base address is jumper selected as described in chapter 3.

Each register will now be described in detail.

Warning
You should not write to, or read from, unused registers. All unused registers are reserved for manufacturing test, or for future developments.

Offset From Base	Register Name	
	Read	Write
0	A/D Low Byte (ADDATL)	——
1	A/D Data/Status(ADDSR)	——
2	Control/Channel (ADCCR)	
3	——	Mode Register (ADMDE)
4	——	Prescaler (PRESCALER)
5	——	Divider (DIVIDER)
6	——	Counter/Timer (USR_CNT)
7	——	Counter Control(TMRCTR)
8	Digital I/O port A (DIOP0)	
9	Digital I/O port B (DIOP1)	
10	Digital I/O port C (DIOP2)	
11	——	Dig. Control (DIOCNTRL)
12	——	DAC0 Low Byte (DADATL0)
13	——	DAC0 High Byte (DADATH0)
16	——	DAC1 Low Byte (DADATL1)
17	——	DAC1 High Byte (DADATH1)
20	——	DAC2 Data (DADAT2)
21	——	DAC3 Data (DADAT3)

Figure 5.1. PC-30 Register Structure.

5.2.1. ADDATL - A/D data low byte (offset 0) (read only)

On completion of an A/D conversion, the A/D converter loads this register with digital data. To retrieve converted data, the user must perform a read operation to ADDATL. Bit 0 is the LSB. The layout of this register is shown in figure 5.2

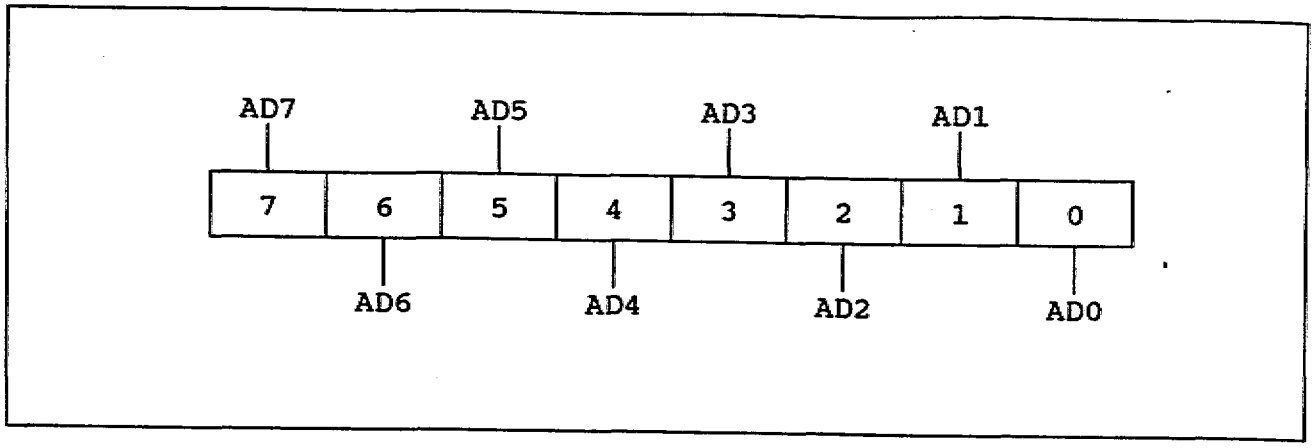


Figure 5.2. A/D data register (low byte).

Bits 7-0 - A/D data (AD)

These bits are the low byte of the 12-bit code which is returned from an A/D conversion.

5.2.2. ADDSR - A/D data/status register (offset 1)

The ADDSR contains the high nibble of the A/D result, and contains the current A/D's status information. The bit functions of the ADDSR are shown in figure 5.3.

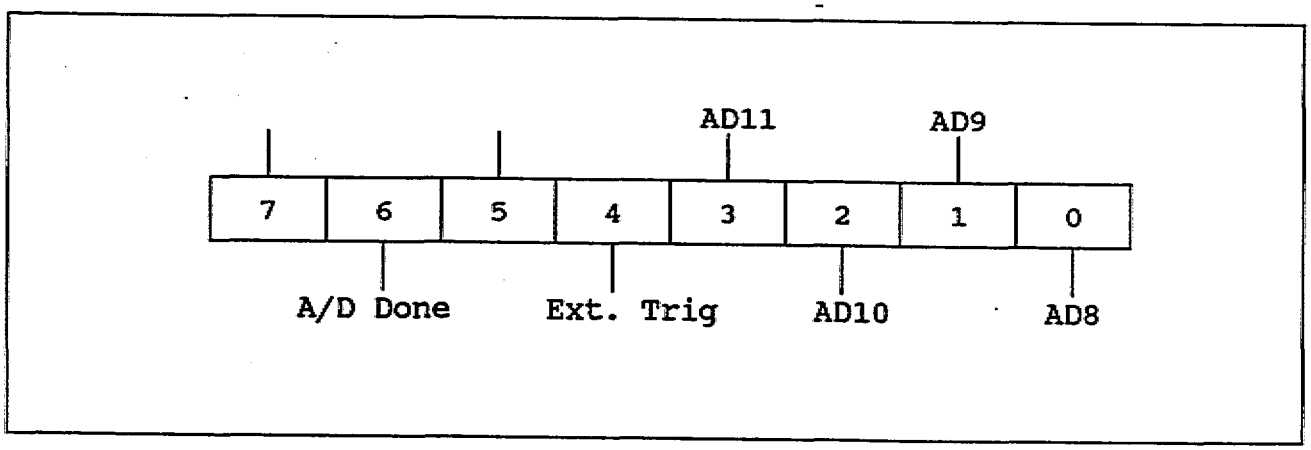


Figure 5.3. A/D data/status register.

Bit 7 - Reserved

Bit 6 - A/D done (PC-39 only)

Set by the A/D converter to indicate that A/D data is available. If bit 3 of the ADCCR is set (interrupts are enabled), then an interrupt will be generated when bit 6 is set. If bit 2 of the ADCCR is set (DMA is enabled), then a DMA cycle will be generated when bit 6 is set.

In PC-39s, bit 6 is cleared on power up, and by a read of the ADDATL.

This bit is not present on PC-26 or PC-30 boards.

Bit 5 - Reserved

The result of reading this bit is undefined.

Bit 4 - Ext. Trig

This bit reflect the status of the external trigger pin.

Bits 3-0 - A/D data (AD)

The four higher bits of 12-bit data from an A/D conversion.

5.2.3. ADCCR - A/D Control/channel register (offset 2)

The ADCCR contains channel address bits, A/D clock control bits as well as DMA and interrupt enable bits. The bit functions of the ADCCR are shown in figure 5.4.

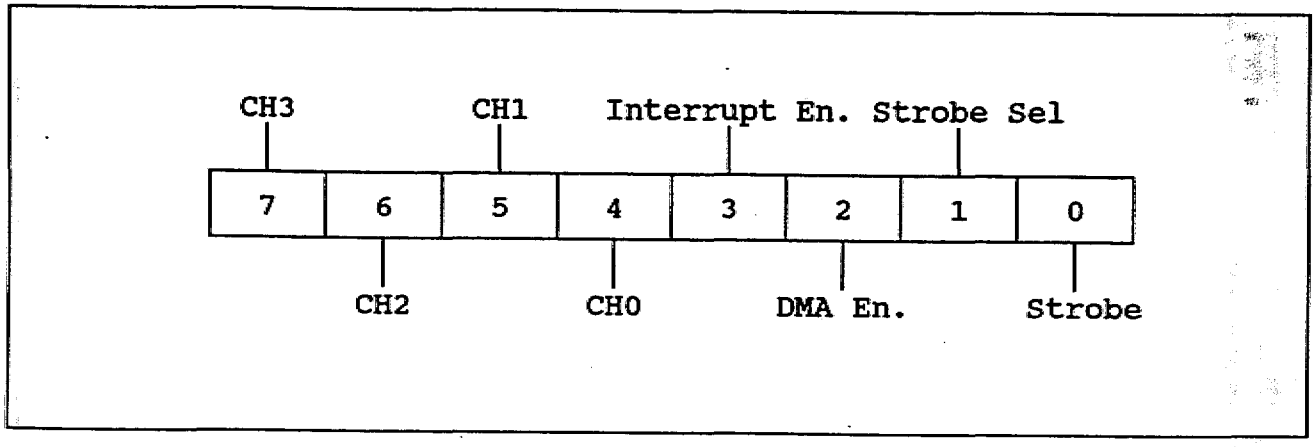


Figure 5.4. A/D control/channel register.

Bits 7-4 - Channel (CH)

These bits specify a four bit channel address. This channel is the channel which will be converted on the next A/D strobe.

Bit 3 - Interrupt enable

If the interrupt enable bit is set, then interrupts will be generated at the end of each A/D conversion.

This bit is controlled by the program in use. All interrupts are disabled when this bit is cleared. Bit 3 is cleared on power up.

Note
When interrupts are disabled, the PC-39 electronically disconnects the interrupt line from the PC's bus. This is not the case for the PC-26 or PC-30.

Bit 2 - DMA enable (PC-39 only).

This bit controls DMA operations. Setting the DMA enable bit enables DMA operation. If this bit is cleared all DMA is disabled, and the DMA request line to the PC bus is tri-stated. Bit 3 is cleared on power up.

This bit is not present on either the PC-26 or the PC-30.

Bit 1 - Strobe Select (STBC).

This bit controls the source of A/D strobes to the A/D converter. If it is set, then software strobes are used. Software strobes are generated by toggling the SSTB bit (bit 0). If the STBC bit is cleared, then A/D clock pulses are used to start A/D conversion cycles. A/D clock pulses are generated from the A/D clock prescaler/divider combination.

Bit 0 - Strobe (SSTB).

Bit 0 of the ADCCR is the software strobe bit. If the STBC bit is set, then a software strobe is generated by taking the strobe bit high, then low. If the STBC bit is low, then the SSTB bit is ignored.

Warning
Whenever you change the state of the STBC bit, the SSTB bit must be 0 (cleared). If the SSTB bit is not clear, spurious A/D conversion cycles may be generated.

5.2.4. ADMDE - A/D mode register (offset 3)

The ADMDE register contains A/D mode selection bits. The bit functions of the ADMDE register are shown in figure 5.5.

Note
The value 92 (hex) must be written to this register prior to any other activity.

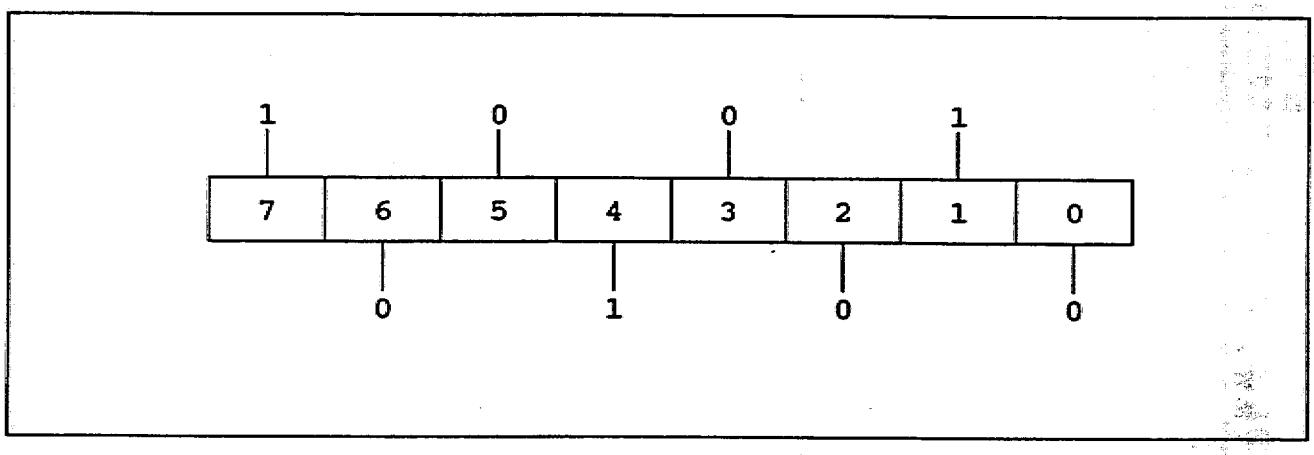


Figure 5.5. A/D mode register.

Bit 7 - Reserved (1)

For compatibility with future products, you must write a 1 to this bit. The results of reading this bit are undefined.

Bit 6 - Reserved (0)

For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

Bit 5 - Reserved (0)

For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

Bit 4 - Reserved (1)

For compatibility with future products, you must write a logical 1 to this bit. The results of reading this bit are undefined.

Bit 3 - Reserved (0)

For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

Bit 2 - Reserved (0)

For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

Bit 1 - Reserved (1)

For compatibility with future products, you must write a 1 to this bit. The results of reading this bit are undefined.

Bit 0 - Reserved (0)

For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

5.2.5. PRESCALER - A/D clock prescaler register (offset 4)

The register is used to program the A/D clock prescaler. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter. Binary mode is almost always used.

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

The input to the prescaler is always the PC's bus clock divided by four.

This register is register 0 (counter 0) of the 8253 on the PC-30 board. The bit layout of the prescaler is shown in figure 5.6.

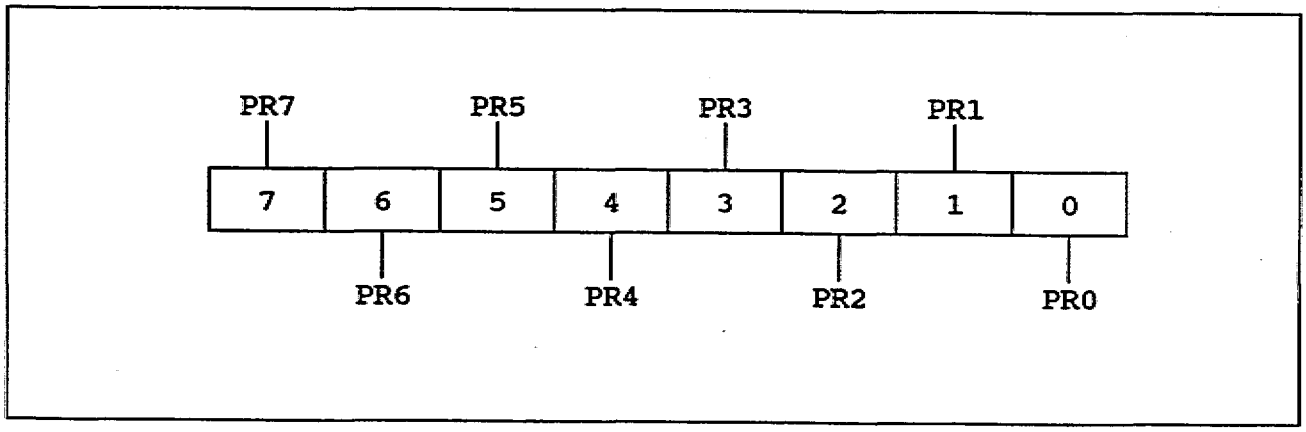


Figure 5.6. Prescaler register.

Bits 7-0 - Prescaler data (PR)

The prescaler register can be configured to write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to write both bytes. In this case the LSB is written first, then the MSB.

Note
It is important to always write two bytes to the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

5.2.6. DIVIDER - A/D clock divider register (offset 5)

The divider register is used to program the A/D clock divider. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter. Binary mode is almost always used.

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

The input to the clock divider is always the output from the A/D clock prescaler.

This register is register 1 (counter 1) of the 8253 on the PC-30 board. The bit layout of the divider is shown in figure 5.7.

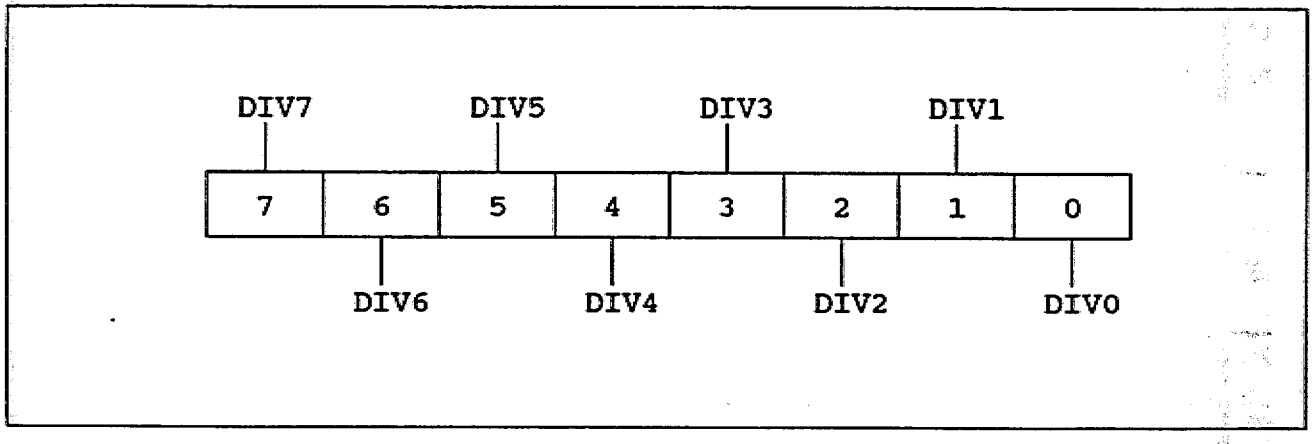


Figure 5.7. A/D clock divider register.

Bits 7-0 - A/D clock divider data (DIV)

The clock divider register can be configured to write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to write both bytes. In this case the LSB is written first, then the MSB.

Note:
It is important to always write two bytes to the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

5.2.7. USR_CNT - User counter register (offset 6)

The register is used to program the user configurable counter/timer. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter. Binary mode is almost always used.

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

This register is register 2 (counter 2) of the 8253 on the PC-30 board. The bit layout of the USR_CNT register is shown in figure 5.8.

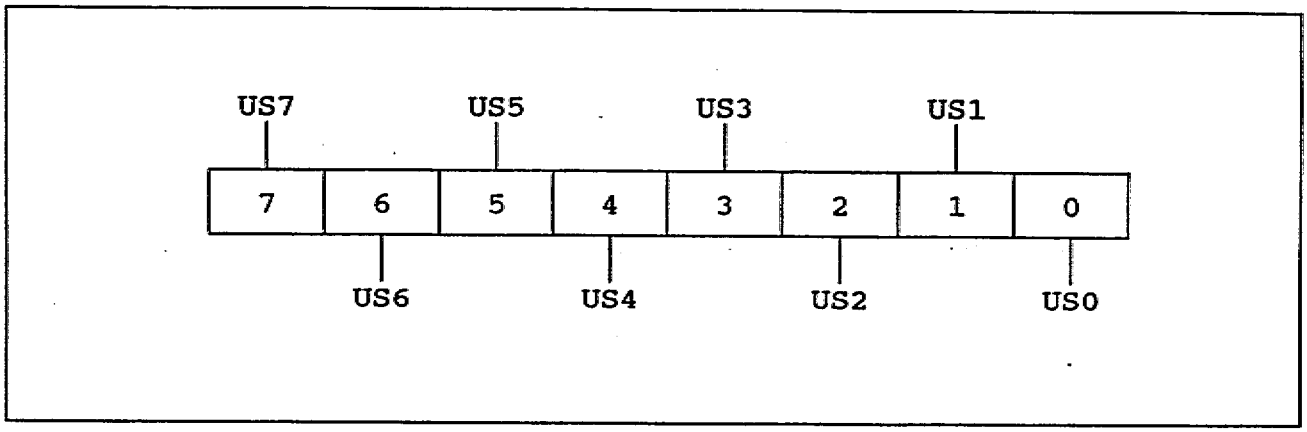


Figure 5.8. User counter register.

Bits 7-0 - User counter data (US)

The user counter register can be configured to write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to write both bytes. In this case the LSB is written first, then the MSB.

Note:
It is important to always write two bytes to the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

5.2.8. TMRCTR - Timer control register (offset 7)

The timer control register is used to configure the three 16-bit counters on the PC-30 board. It is

register 3 (Mode word) of the 8253 on the board. If you intend to program the 8253 extensively, you should obtain a copy of the data sheet for an 8253 type counter/timer.

The register layout is shown in figure 5.9.

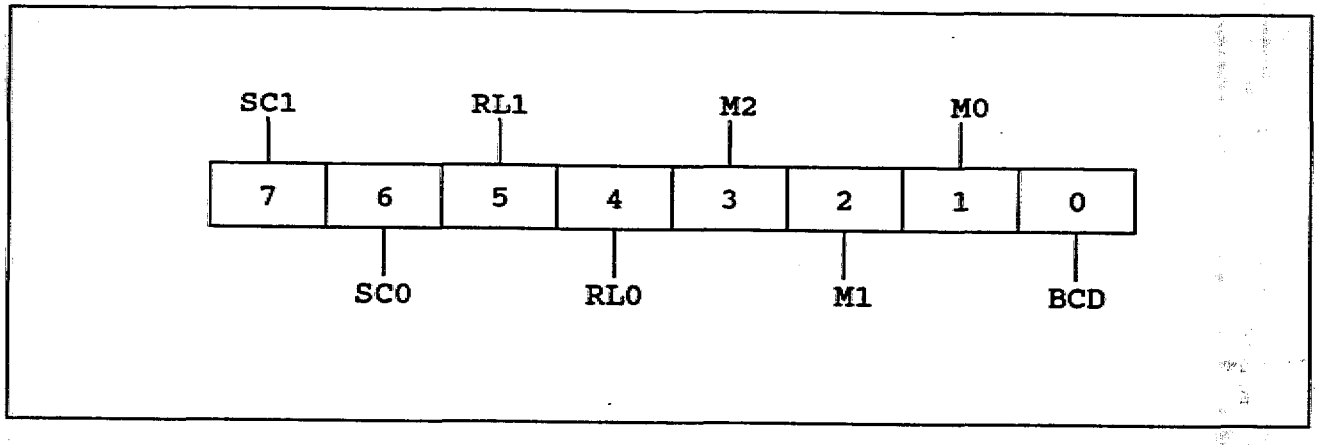


Figure 5.9. Timer controller register.

Bit 7-6 - Select Counter (SC)

These two bits are used to select the counter to be configured. They function as follows:

SC1	SC0	Function
0	0	Select counter 0 (A/D clock prescaler). The rest of the information in the byte written is used to configure the prescaler.
0	1	Select counter 1 (A/D clock divider). The rest of the information in the byte written is used to configure the divider.
1	0	Select counter 2 (User configured counter/timer). The rest of the information in the byte written is used to configure the user counter.
1	1	Reserved; Do not use.

Bits 5-4 - Read/load (RL)

Bits 4 and 5 are used to configure how the counter is read and written. The meaning of the various bit combinations is as follows:

RL1	RL0	Function

0	0	Counter latch. The selected counter is latched. This is used to read the contents of a counter while the counter in question's clock is active. If you intend to use this function you should consult the 8253 data sheet for full details.
0	1	Write LSB only. All write operations will write only the LSB (least significant byte) of the selected counter.
1	0	Write MSB only. All write operations will write only the MSB (most significant byte) of the selected counter.
1	1	Write LSB/MSB. Both the LSB and MSB of the counter are written. The LSB is written first, then the MSB. Note that both bytes must always be written. Writing only one byte will cause unpredictable results.

Bits 3-1 - Mode (M)

The counters can be programmed into various modes, as described below. The normal mode of operation for counters 0 and 1 (the A/D prescaler and divider) is mode 2. The mode for counter 2 (the user configurable counter) depend on its intended application.

Mode bits (M2 M1 M0)	Mode description
000	Mode 0, interrupt on terminal count. After the mode byte is written, the output is low. Once a count value is written, the output remains low until the counter counts down to 0. The output then goes high, and remains high until a new count or mode is written to the counter. The gate input of the counter disables counting when low. This mode can be used to generate a positive edge on the external output after a programmable time.
001	Mode 1, programmable one shot. Any rising edge on the clock input to the counter causes the output of the counter to go low for the number of clock cycles programmed into the counter. This mode can be used to generate a pulse of programmable length to external circuitry on each A/D conversion.

010	Mode 2, rate generator. The output of the counter goes low for one clock period in every N clocks, where N is the number programmed into the counter. This is the normal mode for the A/D prescaler and clock divider. The gate input of the counter disables counting when low. This mode can also be used in the configurable counter to generate an output frequency.
011	Mode 3, square wave generator. The output of the counter goes low for N/2 clocks in every N, where N is the number programmed into the counter. The counter hence generates square waves. The gate input of the counter disables counting when low. This mode is normally used in the uncommitted timer to generate an output frequency. Note that the minimum value of N is 4.
100	Mode 4, software triggered strobe. After the mode byte or a count value is written, the output is high. The output remains high until the counter counts down to 0. The output then goes low for one clock period. The gate input of the counter disables counting when low. This mode can be used to generate a pulse on the external output after a programmable time.
101	Mode 5, hardware triggered strobe. After the mode byte or a count value is written, the output is high. The counter begins counting down after a rising edge on the gate input. When the count reaches 0, the output goes low for one clock period. This mode can be used to generate a pulse on the external output after a programmable time.

Bit 0 - BCD

If this bit is 0, the counter is configured as a binary counter. If it is 1, the counter is configured as a BCD counter. Note that in either case the counter is a down counter.

5.2.9. DIOP0 - Digital I/O port 0 (offset 8)

This register is digital I/O port 0 (port A). It can be operated in one of several modes, as discussed in chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP0 register is shown in figure 5.10.

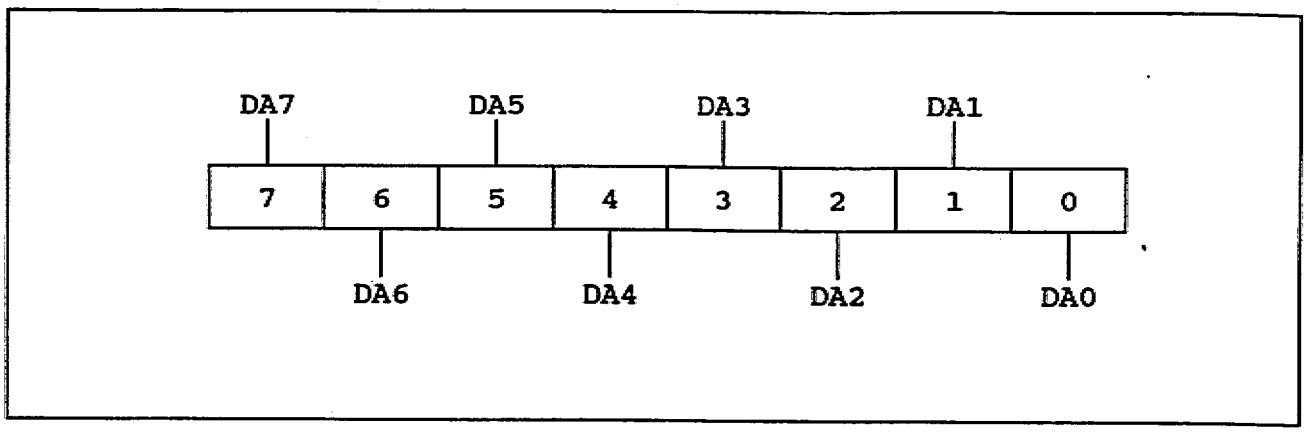


Figure 5.10. Digital I/O port 0 register.

Bits 7-0 digital I/O port 0 (DA)

These bits are port 0 of the 8255 on the PC-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bidirectional or handshake lines, depending on mode. The various modes are discussed in detail in chapter 4, and in the description of the DIOCNTRL register.

5.2.10. DIOP1 - Digital I/O port 1 (offset 9)

This register is digital I/O port 1 (port B). It can be operated in one of several modes, as discussed in chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP1 register is shown in figure 5.11.

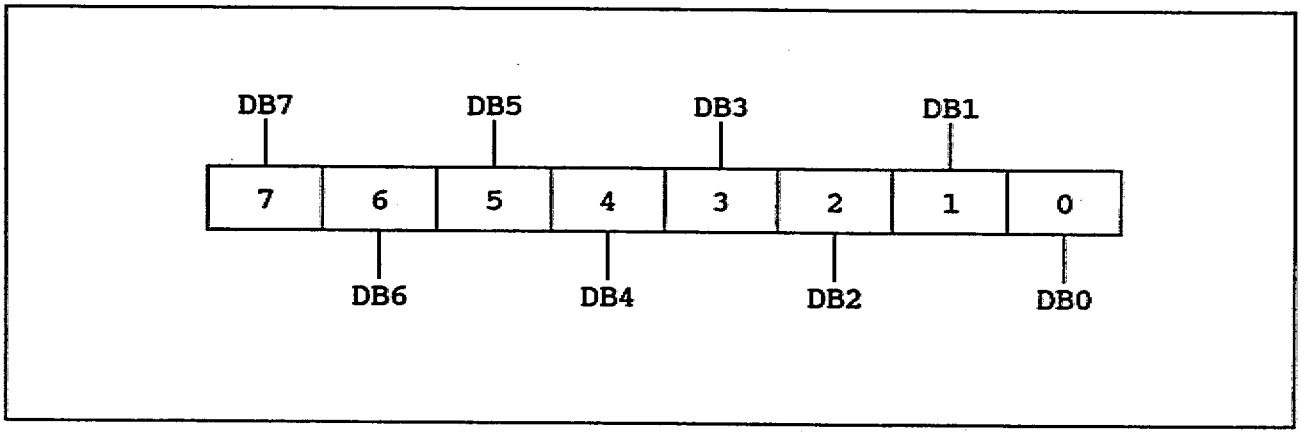


Figure 5.11. Digital I/O port 1 register.

Bits 7-0 digital I/O port 1 (DB)

These bits are port 1 of the 8255 on the PC-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bidirectional or handshake lines, depending on mode. The various modes are discussed in detail in chapter 4, and in the description of the DIOCNTRL register.

5.2.11. DIOP2 - Digital I/O port 2 (offset 10)

This register is digital I/O port 2 (port C). It can be operated in one of several modes, as discussed in chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP2 register is shown in figure 5.12.

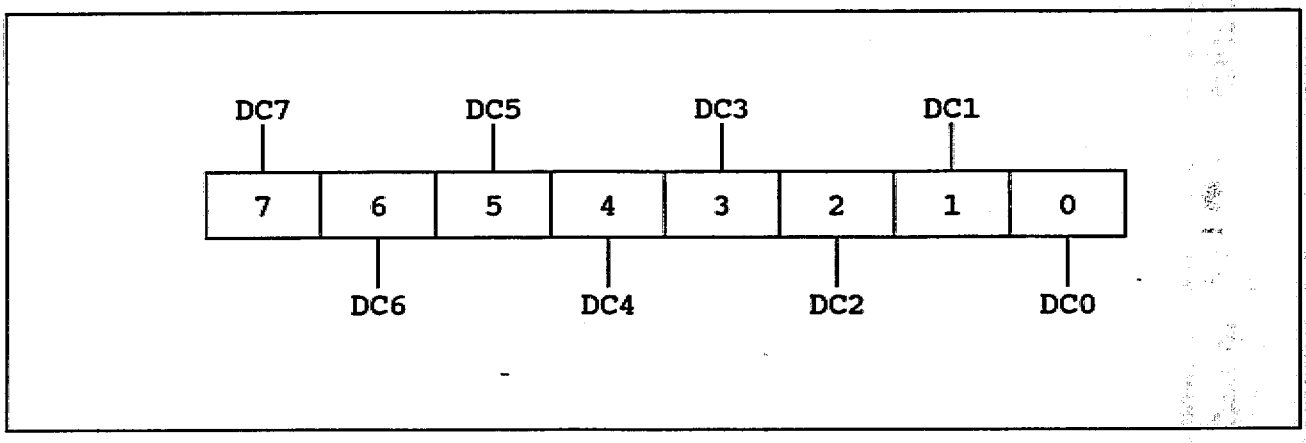


Figure 5.12. Digital I/O port 2 register.

Bits 7-0 digital I/O port 2 (DC)

These bits are port 2 of the 8255 on the PC-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bidirectional or handshake lines, depending on mode. The various modes are discussed in detail in chapter 4, and in the description of the DIOCNTRL register.

5.2.12. DIOCNTRL - Digital I/O control (offset 11)

This register can either be used to control the mode of the three digital I/O ports, or to set and reset individual bits in port C. This is the control register of the 8255 on the PC-30 board.

The various possible modes were described in chapter 4. If you intend to program the 8255 extensively, you should also obtain a data sheet for an 8255 type device for further information.

The layout of this register is shown in figure 5.13. Note that the register function and layout depends on the setting of bit 7.

Bit 7 - Function select

If this bit is 1, the register is in configuration mode. If the bit is 0, then it is in bit set/reset mode.

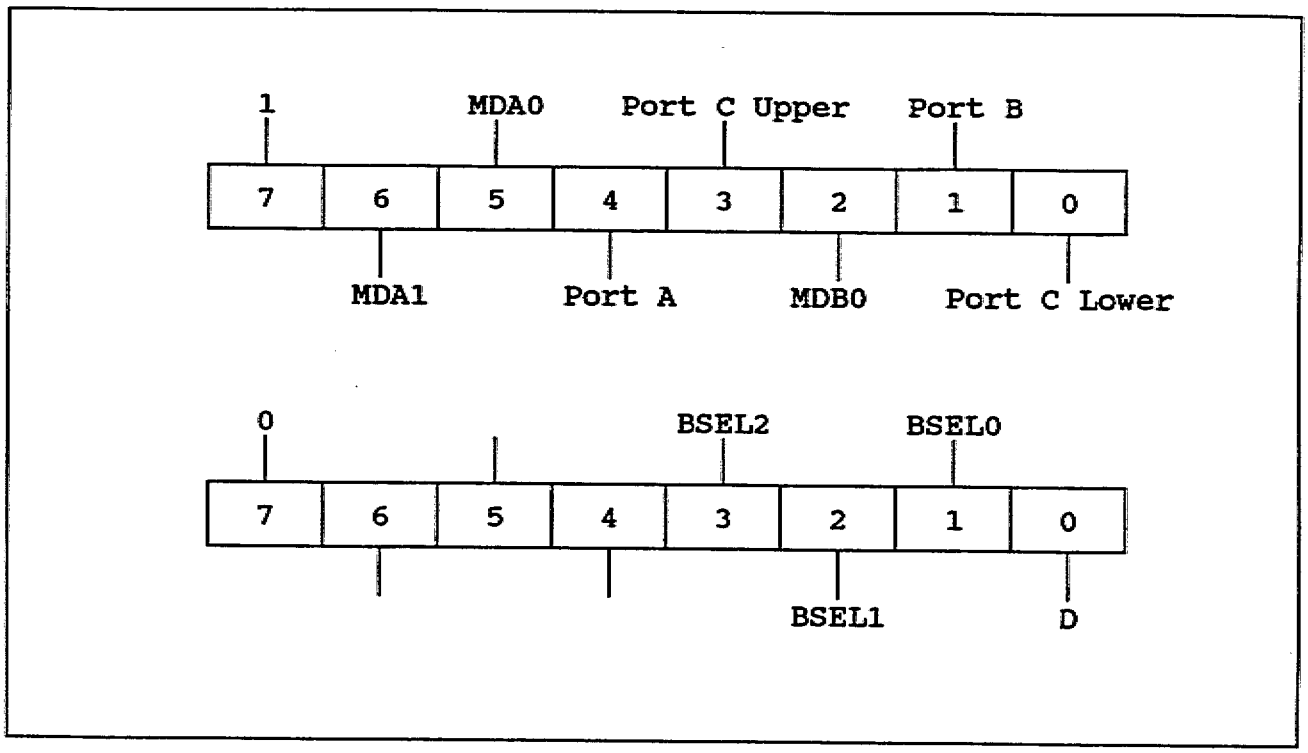


Figure 5.13. Digital I/O control register.

5.2.12.1 Configuration mode.

Bit 6-5 Group A mode set (MDA)

These two bits set the mode of the group A ports. These are port A, and the upper nibble of port C. The bit combinations are as follows:

MDA1	MDA0	Mode
0	0	Mode 0, simple I/O.
0	1	Mode 1, strobed I/O.
1	0	Mode 2, strobed bidirectional I/O.
1	1	Reserved; do not use.

Bit 4 - Port A input

If this bit is set, then port A functions as an input. If it is zero, then port A is an output.

Bit 3 - Port C upper input

If this bit is set, then the group A portion of port C (the upper nibble) functions as an input. If it is zero, then the upper nibble of port C is an output.

Bit 2 Group B mode set (MDB)

This bit sets the mode of the group B ports. These are port B, and the lower nibble of port C. The bit settings are as follows:

MDB0	Mode
0	Mode 0, simple I/O.
1	Mode 1, strobed I/O.

Bit 1 - Port B input

If this bit is set, then port B functions as an input. If it is zero, then port B is an output.

Bit 0 - Port C lower input

If this bit is set, then the group B portion of port C (the lower nibble) functions as an input. If it is zero, then the lower nibble of port C is an output.

5.2.12.2 Bit set/reset mode.

Bits 6-4 - Reserved.

The content of these bits is ignored.

Bits 3-1 - Bit select (BSEL)

The BSEL bits serve to select the bit in port C which is to be modified. A code of 000 will select bit 0, a code of 001 bit 1 etc.

Bit 0 - Data (D)

The data bit represents the value to which the selected bit will be set.

5.2.13. DADATL0 - DAC0 register (low byte) (offset 12)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC0 by software for D/A conversions. Data is left justified. Figure 5.14 shows the register layout. Data is transferred to the output when this register is written.

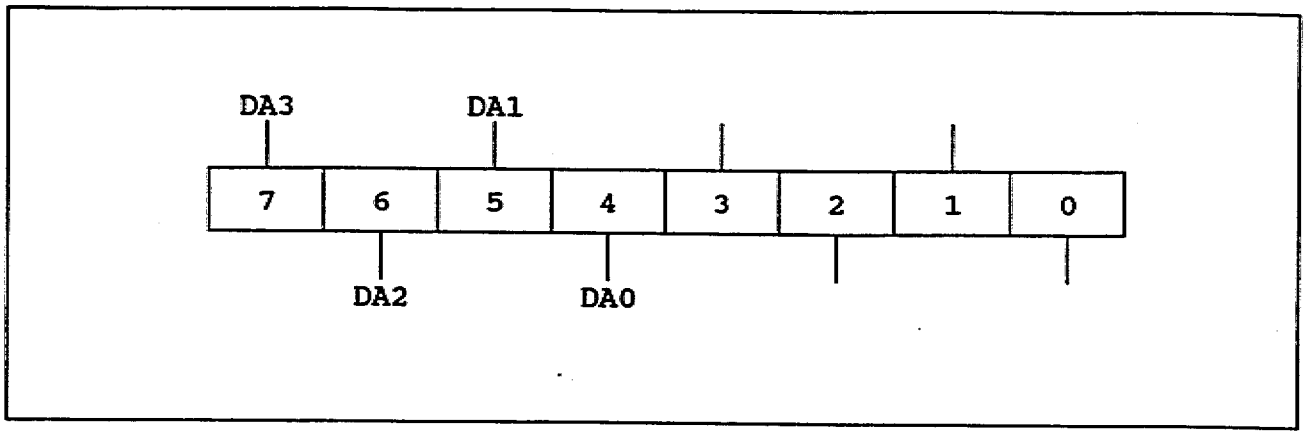


Figure 5.14. DAC0 low byte data register.

Bits 7-4 - DAC0 data (DA)

These bits are the LSB of DAC0 data.

Bits 3-0 - Not used.

5.2.14. DADATH0 - DAC0 register high byte (offset 13)

DADATH0 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in figure 5.15.

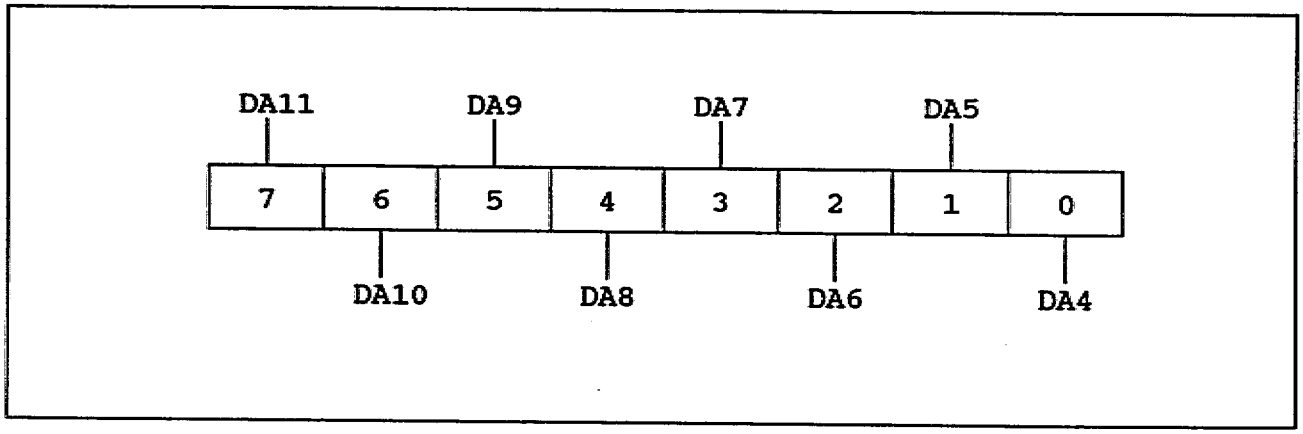


Figure 5.15. DAC0 high byte data register.

Bits 7-0 - DAC) data (DA)

These eight bits are the MSB of the DAC0 data.

5.2.15. DADATL1 - DAC1 register (low byte) (offset 16)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC1 by software for D/A conversions. Data is left justified. Figure 5.16 shows the register layout. Data is transferred to the output when this register is written.

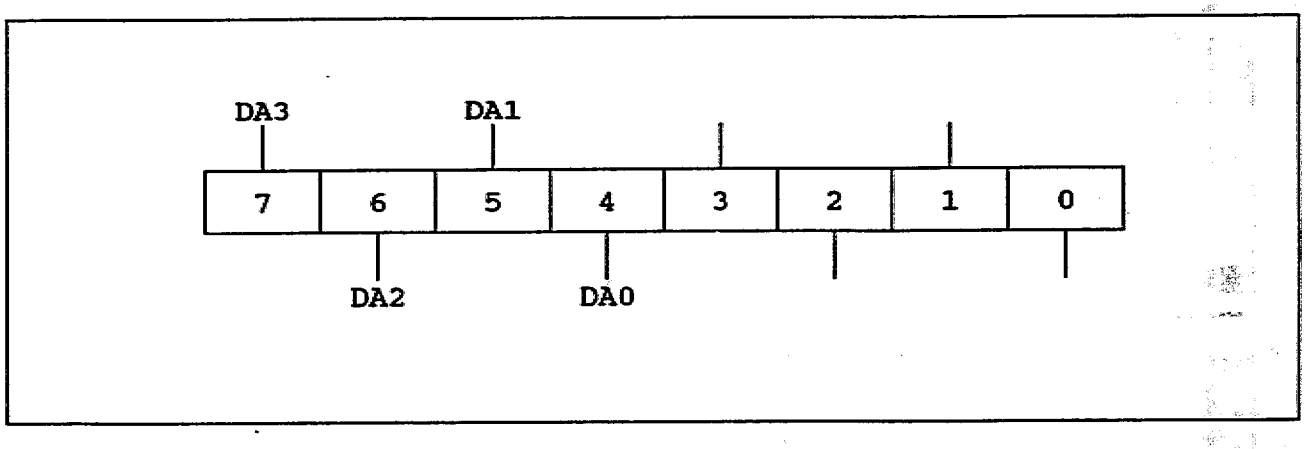


Figure 5.16. DAC1 low byte data register.

Bits 7-4 - DAC1 data (DA)

These bits are the LSB of DAC1 data.

Bits 3-0 - Not used.

5.2.16. DADATH1 - DAC1 register high byte (offset 17)

DADATH1 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in figure 5.17.

Bits 7-0 - DAC1 data (DA)

These eight bits are the MSB of the DAC1 data.

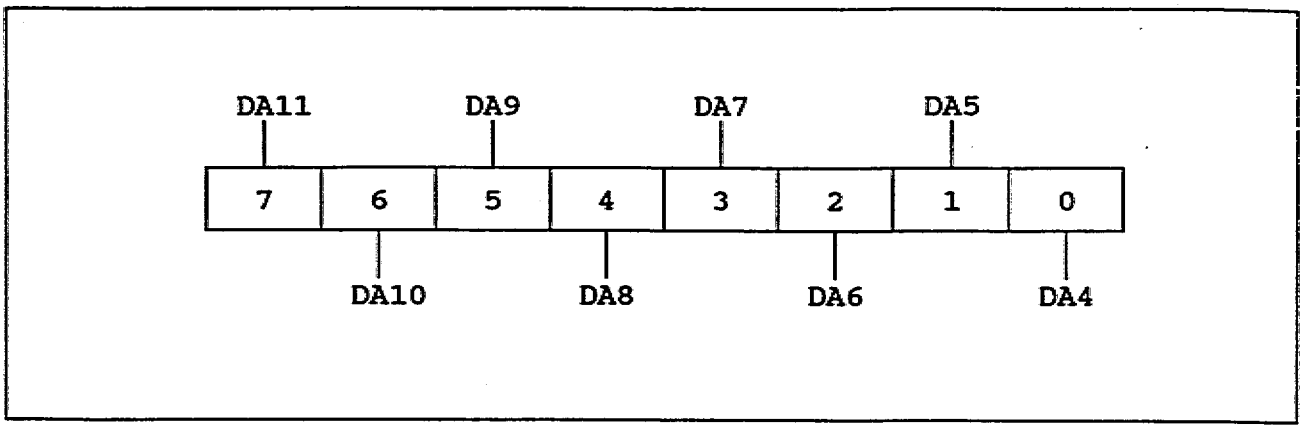


Figure 5.17. DAC1 high byte data register.

5.2.17. DADAT2 - DAC2 register (offset 20)

DADAT2 high byte holds the eight bits of the software-loaded value for D/A conversion. Bit 7 is the MSB. Data is left justified. The layout of this register is shown in figure 5.18.

Bits 7-0 - DAC2 data (DA)

These eight bits contain the code which represents the analog value on the output of DAC2.

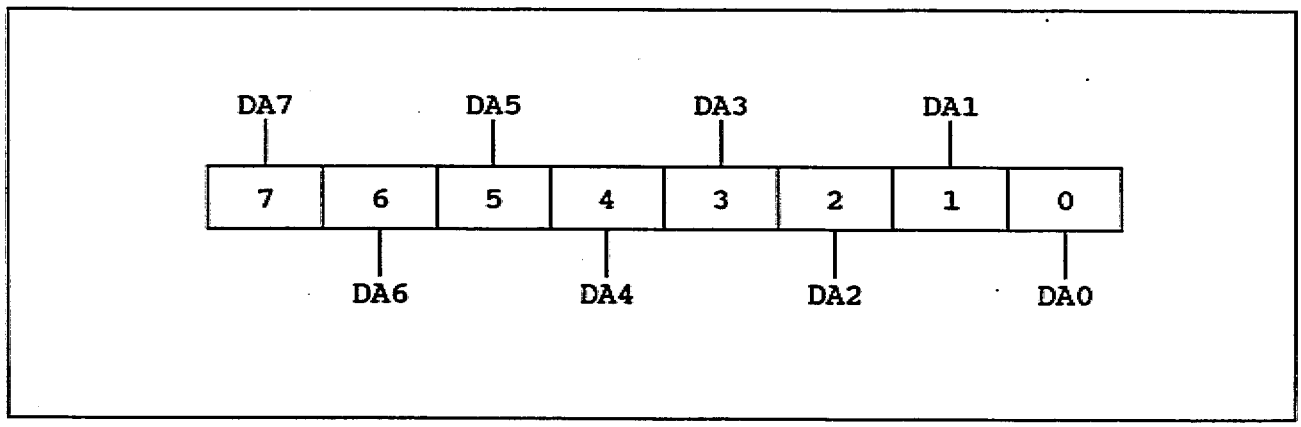


Figure 5.18. DAC2 data register.

5.2.18. DADAT3 - DAC3 register (offset 21)

The DADAT3 register holds the eight bits of the software-loaded value for D/A conversion. Bit 7 is the MSB. Data is left justified. The layout of this register is shown in figure 5.19.

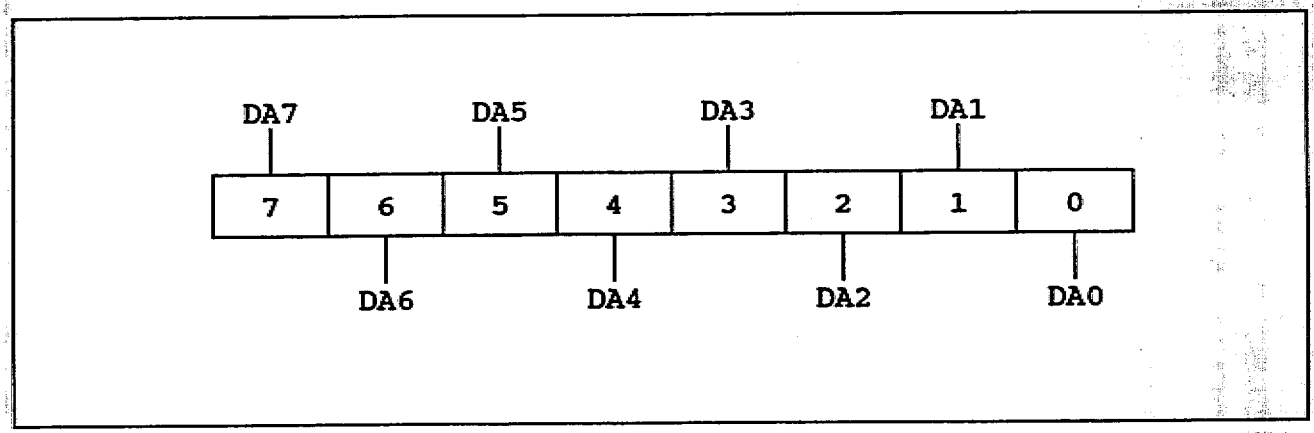


Figure 5.19. DAC3 data register.

Bits 7-0 - DAC3 data (DA)

These eight bits contain the code which represents the analog value on the output of DAC3.

Chapter 6

Programming Guide

This chapter gives a "How To" guide to programming the PC-26, PC-30 and PC-39. In order to make best use of the contents of this chapter, you should be familiar with the contents of the previous chapter.

Driver software for the PC-26, PC-30 and PC-39 is available, and where possible you should make use of this software, rather than write your own. However, if you do decide to write your own low-level code, we strongly recommend that you study the source code for the drivers. This code is found in the \SOURCE directory of the driver diskette supplied with the driver software package. When you study this code however, bear in mind that the code is written to support all PC-30 variants, including boards not described in this manual.

Where possible, in the following guide there are references to where you can find example code showing how to perform various functions.

6.1. Converting from binary to analog values

Analog data from the A/D converter, and to the D/A converters, is always in the form of offset binary code. Analog voltages may be calculated from the digital codes as follows:

6.1.1. A/D converter codes

i. For the 0 to +10V range :

$$\text{Voltage} = (\text{Digital code})(10)/4096$$

ii. For the -5 to +5V range :

$$\text{Voltage} = (\text{Digital code} - 2048)(5)/2048$$

iii. For the -10 to +10V range :

$$\text{Voltage} = (\text{Digital code} - 2048)(10)/2048$$

6.1.2. DAC0 and DAC1 converter codes

DAC0 and DAC1 are 12 bit converters. Note that the bipolar voltage range is inverted. Conversion is as follows:

i. For the 0 to +10V range :

$$\text{Voltage} = (\text{Digital code})(10)/4096$$

ii. For the -10 to +10V range :

$$\text{Voltage} = -(\text{Digital code} - 2048)(10)/2048$$

6.1.3. DAC2 and DAC3 converter codes

DAC2 and DAC3 are 8 bit converters. Note that the bipolar voltage range is inverted. Conversion is as follows:

i. For the 0 to +10V range :

$$\text{Voltage} = (\text{Digital code})(10)/256$$

ii. For the -10 to +10V range :

$$\text{Voltage} = -(\text{Digital code} - 128)(10)/256$$

Note:
The above formulas all assume that the PC-30 is calibrated as described in chapter 7. If you do not use the recommended calibration procedure, these formulas may not apply.

6.2. Initialization

In order to initialize the PC-30, the following steps should be performed. The function Init, supplied with the PC-30 driver software, performs this function. This sequence should be followed prior to attempting any function.

i. Write 92(hex) to the A/D mode register (ADMDE).

ii. Write 34(hex) to the counter control register (TMRCTR). This sets the mode of the A/D clock prescaler to 2.

iii. Write 30(hex) to the counter control register (TMRCTR). This sets the mode of the A/D clock divider to 2.

iv. Write B6(hex) to the counter control register (TMRCTR). This sets the mode of the uncommitted counter/timer to 3.

v. Write 02(hex) to the A/D control/channel register (ADCCR). This disables DMA and interrupts, and sets the A/D for software strobes.

vi. Write 0 to the digital I/O control register (DIOCNTRL). This configures all digital input lines as inputs.

vii. Wait at least 100 uS.

viii. Read the high and low byte of the A/D data register.

The PC-30 is then ready for operation.

6.3. Clearing the A/D subsystem

Before using the A/D subsystem, it is important to wait for any current A/D conversion to complete, and to clear any information in the A/D data registers. The sequence below performs this function, as well as clearing the PC-30D's FIFO buffer. The function Clean, supplied with the PC-30 driver software, performs this function. This sequence should be followed prior to attempting any A/D input function.

i. Write 92(hex) to the A/D mode register (ADMDE).

ii. Write 02(hex) to the A/D control/channel register (ADCCR). This disables DMA and interrupts, and sets the A/D for software strobes.

iii. Read the high and low byte of the A/D data register.

iv. Wait at least 100 uS, or until the done bit is set.

v. Read the high and low byte of the A/D data register.

6.4. Writing to the D/A converters

In order to write to the D/A converters, all that is necessary is to convert the required voltage to a digital code, and then write this to the appropriate registers. Remember that, in the case of the 12-bit converters, the code must be shifted left by 4 bits, and that the D/A output is not updated until the low byte register is written.

6.5. Digital I/O

Digital I/O is performed simply by reading or writing the required digital values to the appropriate registers. You must however remember to configure the port in use before reading and writing. The initialization procedure above configures all the digital ports as inputs.

6.6. Obtaining a single A/D reading - PC-26/PC-30

To obtain a single A/D reading under program control, proceed as follows:

i. Clear the A/D subsystem as described above.

ii. Write a byte containing the address of the channel you wish to convert, with the STBC bit set, and all other bits cleared, to the ADCCR.

iii. Write the same byte, but with the SSTB bit set as well as the STBC bit, to the ADCCR.

iv. Write the same byte, but with the SSTB bit cleared, to the ADCCR.

v. Wait for at least 40 uS.

vi. Read the result from the ADDSR and the ADDATL registers.

6.7. Obtaining a single A/D reading - PC-39

To obtain a single A/D reading under program control, proceed as follows:

i. Clear the A/D subsystem as described above.

ii. Write a byte containing the address of the channel you wish to convert, with the STBC bit set, and all other bits cleared, to the ADCCR.

iii. Write the same byte, but with the SSTB bit set as well as the STBC bit, to the ADCCR.

iv. Write the same byte, but with the SSTB bit cleared, to the ADCCR.

v. Wait for the Done bit in the ADDSR to be set.

vi. Read the result from the ADDSR and the ADDATL registers.

6.8. Setting the sample rate

Assuming that the PC-30 is initialized as described above, setting the sampling rate is simple:

i. Decide on values for the A/D clock prescaler and divider. For example, assuming a bus clock of 8 MHz, to sample at 100KHz, you could set the prescaler to 2, and the divider to 10 (or vice-versa). Remember that the maximum value for either the prescaler or the divider is FFFF(hex).

ii. Write the LSB of the prescaler value to the PRESCALER register, then the MSB.

iii. Write the LSB of the divider value to the DIVIDER register, then the MSB.

The sampling rate is then set. The procedures Ad_prescaler and Ad_clock (in the file PC30S.C) show how to perform the above functions.

6.9. Detecting the end of conversion

When you want to obtain a sequence of samples, detecting when A/D conversions end is very important. For the PC-39, all that is required is to monitor the Done bit in the ADDSR register. This bit is however not present in PC-26 or PC-30 boards.

In order to detect end of conversion on PC-26 or PC-30 boards, you have to make use of the interrupt system. There are two ways of doing this :

i. Use the interrupts directly. This is the obvious way, and is discussed in a later section. Using interrupts does require some programming skill, and is difficult to implement

under some environments (OS/2 for example). Fortunately, there is a second way.

ii. It is possible to monitor the status of the PC-26/PC-30 interrupt line via the interrupt controller. All that is required is to read bit 5 of the system board 8259 interrupt controller's control register. This is at address 20 hex. If this bit is high, then the interrupt line is active. Code for doing this can be found in the S_Chan function in the PC30S.C file supplied on the driver disk. Note however the following :

a) This technique only works on 100% PC and PC/XT compatibles. Some PC clones with "chip-set" type motherboards don't fully implement the 8259.

b) The interrupt line is active for only 10 uS. This means that you have to disable all other interrupts, and you have to write in an efficient language. Interpreted Basic is not a good idea!

c) Coded in C, as in the driver software, a standard PC can achieve up to 12 KHz throughput, and a 386 system 25 KHz, using this technique.

d) Remember that PC-26 and PC-30's don't have error detection, so you will have to test your systems maximum speed manually. To do this, use Status-30, and sample a signal of known frequency at a low rate (1 KHz). Then gradually increase the sampling rate until the measured frequency of your test signal shows a change from what it should be.

6.10. Obtaining a series of A/D conversions by polled I/O

Polled I/O is by far the simplest way to obtain a sequence of samples. It is however limited to about 25 KHz throughput. The procedure is as follows:

i. Set the sampling rate, as described above.

ii. Load the channel to be converted into the channel register.

iii. Set the STBC bit in the ADCCR to 0. This enables A/D strobes.

iv. Wait for the A/D conversion to complete. As soon as it is, read the A/D result into memory. How to detect the end of an A/D conversion is discussed above.

v. Repeat step iv until you have collected as many samples as you require.

vi. When the sampling procedure is complete, set the STBC bit to 1.

The procedures S_chan (for single channel operation) and Mb_chan (for multi-channel operations), in the file PC30S.C show how the driver software performs this function.

6.11. Interrupts

Interrupt based I/O allows the PC's CPU to perform other tasks while the PC-30 acquires data. It is however limited to low speed applications. Throughput of about 10KHz is typical. Note that if you intend to write your own interrupt based routines, that you must have a thorough understanding of both the PC and the operating system in use. A complete description of interrupt handlers is well beyond the scope of this manual. However, the basic procedure is described below:

i. Set the sampling rate, as described above.

ii. Load the channel to be converted into the channel register.

iii. Set the PC's interrupt vector to the address of your interrupt handling procedure. This procedure must read in the results of the A/D conversion, as well as halt operations when sufficient samples have been obtained. Remember also that the interrupt handler must send an EOI (end of interrupt) command to the interrupt controller.

iv. Set the interrupt enable bit in the ADCCR to 1. This enables the PC-30 interrupts.

v. If you are using a PC-26 or PC-30, disable the PC's clock interrupt. To see how to do this, take a look at the Rtc_on and Rtc_off functions in the PC-30 driver software. If you are using a PC-39, disabling the PC's clock is not required.

vi. Set the STBC bit in the ADCCR to 0. This enables A/D strobes. The program can then continue with other work.

vii. As soon as an A/D conversion completes, control is passed to the interrupt handling procedure. This continues until the interrupt handler disables interrupts.

viii. When the sampling procedure is complete, set the STBC bit to 1, and the interrupt enable bit to 0.

The procedures Mi_chan, Int_chk and Int_close in the file PC30I.C show how the driver software performs this function.

6.12. Single channel DMA

Only the PC-39 can perform single channel DMA. DMA is normally the only way to achieve full throughput on the PC-39 board, and also allows the program to continue with other activities while the DMA takes place.

If you intend to write your own DMA based routines, you must have a thorough understanding of the both the PC, and the operating system in use. A complete description of DMA procedures is well beyond the scope of this manual. The basic procedure is described below:

i. Set the sampling rate, as described above.

ii. Load the channel to be converted into the channel register.

iii. Set up the PC's DMA hardware with the address of the section of memory into which you wish to transfer the results of the A/D conversions. Also remember to enable the DMA level you intend to use. The DMA controller should be programmed for demand mode operation.

iv. Set the DMA enable bit in the ADCCR to 1. This enables the PC-30 DMA.

v. Set the STBC bit in the ADCCR to 0. This enables A/D strobes. The program can then continue with other work.

vi. As soon as an A/D conversion completes, the results of the conversion are transferred to memory. This continues until the count value programmed into the DMA hardware in the PC reaches 0.

vii. When the sampling procedure is complete, set the STBC bit to 1 and the DMA enable bit to 0.

Note that the memory space to which the data is transferred should start on a even byte, and that the entire memory space must be inside a single 64K segment (the upper 4 bits of the address cannot change from the start to the end of the segment).

Note that the status bits in the ADDSR are transferred to memory along with the data, and you hence have to mask your data prior to use.

The procedures Sd_chan (for single channel operation), Dma_chk and Dma_close in the file PC30C.C show how the driver software performs this function.

6.13. DMA data format

After DMA has completed, the format of the data in the DMA bufferas follows:

	Bit 7 (MSB)				Bit 0 (LSB)
Byte 0	Sample 0 Bits 7-0				
Byte 1	-	Done	-	Trig	Sample 0 Bits 11-8
Byte 2	Sample 1 Bits 7-0				
Byte 3	-	Done	-	Trig	Sample 1 Bits 11-8
Byte 4	Sample 2 Bits 7-0				
Byte 5	-	Done	-	Trig	Sample 2 Bits 11-8
	. . . .				

[This page intentionally left blank.]

Chapter 7

Calibration

7.1. Introduction

This chapter contains information on the calibration procedures for the A/D and D/A subsystems on the PC-30 series of boards.

These procedures should be performed at six month intervals, or whenever the input or output range jumpers are changed.

NOTE
Allow the host PC and the board to warm up for at least one hour before calibration.

7.2. A/D calibration

A/D calibration is performed by adjusting three trimpots, VR1, VR2 and VR3. These trimpots are easily located from the board layout shown in appendix C, or the labels on the PC-30 board itself.

7.2.1. Requirements

i. Calibration is done on channel 1. The recommended connector wiring is shown in figure 7.1.

ii. Calibration is performed with the board jumpered into its intended operating mode.

iii. All cables should be as short as possible.

7.2.2. Equipment required

i. Precision voltage source. Range +10 to -10 V, absolute accuracy better than 0.005%,

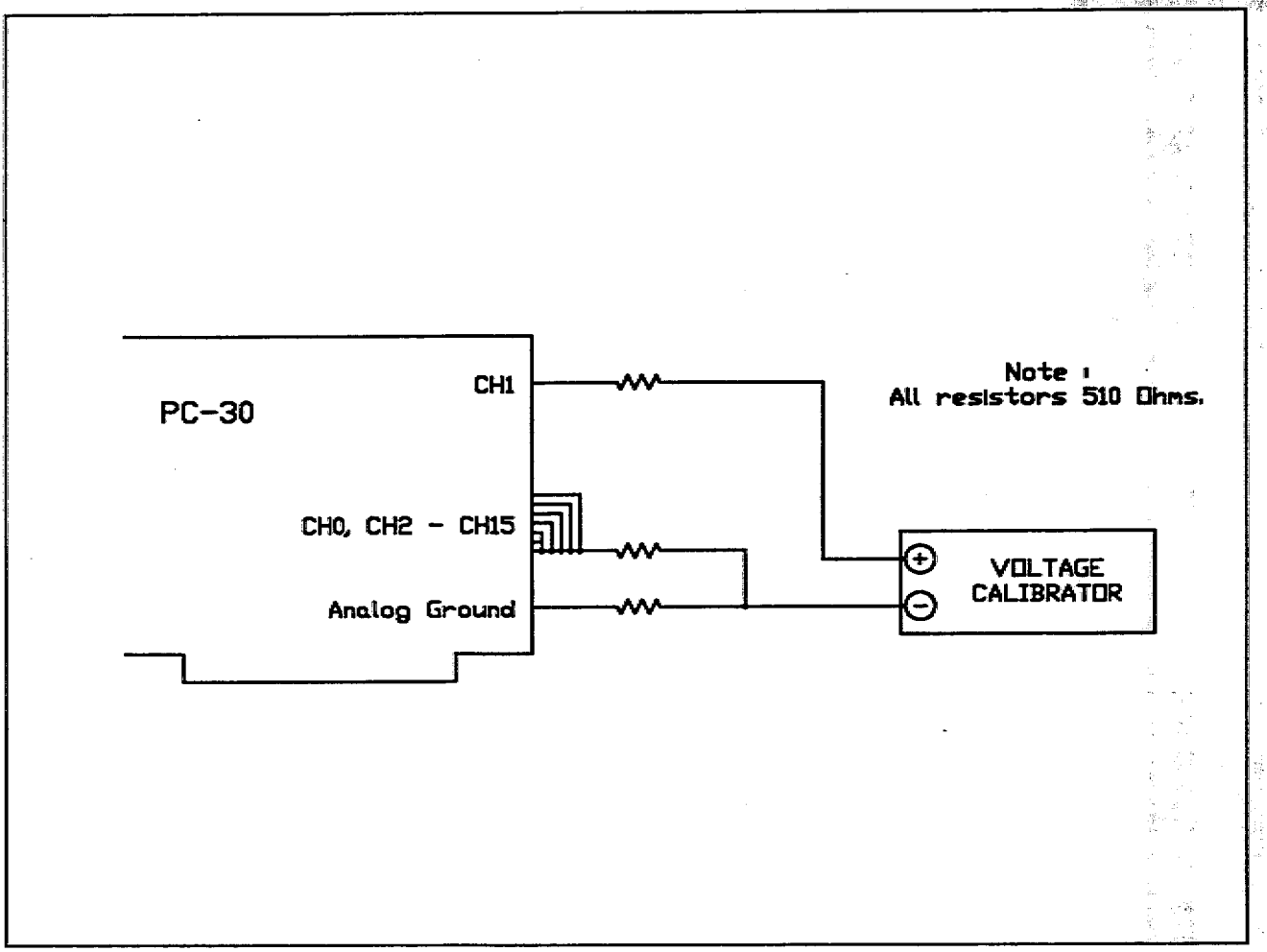


Figure 7.1. A/D calibration connections.

resolution 100 nV or better.

7.2.3. Procedure

7.2.3.1 Unipolar mode

i. Apply (-FS + 1/2 LSB) to channel 1. This is 1.20 mV for the 10 V range. Adjust VR3 (unipolar A/D offset potentiometer) for an output code which flickers evenly between 000H and 001H.

ii. Apply (+FS - 3/2 LSB) to channel 1. This is 9.9963 V for the 10 V range. Adjust VR2 (gain potentiometer) for an output code which flickers evenly between FFEH and FFFH.

iii. Repeat the above two steps until no further adjustment is required.

7.2.3.2 Bipolar mode

i. Apply (-FS + 1/2 LSB) to channel 1. This is -9.9976V for the -10 to 10V range, and -4.9988V for the -5 to 5V range. Adjust VR1 (bipolar A/D offset potentiometer) for an output code which flickers evenly between 000H and 001H.

ii. Apply (+FS - 3/2 LSB) to channel 1. This is +9.9927V for the -10 to 10V range, and +4.9963V for the -5 to 5V range. Adjust VR2 (gain potentiometer) for an output code which flickers evenly between FFEH and FFFH.

iii. Repeat the above two steps until no further adjustment is required.

7.3. A/D calibration Software

The program CAL30.EXE, supplied on the distribution disk, automates the above procedure. Note that for correct operation, the setup information supplied in the first menu must be correct.

7.4. DAC0 and DAC1 calibration

D/A calibration is performed by adjusting three trimpots for each 12-bit D/A (DAC0 and DAC1). These trimpots are easily located from the board layout shown in appendix C, or the labels on the PC-30 board itself.

7.4.1. Requirements

i. The recommended connector wiring is shown in figure 7.2.

ii. Calibration is performed with the board jumpered into its intended operating mode.

iii. All cables should be as short as possible.

7.4.2. Equipment required

i. Precision Multimeter. Range +10 to -10 V, accuracy 10 uV for voltage.

7.4.3. Procedure

7.4.3.1 Monopolar mode

The procedure below gives the steps required to calibrate the 12-bit D/A converters in monopolar (0 to 10V) mode.

i. Set the D/A code to 000H. Use VR5 (DAC0) or VR7 (DAC1) to adjust the D/A output to within 1/2 LSB of the -FS (0.000V) for the range.

ii. Set the D/A code to FFFH. Use VR4 (DAC0) or VR6 (DAC1) to adjust the D/A output to within 1/2 LSB of (+FS - 1 LSB). This is 9.9976 V for the monopolar range.

These two adjustments interact to some extent, and hence the sequence should be repeated until no further adjustment in either trimpot is required.

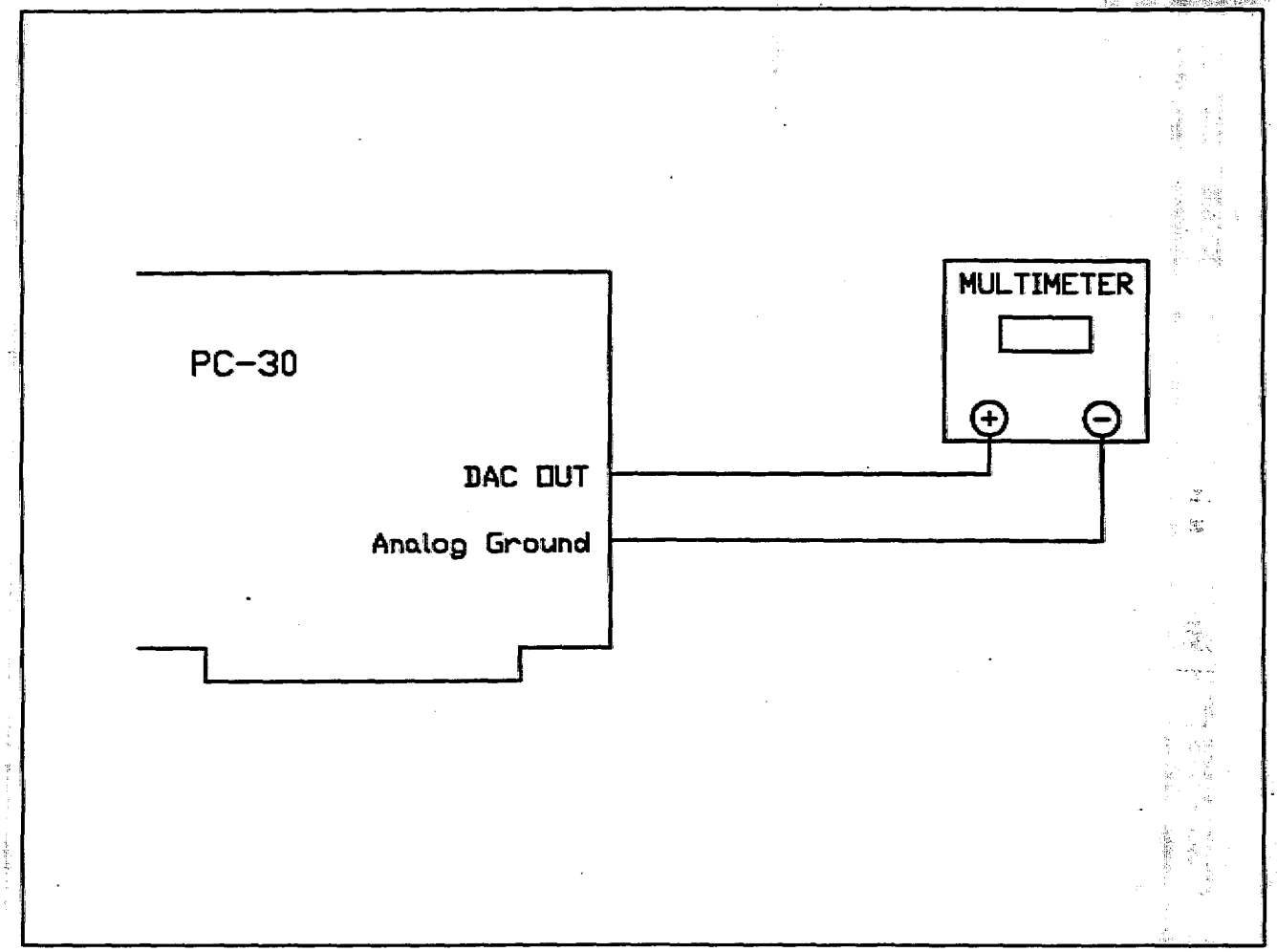


Figure 7.2. D/A calibration connections.

7.4.3.2 Bipolar mode

The procedure below gives the steps required to calibrate the 12-bit D/A converters in bipolar (-10 to 10V) mode.

i. Set the D/A code to 000H. Use VR5 (DAC0) or VR7 (DAC1) to adjust either pin 6 of IC21 (DAC0) or pin 6 of IC23 (DAC1) to 0.000 V +- 1mV.

ii. Set the D/A code to 000H. Use VR10 (DAC0) or VR11 (DAC1) to adjust the D/A output to within 1/2 LSB of the FS (10.000V) for the range.

iii. Set the D/A code to FFFH. Use VR4 (DAC0) or VR6 (DAC1) to adjust the D/A output to within 1/2 LSB of (-FS + 1 LSB). This is -9.9951 V for the bipolar range.

These adjustments interact to some extent, and hence the sequence should be repeated until no further adjustment in any trimpot is required.

7.5. DAC2 and DAC3 calibration

D/A calibration is performed by adjusting a single trimpot for each 8-bit D/A (DAC2 and DAC3). These trimpots are easily located from the board layout shown in appendix C, or the labels on the PC-30 board itself.

7.5.1. Requirements

i. The recommended connector wiring is shown in figure 7.2.

ii. Calibration is performed with the board jumpered into its intended operating mode.

iii. All cables should be as short as possible.

7.5.2. Equipment required

i. Precision Multimeter. Range +10 to -10 V, accuracy 10 uV for voltage.

7.5.3. Procedure

7.5.3.1 Monopolar mode

The procedure below gives the steps required to calibrate the 8-bit D/A converters in monopolar (0 to 10V) mode.

i. Set the D/A code to FFH. Use VR9 (DAC2) or VR8 (DAC3) to adjust the D/A output to within 1/2 LSB of (+FS - 1 LSB). This is 9.961 V for the monopolar range.

7.5.3.2 Bipolar mode

The procedure below gives the steps required to calibrate the 8-bit D/A converters in bipolar (-10 to 10V) mode.

i. Set the D/A code to FFH. Use VR9 (DAC0) or VR8 (DAC1) to adjust the D/A output to within 1/2 LSB of (-FS + 1 LSB). This is -9.922 V for the bipolar range.

7.6. D/A calibration Software

The program CAL30.EXE, supplied on the distribution disk, automates the procedure for calibrating the DAC outputs. Note that for correct operation, the setup information supplied in the first menu must be correct.

[This page intentionally left blank.]

Appendix A

Hardware Specifications

I. Analog Input

1. Number of Input Channels

16 single-ended

2. Resolution

12-bit, 1 in 4096

3. Total System Accuracy

+- 1 LSB

4. Differential Nonlinearity

+- 3/4 LSB max.

5. Quantization Uncertainty

+-1/2 LSB

6. Input Ranges

-5 to +5V, 0 to +10V, -10 to +10V

9. Gain Drift

+- 30 ppm per degree C.

10. Offset Drift

+- 10 ppm per degree C.

11. Input Impedance

PC-26/PC-30

22K/20 pF Off Chan typ.

22K/100 pF On Chan typ.

PC-39

10M/20 pF Off Chan typ.

10M/100 pF On Chan typ.

12. Offset Voltage

+- 5 LSB, adjustable to zero.

13. Monotonicity

0 to 70 degree C

14. Data acquisition rate

25 KHz (PC-26/PC-30)

80 KHz (PC-39)

II. A/D clock

4. Internal clock divider

16-bit prescaler, 16-bit divider.

5. External clock

TTl compatible

6. External Trigger

TTL compatible, readable via software.

III. Analog Output (PC-30/PC-39 only)

1. Number of Channels

4

2. Resolution

Two 12-bit, two 8-bit

3. Accuracy

+- 1 LSB (12-bit), +- 3 LSB (8-bit).

4. Differential Nonlinearity

+- 1 LSB

5. Quantization Uncertainty

+-1/2 LSB

6. Output Ranges

-10 to +10 V, 0 to +10V

7. Gain Error

Adjustable to 0.

8. Offset Error

Adjustable to 0 (12-bit), +-3 LSB (8-bit).

9. Gain Drift

+- 30 ppm per degree C (12-bit)

+- 0.007% per degree C (8-bit)

10. Throughput

130 KHz max.

11. Output compliance

+- 5 mA.

12. System Accuracy

0.025%

13. Monotonicity

0 to 70 degree C

IV. Digital I/O (PC-30/PC-39 only)

1. Number of Lines

24 in 3 ports

2. Compatibility

TTL

3. Interface

Programmable for simple I/O, strobed I/O or handshake I/O.

V. Timer/Counter

1. Resolution

16 bit

2. Compatibility

TTL

VI. I/O Connector

- **PC-26**

25 way female D-type Amphenol DB25 or equivalent.

PC-30/PC39

50 way female D-type Amphenol DB50 or equivalent.

VII. Environmental

1. Operating Temperature

0 to 70 degrees C

2. Storage Temperature

-55 to 150 degrees C

3. Relative Humidity

5% to 95% noncondensing

VIII. PC Interface

1. Base Address

700 or 780, jumper selectable.

2. Number of registers

32 8-bit registers.

3. Interrupts

On end of conversion, software controlled.

4. DMA

Single channel, 8-bit jumper selectable (PC-39 only)

IX. Power

1. +5V

1,5A typ.

2. +12V

100mA typ.

3. -12V

100mA typ.

[This page intentionally left blank.]

Appendix B

PC-26/PC-30/PC-39 Compatibility

This appendix discusses the compatibility of the PC-39 with the older series boards (PC-26 and PC-30). In general, the PC-39 is completely compatible with the older, and any well behaved software written for the PC-26 or PC-30 will operate correctly in conjunction with the PC-39.

I. PC-26/PC-30

1. Analog inputs.

The analog inputs of the PC-30 had an input impedance of 22K. The PC-39 has an industry standard input impedance of 10M/20pF (off channel) and 10M/100pF (on channel). Series resistors are used to provide fault protection.

NOTE:
Because of the 22K pull down resistors used on the PC-26/PC-30, unused PC-26/PC-30 inputs could be left open. THIS IS NO LONGER POSSIBLE. All unused PC-39 analog inputs must be grounded.

---

2. Interrupt system modifications.

The PC-39's interrupt system has improved relative to the PC-26/PC-30. The effect of these changes is as follows :

i. The PC-26/PC-30 deactivates its interrupt line approximately 10 uS after initiating an interrupt. If the host PC does not respond to the interrupt within this time, then the interrupt is lost. This means that when PC-26/PC-30 interrupts are used, all other interrupts (including the PC's clock) have to be disabled, regardless of the PC-26/PC-30's sampling frequency.

ii. The PC-39 keeps its interrupt line active indefinitely. This means the PC-39 cannot miss interrupts (unless of course an interrupt service routine takes up all the processing time from one PC-39 sample to the next). Hence it is no longer necessary to disable all other

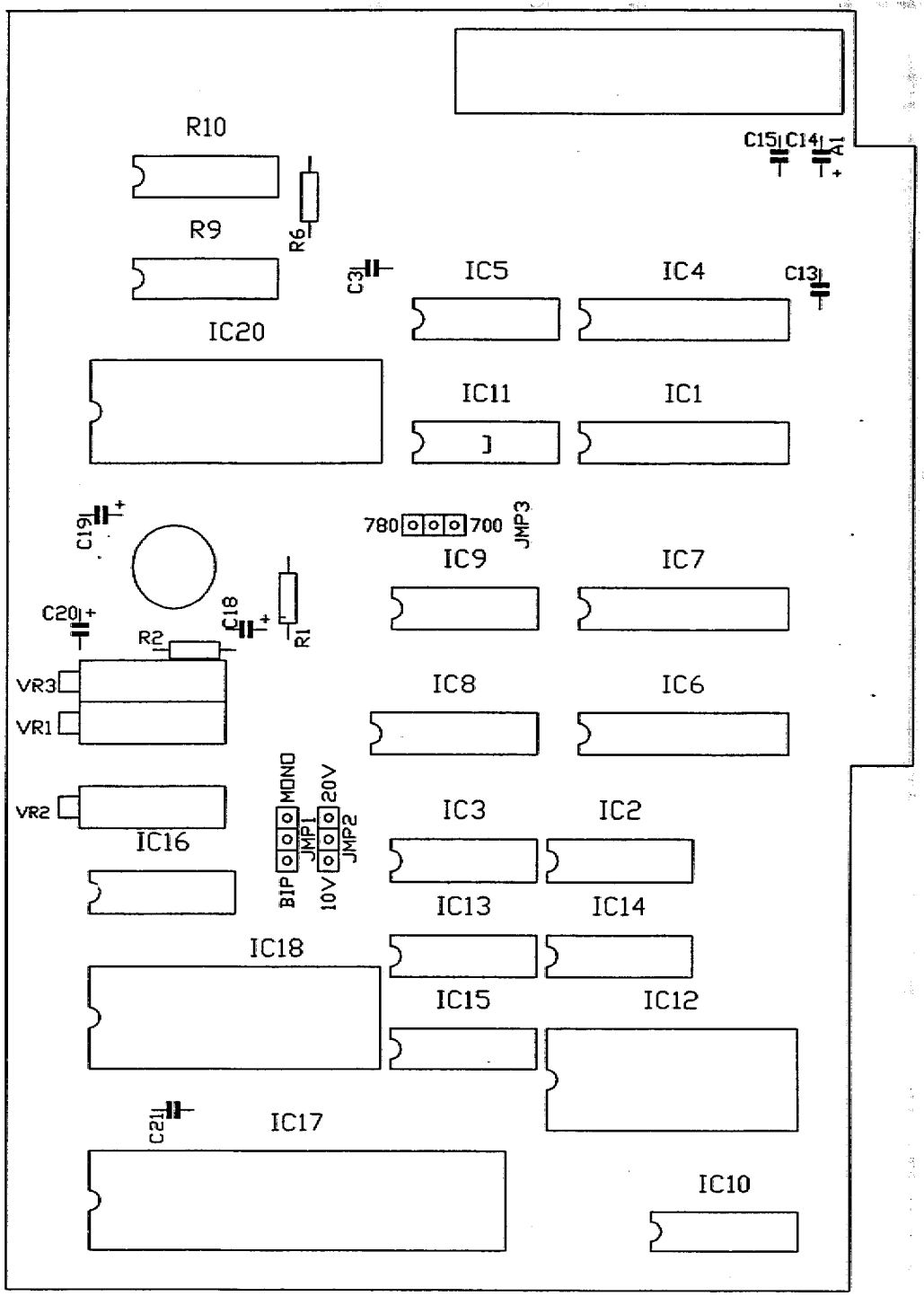
interrupts unless the PC-39 is operating at a high enough sample rate that the processing power of the PC becomes too little to handle the PC-39 as well as other interrupts.

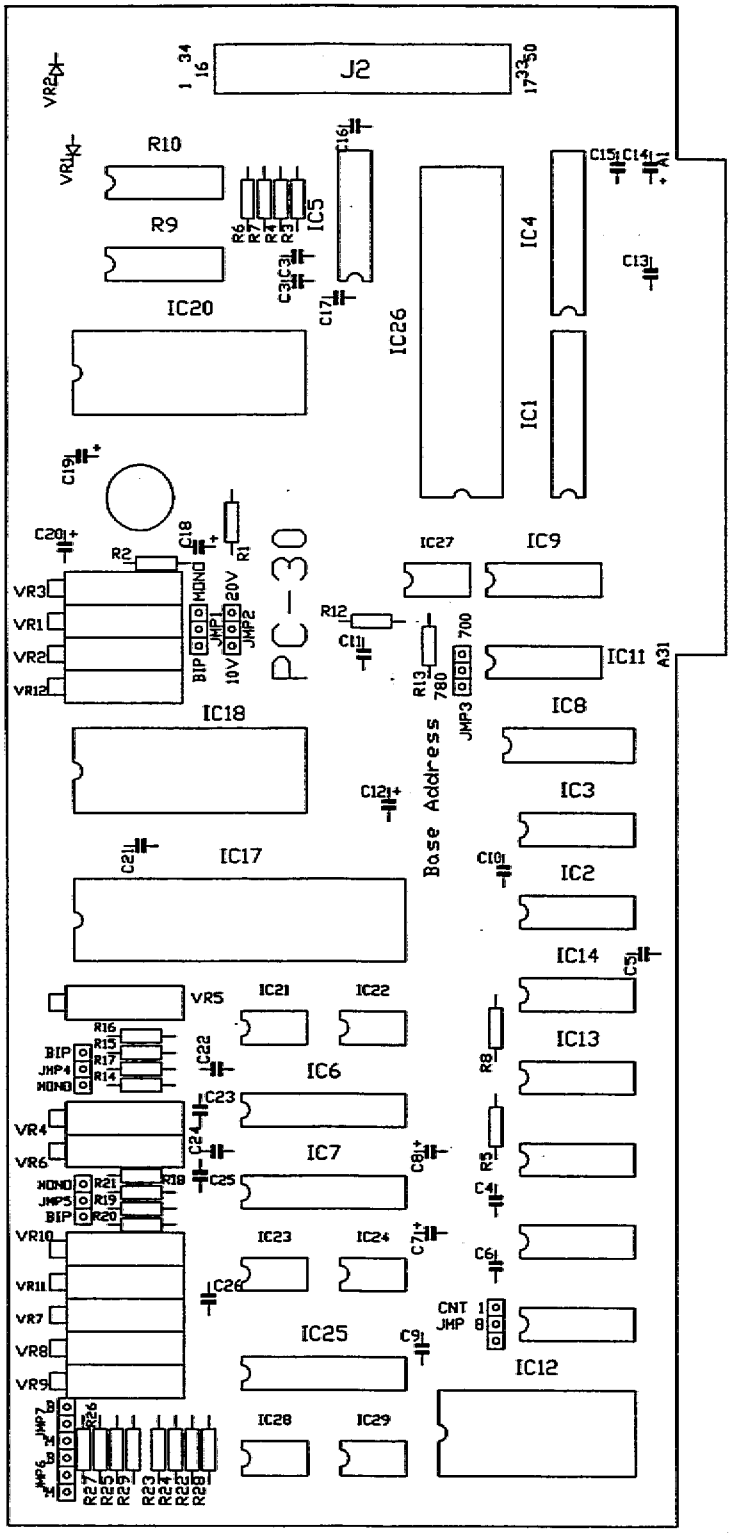
iii. This change is invisible to PC-26/PC-30 applications programs which are well behaved. Note however that the missing of interrupts served as a "safety valve" for programs which set the PC-26/PC-30's sampling rate to higher than what the program could process. In that case interrupts which could not be handled were simply lost. As the PC-39 cannot lose interrupts, such programs will now hang up, spending all processing time on interrupts, and never responding to user inputs. Early versions of the PC-26/PC-30 demo program (which allowed the user to set almost any sample rate) suffer from this problem.

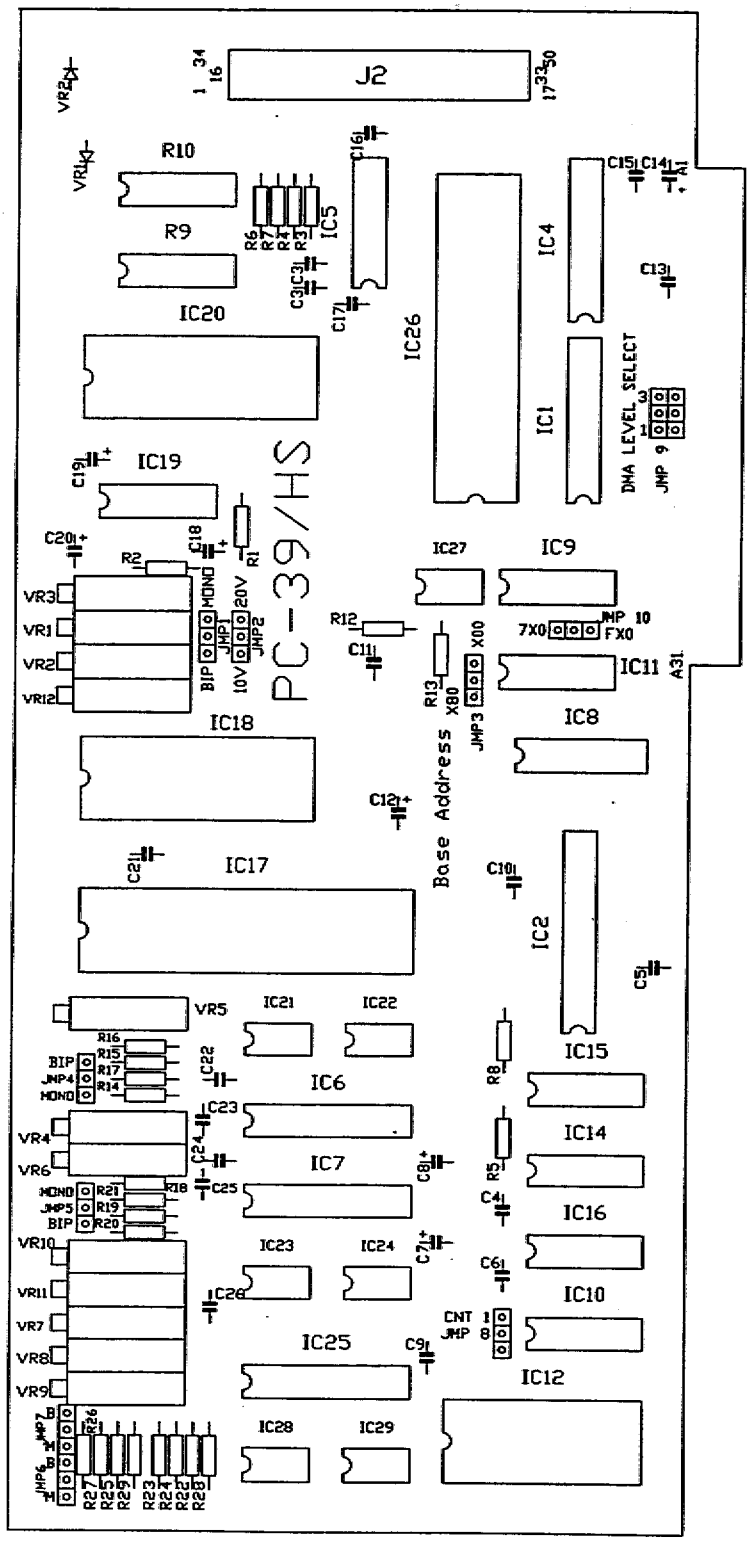
Appendix C

Layout Diagram

The following two pages show the layout diagram for the PC-26 and PC-30, as well as the PC-39.


R10
R9
R6
C3
IC5
IC4
C13
C15
C14
IC20
IC11
IC1
C19
780
700
JMP3
IC9
IC7
C20
C18
R2
R1
VR3
VR1
IC8
IC6
VR2
MONO
20V
JMP1
JMP2
IC3
IC2
IC16
BIP
10V
IC13
IC14
IC18
IC15
IC12
C21
IC17
IC10


J2
PC-30
Base Address
IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
IC11
IC12
IC13
IC14
IC17
IC18
IC20
IC21
IC22
IC23
IC24
IC25
IC26
IC27
IC28
IC29
R10
R9
VR1
VR2
VR3
VR4
VR5
VR6
VR7
VR8
VR9
VR10
VR11
VR12
JMP1
JMP2
JMP3
JMP4
JMP5
JMP6
JMP7
JMP8
CNT JMP


J2
PC-39/HS
Base Address
DMA LEVEL SELECT
JMP 9
IC1
IC2
IC4
IC5
IC6
IC7
IC8
IC9
IC10
IC11
IC12
IC14
IC15
IC16
IC17
IC18
IC19
IC20
IC21
IC22
IC23
IC24
IC25
IC26
IC27
IC28
IC29
R10
R9
VR1
VR2
VR3
VR4
VR5
VR6
VR7
VR8
VR9
VR10
VR11
VR12
JMP1
JMP2
JMP3
JMP4
JMP5
JMP6
JMP7
JMP 10
FX0
7X0
X00
X80
CNT JMP

Appendix D

Problem Determination Guide

I. Introduction

If you are experiencing problems, first check the following:

i. Remove the PC-30, and check that all ICs are firmly seated in their sockets, that there is no obvious damage to any components, and that the edge connector fingers on the PC-30 are clean.

ii. Check that the PC-30 is jumpered correctly for your application.

iii. Replace the PC-30, and check that it seats firmly in the host PC's mother-board. Also check that no components are touching a adjacent board.

iv. Check that the cable is securely plugged into the PC-30.

II. The diagnostics function.

The PC-30 contains a very comprehensive diagnostics program. All of the supplied demo programs as well as the calibration program use this, and can be used to diagnose malfunctions on the PC-30. In fact, the only PC-30 malfunctions which it will not detect are the following:

i. Damaged input multiplexer.

ii. Damaged D/A output amplifier.

iii. Damaged digital input or output lines.

III. Common problems

1. PC-30 diagnostics report board not found.

This is typically as a result of incorrect jumper settings.

2. A/D output code all zeros or all ones.

This is typically as a result of floating inputs, or an overload.

If you have exceeded the maximum input voltage (+- 12 V), you may have damaged the input multiplexers. If so, return the board to your dealer for repair.

3. A/D reading are noisy.

This may be as a result of one or more of several reasons:

i. Long leads.

ii. An electrically noisy environment

iii. Overloads on other input channels. Note also that if an input channels is overloaded it may saturate in such a way as to give a reading which appears to be in the normal range, but is very noisy.

iv. Excessive source resistance. The source resistance of the devices connected to the inputs of a PC-26 or a PC-30 should not be greater than 2.5 Ohms. The source resistance of the devices connected to the inputs of a PC-39 should not be greater than 10 KOhm.

4. The first reading in a series is inaccurate.

This is normally as a result of an overload on another input, or long leads, or a very high source impedance.

Index

+12 V 26

8253 19, 44

8255 11, 28

A/D 1, 7, 10, 13, 18, 36, 37, 38, 39, 55, 57, 58

A/D clock divider 42

A/D clock prescaler 41

A/D jumper 18

A/D mode 39

A/D strobes 9

Accessories 6

Accuracy 27

ADCCR 38

ADDATL 36

Address locations 35

ADDSR 37

ADMDE 39

Analog input 25

Analog inputs 26, 27

Analog output 25

Architecture 7

Backplane 23

Base address 3, 8, 13, 15, 35

Bipolar 2

Bit set/reset 50

C 3

Channel address 38

Clock 26

Clock divider 10, 19

Clock prescaler 9

Clock selection multiplexer 10

Clock/trigger 19

Connections 24, 25

Continuous conversion 2

Counter 19

Counter/timer 26

D/A 2, 7, 13, 17, 51, 52, 53, 55, 57

D/A converters 2

DAC0 25, 50

DAC1 25
DAC2 25
DAC3 25
DADAT2 53
DADAT3 53
DADATH0 51
DADATH1 52
DADATL0 50
DADATL1 52
Device drivers 35
Diagnostics 5
Digital codes 55
Digital I/O 3, 11, 26, 27, 46, 47, 48, 57
DIOCNTRL 48
DIOP0 46
DIOP1 47
DIOP2 48
Direct Memory Access 5
DIVIDER 42
DMA 2, 5, 11, 15, 39
DMA buffer 61
DMA level 13
Expansion slots 23
External Trigger 26, 38
Ground 25
Handshake protocol 31
Hardware strobes 9
Initialization 56
Input lines 27
Input range 18
Inputs 2
Interface 3, 8
Interrupt 39, 59
Interrupt level 13
Interrupts 2, 5, 8
J8 slot 23
Mode 0 29
Mode 1 30
Mode 2 31
Multiplexer 7
Offset binary code 55
PC-22 6
PC-30 1
PC-39 1
PC-68 6
PC-81 6
Polled I/O 2, 5, 10, 59
PPI 11, 28
Prescaler 19, 41, 58
Sample and hold 7
Sampling rate 58
Single channel DMA 60
Single conversion 2
Software 3
Software strobes 9
Source impedance 27
Specifications 2
SSTB 39

STBC 39

Throughput 5

Timing and control 8

TMRCTR 42, 43

Turbo C 3

Turbo Pascal 3

Uncommitted counter/timer 10, 19

Unipolar 2

User configurable counter/timer 43

USR_CNT 43

-12 V 26