PCI-1002

User's Manual

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

1. Introduction

The PCI-1002H/L card provides 12-bit ADC and two 16-bit digital I/O ports.

1.1 General Description

The PCI-1002L and PCI-1002H are high performance multifunction cards, with both A/D and digital I/Os in a 5V PCI slot. This family features one 12-bit 125K A/D converter, 16-channel D/I, 16-channel D/O, a programmable interrupt source and support truly “Plug and Play”. PCI-1002H/L provides either 32-channel, single-ended analog inputs or 16-channel differential analog inputs which are jumper selectable. The input channel number is also easily programmable with our software. The PCI-1002L, Our low-gain model, is equipped with a high speed PGA (programmable gain amp.) featuring programmable gain control (1,2,4,8). PCI-1002H is our high-gain model, equipped with a high-gain/high-resolution PGA with programmable gain control (1,10,100,1000). There are 16 channels of TTL-compatible digital output ports and 16 channel of TTL-compatible digital input ports. This series provides three programmable trigger methods: software trigger, pacer trigger and external trigger. The external trigger control can also be programmed into a post-trigger, pre-trigger, or external pacer trigger mode. There are also several interrupt sources: A/D conversion interrupt, pacer interrupt, external interrupt. This multifunction card also provides an A/D buffer and a high data transfer rate: 2.7M words per second in non-burst mode. This powerful A/D control mechanism offers the largest flexibility for various kinds of usage and the least I/O overhead for your system!! PCI-1002 series fully support “Plug and Play” under your system, and can operate even on full speed of a PCI bus (33MHz). This important feature fits the latest version of computer I/O system and yields the best performance. The low cost and extensive features of this series make it especially useful for any cost-effective application.

1.2 The Block Diagrams

Here's the block diagram of PCI-1002:

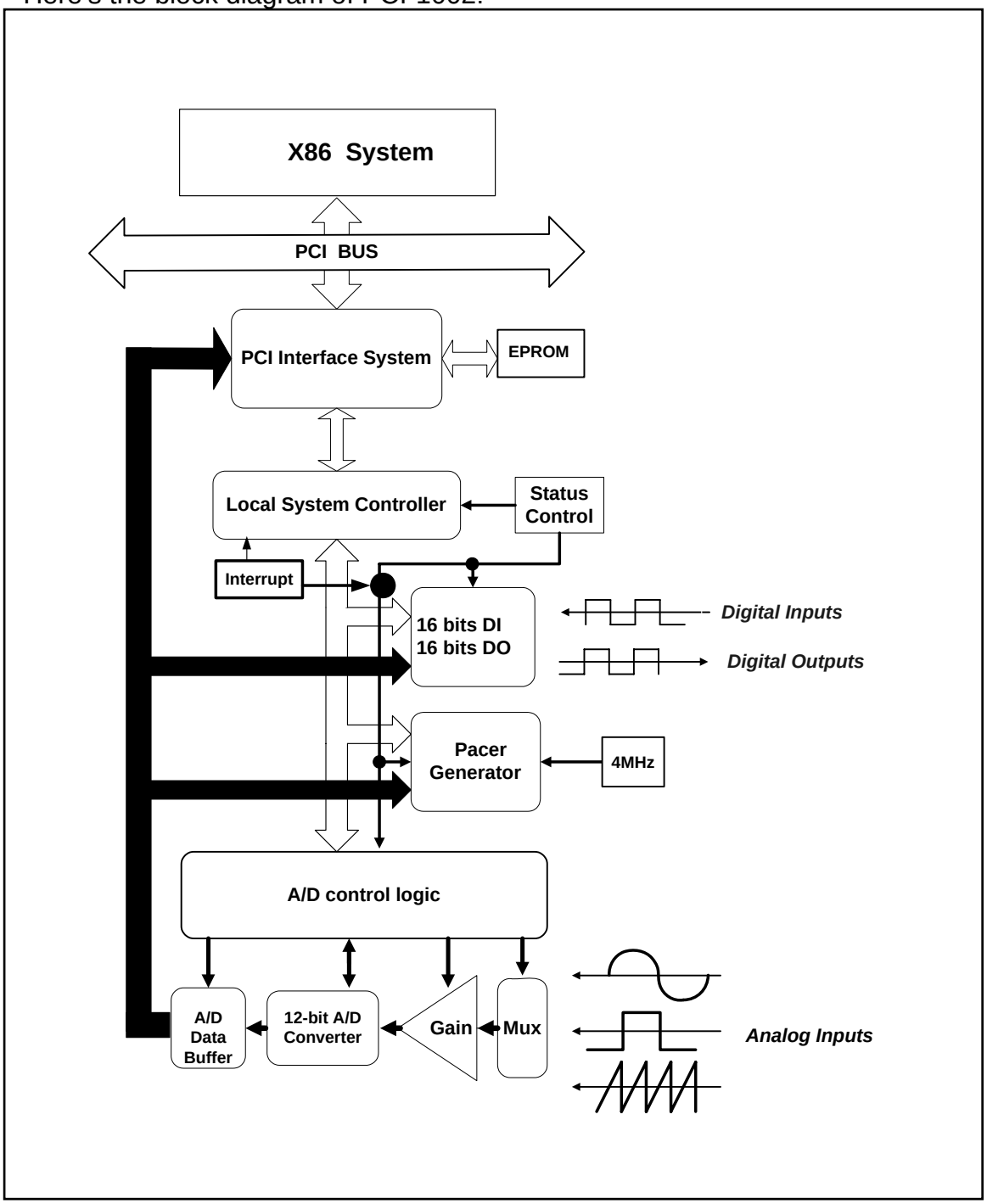


Figure 1-1. The PCI-1002 series block diagram.

1.3 Features

The following is a list of general features for the PCI-1002 series. Check section 1.4 for more details.

- Bus: 5V PCI (Peripherals Component Interface) bus.

1. A/D:
- One A/D converter with maximum 110Ksamples/second.
- 32 single-ended / 16 differential programmable inputs for PCI-1002L/H.
- Provides three different A/D trigger methods.
- Provides three different external trigger methods.
- Programmable gain control, programmable offset control.

2. DIO:
- 16 digital inputs and 16 digital outputs (TTL compatible) .
- High speed data transfer rate: 2.7M word/sec (non-burst mode).

3. Timer:
- One 16-bit machine independent timer for software (Timer 2).
- Two 16-bit pacer timers for A/D converter and interrupt (Timer0, Timer1).

1.4 Specifications

1.4.1 Power Consumption:

- +5V @960mA maximum, PCI-1002L/H
- Operating temperature: 0°C ~ +70°C

1.4.2 Analog Inputs

- Channels:(software programmable)
- 32 single-ended inputs/16 differential inputs, jumper selectable.
- Gain control: (software programmable)
- PCI-1002H: 1,10,100,1000.
- PIC-1002L: 1,2,4,8.
- Input signal ranges:
  PCI-1002L: Bipolar, with range of ±10V, ±5V, ±2.5V, ±1.25V
  PCI-1002H: Bipolar, with range of ±10V, ±1V, ±0.1V, ±0.01V
- Input current: 250 nA max (125 nA typical) at 25 °C.
- Maximum voltage: continuous single channel to **70Vp-p**
- Input impedance:
  PCI-1002H/L: $10^{10}\Omega$ // 6pF
- Settling time
  1002L --> 3 us for all input range
  1002H -->23 us for +/- 10V input range
  28 us for +/- 1V input range
  140us for +/-0.1V input range
  1300us for +/-0.01V input range

1.4.3 D/I and D/O

- Channels: 16-channel DI, 16-channel DO.
- DO: Digital output port
- Output level: TTL compatible
- Output current: I_{oh} = 0.5mA,I_{ol} = 8mA
- DI: Digital input port
- Input level: TTL compatible
- Input current: 50uA (max).

1.4.4 A/D Trigger Methods

- Trigger methods:
  Software trigger.
  Pacer trigger: 16-bit programmable timer/counter.
  External trigger: Pre-trigger, Post-trigger, external Pacer trigger.

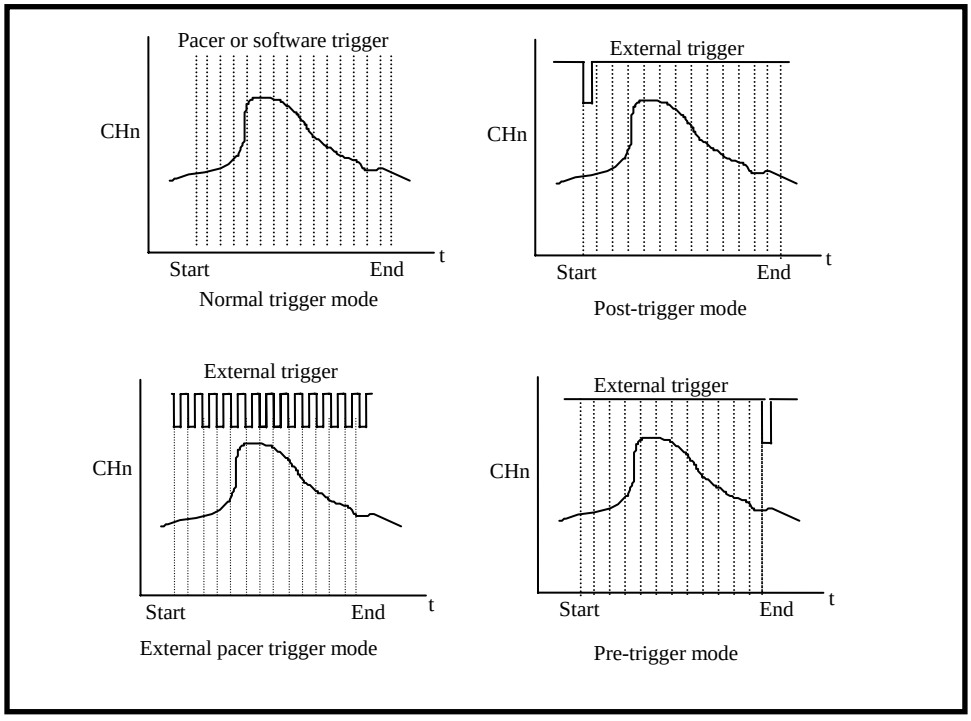


Figure 1-2. Trigger methods of PCI-1002.

1.4.5 Interrupt Channel

- Interrupt: INTA (Automatically assigned by PCI-initiator).
- Enable/Disable: Via PCI control register and add-on control register.
- Interrupt source: (Selected by on-board control register)

1. A/D conversion interrupt.
2. Pacer 0 interrupt (Timer 0).
3. Pacer 1 interrupt (Timer 1).
4. External interrupt.

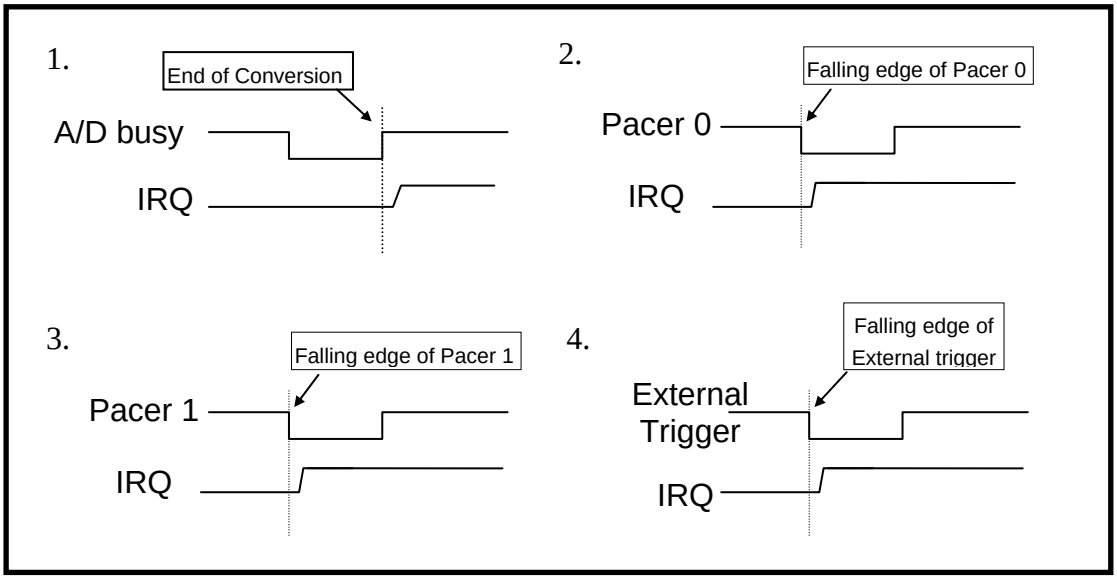


Figure 1-3. Programmable interrupt source.

1.4.6 Programmable Timer/Counter

- Type: 82C54 -8 programmable timer/counter.
- Timers: Timer 0 for Pacer trigger and interrupt.
- Timer 1 for External trigger and interrupt.
- Timer 2 for software machine independent timer.
- Input Clock: 4 M Hz.

1.5 Applications

- Signal analysis.
- FFT & frequency analysis.
- Transient analysis.
- Speech analysis.
- Temperature monitor.
- Vibration analysis.
- Energy management.
- Other industrial and laboratory measurement and control.

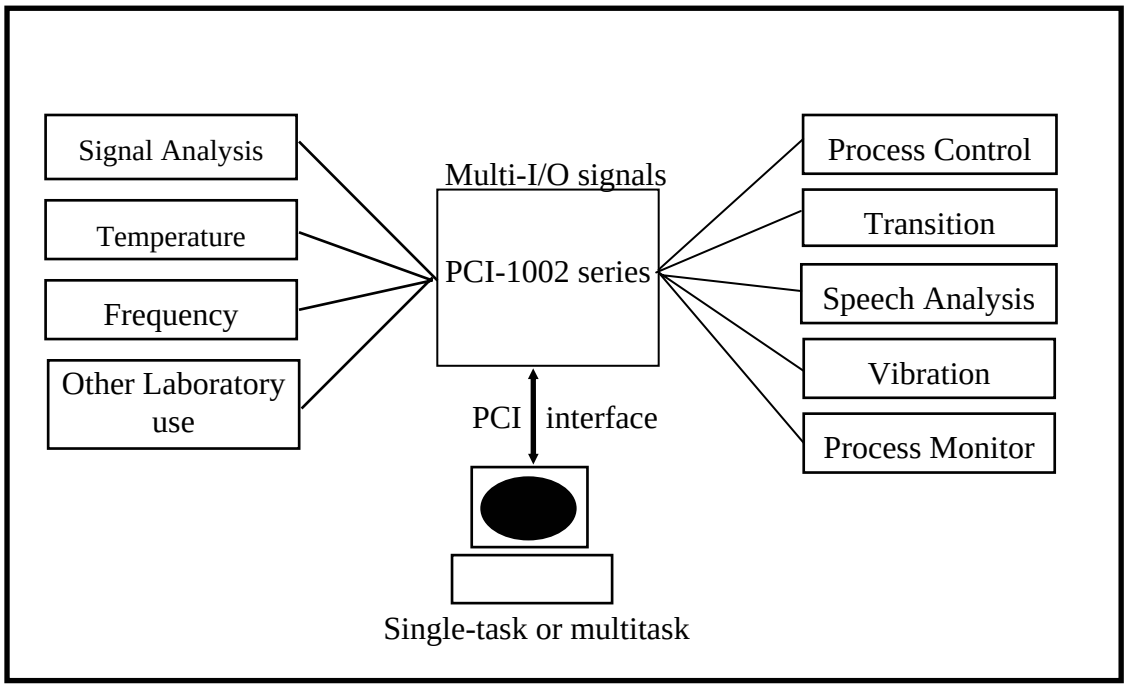


Figure 1-4. PCI-1002 series multifunction cards.

1.6 Product Checklist

In addition to this manual, the package includes the following items:

- One PCI-1002 card
- One ICP-DAS floppy diskette or CD-ROM
- One release note

It is recommended to read the release note first. The following important information will be given in the release note:

1. Where you can find the software driver & utility
2. How to install software & utility
3. Location of the diagnostic program
4. FAQ

Attention!

If any of these items are missing or damaged, please contact your local field agent. Save the shipping materials and carton in case you want to ship or store the product in the future.

2. Hardware Configuration

2.1 Board Layout

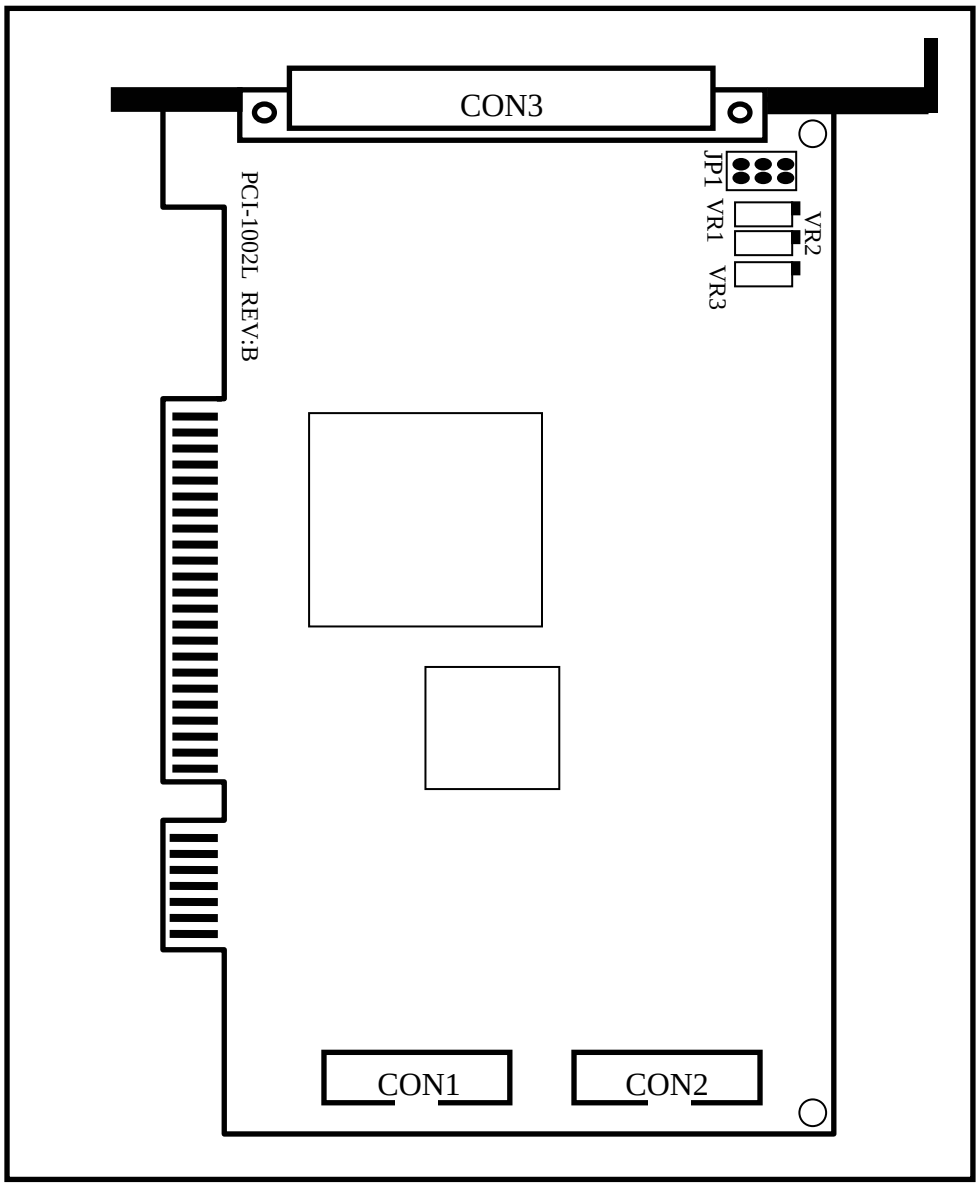
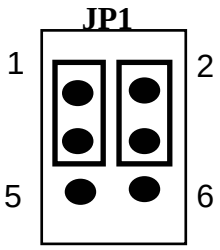


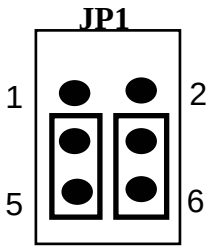
Figure 2-1. PCI-1002 board layout.

2.2 Jumper Settings

The PCI-1002 has only one jumper. JP1 is used to select the analog input type. For single-ended inputs, users should connected Pins 1&3 and Pins 2&4. For differential inputs, Pins 3&5 and Pins 4&6 should be connected.



Single-ended Inputs (Default)



Differential Inputs

2.3 AD Calibration

- AD Calibration for PCI-1002 card

  Step 1: Apply +10V to channel 0.

  Step 2: Apply +0V to channel 1.

  Step 3: Apply -10V to channel 2.

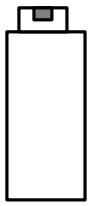
  Step 4: Run DEMO6.EXE.

  Step 5: Adjust **VR1** until channel 0 = fff or ffe

  Step 6: Adjust **VR2** until channel 1 = 800 or 801

  Step 7: Adjust **VR3** until channel 2 = 000 or 001

  Step 8: Repeat Step 4, Step 5, Step 6 until all are Ok.



VR1, VR2, VR3

2.4 System Block

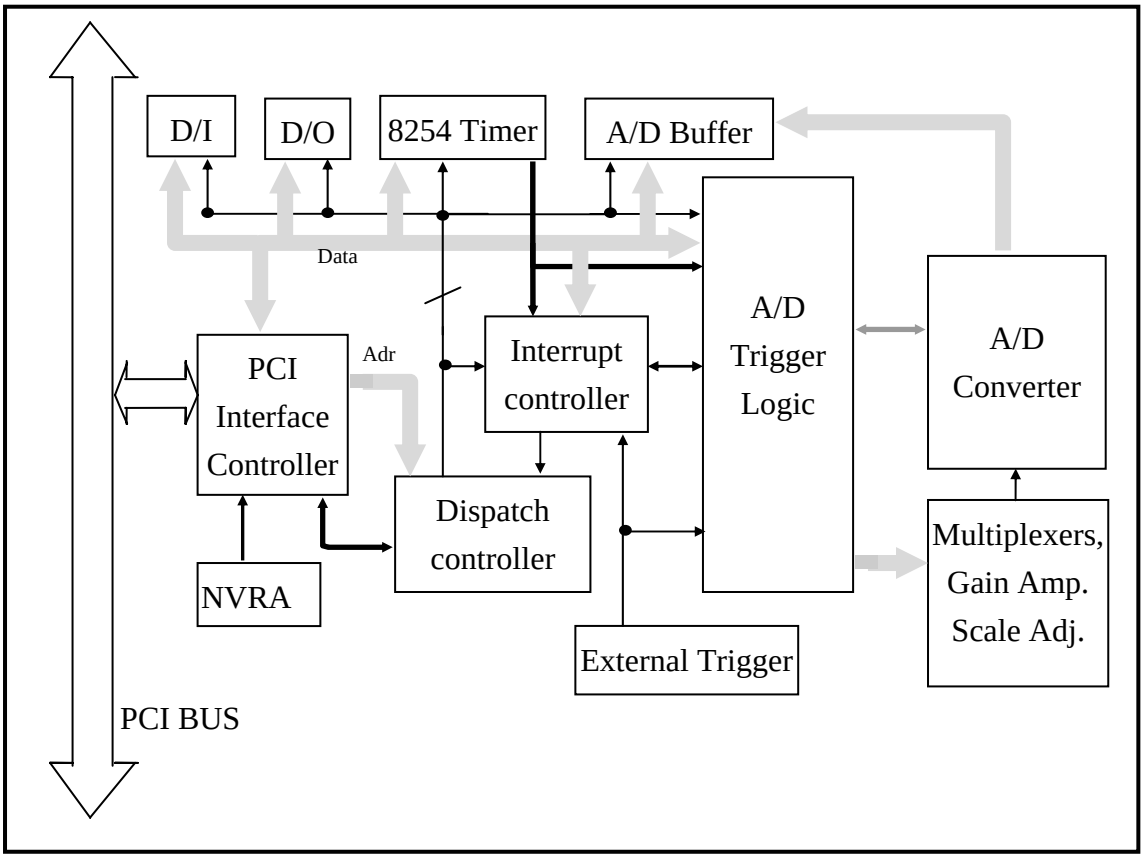
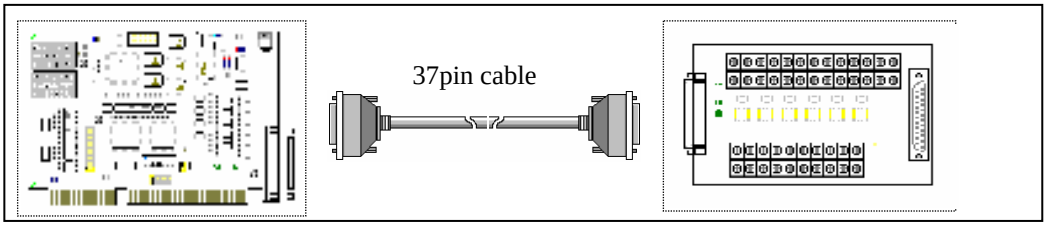


Figure 2-2. PCI-1002 System Function Block.

2.5 Daughter Boards

2.5.1 DB-1825

The DB-1825 is a daughter board designed for 32-channel AD cards such as ISO-AD32 and PCI-1002. Refer to “DB-1825 User Manual ” .

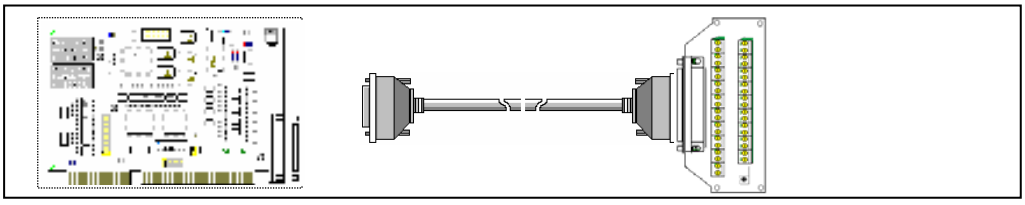


2.5.2 DB-8225

The DB-8225 provides an **on-board CJC**(Cold Junction Compensation) circuit for thermocouple measurement and a **terminal block** for easy signal connection and measurement. The CJC is connected to A/D channel_0. The PCI-1002 can connect CON3 directly to DB-8225 through a 37-pin D-sub connector. Refer to “DB-8225 User Manual” for details.

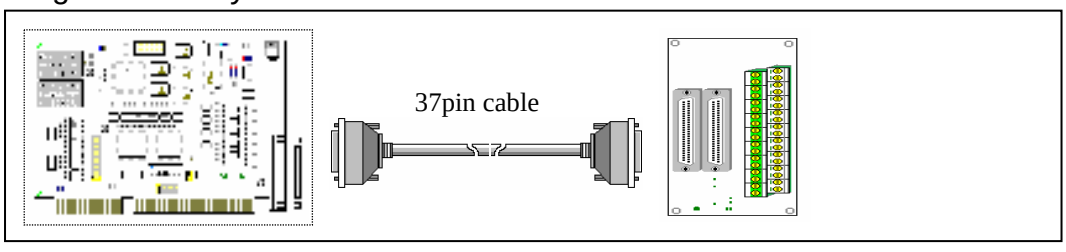
2.5.3 DB-37

The DB-37 is a general-purpose daughter board for D-sub 37 pins. It is designed for easy wire connection.



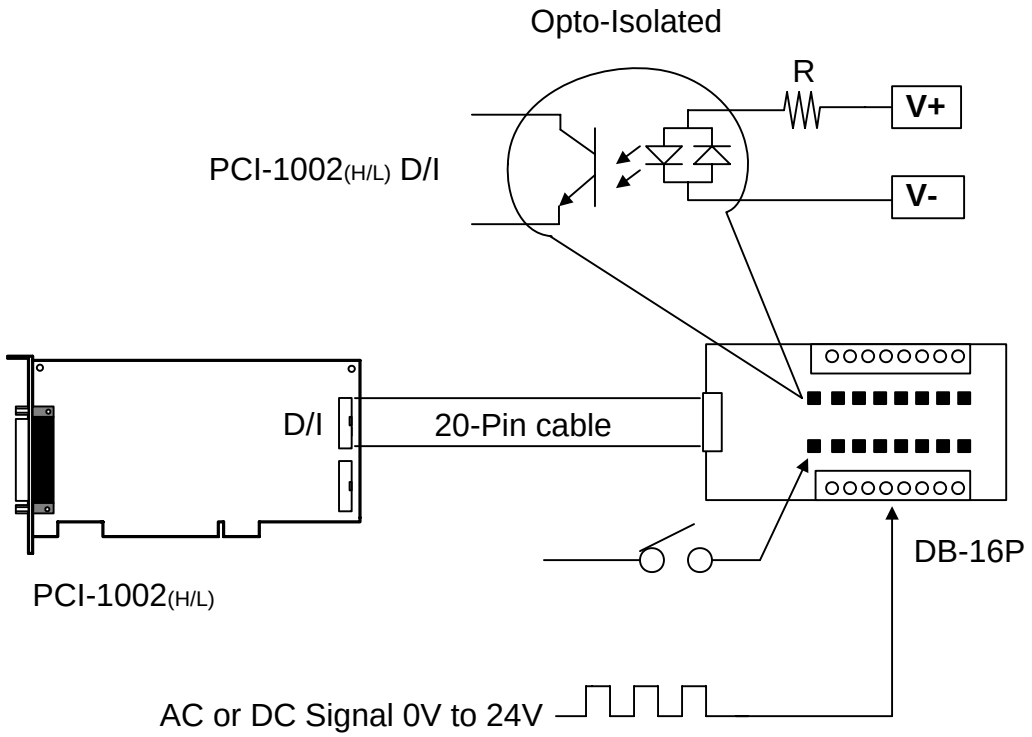
2.5.4 DN-37

The DN-37 is a general purpose daughter board for DIN Rail Mounting. It is designed for easy wire connection. It is Din-Rail mounted.



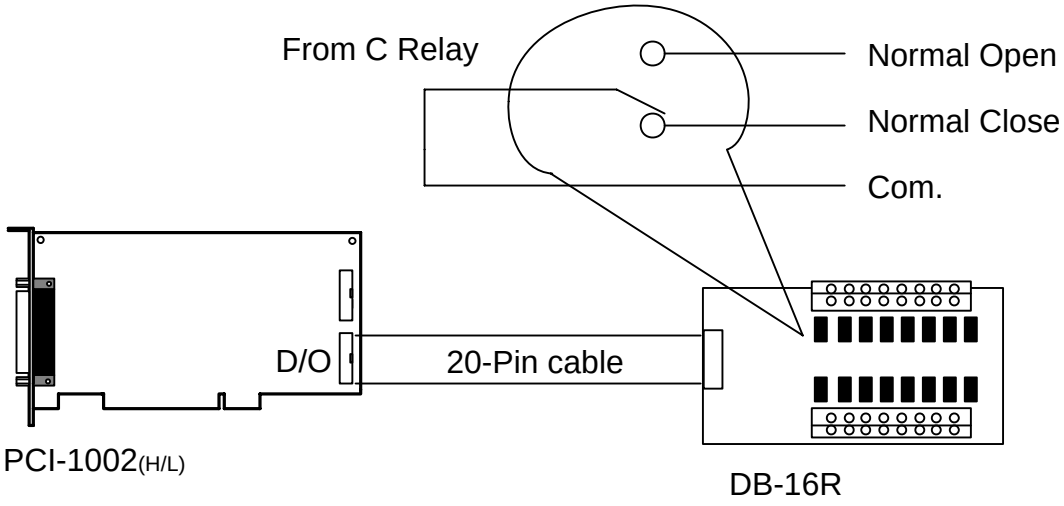
2.5.5 DB-16P Isolated Input Board

The DB-16P is a 16-channel isolated digital input daughter board. The optically isolated input of the DB-16P consists of a bi-directional optocoupler with a resistor for current sensing. You can use the DB-16P to sense DC signal from TTL levels up to 24V, or use the DB-16P to sense a wide range of AC signals. You can use this board to isolate the computer from large common-mode voltage, ground loops and transient voltage spikes that often occur in industrial environments.



2.5.6 DB-16R Relay Board

The DB-16R, a 16-channel relay output board, consists of 16 from C relays for efficient load switching via programmable controls. It is connected and functionally compatible with 785 series board but feature an industrial-type terminal block. Relays are energized by applying 5-volt signal to the appropriate relay channel on the 20-pin flat connector. There are 16 enunciator LEDs for each relay, light when their associated relay is activated. To avoid overloading your PC's power supply, this board provides a screw terminal for an external power supply.



Note: Channel: 16 From C Relay

Relay: Switching up to 0.5A at 110ACV or 1A at 24 DCV

2.6 Analog Input Signal Connections

The PCI-1002 can measure single-ended or differential-type analog inputs signal. Some analog signals can be measured in both modes. However, some analog signals only can be measured in one or the other. The user must decide which mode is suitable for measurement.

In general, there are 4 different analog signal connection methods (shown from Figure 2-3 to Figure 2-6). The connection in Figure 2-3 is suitable for grounding source analog input signals. The Figure 2-4 connection is used to measure more channels than in Figure 2-3, but it is only suitable for large analog input signals. The connection in Figure 2-5 is suitable for thermocouple and the Figure 2-6 connection is suitable for floating source analog input signals. **Note: In Figure 2-5, the maximum common mode voltage between the analog input source and the AGND is 70Vp-p, so the user must take care that the input signal is under this specification first. If the common mode voltage is over 70Vp-p, the input multiplexer will be permanently damaged!**

The simple way to select your input signal connection configuration is listed below.

1. Grounding source input signal → select Figure 2-3
2. Thermocouple input signal → select Figure 2-5
3. Floating source input signal → select Figure 2-6
4. If Vin > 0.1V, the gain<=10 and more channels are needed → select Figure 2-4

If you are unsure of the characteristics of your input signal, follow these test step:

1. Step1 : Try Figure 2-3 and record the measurement result
2. Step2 : Try Figure 2-6 and record the measurement result
3. Step3 : Try Figure 2-4 and record the measurement result
4. Compare the three results and select the best one

Figure 2-3. Connecting to grounding source input (Right way)

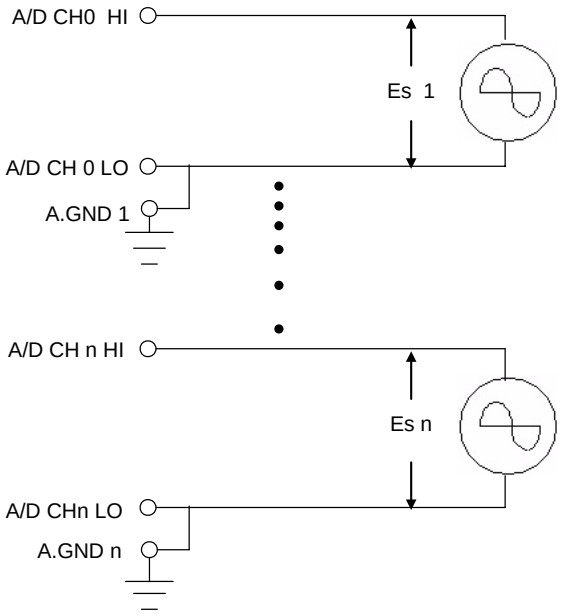


Figure 2-3. Wrong way

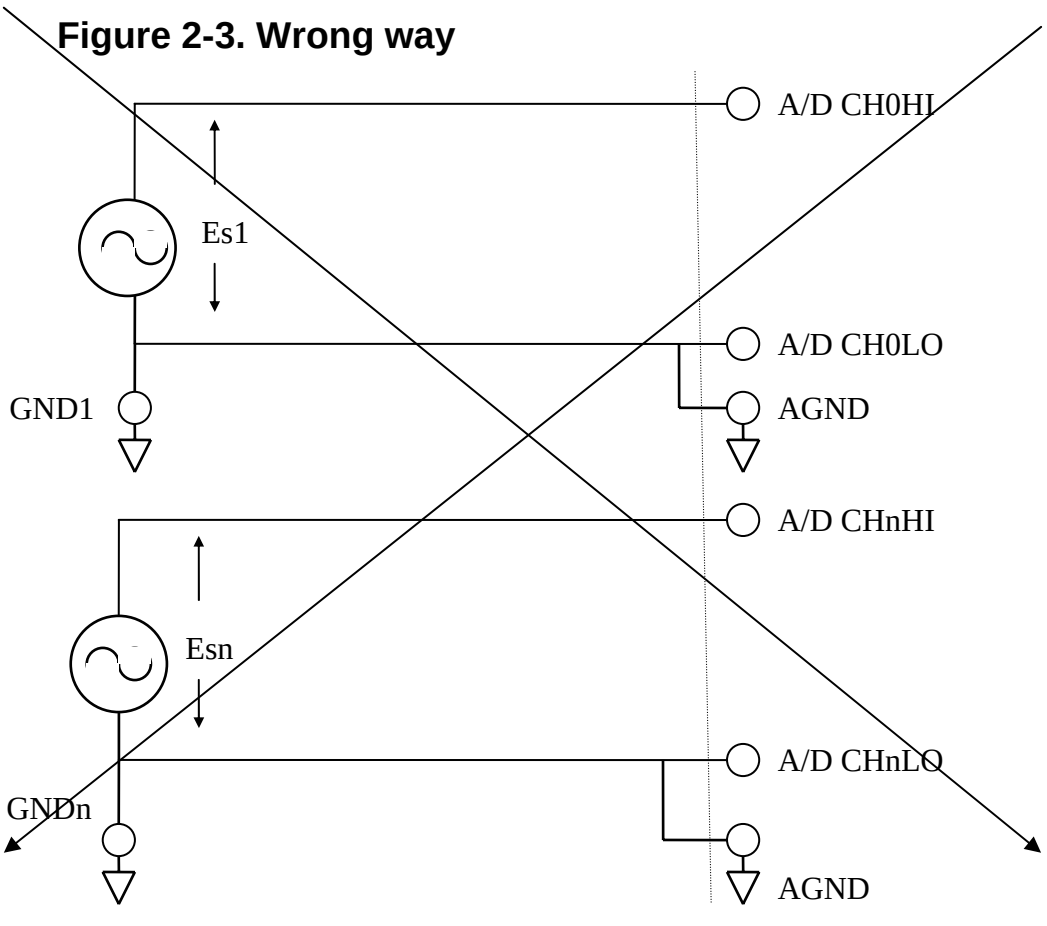


Figure 2-4. Connecting to single-ended input configuration

PCI-1002

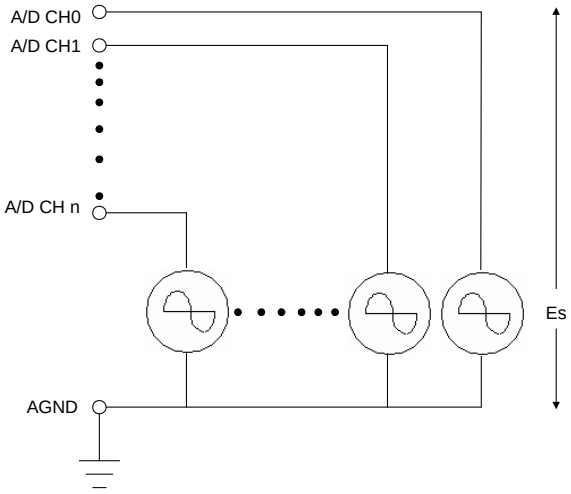
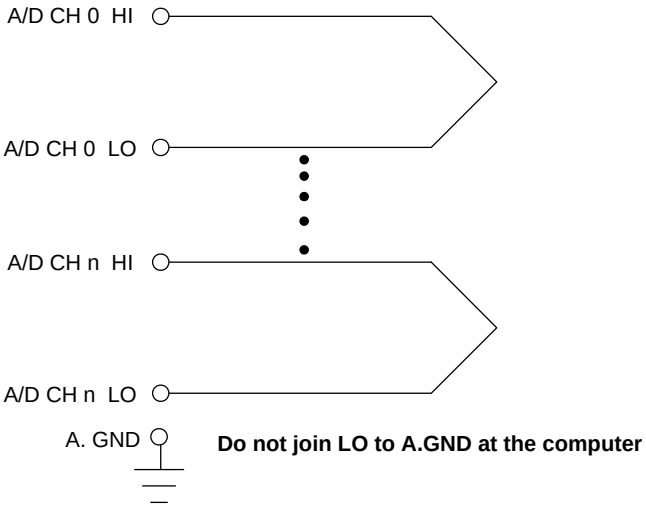


Figure 2-5. connecting to thermocouple configuration

PCI-1002

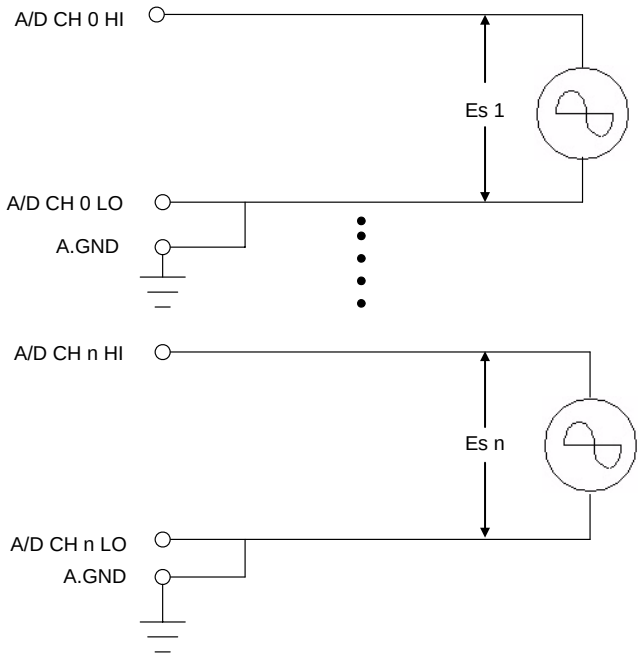


Note: If the input signal is not thermocouple, the user should use an oscilloscope to measure common mode voltage of Vin before connecting to PCI-1002. Don't use a voltage meter or multimeter.

CAUTION: In Figure 2-5, the maximum common mode voltage between the analog input source and the AGND is 70Vp-p. Make sure that the input signal is under specification first! If the common mode voltage is over 70Vp-p, the input multiplexer will be permanently damaged.

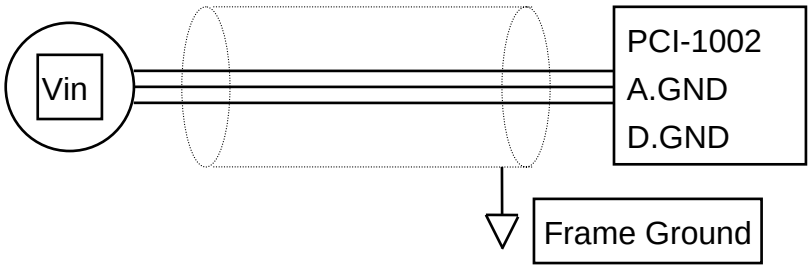
Figure 2-6. Connecting to floating source configuration

PCI-1002



Signal Shielding

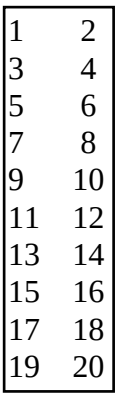
- Signal shielding connections in Figure 2-3 to Figure 2-6 are all the same
- Use a single-point connection to **frame ground (not A.GND or D.GND)**



2.7 The Connectors

CON1: Digital output connector pin assignments.

Pin	Name	Pin	Name
1	Digital output 0	2	Digital output 1
3	Digital output 2	4	Digital output 3
5	Digital output 4	6	Digital output 5
17	Digital output 6	8	Digital output 7
9	Digital output 8	10	Digital output 9
11	Digital output 10	12	Digital output 11
13	Digital output 12	14	Digital output 13
15	Digital output 14	16	Digital output 15
17	PCB ground	18	PCB ground
19	PCB +5V	20	PCB +12V



CON2: Digital input connector pin assignments.

Pin	Name	Pin	Name
1	Digital input 0	2	Digital input 1
3	Digital input 2	4	Digital input 3
5	Digital input 4	6	Digital input 5
17	Digital input 6	8	Digital input 7
9	Digital input 8	10	Digital input 9
11	Digital input 10	12	Digital input 11
13	Digital input 12	14	Digital input 13
15	Digital input 14	16	Digital input 15
17	PCB ground	18	PCB ground
19	PCB +5V	20	PCB +12V

CON3: Analog input/output connector pin assignment. (For PCI-1002H/L)

Pin	Name	Pin	Name
1	Analog input 0/0+	20	Analog input 16/0-
2	Analog input 1/1+	21	Analog input 17/1-
3	Analog input 2/2+	22	Analog input 18/2-
4	Analog input 3/3+	23	Analog input 19/3-
5	Analog input 4/4+	24	Analog input 20/4-
6	Analog input 5/5+	25	Analog input 21/5-
7	Analog input 6/6+	26	Analog input 22/6-
8	Analog input 7/7+	27	Analog input 23/7-
9	Analog input 8/8+	28	Analog input 24/8-
10	Analog input 9/9+	29	Analog input 25/9-
11	Analog input 10/10+	30	Analog input 26/10-
12	Analog input 11/11+	31	Analog input 27/11-
13	Analog input 12/12+	32	Analog input 28/12-
14	Analog input 13/13+	33	Analog input 29/13-
15	Analog input 14/14+	34	Analog input 30/14-
16	Analog input 15/15+	35	Analog input 31/15-
17	Analog ground	36	N.C.
18	N.C.	37	Digital ground
19	External trigger		

Note:

1. Once differential analog input is selected (JP1 3-5, 4-6), Pins 1-16 will be assign as the positive inputs while Pins 20-35 will act as the negative inputs of the channel.

2. ‘ N.C. ’ is short for “Not Connected”.

3. I/O Register Address

3.1 How to Find the I/O Address

The Plug&Play BIOS will assign a proper I/O address to every PCI-1002 card in the power-on stage. The IDs of PCI-1002 are as follows:

- **Vendor ID = 1234**
- **Device ID = 1002**

We provide the following necessary functions:

1. **P1002_DriverInit(&wBoard)**

   This function can detect how many PCI-1002 cards are in the system. The function is implemented based on the PCI Plug & Play mechanism-1. It will find all PCI-TMC12 cards installed in this system & save all their resources in the library.

   - wBoard=1 → only one PCI-1002 in this PC system.
   - wBoard=2 → there are two PCI-1002 in this PC system.

2. **P1002_GetConfigAddressSpace(wBoardNo, *wBase, *wIrq, *wPLX)**

   Use this function to save resources of all PCI-1002 installed in this system. Then the application program can control all functions of PCI-1002 directly.

   - wBoardNo=0 to N → totally N+1 cards of PCI-1002
   - wBase → base address of the board control word
   - wIrq → allocated IRQ channel number of this board
   - wPLX → base address of PCI-interface-IC

Here's the sample program source code:

```
/* Step1: Detect all PCI-1002 cards first */
wRetVal=P1002_DriverInit(&wBoards);
printf("Threr are %d PCI-1002 Cards in this PC\n",wBoards);

/* Step2: Save resources of all PCI-1002 cards installed in this PC */
for (i=0; i<wBoards; i++)
  {
  P1002_GetConfigAddressSpace(i,&wBase,&wIrq,&wPLX);
  printf("\nCard_%d: wBase=%x, wIrq=%x, wPLX=%x", i,wBase,wIrq,wPLX);
  wConfigSpace[i][0]=wBaseAddress;    /* save all resource of this card */
  wConfigSpace[i][1]=wIrq;            /* save all resource of this card */
  wConfigSpace[i][2]=wPLX;            /* save all resource of this card */
  }

/* Step3: Control the PCI-1002 directly */
   wBase=wConfigSpace[0][0];          /* get base address the card_0  */
   outpw(wBase+0x20,wDoValue);        /* control the D/O states of card_0 */
   wDiValue=inpw(wBase+0x20);         /* read the D/I states of card_0 */

   wBase=wConfigSpace[1][0];          /* get base address of card_1 */
   outpw(wBase+0x20,wDoValue);        /* control the D/O states of card_1 */
   wDiValue=inpw(wBase+0x20);         /* read the D/I states of card_1 */

   wPLX=wConfigSpace[2][2];       /* get PCI-interface base address of card-2 */
   _outpd(wPLX+0x4c,0x41);            /* channel_1, interrupt active_Low   */
     …
     …
     …
 _outpd(wPLX+0x4c,0);               /* disable all interrupt   */
```

3.2 The I/O Address Map

The list of PCI-1002 registers is given below. The address of each register is found by simply adding the offset to the base address of the corresponding section. More detailed descriptions of each register will be shown in the following text and the software manual.

Section	Offset	Name	Access	Length
1	4ch	PCI interrupt control register	R/W	8/16/32 bits
2	00h	8254 timer1	R/W	8/16/32 bits
	04h	8254 timer2	R/W	8/16/32 bits
	08h	8254 timer3	R/W	8/16/32 bits
	0Ch	8254 control register	W	8/16/32 bits
	10h	Analog input channel control register	W	8/16/32 bits
	10h	Status register	R	8/16/32 bits
	14h	Analog input gain control register	W	8/16/32 bits
	18h	General control register	W	8/16/32 bits
	1Ch	A/D software trigger	W	8/16/32 bits
	1Ch	Clear Interrupt	R	8/16/32 bits
	20h	Digital output register	W	16/32 bits
	20h	Digital input register	R	16/32 bits
	30h	A/D data register	R	16/32 bits

3.2.1 Section 1

Although there are 128 I/O ports used by the on-board PCI interface controller, only one register is used in real applications!! Users should keep the other registers from being modified! The PCI interrupt control register (**4Ch**) controls the interrupt sent to your system. The register is set to "disable interrupt" after power-on or a hardware reset signal. Thus, no interrupt will be generated before this register is activated even if user enables the add-on interrupt! In order to enable the PCI-interrupt, always write **43h** to this register. Write **03h** to this register if you want to disable the PCI interrupt.

Here's the format of the PCI interrupt control register:

Bit 31-Bit 7	Bit6	Bit5-Bit3	Bit2	Bit1-Bit0
Not used	Interrupt Enable	Not used	Interrupt Flag	Interrupt Select

Bit 6: Write a '1' to enable the PCI-interrupt and a '0' to disable PCI interrupt.
Bit2: This bit is readable but can't be written. A '1' indicates that Add-on has generated interrupt, '0' means that add-on hasn't generated interrupt.
Bit1-0: Always write 1 to these two bits.

Note:

1. Because PCI-1002 supports "Plug and Play", the interrupt number will be automatically assigned by your system. Use the standard PCI mechanism or the software in our library to find out the interrupt number.
2. If your system supports "Shared IRQ", several peripherals will share the same IRQ at the same time. You must use Bit-2 to find out if this IRQ was generated from your PCI-1002!!!
3. For more information about the PCI interrupt control, refers to the PLX-9050 user reference manual.

3.2.2 Section 2

This section is used by the add-on control logic. 64 bytes of I/O locations are used. Detailed descriptions are shown below.

3.2.2.1 The 8254 registers

The 8254, programmable timer/counter is used to generate periodic A/D trigger signals, periodic interrupt signals and the machine-independent timer for PCI-1002. Addresses 00h, 04h, 08h and 0Ch are used to control the 8254.

Timer 0 is used as Pacer 0. Timer 1 is used as Pacer 1. Timer 2 is used as a machine-independent timer, P1002_Dealy(). For more details about the programming information, please refer to Intel's "Microsystem Components Handbook".

3.2.2.2 The DI / DO register

Address 20h is used for DI / DO ports. Writing to this port will write data to DO register. Reading from this port will read the data from DI.

3.2.2.3 The A/D buffer

Address 30h is used for A/D buffer. Only read operations are available at this address. Reading from this port will read the data from A/D buffer. The format of A/D buffer is:

Bit15-12	Bit11-0
Analog input channel	A/D data

Bit15-12: The channel number of analog input. Only the lower 4 bits of the channel number are shown in this register.

Bit11-0: The A/D data.

3.2.2.4 The status register

Address **10h** is used by the status register. Reading from this address will get the data from the status register. The format of status register is:

Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Gain Control	8245 Timer 1	8245 Timer 0	8245 Timer 2	Reserved	Analog input type	A/D Busy

Bit 7-6: Current A/D gain control.
Bit 5 : Output of 8254 timer 1.
Bit 4 : Output of 8254 timer 0.
Bit 3 : Output of 8254 timer 2.
Bit 2 : Reserved. Used for hardware testing.
Bit 1 : Analog input type, '1' indicated that analog input type is single-ended and '0' indicated analog input is differential.
Bit 0 : The A/D busy signal. '0' indicates busy, A/D is under conversion. '1' indicates not busy, A/D is completely converted and is idle now.

3.2.2.5 The A/D software trigger register

Writing to this port (**1Ch**) will generate an A/D trigger pulse signal.

Note: Although a very fast trigger can be performed (more than the speed of A/D controller, 125K) via this method, a reasonable delay time should be left between the two triggers.

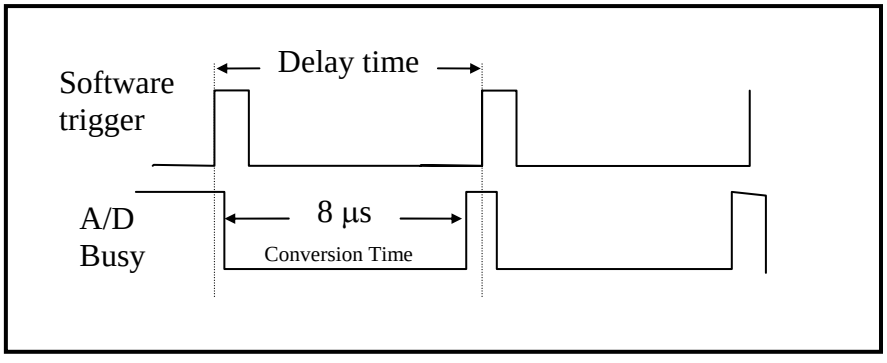


Figure 3-1. Software trigger delay time.

3.2.2.6 Clear interrupt

Reading from 1Ch will clear the add-on interrupt.

3.2.2.7 The analog input selection register

Address **10h** is used by the analog input channel selection register and address **14h** is used by the analog gain control selection register. Write 0-31 to port **10h** to select the channel number (for differential input, write 0-15). Write 0-3 to port **14h** to select the gain control.

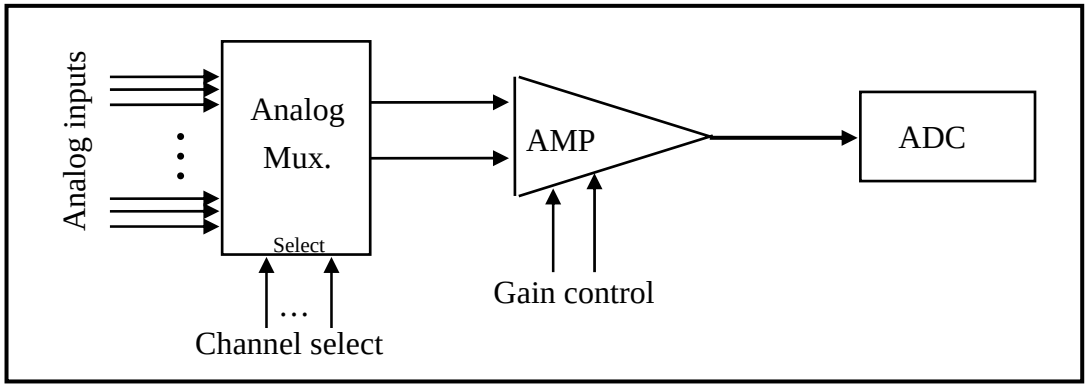


Figure 3-2. Analog input control.

Note:

1. For single ended inputs, channels 0-31 are available. For differential inputs, channels 0-15 are available. Input numbers which are more than the available channel will be discarded. Thus, for single-ended inputs, only the last 5 bits are taken as the channel number. And for differential inputs, only the last 4 bits are taken as the channel number.
2. Only the last two digits are taken as the gain control code. The gain control code and the corresponding gain is:

For PCI-1002L:

Gain code	[0 0]	[0 1]	[1 0]	[1 1]
Gain	1	2	4	8

For PCI-1002H:

[Bit1, Bit 0]	[0 0]	[0 1]	[1 0]	[1 1]
Gain	1	10	100	1000

3. These registers are set to 0 after powered-on or hardware reset signals.

3.2.2.8 The general control register

A general control register (18h) is used to control the add-on interrupt signal source and the A/D trigger method. The format of this register is:

Bit4-2	Bit 1-0
Interrupt source selection register	A/D trigger method selection register

3.2.2.8.1 Interrupt source selection

There are four interrupts selectable for PCI-1002 (see section 1.4.4).

[Bit4,Bit3,Bit2]	Descriptions
[0, 0, 0]	No interrupt source, disable all interrupts.
[0, 0, 1]	Interrupt after A/D completes conversion.
[0, 1, 0]	Interrupt after 8254 timer 0 falls.
[0, 1, 1]	Interrupt after external trigger falls.
[1, 0, 0]	Interrupt after 8254 timer 1 falls.
Others	No interrupt source, Disable all interrupts.

Note: Bit 3-3 of general control register is set to 0 after hardware reset.

3.2.2.8.2 Trigger method selection

Here's a list of our trigger options (see section 1.4.5):

[Bit1,Bit0]	Descriptions
[0, 0]	General trigger mode. 8254 timer 0 trigger (internal pacer trigger) or software trigger.
[0, 1]	External clock trigger mode.
[1, 0]	Pre-trigger mode.
[1, 1]	Post-trigger mode.

Note:

1. In general trigger mode, both 8254 timer 0 and software triggers are treated as A/D trigger signals. In this mode, 8254 timer 0 and software trigger should not working at the same time!! This means **users should not generate the software trigger while 8254 timer 0 is activated**!!
2. In external clock trigger mode, external trigger input is taken as the A/D trigger signal. If the external trigger input falls(falling edge),will generate one A/D trigger.
3. The pre-trigger mode is used for pre-trigger method. The mode is incorporated with the 8254 timer 1. First, setup 8254 timer 1 properly. Then set the trigger mode to pre-trigger. Once pre-trigger mode has been activated, the 8254 timer 1 will automatically turn on and start to perform A/D triggers. It will continue until the A/D trigger logic receives a falling external trigger signal. Any change to the trigger mode selection will turn off the pre-trigger mode.
4. The post-trigger mode is used for post-trigger method. The mode working incorporated with the 8254 timer 1. First, setup 8254 timer 1 properly. Then set the trigger mode to post-trigger. Once post-trigger mode has been activated, the 8254 timer 1 will automatically turn off until it receives a falling external trigger signal. Any change to the trigger mode selection will turn off the post-trigger mode.
5. The A/D trigger is set to 0 after either power-on or hardware reset.

4. Function Operations

4.1 Digital I/O

The PCI-1002 series provide 16 digital input channels and 16 digital output channels. All levels are TTL compatible. The connection diagram and block diagram are given below:

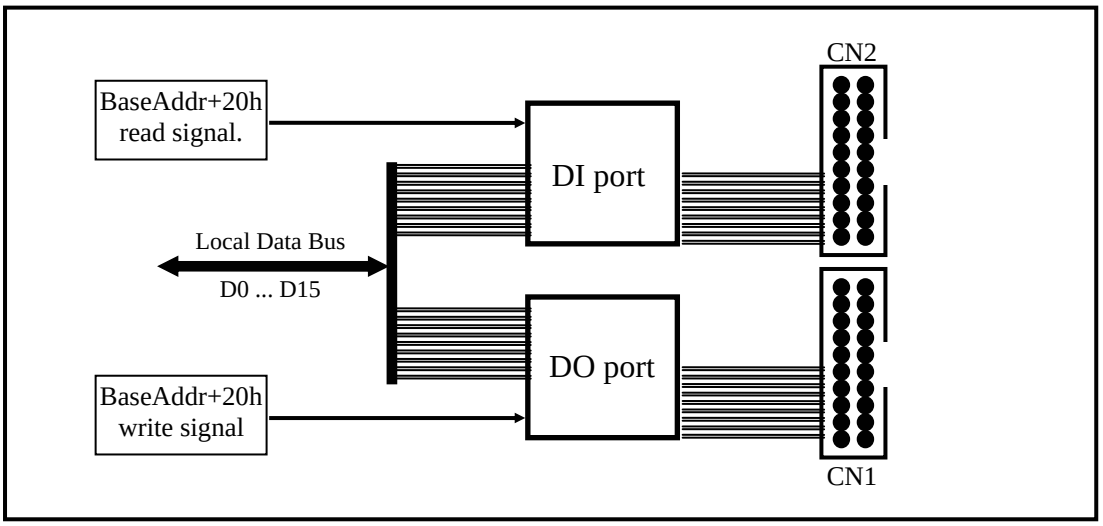


Figure 4.1. DIO function diagram.

4.2 The 8254 Timer

The PCI-1002 series provide 3 independent,16-bit timer/counters. Each timer has different functions. **Timer 0 is uses Pacer 0. Timer 1 is uses Pacer 1. Timer 2 is uses a machine independent timer.** The block diagram is given as follows:

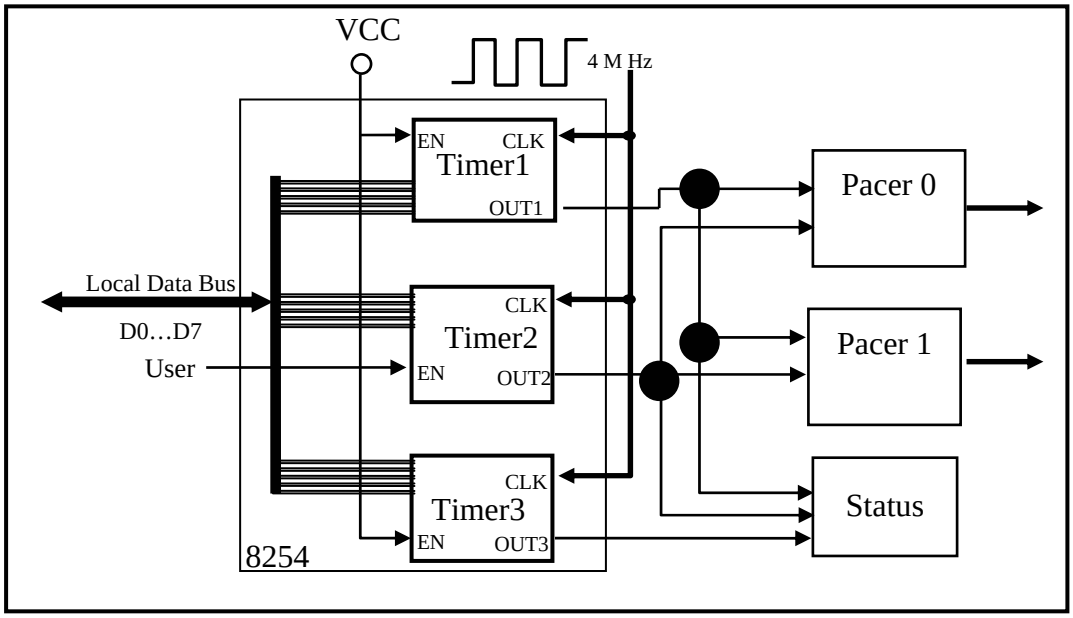


Figure 4-2. 8254 control diagram.

4.3 The A/D trigger

The A/D trigger is controlled by on-board A/D trigger controller. The function diagram of A/D trigger is shown below:

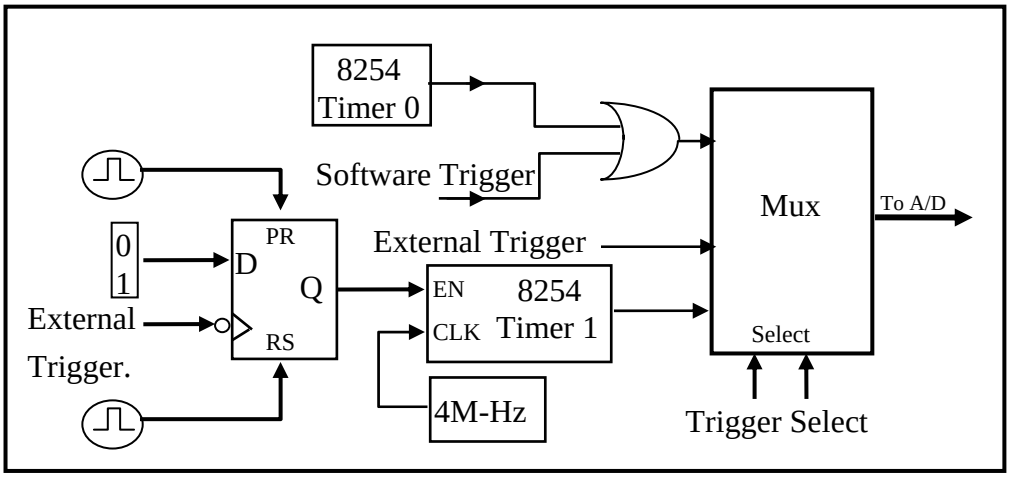
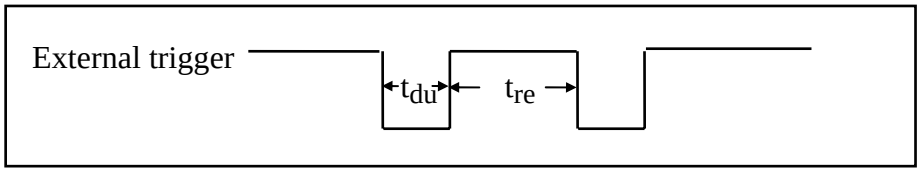


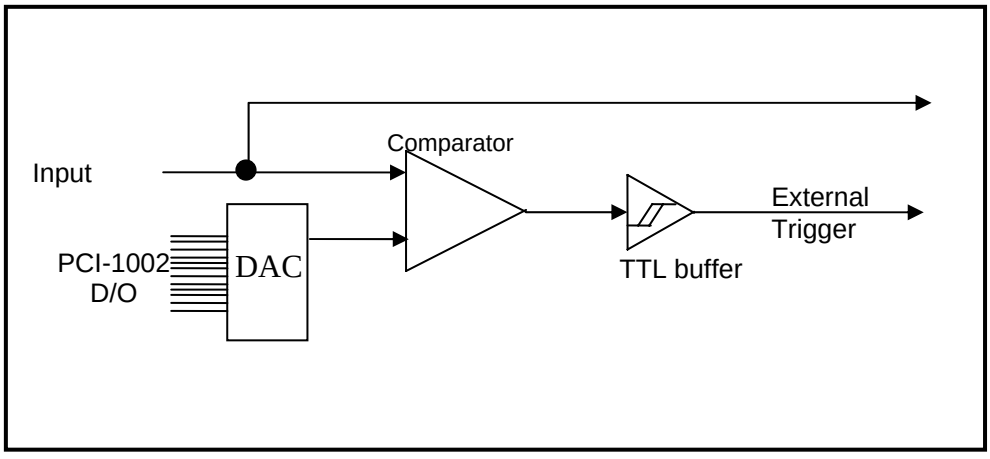
Figure 4-3. A/D trigger controller.

A/D trigger logic receives the external trigger and then performs the correct A/D trigger function. In order to be recognized by the A/D trigger controller, the external trigger signal must be a TTL compatible signal, with the minimum duration of pulse width to avoid noise. This signal must satisfy the following specifications:



Symbol	Name	Minimum	Maximum
T_{du}	Duration time	40ns	∞
T_{re}	Recover time	100ns	∞

Note: The PCI-1002 is designed only as a time sensitive trigger (trigger depends only on receiving a falling edge external trigger signal). For a level sensitive external trigger (trigger depends only on the level of the input signals), make the following circuit outside the PCI-1002:

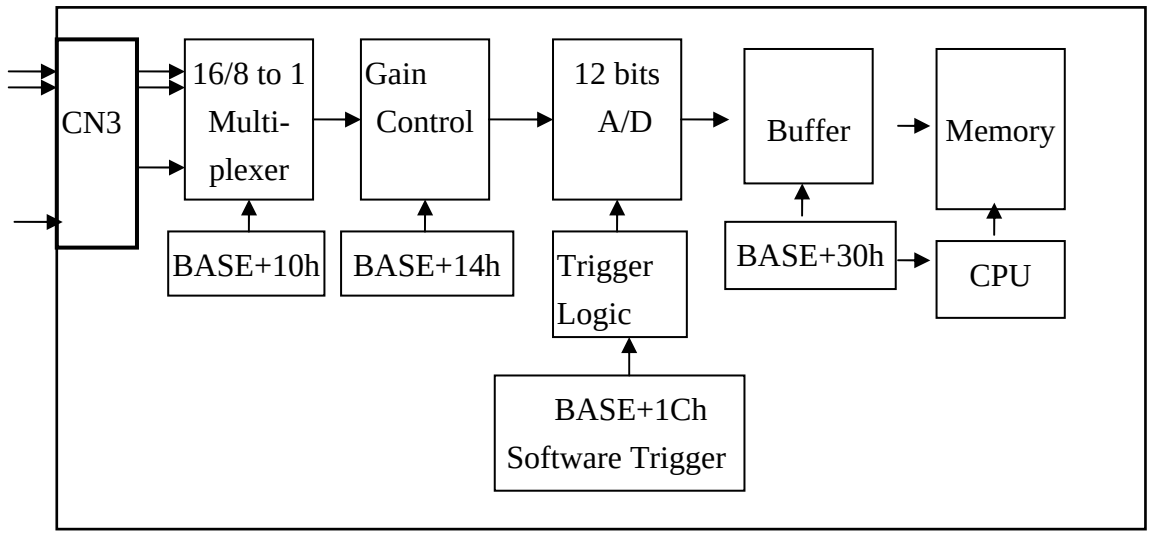


4.4 A/D Conversion

An A/D conversion can be initiated in one of three ways: software command, internal programmable interval timer, or by external trigger to the A/D. At the end of the A/D conversion, it is possible to transfer the data in one of three ways: by polling a status register and reading data when ready, by generating a hardware interrupt, or by interrupting the service routine. All operating modes are selected by a control register on the PCI-1002 and are supported by the utility software.
Before using the A/D conversion function, please follow this checklist:

- A/D data register (BASE+30h)=> store the A/D conversion data.
- A/D data conversion ready register (BASE +10hH)=> Check A/D conversion ready.
- A/D gain control register(BASE+14h)=> Select gain.
- A/D multiplex control register(BASE+10h)=>Select analog input.
- A/D mode control register(BASE+0Ch)=>Select trigger type and transfer type.
- A/D software trigger control register(BASE+1Ch)**.**
- JP1 to select single-ended or differential input.
- **3 Trigger logic: Software, Pacer, or External trigger.**
- **2 Transfer logic: Polling, or Interrupt.**

Here's the block diagram:



PCI-1002H/L

A/D conversion flow

Before using the A/D converter, please select either single-ended or differential input (JP1).

The software driver supports two different modes: **polling and interrupt.** The user can control the A/D conversion by polling mode very easy. Using the software driver is recommended if using interrupt.

The multiplexer can select 16 single-ended or 8 differential signals into the gain control module. T**he settling time of multiplexer depends on the impedance of the signal source.** Because the software **doesn't control the settling time**, **please make sure to leave enough settling time if switching from one channel to the next channel.**

The gain control module also needs settling time if gain control code changed. Again, because the software **doesn't control settling time**, **please delay enough settling time if the gain control code is changed.**

Remember to delay the extra setting time when gain of channel is changed.

The software driver provides **a machine independent timer, P1002_Delay()**, for settling time delays. This subroutine assumes a machine independent timer will be implemented. However, if using call P1002_Delay(), the counter 0 will be reserved and can't be used as a user programmable timer/counter.

The gain control module's output feeds into the A/D converter. **The A/D converter needs a trigger signal to start a A/D conversion cycle**. The PCI-1002 supports three trigger modes, **software, pacer and external trigger**.

4.4.1 A/D Conversion Trigger Modes

A-826PG supports three trigger modes.

1 : Software Trigger :

Write any value to the A/D software trigger control register, BASE+1Ch, and it will initiate a A/D conversion cycle. This mode is very simple but controlling the sampling rate very difficult.

2 : Pacer Trigger Mode :

See section 3.2 for a block diagram for this pacer timer. The sample rate of pacer is very precise.

3 : External Trigger Mode :

When a rising edge of external trigger signal is applied, a A/D conversion will be performed. The external trigger source comes from Pin-17 of CON3.

4.4.2 A/D Transfer Modes

PCI-1002 supports two transfer modes.

1 : Polling transfer :

This mode can be used with all trigger modes. You have to **disable timer 0** before polling. The software reads the A/D data register from BASE+30h, Register BASE +10h READY_BIT=0.

2 : Interrupt transfer:

This mode can be used with either a pacer trigger or external. A hardware interrupt signal is sent to the PC when a A/D conversion is completed.

If using interrupt transfer, it is recommended to use PCI-1002 software driver.

4.4.3 Software triggers and Polling techniques

The easiest way to control is by following these steps:

1. Send 00h to A/D mode control register (software trigger + polling transfer)
2. Send channel number to multiplexer control register.
3. Send the gain control code value to gain control register.
4. Send any value to software trigger control register to generate a software trigger signal.
5. Scan the READY bit of the A/D high byte data until READY=0
6. Read the 12-bit A/D data.
7. Convert this 12-bit binary data to the floating point value.

For example:

```
/* ---------------------------------------------------------------- */
/* DEMO 3: AdPolling                                                */
/* Compiler: Borland C++ 3.1, Mode Large                            */
/* Output Code: HEX code                                            */
/* ---------------------------------------------------------------- */
#include "P1002.H"
WORD wBaseAddr,wIrq;
//--------------------------------------------------------
WORD P1002_Delay(WORD wDownCount)
{
  WORD h,l;
  int count;
  wDownCount &= 0x7fff;
  if (wDownCount<1) wDownCount=1;
 /* Clock in=4M --> count 4000 = 1 ms, count 1 = 0.25 us */

  l=wDownCount&0xff;
  wDownCount=wDownCount / 256;
  h=wDownCount&0xff;

  outp(wBaseAddr+3*4,0xB0);    /* mode_0, counter_2 */
  outp(wBaseAddr+2*4,l);       /* counter_2 low byte first */
  outp(wBaseAddr+2*4,h);       /* counter_2 high byte ,0x07D0=2000 */
```

```
   outp(wBaseAddr+3*4,0x80);         /* latch counter_2 */
  l=inp(wBaseAddr+2*4);              /* delay starting two clks */
  h=inp(wBaseAddr+2*4);

  for (count=32767;count>0;count--){
          outp(wBaseAddr+12,0x80);       /* latch counter_2 */
          l=inp(wBaseAddr+8);
          h=inp(wBaseAddr+8);
          if (h>=0x80) return NoError;
  }
  return TimeOut;
}

//-------------------------------------------------------
void AdPolling(UCHAR channel, UCHAR gain, WORD delay)
    {
     outp(wBaseAddr+0x18,0);       // Select Mode 0
     outp(wBaseAddr+0x10,channel);
     outp(wBaseAddr+0x14,gain);
     P1002_Delay(delay);
     outp(wBaseAddr+0x1c,01);       // A/D software tirgger
    }

void SetupTimer(WORD wChannel, WORD wCoef)
{
   WORD   cmd;
   wChannel=wChannel&0x03;
   cmd=0x34+(wChannel<<6);
   outpw(wBaseAddr+3*4, cmd);
   outp(wBaseAddr+wChannel*4, (UCHAR)(wCoef&0xff));
   outp(wBaseAddr+wChannel*4, (UCHAR)(wCoef>>8));
}

//=========================================================
void main()
 {
 int  i,j;
 WORD wBoards,wRetVal,wPLX;
```

```
 WORD Drdy,wAdData=0;
 char c;

 clrscr();
 P1002_DriverInit(&wBoards);
 printf("\n(1) Threr are %d PCI-1002 Cards in this PC",wBoards);
 if ( wBoards==0 )
  {
  putch(0x07);  putch(0x07);  putch(0x07);
  printf("(1) There are no PCI-1002 card in this PC !!!\n"); exit(0);
  }

printf("\n(2) Show the Configuration Space of all PCI-1002:");
for(i=0; i<wBoards; i++)
  {
  P1002_GetConfigAddressSpace(i,&wBaseAddr,&wIrq,&wPLX);
  printf("\n    Card_%d: wBaseAddr=%x, wIrq=%x, wPLX=%x",i,wBaseAddr,wIrq,wPLX);
  }

 P1002_GetConfigAddressSpace(0,&wBaseAddr,&wIrq,&wPLX); /* select card_0 */
 printf("\n(3) *** Card_0, wBaseAddr=%x ***\n",wBaseAddr);
 SetupTimer(0,1);            // AdPolling have to disable timer 0
 AdPolling(0,0,23);          // channel=0, gain=+/-10, delay=23us

for(i=0;i<10;i++)
  {
   outp(wBaseAddr+0x1c,01);    // A/D software tirgger
   while(1)
       {
        if( ((inpw(wBaseAddr+0x10))&0x01)==1) // check if A/D busy?
        break;
       }
    wAdData=((inpw(wBaseAddr+0x30))&0x0fff);
   printf("\nRang:+/-10V, Counter %d ,ADC channel 0 value: 0x%xH",i,wAdData);
 }
P1002_DriverClose();
}
```

5. Software and Demo Program

1. Demo program for DOS

- …\1002\BC\LARGE\DEMO> ← demo program
- …\1002\BC\LARGE\LIB> ←library and driver

- DEMO1: Digital output.
- DEMO2: Digital output and Digital input test by itself.
- DEMO3: ADC Polling for channel 0.
- DEMO4: ADC Polling for channel 0,1,2,3 using different gains-1,2,4 or 8.
- DEMO5: ADC Pacer trigger.
- DEMO6: AD Calibration.
- DEMO7: Find card number.

2. Demo program for Windows95/98/NT/2000/XP and LINUX

- Refer to CD-ROM.

6. Diagnostic Program

6.1 Power-ON Plug & Play Test

The operation steps for a power-on Plug & Play test are as follows:
Step 1: Power-off PC
Step 2: Install PCI-1002 without any extra external connector
Step 3: Power-on PC and check the PC screen very carefully
Step 4: The PC will perform a self-test first
Step 5: Detect the non-PCI physical devices installed in the system
Step 6: Show the information of these device in screen
Step 7: Detect the PCI Plug & Play devices installed in the system
show all PCI-device information → check here carefully
→ There will be a PCI device with vendor_ID=1234, device_ID=1002 (PCI-1002)

If the Plug & Play ROM-BIOS detects the PCI-1002 card during the power-on time, the software driver of DOS,and Windows 95/NT/2000/XP will function well later. If the Plug & Play ROM-BIOS can't find the PCI-1002, all software drivers will not function. Therefore the user must make sure that the power-on procedure is correct.

6.2 Driver Plug & Play Test

Step 1: Power-off PC.
Step 2: Install PCI-1002 without any extra external connectors.
Step 3: Power-on PC. Run DEMO7.EXE.
Step 4: The I/O base address of all PCI-1002 installed in the system will be shown in screen.
Step 5: Is the total number of boards correct?
Step 6: Install a 20-pin flat cable into one of these PCI-1002 cards.
Step 7: One card's D/O=D/I → This is the physical card number. Remember this number.
Step 8: Repeat the previous two steps to find the physical card number for each board.

6.3 D I/O Test

Step 1: Power-off PC.
Step 2: Install one PCI-1002 card with a 20-pin flat cable between CON1&CON2.
Step 3: Power-on PC, Then run DEMO2.EXE.
Step 4: The DO and DI will show either TEST OK or TEST ERROR.

6.4 A/D Test

- A/D Test for PCI-1002 card

Step 1: Power-off PC.
Step 2: Install one PCI-1002 card.
Step 3: Power-on PC, run DEMO6.EXE
Step 4: Apply +10V to channel 0.
Step 5: Apply +0V to channel 1.
Step 6: Apply -10V to channel 2.
Step 7: Run DEMO6.EXE.
Step 8: Check channel 0 = fff or ffe?
Step 9: Check channel 1 = 800 or 801?
Step 10: Check channel 2 = 000 or 001?