NuDAQ®

PCI-8554

Multi-functions Counter / Timer Card

User's Guide

Manual Rev. 1.10: April 11, 2000

Getting service from ADLINK

- Customer Satisfaction is always the most important thing for ADLINK Tech Inc. If you need any help or service, please contact us and get it.

ADLINK Technology Inc.			
Web Site	http://www.Adlink.com.tw		
Sales & Service	service@Adlink.com.tw		
Technical Support	NuDAQ	nudaq@Adlink.com.tw	
	NuDAM	nudam@Adlink.com.tw	
	NuIPC	nuipc@Adlink.com.tw	
	NuPRO	nupro@Adlink.com.tw	
	Software	sw@Adlink.com.tw	
	AMB	amb@Adlink.com.tw	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan, R.O.C.		

- Please inform or FAX us of your detailed information for a prompt, satisfactory and constant service.

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			
Questions			
Product Model			
Environment to Use	OS_____ Computer Brand _____ M/B: CPU: Chipset: Bios: Video Card: Network Interface Card: Other:		
Challenge Description			
Suggestions for ADLINK			

Table of Contents

How to Use This Guide

This manual is designed to help you use the PCI-8554. The manual describes how to modify various settings on the PCI-8554 card to meet your requirements. It is divided into 5 chapters:

- Chapter 1, "Introduction," gives an overview of the product features, applications, and specifications.
- Chapter 2, "Installation & Configurations" describes the operation method and multi-functions of the PCI-8554. Users should read through this chapter to understand the configurations of the PCI-8554. The chapter will also teach user how to install the PCI-8554.
- Chapter 3, "Register Format," describes the details of register format of the PCI-8554, this information is very useful for the programmers who want to control the hardware by low level programming.
- Chapter 4, "Signal Connection & Applications," describes the connectors' pin assignment and how to connect the outside signal and devices to / from the PCI-8554. Some applications also are introduced.
- Chapter 5, "High-level Programming," introduces the C-language library for operating the PCI-8554. Some examples are shown too.

1

Introduction

PCI-8554 is a general-purpose counter / timer and digital I/O card. This card have four 8254 chips on board, so it provides twelve 16 bits down counter or frequency dividers.

This card has multi-configurations. The counters can be set as independent counter or cascaded counter. The gate control of counter come from either external source or internal default enable signal. The clock source of the counters can be set as internal or external clock source, when external clock is used, user can set the jumper to decide whether the debounce function is used or not used. An 8 MHz crystal is used as internal clock source. It is possible to use this card on variety of powerful counter / timer functions to match your industry and laboratory applications. Users can set the configuration to fit the variety of applications.

The card also provides digital output and input port. There are 8 bits digital output and 8 bits digital input channels which can be used to control or monitor the external devices.

PCI-8554 provides one interrupt signal which comes from internal or external interrupt sources, the internal interrupt sources come from the counter output. The interrupt can be used for watchdog timer or others applications. The maximum interrupt time interval can be 536 sec.

The I/O signals are via a 100 pin SCSI-II connector that project through the computer case at the rear of the board. The figure 1.1 shows the block diagram of the PCI-8554.

PCI-8554 uses ASIC PCI controller to interface the board to the PCI bus. The ASIC fully implement the PCI local bus specification Rev 2.0. All bus relative configurations, such as base memory and interrupt assignment, are automatically controlled by BIOS software. It does not need any user interaction and pre-study for the configurations. This removes the burden of searching for a conflict-free configuration, which can be very time consuming and difficult with some other bus standards.

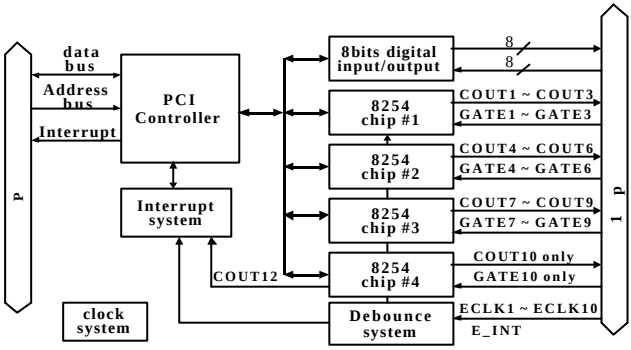


Figure 1.1. Block diagram of the PCI-8554

1.1 Features

The PCI-8554 Counter / Timer and digital I/O Card provides the following advanced features:

- Four 8254 chips provide twelve 16 bits down counters
- Multi-configurations of counters / timers:
- Flexible setting for each independent counter, the clock source could be external, internal or cascaded. The gate signal is external controlled or internal enabled.
- Provide debounce function with flexible setting to prevent from bounce phenomenon when using external clock.
- 8 digital output channels
- 8 digital input channels
- Dual interrupt sources
  The first interrupt source comes from output of counter #12

The second interrupt source comes from external source.

- 100-pin SCSI-II female connector.
- PCI-Bus

1.2 Applications

- Event counter
- Frequency generator
- Frequency synthesizer
- Pulse width measurement
- Low level pulse generator
- Time delay
- Industry automation
- Watchdog timer

1.3 Specifications

- **Programmable Counter / Timer**
  - **Device :** 82C54x4
  - **Number of Counters /timers:**
    10 independent timers / counters
    Cascaded 32-bit counters with fixed 8MHz internal clock
  - **Counter mode:** 16-bit down counter
  - **Maximum input frequency:** 10MHz
  - **Clock sources of independent counters:**
    External clock
    Prior counter output
    CK1 (Programmable)
    Clock #10 output
  - **CK1 clock sources: (Programmable)**
    8MHz internal base clock
    Programmable counter 11 output
  - **Gate control:** default enable or external control
- **Digital Filter Circuits**
  - **Device: MC14490**
  - **De-bounce clock: (Programmable)**
    8MHz internal base clock
    Programmable counter 11 output
- **Digital I/O (DIO)**
  - **# of input channels :** 8 channels
  - **# of output channels :** 8 channels (dedicated output)

- **Electronics characteristics:** TTL compatible signal

¨ **General Specifications**

- **Connector**: 100-pin SCSI-II female connector
- **Operating Temperature**: 0°C ~ 60°C
- **Storage Temperature**: -20°C ~ 80°C
- **Humidity**: 5 ~ 95%, non-condensing
- **Power Consumption**: +5 V @ 350 mA typical
- **Dimension**: 134mm(L) X 107mm(W)

1.4 Software Supporting

ADLink provides versatile software drivers and packages for users' different approach to built-up a system. We not only provide programming library such as DLL for many Windows systems, but also provide drivers for many software package such as LabVIEW®, HP VEE™, DASYLab™, InTouch™, InControl™, ISaGRAF™, and so on.

All the software options are included in the ADLink CD. The non-free software drivers are protected with serial licensed code. Without the software serial number, you can still install them and run the demo version for two hours for demonstration purpose. Please contact with your dealer to purchase the formal license serial code.

1.4.1 Programming Library

For customers who are writing their own programs, we provide function libraries for many different operating systems, including:

- DOS Library: Borland C/C++ and Microsoft C++, the functions descriptions are included in this user' s guide.
- Windows 95 DLL: For VB, VC++, Delphi, BC5, the functions descriptions are included in this user' s guide.
- PCIS-DASK: Include device drivers and DLL for Windows 98, Windows NT and Windows 2000. DLL is binary compatible across Windows 98, Windows NT and Windows 2000. That means all applications developed with PCIS-DASK are compatible across Windows 98, Windows NT and Windows 2000. The developing environment can be VB, VC++, Delphi, BC5, or any Windows programming language that allows calls to a DLL. The user' s guide and function reference manual of

PCIS-DASK are in the CD. Please refer the PDF manual files under \\Manual_PDF\Software\PCIS-DASK

The above software drivers are shipped with the board. Please refer to the "Software Installation Guide" to install these drivers.

1.4.2 PCIS-LVIEW: LabVIEW® Driver

PCIS-LVIEW contains the VIs, which are used to interface with NI's LabVIEW® software package. The PCIS-LVIEW supports Windows 95/98/NT/2000. The LabVIEW® drivers are free shipped with the board. You can install and use them without license. For detail information about PCIS-LVIEW, please refer to the user's guide in the CD.
(\\Manual_PDF\Software\PCIS-LVIEW)

1.4.3 PCIS-VEE: HP-VEE Driver

The PCIS-VEE includes the user objects, which are used to interface with HP VEE software package. PCIS-VEE supports Windows 95/98/NT. The HP-VEE drivers are free shipped with the board. You can install and use them without license. For detail information about PCIS-VEE, please refer to the user's guide in the CD.
(\\Manual_PDF\Software\PCIS-VEE)

1.4.4 DAQBench™: ActiveX Controls

We suggest the customers who are familiar with ActiveX controls and VB/VC++ programming use the DAQBench™ ActiveX Control components library for developing applications. The DAQBench™ is designed under Windows NT/98. For more detailed information about DAQBench, please refer to the user's guide in the CD.
(\\Manual_PDF\Software\DAQBench\DAQBench Manual.PDF)

2

Installation

This chapter describes the configurations and multi-functions of the PCI-8554 and teach user to install PCI-8554. At first, the contents in the package and unpacking information that you should care about are described, then versatile configurations of PCI-8554 are introduced so that you can configure it according to your applications. The default jumper setting of PCI-8554 is shown in this chapter also.

2.1 What You Have

In addition to this *User's Manual*, the package includes the following items:

- PCI-8554 Enhanced Multi-function Counter / Timer Card
- ADLINK CD
- Software Installation Guide

If any of these items is missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton in case you want to ship or store the product in the future.

2.2 Unpacking

Your PCI-8554 card contains sensitive electronic components that can be easily damaged by static electricity. The card should be unpacked on a grounded anti-static mat. The operator should be wearing an anti-static wristband, grounded at the same point as the anti-static mat.

Inspect the card module carton for obvious damage. Shipping and handling may cause damage to your module. Be sure there are no shipping and handing damages on the module before processing.

After opening the card module carton, extract the system module and place it only on a grounded anti-static surface component side up.

Again inspect the module for damage. Press down on al the socketed IC's to make sure that they are properly seated. Do this only with the module place on a firm flat surface.

Note: DO NOT APPLY POWER TO THE CARD IF IT HAS BEEN DAMAGED.

You are now ready to install your PCI-8554.

2.3 PCB Layout of PCI-8554

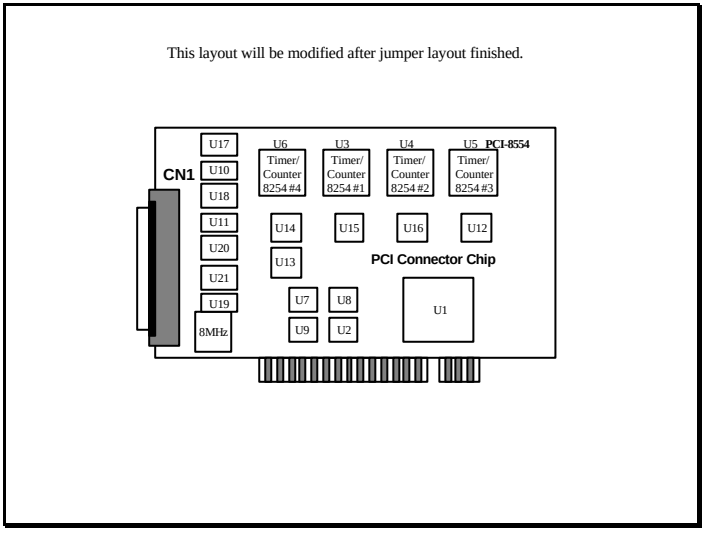


Figure 2.1 PCB Layout of PCI-8554

2.4 Default Jumper Setting

To operate the PCI-8554 correctly, users should understand the structure of PCI-8554 and details of the possible configurations. The block diagram of the PCI-8554 is shown in chapter 1. It contains the clock system, counters confederation, interrupt system and PCI controller. The following sections teach you the jumper setting and the default setting listed in Table 2.1.

Items	Default Configuration	Setting by:
ECLK1	Debounce function	JP1
ECLK2	Debounce function	JP2
ECLK3	Debounce function	JP3
ECLK4	Debounce function	JP4
ECLK5	Debounce function	JP5
ECLK6	Debounce function	JP6
ECLK7	Debounce function	JP7
ECLK8	Debounce function	JP8
ECLK9	Debounce function	JP9
ECLK10	Debounce function	JP10
E_INT	Debounce function	JP11

Table 2.1 Default Configuration of PCI-8554

There are eleven jumpers on PCI-8554, these jumpers are used to select debounce function. You can change PCI-8554's default configuration by setting jumpers on the card for your own applications. The card's jumpers are preset at the factory.

Before changing the default configuration, users must fully understand the operation of the debounce function. The setting and the basic operation theorem are not discussed in this chapter. It is recommended to refer chapter 2.12 for details of the operation theorem and to refer chapter 4 for application notes.

2.5 PCI-8554 Installation Outline

Hardware configuration

The PCI cards (or CompactPCI cards) are equipped with plug and play PCI controller, it can requests base addresses and interrupt according to PCI standard. The system BIOS will install the system resource based on the PCI cards' configuration registers and system parameters (which are set by system BIOS). Interrupt assignment and memory usage (I/O port locations) of the PCI cards can be assigned by system BIOS only. These system resource assignment is done on a board-by-board basis. It is not suggested to assign the system resource by any other methods.

1. PCI slot selection

The PCI card can be inserted to any PCI slot without any configuration for system resource.

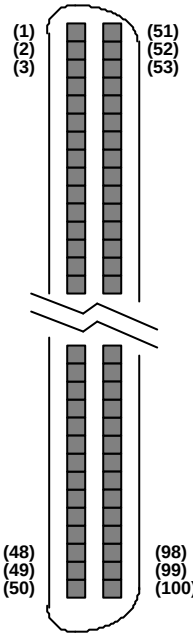
2. Installation Procedures

1. Turn off your computer
2. Turn off all accessories (printer, modem, monitor, etc.) connected to your computer.
3. Remove the cover from your computer.
4. Setup jumpers on the PCI or CompactPCI card.
5. Select a 32-bit PCI slot. PCI slot are short than ISA or EISA slots, and are usually white or ivory.
6. Before handling the PCI cards, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge and do not touch the components.
7. Position the board into the PCI slot you selected.
8. Secure the card in place at the rear panel of the system.

2.6 Device Installation for Windows Systems

Once Windows 95/98/2000 has started, the Plug and Play function of Windows system will find the new NuDAQ/NuIPC cards. If this is the first time to install NuDAQ/NuIPC cards in your Windows system, you will be informed to input the device information source. Please refer to the "***Software Installation Guide***" for the steps of installing the device.

2.7 Pin Assignment of Connector



(1) +12V	(26) GND	(51) GND	(76) ECLK9
(2) +12V	(27) GND	(52) GOUT2	(77) COUT8
(3) +12V	(28) GND	(53) GIN2	(78) GATE8
(4) Vcc	(29) GND	(54) GND	(79) ECLK8
(5) Vcc	(30) GND	(55) GOUT1	(80) COUT7
(6) Vcc	(31) GND	(56) GIN1	(81) GATE7
(7) NC	(32) GND	(57) E_INT	(82) ECLK7
(8) DI_6	(33) GND	(58) DI7	(83) COUT6
(9) DI_4	(34) GND	(59) DI5	(84) GATE6
(10) DI_2	(35) GND	(60) DI3	(85) ECLK6
(11) DI_0	(36) GND	(61) DI1	(86) COUT5
(12) DO_6	(37) GND	(62) DO7	(87) GATE5
(13) DO_4	(38) GND	(63) DO5	(88) ECLK5
(14) DO_2	(39) GND	(64) DO3	(89) COUT4
(15) DO_0	(40) GND	(65) DO1	(90) GATE4
(16) NC	(41) GND	(66) NC	(91) ECLK4
(17) GND	(42) GND	(67) COUT12	(92) COUT3
(18) GND	(43) GND	(68) GND	(93) GATE3
(19) GND	(44) GND	(69) COUT11	(94) ECLK3
(20) GND	(45) GND	(70) GND	(95) COUT2
(21) GND	(46) GND	(71) COUT10	(96) GATE2
(22) GND	(47) GND	(72) GATE10	(97) ECLK2
(23) GND	(48) GND	(73) ECLK10	(98) COUT1
(24) GND	(49) GND	(74) COUT9	(99) GATE1
(25) GND	(50) GND	(75) GATE9	(100) ECLK1

Legend:

ECLK n	: External clock source for counter #*n*
ExtG n	: External gate signal for counter #*n*
COUT n	: Counter / Timer output of counter #*n*
DO_m	: Digital output port channel #*m*
DI_ m	: Digital input port channel #*m*
E_int	: External interrupt signal input
GOUT1	: Inverse TTL signal of **GIN1**
GOUT2	: Inverse TTL signal of **GIN2**

Figure 2.2 Pin Assignment of Connector CN1

2.8 Clock System

The clock system of PCI-8554 provides the internal clock source for the 8254 chips. The clock of counter / timer #1 ~ #10 can be one of the 4 sources: ***external clock source*** or ***cascaded source*** from the ' last' channel or **CK1** or **COUT10**. The next section will give you detail description about setting clock for each counter / timer and definition of **CK1**. The clock of counter / timer #11 is fixed at 8Mhz, and clock of counter / timer #12 is connected to **COUT11**

2.9 Counters Architecture

There are four 8254 chips on PCI-8554 card. The counters on chip #1 ~ #4 are labeled from counter #1 to counter #12. Counters #11 and #12 are **cascaded counters**, and counter #1~ #10 can be programming to independent or cascaded counters. Table 2.2 illustrates the relationship between the reference number of chips and the counters number.

8254 Chip Number	Reference Number	Counter Number	Type of Counter
Chip #1	U3	Counter #1	Independent or Cascaded
		Counter #2	Independent or Cascaded
		Counter #3	Independent or Cascaded
Chip #2	U4	Counter #4	Independent or Cascaded
		Counter #5	Independent or Cascaded
		Counter #6	Independent or Cascaded
Chip #3	U5	Counter #7	Independent or Cascaded
		Counter #8	Independent or Cascaded
		Counter #9	Independent or Cascaded
Chip #4	U6	Counter #10	Independent or Cascaded
		Counter #11	Cascaded
		Counter #12	Cascaded

Table 2.2 Counters Architecture

There are three signals (2 input,1 output) for each counter, a clock input signal, a gate control signal, and an output signal. The Figure 2.3 illustrates the block diagram of 8254 counter. ***CLK1*** ~ ***CLK12*** are clock sources and ***GATE1*** ~ ***GATE12*** are gate control signals. The ***COUT1*** ~ ***COUT12*** are output of the counters. The Figure 2.4

shows all the labels and the inter-connection of the 8254 counters.

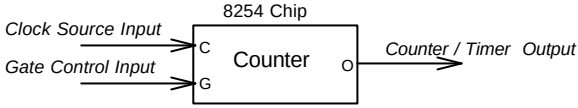


Figure 2.3 Block Diagram of 8254 Counter

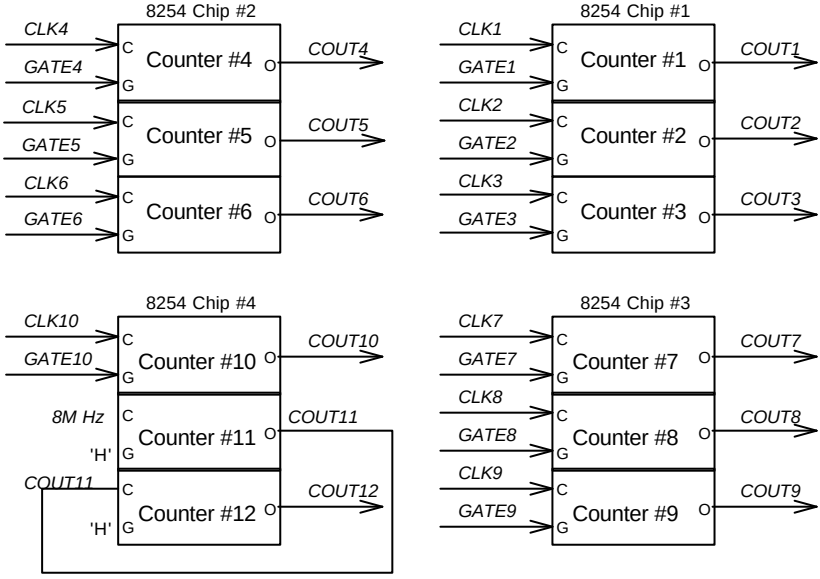


Figure 2.4 Counters Architectural

Independent Counters (Counter #1~#10)

The Counter #1 to Counter #10 are independent counters because the clock source and gate control of those counters can be set independently. These 10 counters are named as independent counter.

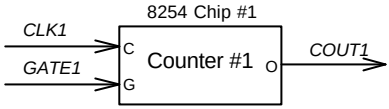


Figure 2.5 Example of ' independent counters'

Cascaded Counters

The connection of Counter #11 and #12 are different from other independent counters. These two counters are named as cascaded counters because the clock sources of counter #11 come from fixed 8 **MHz** and its output are cascaded to counter #12. In fact, counter #11and #12 are designed for frequency divider by using 8254's square wave generator mode. The gate of these counters keep at 'H' level for enabling counters all the time. The ***COUT12*** can precisely generate frequency upper to 2MHz and lower to 0.00186 Hz. Note that the signals ***COUT12*** can also be used as interrupt source. See ' Interrupt Sources' section for details. The following figure demonstrates cascaded counter - counter #11 and #12.

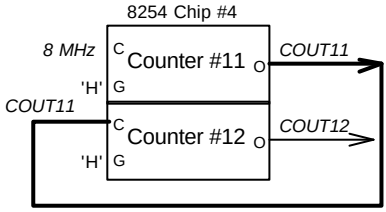


Figure 2.6 Example of ' cascaded counter'

User Configurable Cascaded Counters

Although there is one set cascaded counter on board, users may need more cascaded counters. User can set the clock source of every independent counters by program. Therefore, the independent counter output can be cascaded to the next counter's clock source to implement cascaded counter. Figure 2.7 demonstrate an example of the user programmable cascaded counter. Refer to next section for details of the clock source setting.

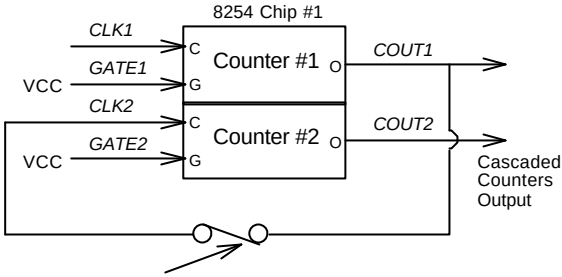


Figure 2.7 Example of ' user programmable cascaded counters'

2.10 Clock Source Configurations

For every independent counter, four signals can be chosen as clock source by software . The clock source of counter #n comes from either external clock source (***ECLK n***) or the cascaded counter output (**COUTn-1**) or **CK1** or **COUT10**. (Note: 1.The clock source of the cascaded counters #11 is fixed to **C8M** and counter #12 is fixed to **COUT11** 2. The external clock source named as ***ECK n*** comes from jumper **JP1** ~ **JP10**, please see section 2.12 for detail description.)

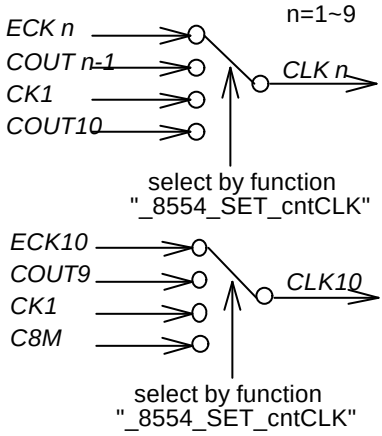


Figure 2.8 Clock Source of Counter #n

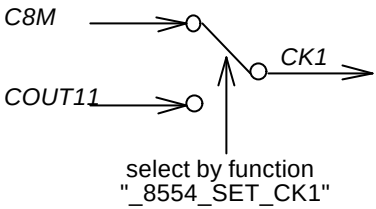


Figure 2.9 Clock Source of CK1

The internal clock sources ***CK1*** comes from the clock system **C8M** or **COUT11**, selected by function "**_8554_SET_CK1**", and counters can be set to cascaded mode, then clock source comes from the output of the counter with smaller channel number. For example, the ***COUT1*** is cascaded to ***CLK2***, the ***COUT3*** is cascaded to ***CLK4***. (Note: If counter #1 is set to cascaded mode, **CLK1** is connected to GND because **COUT0** doesn' t exist.)

2.11 Gate Control Configurations

The gate control signals of the independent counters are internally pulled high hence they are default enable if no external gate used. When the external gate signals are used, the counters can be used to measure pulse width. Therefore, the time interval of the counter gate can be precisely controlled and frequency measurement is possible. Figure 2.10 shows the jumper setting of gate control of counter #1~ #10. (Note: The gate control of counter #11 and #12 are always enable.)

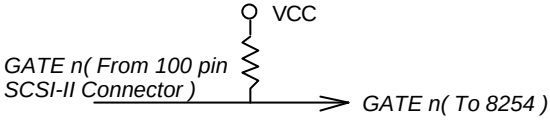


Figure 2.10 Gate source of counter #1 ~#10

2.12 Counter Outputs

The timer / counter output signals (***COUT n***) of 8254 are controlled by clock source, gate control and software programming. All the output of the 12 counters are sent to the 100 pins connector directly, please see 'Pin assignment' for corresponding signal pin number.

In addition, the output signal may be used as clock source for cascaded counters, see the above sections. It is possible to cascaded ten counters by software setting, see 2.8 for reference. The counters output ***COUT12*** is also used as internal interrupt source (refers to 'Interrupt System').

2.13 Debounce System

Debounce system is used to eliminate bounce phenomenon. If external clock is used, user can set jumper **JP1** ~ **JP11** to select if debounce system is used or not used. If debounce system is used, the debounce output signal will be the same state as the input only if the input signal keep the same state for four **DB_CLK**, otherwise the input signal will be treated as glitch and the debounce output signal will keep previous state, figure 2.11 show you the how to set these jumpers, figure 2.12 show you how to select **DB_CLK**, figure 2.13

show you the basic theorem of debounce system.(Note: **DB_CLK can' t be higher than 2MHz.**)

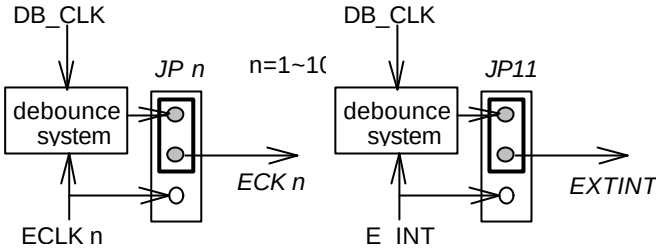


Figure 2.11 Structure of JP1 ~ JP11

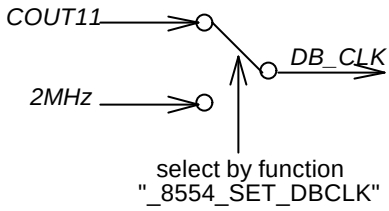


Figure 2.12 Clock Source of DB_CLK

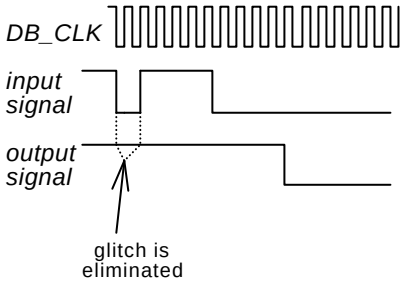


Figure 2.13 Basic theorem of debounce system

2.14 Interrupt System

The PCI-8554' s interrupt system is a powerful and flexible system, which is suitable for many applications. The system is a *Dual Interrupt System*. The dual interrupt means the hardware can

generate two interrupt request signals at the same time and the software can service these two request signals by ISR. Note that the dual interrupt do not mean the card occupy two IRQ levels. These two interrupt request signals (INT1 and INT2) comes from external interrupt signal **EXTINT** and the timer / counter #12 output. Fig 2.14 show you the structure of interrupt system.

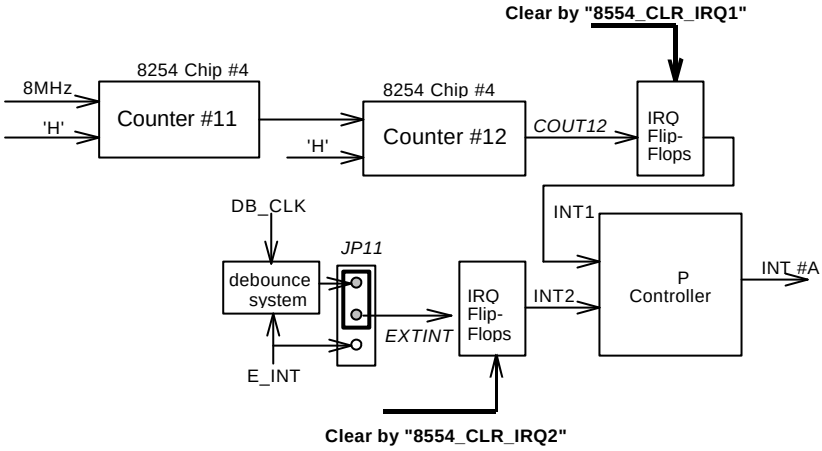


Fig 2.14 Dual Interrupt System of PCI-8554

There is only one IRQ level used by this card although it is a dual interrupt system. This card uses INT #A interrupt request signal to PCI bus. The mother board circuits will transfer INT #A to one of the AT bus IRQ levels. The IRQ level is set by the PCI plug and play BIOS and saved in the PCI controller. It is not necessary for users to set the IRQ level. Users can get the IRQ level setting by software library. Refer the section 5.4.

The PCI controller of PCI-8554 can receive two hardware IRQ sources. However, a PCI controller can generate only one IRQ to PCI bus, the two IRQ sources must be distinguished by ISR of the application software if the two IRQ are all used. The application software can use the "_8554_GET_IRQ_Status" function to distinguish which interrupt is inserted and servicing that IRQ then users must clear current IRQ to allow the next IRQ coming in.

If the application need only one IRQ, you can disable one of the IRQ

sources by software. If your application do not need any IRQ source, you can disable both the two interrupts . However, the PCI BIOS still assign a IRQ level to the PCI card and occupy the PC resource if you only disable the IRQ sources without change the initial condition of the PCI controller. It is not suggested to re-design the initial condition of the PCI card by users' own application software. If users want to disable the IRQ level, user can use the ADLINK's utility 'INIT8554.EXE' to change power on interrupt setting.

2.15 Digital Input and Output

To program digital I/O operation is fairly straight forward. The digital input operation is just to read data from the corresponding registers, and the digital output operation is to write data to the corresponding registers. The digital I/O registers' format are shown in section 3.4 and 3.5.

2.16 12V and 5V Power Supply

The 100 pin SCSI-II connector provides +12 volts and +5 volts power. To avoid short or overload of the power supply, the fuses are added on all the power supply signals. The maximum current for 5 volts on every fuse is 0.5 A. If the load current is larger than 0.5 A, the resistance of the fuse will increase because of the temperature rising. The rising resistance will cause the power supply drop and reduce current. If the overload or short condition is removed, the fuse will return to normal condition. It is no necessary to repair or re-install the fuse.

The maximum current of 12 volts for all the four connectors is also 0.5 A. The action of the fuse is the same as which used for +5V power. The limitation is more restrict than 5V power supply because the PCI bus can not provide large current.

3

Registers Format

The detailed descriptions of the registers format are specified in this chapter. This information is quite useful for the programmers who wish to handle the card by low-level programming. However, we suggest user have to understand more about the PCI interface then start any low-level programming. In addition, the contents of this chapter can help users understand how to use software driver to manipulate this card.

3.1 PCI PnP Registers

This PCI card functions as a 32-bit PCI target device to any master on the PCI bus. There are three types of registers: PCI Configuration Registers (PCR), Local Configuration Registers (LCR) and PCI-6308 registers.

The PCR, which is compliant to the PCI-bus specifications, is initialized and controlled by the plug & play (PnP) PCI BIOS. User' s can study the PCI BIOS specification to understand the operation of the PCR. Please contact with PCISIG to acquire the specifications of the PCI interface.

The PCI bus controller PCI-9050 is provided by PLX technology Inc. (www.plxtech.com). For more detailed information of LCR, please visit PLX technology' s web site to download relative information. It is not necessary for users to understand the details of the LCR if you use the software library. The PCI PnP BIOS assigns the base address of the LCR. The assigned address is located at offset 14h of PCR.

The PCI-6308 registers are shown in the next section. The base address, which is also assigned by the PCI PnP BIOS, is located at

offset 18h of PCR. Therefore, users can read the 18h of PCR to know the base address by using the BIOS function call.

Please do not try to modify the base address and interrupt which assigned by the PCI PnP BIOS, it may cause resource confliction in your system.

3.2 I/O Address Map

All the PCI-8554 registers are 8 bits. The users can access these registers by 8 bits I/O instructions. The following table shows the registers map, including descriptions and their offset addresses relative to the base address.

I/O Address	Write	Read
Base + 0x00 ~ 0x03	Counter 1 ~ 3 & Mode Control	Counter 1 ~ 3
Base + 0x04 ~ 0x07	Counter 4 ~ 6 & Mode Control	Counter 4 ~ 6
Base + 0x08 ~ 0x0B	Counter 7 ~ 9 & Mode Control	Counter 7 ~ 9
Base + 0x0C ~0x0F	Counter 10 ~ 12 & Mode Control	Counter 10 ~ 12
Base + 0x10 ~ 0x12	Clock Mode Control	- -
Base + 0x18	Digital Output	Digital Input

Table 3.1. I/O Address Map of PCI-8554

3.3 Timer/Counter Registers

The 8254 occupies 4 I/O address locations in PCI-8554 as shown blow. Users can refer to Tundra's or Intel's data sheet for a full description of the 8254 features, condensed information is specified in Appendix A.

Address : BASE + 0x00 ~ BASE + 0x0F
Attribute: read / write
Data Format:

Base + 0	Counter 1 Register (R/W)
Base + 1	Counter 2 Register (R/W)
Base + 2	Counter 3 Register (R/W)

Base + 3	8254 Mode Control Register (W) 8254 Read Back Register (R)
Base + 4	Counter 4 Register (R/W)
Base + 5	Counter 5 Register (R/W)
Base + 6	Counter 6 Register (R/W)
Base + 7	8254 Mode Control Register (W) 8254 Read Back Register (R)
Base + 8	Counter 7 Register (R/W)
Base + 9	Counter 8 Register (R/W)
Base + A	Counter 9 Register (R/W)
Base + B	8254 Mode Control Register (W) 8254 Read Back Register (R)
Base + C	Counter 10 Register (R/W)
Base + D	Counter 11 Register (R/W)
Base + E	Counter 12 Register (R/W)
Base + F	8254 Mode Control Register (W) 8254 Read Back Register (R)

3.4 Timer / Counter Clock Mode Control

There are total twenty two bits on PCI-8554 to select clock source of Timer / Counter #1 ~ #10 and **CK1** and debounce clock.
Address : BASE + 0x10 ~ 0x12
Attribute: write only
Data Format:

Bit	7	6	5	4	3	2	1	0
Base+0x10	*C4N2*	*C4N1*	*C3N2*	*C3N1*	*C2N2*	*C2N1*	*C1N2*	*C1N1*
Base+0x11	*C8N2*	*C8N1*	*C7N2*	*C7N1*	*C6N2*	*C6N1*	*C5N2*	*C5N1*
Base+0x12	-	*DBCSEL*	-	*CK1SEL*	*C10N2*	*C10N1*	*C9N2*	*C9N1*

C**n**N1 、 C**n**N2: these two bits are used to control clock source of Timer / Counter **n**, **n** = 1 ~ 10
CK1SEL: select source of **CK1**
DBCSEL: select debounce clock

3.5 Digital Input Registers

There are 8 digital input channels on the PCI-8554.
Address : BASE + 0x18
Attribute: read only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base+0x18	*DI7*	*DI6*	*DI5*	*DI4*	*DI3*	*DI2*	*DI1*	*DI0*

3.6 Digital Output Register

The register is a general purpose 8 bits digital output port. These signals can be used to control external devices.

Address : BASE + 0x18
Attribute: write only
Data Format:

Bit	7	6	5	4	3	2	1	0
Base+0x18	***DO7***	***DO6***	***DO5***	***DO4***	***DO3***	***DO2***	***DO1***	***DO0***

4

Signal Connections & Applications

This chapter describes the connectors and some application of the PCI-8554. including the signal connection between the PCI-8554 and external devices

4.1 Connectors Pin Assignment

The PCI-8554 comes equipped with a 100 pin SCSI-II female connector (CN1). The CN1 is located at the rear plate. The pin assignment of the connector is illustrated in the Figure 2.1. . Refer to section 2.1 for details of pin assignment.

4.2 Digital I/O Connection

The PCI-8554 provides 8 digital input and 8 digital output channels through the connector CN1. The digital I/O signals are fully TTL compatible.

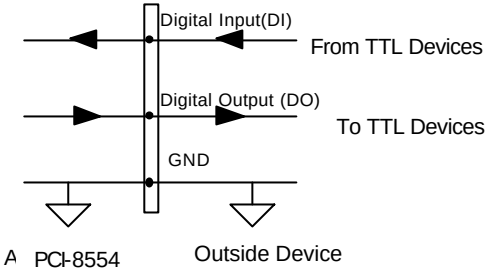


Figure 4.1 Digital I/O Connection

4.3 Timer / Counter Connection

The PCI-8554 has four 8254 chips on board. It can offer 10 independent 16-bit programmable down counters and cascaded counters. To implement your application, you can following the procedure to design your application and connect the signals.

1. Check if use a clock source with fixed frequency, if answer is ' No' , external clock source must be used and go to step 3.
2. **Calculate the frequency** of clock according to your application and **decide the clock source:** internal, external, or cascaded, then decide which counter is used.
3. If external clock source is used, generate a clock source outside the board and check the frequency. If external clock source is used, you have to decide whether debounce function is used or not, and then set the jumper JP1 ~ JP10.
4. **Decide the gate control source:** always enable or externally control, if gate control is necessary, connect the gate signal
5. Program the counter / timer using desired mode.

4.4 Frequency Generator

Example 1: To generate a 250 K Hz Square Wave.

step 1: To use **fixed** clock source because the output is a fixed frequency.

step 2: I**nternal 8M Hz** is suitable to generate 250K Hz frequency. Use Counter #1 for this application.

∵ 250 kHz = 8 M Hz / 32

step 3: Skip these steps.

step 4: The gate source is enable always, so let SCCI-II connector pin99 **GATE1** open

step 5: Connect the counter output to external device and write the control program. Please refer the ' DEMO1.C' source code.

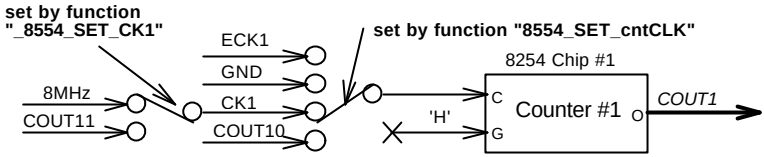


Figure 4.2 Example of frequency generator (1)

Example 2: To generate a very low frequency of 1 pulse / 1 hour

step 1: To use **fixed** clock source because the output is a fixed frequency.

step 2: Because the desired frequency (1/3600sec=0.000278Hz) is too slow to use one counter to generate, set the independent counter #1 & #2 & #3 to cascade mode: Clock source of counter #1 comes from **C8M**, clock source of counter #2 comes from **COUT1,** clock source of counter #3 comes from **COUT2.** Divider value of counter #1 was set to 4000, divider value of counter #2 was set to 2000, divider value of counter #3 was set to 3600, **8MHz**/4000/2000/3600 = 1/3600, so **COUT3** will generate a pulse every hour.

step 3: Skip these steps.

step 4: The gate source is enable always, so let **GATE1 GATE2 GATE3** open.

step 5: Write and verify the control program. Please refer the ' DEMO2.C' source code.

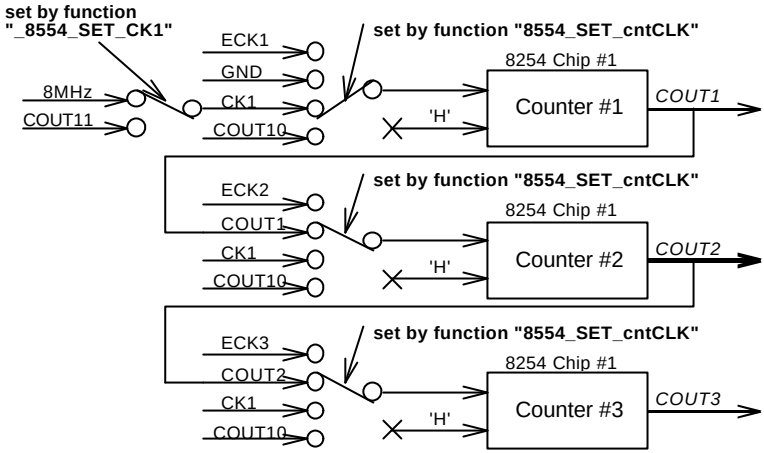


Figure 4.3 Example of frequency generator (2)

4.5 Pulse Width Measurement

Example : To measure pulse width

step 1: To use **fixed** clock source as base time interval (or base frequency).

step 2: Assume I**nternal 2M Hz** clock is used. The time base is

$$\Delta t = 1/2M = 5x10e\text{-}7 \text{ sec}$$

The count range for measuring pulse width is:

$$\Delta t < \text{pulse width} < \Delta t * 65535 \ (=32.768 \text{ msec})$$

If the specification of the pulse width to be measured is in the range, the 2M Hz can be used. Otherwise changing the base frequency of the counter, for example, you can set counter #2 to cascaded counter mode, and use counter #2 to measure pulse width, then the count range can increase but the resolution will decrease. Counter #1 #2 #3 are used in this example.

step 3: Skip these steps.

step 4: Connect **GATE1** to the signal to be measured.

step 5: Write and verify the control program. Please refer the ' DEMO3.C' source code. Note that if the pulse is shorter, the time resolution is worse. If the pulse is wider, the limitation of the maximum pulse width should be care.

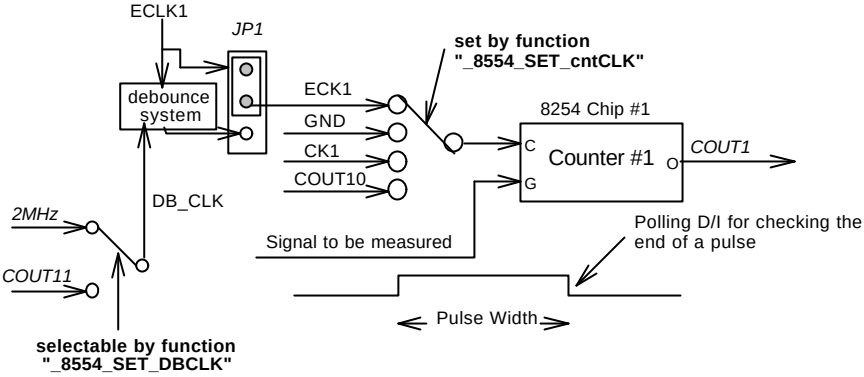


Figure 4.4 Example of pulse width measurement

4.6 Frequency Measurement

Example: To measure frequency around 1~100 K Hz

step 1: This application need two counters. One counter is used to generate a pulse whose time interval is very precise. The pulse is used to enable the other counter (counting counter) by gate control. The gate control is coming from **/COUT3**. In this example, cascaded counter is used, the pulse generator is counter #3 (clock is from **COUT2**) and the counter #1 is used to measure frequency.

step 2: The maximum value of counting counter is no more than 65535, For measuring 100 K Hz frequency, the time interval should be within 1/100 K Hz x 65535 = 0.655 sec. If the time interval is wider, then the measurement resolution is better, however, the counting value will be overflow if time interval is too long. That means the low pulse width of counter #3 output should shorter than 0.655 sec. User can try to generate the pulse by counter #3 by yourself.

step 3: Connect the signal to be measured to the **ECLK1** and adjust **JP1** to select debounce function.

step 4: Connect **GATE1** to **/COUT3**.

step 5: The following block diagram illustrates the application. Write and verify the control program. The frequency of the signal is:

frequency = counting value of counter #1 / precise time interval

Please refer the ‘ DEMO4.C’ source code.

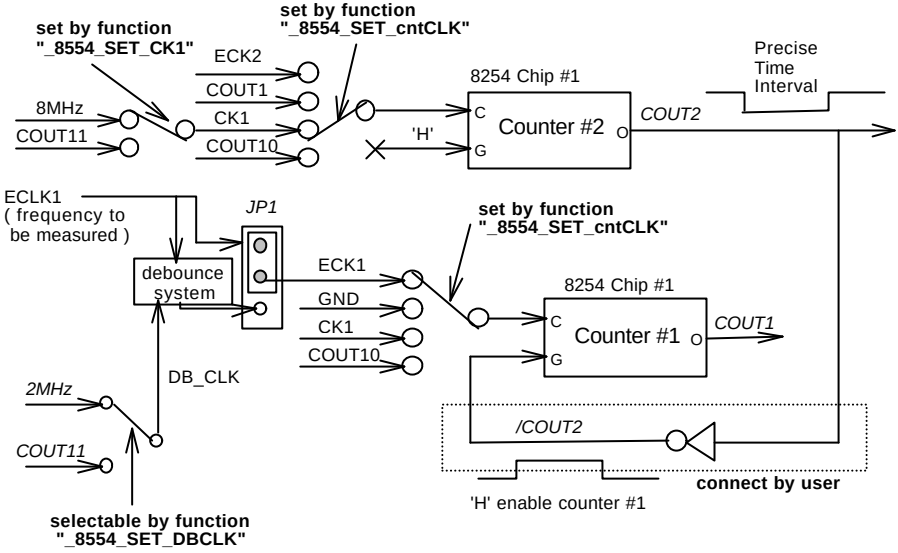


Figure 4.5 Example of frequency measurement (1)

4.7 Event Counter

Example: To count external event in 1 sec

step 1: This application needs one counter to generate a time base of 1 sec and the second counter to count the event. The cascaded counter #11, #12. can perform the watchdog timer. The another counter #1 is used as an example to count external event. The clock source of counter #1 is the event signal and the frequency is not fixed.

step 2: Skip this steps.

step 3: Connect **ECLK1** to the signal to be measured and adjust **JP1** to select debounce function.

Step 4: The gate source of counter 1 is always enable, so let the external gate open.

step 5: Write the control program. Please refer the ‘ DEMO5.C’ source code.

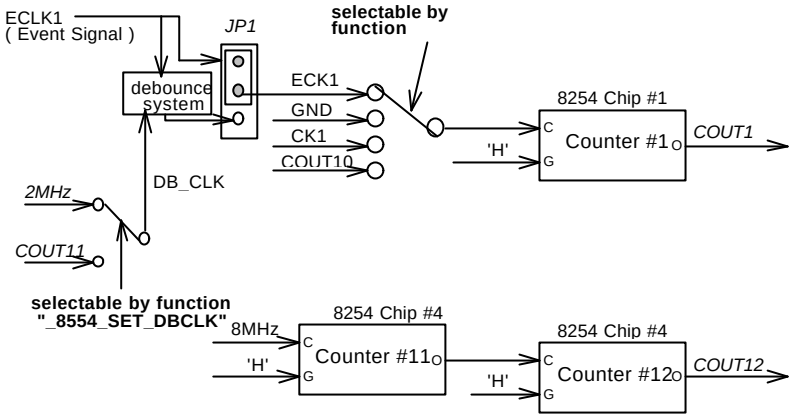


Figure 4.6 Example of event counter

4.8 Dual Interrupt System

One Internal plus one external interrupt sources

The PCI-8554 provides double interrupt sources which is very useful in some application. For example, most of the application needs a watchdog timer to monitor the system periodically, hence, an IRQ channel is used. In addition, the emergency control may be necessary, hence, an additional external IRQ channel is helpful to handle the situation. Therefore, dual interrupt level is necessary.

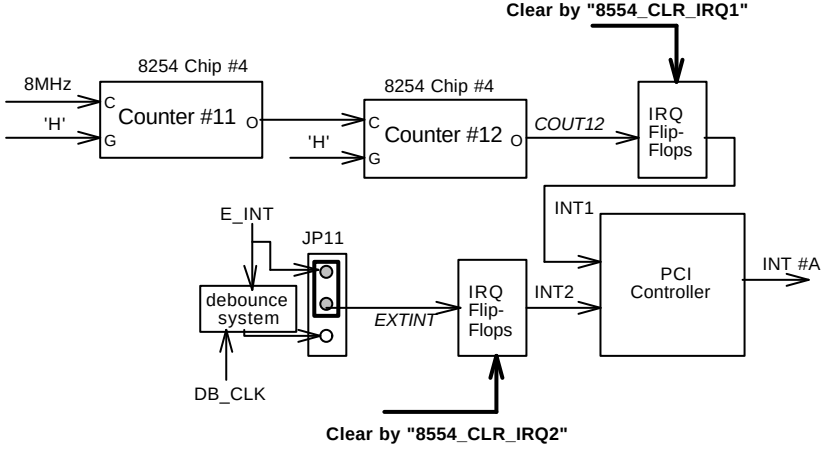


Figure 4.7 Example of dual interrupt system

5

C/C++ Library

This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 95 DLL are described. Please refer to the PCIS-DASK function reference manual, which included in ADLINK CD, for the descriptions of the Windows 98/NT/2000 DLL functions.

The function prototypes and some useful constants are defined in the header files LIB directory (DOS) and INCLUDE directory (Windows 95). For Windows 95 DLL, the developing environment can be Visual Basic 4.0 or above, Visual C/C++ 4.0 or above, Borland C++ 5.0 or above, Borland Delphi 2.x (32-bit) or above, or any Windows programming language that allows calls to a DLL. It provides the C/C++, VB, and Delphi include files.

5.1 Libraries Installation

Please refer to the "***Software Installation Guide***" for the detail information about how to install the software libraries for DOS, or Windows 95 DLL, or PCIS-DASK for Windows 98/NT/2000.

The device drivers and DLL functions of Windows 98/NT/2000 are included in the PCIS-DASK. Please refer the PCIS-DASK user's guide and function reference, which included in the ADLINK CD, for detailed programming information.

5.2.1 Naming Convention

The functions of the NuDAQ PCI cards or NuIPC CompactPCI cards' software driver are using full-names to represent the functions' real meaning. The naming convention rules are:

In DOS Environment:

_{hardware_model}_{action_name}. **e.g. _8554_Initial().**

In order to recognize the difference between DOS library and Windows 95 library, a capital "**W**" is put on the head of each function name of the Windows 95 DLL driver. e.g. **W_8554_Initial().**

5.2.2 Data Types

We defined some data type in Pci_8554.h (DOS) and Acl_pci.h (Windows 95). These data types are used by NuDAQ Cards' library. We suggest you to use these data types in your application programs. The following table shows the data type names and their range.

Type Name	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed integer	-2147483648 to 2147483647
U32	32-bit single-precision floating-point	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

5.3 _8554_Initial

@ Description

The PCI-8554 cards are initialized by this function. The software library could be used to control multiple PCI-8554 cards. Because PCI-8554 is in PCI bus architecture and meets the plug and play specifications, the ***IRQ*** and ***I/O address*** are assigned by system BIOS directly.

@ Syntax

C/C++ (DOS)

U16 _8554_Initial (U16 *existCards, PCI_INFO *pciinfo)

C/C++ (Windows 95)

U16 W_8554_Initial (U16 *existCards, PCI_INFO *pciinfo)

Visual Basic (Windows 95)

W_8554_Initial (existCards As Integer, pciInfo As PCI_INFO) As Integer

@ Arguments

existCards: The numbers of installed PCI-8554 cards. The returned value shows how many PCI-8554 cards are installed in your system.

pciinfo: It is a structure to memorize the PCI bus plug and play initialization information which is decided by PnP BIOS. The PCI_INFO structure is defined in PCI_8554.H. The base I/O address and the interrupt channel number is stored in pciinfo which is for reference.

@ Return Value

ERR_NoError, ERR_PCIBiosNotExist

5.4 _8554_Write_Counter

@ Description

User can directly write command to counter #1 ~ #12 by this function. Using this function, user can assign the counter number 1~12 directly without care about the chips number and other details.

@ Syntax

C/C++ (DOS)

U16 _8554_Write_Counter (U16 cardNo, U16 cntNo, U16 mode, U16 cntrVal)

C/C++ (Windows 95)

U16 W_8554_Write_Counter (U16 cardNo, U16 cntNo, U16 mode, U16 cntrVal)

Visual Basic (Windows 95)

W_8554_Write_Counter (ByVal cardNo As Integer, ByVal cntNo As Integer, ByVal mode As Integer, ByVal cntrVal As Integer) As Integer

@ Arguments

cardNo:	card number to select board
cntNo:	Counter / Timer number. This value must between 1 and 12.
mode:	Counter operation mode. This value must between 0 and 5.
cntrVal:	The counter value to be written to the counter.

@ Return Value

ERR_NoError
ERR_BoardNoInit
ERR_InvalidCounterNo: **cntNo** is out of range.
ERR_TimerMode: **mode** is out of range

5.5 _8554_Read_Counter

@ Description

User can directly read counter information by this function.

@ Syntax

C/C++ (DOS)

U16 _8554_Read_Counter (U16 cardNo,U16 cntNo,U16 *mode,U16 *cntrVal)

C/C++ (Windows 95)

U16 W_8554_Read_Counter (U16 cardNo,U16 cntNo,U16 *mode,U16 *cntrVal)

Visual Basic (Windows 95)

W_8554_Read_Counter (ByVal cardNo As Integer, ByVal cntNo As Integer, mode As Integer, cntrVal As Integer) As Integer

@ Arguments

cardNo:	card number to select board
cntNo:	Counter / Timer number. This value must between

1 and 12.

mode: Counter operation mode.

cntrVal: Counter value read back from counter

@ Return Value

ERR_NoError , ERR_BoardNoInit

ERR_InvalidCounterNo: **cntNo** is out of range.

5.6 _8554_Stop_Counter

@ Description

User can directly stop counter by this function. This function will stop counter by setting counter to mode 5.

@ Syntax

C/C++ (DOS)

U16 _8554_Stop_Counter (U16 cardNo,U16 cntNo,U16 *cntrVal)

C/C++ (Windows 95)

U16 W_8554_Stop_Counter (U16 cardNo,U16 cntNo,U16 *cntrVal)

Visual Basic (Windows 95)

W_8554_Stop_Counter (ByVal cardNo As Integer, ByVal cntNo As Integer, cntrVal As Integer) As Integer

@ Arguments

cardNo: card number to select board

cntNo: Counter / Timer number. This value must between 1 and 12.

cntrVal: Counter value read back from counter

@ Return Value

ERR_NoError , ERR_BoardNoInit

ERR_InvalidCounterNo: if **cntNo** is not in the range of 1 ~ 12

5.7 _8554_Read_Status

@ Description

User can directly read current counter status by this function.

@ Syntax

C/C++ (DOS)

U16 _8554_Read_Status (U16 cardNo, U16 cntNo, U16 * cntrVal, U16 *status)

C/C++ (Windows 95)

U16 W_8554_ Read_Status (U16 cardNo, U16 cntNo, U16 * cntrVal, U16 *status)

Visual Basic (Windows 95)

W_8554_Read_Status (ByVal cardNo As Integer, ByVal cntNo As Integer, cntrVal As Integer, status As Integer) As Integer

*@ **Arguments***

cardNo:	card number to select board
cntNo:	Counter / Timer number. This value must be between 1 and 12.
cntrVal:	Counter value read back from counter
status:	current status read back from counter, please refer to 8254's datasheet for detail information

*@ **Return Value***

ERR_NoError , ERR_BoardNoInit

ERR_InvalidCounterNo: if **cntNo** is not in the range of 1 ~ 12

5.8 _8554_DO

*@ **Description***

To write a 8 bits data to the digital output port.

*@ **Syntax***

C/C++ (DOS)

U16 _8554_DO (U16 cardNo, U16 doData)

C/C++ (Windows 95)

U16 W_8554_DO (U16 cardNo, U16 doData)

Visual Basic (Windows 95)

W_8554_DO (ByVal cardNo As Integer, ByVal doData As Integer) As Integer

*@ **Arguments***

cardNo:	card number to select board
doData:	the value to write to digital output port

*@ **Return Value***

ERR_NoError

ERR_PCIBiosNotExist

5.9 _8554_DI

*@ **Description***

To read 8 bits data from digital input port.

*@ **Syntax***

C/C++ (DOS)

U16 _8554_DI (U16 cardNo,U16 *diData)

C/C++ (Windows 95)

U16 W_8554_DI (U16 cardNo,U16 *diData)

Visual Basic (Windows 95)

W_8554_DI (ByVal cardNo As Integer, diData As Integer) As Integer

@ Arguments

cardNo: card number to select board

doData: the value read from digital input port

@ Return Value

ERR_NoError , ERR_BoardNoInit

5.10 _8554_SET_cntCLK

@ Description

To select 8254 counter #1 ~ #10 clock source.(Clock source of counter #11 is 8MHz and clock source of counter #12 is from **COUT11,** both clock source are fixed.)

@ Syntax

C/C++ (DOS)

U16 _8554_DI (U16 cardNo, U16 cntNo, U16 clkMODE)

C/C++ (Windows 95)

U16 W_8554_DI (U16 cardNo, U16 cntNo, U16 clkMODE)

Visual Basic (Windows 95)

W_8554_SET_cntCLK (ByVal cardNo As Integer, ByVal cntNo As Integer, ByVal clkMODE As Integer) As Integer

@ Arguments

cardNo: card number to select board

cntNo: Counter / Timer number. This value must be between 1 and 10.

clkMODE: Select clock source. **"0"** select **ECLKn**, **"1"** select **COUTn-1**, **"2"** select **CK1**, **"3"** select **COUT10**.

@ Return Value

ERR_NoError

ERR_BoardNoInit

ERR_InvalidCounterNo: **cntNo** is not in the range of 1 ~ 12

ERR_InvalidMode: **clkMODE** is not in the range of 1 ~ 3

5.11 _8554_SET_CK1

*@ **Description***

To select source of **CK1**.

*@ **Syntax***

C/C++ (DOS)

U16 _8554_SET_CK1 (U16 cardNo, U16 selCK1)

C/C++ (Windows 95)

U16 W_8554_SET_CK1 (U16 cardNo, U16 selCK1)

Visual Basic (Windows 95)

W_8554_SET_CK1 (ByVal cardNo As Integer, ByVal selCK1 As Integer) As Integer

*@ **Arguments***

cardNo:	card number to select board
selCK1:	if set selCK1 " 0" then **CK1** is **C8M**, if set selck1 " 1" then **CK1** is **COUT11**

*@ **Return Value***

ERR_NoError

ERR_BoardNoInit

ERR_InvalidMode: **selCK1** is out of range

5.12 _8554_SET_DBCLK

*@ **Description***

To select debounce clock.

*@ **Syntax***

C/C++ (DOS)

U16 _8554_SET_DBCLK (U16 cardNo,U16 DBCLK)

C/C++ (Windows 95)

U16 W_8554_SET_DBCLK (U16 cardNo,U16 DBCLK)

Visual Basic (Windows 95)

W_8554_SET_DBCLK (ByVal cardNo As Integer, ByVal DBCLK As Integer) As Integer

*@ **Arguments***

cardNo:	card number to select board
DBCLK:	if set dbclk " 0" then **DB_CLK** is **COUT11**, if set dbclk " 1" then **DB_CLK** is **2MHz**

*@ **Return Value***

ERR_NoErro, ERR_BoardNoInit

ERR_InvalidMode: **DBCLK** is out of range

5.13 _8554_Set_INT_Control

*@ **Description***

The PCI-8554 has dual interrupts system. Two interrupt sources can be generated and be checked by the software. This function is used to select and control PCI-8554 interrupt sources. The interrupt source can be set as from counter #12 output COUT12 (INT1) or external interrupt signal EXTINT (INT2).

*@ **Syntax***

C/C++ (DOS)

U16 _8554_Set_INT_Control (U16 cardNo, U16 int1Flag, U16 int2Flag)

C/C++ (Windows 95)

U16 W_8554_Set_INT_Control (U16 cardNo, U16 int1Flag, U16 int2Flag)

Visual Basic (Windows 95)

W_8554_Set_INT_Control (ByVal cardNo As Integer, ByVal int1Falg As Integer, ByVal int2Falg As Integer)

*@ **Arguments***

cardNo:	card number to select board
int1Flag:	INT1 setting; 0: disable, 1: enable
int2Flag:	INT2 setting; 0: disable, 1: enable

5.14 _8554_Get_IRQ_Status

*@ **Description***

The PCI-8554 has dual interrupts system. Two interrupt sources can be generated and be checked by the software. This function is used to distinguish which interrupt is inserted if both INT1 and INT2 interrupts are used.

*@ **Syntax***

C/C++ (DOS)

U16 _8554_Get_IRQ_Status (U16 cardNo, U16 *ch1, U16 *ch2)

C/C++ (Windows 95)

U16 W_8554_Get_IRQ_Status (U16 cardNo, U16 *ch1, U16 *ch2)

Visual Basic (Windows 95)

W_8554_Get_IRQ_Status (ByVal cardNo As Integer, ch1 As Integer, ch2 As Integer)

*@ **Arguments***

cardNo:	card number to select board
ch1:	INT1 status; 0: interrupt is not from INT1, 1: interrupt is from INT1
ch2:	INT2 status; 0: interrupt is not from INT2, 1:

interrupt is from INT2

5.15 _8554_INT_Enable

@ Description

This function is only available in Windows 95 driver. This function is used to start up the interrupt control. After calling this function, every time an interrupt request signal generated, a software event is signaled. So that in your program, you can use wait operation to wait for the event. When the event is signaled, it means an interrupt is generated.

@ Syntax

C/C++ (Windows 95)

U16 W_8554_INT_Enable (U16 cardNo, HANDLE *hEvent)

Visual Basic (Windows 95)

W_8554_INT_Enable (ByVal cardNo As Integer, hEvent As Long) As Integer

@ Arguments

cardNo: card number to select board

hEvent: the address of an array of two handles. HEvent[0] and hEvent[1] are the events for interrupt signals INT1 and INT2 respectively.

@ Return Value

ERR_NoError

ERR_BoardNoInit

5.16 _8554_INT_Disable

@ Description

This function is only available in Windows 95 driver. This function is used to disable the interrupt signal generation.

@ Syntax

C/C++ (Windows 95)

U16 W_8554_INT_Disable (U16 cardNo)

Visual Basic (Windows 95)

W_8554_INT_Disable (ByVal cardNo As Integer) As Integer

@ Arguments

cardNo: card number to select board

@ Return Value

ERR_NoError , ERR_BoardNoInit

5.17 _8554_CLR_IRQ1

@ Description

This function is only needed in DOS driver. It is used to clear interrupt request which is requested by INT1. You should use this function to clear interrupt request status, otherwise the new coming interrupt will not be generated.

@ Syntax

C/C++ (Windows 95)

U16 _8554_CLR_IRQ1 (U16 cardNo)

@ Arguments

cardNo: card number to select board

5.18 _8554_CLR_IRQ2

@ Description

This function is only needed in DOS driver. It is used to clear interrupt request which is requested by INT2. You should use this function to clear interrupt request status, otherwise the new coming interrupt will not be generated.

@ Syntax

C/C++ (Windows 95)

U16 _8554_CLR_IRQ2 (U16 cardNo)

@ Arguments

cardNo: card number to select board

Appendix A. Timer/Counter Operation

The PCI-8554 has at most four interval 8254 chips on board. Refer to chapter 2 and 4 for the signal connection and the configuration of the counters. The following sections describe the details of the 8254 chip.

The 8254 Timer / Counter Chip

The Intel (Tundra) 8254 contains three independent, programmable, multi-mode 16 bit counter/timers. The three independent 16 bit counters can be clocked at rates from DC to 8MHz MHz. Each counter can be individually programmed with 6 different operating modes by appropriately formatted control words. The most commonly uses for the 8254 in microprocessor based system are:

- programmable baud rate generator
- event counter
- binary rate multiplier
- real-time clock
- digital one-shot
- motor control

For more information about the 8254 , please refer to the Tundra Microprocessors and peripherals or Intel Microsystems Components Handbook.

I/O Address

The 8254 in the PCI-8554 occupies 4 I/O address as shown below. Although there are four 8254 chips on board, however, only one chip is selected in one moment. The programming of 8254 is control by the registers BASE+0 to BASE+3. The functionality of each register is specified in the following sections. For more detailed information, please refer handbook of 8254 chip.

BASE + 0	LSB OR MSB OF COUNTER 0
BASE + 1	LSB OR MSB OF COUNTER 1
BASE + 2	LSB OR MSB OF COUNTER 2
BASE + 3	CONTROL BYTE

Control Byte

Before loading or reading any of these individual counters, the control byte (BASE+3) must be loaded first. The format of the control byte is:

Bit	7	6	5	4	3	2	1	0
	SC1	SC0	RL1	RL0	M2	M1	M0	BCD

- SC1 & SC0 - Select Counter (Bit 7 & Bit 6)

SC1	**SC0**	**COUNTER**
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	ILLEGAL

- RL1 & RL0 - Select Read/Load operation (Bit 5 & Bit 4)

RL1	**RL0**	**OPERATION**
0	0	COUNTER LATCH FOR STABLE READ
0	1	READ/LOAD LSB ONLY
1	0	READ/LOAD MSB ONLY
1	1	READ/LOAD LSB FIRST, THEN MSB

- M2, M1 & M0 - Select Operating Mode (Bit 3, Bit 2, & Bit 1)

M2	**M1**	**M0**	**MODE**
0	0	0	0
0	0	1	1
x	1	0	2
x	1	1	3
1	0	0	4
1	0	1	5

- BCD - Select Binary/BCD Counting (Bit 0)

0	16-BITS BINARY COUNTER
1	BINARY CODED DECIMAL (BCD) COUNTER (4 DIGITAL)
Note	The count of the binary counter is from 0 up to 65,535 and the count of the BCD counter is from 0 up to 9,999

Mode Definitions

In 8254, six operating modes can be selected. they are:

- **Mode 0**: Interrupt on Terminal Count
- **Mode 1**: Programmable One-Shot.
- **Mode 2**: Rate Generator.
- **Mode 3**: Square Wave Rate Generator.
- **Mode 4**: Software Triggered Strobe.
- **Mode 5**: Hardware Triggered Strobe.

All detailed description of these six modes are written in Intel Microsystem Components Handbook Volume II Peripherals.

Timer / Counter Applications

Please refer to Chapter 4.

Product Warranty / Service

Seller warrants that equipment furnished will be free form defects in material and workmanship for a period of one year from the confirmed date of purchase of the original buyer and that upon written notice of any such defect, Seller will, at its option, repair or replace the defective item under the terms of this warranty, subject to the provisions and specific exclusions listed herein.

This warranty shall not apply to equipment that has been previously repaired or altered outside our plant in any way as to, in the judgment of the manufacturer, affect its reliability. Nor will it apply if the equipment has been used in a manner exceeding its specifications or if the serial number has been removed.

Seller does not assume any liability for consequential damages as a result from our products uses, and in any event our liability shall not exceed the original selling price of the equipment.

The equipment warranty shall constitute the sole and exclusive remedy of any Buyer of Seller equipment and the sole and exclusive liability of the Seller, its successors or assigns, in connection with equipment purchased and in lieu of all other warranties expressed implied or statutory, including, but not limited to, any implied warranty of merchant ability or fitness and all other obligations or liabilities of seller, its successors or assigns.

The equipment must be returned postage-prepaid. Package it securely and insure it. You will be charged for parts and labor if you lack proof of date of purchase, or if the warranty period is expired.