NuDAQ® cPCI-7452

256-CH Isolated Digital I/O Card

User's Manual

Manual Rev. 2.01

Revision Date: December 21, 2006

Part No: 50-15053-1000

Recycled Paper

Advance Technologies; Automate the World.

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Trademarks

NuDAQ, NuIPC, DAQBench are registered trademarks of ADLINK TECHNOLOGY INC.

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting Service from ADLINK

Customer Satisfaction is top priority for ADLINK Technology Inc. Please contact us should you require any service or assistance.

ADLINK TECHNOLOGY INC.

Web Site:	http://www.adlinktech.com
Sales & Service:	Service@adlinktech.com
TEL:	+886-2-82265877
FAX:	+886-2-82265717
Address:	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan

Please email or FAX this completed service form for prompt and satisfactory service.

Company Information	
Company/Organization	
Contact Person	
E-mail Address	
Address	
Country	
TEL	FAX:
Web Site	
Product Information	
Product Model	
Environment	OS: M/B: CPU: Chipset: BIOS:

Please give a detailed description of the problem(s):

Table of Contents

List of Tables

List of Figures

1 Introduction

The ADLINK cPCI-7452 Isolated Digital I/O card is a 256-CH extra-high-density digital input and output card for the PCI-bus computer in industrial applications.

cPCI-7452 provides 128 opto-isolated digital inputs as well as 128 opto-isolated digital outputs. It provides a robust 2,500VRMS isolation protection, which is suitable for most industrial applications. All of the opto-isolated digital inputs are identical non-polarity; each digital input line is isolated respectively and suited to collect digital inputs in noisy environments. The function of Change-of-State (COS) interrupt is featured. The COS function means when any of these digital inputs changes the state, an interrupt will consequently rise for user to handle this external event.

Another useful feature is the design of Board ID, which is convenient for users to identify a specified card by means of setup of the switch jumper when users have two or more cPCI-7452 cards in one system.

The I/O signals are linked outward via two 100-pin stacking Mini-SCSI connectors for multi-channel instrument loads.

1.1 Features

The cPCI-7452 Isolated Digital I/O Card provides the following advanced features:

- 32-bit CompactPCI Bus, Plug and Play
- High-density 128 opto-isolated digital outputs
- High-density 128 opto-isolated digital inputs
- Programmable Change-of-State (COS) detection for all digital input channels
- Watch-Dog-Timer counter to prevent system from crashing
- Dry contact input available
- Board ID

1.2 Applications

- Machine automation
- Industrial ON/OFF control
- External relay driving
- Signal switching
- Laboratory automation

1.3 Specifications

Optical Isolated Digital Input

Input channels	128
Input type	Opto-isolated
Input Device	PC-3H4
Maximum input range	24V, non-polarity
Digital logic levels	0-24V, non-polarity Input high voltage: 5-24V Input low voltage: 0-2V
Input resistance	2.4kΩ @ 0.5W
Allowed Input Current	10 mA per channel (typical) 50 mA per channel (max)
Isolated Voltage	$2{,}500V_{RMS}$
Input Protection	ESD protection circuit (Forward direction)
Interrupt Source	Change-of-State for 128 lines Watch-Dog-Timer Counter
Data Transfers:	Programmed I/O

Table 1-1: Digital Input Specifications

Optical Isolated Digital Output

Output channels	128
Output type	Open collector Darlington transistor
Output Device	ULN2803A (common ground)
Sink Current	500mA for one channel @ 100% duty 500mA per line if all channels @ 20% duty
Isolated Voltage	$2{,}500V_{RMS}$
Data Transfers:	Programmed I/O

Table 1-2: Digital Output Specifications

Isolated +5V Power Supply

Output Voltage	+5V

Table 1-3: Power Supply Specifications

Output Current	150mA max. @ 40°C

Table 1-3: Power Supply Specifications

General Specifications

Dimension	233.35mm (L) x 160mm (W), standard CompactPCI 6U size
Bus	32-bit CompactPCI bus
Operating temperature	0°C - 60°C
Storage temperature	-20°C - 80°C
Humidity	5 to 95% non-condensing
I/O connector	100-pin Dual port SCSI-II female x 2 (HDRA-E100W1LFDT1EC-SL) Stacking arrangements

Table 1-4: General Specifications

Power Consumption

cPCI-7452	3.3V, 300mA; 5V, 1258 mA (typical)

Table 1-5: Power Consumption

1.4 Software Support

ADLINK provides versatile software drivers and packages to address different approaches to building a system. We not only provide programming libraries such as DLLs for many Windows systems, but also provide drivers for many software packages such as LabVIEW®. All software options are included in the ADLINK CD.

Programming Library

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

- DOS Library: For 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: Included device drivers and DLL for Windows 98/NT/2000/XP. A DLL is a binary compatible across Windows 98/NT/2000/XP. That means all applications developed with PCIS-DASK are compatible across Windows 98/NT/2000/XP. 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\Software Package\PCIS-DASK

The above software drivers are shipped with the board. Please refer to the "Software Installation Guide" for installation procedures.

DAQ-LVIEW PnP: LabVIEW® Driver

DAQ-LVIEW PnP contains VIs that are used to interface with the LabVIEW® software package. DAQ-LVIEW PnP supports Windows 95/98/NT/2000/XP. The LabVIEW® drivers are shipped free with the board. You can install and use them without a license. For more information about DAQ-LVIEW PnP, please refer to the user's guide in the CD.

DAQBenchTM: ActiveX Controls

We suggest customers who are familiar with ActiveX controls and VB/VC++ programming use the DAQBench^{TM} ActiveX Control component library for developing applications. The DAQBenchTM is designed under Windows NT/98. For more information about DAQBench, please refer to the user's guide in the CD.

2 Installation

This chapter describes how to install and setup the cPCI-7452. Please follow these instructions carefully.

2.1 Unpacking Checklist

Check the shipping carton for any damage. If the shipping carton and contents are damaged, notify the dealer for a replacement. Retain the shipping carton and packing materials for inspection by the dealer. Obtain authorization before returning any product to ADLINK.

Check the following items are included in the package, if there are any items missing, please contact your dealer:

Included Items

- cPCI-7452 isolated digital input and isolated digital output cards
- This User's Manual

Note: The packaging of OEM versions with non-standard configuration, functionality, or package may vary according to different configuration requests.

CAUTION: The boards must be protected from static discharge and physical shock. Never remove any of the socketed parts except at a static-free workstation. Use the anti-static bag shipped with the product to handle the board. Wear a grounded wrist strap when servicing

2.2 PCB Layout

The location of connector, switch and jumpers are shown in the figure below and are described in the following sections.

cPCI-7452 PCB Layout

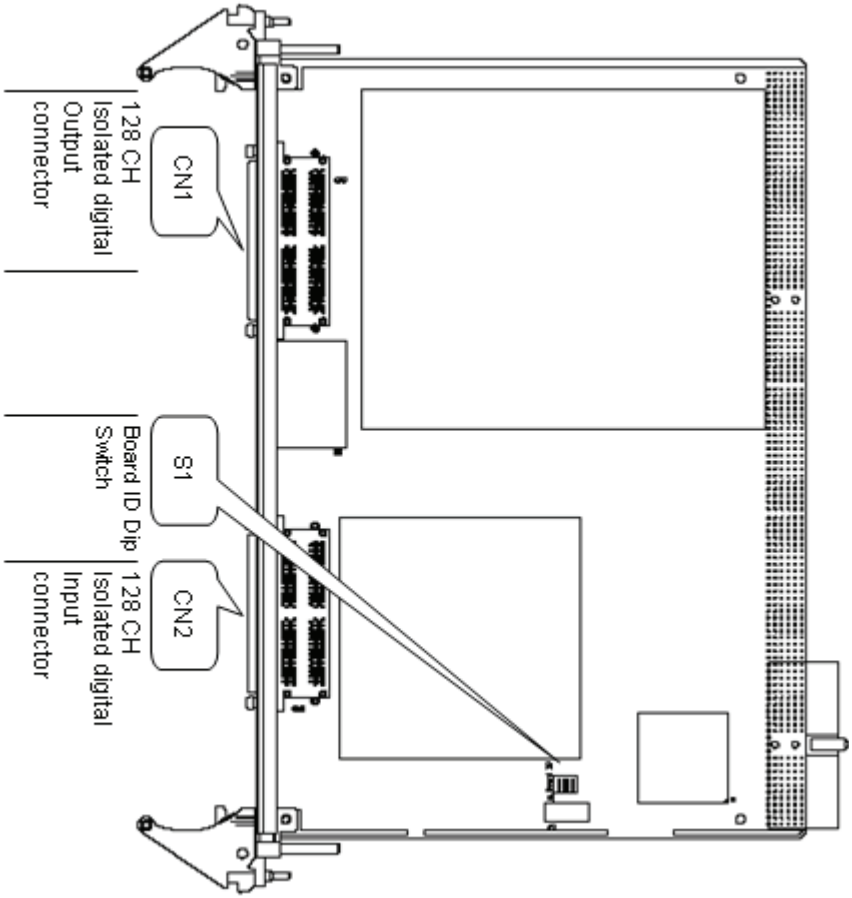


Figure 2-1: cPCI-7452 PCB Layout

Front Panel View

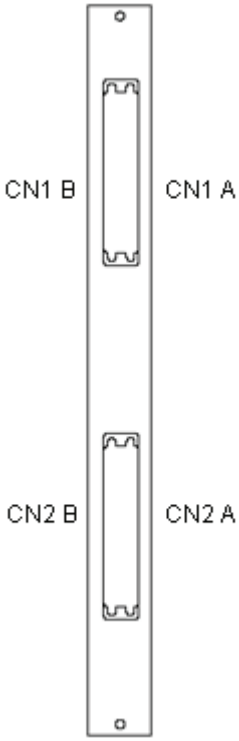
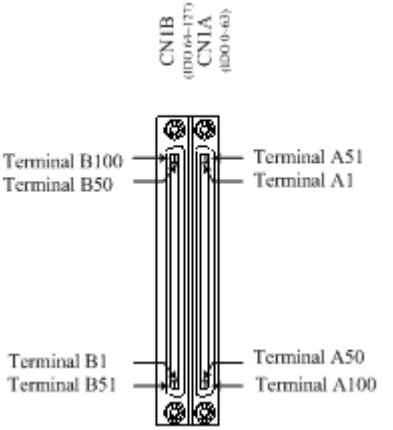


Figure 2-2: cPCI-7452 Front Panel View

2.3 DO Connector Pin Assignments (CN1)

N/C	100	50	N/C
IGND	99	49	IGND
IGND	98	48	IGND
IGND	97	47	IGND
VDD	96	46	VDD
IDO_127	95	45	IDO_119
IDO_126	94	44	IDO_118
IDO_125	93	43	IDO_117
IDO_124	92	42	IDO_116
IDO_123	91	41	IDO_115
IDO_122	90	40	IDO_114
IDO_121	89	39	IDO_113
IDO_120	88	38	IDO_112
IGND	87	37	IGND
IGND	86	36	IGND
IGND	85	35	IGND
VDD	84	34	VDD
IDO_111	83	33	IDO_103
IDO_110	82	32	IDO_102
IDO_109	81	31	IDO_101
IDO_108	80	30	IDO_100
IDO_107	79	29	IDO_99
IDO_106	78	28	IDO_98
IDO_105	77	27	IDO_97
IDO_104	76	26	IDO_96
N/C	75	25	N/C
IGND	74	24	IGND
IGND	73	23	IGND
IGND	72	22	IGND
VDD	71	21	VDD
IDO_95	70	20	IDO_87
IDO_94	69	19	IDO_86
IDO_93	68	18	IDO_85
IDO_92	67	17	IDO_84
IDO_91	66	16	IDO_83
IDO_90	65	15	IDO_82
IDO_89	64	14	IDO_81
IDO_88	63	13	IDO_80
IGND	62	12	IGND
IGND	61	11	IGND
IGND	60	10	IGND
VDD	59	9	VDD
IDO_79	58	8	IDO_71
IDO_78	57	7	IDO_70
IDO_77	56	6	IDO_69
IDO_76	55	5	IDO_68
IDO_75	54	4	IDO_67
IDO_74	53	3	IDO_66
IDO_73	52	2	IDO_65
IDO_72	51	1	IDO_64



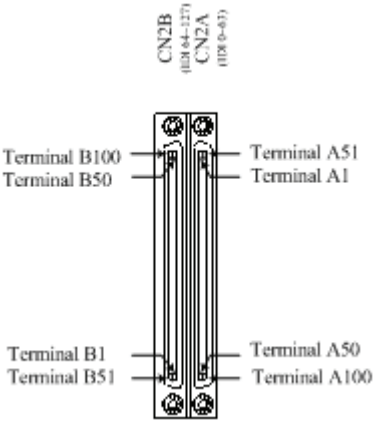
IDO_0	1	51	IDO_8
IDO_1	2	52	IDO_9
IDO_2	3	53	IDO_10
IDO_3	4	54	IDO_11
IDO_4	5	55	IDO_12
IDO_5	6	56	IDO_13
IDO_6	7	57	IDO_14
IDO_7	8	58	IDO_15
VDD	9	59	VDD
IGND	10	60	IGND
IGND	11	61	IGND
IGND	12	62	IGND
IDO_16	13	63	IDO_24
IDO_17	14	64	IDO_25
IDO_18	15	65	IDO_26
IDO_19	16	66	IDO_27
IDO_20	17	67	IDO_28
IDO_21	18	68	IDO_29
IDO_22	19	69	IDO_30
IDO_23	20	70	IDO_31
VDD	21	71	VDD
IGND	22	72	IGND
IGND	23	73	IGND
IGND	24	74	IGND
N/C	25	75	N/C
IDO_32	26	76	IDO_40
IDO_33	27	77	IDO_41
IDO_34	28	78	IDO_42
IDO_35	29	79	IDO_43
IDO_36	30	80	IDO_44
IDO_37	31	81	IDO_45
IDO_38	32	82	IDO_46
IDO_39	33	83	IDO_47
VDD	34	84	VDD
IGND	35	85	IGND
IGND	36	86	IGND
IGND	37	87	IGND
IDO_48	38	88	IDO_56
IDO_49	39	89	IDO_57
IDO_50	40	90	IDO_58
IDO_51	41	91	IDO_59
IDO_52	42	92	IDO_60
IDO_53	43	93	IDO_61
IDO_54	44	94	IDO_62
IDO_55	45	95	IDO_63
VDD	46	96	VDD
IGND	47	97	IGND
IGND	48	98	IGND
IGND	49	99	IGND
V5V	50	100	V5V

Legend	
IDO_n	Isolated digital output channel n
VDD	Common pin for isolated output channel (Note: All outputs share the same power supply)
IGND	Ground return path for isolated output channels
V5V	Onboard un-regulated 5V power supply output
N/C	Not used

Table 2-1: cPCI-7452 CN1 Pin Assignment

2.4 DI Connector Pin Assignments (CN2)

N/C	100	50	N/C
COM16	99	49	COM15
COM16	98	48	COM15
COM16	97	47	COM15
COM16	96	46	COM15
IDI_127	95	45	IDI_119
IDI_126	94	44	IDI_118
IDI_125	93	43	IDI_117
IDI_124	92	42	IDI_116
IDI_123	91	41	IDI_115
IDI_122	90	40	IDI_114
IDI_121	89	39	IDI_113
IDI_120	88	38	IDI_112
COM14	87	37	COM13
COM14	86	36	COM13
COM14	85	35	COM13
COM14	84	34	COM13
IDI_111	83	33	IDI_103
IDI_110	82	32	IDI_102
IDI_109	81	31	IDI_101
IDI_108	80	30	IDI_100
IDI_107	79	29	IDI_99
IDI_106	78	28	IDI_98
IDI_105	77	27	IDI_97
IDI_104	76	26	IDI_96
N/C	75	25	N/C
COM12	74	24	COM11
COM12	73	23	COM11
COM12	72	22	COM11
COM12	71	21	COM11
IDI_95	70	20	IDI_87
IDI_94	69	19	IDI_86
IDI_93	68	18	IDI_85
IDI_92	67	17	IDI_84
IDI_91	66	16	IDI_83
IDI_90	65	15	IDI_82
IDI_89	64	14	IDI_81
IDI_88	63	13	IDI_80
COM10	62	12	COM9
COM10	61	11	COM9
COM10	60	10	COM9
COM10	59	9	COM9
IDI_79	58	8	IDI_71
IDI_78	57	7	IDI_70
IDI_77	56	6	IDI_69
IDI_76	55	5	IDI_68
IDI_75	54	4	IDI_67
IDI_74	53	3	IDI_66
IDI_73	52	2	IDI_65
IDI_72	51	1	IDI_64



IDI_0	1	51	IDI_8
IDI_1	2	52	IDI_9
IDI_2	3	53	IDI_10
IDI_3	4	54	IDI_11
IDI_4	5	55	IDI_12
IDI_5	6	56	IDI_13
IDI_6	7	57	IDI_14
IDI_7	8	58	IDI_15
COM1	9	59	COM2
COM1	10	60	COM2
COM1	11	61	COM2
COM1	12	62	COM2
IDI_16	13	63	IDI_24
IDI_17	14	64	IDI_25
IDI_18	15	65	IDI_26
IDI_19	16	66	IDI_27
IDI_20	17	67	IDI_28
IDI_21	18	68	IDI_29
IDI_22	19	69	IDI_30
IDI_23	20	70	IDI_31
COM3	21	71	COM4
COM3	22	72	COM4
COM3	23	73	COM4
COM3	24	74	COM4
N/C	25	75	N/C
IDI_32	26	76	IDI_40
IDI_33	27	77	IDI_41
IDI_34	28	78	IDI_42
IDI_35	29	79	IDI_43
IDI_36	30	80	IDI_44
IDI_37	31	81	IDI_45
IDI_38	32	82	IDI_46
IDI_39	33	83	IDI_47
COM5	34	84	COM6
COM5	35	85	COM6
COM5	36	86	COM6
COM5	37	87	COM6
IDI_48	38	88	IDI_56
IDI_49	39	89	IDI_57
IDI_50	40	90	IDI_58
IDI_51	41	91	IDI_59
IDI_52	42	92	IDI_60
IDI_53	43	93	IDI_61
IDI_54	44	94	IDI_62
IDI_55	45	95	IDI_63
COM7	46	96	COM8
COM7	47	97	COM8
COM7	48	98	COM8
COM7	49	99	COM8
N/C	50	100	N/C

Legend	
IDI_n	Isolated digital input channel n
COM1	common junction for input channel 0-7
COM2	common junction for input channel 8-15
COM3	common junction for input channel 16-23
COM4	common junction for input channel 24-31
COM5	common junction for input channel 32-39
COM6	common junction for input channel 40-47
COM7	common junction for input channel 48-55
COM8	common junction for input channel 56-63
COM9	common junction for input channel 64-71
COM10	common junction for input channel 72-79
COM11	common junction for input channel 80-87
COM12	common junction for input channel 88-95
COM13	common junction for input channel 96-103
COM14	common junction for input channel 104-111
COM15	common junction for input channel 112-119
COM16	common junction for input channel 120-127
N/C	Not used

Table 2-2: cPCI-7452 CN2 Pin Assignment

2.5 Board ID (S1)

When users plug two or more data acquisition cards in one system, it can take lots of efforts to identify one specific card. For easier identification, cPCI-7452 provides the Board ID function. According to a DIP switch configuration located in S1, users can assign a specific board ID to a specially designated card and access the card correctly simply through software programming. For more details about Board ID in programming, please refer to Chapter 5.

The table below shows all of the switch setting conditions. '1' means DIP is at "ON" position; '0' means DIP is at "OFF"

Figure 2-3: Board ID Settings

Board ID	Switch No.			
	1	2	3	4
0	1	1	1	1
1	0	1	1	1
2	1	0	1	1
3	0	0	1	1
4	1	1	0	1
5	0	1	0	1
6	1	0	0	1
7	0	0	0	1
8	1	1	1	0
9	0	1	1	0
10	1	0	1	0
11	0	0	1	0
12	1	1	0	0
13	0	1	0	0
14	1	0	0	0
15	0	0	0	0

Table 2-3: Board ID Setting Conditions

Note: 1 = ON, 0 = OFF

3 Register Format

The detailed descriptions of the register format are specified in this chapter. This information must be quite useful and helpful for the programmers who want to handle the cPCI-7452 by low-level programming. We suggest that users understand more about the PCI interface before starting low-level programming.

3.1 I/O Address Map

The cPCI-7452 registers are all 32-bit wide. Therefore, users can access these registers only by 32 bits I/O instructions. The control of the isolated digital outputs and isolated digital inputs is by accessing registers mentioned in this chapter. The following table outlines the register map, including descriptions and offset addresses corresponding to the base address.

3.2 I/O Registers Map

Offset	Write	Read
0x00h	Bank0 Isolated DO	Bank0 DO Read Back
0x04h	WDT/Bank0 COS Interrupt Control	Interrupt Status/Bank0 COS Interrupt Control Read Back/WDT Interrupt Control Read Back
0x08h	Bank0 COS Setup	Bank0 COS Latch
0x0Ch	---	Bank0 Isolated DI
0x10h	WDTimer Load Config	---
0x40h	Bank1 Isolated DO	Bank1 DO Read Back
0x44h	Bank1 COS Interrupt Control	Bank1 COS Interrupt Control Read Back
0x48h	Bank1 COS Setup	Bank1 COS Latch
0x4Ch	---	Bank1 Isolated DI
0x50h	---	---
0x80h	Bank2 Isolated DO	Bank2 DO Read Back
0x84h	Bank2 COS Interrupt Control	Bank2 COS Interrupt Control Read Back
0x88h	Bank2 COS Setup	Bank2 COS Latch
0x8Ch	---	Bank2 Isolated DI
0x90h	---	---
0xC0h	Bank3 Isolated DO	Bank3 DO Read Back
0xC4h	Bank3 COS Interrupt Control	Bank3 COS Interrupt Control Read Back
0xC8h	Bank3 COS Setup	Bank3 COS Latch
0xCCh	---	Bank3 Isolated DI
0xD0h	---	---

Table 3-1: cPCI-7452 Register Map

3.3 Isolated DO Register Bankn, n = 0 - 3

There are 128 isolated digital outputs on each cPCI-7452 board. These lines are divided between two output connectors, CN1A and CN1B. They are controlled by four 32-bit registers. Each 32-bit registers corresponding to 32 lines is called a bank.

Each digital output line is controlled by each bit of the four control registers. The setting “1” means that the corresponding output line is ON. The setting “0” means that it is OFF.

- Bank0 Isolated DO Register
- Address: BASE + 0x00h
- Attribute: Write

7	6	5	4	3	2	1	0
DO[7..0]							
15	14	13	12	11	10	9	8
DO[15..8]							
23	22	21	20	19	18	17	16
DO[23..16]							
31	30	29	28	27	26	25	24
DO[31..24]							

- Bank1 Isolated DO Register
- Address: BASE + 0x40h
- Attribute: Write

7	6	5	4	3	2	1	0
DO[39..32]							
15	14	13	12	11	10	9	8
DO[47..40]							
23	22	21	20	19	18	17	16
DO[55..48]							
31	30	29	28	27	26	25	24
DO[63..56]							

- Bank2 Isolated DO Register
- Address: BASE + 0x80h
- Attribute: Write

7	6	5	4	3	2	1	0
DO[71..64]							
15	14	13	12	11	10	9	8
DO[79..72]							
23	22	21	20	19	18	17	16
DO[87..80]							
31	30	29	28	27	26	25	24
DO[95..88]							

- Bank3 Isolated DO Register
- Address: BASE + 0xC0h
- Attribute: Write

7	6	5	4	3	2	1	0
DO[103..96]							
15	14	13	12	11	10	9	8
DO[111..104]							
23	22	21	20	19	18	17	16
DO[119..112]							
31	30	29	28	27	26	25	24
DO[127..120]							

- DOn: isolated digital output data of a certain output line n, n = 0-127
- 1: output channel is ON
- 0: ouput channel is OFF

3.4 DO Read Back Register Bankn, n = 0 - 3

The isolated DO statuses from Bank0 to Bank3 can be readback from the readback register. If the output line is ON, the corresponding bit value is 1. If the output line is OFF, the corresponding bit value is 0.

- Bank0 DO Read Back Register
- Address: BASE + 0x00h
- Attribute: Read

7	6	5	4	3	2	1	0
RBK[7..0]							
15	14	13	12	11	10	9	8
RBK[15..8]							
23	22	21	20	19	18	17	16
RBK[23..16]							
31	30	29	28	27	26	25	24
RBK[31..24]							

- Bank1 DO Read Back Register
- Address: BASE + 0x40h
- Attribute: Read

7	6	5	4	3	2	1	0
RBK[39..32]							
15	14	13	12	11	10	9	8
RBK[47..40]							
23	22	21	20	19	18	17	16
RBK[55..48]							
31	30	29	28	27	26	25	24
RBK[63..56]							

- Bank2 DO Read Back Register
- Address: BASE + 0x80h
- Attribute: Read

7	6	5	4	3	2	1	0
RBK[71..64]							
15	14	13	12	11	10	9	8
RBK[79..72]							
23	22	21	20	19	18	17	16
RBK[87..80]							
31	30	29	28	27	26	25	24
RBK[95..88]							

- Bank3 DO Read Back Register
- Address: BASE + 0xC0h
- Attribute: Read

7	6	5	4	3	2	1	0
RBK[103..96]							
15	14	13	12	11	10	9	8
RBK[111..104]							
23	22	21	20	19	18	17	16
RBK[119..112]							
31	30	29	28	27	26	25	24
RBK[127..120]							

- RBKn: Read back data of output line n, n = 0-127
- 1: output channel is ON
- 0: ouput channel is OFF

3.5 Bank0, Bank1, Bank2, Bank3 COS Interrupt, and WDT Control Registers

There are two different interrupt modes in cPCI-7452. In default, both interrupt modes are disabled. Users can write registers mentioned later to enable them. In the first mode, users enable the COS (Change of State) interrupt function to monitor the statuses of enabled input channels and whenever the statuses change from 0 to 1 or 1 to 0. In the second mode, users can enable Watch-Dog-Timer (WDT) Counter and let it count down. The interrupt assertes when Watch-DogTimer Counter counts to zero.

After processing the interrupt request event, users have to clear the interrupt request in order to handle another interrupt request. Noteworthy is the fact that it takes time for a system to clear the interrupt. That is, any COS interrupt or WDT interrupt comes before the previous interrupt is still not cleared is neglected. To clear the interrupt request, write 1 to the corresponding bit.

The COS interrupt is enabled by four registers (WDT/Bank0 COS Interrupt Control Reg, Bank1 COS Interrupt Control Reg, Bank2 COS Interrupt Control Reg, Bank3 COS Interrupt Control Reg). Because the 128 digital inputs are divided among four 32-bit onboard buses, every 32 inputs is connected to a CPLD. When users enable Bank0 COS interrupt, the first CPLD (CPLD0) will produce interrupt signal while the first 32-bit inputs DI[31..0] have change of state. When users enable Bank1 COS interrupt, the second CPLD (CPLD1) will produce interrupt signal while the second 32-bit inputs DI[63..32] have change of state. When users enable Bank2 COS interrupt, the third CPLD (CPLD2) will produce interrupt signal while the thrid 32-bit inputs DI[95..64] have change of state. When users enable Bank3 COS interrupt, the fourth CPLD (CPLD3) will produce interrupt signal while the fourth 32-bit inputs DI[127..96] have change of state.

- WDT/Bank0 COS Interrupt Control Register
- Address: BASE + 0x04h
- Attribute: Write

7	6	5	4	3	2	1	0
---						WDT CLR	COS0 CLR
15	14	13	12	11	10	9	8
---						WDT Int_EN	COS0 Int_EN
23	22	21	20	19	18	17	16

31	30	29	28	27	26	25	24

- COS0 CLR (bit 0): write 1 to clear the Bank0 COS interrupt.
  - 1: clear the Bank0 COS interrupt
  - 0: no effect
- WDT CLR (bit 1): write 1 to clear WDT interrupt.
  - 1: clear WDT interrupt
  - 0: no effect
- COS0 Int_EN (bit 8): Write Bank0 DI[31..0] Change-of-State interrupt enable control
  - 1: enable
  - 0: disable
- WDT Int_EN (bit 9): Write WDTimer counter/interrupt enable control
  - 1: enable
  - 0: disable

- Bank1 COS Interrupt Control Register
- Address: BASE + 0x44h
- Attribute: Write

7	6	5	4	3	2	1	0
---							COS1 CLR
15	14	13	12	11	10	9	8
---							COS1 Int_EN
23	22	21	20	19	18	17	16

31	30	29	28	27	26	25	24

- COS1 CLR (bit 0): write 1 to clear the Bank1 COS interrupt.
  - 1: clear the Bank1 COS interrupt
  - 0: no effect
- COS1 Int_EN (bit 8): Write Bank1 DI[63..32] Change-of-State interrupt enable control
  - 1: enable
  - 0: disa

- Bank2 COS Interrupt Control Register
- Address: BASE + 0x84h
- Attribute: Write

7	6	5	4	3	2	1	0
---							COS2 CLR
15	14	13	12	11	10	9	8
---							COS2 Int_EN
23	22	21	20	19	18	17	16

31	30	29	28	27	26	25	24

- COS2 CLR (bit 0): write 1 to clear the Bank2 COS interrupt.
  - 1: clear the Bank2 COS interrupt
  - 0: no effect
- COS2 Int_EN (bit 8): Write Bank2 DI[95..64] Change-of-State interrupt enable control
  - 1: enable
  - 0: disable

- Bank3 COS Interrupt Control Register
- Address: BASE + 0xC4h
- Attribute: Write

7	6	5	4	3	2	1	0
---							COS3 CLR
15	14	13	12	11	10	9	8
---							COS3 Int_EN
23	22	21	20	19	18	17	16

31	30	29	28	27	26	25	24

- COS3 CLR (bit 0): write 1 to clear the Bank3 COS interrupt.
  - 1: clear the Bank3 COS interrupt
  - 0: no effect
- COS3 Int_EN (bit 8): Write Bank3 DI[127..96] Change-of-State interrupt enable control
  - 1: enable
  - 0: disable

3.6 Interrupt Status/Bank0 COS Interrupt Control Read Back/WDT Interrupt Control Read Back

When interrupt occurs, this register provides information for users to recognize the interrupt status and the interrupt setup condition read back.

- Address: BASE + 0x04h
- Attribute: Read

7	6	5	4	3	2	1	0
			WDT Int_Stat	COS3 Int_Stat	COS2 Int_Stat	COS1 Int_Stat	COS0 Int_Stat
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
						WDT Int_EN_Stat	COS0 Int_EN_Stat
31	30	29	28	27	26	25	24

- COS0 Int_Stat (bit 0): Bank0 DI[31..0] COS interrupt status register
  - 0: Bank0 COS interrupt de-asserts
  - 1: Bank0 COS interrupt asserts
- COS1 Int_Stat (bit 1): Bank1 DI[63..32] COS interrupt status register
  - 0: Bank1 COS interrupt de-asserts
  - 1: Bank1 COS interrupt asserts
- COS2 Int_Stat (bit 2): Bank2 DI[95..64] COS interrupt status register
  - 0: Bank2 COS interrupt de-asserts
  - 1: Bank2 COS interrupt asserts
- COS3 Int_Stat (bit 3): Bank3 DI[127..96] COS interrupt status register
  - 0: Bank3 COS interrupt de-asserts
  - 1: Bank3 COS interrupt asserts
- WDT Int_Stat (bit 4): Watch Dog Timer interrupt status
  - 0: WDTimer interrupt de-asserts
  - 1: WDTimer interrupt asserts

- COS0 Int_EN_Stat (bit 16): Bank0 DI[31..0] COS interrupt enable status register
  - 0: Bank0 COS interrupt disabled
  - 1: Bank0 COS interrupt enabled
- WDT Int_EN_Stat (bit 17): Watch Dog Timer interrupt enable status
  - 0: WDT interrupt disabled
  - 1: WDT interrupt enabled

3.7 COS Interrupt Control Read Back Register Bankn, n = 1 - 3

When any COS interrupt of Bank1 - Bank3 occurs, these registers provide information for users to recognize the interrupt setup condition read back.

- Bank1 COS Interrupt Control Read Back Register
- Address: BASE + 0x44h
- Attribute: Read

7	6	5	4	3	2	1	0

15	14	13	12	11	10	9	8

23	22	21	20	19	18	17	16
---							COS1 Int_EN_Stat
31	30	29	28	27	26	25	24

- COS1 Int_EN_Stat (bit 16): Bank1 DI[63..32] COS interrupt enable status register
  - 0: Bank1 COS interrupt disabled
  - 1: Bank1 COS interrupt enabled

- Bank2 COS Interrupt Control Read Back Register
- Address: BASE + 0x84h
- Attribute: Read

7	6	5	4	3	2	1	0

15	14	13	12	11	10	9	8

23	22	21	20	19	18	17	16
---							COS2 Int_EN_Stat
31	30	29	28	27	26	25	24

- COS2 Int_EN_Stat (bit 16): Bank2 DI[95..64] COS interrupt enable status register
  - 0: Bank2 COS interrupt disabled
  - 1: Bank2 COS interrupt enabled

- Bank3 COS Interrupt Control Read Back Register
- Address: BASE + 0xC4h
- Attribute: Read

7	6	5	4	3	2	1	0

15	14	13	12	11	10	9	8

23	22	21	20	19	18	17	16
---							COS3 Int_EN_Stat
31	30	29	28	27	26	25	24

- COS3 Int_EN_Stat (bit 16): Bank3 DI[127..96] COS interrupt enable status register
  - 0: Bank3 COS interrupt disabled
  - 1: Bank3 COS interrupt enabled

3.8 COS Setup Register Bankn, n = 0 - 3

The cPCI-7452 provides a Change-of-State(COS) interrupt function on any one of digital input channel. This function allows users to monitor the status of input channels. Because these digital input channels are divided among four 32-bits banks. Users monitor the digital input channels by setting four Bank COS Setup registers. By enabling the COS Setup registers, it will generate an interrupt when the corresponding channnel changes its state. For more detailed information, please refer to Chapter 4.

- Bank 0 COS Setup Register
- Address: BASE + 0x08h
- Attribute: Write

7	6	5	4	3	2	1	0
COS SET[7..0]							
15	14	13	12	11	10	9	8
COS SET[15..8]							
23	22	21	20	19	18	17	16
COS SET[23..16]							
31	30	29	28	27	26	25	24
COS SET[31..24]							

- Bank 1 COS Setup Register
- Address: BASE + 0x48h
- Attribute: Write

7	6	5	4	3	2	1	0
COS SET[39..32]							
15	14	13	12	11	10	9	8
COS SET[47..40]							
23	22	21	20	19	18	17	16
COS SET[55..48]							
31	30	29	28	27	26	25	24
COS SET[63..56]							

- Bank 2 COS Setup Register
- Address: BASE + 0x88h
- Attribute: Write

7	6	5	4	3	2	1	0
COS SET[71..64]							
15	14	13	12	11	10	9	8
COS SET[79..72]							
23	22	21	20	19	18	17	16
COS SET[87..80]							
31	30	29	28	27	26	25	24
COS SET[95..88]							

- Bank 3 COS Setup Register
- Address: BASE + 0xC8h
- Attribute: Write

7	6	5	4	3	2	1	0
COS SET[103..96]							
15	14	13	12	11	10	9	8
COS SET[111..104]							
23	22	21	20	19	18	17	16
COS SET[119..112]							
31	30	29	28	27	26	25	24
COS SET[127..120]							

- COS SETn: change-of-state setup of DI channel n, n=0 - 127
  - 1: enable monitoring channel n COS interrupt
  - 0: disable monitoring channel n COS interrupt

3.9 COS Latch Register Bankn, n = 0 - 3

When COS occurs, the Bank0 - Bank3 COS Latch registers will also latch the DI[31..0], DI[63..32], DI[95..64], DI[127..96] data respectively. Once the users clear the interrupt request, the COS Latch register will be also cleared automatically. Because users can simply read these registers to know the statuses after interrupts, these registers can release the CPU from overwhelming burden on polling all of the inputs always and enable the computer to handle more tasks.

- Bank 0 COS Latch Register
- Address: BASE + 0x08h
- Attribute: Read

7	6	5	4	3	2	1	0
CL[7..0]							
15	14	13	12	11	10	9	8
CL[15..8]							
23	22	21	20	19	18	17	16
CL[23..16]							
31	30	29	28	27	26	25	24
CL[31..24]							

- Bank 1 COS Latch Register
- Address: BASE + 0x48h
- Attribute: Read

7	6	5	4	3	2	1	0
CL[39..32]							
15	14	13	12	11	10	9	8
CL[47..40]							
23	22	21	20	19	18	17	16
CL[55..48]							
31	30	29	28	27	26	25	24
CL[63..56]							

- Bank 2 COS Latch Register
- Address: BASE + 0x88h
- Attribute: Read

7	6	5	4	3	2	1	0
CL[71..64]							
15	14	13	12	11	10	9	8
CL[79..72]							
23	22	21	20	19	18	17	16
CL[87..80]							
31	30	29	28	27	26	25	24
CL[95..88]							

- Bank 3 COS Latch Register
- Address: BASE + 0xC8h
- Attribute: Read

7	6	5	4	3	2	1	0
CL[103..96]							
15	14	13	12	11	10	9	8
CL[111..104]							
23	22	21	20	19	18	17	16
CL[119..112]							
31	30	29	28	27	26	25	24
CL[127..120]							

- CL x: COS latch register of DI channel n, n = 0 - 127
  - 1: digital input is ON
  - 0: digital input is OFF

3.10 Isolated Digital Input Register Bankn, n = 0 - 3

There are 128 isolated inputs on a cPCI-7452 card. The statuses of the 128 lines can be read from the four isolated input registers. Each bit corresponds to each channel. The bit value 0 means that the input is ON and 1 menas that the input is OFF.

- Bank0 Isolated Digital Input Register
- Address: BASE + 0x0Ch
- Attribute: Read

7	6	5	4	3	2	1	0
DI[7..0]							
15	14	13	12	11	10	9	8
DI[15..8]							
23	22	21	20	19	18	17	16
DI[23..16]							
31	30	29	28	27	26	25	24
DI[31..24]							

- Bank1 Isolated Digital Input Register
- Address: BASE + 0x4Ch
- Attribute: Read

7	6	5	4	3	2	1	0
DI[39..32]							
15	14	13	12	11	10	9	8
DI[47..40]							
23	22	21	20	19	18	17	16
DI[55..48]							
31	30	29	28	27	26	25	24
DI[63..56]							

- Bank2 Isolated Digital Input Register
- Address: BASE + 0x8Ch
- Attribute: Read

7	6	5	4	3	2	1	0
DI[71..64]							
15	14	13	12	11	10	9	8
DI[79..72]							
23	22	21	20	19	18	17	16
DI[87..80]							
31	30	29	28	27	26	25	24
DI[95..88]							

- Bank3 Isolated Digital Input Register
- Address: BASE + 0xCCh
- Attribute: Read

7	6	5	4	3	2	1	0
DI[103..96]							
15	14	13	12	11	10	9	8
DI[111..104]							
23	22	21	20	19	18	17	16
DI[119..112]							
31	30	29	28	27	26	25	24
DI[127..120]							

- DIn: isolated digiatal input channel n, n = 0 - 127
  - 1: input voltage is ON
  - 0: input voltage is OFF

3.11 Watch-Dog-Timer Counter Load Config Register

Watch-Dog-Timer Counter loads the value written in the the register. The corresponding hexadecimal value setted by user determines the overflow time of Watch-Dog-Timer Counter. The overflow time is calculated by the value that users set multiplied 1.056 μs. The timer interval is from 0 to 4535.5 seconds.

- Address: BASE + 0x10h
- Attribute: Write

7	6	5	4	3	2	1	0
WDT Cnt[7..0]							
15	14	13	12	11	10	9	8
WDT Cnt[15..8]							
23	22	21	20	19	18	17	16
WDT Cnt[23..16]							
31	30	29	28	27	26	25	24
WDT Cnt[31..24]							

- WDT Cntn: Counter value determined by each binary bit n, n = 0 – 31
- Ex: If users write 1 to each bit, then the maximal time interval is obtained.

3.12 Handling PCI Controller Registers

The PCI bus controller adopted in cPCI-7452 is PCI-9030 which is provided by PLX technology Inc. When users attempt to handle low-level programming, some registers in PCI-9030 should be noticed. The interrupt control register(INTCSR; 0x4Ch) of PCI-9030 takes charge of all interrupt information from local bus to PCI bus. When users want to develop their own interrupt function driver, both interrupt registers in PCI-9030 and in cPCI-7452 have to work together. For more detailed information about the interrupt control register in PCI-9030, please refer to the PCI-9030 databook.

In the cPCI-7452 software funciton library, we provide simple and easy-to-use functions to handle the procedure of interrupt. Using these functions, users don't need to care about the interrupt register in PCI controller. We suggest users use these functions instead of developing interrupt functions by themselves. For more information about cPCI-7452 funciton library, please refer to Chapter 5.

4 Operation Theory

4.1 Watchdog Timer

In safety-critical applications, users can enable the watchdog timer function on cPCI-7452 to automatically generate interrupt signal, in case the operating system or cPCI-7452 crashes. To access this function, users must first configure the watchdog timer overflow counter by windows API. Generally, the trigger source would come from the onboard 32-bit watchdog timer.

The watchdog timer overflow interval can be programmed through API. After enabling the watchdog timer, users must periodically reset the timer by software command. If the timer is not being reset within the specified interval, the WDT module will generate an overflow signal and set the relay pattern to the one specified by users. This function is disabled in default.

The Watch Dog Timer function operation flow is listed as follows:

1. Load the WDTimer counter value through the function _7452_WDTimerSReload_Config(), this function allows users to set WDTimer overflow time from 1s to 4500 s.
2. Enable the WDTimer counter to count down through the function _7452_INT_Control(), this function allows WDT function in the CPLD (as Figure 4.3) starting to count down. After enabling the function, users must reload WDTimter by step (1) before it overflows. This makes cPCI-7452 work normally under WDT monitoring. When it overflows, CPLD interrupt will occurs.
3. When WDTimer interrupt occurs, users must reload the WDTimer through function _7452_WDTimerSReload_Config() to clear WDTimer carry out, and then clear the system interrupt through function _7452_CLR_IRQ(). After doing this, the WDTimer will count again unless user disable WDTimer counter.

4.2 Isolated Digital Input

The cPCI-7452 contains 128 opto-isolated digital input channels. The circuit diagram of the isolated input channel is shown below.

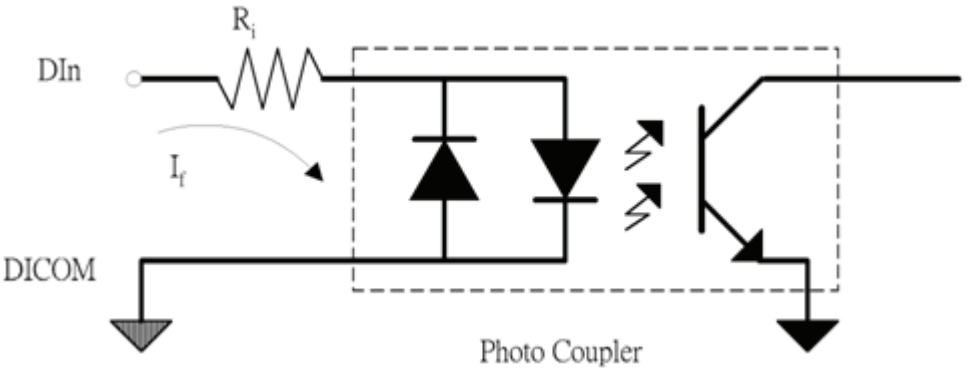


Figure 4-1: Photo Coupler

The digital input is first routed through a photo-coupler (PC3H4), so that the connection are not polarly sensitive whether using positive or negative voltage.

In addition, a first order-filter with time constant about 1.5ms is provided to filter high frequency noise. The normal input voltage range for high state is from 5 to 24V.

The cPCI-7452 provides an isolated +5V power for dry contact input. When the external circuit has no voltage source (e.g. a switch), users can use the on board +5V to respond the change of external circuit. The maximum output current of the on board isolated power is 170mA (@40°C). Please pay attention to the current consumption of the external circuit, which should not exceed the limitation. The dry contact architecture is shown below:

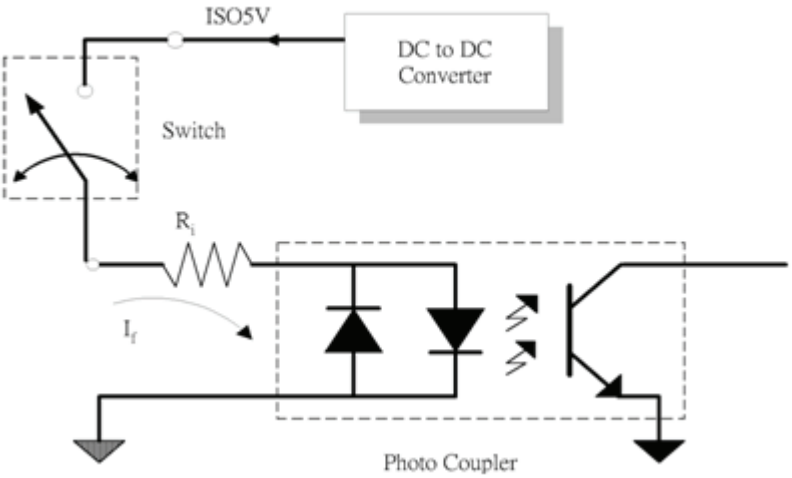


Figure 4-2: Dry Contact

4.3 Isolated Digital Output Channels

The common ground connection of isolated digital output is shown in the figure below. When the isolated digital output goes “ON”, the sink current will be conducted through the transistors. When the isolated digital output goes “OFF”, no current is conducted flow through the transistors. Please note that when the load is of an “inductance nature” such as a relay, coil or motor, the VDD pin must be connected to an external power source. The extra connection is utilized for the ‘fly-wheel diode’ to form a current-release closed loop, so that the transistors are protected from any high reverse voltage which can be generated by the inductance load when the output is switched from “ON” to “OFF”.

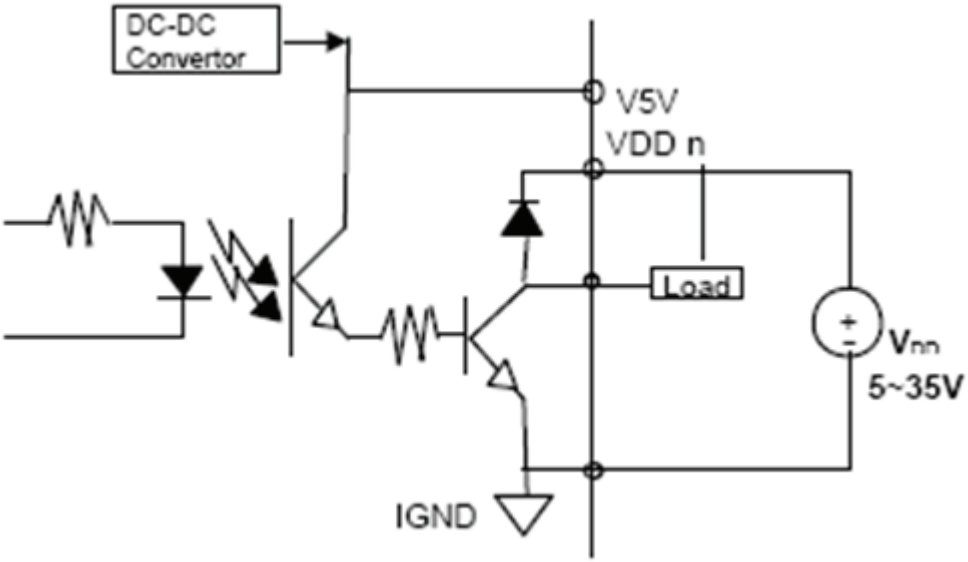


Figure 4-3: Common Ground Connection for Sink type Darlington ICs

4.4 Interrupt Architecture

cPCI-7452 has a powerful dual interrupt routing scheme including change-of-state detection and interrupt sources on watch-dog-timer. these interrupts well can make you handle more complicated information from outside enviroment and release your computer from a heavy burden in dealing with digital input data. Note that the dual interrupts do not mean the card occupies two IRQ levels.

4.5 Change of State(COS) Interrupt

What is COS?

The COS (Change of State) means either the input state(logic level) changes from low to high, or from high to low. The COS detection circuit will detect the edge of level change. In the cPCI-7452 card, the COS detection circuit is applied to all the input channels. When any channel changes its logic level, the COS detection circuit generates an interrupt request to PCI controller.

COS Detection

The following timing is an example of 8-CH COS operation. All of the enabled DI channels' signal level change will be detected to generate the interrupt request.

While the interrupt request generates, the corresponding DI data will also be latched into the COS latch register. In our COS architecture, the DI data are sampled by a 16.5MHz clock. It means the pulse width of the digital input have to last longer than 61ns, or the

COS latch register won't latch the correct input data. The COS latch register will be erased after clearing the interrupt request.

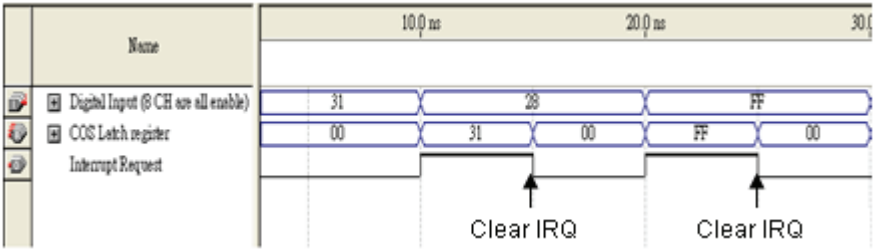
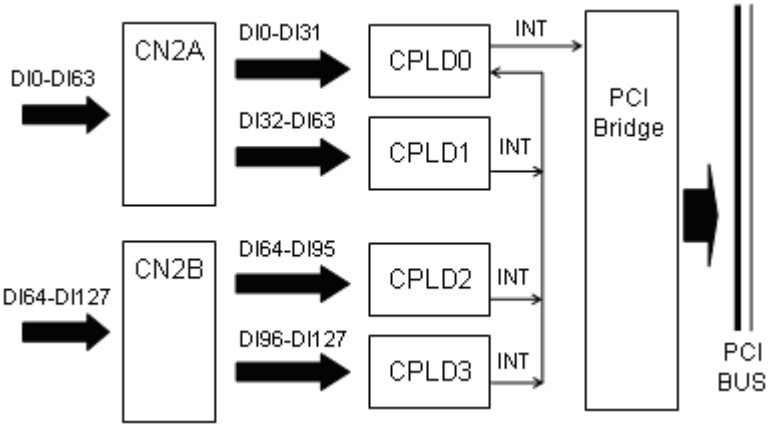


Figure 4-4: COS Timing

Architecture of COS detection

The COS interrupt system is used in cPCI-7452. COS interrupt occurs when the any of enabled DI line sense the status changes either from HIGH to LOW or from LOW to HIGH. The COS interrupt system can generate an interrupt request signal and the software can service this request with ISR. Note that there are four banks: bank 1 from DI0 to DI31, bank 2 from DI32 to 63, bank 3 from DI64 to DI95, bank 4 from DI 96 to 127. These banks are casacaded together toward the same IRQ line via CPLD. Users can use commands to know which bank or which DI line has COS if it happens. Also, users can use commands to disable or enable the COS function of certain DI lines. The COS function for each in default is disabled. Please refer to the following diagram to know the architecture of COS detection.


DI0-DI63
CN2A
DI0-DI31
DI32-DI63
CPLD0
CPLD1
INT
INT
PCI
Bridge
DI64-DI127
CN2B
DI64-DI95
DI96-DI127
CPLD2
CPLD3
INT
INT
PCI
BUS

5 C/C++ DOS Libraries

5.1 Programming Guide

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 conventions are:

```
_{hardware_model}_{action_name}. e.g.
     _7452_Initial().
```

All functions in the cPCI-7452 drivers are with 7452 as {hardware_model}.

Data Types

We have defined some data types in the Pci_9112.h (DOS) and Acl_pci.h (Windows 95) header files. These data types are used by the NuDAQ card library. We recommend you 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 long integer	-2147483648 to 2147483647
U32	32-bit unsigned long integer	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

5.2 _7452_Initial

@ Description

The cPCI-7452 card is initialized according to the card number. Because the cPCI-7452 is PCI bus architecture and meets the plug and play design, the IRQ and base_address (pass-through address) are assigned by the system BIOS directly. Every cPCI-7452 card must be initialized by this function before using other functions.

@ Syntax

```
U16 _7452_Initial (U16 *existCards, PCI_INFO
     *pciInfo)
```

@ Argument

existCards: The number of installed cPCI-7452 cards. The returned value shows how many cPCI-7452 cards are installed in your system.

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

@ Return Code

```
ERR_NoError,  ERR_PCIBiosNotExist,
     ERR_BoardNoInit, ERR_InvalidBoardNumber
```

@ Example

```
#include "7452.h"
PCI_INFO info_7452;

Main()
{

int ErrCode;
U16 existCards;
```

```
 ErrCode = _7452_Initial(&existCards,
      &info_7452);
       if(ErrCode != ERR_NoError){
         printf("PCI BIOS DOES NOT EXIST\n");
         exit(1);
         }
.
.
}
```

5.3 _7452_DI_Bankn, n = 0 - 3

@ Description

These functions are used to read data from digital input port. On the cPCI-7452, there are four banks, each of which has 32-bit digital inputs. You can get 128 input data by using these four functions.

@ Syntax

```
U16 _7452_DI_Bank0 (U16 boardID, U32 *diData)
U16 _7452_DI_Bank1 (U16 boardID, U32 *diData)
U16 _7452_DI_Bank2 (U16 boardID, U32 *diData)
U16 _7452_DI_Bank3 (U16 boardID, U32 *diData)
```

@ Argument

boardID: Board ID to the specific borad.

diData: return 32-bit value from each digital input port.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

```
#include "7452.h"
PCI_INFO info_7452;
U16   curCardNo_7452 = 0;

Main()
{

int ErrCode;
U16 existCards;
U32  InData0,InData1, InData2, InData3;

ErrCode = _7452_Initial(&existCards, &info_7452);
     /* Assume NoError when initialize cPCI-7452
     */

_7452_DI_Bank0(curCardNo_7452,InData0);
```

```
Printf("The value of DI[31..0] are 0x%lX \n",
      InData0);
_7452_DI_Bank1(curCardNo_7452,InData1);
Printf("The value of DI[63..32] are 0x%lX \n",
      InData1);
_7452_DI_Bank2(curCardNo_7452,InData2);
Printf("The value of DI[95..64] are 0x%lX \n",
      InData2);
_7452_DI_Bank3(curCardNo_7452,InData3);
Printf("The value of DI[127..96] are 0x%lX \n",
      InData3);
/* Bank0 - Bank3 are corresponding to DI[31..0],
      DI[63..32], DI[95..64], DI[127..96] */
.
}
```

5.4 _7452_DO_Bankn, n = 0 - 3

@ Description

These functions are used to write data to digital output ports which can be used to energize the external power coils. You can control all 128 channels by using _7452_DO_Bankn functions. Bit 1 represent the ON condition, and Bit 0 represent the OFF condition.

@ Syntax

```
U16 _7452_DO_Bank0 (U16 boardID, U32 doData)
U16 _7452_DO_Bank1 (U16 boardID, U32 doData)
U16 _7452_DO_Bank2 (U16 boardID, U32 doData)
U16 _7452_DO_Bank3 (U16 boardID, U32 doData)
```

@ Argument

boardID: Board ID to the specific borad.

doData: 32-bits value which will be written to each digital output port.

@ Return Code

```
ERR_NoError,  ERR_BoardNoInit
```

@ Example

```
#include "7452.h"
PCI_INFO info_7452;
U16   curCardNo_7452 = 0;

Main()
{
int ErrCode;
U16 existCards;
U32  OutData0, OutData1, OutData2, OutData3;

ErrCode = _7452_Initial(&existCards, &info_7452);
     /* Assume NoError when initialize cPCI-7452
     */

_7452_DO_Bank0(curCardNo_7452,OutData0);
```

```
Printf("The value of DO[31..0] are 0x%lX \n",
     OutData0);
_7452_DO_Bank1(curCardNo_7452,OutData1);
Printf("The value of DO[63..32] are 0x%lX \n",
     OutData1);
_7452_DO_Bank2(curCardNo_7452,OutData2);
Printf("The value of DO[95..64] are 0x%lX \n",
     OutData2);
_7452_DO_Bank3(curCardNo_7452,OutData3);
Printf("The value of DO[127..96] are 0x%lX \n",
     OutData3);
/* Bank0 - Bank3 are corresponding to DO[31..0],
     DO[63..32], DO[95..64], DO[127..96] */
.
}
```

5.5 _7452_DORBk_Bankn, n = 0 - 3

@ Description

These functions are used to read data back from each digital output port control by 7452_DO_Bankn function. There are 128-bit digital outputs on the cPCI-7452. You can get back all DO data by using this function. Bit 1 represent the ON condition, and Bit 0 represent the OFF condition

@ Syntax

```
U16 _7452_DORBK_Bank0 (U16 boardID, U32
     *DoReadBackData)
U16 _7452_DORBK_Bank1 (U16 boardID, U32
     *DoReadBackData)
U16 _7452_DORBK_Bank2 (U16 boardID, U32
     *DoReadBackData)
U16 _7452_DORBK_Bank3 (U16 boardID, U32
     *DoReadBackData)
```

@ Argument

boardID: Board ID to the specific borad.

DoReadBackData: value read back from each 32-bit digital output port.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

```
#include "7452.h"
PCI_INFO info_7452;
U16   curCardNo_7452 = 0;

Main()
{

int ErrCode;
U16 existCards;
U32  OutData0, OutData1, OutData2, OutData3;
U32  RBKData0, RBKData1, RBKData2, RBKData3;
```

```
ErrCode = _7452_Initial(&existCards, &info_7452);
      /* Assume NoError when initialize cPCI-7452
      */

_7452_DO_Bank0(curCardNo_7452,OutData0);
Printf("The value of DO[31..0] are 0x%lX \n",
      OutData0);
_7452_DORBK_Bank0(curCardNo_7452,BBKData0);
Printf("The value of DORBK[31..0] are 0x%lX \n",
      RBKData0);
/* Write data to DO[31..0] and read the setting
      back*/

_7452_DO_Bank1(curCardNo_7452,OutData1);
Printf("The value of DO[63..32] are 0x%lX \n",
      OutData1);
_7452_DORBK_Bank1(curCardNo_7452,RBKData1);
Printf("The value of DORBK[63..32] are 0x%lX \n",
      OutData1);
/* Write 32-bit data to DO[63..32] and read the
      setting back*/
.
}
```

5.6 _7452_COSETUP_Bankn, n = 0 - 3

@ Description

These functions are used to enable the COS channel for each port.

@ Syntax

```
U16  _7452_COSETUP_Bank0 (U16 boardID, U32
     COS_Enable_Data)
U16  _7452_COSETUP_Bank1 (U16 boardID, U32
     COS_Enable_Data)
U16  _7452_COSETUP_Bank2 (U16 boardID, U32
     COS_Enable_Data)
U16  _7452_COSETUP_Bank3 (U16 boardID, U32
     COS_Enable_Data)
```

@ Argument

boardID: Board ID to the specific borad.

COS_Enable_Data: COS channel enable. ‘1’ enable the corresponding channel and ‘0’ disable the corresponding channel.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

See Demo program. Demo Program ‘ COS_Interrup_Demo.C’

5.7 _7452_COSLatch_Bankn, n = 0 - 3

@ Description

These functions are used to latch digital input data for each port after COS interrupt occurs.

@ Syntax

```
U16  _7452_COSLatch_Bank0 (U16 boardID, U32
     *COS_Latch_Data)
U16  _7452_COSLatch_Bank1 (U16 boardID, U32
     *COS_Latch_Data)
U16  _7452_COSLatch_Bank2 (U16 boardID, U32
     *COS_Latch_Data)
U16  _7452_COSLatch_Bank3 (U16 boardID, U32
     *COS_Latch_Data)
```

@ Argument

boardID:Board ID to the specific borad.

COS_Latch_Data:Digital input data when COS occurs. This register will be erased when clearing IRQ.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

See Demo program. Demo Program ‘ COS_Interrup_Demo.C’

5.8 _7452_INT_Control

@ Description

This function is used to control the interrupt source of cPCI-7452. For more details about interrupt sources, refer to section 3.4

@ Syntax

```
U16  _7452_INT_Control (U16 boardID, U16
     COS3_Enable, U16 COS2_Enable, U16
     COS1_Enable, U16 COS_Enable,U16 WDT_Enable)
```

@ Argument

boardID: Board ID to the specific borad.

COS_Enable: Bank0 COS interrupt function enable/disable.

COS1_Enable: Bank1 COS interrupt function enable/disable.

COS2_Enable: Bank2 COS interrupt function enable/disable.

COS3_Enable: Bank3 COS interrupt function enable/disable.

WDT_Enable: Watch Dog Timer interrupt counter enable/disable

The possible combinations of interrupt source are shown in the following table.

WDT_ Enable	COS3_ Enable	COS2_Enable	COS1_Enable	COS_ Enable	IRQ source	IRQ trigger condition
0	0	0	0	0	Interrupt disable	--
0	0	0	0	1	Bank0 COS interrupt	Change of state in the enabled channel DI[31..0]
0	0	0	1	0	Bank1 COSd interrupt	Change of state in the enabled channel DI[63..32]
0	0	1	0	0	Bank2 COS interrupt	Change of state in the enabled channel DI[95..64]
0	1	0	0	0	Bank3 COS interrupt	Change of state in the enabled channel DI[127..96]

WDT_ Enable	COS3_ Enable	COS2_Enable	COS1_Enable	COS_ Enable	IRQ source	IRQ trigger condition
0	…	…	…	…	Any Arrange-ment and Combina-tion of COSx_Ena ble	
1	x	x	x	x	WDT inter-rupt enable	Rising edge of WDT carry out overflow

@ Return Code

```
ERR_NoError, ERR_BoardNoInit, ERR_INTNotSet
```

@ Example

See Demo program. Demo Program ‘ COS_Interrup_Demo.C’

5.9 _7452_CLR_IRQ

@ Description

This function is used to clear the interrupt request of cPCI-7452.

@ Syntax

```
U16  _7452_CLR_IRQ (U16 boardID, U16 COS_CLR3,
     U16 COS_CLR2, U16 COS_CLR1, U16 COS_CLR, U16
     WDT_CLR)
```

@ Argument

boardID: Board ID to the specific borad.

COS_CLR: Clear Bank0 COS interrupt request.

COS_CLR1: Clear Bank1 COS interrupt request.

COS_CLR2: Clear Bank2 COS interrupt request.

COS_CLR3: Clear Bank3 COS interrupt request.

WDT_CLR: Clear Watch Dog Timer interrupt request.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

See Demo program. Demo Program ‘ COS_Interrup_Demo.C’

5.10 _7452_GET_IRQ_Status

@ Description

This function is used to get the interrupt status of cPCI-7452.

@ Syntax

```
U16  _7452_GET_IRQ_Status (U16 boardID, U16
     *WDT_Status, U16 *COS3_Status, U16
     *COS2_Status, U16 *COS1_Status, U16
     *COS_Status)
```

@ Argument

boardID: Board ID to the specific borad.

COS_Status: Bank0 COS interrupt status.

- '1' represents interrupt asserts.
- '0' represents interrupt de-asserts.

COS1_Status: Bank1 COS interrupt status.

- '1' represents interrupt asserts.
- '0' represents interrupt de-asserts.

COS2_Status: Bank2 COS interrupt status.

- '1' represents interrupt asserts.
- '0' represents interrupt de-asserts.

COS3_Status: Bank3 COS interrupt status.

- '1' represents interrupt asserts.
- '0' represents interrupt de-asserts.

WDT_Status: Watch Dog Timer interrupt status.

- '1' represents interrupt asserts.
- '0' represents interrupt de-asserts.

@ Return Code

```
ERR_NoError, ERR_BoardNoInit
```

@ Example

See Demo program. Demo Program ‘ COS_Interrup_Demo.C’ and ‘WDT_Interrupt_Dmo.C’

5.11 _7452_WDTimerSReload_Config

@ Description

This function is used to clear the Watch-Dog-Timer Counter zero state of cPCI-7452 and reload the count value.

@ Syntax

```
U16  _7452_ WDTimerSReload_Config (U16 boardID,
     Int WDT_CountData)
```

@ Argument

boardID: Board ID to the specific borad.

WDT_CountData: decimal value from 1 (s) to 4500 (s) which will be written to WDT reload value.

@ Return Code

ERR_NoError, ERR_BoardNoInit

@ Example

See Demo program. Demo Program ‘WDT_Interrupt_Dmo.C’

Warranty Policy

Thank you for choosing ADLINK. To understand your rights and enjoy all the after-sales services we offer, please read the following carefully.

1. Before using ADLINK's products please read the user manual and follow the instructions exactly. When sending in damaged products for repair, please attach an RMA application form which can be downloaded from: http://rma.adlinktech.com/policy/.
2. All ADLINK products come with a limited two-year warranty, one year for products bought in China:
   - The warranty period starts on the day the product is shipped from ADLINK's factory.
   - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers' warranty.
   - For products containing storage devices (hard drives, flash cards, etc.), please back up your data before sending them for repair. ADLINK is not responsible for any loss of data.
   - Please ensure the use of properly licensed software with our systems. ADLINK does not condone the use of pirated software and will not service systems using such software. ADLINK will not be held legally responsible for products shipped with unlicensed software installed by the user.
   - For general repairs, please do not include peripheral accessories. If peripherals need to be included, be certain to specify which items you sent on the RMA Request & Confirmation Form. ADLINK is not responsible for items not listed on the RMA Request & Confirmation Form.

3. Our repair service is not covered by ADLINK's guarantee in the following situations:
   - Damage caused by not following instructions in the User's Manual.
   - Damage caused by carelessness on the user's part during product transportation.
   - Damage caused by fire, earthquakes, floods, lightening, pollution, other acts of God, and/or incorrect usage of voltage transformers.
   - Damage caused by unsuitable storage environments (i.e. high temperatures, high humidity, or volatile chemicals).
   - Damage caused by leakage of battery fluid during or after change of batteries by customer/user.
   - Damage from improper repair by unauthorized ADLINK technicians.
   - Products with altered and/or damaged serial numbers are not entitled to our service.
   - This warranty is not transferable or extendible.
   - Other categories not protected under our warranty.
4. Customers are responsible for shipping costs to transport damaged products to our company or sales office.
5. To ensure the speed and quality of product repair, please download an RMA application form from our company website: http://rma.adlinktech.com/policy. Damaged products with attached RMA forms receive priority.

If you have any further questions, please email our FAE staff: service@adlinktech.com.