Acknowledgments

PC-LabCard is a trademark of Advantech Co., Ltd. IBM and PC are trademarks of International Business Machines Corporation. MS-DOS, Windows, Microsoft Visual C++ and Visual BASIC are trademarks of Microsoft Corporation. Intel and Pentium are trademarks of Intel Corporation. Delphi and C++ Builder are trademarks of Borland Corporation.

CE notification

The PCI-1755, developed by ADVANTECH CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

On-line Technical Support

For technical support and service, please visit our support website at:
http://www.advantech.com/support

Part No. 2003175500 1st Edition
Printed in Taiwan Jul 2003

(This page is left blank for hard printing.)

Contents

(This page is left blank for hard printing.)

Chapter 1

Introduction

(This page is left blank for hard printing.)

1. Introduction

Thank you for buying the Advantech PCI-1755. The PCI-1755 is a Ultra-Speed 32-ch Digital I/O card for the PCI bus. Its digital I/O channels are TTL-compatible and use 74LS244 driver/buffer circuits to provide high output driving capacity. These buffered circuits also require lower input loading current than regular TTL circuits. The ultra speed data transfer functions fulfill your industrial or laboratory application needs.

The following sections of this chapter will provide further information about features of the multifunction cards, a Quick Start for installation, together with some brief information on software and accessories for the PCI-1755 card.

1.1 Features

- Bus-mastering DMA data transfer with Scatter-Gather™ technology
- 32/16/8-bit Pattern I/O with start and stop trigger function, 2-mode Handshaking I/O
- Interrupt handling capability
- On-board active terminators for high speed and long distance transfers
- Pattern match and Change state detection interrupt function
- General-purposed 8-ch DI/O

The Advantech PCI-1755 offers the following main features:

PCI-Bus Mastering Data Transfer

The PCI-1755 supports PCI-Bus mastering DMA for high-speed data transfer. By setting aside a block of memory in the PC, the PCI-1755 performs bus-mastering data transfers without CPU intervention, freeing the CPU to perform other more urgent tasks such as data analysis and graphic manipulation. The function allows users to run all I/O functions simultaneously at full speed without losing data.

Special Shielded Cable for Noise Reduction

The PCL-101100 shielded cable is specially designed for the PCI-1755 for reducing noise. Its wires are all twisted pairs, with input signals and output signals separately shielded, providing minimal cross talk between signals and offering the best protection against EMI/EMC problems.

Keeping the Output Values after System Reset

When the system is hot reset (power is not shut off), the PCI-1755 can either retain the last digital output values, or return to its default configuration, depending on the jumper setting. This practical function eliminates dangers and problems caused by an unexpected system reset.

On-board FIFO Memory

The PCI-1755 provides an on-board FIFO (Fist In First Out) memory buffer, storing up to 16K samples for digital input and 16K for digital output conversion.

Pattern Match Function

The PCI-1755 provides "Pattern Match" interrupt function for digital input channels. The card monitors the state of digital inputs and compares them with a pre-set pattern. When the received state matches the pre-set pattern, the PCI-1755 generates an interrupt signal to system.

Change of State Function

"Change of State" interrupt function is provided at-- digital input channels. When any signal line changes its state, the card generates an interrupt to the system to handle this event.

Note:

☞ For detailed specifications of the PCI-1755, please refer to *Appendix A, Specifications.*

1.2 Applications

- High speed IC function test
- Parallel data transfer
- TTL, DTL and CMOS logic signal sensing
- Relay and switch monitoring and controlling
- Indicator LED driving

1.3 Installation Guide

Before you install your PCI-1755 card, please make sure you have the following necessary components:

PCI-1755 DA&C card	
PCI-1755 User's Manual	
Driver software	Advantech DLL drivers (included in the companion CD-ROM)
Wiring cable	PCL-101100 (option)
Wiring board	ADAM-39100 (option)
Computer	Personal computer or workstation with a PCI-bus slot (running Windows 2000/95/98/ ME/NT/XP)

Some other optional components are also available for enhanced operation:

Application software	ActiveDAQ or other third-party software packages

After you get the necessary components and maybe some of the accessories for enhanced operation of your Multifunction card, you can then begin the Installation procedures. ***Fig. 1-1*** on the next page provides a concise flow chart to give users a broad picture of the software and hardware installation procedures:

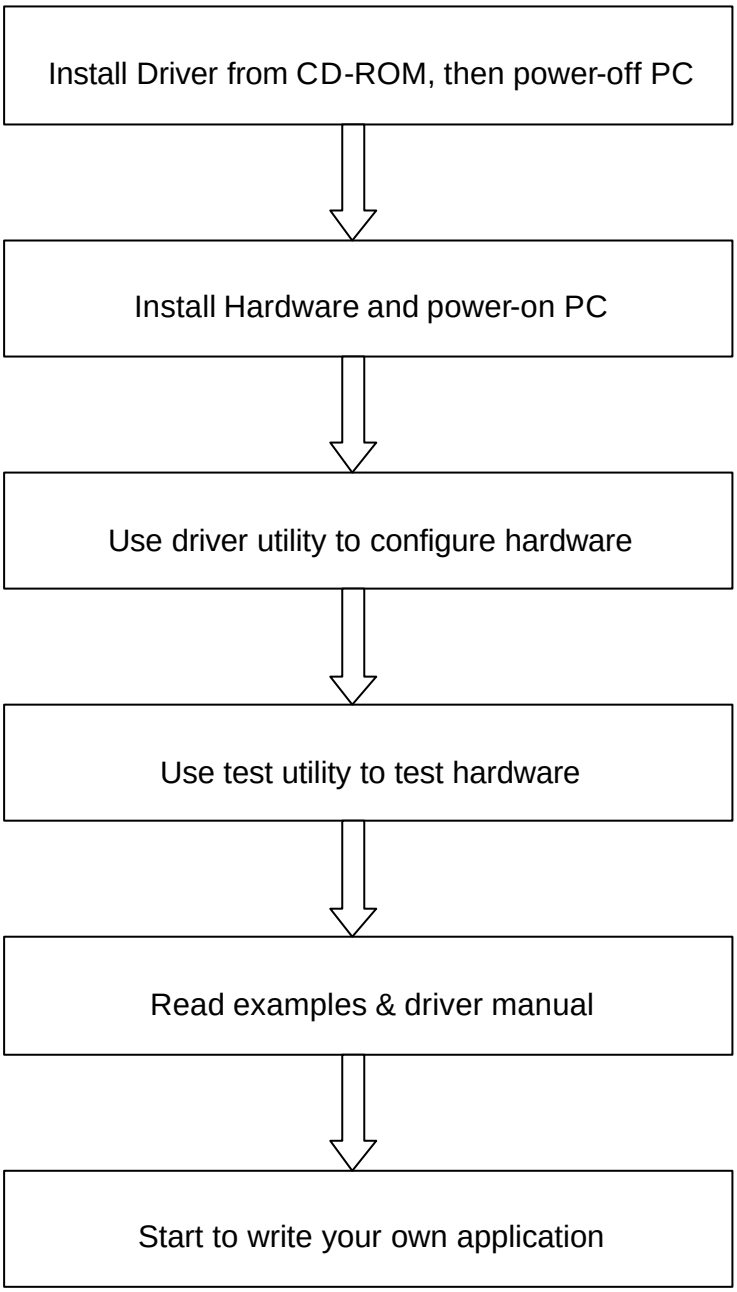


Fig. 1-1 Installation Flow Chart

1.4 Software Overview

Advantech offers a rich set of DLL drivers, third-party driver support and application software to help fully utilize the functions of your PCI-1755 card:

Device Drivers (on the companion CD-ROM)
Advantech ActiveDAQ
Advantech GeniDAQ

Programming choices for DA&C cards: You may use Advantech application software such as Advantech Device Drivers. On the other hand, advanced users can use another option for register-level programming, although it is not recommended due to its laborious and time-consuming nature.

Device Drivers

The Advantech Device Drivers software is included on the companion CD-ROM at no extra charge. It also comes with all Advantech DA&C cards. Advantech's device drivers feature a complete I/O function library to help boost your application performance. The Advantech Device Drivers for Windows 2000/95/98/ME/NT/XP works seamlessly with development tools such as Visual C++, Visual Basic, Borland C++ Builder and Borland Delphi.

Register-level Programming

Register-level programming is reserved for experienced programmers who find it necessary to write code directly at the level of device registers. Since register-level programming requires much effort and time, we recommend that you use the Advantech Device Drivers instead. However, if register-level programming is necessary, you should refer to the relevant information in *Appendix C, Register Structure and Format*, or to the example codes included on the companion CD-ROM.

1.5 Device Drivers Programming Roadmap

This section will provide you a roadmap to demonstrate how to build an application from scratch using Advantech Device Drivers with your favorite development tools such as Visual C++, Visual Basic, Delphi and C++ Builder. The step-by-step instructions on how to build your own applications using each development tool will be given in the *Device Drivers Manual*. Moreover, a rich set of example source code is also given for your reference.

Programming Tools

Programmers can develop application programs with their favorite development tools:

Visual C++
Visual Basic
Delphi
C++ Builder

For instructions on how to begin programming works in each development tool, Advantech offers a *Tutorial* Chapter in the *Device Drivers Manual* for your reference. Please refer to the corresponding sections in this chapter of the *Device Drivers Manual* to begin your programming efforts. You can also look at the example source code provided for each programming tool, since they can get you very well oriented.

The *Device Drivers Manual* can be found on the companion CD-ROM. Or if you have already installed the *Device Drivers* on your system, the *Device Drivers Manual* can be readily accessed through the ***Start*** button:

Start/Programs/Advantech Automation/Device Manager/Device Driver's Manual

The example source codes could be found under the corresponding installation folder such as the default installation path:
\Program Files\Advantech\ADSAPI\Examples

For information about using other function groups or other development tools, please refer to the *Creating Windows 95/NT/2000 Application with Device Drivers*

chapter and the *Function Overview* chapter on the *Device Drivers Manual.*

Programming with Device Drivers Function Library

Advantech Device Drivers offers a rich function library to be utilized in various application programs. This function library consists of numerous APIs that support many development tools, such as Visual C++, Visual Basic, Delphi and C++ Builder.

According to their specific functions or services, those APIs can be categorized into several function groups:

Digital Input/Output Function Group

Port Function Group (direct I/O)

Event Function Group

For the usage and parameters of each function, please refer to the *Function Overview* chapter in the *Device Drivers Manual.*

Troubleshooting Device Drivers Error

Driver functions will return a status code when they are called to perform a certain task for the application. When a function returns a code that is not zero, it means the function has failed to perform its designated function. To troubleshoot the *Device Drivers* error, you can pass the error code to **DRV_GetErrorMessage** function and it will return the error message. You can refer to the *Device Drivers Error Codes Appendix* in the *Device Drivers Manual* for a detailed listing of the **Error_Code**, **Error_ID** and the **Error_Message**.

1.6 Accessories

Advantech offers a complete set of accessory products to support the PCI-1755 card. These accessories include:

Wiring Cable

PCL-101100 The PCL-101100 shielded cable is specially designed for PCI-1755 cards to provide high resistance to noise. To achieve better signal quality, the signal wires are twisted in such a way as to form a "twisted-pair cable," reducing cross-talk and noise from other signal sources. Furthermore, its analog and digital lines are separately sheathed and shielded to neutralize EMI/EMC problems.

Wiring Board

ADAM -39100 The ADAM-39100 is a 100-pin SCSI-II wiring terminal module for DIN-rail mounting. This terminal module can be readily connected to the Advantech PC-Lab cards and allow easy yet reliable access to individual pin connections for the PCI-1755 card.

Chapter 2

Installation

(This page is left blank for hard printing.)

2. Installation

This chapter gives users a package item checklist, proper instructions about unpacking and step-by-step procedures for both driver and card installation.

2.1 Unpacking

After receiving your PCI-1755 package, please inspect its contents first. The package should contain the following items:

☑ PCI-1755 card
☑ Companion CD-ROM (DLL driver included)
☑ User's Manual

The PCI-1755 card harbors certain electronic components vulnerable to *electrostatic discharge* (ESD). ESD could easily damage the integrated circuits and certain components if preventive measures are not carefully paid attention to.

Before removing the card from the antistatic plastic bag, you should take following precautions to ward off possible ESD damage:

- Touch the metal part of your computer chassis with your hand to discharge static electricity accumulated on your body. Or use a grounding strap.
- Touch the anti-static bag to a metal part of your computer chassis before opening the bag.
- Take hold of the card only by the metal bracket when removing it from the bag.

After taking out the card, first you should:

- Inspect the card for any possible signs of external damage (loose or damaged components, etc.). If the card is visibly damaged, please notify our service department or the local sales representative immediately. Avoid installing a damaged card into your system.

Also, pay extra caution to the following aspects to ensure proper installation:

- Avoid physical contact with materials that could hold static electricity such as plastic, vinyl and Styrofoam.
- Whenever you handle the card, grasp it only by its edges. DO NOT TOUCH

the exposed metal pins of the connector or the electronic components.

Note:

☞ Keep the anti-static bag for future use. You might need the original bag to store the card if you have to remove the card from the PC or transport it elsewhere.

2.2 Driver Installation

We recommend you to install the driver before you install the PCI-1755 card into your system, since this will guarantee a smooth installation process.

The *Advantech Device Drivers* setup program for the PCI-1755 card is included on the companion CD-ROM that is shipped with your DA&C card package. Please follow the steps below to install the driver software:

Step 1: Insert the companion CD-ROM into your CD-ROM drive.

Step 2: The *Setup* program will be launched automatically if you have the AUTORUN function enabled on your system. When the Setup program is launched, you' ll see the following Setup Screen.

Note:

If the AUTORUN function is not enable on your computer, use Windows Explorer or the Windows ***Run*** command to execute SETUP.EXE on the companion CD-ROM.

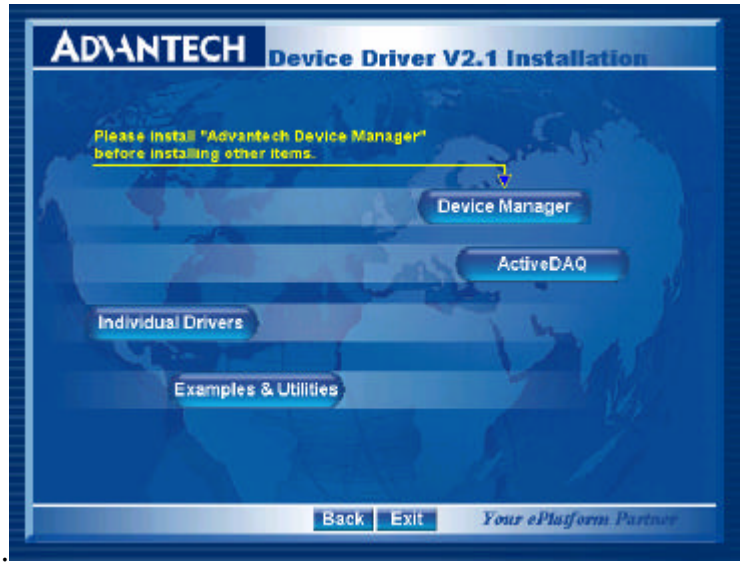


Fig. 2-1 The Setup Screen of Advantech Automation Software

Step 3: Select the *Individual Drivers* option.

Step 4: Select the specific device then just follow the installation instructions step by step to complete your device driver setup.

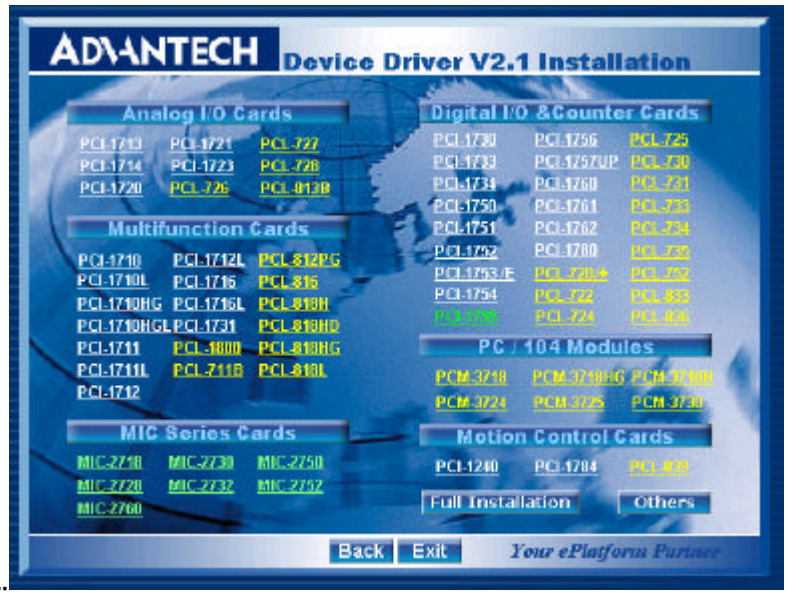


Fig. 2-2 Different options for Driver Setup

For further information on driver-related issues, an online version of *Device Drivers Manual* is available by accessing:

Start /Programs /Advantech Automation /Device Manager /Device Driver's Manual

2.3 Hardware Installation

After the DLL driver installation is completed, you can now go on to install the PCI-1755 card in any PCI slot on your computer. But it is suggested that you should refer to the computer user manual or related documentation if you have any doubt. Please follow the steps below to install the card on your system.

Note:

☞ Make sure you have installed the driver first before you install the card (please refer to *2.2 Driver Installation*)

Step 1: Turn off your computer and unplug the power cord and cables. TURN OFF your computer before installing or removing any components on the computer.

Step 2: Remove the cover of your computer.

Step 3: Remove the slot cover on the back panel of your computer.

Step 4: Touch the metal part on the surface of your computer to neutralize the static electricity that might be on your body.

Step 5: Insert the PCI-1755 card into a PCI slot. Hold the card only by its edges and carefully align it with the slot. Insert the card firmly into place. Use of excessive force must be avoided, otherwise the card might be damaged.

Step 6: Fasten the bracket of the PCI card on the back panel rail of the computer with screws.

Step 7: Connect appropriate accessories (100-pin cable, wiring terminals, etc. if necessary) to the PCI card.

Step 8: Replace the cover of your computer chassis. Re-connect the cables you removed in Step 2.

Step 9: Plug in the power cord and turn on the computer.

Note:

☞ In case you installed the card without installing the DLL driver first, Windows 95/98/ME will recognize your card as an "unknown device" after rebooting, and will prompt you to provide the necessary driver. You should ignore the prompting messages (just click the ***Cancel*** button) and set up the driver according to the steps described in *2.2 Driver Installation.*

After the PCI-1755 card is installed, you can verify whether it is properly installed on your system in the *Device Manager*:

1. Access the *Device Manager* through:
   Start /Control Panel /System /Device Manager.

2. The *device name* of the PCI-1755 should be listed on the *Device Manager* tab on the *System Property* Page.

..

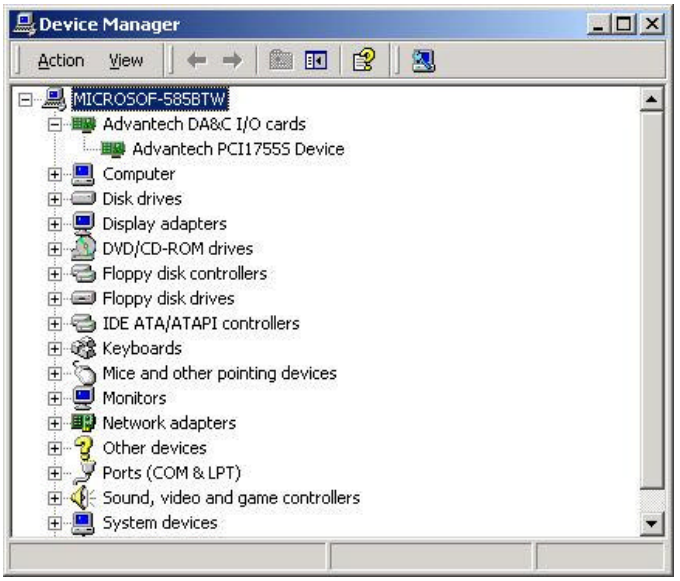


Fig. 2-3 The device name listed in the Device Manager

Note:

☞ If your card is properly installed, you should see the *device name* of your card listed on the *Device Manager* tab. **If you do see your device name listed on it but marked with an exclamation sign "!", it means your card has not been correctly installed.** In this case, remove the card device from the *Device Manager* by selecting its device name and press the ***Remove*** button. Then go through the driver installation process again.

After your card is properly installed on your system, you can now configure your device using the *Device Manager* program that has itself already been installed on your system during driver setup. A complete device installation procedure should include *board selection* and *device setup*. After that, you can operate this card through the *operation*. The following sections will guide you through the *board selection, device setup* and *operation* of your device.

2.4 Device Setup & Configuration

The *Device Manager* program is a utility that allows you to setup, configure and test your device, and later store your settings on the system registry. These settings will be used when you call the APIs of *Advantech Device Drivers*.

Setting Up and Configuring the Device

Step 1: To install the I/O device for your card, you must first run the *Device Manager* program by accessing:

Start /Programs /Advantech Automation /Device Manager /Advantech Device Manager

Step 2: You can then view the device(s) already installed on your system (if any) in the *Installed Devices* list box. Since you haven't installed any device yet, you might see a blank list such as the one below (***Fig. 2-4***).

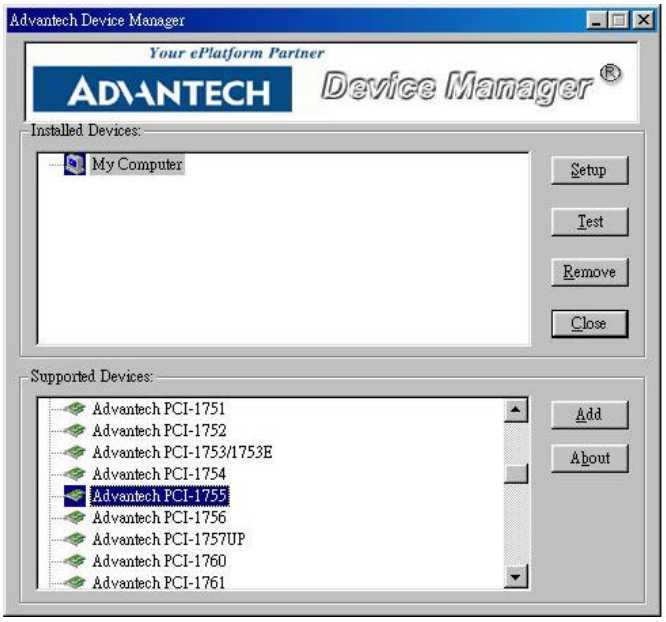


Fig. 2-4 The Device Manager dialog box

Step 3: Scroll down the *Supported Devices* box to find the device that you want to install, then click the ***Add…*** button to evoke the *Device(s) Found* dialog box. It lists all the installed devices on your system. Select the device you want to configure from the list box and press the ***OK*** button.

Step 4: After you have finished configuring the device, click ***OK*** and the *device name* will appear in the *Installed Devices* box as the following (***Fig. 2-5***).

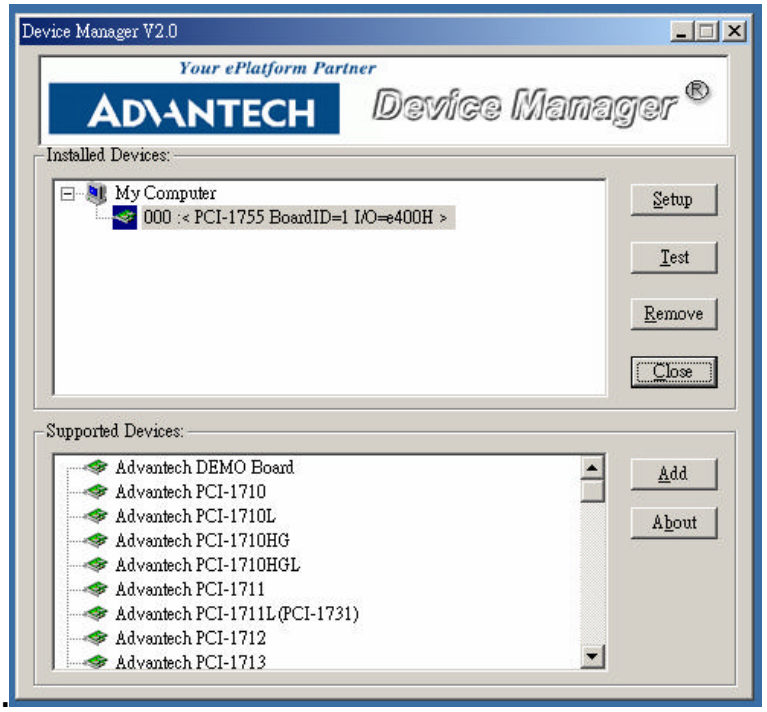


Fig. 2-5 The Device Manager dialog box

Note:

As we have noted, the *device name* **"000:<PCI-1755 BoardID=1 I/O=e400H>"** begins with a *device number* "000", which is specifically assigned to each card. The *device number* is passed to the driver to specify which device you wish to control.

After your card is properly installed and configured, you can click the ***Test…*** button to test your hardware. For more detailed information, please refer to *Chapter 2* of the *Device Drivers Manual*.

You can also find the rich examples on the CD-ROM to speed up your programming.

Chapter

3

Signal Connections

(This page is left blank for hard printing.)

3. Signal Connections

3.1 Overview

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly. A good signal connection can avoid unnecessary and costly damage to your PC and other hardware devices. This chapter provides useful information about how to connect input and output signals to the PCI-1755 via the I/O connector.

3.2 Switch and Jumper Settings

The PCI-1755 card has one function switch and five jumper settings.

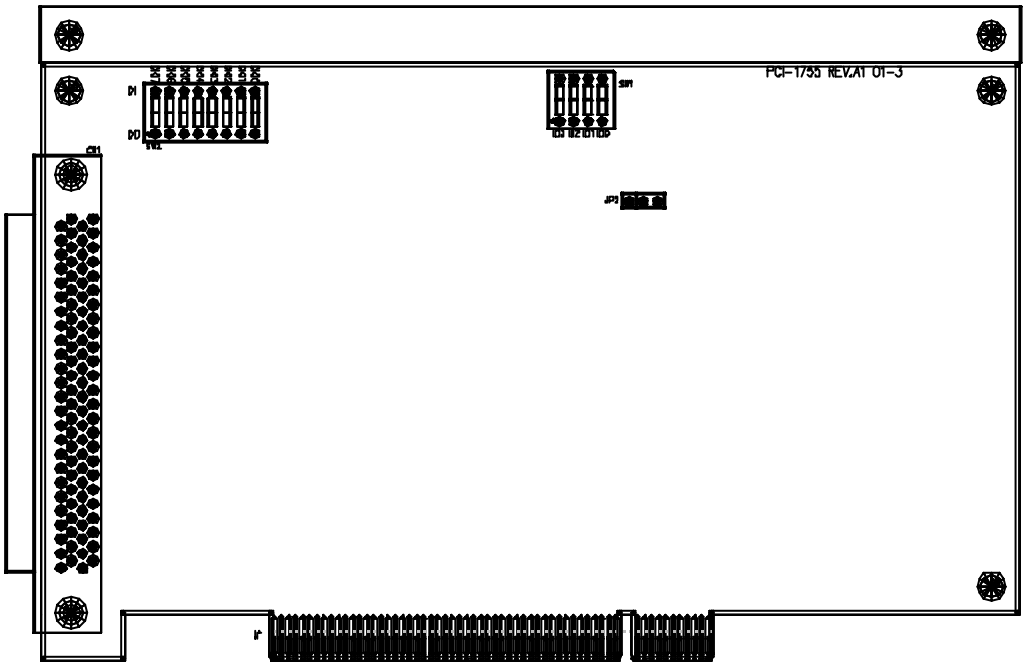


Fig. 3-1 Card connector, jumper and switch locations

Table 1-1: Summary of jumper settings

Names of Jumpers	Function Description	
JP2		Keep last status after hot reset
		Default configuration

Board ID setting (SW1)

ID3	ID2	ID1	ID0	Board ID
1	1	1	1	0
1	1	1	0	1
1	1	0	1	2
1	1	0	0	3
1	0	1	1	4
1	0	1	0	5
1	0	0	1	6
1	0	0	0	7
0	1	1	1	8
0	1	1	0	9
0	1	0	1	10
0	1	0	0	11
0	0	1	1	12
0	0	1	0	13
0	0	0	1	14
0	0	0	0	15

You can configure the Auxiliary DIO0~DIO7 (Pin23~Pin29) randomly by SW2. For instance, if you configure SW2 as (A3)H, it means DIO2, DIO3, DIO4, DIO6 were configured as digital output and DIO0, DIO1, DIO5, DIO7 were configured as digital input.

Auxiliary DI/O Setting (SW2)

	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
Digital Output	0	0	0	0	0	0	0	0
Digital Input	1	1	1	1	1	1	1	1

3.3 Signal Connections

Pin Assignment

Fig. 3-2 shows the pin assignments for the 100-pin I/O connector on the PCI-1755.

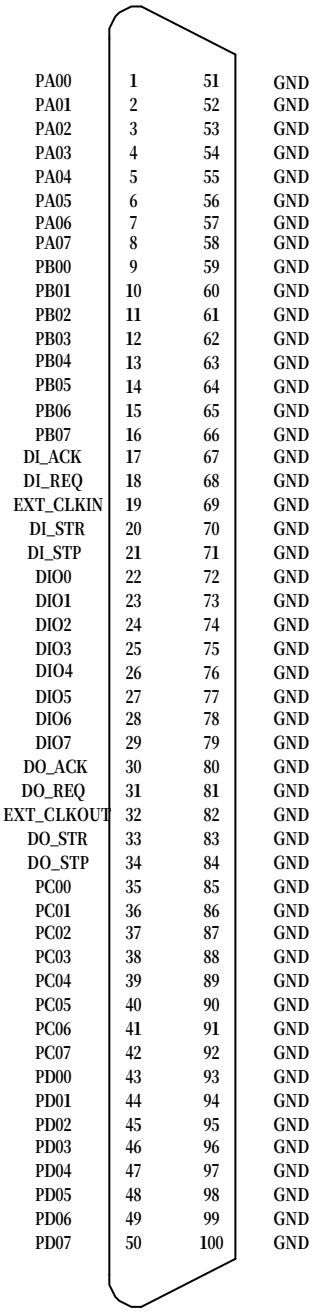


Fig. 3-2 I/O connector pin assignments for the PCI-1755

I/O Connector Signal Description

Table 3-2 I/O connector signal descriptions

Signal Name	Reference	Direction	Description
PA00~PA07	GND	I/O	Port A bi-directional DIO channels
PB00~PB07	GND	I/O	Port B bi-directional DIO channels
PC00~PC07	GND	I/O	Port C bi-directional DIO channels
PD00~PD07	GND	I/O	Port D bi-directional DIO channels
DI_ACK	GND	Output	Acknowledge line for digital input channels
DI_REQ	GND	Input	Request line for digital input channels
EXT_CLKIN	GND	Input	Clock input channel
DI_STR	GND	Input	Start trigger line for digital input channels
DI_STP	GND	Input	Stop trigger line for digital input channels
DO_ACK	GND	Input	Acknowledge line for digital output channels
DO_REQ	GND	Output	Request line for digital output channels
EXT_CLKOUT	GND	Output	Clock output channel
DO_STR	GND	Input	Start trigger line for digital output channels
DO_STP	GND	Input	Stop trigger line for digital output channels
DIO0~DIO7	GND	I/O	General-purpose digital input/output channels
GND	-	-	Ground reference for all other signals

Appendixes

(This page is left blank for hard printing.)

Appendix A. Specifications

Digital Input /Output (Part 1):

Channels	32 TTL compatible	
Number of ports	Port A, Port B, Port C and Port D (8 bits/port)	
I/O Configuration	32DI(PA~PD) (default); 32DO(PA~PD); 16DI(PA~PB) & 16DO (PC~PD); 8DI(PA) & 8DO(PC) (Programmable)	
On-board FIFO	16KB for DI & 16KB DO channels	
Transfer Characteristics	Data Transfer Mode	Bus Mastering DMA with Scatter-Gather
	Data Transfer Bus Width	8/16/32 bits (programmable)
	Max. Transfer Rate	DI: 80MBytes/sec, 32-bit@20MHz 120MBytes/sec, 32-bit@30MHz external pacer when data length is less than FIFO size DO: 80MBytes/sec, 32-bit@20MHz 120MBytes/sec, 32-bit@30MHz external pacer when data length is less than FIFO size
	Operation Mode	Handshaking
Handshaking Mode	Direction	I/O
	Asynchronous	8255 Emulation
	Synchronous	Burst Handshaking
	Clock source for Burst Handshaking	Internal: 30MHz, 15MHz, 10MHz, Timer#0 for DI & Timer#1 for DO External: EXT_CLKIN for DI & EXT_CLKOUT for DO
	Samples No.	Finite transfer, Continuous I/O
Normal Mode	Input	Data Acquisition at a predetermined rate by internal/external clock
	Output	Pattern Generation at a predetermined rate by internal/external clock
	Clock Source for DI	Internal: 30MHz, 15MHz, 10MHz, Timer#0 External: EXT_CLKIN
	Clock Source for DO	Internal: 30MHz, 15MHz, 10MHz, Timer#1 External: EXT_CLKIN
	Start Mode	Software command / Trigger signal occurred from DI_STR or DO_STR / Pattern DI
	Stop Mode	Software command / Trigger signal occurred from DI_STP (for DI) or DO_STR (for DO) / Pattern DI / "Finite transfers"
Change Detection (DI only)	Monitor the selected input channel and capture data whenever there is a transition on one of the channels, and then issue a IRQ	
	Clock Source for DI	Internal: 30MHz, 15MHz, 10MHz, Timer#0 External: EXT_CLKIN
	Start Mode	Software command / Trigger signal occurred from DI_STR / Pattern DI
	Stop Mode	Software command / Trigger signal occurred from DI_STP / Pattern DI / "Finite transfers"

Digital Input /Output (Part 2):

Trigger Capability	DI trigger signal	DI_STR, DI_STP
	DO trigger signal	DO_STR, DO_STP
	Low	0.8 V max.
	High	2.0 V min.
	Trigger Type	Rising or falling edge, or digital pattern (for DI only)
	Pulse width for edge triggers	10 ns min.
	Pattern trigger detection capabilities	Detect pattern match on user-selected data lines
Messaging	The messages can be generated when 1. Specified number of bytes have been transferred 2. When a specified input pattern is matched 3. When a measurement operation completes.	
Terminator	On-board Schottky diode termination	
Input Voltage	Low	0V min.; 0.8 V max.
	High	2.0 V min.; 5V max.
Input Load	Terminator OFF: TTL compatible	
	Low	+0.5V@ ±20mA
	High	+2.7V@ ±1mA max.
	Terminator ON	
	Terminator Resistor	110Ω
	Termination Voltage	2.9 V
	Low	+0.5V@ ±22.4mA
	High	+2.7V@ ±1mA max.
Output Voltage	Low	0.5V max.
	High	2.7V min.
Driving Capacity	Low	0.5 V max.@+48 mA (sink)
	High	2.4 V min.@-15 mA (source)
Hysteresis	500 mV	
General-purpose DI/O	DI Channels	DI0 ~ DI7 (TTL compatible)
	DO Channels	DO0 ~ DO7 (TTL compatible)
Interrupt Source	DI0~7 and Timer#2, Pattern match and Change detection, DI FIFO overflow and DO FIFO underflow, DI_STP and DO_STP	
Power Available at I/O connector	+4.65 ~ +5.25 V_{DC} @ 1A	

Pacer:

Channels	Timer#0, Timer#1 and Timer#2
Timer#0	Timer pacer for digital input
Timer#1	Timer pacer for digital output
Timer#2	Interrupt source
Resolution	16-bit
Base Clock	10MHz

Cable:

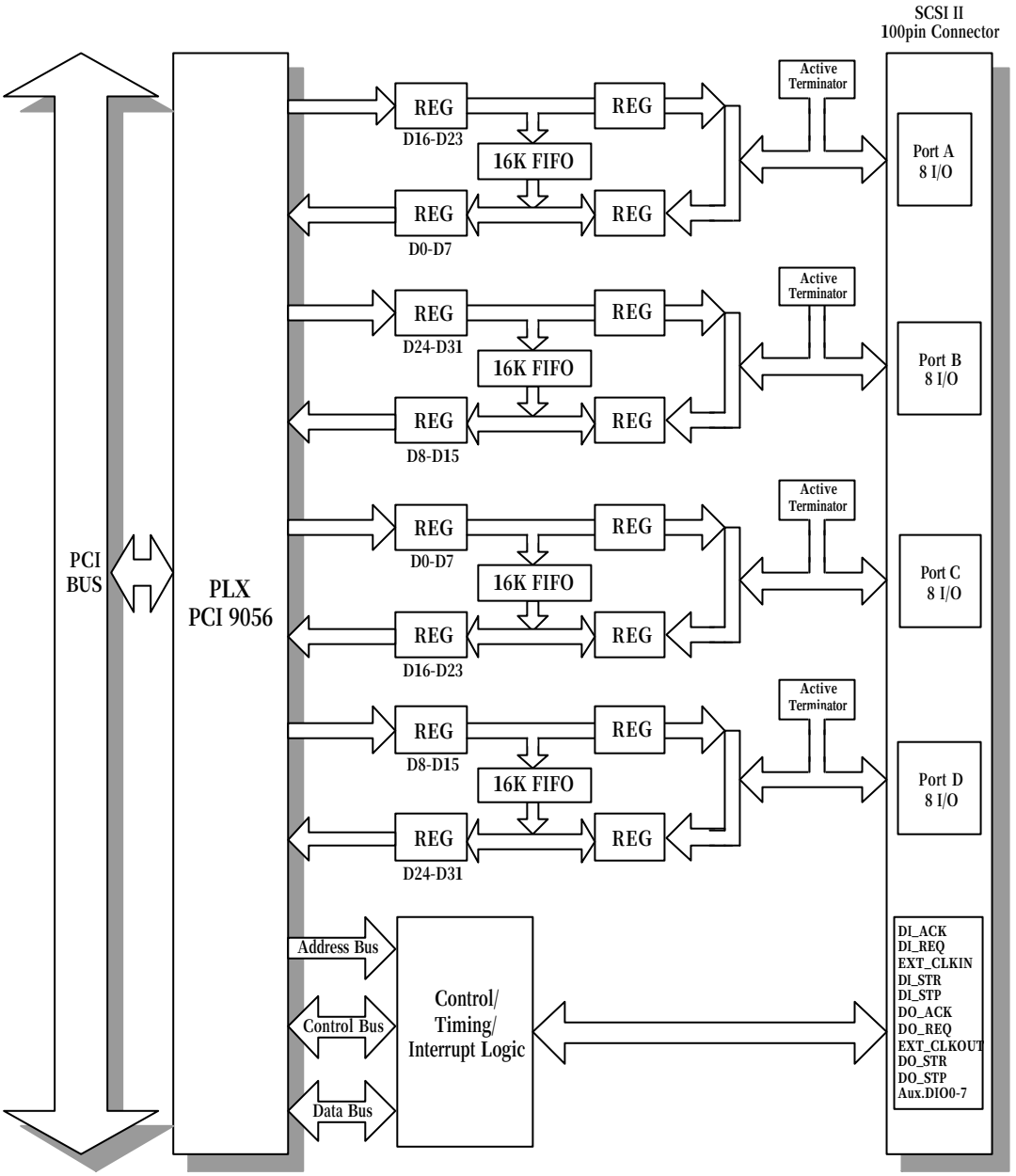
I/O Connector Type	100/100-pin SCSI-II male/male
Length	1m
Type	Twisted-pair cable. Each signal conductor is twisted with a ground conductor that establishes a low-inductance uniform transmission line
Termination scheme	Using the Schottky-Diode Termination Scheme to prevent from overshooting, undershooting and reflection phenomenon (Resistive termination scheme is not recommended because of the current drawn by the termination resistors)

General:

I/O Connector Type	100-pin SCSI-II female	
Dimensions	175 mm x 100 mm (6.9" x 3.9")	
Power Consumption	Typical	Terminator OFF: +5 V @ 1.07 A Terminator ON: +5 V @ 1.1 A
	Max.	Terminator OFF: +5 V @ 1.32 A Terminator ON: +5 V @ 1.36 A
Temperature	Operation	0~+60°C(32~140°F) (refer to *IEC 68-2-1,2)*
	Storage	-20~+85°C(-4~185°F)
Relative Humidity	5~95%RH non-condensing (refer to *IEC 68-2-3*)	
Certification	CE certified	

(This page is left blank for hard printing.)

Appendix B. Block Diagram



(This page is left blank for hard printing.)

Appendix C. Register Structure and Format

C.1 Overview

The PCI-1755 is delivered with an easy-to-use 32-bit DLL driver for user programming under the Windows 2000/95/98/NT/ME/XP operating system. We advise users to program the PCI-1755 using the 32-bit DLL driver provided by Advantech to avoid the complexity of low-level programming by register.

The most important consideration in programming the PCI-1755 at register level is to understand the function of the card's registers. The information in the following sections is provided only for users who would like to do their own low-level programming.

C.2 I/O Port Address Map

The PCI-1755 requires 32 consecutive addresses in the PC's I/O space. The address of each register is specified as an offset from the card's base address. For example, BASE+0 is the card's base address and BASE+6 is the base address plus six bytes.

Table C-1 shows the function of each register of the PCI-1755 or driver and its address relative to the card's base address.

Table C-1 PCI-1755 register format (Part 1)

Base Address + HEX		PCI-1755 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	W	**DI Control Register**															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
	R	**DI Status Register**															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
02H	W	**DI Control Register**															
									HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF
	R	**DI Status Register**															
		OV	FF	FH	FE				HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF
04H	W	**DO control register**															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
	R	**DO status register**															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
06H	W	**DO control register**															
									HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF
	R	**DO status register**															
		UN	FF	FH	FE				HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF
08H	W	**Pattern match register**															
		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	*C2*	C1	C0
	R	**Pattern match register**															
		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	*C2*	C1	C0
0AH	W	**Pattern match register**															
		C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16
	R	**Pattern match register**															
		C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16
0CH	W	**Interrupt control register**															
		DI7 RF	DI6 RF	DI5 RF	DI4 RF	DI3 RF	DI2 RF	DI1 RF	DI0 RF	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	R	**Interrupt status register**															
		DI7 RF	DI6 RF	DI5 RF	DI4 RF	DI3 RF	DI2 RF	DI1 RF	DI0 RF	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0EH	W	**Interrupt control register**															
		IE		DO_ STP	DI_S TP	UN	OV	TM	PM	CD			CH4	CH3	CH2	CH1	CH0
	R	**Interrupt status register**															
		IF		DO_S TP	DI_S TP	**UN**	OV	TM	PM	CD			CH4	CH3	CH2	CH1	CH0

Table C-1 PCI-1755 register format (Part 2)

Base Address + HEX		PCI-1755 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10H	W	**DO FIFO direct**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	*D2*	D1	D0
	R	**DI FIFO direct**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	*D5*	D4	D3	*D2*	D1	D0
12H	W	**DO FIFO direct**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
	R	**DI FIFO direct**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
14H	W	**DI/O CLK STR and STP**															
																DO_CLK	DI_CLK
	R	**Board ID**															
														BD3	BD2	BD1	BD0
16H	W	**DI/O CLK STR and STP**															
														DW1	DW0	DO_TERM	DI_TERM
	R	**Board ID**															
														DW1	DW0	DO_TERM	DI_TERM
18H	W	**Clear Interrupt**															
		Write this address clear interrupt															
	R																
1AH	W	**Clear Interrupt**															
		Write this address clear interrupt															
	R																
1CH	W	**Clear FIFO**															
														UN	OV	DO	DI
	R																
1EH	W	**Clear FIFO**															
	R																

Table C-1 PCI-1755 register format (Part 3)

Base Address + HEX		PCI-1755 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
20H	W	8254 Counter 0 Register															
	R	8254 Counter 0 Register															
22H	W	8254 Counter 0 Register															
	R	8254 Counter 0 Register															
24H	W	8254 Counter 1 Register															
	R	8254 Counter 1 Register															
26H	W	8254 Counter 1 Register															
	R	8254 Counter 1 Register															
28H	W	8254 Counter 2 Register															
	R	8254 Counter 2 Register															
2AH	W	8254 Counter 2 Register															
	R	8254 Counter 2 Register															
2CH	W																
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	*D2*	D1	D0
	R	8254 Control Register															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
2EH	W																
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
	R	8254 Control Register															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

Base Address + HEX		PCI-1755 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30H	W	**Auxiliary DO**															
										DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	R	**Auxiliary DI/O**															
		DIP7	DIP6	DIP5	DIP4	DIP3	DIP2	DIP1	DIP0	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
32H	W	**Auxiliary DO**															
	R	**Auxiliary DI/O**															
34H	W	**DO value preset**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	*D2*	D1	D0
	R	**DI Value Preview**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
36H	W	**DO value preset**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
	R	**DI Value Preview**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
38H	W	**DO FIFO Out**															
		Write this address DO FIFO out to DO port															
	R	**DI FIFO in**															
		Read this address DI port value into DI FIFO															
3AH	W	**DO FIFO Out**															
		Write this address DO FIFO out to DO port															
	R	**DI FIFO in**															
		Read this address DI port value into DI FIFO															

C.3 DI Control Register--- BASE+0

Table C-2 PCI-1755 Register for DI Control Register

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	W	DI Control Register															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
02H	W	DI Control Register															
									HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF

1. M1:M0 = Digital input mode
   - 00 Disable
   - 01 Normal mode
   - 10 Handshaking mode
2. SC2:SC0 = Sampling clock select
   - 000 Disable
   - 001 30 MHz sampling clock
   - 010 15 MHz sampling clock
   - 011 10 MHz sampling clock
   - 100 Timer0 output of 8254
   - 101 External clock input by EXT_CLKIN
3. STR1:STR0 = Start mode of normal DI
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DI_STR
   - 11 Pattern DI
4. STP1:STP0 = Stop mode of normal DI
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DI_STP
   - 11 Pattern DI
5. STRRF = DI_STR triggering control
   - 0 Rising edge trigger
   - 1 Falling edge trigger
6. STPRF = DI_STP triggering control
   - 0 High level active
   - 1 Low level active

7. REQRF = DI_REQ triggering control

0 High level active

1 Low level active

8. ACKRF = DI_ACK triggering control

0 High level active

1 Low level active

9. CLKRF = Sampling clock triggering control

0 Rising edge trigger

1 Falling edge trigger

10.HS0 = Handshaking mode

0 Burst Handshaking

1 8255 Emulation

C.4 DO Control Register--- BASE+4H

Table C-3 PCI-1755 Register for DO Control Register

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
04H	W	**DO control register**															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
06H	W	**DO control register**															
									HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF

1. M1:M0 = Digital output mode
   - 00 Disable
   - 01 Normal mode
   - 10 Handshaking mode
2. SC2:SC0 = Sampling clock select
   - 000 Disable
   - 001 30 MHz sampling clock
   - 010 15 MHz sampling clock
   - 011 10 MHz sampling clock
   - 100 Timer1 output of 8254
   - 101 External clock input by EXT_CLKIN
3. STR1:STR0 = Start mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DO_STR
4. STP1:STP0 = Stop mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DO_STP
5. STRRF = DO_STR triggering control
   - 0 Rising edge trigger
   - 1 Falling edge trigger
6. STPRF = DO_STP triggering control
   - 0 High level active
   - 1 Low level active
7. REQRF = DO_REQ triggering control
   - 0 High level active

1 Low level active

8. ACKRF = DO_ACK triggering control

0 High level active

1 Low level active

9. CLKRF = Sampling clock triggering control

0 Rising edge trigger

1 Falling edge trigger

10.HS0 = Handshaking mode

0 Burst Handshaking

1 8255 Emulation

C.5 DO Pattern Match Register--- BASE+8H

Table C-4 PCI-1755 Register for DO Pattern Match Register

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
08H	W	**Pattern match register**															
		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	*C2*	C1	C0
0AH	W	**Pattern match register**															
		C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16

1. C31:C0 = Compare data

C.6 Interrupt Control Register--- BASE+0CH

Table C-5 PCI-1755 Register for Interrupt Control

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	W	**Interrupt control register**															
		DI7 RF	DI6 RF	DI5 RF	DI4 RF	DI3 RF	DI2 RF	DI1 RF	DI0 RF	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0EH	W	**Interrupt control register**															
		IE		DO_ STP	DI_S TP	UN	OV	TM	PM	CD			CH4	CH3	CH2	CH1	CH0

1. DIn = Interrupt by digital input enable
1. DInRF = DI triggering control
   0 Rising edge trigger
   1 Falling edge trigger
3. CH4:CH0 = Channel of change detection
4. CD = Interrupt by DI change detection enable
5. PM = Interrupt by pattern match enable
6. TM = Interrupt by 8254 timer2 enable
7. OV = DI FIFO overflow enable
8. UN = DO FIFO underflow enable
9. DI_STP = Interrupt by DI_STP enable
10. DO_STP = Interrupt by DO_STP enable
11. IE = Interrupt enable
    0 No occurred
    1 Occurred

(n = 0 to 7)

C.7 DO FIFO Direct --- BASE+10H

Table C-6 PCI-1755 Register for DO FIFO Direct

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10H	W	**DO FIFO direct**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	*D2*	D1	D0
12H	W	**DO FIFO direct**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

1. D31:D0 = DO FIFO direct data

C.8 DI/O CLK STR and STP --- BASE+14H

Table C-7 PCI-1755 Register for DI/O CLK STR and STP

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14H	W	**DI/O CLK STR and STP**															
																DO_CLK	DI_CLK
16H	W	**DI/O CLK STR and STP**															
														DW1	DW0	DO_TERM	DI_TERM

1. DI_CLK = DI CLK command
   - 0 DI stop triggering
   - 1 DI start triggering
2. DO_CLK = DO CLK command
   - 0 DO stop triggering
   - 1 DO start triggering
3. DI_TERM = DI Terminator OFF/ON
   - 0 Terminator ON
   - 1 Terminator OFF
4. DO_TERM = DO Terminator OFF/ON
   - 0 Terminator ON
   - 1 Terminator OFF
5. DW1:DW0 = Double word wide
   - 00 DI port is Double word (32-bit) wide
   - 01 DO port is Double word (32-bit) wide
   - 10 Both of DI and DO ports are Word (16-bit) wide
   - 11 Both of DI and DO ports are Byte (8-bit) wide

C.9 Clear Interrupt --- BASE+18H

Table C-8 PCI-1755 Register for Clear Interrupt

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
18H	W	**Clear Interrupt**															
		Write this address clear interrupt															
0AH	W	**Clear Interrupt**															
		Write this address clear interrupt															

Write this address clear interrupt

C.10 Clear FIFO --- BASE+1C H

Table C-9 PCI-1755 Register for Clear FIFO

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	W	**Clear FIFO**															
														UN	OV	DO	DI
1EH	W	**Clear FIFO**															

1. DI = Clear DI FIFO command
   0 N/A
   1 Clear DI FIFO
2. DO = Clear DO FIFO command
   0 N/A
   1 Clear DO FIFO
3. OV = Clear DI FIFO overflow flag
   0 N/A
   1 Clear DI FIFO overflow flag
4. UN = Clear DO FIFO underflow flag
   0 N/A
   1 Clear DO FIFO underflow flag

C.11 Auxiliary DO --- BASE+30 H

Table C-10 PCI-1755 Register for Auxiliary DO

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30H	W	**Auxiliary DO**															
										DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
32H	W	**Auxiliary DO**															

1. DOn = Digital output data

(n = 0 to 7)

C.12 DO Value Preset --- BASE+34 H

Table C-11 PCI-1755 Register for DO Value Preset

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
34H	W	**DO value preset**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	*D2*	D1	D0
36H	W	**DO value preset**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

1. D31:D0 = DO port value preset

C.13 DO FIFO Out--- BASE+38 H

Table C-12 PCI-1755 Register for DO FIFO Out

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
38H	W	**DO FIFO Out**															
		Write this address DO FIFO out to DO port															
3AH	W	**DO FIFO Out**															
		Write this address DO FIFO out to DO port															

Write this address DO FIFO out to DO port

C.14 DI Status Register--- BASE+0 H

Table C-13 PCI-1755 Register for DI Status

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	R	DI Status Register															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
02H	R	DI Status Register															
		OV	FF	FH	FE				HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF

1. M1:M0 = Digital input mode
   - 00 Disable
   - 01 Normal mode
   - 10 Handshaking mode
2. SC2:SC0 = Sampling clock select
   - 000 Disable
   - 001 30 MHz sampling clock
   - 010 15 MHz sampling clock
   - 011 10 MHz sampling clock
   - 100 Timer0 output of 8254
   - 101 External clock input by EXT_CLKIN
3. STR1:STR0 = Start mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DI_STR
   - 11 Pattern DI
4. STP1:STP0 = Stop mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DI_STP
   - 11 Pattern DI
5. STRRF = DI_STR triggering control
   - 0 Rising edge trigger
   - 1 Falling edge trigger
6. STPRF = DI_STP triggering control
   - 0 High level active
   - 1 Low level active

7. REQRF = DI_REQ triggering control

0 High level active

1 Low level active

8. ACKRF = DI_ACK triggering control

0 High level active

1 Low level active

9.CLKRF = Sampling clock triggering control

0 Rising edge trigger

1 Falling edge trigger

10.HS0 = Handshaking mode

0 Burst Handshaking

1 8255 Emulation

11.FE = FIFO empty

0 No occurred

1 Occurred

12.FH = FIFO half full

0 No occurred

1 Occurred

13.FF = FIFO full

0 No occurred

1 Occurred

14.OV = FIFO overflow flag

0 No occurred

1 Occurred

C.15 DO Status Register--- BASE+4 H

Table C-14 PCI-1755 Register for DO Status

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
04H	R	DO status register															
						STP1	STP0	STR1	STR0		SC2	SC1	SC0			M1	M0
06H	R	DO status register															
		UN	FF	FH	FE				HS0				CLK RF	ACK RF	REQ RF	STP RF	STR RF

1. M1:M0 = Digital output mode
   - 00 Disable
   - 01 Normal mode
   - 10 Handshaking mode
2. SC1:SC0 = Sampling clock select
   - 000 Disable
   - 001 30 MHz sampling clock
   - 010 15 MHz sampling clock
   - 011 10 MHz sampling clock
   - 100 Timer1 output of 8254
   - 101 External clock input by EXT_CLKIN
3. STR1:STR0 = Start mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DO_STR
4. STP1:STP0 = Stop mode of normal mode
   - 00 Disable
   - 01 Software command
   - 10 Trigger signal occurred from DO_STP
5. STRRF = DO_STR triggering control
   - 0 Rising edge trigger
   - 1 Falling edge trigger
6. STPRF = DO_STP triggering control
   - 0 High level active
   - 1 Low level active
7. REQRF = DO_REQ triggering control
   - 0 High level active

1 Low level active

8. ACKRF = DO_ACK triggering control

0 High level active

1 Low level active

9. CLKRF = Sampling clock triggering control

0 Rising edge trigger

1 Falling edge trigger

10.HS0 = Handshaking mode

0 Burst Handshaking

1 8255 Emulation

11.FE = FIFO empty

0 No occurred

1 Occurred

12.FH = FIFO half full

0 No occurred

1 Occurred

13.FF = FIFO full

0 No occurred

1 Occurred

14.UN = FIFO underflow flag

0 No occurred

1 Occurred

C.16 Pattern Match Register--- BASE+8 H

Table C-15 PCI-1755 Register for Pattern Match

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
08H	R	**Pattern match register**															
		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
0AH	R	**Pattern match register**															
		C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16

1. C31:C0 = Compare data

C.17 Interrupt Status Register--- BASE+0C H

Table C-16 PCI-1755 Register for Interrupt Status

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	R	Interrupt status register															
		DI7 RF	DI6 RF	DI5 RF	DI4 RF	DI3 RF	DI2 RF	DI1 RF	DI0 RF	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0EH	R	Interrupt status register															
		IF		DO_STP	DI_STP	UN	OV	TM	PM	CD			CH4	CH3	CH2	CH1	CH0

1. DIn = Interrupt by digital input flag
2. DInRF = DI triggering control
   - 0 Rising edge trigger
   - 1 Falling edge trigger
3. CH4:CH0 = Channel of change detection
4. CD = Interrupt by DI change detection flag
5. PM = Interrupt by pattern match flag
6. TM = Interrupt by 8254 timer2 flag
7. OV = DI FIFO overflow flag
8. UN = DO FIFO underflow flag
9. DI_STP = Interrupt by DI_STP flag
10. DO_STP = Interrupt by DO_STP flag
11. IF = Interrupt flag
    - 0No occurred
    - 1Occurred

(n = 0 to 7)

C.18 DI FIFO Direct--- BASE+10 H

Table C-17 PCI-1755 Register for DI FIFO Direct

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10H	R	**DI FIFO direct**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	*D5*	D4	D3	*D2*	D1	D0
12H	R	**DI FIFO direct**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

1. D31:D0 = DI FIFO direct data

C.19 Board ID--- BASE+14 H

Table C-18 PCI-1755 Register for Board ID

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14H	R	**Board ID**															
														BD3	BD2	BD1	BD0
16H	R	**Board ID**															
														DW1	DW0	DO_ TERM	DI_ TERM

1. BD3:BD0 = Board ID

2. DI_TERM = DI Terminator OFF/ON

   0 Terminator ON

   1 Terminator OFF

3. DO_TERM = DO Terminator OFF/ON

   0 Terminator ON

   1 Terminator OFF

4. DW1:DW0 = Double word wide

   00 DI port is Double word (32-bit) wide

   01 DO port is Double word (32-bit) wide

   10 Both of DI and DO ports are Word (16-bit) wide

   11 Both of DI and DO ports are Byte (8-bit) wide

C.20 Auxiliary DI/O --- BASE+30 H

Table C-19 PCI-1755 Register for Auxiliary DI/O

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30H	R	**Auxiliary DI/O**															
		DIP7	DIP6	DIP5	DIP4	DIP3	DIP2	DIP1	DIP0	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
32H	R	**Auxiliary DI/O**															

1. DIOn = Digital input/output data
2. DIPn = Dip switch (DI/O) value

   0 Digital output

   1 Digital input

   (n = 0 to 7)

C.21 DI Value Preview--- BASE+34 H

Table C-20 PCI-1755 Register for DI Value Preview

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
34H	R	**DI Value Preview**															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
36H	R	**DI Value Preview**															
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

D31:D0 = DI port value preview

C.22 DI FIFO in--- BASE+38 H

Table C-21 PCI-1755 Register for DI FIFO in

Base Addr.		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
38H	R	**DI FIFO in**															
		Read this address DI port value into DI FIFO															
3AH	R	**DI FIFO in**															
		Read this address DI port value into DI FIFO															

Read this address DI port value into DI FIFO

(This page is left blank for hard printing.)

Appendix D. 82C54 Counter Function

D.1 Overview

The PCI-1755 uses one Intel 82C54 compatible programmable interval timer/counter chip. The popular 82C54 offers three independent 16-bit counters, counter 0, counter 1 and counter 2. Each counter has a clock input, control gate and an output. You can program each counter for maximum count values from 2 to 65535.
The 82C54 has a maximum input clock frequency of 10 MHz. The PCI-1755 provides 10 MHz input frequencies to the counter chip from an on-board crystal oscillator.

Counter 0
On the PCI-1755 counter 0 can be a 16-bit timer or an event counter, selectable by users. When the clock source is set as an internal source, counter 0 is a 16-bit timer; when set as an external source, then counter 0 is an event counter and the clock source comes from **CNT0_CLK**. The counter is controlled by **CNT0_GATE**. When **CNT0_GATE** input is high, counter 0 will begin to count.

Counter 1 & 2
Counter 1 and counter 2 of the counter chip are cascaded to create a 32-bit timer for the pacer trigger. A low-to-high edge of counter 2 output (**PACER_OUT**) will trigger an A/D conversion. At the same time, you can use this signal as a synchronous signal for other applications.

D.2 Counter Read/Write and Control Registers

The 82C54 programmable interval timer uses four registers at addresses **BASE + 20H**, **BASE + 24H**, **BASE + 28H** and **BASE + 2CH** for read, write and control of counter functions. Register functions appear below:

Register	Function
BASE + 20H	Counter 0 read/write
BASE + 24H	Counter 1 read/write
BASE + 28H	Counter 2 read/write
BASE + 2CH	Control register

Since the 82C54 counter uses a 16-bit structure, each section of read/write data is split into a least significant byte (LSB) and most significant byte (MSB). To avoid errors it is important that you make read/write operations in pairs and keep track of the byte order. The data format for the control register is as below:

BASE+2CH 82C54 control bit, standard mode								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Description

SC1 & SC0 Select number

Counter	SC1	SC0
0	0	0
1	0	1
2	1	0
Read-back command	1	1

RW1 & RW0 Select read/write operation

Operation	RW1	RW0
Counter latch	0	0
Read/Write LSB	0	1
Read/Write MSB	1	0
Read/Write LSB first, then MSB	1	1

M2, M1 and M0 Select operation mode

M2	M1	M0	Mode	Description
0	0	0	0	Stop on terminal count
0	0	1	1	Programmable one shot
X	1	0	2	Rate generator
X	1	1	3	Square wave rate generator
1	0	0	4	Software triggered strobe
1	0	1	5	Hardware triggered strobe

BCD Select binary or BCD countering.

BCD	**Type**
0	Binary counting 16-bits
1	Binary coded decimal (BCD) counting

If you set the module for binary counting, the count can be any number from 0 up to 65535. If you set it for BCD (Binary Coded Decimal) counting, the count can be any number from 0 to 9999.

If you set both SC1 and SC0 bits to 1, the counter control register is in read-back command mode. The control register data format thenbecomes:

BASE+2CH 82C54 control bit, read-back mode								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	1	1	CNT	STA	C2	C1	C0	X

CNT=0 Latch count of selected counter(s)
STA=0 Latch status of selected counter(s)
C2, C1&C0 Select counter for a read-back operation
C2 = 1 select Counter 2
C1 = 1 select Counter 1
C0 = 1 select Counter 0

If you set both SC1 and SC0 to 1 and STA to 0, the register selected by C2 to C0 contains a byte which shows the status of the counter. The data format of the counter read/write register then becomes:

BASE+20/24/28H 82C54 Standard read-back mode								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	OUT	NC	RW1	RW0	M2	M1	M0	BCD

OUT	Current state of counter output
NC	Null count is 1 when the last count written to the counter register has been loaded into the counting element

D3. Counter Operating Modes

MODE 0 — Stop on Terminal Count

The output will initially be low after you set this mode of operation. After you load the count into the selected count register, the output will remain low and the counter will count. When the counter reaches the terminal count, its output will go high and remain high until you reload it with the mode or a new count value. The counter continues to decrement after it reaches the terminal count. Rewriting a counter register during counting has the following results:

1. Writing to the first byte stops the current counting.
2. Writing to the second byte starts the new count.

MODE 1 — Programmable One-shot Pulse

The output is initially high. The output will go low on the count following the rising edge of the gate input. It will then go high on the terminal count. If you load a new count value while the output is low, the new value will not affect the duration of the one-shot pulse until the succeeding trigger. You can read the current count at any time without affecting the one-shot pulse. The one-shot is retriggerable, thus the output will remain low for the full count after any rising edge at the gate input.

MODE 2 — Rate Generator

The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the counter register. If you reload the counter register between output pulses, the present period will not be affected, but the subsequent period will reflect the value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. You can thus use the gate input to synchronize the counter.

With this mode the output will remain high until you load the count register. You can also synchronize the output by software.

MODE 3 － Square Wave Generator

This mode is similar to Mode 2, except that the output will remain high until one half of the count has been completed (for even numbers), and will go low for the other half of the count. This is accomplished by decreasing the counter by two on the falling edge of each clock pulse. When the counter reaches the terminal count, the state of the output is changed, the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the count by 2. After time-out, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by two until time-out, then the whole process is repeated. In this way, if the count is odd, the output will be high for (N+1)/2 counts and low for (N-1)/2 counts.

MODE 4 － Software-Triggered Strobe

After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period then go high again.

If you reload the count register during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

MODE 5 － Hardware-Triggered Strobe

The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable.

D4. Counter Operations

Read/Write Operation

Before you write the initial count to each counter, you must first specify the read/write operation type, operating mode and counter type in the control byte and write the control byte to the control register [**BASE +2CH**].

Since the control byte register and all three counter read/write registers have separate addresses and each control byte specifies the counter it applies to (by **SC1** and **SC0**), no instructions on the operating sequence are required. Any programming sequence following the 82C54 convention is acceptable.

There are three types of counter operation: Read/load LSB, read /load MSB and read /load LSB followed by MSB. It is important that you make your read/write operations in pairs and keep track of the byte order.

Counter Read-back Command

The 82C54 counter read-back command lets you check the count value, programmed mode and current states of the **OUT** pin and **Null Count** flag of the selected counter(s). You write this command to the control word register. Format is as shown at the beginning of this section.

The read-back command can latch multiple counter output latches. Simply set the **CNT** bit to 0 and select the desired counter(s). This single command is functionally equivalent to multiple counter latch commands, one for each counter latched.

The read-back command can also latch status information for selected counter(s) by setting **STA** bit = 0. The status must be latched to be read; the status of a counter is accessed by a read from that counter. The counter status format appears at the beginning of the chapter.

Counter Latch Operation

Users often want to read the value of a counter without disturbing the count in progress. You do this by latching the count value for the specific counter then reading the value.

The 82C54 supports the counter latch operation in two ways. The first way is to set bits

RW1 and **RW0** to 0. This latches the count of the selected counter in a 16-bit hold register. The second way is to perform a latch operation under the read-back command. Set bits **SC1** and **SC0** to 1 and **CNT** = 0. The second method has the advantage of operating several counters at the same time. A subsequent read operation on the selected counter will retrieve the latched value.

Appendix E. Waveform of each mode

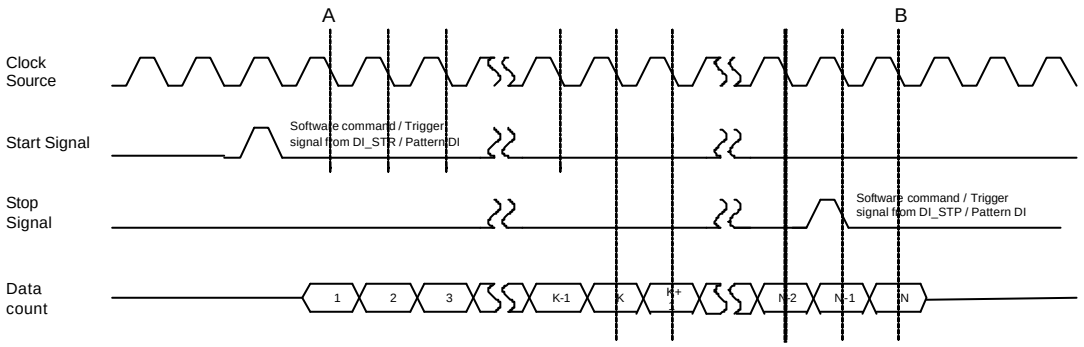
PCI-1755 provides two types of transmit modes for sample input data from external device to the PCI-1755 or output data from PCI-1755 to external device.

- Normal Mode
- Handshaking Mode

Normal Mode of Ultra-Speed Digital Input

In Normal mode of PCI-1755, you can start to transmit the data from external device to the PCI-1755 by start signal or stop it by stop signal. You can generate start or stop signal by software command, external trigger via **DI_STR/DI_STP** and pattern **DI**.

When PCI-1755 gets the start signal, it will start to receive data from external device at next clock (Point A). When PCI-1755 gets the stop signal, it will stop to receive the data at next clock (Point B).



Normal Mode of Ultra-speed Digital Input

NOTE:

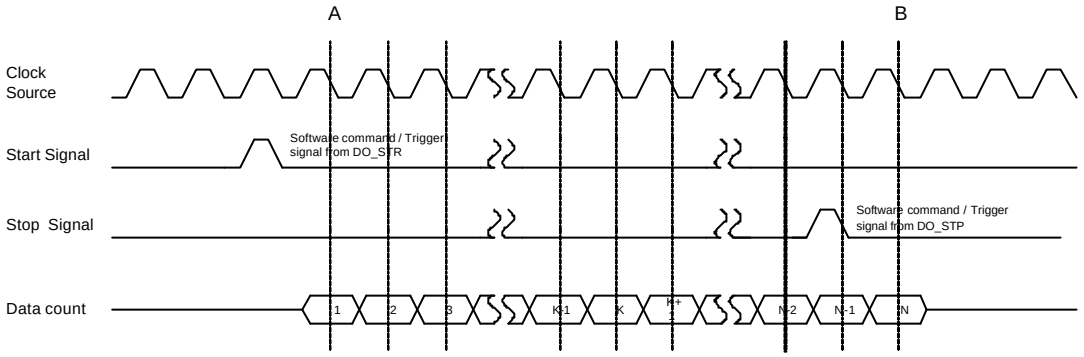
1. In this instance, start/stop signal are shown as active high.
2. Note that you can't generate start and stop signal by pattern **DI** at the same time.
3. There are two types of **DI** clock source listed below.

Internal	30MHz, 15MHz, 10MHz, timer#0
External	EXT_CLKIN

Normal Mode of Ultra-Speed Digital Output

In Normal mode of PCI-1755, you can start to transmit the data from PCI-1755 to the external device by start signal or stop it by stop signal. You can generate start or stop signal by software command, external trigger via **DO_STR**/**DO_STP**.

When PCI-1755 gets the start signal, it will start to send data to external device at next clock (Point A). When PCI-1755 gets the stop signal from external device, it will stop to send the data at next clock (Point B).



Normal Mode of Ultra-speed Digital Output

NOTE:

1. In this instance, start/stop signal are shown as active high.
2. There are two types of **DO** clock source listed below.

Internal	30MHz, 15MHz, 10MHz, timer#1
External	EXT_CLKIN

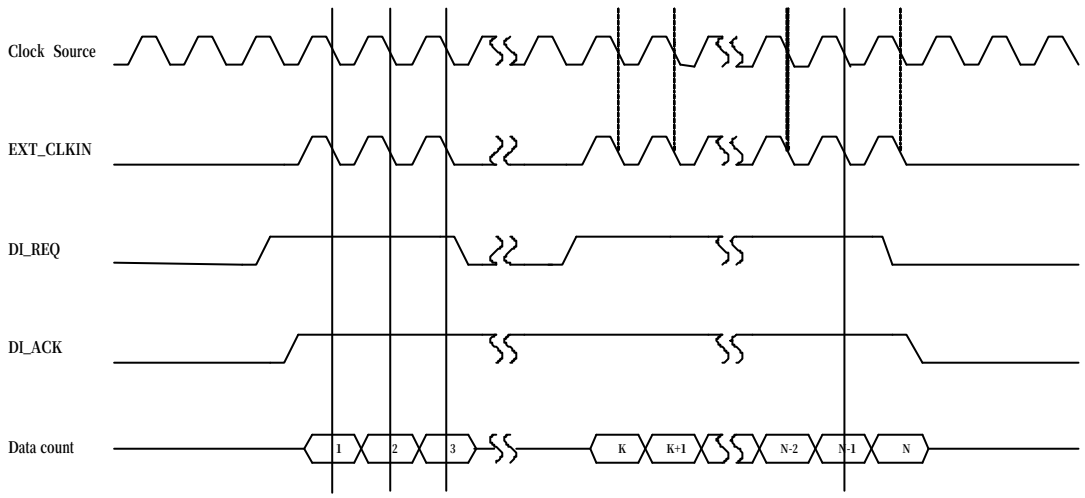
Handshaking mode

There are two different transmit modes for handshaking.

- Burst
- 8255 Emulation.

Handshaking mode of Burst Ultra-Speed Digital Input

For the Burst Ultra-Speed Digital Input, if the external device would like to transmit the data to PCI-1755, it will enable the **DI_REQ** signal to PCI-1755. If PCI-1755 is ready to get the data, it will also enable the **DI_ACK** signal to external device and then the data will be transmitting from external device to the PCI-1755 according to **EXT_CLKIN**.



Handshaking Mode of Ultra-speed Digital Input (Burst)

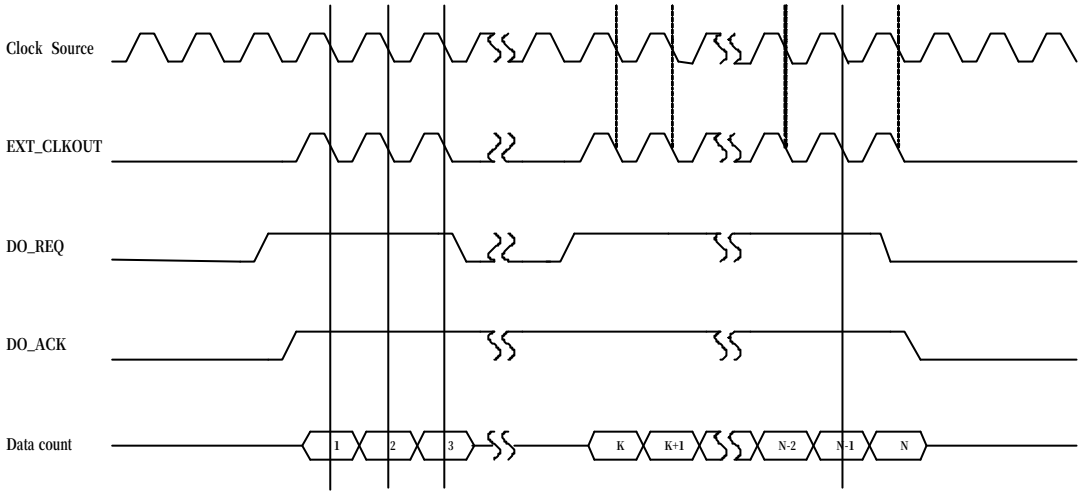
NOTE:

1. In this instance, **DI_REQ** and **DI_ACK** signal are shown as active high.
2. There are two types of **DI** clock source listed below.

Internal	30MHz, 15MHz, 10MHz, timer#0
External	EXT_CLKIN

Handshaking mode of Burst Ultra-Speed Digital Output

For the Burst Ultra-Speed Digital Output, if the PCI-1755 would like to transmit the data to the external device, it will enable the **DO_REQ** signal to external device. If the external device is ready to get the data, it will also enable the **DO_ACK** signal to PCI-1755 and then the data will be transmitting from PCI-1755 to the external device according to **EXT_CLKOUT**.



Handshaking Mode of Ultra-speed Digital Output (Burst)

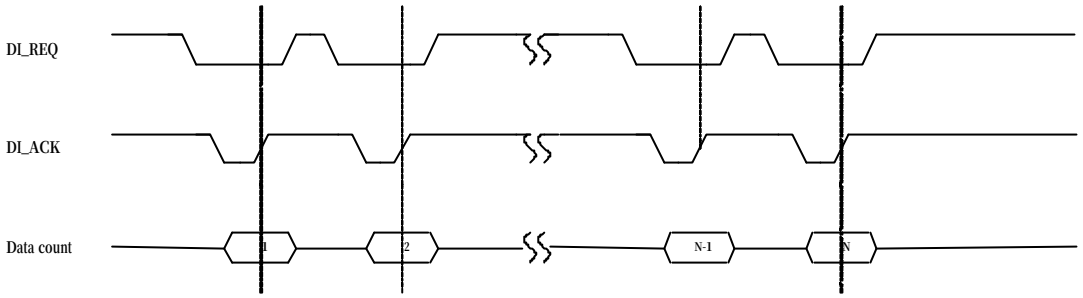
NOTE:

1. In this instance, **DO_REQ** and **DO_ACK** signal are shown as active high.
2. There are two types of **DO** clock source listed below.

Internal	30MHz, 15MHz, 10MHz, timer#1
External	EXT_CLKIN

Handshaking mode of 8255 Emulation Ultra-Speed Digital Input

For the 8255 Emulation Ultra-Speed Digital Input, if the external device would like to transmit the data to PCI-1755, it will send a **DI_REQ** signal to PCI-1755. If PCI-1755 is ready to get the data, it will also response a **DI_ACK** signal to external device and then one unit of data will be transmitting from external device to the PCI-1755.

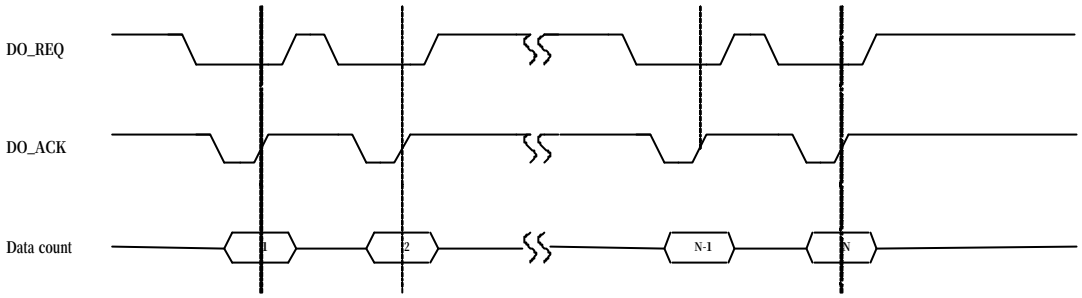


Handshaking Mode of Ultra-speed Digital Input (8255 Emulation)

NOTE:
The **DI_REQ** and **DI_ACK** signal are shown as active low ONLY in handshaking mode of 8255 Emulation.

Handshaking mode of 8255 Emulation Ultra-Speed Digital Output

For the 8255 Emulation Ultra-Speed Digital Output, the PCI-1755 would like to transmit the data to the external device; it will send a **DO_REQ** signal to external device. If the external device is ready to get the data, it will also response a **DO_ACK** signal to PCI-1755 and then one unit of data will be transmitting from PCI-1755 to the external device.



Handshaking Mode of Ultra-speed Digital Output (8255 Emulation)

NOTE:
The **DO_REQ** and **DO_ACK** signal are shown as active low ONLY in handshaking mode of 8255 Emulation.