User manual for

PC-30PGL and PC-30PGH

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

Second edition.

November 1990

December 1990 Printing

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 provided along with the board. The use of these allows access to all board functions. These drivers are described accompanying manual "User Manual for PC-30 Driver Software".

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 DIP switch 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.

Preface

This manual is written for users of the PC-30PGL and PC-30PGH 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-30PG 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-30PGL and PC-30PGH. 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 - Compatibility.

- Appendix B discusses the compatibility of the PC-30B, PC-30C, PC-30D, PC-30DS and PC-30PG to each other, and older series boards.

Appendix C - PC-30 Component Layout.

- Appendix C contains layout diagrams of the PC-30PG.

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-30PGL and PC-30PGH boards are full size, low cost, high accuracy analog and digital I/O board for the IBM PC/AT and compatible series of computers. The PC-30PG features software programmable gain on its analog input channels, and is a development from the well known PC-30C, PC-30C and PC-30D family, and is fully compatible with these boards. The PC-30PG is also fully compatible with the older PC-26/PC-30/PC-39 series of boards.

The PC-30PG is available in two versions:

i. PC-30PGL. The analog input channel gain is programmable to either 1, 10, 100 or 1000.

ii. PC-30PGH. The analog input channel gain is programmable to either 1, 2, 4 or 8.

The PC-30PG is an integral part of the PC-30 range of boards. The full range of boards consists of the following:

- PC-30B. The PC-30B forms the basis of the range, and features 16 analog input channels with 30 KHz throughput, two 12 bit D/A outputs, two 8 bit D/A outputs and 24 digital I/O lines. Advanced DMA circuitry allows multi-channel operations at full throughput.

- PC-30C. This board has all the features of the PC-30B, but features an A/D throughput of 100 KHz.

- PC-30D. The PC-30D is a development of the PC-30C, with A/D throughput of 200 KHz. It is designed for use in PC/AT and AT compatible PC's. In addition to improved throughput, the PC-30D also contains FIFO buffers for A/D data, for improved operation in conjunction with multi-tasking operating system such as OS/2.

- PC-30DS. The PC-30DS is a development of the PC-30D, and is identical in operation and capabilities to the PC-30D, except for the addition of simultaneous sample and hold operation on all 16 input channels. The available analog input ranges of the PC-30DS have also been modified to support this capability.

- PC-30DS/4. The PC-30DS/4 is a low cost version of the PC-30DS, with only four simultaneously sampled input channels. In all other respects it is identical to the PC-30DS.

1.2. Features

The PC-30PG can be plugged into any slot of a PC/AT or compatible.

1.2.1 Programmable gain

The PC-30PG features software programmable gain. The gain of each of the 16 input channels can be independently set to either 1, 2, 4 or 8 (PC-30PGH), or 1, 10, 100 or 1000 (PC-30PGL). The gain for each channel is stored in the PC-30's internal gain memory. In addition, the PC-30PG's instrumentation amplifier can be configured to provide differential inputs.

1.2.2 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) 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.

The PC-30 contains logic which allows any sequence of channels, up to a sequence length of 31, to be selected and sampled under hardware control. This allows full throughput to be achieved even when converting multiple input channels.

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. 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.
- Number of A/D inputs: 16 single ended or 8 differential.

- A/D throughput rate: 200 KHz.

1.2.3 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. 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.

1.2.4 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. There are three bidirectional eight bit digital I/O ports, which can each be used in a variety of operating modes.

1.2.5 Interface logic

The PC-30 is accessed via I/O operations performed by the host processor. Of the 13 bit address received by the board, the most significant 8 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 board can be selected to be located anywhere between 0000 (hex) and 1FE0 (hex).

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

1.3. Software support

Supplied with the PC-30 board is a set of real time device drivers for use with a wide variety of

software. These device drivers are written in C, and are callable from the following languages:

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

In addition to the software supplied with the board, other software is also available:

1.3.1 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.

1.3.1.1 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.

1.3.1.2 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). In this case, the processing power of the host PC system is of no consequence, and the throughput of the PC-30 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 supplied with 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.

Note also that the throughput of the PC-30 is unaffected by the number of channels which are being converted, as long as the sequence of channels is less than 31. Longer sequences cannot use DMA, and must use program transfer techniques.

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 DEMO1.EXE program on the driver disk. This program will execute the PC-30 diagnostics routine, and should print out the following message:

PC-30 Driver Version 1.05
PC-30 diagnostics report the following:

PC-30PG found, operating correctly.

iv. If the "found" message is displayed, then you can run the program DEMO2.EXE on the distribution disk. This program displays the voltage on all the PC-30 input channels, and allows you to vary the analog outputs. Note that the program assumes that the PC-30 is in its factory standard configuration.

v. If the DEMO1 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-78

The PC-78 is a multi-interface adapter which allows the PC-30 to be connected to as many industry standard signal conditioning devices as possible, using only ribbon cable and standard insulation displacement connectors. It allows direct interface to:

- PC-77 screw terminal board.
- 3B series analog signal conditioning backplanes. Up to 16 isolated signal conditioning modules are supported.
- 5B series analog signal conditioning backplanes. Both standard and multiplexed backplanes are supported. Using the multiplexed backplanes, up 64 input and 64 output modules can be connected to a single PC-30. 5B series modules are also isolated.
- Industry standard digital I/O backplanes. The PC-78 can be connected directly to any digital signal conditioning panel which uses standard 24 line/50-way cable type connectors. Panels are available from a variety of vendors, supporting solid state relays, isolated input modules etc.

1.6.2 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.

1.6.3 PC-22

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

1.6.4 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.

[This page intentionally left blank.]

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-bit 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 or 8 differential input channels. This channel is selected by a channel address, obtained from the channel list. The channel list contains a list of channels to be converted. This may be up to 31 channels in length. When the end of the list is reached, the A/D loops back to the first channel in the list. This channel list may also be disabled for purposes of compatibility with older products.

- The programmable gain amplifier. The instrumentation/programmable gain amplifier amplifies the signal from the multiplexer by one of the four programmable gains. The gain is automatically selected by the gain code stored in the board's gain memory.

- 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.

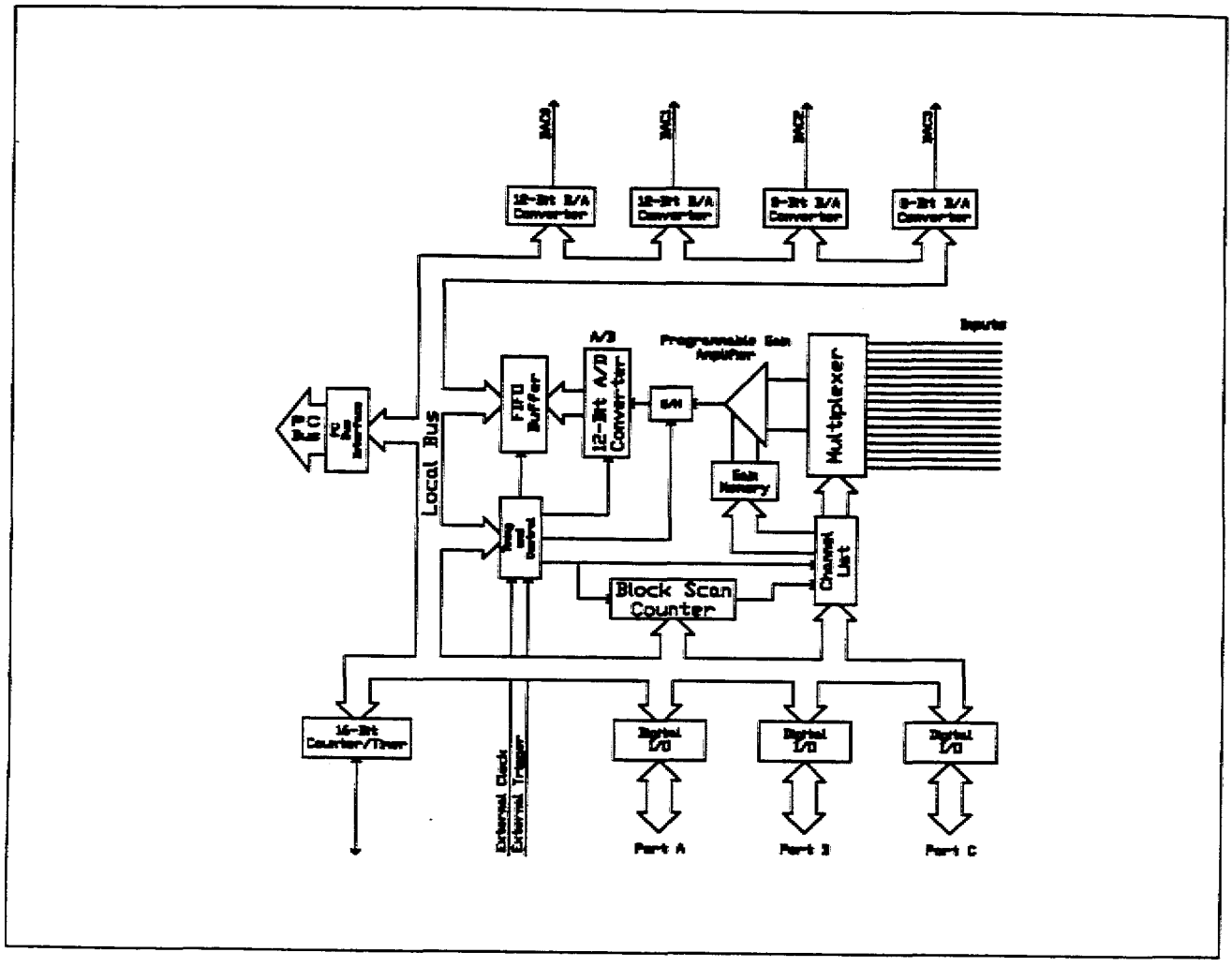


Figure 2.1.

- 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.

- The FIFO buffer. The FIFO (First In First Out) buffer is a temporary store for converted results. The FIFO stores data while the PC is performing other functions, such as memory refresh, and also stores data while the CPU changes memory buffers. Changing memory buffers allows the PC-30PG to perform DMA into the entire memory space of the PC without break. This is discussed in detail later in the chapter. The FIFO can store up to 16 samples.

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

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 a DIP switch.

ii. The generation of interrupts. Interrupts can be generated under one of three jumper selectable conditions:

   a) The end of each A/D conversion.

   b) The end of a DMA block.

   c) On each pulse from the uncommitted counter/timer.

iii. The generation of DMA signals. 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 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 jumper selected from one of two sources:

   a) The external clock, which is obtained from the J1 connector. This is a TTL level signal. Conversions are started on positive edges of this signal.

   b) The internal clock. This is derived from a crystal controlled oscillator, which operates at 2 MHz. The 2 MHz signal is then divided down by a programmable ratio.

The timing and control section contains four major subsections:

2.4.1 Crystal Oscillator

The master clock oscillator for the PC-30 is a crystal controlled 2 MHz oscillator. This is used to drive the clock prescaler. This is a 16 bit counter, the output from which drives the clock divider.

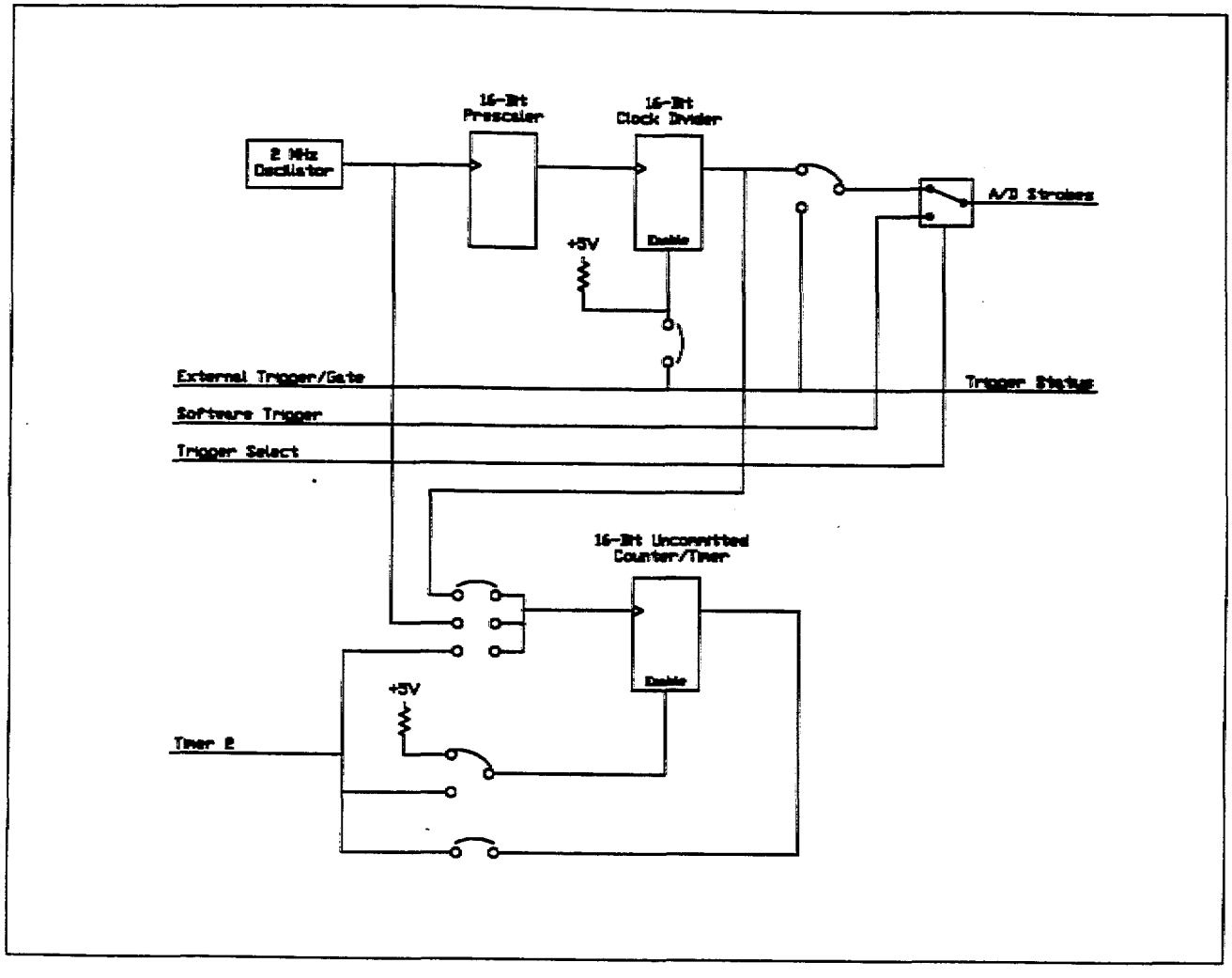


Figure 2.2.

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 can be configured by jumpers to count either

i. pulses from the 2 MHz master oscillator,

ii. pulses from the output of the clock divider, or

iii. pulses from an external source.

This allows the counter to be used for measuring frequency, or for generating a frequency. Counting can also be enabled from an external source, so allowing pulse width to be measured.

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 and trigger modes are selected, and the clock divider programmed.

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

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

   d) The required gain settings are written to the gain memory.

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 monitors the status register, until data is available.

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 3 KHz)

2.6.3 Single block DMA

DMA stands for Direct Memory Access. 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-30 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.

The number of samples which can be acquired in a single block is limited by the PC hardware to 32768. The PC-30PG however has special provision for chaining as many blocks as required. This is described in the next section.

2.6.4 Multi Block DMA

The PC-30PG has the ability to make use of two DMA channels. These two DMA channels are selected by jumpers on the PC-30PG board. Multi block DMA proceeds as follows:

i. The area of memory into which the samples are to be transferred is split into as many blocks of 64K as required. In practice the PC/AT can transfer up to 128K, but many operating systems provide data in blocks of 64K, as this is the segment size used by the processor in the PC/AT.

ii. The address of the first block is then programmed into the first channel of the DMA controller, the address of the second block is programmed into the second DMA channel, and the A/D sampling initiated. The PC-30PG then begins to transfer A/D samples into memory via the first DMA channel.

iii. When the DMA controller reaches the end of the first block, it generates a TC (Terminate Cycle) signal. The PC-30 uses this to change to the second DMA channel, and set a status bit indicating block completion. Samples are now transferred via the second DMA channel into the second block of memory. If the board is jumpered to generate an end of DMA block interrupt, it occurs at the same time as this changeover.

iv. Once the data acquisition program detects the end of block condition it programs the address of the next block into the first channel of the DMA controller. When the second channel of DMA reaches the end of its block, DMA swops back to the first channel. This process continues, with DMA channel continually being swopped, until all blocks have been filled.

Note that the DMA system must be reprogrammed before the memory block used by the DMA channel becomes full. Under normal circumstances this requires a response time of less than 0.16 seconds.

2.7. Block Mode triggering

The PC-30 can operate in one of two modes. These are block and normal modes.

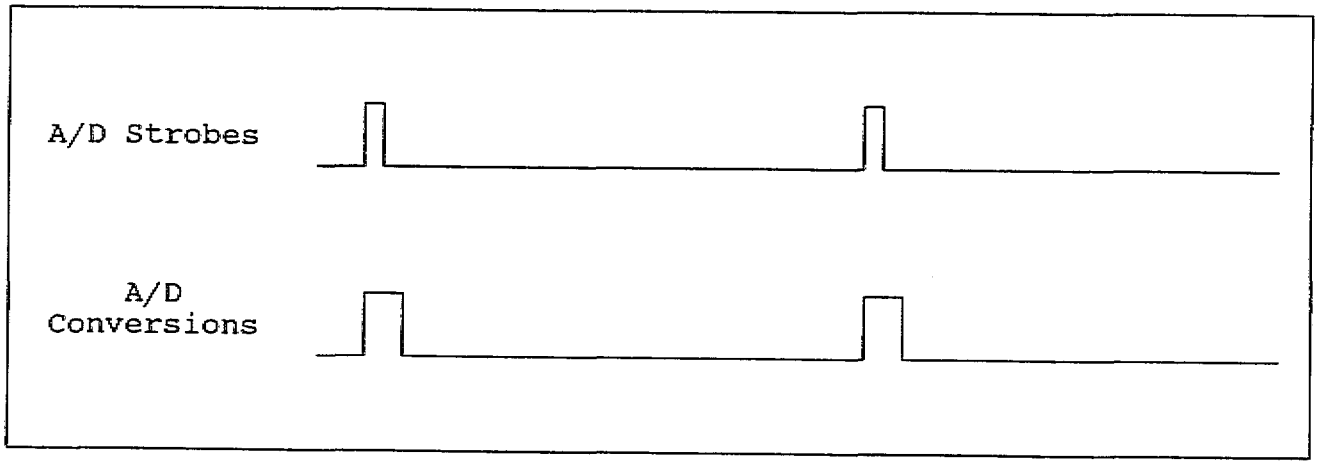


Figure 2.3.

2.7.1 Normal Mode

In normal mode, for each A/D strobe, one A/D conversion is performed. This is illustrated in figure 2.3. Note that in normal mode, the inputs are sampled prior to each individual A/D conversion.

2.7.2 Block Trigger Mode

In block trigger mode, for each A/D strobe, as many A/D conversions as are programmed into the block counter are performed. This value may range from 2 to 256. In block mode operation, the inputs are sampled only on each A/D strobe, and NOT prior to each individual A/D conversion.

Warning
Block mode operation is not available for PC-30PGL input channels operating at a gain of 1000.

Block mode operation is shown in figure 2.4, for a block count of 3.

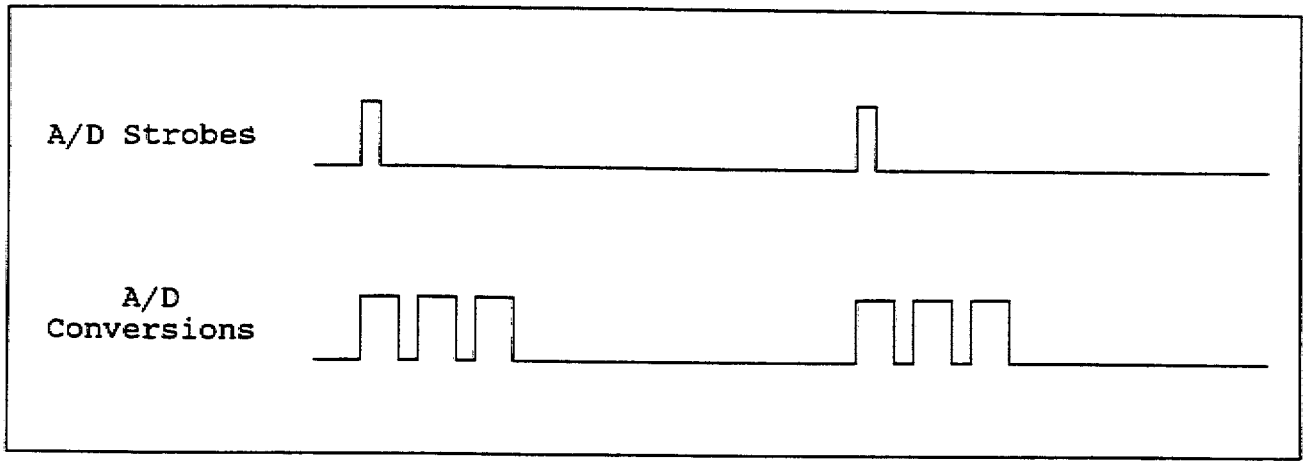


Figure 2.4.

2.8. Operation of the channel list

The channel list allows the PC-30 to convert a series of channels, without program intervention. For example, assume that we wish to convert the voltages on channel 2, 15 and 6, in that order.

We would write the following values to the channel register.

2

15

6

When the PC-30 is triggered, first channel 2 will be converted, then channel 15, then channel 6. The channel list will then loop round, and start at 2 again.

Note that the sequence of channels to be converted may be no longer than 31.

2.9. 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.

2.9.1 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.

2.9.2 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.

2.9.3 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.

2.9.4 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 PC-30 board

3.1. Introduction

The PC-30 board 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, the DMA level and the interrupt level used by the board can be set. As supplied by the factory, the base address is set to 700H, the DMA levels 5/6, and the interrupt level to 5. This allows operation in a standard PC/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 both the DMA request and the interrupt line are electronically disconnected from the PC bus unless specifically enabled by software, even if an interrupt level selection jumper or 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 or +/-5V. In addition, the board may be configured to provide either 16 single ended or 8 differential inputs.

iv. Uncommitted counter/timer. The PC-30 contains a single 16-bit counter, which can be used in a variety of roles, including counting events, generating pulses, and measuring frequency.

v. Output connector. The function of a power supply/ground pin can be set.

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
0H	on	on	on	on	on	on	on	on
20H	on	on	on	on	on	on	on	off
40H	on	on	on	on	on	on	off	on
60H	on	on	on	on	on	on	off	off
80H	on	on	on	on	on	off	on	on
A0H	on	on	on	on	on	off	on	off
C0H	on	on	on	on	on	off	off	on
E0H	on	on	on	on	on	off	off	off
100H	on	on	on	on	off	on	on	on
120H	on	on	on	on	off	on	on	off
140H	on	on	on	on	off	on	off	on
160H	on	on	on	on	off	on	off	off
180H	on	on	on	on	off	off	on	on
1A0H	on	on	on	on	off	off	on	off
1C0H	on	on	on	on	off	off	off	on
1E0H	on	on	on	on	off	off	off	off
200H	on	on	on	off	on	on	on	on
220H	on	on	on	off	on	on	on	off
240H	on	on	on	off	on	on	off	on
260H	on	on	on	off	on	on	off	off
280H	on	on	on	off	on	off	on	on
2A0H	on	on	on	off	on	off	on	off
2C0H	on	on	on	off	on	off	off	on
2E0H	on	on	on	off	on	off	off	off
300H	on	on	on	off	off	on	on	on
320H	on	on	on	off	off	on	on	off
340H	on	on	on	off	off	on	off	on
360H	on	on	on	off	off	on	off	off
380H	on	on	on	off	off	off	on	on
3A0H	on	on	on	off	off	off	on	off
3C0H	on	on	on	off	off	off	off	on
3E0H	on	on	on	off	off	off	off	off
400H	on	on	off	on	on	on	on	on
420H	on	on	off	on	on	on	on	off
440H	on	on	off	on	on	on	off	on
460H	on	on	off	on	on	on	off	off
480H	on	on	off	on	on	off	on	on
4A0H	on	on	off	on	on	off	on	off
4C0H	on	on	off	on	on	off	off	on
4E0H	on	on	off	on	on	off	off	off
500H	on	on	off	on	off	on	on	on
520H	on	on	off	on	off	on	on	off
540H	on	on	off	on	off	on	off	on
560H	on	on	off	on	off	on	off	off

Table 3.1. Base address Setting.

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
580H	on	on	off	on	off	off	on	on
5A0H	on	on	off	on	off	off	on	off
5C0H	on	on	off	on	off	off	off	on
5E0H	on	on	off	on	off	off	off	off
600H	on	on	off	off	on	on	on	on
620H	on	on	off	off	on	on	on	off
640H	on	on	off	off	on	on	off	on
660H	on	on	off	off	on	on	off	off
680H	on	on	off	off	on	off	on	on
6A0H	on	on	off	off	on	off	on	off
6C0H	on	on	off	off	on	off	off	on
6E0H	on	on	off	off	on	off	off	off
700H	on	on	off	off	off	on	on	on
720H	on	on	off	off	off	on	on	off
740H	on	on	off	off	off	on	off	on
760H	on	on	off	off	off	on	off	off
780H	on	on	off	off	off	off	on	on
7A0H	on	on	off	off	off	off	on	off
7C0H	on	on	off	off	off	off	off	on
7E0H	on	on	off	off	off	off	off	off
800H	on	off	on	on	on	on	on	on
820H	on	off	on	on	on	on	on	off
840H	on	off	on	on	on	on	off	on
860H	on	off	on	on	on	on	off	off
880H	on	off	on	on	on	off	on	on
8A0H	on	off	on	on	on	off	on	off
8C0H	on	off	on	on	on	off	off	on
8E0H	on	off	on	on	on	off	off	off
900H	on	off	on	on	off	on	on	on
920H	on	off	on	on	off	on	on	off
940H	on	off	on	on	off	on	off	on
960H	on	off	on	on	off	on	off	off
980H	on	off	on	on	off	off	on	on
9A0H	on	off	on	on	off	off	on	off
9C0H	on	off	on	on	off	off	off	on
9E0H	on	off	on	on	off	off	off	off
A00H	on	off	on	off	on	on	on	on
A20H	on	off	on	off	on	on	on	off
A40H	on	off	on	off	on	on	off	on
A60H	on	off	on	off	on	on	off	off
A80H	on	off	on	off	on	off	on	on
AA0H	on	off	on	off	on	off	on	off
AC0H	on	off	on	off	on	off	off	on
AE0H	on	off	on	off	on	off	off	off

Table 3.2. Base address Setting (cont).

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
B00H	on	off	on	off	off	on	on	on
B20H	on	off	on	off	off	on	on	off
B40H	on	off	on	off	off	on	off	on
B60H	on	off	on	off	off	on	off	off
B80H	on	off	on	off	off	off	on	on
BA0H	on	off	on	off	off	off	on	off
BC0H	on	off	on	off	off	off	off	on
BE0H	on	off	on	off	off	off	off	off
C00H	on	off	off	on	on	on	on	on
C20H	on	off	off	on	on	on	on	off
C40H	on	off	off	on	on	on	off	on
C60H	on	off	off	on	on	on	off	off
C80H	on	off	off	on	on	off	on	on
CA0H	on	off	off	on	on	off	on	off
CC0H	on	off	off	on	on	off	off	on
CE0H	on	off	off	on	on	off	off	off
D00H	on	off	off	on	off	on	on	on
D20H	on	off	off	on	off	on	on	off
D40H	on	off	off	on	off	on	off	on
D60H	on	off	off	on	off	on	off	off
D80H	on	off	off	on	off	off	on	on
DA0H	on	off	off	on	off	off	on	off
DC0H	on	off	off	on	off	off	off	on
DE0H	on	off	off	on	off	off	off	off
E00H	on	off	off	off	on	on	on	on
E20H	on	off	off	off	on	on	on	off
E40H	on	off	off	off	on	on	off	on
E60H	on	off	off	off	on	on	off	off
E80H	on	off	off	off	on	off	on	on
EA0H	on	off	off	off	on	off	on	off
EC0H	on	off	off	off	on	off	off	on
EE0H	on	off	off	off	on	off	off	off
F00H	on	off	off	off	off	on	on	on
F20H	on	off	off	off	off	on	on	off
F40H	on	off	off	off	off	on	off	on
F60H	on	off	off	off	off	on	off	off
F80H	on	off	off	off	off	off	on	on
FA0H	on	off	off	off	off	off	on	off
FC0H	on	off	off	off	off	off	off	on
FE0H	on	off	off	off	off	off	off	off
1000H	off	on	on	on	on	on	on	on
1020H	off	on	on	on	on	on	on	off
1040H	off	on	on	on	on	on	off	on
1060H	off	on	on	on	on	on	off	off

Table 3.3. Base address Setting (cont).

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1080H	off	on	on	on	on	off	on	on
10A0H	off	on	on	on	on	off	on	off
10C0H	off	on	on	on	on	off	off	on
10E0H	off	on	on	on	on	off	off	off
1100H	off	on	on	on	off	on	on	on
1120H	off	on	on	on	off	on	on	off
1140H	off	on	on	on	off	on	off	on
1160H	off	on	on	on	off	on	off	off
1180H	off	on	on	on	off	off	on	on
11A0H	off	on	on	on	off	off	on	off
11C0H	off	on	on	on	off	off	off	on
11E0H	off	on	on	on	off	off	off	off
1200H	off	on	on	off	on	on	on	on
1220H	off	on	on	off	on	on	on	off
1240H	off	on	on	off	on	on	off	on
1260H	off	on	on	off	on	on	off	off
1280H	off	on	on	off	on	off	on	on
12A0H	off	on	on	off	on	off	on	off
12C0H	off	on	on	off	on	off	off	on
12E0H	off	on	on	off	on	off	off	off
1300H	off	on	on	off	off	on	on	on
1320H	off	on	on	off	off	on	on	off
1340H	off	on	on	off	off	on	off	on
1360H	off	on	on	off	off	on	off	off
1380H	off	on	on	off	off	off	on	on
13A0H	off	on	on	off	off	off	on	off
13C0H	off	on	on	off	off	off	off	on
13E0H	off	on	on	off	off	off	off	off
1400H	off	on	off	on	on	on	on	on
1420H	off	on	off	on	on	on	on	off
1440H	off	on	off	on	on	on	off	on
1460H	off	on	off	on	on	on	off	off
1480H	off	on	off	on	on	off	on	on
14A0H	off	on	off	on	on	off	on	off
14C0H	off	on	off	on	on	off	off	on
14E0H	off	on	off	on	on	off	off	off
1500H	off	on	off	on	off	on	on	on
1520H	off	on	off	on	off	on	on	off
1540H	off	on	off	on	off	on	off	on
1560H	off	on	off	on	off	on	off	off
1580H	off	on	off	on	off	off	on	on
15A0H	off	on	off	on	off	off	on	off
15C0H	off	on	off	on	off	off	off	on
15E0H	off	on	off	on	off	off	off	off

Table 3.4. Base address Setting (cont).

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1600H	off	on	off	off	on	on	on	on
1620H	off	on	off	off	on	on	on	off
1640H	off	on	off	off	on	on	off	on
1660H	off	on	off	off	on	on	off	off
1680H	off	on	off	off	on	off	on	on
16A0H	off	on	off	off	on	off	on	off
16C0H	off	on	off	off	on	off	off	on
16E0H	off	on	off	off	on	off	off	off
1700H	off	on	off	off	off	on	on	on
1720H	off	on	off	off	off	on	on	off
1740H	off	on	off	off	off	on	off	on
1760H	off	on	off	off	off	on	off	off
1780H	off	on	off	off	off	off	on	on
17A0H	off	on	off	off	off	off	on	off
17C0H	off	on	off	off	off	off	off	on
17E0H	off	on	off	off	off	off	off	off
1800H	off	off	on	on	on	on	on	on
1820H	off	off	on	on	on	on	on	off
1840H	off	off	on	on	on	on	off	on
1860H	off	off	on	on	on	on	off	off
1880H	off	off	on	on	on	off	on	on
18A0H	off	off	on	on	on	off	on	off
18C0H	off	off	on	on	on	off	off	on
18E0H	off	off	on	on	on	off	off	off
1900H	off	off	on	on	off	on	on	on
1920H	off	off	on	on	off	on	on	off
1940H	off	off	on	on	off	on	off	on
1960H	off	off	on	on	off	on	off	off
1980H	off	off	on	on	off	off	on	on
19A0H	off	off	on	on	off	off	on	off
19C0H	off	off	on	on	off	off	off	on
19E0H	off	off	on	on	off	off	off	off
1A00H	off	off	on	off	on	on	on	on
1A20H	off	off	on	off	on	on	on	off
1A40H	off	off	on	off	on	on	off	on
1A60H	off	off	on	off	on	on	off	off
1A80H	off	off	on	off	on	off	on	on
1AA0H	off	off	on	off	on	off	on	off
1AC0H	off	off	on	off	on	off	off	on
1AE0H	off	off	on	off	on	off	off	off
1B00H	off	off	on	off	off	on	on	on
1B20H	off	off	on	off	off	on	on	off
1B40H	off	off	on	off	off	on	off	on
1B60H	off	off	on	off	off	on	off	off

Table 3.5. Base address Setting (cont).

Base Address	DIP Switch Setting							
	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1B80H	off	off	on	off	off	off	on	on
1BA0H	off	off	on	off	off	off	on	off
1BC0H	off	off	on	off	off	off	off	on
1BE0H	off	off	on	off	off	off	off	off
1C00H	off	off	off	on	on	on	on	on
1C20H	off	off	off	on	on	on	on	off
1C40H	off	off	off	on	on	on	off	on
1C60H	off	off	off	on	on	on	off	off
1C80H	off	off	off	on	on	off	on	on
1CA0H	off	off	off	on	on	off	on	off
1CC0H	off	off	off	on	on	off	off	on
1CE0H	off	off	off	on	on	off	off	off
1D00H	off	off	off	on	off	on	on	on
1D20H	off	off	off	on	off	on	on	off
1D40H	off	off	off	on	off	on	off	on
1D60H	off	off	off	on	off	on	off	off
1D80H	off	off	off	on	off	off	on	on
1DA0H	off	off	off	on	off	off	on	off
1DC0H	off	off	off	on	off	off	off	on
1DE0H	off	off	off	on	off	off	off	off
1E00H	off	off	off	off	on	on	on	on
1E20H	off	off	off	off	on	on	on	off
1E40H	off	off	off	off	on	on	off	on
1E60H	off	off	off	off	on	on	off	off
1E80H	off	off	off	off	on	off	on	on
1EA0H	off	off	off	off	on	off	on	off
1EC0H	off	off	off	off	on	off	off	on
1EE0H	off	off	off	off	on	off	off	off
1F00H	off	off	off	off	off	on	on	on
1F20H	off	off	off	off	off	on	on	off
1F40H	off	off	off	off	off	on	off	on
1F60H	off	off	off	off	off	on	off	off
1F80H	off	off	off	off	off	off	on	on
1FA0H	off	off	off	off	off	off	on	off
1FC0H	off	off	off	off	off	off	off	on
1FE0H	off	off	off	off	off	off	off	off

Table 3.6. Base address Setting (cont).

3.2. Changing the Configuration

The DIP switch and 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 jumper or DIP switch settings follow the procedure below :

i. Switch off the computer.

ii. Remove the board.

iii. Change the required jumpers or DIP switch settings.

iv. Replace the board in the PC.

v. Power up, and run a program (such as DEMO1), which executes the PC-30 diagnostics routines.

3.3. Bus Interface Configuration

3.3.1 Base address

The base address setting is controlled by the DIP switch on the board. As supplied by the factory, the address is set to 700H. The board occupies 32 consecutive locations. Tables 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 shows the DIP switch settings for a particular base address.

3.3.2 Interrupt level

The interrupt setting block consists of jumper locations 26 to 38. The interrupt level may be set to level 2, 3, 5, 7, 10, 11, 12, 14 or 15.

3.3.2.1 Jumper settings

The interrupts jumper settings for the "8-bit" (PC) interrupts are described in table 3.7, and the additional "16-bit" jumpers in table 3.8.

Warning
Only one interrupt level may be installed at any one time. Selecting more than one interrupt level may cause permanent damage to your PC-30.

Interrupt Level	Jumpers			
	W26	W27	W28	W33
2	Out	Out	Out	In
3	In	Out	Out	Out
5	Out	In	Out	Out
7	Out	Out	In	Out
None	Out	Out	Out	Out

Table 3.7. Interrupt jumper settings.

Interrupt Level	Jumpers				
	W34	W35	W36	W37	W38
10	Out	Out	Out	Out	In
11	Out	Out	Out	In	Out
12	Out	Out	In	Out	Out
14	Out	In	Out	Out	Out
15	In	Out	Out	Out	Out
None	Out	Out	Out	Out	Out

Table 3.8. Additional interrupt jumper settings.

3.3.2.2 Selection of interrupt level

In a standard PC, interrupts are allocated as follows :

level 3	Used by COM2: (if installed)
level 4	Used by COM1: (if installed)
level 5	Used by fixed disks (XT and AT)
level 7	Used by LPT1: (if installed)
level 10	Unused
level 11	Unused
level 12	Unused
level 14	Used by fixed disks (AT)
level 15	Unused

An interrupt level which is not already used must be selected. Level 5 is the factory default. Note that unless the interrupts are specifically enabled by software, the interrupt output from the board is tri-stated (does not have any effect on the PC bus).

3.3.3 Interrupt Source

The source of PC-30 interrupts can be jumpered to be one of three events:

i. End of Conversion. In this case, an interrupt is generated each time that the PC-30 completes an A/D conversion.

ii. End of DMA block. This option is only available for the PC-30PG. In this case, an interrupt is generated on completion of each DMA block. This can be used to signal the end of an operation or to signal that the DMA controller must be reprogrammed in dual DMA channel "ping-pong" operations.

Note
End of DMA block interrupts are only generated if the PC-30PG registers are programmed correctly; see the description of the DMA mode bit for more information.

iii. On each pulse from the counter/timer. If this jumper option is selected, an interrupt is generated on each pulse from the uncommitted counter/timer. The counter/timer is usually configured to generate a constant frequency, as discussed in chapter 5.

Jumper settings for the interrupt source are shown in table 3.9.

Interrupt Source	Jumpers		
	W52	W53	W54
End of Conversion	In	Out	Out
End of DMA Block	Out	Out	In
Counter/Timer Pulse	Out	In	Out

Table 3.9. Interrupt source jumper settings.

3.3.4 DMA level

On the PC-30PG, the DMA setting block consists of jumper locations 39 to 47, and 49. The PC-30PG supports either normal single DMA channel, or dual DMA channel "ping-pong" mode DMA operations. DMA levels may be set to levels 5, 6 or 7. The DMA jumper settings for single channel operation are shown in table 3.10, and those for dual channel mode in table 3.11. Default setting is for dual channel mode on channels 5 and 6.

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

DMA Level	Jumpers									
	W39	W40	W41	W42	W43	W44	W45	W46	W47	W49
5	Out	Out	Out	In	Out	Out	In	Out	Out	Out
6	Out	Out	In	Out	Out	In	Out	Out	Out	Out
7	In	Out	Out	Out	In	Out	Out	Out	Out	Out

Table 3.10. PC-30PG single channel DMA jumper settings.

DMA Levels Pri/Sec	Jumpers									
	W39	W40	W41	W42	W43	W44	W45	W46	W47	W49
5/6	Out	In	Out	In	Out	Out	In	Out	In	Out
6/7	In	Out	In	Out	Out	In	Out	In	Out	Out

Table 3.11. PC-30PG dual channel DMA jumper settings.

3.3.4.1 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)
level 5	Unused
level 6	Unused
level 7	Unused

A DMA level which is not already used must be selected. Note that unless DMA is specifically enabled by software, the DMA outputs from the board are tri-stated (do not have any effect on the PC bus).

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 installed in the locations numbered 1 to 8 as shown in table 3.12, 3.13, 3.14 and 3.15.

DAC 0 Output Range	Jumpers	
	W3	W4
0 to +10V	Out	In
-10 to +10V	In	Out

Table 3.12. DAC 0 output ranges jumper settings.

DAC 1 Output Range	Jumpers	
	W2	W1
0 to +10V -10 to +10V	Out In	In Out

Table 3.13. DAC 1 output ranges jumper settings.

DAC 2 Output Range	Jumpers	
	W8	W7
0 to +10V -10 to +10V	In Out	Out In

Table 3.14. DAC 2 output ranges jumper settings.

DAC 3 Output Range	Jumpers	
	W5	W6
0 to +10V -10 to +10V	In Out	Out In

Table 3.15. DAC 3 output ranges jumper settings.

3.5. A/D configuration

Three aspects of A/D operation can be configured. These are the input mode (single ended or differential), the A/D input voltage range, and the A/D clock/triggering.

3.5.1 A/D input mode

The PC-30PG can provide either 16 single ended or eight differential inputs. The use of differential inputs is recommended in environments with high levels of electrical noise, when using long lines to connect the analog inputs, or for any input operating at a gain of greater than 10. Default configuration for the PC-30PG is differential inputs. Jumper settings are shown in table 3.16.

A/D Input Mode	Jumpers			
	W56	W57	W58	W59
Single Ended	In	Out	In	Out
Differential	Out	In	Out	In

Table 3.16 PC-30PG A/D Input mode settings.

3.5.2 A/D voltage range jumper settings

The A/D may be configured for input range. The A/D jumper settings for the PC-30PG are given in table 3.17. The default range setting is -5 to +5V.

A/D Range	Jumpers			
	W22	W23	W50	W51
-5 to +5V	In	Out	Out	In
0 to +10V	Out	In	In	Out

Table 3.17. PC-30PG A/D jumper settings.

3.5.3 A/D clock/trigger.

Figure 3.1 shows a partial schematic of the PC-30 clock/trigger system. Counter 0 of the 8254 serves as the A/D clock prescaler, and counter 1 as the A/D clock divider. Counter 2 is the uncommitted counter/timer. Jumpers 18, 19 and 20 are used to configure the source of A/D clock pulses. Three possible configurations are supported:

i. Internal clock. A/D clock pulses are obtained from the clock divider. This is the normal configuration. The external trigger input has no direct effect on the operation of the board, but can be read by software. The A/D clock frequency is the frequency of the 2 MHz master clock divided by the values programmed into the A/D prescaler and the A/D clock divider.

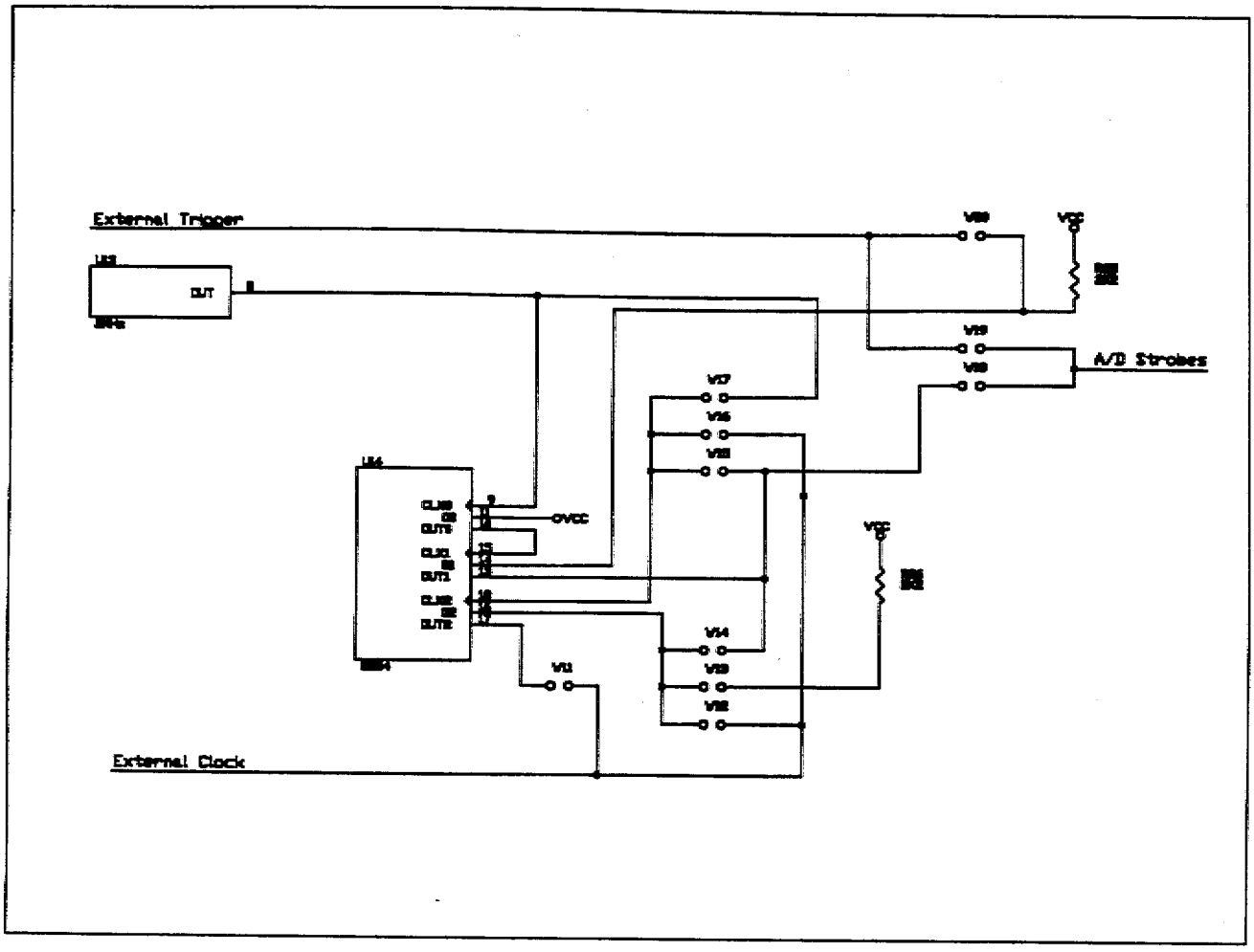


Figure 3.1. PC-30 clock generation schematic.

ii. Internal clock/external trigger. A/D clock pulses are obtained from the clock divider, and the external trigger input is a TTL compatible clock enable. In this case a 2K2 pull-up is attached to the trigger input. If the external trigger input is logically high, A/D clock pulses are produced. If the external trigger input is logically low, then clock pulses are not produced. The A/D clock frequency is the frequency of the 2 MHz master clock divided by the values programmed into the A/D prescaler and the A/D clock divider.

iii. External clock. A/D clock pulses are obtained from the external trigger input. In this case, the external trigger input serves directly as the source of A/D clock pulses. Note that in this case neither the A/D prescaler or clock divider have any effect on the sample rate.

Table 3.18 shows the A/D clock source jumper settings. The default setting is for A/D clock pulses obtained from the clock divider.

A/D Clock Configuration	Jumpers		
	W18	W19	W20
Internal clock	In	Out	Out
Internal clock/External trig.	In	Out	In
External clock	Out	In	Out

Table 3.18. A/D clock mode configuration jumper settings.

3.6. Uncommitted Counter/Timer

The PC-30 contains an uncommitted counter/timer, which can be used for a variety of tasks, including counting pulses, generating pulses, generating a constant frequency, and measuring frequency and period. Figure 3.1 shows a partial schematic of the PC-30 clock/trigger system.

Three aspects of the counter timer operation can be configured:

3.6.1 External clock line

The external clock lines can either be an input to, or output from, the counter/timer. This is configured by jumper 11, as shown in table 3.19. Default configuration is as an output.

Clock Line Configuration	Jumper
	W11
Input	Out
Output	In

Table 3.19. External clock line configuration.

3.6.2 Counter/timer clock source

The counter/timer can be configured to count one of three things:

i. Pulses from the master clock. In this case, the counter/timer counts pulses from the 2MHz master clock oscillator. This is generally used either to generate a frequency, or to measure period.

ii. Pulses from the clock divider. In this case, the counter timer counts at the rate set by the clock prescaler and clock divider. This clock is also derived from the master clock oscillator, but can be of a much lower frequency.

iii. Pulses from the external clock line. To use this option, the eternal clock line must be configured as an input, as described above. This option is generally used to count external events, or frequency.

The counter timer clock source is set by jumpers 15, 16 and 17, as described in table 3.20. Default setting is for counting the master clock.

Counter/Timer Clock Source Configuration	Jumpers		
	W15	W16	W17
Master clock	Out	Out	In
A/D Clock Divider	In	Out	Out
External clock input	Out	In	Out

Table 3.20. Counter/timer clock source jumper settings.

3.6.3 Counter/Timer enable

The counter/timer counts only if the gate input is at logical high. The input to this gate can be obtained from three different sources, as set by jumpers 12, 13 and 14.

i. Constantly enabled. The gate input to the counter timer can be constantly enabled. This generally is used for generating frequencies and pulses, or for counting events.

ii. Enabled from the output of the A/D clock divider. This mode is generally used for measuring frequency. The A/D divider is programmed to generate a pulse of a known length, and the counter/timer set to count input pulses. The count gives a measure of input frequency.

iii. Enabled from the external clock line. In this case the external clock must be configured as an input, as described above. This configuration is generally used to measure period, in which case the counter is configured to count master clock pulses.

The counter timer enable source is set by jumpers 12, 13 and 14, as described in table 3.21. Default setting is for constant enable.

Counter/Timer Enable Source Configuration	Jumpers		
	W12	W13	W14
Constant Enable	Out	In	Out
A/D Clock Divider	Out	Out	In
External clock input	In	Out	Out

Table 3.21. Counter/timer enable source jumper settings.

3.7. Output connector

Pin 17 of the PC-30 I/O connector can be configured as either a power supply (+5V) pin, or as digital ground. For compatibility with earlier versions of the PC-30, this is set to +5V operation at the factory. However, if you do not need the +5V supply, reconfiguring pin 17 to digital ground can provide improved noise performance.

Pin 17 configuration is set by jumpers 9 and 10, as described in table 3.22.

Pin 17 Connection	Jumpers	
	W9	W10
+5V	In	Out
Digital Gnd.	Out	In

Table 3.22. Connector pin 17 jumper settings.

Warning
Only jumper 9 or jumper 10 may be installed at any one time. Installing both jumpers will cause permanent irreparable damage to your PC-30.

3.8. Fixed jumper settings

Certain jumpers are fixed. These jumpers are set at the factory, and do not require modification. They are shown in table 3.23.

Jumper
W21
Always installed

Table 3.23. Fixed Jumper settings.

3.9. Default jumper settings

If your PC-30 has been used by a colleague, or you wish to test it, you may wish to return it to its factory default setting. This is given below:

Jumpers Installed (All other jumpers Out)

PC-30PG	W2, W3, W6, W7, W9, W11, W13, W17, W18, W21, W22, W27, W40, W42, W45, W47, W51, W52, W57, W59.

Chapter 4

Interconnections

4.1. Introduction

The PC-30PG plugs into any 16-bit IBM PC/AT expansion slot, 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-30PG boards plug into any slot of the IBM AT backplane. All communication to and from the host processor is carried out via this connector.

4.3. User connection

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

- 16 single ended or 8 differential 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 an optional +5V supply.

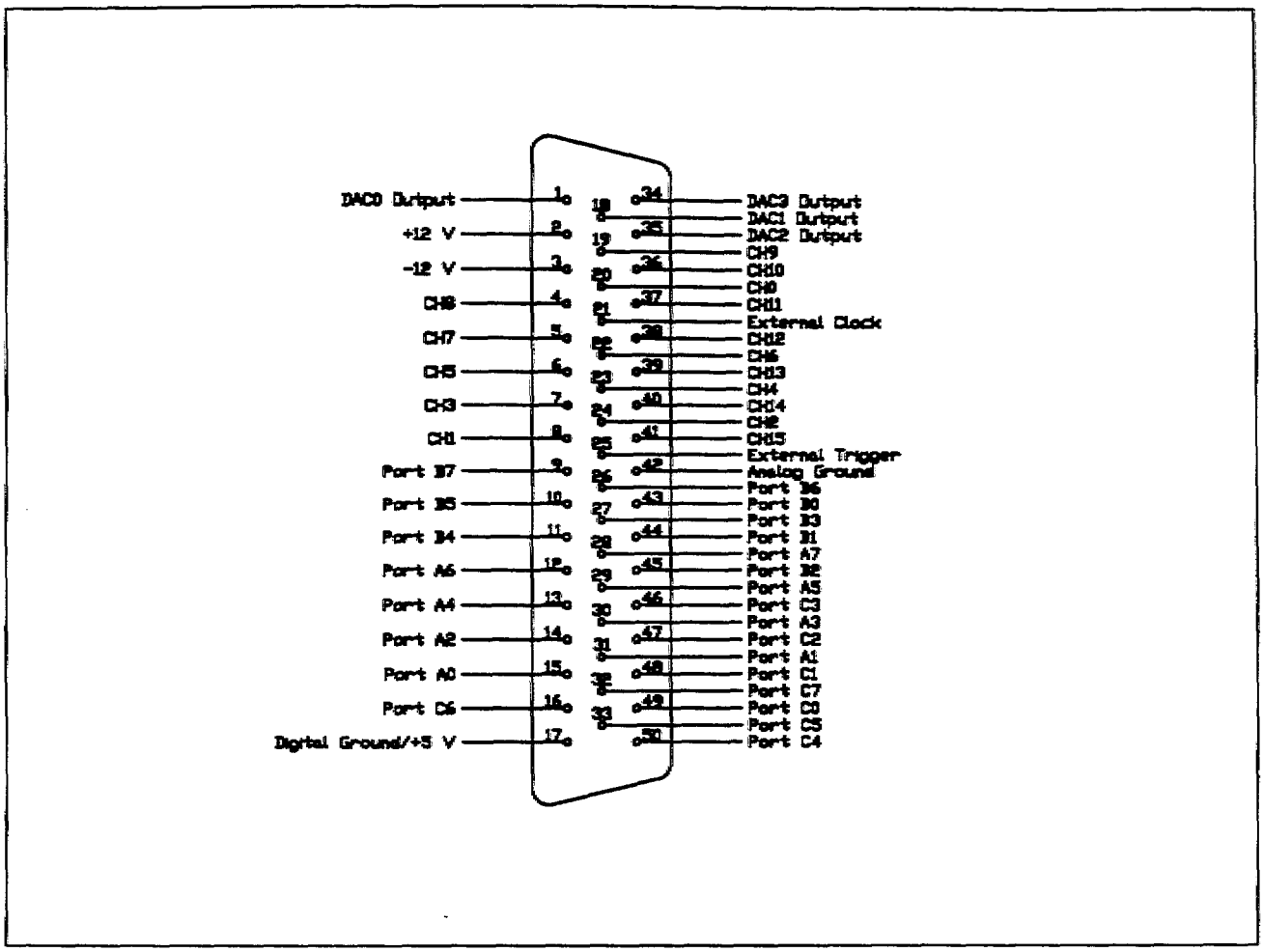


Figure 4.1. PC-30 connector, as seen from the rear of the PC.

Figure 4.1. 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.

4.3.1 Signal definitions.

i. CH0 - CH15. These are the analog input lines. Note that no more than +- 10V must be applied to these pins. In differential mode, channel 8 serves as the return line for input channel 0, channel 9 as that for input channel 1 etc.

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. Digital Ground/+5V. This line is jumper selectable to provide either a digital ground connection, or a source of +5V power. Digital ground is the ground return line for the digital inputs and outputs. Any digital circuitry tied to the digital lines should be referenced to these lines. It is internally connected to analog ground.

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 is jumper selectable to provide either a clock or trigger signal to the A/D, and may be read under software control. It is TTL compatible.

xiv. Ext. Clock. This line interfaces to the uncommitted counter/timer, and can be jumpered to perform a variety of functions, as described in the previous chapter. It may be configured either as an input or output. It is TTL compatible.

4.4. Analog I/O

4.4.1 Recommended analog input schemes.

Analog signals are input into the PC-30 either as single ended inputs or differential inputs.

Warning:
Overloading any analog input by more than 10% may cause other input channels to become inaccurate or noisy. For PC-30PGL inputs operating at maximum gain, this corresponds to an input voltage of 5.5 mV!!

4.4.1.1 Single ended inputs.

In single ended connections, input signals share a common low side, which is analog ground. Single ended inputs are shown diagrammatically in figure 4.2.

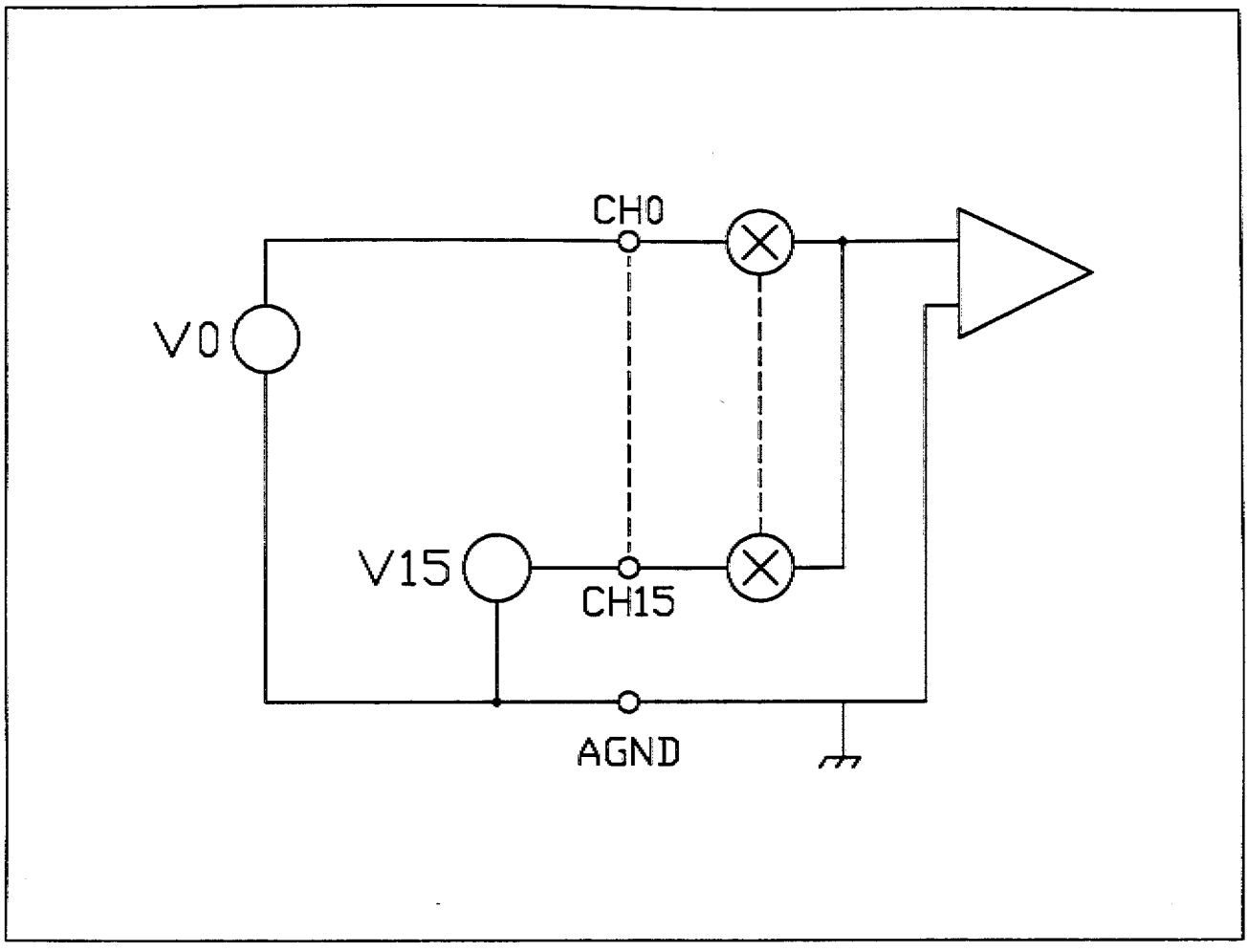


Figure 4.2. Single ended analog inputs.

This has the advantage of giving the maximum number of inputs. Its major disadvantage is the loss of common mode rejection obtainable from differential mode. Single ended inputs are very sensitive to noise, and should not be used with lead lengths of greater than 18 inches, or for inputs with a gain of greater than 10.

4.4.1.2 Differential inputs

In differential input mode, two multiplexer switches per channel are used, and the A/D converter measures the difference between the high and low input lines of each channel. Figure 4.3 shows differential inputs diagrammatically.

In differential mode, channels 0 and 8 form the high and low inputs of input channel 0, channels 1 and 9 that of input channel 1 etc.

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

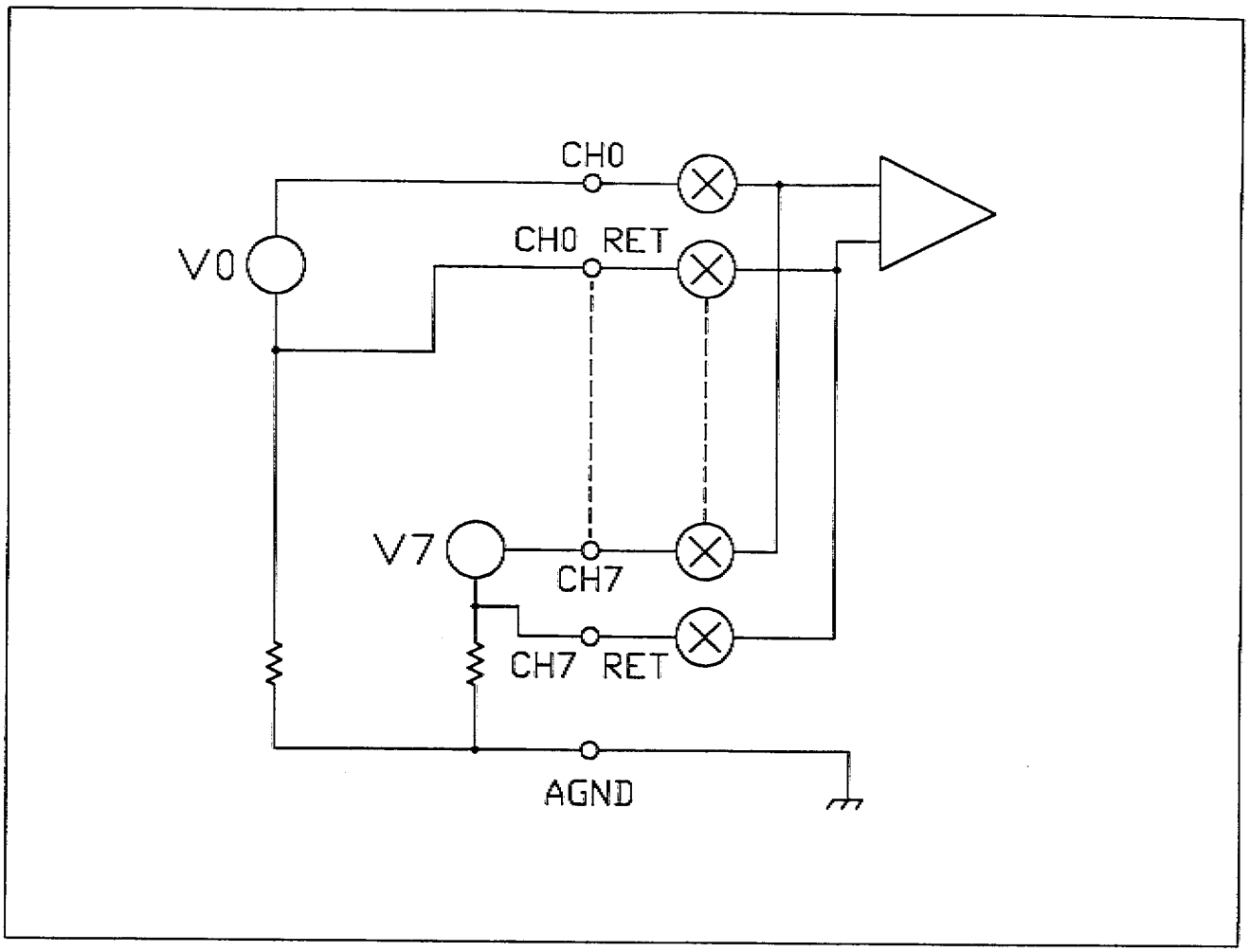


Figure 4.3. Differential analog inputs.

Note
In differential mode, all signal inputs to the PC-30 must be referred to ground. This can be done by connecting a 1 to 10 kilohm resistor from the low end of each input to ground.

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.

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 Grounding.

If user circuitry is connected to the PC-30 it is of the utmost importance to keep the digital and analog ground separate.

4.4.3.3 Input voltages.

To maintain stated accuracy, all inputs to the PC-30PG must be within 110% of full scale.

4.4.3.4 Source Impedance.

To maintain stated accuracy, all devices connected to the analog inputs of the PC-30 must have a source impedance of less than 1K Ohm.

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 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.
- 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.

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.

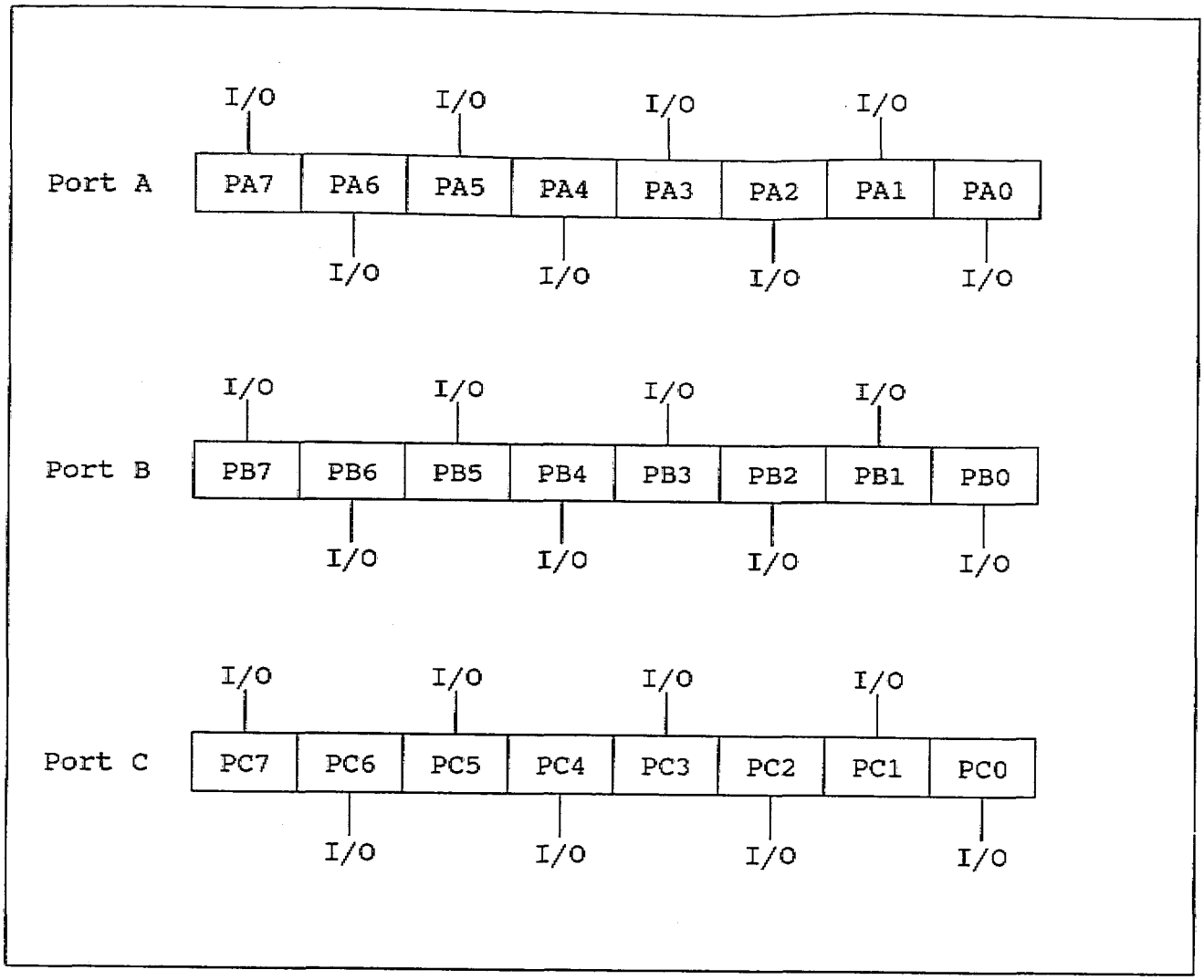


Figure 4.4. Mode 0 digital I/O.

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.

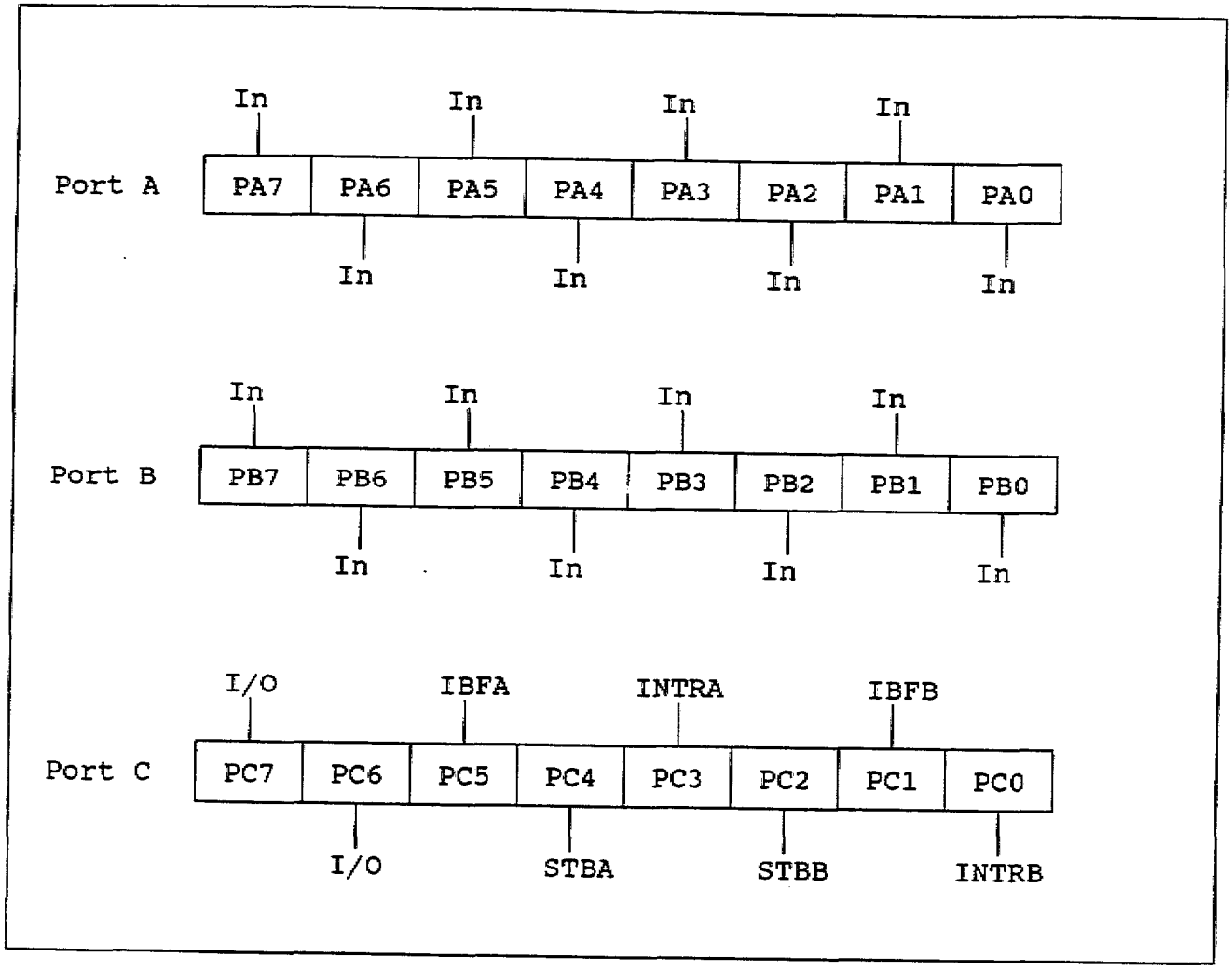


Figure 4.5. Mode 1 digital input.

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.

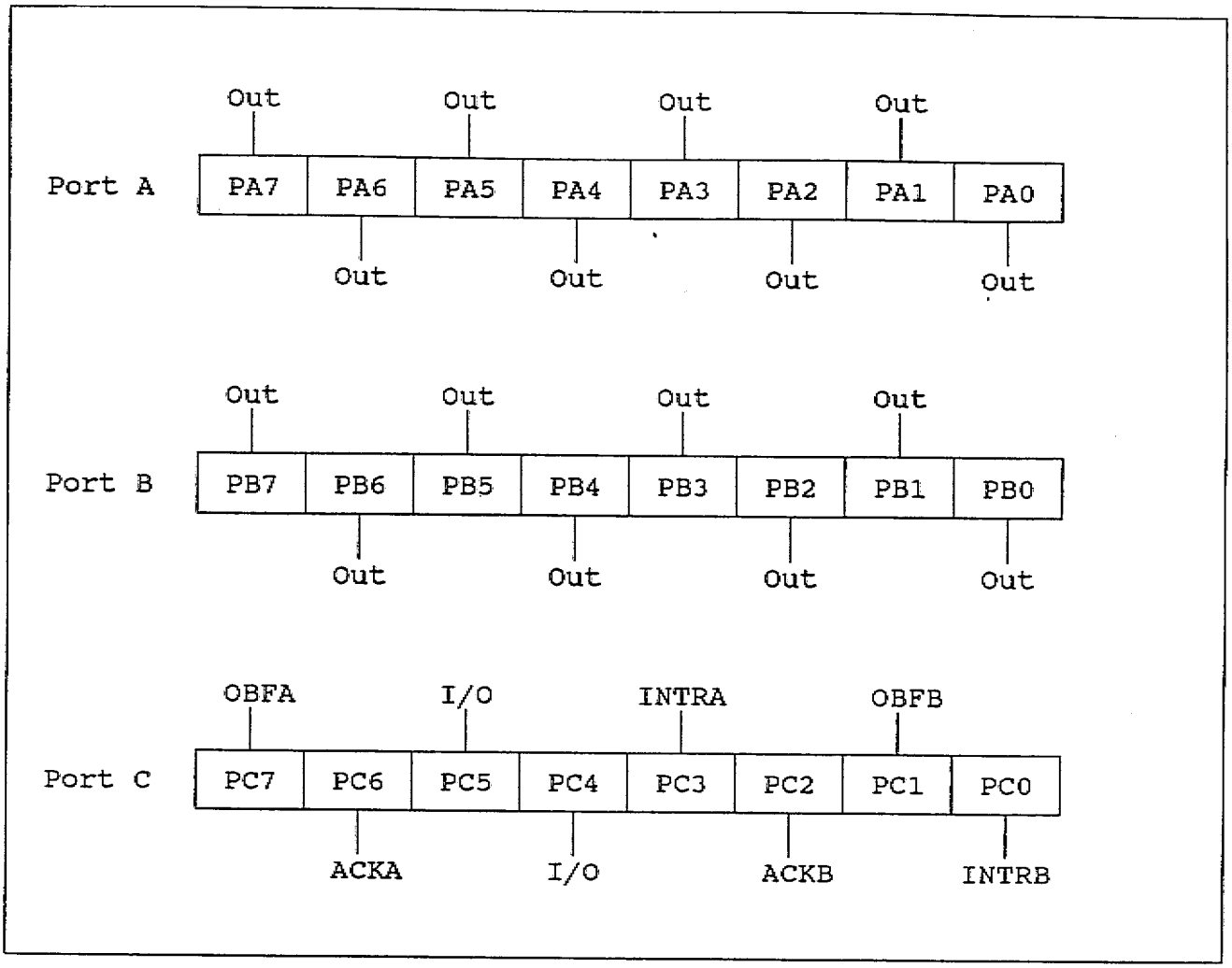


Figure 4.6. Mode 1 digital output.

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:

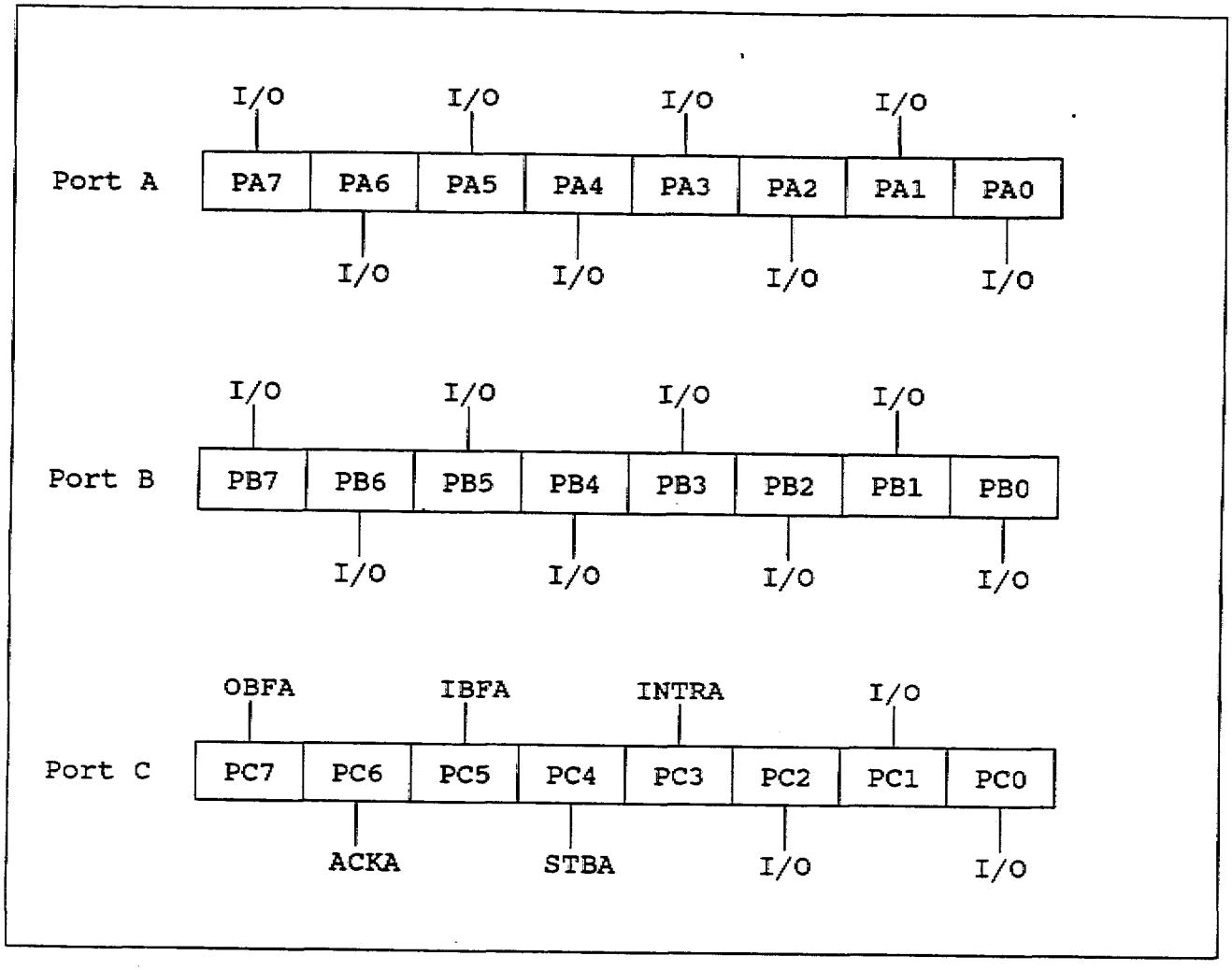


Figure 4.7. Mode 2 Digital I/O.

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

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 provided along with the board. The use of these allows access to all board functions. These drivers are described accompanying manual "User Manual for PC-30 Driver Software".

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 DIP switch 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)	Block Count (BLKCNT)
1	A/D Data/Status(ADDSR)	——
2	Control/Channel (ADCCR)	
3	A/D Mode Register (ADMDE)	
4	A/D Clock Prescaler (PRESCALER)	
5	A/D Clock Divider (DIVIDER)	
6	Uncommitted 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)
24	Gain Read (GAINREG)	Gain Memory 0 (GMEM0)
25	——	Gain Memory 1 (GMEM1)
26	——	Gain Memory 2 (GMEM2)
27	——	Gain Memory 3 (GMEM3)

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

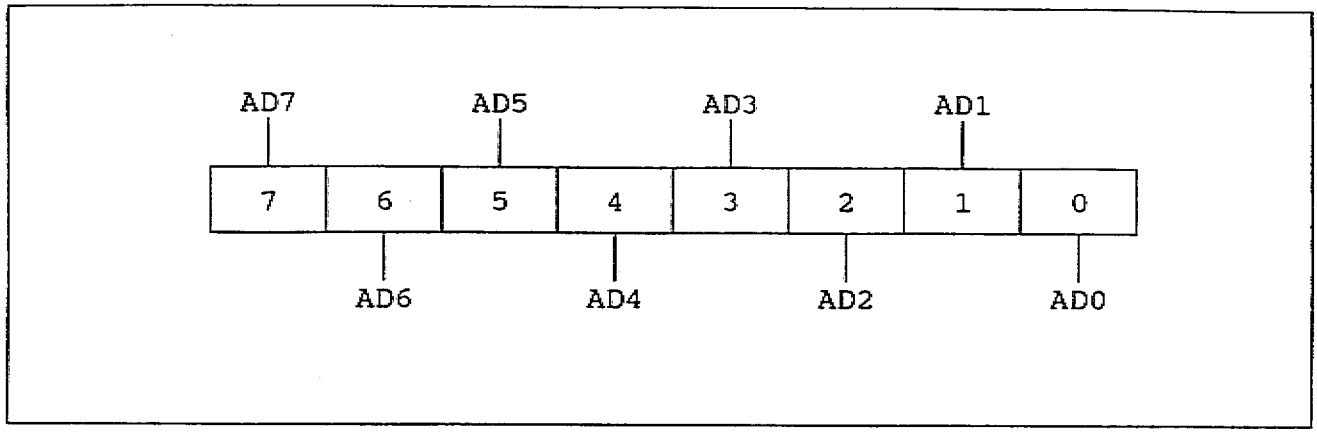


Figure 5.2. A/D data register (low byte).

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 BLKCNT - Block counter (offset 0)

The block counter indicates the number of A/D conversions to perform on each A/D strobe. Note that the A/D mode must be 1. The A/D mode is set in the ADMDE register, discussed later.

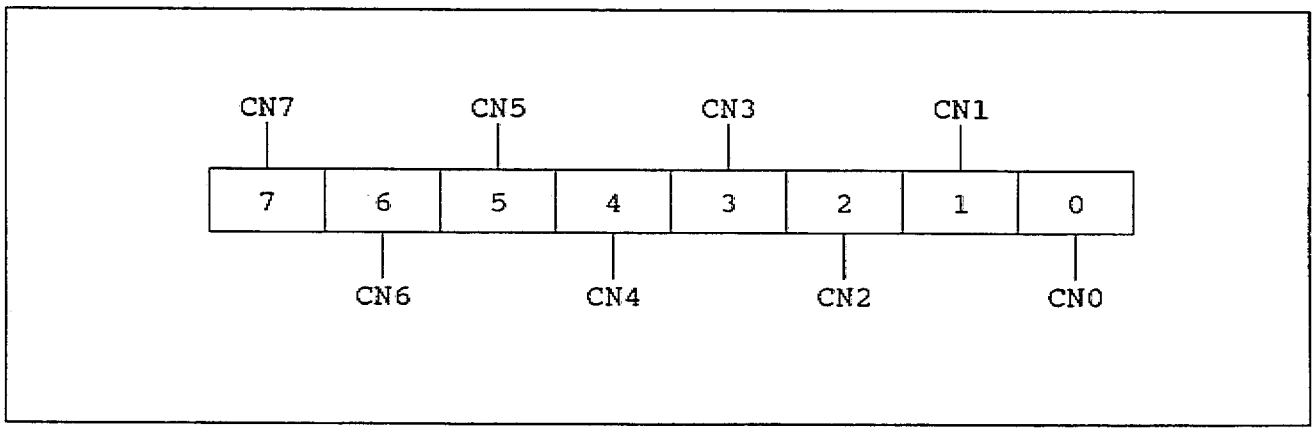


Figure 5.3. Block count register.

The value written to the BLKCNT register is calculated from :

257 - (Number of conversions per block)

For example, to perform 3 conversions per block, the value 254 must be written to the BLKCNT register. The value written to the BLKCNT register can range from 255 to 1 (block sizes of 2 to 256). Figure 5.3 shows the layout of this register.

Note
You must write the block count register prior to writing the first value to the channel list. Failure to do this may cause unpredictable operation.

Bits 7-0 - Block count value (CN)

These bits represent 257 - the number of conversions to perform per clock pulse.

5.2.3 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 5.4.

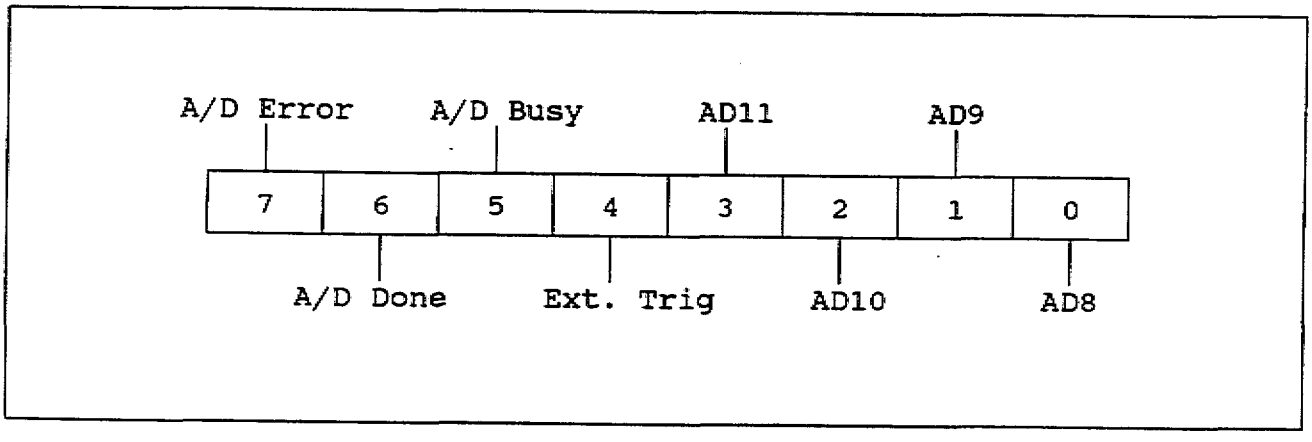


Figure 5.4. A/D data/status register.

Bit 7 - A/D error

Set when an error occurs during an A/D conversion. Such conditions can result from two causes: either the current conversion has been completed while data from the previous conversion has not been completely read (ADDATL low byte not read), which is a data overflow error, or an attempt has been made to initiate a new conversion while the current conversion is still in progress, which is a trigger error.

Note that on DMA operations it is normal to have a data overflow error on completion of the operation. There should not however be any error while the transfer is in progress.

Bit 7 is cleared on power up, and by writing a 1 to bit 2 of the ADMDE register.

Bit 6 - A/D done

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.

The A/D done bit remains set as long as there is data in the FIFO buffer.

Bit 5 - A/D busy

Set by an A/D strobe, and indicates that a conversion is in progress. The bit is cleared at end of conversion, and any trigger while the bit is set will cause an error condition.

Bit 4 - Ext. Trig

This bit reflects the status of the external trigger pin (pin 25 of the user connector).

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

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

5.2.4 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 5.5.

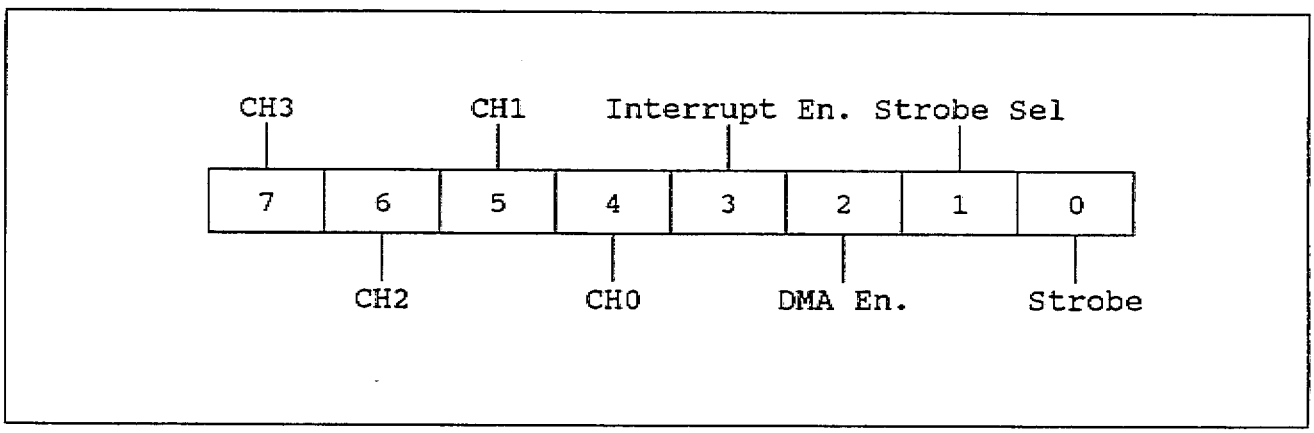


Figure 5.5. A/D control/channel register.

Bits 7-4 - Channel (CH)

These bits specify a four bit channel address. Depending on the A/D mode, the channel specified

by these four bits either replaces the current channel list, is added to the current channel list, or is ignored. The A/D mode is set in the ADMDE register, discussed in the next section. The channel list is discussed in detail in that section.

On read, these bits represent the channel at the head of the channel list, not the most recently written channel value. 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 under one of three conditions:

a) The end of each A/D conversion.

b) On each pulse from the uncommitted counter/timer.

c) The end of a DMA block.

The selection between these conditions is jumper selected, as described in chapter 3.

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.

Bit 2 - DMA enable.

DMA operations are controlled by the combination of bit 2, and the DMA mode bit in the ADMDE register. This interaction of the DMA enable bit and the DMA mode bit is explained in the description of the ADMDE register.

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 may be generated from the A/D clock prescaler/divider combination, or from an external source.

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.

5.2.5 ADMDE - A/D mode register (offset 3)

The ADMDE register contains A/D mode selection bits, the error reset control bit and the DMA mode control bit. The bit functions of the ADMDE register are shown in figure 5.6.

If your application requires 100% compatibility with older PC-26, PC-30 or PC-39 boards, you

should write the value 92(hex) to this register.

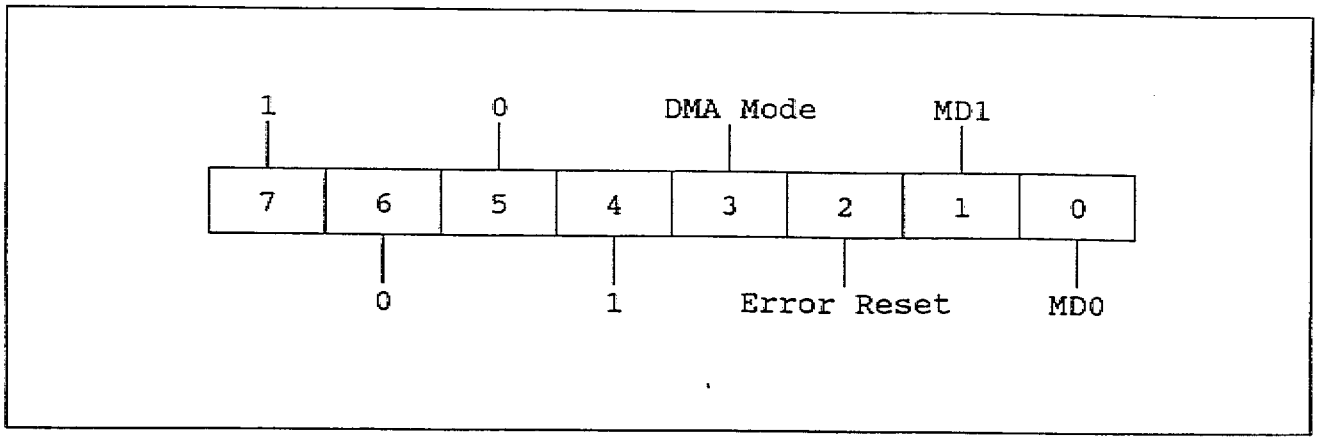


Figure 5.6. A/D mode register.

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 - DMA mode

The table below describes the effect of the various combination of it and the DMA enable bit in the ADCCR. There are four possible states:

Enable Bit	Mode Bit	Description
0	0	Disable DMA. The PC-30PG DMA request lines are held tri-state.

0	1	Swop on TC. This mode is used during dual channel DMA. On each TC pulse, the DMA enable bit is set, and the PC-30PG's internal block flip-flop is toggled. The PC-30PG hence begins DMA on the next DMA channel.
1	0	Normal DMA. This mode can be used for normal single channel DMA. TC pulses are ignored. This mode can hence be used for "circular buffer" applications where DMA runs continuously.
1	1	Terminate on TC. The PC-30PG switches itself to this state from the 0-1 state. On the next TC pulse, all DMA activities are halted. This is the recommended mode for single channel DMA transfers.

End of DMA block interrupts are produced by the DMA block toggle signal described above. Hence, to generate DMA interrupts, the PC-30PG must be programmed for "Swop on TC" mode operation.

Mode Bits		Channel List Mode	Trigger Mode	FIFO Mode
MD1	MD0			
0	0	Ignored	Normal	Enabled
0	1	Ignored	Burst	Enabled
1	0	Replace	Normal	Disabled
1	1	Add	—	Disabled

Table 5.1. A/D modes.

Bit 2 - Error reset.

Writing a 1 to this bit clears the error detection bit in the ADDSR.

Bits 1-0 - Mode (MD)

The two mode bits set the A/D conversion mode. The A/D conversion mode sets the channel list mode, selects the trigger mode and enables or disables the FIFO buffer. Table 5.1 details the effect

of the various modes.

i. Channel list mode.

There are three possible channel list modes: replace, ignore and add.

a) Replace. In replace mode, the current channel list is cleared, and the channel value written to the ADCCR register is inserted into the head of the list. This entry is then the only value in the channel list.

b) Ignore. In ignore mode, the channel bits in the ADCCR register have no effect. This is used to change control bits (for example the DMA and interrupt enable bits) without modifying the current channel list.

c) Add. In add mode, the channel value written is added to the current channel list.

ii. Trigger mode. Trigger mode was discussed extensively in chapter 2. Either normal or burst (block) mode can be selected. Note that in block mode, simultaneous sample and hold operation is active.

iii. FIFO mode. The FIFO buffer can be either enabled or disabled. Note that disabling the FIFO buffer resets it, destroying all data currently stored in it.

5.2.6 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.

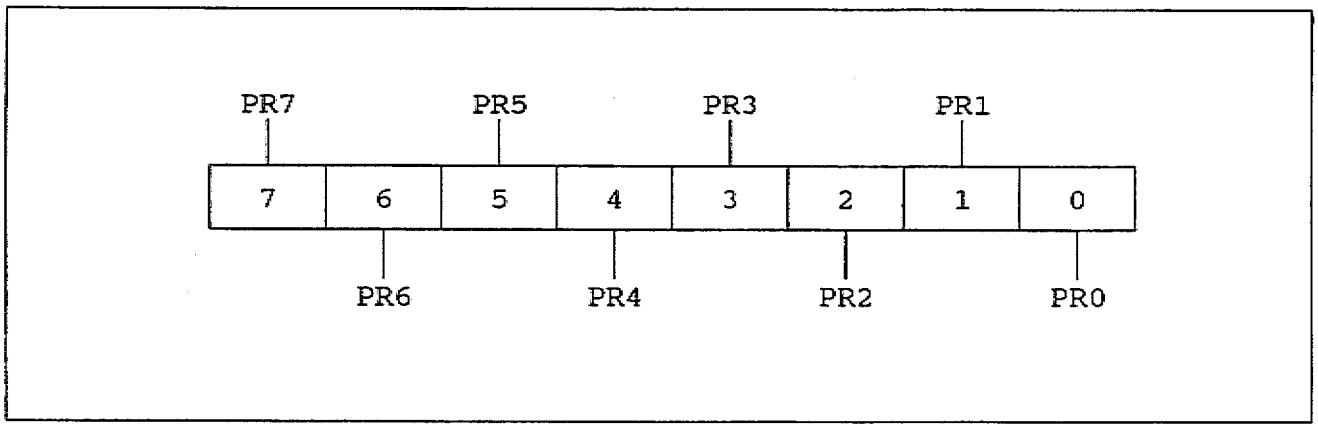


Figure 5.7. Prescaler register.

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 2 MHz master clock of the PC-30.

This register is register 0 (counter 0) of the 8254 on the PC-30 board. The bit layout of the prescaler

is shown in figure 5.7.

Bits 7-0 - Prescaler data (PR)

The prescaler register can be configured to read/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 read/write two bytes from the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

5.2.7 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 8254 on the PC-30 board. The bit layout of the divider is shown in figure 5.8.

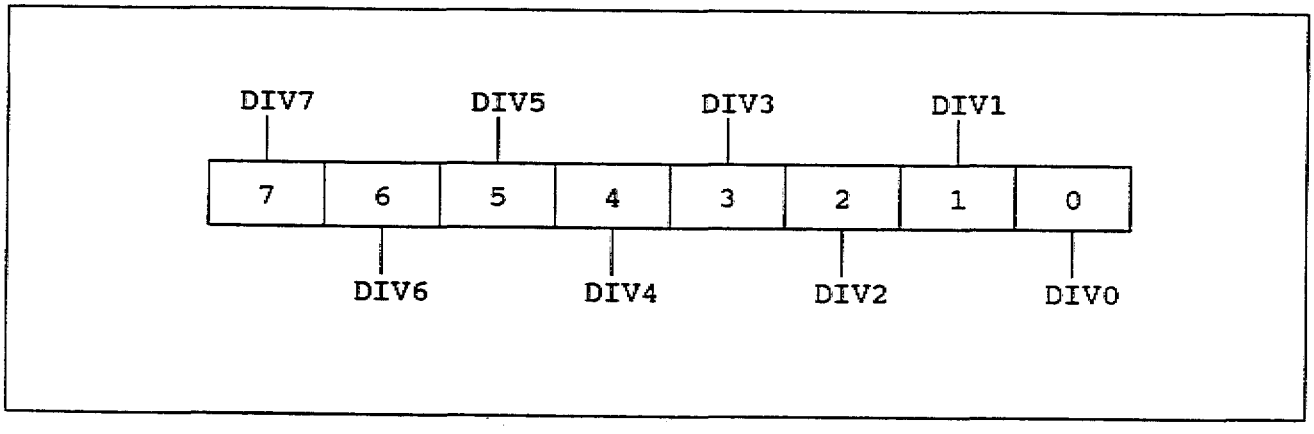


Figure 5.8. A/D clock divider register.

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

The clock divider register can be configured to read/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 read/write two bytes from the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

5.2.8 USR_CNT - User counter register (offset 6)

The register is used to program and to read 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 8254 on the PC-30 board. The bit layout of the USR_CNT register is shown in figure 5.9.

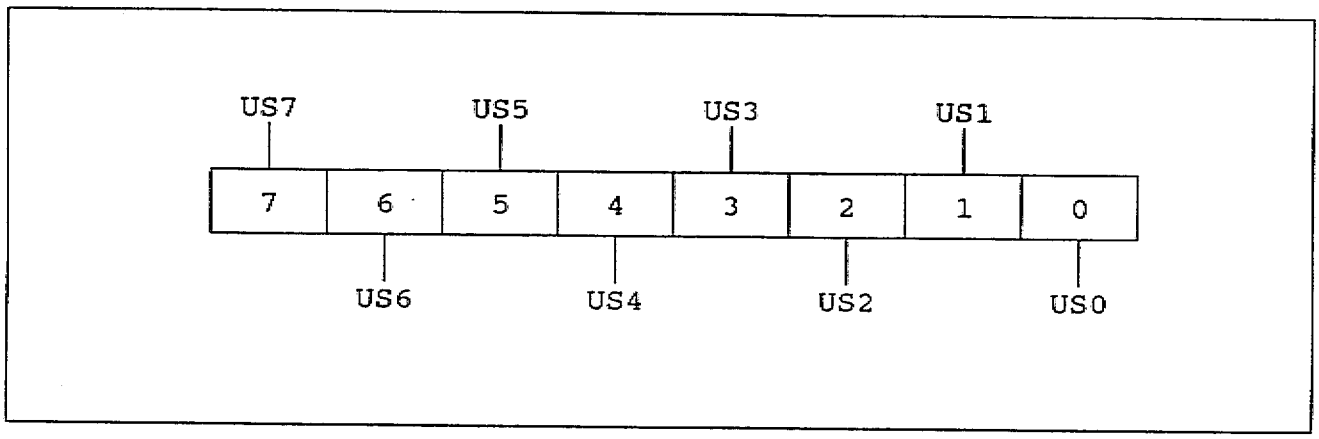


Figure 5.9. User counter register.

Bits 7-0 - User counter data (US)

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

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

5.2.9 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 8254 on the board. If you intend to program the 8254 extensively, you should obtain a copy of the data sheet for an 8254 type counter/timer.

The register layout is shown in figure 5.10.

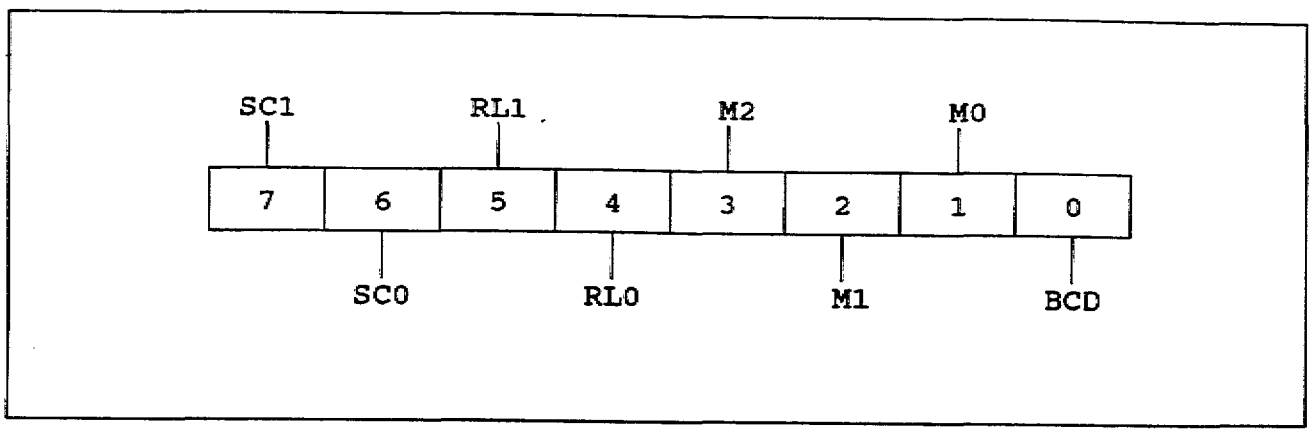


Figure 5.10. Timer controller register.

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 8254 data sheet for full details.
0	1	Read/write LSB only. All read/write operations will read/ write only the LSB (least significant byte) of the selected counter.
1	0	Read/write MSB only. All read/write operations will read/write only the MSB (most significant byte) of the selected counter.
1	1	Read LSB/MSB. Both the LSB and MSB of the counter are read/written. The LSB is read/written first, then the MSB. Note that both bytes must always be read/written. Reading/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) depends 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 or, if the board is jumpered for timer interrupts, to generate an interrupt after a programmable time. Mode 0 is also used to count events or frequency.
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, or to generate a periodic interrupt.
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.

5.2.10 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 5.11.

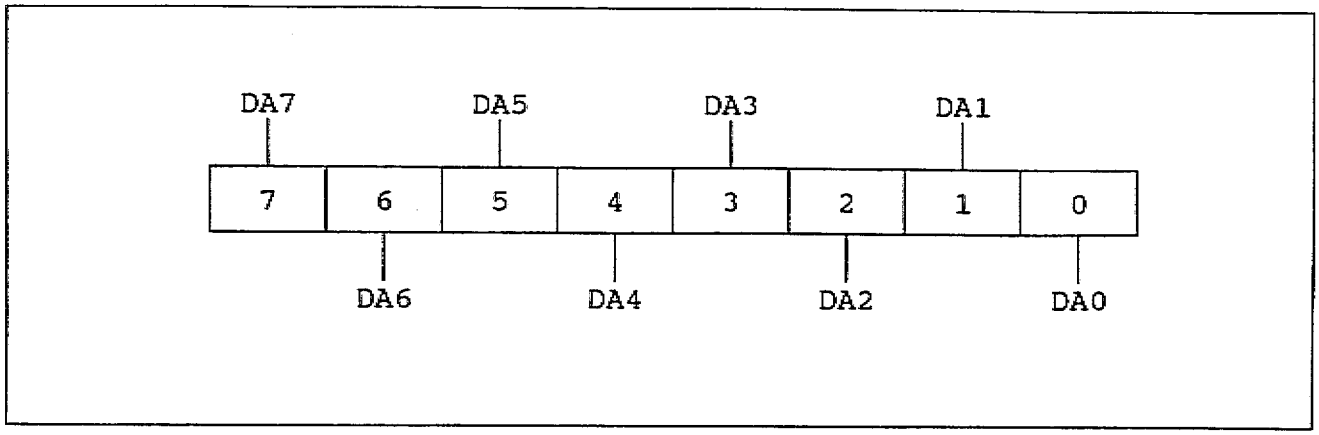


Figure 5.11. Digital I/O port 0 register.

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.

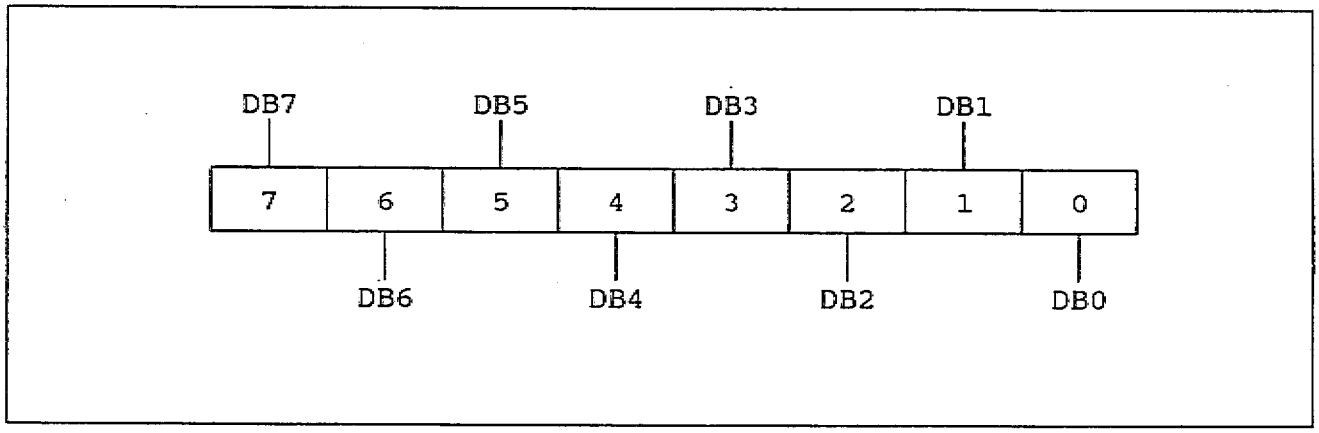


Figure 5.12. Digital I/O port 1 register.

5.2.11 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 5.12.

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.

5.2.12 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 5.13.

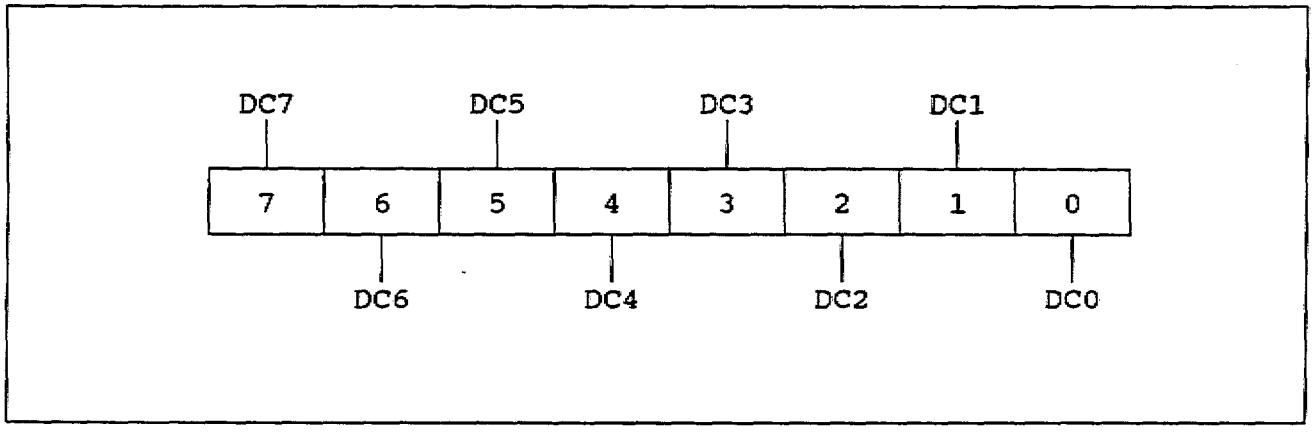


Figure 5.13. Digital I/O port 2 register.

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.

5.2.13 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 5.14. 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.

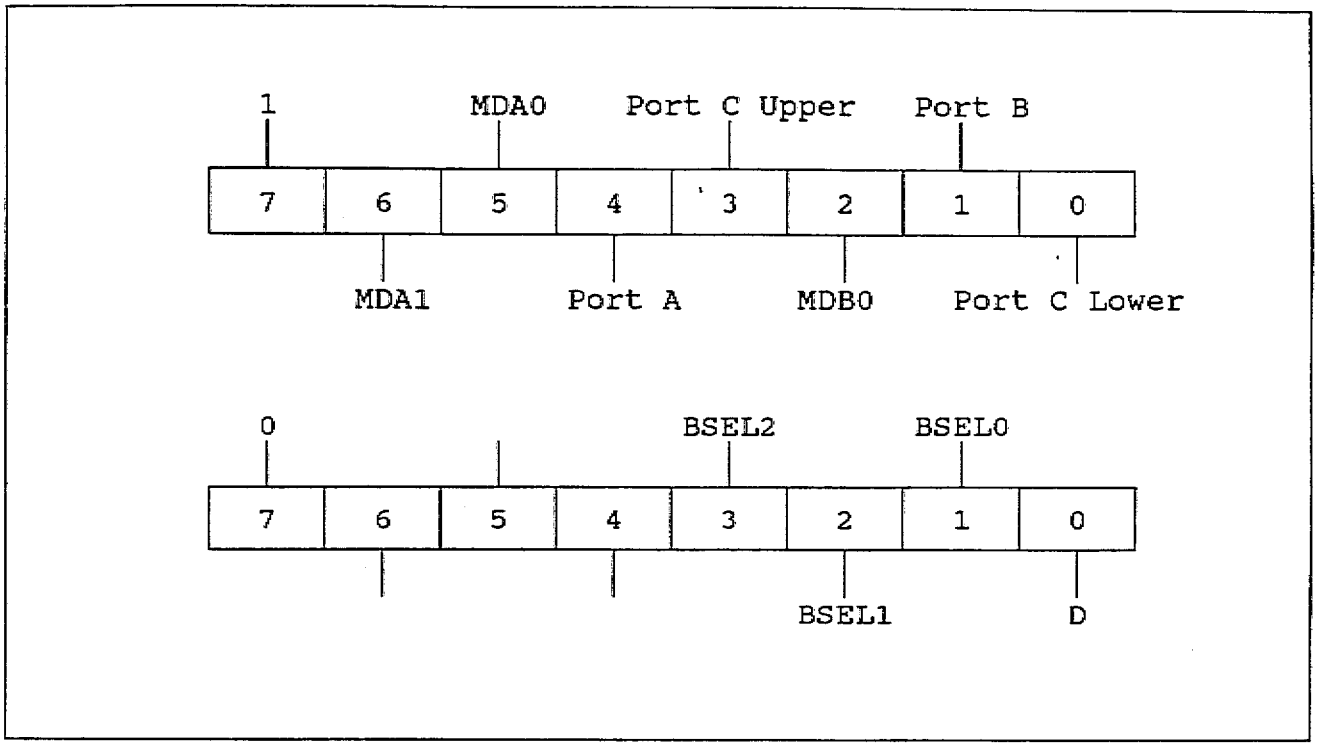


Figure 5.14. Digital I/O control register.

5.2.13.1 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.

5.2.13.2 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.

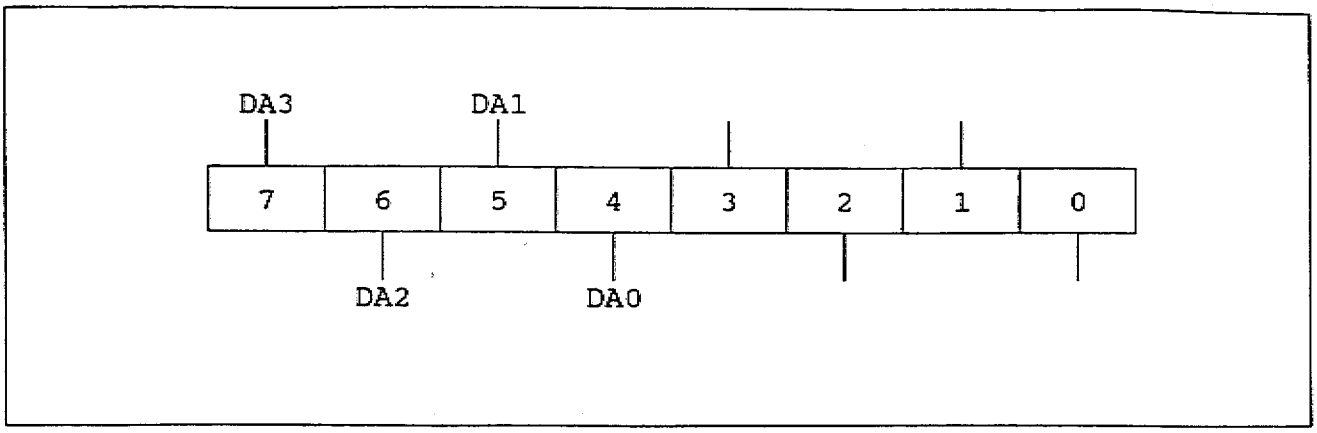


Figure 5.15. DAC0 low byte data register.

5.2.14 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 5.15 shows the register layout. Data is transferred to the output when this register is written.

Bits 7-4 - DAC0 data (DA)

These bits are the LSB of DAC0 data.

Bits 3-0 - Not used.

5.2.15 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 5.16.

Bits 7-0 - DAC0 data (DA)

These eight bits are the MSB of the DAC0 data.

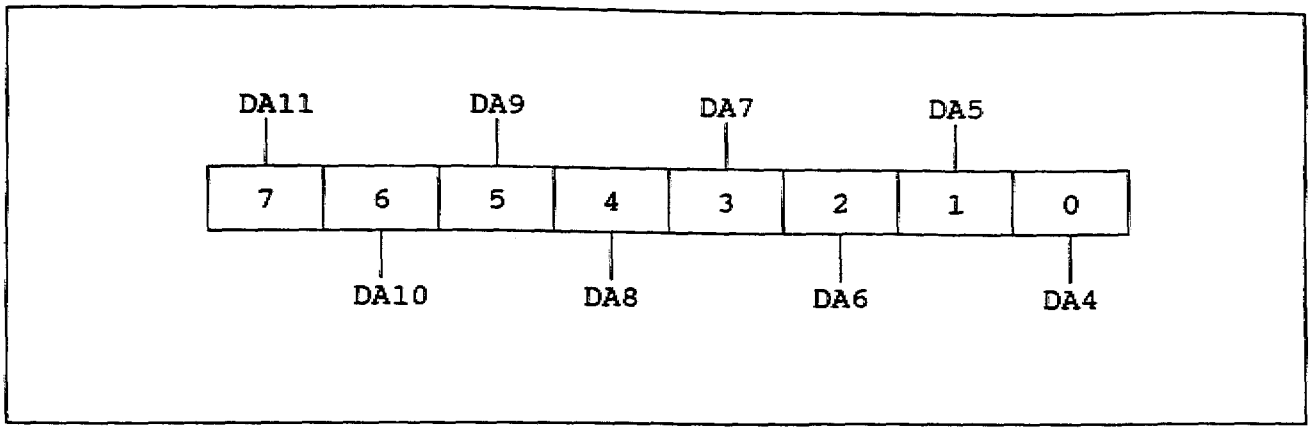


Figure 5.16. DAC0 high byte data register.

5.2.16 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 5.17 shows the register layout. Data is transferred to the output when this register is written.

Bits 7-4 - DAC1 data (DA)

These bits are the LSB of DAC1 data.

Bits 3-0 - Not used.

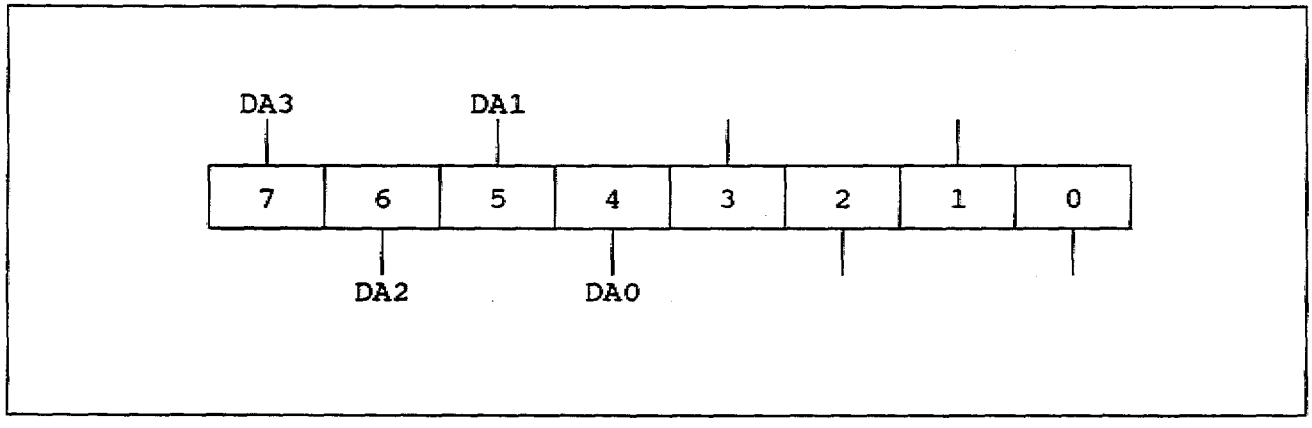


Figure 5.17. DAC1 low byte data register.

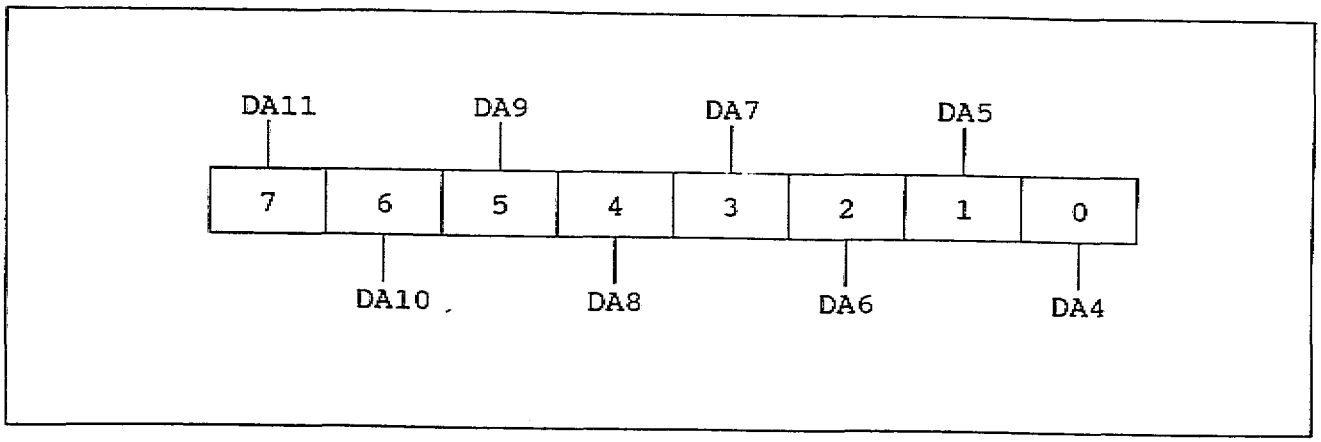


Figure 5.18. DAC1 high byte data register.

5.2.17 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 5.18.

Bits 7-0 - DAC1 data (DA)

These eight bits are the MSB of the DAC1 data.

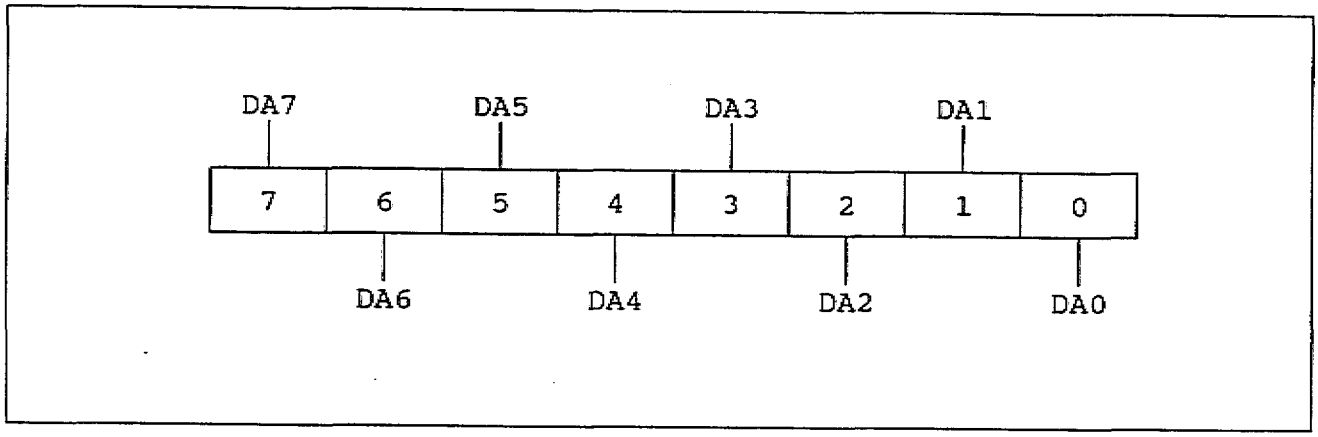


Figure 5.19. DAC2 data register.

5.2.18 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 5.19.

Bits 7-0 - DAC2 data (DA)

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

5.2.19 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 5.20.

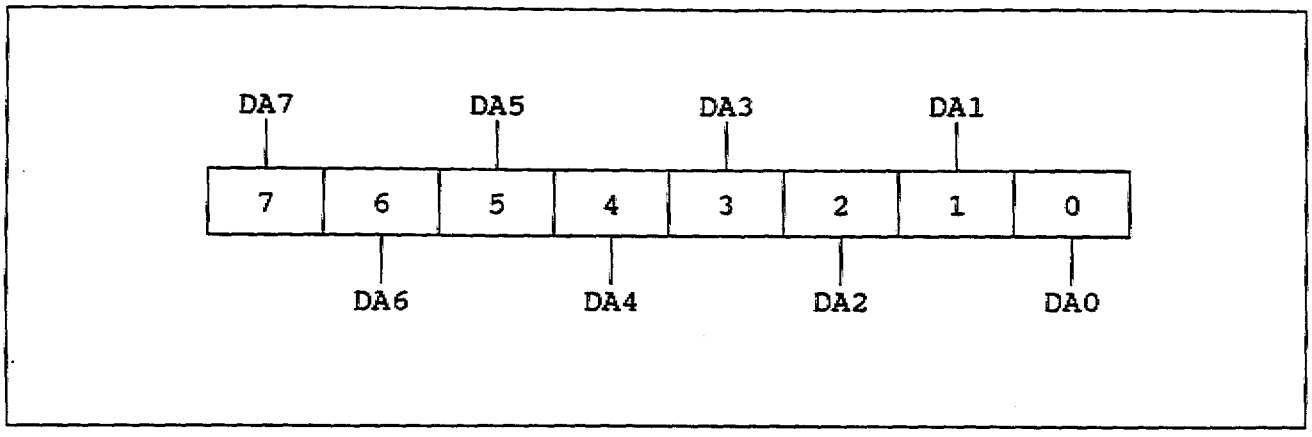


Figure 5.20. DAC3 data register.

Bits 7-0 - DAC3 data (DA)

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

5.2.20 GAINREG - Gain read back register (offset 24)

The GAINREG register reflects the gain setting for the current input channel. The current input channel can be found be reading the ADCCR register. The layout of this register is shown in figure 5.21.

Channel gain is set in the GMEM0, GMEM1, GMEM2 and GMEM3 registers.

Bits 1-0 - Gain (GB and GA)

These bits reflect the gain setting of the current channel.

Bits 7-3 - Unused

Bits 7-3 are not used. The result of reading these bits is undefined.

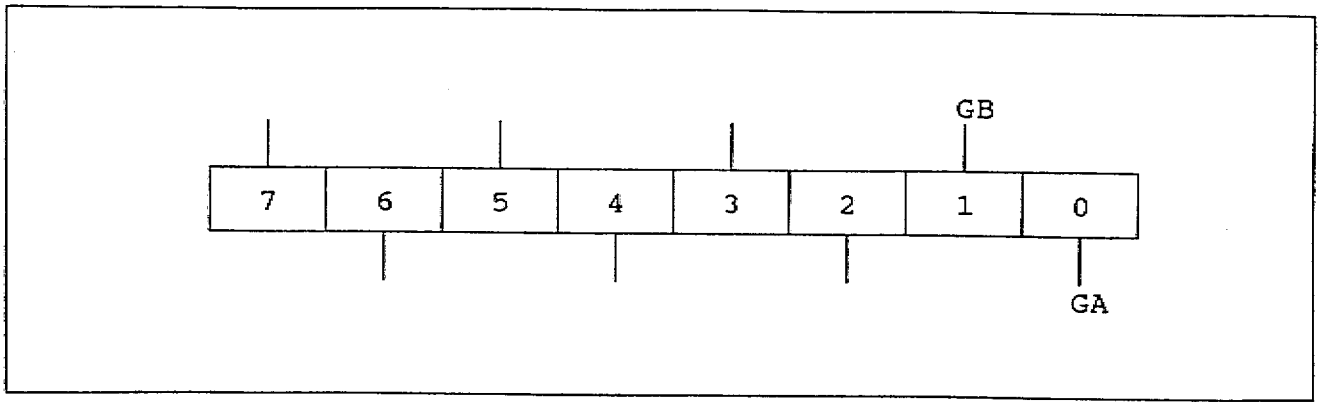


Figure 5.21 GAINREG register.

5.2.21 GMEM0 - Gain memory 0 register (offset 24)

The GMEM0 register holds the programmable gain settings for channels 0, 4, 8 and 12. The layout of this register is shown in figure 5.22.

For each channel, the gain is set by two bits, GA and GB, as follows:

GB	GA	Gain Setting
0	0	1
0	1	2 (PGH) or 10 (PGL)
1	0	4 (PGH) or 100 (PGL)
1	1	8 (PGH) or 1000 (PGL)

The gain of the current channel can be read back from the GAINREG register.

Bits 1-0 - Channel 0 Gain (GB0 and GA0)

Channel 0 gain setting bits.

Bits 3-2 - Channel 4 Gain (GB4 and GA4)

Channel 4 gain setting bits.

Bits 5-4 - Channel 8 Gain (GB8 and GA8)

Channel 8 gain setting bits.

Bits 1-0 - Channel 12 Gain (GB12 and GA12)

Channel 12 gain setting bits.

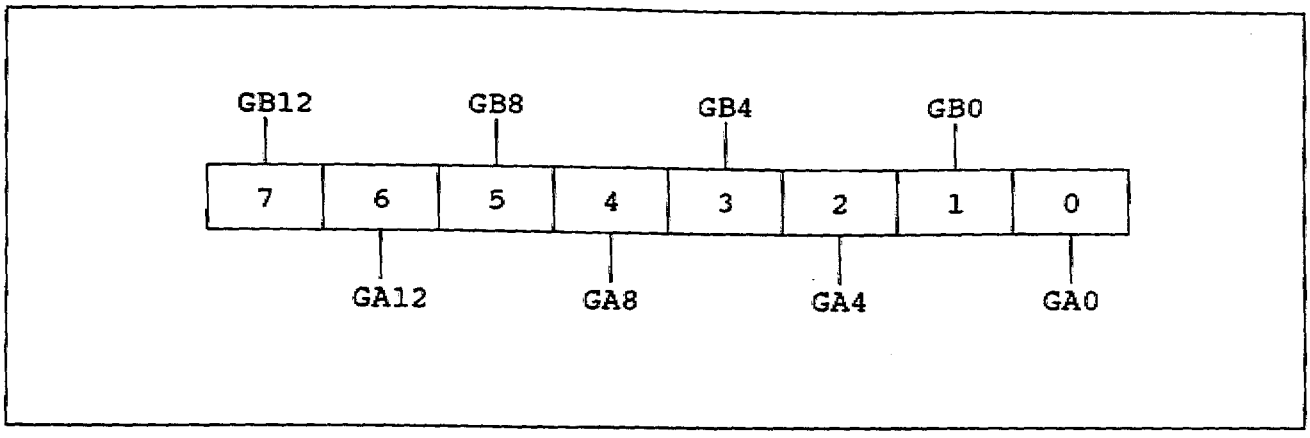


Figure 5.22 GMEM0 register.

5.2.22 GMEM1 - Gain memory 1 register (offset 25)

The GMEM1 register holds the programmable gain settings for channels 1, 5, 9 and 13. The layout of this register is shown in figure 5.23.

Gain codes are discussed in the description of the GMEM0 register.

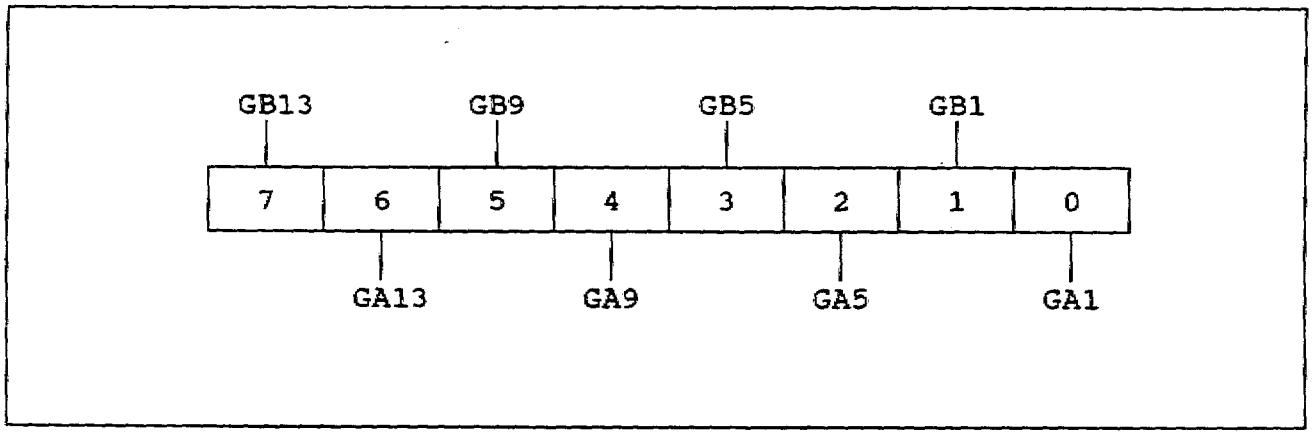


Figure 5.23 GMEM1 register.

Bits 1-0 - Channel 1 Gain (GB1 and GA1)

Channel 1 gain setting bits.

Bits 3-2 - Channel 5 Gain (GB5 and GA5)

Channel 5 gain setting bits.

Bits 5-4 - Channel 9 Gain (GB9 and GA9)

Channel 9 gain setting bits.

Bits 1-0 - Channel 13 Gain (GB13 and GA13)

Channel 13 gain setting bits.

5.2.23 GMEM2 - Gain memory 2 register (offset 26)

The GMEM2 register holds the programmable gain settings for channels 2, 6, 10 and 14. The layout of this register is shown in figure 5.24.

Gain codes are discussed in the description of the GMEM0 register.

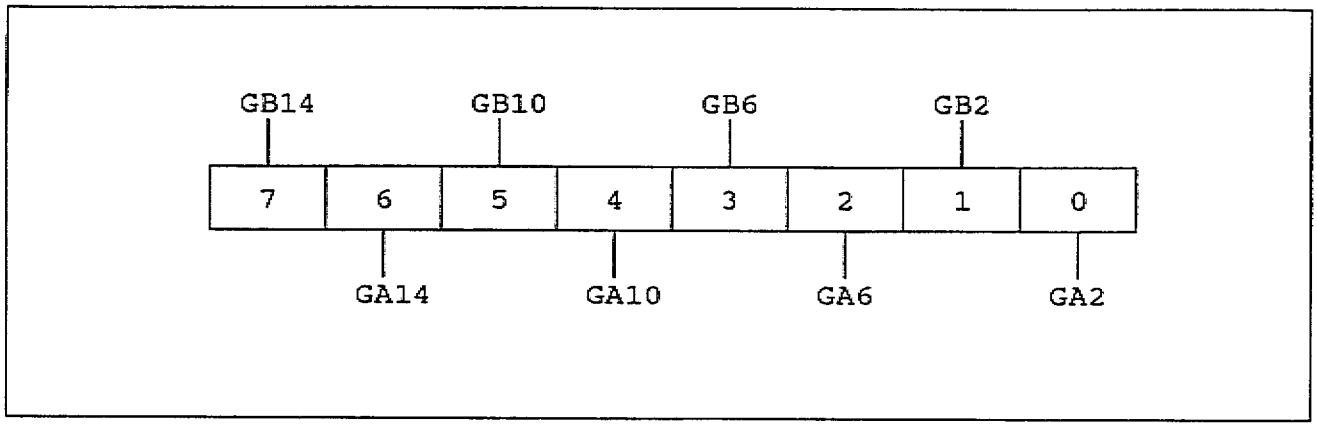


Figure 5.24 GMEM2 register.

Bits 1-0 - Channel 2 Gain (GB2 and GA2)

Channel 2 gain setting bits.

Bits 3-2 - Channel 6 Gain (GB6 and GA6)

Channel 6 gain setting bits.

Bits 5-4 - Channel 10 Gain (GB10 and GA10)

Channel 10 gain setting bits.

Bits 1-0 - Channel 14 Gain (GB14 and GA14)

Channel 14 gain setting bits.

5.2.24 GMEM3 - Gain memory 3 register (offset 27)

The GMEM3 register holds the programmable gain settings for channels 3, 7, 11 and 15. The layout of this register is shown in figure 5.25.

Gain codes are discussed in the description of the GMEM0 register.

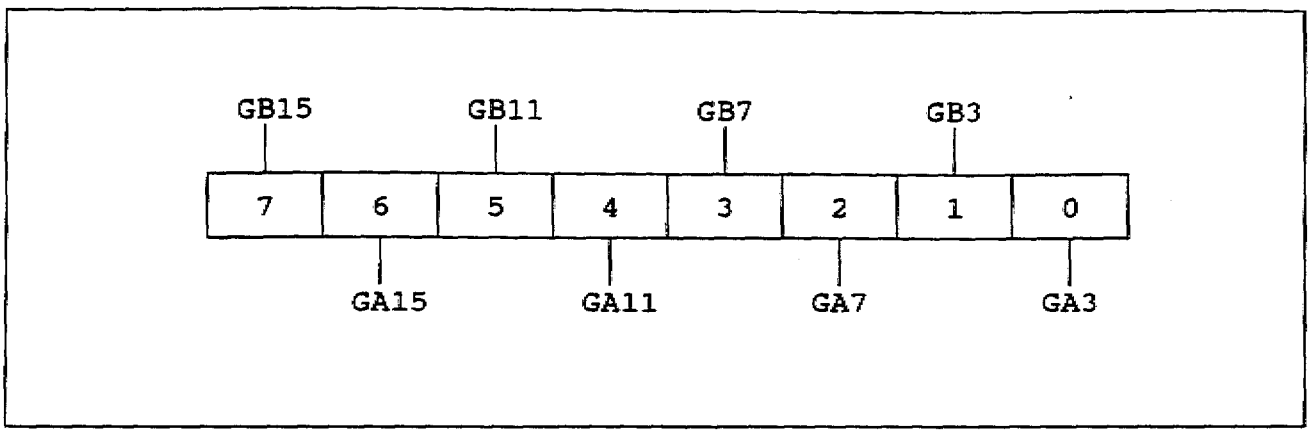


Figure 5.25 GMEM3 register.

Bits 1-0 - Channel 3 Gain (GB3 and GA3)

Channel 3 gain setting bits.

Bits 3-2 - Channel 7 Gain (GB7 and GA7)

Channel 7 gain setting bits.

Bits 5-4 - Channel 11 Gain (GB11 and GA11)

Channel 11 gain setting bits.

Bits 1-0 - Channel 15 Gain (GB15 and GA15)

Channel 15 gain setting bits.

Chapter 6

Programming Guide

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

As mentioned in the previous chapter, where possible you should make use of the supplied driver 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 provided drivers. This code is found in the \SOURCE directory of the driver diskette supplied with your PC-30. When you study this code, 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 +5V range :

$$\text{Voltage} = (\text{Digital code})(5)/4096$$

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

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

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 74(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. Write 0 to the GMEM0, GMEM1, GMEM2 and GMEM3 registers. This selects a channel gain of 1 for all input channels.

viii. Wait at least 100 uS.

ix. 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-30PG'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. Setting channel gain

The gains of channels 0, 4, 8 and 12 are set in the GMEM0 register, the gains of channel 1, 5, 9 and 13 in the GMEM1 register, the gains of channel 2, 6, 10 and 14 in the GMEM2 register and the gains of channel 3, 7, 11 and 15 in the GMEM3 register. For example, using a PGL board, to set the gain of channels 0, 1 and 4 to 1000, and that of all other channels to 1 the following values would be used :

GMEM0	00001111 (binary)
GMEM1	00000011 (binary)

GMEM2	00000000 (binary)
GMEM3	00000000 (binary)

6.7. Obtaining a single A/D reading

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

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

ii. Set the channel gain as described above.

iii. 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.

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

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

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

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

6.8. Setting the sample rate

Assuming that the PC-30PG 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, 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. Bear in mind that if the board is not jumpered for internal clock, then neither of the above registers will have any effect.

The procedures Ad_prescaler and Ad_clock (in the file PC30S.C) show how to perform the above functions.

6.9. Loading the channel list/block counter

The easiest way to perform multi-channel data input operations on the PC-30, is to make use of the channel list. Any sequence of channels, in any order, up to a maximum length of 31, can be loaded into the channel list. On completion of each A/D conversion, the PC-30 automatically loads the next channel into the channel address register. The procedure for loading this channel list is as follows:

i. If you are using block mode, then write (257 - N) to the BLKCNT register, where N is the number of samples per block.

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

iii. Write the first channel address to the ADCCR. All other bits except the STBC should be cleared. The channel list now contains only the first channel.

iv. Set the A/D mode register (ADMDE) to 9F(hex). This sets the channel list mode to add.

v. Write, in sequence, the rest of the channels to be converted to the ADCCR, as above. Note that on read the ADCCR will reflect the first channel address written.

vi. Set the A/D mode register (ADMDE) to either 90(hex) (for normal trigger mode) or 91(hex) (for block trigger mode). This sets the channel list mode to ignore. You can now write control bits to the ADCCR without disturbing the channel list.

The channel list is now ready for operation.

The driver procedure Ch_list_load, found in the file PC30S.C, shows how the driver software performs this function.

If you need to convert only a single channel, all that is required is to write the channel address to the ADCCR. It's still a good idea to set the mode register to 90(hex), as described above, as this enables the PC-30PG FIFO.

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 50KHz on PC-30PG boards. The procedure is as follows:

i. Set the sampling rate, as described above.

ii. Set the channel gain, as described above.

iii. Load the channel or channels to be converted into the channel list, and set the block counter as described above.

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

v. Wait for the A/D done bit in the ADDSR to be set. As soon as it is, read the A/D result into memory.

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

vii. 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. Set the channel gain, as described above.

iii. Load the channel or channels to be converted into the channel list, and set the block counter as described above.

iv. 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. In the case of a PC-30PG using an interrupt level greater than 7, both interrupt controllers must receive this command.

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

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

Note that the "single channel" refers to the number of DMA channels used, NOT to the number of input channels that can be sampled. All the PC-30 boards described in this manual can scan multiple channels under the control of the channel list hardware. DMA is normally the only way to achieve full throughput on the various PC-30 boards, 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 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. Set the channel gain, as described above.

iii. Load the channel or channels to be converted into the channel list, and set the block counter as described above.

iv. 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.

v. Set the DMA enable bit in the ADCCR to 1, and the DMA mode bit in the ADMDE

register to 1. This enables the PC-30 DMA. In the case of the PC-30PG, it also sets the DMA mode to terminate on TC. This is not strictly necessary, but should be done for compatibility with Microchannel products.

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, 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.

viii. When the sampling procedure is complete, set the STBC bit to 1, the DMA mode bit to 0, 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).

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

6.13. Dual channel gap-free DMA

The PC-30PG can perform dual channel gap-free DMA. Normal DMA can only transfer up to 32768 samples, but dual channel DMA can transfer to the limits of installed memory. Dual channel DMA also has the advantage the restrictions due to the host PC's requirement that the entire memory space must be in a single 64K block fall away.

Dual channel gap-free DMA is very similar to conventional DMA. The basic procedure is described below:

i. Set the sampling rate, as described above.

ii. Set the channel gain, as described above.

iii. Load the channel or channels to be converted into the channel list, and set the block counter as described above.

iv. Split the memory space into which you wish to transfer the data into segments. Each segment must be chosen such that only the lower 16 bits of the address vary from start to end.

v. Set up the primary DMA channel with the address of the first segment, and the secondary DMA channel with the address of the second segment. Enable both levels. The DMA controller should be programmed for demand mode operation.

vi. Set the DMA enable bit in the ADCCR to 0, and the DMA mode bit in the ADMDE register to 1. This enables the PC-30 DMA in the swop on TC mode.

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

viii. 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.

ix. Once the primary DMA channel completes, the PC-30PG automatically swops to the secondary channel, and sets the DMA enable bit in the ADCCR. You must now reprogram the primary DMA channel with the address of the next segment, and clear the DMA enable bit.

x. When the secondary DMA channel completes, the PC-30PG swops back to the primary DMA channel, and again sets the DMA enable bit in the ADCCR. You must then reprogram the secondary DMA channel with the address of the next DMA channel, and clear the DMA enable bit.

xi. This process of swopping buffers continues until the last segment in the memory space. When this occurs, rather than clearing the DMA enable bit, you should leave it set. The PC-30 then automatically shuts down DMA operation at the end of the last segment.

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

The procedures Mdh_chan, Dma_chk and Dma_close in the file PC30C.C show how the driver software performs this function.

6.14. DMA data format

After DMA has completed, the format of the data in the DMA buffer is as 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
	. . . .				

6.15. Dealing with extended memory

Any version of the PC-30 can execute DMA transfers into extended (NOT expanded) memory. This is done precisely as described above. Note however that you will need to make use of BIOS functions to obtain this data, as your program cannot access this memory directly under DOS.

The procedures Mde_chan, E_mem_size and Xfer_dma_res in the file PC30C.C show how the

driver software accesses extended memory.

6.16. Error detection

All versions of the PC-30 described in this manual implement error detection. If an error occurs, the error bit in the ADDSR is set. There are two approaches to checking this bit.

i. Check only at the end of an operation. As long as you do not write a one to the error reset bit in the ADMDE, the error bit will remain set. You need hence only check the bit at the end of an entire operation. Note however that you must check the bit fast enough such that it is not set due to A/D conversions which occur after you have obtained all the samples you require. For this reason, this technique is not suitable for error detection of DMA operations, as the error bit will almost inevitably have been set before a program can check it, regardless of whether it was or was not set during the actual DMA transfer.

ii. Check the stored data. The PC-30 has been designed in such a way that the error detection bit is in the same register as the A/D data. It can hence be transferred to memory along with the data, at no extra overhead. This always occurs during DMA transfers. Error checking can thus be performed on the actual data, after the completion of sampling.

Almost all of the routines in the PC-30 driver software implement error detection.

6.17. End of DMA block interrupt

Note that for the PC-30PG to generate an end of DMA block interrupt, the DMA mode must be set to "swop on TC", as discussed in the description of the ADMDE register.

[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-30PG 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, R32, R33 and R36. 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%,

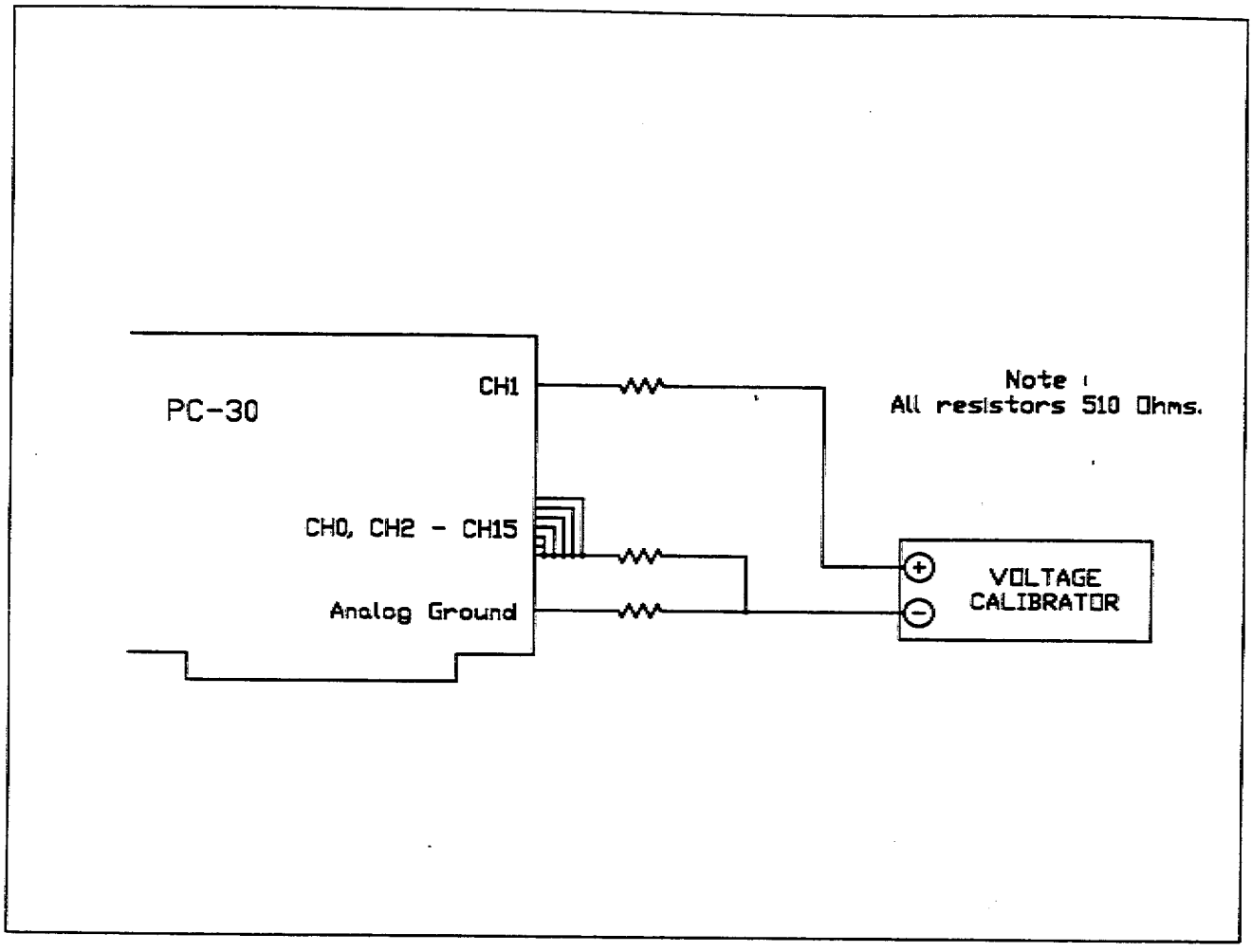


Figure 7.1. A/D calibration connections.

resolution 100 nV or better.

7.2.3 Procedure

7.2.3.1 Unipolar mode

i. Set the A/D for a gain of 1, and apply (-FS + 1/2 LSB) to channel 1. This is 0.610 mV for the 5 V range. Adjust R36 (unipolar A/D offset potentiometer) for an output code which flickers evenly between 000H and 001H.

i. Set the A/D for its maximum gain, and apply 0.000V to channel 1. Adjust R70 (instrumentation amplifier offset potentiometer) for an output code of 000H.

ii. Set the A/D for a gain of 1, and apply (+ FS - 3/2 LSB) to channel 1. This is 4.9982 V for the 5 V range. Adjust R32 (gain potentiometer) for an output code which flickers evenly between FFEH and FFFH.

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

7.2.3.2 Bipolar mode

i. Set the A/D for a gain of 1, and apply (-FS + 1/2 LSB) to channel 1. This is -4.9988V for the -5 to 5V range. Adjust R33 (bipolar A/D offset potentiometer) for an output code which flickers evenly between 000H and 001H.

i. Set the A/D for its maximum gain, and apply 0.000V to channel 1. Adjust R70 (instrumentation amplifier offset potentiometer) for an output code of 800H.

ii. Set the A/D for a gain of 1, and apply (+ FS - 3/2 LSB) to channel 1. This is + 4.9963V for the -5 to 5V range. Adjust R32 (gain potentiometer) for an output code which flickers evenly between FFEH and FFFH.

iii. Repeat the above three 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 R4 (DAC0) or R11 (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 R2 (DAC0) or R9 (DAC1) to adjust the D/A output to

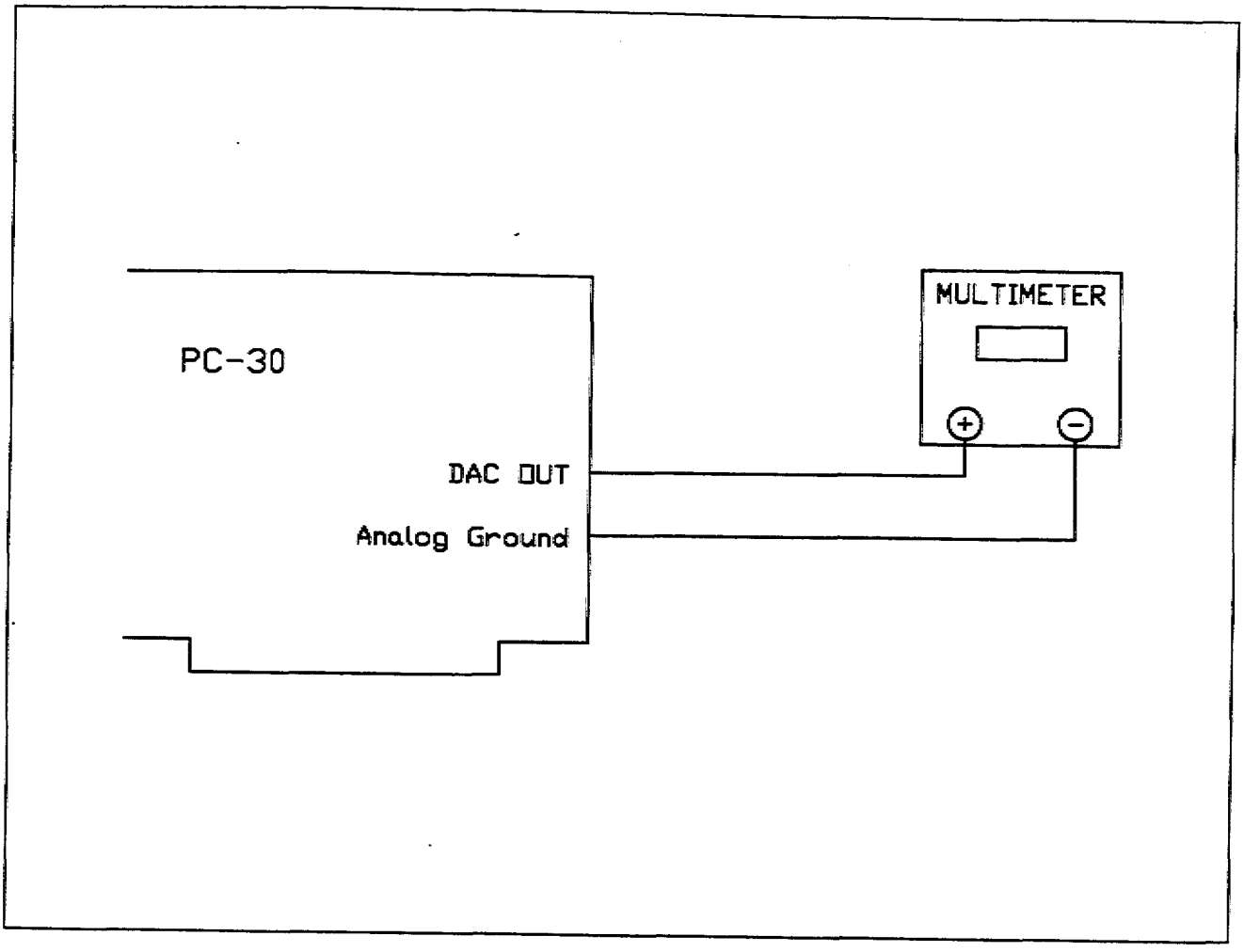


Figure 7.2. D/A calibration connections.

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.

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 R4 (DAC0) or R11 (DAC1) to adjust either pin 6 of U2 (DAC0) or pin 6 of U5 (DAC1) to 0.000 V + - 1mV.

ii. Set the D/A code to 000H. Use R1 (DAC0) or R8 (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 R2 (DAC0) or R9 (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 R19 (DAC2) or R24 (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 R19 (DAC0) or R24 (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 or 8 differential

2. Resolution

12-bit, 1 in 4096

3. Total System Accuracy

+- 1.5 LSB

4. Differential Nonlinearity

+- 3/4 LSB max.

5. Quantization Uncertainty

+-1/2 LSB

6. Input Ranges

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

7. Input Bias Current

+- 200 nA max

8. Input Bias Current Drift

+ - 100 pA per °C.

9. Gain Drift

+ - 30 ppm per °C.

10. Offset Drift

+ - 30 ppm per °C.

11. Input Impedance

10G/20 pF (off channel).

10G/100 pF (off channel).

12. Offset Voltage

+ - 5 LSB, adjustable to zero.

13. Gain Error

Adjustable to 0

14. Monotonicity

0 to 70 °C

15. Programmable Gain Ranges

1, 2, 4, 8 (PGH)

1, 10, 100, 1000 (PGL)

16. Gain accuracy

0.25% maximum, 0.05% typical for gains < 1000

1% maximum, 0.1% typical for gain = 1000

17. Data acquisition rate

200 KHz (Gains of 1, 2, 4, 8, 10, 100)

100 KHz (Gain of 1000)

II. A/D clock

1. Internal clock

2 MHz, crystal controlled.

2. Clock frequency initial tolerance

0.01%

3. Clock drift

10 ppm per °C.

4. Internal clock divider

16-bit prescaler, 16-bit divider.

5. External clock

TTL compatible

6. External Trigger

TTL compatible, enables or disables conversions.

7. Channel List Length

31 entries max.

8. Block scan mode

Up to 256 channels per block, all channels in block converted at maximum throughput on each clock pulse.

III. Analog Output

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

+-0.2 LSB (12-bit), +-3 LSB (8-bit), after calibration.

9. Gain Drift

+- 30 ppm per °C (12-bit)

+- 0.007% per °C (8-bit)

10. Throughput

130 KHz max.

11. Output compliance

+- 5 mA.

12. System Accuracy

0.025%

13. Monotonicity

0 to 70 °C

IV. Digital I/O

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

50 way female D-type Amphenol DB50 or equivalent.

VII. Environmental

1. Operating Temperature

0 to 70 °C

2. Storage Temperature

-55 to 150 °C

3. Relative Humidity

5% to 95% noncondensing

VIII. PC Interface

1. Base Address

0 to 1FFF, DIP switch selectable.

2. Number of registers

32 8-bit registers.

3. Interrupts

Jumper selectable on end of conversion, from timer counter, or on end of DMA block.

4. DMA

Dual channel, 16-bit jumper selectable

IX. Power

1. +5V

1,5A typ.

2. +12V

100mA typ.

3. -12V

100mA typ.

Appendix B

Compatibility

This appendix discusses the compatibility of the PC-30PG with the other new series PC-30 boards (PC-30B, PC-30C, PC-30D and PC-30DS) and with the older series boards (PC-26, PC-30 and PC-39).

I. New series boards

The PC-30PG is completely compatible with all the new series boards (PC-30B, PC-30C, PC-30D and PC-30DS), with the exception of the PC-30PG's programmable gain capability.

1. Running old software on the PC-30PG

Pre PC-30PG software for the new series boards can be run on the PC-30PG as long as the gain memory of the PC-30PG has been set to a known state prior to running the old software. Included on the distribution disk is a a utility program which sets the gain of all the PC-30PG channel to 1. This is ZEROGAIN.EXE. It is used as follows:

ZEROGAIN <base address>

The <base address> parameter is the board's base address, in HEX. If the base address is not given, then the default base address of 700 HEX will be used.

2. Running PC-30PG software on older "new series" boards

Any PC-30PG software can be run on the PC-30B, PC-30C, PC-30D and PC-30DS without modification. Accesses to the gain memory are simply ignored.

II. Old series boards

In general, the new series boards are completely compatible with the old, and any well behaved software written for the PC-26, PC-30 or PC-39 will operate correctly in conjunction with the PC-30B, PC-30C, PC-30D, PC-30DS or PC-30PG.

In terms of software, all that is required to set the new boards into a mode compatible with the old boards is to write 92(hex) to the ADMDE register. As all well behaved PC-26/PC-30/PC-39 software should do this in any event, no special setup procedures are required for compatibility. When using a PC-30PG however, the gain memory must be set as described above.

1. PC-26/PC-30

A. Analog inputs.

The analog inputs of the PC-30 had an input impedance of 22K. The PC-30B, PC-30C, PC-30D and PC-30PG have an industry standard input impedance of 10M/20pF (off channel) and 10M/100pF (on channel). Series resistors are used to provide fault protection. The PC-30DS has an fixed input impedance of 50M/20pF.

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-30B, PC-30C, PC-30D, PC-30DS and PC-30PG analog inputs must be grounded.

B. Interrupt system modifications.

The PC-30B, PC-30C, PC-30D, PC-30DS and PC-30PG'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-30B, PC-30C, PC-30D, PC-30DS and PC-30PG keep their interrupt lines active indefinitely. This means the PC-30B, PC-30C, PC-30D, PC-30DS and PC-30PG cannot miss interrupts (unless of course an interrupt service routine takes up all the processing time from one PC-30B, PC-30C, PC-30D, PC-30DS or PC-30PG sample to the next). Hence it is no longer necessary to disable all other interrupts unless the PC-30B, PC-30C, PC-30D, PC-30DS or PC-30PG is operating at a high enough sample rate that the processing power of the PC becomes too little to handle the PC-30B, PC-30C, PC-30D, PC-30DS or PC-30PG as well as other interrupts.

iii. This change is invisible to PC-26/PC-3ʊ 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-30B, PC-30C, PC-30D, PC-30DS and PC-30PG 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.

2. PC-39

The new series boards are fully compatible with the PC-39. The PC-39 already has the modifications detailed under the PC-26/PC-30 section above.

3. Jumpers

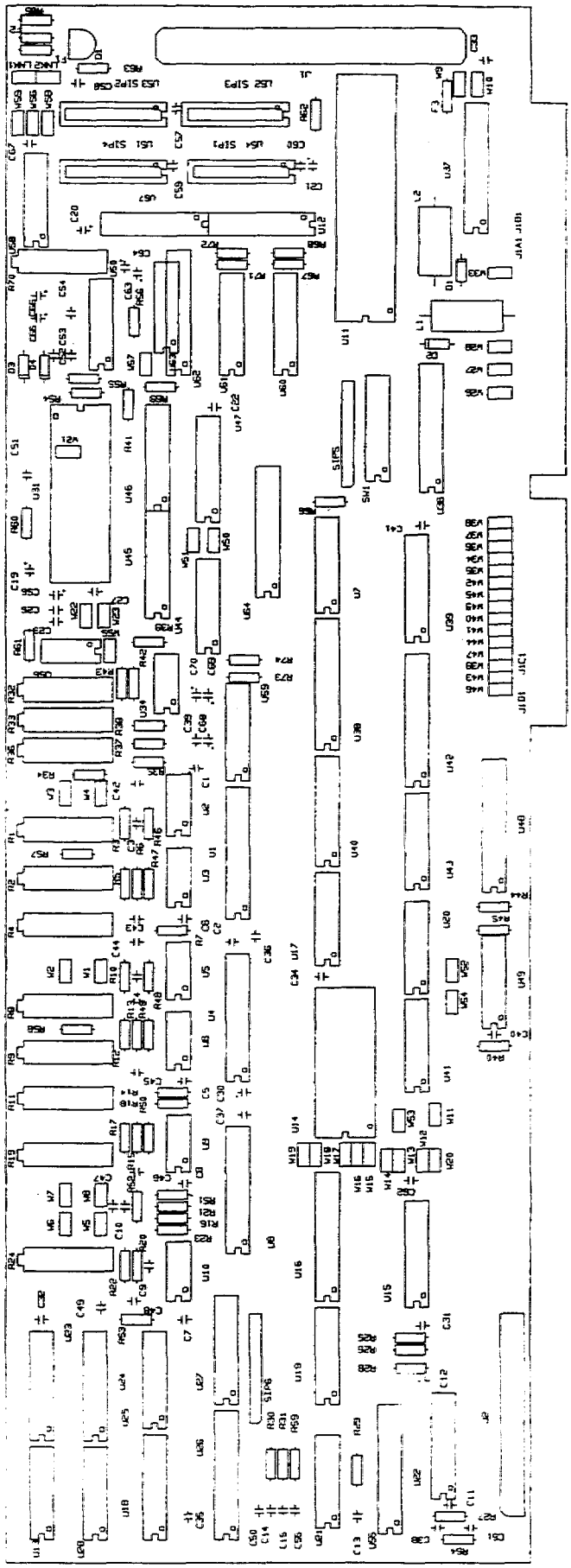
The PC-30B, PC-30C, PC-30D, PC-30DS and PC-30PG have a significantly larger number of possible jumper options than the older boards. You must ensure that you select jumper options the same as the fixed setting on the older boards. As supplied by the factory, the new series boards are configured as were the older series boards.

[This page intentionally left blank.]

Appendix C

Layout Diagram

The following page shows the layout diagram for the PC-30PG



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 an 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 it 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 the PC-30 should not be greater than 1K.

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.

5. The board does not operate at full throughput.

The PC-30PG is a very high performance board, and makes correspondingly high demands on the PC in which it is installed. If you find that your PC-30 cannot operate at full throughput, here are some points to check.

i. To achieve full throughput, you will need to make use of DMA transfers. Program transfer techniques cannot transfer data sufficiently fast to achieve full throughput on the PC-30PG.

ii. If your PC has a high speed or "Turbo" mode, make sure the PC is in this mode. You may have to run a special program, or press a combination of keys on the keyboard to switch to high speed mode. Consult the manual supplied with your PC for more information. Note that the speed setting in most clones effects not only the CPU speed, but the DMA speed as well.

iii. Some PC compatibles have a jumper setting for number of bus wait states. This should be set to the minimum. Consult the manual supplied with your PC for more information.

6. The board does not operate in block trigger mode.

When using block trigger mode, you must remember that the board operates at its maximum

throughput; in fact, generally significantly faster than the rated maximum. Hence you must use DMA for block mode transfers, except in the case of a block length no greater than 16. The PC-30PG can store up to 16 samples in its FIFO buffer, so easing the transfer rate requirements on the host PC.

[This page intentionally left blank.]

Index